NEAR FIELD COMMUNICATION

NEAR FIELD COMMUNICATION

FROM THEORY TO PRACTICE

Vedat Coskun, Kerem Ok and Busra Ozdenizci
NFC Lab – Istanbul, ISIK University, Turkey

WILEY
A John Wiley & Sons, Ltd., Publication

This edition first published 2012

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

Library of Congress Cataloging-in-Publication Data

Coskun, Vedat.
Near field communication : from theory to practice / Vedat Coskun, Kerem Ok, and Busra Ozdenizci.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-119-97109-2 (cloth)
1. Near field communication. I. Ok, Kerem. II. Ozdenizci, Busra. III. Title.
TK6570.N43C67 2012
621.384–dc23

2011033663

A catalogue record for this book is available from the British Library.

ISBN: 9781119971092

Typeset in 10/12pt Times by Aptara Inc., New Delhi, India

Vedat Coskun:
To lovely members of my family;
Mehmet & Fatma
Filiz & Ozgur & Arda
Mujdat & Kilinc & Muge & Selma.

Kerem Ok:
To my family who supported me all the time

Busra Ozdenizci:
To my dear parents and my brother Ozan

Contents

Preface

Near Field Communication (NFC) is a currently emerging and yet promising area which will have an enormous impact on the financial ecosystem as well as mobile technology throughout the world within just a few years. Mobile phone manufacturers, mobile network operators (MNOs), financial institutions such as banks, and information technology firms are performing R&D activities just to increase their share of the pie as much as possible. NFC, being a short range wireless communication technology that potentially facilitates mobile phone usage of billions of people throughout the world offers an enormous number of use cases including credit cards, debit cards, loyalty cards, car keys, access keys for hotels, offices and houses, eventually integrating all such materials into one single mobile phone. A solid and complete understanding of NFC requires knowledge in four areas: NFC technology; NFC security and privacy; NFC application development; and NFC business ecosystem. All of these issues are covered in this book which aims to present knowledge from theory to practice; including short use cases, case studies, application development examples, and finally existing featured trials from all over the world. This book provides information on NFC technology that appeals to the needs of almost all users interested in NFC technology and its ecosystem.

NFC Lab - Istanbul (www.NFCLab.com)

This book is the collective effort of NFC Lab - Istanbul researchers, which is an integral part of Işık University, Istanbul.

NFC Lab - Istanbul considers NFC as an emerging technology that transforms innovative ideas into reality for the future information and communication society.

NFC Lab - Istanbul strives for research excellence in focused research areas relevant to NFC. The Lab aims to collaborate with MNOs, financial institutions, government agencies, research institutes, trusted third parties, and other service providers to facilitate widespread usage of NFC applications.

The Lab is committed to work on NFC technology with a multidisciplinary network of expertise all around the world. The core team is accountable for creating and maintaining business and academic partnerships and dynamically generates networks on a project basis.

Our Motivation to Write This Book

We, the members of NFC Lab - Istanbul, have performed many academic and industrial tasks over the last few years. As we required information, we had to use web sources, white papers,

academic papers, works of the standardization bodies and so on. Many times the information we retrieved from several sources conflicted. Indeed, sometimes the information we obtained was inaccurate. Also, the sources involved different approaches which could not be compared easily.

Despite the fact that NFC is an emerging area, the volume of new articles being generated is increasing on a daily basis, and a satisfactory amount of information can be found on the Internet. However, the task of gathering the information and then retrieving the required knowledge is tiresome and time consuming.

When a practitioner with some expertise of programming in Java decides to access this new area, it is necessary to find the required sources from different sources. This will not be trivial, because in order to build NFC applications using Java language, the practitioner needs to collect scattered information, and then merge it for a better understanding. Even in this case, the user would be able to collect a small amount of information.

Some basic information exists in the public domain, and much more exists in academic literature, which either is not publicly available, or not easy to combine with the public information by non-academics.

Although some basic information exists in the current literature, other information is still not available. For example, we have performed extensive ecosystem analysis in this work.

As a result, we recognized the lack of and the need for a solid source that contains accurate information and addresses all of those involved in NFC technology and the NFC business ecosystem with a standard content.

As a matter of fact, we were the right team for NFC. We were pleased when we produced some tangible products such as conference papers, journal papers, applications, scientific projects, commercial projects, courses and so on. We then thought that it might be beneficial to share our knowledge with others. But how could we do this? The solution was to write a book containing the collective knowledge of the members.

Positive Discrimination

For the sake of simplicity on using pronouns and positive discrimination, we have ignored male pronouns throughout this book. Hence, we have used only *she*, *her*, and *herself* and chosen to ignore he, him, his, and himself.

NFC: Is it a Big Success or Another Failure Instead?

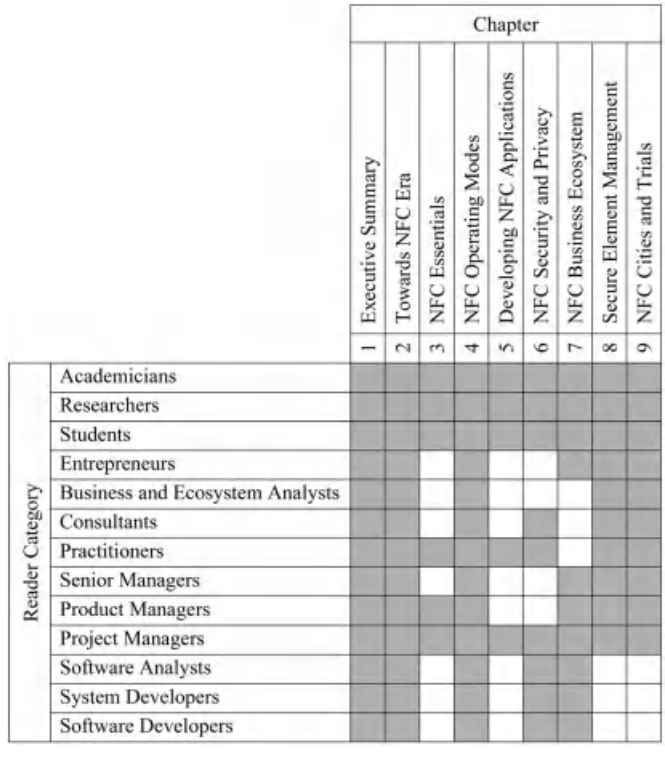
History is indeed full of technological and ecosystem failures. The satellite phone network is one example. Only a few years ago, NFC was unknown. In a short period, NFC has been introduced to great enthusiasm by several organizations including governmental departments, research centers, and companies.

NFC usage was expected to boom over the last few years but this seems to have been postponed repeatedly until now. This has made many people from the industry suspicious about the potential success of NFC.

As a matter of fact, there are two major stumbling blocks to the success of NFC. The first is that of technological sufficiency, and the second is the ecosystem agreement by the interested parties. These issues are indeed very much related. As the involved parties become more convinced about the technical success of the new model, they tend to invest more resources for development, and as new technical improvements occur, the ecosystem becomes more established and ready for the boom. Also, when a party puts more investment into a project, it seems more eager to make an agreement with the other parties involved, in order to get their money back and hence to provide better Return of Investment (RoI).

Audience

This book is aimed at academicians, researchers, students, entrepreneurs, business and ecosystem analysts, consultants, practitioners, senior managers, product managers, project managers, software analysts, system developers and software developers who intend to invest in NFC, or at least want to have a broad knowledge of the area. The audience for each chapter is given below.

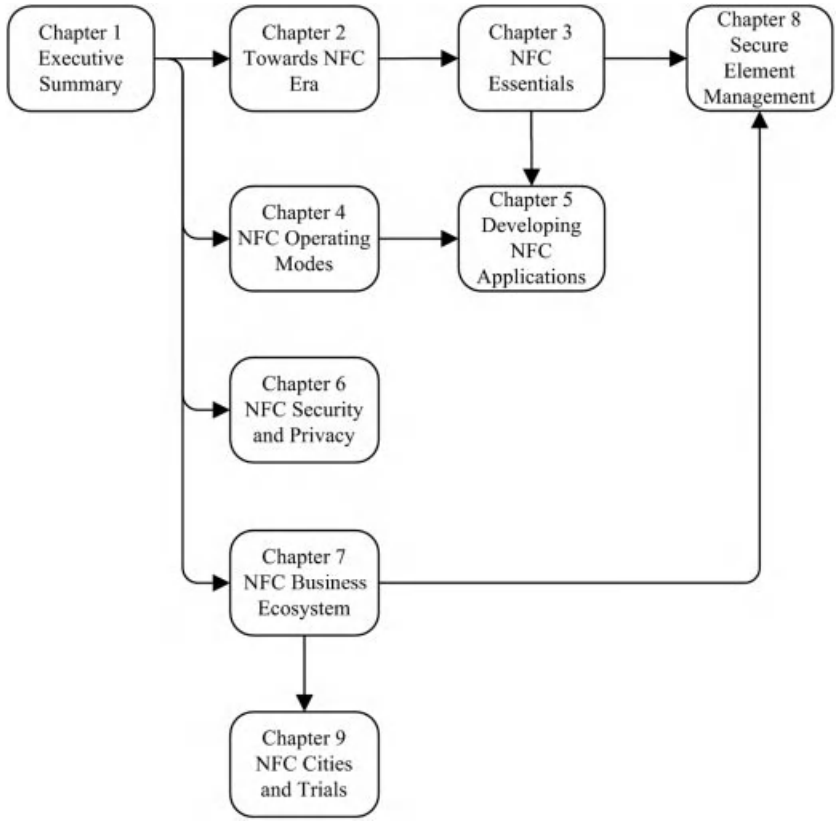


Chapter Dependencies

1. *Executive Summary:* This chapter does not require any prerequisite knowledge.
2. *Towards NFC Era:* Although this chapter does not require much background knowledge, we advise readers to read Chapter 1 to obtain an overall understanding of NFC technology. Readers of this chapter will learn the foundations of NFC technology and its evolution in detail.
3. *NFC Essentials*: We advise readers of this chapter to read Chapter 1 to gain general knowledge on NFC, and Chapter 2 to learn the foundation technologies of NFC in detail, so that the readers will appreciate the technical and communication essentials of NFC operating modes covered in this chapter.
4. *NFC Operating Modes:* Although this chapter does not require much background knowledge, we advise readers to read Chapter 1 to obtain an overall understanding of NFC technology. Readers of this chapter will learn about each operating mode in detail. They will be able to design NFC applications in each operating mode and also know the benefits of the designed application to the user.
5. *Developing NFC Applications:* This chapter requires general programming skills, as well as basic knowledge of Java technology. The readers should also have knowledge of the NFC operating modes described in Chapter 4 and some of the NFC technical essentials described in Chapter 3. Readers of this chapter will gain fundamental knowledge in NFC programming.
6. *NFC Security and Privacy:* Although this chapter does not require much background knowledge, we advise readers to read Chapter 1 to obtain an overall understanding of NFC technology. Readers of this chapter will learn about the security and privacy issues of NFC technology.
7. *NFC Business Ecosystem:* Although this chapter does not require much background knowledge, we advise readers to read Chapter 1 to obtain an overall understanding of NFC technology. Readers of this chapter will learn about the nature of a NFC business ecosystem from a holistic perspective.
8. *Secure Element Management:* We advise readers to read Chapter 1 to obtain an overall understanding of NFC technology, Chapter 3 to learn about the secure element and its infrastructure within NFC mobiles and Chapter 7 to appreciate the importance of the secure element and Over-the-Air (OTA) concepts in NFC business models.
9. *NFC Cities and Trials*: Although this chapter does not require much background knowledge, we advise readers to read Chapter 1 to obtain an overall understanding of NFC technology. Moreover, executive audiences of this chapter should also read Chapter 7. Readers of this chapter will learn about conducted projects, trials and implementations from all over the world and will analyze different NFC services from a holistic perspective. This will help readers to gain important knowledge about NFC service implementations around the world.

Organization

This book consists of nine chapters. They are arranged to make it easy for the reader to be able to select the material relevant to them. Potential readers of each chapter are specified in the chapter contents below.



Chapter 1: Executive Summary

This chapter consists of an executive and managerial summary on the NFC ecosystem and technology. The readers of this chapter will be able to obtain a high level introductory knowledge without being exposed to any unnecessary technical details. The readers will get enough information to understand what NFC is in almost all scenarios. People who intend to cover NFC system development, NFC application development, NFC service providing and so on will need to read related chapters additionally in order to obtain a detailed knowledge. The audience will be comfortable reading the other chapters after being exposed to NFC in this chapter.

Executive audiences of this chapter can additionally read Chapter 7 and Chapter 9. Readers who are interested in more technical aspects of NFC should read Chapters 2–4, 6 and 8 for further technical understanding. Potential members of NFC application development teams are encouraged to read Chapter 5 as well.

Chapter 2: Towards NFC Era

This chapter consists of basic knowledge on ubiquitous computing, mobile phones, radio frequency identification, smart cards, and NFC technologies. Although this chapter does not

require much background knowledge, we advise readers to read Chapter 1 to obtain an overall knowledge of NFC technology for a better understanding. This chapter addresses readers of all categories.

Readers of this chapter will be able to learn and appreciate the NFC concept and also understand other chapters more easily. Executive audiences of this chapter are encouraged to also read Chapters 7 and 9. Readers who are interested in the more technical aspects of NFC should read Chapters 3, 4, 6 and 8 for further technical understanding. Chapter 5 should be read if they intend to be part of an NFC application development team.

Chapter 3: NFC Essentials

This chapter contains vast amount of information on NFC technology in order to give a solid and complete understanding. This includes sufficient knowledge on NFC devices and their capabilities, NFC enabled mobile phones and their architectural details, and standardization bodies in the development of NFC enabled mobile phones, radio frequency layer features of NFC and also technical essentials and communication architectures of each operating mode. Potential readers of this chapter are academicians, researchers, students, practitioners, product managers, and system developers.

Readers of this chapter will learn about each NFC operating mode's communication architecture and other technical essentials in detail. This knowledge will help the reader to understand further concepts in NFC technology better and also to develop NFC based systems.

Chapter 4: NFC Operating Modes

This chapter mainly consists of NFC operating modes including descriptions, essentials, use cases, generic usage models, and benefits of each operating mode. This chapter addresses readers of all categories.

Readers of this chapter will learn about each operating mode in detail. They will be able to design NFC applications in each operating mode and also know the benefits of the designed application to the user. Our readers may read Chapter 5 which covers NFC application programming in each operating mode. Readers who are interested in the more technical aspects of NFC should read Chapters 6 and 8 for further technical understanding.

Chapter 5: Developing NFC Applications

This chapter presents all the necessary information to develop NFC applications. General information regarding NFC project design and application team management is initially provided. On account of the number of affirmative features, availability of application programming interfaces and high usage experience in the market, Java is used as the development technology. Potential readers of this chapter are academicians, researchers, students, practitioners, project managers, software analysts, system developers, and software developers.

Readers of this chapter will gain fundamental knowledge in NFC programming and the learned NFC programming skills from Java technology will help users to develop NFC based applications in other languages and development platforms more easily. Our readers are

encouraged to also read Chapters 7 and 9. Readers interested in obtaining further technical understanding should read Chapters 6 and 8.

Chapter 6: NFC Security and Privacy

This chapter presents introductory knowledge on security; vulnerability, threat, attack, and risk; cryptography; NFC security issues and preventive mechanisms to handle the risks. The content is provided for both those in higher, or managerial and lower, or technical levels. Potential readers are academicians, researchers, students, consultants, practitioners, project managers, software analysts, system developers, and software developers.

Readers of this chapter will learn about the security and privacy issues concerning NFC technology in detail. Hence they will be able to design secure and interoperable NFC based systems and services in each operating mode. Readers who are interested in obtaining further technical understanding should read Chapter 8.

Chapter 7: NFC Business Ecosystem

This chapter consists of the NFC business ecosystem including comprehensive analysis of the ecosystem, generic roles, viewpoints of different associations, business model alternatives with driving factors, and a case study. Potential readers of this chapter are academicians, researchers, students, entrepreneurs, business and ecosystem analysts, consultants, practitioners, senior managers, product managers, and project managers.

Readers of this chapter will learn the nature of the NFC business ecosystem and understand participating stakeholders' business requirements as well. They will be able to evaluate and design NFC services with sustainable business models. Readers may also read Chapter 9 which gives real NFC trials throughout the world in diverse business environments.

Chapter 8: Secure Element Management

This chapter mainly consists of NFC technology including essentials of OTA technology, an overview on GlobalPlatform specifications, the Universal Integrated Circuit Card (UICC) based secure element and its security domain architecture, different UICC based secure element management models, multiple secure element management within a single NFC mobile and an alternative OTA management model. Potential readers of this chapter are academicians, researchers, students, practitioners, product managers, project managers, and system developers.

Readers of this chapter will gather a broad knowledge of OTA technology; UICC based secure elements and their life-cycle management via OTA technology, and also an alternative OTA management model.

Chapter 9: NFC Cities and Trials

This chapter consists of NFC cities, launched NFC trials and projects that have been developed all over the world so far. Potential readers of this chapter are academicians, researchers,

students, entrepreneurs, business and ecosystem analysts, consultants, practitioners, senior managers, product managers, and project managers.

The chapter starts by describing NFC cities which are the most popular implementations of NFC technology. The purpose of an NFC city may be either to test implementations or even to actually use them in a defined arena. NFC cities mostly test the social aspect of the NFC technology when compared with trials and projects. The usability issues together with the problems residing in the technology are easily obtained continuously through tests in NFC cities. In the case of NFC trials and projects, the business aspect of an application or an NFC ecosystem is tested more than the social aspect.

Call for Feedback and Help

This is one of the pioneer materials on NFC technology providing such an extensive content. One of the challenges that we have faced is that improvements are continuously being introduced, almost exponentially. Hence, the book will already need to be updated within a few years. We intend to update this book by improving it frequently. Our motivation here is to put this book at the center of the NFC ecosystem in the world. In addition to the work we intend to carry out, we would appreciate receiving any feedback, additional supportive material, or information that might help improve future editions for potential readers.

We would appreciate receiving comments such as:

- Any mistakes that we have inserted, or possibly ignoring some facts;
- Any missing, yet important points that the user considers essential;
- Any missing, yet important contribution to the NFC technology and ecosystem that we have ignored;
- Proposals for new chapters or additional material that should be added in future edition(s).

For more information, please visit the companion website -www.wiley.com/go/coskun

Any such e-mails should be directed to: info@nfc-book.com. Any other material can be sent to:

Vedat Coskun
ISIK University
Sile 34980 Istanbul
Turkey

Acknowledgments

We are grateful for the support of ISIK University to NFC Lab - Istanbul; especially Prof. Dr. Siddik Binboga Yarman, Chairman, Board of Trustees; Prof. Dr. Nafiye Gunec Kiyak, Rector; Aziz Genc, Secretary General. We are also grateful to Prof. Dr. Erdal Cayirci, who has given support for this book.

List of Acronyms

3DES	Triple DES
3G	Third Generation
3GPP	3rd Generation Partnership Project
AC	Alternating Current
AES	Advanced Encryption Standard
AMS	Application Management Software
ANSI	American National Standards Institute
APDU	Application Protocol Data Unit
API	Application Programming Interface
APSD	Application Provider Security Domain
ASK	Amplitude Shift Keying
BPSK	Binary Phase Shift Keying
BS	Base Station
CA	Certificate Authority
CASD	Controlling Authority Security Domain
CDC	Connected Device Configuration
CIB	Card Issuing Bank
CLDC	Connected Limited Device Configuration
CLF	Contactless Front-end
CSMA	Carrier Sense Multiple Access
DC	Direct Current
DES	Data Encryption Standard
DoS	Denial of Service
DSP	Digital Signal Processor
EAN	European Article Number
ECC	Elliptic Curve Cryptography
ECMA	European Computer Manufacturer Association
EDGE	Enhanced Data for GSM Evolution
EMV	Europay, Mastercard and Visa
EPC™	Electronic Product Code
ETSI	European Telecommunications Standards Institute
FIPS	Federal Information Processing Standard
GPRS	General Packet Radio System
GPS	Global Positioning System

GSM	Global System for Mobile Communications
GSMA	GSM Association
GUI	Graphical User Interface
HCP	Host Controller Protocol
HCI	Host Controller Interface
HDLC	High-Level Data Link Control
HMAC	Hash-based Message Authentication Code
HSPA	High Speed Packet Access
I/O	Input/Output
IC	Integrated Circuit
ICAO	International Civil Aviation Organization
ICT	Information and Communications Technologies
IDPS	Intrusion Detection and Prevention System
IEC	International Electrotechnical Commission
IFF	Identify Friend or Foe
ISD	Issuer Security Domain
ISO	International Organization of Standardization
ITU	International Telecommunication Union
J2EE	Java™ 2 Enterprise Edition
J2ME	Java™ 2 Micro Edition
J2SE	Java™ 2 Standard Edition
JAD	Java Application Descriptor
JAR	Java Archive
JCP	Java Community Process
JCVM	JavaCard Virtual Machine
JCRMI	JavaCard Remote Method Invocation
JIS	Japanese Industrial Standard
JLS	Java Language Specification
JSP	Java Community Process
JSR	Java Specification Requests
JVM	Java Virtual Machine
KDF	Key Derivation Function
KMA	Key Management Authority
KVM	Kilobyte Virtual Machine
LLCP	Logical Link Control Protocol
MAC	Message Authentication Code
MIDP	Mobile Information Device Profile
MIM	Man in the Middle
MMS	Multimedia Messaging Service
MNO	Mobile Network Operator
MULTOS	Multi-application Operating System
MVNO	Mobile Virtual Network Operator
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
NFCIP	Near Field Communication Interface and Protocol

NFCIP-1	Near Field Communication Interface and Protocol-1
NFCIP-2	Near Field Communication Interface and Protocol-2
OMA	Open Mobile Alliance
OS	Operating System
OTA	Over-the-Air
PC	Personal Computer
PCD	Proximity Coupling Device
PDA	Personal Digital Assistant
PICC	Proximity Integrated Circuit Card
PIN	Personal Identification Number
PKI	Public Key Infrastructure
POS	Point of Sale
PSK	Phase Shift Keying
QoS	Quality of Service
RF	Radio Frequency
RFID	Radio Frequency Identification
RoI	Return of Investment
RST	Reset the microprocessor
RTD	Record Type Definition
SAT	SIM Application Toolkit
SCOS	Smart Card Operating System
SCP	Smart Card Platform
SCP	Secure Channel Protocol
SDK	Software Development Kit
SE	Secure Element
SEP	Secure Exchange Protocol
SIM	Subscriber Identity Module
SMC	Secure Memory Card
SMS	Short Messaging Service
SSD	Supplementary Security Domain
SSL	Secure Sockets Layer
STEP	Secure Trusted Environment Provisioning
SWP	Single Wire Protocol
TLS	Transport Layer Security
TNF	Type Name Format
TSM	Trusted Service Manager
TTP	Trusted Third Party
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunication System
UPC	Universal Product Code
URI	Uniform Resource Identifier
USIM	Universal Subscriber Identity Module
VM	Virtual Machine
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy

WI	Wired Interface
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Networks

1

Executive Summary

Near Field Communication (NFC) is a new technology and ecosystem that has emerged in the last decade. NFC technology is a short range, high frequency, low bandwidth and wireless communication technology between two NFC enabled devices. Communication between NFC devices occurs at 13.56 MHz high frequency which was originally used by Radio Frequency Identification (RFID). Although RFID is capable of reception and transmission beyond a few meters, NFC is restricted to within very close proximity. Currently, integration of NFC technology into mobile phones is considered as the most practical solution because almost everyone carries one.

NFC technology enables communication between an NFC enabled mobile phone at one end, and another NFC enabled mobile phone, an NFC reader or an NFC tag at the other end. Potential NFC applications and services making use of NFC technology include e-payment, e-ticketing, loyalty services, identification, access control, content distribution, smart advertising, data/money transfer and social services. Due to its applicability to a wide range of areas and the promising value added opportunities, it has attracted many academicians, researchers, organizations, and commercial companies.

The changes or improvements on RFID to expose NFC technology can be described as:

- Short range communication, where RFID may use long range especially for active tags that contain embedded energy.
- Passive tag usage only (actually occurs only in reader/writer mode) whereas both active and passive tags are possible in RFID.
- Inherent secure data exchange because of short range communication.
- Implicit matching of pairs that express their willingness to perform NFC communication by bringing themselves close to each other.
- Interest from companies to integrate many services such as payment with debit and credit cards, loyalty, identification, access control and so on, because of the secure communication and implicit matching as described in the previous item.

Technology usage is now in the pilot phase in many countries. Usability issues and technology adoption are being explored by many academicians and industrial organizations. Many mobile

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.
 Published 2012 by John Wiley & Sons, Ltd.

phone manufacturers have already put their NFC enabled mobile phones into the market. As NFC enabled mobile phones spread and commercial services are launched, people will be able to pay for goods and services, access hotel rooms or apartments, update their information in social networks, upload their health data to hospital monitoring systems from their homes, and benefit from many more services by using their NFC enabled phones.

The success of NFC technology is bound to advances in other fields as well. Over-the-Air (OTA) technology among ecosystem actors is definitely a prerequisite to operate NFC systems satisfactorily. Secure Element (SE) is also a requirement to store valuable digital information and to provide concurrent execution of multiple NFC services on the same smart card securely. Dependence on other technologies is one of the challenges that NFC currently faces now.

Another important challenge is about the potential stakeholders in the NFC ecosystem. NFC has a complex and dynamic environment with high number of participating organizations. They have already recognized the possible added values, and each party is trying to maximize the value of their stake. The ownership and management of the SE is a dominant factor in getting a greater share, because each transaction has to use some applications installed on the SE, and the owner can always demand a higher share. Currently Mobile Network Operators (MNOs) own and issue the UICC as SE on mobile phones, and alternative SE ownerships are being negotiated among MNOs financial organizations, and even smart card manufacturers.

Please note that this chapter is an executive summary of the book and hence references are provided at the end of the related chapters.

1.1 Towards NFC Era

NFC is a technology that simplifies and secures the interaction with the automation ubiquitously around us. The NFC concept is designed from the synergy of several technologies including wireless communications, mobile devices, mobile applications and smart cards. Server side programming, web services, and XML technologies also contribute fast improvement and the spread of NFC technology. Many daily applications, such as credit cards, car keys, and hotel room access cards will presumably cease to exist because an NFC enabled mobile phone will suffice to provide all of their functionalities.

Currently, NFC is one of the enablers for ubiquitous computing. Therefore the origin of the idea is closely related to ubiquitous computing. In order to understand the relation of NFC and ubiquitous computing, we need to start with the history of ubiquitous computing.

1.1.1 Ubiquitous Computing

The essence of modern computers is automated calculation and programmability. The history of modern computers includes the work of pioneers over almost two hundred years. Personal Computers (PCs) are an important step after early computers, changing the way that a user interacts with computers by using keyboards and monitors for input and output instead of punch cards, cables and so on. The mouse has also changed the way humans and computers interact because it enabled users to input spatial data to a computer. The hand became accustomed to holding the mouse, and the pointing finger became accustomed to clicking it. The movements of the pointing device are echoed on the screen by the movements of the cursor, creating a simple and intuitive way to navigate a computer's Graphical User Interface (GUI).

Touch screens changed the form of interaction dramatically. They removed the need for earlier input devices, and the interaction was performed by directly touching the screen, the new input device. In the meantime, mobile phones had been introduced, initially for voice communication. Early forms of mobiles contained a keypad. Mobile phones with touch screens can be assumed to be the state of the art technology as the same screen is used as both the input and output unit, allowing the user to act more intuitively.

Ubiquitous computing is the highest level of interaction between humans and computers, where computing devices are completely integrated into everyday life and the objects around, and are simple to use. Ubiquitous computing is a model in which humans do not design their activities according to the machines they need to use; instead, the machines adjust to human needs. Eventually, the primary aim is that humans using machines will not need to change their normal behaviors and also will not even notice that they are performing activities with the help of machines.

1.1.2 Mobile Phones

A mobile phone is an electronic device which is primarily used to make voice calls while the user is mobile. The user of the mobile phone must be registered to a mobile phone network where the service is provided by a MNO. The call can be made to or received from any other phone which is a member of either the same or another mobile phone network, a fixed line network, or even an internet based network. Mobile phones support the anytime, anywhere motto. Mobile phones are also referred to as mobiles.

Mobile phones are very convenient to use and handy. Therefore in addition to the voice call capability, a vast amount of additional services are bundled to it, and many new future services are still on the way, such as NFC technology. Currently supported mobile phone communication services can be viewed based on whether they are wired or wireless services. Mobile phones also include a vast amount of integrated services.

USB and PC synchronization are the most significant wired services. The phones are connected to the computers to enable data transfer, synchronization and so on.

The amount of wireless services, on the other hand, is much greater. GSM communication is obviously the primary service that a mobile phone provides. As a matter of fact it was the one that the pioneer phones provided. Later, Short Messaging Service (SMS) was introduced. Multimedia Messaging Service (MMS) could be enabled only after high data transfer rates between the base stations and the mobile phones. Moreover, users can experience mobile radio and television services with mobile phones. Localization services, specifically Global Positioning System (GPS), allowed phones to enable applications such as navigation and social media interaction. One added communication capability to mobile phones is via several peer-to-peer services such as Infrared, Bluetooth, and finally NFC. Infrared requires line of sight, NFC requires very close interaction, namely touching, and Bluetooth requires communication within small distance. Wi-Fi connectivity allowed mobile phones to access the Internet with low bandwidth. Electronic mail (e-mail) enabled users to access their inboxes or send e-mails while mobile.

Storing contact and communication details are the most important integrated services, since they simplify Global System for Mobile Communications (GSM). There are other integrated services that are not related to GSM, at least not directly. Instead, the main objective of those

services is to eliminate additional devices and integrate all into one device. Calculator is one primitive function and eliminates the physical need for a calculator. Gaming is the one that most people like to have in their mobiles. Taking photos and videos, and even editing them using additional applications are simple to use so that additional cameras or video recorders are not required. Music and video playback are two other attractive facilities. Moreover, clock and alarm capability has removed the need for watches.

Some major wireless services currently enabled by mobile phones are GPS Navigation, Wireless Internet services, GSM, Bluetooth, Wi-Fi, and NFC technologies.

1.1.3 Technological Motivation of NFC

The main motivation for NFC is the integration of personal and private information such as credit card or debit card data into mobile phones. Therefore, security is the most important concern, and the wireless communication range provided even by RFID technology is considered too long. Mechanisms such as shielding are necessary to prevent unauthorized people from eavesdropping on private information because even non-powered, passive tags can be read over 10 m. This is where NFC comes in.

1.1.4 Wireless Communication, RFID, and NFC

Wireless communication refers to data transfer without using any cables. When communication is impossible and impractical for cable usage, wireless communication is the solution. The communication range may vary from a few centimeters to many kilometers. Wireless is generally mobile, and mobile is essentially wireless. We also distinguish nomadic communication from mobile. Devices that allow nomadic communication may perform either wireless or wired communication at a given time. An example of nomadic communication is the laptop.

The direct consequence of wireless communication is mobility. Mobility allowed people to be flexible, since they can be reached anywhere. It is obvious that mobility increased productivity, since mobile communication enabled people to be reached at anytime and anywhere. This has a big impact on our daily lives. People become reachable not only for commercial purposes, but also for social reasons. The currently available mobile communication services that support mobility are GSM, Bluetooth, Wi-Fi, WiMAX, and ZigBee.

1.2 Evolution of NFC

NFC can be taught as an extension to RFID that also uses smart card technologies' interfaces. To understand NFC technology, we need to have a brief knowledge of the forerunners of NFC technology: the barcode as an earlier form of RFID technology, RFID, and the magnetic stripe card as an earlier form of smart cards and smart card technologies.

1.2.1 Earlier Form of RFID: Barcode Technology

A barcode is a visual representation of data of the object to which it is attached. The information on the barcodes is scanned by barcode readers and transferred to the computing devices that are connected to the readers. Then the device processes the information.

Early barcodes represent data by varying the widths and spacing of parallel lines, and are referred to as linear or one-dimensional (1D). Due to the used space, minimum thickness of each bar and orientation restrictions, the maximum addressable 1D code is not high. In contrary to limited number of available 1D codes, later two-dimensional (2D) barcodes evolved which have a larger data storage capacity. As an example, 2D barcodes on a medicine box may contain specific identification information for that medicine box, so each specific patient and each specific medicine can be tracked.

Some major examples of linear barcodes are UPC (Universal Product Code) and EAN13 (European Article Number) Barcodes. Also QR (Quick Response) Code Barcode is an example of a 2D barcode.

1.2.2 RFID Technology

RFID is a technology that uses communication via radio waves to exchange data between an RFID reader and an electronic RFID tag (label), traditionally attached to an object, mostly for the purpose of identification and tracking. The data transmission results from electromagnetic waves, which can have different ranges depending on the frequency and the magnetic field.

RFID tags are small integrated circuits which can hold small applications as well as tiny amount of data. There are two types of RFID tags; passive tags and active tags. Passive tags have no internal power supply, have an IC (Integrated Circuit) and antenna embedded in them. They are powered by the incoming signal from a Radio Frequency (RF) field. Passive tags have practical read distances ranging from about 10 cm up to a few meters depending on the chosen RF and antenna design and size. Unlike passive RFID tags, active RFID tags have their own internal power source which is used to power any ICs that generate the outgoing signal. Active tags are typically much more reliable than passive tags due to the ability to conduct a session with a reader at longer distances. The major drawback of RFID tags when compared with paper barcodes is their higher price.

Both barcodes and RFID tags can be copied, however in different ways. Barcodes can be distributed electronically which enables printing and displaying on a digital device such as a PC or a mobile phone. You can e-mail a barcode image to a vast number of people and all of the receivers can print the barcode onto ordinary paper immediately. RFID tag content can be electronically spread as well. However, a digital chip is required for each copy instead of paper. When compared with barcodes, producing the original as well as copies is more expensive. The RFID tag has large data capacity, and each individual tag has a unique code which is similar to 2D barcodes. The uniqueness of RFID tags provides a product that can be tracked as it moves from one location to another.

RFID was a relatively early technology, and many RFID applications have been developed so far. Some of those applications are as follows:

- *Inventory control:* Most RFID applications are for managing assets. Retail stores use RFID tags on their items to control purchase, decrease in inventory and so on.
- *Toll roads:* Active RFID tags are fixed on vehicles so that during the vehicle's journey, the toll cost can easily be deducted from the owner's account.

- *Public transportation:* Many cities use RFID enabled payment systems on public transportation to make payment easier.
- *Passports:* It has become an ordinary process to insert RFID tags into passports to prevent counterfeiting them. Information such as owner's photo, fingerprint, address, some private data and so on are embedded into the tag, so that modification and illegal usage is harder than using printed material alone.

1.2.3 Earlier Form of Smart Cards: Magnetic Stripe Cards

A magnetic stripe card is one that contains a digital storage space where the data are loaded during the manufacturing phase. The stripe is made up of tiny magnetic particles in a resin. It is traditionally a read-only item. It is read by physical contact by swiping the card past a device with a magnetic reading head. Currently, magnetic stripes are mostly used on bank debit and credit cards, loyalty cards, airline tickets and boarding passes.

1.2.4 Smart Card Technology

A smart card is an item that contains an embedded IC that has integrated memory, which mostly involves a secure microcontroller or an equivalently intelligent device. In terms of mechanism, smart cards can be considered in three groups; contact and contactless smart cards, and hybrid models.

Smart cards do not contain any power source; hence energy is supplied by the external device, or the reader that the card interacts with. Contact cards receive the required energy via physical contact whereas contactless cards receive power via an electromagnetic field.

A contact smart card communicates with a card reader by direct physical contact, whereas a contactless smart card uses an RF interface for the same purpose. Contact smart cards contain a micro module containing a single silicon IC card with memory and microprocessor. An external device provides a direct electrical connection to the conductive contact plate when the contact smart card is inserted into it. Transmission of commands, data, and card status information takes place over these physical contact points.

In the case of contactless smart cards, the communication is performed only when the devices are in close proximity. One reason for this is increasing security of the communication, and another is enabling higher energy transfer from the active (the device that has embedded power source) to the passive device. As a contactless smart card is brought within the electromagnetic field range of the smart card reader, the card reader spreads out an electromagnetic signal and the smart card is powered by the signal. Once the smart card is powered, it can respond to the request of the reader.

The three major contactless smart cards are ISO/IEC 10536 Close Coupling Smart Cards, ISO/IEC 14443 Proximity Coupling Smart Cards and ISO/IEC 15693 Vicinity Coupling Smart Cards. Close coupling smart cards operate at a distance of up to 1 cm, and proximity coupling smart cards operate at a distance of less than 10 cm (less than 4 in.) at 13.56 MHz. Vicinity coupling smart cards operate in a range of up to 1 m at 13.56 MHz, such as those used in access control systems.

The popular cards are the proximity contactless smart cards which enable a wide range of usage in a wide range of areas from health to entertainment. Various proximity coupling smart card technologies have emerged; however, only a few of them have become ISO/IEC 14443 standard which also provides interface for NFC transactions depending on the operating modes. Currently, the most famous and competing proximity contactless smart cards are MIFARE, Calypso, and FeliCa.

(i) *MIFARE*
MIFARE is a well-known and widely used 13.56 MHz contactless proximity smart card system that is being developed and is owned by NXP Semiconductors which is a spin-off company of Philips Semiconductors. MIFARE is ISO/IEC 14443 Type A Standard. Today, MIFARE is used in more than 80% of all contactless smart cards in the world.

(ii) *Calypso*
Calypso is an international electronic ticketing standard for a microprocessor contactless smartcard, originally designed by a group of European transit operators from Belgium, Germany, France, Italy and Portugal. It ensures multi-sources of compatible products, and makes possible the interoperability between several transport operators in the same area.

(iii) *FeliCa*
FeliCa is a 13.56 MHz contactless proximity high speed smart card system from Sony and is primarily used in electronic money cards. However, FeliCa did not become an ISO/IEC standard.

1.2.5 NFC as a New Technology

NFC operates between two devices over a very short communication range. NFC communication uses the 13.56 MHz spectrum as in RFID. Currently data transfer speed options are 106, 212, and 424 kbps. NFC technology operates in different operating modes; reader/writer, peer-to-peer, and card emulation where communication occurs between an NFC mobile on one side, and a passive RFID tag (NFC tag), an NFC mobile or an NFC reader on the other side. NFC technology is compared with the other technologies in terms of data transfer rate in Chapter 2, Figure 2.23. One of the NFC technology's major properties is its implicit security because of short communication distance. Close proximity of two devices makes the signal interception probability very low. The other property is the automatic implicit pairing capability of NFC. An installed application on a mobile device is automatically launched when it finds the matching pair.

1.3 NFC Essentials

As the basics of the used technology are provided above, we can now introduce essential NFC technical details. In order to do this, NFC structure and the NFC devices (NFC tag, NFC reader, and NFC mobile) must be explained in enough detail. The communication is based on the existing standards, and the devices stick to those standards for a seamless activation. Hence, we also will provide information on the standardization bodies which steer NFC technology.

1.3.1 Smart NFC Devices

NFC devices are the acting components of NFC. NFC is available using three NFC devices: the NFC mobile, NFC reader and NFC tag.

- *NFC enabled mobile phone:* NFC enabled mobile phones which are also referred to as NFC mobiles are the most important NFC devices. Currently, integration of NFC technology with mobile phones (thus introducing NFC enabled mobile phones) creates a big opportunity for the ease of use and acceptance of the NFC ecosystem.
- *NFC reader:* An NFC reader is capable of data transfer with an NFC component. The most common example is the contactless POS (Point of Sale) terminal which can perform contactless NFC enabled payments when an NFC device is touched against the NFC reader.
- *NFC tag:* An NFC tag is actually an RFID tag that has no integrated power source.

NFC works in a very intuitive way. Two NFC devices immediately start their communication as they are touched. The touching action is taken as the triggering condition for NFC communication. This is actually an important feature of NFC technology. In the NFC case, the NFC application is designed so that when the mobile touches some NFC component with the expected form of data, it boots up. Hence, the user does not need to interact with the mobile device after she touches one appropriate NFC device which may be an NFC tag, an NFC reader, or another NFC enabled mobile phone. This is a very useful property of NFC communication that provides ubiquitous computing.

For each NFC communication session, the party that starts or initiates the communication is called the initiator, whereas the device that responds to the requests of the initiator is called the target. This case is analogous to the well-known client server architecture. Remember that in a client server communication the client initiates the communication and the server responds. In NFC communication, it is no different.

In an active/passive device approach, when an NFC component has an embedded power source, it can generate its own RF field, and naturally initiates and leads communication. This device is called an active device. On the other hand, if it does not have any embedded power source, it is called a passive device and can only respond to the active device.

The initiator always needs to be an active device, because it requires a power source to initiate the communication. The target, however, may be either an active or a passive device. If the target is an active device, then it uses its own power source to respond; if it is a passive device, it uses the energy created by the electromagnetic field which is generated by the initiator that is an active device.

Consider an NFC tag which is a low cost and low capacity device. It does not contain any power source and needs an external power source to perform any activity. Thus, an NFC tag is always a passive device and always a target, since it does not include any energy source by design. It stores data that can be read by an active device.

1.3.2 Standardization of NFC Enabled Mobile Phones

NFC technology was jointly developed by Philips and Sony in late 2002 for contactless communications. Europe's ECMA International adopted the technology as a standard in December

2002. The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) adopted NFC technology in December 2003. In 2004, Nokia, Philips, and Sony founded the NFC Forum to promote the technology. NFC technology standards are acknowledged by ISO/IEC (International Organization for Standardization/ International Electrotechnical Commission), ETSI (European Telecommunications Standards Institute), and ECMA (European Computer Manufacturers Association).

NFC is a joint adventure of various technologies. Smart cards, mobile phones, card readers, short range communication, secure communication, transaction and payment systems are the most significant leading technologies. As several technologies are involved, related organization bodies have provided the respective standards. The integrated form of those standards will hopefully define a common vision for secure and yet functional usage and transaction. An interoperable set of standards is essential for a successful NFC ecosystem. The most dominant standardization organizations are:

(i) *NFC Forum*

NFC Forum is an alliance for specifying the NFC standards built on ISO/IEC standards. NFC Forum was established with the aim of enabling NFC technology and making it spread throughout the world. NFC Forum is a non-profit industry association formed to improve the use of NFC short range wireless interaction in consumer electronics, mobile devices, and PCs. NFC Forum promotes implementation and standardization of NFC technology to ensure interoperability between devices and services. The mission of the NFC Forum is to promote the usage of NFC technology by developing specifications, ensuring interoperability among devices and services, and educating the market about NFC technology.

NFC Forum has standardized two operating modes (reader/writer and peer-to-peer operating modes) up to now. Record Type Definition (RTD) and NFC Data Exchange Format (NDEF) specifications are provided by NFC Forum for reader/writer mode communication. Within peer-to-peer mode, Logical Link Control Protocol (LLCP) is used to connect peer-to-peer based application to the RF layer. Card emulation mode on the other hand, provides smart card capability for mobile phones.

Another important development introduced by NFC Forum is the "N-Mark" trademark which is a universal symbol for NFC, so that consumers can easily identify where their NFC enabled devices can be used.

(ii) *GlobalPlatform*

GlobalPlatform is a cross industry, non-profit association which identifies, develops and publishes specifications that facilitate secure and interoperable deployment and management of multiple embedded applications on secure smart cards. The goal of the GlobalPlatform specifications is to ensure interoperability on content management of smart cards, managing smart cards without any dependencies on hardware, manufacturers, or applications.

(iii) *GSM Association* (*GSMA*)

GSMA is an association of mobile operators and related companies devoted to supporting the standardization, deployment and promotion of GSM. GSMA represents the interests of the worldwide mobile communications industry. GSMA is focused on innovating, incubating and creating new opportunities for its members, all with the ultimate goal of driving the growth of the mobile communications industry.

(iv) *ISO/IEC*

ISO is the world's largest developer and publisher of international standards. It is a non-governmental organization that forms a bridge between the public and private sectors. IEC is a non-profit international organization that prepares and publishes international standards for all electrical, electronic and related technologies. ISO and IEC work together to provide worldwide standards.

(v) *ECMA International*

ECMA International is an international non-profit standardization organization for information and communications systems. ECMA studies include mobile devices and NFC.

(vi) *ETSI and ETSI Smart Card Platform* (*ETSI SCP*)

ETSI is a non-profit organization with 700+ members. The ETSI produces globally applicable standards for Information and Communications Technologies (ICT), including fixed/mobile, radio, broadcast and Internet technologies. ETSI SCP handles Subscriber Identity Module (SIM) specifications that would enable SIM cards to carry NFC applications or to play other roles within NFC phones.

(vii) *Java Community Process* (*JCP*)

JCP holds the responsibility for the development of Java technology which indeed is a prominent candidate for NFC applications as well. As an open, inclusive organization of active members and non-member public input, it primarily guides the development and approval of Java technical specifications.

(viii) *Open Mobile Alliance* (*OMA*)

OMA develops open standards for the mobile phone industry. OMA members include many companies including the world's leading mobile operators, device and network suppliers, information technology companies and content and service providers.

(ix) *3rd Generation Partnership Project* (*3GPP*)

3GPP is a collaboration between groups of telecommunications associations to make a globally applicable third generation (3G) mobile phone system specification. 3GPP specifications are based on evolved GSM specifications.

(x) *EMVCo*

EMVCo aims to ensure global interoperability between chip cards and terminals on a global basis regardless of the manufacturer, the financial institution or the card issuer. EMV 2000 specifications are an open standard set for smart card based payment systems worldwide and seek collaboration in mobile payment standards.

1.3.3 General Architecture of NFC Enabled Mobile Phones

Mobile devices those are integrated with NFC technology contain NFC specific ICs such as SEs and an NFC interface (see Chapter 3, Figure 3.7). The NFC interface is composed of an analog/digital front-end called an NFC Contactless Front-end (NFC CLF), an NFC antenna and an NFC controller to enable NFC communication. The NFC controller enables NFC communication of the mobile phone with the external NFC device. An NFC enabled mobile phone requires an SE for performing secure transactions with the external NFC devices. The SE provides a secure environment for related programs and data. It enables storage of sensitive data of the user. It also enables secure storage and execution of NFC enabled services such as

contactless payments. Various standards have already been defined for NFC communication between two NFC enabled devices, and data transfer within the NFC mobile phone such as Single Wire Protocol (SWP) or the NFC Wired Interface (NFC-WI).

The host controller can be identified as the heart of any mobile phone. A Host Controller Interface (HCI) creates a bridge between the NFC controller and the host controller. The HCI is a logical interface which allows an NFC interface including front-end to communicate directly with an application processor and multiple SEs in mobile devices.

1.3.4 Near Field Communication Interface and Protocol (NFCIP)

At the physical layer, the Near Field Communication Interface and Protocol (NFCIP) is standardized in two forms as NFCIP-1 which defines the NFC communication modes on the RF layer and other technical features of the RF layer, and NFCIP-2 which supports mode switching by detecting and selecting one communication mode.

(i) *Near Field Communication Interface and Protocol-1 (NFCIP-1)*
NFCIP-1 standard defines two communication modes as active and passive. It also defines RF field, RF communication signal interface, and general protocol flow. Moreover, it defines transport protocol including protocol activation, data exchange protocol with frame architecture and error detecting code calculation (CRC for both communication mode at each data rate), and protocol deactivation methods.

(ii) *Near Field Communication Interface and Protocol-2 (NFCIP-2)*
NFCIP-2 standard specifies the communication mode selection mechanism and is designed not to disturb any on-going communication at 13.56 MHz for devices implementing NFCIP-1, ISO/IEC 14443 and ISO/IEC 15693.

1.4 NFC Operating Modes and Essentials

Remember that there may three major smart devices in NFC; NFC enabled mobile phones, NFC readers, and NFC tags. NFC communication occurs between two NFC devices with some valid combinations. For example, a mobile phone may communicate with an NFC reader.

As NFC occurs within a very close range, it is very common to touch the communicating devices against each other. For this reason, this process is called touching paradigm. User awareness is definitely a must in order to perform NFC. The user first interacts with a smart object (that is either an NFC tag, NFC reader, or another NFC mobile) using a mobile phone (see Chapter 4, Figure 4.3). After the touching activity occurs, the mobile device may make use of the received data and use mobile services as well, such as opening a web page, making a web service connection and so on.

1.4.1 NFC Operating Modes

There are three operating modes, reader/writer, peer-to-peer, and card emulation, as already mentioned. The reader/writer mode enables one NFC mobile to exchange data with one NFC tag. The peer-to-peer mode enables two NFC enabled mobiles to exchange data with each

other. In card emulation mode, a mobile phone can be used as a smart card to interact with an NFC reader. Each operating mode has a different technical infrastructure as well as benefits for the users.

(i) *Reader/writer mode*
This mode provides communication of an NFC mobile with an NFC tag. The purpose of the communication is either reading or writing data from or to a tag by the mobile phone. We can further categorize the mode into two different modes: reader mode and writer mode. In reader mode, the mobile reads data from an NFC tag; whereas in writer mode, the mobile phone writes data to an NFC tag.

(ii) *Peer-to-peer mode*
Two NFC mobiles using this mode exchange any data between each other. Since both mobiles have integrated power, each one uses its own energy by being in active mode in this mode. Bidirectional half duplex communication is performed in this mode similar to other modes, meaning that when one device is transmitting, the other has to listen and can start transmitting data after the first one finishes.

(iii) *Card emulation mode*
This mode provides the opportunity for an NFC mobile to function as a contactless smart card. Some examples of emulated contactless smart card are credit cards, debit cards, loyalty cards and so on. One NFC mobile may even store multiple contactless smart card applications concurrently. The card emulation mode is an important mode since it enables payment and ticketing applications and is compatible with existing smart card infrastructure.

1.4.2 Reader/Writer Mode Essentials

As already mentioned, the underlying technical architecture of each mode differs. The standards and specifications used by each mode may also differ. In reader/writer operating mode, an active NFC enabled mobile phone initiates the wireless communication, and can read and alter data stored in NFC tags. First, the used RF interface in this mode is compliant to ISO/IEC 14443 Type A, Type B and FeliCa schemes which are contactless smart card interfaces (see Chapter 3, Figure 3.24). The applications operating in reader/writer mode usually do not need a secure area in the NFC enabled mobile phone; the process is only reading data stored inside the tag and writing data to the tag.

In this operating mode, NFC Forum performed various specifications and standards in tag types, operation of tag types, and data exchange format between devices. An NFC enabled mobile phone is capable of reading NFC Forum mandated tag types. Four tag types have been defined by NFC Forum, and are designated as Type 1, Type 2, Type 3 and Type 4. Each tag type has a different format and capacity. NFC tag type formats are based on either ISO 14443 Type A, ISO 14443 Type B, or Sony FeliCa.

The other important standard is the NDEF. NDEF is a data format to exchange information between two NFC devices; namely, between an active NFC mobile and a passive tag, or an active NFC mobile and an active NFC mobile.

NDEF is a binary message format designed to encapsulate one or more application-defined payloads into a single message construct. An NDEF message contains one or more NDEF

records and those records can be chained together to support larger payloads. Various record types for NDEF messaging format are defined by NFC Forum for specific cases; smart posters, URIs, digital signature, and text.

The record types defined for smart posters are the most used. For example, with the defined smart poster record types, URLs, SMSs or phone numbers can be put on an NFC Forum mandated tag. By touching an NFC device to the tag, this information can be read and processed afterwards. The smart poster contains data that will trigger an application in the device such as launching a browser to view a website, sending an SMS to a premium service to receive a ring tone, and so on.

1.4.3 Peer-to-Peer Mode Essentials

In peer-to-peer mode, two NFC enabled mobile phones establish a bidirectional, link level connection to exchange information as depicted in Chapter 3, Figure 3.29. They can exchange virtual business cards, digital photos, and any other kind of data or perform Bluetooth pairing, and so on. Peer-to-peer operating mode's RF communication interface is standardized by ISO/IEC 18092 as NFCIP-1. Also NDEF message is used in this mode which is received over LLCP that is also defined by NFC Forum. The data format is the same as that used in reader/writer mode.

LLCP as a data link layer protocol supports peer-to-peer communication between two NFC enabled devices which is essential for any NFC application that involves a bidirectional communication. LLCP specification defines five major services: connectionless transport, connection oriented transport, link activation-supervision-deactivation, asynchronous balanced communication and protocol multiplexing.

1.4.4 Card Emulation Mode Essentials

In card emulation mode, an NFC enabled mobile phone acts as a smart card. Either an NFC enabled mobile phone emulates an ISO 14443 smart card or a smart card chip integrated in a mobile phone is connected to the antenna of the NFC module. When the user touches the mobile phone to an NFC reader, the NFC reader initiates the communication. This operating mode is useful for secure transactions such as contactless payment, ticketing applications and access control.

As depicted in Chapter 3, Figure 3.32, when an NFC reader interacts with an NFC device, the NFC device acts like a standard smart card, thus the NFC reader interacts with the SE and its applications. Only the card emulation mode uses an SE efficiently and performs functions securely.

1.4.5 Case Studies

We present the following three case studies at the end of Chapter 4 to clarify the three operating modes and their usages thoroughly:

1. The NFC enabled shopping system enables users to shop online anywhere they want, so that no geographical restrictions are set. This use case employs the reader/writer mode.

2. The NFC based gossiping application works in the same way as gossiping and disseminates information between the parties. This use case employs the peer-to-peer mode.
3. The cinema ticketing application enables payment to be made. This use case employs the card emulation mode.

In each use case, initially the description of the case is given. Use case diagrams, activity diagrams, and generic usage models follow. The first and second use cases are also implemented with Java in Chapter 5. The codes may run in emulator or mobile phone after successful implementation. The third use case's ecosystem environment and business models are analyzed in Chapter 7.

1.5 SE and Its Management

In order to provide secure storage and execution of NFC enabled applications, an SE is essential. The SE is actually a combination of hardware, software, interfaces, and protocols. Since secure functions are mostly provided in card emulation mode, an SE is mostly used in that mode as well. When an SE is used appropriately, that is, according to the provided standards, the users and service providers are assured about the security of the overall process. Currently various SE alternatives are being considered, but the most popular ones are (see Chapter 3, Figure 3.10):

- Embedded hardware;
- Secure Memory Card (SMC);
- Universal Integrated Circuit Card (UICC).

(i) *Embedded hardware*

The embedded SE is a non-removable component within a mobile phone. This chip is embedded into a mobile phone during the manufacturing stage and it must be personalized after the device is delivered to the end user. This embedded SE chip obviously cannot be transferred to other mobile phones. It has to be replaced and personalized every time the mobile phone is used by another user. The SE of a new mobile phone must be personalized for the user.

(ii) *SMC*

A removable SMC is made up of memory, embedded smart card element and smart card controller. In other words, it is a combination of a memory card and a smart card. With the removable property and a large capacity memory, the SMC based SE can host a large number of applications, and does not need to be reissued when the customer buys a new mobile phone.

(iii) *UICC*

The UICC is the physical smart card that the Subscriber Identity Module (SIM) or Universal Subscriber Identity Module (USIM) is implemented upon. Therefore it is commonly known as a SIM or USIM. A UICC based SE is a removable smart card used in mobile terminals in GSM and UMTS networks.

A UICC based SE provides an ideal environment for NFC applications. It is personal, secure, portable and easily managed remotely via OTA technology. The card holder can be assured that transactions are executed with their personal information protected. It has appropriate card structure based on the GlobalPlatform card specification that allows multiple security domains for different applications on the same smart card. By use of OTA, new NFC applications can remotely be installed onto the UICC easily, personalized afterwards, and the life cycle of the SE can be managed easily thereafter. Hence, the user does not need to physically touch the NFC enabled mobile device to a system in order to perform any of the processes mentioned.

1.5.1 Over-the-Air Technology

OTA is the standard for exchanging applications and application related information through wireless communications media. By facilitating an OTA platform, new services can be introduced; the SEs can be accessed, manipulated, and modified in a rapid and cost effective way through an OTA platform. Based on the ecosystem that has been agreed upon, the OTA service can be provided by an MNO or another trusted entity.

Currently, UICC cards are produced, submitted to the user, and hence owned by the MNOs in an ad hoc fashion. MNOs control and manage the UICC, and use OTA functionality provided by the same MNO to manage the UICC. Hence, OTA capabilities are considered as part of the main functionalities of MNOs as well as part of mobile device management.

1.5.2 GlobalPlatform Card Specification

Remember that GlobalPlatform card specification contains the details of the smart card. Based on the standards defined by the GlobalPlatform, the logical and physical components of the smart card aim to provide application interoperability and security in an issuer controlled environment.

Security domains are extremely important for the GlobalPlatform based smart cards to provide enough level of security. As a matter of fact, security domains are the on-card representatives of off-card authorities. They enable management of applications in a secure fashion by providing a complete separation of cryptographic keys and the security domains can be categorized, reflecting different types of off-card authority recognized by a card:

- The Issuer Security Domain (ISD) is the on-card representative of the card issuer. This component represents the issuer's area on the card that controls the issuer's applications.
- The Supplementary Security Domain (SSD) is the on-card representative of the application providers and card issuers. This component allows the application providers to share and utilize a territory on the card without the risk of compromising management of the card or any application on the card.
- The Controlling Authority Security Domain (CASD) is actually a sub type of SSD. The role of the controlling authority is to enforce security policy to all applications on the card.

1.5.3 Trusted Service Manager

In order to deploy NFC applications and services onto the user's SEs, service providers and MNOs have distinct requirements. To create and manage a trusted environment and to allow actors to communicate among themselves securely, an additional trusted actor is required; Trusted Service Manager (TSM) which is an independent party serving the other actors of the NFC ecosystem as required. The TSM ensures a level of trust and confidentiality between major actors of the system such as service providers and MNOs during the application life-cycle management.

1.5.4 UICC Management Models

GlobalPlatform smart card standards define the card content management process to contain several activities: loading the initial key set of the security domain, application coding of a third party, application loading, personalization and so on.

GlobalPlatform defines three card content management models as simple mode, delegated mode and authorized mode. These models cover application loading and personalization processes on SEs. The simple mode is a completely card issuer centric model, whereas the delegated mode and authorized mode are more TSM centric models. GlobalPlatform messaging specification supports all deployment models.

(i) *Simple mode using MNO OTA platform*
In simple mode, the service provider delegates full management of its NFC enabled application to a TSM. TSM manages the security domain on behalf of the service provider. MNO is authorized to perform the card content management functions, namely loading, installing, activating and removing the application on the SE. TSM only manages the application lock, unlock, and personalization processes using its own OTA server and network of the MNO.

(ii) *Delegated mode with full delegation to the TSM*
The delegated management case can be described as TSM centric loading. In this case the MNO is no longer in charge of loading, installing, activating or removing the application. Card content management is performed by the TSM with a pre-authorization from the MNO. In some cases, the service provider may need to manage its own application personalization process to prevent any third party manipulation of application keys or application data.

(iii) *Authorized mode with full delegation to TSM*
The authorized management deployment is completely organized around the TSM centric loading option. TSM has service provider applications, and is able to perform card content management without authorization (or being forced to use a token) from the MNO. As in delegated mode, the service provider can manage its own application personalization instead of delegating it to the TSM.

1.5.5 Multiple SE Environments

It is also possible to see that a single NFC enabled mobile phone can host multiple SEs. For example, a mobile phone may contain an embedded hardware or SMC that is integrated

by the manufacturer during the production process, together with an UICC based SE that is embedded by the MNO before handing the mobile phone to the customer. Many problems may emerge when multiple SEs reside on the same NFC enabled mobile handset. One of the problems can emerge when implementing card emulation mode applications. The NFC reader needs to initiate communication with only one SE on the NFC enabled mobile phone. Another problem is the management of multiple SEs at the same time. According to GlobalPlatform, two business models can be performed: architecture without aggregation and architecture with aggregation:

- *Architecture without aggregation*: The NFC controller can be used by only one SE at a given time, thus only one SE can be active at a time. The SE is activated by the user so that the activated SE is able to perform NFC enabled contactless transactions. The user is responsible for selecting the correct SE in this model.
- *Architecture with aggregation*: All SEs hosted in NFC enabled mobile phones are active at the same time. Any application on any SE can perform contactless NFC transactions at any given time. Obviously each SE can contain one or more applications. The application should be selectable by the external NFC reader.

1.6 NFC Application Development

Developing NFC applications is an important part of NFC technology. In order to develop NFC applications, complete understanding of NFC technology and operating modes are required. There are two different types of applications in NFC services. The first one is a Graphical User Interface (GUI) application which exists in all operating mode applications. A GUI application provides an interface which allows a user to interact with the mobile device. It also provides the capability to read and write from and to NFC components. The second type is an SE application which is needed in order to provide a secure and trusted environment for security required applications (e.g., credit card). There are various development tools on the market targeting different mobile phones. Some of these development tools are:

- Android SDK (Software Development Kit) for Android mobile phones;
- Qt SDK for Symbian^3 mobile phones;
- Series 40 Nokia 6212 NFC SDK for Nokia 6212 devices.

Java is a well-known object-oriented programming language, and can be used in various environments from PCs to refrigerators, and from servers to mobile phones. Java, in the NFC context, provided one of the first APIs (Application Programming Interfaces). We have provided information on how to develop NFC applications using Java in Chapter 5. Lessons learned from Chapter 5 will help users develop NFC based applications in other development platforms as well.

Java provides two APIs for NFC development: JSR 257 (Contactless Communication API) and JSR 177 (Security and Trust Services API). JSR 257 mainly enables reader/writer mode application programming resources, whereas JSR 177 and some classes in JSR 257 provide access to SEs to implement card emulation mode projects. For peer-to-peer mode

Table 1.1 JSR 257 packages

Package	Description
javax.microedition.contactless	Provides common functionalities to all contactless targets such as discovering the target
javax.microedition.contactless.ndef	Provides functionality to exchange NFC Forum formatted data with RFID tags
javax.microedition.contactless.rf	Enables communication with RFID tags that contain non-NFC Forum formatted data
javax.microedition.contactless.sc	Enables communication with ISO14443-4 smart cards
javax.microedition.contactless.visual	Enables communication with visual tags

programming, propriety APIs are required, since this mode is not supported by the standard Java APIs.

1.6.1 JSR 257

This API provides an application programming interface that allows applications to access RFID tags, smart cards, and visual tags (bar codes). Different packages are defined for each target type and an application using this API can discover contactless targets in the proximity, notify applications upon discovery and perform contactless operations. A few classes in this API also allow access to SEs using ISO 14443 connection. Table 1.1 summarizes the packages in JSR 257.

1.6.2 JSR 177

This API defines optional packages to support smart card communication and security operations. Digital signature services, user credential management, cryptographic operations and more can be implemented using this API. It enables mobile phones to access smart cards using APDU (Application Protocol Data Unit) and JavaCard Remote Method Invocation (RMI) protocols. JSR 177 consists of four optional packages (summarized in Table 1.2) which also consist of different packages and classes.

Table 1.2 JSR 177 packages

Optional Package	Description
SATSA-APDU	Enables communication with smart cards using a protocol based on APDUs
SATSA-JCRMI	Enables communication with smart cards using JavaCard RMI protocol
SATSA-PKI	Enables smart cards to manage digital signatures and certificates
SATSA-CRYPTO	Provides cryptographic operations such as message digests and digital signatures to increase security

Peer-to-peer mode is not supported by standard Java APIs; however, there are extensions to Contactless Communication API that provide programming in peer-to-peer mode. These are generally mobile device specific APIs, namely proprietary APIs.

Push registry is another important topic in NFC and can increase the usability of applications. Normally, applications can be executed by a user's action from the mobile phone's menu. However, an application can also run without any user action, but with the 'Push' feature. For example, an application can be triggered with a network package received by an incoming SMS. Push registry simply maintains a list of inbound connections and runs applications automatically based on received connections. In the case of NFC, it enables applications to launch with an NFC connection.

In order to activate a push registry service, an application should be registered to run with a specified connection. Two types of registrations are possible: dynamic and static. With static registration, an application is registered for a push record when installing it in the mobile device. On the other hand, dynamic registration is performed at the first execution of the application.

Dynamic and static registrations have their own advantages and disadvantages. Static registration enables a push connection to register easily without a user's action at installation stage. However, if there is a conflicting push registry entry in the device, the application cannot be installed. On the other hand, dynamic registration eliminates this issue. One disadvantage of dynamic registration is that in order to save the push registry record, the first execution of the application needs to be performed with a user's action.

1.7 NFC Security and Privacy

Although a mobile phone is almost identical to a PC in technical terms, it is different since it is more a personal item and mostly carried by people in daily life. Users generally believe that their mobile phones are an important part of their lives and they usually put them under physical surveillance. However, a mobile phone is still subject to physical attacks such as theft, and technical wireless attacks using Bluetooth or WI-FI communication technologies. Integrated NFC capability also imposes some additional risks to mobile phones.

1.7.1 Why is Security Important?

A service is useful only when it is both functional and secure enough. A user initially cares about the functionality, and only subsequently notices the importance of the security. In conclusion, a user is eager to use a service only when it is functional and yet secure. Technical deficiencies and security related obstacles are potentially of concern to users.

People were not aware of the importance of the security of services decades ago. As malicious activities started to emerge, both people and companies started to notice the importance of security but only after the cost of the damage became sizeable. The hackers also noticed that they could gain financially in addition to being proud of gaining illegal access to the assets and causing damage as well.

There are some obvious reasons why the security risk has increased so much lately. Three viewpoints are given.

Hackers' point of view:

- They may get financial reward.
- They may satisfy their egos.
- They may even become famous when they perform incredibly successful actions.

Users' point of view:

- The number of Internet users is increasing exponentially; hence the opportunity for malicious actions is also increasing. One hacker can try the same method to hack many potential victims.
- The amount of financial assets has grown, resulting in greater financial reward to hack.

Technical point of view:

- Organizations traditionally ignore security during the initial stage of software development. One reason for this is to keep expenditure within the budget, and security seems the most obvious area in which to reduce costs. This results in failure in security of the system even if a vast amount of effort is spent subsequently, since it is not easy to integrate security if it is ignored during the design phase.

Now it is time to make a projection of the importance of security in general in the NFC ecosystem. The major reasons why NFC is an alluring target are:

- NFC is an emerging technology and integrated with mobile phones.
- Nearly everybody owns a mobile.
- NFC is heavily promoted by service providers, MNOs, and banks.
- NFC potentially has a big financial market, which is tempting for the hackers.

1.7.2 Primary Goals of Security Measures

The security requirements of each system are different; their priorities are different as well. Some prioritize keeping the information available to only one or more people, while others demand keeping the application data contents unchanged by illegal parties.

The most common security requirements can be listed as follows:

- Secrecy ensures that information is accessible only to those with authorized access.
- Authentication approves the identity of a person, a process, or a device based on the provided information.
- Authorization allows different actions on the object (file, application, or machine) by the subject (user) after authentication is provided.
- Non-repudiation of a party prevents the sender from denying sending a message to the receiver, so that a referee can prove the case.
- Availability ensures that the system responds correctly and completely to the requests of the authorized users at any given time.

- Data integrity ensures that the received information is exactly the same as the information sent, and thus confirms that it has not been accidentally or maliciously modified, altered, or destroyed.
- Accountability guarantees tracing back all the actions together with the actor who performs.

1.7.3 Vulnerability, Threat, Attack, and Risk

In order for a malicious action to create damage on the secured system, initially the system should be vulnerable to attacks. In this sense, vulnerability is a weakness in a system which allows an attacker to perform some actions that threatens its information assurance.

A threat is a possible danger that may cause an unfair benefit to the unauthorized user or cause harm by making use of vulnerability.

An intentional attempt by intruders to perform an unauthorized access to information is called an attack.

Attacks are classified as active or passive. If an attack does not modify or delete a resource it is classified as passive, otherwise it is classified as an active attack.

The potential harm that may arise after the realization of some threat is further defined as the risk.

1.7.4 Security Tools and Mechanisms

In order to satisfy the security requirements, cryptography is the primarily used technique. Most of the security mechanisms rely on cryptography. Cryptography is used for providing a secure channel, storing password information within the hard disk, digitally signing the financial transactions, and so on.

Cryptography is also used for many purposes such as hiding the content of the data from an unauthorized third party, or preventing illegal modification of some transmitted data. The following are the basic services that are provided by cryptography:

- The stored or exchanged information is not revealed to the unauthorized parties.
- The content of the stored or exchanged data cannot be changed by unauthorized parties, or it will be noticed if it occurs.
- When the data are created or sent by some party, the party cannot deny creating or sending them.

In cryptography the original data (plaintext) is initially encrypted by using an encryption key to create a modified form of the plaintext, called the ciphertext. The data can simply be stored or be transferred to the receiver. The receiver decrypts the data by using the decryption key in order to recreate the original message.

The idea of satisfying secrecy using cryptography is being able to send the message in a scrambled form called a ciphertext, so that communication between the sender and the receiver is still possible, and can be performed using public channels such as the Internet.

1.7.5 NFC Security

As with all information systems, NFC based systems are subject to attacks that threaten system security and user privacy. Each operating mode of NFC has a different architecture. Hence, attacks and defense mechanisms are mostly subject to different use cases. When NFC based systems are analyzed from the security point of view, we should consider the security concerns related to the NFC tag, NFC reader, smart card, communication and backend systems separately.

1.7.5.1 Security Issues on NFC Tag

Remember that the NFC tag is involved in reader/writer mode. In this mode traditionally an NFC mobile interacts with an NFC tag. In order to satisfy the overall security requirements, the security of the data on the NFC tag as well as the security of the communication between NFC devices must be secured. Remembering that the NFC tag is actually an RFID tag, we can make use of the knowledge that is accumulated by using RFID tags to handle the security of the same tag in NFC. Traditionally, the following are the security issues related to the NFC, or RFID tag:

(i) *Tag cloning*
The attacker may try to clone, or create an exact copy of a valid tag. In order to insert preventive mechanisms to the system, applications that require high processing capability are required, increasing the cost of low-cost tags. Obviously this is unfeasible and unacceptable, since the major point here is enabling low-cost NFC tags.

(ii) *Tag content changes*
The attacker may try to modify an NFC tag to change its content. In this way, several attacks become possible:
- Spoofing attacks
  Spoofing attack is providing false information to the user which seems valid, and hence possibly will be accepted by the user. By spoofing attack, the user may insert a fake domain name, telephone number or false information about the identification of some person, item, or activity on to the tag.
- Manipulating tag data
  The content of the tag might be changed by the attacker for some malicious purpose.
- Denial of Service (DOS) attack.
  DoS attacks aim to damage the relationship between the customer and the service provider. The primary way to do this is by exhausting the system's resources by forcing it to perform some unnecessary and illegal action. This results in decreasing and eventually exhausting the power source of the server.

(iii) *Tag replacement and tag hiding*
The NFC tag may be replaced by a malicious tag, so that the latter tag performs illegal actions as it is designed to do. Sticking a malicious tag on top of the original tag or replacing the original tag with a malicious tag is called tag hiding, and is enough to let the system work as the attacker desires.

1.7.5.2 Security Issues on NFC Reader

Remember that card emulation mode involves the interaction of the mobile phone with the NFC reader. Hence, security of the reader is the concern in this mode. An NFC reader is similar to the RFID reader; therefore the security concerns are consequently similar. These kinds of similarities are important and reassuring, since the potential problems have been mostly solved up to now.

1.7.5.3 Security Issues on Smart Cards

Smart cards on the mobile phone are mostly used by applications operating in card emulation mode in an NFC ecosystem. Hence, the security of a smart card is a major concern in this mode. The attacks on smart cards can be categorized into two groups: invasive attacks and side channel attacks.

1.7.5.4 Security Issues on Communication

In all operating modes of NFC technology, it is obvious that a short range communication is used. Attackers with enhanced radio devices can communicate with the contactless smart cards within several meters. Therefore, attacks and threats during the communication are valid in all modes.

- *Eavesdropping:* An unauthorized individual may use an antenna in order to record communication between NFC devices.
- *Data Corruption:* In addition to eavesdropping, an attacker can try to modify the transmitted data.
- *Data Modification:* The attacker may try to modify or delete valuable information by intercepting the communication.
- *Data Insertion:* Data may be inserted into the exchanged messages between two NFC devices. The attacker must be fast enough to send the data before the valid responder. The insertion will be successful only if the inserted data can be transmitted before the original device responds. If both data streams overlap, the data will be corrupted.
- *Man-in-the-Middle Attack:* These attacks are performed by unknown parties in a communication, who relay information back and forth by giving the simultaneous appearance of being the other party.
- *Relay Attack:* The attacker uses wireless communication to borrow the data from the victim's tag into another tag. This means that the attacker inserts messages into the exchanged data between two devices.
- *Replay Attack:* A valid NFC signal is intercepted and its data is recorded first; this is later transmitted to a reader so that it is "played back". Since the data appear valid, the reader accepts them unless suitable prevention mechanisms are used.

1.7.5.5 Middleware and Backend System Security

An NFC based system contains NFC readers, NFC mobiles and NFC tags in technical terms. A complete NFC system includes servers to store and manage data such as banking servers, credit

card middleware, authentication subsystems, and so on. Hence, security of an NFC system is not complete unless the security of all components of the system is provided. Security issues of the middleware and the backend systems are not within the scope of this book. However, the reader should be aware of the fact that the middleware and backend systems should be secure.

1.7.5.6 Standardized NFC Security Protocols

All NFC related security protocols are standardized in ECMA 385 as NFC-SEC and in ECMA 386 as NFC-SEC-01. These protocols deal with the NFCIP-1 security issues since NFCIP-1 provides no security at all. NFC-SEC provides security standard for peer-to-peer NFC communication. On top of NFCIP-1, NFC-SEC is promoted to insert security capabilities.

1.7.6 Privacy, Legal, and Ethical Aspects

Information that is voluntarily shared but is later stolen or misused is an important issue nowadays. Privacy requires the appropriate use of information. When something is private to a person, it usually means that it is considered inherently and personally special. Privacy is broader than security and includes the concepts of appropriate use and protection of information.

A few decades ago it was considered to be the information era. Now is regarded to be the age of information management. The reason for this difference is obvious: data and information are everywhere. It is more important to retrieve the useful information than to create it.

A huge amount of data is generated by various means all over the world. Data are collected worldwide using sensors, applications, and other data collecting devices. Wireless sensor nodes and cameras are two examples of data collecting devices. When you call somebody or even carry your mobile without using it intentionally, buy something even by using cash, or click on a website on your browser, many data are recorded. Hence, generating data is no longer a problem. The issue is how to make use of the data.

It is true that people do not like to be recorded when entering a building, walking on a road, meeting a friend in a café, and so on. It is unfortunate that this trend will not be changed even if it can be slowed down temporarily. It is even more annoying if the captured data are used in activities invading users' privacy.

The rules must be changed worldwide, because currently it is so easy to use the information gathered about people even without their permission.

1.7.6.1 What to do to Protect Privacy

It is true that wireless communication is subject to a higher number of security risks when compared with wired media. It is also true that mobile devices have lower processing capability because of higher hardware cost, which results in vulnerability in carrying out services implemented to build security measures.

Another fact is that companies traditionally cared about the security of the products much less than the functionality. Hence, initially people were happy when they bought new applications,

but then encountered damage as the vulnerabilities were misused by malicious actions. Hence, public suspicion towards the new technology greatly increased.

Since the public became more suspicious of mobile and wireless systems, RFID also received public opposition as well. Hence, RFID met with fear and rejection over time.

In the RFID case, customers complained when they were not informed by companies that they had embedded RFID tags to their products. Even if they had been informed, the customers were not told of all the risks. Hence, people became angry when they eventually learned the facts. It is possible to leak sensitive data that is on the tag especially if the individual is unaware of the tag.

Since NFC is an emerging technology, the collection, storage and usage of sensitive private data is of public concern as well. Valuable private data such as payment card information and personal identity information are at risk when appropriate measures are not taken.

In order to appease public concern, detailed information as well as all the risks should be given to the potential customers and appropriate security measures should be applied as well. In order to achieve the trust of users there is a clear need for effective tools that support users to protect their privacy. Supporting data protection legislations, other legal measurements and auditing mechanisms can also be required. In this way, NFC application users might be convinced that these applications will not misuse their personal data and privacy.

1.8 NFC Business Ecosystem

Many technological improvements have been introduced so far. Some historical examples include radio receivers, televisions, cable television channels, satellite television channels, phones, mobile phones and satellite phones. Some of these technologies have been successfully accepted whilst others have just disappeared. There are various reasons for acceptance or rejection of such technologies. The primary requirement for wide usage of such technology is technical success of course. Without technical sufficiency, no technical device can be used.

The next factor that affects the success of the technology is public acceptance. If people find it useful enough, then they are eager to pay for it. As more people pay for the devices, the companies earn more income and provide further investment to improve the technology. As the technology is improved the unit price is reduced, which makes more people willing to own the device. This cycle carries the price of the item to a reasonable level so that the new technology eventually becomes successful.

As we can see from the last paragraph, the dominant factor of the success of the model is actually more financial than technological. Economical motivations enable technical sufficiency over a period of time. We have already examined the user's involvement in the process. From the investor's perspective, the Return of Investment (RoI) is the primary motivation of companies. Companies always request a small timeframe in which to get their money back.

It is true that in any technological device usage, there is more than one company involved. Consider television broadcasting for example. There are manufacturers, cable companies, advertisement firms, infrastructure firms and many others. Each company tries to maximize their profit, thus decreasing their RoI as much as possible. We can illustrate the financial aspect using a cake. The size of the cake is affected by the number of users that are willing to pay money for the products and services. The share of each company is then defined. Each company tries to get the most. If the resultant sharing set-up is healthy then the system

works. In contrast, if any major company cannot receive enough money, its RoI increases too much and hence that company does not invest any money. Let us use the television case again. If the television manufacturer tries to get most of the money by increasing the cost of the televisions, people will stop buying televisions, and hence nobody wins. As another alternative, if the channel service provider requests too much money, then people will stop purchasing televisions, since they will not be willing to use televisions any more. Hence, a good agreement on who gets what and how defines the success of the outcome.

NFC technology is no different. A success story will be written only when the players agree on how to share the profit, which is not settled yet.

It is true that the NFC ecosystem exposes an inviting financial share to all related partners. NFC creates a new business environment and large value chain. The financial shares whet the appetite of many companies.

The potential of NFC technology in business opportunities (especially in the mobile financial services industry) has caused great excitement in many organizations. Since NFC technology is made up of several components, it cuts across boundaries of many organizations from diverse business sectors.

The leading companies that might be enthusiastic about the emerging ecosystem are MNOs, banking and payment services, product manufacturers including mobile handset makers, software developers, and other merchants including transport operators and retailers.

Some worldwide companies try to get all possible profit from the NFC ecosystem. Some MNOs for example, try to circumvent the banks in order to get all the added value. The same is true for some banks as well. Some mobile manufacturers try to create appropriate models so that they will not need any MNOs or banks for facilitating NFC services. This is rather disappointing though. What the companies should understand is that no single company can get all the profit. The money to be shared can be increased only when all related sectors agree upon a share model which satisfies all companies. Otherwise, the money to be shared will only be a small amount.

NFC take-off has been slower than expected. The reason for this is not technical. The foremost reason is the lack of formation of a common understanding and vision in NFC technology among participating organizations and industries. Thus a mutually beneficial business model could not be sustained yet. The main reasons for this lack of common understanding and vision are:

- The profit that will be shared is enormous. Hence every company is eager to get a big share.
- All parties are powerful companies so that they assume that the other parties are obliged to follow their demands.
- Different technical solutions and infrastructures for a specific NFC enabled service are an issue as each party proposes a model which brings more advantage to itself than others. For example, MNOs propose SIM based models, since they can control the SIM cards and hence can receive more profit if this model is used. Nobody tries to find solutions that would possibly make all parties happy.

To achieve a fair and yet a profitable business model, interoperability, compatibility and standardization of an NFC technology model that is agreed upon are essential. It is also important to get customer acceptance.

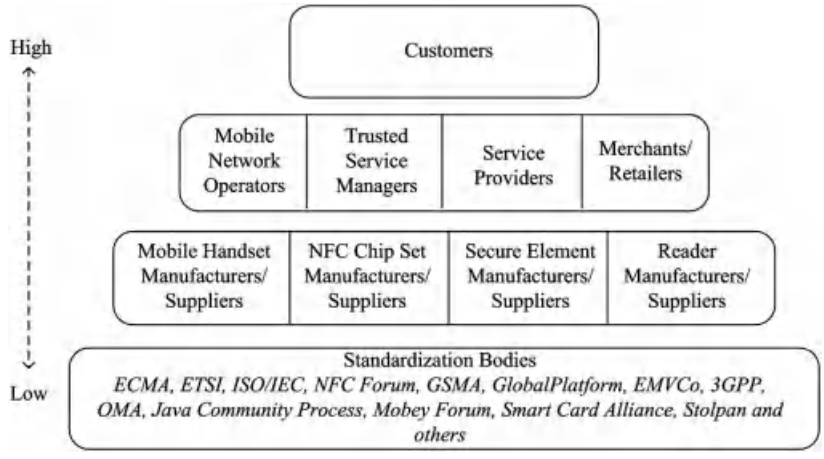


Figure 1.1 Stakeholders in NFC ecosystem.

1.8.1 Stakeholders in NFC Ecosystem

A wide range of stakeholders are potentially involved in each NFC project depending on the type of services provided. Some of the potentially leading services are smart poster, payment, ticketing, and social media processing. The type of the used service defines the potential players, details of the need for collaboration among those players, the money to be shared and so on.

As depicted in Figure 1.1, we can examine the participating entities within the NFC ecosystem on four levels. At the lowest level the standardization bodies exist who aim to develop global, interoperable standards for NFC and its dependent technologies such as smart cards and mobile phones. This level can be seen as the fundamental level for NFC technology and its ecosystem. The next level involves the hardware (i.e., mobile handset, NFC chip set, SE, NFC reader) manufacturers and suppliers. They are responsible for producing and selling products conforming to standards for enabling NFC based systems. For instance consider SE manufacturers and suppliers. They need to provide appropriate SEs to the right customers. Embedded hardware and secure memory card based SEs need to be manufactured for the mobile handset manufacturers or other retailers, and UICC based SE for MNOs. UICC manufacturers and suppliers are contracted and forced to provide required UICC hardware to the MNOs. Thus, they have a direct relationship with the MNO, who also defines the requirements for the UICCs and issues the SE.

The major players exist in the next level. These parties are the ones who effectuate the NFC enabled applications and services in a secure, trusted environment. They can be described as follows:

- MNOs traditionally provide mobile phone owners access to the communication and data network. They are actually currently responsible for – indeed have the privilege of – providing all kinds of mobile services to their subscribers. MNOs, in the meantime, can provide and maintain the network infrastructure that enables the secure OTA solutions to provide remote management and maintenance of applications stored on SEs.

- TSM is required to create and manage a trusted environment among actors of the NFC ecosystem. TSM generally offers a single point of contact with MNOs for service providers such as financial institutions, banks, transit authorities, retailers and others who want to provide a payment, ticketing, or loyalty application to customers with NFC enabled mobile phones.
- The service provider deploys as well as manages a service to the mobile devices of its customers. A service provider may be a financial institution, bank, transport authority, or some other organization.
- In the NFC ecosystem context, merchants and retailers are the stakeholders who are accepting contactless proximity mobile payment services.

These actors have a direct relationship with the end users as customers at the highest level. Customers are always the principal stakeholders in any business and are the focus of service providers; the reason for this is obvious. Customers will sustain the business only if their needs are met. All marketing activities always focus on them.

As already mentioned, several NFC trials have been implemented around the world since 2005; and operators, banks, and SE issuers are ambitious to increase their revenue through NFC services. Especially in European industry, banks and MNOs are competing industries, thus different business models and approaches can be seen to enable NFC services. For example, in Austria some MNOs have obtained banking licenses, so that they are able to provide financial and banking services. The Mobile Virtual Network Operator (MVNO) model has gained popularity in recent years. MVNOs allow MNOs and non-telecom companies to participate in mobile services. For example, in the Netherlands, a bank called Rabobank started to act as an MVNO by collaborating with an MNO to provide mobile services with financial services to its customers. Such an MVNO model creates a win–win model through MNO cooperation; the model has been rapidly expanding in countries such as Spain, Italy, and Portugal and more recently Turkey. MVNOs can increase their brand awareness, distribution channels, and customer base to provide value added services to their customers. On the other hand, MNOs can increase their network capacity, IT infrastructure and service portfolio, add a new revenue stream and so on.

1.8.2 Understanding NFC Business Models

In NFC business models, problems to be solved are mostly business related issues than technology or infrastructure related. Business models are important to deliver value to all stakeholders. Some business models do not encourage strong cooperation between all bodies; however in NFC it is a must.

Currently, there is a pervasive amount of uncertainty on which business model is the best; which firm will perform exactly which activity, and who will pay whom for which service and eventually how much profit is to be earned or shared by any stakeholder. Due to the novelty of the NFC technology, there is still no agreement or common vision of a business model that sufficiently satisfies all stakeholders. Actually, many business model proposals and specifications have been published and various projects are implemented through a vast number of trials throughout the world, especially in mobile financial services due to the high degree of complexity in ecosystem and technological infrastructure. NFC Forum, GlobalPlatform,

GSMA, and EMVCo are some of the important associations who intensively work on the NFC ecosystem and business models as well as work on the underlying technological infrastructure.

In summary, it is essential to harmonize the interests of all participants in forming sustainable business models. Otherwise conflicting, not feasible, and not interoperable solutions will be performed, and hence the technology will not be able to be improved.

There are some key questions that help us to understand the business environment of NFC enabled applications and services. These questions are:

- Who will issue and own the control of the SE?
- Who will manage the life cycle of the SE platform?
- Whose OTA platform will be used for management of the SE platform?

Relating to these questions, we have identified three main issues that determine business model alternatives for NFC; SE issuer, platform manager, and OTA provider. These issues can also be referred to as functional roles and responsibilities that need to be handled by a single entity or multiple entities in an NFC business model.

(i) *SE Issuer*

The SE issuer is the entity who issues and owns the SE. Currently, in almost all cases MNOs or service providers (e.g., banks) concurrently play the role of the SE issuer. In an ideal case, the SE issuer should be an independent actor, but currently this is not the case.

If UICC based SE is used in a business model, MNO holds the responsibility of SE issuer, since the SE is given to the user when the SIM containing the SE is provided. If embedded hardware based SE is used, the actor who gives the mobile phone to the user plays the role. If the mobile phone is given to the current customer of a MNO within the context of a campaign for example, the MNO is the card issuer. If the mobile is purchased by the user from a retailer or from a service provider, the card issuer can be any independent partner theoretically such as the retailer, service provider or other.

(ii) *Platform Manager*

The second important issue is the control and management of the SE platform. The platform manager owns the cryptographic keys that are used to control the SE in its lifecycle. The platform manager allows authorized service providers to install applications on the SE preferably using an OTA infrastructure.

The business model is simpler when the SE issuer and platform manager are the same entity. Delegating the platform manager role to some other party than the SE issuer is yet another option. Indeed this entity can be a TSM as an independent neutral party. In order to realize this option, the TSM and SE issuer previously agree on the business model. The agreement possibly consists of the details from technical infrastructure to revenue sharing. The TSM can handle all SE management functions by using its own OTA platform or MNO OTA link, which is the third important issue covered next.

(iii) *OTA Provider*

The final important issue is about the provision of the OTA Platform. Providing a flexible and interoperable OTA solution is a key requirement in the NFC ecosystem. It enables secure wireless communication between two end parties and provides transmission and reception of application related information in a wireless communication system. OTA

enables remote download, installation, and management of applications such as updating, activating or deactivating an application stored on SE.

1.8.3 Business Model Approaches

Currently available business models are MNO centric, distributed, and TSM centric alternatives:

(i) *MNO centric business model*
In the MNO centric business model, MNO issues SEs, and acts as SE issuer, platform manager and OTA provider at the same time. There is no other independent trusted entity; MNO performs all capabilities of TSM, owns, and manages the life cycle of SE using its own OTA platform. MNO also performs loading, installation, and personalization processes, as well as security domain creation on the SE. Service providers have to pay the MNO in order for their applications to run on the SE, and even share their personalization data with the MNO.

(ii) *Distributed business model*
In the distributed business model, the platform management services are distributed between the SE issuer and the service provider. There can be a separate TSM infrastructure, and using no infrastructure is still another option as well. Actually the choice depends on the entity's TSM capabilities. If the service provider has no TSM capabilities, they need to make an agreement with an existing TSM. Currently service providers prefer to collaborate with trusted third parties rather than making high investments to build OTA infrastructure within their organizations.

(iii) *TSM centric business model*
For an NFC service, a single TSM centric business model is actually the best option and a less complex business model at the same time. MNOs and service providers that want to participate in that specific NFC service ecosystem need to sign and agree with an existing TSM in the market. The TSM performs the platform manager's role completely on behalf of the service providers; this is done by realizing loading, installation, and personalization processes via its own OTA platform. The number of TSMs may increase depending on the available services and the agreements of the actors in the NFC ecosystem. For instance, NFC enabled payment and transportation services may use the same TSM platform as well as different TSM platforms.

In order to create a sustainable model, it is really important to create a win–win business model for all stakeholders in a market with many additional revenue and marketing opportunities. In order to move forward successfully with all kinds of business models, understanding the business requirements of all stakeholders especially MNOs and service providers, and establishing trust between them is essential. A secure and simple ecosystem can be acquired afterwards.

1.9 Usability in NFC

NFC technology is declared as easy to use and simple in NFC Forum. In order to use NFC, all a user needs to perform is to hold NFC enabled devices together. In this way, users can

access services, set up connections, make payment, or use a ticket [1]. Up to now, only a few studies have performed usability analysis on NFC to measure the success of trials. Here we summarize some of the studies on NFC usability.

In [2] subjective usability of a student council voting is studied and compared with a web based voting scenario. NFC voting gained a higher usability than web based voting with a score of 82.75 whereas web based voting gained a score of 78.50 out of 100. The results of the usability test showed that NFC technology has the potential to increase the usability of systems. As a result, the rise of NFC compatible mobile phones and services will bring new opportunities to make our lives easier. In the context of voting, NFC provided a practical and easy to use environment.

Another study [3] also performed usability tests on NFC to identify how NFC based systems could be used to improve mobile solution workflows and usability.

The study showed that NFC can improve mobile workflows by solving different related problems. In the pilot cases, NFC technology dealt with the following problems:

- Access to real-time information, applications and instructions in the field;
- Real time updating of data;
- Removal of human errors;
- Reducing users' memory payload;
- Creating ability to verify people's presence in different locations.

The study concluded that NFC based solutions are easy to use, but the small and limited keyboard of mobile devices causes difficulties for the design of the models. NFC based solutions should take into account the place of the tags, ease of the application usage, and the amount of textual input. The study showed that user friendliness was taken into account in the pilots, but it did not always impact on the user experience.

Another study on user experiences and acceptance scenarios of NFC applications [4] showed that simple NFC technology must beat the alternative technologies in terms of user experience and performance criteria especially when both technologies provide nearly the same end-user functionalities.

1.10 Benefits of NFC Applications

Numerous NFC applications are designed, prototyped, and developed so far. Service providers are eager to offer NFC based services, however sometimes they do not decide which service to offer. A study [5] analyzed most of the NFC applications in the literature and highlighted the benefits of those applications from the users' perspective. In addition, the study gave possible future usage scenarios based on the discovered benefits.

The study initially identified that each operating mode provides different benefits to users [5]. Thus they analyzed applications based on the used operating mode.

In reader/writer mode, data stored in an NFC tag are read by an NFC enabled mobile phone and then it is used to process further operations. Transferred data can be any type of text, such as a web address, data of an event, or some other data. After transfer operation, the data can be used for many purposes (e.g., display information on a mobile device's screen on the go). Moreover, based on the design of the application, this mode is able to provide mobility and to

decrease physical effort. Increasing processing power and wireless Internet access of mobile devices also helped with this issue and made this mode more attractive. For example; patients can upload their medical information using NFC technology from their homes and elderly people can order their meals from their homes. Clients can shop from home by touching their mobile devices to NFC tags placed on brochures.

Many more applications using reader/writer mode are developed than other modes. The most important reason for such development is that there are so many interesting and easy to implement use case scenarios that can be developed in reader/writer mode. Also developments and implementations of reader/writer mode applications are relatively easier to implement than others.

Peer-to-peer mode is rare when compared with other modes, which is studied mostly for device pairing, social networking, and file transfer operations. Peer-to-peer mode provides easy data exchange between two devices and enables some social networking cases (e.g., updating presence information on social networks).

In the study, it is found out that card emulation mode is mainly concerned with eliminating the need for a physical object. For example, the usage of a mobile phone eliminates the need to carry a credit card, a debit card, or even cash. Instead, a user makes payment with her mobile phone. NFC usage eliminates the need to carry a physical key and contactless smart key. As NFC can be used to enter rooms instead of electronic keys, it provides access control. Moreover, card emulation mode is used while cashing in ticket and mobile coupons. Actually these two processes also achieved the elimination of physical objects (paper-based tickets, coupons and so on) . The most important features of card emulation mode are the elimination of physical objects and providing access control. Also, the study stated that the commercially available applications are mostly developed using card emulation mode.

1.10.1 Future Scenarios on NFC

The main benefits of the reader/writer mode are identified as increasing mobility and decreasing physical effort. These benefits are in accordance with the mobility property of the mobile phone which in turn generally decreases physical effort. For example, calling someone provides mobility and eliminates the need to communicate face to face. Moreover with the mobile services usage, e-mail applications are developed for mobile phones and these e-mail applications enable users to read and write e-mails without any geographical restriction. It is seen in the study that the majority of real life scenarios can be adapted to this mode's applications. Application designs should include the data transfer from an NFC tag to a mobile phone and displaying it to the user. Moreover mobile phones can do additional processing with transferred data (e.g., can store the data in the mobile phone, and can transfer the data to any server on the Internet).

It has been seen that the peer-to-peer mode's major benefit is exchanging data easily. Data exchange between two NFC devices provides the possibility of secure transfer of critical data and social interaction. Since NFC devices can transfer data within 4 cm, exchanging critical data can be one of the key future applications of this mode.

It is stated in [5] that the card emulation mode's main aim is to make the mobile phone tightly coupled to its users. This can be considered as a challenge to the mobility property of mobile phones, however people carry mobile phones with them most of the time, and the

coupling of mobile phones with the human body actually fits with the usage of mobile phones. In the near future, there may be the opportunity for people to carry NFC enabled mobile phones not only to gain mobility, but also to perform daily functions. Credit cards, keys, and tickets will be embedded into mobile phones. Hence, there will be more opportunities to integrate daily objects into NFC enabled mobile phones. In addition to the current usage areas, many objects such as identification cards, passports, fingerprints, and driver licenses can be stored in mobile phones and be employed by making use of NFC. As the mobile phone becomes part of human daily life, additional opportunities may also arise. One opportunity will be using NFC enabled mobile phones as a memory area for users' data. One of the most concrete examples of this usage is to store the user's patient information into an NFC enabled mobile phone.

1.11 NFC Throughout the World

There have been many attempts to realize NFC technology so far. Some models are developed by universities, others by companies, and even more by a joint effort between universities and companies. Many models are only theoretical, some cannot be used because of missing features but others are completely developed.

NFC city describes an area where several NFC applications are being used. The purpose of an NFC city may be either to test an implementation, or even to actually use one. NFC cities are important for the improvement of NFC technology, since they are the actual arena of a moderate sized usage media.

In NFC trials and projects, applications and the NFC ecosystem are tested; thus usability issues together with the problems of the technology can be obtained through the tests in the initial phase of NFC cities. NFC technology, applications and usability issues can be tested during this phase. During the testing period, one of the purposes is to evaluate the applicability of the NFC technology. However, it is not the only reason for this effort. One major aim is to test NFC ecosystem issues. The NFC ecosystem usability is at least as important as the technical appropriateness, since the model cannot be used when there is disagreement between the actors involved.

1.11.1 NFC Cities

1.11.1.1 City of Oulu

Oulu City was used as a test bed for the SmartTouch project. The project ran from 2006 to the end of 2008 and examined the role of NFC in city life, the home, wellbeing and health, entertainment, technological, and business building blocks. Citizens living in Oulu had the opportunity to test commercial and public NFC services as the first users of the technology over a broad aspect. The main applications that were tested are:

- Meal ordering for the elderly;
- Attendance process of students in a secondary school;
- Attendance process for a primary school using NFC;
- City orienteering for schools;
- Smart parking by eliminating coins and parking tickets;

- Retrieving news, and downloading videos from smart posters at theaters;
- Ordering lunch quickly at restaurants and obtaining coupons;
- Bus ticketing;
- Work time management and drivers' diary;
- Lock management in public sport halls;
- Information tags in the city environment;
- Future shop concept;
- NFC enabled blood glucose meter.

1.11.1.2 City of Nice

Nice is assumed to be the first NFC city in Europe with a commercial NFC roll out by every French MNO. The projects are conducted mainly on travel, m-tourism, healthcare, assisted living, m-payment, m-culture, m-government, m-education and fair trade. Implemented projects in Nice consist of all operating modes; reader/writer, peer-to-peer and card emulation modes. Also applications are stored and implemented on a single SIM card. The projects are developed in France, Morocco, Russia and Haiti with different partnerships and contracts.

These projects also involved the University of Nice hosting an important NFC pilot project in 2010–2011. The Nice Future Campus project's aim is to replace the physical student ID card with NFC enabled mobile phones and to enable multiple applications on a single SIM card. A student using this project was able to pay, manage tickets and coupons, share an opinion regarding a book, get contextual information, communicate with her friends and much more with her NFC enabled mobile phone within the campus. The main applications provided by NFC technology were for payment, the university restaurant ticket, library access, access control, location based services, social networking, and information services.

1.11.1.3 Smart Urban Spaces

Smart Urban Spaces (SUS) is a collaborative European project focusing on designing and adopting context aware services and e-city services based on NFC with the latest mobile and ubiquitous technologies. SUS is a three year project running from mid 2009 to mid 2012 and involves four countries: Finland, France, Spain and Greece. Oulu, Helsinki, Caen, Valencia and Seville are some of the cities included. The SUS project involves many organizations from all of the four countries. The main purposes of the SUS project are to provide a framework for adopting e-city services, technical and operational service analysis, interoperability analysis and testing other issues through pilots and trials. The e-city services included in the SUS projects can be categorized as transport, family and community, leisure, culture and sports, utility tools, education and learning, NFC city ecosystem, and other specific services.

1.11.2 NFC Trials and Projects

There have been various NFC trials and projects all over the world. Payment and ticketing applications are possibly the most well-known and promising everyday applications of NFC technology, and are the most complex from the ecosystem aspect as well. Thus, most of the tests

and trial projects are implemented in this application domain. Some of these projects have been completed or expanded into different application domains with growing participating entities or are still continuing. Some of the trials and projects are as follows:

- *Payez Mobile Project:* This is a joint initiative launched in November 2007. It is a mobile payment service pilot implemented with about 1000 testers and 500 retailers in Caen and Strasbourg. The global objective of the participants in this trial is to create a common vision, business solution for banks and MNOs in the contactless payment application domain.
- *C1000 NFC Pilot with Rabo Mobile in the Netherlands:* The Dutch based Rabobank has become the first bank in Europe to introduce mobile banking and low-cost calling services in a different way with Rabo Mobile (originally named Rabo Mobiel). It is a MVNO that is fully owned by Rabobank. Rabo Mobile initiated a new NFC pilot called 'Pay with your mobile phone at C1000' in the Netherlands. C1000 is one of the largest Dutch based supermarket chains. A number of NFC enabled applications in C1000 retail stores including mobile payment, and loyalty services were implemented over 6 months. Moreover, customers can bring their empty bottles and receive discount receipts to be used at the checkout from the bottle machines which are located within the supermarket or they can have a refund credited to their Rabobank accounts.
- *NFC Stadium experience in Manchester:* Manchester City Football Club and Orange UK provided an NFC enabled ticketing application. The fans are allowed to use their NFC enabled mobile phones to touch to the NFC readers at the stadium gates and enter through turnstiles to attend home games easily.
- *Bouygues Telecom trials in Paris:* France's major MNO Bouygues Telecom, RATP and SNCF who are the providers of Navigo contactless transit fare cards performed a 3-month NFC enabled transit ticketing trial in Paris. This trial's aim was to enable users to pay their fares at gates or at readers on buses which accept the Navigo ticketing application using their NFC enabled mobile phones.
- *O2 Wallet:* Telefonica O2, as one of the largest MNOs, announced O2 Wallet in November 2007 and performed a 6-month trial with various service providers. The O2 Wallet pilot paves the way for large usage of mobile phones as Oyster cards for travel around London, pay for purchases by Barclaycard, or access events. This application eliminates the need for users to carry Oyster smart cards in their wallets. Users can pay for their travel expenses through the Oyster application by simply touching their mobile phones to the Oyster NFC readers at London underground tube stations, on buses, and on trams.
- *London Fashion Week:* One of the largest MNOs in Europe, Telefonica O2, organized and performed a small trial at London Fashion Week which is the key event for designers in London to show their designs to fashion buyers throughout the world. The aim of this trial was to provide fashion buyers an opportunity to give instant feedback on the collection of designer Emilio de la Morena. This NFC enabled messaging trial was performed with a limited number of users.
- *Pass and Fly in Nice Airport:* Pass and Fly was a joint project of Nice Cote d'Azur Airport and Air France in partnership with Amadeus and IER. This pilot was launched in April 2009 and lasted for 6 months in Nice Cote d'Azur Airport. The aim of the pilot was to enable passengers to download digital boarding passes to their mobile phones using NFC technology.

1.12 Status of Academic Research on NFC Literature

Today, NFC has become an attractive research area for many academicians due to its exploding growth and its promising applications and related services. Due to its nature, a large proportion of NFC research can be represented as design science research which aims to propose an innovative design artifact and has problem relevance and a rigorous nature. A study on NFC [6] provides a rigorous academic review on NFC literature. For the last few years, there has been a considerable increase in the number of research papers and activities concerning NFC. However, a better understanding of the current status of the NFC research area through an academic review of literature is necessary to maintain the advancement of knowledge in NFC research and to identify the gap between theory and practice.

The conducted survey is based on articles in journals and mostly conference proceeding papers. The study exclude master theses, doctoral dissertations, textbooks, unpublished working papers, white papers, editorials, news reports, and book reviews. Researchers and practitioners often use journal papers to acquire information and to disseminate new research findings, thus most of the existing literature reviews exclude conference proceeding papers too. However, conference papers are not excluded which provide "the high level of research, both in width and breadth" after journals.

(i) *NFC research framework*

In accordance with the study, an NFC research framework is proposed which includes a content-oriented classification of the NFC literature. This framework classifies the NFC academic literature in four major categories and signifies the bidirectional relationships between categories: NFC Theory and Development, NFC Infrastructure, NFC Applications and Services, and NFC Ecosystem (see Figure 1.2). Most of the papers in NFC research can be considered as design science research which provides an innovative, purposeful design artifact in the form of a construct, a model, a method, or an instantiation. The design artifact has to solve a specific problem or develop technology based solutions.

The NFC Theory and Development level is the fundamental level of the NFC research framework. It includes the studies related to the development of NFC technology and applications; "Overviews, Context and Foundations" includes general introductions, assessments, reviews about NFC, foundations and standards on NFC technology, performance analysis and measurements and new guidelines for the development of NFC enabled applications or services, and "Policy, Legal and Ethical Issues" includes security and privacy issues, regulations, and legal requirements. The papers in this category focus on more behavioral issues and behavioral sciences which seek to develop and justify theories, rather than developing a design artifact. Papers dealing with this level influence upper levels that focus on design science in NFC research.

The NFC Infrastructure level is the intermediate level that includes studies related to "Network and Communication" issues (e.g., data aspect, new communication protocols, and OTA transactions), hardware issues dealing with "Tags, Antenna, Reader and NFC Chip", "Security and Privacy" issues (e.g., vulnerability analysis, availability, confidentiality, integrity, authentication, authorization, and non-repudiation) that focus on developing design artifacts rather than behavioral issues. This layer is positioned with pre-defined business related to existing technology infrastructure, applications, and the existing ecosystem. The proposed framework shows the direct linkages of NFC

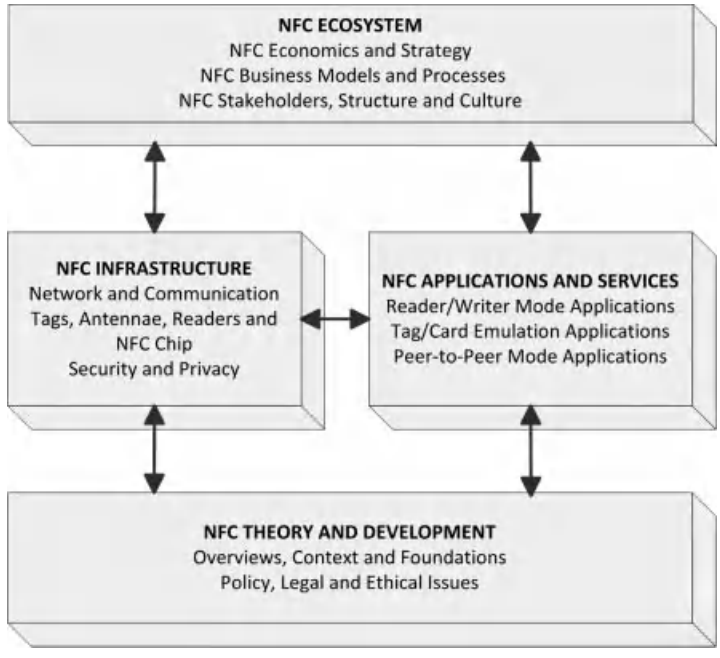


Figure 1.2 Classification framework for NFC research.

Infrastructure with other categories. Moreover, NFC Infrastructure related research facilitates new business needs.

NFC Applications and Services is the other intermediate level which is influenced by the other three categories. The papers within this category provide a problem space or new business needs. NFC technology covers a wide range of applications that provide real implementations or prototypes with rigorous design artifact evaluations such as experimental, testing, field studies and so on. The papers in this category are also grouped according to the application's operating modes: "Reader/Writer Mode Applications", "Card Emulation Mode Applications" and "Peer-to-Peer Mode Applications". Indeed, design artifacts which propose applications or services operating in two or more modes can be seen in NFC literature.

The NFC Ecosystem is the highest level of the NFC research framework. This level is part of the problem space or environment of NFC research, the improvements or changes in the middle and fundamental layers affect NFC Ecosystem significantly. The papers within this category are grouped under "NFC Economics and Strategy", "NFC Business Models and Processes" and "NFC Stakeholders, Structure and Culture". The first two groups deal with the business requirements, analysis and managerial sides of the NFC technology. The third group deals with the social sides of NFC technology such as roles, characteristics and capabilities (e.g., user acceptance, usability, adoption, reliability, and manageability) of stakeholders (e.g., MNOs, service providers, and end users), and culture of NFC enabled services. Stakeholders play a crucial role in facilitating NFC research and development. In an NFC ecosystem, the goals, tasks, problems, and

opportunities define business needs as they are perceived by the stakeholders. These perceptions are shaped by the roles, capabilities and characteristics of the stakeholders evaluated within the context of economics and strategies, structure and culture, and business models and processes.

(ii) *Evaluation of the academic review*

NFC as a new emerging research area has attracted the attention of both practitioners and academicians. As cited earlier, academic research activities in the NFC area have significantly increased since 2006. This literature review wants to shed light on the current status of NFC research. The results from the NFC classification scheme have several important implications:

- There is a clear need for more journal publications to provide business related and rigorous research papers on NFC technology.
- It is not surprising that most of the academic research papers are related to NFC Applications and Services, especially operating in reader/writer mode. The reason for such interest on this mode is that development and implementation of such services or applications are much easier than developing applications operating in other modes. Unfortunately there are only a few research papers on "Peer-to-Peer Mode Applications".
- Another large proportion of the papers are related to NFC Infrastructure. Our review shows the importance of focusing on technical issues of a new technology again, rather than issues related to realizing economics, business values or strategies for NFC development, dissemination, and marketing. More specific research study needs to be conducted on business issues, and economics of NFC technology.
- When developing a NFC service, the ecosystem and business environment of that NFC service need to be considered as well. Such new applications can bring new business models and processes with new players. In particular, the capabilities, characteristics and roles of stakeholders need to be evaluated and modified when necessary, in order to satisfy the requirements of new business models and processes. Cultural differences on adopting NFC enabled technologies could be an interesting area for investigation.
- Policy, ethical and legal problems which can be referred to as behavioral issues were other important and demanding research areas for development of a new, emerging technology. However, it is hard to find papers dealing with the public policy and legal problems (e.g., taxation problems, trust, fraud, privacy issues for Internet privacy, and financial privacy). Indeed, this should prompt academicians to investigate this area.
- Today it seems that the most popular NFC related research areas are on developing new NFC enabled applications and developing NFC infrastructure. Thus, NFC research can be mostly referred to as design science research that proposes innovative artifacts and provides utility for relevant business problems. In the meantime, utility and efficiency of the proposed artifact must be demonstrated in well-defined methods. Most of the papers use more descriptive (e.g., scenarios, and use cases to demonstrate its utility) or analytical (e.g., architecture analysis) methods while developing an application or service, rather than performing experimental and testing methods. In some papers, inadequate design evaluations are performed, or implications of the proposed design artifacts are observed. Instead of evaluations through scenarios or use cases, field studies, controlled experiments, or simulations will be more useful for representing the proposed artifact rigorously.

(iii) *Further research in NFC*

The literature review presented in [6] aims to provide a holistic review and a comprehensive base for understanding of NFC research. In addition to the evaluations, some academic research questions for further research in NFC are proposed:

- Are there any public policies, regulations and legal standards for the development and adoption of NFC technology at the individual and corporate level?
- How are the required NFC standards developed from policy, regulations and legal points of view?
- What are the impacts on the adoption and acceptance of NFC applications on the user side?
- What are the possible implications of cultural differences on adoption of NFC technologies and new business opportunities?
- How can developments in NFC technology as an information technology tool be evaluated in terms of economic performance and economic decision rules?

1.13 Chapter Summary

NFC as a promising research and development area has attracted the attention of both academicians and practitioners in the last decade. NFC technology aims to simplify daily human life by a simple touch and in the very near future, people will be able to benefit from several services by using their NFC mobiles. They will be able to purchase goods and services, access hotel rooms, apartments or cars, configure Wi-Fi and Bluetooth settings, upload health related data to hospital monitoring systems, and so on.

The underlying layers of NFC technology follow globally accepted standards of ISO, ETSI, ECMA and so on as well as the specifications of industry pioneers. NFC Forum is the most crucial association in advancement of the technology.

Although interoperable sets of standards and specifications are currently available, there is still a lack of common understanding of the need for collaboration. Since the profit within the market is very big, companies are ambitious to share this profit. Hence this creates a strong rivalry among MNOs, financial institutions, transport authorities, IT firms and so on which affects the progress of NFC ecosystem and business models. Up to now, various NFC trials and usability studies especially in the payment and ticketing service domain have been conducted throughout the world. Some led to commercial launches whilst others did not, due to failing to generate consistent business models.

This chapter provides a good introductory knowledge on NFC technology from various perspectives including technical, business, usability, and so on. It also presents valuable academic studies on the benefits of NFC applications and the status of academic research in NFC literature.

References

[1] NFC Forum, http://www.nfc-forum.org/ (accessed 10 July 2011).

[2] Ok, K., Coskun, V., and Aydin, M. N. Usability of Mobile Voting with NFC Technology. Proceedings of IASTED International Conference on Software Engineering, Innsbruck, Austria, 16–18 February 2010, pp. 151–158.

[3] Jaring, P., Törmänen, V., Siira, E., and Matinmikko, T. Improving Mobile Solution Workflows and Usability Using Near Field Communication Technology. Proceedings of the 2007 European Conference on Ambient Intelligence, Darmstadt, Germany, 7–10 November 2007, pp. 358–373.

[4] Franssila, H. User Experiences and Acceptance Scenarios of NFC Applications in Security Service Field Work. Proceedings of the 2010 Second International Workshop on Near Field Communication, Monaco, 20–22 April 2010, pp. 39–44.

[5] Ok, K., Coskun, V., Aydin, M. N., and Ozdenizci, B. Current Benefits and Future Directions of NFC Services. Proceedings of 2010 International Conference on Education and Management Technology (ICEMT), Cairo, Egypt, 2–4 November 2010, pp. 334–338.

[6] Ozdenizci, B., Aydin, M. N., Coskun, V., and Ok, K. NFC Research Framework: A Literature Review and Future Research Directions. Proceedings of 14th International Business Information Management Association Conference on Global Business Transformation through Innovation and Knowledge Management, Istanbul, Turkey, 23–24 June 2010, pp. 2672–2685.

2

Towards NFC Era

Near Field Communication (NFC) is a technology that simplifies and secures the interaction with the NFC capable devices around us. It integrates Radio Frequency Identification (RFID) technology and contactless smart card interface with mobile phones. The NFC concept arises from the synergy of several technologies including wireless communication, mobile devices, mobile applications, RFID communication and smart cards. Server side programming, web services, and XML technologies also contribute to the fast improvement and spread of NFC by enabling online services over the Internet. Many familiar items, such as credit cards, car keys, and hotel room access cards will eventually cease to exist because an NFC enabled mobile phone will suffice to provide all of their functionalities.

NFC technology has already emerged and is being used more and more commonly. You will notice this clearly while reading this book, which explains NFC using a comprehensive approach from the concept and technological components to its application areas and procedural issues. In this chapter, we start with a short background history that elaborates how all the technologies mentioned above paved the way for the development of NFC.

2.1 Ubiquitous Computing and NFC

Currently, NFC is one of the enablers for ubiquitous computing. Therefore the origin of the idea is closely related to ubiquitous computing. In order to understand the background of NFC, we need to start with the history of ubiquitous computing.

2.1.1 Ubiquitous Computing

Automated calculation and programmability are the essence of modern computers and therefore ubiquitous computing. The history of modern computers includes the work of pioneers over almost two hundred years, such as the textile loom, the punched paper cards by Joseph Marie Jacquard which resulted in the Jacquard loom, a fully programmable mechanical computer by Charles Babbage in 1830s, data recording on a machine readable medium by Herman Hollerith in the late 1880s, the formalization of the concept of algorithms and computation

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.
 Published 2012 by John Wiley & Sons, Ltd.

with the Turing machine by Alan Turing in 1930s, and the first electronic computers in the mid-twentieth century which required big rooms full of devices.

Personal computers are the one important step that came after these developments, changing the way that a user interacts with computers by using keyboards and monitors for input and output. The mouse further improved the interaction between humans and computers because it enabled users to input spatial data to a computer. The hand became accustomed to holding the mouse, and the pointing finger became accustomed to click it. Three dimensional (3D) mice, joysticks, and pointing sticks were other forms used to input data. The movements of the pointing device are echoed on the screen by the movements of the cursor, creating a simple and intuitive way to navigate a computer's GUI.

Touch screens changed the form of interaction dramatically as well. They decreased the interest in all earlier input devices, as the hands could move more freely on the new input device of the computer. In the meantime, mobile phones had been introduced, initially for voice communication. Early forms contained a keypad, with less number of keys than regular keyboards, since regular keyboards would not easily fit in such a small area. Mobile phones with touch screens can be assumed to be the state of the art technology as the same screen is used as both input and output unit.

Ubiquitous computing refers to the next level of interaction between humans and computers, where computing devices are completely integrated into everyday life and the objects around. Ubiquitous computing is a model in which humans do not design their activities according to the machines they need to use; instead, the machines change their forms to adjust to human needs. Eventually, the primary purpose is that humans using machines will not need to change their daily behaviors and also will not even notice that they are performing activities with the help of machines.

2.1.2 New Communication Interface Alternative for Mobile Phones: NFC Technology

As with modern computers and interfaces, the increasing mobility of computing devices provided by mobile communications is also an important step in the development of ubiquitous computing capabilities and NFC. Mobile phones already had several communication options with external environments before the introduction of NFC. When mobile phones were initially introduced, their primary goal was to enable voice (and lately data) communications among mobile phones. GSM communication enabled functionality of mobile phones for several services, such as voice communication, Short Messaging Service (SMS), Multimedia Messaging Service (MMS) and limited Internet access. One immediate communication option for mobile phones with other devices was cabled data transfer with computers.

Bluetooth technology was introduced later to create personal area networks that connect peripherals with computing devices including mobile phones. Bluetooth became very popular in the early 2000s. Perhaps the most widely used function of Bluetooth is data exchange among mobile phones and between a mobile phone and another Bluetooth enabled device such as a computer. Bluetooth enables communication between devices in the vicinity. However secure transactions could not be performed with this technology since it is designed for wireless communication within tens of meters, which allows malicious devices in the vicinity to tap the communications among the Bluetooth nodes.

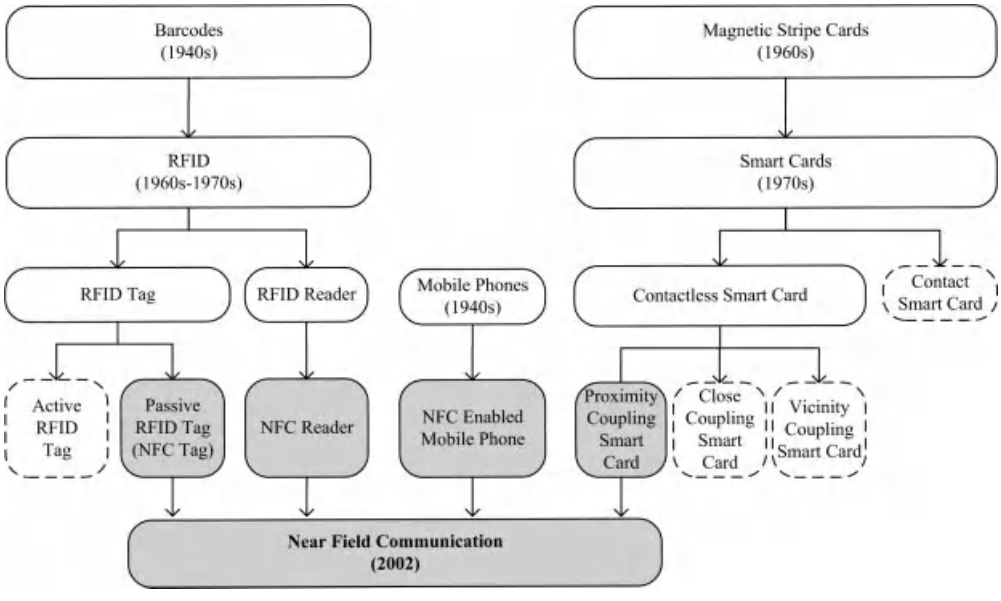


Figure 2.1 Evolution of NFC technology.

The main motivation for the NFC vision is the integration of personal and private information such as credit card or debit card data into mobile phones. Because of the sensitivity of the exchanged information, security is the most important concern, and the wireless communication range provided even by RFID technology is considered too long. Shielding is necessary to prevent unauthorized people from eavesdropping private information because even non-powered, passive tags can be read over 10 m. This is where NFC comes in.

NFC operates between two compatible devices over a very short communication range. Communication is performed by an NFC mobile on one side and on the other side an NFC tag (a passive RFID tag), an NFC reader, or an NFC mobile. RFID is capable of accepting and transmitting beyond a few meters and has a wide range of uses. However, NFC is restricted to within very close proximity and is also designed for secure data transfer. Currently, integration of NFC technology into mobile phones is considered the most practical solution for NFC because almost everyone carries a mobile phone.

The evolution of NFC technology is illustrated in Figure 2.1. In essence NFC makes use of RFID technology as well as is compatible with contactless smart card interfaces. The gray areas in Figure 2.1 are the technological developments that support the NFC environment directly. The technologies that make NFC evolution possible are overviewed briefly within this chapter. Learning the technological background of NFC will be helpful for the reader to understand subsequent chapters.

2.2 Mobile Phones

A mobile phone is an electronic device which is primarily used to make telephone calls while the user is mobile. Mobile phones are simply referred as mobiles. The user of the mobile

must be registered to a mobile telephone network where the service is provided by a Mobile Network Operator (MNO). The call can be made to or received from any other phone which is a member of either the same or another mobile phone network, a fixed-line network, or even the Internet. Mobile phones support the anytime, anywhere motto.

2.2.1 Features of a Mobile Phone

Mobile phones are very convenient to use and handy. Therefore in addition to the phone call capability, a vast amount of additional services are bundled to it, and many new future services are still on the way. Some of the currently supported services are as follows:

Wired communication services

- USB;
- PC synchronization.

Wireless communication services

- GSM communication;
- SMS (Short Messaging Service or Text Messaging);
- MMS (Multimedia Messaging Service);
- RDS radio receiver;
- Global Positioning System (GPS) navigation;
- Short range wireless communications; Infrared, Bluetooth, NFC, and so on;
- Wi-Fi connectivity;
- Electronic mail;
- Internet access.

Integrated services

- Storing contact and communication details;
- Scheduling;
- Calculator;
- Gaming;
- Photography;
- Music (MP3) and video (MP4) playback;
- Alarms;
- Memo recording;
- Personal Digital Assistant (PDA) functions;
- Camcorder (video recording);
- Secure Memory Card (SMC, MSD, SD, and so on).

Figure 2.2 shows the major wireless services currently enabled by mobile phones: GPS Navigation, Wireless Internet services, GSM, Bluetooth, Wi-Fi, and NFC technologies. GPS technology, as a space based global navigation satellite technology, is one of the most popular services used by mobiles. It provides reliable location information in almost all weather

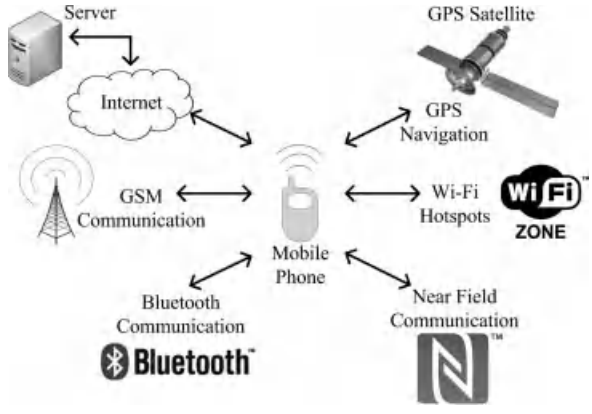


Figure 2.2 Major wireless services enabled by mobile phones.

conditions and at all times all over the world. Since a lot of people use smart mobile phones or PDAs, and navigation systems can run on those devices, the demand for and use of navigation systems are increasing. Wireless Internet services can be enabled in a variety of ways; through Wi-Fi, cellular networks and so on. GSM, Bluetooth, and Wi-Fi wireless technologies are presented in Section 2.3.2.

2.2.2 Mobile Phone Network

The mobile telephone network consists of a MNO, Base Stations (BSs), Mobile Switching Center (MSC), and mobile phones (see Figure 2.3). The mobile transmits a radio signal, which is captured by a nearby BS if there is one. If the signal is caught by a BS, than the mobile can access the network, otherwise the user is informed that the mobile is out of network

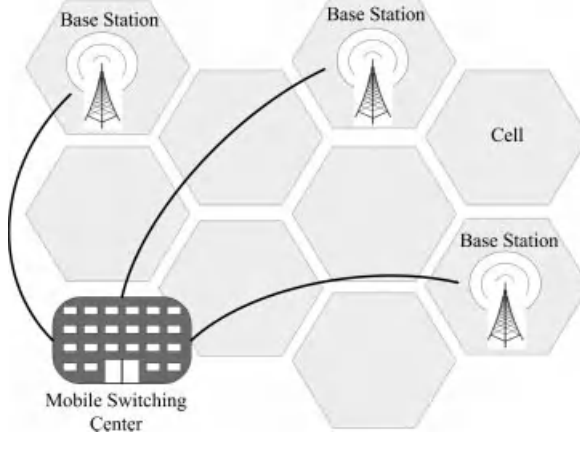


Figure 2.3 Mobile phone network.

coverage. The mobile within the coverage of the network via any BS can then receive any provided service that is requested by the user. In the case of a phone call to another mobile, the requested phone must be within the coverage area of the same or another BS, otherwise the call cannot be processed. The circuit that is built on the fly between the two mobiles might include ground communication legs in addition to the mobile communication legs.

Each BS covers and thus maintains the first (and last, depending on the direction of the call) communication leg between the mobile phone and the mobile phone network. The area that is maintained by each BS is called a cell. The size of the cells technically depends on the receiving and transmitting range, or which mainly depends on the power of the BS. Smaller cells are used in areas where there is a big demand which is mainly in crowded urban areas, and vice versa for a good Quality of Service (QoS).

In cellular telecommunications, handover (handoff) is the process of transferring an ongoing call or data session from one core network channel to another. In satellite communications, it refers to the process of transferring satellite control responsibility from one earth station to another.

When a mobile is on the move during a call, the connection is maintained by the BS of the cell that the mobile is moved into. If the mobile is out of coverage area of a BS, or in other words out of a cell during a conversation and there is no coverage by a BS in that location, then the connection cannot be maintained, so the call is broken. There are algorithms to carry on the conversation with a high quality so that the people at both ends of the conversation are not aware of the fact that there is a handover during the cell passage.

2.2.3 Mobile Phone Architecture

A mobile consists of several components, each of which fulfills a specific need:

- A microphone captures voice for conversion from analog to digital mode.
- A display unit shows call, phone, signal, and network information.
- An input mechanism allows the user to interact with the phone. The two most common input mechanisms are keypads and touch screens.
- A rechargeable battery supplies power for the cell phone.
- A Universal Subscriber Identity Module (USIM) card allows access to a network that is managed by a MNO.
- An antenna allows the reception and transmission of signals between the mobile and the BS.
- A Digital Signal Processor (DSP) takes a digital signal and processes it for further use.
- A microprocessor coordinates almost all functions on the board.
- A memory unit stores all types of data such as telephone numbers, conversation records, and message information.
- A USIM card acts as a smart card that stores the subscriber identity information. It holds the details of the subscriber, security data which allows access to the secured services, and memory to store personal data such as telephone numbers.

The USIM card includes a microchip that contains the required information and access keys to access a mobile phone network managed by a MNO.

The reason for providing a mobile USIM card instead of embedding it in the mobile device is to provide flexibility and portability for the user. A user may just switch the USIM from a mobile to another while being able to connect to the same mobile network provider without any additional procedure.

USIM is managed by a mobile operating system. Currently a wide range of options exist, and the market share of the operating systems differs as time passes.

2.3 Wireless Communication as a Communication Media for NFC Technology

Wireless communication refers to data transfer without using any cables. When communication is impossible and impractical by using wires or cables, wireless communication is preferred. The range may vary from a few centimeters to many kilometers. Wireless communication devices include various types of fixed, mobile, and portable two way radios, cellular telephones, PDAs, GPS units, wireless computer mice, keyboards, headsets, satellite television and cordless telephones, and allows communication without requiring a physical connection to the network.

Wireless communications introduce challenges that are somewhat harder to handle compared with wired communications such as interference, attenuation, unreliability, higher cost, and lower security. The term wireless refers to the transmission of data over electromagnetic waves, which uses the electromagnetic spectrum as depicted in Figure 2.4.

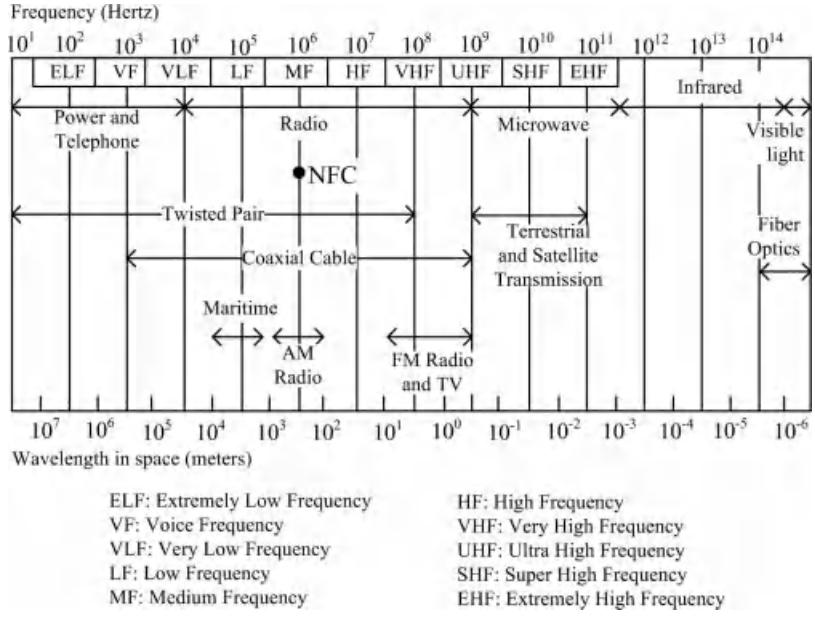


Figure 2.4 Electromagnetic spectrum for communication.

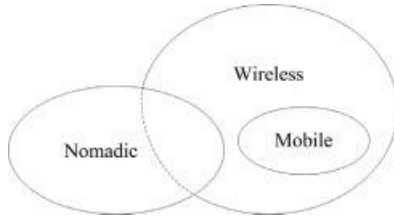


Figure 2.5 Wireless, mobile and nomadic communication.

2.3.1 Wireless, Mobile, and Nomadic Communication

Wireless is generally mobile, and mobile is essentially wireless at least in one link as shown in Figure 2.5. An example of wireless but not mobile is the desktop which uses wireless communication to connect to the Wireless Local Area Network (WLAN). We also distinguish nomadic communication from mobile. In a nomadic communication, a node moves in between communication, but does not move during a communication. An example of nomadic communication is the laptop. If mobility is not needed during a communication, wireless or wired communication can be used although the laptop can be moved to anywhere in the world. The capability of being mobile during communications can only be provided by a wireless medium.

An end-to-end communication between two wireless devices does not need a completely wireless path. A wired infrastructure can be used to complete the path. Only the first communication leg between the mobile user and one terminal of the network must be wireless, whereas the remaining part might be still based on wired connections, and thus a wired network. The architecture is called a cellular network or infrastructured wireless network. Alternatively when all the links are wireless and there is not a pre-engineered infrastructure, the architecture is classified as an ad hoc or infrastructureless wireless network. If you consider a GSM network for example, the mobile phone performs a wireless communication with the BS that it is connected to, but most of the remaining communication between the BS and the network servers use wired communication thereafter. Therefore, GSM is based on a cellular architecture. In the wireless sensor network case, the wireless sensor nodes build a wireless network among themselves as well as with the sink, hence they create an ad hoc network.

2.3.2 Wireless and Mobile Communication Technologies

The most straightforward benefit of wireless communication is mobility, which indeed has a big impact on our daily lives. Mobile communications do not support only the productivity and flexibility of organizations but also support the social life of individuals because people can stay continuously connected to their social networks. Widely used wireless technologies include GSM, Bluetooth, Wi-Fi, WiMAX, and ZigBee:

(i) *Global System for Mobile (GSM) communications*
GSM is one of the widely used mobile communication technologies. GSM uses a bidirectional, full duplex connection between two devices. It is a connection oriented system

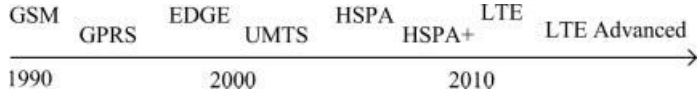


Figure 2.6 Evolution of GSM technology.

that provides highly reliable functionalities and reduces terminal power consumption considerably compared with the earlier generations of mobile communication.

The evolution of GSM based services are depicted, depending on their date of introduction to the market, in Figure 2.6. This is the widest variety of any wireless technology. The evolution starts from 2G GSM and General Packet Radio System (GPRS) to 3G Enhanced Data for GSM Evolution (EDGE), Universal Mobile Telecommunication System (UMTS), and High Speed Packet Access (HSPA).

There are several features which make GSM so popular among operators and their customers. It enables data support including SMS and web browsing, clear voice quality, and international roaming with services available in more than 200 countries. For more detailed information visit the website (http://www.gsm.org).

(ii) *Bluetooth*

Bluetooth is an example of a Wireless Personal Area Network (WPAN) based on IEEE 802.15 standard. It is an open wireless technology standard. Any Bluetooth enabled device can connect to another Bluetooth enabled device in the proximity using a radio technology called a frequency hopping spread spectrum. It uses a 2.4 GHz short range radio frequency (RF) band.

Bluetooth enabled devices can communicate using short range piconets. Piconets allow one master device to connect up to seven active slave devices. When a Bluetooth enabled device enters radio proximity, piconets are established automatically. Hence, you can easily connect in everywhere.

The most commonly used Bluetooth applications are data transfer between a mobile phone and a Bluetooth enabled device, input and output devices communicating with a PC, and a mobile phone headset communicating with a mobile phone base. Today Bluetooth is managed by the Bluetooth Special Interest Group. For more detailed information visit the website (http://bluetooth.com).

(iii) *Wi-Fi*

Wi-Fi is one of the popular wireless technologies and is a superset of IEEE 802.11 standard which addresses Wireless Local Area Networks (WLANs). The Wi-Fi trademark is owned by Wi-Fi Alliance. Wi-Fi is a wireless version of an Ethernet network. It uses RFs similar to Bluetooth, but with higher power. This actually results in better and faster connection within long range.

Currently, it is possible to use Wi-Fi everywhere. It is installed in many PCs, MP3 players, smart phones, printers, video game consoles, and laptop computers. These devices can easily connect to the Internet as they are in the range of a wireless network that is connected to the Internet. The size of the range depends on the number of interconnected access points. For more detailed information visit the website (http://www.wi-fi.org).

(iv) *WiMAX*

WiMAX (Worldwide Interoperability for Microwave Access) is a telecommunications protocol that provides fixed and fully mobile Internet access. WiMAX Forum describes

Table 2.1 Overview of some wireless technologies

Wireless Technology	Operating Frequency	Data Rate	Operating Range
UMTS	900, 1800, 1900 MHz	2 Mbps	Wide range
EDGE	900, 1800, 1900 MHz	160 kbps	Wide range
GPRS	900, 1800, 1900 MHz	160 kbps	Wide range
802.16 WiMAX	10–66 GHz	134 Mbps	1–3 miles
802.11b/g Wi-Fi	2.4 GHz	54 Mbps	100 m
802.11a Wi-Fi	5 GHz	54 Mbps	100 m
802.15.1 Bluetooth 2.0	2.4 GHz	3 Mbps	10 m
802.15.4 ZigBee	2.4 GHz	250 kbps	70 m
NFC	13.56 MHz	106, 212, 424 kbps	0–4 cm
RFID	125–134 kHz (LF) 13.56 MHz (HF) 400–930 MHz (UF) 2.5 GHz and 5 GHz (microwave)	1–200 kbps	20 cm for passive 400 cm for active

WiMAX as "a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL". For more detailed information visit the website (http://www.wimax.com).

(v) *ZigBee*

ZigBee is a wireless mesh networking standard. It is a low cost, low power and low rate technology based on the IEEE 802.15.4 (Low Rate Wireless Personal Area Networks) standard. ZigBee is simpler and less expensive than other WPAN technologies such as Bluetooth. ZigBee enables applications and services such as remote control, door control, and lighting control that require a low data rate and long battery life. With reliable wireless performance and battery operation, ZigBee provides freedom and flexibility for users. For more detailed information visit the website (http://www.zigbee.org).

Table 2.1 gives a brief summary of popular wireless technologies – according to their operating frequency, operating range and data rate – currently used throughout the world and integration of another two wireless technologies in this context. These two other major wireless technologies are RFID and NFC which is an extension of short range RFID. The details of RFID technology are explained in the next section.

In Table 2.1 GPRS, EDGE, and UMTS technologies represent Wireless Wide Area Networks (WWANs). These technologies are followed by the WLAN with different operating frequencies and range. Then the Wireless Personal Area Network (WPAN) technologies follow such as ZigBee and Bluetooth. NFC has the shortest communication range when compared with other technologies.

2.4 RFID Technology

RFID is a communication technology to exchange data between an RFID reader and an electronic RFID tag (label) by using radio waves (see Figure 2.7). These tags are traditionally

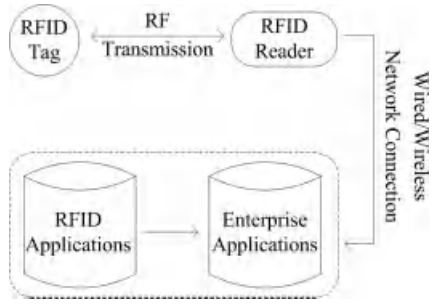


Figure 2.7 Architecture of an RFID system.

attached to an object, mostly for the purpose of identification and tracking. The data transmission results from electromagnetic waves, which can have different ranges depending on the frequency and the magnetic field. RFID readers can read/write data from/to RFID tags.

The connection between RFID readers and the RFID host application uses wired or wireless networks. In the backend system, the RFID application is assigned specific information. RFID tags generally contain an integrated circuit (IC) and an antenna. The IC provides data storage and processing, modulating and demodulating an RF signal, and other functions. The antenna enables the signal to be received and transmitted. Tags, readers and transmission details about the RFID system's components are explained later in this section.

The IFF (Identify Friend or Foe) system was the first common use of RFID technology in the Second World War used to distinguish friendly aircraft from foe. Commercial usage of RFID technology dates back to 1960s and 1970s with door key opening systems. Technology advances in a variety of different fields (e.g., computers, radio, radar, supply chain management, transportation, quality management, and engineering) have made RFID technology more useful with applications in asset management, payments, ticketing, livestock tracking, and transportation.

2.4.1 Earlier Form of RFID: Barcode Technology

A barcode is a way of representing data which is readable by an optical machine or scanner. Barcodes contain data about the object to which it is attached to. Barcodes are scanned by optical scanners called barcode readers. Early barcodes represent data by varying widths and the spacing of parallel lines. These are referred to as linear or one-dimensional (1D) barcodes. Later they evolved into rectangles, dots, hexagons and other geometric patterns in two dimensions. Although two-dimensional (2D) barcodes use a variety of symbols other than bars, they are still referred to as barcodes.

Barcodes have small storage capacity and they are easy and cheap to produce. The linear barcode range is low, but the 2D barcode range is much higher. Linear barcodes are typically used to identify a group of items, such as candy bars of some specific brand, but are not used to identify each specific candy bar. Hence, all candy bars of the same brand contain the same barcode, even in different department stores, different cities and countries. 2D barcodes have larger data storage capacity than linear barcodes, hence may contain a unique key value for each

Figure 2.8 Example of a UPC-A Barcode.

specific item. For example, 2D barcodes in a medicine box may contain specific identification for that medicine box, so patients and medicines can be tracked from a centralized database. Some major examples of linear barcodes are UPC (Universal Product Code) and EAN-13 (European Article Number) barcodes. The QR (Quick Response) Code Barcode is an example of a 2D barcode.

(i) *UPC-A Barcode*

The UPC Barcode allows the unique identification of a product and its manufacturer. UPC-A is a well-known type of UPC and is a 12 digit code as shown in Figure 2.8. The first six digits represent the number system character and the manufacturer of the labeled item. The next five digits identify uniquely the product and the 12th digit is a check character.

UPC encodes 12 decimal digits as SLLLLLLMRRRRRRE. The S refers to the start which is a prefix, and the E refers to the end which is an error checking character. The S and E are the bit pattern 101. The M character that is located in the middle is the bit pattern 01 010. S, M, and E are the guard bars of the UPC Barcode. The L (left) and R (right) are digits that are represented by a seven bit code. The UPC code encodes 84 digits and 11 digits as guard bars.

(ii) *EAN-13 Barcode*

The EAN-13 Barcode refers to the European Article Number-13 Barcode, which has now been renamed the International Article Number. It uses 13 digits made up of 12 data digits and a check digit. The EAN-13 barcodes are used worldwide for marking products often sold at retail point of sale. The numbers encoded in EAN-13 barcodes are product identification numbers including the country, manufacturer, and product codes. A sample EAN-13 Barcode is shown in Figure 2.9.

(iii) *QR Code Barcode*.

The QR Code Barcode was designed by Denso in Japan. It is a 2D general purpose matrix designed for fast information scanning. The QR Code Barcode symbol is square in shape and has dark and light squares in three corners of the symbol which enables them to be easily identified. A sample QR Barcode is shown in Figure 2.10.

Figure 2.9 Example of a EAN-13 Barcode.

Figure 2.10 Example of a QR Code Barcode.

2.4.2 *Barcodes* vs. *RFID Tags*

It is important to say that RFID tags can be identified as a complement but not a substitute for barcodes. Both of them have different advantages and drawbacks.

Let us consider barcodes first. Barcodes are printed on paper or a sticker which needs to be visible. Since papers or stickers are used for creating barcodes, they can be generated and distributed easily. The barcode scanner is used to easily read the sticker. Barcodes are however unsuitable in some applications. They can easily get damaged or worn out and can be unsuitable for applications that require high security. A barcode scanner reflects a light on the black bars of the barcode and the read data is converted to its numeric equivalent for further processing. Positioning the scanner on the barcode and scanning each item is a somewhat difficult process.

In the case of RFID systems, RFID tags do not suffer from any visibility issues. Radio waves can easily get the data stored on RFID tags; there is no requirement for line of sight for reading and getting the data. The data stored within the tag can be encrypted in order to prevent malicious copying. Thus RFID tags provide more security than barcodes. However, the major drawback of RFID tags is their cost.

Another important issue is that the RFID tag's data capacity is sufficiently large so that each individual tag can provide a unique code and a unique meaning. Hence, with the uniqueness of RFID tags, each individual product can be simply tracked when it moves from one location to another. Keeping track of products is an important feature of RFID tags. This is very useful for handling and coping with any theft or product loss cases.

2.4.3 *Essentials of RFID Technology*

Several organizations are involved in the development and definition of RFID technology such as in the hardware concepts, the applications environment, and so on. Various organizations take part in standardization such as the International Organization of Standardization (ISO), EPCglobal Inc., the European Telecommunications Standards Institute (ETSI), and the Federal Communications Commission (FCC) [1].

An RFID system is made up of two major components; the transponder and the reader [2]. The transponder is the component which is located on a product or object to be identified, and

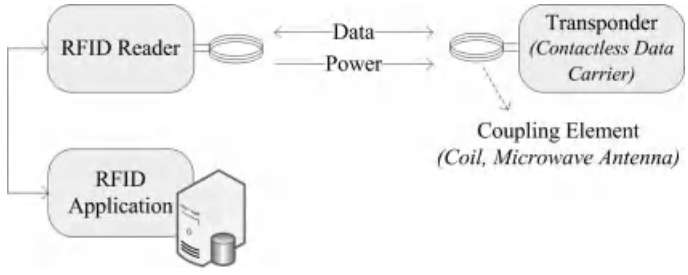


Figure 2.11 Main components of the RFID system.

the reader is the component which reads data from the transponder or read/writes from/to the transponder (see Figure 2.11):

- The transponder consists of a coupling element and an IC which holds the actual data. The transponder is actually the RFID tag. The transponder can be either passive or active. When the transponder is in the range of an RFID reader, it is powered by the incoming signals.
- The reader typically contains a transceiver (or high frequency module) with a decoder which is for interpreting data, a control unit and an antenna. Many RFID readers are fitted with an additional interface to enable them to forward the data received to another system.

2.4.4 RFID Tags as Transponders

As mentioned, RFID tags are small ICs with coupling elements. They have enough capability to store large amounts of data. They are divided into two major groups: passive tags which have no power supply; and active tags which have their own power supply.

2.4.4.1 Passive RFID Tags

Passive RFID tags have an embedded IC and antenna, but have no internal power supply. The incoming RF signals provide enough power to boot up the IC in the tag and transmit a response. Due to the lack of an embedded power supply or battery, passive RFID tags are quite small. In terms of size, they can vary between the size of a postage stamp to that of a post card. Also they can be read only at short distances which change from about 10 cm to a few meters. This range actually depends on the selected radio frequency, antenna design and size. The passive RFID tags are useful only for a limited set of applications because they have no internal power source and can be read only at short distances. The passive tag remains readable for a very long time; even after the commercial product containing passive RFID tag has been sold. These tags are very cheap to manufacture.

2.4.4.2 Active RFID Tags

Similarly, active RFID tags also have an IC and an antenna. However, unlike passive RFID tags, active RFID tags have their own embedded power source. This is used to power the IC within the tag to generate an outgoing signal. Active RFID tags have the ability to conduct a

Table 2.2 Comparison of active tags and passive tags

Parameters	Active Tags	Passive Tags
Power Source	Embedded power source	Power from RF field
Battery	Yes	No
Signal Strength to Tag	Very low	Very high
Operating Range	Long range	Up to a few meters
Data Storage Capacity	High	Low
Manufacturing Cost	Expensive	Cheap

session with an RFID reader. They can transmit at higher power levels and are more reliable than passive tags. These tags can be also effective in RF challenged environments such as in water, in shipping containers and in vehicles over long distances.

Active RFID tags have much longer range and larger memory size than passive RFID tags. The active RFID tag is typically more expensive, and physically larger than a passive RFID tag. They have the ability to store additional information sent by the RFID reader. Some active RFID tags can be integrated with sensors such as for logging temperature, humidity, shock/vibration, light, and atmospherics. The distinctive properties of active and passive RFID tags are compared in Table 2.2.

2.4.5 RFID Readers

An RFID reader is a device that is used to interrogate an RFID tag. As mentioned earlier, it contains a transceiver, a control unit, and an antenna. The antenna emits radio waves and the reader captures the data transmitted by the tag, and delivers it to the infrastructure for further processing, for example, retrieving content connected to the identification number on the tag. Readers are either capable of reading single frequencies whereas multi-protocol readers can read the spectrum of most available bands.

2.4.6 Frequency Ranges

RFID makes use of frequencies ranging from 300 kHz to 3 GHz; the exact range depends on country regulations. Active tags transmit only at higher RF frequencies, while passive tags transmit at all frequencies. Figure 2.12 shows the communication range of different frequency bands and system types.

2.4.7 Operating Principles of RFID Technology

The most important operating principles of RFID technology are inductive coupling and backscatter coupling:

2.4.7.1 Inductive Coupling

In accordance with [2], an inductively coupled transponder comprises of an electronic data carrying device, usually a single microchip and a large area coil that functions as an antenna.

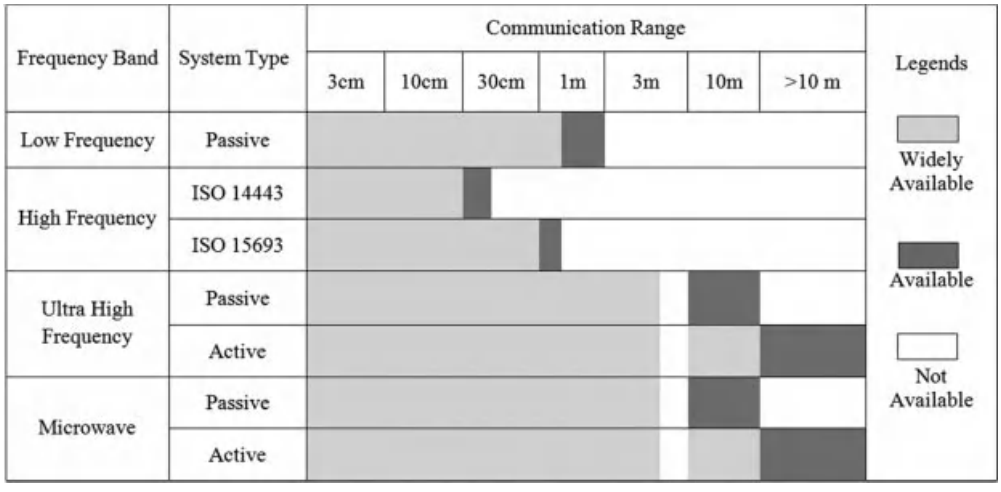


Figure 2.12 Communication range of the RFID system [3].

Inductively coupled transponders or tags are generally passive tags which have no internal power source. Thus they can be only used in near field cases. This means that all the energy for the embedded microchip within the tag has to be provided by the RFID reader in order for the microchip to operate.

For this purpose, the RFID reader's antenna generates a high frequency electromagnetic field. This field penetrates the cross section of the antenna's coil area and the area around the coil. The wavelength of the frequency range is several times greater than the distance between the RFID reader's antenna and the passive RFID tag. This electromagnetic field can be also identified as a simple magnetic alternating field.

When the RFID tag is placed in the electromagnetic field of the RFID reader, then the transponder gets energy from this magnetic field (see Figure 2.13). This power consumption can be described as a voltage drop at the internal resistance in the RFID reader's antenna through the supply current to the RFID reader's antenna. So, switching on and off a load resistance (or load modulator) at the transponder's antenna effects voltage changes at the RFID reader's antenna. If switching on and off the load modulator is controlled by data, then

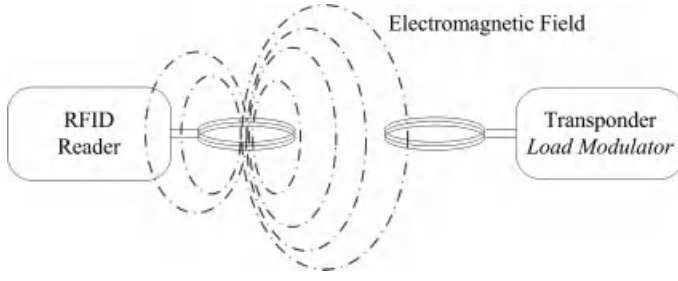


Figure 2.13 Inductive coupling.

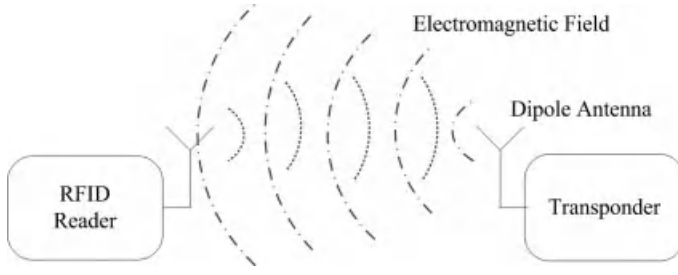


Figure 2.14 Backscatter coupling.

these data can be transferred from the RFID tag to the RFID reader. This type of data transfer is called load modulation.

2.4.7.2 Backscatter Coupling

Let us consider the electromagnetic field of RADAR technology [2]. The electromagnetic waves hit objects, and the objects reflect them. These objects have a large reflection cross section feature which is a measure of efficiency in an object reflecting waves.

In a typical RFID system, an electromagnetic field propagates outwards from the RFID reader's antenna. A small proportion of that field is reduced for free space reasons. The rest of the field reaches the transponder's or RFID tag's antenna. The antenna supplies the high frequency voltage. After rectification by diodes, this power can be used to turn on the voltage for the deactivation or activation of the power saving mode.

In the case of backscatter coupling, some of the incoming RF energy is reflected by the transponder's antenna as illustrated in Figure 2.14. The *reflection* cross section of the transponder's antenna can be influenced by altering the load connected to the antenna. In order to transmit data from the RFID tag to the RFID reader, a load resistor connected in parallel with the tag's antenna is switched on and off in time with the transmitted data. Thus the strength of the signal reflected from the transponder can be modulated. This is called *backscatter modulation*.

The signal from the transponder is radiated into free space. Some of this signal is received by the RFID reader's antenna. So, the reflected signal goes in a backwards direction to the RFID reader's antenna. It can be decoupled with the help of a *directional coupler.*

2.4.8 Near Field vs. Far Field Transmission

Two different RFID design approaches exist for transferring energy from the reader to the tag: magnetic induction and electromagnetic wave capture. The two designs make use of these approaches and are called near field and far field [1]:

- *Near field RFID*

  Near field RFID uses magnetic induction between a reader and a transponder. While an RFID is generating a magnetic field in its location, it passes an alternating current through a reading coil. If an RFID tag with a smaller coil is placed within range of the reader, the

alternating voltage appears across it and the magnetic field is affected by data stored on the tag. The voltage is rectified and powers the tag. As it is powered, the data are sent back to the reader using load modulation.

- *Far field RFID*

  Tags using far field principles operate above 100 MHz, typically in the 865–915 MHz range up to 2.45 GHz. They use backscatter coupling operating principles. In far field the reader's signal is reflected and it is modulated to an alternating potential difference in order to transmit data. The system's range is limited by the energy transmission sent by the reader. Due to advances in semiconductor manufacturing, the energy required to power a tag continues to decrease. The possible maximum range increases accordingly.

2.4.9 Common RFID Applications Throughout the World

RFID applications have been implemented throughout the world including:

- *Casino chip tracking:* Some casinos are placing RFID tags on their high value chips to track and detect counterfeit chips, to follow the betting habits of individual players, to speed up chip tallies and to determine counting mistakes of dealers.
- *Animal identification:* Using RFID tags for animals is one of the oldest RFID applications. RFID provides animal identification management for large ranches and rough terrain. An implantable variety of RFID tag is also used for animal identification.
- *Inventory systems:* RFID technology enables fast and easy management of an inventory for companies. It also enables to track reduction in out-of-stocks, increase in product selling as well as reduction in labor costs, simplification of business processes and reduction in inventory inaccuracies.
- *Hospital operating rooms:* Using an RFID reader and RFID tagged disposable gauze and towels is designed to improve patient safety and operational efficiency in hospitals.
- *Ski resorts:* Many ski resorts have adopted RFID tags to provide skiers hands-free access to ski lifts. It enables collecting information such as vertical feet skied and number of runs taken, which can also be made available to the user online.
- *Human implants:* Implantable RFID chips designed for animal tagging are also being used in humans.

2.5 Smart Card Technology

A smart card includes an embedded IC which can be a memory unit with or without a secure microcontroller. A typical smart card system contains cards, readers, and a backend system. The smart card connects to a reader with direct physical contact (contact smart card) or with a remote contactless RF interface (contactless smart card). The reader connects to the backend system which stores and manages the information.

The motivation for smart cards is that there is a clear need for a portable record of applications and these records require updating over time. The security of these records, especially confidentiality and integrity, is important as well. Smart cards are promising solutions for efficient data processing and transfer, and for providing secure multi application environments.

In terms of capability, smart cards are divided into two groups: memory based and microprocessor based smart cards (see Section 2.5.3). Smart cards with embedded secure

microcontroller can store large amount of data and perform their own on-card functions such as security related operations and mutual authentication. These smart cards can interact intelligently with a smart card reader. These cards have their own operating system (see Section 2.5.4). Similarly, in terms of operating mechanism, smart cards are divided into three groups: contact, contactless, and hybrid smart cards. The details of the general features and interfaces are explained later in Section 2.5.5.

A smart card should obviously conform to the international standards. There are a number of standards and specifications that are relevant for smart card implementations and some of them are relevant for industry-specific applications. The complete smart card standardization bodies and specifications are ISO/IEC Standards, EMV 2000 Specifications, Federal Information Processing Standard 201 (FIPS 201), other Federal Information Processing standards, American National Standards Institute (ANSI) Standards, GlobalPlatform Specifications, Common Criteria (CC) Specifications, International Civil Aviation Organization (ICAO), International Airline and Transportation Association (IATA) Standards, G-8 Health Standards, The Health Insurance Portability and Accountability Act (HIPAA) of 1996 (Public Law 104-191) Standards, Global System for Mobile Communication (GSM) Standards, Personal Computer/Smart Card (PC/SC) Workgroup Open Specifications, Open Card TM Framework, American Public Transportation Association's Contactless Fare Media System (CFMS) Standard, and Biometric Standards [4].

2.5.1 Earlier Form of Smart Card: Magnetic Stripe Cards

The process of attaching a magnetic stripe to a plastic card was invented by IBM in the 1960s. A magnetic stripe is the black or brown stripe placed on typically a credit card, or it can be placed on the back of an airline ticket or a transit card. The stripe is composed of tiny magnetic particles in a resin. The particles can be applied directly to the card or can be made into a stripe on a plastic backing which is applied to the card. Magnetic stripe cards are capable of storing data. These cards can be read by physical contact, by swiping the card on an external device which has magnetic reading head as depicted in Figure 2.15. Currently, magnetic stripes are mostly visible on financial debit or credit smart cards, airline tickets, and boarding passes.

Figure 2.15 Magnetic stripe card.

The material used to make the particles defines the coercivity of the stripe. Coercivity is the measure of difficulty to encode information on the magnetic stripe. The measure of coercivity is adjusted by the material used to make the particles. For instance, low coercivity stripes may use iron oxide and high coercivity stripes may use barium ferrite. The advantage of high coercivity is that it is harder to encode the information on the stripe. Hence, it is more difficult to erase the information on the card, so problems of accidental erasure are eliminated.

2.5.2 Evolution of Smart Cards

Smart cards were invented in the 1970s. The first mass use of the cards was for telephone payments in the 1980s. In the meantime, microprocessor smart cards were introduced. Microchips were integrated into debit cards in the 1990s. Smart card based electronic purse systems which store values on a card and do not need network connectivity, began to be used in Europe from the mid 1990s. One major improvement in smart card technology occurred in the 1990s; smart card based SIMs were introduced and started to be used in GSM based mobile phone environments in Europe. The use of smart cards increased with the ubiquity of mobile phones in Europe. In 1993, the international payment brands Europay, MasterCard, and Visa (EMV) collaborated to develop new specifications for smart cards in order to use them in payments both as a debit and a credit card. The first version of the EMV specifications, which stands for Europay, MasterCard, and Visa specifications, was released in 1994. EMVCo upgraded the specification in 2000. The specification was most recently upgraded in 2004. With the exception of some countries, there has been significant progress in the deployment of EMV-compliant Point of Sale (POS) equipment, as well as in the issuance of debit and credit cards using the EMV specifications. At that time, typically each country's national payment association was coordinated either by MasterCard International, Visa International, American Express, or JCB. They jointly planned and implemented EMV systems by considering various stakeholders. With the introduction of EMV specifications and systems throughout Europe, payment with contact smart card systems improved drastically. From a contactless smart card technology perspective, the major progress was the agreement of Visa and MasterCard in 2004–2006 to implement contactless payment and ticketing applications such as mass transit and highway tolls in the USA. With the introduction of contactless smart cards such as the MIFARE proximity smart card by Philips, contactless smart card applications started to have a considerable market share in Europe and the US.

2.5.3 Types of Smart Cards: Capability Based Classification

Smart cards are plastic cards with an embedded microprocessor and memory. Some smart cards have only non-programmable memory, thus they have limited capabilities. The smart cards that have microprocessors have various functionalities. Smart cards, in terms of their capability, can be divided into two major groups: memory based and microprocessor based smart cards.

2.5.3.1 Memory Based Smart Cards

Memory based smart cards can store any kind of data including financial, personal and other special information. However, they do not have a processing capability. These cards need to

Figure 2.16 Sample microprocessor based smart card.

communicate with an external device such as a card reader using synchronous protocols to manipulate the data on the card. These cards are widely used as prepaid telephone cards.

2.5.3.2 Microprocessor Based Smart Cards

The microprocessor based smart cards have on-card dynamic data processing capabilities. They have a microprocessor, as well as a memory. A sample microprocessor card is shown in Figure 2.16. The microprocessor within the card manages the memory allocation and data management. Microprocessor based smart cards are comparable with tiny computers, ones without internal power source. These smart cards have an Operating System (OS), namely the Smart Card Operating System (SCOS), enabling the data to be managed on the smart card and allowing the smart cards to be multi functional. They can store and process information, and perform complex calculations on the stored data. They can record, modify and process the data unlike memory based smart cards. Also, microprocessor based smart cards have the ability to store high amount of data compared with memory cards.

2.5.4 Smart Card Operating System (SCOS)

Until the end of the 1990s, it was very difficult to have more than one application running on a smart card due to the memory constraints of the IC chips. With the development of SCOSs, implementing several applications, running them simultaneously, and loading new ones during a card's active life became possible. Now, SCOSs enable more dynamic, multi application platforms, and they are considered to be a really smart and powerful secure computing object for a lot of new application domains.

Today each smart card has its own SCOS which can be defined as a set of instructions embedded in the ROM of the smart card. SCOSs are generally divided into two categories [5]: general purpose SCOS which has a generic command set in which the various sequences cover most applications; and dedicated SCOS which has commands designed for specific applications and can contain only the related application(s) (e.g., a payment smart card which

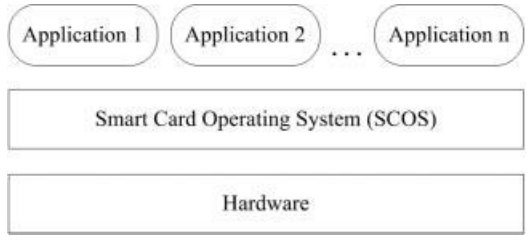


Figure 2.17 Smart card architecture [6].

is designed to support only payment transactions). Smart card architecture is depicted in Figure 2.17.

The basic functions of SCOS include:

- Managing interchanges between a smart card and an external device such as a POS terminal.
- Managing the data stored in memory.
- Controlling the access to information and functions.
- Managing security of the smart card especially in terms of data integrity.
- Managing the smart card's life cycle from its personalization to usage and termination.

Earlier in SCOS evolution, an application or service on a smart card was written for a specific OS. Thus the card issuer had to agree with a specific application developer as well as an operating system. This was a costly and an inflexible solution. Consumers needed to carry different smart cards for each service. Today the trend is towards an open operating system that supports multiple applications running on a single smart card. Currently, the most notable OSs that have bigger market exposure are MULTOS and JavaCard OS [5,6].

2.5.4.1 MULTOS

MULTOS (Multi-application Operating System) is an operating system for smart cards which is one of the ideal OS solutions with enhanced security features. MULTOS is controlled by an industrial MULTOS consortium which involves entities from smart card chip and silicon manufacturers to card management and personalization system providers.

MULTOS aims to provide a standard secure SCOS that could be implemented on any silicon chip and execute any smart card application (e.g., payment, identity, ticketing). Hence, various applications can run and operate on the same smart card both independently and securely. This provides a combination of diverse applications from different vendors and all can exist independent of each other.

One key difference of MULTOS when compared with other SCOSs is that it is specifically designed for secure applications. It implements Secure Trusted Environment Provisioning (STEP), which is a patented mechanism. STEP enables manufacture, issuance, and dynamic updates of MULTOS smart cards under card issuer's control, which is enabled by MULTOS Key Management Authority (KMA). KMA provides card issuers the required cryptographic

data, so that card issuers can take control of the smart card, and generate permission certificates for management of applications.

2.5.4.2 JavaCard OS

Another new trend in SCOSs is the JavaCard OS. JavaCard enables applications (or applets) written in the JavaCard Language (a subset of the Java programming language) to run on smart cards. This technology is standardized by Sun Microsystems and the JavaCard Forum.

JavaCard provides a secure, interoperable, and multi application platform for smart cards by using the advantages of the Java language; object-oriented programming, reuse of existing development environments, strongly typed language, several levels of access control to methods and variables, and interoperability. It means JavaCard OS based applications could be used on any vendor of smart cards that support the JavaCard OS. JavaCard OS gives programmers independence over architecture.

Another advantage of the JavaCard OS is that it allows post-issuance of applications. This allows upgrading and updating the applications on the smart card after delivering the card to the end-user when necessary. JavaCard OS has a JavaCard Virtual Machine (JCVM) like a converter which allows JavaCard OS to run applications written in Java programming language. Part of the JCVM runs outside the card as converter, and another part runs on the card as the byte code interpreter. Thus, in JavaCard, many tasks are not carried out at execution but are deported to the part of the virtual machine outside the card (e.g., the loading of the classes, the verification of the byte codes, the resolution of links, and the optimization).

2.5.5 Types of Smart Cards: Mechanism Based Classification

Smart cards are divided into three major groups in terms of the communication mechanism with outer devices: contact smart cards, contactless smart cards, and hybrid models.

2.5.5.1 Contact Smart Cards

Contact smart cards are embedded with a micro module containing a single silicon "IC card" (or chip card) with memory and microprocessor. This IC card is a conductive contact plate of approximately 1 cm^2 (0.16 $in.^2$) placed on the surface of the smart card which is typically gold plated. An external device provides a direct electrical connection to the conductive contact plate when the contact smart card is inserted into it. Transmission of commands, data, and card status information takes place over these physical contact points.

Cards do not contain any power source; hence energy is supplied by the external device that the card interacts with. These external devices are used as a communications medium between the contact smart card and a host. These external devices can be a computer, a POS terminal or a mobile device. Contact smart cards interacting with POS devices are used for payment purposes. The dimensions of a contact smart card for payment purposes are standardized as 85.6 mm $\times$ 53.98 mm $\times$ 0.76 mm, similar to current bank cards (see Figure 2.18). These cards also have a magnetic stripe capability. Actually the IC cards used on contact smart cards for payment purposes are the same as those used in Subscriber Identity Modules (SIMs) in

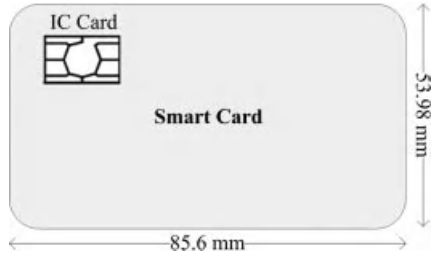


Figure 2.18 Financed based contact smart card standard dimensions.

mobile phones. They are programmed differently and embedded in a different piece of PVC. These cards interact with external readers such as mobile phones. The physical shape of SIM modules with IC cards can differ from bank cards, and typically they are smaller. Contact smart cards can be used by inserting them in the reader device.

The standards related to contact smart cards are ISO/IEC 7810 and mainly ISO/IEC 7816. They define the physical shape and characteristics of contact smart cards, electrical connector positions and shapes, electrical characteristics, communication protocols including commands exchanged with the card, and basic functionality. According to the ISO/IEC 7816 Standard, the IC card has eight electrical gold-plated contact pads on its surface (see Figure 2.19): VCC (power supply voltage), RST (reset the microprocessor), CLK (clock signal), GND (ground), VPP (programming or write voltage), and I/O (serial input/output line). Only the I/O and GND contacts are mandatory on a typical smart card; the others are optional. Two contacts are reserved for future use (RFU). The usage of communication paths C1 through C8 are [4,7]:

- *C1:* VCC is the supply voltage that drives the chips and is generally 5 V. It is possible to foresee chips that allow lower energy consumption.
- *C2:* RST is the reset signal line that initiates the state of the IC after the power is on.
- *C3:* CLK is the clock signal which is to drive the logic of the ICC and is also used as the reference for the serial communications link. This line controls the operation speed and provides a common framework for data communication between the interface device (IFD) and the integrated circuit chip (ICC).
- *C4:* RFU.
- *C5:* GND refers to the ground signal line which provides a common electrical ground between the IFD and the ICC.

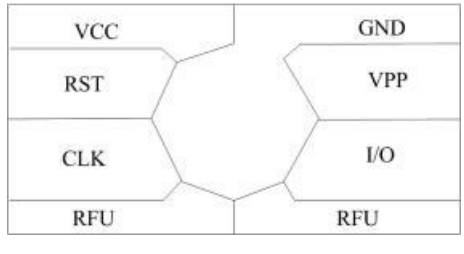


Figure 2.19 Contact pads on an IC card.

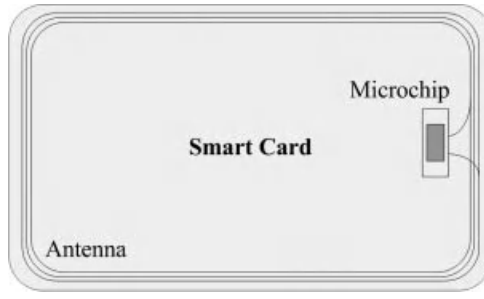


Figure 2.20 Contactless smart card.

- *C6:* VPP is the power connection or programming voltage input that is used for the high voltage signal which is necessary to program the EPROM memory. Until the late 1980s, it was necessary to apply an external voltage to program and erase the EEPROM, since the microcontrollers used at that time did not have charge pumps. However, since the early 1990s it has been standard practice to generate this voltage directly from the chip using a charge pump, so this contact is no longer used. The C6 contact cannot be employed for any other function, otherwise it would conflict with the ISO standard. C6 is right in the middle between the GND and I/O contact, so C6 is neither used nor eliminated.
- *C7:* I/O is the serial input/output connector that provides a half duplex communication channel between the card reader and the smart card. This signal line is used for communication and sending commands to the chip within the smart card. The protocols used to communicate with a smart card are referred to as T0 and T1.
- *C8:* RFU.

2.5.5.2 Contactless Smart Cards

The contactless smart card is a type of smart card that is processed without having to come into contact with an external device. It is a combination of a microchip (or IC chip) embedded within it and antenna which allows the card to be tracked (see Figure 2.20). This antenna is formed by several wires. In contactless smart cards, information is stored in the microchip which has a secure microcontroller and internal memory. Unlike the contact smart card, power supply to the contactless smart card is achieved with the embedded antenna on the smart card. Data exchange between the smart card and the external device such as a smart card reader is performed with the help of this antenna. Electromagnetic fields provide power for the card as well as exchanging data with the external device.

Contactless smart cards have the ability to securely store and manage data. They also provide access to the data stored on the card; they perform on-card functions such as enabling mutual authentication. They can easily and securely interact with a contactless external reader.

The contactless communication can be performed only with devices in close proximity. Readers with RFID reading capability are marked with a special symbol as shown in Figure 2.21. Both the external device and the contactless smart card have antenna, and communicate using RF technology at data rates of 106–848 kbps. Most contactless cards also derive power for the microchip from this electromagnetic signal.

Figure 2.21 Universal contactless smart card reader symbol.

When a contactless smart card is brought into the electromagnetic field of the card reader, the energy transfer starts from the card reader to the microchip on the smart card. The microchip is powered by the incoming signal from the card reader. Once the microchip is powered, the wireless communication is established between the smart card and the card reader for data transfer.

Contactless smart card technology is used in applications that process private information such as health data and identity data to be protected. It is also used in applications when fast and secure transactions are required such as transit fare payment, electronic passports, and visa control. Contactless smart cards are often used for hands-free transactions. Applications using contactless smart cards must support many security features such as mutual authentication, strong information security through dynamic cryptographic keys, strong contactless device security, and individual information privacy. Contactless smart card technology is available in a variety of forms such as in plastic cards, watches, key fobs, documents, mobile phones, and other mobile devices or handsets.

Currently, there are three different major standards for contactless smart cards based on a broad classification range: ISO/IEC 10536, ISO/IEC 14443 and ISO/IEC 15693:

(i) *ISO/IEC 10536 – Close Coupling Smart Cards*
The ISO/IEC standard 10536 entitled "Identification Cards – Contactless Integrated Circuit Cards" describes the structure and operating parameters of contactless close coupling smart cards. These cards operate at a distance of up to 1 cm. This standard has four parts: physical characteristics, dimensions and location of coupling areas, electronic signal and reset procedures, and response to reset and transmission protocols.

(ii) *ISO/IEC 14443 – Proximity Coupling Smart Cards*
The ISO/IEC standard 14443 entitled "Identification Cards – Proximity Integrated Circuit Cards" describes the operating method and operating parameters of contactless proximity coupling smart cards. According to this standard a contactless proximity smart card operates at a distance of less than 10 cm (less than 4 in.) at 13.56 MHz. This standard's details are explained in Chapter 3.

(iii) *ISO/IEC 15693 – Vicinity Coupling Smart Cards*
The ISO/IEC standard 15693 entitled "Identification Cards – Contactless Integrated Circuit Cards – Vicinity Cards" describes the method of functioning and operating parameters of contactless vicinity coupling smart cards. These smart cards operate in a range of up to 1 m at 13.56 MHz, like those used in access control systems. This standard has four parts: physical characteristics, air interface initialization, anti-collision protocols, and transmission protocol.

Table 2.3 Some novel smart card applications

Application Areas	Examples
Finance	Electronic purse cards to replace coins for small purchases
	Credit and debit payment cards
Communication	GSM based SIM cards which are popular for secure communication
Identification and Physical Access Control	Employee access cards with secured ID and passwords
	Student ID or campus cards which can contain a variety of applications such as authentication, electronic purse, library card, and meal card
Transportation	Mass transit fare collection systems
	Driver licenses
	Electronic toll collection systems
Loyalty	Membership or loyalty cards for consumer reward tracking systems
Health	Health cards for storing insurance and emergency medical data of the user

2.5.5.3 Hybrid Models

Additionally, it is possible to see hybrid models of smart cards such as dual interface cards and hybrid cards:

- A dual interface card has both contact and contactless interfaces, that contains only one chip. Such a model enables the same chip to be accessed by both the contact and the contactless interface with a high level of security.
- A hybrid card contains two chips. One of those chips is used for a contact interface and the other one is used for a contactless interface. These chips are independent and not connected.

2.5.6 Smart Card Applications

The first smart cards were prepaid telephone cards implemented in Europe in the mid 1980s. They were simple memory smart cards. Today, some of the major application areas for microprocessor based smart cards are finance, communications, identification and physical access control, transportation, loyalty, and health care. A smart card can carry applications from more than one area such as combining physical security access, financial and loyalty applications on the same card. Some major smart card application areas and examples are presented in Table 2.3.

2.6 Comparison between RFID Tags and Contactless Smart Cards

Currently, many applications are using RF technology to automatically identify objects and people. These applications range from tracking animals and products for enabling fast payment and securely identifying people. All these applications use radio waves to enable wireless communication and data transfer. However, these applications differ in the RF technology they use depending on the application requirements [8]. As a general definition, RFID technology uses RFID tags in applications that identify or track objects, whereas contactless smart card

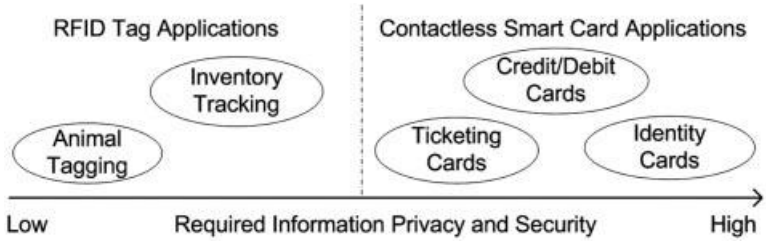


Figure 2.22 RFID tag vs. contactless smart card applications.

technology uses contactless smart cards in applications that identify people, or store financial and personal information.

As already mentioned, RFID tags are the major components in RFID systems. They are simple, low cost, and disposable. They are used in various projects such as in identifying animals and tracking goods logistically and are replacing printed barcodes at retailers. RFID tags include an IC that typically stores static data and an antenna which enables the chip to transmit the stored data to a reader. When the tag comes within the range of the RFID reader, the data stored on tag are transferred to the reader. No security or authentication scheme is provided for the data on the tag or transmitted between the tag and the reader. Thus it is possible for any reader using the appropriate RF signal to obtain the content of an RFID tag. Typical RFID tags can be easily read from distances of several centimeters to several meters to allow easy tracking of objects based on the used tag type.

Like RFID tags, contactless smart cards also interact intelligently with an external device using RF technology. However contactless smart cards are used in applications that need a high level of security to protect sensitive information and perform secure transactions. As already mentioned, a contactless smart card includes a secure microcontroller and internal memory. It also has the ability to perform complex functions (e.g., encryption or other security functions) and securely manage, store and provide access to data on the card. Applications that require a high level of security (e.g., payment applications, government IDs, electronic passports) use contactless smart card technology. A distance of approximately 10 cm (4 in.) is usually preferred for those applications. Applications that need longer reading distances can use other types of contactless technologies such as vicinity coupling contactless smart cards. Applications using contactless smart cards ensure data integrity and confidentiality, and privacy of information stored or transferred. When compared with RFID tags, contactless smart card technology is the ideal solution for applications that require security and privacy.

Applications often have different requirements and capabilities in their use of RF technology. Figure 2.22 summarizes the types of technologies used in some key applications and the level of information protection that they provide. For example, in the inventory tracking and animal tagging applications, RFID tags store information such as an ID number or electronic code which does not have high sensitivity. Thus they do not need a high level of security and privacy when compared with contactless smart card applications.

2.7 More on NFC

NFC technology was jointly developed by Philips and Sony in late 2002 for contactless communications. Europe's ECMA (European Computer Manufacturers Association) International

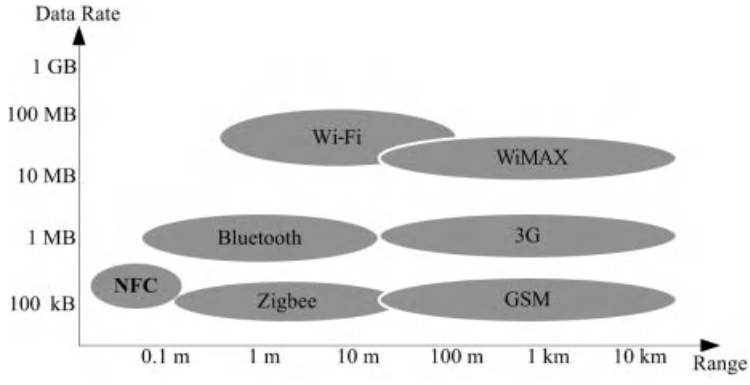


Figure 2.23 Comparison of wireless technologies based on range and data exchange rate.

adopted the technology as a standard in December 2002. The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) adopted NFC technology in December 2003. In 2004, Nokia, Philips, and Sony founded the NFC Forum to promote the technology. NFC technology standards are acknowledged by ISO/IEC (International Organization for Standardization/International Electro-technical Commission), ETSI (European Telecommunications Standards Institute), and ECMA. The other participating organizations in standardization of NFC technology and NFC devices are discussed in detail in Chapter 3.

NFC is a bidirectional and short range wireless communication technology. The communication occurs between two NFC compliant devices within a few centimeters. A 13.56 MHz signal with a bandwidth not more than 424 kbps is used. NFC technology operates in different operating modes; reader/writer, peer-to-peer, and card emulation where communication occurs between an NFC mobile on one side, and a passive RFID tag (NFC tag), an NFC mobile or an NFC reader on the other side, respectively. Each operating mode uses distinct communication interfaces (i.e., ISO/IEC 14443, FeliCa, NFCIP-1 interfaces) on the RF layer as well as has different technical, operational and design requirements that are explicitly presented as illustrated in Chapter 3.

NFC technology is compared with other technologies [9] in terms of data transfer rate in Figure 2.23 and in terms of security, personalization, flexibility and power consumption in Table 2.4.

In NFC, a device generates a low frequency radio wave field in the 13.56 MHz spectrum. When another NFC device gets close enough to contact the field, magnetic inductive coupling

Table 2.4 Comparison of NFC with other wireless technologies [10]

Parameter	RFID	Bluetooth	ZigBee	NFC
Security	High	Low	Low	High
Personalization	High	Medium	Low	High
Flexibility	Low	High	High	High
Power Consumption	No	High	Medium	Low

transfers energy and data from one device to the other. An NFC device with an internal power supply is considered active whereas a device with no internal power supply such as a smart card is considered passive. Inductive coupling causes a passive device to absorb energy from an active device when it gets close enough. Once the passive device is powered up, it can communicate and exchange data with the other device.

To date, many NFC trials have been conducted throughout the world, especially in the payment domain. All trials conclude that with the development of NFC technology, the mobile phone will become safer, more convenient, faster and a more fashionable item. NFC technology allows people to integrate the cards they use daily, such as loyalty and credit cards, into their mobile phones. In addition to integrating cards into mobile devices, NFC technology brings innovations to mobile communications. It enables two users to easily communicate and exchange data simply by touching two mobile phones together. Moreover NFC technology gives NFC reader capability to mobile phones, so that NFC can be read by them. The increasing processing power of mobile phones, Internet access and many other features gain advantages from this reader functionality that will lead to new and innovative services. It is true that NFC technology brings simplicity to transactions, provides easy content delivery and enables information sharing. At the same time it builds new opportunities for various stakeholders; mobile operators, banks, transport operators and merchants will presumably have faster transactions, less cash handling and new operator services.

2.7.1 Inherent Security and Pairing Capability of NFC

One of the major properties of NFC technology is its implicit security because of the short communication distance involved. The close proximity of the two devices makes the signal interception probability very low. The other property is the implicit pairing capability of NFC. An installed application on a mobile device is automatically launched when it finds the matching pair. Consider an application that reads concert events from an NFC tag. This application can be programmed to run immediately and automatically when a mobile phone is touched to an NFC tag which consists of the event information. It can perform all of the processing until user intervention is required. If no user intervention is needed according to the embedded algorithm, the application may finalize its processes and may exit as well. In this case, the user may not even notice that the application has been started, processed, and finalized. This kind of processing is not suitable for services with security requirements. In the case of a payment for example, user intervention is essential and the payment cannot proceed without the user's approval and without details such as a payment password being entered. Automatically firing the NFC application allows automatic pairing of the related NFC components, which is not true in Bluetooth. The user shows her intention for this automatic pairing by bringing her device close to the other device. In order to show intention in a peer-to-peer communication, two users need to bring their devices close to each other. So, pairing two NFC devices is completely different from the processes required in Bluetooth. Remember that pairing devices in Bluetooth includes searching, waiting, pairing, and authorization.

2.8 Chapter Summary

Ubiquitous computing refers to the next level of interaction between humans and computers where computing devices are completely integrated into our everyday life. NFC is mostly

regarded as an important step towards ubiquitous computing. NFC uses the touching paradigm for interaction. The users need to touch their mobile phones to a reader or a tag in order to establish a connection. NFC is an extension of RFID technology and compatible with contactless smart card technology interfaces.

RFID technology is used for tagging and identifying objects over large ranges. An RFID system typically includes two major components: the transponder which is placed on the object to be identified; and the reader which has reading and/or writing capability. The transponders are usually RFID tags (either passive or active).

Contactless smart card technology uses contactless smart cards that need to protect private information and also perform fast and secure transactions. Some examples of smart cards are personal identity verification, transit fare payment cards, and electronic passports. The major standards for contactless smart cards are ISO/IEC 10536 for close coupling, ISO/IEC 14443 for proximity coupling, and ISO/IEC 15693 for vicinity coupling smart cards.

NFC technology is defined by the NFC Forum founded by Nokia, Philips, and Sony which allow communication based on RFID technology and ISO/IEC 14443 infrastructures. It operates in three modes (reader/writer, peer-to-peer, and card emulation) with RF of 13.56 MHz where communication occurs on one side between a mobile phone with NFC capability, and on the other side a passive RFID tag, an NFC device or an NFC reader, respectively.

The major properties of NFC technology are automatic pairing and implicit security due to its short range communication capability. When compared with other wireless technologies, it allows low data rate transfer within very close distances. It is more human centric, easy, fast and enables high security and privacy.

Chapter Questions

1. Explain ubiquitous computing, and its relationship with NFC.
2. What are the similarities and the differences between QR Code and RFID technology?
3. Explain the difference between active and passive RFID tags.
4. What are the differences between near field and far field transmission concerning energy transfer between an active and a passive device?
5. What are the differences between memory based and microprocessor based smart cards?
6. What are the differences between microprocessor based smart cards and PCs?
7. What are the challenges that a mobile OS faces, but a PC OS does not?
8. Draw the universal contactless smart card reader symbol.
9. What are the differences between proximity and vicinity coupling smart cards?
10. What are the differences between RFID tags and contactless smart cards?
11. What are the most important integral properties of NFC technology?

References

[1] Resarsch, F. (2010) *Ubiquitous Computing, Developing and Evaluating Near Field Communication Applications*, Gabler, ISBN: 978-3-8349-2167-3.

[2] Finkenzeller, K. (2010) *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, John Wiley & Sons, Ltd, ISBN: 978-0-470-69506-7.

[3] Dressen, D. (2004) Considerations for RFID technology selection. *Atmel Applications Journal*, **3**, 45–47.

[4] Smart Card Alliance, http://www.smartcardalliance.org/ (accessed 10 July 2011).
[5] Cardwerk, http://www.cardwerk.com/smartcards/ (accessed 10 July 2011).
[6] Sauveron, D. (2009) Multiapplication smart card: Towards an open smart card? *Information Security Technical Report*, **14**(2), 70–78.
[7] Leng, X. (2009) Smart card applications and security. *Information Security Technical Report*, **14**(2), 36–45.
[8] Smart Card Alliance, *RFID Tags & Contactless Smart Card Technology: Comparing and Contrasting Applications and Capabilities*. Available at: http://www.hidglobal.com/documents/tagsVsSmartcards_wp_en.pdf (accessed 10 July 2011).
[9] NFC Forum, http://www.nfc-forum.org/ (accessed 10 July 2011).
[10] Chang, Y. *et al.* (2010) NCASH: NFC phone-enabled personalized context awareness smart-home environment, *Cybernetics and Systems*, **41**(2), 123–145.

3

NFC Essentials

We are now ready to provide the reader with an in depth view of the NFC technical essentials. We first introduce the underlying structure of NFC, and its devices (NFC tag, NFC reader, and NFC enabled mobile phone). It is also important to understand the standardization bodies which steer NFC technology. Different standardization bodies are composed of different types of actors in the NFC ecosystem. Mobile device and smart card manufacturers, banks and mobile network operators (MNOs) are among the major stakeholders who influence the NFC standardization efforts.

In this chapter, we elaborate the communication architecture and essentials of each operating mode (reader/writer, peer-to-peer and card emulation) and radio frequency (RF) layers of NFC technology. NFC Forum has almost completed the standardization of the reader/writer and peer-to-peer operating modes' communication between application level and RF layer. In the card emulation operating mode however, the technical infrastructure is quite different from the other two modes and the standardization process is not completed yet. This mode implicitly gives the smart card operating capability to an NFC mobile owning a secure environment.

3.1 Introduction to NFC

NFC occurs between two NFC devices in a close proximity range (within a few centimeters). These two NFC devices can operate in several modes. NFC modes are distinguished based on whether each device has an embedded power source, so that they can generate their own RF, or one retrieves the power from the RF field generated by the other. In terms of power, if a device generates its own RF field, it is called an active device; otherwise, it is called a passive device.

Alternatively devices can be classified based on an algorithmic point of view; if a device actively leads in a communication session such as by asking for a credit card number, or by passing some information to the partner based on some intelligent decisions, it is assumed to be an active device. However, if it replies to the interrogator, it is assumed to be a passive device. In essence, both points of view (classification of devices based on their internal power source or the algorithm used) mostly match. When a device has an embedded power source, it naturally initiates and leads communication. On the other hand, if it does not have any

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.
 Published 2012 by John Wiley & Sons, Ltd.

Table 3.1 Active vs. passive communication mode

Device A	Device B	Description	Communication Mode
Active	Active	RF field is generated by both devices	Active mode
Active	Passive	RF field is generated by Device A only	Passive mode
Passive	Active	RF field is generated by Device B only	Passive mode

embedded power source, it can only respond to the active device. In this book, we refer to a device as an active device when it has its own embedded power source; otherwise we refer to it as a passive device.

Similarly, operation modes defined in NFC protocol are also classified as active and passive communication modes. In active communication mode, both devices generate their own RF field to exchange data (see Table 3.1). In passive communication mode, only one device generates an RF field, while the other uses load modulation to transfer data; this has already been explained in Chapter 2.

In addition, there are two different roles that a device can play in NFC which can be illustrated as a "request and reply" concept as shown in Figure 3.1. The initiator sends a request message to a target and the target replies by sending a message back to the initiator. In this case the role of the initiator is to start the communication. The role of the target is to respond to the requests coming from the initiator.

Table 3.2 shows the possible combinations of initiator/target roles with respect to active/passive roles for a device. An active device can act as both an initiator and a target. However, a passive device cannot be an initiator.

As already mentioned in Chapter 2, NFC occurs between two NFC devices. These devices can play an initiator or a target role as well as being an active or a passive device. Table 3.3 shows the possible interaction styles between NFC devices which are NFC enabled mobile phones, NFC tags, and NFC readers:

(i) *NFC enabled mobile phones*

NFC enabled mobile phones are also referred to as NFC mobiles. Currently, integration of NFC technology in NFC mobiles creates a big opportunity to show the ease of use and for acceptance of NFC ecosystem. Many NFC mobile models are already available in the market. Mobile phones enable secure storage of data and hence are able to behave like secure smart cards. By using this feature, NFC enables the use of NFC enabled applications that require secure implementation such as payment, ticketing, loyalty applications, and access control.

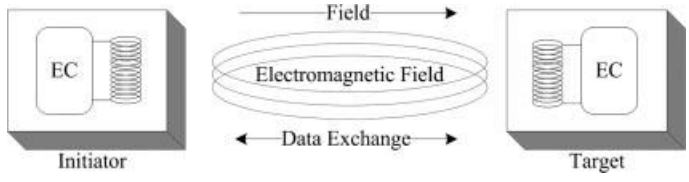


Figure 3.1 Initiator and target device.

Table 3.2 Possible combinations of active/passive device with initiator/target device

Device Role	Active Device	Passive Device
Initiator	Possible	Not possible
Target	Possible	Possible

(ii) *NFC tags*

An NFC tag is actually a passive RFID tag. Currently only a small amount of data can be stored on an NFC tag. In order to power an NFC tag, the user touches an active device such as an NFC mobile or an NFC reader to it. As the active device creates an RF signal, the energy is consumed by the NFC tag so that it boots up immediately and data are transferred back, since the pre-loaded algorithm is designed to do so.

By being a passive device, an NFC tag can communicate with only an active NFC device (NFC mobile or NFC reader); two tags cannot communicate with each other because no power source exists in NFC tags to enable the communication. NFC tags can be used in applications that require small capability such as smart posters. NFC tags may contain any type of data, but the size limitation cannot be exceeded. NFC Forum as a standardization body has defined four NFC tag types (Type 1, Type 2, Type 3, and Type 4). The details about NFC Forum mandated tag types and the record types will be explained in Section 3.5.

(iii) *NFC readers*

An NFC reader is always an active device and is capable of bidirectional information transfer with another NFC device. An NFC reader can be in one of two forms: internal and external. An internal NFC reader can be integrated to an NFC enabled mobile phone, so that when the NFC mobile is touched to an NFC tag, it can read/write data from/to the tag. Thus the interaction between an NFC mobile (with embedded NFC reader) and an NFC tag is, respectively, the interaction between active and passive devices.

In the case of the interaction between two NFC mobiles, a link-level communication is created between them. An NFC reader embedded in an NFC mobile is always active and generating its own RF field, unless the NFC mobile is in standby or airplane mode. An external NFC reader is generally used for reading data from an NFC mobile (see Figure 3.2). The most common example of an external NFC reader is a contactless POS terminal which can perform contactless NFC enabled payments when an NFC device is touched to the NFC reader.

Table 3.3 Interaction styles of NFC devices

Initiator Device	Target Device
NFC mobile	NFC tag
NFC mobile	NFC mobile
NFC reader	NFC mobile

Figure 3.2 NFC mobile and NFC reader.

3.2 Standardization and Development Efforts of NFC Enabled Mobile Phones

NFC technology benefits from various elements such as smart cards, mobile phones, card readers, payment systems and so on. All these elements need to acquire accreditation from an assortment of governing bodies that have the responsibility for the security and interoperability of various NFC devices. As mobile phones became the best solution for NFC technology, especially for secure transactions, various standardization bodies defined how the NFC technology should be integrated into mobile phones and other related devices. Some other bodies defined the architectures and standards for the security as well as the ancillary technologies for NFC enabled mobile phones, such as smart cards for NFC transactions. The common vision of all standardization bodies is to increase the ease of access, interoperability and security for NFC technology. Figure 3.3 gives a summary of the standardization bodies that play a role in the development of NFC technology and Figure 3.4 gives the "big picture" for mobile phones and the bodies supporting them.

3.2.1 NFC Forum

NFC Forum is a non-profit industry association that was established with the aim of enabling NFC technology and making it spread around the world. NFC Forum is an alliance for specifying the NFC standards built on ISO/IEC standards. The introduction of this organization dates back to 2004 when a number of major companies including Nokia, Philips, and Sony formed an alliance for advanced usage of RFID technology in consumer applications. The NFC Forum later included companies such as Visa, MasterCard, Samsung, Microsoft, and Motorola.

NFC Forum focuses on improving the use of short range wireless interaction through NFC technology in consumer electronics, mobile devices, and PCs. The mission of NFC Forum is to promote the usage of NFC technology by developing specifications, ensuring interoperability

Organization	Standards or Activities Governed within NFC Ecosystem			Responsibility
	Only Mobile Phones	Supportive Technologies	NFC Technology	
GSMA	✓	✓		Engages in technical, commercial and public policy initiatives to ensure that mobile services are interoperable worldwide
OMA	✓	✓		Develops specifications for mobile service enablers to promote interoperability
JCP		✓	✓	Establishes specifications for the development of Java technology on mobile phones
ETSI and its Smart Card Platform	✓	✓		Develops globally interoperable standards and handles SIM specifications
3GPP	✓	✓		Develops globally applicable technical specifications for the third generation GSM
EMVCo		✓	✓	Provides specifications to ensure interoperability of smart card based payment systems as well as mobile payment standards worldwide
GlobalPlatform		✓	✓	Provides open and interoperable infrastructure and standards for transactions performed on smart cards
NFC Forum	✓	✓	✓	Develops specifications for NFC devices that are based on ISO/IEC 18092 contactless interface ensuring interoperability among devices and services
ISO/IEC	✓	✓	✓	Provides worldwide international standards for business, government and society
ECMA International	✓	✓	✓	Provides international standards and technical reports for information communication

Figure 3.3 List of standardization bodies in NFC ecosystem.

between devices and services, and educating the market about NFC technology [1]. The objectives of NFC Forum are to:

- Develop NFC specifications that define a modular architecture for NFC devices.
- Define protocols for interoperable data exchange and device independent service delivery.
- Define protocols for device discovery and device capability.

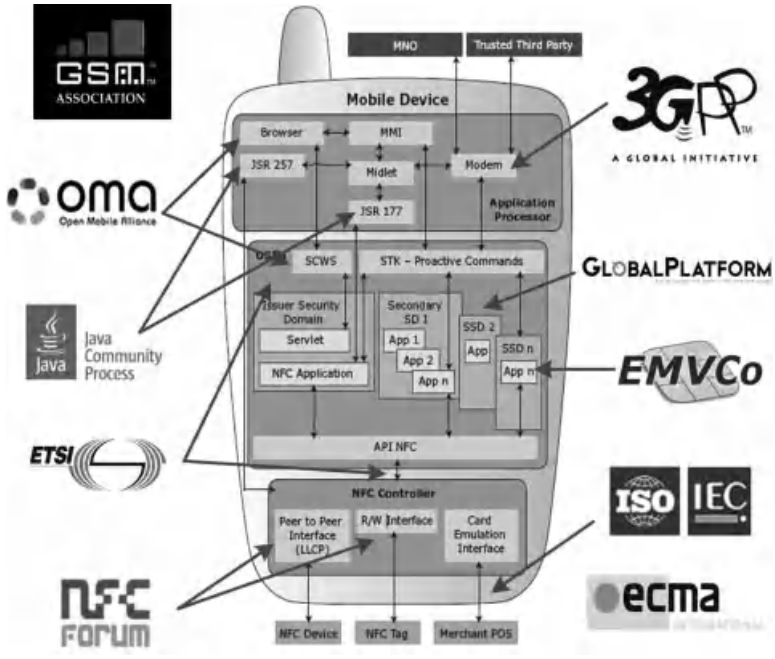


Figure 3.4 Standards and standardization bodies of NFC mobile. Reproduced by permission from GSMA. All rights reserved.

- Encourage technology providers to develop and deploy NFC enabled products around a common set of specifications.
- Establish a certification program that ensures compliant products according to NFC Forum specifications.
- Promote global use of NFC technology by educating consumers and enterprise users on the applications and benefits of NFC technology.

NFC Forum standardized only two operating modes (reader/writer and peer-to-peer) from the application layer to the RF layer (see Figure 3.5). As mentioned in Table 3.4, for reader/writer mode, Record Type Definition (RTD) and NFC Data Exchange Format (NDEF) specifications are being used. In peer-to-peer mode, Logical Link Control Protocol (LLCP) is used to connect the peer-to-peer based application to the RF layer, while the card emulation mode provides smart card capability for mobile phones. Details from Figure 3.5 will be briefly touched upon for each operating mode throughout this chapter. For more details about these specifications, please visit the NFC Forum website (http://www.nfc-forum.org).

Another important development introduced by the NFC Forum is the "N-Mark" trademark which is a universal symbol for NFC (see Figure 3.6), so that consumers can easily identify where their NFC enabled devices can be used. The N-Mark has two meanings [1]:

- The existence of an N-Mark on an active device means that the device has passed NFC Forum certification testing. Once a product is certified, it can contain the N-Mark on it. The

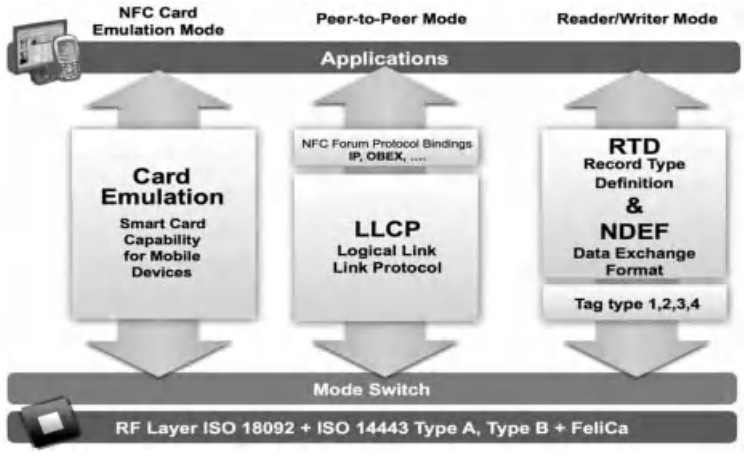


Figure 3.5 NFC Forum technical architecture. Reproduced by permission from NFC Forum.

N-Mark on the product is the touch point for NFC interactions. Currently, only members of NFC Forum can certify their devices.

- The N-Mark on posters, signs, badges, labels, and so on means that NFC services are available on the specified location which is important for customers. It indicates where the touching point on the target device is to enable NFC services. The N-Mark can be used freely on tags and other media after receiving the N-Mark license.

3.2.2 GlobalPlatform

GlobalPlatform represents the interests of the smart card industry, card issuers, and vendors. It is a cross industry, non-profit association which identifies, develops and publishes specifications that facilitate secure and interoperable deployment and management of multiple

Table 3.4 Overview on NFC Forum specifications

Specification	Purpose
NFC Data Exchange Format (NDEF)	Common data format for devices and tags
NFC Record Type Definition (RTD)	Standard record types used in messages between devices/tags
Smart Poster RTD	For posters containing NFC tags with text, audio or other data
Text RTD	For records containing plain text
Uniform Resource Identifier (URI)	For records that refer to an Internet resource
Connection Handover	Defines how to establish a connection with other wireless technologies
NFC Tag Types 1–4 Operation	Defines NFC Forum mandated tag types
Logical Link Control Protocol (LLCP)	Supports P2P operation for NFC applications

Figure 3.6 N-Mark trademark for NFC devices.

embedded applications on secure smart cards. The goal of the GlobalPlatform specifications is to ensure interoperability on content management of smart cards, managing smart cards without any dependencies on hardware, manufacturers, or applications. The GlobalPlatform specifications are related to cards, card terminals and global management of systems using smart cards. For more detailed information, please visit the website (http://www.globalplatform.org).

3.2.3 GSM Association (GSMA)

GSMA is an association of mobile operators and closely related companies devoted to supporting the standardization, deployment and promotion of GSM (Global System for Mobile) technology. Actually, GSMA represents the interests of all organizations in the mobile communications industry. GSMA unites 800+ of the world's mobile operators and 200+ additional companies within the mobile ecosystem. These companies are handset makers, software companies, equipment providers, Internet companies, and media and entertainment organizations. GSMA is focused on innovating and creating new business opportunities for its members as well. The main objective of GSMA is to enable the continuous growth of the mobile communications industry.

GSMA has also been working with large numbers of MNOs to develop a common vision on mobile NFC services by promoting the development of a stable and efficient ecosystem to prevent market fragmentation. Furthermore, GSMA has performed analysis of the UICC to NFC chip interface (Host Controller Interface, HCI), UICC Run-Time Environment, Over-the-Air (OTA) Provisioning, and mobile NFC device security.

The guidelines of GSMA are intended to provide the MNO vision for mobile NFC solutions by identifying technical options and providing recommendations. GSMA has an NFC project, the purpose of which is to define technical guidelines for NFC usage in mobile handsets. GSMA envisages payment, public transport ticketing, and loyalty services as the application areas for NFC enabled GSM. For more detailed information, visit the website (http://www.gsm.org).

3.2.4 International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)

ISO is the world's largest developer and publisher of international standards. It is a network of the standardization institutes worldwide. It is a non-governmental organization that forms a bridge between the public and private sectors. IEC is a non-profit and non-governmental

international organization which provides international standards for all electrical, electronic and other related technologies.

ISO and IEC work together to provide stakeholders with information about standardization, standards and related matters. They provide worldwide standards to make the development and manufacture of products and services more efficient and safer, ensuring the safety of consumers and users, and making life simpler by providing solutions to common problems. For more detailed information, please visit the website (http://www.standardsinfo.net).

3.2.5 ECMA International

ECMA International is an international non-profit and non-governmental standardization organization for information and communications systems. The organization was formed in 1961 to standardize computer systems. It is a membership based private organization. Membership is open to companies of any size that develop, produce or sell computer and communications systems. ECMA's studies include mobile devices and NFC. For more detailed information, please visit the website (http://www.ecma-international.org).

3.2.6 ETSI and ETSI Smart Card Platform (ETSI SCP)

The European Telecommunications Standards Institute (ETSI) is a non-profit organization with 700+ members organizations drawn from 60+ countries across 5 continents worldwide. The ETSI produces globally interoperable and applicable standards for Information and Communications Technologies (ICT) including fixed, mobile, radio, broadcast, and Internet technologies.

On the other hand, ETSI SCP handles Subscriber Identity Module (SIM) specifications which include adopting any hardware and sometimes software changes in SIMs that would enable the cards to carry NFC applications or to play other roles within NFC phones. For more detailed information, please visit the website (http://www.etsi.org).

3.2.7 Java Community Process (JCP)

JCP holds the responsibility for the development of Java technology. The organization primarily guides the development and approval of Java technical specifications. JCP has already adopted two important API specifications on mobile phones for NFC devices [Contactless Communications API (JSR257) and Secure and Trust Service API (SATSA, JSR177)] which will be explained in detail in Chapter 5. For more detailed information, please visit the website (http://jcp.org).

3.2.8 Open Mobile Alliance (OMA)

OMA develops open standards for the mobile phone industry. It was formed in June 2002 by approximately 200 companies including leading MNOs, mobile device and network suppliers, information technology companies, and content and service providers.

OMA supports the development of mobile service enabler specifications and the creation of interoperable end-to-end mobile services. The specifications are generally about the service enabler architectures and open interfaces that are completely independent of underlying wireless networks and platforms. For more detailed information, please visit the website (http://www.openmobilealliance.org).

3.2.9 3rd Generation Partnership Project (3GPP)

3GPP is an organization that was formed by the collaboration of telecommunications associations. Its aim is to make globally applicable third generation (3G) mobile phone system specifications within the scope of the *International Mobile Telecommunications 2000* project of the International Telecommunication Union (ITU). 3GPP specifications are based on GSM specifications. For more detailed information, visit the website (http://www.3gpp.org).

3.2.10 EMVCo

EMV 2000 specifications are open standard set of specifications for smart card based payment systems and seek collaboration in mobile payment standards worldwide. EMVCo, owned by American Express, JCB, MasterCard, and Visa, manages, maintains and enhances the EMV specifications to ensure global interoperability between chip cards and terminals on a global basis regardless of the manufacturer, the financial institution or the card issuer. EMVCo also performs valuable specifications on infrastructure of NFC enabled payment systems in a secure environment. For more detailed information, please visit the website (http://www.emvco.com).

3.3 General Architecture of NFC Enabled Mobile Phones

A mobile device integrated with NFC technology is typically composed of various integrated circuits (ICs), SEs and an NFC interface (see Figure 3.7). The NFC interface is composed of a contactless, analog/digital front-end called an NFC Contactless Front-end (NFC CLF), an NFC antenna and an IC called an NFC controller to enable NFC transactions. Apart from an NFC controller, an NFC enabled mobile phone has at least one SE which is connected to the NFC controller for performing secure proximity transactions with external NFC devices. The SE provides a dynamic and secure environment for programs and data. It enables secure storage of valuable and private data such as the user's credit card information, and secure execution of NFC enabled services such as contactless payments. Also, more than one SE can be directly connected to the NFC controller. The supported common interfaces between SEs and the NFC controller are the Single Wire Protocol (SWP) and the NFC Wired Interface (NFC-WI). The SE can be accessed and controlled from the host controller internally as well as from the RF field externally.

The host controller (baseband controller) is the heart of any mobile phone. A Host Controller Interface (HCI) creates a bridge between the NFC controller and the host controller. An ISO/IEC 7816 interface supports the linkage of SEs to the host controller. The host controller sets the operating modes of the NFC controller through the HCI, processes data that are sent and received, and establishes a connection between the NFC controller and the SE. Furthermore, it maintains the communication interface, peripherals, and the user interface.

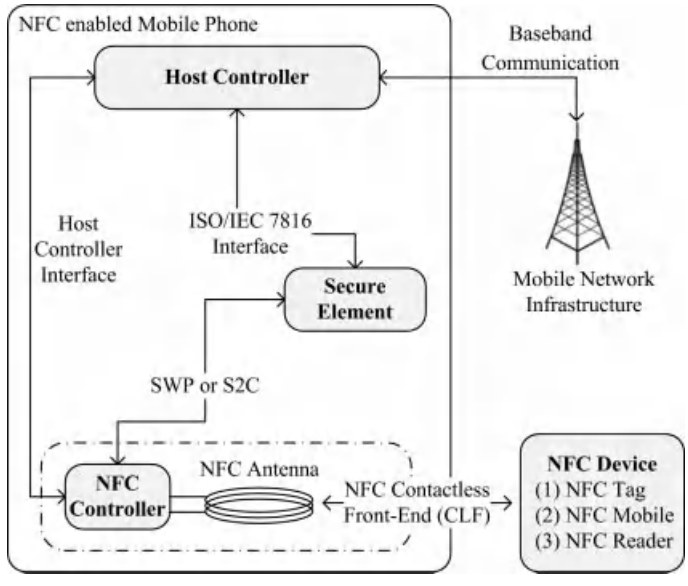


Figure 3.7 Technical architecture of NFC enabled mobile phone.

3.3.1 Secure Element

NFC enabled services must reassure users and service providers that the transaction takes place in a protected environment. This protection is achieved by using an SE which provides the security mechanisms required to support various business models. The SE is a combination of hardware, software, interfaces, and protocols embedded in a mobile handset that enables secure storage.

An SE needs to have an operating system as usual. The operating system (e.g., MULTOS, JavaCard OS) supports the secure execution of applications and the secure storage of application data. The operating system may also support the secure loading of applications.

If NFC enabled applications are saved and executed in the memory of the NFC enabled mobile phone's host controller, these applications are not protected against unintentional deletion or intentional manipulation of the saved data in the memory. They only transmit data between NFC enabled mobile phones or collect information from smart posters. In contactless ticketing, payment and other similar application cases, security is an important issue. These applications use valuable data, and the storage of valuable, private information (e.g., credit card information) in an unsecured memory is unacceptable. The data could be transmitted via a GSM interface to a third party who may misuse the information.

To solve this issue, relevant NFC applications need to be executed and saved in the memory of an SE of the NFC enabled mobile phone (see Figure 3.8). A variety of modules can serve as SEs such as Universal Integrated Circuit Cards (UICCs) (i.e., SIMs), memory cards or embedded hardware. An SE is necessary for various applications such as payment, ticketing, government and other applications where secure authentication and a trusted environment are among the prerequisites.

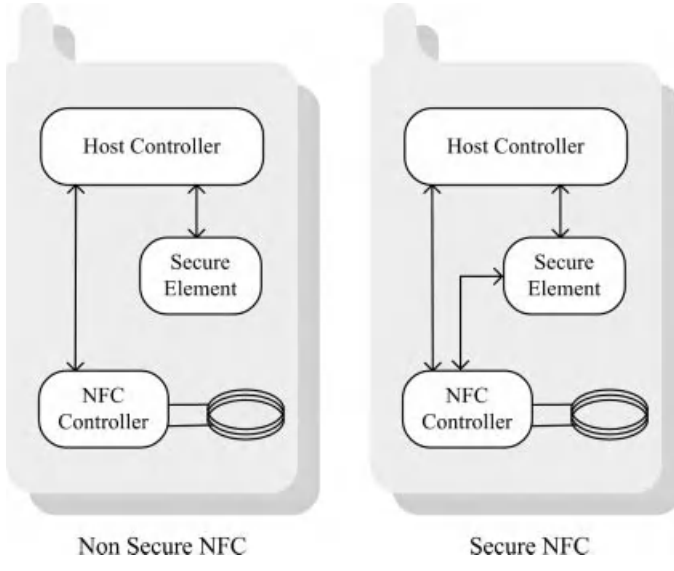


Figure 3.8 Non secure vs. secure NFC [2].

To date, various SE solutions have been introduced to the industry for enabling NFC based systems. Today the most preferred and used SE options for an NFC enabled mobile phone are (see Figure 3.9 and Figure 3.10):

- *Embedded hardware* in a mobile device as an integral, non-removable part of the device.
- *Secure Memory Card (SMC)* as a secure storage area in a removable smart card.
- *UICC* as a physical smart card and maybe the most popular one.

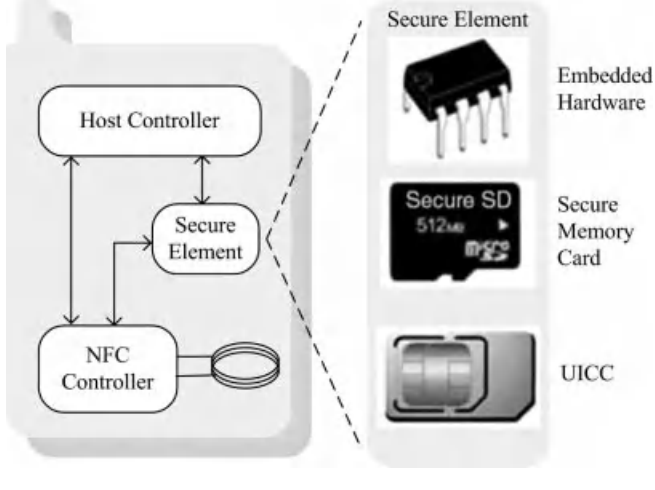


Figure 3.9 Secure element alternatives [2].

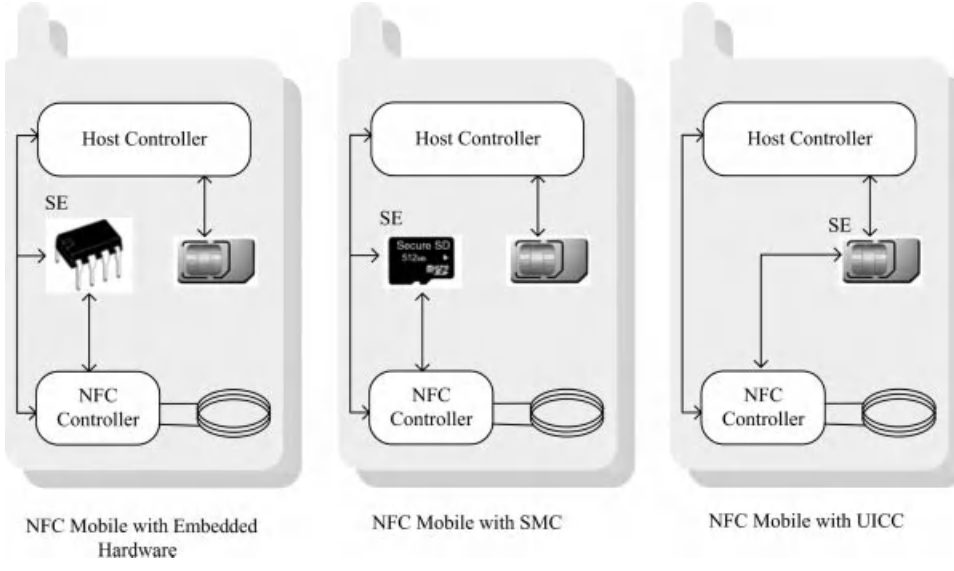


Figure 3.10 Different Design models for secure NFC [2].

(i) *Embedded hardware*

This SE alternative is a smart card soldered onto the mobile phone that cannot be removed. Thus, the level of security provided by this SE is as high as that supported by a smart card. This chip is embedded into the mobile phone during the manufacturing stage and must be personalized after the device is delivered to the end user. Soldered onto the handset, the SE chip obviously cannot be transferred to another mobile. It has to be personalized every time the mobile phone is used by another user. Although the embedded hardware based SE is compliant with all the smart card standards, the communication within the mobile phone is not standardized yet.

(ii) *SMC*

A removable SMC is made up of memory, an embedded smart card element and a smart card controller. In other words, it is a combination of a memory card and a smart card. Thereby, an SMC provides the same high level of security as a smart card and is compliant with most of the major standards, interfaces and environment for smart cards (e.g., EMV, GlobalPlatform, ISO/IEC 7816, JavaCard). With the removable property and a large capacity memory, the SMC based SE can host a large number of applications, and does not need to be reissued when the customer buys a new mobile phone. The SMC based SE can be inserted into the new device easily.

(iii) *UICC*

The UICC is the physical smart card that the SIM or USIM is implemented upon. Therefore it is commonly known as a SIM or USIM. The UICC based SE is compliant with all smart card standards and can host multiple applications issued by different application providers. The UICC ensures the integrity and security of all kinds of personal data. The UICC can host GSM/UMTS applications such as SIM/USIM applications, as

well as non-telecom applications such as payment, loyalty, ticketing, e-passport, and so on.

Today, UICC based SEs provide an ideal environment for NFC applications. They are personal, secure, portable and easily managed remotely via OTA technology (see Chapter 8). The cardholders can be reassured that transactions are executed with their personal information protected. Service providers, who deploy NFC enabled applications, do not risk losing customers when cardholders change handsets.

3.3.2 NFC Interface

The NFC interface is composed of a contactless, analog/digital front-end called an NFC Contactless Front-end (NFC CLF), an NFC antenna and an IC called an NFC controller to enable NFC transactions as shown in Figure 3.11.

The NFC controller enables the NFC link in a mobile phone. It works as a modulator and demodulator between the analog RF signal and the NFC antenna. To connect the NFC antenna to the NFC controller, usually a few passive components are needed such as capacitors, resistors, and inductors. The NFC controller supports both active and passive communication with various modulation types. Typically, an NFC controller is compliant with NFCIP-1 protocol (peer-to-peer mode), and with other two operating modes (reader/writer and card emulation modes). Also, other RFID protocols such as ISO/IEC 15693 are often supported.

NFC CLF is the analog front-end of the NFC controller. The NFC CLF logical interface defines the protocol on top of the data link layer, as well as how the messages are transmitted between the SE and the NFC CLF. It is theoretically independent from the underlying interface (i.e., physical and data link interface) which carries the messages.

All SE design models have an interface between the SE and the NFC controller and also between the host controller and the NFC controller. The data transmitted via the contactless interface are directly forwarded by the NFC controller to the SE and vice versa. The host controller (i.e., the non-secure part of the system) is not involved in the transaction.

3.3.3 Interface between SE and NFC Controller

There are various technical options for designing the interface between the SE and the NFC controller. The most promising two options are NFC-WI and SWP. The most important difference between them is that SWP uses one physical line whereas NFC-WI uses two lines. It is worth mentioning that they are not alternatives to each other but options to be used in certain places instead.

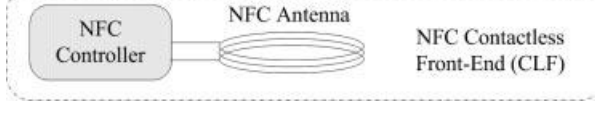


Figure 3.11 NFC interface.

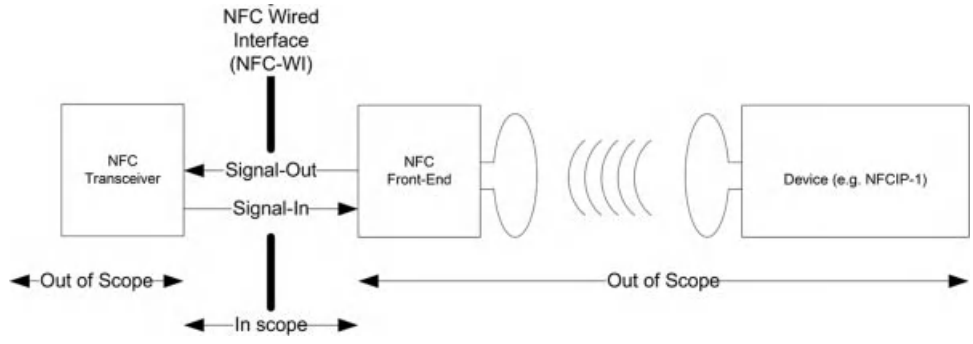


Figure 3.12 NFC-WI architecture [3].

(i) *NFC-WI*

NFC-WI (also called S2C) is a digital wire interface standardized by ECMA 373, ISO/IEC 28361 as well as ETSI TS 102 541. The SE is defined as a transceiver and the NFC controller is defined as front-end in this protocol. The SE is connected to the NFC controller via two wires [3]. NFC-WI defines the Signal-In (SIGIN) and the Signal-Out (SIGOUT) wires between the transceiver and front-end as illustrated in Figure 3.12. In the standard [3], the transceiver is the entity that drives the SIGIN wire and receives on the SIGOUT wire. The front-end is the entity that drives the SIGOUT wire and receives on the SIGIN wire.

This digital wire interface carries two binary signals which are defined as HIGH and LOW. Both of them transmit modulation signals between the NFC controller and the SE and are digitally received or sent by the RF interface. The transceiver drives the SIGIN wire with a binary signal of either HIGH or LOW. The front-end receives the binary signal that is on the SIGIN wire. The front-end drives the SIGOUT wire with a binary signal of either HIGH or LOW. The transceiver receives the binary signal that is on the SIGOUT wire.

Three transmission rates supported by NFC-WI are 106, 212 and 424 kbps. At 106 kbps (see Figure 3.13), the data stream from the NFC controller to the transceiver (SIGIN) shall carry the AND combination of the Modified Miller bit encoded data with 13.56 MHz. In the opposite direction (SIGIN) the data stream is Manchester encoded and then inverted by a logical OR operation with 848 kHz. At 212 and 424 kbps, the data stream from the NFC controller for transceiver (SIGIN) is Manchester encoded and then inverted by a logical XOR operation with 13.56 MHz. This corresponds to a PSK (Phase Shift Keying) modulation of the clock signal. In the opposite direction (SIGIN), the data stream is again Manchester encoded.

NFC-WI is fully compliant and directly coupled with all modes, types and data rates of ISO/IEC 18092 and ISO/IEC 14443, and no additional adaptation and no protocol conversion is required. It is a reliable concept which is feasible for immediate implementation as well.

(ii) *SWP*

The next physical interface option is the SWP which defines a single-wire connection between the SE and the NFC controller in the mobile phone in contrast to the NFC-WI

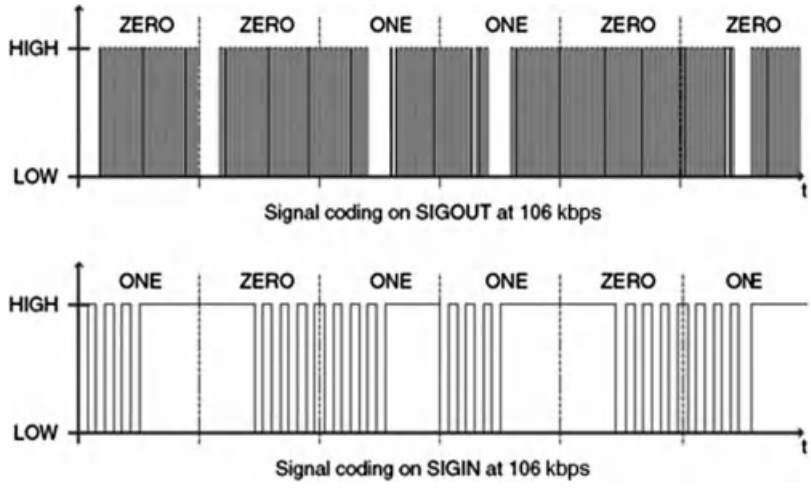


Figure 3.13 Transmission of NFC controller's modulation signals at 106 kbps [2].

double wire connection. SWP is standardized by ETSI TS 102 613. SWP is a digital full duplex protocol [4]. The data rate is scalable from 212 kbps to 1.6 Mbps for a distance that is less than 10 cm. The SWP interface is a bit oriented and point-to-point communication protocol between an SE and an NFC controller as shown in Figure 3.14. The working principle is similar to that of master and slave; the NFC controller is comparable with the master and the SE is comparable with the slave.

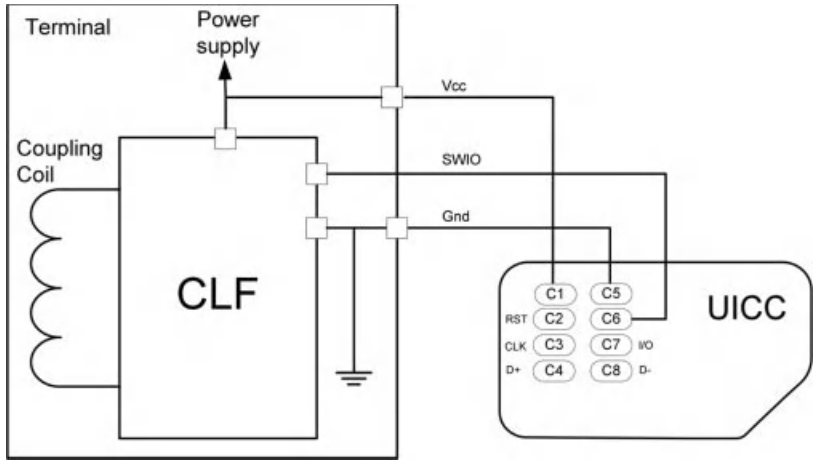


Figure 3.14 CLF-UICC physical link (SWP Architecture). © European Telecommunications Standards Institute 2008. Further use, modification, redistribution is strictly prohibited. ETSI standards are available from http://pda.etsi.org/pda/.

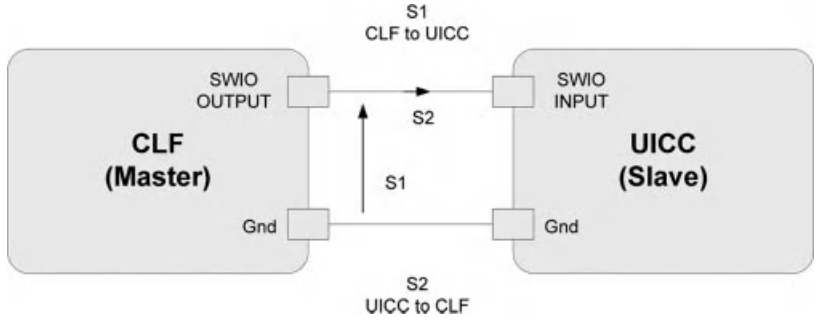


Figure 3.15 SWP data transmission. © European Telecommunications Standards Institute 2008. Further use, modification, redistribution is strictly prohibited. ETSI standards are available from http://pda.etsi.org/pda/.

The SWP is mainly intended to be used by UICC cards in mobile phones since only one of the standard eight contact paths is available for the SWP function. A special case occurs here as shown in Figure 3.14, so that the voltage (Vcc) of the UICC card is not directly supplied by the mobile phone, but supplied through the NFC interface instead. This is necessary for contactless data transmission with SEs even when the battery is exhausted. If the NFC interface is close to an NFC reader, the reader field supplies energy to the SE through the NFC interface. Remember that the NFC mobile operates in the passive mode in this case.

This way the NFC controller and the SE can be used in card emulation mode. Thus card emulation mode applications which have a high operational security and contactless functionality requirement no longer rely on the battery's charging state.

The data to be transmitted are represented by the binary states of voltage (S1) and current (S2) on the single wire (see Figure 3.15). The data transmission from the NFC interface to the SE is carried out by modulating signal S1. In the reverse direction, the data are transmitted by modulating signal S2.

On top of the physical layer, a bit oriented Medium Access and Control (MAC) layer based HDLC (High-Level Data Link Control) is implemented. The HDLC protocol is used for controlling data transmission between the NFC interface and the SE. The HDLC protocol is standardized as ISO/IEC 13239 which is one of the oldest communication protocols. It provides efficient error detection and correction, sign synchronization, and flow control.

SWP carries short packets (payload less than 30 bytes) to the application layer and allows the routing of messages to different components within the mobile. Using short packets and pipelining allow communication with low latency between the SE and the NFC interface. It uses a tunneling technique, so that any frame can be encapsulated in a SWP frame from/to the SE [5].

3.3.4 Host Controller and HCI

The HCI is a logical interface which allows an NFC interface to communicate directly with an application processor and multiple SEs. The HCI may be used in various electronic devices

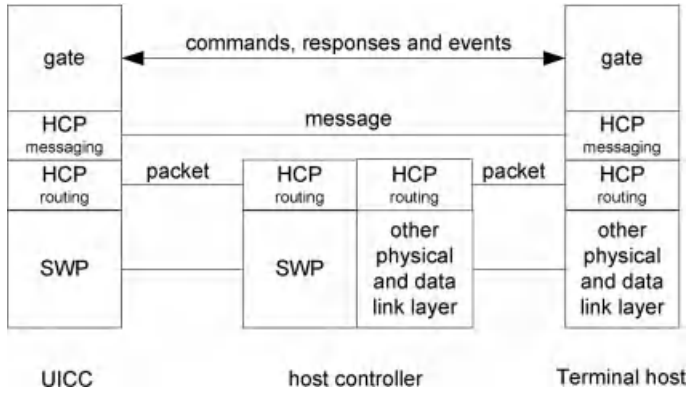


Figure 3.16 HCP Stack in a host network. © European Telecommunications Standards Institute 2008. Further use, modification, redistribution is strictly prohibited. ETSI standards are available from http://pda.etsi.org/pda/. Note: For clarity, only two gates are shown. Host controller has also gates to connect via HCP to other hosts.

such as mobile devices, PDAs and PC peripherals. For NFC enabled mobile phones, it enables faster integration of NFC functionality. The HCI is standardized in ETSI TS 102 622 [6].

The HCI defines the interface between logical entities called *hosts* that operate one or more service(s). According to the ETSI terminology, a network of two or more hosts is called a *host network*; and one of the hosts that is also responsible for managing a host network is called a *host controller*. In a host network which has a star topology, all hosts are connected to the host controller. The HCI has three components: a collection of gates that exchange commands, responses, and events; a Host Controller Protocol (HCP) messaging mechanism; and an HCP routing mechanism that may optionally segment messages when required (see Figure 3.16).

Data link layers of the HCP need to be error free so that the order of sent and received data can be trusted. The data link layer should also enable data flow control, fragment the upper layer's packets to a maximum size, and report each received packet's size to its upper layer.

(i) *Gates*

A gate provides an entry point for the services operating in a host. HCP enables gates from several hosts to exchange messages. There are two types of gates: management gates that host network management, and generic gates.

Each gate type has a gate identifier which is either unique (values "10" to "FF") or not (values "00" to "0F"). Each host must have several pre-defined gates, including administration, identity management, link management, and loop back gates.

(ii) *Pipes*

Gates communicate with each other through logical communication channels called pipes. There are two types of pipes: static pipes and dynamic pipes. Static pipes do not need to be created and cannot be deleted; however, dynamic pipes can be created and deleted. The state of a pipe can be either opened or closed. The state of the pipe

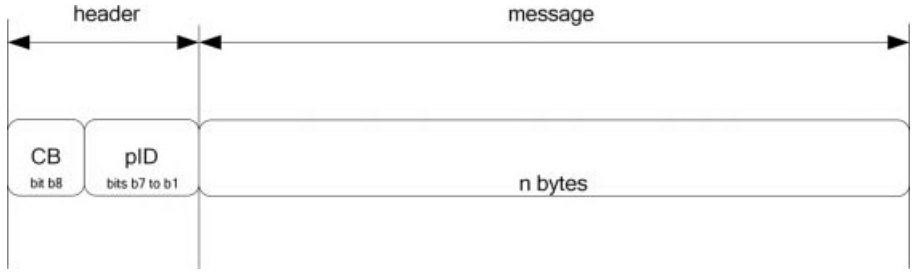


Figure 3.17 HCP packets. © European Telecommunications Standards Institute 2008. Further use, modification, redistribution is strictly prohibited. ETSI standards are available from http://pda.etsi.org/pda/.

remains persistent even if the host is powered down and up or a host is removed from the network.

The length of the pipe identifier (pID) is 7 bits. It is used in the header of HCP packets for routing information. Pipe identifiers of dynamic pipes are dynamically defined by the host controller, whereas pipe identifiers of static pipes are pre-defined.

(iii) *HCP packets*

The HCP packets are exchanged between the hosts and the host controller (see Figure 3.17); and it contains the following segments:

- *CB* is the chaining value which takes value "1" for the last packet of a fragmented message or value "0" for a packet that belongs to a fragmented message.
- *pID* is the pipe identifier.
- *Message* carries one instruction and optional data as depicted in Figure 3.18. TYPE identifies the type of instruction. INSTRUCTION can be a command (TYPE = 0), an event (TYPE = 1) or a command response (TYPE = 2).

(iv) *Registries*

Registries are 1 byte long identifiers and are used to define unique parameters related to every gate. The host is responsible for management of its registries. For every pipe, a new registry instance is created and registry parameters are set. When a pipe is removed from the network, its registry instance is removed too.

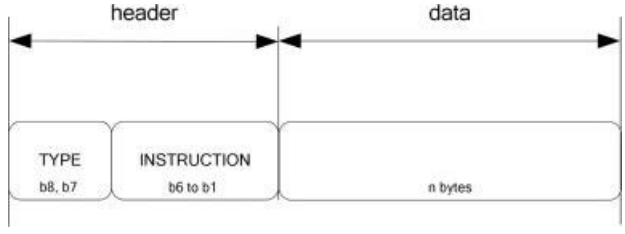


Figure 3.18 HCP message structure. © European Telecommunications Standards Institute 2008. Further use, modification, redistribution is strictly prohibited. ETSI standards are available from http://pda.etsi.org/pda/.

(v) *HCI procedures*

The following procedures are defined in ETSI TS 102 622:

- *Pipe creation* which describes how a host requests dynamic pipe creation between one of its gates and a gate in another host.
- *Registry access* which defines how a host can read/write parameters in the registry of another host.
- *Host and gate discovery* which defines how a host discovers other hosts in a host network as well as the gates.
- *Session initialization* which defines how a host can detect changes in recovery mechanisms.
- *Loop back testing* which defines how a host can verify the pipe connectivity to another host.

3.4 Physical Layer of NFC

NFC technology is based on RFID technology in a proximity range at 13.56 MHz. NFC transfers data at up to 424 kbps. The communication between two NFC devices is standardized in ISO/IEC 18092 standard as NFCIP-1. This standard defines only device-to-device communication for both active and passive communication modes. However, the RF layer of NFC is a super set of the standard protocols which is also compatible with the ISO/IEC 14443 standard (i.e., contactless proximity smart card standard) and JIS X 6319 standard as FeliCa (i.e., another contactless proximity smart card standard by Sony) as well as ISO/IEC 15693 standard (i.e., contactless vicinity smart card standard). These smart card interfaces similarly operate at 13.56 MHz from card reader to smart card with distinct data rates and communication ranges. Table 3.5 gives a short summary and comparison of ISO/IEC 14443, ISO/IEC 15693, and ISO/IEC 18092 communication interfaces. In this section, background knowledge about the proximity range communication interfaces used by NFC technology is given and then data transmission features of the RF layer are explained.

3.4.1 ISO/IEC 14443 – Proximity Contactless Smart Card Standard

As described in ISO/IEC 14443, proximity transactions are based on an electromagnetic coupling between a proximity card and RFID reader which uses an embedded microcontroller (including its own processor and one of several types of memory) and a magnetic loop antenna that operates at 13.56 MHz. ISO/IEC 14443 enables contactless transactions between a reader device and a proximity card used for identification. Usually this proximity card uses the

Table 3.5 Summary of communication interface standards

Parameters	ISO/IEC 18092	ISO/IEC 14443	ISO/IEC 15693
Operating Mode	Peer-to-peer	Reader-to-card	Reader-to-card
Communication Mode	Active and passive	Passive	Passive
Range	Proximity	Proximity	Vicinity
Data Rate	106, 212, 424 kbps	106 kbps	$\leq$26 kbps

Table 3.6 Parts of ISO/IEC 14443 standard

Part Name	Content
Part 1- Physical Characteristics of Contactless Smart Cards (PICC)	Defines physical characteristics of a contactless smart card, lists several requirements and tests that need to be performed at card level for construction of the card and antenna design and so on.
Part 2 - RF Power and Signal Interface	Defines the RF power and signal interface, Type A and Type B signaling schemes, and also determines how the card is powered by RF field and so on.
Part 3 – Initialization and Anti-collision	Defines the initialization and anti-collision protocols for Type A and Type B, as well as anti-collision commands, responses, data frame and timing
Part 4 – Transmission Protocol	Determines the high-level data transmission protocols for Type A and Type B which are optional protocols, thus proximity cards may be designed with or without support for Part 4 protocols

standard credit card form factor defined by ISO/IEC 7810 ID-1, but other form factors – such as tokens or key fobs – are also possible. ISO/IEC 14443 uses the terms PICC (Proximity Integrated Circuit Card) and PCD (Proximity Coupling Device) to describe the components taking part in a transaction. PCD refers to a proximity card reader whereas PICC refers to a proximity card.

The ISO/IEC 14443 standard contains four major parts (see Table 3.6): the physical characteristics are explained in the first part, RF power and signal interface are explained in the second part, initialization and anti-collision protocols constitute the third part, and the transmission protocol is defined in the fourth part [2]. It also defines two major contactless cards, namely Type A and Type B.

3.4.1.1 Major Operating Principles of ISO/IEC 14443

Contactless proximity smart cards operating at 13.56 MHz are powered by and communicate with the reader via inductive coupling of the reader antenna to the card antenna. An alternating magnetic field is produced by sinusoidal current flowing through the reader antenna loop. When a card enters the alternating magnetic field generated by a reader, an alternating current (AC) is induced in the card loop antenna. The PICC IC contains a rectifier and powers the regulator to convert the AC to direct current (DC) to power the IC. The reader amplitude modulates the RF field to send information to the card. The IC contains a demodulator to convert the amplitude modulation to digital signals. The IC also contains a clock extraction circuit that produces a 13.56 MHz digital clock signal for use within the IC. The data from the reader are clocked in, decoded, and processed by the IC. The IC communicates with the reader by modulating the load on the card antenna and the load on the reader antenna. ISO/IEC 14443 PICC uses a 847.5 kHz subcarrier for load modulation which allows the reader to filter the subcarrier frequency off the reader antenna and decode the data.

3.4.1.2 Major Proximity Contactless Smart Card Technologies

Up to now, various proximity coupling smart card technologies have emerged; however only a few of them are compatible with ISO/IEC 14443 standard. Currently, the most famous and competing contactless smart cards are MIFARE, Calypso, and FeliCa:

(i) *MIFARE*

MIFARE is a well-known and widely used 13.56 MHz contactless proximity smart card system that is being developed and is owned by NXP semiconductors which is a spin-off company of Philips Semiconductors. MIFARE is ISO/IEC 14443 Type A standard. As of 2011, MIFARE is used in more than 80% of all contactless smart cards in the world. The MIFARE family contains different types of cards such as Ultralight, Standard, Desfire, Classic, Plus, and SmartMX [7]. MIFARE Classic cards have varying memory sizes. MIFARE based smart cards are being used in an increasingly broad range of applications mostly in public transport ticketing and also for access management, e-payment, road tolling, and loyalty applications.

(ii) *FeliCa*

FeliCa is a 13.56 MHz contactless proximity high speed smart card system from Sony and is primarily used in electronic money cards [8]. The name FeliCa comes from the word "felicity," suggesting that the technology will make our daily lives more convenient and enjoyable. Sony applied for standardizing FeliCa into ISO/IEC 14443 as Type C standard, but failed. FeliCa eventually did not become an ISO/IEC standard. FeliCa currently only complies with Japanese Industrial Standard (JIS) X 6319 Part 4 which defines high speed proximity cards.

(iii) *Calypso*

Calypso is another example of a contactless proximity smart card which is also international public transportation standard [9]. It was originally designed by a group of European transit operators from Belgium, Germany, France, Italy, and Portugal. It enables interoperability between several transport operators in the same area. Calypso was created in 1993 by a partnership between transit operator RATP and Innovatron Group. The creation of this technology was patented to allow enforcement of a technical interoperability and to finance the developments through an open license scheme. Both the ISO/IEC 14443 Type B standard and the European standard EN 1545 which defines the ticketing data for smart cards are direct results of this work.

3.4.2 Near Field Communication Interface and Protocol (NFCIP)

NFCIP is standardized in two forms as NFCIP-1which defines the NFC communication modes on the RF layer and other technical features of the RF layer; and NFCIP-2 which supports mode switching by detecting and selecting one of the communication modes.

3.4.2.1 NFCIP-1

NFCIP-1 is standardized in ISO/IEC 18092, ECMA 340, and ETSI TS 102 190. This standard defines two communication modes as active and passive. It also defines the RF field, RF communication signal interface and general protocol flow, initialization conditions for the

supported data rates of 106, 212, and 424 kbps in detail. Moreover, it defines transport protocol including protocol activation, data exchange protocol with frame architecture and error detecting code calculation (CRC for both communication mode at each data rate), and protocol deactivation methods [10].

The carrier frequency of the RF field is 13.56 MHz. The minimum unmodulated RF field is denoted by H_{min} and has a value of 1.5 A/m rms. The maximum unmodulated RF field is denoted by H_{max} and has a value of 7.5 A/m rms. This RF field needs to be modulated during communication.

As mentioned before for the active communication mode, both the initiator and target use their own RF field to enable communication. The initiator starts the NFCIP-1 communication whereas the target responds to an initiator command in the active communication mode using self-generated modulation of the self-generated RF field. The target is powered by inductive coupling and is able to send and receive data. In the passive communication mode, the initiator generates the RF field and starts the communication. The target responds to an initiator command in the passive communication mode using a load modulation scheme as described in Chapter 2.

The communication scheme over the RF interface in active and passive communication modes includes modulation schemes, transfer speed and bit coding. Additionally it includes start of communication, end of communication, bit and byte representation, framing and error detection, single device detection, protocol and parameter selection, data exchange and de-selection of NFCIP-1 devices.

All NFCIP-1 devices have communication capability on 106, 212, or 424 kbps and may switch to another transfer speed or stay on the same transfer speed. The transfer speed of the initiator to target and the transfer speed of the target to the initiator do not need to be kept the same during a transaction. The change of transfer speed during a transaction session may be performed by a parameter change procedure. The mode (active or passive) cannot be changed within one transaction session. The transaction is started by device initialization and terminated by device de-selection (or equivalent).

3.4.2.2 NFCIP-2

NFCIP-2 is a standard specified in ISO/IEC 21481, ECMA 352 and ETSI TS 102 312. The standard specifies the communication mode selection mechanism and is designed not to disturb any ongoing communication at 13.56 MHz for devices implementing ISO/IEC 18092 (i.e., NFCIP-1), ISO/IEC 14443 (e.g., MIFARE) or ISO/IEC 15693 (e.g., long range-vicinity communication, RFID tags) [11]. Although all of the ISO/IEC 18092, ISO/IEC 14443 and ISO/IEC 15693 standards specify 13.56 MHz as their working frequency, they may specify distinct communication modes. These are defined as NFC, PCD, PICC and VCD communication modes. This is achieved using Carrier Sense Multiple Access (CSMA), hence an NFCIP-2 device will not activate its RF field when it detects an external RF that exceeds a certain threshold [11].

Devices following NFCIP-2 need to implement the functions of the proximity coupling device (ISO/IEC 14443), vicinity coupling device (ISO/IEC 15693) and the initiator and the target functions defined in ECMA-340. This makes NFC devices compatible with existing commercially deployed FeliCa and MIFARE systems. However, compatibility is not achieved

on the smart card emulation side for the standards ISO/IEC 14443 Type B and ISO/IEC 15936, although read-out and editing is possible [5].

3.4.3 Data Transmission on RF Layer

The reader/writer mode allows data connection only at 106 kbps and relies on the RF interface that is compliant with the ISO/IEC 14443 (Type A, Type B) and FeliCa schemes. In peer-to-peer mode, the RF interface that allows all data connections such as 106, 212, and 424 kbps is based on the ISO/IEC 18092 (NFCIP-1) standard. In card emulation mode, the RF interface is based on the ISO/IEC 14443 (Type A, Type B) standard and FeliCa. The Type B is especially used for highly secure transactions such as contactless mobile payments and ticketing. In this section the modulation and coding techniques used by NFC are explained.

(i) *Modulation*

Like the RFID standards 14443 and FeliCa, NFC uses inductive coupling. The operating frequency is 13.56 MHz, and commonly a bit rate of 106 kbps (partly also 212 kbps and 424 kbps) is used. Modulation schemes used by NFC are ASK (Amplitude Shift Keying) with different modulation depth (100% or 10%) and load modulation:

- In the case of data transmission *from the initiator to the target* such as an NFC enabled mobile phone in card emulation mode, the target device uses 13.56 MHz carrier signal of the initiator device as energy source. The modulation scheme of the initiator device is ASK modulation. In peer-to-peer mode, both directions are modulated and coded like an initiator device. However, less power is required because both active NFC devices use their own power supply, generate their own RF field, and the carrier signal is switched off at the end of transmission.
- In the case of data transmission *from the target to the initiator,* due to the coupling of the coils of initiator and target devices, the passive target device also affects the active initiator device. A variation in the impedance of the target device causes amplitude or phase changes on the antenna voltage of the initiator device, which is detected by the initiator device. This technique is called load modulation. Load modulation is carried out in listening mode using an auxiliary carrier at 848 kHz which is modulated by the baseband and varies the impedance of the target device.

(ii) *Coding*

NFC employs three different coding techniques to transfer data: *NRZ-L, Manchester*, and *Modified Miller coding* (see Figure 3.19):

- In NRZ-L coding: a high state during one bit duration refers to logic 1 and a low state refers to logic 0.
- In Manchester coding: at logic 1, the first half of a bit is set to high state, and the second half of that bit is set to low state. At logic 0, the first half of a bit is set to low state and the second half is set to high state.
- In Modified Miller coding: at logic 1, a low pulse occurs after half of the bit duration. At logic 0, a low pulse occurs at the beginning of a bit. If logic 0 comes after logic 1, no pulse occurs at logic 0, hence the signal remains high.

In Manchester and Modified Miller coding schemes a single data bit is sent in a fixed time slot. This time slot is divided into two halves, called half bits. In Miller coding a 0 is encoded

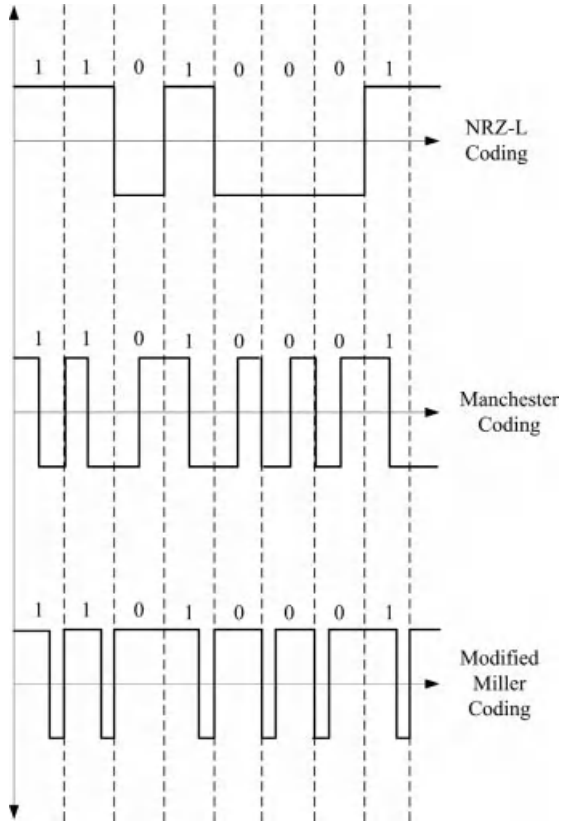


Figure 3.19 Coding with NFC is either NRZ-L Coding, Manchester Coding, or Modified Miller Coding.

with a pause in the first half bit and no pause in the second half bit. A 1 is encoded with no pause in the first bit, but a pause in the second half bit. In the Modified Miller coding some additional rules apply on the coding of 0s. In the case of 1 followed by 0, two subsequent half bits would have a pause. Modified Miller coding avoids this by encoding 0 which directly follows 1 with two half bits with no pause. In Manchester coding the situation is nearly the same, but instead of having a pause in the first or second half bit, the whole half bit is either paused or modulated. Besides the coding scheme, the strength of the modulation also depends on the baud rate. For 106 kbaud, 100% modulation is used. This means that in a pause the RF signal is actually zero. No RF signal is sent in a pause. For baud rates greater than 106 kbaud a 10% modulation ratio is used. This difference in the modulation strength is very important from a security point of view and will be described later in the security section [12].

3.4.3.1 ISO/IEC 14443 Type A

Type A signaling utilizes 100% ASK modulation of the RF field for communication from the reader (PCD) to the card (PICC) with Modified Miller encoded data. Communication

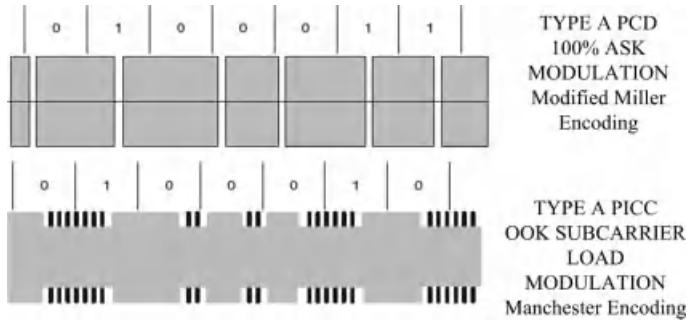


Figure 3.20 Summary of ISO/IEC 14443 Type A interface [2].

from card to reader utilizes OOK Subcarrier (847.5 kHz subcarrier) Load Modulation with Manchester encoded data. In Type A signaling, the RF field is turned off for short periods of time when the reader is transmitting (see Figure 3.20).

3.4.3.2 ISO/IEC 14443 Type B

Type B signaling utilizes 10% ASK modulation of the RF field for communication from the reader (PCD) to the card (PICC) with NRZ-L encoded data. Communication from card to the reader utilizes BPSK (Binary Phase Shift Keying) Subcarrier Load modulation of an 847.5 kHz subcarrier with NRZ-L encoded data (see Figure 3.21).

3.4.3.3 JIS X 6319 Type F

Another important standard which is provided by Japanese Industrial Standard is JIS X 6319 Type F. JIS X 6319 uses Manchester coding in both transfer directions. The bandwidth needed to carry the data is twice as much as the data frequency. This technology is standardized for 212 kbps and 424 kbps only [5].

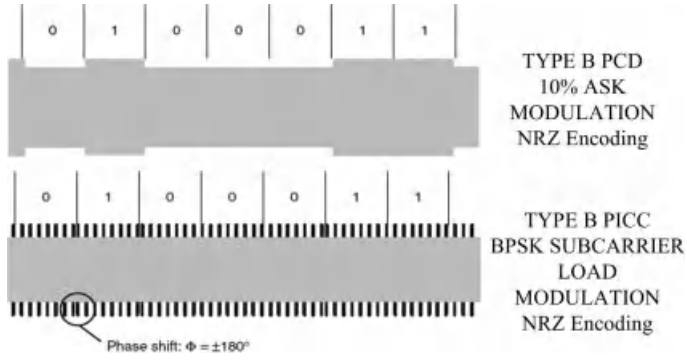


Figure 3.21 Summary ISO/IEC14443 Type B interface [2].

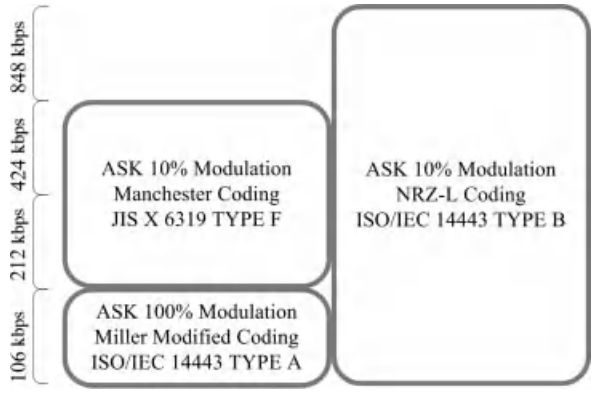


Figure 3.22 Coding and modulation from PCD to PICC [5].

3.4.3.4 Summary

ISO/IEC 18092 (NFCIP-1) is the combination of ISO/IEC 14443 Type A and JIS X 6319 Type F. The ISO/IEC 14443 Type A high speed has been standardized since 2005 but not integrated to the NFCIP-1. Thus NFCIP-1 does not define coding and modulation based on the ISO/IEC 14443 Type A greater than 106 kbps (see Figures 3.22 and 3.23). For high speed data transfers from PICC to PCD and vice versa, Type B is used.

3.5 Reader/Writer Operating Mode Essentials

In reader/writer operating mode, an active NFC enabled mobile phone initiates the wireless communication, and can read and alter data stored in NFC tags. In this operating mode, an NFC enabled mobile phone is capable of reading NFC Forum mandated tag types, such as NFC smart poster tags. This enables the mobile user to retrieve the data stored in the tag and take appropriate actions afterwards.

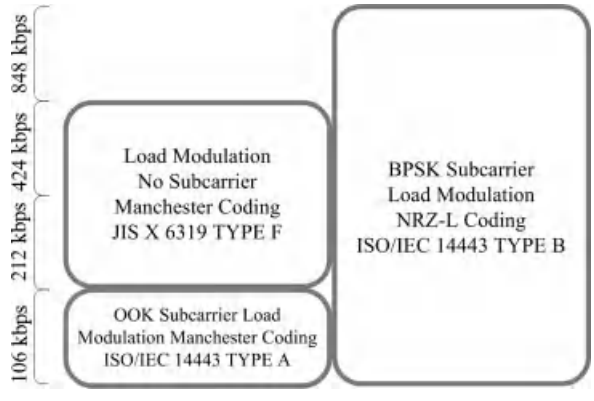


Figure 3.23 Coding and modulation from PICC to PCD [5].

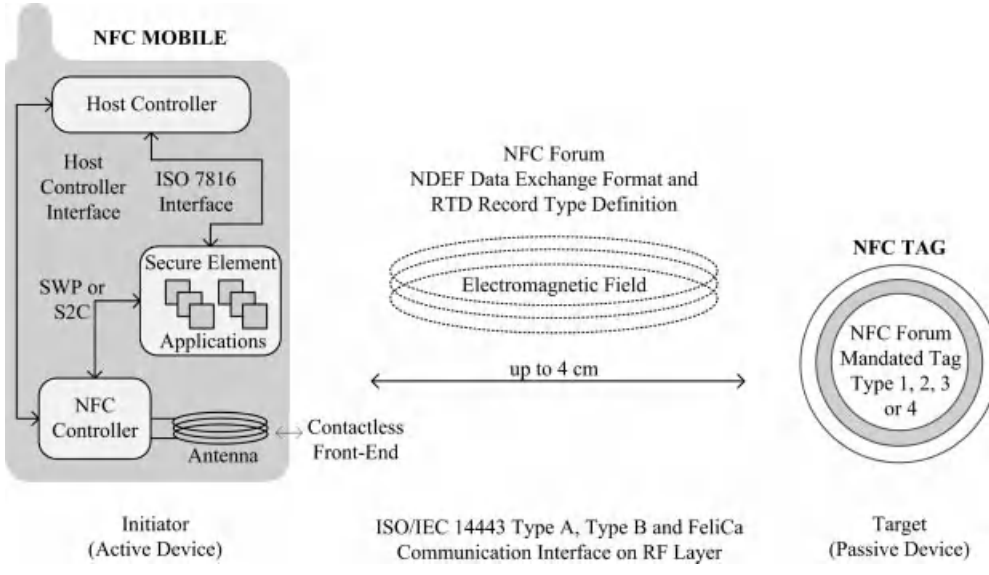


Figure 3.24 Communication architecture of reader/writer operating mode.

As illustrated in Figure 3.24, the reader/writer mode's RF interface is compliant with ISO/IEC 14443 Type A, Type B and FeliCa schemes. NFC Forum has standardized tag types, operation of tag types and data exchange format between components. The reader/writer operating mode usually does not need a secure area, SE in other words, of the NFC enabled mobile phone. The process consists of only reading data stored inside the passive tag and writing data to the passive tag. The protocol stack architecture of the reader/writer operating mode, the NDEF and record types are explained in Sections 3.5.1 and 3.5.2.

3.5.1 Protocol Stack Architecture of Reader/Writer Mode

Figure 3.25 provides a useful protocol stack illustration of the reader/writer mode. The NFC device operating in reader/writer mode has the following protocol stack elements:

- Analog and digital protocols are the lower protocols which have been explained in Section 3.4. Analog is related to RF characteristics of NFC devices and determines the operating range of devices. Digital protocols refer to the digital aspects of ISO/IEC 18092 (see Section 3.4.2.1) and ISO/IEC 14443 (see Section 3.4.1) standards, and define building blocks of the communication. There is also another important specification by NFC Forum at this level which is the NFC Activities Specification [13]. This specification defines the required activities that set up communication in an interoperable manner based on digital protocol specification such as polling cycles, when to perform collision detection.
- Tag operations indicate the commands and instructions used by NFC devices to operate NFC Forum mandated tags which are Type 1, Type 2, Type 3, and Type 4. They enable read and

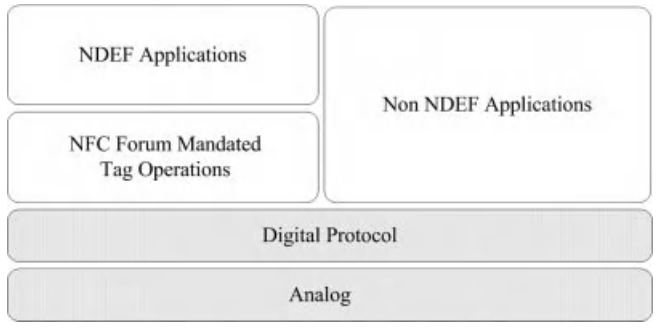


Figure 3.25 Protocol stack of reader/writer operating mode.

write operations by using the NDEF data format and RTDs (i.e., smart poster, URI RTDs) from/to a tag.
- NDEF applications are based on NDEF specifications such as smart poster and reading product information from NFC enabled smart shopping fliers.
- Non NDEF applications are vendor specific applications such as an electronic purse balance reader and contactless ticket reader which are not based on NDEF specifications.

3.5.2 NFC Forum Mandated Tag Types

Four tag types have been defined by NFC Forum, and are given designations between 1 and 4. Each tag type has a different format and capacity. NFC tag type formats are based on either ISO/IEC 14443 Type A, ISO/IEC 14443 Type B, or Sony FeliCa. Different NFC tag type definitions are listed below and Table 3.7 gives a comparison of the tag types.

- *Type 1 tag:* NFC Type 1 tag is based on the ISO/IEC 14443 Type A standard. These NFC tags are both readable and writable. Data on these tags may also be modified and users can configure the tag to make it read-only when required. Memory availability is up to 1 kB which is just enough to store a website URL or similar small amount of data. The memory size can be expanded up to 2 kB. The communication speed of this NFC tag is 106 kbps. As a result of its simplicity, this tag type is cost effective and can be used in many NFC applications.
- *Type 2 tag:* NFC Type 2 tag is also based on ISO/IEC 14443 Type A standard. These NFC tags are also both readable and writable, and users can configure the tag to make it read-only when required. The memory size of this tag type can be expanded up to 2 kB. Similarly the communication speed is 106 kbps.
- *Type 3 tag:* NFC Type 3 tag is based on the Sony FeliCa contactless smart card interface. It currently has a 2 kB memory capacity and the data communications speed is 212 kbps. This tag type is more suitable for complex applications. However, it is more expensive compared with other tag types.
- *Type 4 tag:* NFC Type 4 tag is compatible with both ISO14443 Type A and Type B standards. These NFC tags are pre-configured during the manufacturing phase, and they are either writable or read-only; and the type is defined during the manufacturing phase. The

Table 3.7 Summary of NFC Forum tag types

Parameter	Type 1	Type 2	Type 3	Type 4
Based On	ISO/IEC 14443 Type A	ISO/IEC 14443 Type A	FeliCa	ISO/IEC 14443 Type A, Type B
Chip Name	Topaz	MIFARE	FeliCa	DESFire, SmartMX-JCOP
Memory Size	Up to 1 kB	Up to 2 kB	Up to 1 MB	Up to 64 kB
Data Rate (kbps)	106	106	212	106–424
Security	16- or 32-bit digital signature	Insecure	16- or 32-bit digital signature	Variable
Provided By	Innovision Research and Technology	Philips/NXP	Sony	Several
Unit Price	Low	Low	High	Medium /high
Use Cases	Tags with small memory for single application		Flexible tags with larger memory offering multi-application capabilities	

memory capacity can be up to 32 kB and the communication speed is between 106 and 424 kbps.

3.5.3 NDEF

NDEF specification is a standard defined by NFC Forum [14]. NDEF is a data format to exchange information between NFC devices; namely, between an active NFC device and passive tag, or two active NFC devices. The NDEF message is exchanged when an NFC device is in the proximity of an NFC Forum mandated tag, the NDEF message is received from the NFC Forum mandated tag, as well as over the NFC Forum LLCP (see Section 3.6). The data format is the same in both cases.

NDEF is a binary message format that encapsulates one or more *application defined payloads* into a single message as depicted in Figure 3.26. An NDEF message contains one or more NDEF records. Each record consists of a payload up to 2^{32}-1 octets in size. Records can be chained together to support larger payloads as well.

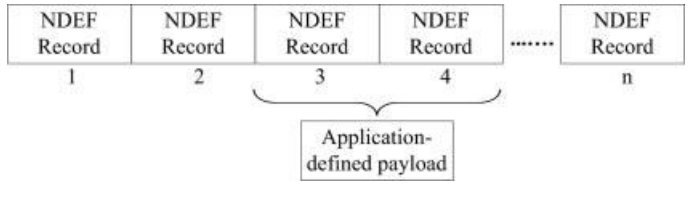


Figure 3.26 NDEF structure.

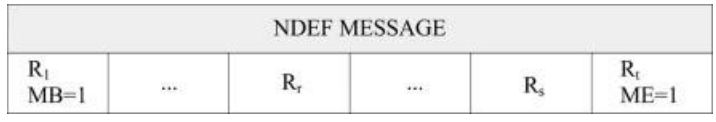


Figure 3.27 NDEF message with a set of records. Reproduced by permission from NFC Forum.

In an NDEF message, the first record is marked with the MB (Message Begin) flag set and the last record is marked with the ME (Message End) flag set (see Figure 3.27). The minimum message length is one record which is achieved by setting both the MB and the ME flag in the same record. In a message, the maximum number of NDEF records that can be carried is unlimited.

The message head starts from left (head) to right (tail). The first flag set (MB) is referred to by the index 1 and the last one (ME) is referred to by the index t. So, the logical record indices are in the order: t>s>r>1(see Figure 3.27).

A record is the unit for carrying a payload within an NDEF message. Each NDEF record carries three parameters for describing its payload: payload length, payload type, and an optional payload identifier. The purposes of using these parameters are as follows:

- The *payload length* indicates the total number of octets in the payload.
- The *payload type* identifier indicates the type of payload. NDEF supports URIs, MIME media type constructs and an NFC specific type format as type identifiers. By indicating the type of a payload, it is possible to dispatch the payload to the appropriate application.
- The *optional payload identifier* allows user applications to identify the payload carried within an NDEF record.

NDEF records are variable length records with a common format illustrated in Figure 3.28. Each individual field in a record has different features. The details of each record can be found in the specifications. The most important ones are the TNF (Type Name Format) field, and the TYPE field. The TNF field value indicates the structure of the value of the TYPE field. The TYPE field describes the type of the payload. TNF covers a 3-bit field and the possible TNF field values are listed in Table 3.8.

Various record types for NDEF messaging format are defined by NFC Forum [15]. The *record type* string field contains the name of the record type as *record type name*. Record type names are used by NDEF applications to identify the semantics and structure of the record content. Record type names can be specified in several formats in the TNF field of the NDEF record header. Record type names may be MIME media types, absolute URIs, NFC Forum external type names or well-known NFC type names. Each record type definition is identified by its record type name:

- *NFC Forum Well-known Type:* This is a dense format designed for tags and creating primitives or common types. It is identified inside an NDEF message by setting the TNF field of a record to the value of 0×01. NFC Forum Well-known Type is a URN with "nfc" Namespace Identifier (NID). The Namespace Specific String (NSS) of NFC Forum Well-Known Type

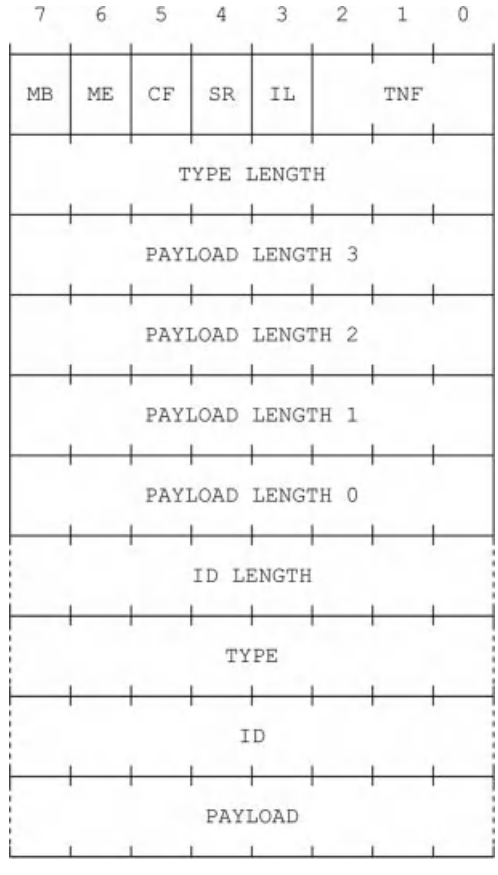


Figure 3.28 NDEF record. Reproduced by permission from NFC Forum.

is prefixed with "wkt:" (see Table 3.9). When it is encoded in NDEF message, it must be written as a relative-URI construct where the NID and "wkt:" prefixes are discarded. For instance, "urn:nfc:wkt:very-complicated-type" as a NFC Forum Well-known Type will be encoded as "very-complicated-type". The two major types are as follows:

- NFC Forum Global Type: NFC Forum defines and manages this type. NFC Forum Global Types start with an upper-case letter in character set which can be found in RTD specification. For instance, "A", "Trip-to-Istanbul".
- NFC Forum Local Type: NFC Forum Local Types start with a lower-case character in character set or with a number in character set which can be found in RTD specification. For instance, "2", "a", "trip". An NFC Forum Local Type can be reused by another application in a different context and with a different content.

- *NFC Forum External Type:* The External Type name is identified inside an NDEF message by setting the TNF field or a record to the value of 0×04. NFC Forum External Type is a URN with "nfc" NID. The NSS part is prefixed with "ext:". An example might

Table 3.8 Possible TNF field values

Type Name Format (TNF)	Value
Empty	0×00
NFC Forum Well-Known Type	0×01
Media Type	0×02
Absolute URI	0×03
NFC Forum External Type	0×04
Unknown	0×05
Unchanged	0×06
Reserved	0×07

be "urn:nfc:ext:yourcompany.com:f". As with NFC Forum Well-known Types, the binary encoding of NFC Forum External Type discards the NID and NSS prefix of "ext:".

NFC Forum defines various record types for specific cases; smart posters, URIs, digital signature, and text. These RTDs can be found on NFC Forum's website in detail. A short description for each one is given below:

(i) *URI RTD*

The URI Service RTD defines a record to be used with the NDEF to retrieve a URI stored on an NFC tag or to transport a URI from one NFC device to another [16]. The Well-Known Type for a URI record is "U" (0×55 in the NDEF binary representation). The structure of a URI record is shown in Table 3.10. There are 256 URI identifier codes, the details of which can be found in URI RTD specification of NFC Forum. Some examples of URI identifier codes are shown in Table 3.11. The URI field is defined by the UTF-8 coding which is also called NFC binary encoding in specifications.

Tables 3.12 and 3.13 give examples of storing different URIs in tags. Table 3.12 shows a simple example on launching NFC Lab - Istanbul's website "http://www.nfclab.com" where the URI identifier code field is 0×01. If the content of this field is 0×02, and the content of the URI field reads as "nfclab.com", the resulting URI is "https://www.nfclab.com". Table 3.13 is about storing a mail address on a tag. The content of the URI field reads as "info@nfclab.com"; the resulting URI is "mailto:info@nfclab.com".

Table 3.9 Well-known NDEF record type examples

NDEF Record Type	Description	URI Reference
Sp	Smart Poster	urn:nfc:wkt:Sp
T	Text	urn:nfc:wkt:T
U	URI	urn:nfc:wkt:U
Sig	Signature	urn:nfc:wkt:Sig

Table 3.10 URI record contents

Name	Value	Description
URI Identifier Code	URI identifier code	The URI identifier code (256 codes)
URI Field	UTF-8 encoded data	The rest of the URI, or the entire URI (if identifier code is 0 × 00)

Table 3.11 Some examples of URI identifier codes

Decimal	Hex	Protocol Type
0	0 × 00	Not available. The URI field contains the unabridged URI
1	0 × 01	http://www.
2	0 × 02	https//www.
5	0 × 05	tel:
6	0 × 06	mailto:
8	0 × 08	ftp://ftp.

Table 3.12 Storing a URL

Offset	Content	Explanation
0	0 × D1	SR = 1, TNF = 0 × 01 (NFC Forum Well-Known Type), ME = 1, MB = 1
1	0 × 01	Length of the record type (1 byte)
2	0 × 0B	Length of the payload (11 bytes)
3	0 × 55	The URI record type (“U”)
4	0 × 01	URI identifier code as “http://www.”
5	0 × 6e 0 × 66 0 × 63 0 × 6c 0 × 61 0 × 62 0 × 2e 0 × 63 0 × 6f 0 × 6d	The string “nfclab.com” in UTF-8

Table 3.13 Storing an e-mail address

Offset	Content	Explanation
0	0 × D1	SR = 1, TNF = 0 × 01 (NFC Forum Well-Known Type), ME = 1, MB = 1
1	0 × 01	Length of the record type (1 byte)
2	0 × 10	Length of the payload (16 bytes)
3	0 × 55	The URI record type (“U”)
4	0 × 06	URI identifier code as “mailto:”
5	0 × 69 0 × 6e 0 × 66 0 × 6f 0 × 40 0 × 6e 0 × 66 0 × 63 0 × 6c 0 × 61 0 × 62 0 × 2e 0 × 63 0 × 6f 0 × 6d	The string “info@nfclab.com” in UTF-8

Table 3.14 URI example on a smart poster

Offset	Content	Length	Explanation
0	0 × D1	1	NDEF header, TNF = 0 × 01(Well-Known Type), SR = 1, MB = 1, ME = 1
1	0 × 02	1	Record Name Length (2 bytes)
2	0 × 12	1	Length of the smart poster data (18 bytes)
3	"Sp"	2	The record name
5	0 × D1	1	NDEF header, TNF = 0 × 01(Well-Known Type), SR = 1, MB = 1, ME = 1
6	0 × 01	1	Record Name Length (1 bytes)
7	0 × 0A	1	Length of the URI payload (11 bytes)
8	"U"	1	Record Type: "U"
9	0 × 01	1	Abbreviation: "http://www."
10	"nfclab.com"	10	The URI itself

(ii) *Smart Poster RTD*

The Smart Poster RTD defines an NFC Forum Well-Known Type on how to put URLs, SMSs, and phone numbers on an NFC Forum mandated tag, and how to transport them between devices [17]. Smart posters are the popular use cases of NFC enabled applications. The idea is that an object can be made "smart", so that it is capable of storing additional information in the form of an NFC Forum mandated tag. By touching an NFC device to the tag, this information can be read and processed afterwards. The smart poster contains data that will trigger an application in the device such as launching a browser to view a website, sending an SMS to a premium service to receive a ring tone, and so on.

The smart poster concept is mostly built around URIs, which became the standard for referencing information around the Internet. URIs are very powerful, and as already mentioned they can represent anything from unique identifiers and web addresses to SMS messages, phone calls, and so on.

The content of a smart poster payload is an NDEF message. The content of this message consists of several NDEF records. The most important ones are as follows:

- *Title* record for the service.
- *URI* record which is the core of the smart poster.
- *Action* record which describes how the service should be treated.
- *Icon* record which refers to one or many MIME-typed image records (icons) within the smart poster.
- *Size* record for telling the size of the URI references which has an external entity (e.g., by URL).
- *Type* record for declaring the MIME type of the URI references which has an external entity (e.g., via a URL).

As shown in Table 3.14, the content of this message represents a web address of the NFC Lab-Istanbul, so that when the user touches the tag on this smart poster, it triggers the browser on the NFC device and displays the NFC Lab Istanbul's website.

(iii) *Signature RTD*

The signature record contains a digital signature related to one or more records within an NDEF message [18]. The signature can be used to verify the integrity and authenticity of

the overall NDEF message by signing them. Digital signing of NDEF data is a trustworthy method for providing information about the origin of the NDEF data in an NFC Forum mandated tag and NFC device. It enables the authenticity and integrity of data within the NDEF message to be verified. The goal is not to define or mandate a specific PKI or certification system, or to define a new algorithm for use with the signature RTD.

The NFC Forum Well-Known Type for the signature record is "Sig" which is 0×53, 0×69, 0×67 in UTF-8 encoding. The contents of the payload of a signature record are as follows:

- *Version* which refers to the version of the specification.
- *Signature* which includes the actual signature or reference to the location of the signature.
- *Certificate* chain which includes optional and mandatory fields.

The signature record can be used within an NDEF message as well as other types of data. The details about the signature record can be found on the NFC Forum's website.

(iv) *Text RTD*

The text record contains free form plain text [19]. The text record may appear as a sole record in an NDEF message, but in this case the behavior is undefined and the application should handle this. Typically, the text record should be used in conjunction with other record types to provide explanatory text. The NFC Forum Well-Known Type for the text record is "T" which is 0×54 in UTF-8 encoding. In text record types, the text can be encoded in either UTF-8 or UTF-16, which is defined by the status byte in the text record. The text record is composed typically of a NDEF record header, a payload and the actual body text in UTF format. In the payload, the status byte defines the encoding structure.

3.6 Peer-to-Peer Operating Mode Essentials

In peer-to-peer mode, two NFC enabled mobile phones establish a bidirectional connection to exchange information as depicted in Figure 3.29. They can exchange virtual business cards, digital photos, and any other kind of data. Peer-to-peer operating mode's RF communication interface is standardized by ISO/IEC 18092 as NFCIP-1.

3.6.1 Protocol Stack Architecture of Peer-to-Peer Mode

According to the NFC Forum Specifications, an NFC device operating in peer-to-peer mode has the following protocol stack elements (see Figure 3.30):

- Analog and digital protocols are lower layer protocols standardized by NFCIP-1.
- LLCP allows the transfer of upper layer information units between two NFC devices (see next section).
- Protocol bindings provide standard bindings to NFC Forum protocols and allow interoperable use of registered protocols.
- NFC Forum protocols are the ones that NFC Forum defines as binding to LLCP, such as OBEX and IP.

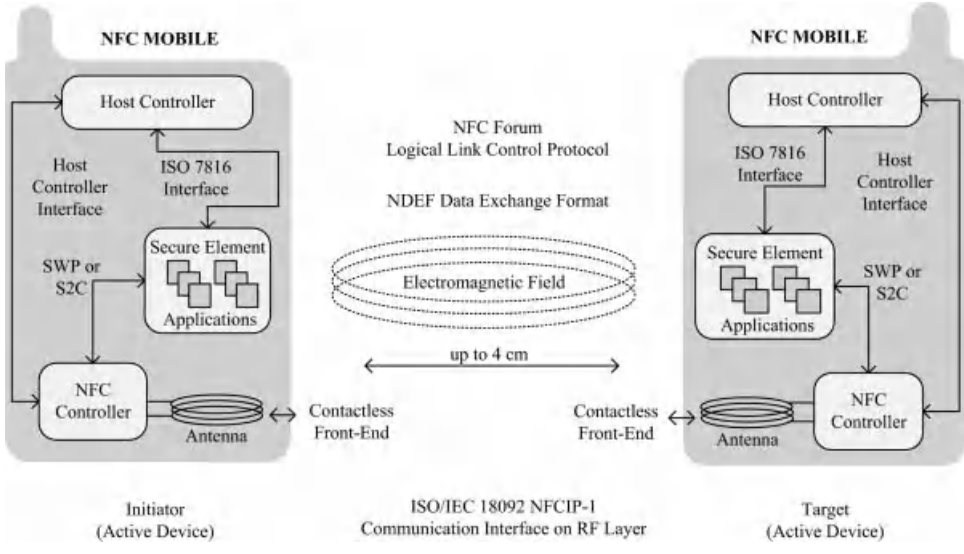


Figure 3.29 Communication architecture of peer-to-peer operating mode.

- Simple NDEF exchange protocol allows exchange of NDEF messages. It is also possible to run other protocols over the data link layer provided by LLCP.
- Applications may run over the simple NDEF exchange protocol, other protocols, or NFC Forum protocols. Example applications are printing from a camera, business card exchange, and so on.

3.6.2 LLCP

The LLCP defines an OSI data link protocol to support peer-to-peer communication between two NFC enabled devices. LLCP is essential for any NFC application that involves a bi-directional communication (see Figure 3.31) [20].

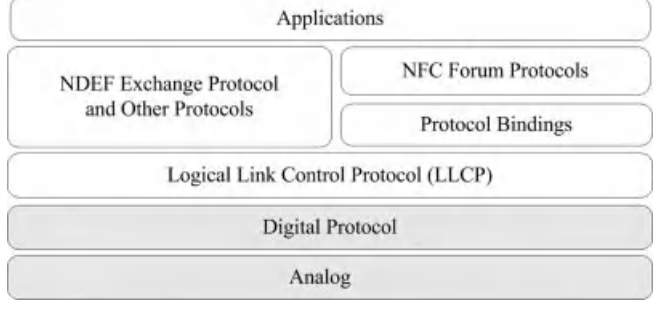


Figure 3.30 Protocol stack of peer-to-peer operating mode.

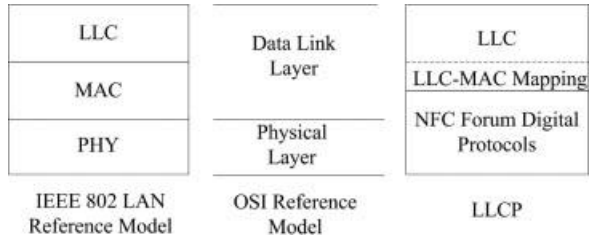


Figure 3.31 Relationship between LLCP and OSI Reference Model. Reproduced by permission from NFC Forum.

LLCP provides a solid ground for peer-to-peer applications. It enhances the basic functionalities provided by NFCIP-1 protocol as well. NFCIP-1 protocol provides a SAR (Segmentation and Reassembly) Level 1 Capability, as well as data flow control depending on the “Go and Wait" principle usual for half duplex protocols. Furthermore NFCIP-1 protocol allows error handling by using acknowledgement (ACK) frame and negative acknowledgement (NACK) frame, and provides an ordered data flow. It provides a link layer which is reliable and error free [5].

LLCP provides five important services: connectionless transport; connection oriented transport; link activation, supervision and deactivation; asynchronous balanced communication; and protocol multiplexing. These are described below [20]:

- *Connectionless transport:* Connectionless transport provides an unacknowledged data transmission service. This transport mode can be used if upper protocol layers implement their own flow control mechanisms. Therefore these layers do not need to rely on the data link layer’s flow control mechanism.
- *Connection oriented transport:* This mode provides a data transmission service with sequenced and guaranteed delivery of service data units. Data transmission is controlled through sliding window protocol.
- *Link activation, supervision and deactivation:* LLCP specifies how two NFC Forum devices within communication range recognize compatible LLCP implementations, establish an LLCP Link, supervise the connection to the remote peer device, and deactivate the link when requested.
- *Asynchronous balanced communication:* Typical NFC MACs operate in Normal Response Mode where only a master, called the initiator, is allowed to send data to the target and request data from the tag as well. The LLCP enables Asynchronous Balanced Mode (ABM) between service endpoints in the two peer devices by use of a symmetry mechanism. Using ABM, service endpoints may initialize, supervise, recover from errors, and send information at any time.
- *Protocol multiplexing:* The LLCP is able to accommodate several instances of higher level protocols at the same time.

Details about the LLCP protocol, architecture, and procedures can be found in the NFC Forum’s LLCP specification on the NFC Forum website.

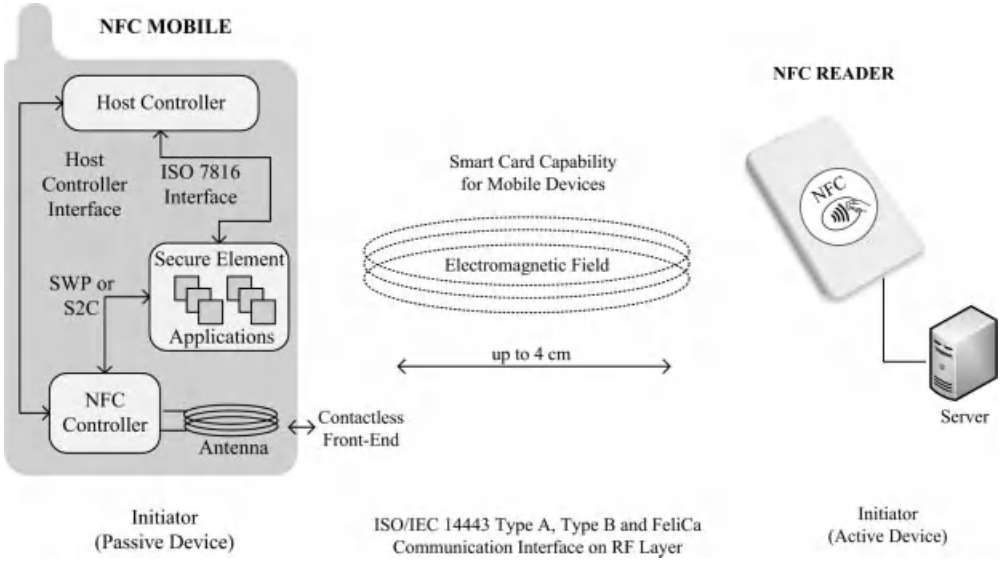


Figure 3.32 Communication architecture of card emulation operating mode.

3.7 Card Emulation Operating Mode Essentials

In card emulation mode, the NFC enabled mobile phone acts as a smart card. Either an NFC enabled mobile phone emulates an ISO 14443 smart card or a smart card chip integrated in a mobile phone is connected to the antenna of the NFC module. As the user touches her mobile phone to an NFC reader, the NFC reader initiates the communication. The communication architecture of this mode is illustrated in Figure 3.32.

3.7.1 Protocol Stack Architecture of Card Emulation Mode

NFC devices that are operating in card emulation mode use similar digital protocol and analog techniques as smart cards and they are completely compatible with the smart card standards (see Figure 3.33). Card emulation mode includes proprietary contactless card applications such as payment, ticketing and access control. These applications are based on ISO/IEC 14443 Type A, Type B and FeliCa communication interfaces.

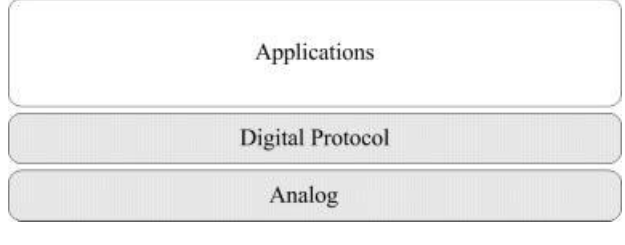


Figure 3.33 Protocol stack of card emulation operating mode.

As depicted in Figure 3.32, when an NFC reader interacts with an NFC mobile, the NFC mobile behaves like a standard smart card, thus the NFC reader interacts with the payment applications on the SE. Only card emulation mode uses SE efficiently and securely to perform functions that require high security.

There are various studies and specifications to manage the SE content remotely via Over the Air (OTA) technology which is covered in Chapter 8.

3.8 Chapter Summary

NFC occurs between two smart NFC devices which can play either an initiator or a target role. These NFC devices are the NFC enabled mobile phone, NFC tag and NFC reader. The NFC enabled mobile phone is the major device of NFC transactions that exists in each interaction type.

For the development of NFC technology in mobile phones, various standardization bodies provide open specifications and standards to increase the ease of access, interoperability, and security of NFC technology as well as mobile phone technology. The leading bodies are NFC Forum, GSMA, GlobalPlatform, OMA, and EMVCo. NFC enabled mobile phones have a complex architecture including the NFC interface which is the major component of the NFC devices, SEs with diverse alternatives, host controller, and host controller interface.

NFC operates at 13.56 MHz and transfers data up to 424 kbps. Communication between two NFC devices is standardized in ISO/IEC 18092 standard as NFCIP-1 which includes only device-to-device, peer-to-peer communication and active/passive communication reader/writer modes. However, an RF layer of NFC communication is also compatible with ISO/IEC 14443 standard (i.e., contactless proximity smart card standard), Japanese JIS X 6319 standard as FeliCa (i.e., another contactless proximity smart card standard by Sony) and ISO/IEC 15693 standard (i.e., contactless vicinity smart card standard). These smart card interfaces operate at 13.56 MHz with distinct data rates, communication ranges, as well as different modulation and coding features.

ISO/IEC 14443 is the main standard that provides the proximity contactless communication interface for NFC enabled transactions which occurs in passive communication mode. ISO/IEC 14443 has two major forms as Type A (e.g., MIFARE smart cards) and TYPE B (e.g., Calypso smart cards). FeliCa is another important 13.56 MHz high speed proximity contactless smart card interface which is also compatible with the RF layer of NFC.

NFC can occur in three interaction styles: between an NFC enabled mobile phone and an NFC tag, between an NFC enabled mobile phone and an NFC reader, and between two NFC enabled mobile phones. Related to these interactions, NFC technology offers three operating modes: reader/writer operating mode, peer-to-peer operating mode, and card emulation operating mode. Each operating mode's communication essentials are discussed in this chapter. NFC Forum standardized all layers from the application level to RF layer of the reader/writer mode and peer-to-peer mode.

In reader/writer operating mode, an active NFC enabled mobile phone initiates the wireless communication, and reads and/or alters data stored in the NFC tag afterwards. In this mode, an NFC enabled mobile phone is capable of reading NFC Forum mandated tag types. It is compatible with the ISO/IEC 14443 Type A, Type B and FeliCa communication interface on the RF layer. NFC Forum standardized the NFC Forum mandated tag types as well as NFC Data

Exchange Format (NDEF) and various record types from the application layer to the RF layer. In the case of peer-to-peer mode, communication occurs between two active NFC mobiles. One of the mobiles initiates the communication and a link-level communication is established between them thereafter. It uses ISO/IEC18092 NFCIP-1 standard for the communication on the RF layer. NFC Forum standardized the LLCP from the application layer to the physical layer. Card emulation mode enables security and privacy required transactions with NFC. It gives smart card capability to mobile phones and uses ISO/IEC 14443 Type A, Type B and FeliCa. The SE concept is an important issue in this mode to store and process valuable data and applications.

Chapter Questions

1. Why can two passive devices not establish an NFC communication?
2. What are the standardization bodies related to NFC technology?
3. What is the eventual aim of NFC Forum?
4. What is the eventual aim of GlobalPlatform?
5. What is an NFC interface? Explain its components.
6. List and explain secure element (SE) alternatives.
7. What are the differences between an NFC wired interface (NFC-WI) and single wire protocol (SWP)? Explain.
8. What is the difference between ISO/IEC 14443 and ISO/IEC 15693 in terms of operating mode and range?
9. What are the major implementations of proximity contactless smart cards?
10. What is the usage of NFCIP-1?
11. What is the usage of NFCIP-2?
12. What are the differences between Type 1, Type 2, Type 3, and Type 4 tags?
13. Explain the basic structure of NDEF.
14. What is the importance of RTD, and what are the possible use cases of it?
15. What is the well-known NDEF record type? Give some examples.
16. What is Logical Link Layer Protocol (LLCP)? Explain its relationship with the OSI Reference Model.
17. Explain the connectionless and connection oriented transport services of LLCP.
18. Explain the importance of secure element in card emulation operating mode.

References

[1] NFC Forum, http://www.nfc-forum.org/ (accessed 10 July 2011).

[2] Finkenzeller, K. (2010) *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, John Wiley &Sons, Lts, ISBN: 978-0-470-69506-7.

[3] ECMA International (2006) *ECMA 373: Near Field Communication Wired Interface (NFC-WI)*, June 2006. Available at: http://www.ecma-international.org/memento/TC47-M.htm (accessed 10 July 2011).

[4] ETSI TS (2008) ETSI TS 102 613, Smart Cards; UICC - Contactless Front-end (CLF) Interface; Part 1: Physical and data link layer characteristics. Technical Specification, September 2008.

[5] Tuikka, T. and Isomursu, M. (eds) (2009) *Touch the Future with a Smart Touch*, VTT Tiedotteita – Research Notes 2492, Espoo, Finland, 2009. Available at: www.vtt.fi/inf/pdf/tiedotteet/2009/T2492.pdf (accessed 10 July 2011).

[6] ETSI TS (2008) *ETSI TS 102 622, Smart Cards; UICC - Contactless Front-end (CLF) Interface; Host Controller Interface (HCI)*, Technical Specification, February 2008.
[7] MIFARE, http://www.MIFARE.net/products/MIFARE-smart card-ic-s/ (accessed 10 July 2011).
[8] FELICA, http://www.sony.net/Products/felica/ (accessed 10 July 2011).
[9] Calypso, http://www.calypsonet-asso.org/index.php (accessed 10 July 2011).
[10] ECMA International (2004) *ECMA 340: Near Field Communication Interface and Protocol (NFCIP-1)*, December 2004. Available at: http://www.ecma-international.org/memento/TC47-M.htm (accessed 10 July 2011).
[11] ECMA International (2010) *ECMA 352: Near Field Communication Interface and Protocol (NFCIP-2)*, June 2010. Available at: http://www.ecma-international.org/memento/TC47-M.htm (accessed 10 July 2011).
[12] Haselsteiner, E. and Breitfuß, K. *Security in Near Field Communication (NFC)*, Philips Semiconductors. Available at: http://events.iaik.tugraz.at/RFIDSec06/Program/papers/002%20-%20Security%20in%20NFC.pdf (accessed 10 July 2011).
[13] NFC Forum, *NFC Activity Specification*, Technical Specification, Version 1.0, November 2010.
[14] NFC Forum, *NFC Data Exchange Format (NDEF)*, Technical Specification, Version 1.0, July 2006.
[15] NFC Forum, *Record Type Definition (RTD)*, Technical Specification, Version 1.0, July 2006.
[16] NFC Forum, *URI Record Type Definition*, Technical Specification, Version 1.0, July 2006.
[17] NFC Forum, *Smart Poster Record Type Definition*, Technical Specification, Version 1.1, July 2006.
[18] NFC Forum, *Signature Record Type Definition*, Technical Specification, Version 1.0, November 2010.
[19] NFC Forum, *Text Record Type Definition*, Technical Specification, Version 1.0, July 2006.
[20] NFC Forum, *Logical Link Control Protocol*, Technical Specification, Version 1.0, December 2009.

4

NFC Operating Modes

There are three major devices in NFC; namely NFC enabled mobile phones, NFC readers, and NFC tags. Various combinations of these devices are available for mobile interaction. For example, a mobile phone may communicate with an NFC reader, another mobile phone, or an NFC tag. Please note that the communication is a pairwise activity here, that is, only two NFC devices can interact with each other at a time.

Also remember that the three existing operating modes are the reader/writer, peer-to-peer and card emulation modes. The reader/writer mode enables NFC enabled mobile devices to exchange data with NFC Forum mandated NFC tags. The peer-to-peer mode enables two NFC enabled mobile devices to exchange data with each other. In the card emulation mode, the user interacts with an NFC reader in order to use her mobile phone as a smart card such as a credit card. Each operating mode has different use case scenarios and each provides various underlying benefits to users.

In parallel to the design considerations and implementations of the technology, communication between devices must occur within a very close range. In order to ensure a successful communication, it is very common to touch the communicating devices to each other. For this reason, this process is called touching paradigm.

This chapter initially describes the details of mobile interaction techniques including NFC technology interaction. It further emphasizes details of NFC operating modes, use case scenarios, and underlying benefits of each NFC operating mode. Finally, case studies analyses are given in order to describe the usage of NFC services in detail.

4.1 Mobile Interaction Techniques

Mobile devices are used to enable mobile services such as phone calls, Short Messaging Service (SMS) message exchange, Internet usage, web services, and so on. In a simple mobile service usage, three important components are involved:

- A *user* who uses the mobile device.
- A *mobile device* which is the tool used by the user.
- A *mobile service* which is launched via the mobile device.

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.
 Published 2012 by John Wiley & Sons, Ltd.

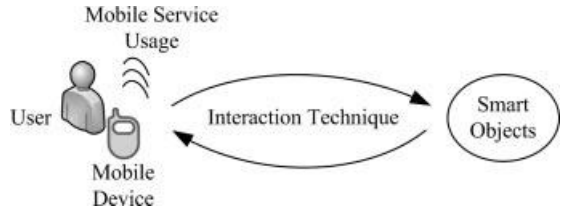


Figure 4.1 Mobile interaction.

These are the minimum requirements for mobile service execution. However, when mobile devices are used to interact with smart objects in the environment, additional components are included. The first one is the smart object that the mobile device interacts with. Additionally, when a user interacts with a smart object, an *interaction technique* is used, which is the communication paradigm between the smart object and the user via the mobile device. The available interaction techniques that the mobile devices use, which are called *mobile interaction techniques*, are touching, pointing, and scanning. Hence, a mobile interaction consists of five elements which are the user, mobile device, mobile service, smart object, and the interaction technique (Figure 4.1).

Smart objects come in a variety of forms and generally consist of sensors. NFC tags, RFID tags and barcodes are a few examples of smart objects. A user uses one of the mobile interaction techniques when interacting with a smart object via a mobile device. We will define available mobile interaction techniques here and then give details of the NFC interaction:

(i) *Touching*

Touching, as its name suggests, occurs by touching the mobile device to the smart object. Touching can also refer to communicating with a smart object in very close range. The typical range varies between 0 and 10 cm [1].

In order for a user to interact with a smart object by the touching interaction technique, the user, the mobile device, and the smart object need to be very close to each other. Touching is a very natural, intuitive and the most usable interaction technique, since all a user needs to do is physically touch the mobile device to a smart object.

Well-known examples of touching are RFID technology applications that communicate in very close range. Since NFC is based on RFID technology and the user needs to touch her mobile device to the smart object, NFC also uses a touch interaction technique.

(ii) *Pointing*

The pointing interaction technique is analogous with someone pointing an object. In the case of mobile interaction, we point our mobile device to the smart object from a distance. Examples to this kind of interaction can be seen in visual markers and pointing based cameras. Reading QR (Quick Response) codes [2] is one example of this interaction technique.

A QR Code (which is also described in Chapter 2) is similar to the barcode and is a specific matrix code that can be readable with QR barcode readers or camera equipped mobile phones. The difference between a QR Code and a barcode is that since QR Code is a two-dimensional matrix barcode it can hold much more information than a barcode. Any type of information such as text, URL, or other data can be saved in a QR Code.

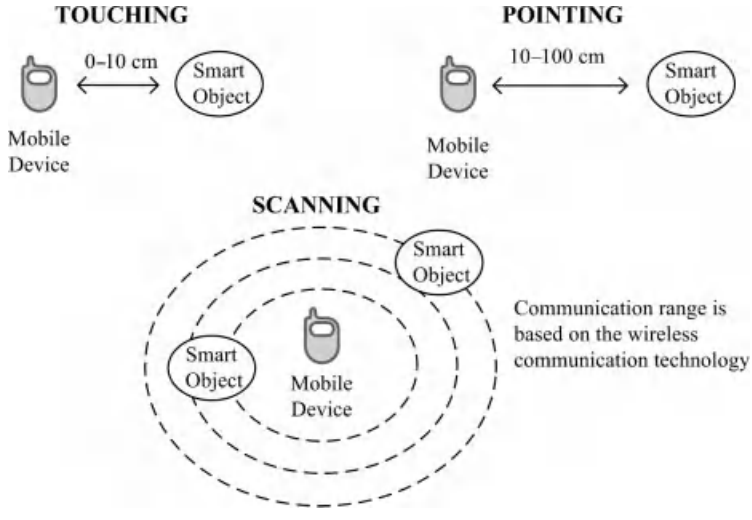


Figure 4.2 Mobile interaction techniques.

This interaction typically occurs between 10 cm and 100 cm. Interactions that occur out of this range are no longer referred to as pointing [1].

(iii) *Scanning*

The scanning interaction technique typically involves discovery of smart objects within the environment using wireless communication technologies. The best practical example is Bluetooth technology. One advantage of this technique is that the user does not need to be aware of the exact location of the smart object; the device simply searches for the smart objects in the range. When any object is discovered, the user can interact with the discovered smart object using her mobile device [1].

The three interaction techniques are depicted in Figure 4.2. The differences between the interaction techniques are given in Table 4.1.

4.1.1 NFC Technology Interaction Technique

As defined earlier, NFC technology requires touching two NFC compatible devices together over a few centimeters. User awareness is essential in order to perform NFC communication.

Table 4.1 Mobile interaction techniques

Mobile Interaction Technique	Communication Range	Smart Object Awareness
Touching	From 0 to 10 cm	Yes
Pointing	From 10 to 100 cm	Yes
Scanning	Based on wireless communication technology	No

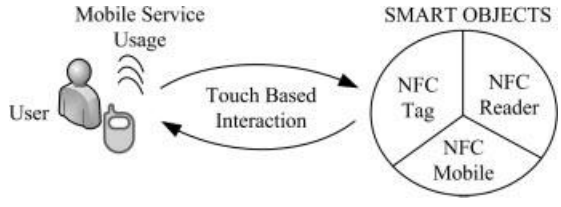


Figure 4.3 NFC technology interaction.

The NFC technology interaction technique is touch based as depicted in Figure 4.3. The user first interacts with a smart object (either an NFC tag, NFC reader or another NFC enabled mobile phone) using her mobile phone. After the touching occurs, the mobile device may make use of received data and thus may additionally use mobile services as well, such as opening a web page, making a web service connection, and so on.

4.2 Classification of NFC Devices

NFC works in a very intuitive way. Two NFC devices immediately start their communication as they are touched. The touching action is taken as the triggering condition for NFC communication. This simple model is actually one important property of NFC technology. An application on a computing device can be started either by a user or another application initiative. In the mobile device case, one typical scenario may be that the user opens the menu, selects a shortcut item, and presses OK, and so on. In the NFC case, it is so trivial. The NFC application is designed so that when the mobile touches some NFC device with the expected form of data, it boots up immediately. Hence, the user does not need to interact with the mobile device anymore but just touches one appropriate NFC device which may be an NFC tag, an NFC reader, or another mobile phone. More sophisticated scenarios are also possible. Let us consider that two or more NFC applications are available on the same mobile phone. As the user touches another NFC device, the application that is designed to interact with the touched object boots up but not the other applications. Considering ubiquitous computing requirements, this is a very useful property of NFC communication.

We can classify the NFC devices in the communication based on two parameters. The first parameter is the energy supply which results in active and passive devices. The second one is initiating the communication and leads to initiator and target devices.

4.2.1 Active vs. Passive Devices

Active and passive device definitions are important to understand the NFC communication. An active device is one that is powered by some power source – such as a battery – so that it generates its own electromagnetic field. On the other hand, a passive device is one that does not have any integrated power source. It is a rule of nature that each activity requires energy; hence even a passive device requires some power to perform as programmed previously. In NFC communication the energy is supplied by the other (active) party for the passive device. To summarize, an active device powers the passive device by creating the electromagnetic field.

Table 4.2 Combinations of NFC devices

Devices	Initiator	Target
Active	√	√
Passive	X	√

4.2.2 Initiator vs. Target Devices

NFC always occurs between two parties, so that one party is called the initiator, and the other party is called the target. The initiator is the one that initiates the communication; the target responds to the request that is made by the initiator. This case is analogous to the well known client–server architecture. Remember that in a client–server communication the client initiates the communication and the server responds. In NFC communication, it is no different.

An initiator obviously always needs to be an active device, because it requires a power source to initiate the communication. The target, on the other hand, may be either an active or a passive device. If the target is an active device, then it uses its own power source to respond; if it is a passive device, it uses the energy created by the electromagnetic field which is generated by the initiator that is an active device.

An RFID tag or an NFC tag is a low-cost and low-capacity device. Hence, it does not contain any power source and needs an external power source to perform any activity. So, an NFC tag is always a passive device, since it does not include any energy source by design. It contains data that can be read by an active NFC device. It also can be only a target and not an initiator for the same reason.

To summarize, NFC occurs between one NFC initiator and one NFC target. An active device can take either role, whereas a passive device can only be a target (see Table 4.2). The initiator sends requests to the target, and the target responds these requests as in client–server architecture.

4.3 Reader/Writer Mode

The reader/writer mode is about the communication of an NFC enabled mobile phone with an NFC tag for the purpose of either reading or writing data from or to those tags. It internally defines two different modes: reader mode and writer mode.

In reader mode, the initiator reads data from an NFC tag which already consists of the requested data. The NFC tag we mention here is one of the NFC Forum mandated tag types as described in Chapter 3. In addition to the requirement that the NFC tag already consists of the requested data, it also consists of the program which performs returning the requested data to the initiator.

In writer mode, the mobile phone acts as the initiator and writes data to the tag. If the tag already consists of any data prior to the writing process, it will be overwritten. The algorithm may even be designed so that the initiator will update by modifying the previously existing data as well. Although it is not a common option, an NFC reader, in addition to the mobile phone, may also be used to read data from a tag as well. We may even envisage an NFC reader

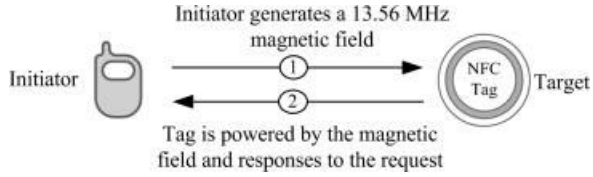


Figure 4.4 Reader/writer mode.

that writes data to an NFC tag. The maximum possible data rate in this mode is 106 kbps. A schematic representation of the reader/writer mode is given in Figure 4.4.

A mobile phone can perform different actions after it reads the data from the tag. If the tag stores push registry data for example, then an application that is previously installed on the mobile phone can be started automatically. Remember that a push registry record is used to start an application on a mobile device automatically. If the tag stores a URL, a web browser can be launched and the received web page can be displayed.

The features of mobile phones such as processing power, audio/video capability and Internet access together with reader/writer mode provide many opportunities for both users and the service providers. A user may purchase a movie ticket after opening a mobile application or a website by touching her mobile device to a smart poster tag. Applications in this mode are countless and can be very innovative.

4.3.1 Smart Poster

The *smart poster* term refers to advertising materials or posters that are equipped with NFC tags. Those tags may contain different types of data such as a URL address, a couponing service, an SMS service, and so on (see Figure 4.5). Smart posters are the most common usage area of the reader/writer mode and some issues of smart posters need to be highlighted:

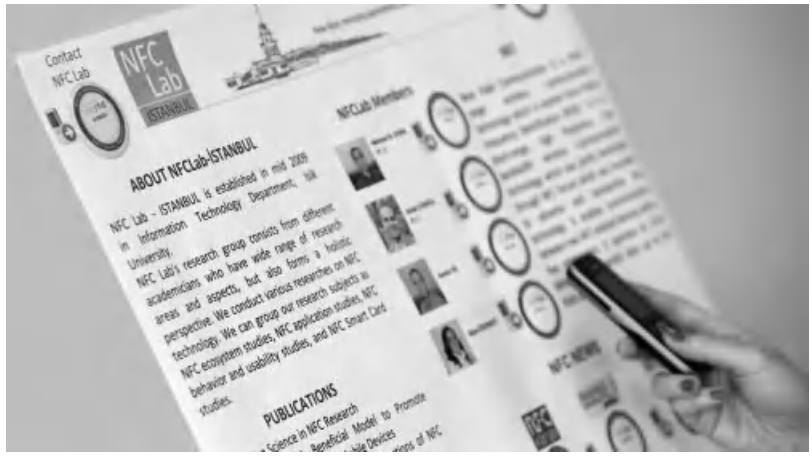


Figure 4.5 Smart poster use case.

(i) *Where to touch*

The most important issue in smart posters is the user's awareness of the touching point. The smart poster should be designed in such a way that the user should not waste time looking for the tag. Different means can be used to achieve this:

- Director sign: Some textual sign such as 'Touch Here' and/or a supplementary image that marks the touching point can be placed near the tag.
- Mobile phone image: A mobile phone image on or next to the tag may be helpful to mark the touching point.
- N-Mark: The N-Mark logo can be used to mark the NFC tag [3]. However end-users may not be aware of the meaning of an N-Mark sign. Hence, we believe that the N-Mark logo should be an additional component and may be used in addition to the above two options. Remember that the N-Mark logo is an important development that has been introduced by NFC Forum and is designed to be a universal symbol for NFC.

(ii) *Awareness of tag content*

Since the digital content of the tag cannot be visually read, it is a question of how to enable the user to guess the content of the tag. The solution is actually simple enough: insert advertisement material or descriptive information close to the tag. For example, if the smart poster advertises a discount coupon and there is only one tag on the smart poster, it is easy to understand that the tag contains the coupon. However, if there are multiple tags on the same smart poster, then descriptive text or image should be placed next to each tag. The descriptive text should be short (one or a few words) and should be easily understood. Example texts include "Info", "Make Reservation" and "Deal of the Day".

(iii) *Trustworthiness*

Another important issue with a smart poster is its trustworthiness. The term "trustworthiness" in the smart poster context refers to "the assurance that the smart poster behaves as it is expected". When a user sees a smart poster, she needs to trust the originality, integrity, and reliability of the smart poster provider. Then the user will be sure that the poster will not harm the mobile device in any way, such as causing it to install some unrequested application. In order to achieve trustworthiness, the easiest way is to use original logo material of the provider. Different options depending on the provided service may also be used to gain trust.

4.3.2 Generic Usage Model

Service usage in each NFC operating mode differs because the interacted smart objects are different and provide distinct usage scenarios. In reader/writer mode, a user interacts with an NFC tag and uses her mobile phone as a reader/writer device. In this case, a service provider owns the NFC tag. However, it is only indirectly included in the process. Therefore NFC interaction occurs offline with respect to the service provider, whereas service usage occurs online as the mobile phone interacts with the server after using the information stored in the NFC tag. Remember that information on the tag is possibly the information that will be used to access some server provided by the service provider such as a URL of some page, reservation information, or ticketing web service data. In another scenario, an NFC tag stores a URL so that the related website can be launched by the browser on the mobile phone after

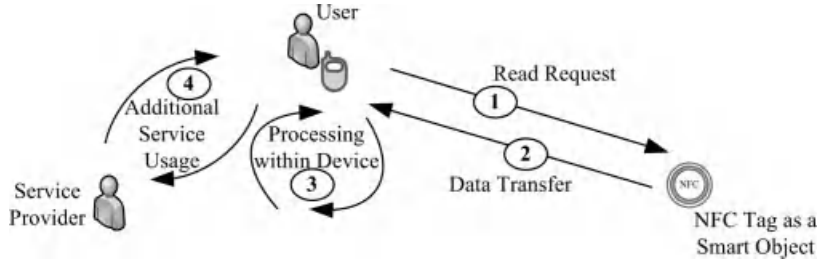


Figure 4.6 Generic usage model of reader mode.

the URL is received. In yet another example, the tag may contain web service information so that the application on the mobile phone may fire the web service provided by the service provider.

Each operating mode has its own characteristics, so that it is possible to define a different usage model for each operating mode. Generic usage models define those characteristics that each operating mode performs mandatorily along with the usage principle of the technology. These models are explained in this chapter to explain the usage of each operating mode in detail. Figures 4.6 and 4.7 show the generic usage model of the reader and writer modes, respectively:

1. *Read request:* User requests data by touching her mobile phone to an NFC tag which can be embedded in various components such as a smart poster, product box, and so on.
2. *Data transfer:* The data that reside in the tag are transferred to the mobile phone.
3. *Processing by the mobile device:* When data are transferred to the mobile phone, they can be used for several purposes such as pushing an application, displaying data to the user, or processing data in an application for additional purposes.
4. *Additional service usage:* This step is optional and takes advantage of the mobile phone's advanced capabilities, and mostly involves the Internet connectivity. When data are processed in the mobile phone, they can be used for further operations via the Internet such as connecting to a service provider by using a web service.
5. *Write request:* The user requests to write data to an NFC tag by touching her mobile phone to it.
6. *Data transfer:* The NFC tag replies with the acknowledgment data to inform the user about the success of the operation.

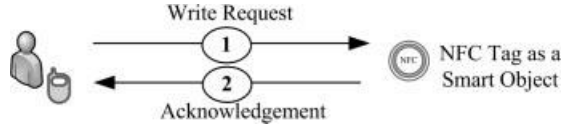


Figure 4.7 Generic usage model of writer mode.

4.3.3 Leading Applications

There are characteristic differences among NFC operating modes so that each mode provides different use cases. For example there are numerous use case opportunities for the reader/writer mode. Transferred data can be a text, a URL, a product identification, or some other type of data. After the transfer operation, data can be used for many purposes by the mobile phone according to the design.

A vast number of scenarios are possible in this mode. Companies or professionals might be willing to design projects in this mode because of the flexibility of different data type to be stored on the NFC tag, as well as the flexibility on how to use that information. Hence, a wide range of applications in health, education, and entertainment may be potentially generated using the reader/writer mode.

4.3.3.1 Information Gathering

Definitely the most important use case for the reader/writer mode is information gathering. Most of the information gathering applications include smart poster applications. In its primitive form, information gathering includes reading information from an NFC tag and then either storing, or displaying it on the mobile phone. There are two options to store the data:

- Actual data might be stored on the tag;
- The tag includes the URL of a webpage that contains the actual data.

The advantage of storing the data on the tag is that reception by the mobile is easy and fast for small amounts; the advantage of using a server is that it can contain larger amounts of data. Storing the data on the server has another benefit: the data can be easily updated with respect to modifying all NFC tags that contain the outdated data. When the data are to be stored on a remote server, using web service provides a convenient and contemporary solution as well. In order to use a web service, a suitable application that can make use of the web service is needed to be installed on the mobile device.

If the amount of data is small and static, it is wiser to embed the data into the tag, since there will be no need to update the data in the tags. However, if the service provider is willing to provide additional dynamic services that may be changed or updated over time, getting the user to connect to the service provider's server is a better option.

(i) *Schedule downloads*
This use case enables users to download schedules (e.g., bus schedules) to a mobile phone via smart posters. As stated above, schedules can be saved within the tag when the amount of data is small and static so that it does not change frequently. If they change frequently, the information should be saved to a website that is accessible on mobile browsers or in a database server that is accessible via a web service client. Schedules can be easily managed if the information is stored in an Internet resource.

(ii) *Product information*
This use case enables users to collect product information from a tag that is embedded onto that product package. For example, technical specifications or content information

of the product can be transferred to mobile phones immediately after reading the tag. It can also be implemented with additional web usage options to provide additional specifications, user comments on the product, price ranges, and much more. Internet usage in this use case is essential, since additional data require extensive storage, which is higher than the capacity of the tag.

(iii) *Product history*

This use case enables the history of a product to be read via an NFC tag attached to it. As an example, when a customer buys beef from a supermarket, production information of the beef such as breed and age of the cow, the location of the farm, and in which abattoir it was produced can be read.

4.3.3.2 Remote Marketing

Remote marketing is another nice use case on reader/writer mode that enable online reservations and purchases.

(i) *Online reservation and purchase*

Reservation and purchase of performance tickets such as cinema and theater can be made online using smart posters. The NFC tag contains summary information about the performance, which may be static. The schedule information is possibly dynamic and stored on some server. After the user is connected to the server, she may either make the reservation or purchase the ticket as well.

(ii) *Online special offers*

Special offers such as mall, store or product discounts and campaign information can be received by touching a mobile device to an NFC tag on the smart poster.

(iii) *Remote shopping*

The remote shopping use case can be implemented in several ways to enable online shopping. Consider that all goods come with an NFC tag attached to their packages. When an item is consumed, the user can touch the mobile device to the tag on the item's package and add it to a shopping list. The user then will be able to order the shopping list online.

4.3.3.3 Mobile Social

(i) *Social networking*

Generally, all social networking cases in the reader/writer mode include updating presence information on social networks. In this case, NFC tags can provide a push registry record together with an additional update information record. Remember that a push registry record is used to start an application on a mobile device automatically. Presence update on social networking sites can become one of the important applications in reader/writer mode. For example, in social networking applications, users can touch NFC tags which are embedded at the entrance of an entertainment venue or educational institution to update status or location information while entering or exiting the place. Status update information can be something like:

I have entered ISIK University.

(ii) *Location based services*

Location based services can be implemented in different contexts. As a service for, say, tourists, users can receive information on the closest cash point or tourist information center as well as city information, a tourist guide, city map or other information from smart posters. NFC based navigations, social event recommendations, and location based mobile advertising are other examples in this context.

There are a tremendous number of use cases and applications in the reader/writer mode. We gave use cases above to show what the reader/writer mode applications can provide.

4.3.4 Use Cases on Reader/Writer Mode

In this section, short use cases on reader/writer mode are presented to give the idea behind the operating mode.

4.3.4.1 Information Gathering

Alice had had a long day at the university and was really tired. She was now sitting in the cafeteria. She wanted to know the next shuttle bus from the university to the city but she did not want to walk to the bus stop. She then saw a smart poster on the cafeteria wall showing a picture of the university shuttle buses. She realized that she could find out the shuttle bus schedule from the poster. She touched her NFC enabled mobile phone to the tag on the smart poster.

In this case, there are three different options of NFC service modeling:

1. The NFC tag stores the schedule information.
2. The NFC tag stores a website URL, so that the website stores the schedule information.
3. The NFC tag stores push registry data and additional service data for the mobile application to process (e.g., the mobile application provides different services and these data enable the application to provide schedule data), and the server stores the schedule information which can be accessed via web services.

(i) *Use Case 1*

1. *Read request:* Alice requests bus schedule data by touching her mobile phone to the tag (see Figure 4.8).
2. *Data transfer:* Bus schedule data that reside in the tag are transferred to Alice's mobile phone.

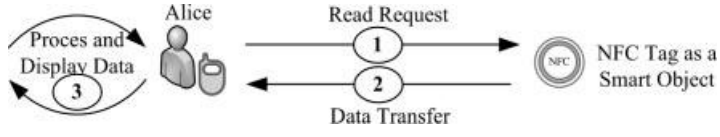


Figure 4.8 Example usage model of schedule information gathering from an NFC tag.

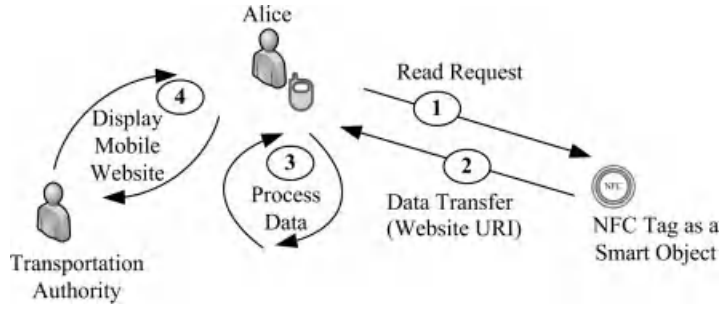


Figure 4.9 Example usage model of schedule information gathering from a mobile website.

3. *Process and display data:* Alice's mobile phone processes the data and displays schedule information.

Note: In order to display the data without any additional application, the data need to be formatted in a specific way that the mobile phone can interpret with its own operating system applications.

(ii) *Use Case 2*

1. *Read request:* Alice requests bus schedule data by touching her mobile phone to the tag (see Figure 4.9).
2. *Data transfer:* The website URL in the tag is transferred to Alice's mobile phone.
3. *Process data:* Alice's mobile phone processes the data and discovers that it is a URL.
4. *Display mobile website:* Alice's mobile phone launches the website with its browser and displays the schedule information.

(iii) *Use Case 3*

1. *Read request:* Alice requests bus schedule data by touching her mobile phone to the tag (see Figure 4.10).
2. *Data transfer:* Push registry and additional required data for the application to process (if any) in the tag are transferred to Alice's mobile phone. Push registry data will be used to launch the application in the mobile phone automatically. Also, the application may require additional data (based on its design) to display the required schedule to the user.

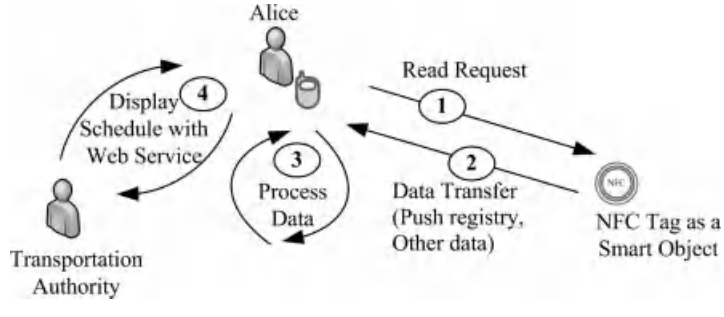


Figure 4.10 Example usage model of schedule information gathering via web service.

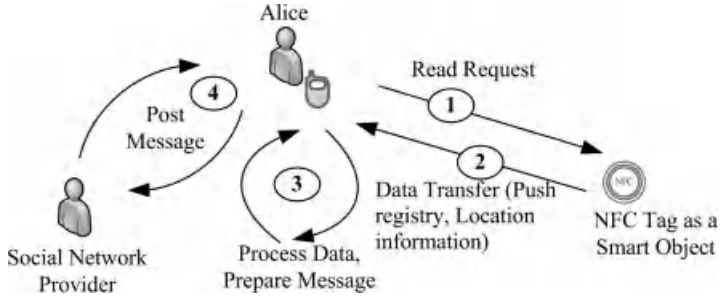


Figure 4.11 Example usage model of presence information updating on a social network.

3. *Process data:* Alice's mobile phone processes data, launches the application and transfers additional required data (if any) to the application.
4. *Display schedule with web service:* The application receives the schedule information from the service provider via web service and displays the schedule information.

Alice felt comfortable after learning the shuttle bus schedule. She avoided waiting at the bus stop and rested in the cafeteria until the shuttle's scheduled time.

4.3.4.2 Mobile Social

Alice went to a recently opened restaurant with her boyfriend Bob. After a couple of minutes, she saw an NFC sign near the table that provided the mechanism to update presence information on her social network. She wanted to let her friends know about the restaurant.

1. *Read request:* Alice touches her mobile phone to the tag to update the status information on her social network (see Figure 4.11).
2. *Data transfer:* Push registry data and location information in the tag are transferred to the mobile phone. Push registry data are used to launch the application automatically.
3. *Process data, prepare message:* The application processes the data, launches the application (if it is not running already) and requests confirmation from Alice to publish "I'm now in Sunday Restaurant". Alice also adds the comment "You must see this restaurant, glorious. . ."
4. *Post message:* Alice posts the message to her social network provider via the mobile application.

In minutes, her best friend, Julia, has commented on her post. Alice was very pleased to be able to share news of the restaurant with her friends.

4.3.5 Underlying Application Benefits

Each NFC operating mode provides different use case opportunities. Each use case provides various underlying benefits to users. In this section, details of these underlying benefits of reader/writer mode applications are given.

In general, the mobile phone and its applications provide mobility; and in turn this generally reduces the requirement for physical effort. For example, making a phone call to someone from a distance provides mobility and eliminates the need to communicate face to face. This results in an increase in mobility and a decrease in physical effort. Moreover, increasing processing power and wireless Internet access of mobile devices enable users to communicate with others on the go and overcome geographical restrictions. This mode is also subject to increase mobile phone usage.

We will now discuss some use cases to give the idea of how the reader/writer mode provides these benefits. In a smart poster case, when the user touches her phone to a tag on the smart poster, previously stored data are transferred from the tag to the mobile device. Assume that this information is the office number of a department employee. Using the transferred information, the user can easily find the staff member's office whilst eliminating the need to memorize office number which is still displayed on the mobile device's screen. These processes provide mobility to user. The user can find her way while the required information can still be read from the screen. It also eliminates any need to print this information.

Consider a remote shopping case in which people can order shopping items using the embedded tags on product packages. First, the user selects the required item by touching her mobile phone to the NFC tag on the package and then she submits the order. In particular, elderly users will be able to easily shop from home which results in decreased physical effort.

If we examine the developments already produced, we can easily envisage many more reader/writer mode applications. Developments and implementations of reader/writer mode applications are relatively easy to carry out when compared with other operating mode applications. There are also many interesting and easy to implement use case scenarios that can be developed in this mode.

4.4 Peer-to-Peer Mode

Peer-to-peer mode enables two NFC enabled mobile devices to exchange information such as a contact record, a text message, or any other kind of data. This mode has two standardized options: NFCIP-1 and LLCP which are already described in Chapter 3. NFCIP-1 takes advantage of the initiator–target paradigm in which the initiator and the target devices are defined prior to starting the communication. However, the devices are identical in LLCP communication. After the initial handshake, the decision is made by the application that is running in the application layer.

On account of the embedded power to mobile phones, both devices are in active mode during the communication in peer-to-peer mode. Data are sent over a bidirectional half duplex channel, meaning that when one device is transmitting, the other one has to listen and should start to transmit data after the first one finishes. The maximum possible data rate in this mode is 424 kbps. A schematic representation of the peer-to-peer mode is given in Figure 4.12.

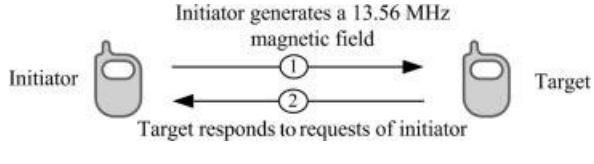


Figure 4.12 Peer-to-peer mode.

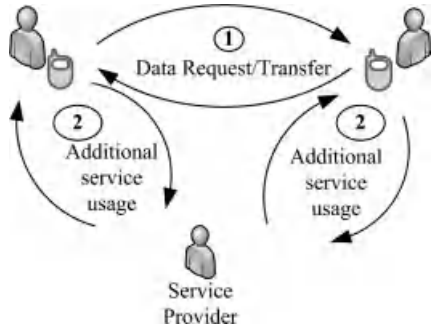


Figure 4.13 Generic usage model of peer-to-peer mode.

4.4.1 Generic Usage Model

In peer-to-peer mode, users communicate with each other using NFC enabled mobile phones. In this mode normally no service provider is used in the process, meaning that users do not communicate with it. If users intend to use any services on the Internet, a service provider may be included in the process as well. Figure 4.13 shows the generic usage model of peer-to-peer mode.

1. *Data exchange:* Two users exchange data via mobile phones.
2. *Additional service usage:* When data are shared between mobile phones, these data can optionally be used for additional purposes such as saving a received business card to a database over the Internet after a successful share or starting a friendship on a social network.

4.4.2 Leading Applications

In this section, we focus on major applications that benefit from peer-to-peer mode.

4.4.2.1 Exchanging Data

Exchanging data is an important use case of this mode (see Figure 4.14).

(i) *Exchanging private data*
Critical information can be stored in a mobile device securely and can further be exchanged with other authorized people using peer-to-peer mode. Since the communication is performed over a few centimeters, users will feel confident to share private and important data using NFC technology. When higher level of security requirements are set, additional security measures need to be provided as well (see Chapter 6).

(ii) *Money transfer*
Two users can exchange money between wallets that are stored on their NFC mobiles. Gifts, coupons, and tickets can be implemented as exchangeable objects as well.

Figure 4.14 Two peers exchanging contact information.

4.4.2.2 Social Networking

Social networking in peer-to-peer mode includes use cases for people to exchange their social information. Starting a friendship on social networks is a good example of this case. Business card exchange is yet another example.

4.4.2.3 Pairing Devices

Two NFC enabled devices can be paired such as headsets with handsets, in-car devices, computer components, and so on.

4.4.3 Use Cases on Peer-to-Peer Mode

(i) *Data exchange*

Alice was having her lunch in her university's cafeteria. Another student, Jennifer, on her course asked if she could sit next to her. They had barely spoken previously. They had their lunch together and chatted in the meantime. Alice asked if she can get Jennifer's contact information. Jennifer took out her mobile phone and Alice saw that it was an NFC enabled mobile phone too.

1. *Business card transfer:* Alice and Jennifer touch their mobile phones to each other. Alice's mobile phone requests sharing business cards and Jennifer accepts this. Then the sharing process occurs (see Figure 4.15).

Both Alice and Jennifer received each other's contact information and hence will be able to communicate in the future.

Figure 4.15 Example usage model of exchanging data.

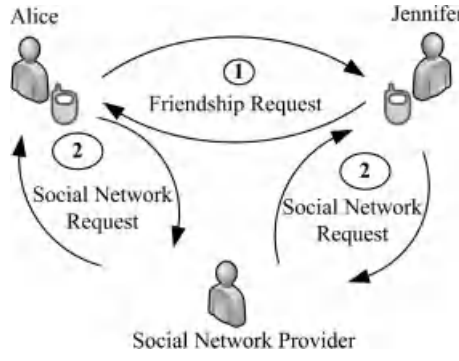


Figure 4.16 Example usage model of social networking.

(ii) *Social networking*

Alice and Jennifer continued to chat while waiting for the next lecture. Jennifer mentioned to Alice about a funny video that she had seen on her social networking site which Alice had also seen the previous day. Jennifer was pleased that Alice is also using the same social networking site and suggested becoming friends on their social networking site.

1. *Friendship request:* Alice and Jennifer request social network friendship by touching their mobile devices to each other (see Figure 4.16).
2. *Social network request:* Alice's mobile application forwards the friendship request and Jennifer's mobile application forwards the friendship acceptance request to the social networking site and then they become friends on the site.

Alice and Jennifer were happy to become friends on their social networking site. They will be able to communicate easily from now on.

4.4.4 Underlying Application Benefits

The number of applications developed using peer-to-peer mode is less than for other operating modes so far. Peer-to-peer mode is generally used for device pairing, networking, and file transfer operations. Pairing Bluetooth devices, exchanging business cards, and making new friends on online networks are possible implementations of this mode. Peer-to-peer mode provides easy data exchange between two devices.

Easy data exchange between two NFC compatible devices provides the possibility for secure exchange of private data. NFC devices can transfer data in a few centimeters, so exchanging private and important data may be one of the key future applications of this mode.

4.5 Card Emulation Mode

Card emulation mode provides the opportunity for an NFC enabled mobile device to function as a contactless smart card. Mobile devices can even store multiple contactless smart card

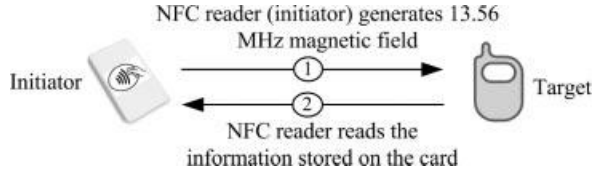


Figure 4.17 Card emulation mode.

applications in the smart card. Examples of emulated contactless smart cards are credit card, debit card, and loyalty card.

In this operating mode an NFC enabled mobile phone does not generate its own RF field; the NFC reader creates this field instead. Currently supported communication interfaces for the card emulation mode are ISO/IEC 14443 Type A and Type B, and FeliCa. Card emulation mode is an important mode since it enables payment and ticketing applications and is compatible with existing smart card infrastructure. A schematic representation of card emulation mode is given in Figure 4.17.

4.5.1 Generic Usage Model

In card emulation mode, the user interacts with an NFC reader, generally using her mobile phone as a smart card. The NFC reader is owned by a service provider which is possibly connected to the Internet as well. In this operating mode, the user connects to a service provider through an NFC reader possibly without notifying the service provider. Figure 4.18 shows the usage model of the card emulation mode.

1. *Service request:* The user makes a request to a service provider by touching her mobile phone to an NFC reader. Required data are transferred from the mobile phone to the service provider through the NFC reader.
2. *Backend services:* The service provider runs the required backend service after getting the required data from the user's mobile device. Examples of these services are credit card authorization and ticket validation.
3. *Service usage:* The service provider returns a service to the user such as issuing a ticket that has already been bought by the payment card, authorizing the payment, and so on.

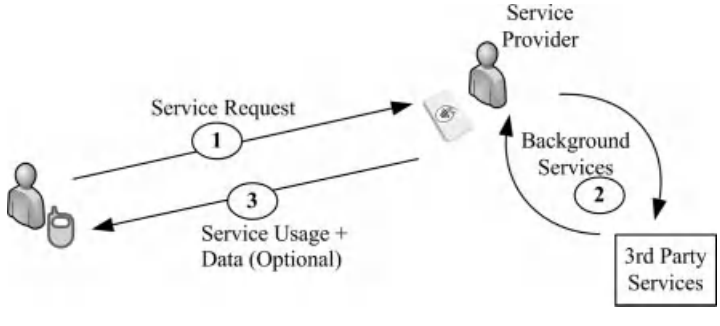


Figure 4.18 Generic usage model of card emulation mode.

The service provider may also optionally send data to the user's mobile device such as a coupon, ticket, receipt, and so on.

4.5.2 Leading Applications

In this section major implementations of card emulation mode are given.

(i) *Payment*
There can be different types of NFC payment applications. There is no doubt that the most important payment applications are credit and debit card usage which can be triggered by NFC readers. There are also other opportunities for NFC payment such as storing and using vouchers, using gift cards, and so on.

(ii) *Loyalty*
Loyalty points can be gained at payment points and can later be used to carry out free shopping or to obtain a gift. Also, coupons that are downloaded via smart posters using reader/writer mode can further be used by NFC readers using this operating mode.

(iii) *Ticketing*
Ticketing use cases can be implemented in different forms. Users can store different types of tickets such as theater, bus, and flight tickets that have been previously downloaded via smart posters or by some other method. These tickets then can be used in turnstiles or validation points via card emulation mode. Prepaid or monthly ticketing cards can also be stored and used.

(iv) *Access control*
Access control use cases enable users to store their access control objects in their mobile devices. Examples of these cases include electronic keys for cars, buildings, secure areas, and hotel rooms. Hotel check-in is an interesting use case that enables the room key to be received via OTA technology prior to arriving at the hotel and directly checking into the room. Thus there is no reason to spend time at the reception on arrival in this instance.

(v) *Identity services*
Another interesting use case of this mode is storing identity based information on mobile devices and enabling authorized personnel to access it. An example of this type of service is storing patient data. A patient's history can be stored on a mobile device and the user can then choose to give permission to a doctor to access those data via an NFC reader. This increases user privacy since undesired third parties such as insurance companies will not be able to access the patient's information. Even more interesting applications can be developed in this category such as integrating national identification cards, passports, fingerprints, and driver licenses to mobile phones.

(vi) *Smart environment*
The smart environment case refers to using NFC technology in smart environments (smart home, office, etc.). The most common example is managing smart environments via preconfigured data on a mobile phone. In this case, when a user enters a smart environment, specific smart environment settings (e.g., brightness level, song selection) can be adjusted by the mobile device. It can also be integrated with an access control mechanism so that when a user opens the door using her NFC electronic key, a personalized smart environment can be activated.

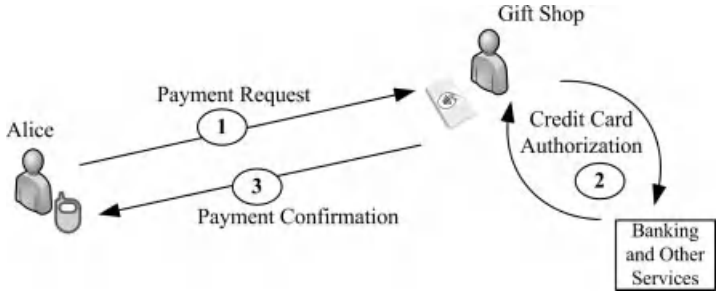


Figure 4.19 Example usage model of credit card payment.

4.5.3 Use Cases on Card Emulation Mode

In this section, credit card payment and ticketing use cases are presented and illustrated in detail.

(i) *Credit card payment*

Alice was happy to find a present for Bob's birthday at the shopping center. She took it to the cashier who asked what payment method Alice would be using. Alice noticed the NFC sign and informed the cashier that she would pay by NFC credit card.

1. *Payment request:* Alice requests NFC payment by touching her mobile phone to the NFC reader (see Figure 4.19). The NFC reader reads the required credit card data and processes them. The mobile phone may be seen as the initiator in this process, however the device that creates the magnetic field is the NFC reader, and thus the NFC reader is the initiator device. The user's touch action can be seen as the triggering action.
2. *Credit card authorization:* The NFC reader sends the card information to the banking services for credit card authorization. The authorization service replies to the transaction request.
3. *Payment confirmation:* The mobile phone is then notified about the transaction.

Alice was happy to pay with her mobile phone.

(ii) *Ticketing*

Alice gave Bob his birthday present. They were out to watch a movie together.. Alice had previously booked their seats and the tickets were saved on her mobile phone. When they arrived at the theater, they directly went in without having to wait at the ticket queue. They then approached the turnstiles. Alice was glad to have an NFC enabled mobile phone and was happy to have booked the tickets with it.

1. *Entrance request:* Alice requests entrance to the theater by touching her mobile phone to the NFC reader. The NFC reader reads the required ticket data stored in the secure element and processes them (see Figure 4.20).
2. *Ticket validation:* The NFC reader sends the ticket information to the ticketing server in order to validate it. Alternatively, validation can also be carried out offline via an application on the NFC reader.
3. *Turnstile opening:* After the ticket is validated, the system opens the turnstile, so that Alice and Bob can pass through.

Alice and Bob did not have to wait in the queue and were not delayed going to the movie.

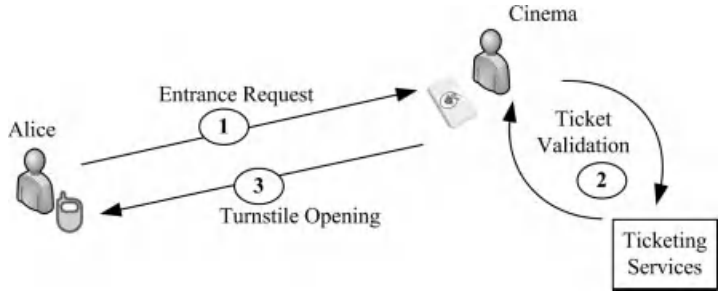


Figure 4.20 Example usage model of ticketing.

4.5.4 Underlying Application Benefits

We previously stated that mobile phones provide mobility to users in reader/writer mode. However, card emulation mode does not support mobility; its aim is to make the mobile phone tightly linked to its user instead. What we mean is that using a mobile phone in card emulation mode does not affect the mobility of the user because the user is already mobile even when she carries credit cards to perform the same functions. Card emulation mode only removes the need for carrying the cards. People carry mobile phones with them most of the time so coupling mobile phones with the human body fits with their use. One can expect that in the near future people will carry NFC enabled mobile phones not just to gain mobility but also to perform daily functions as well. All credit cards, keys, tickets, and so on will be possibly embedded into mobile phones. Hence, there will be more opportunities to integrate daily objects into NFC enabled mobile phones in the future.

To summarize, the dominating characteristic of the card emulation mode involves eliminating the need for a physical object. A few examples may illustrate this characteristic. In payment applications, the use of a mobile phone eliminates the need to carry contact and contactless credit cards, debit cards, and cash. Moreover in ticketing applications, an NFC enabled mobile phone eliminates physical objects by removing the need to carry paper based or electronic tickets. In an access control or an electronic key application, NFC usage eliminates the need to carry a physical key or a contactless smart key. As it is used to enter rooms without using electronic keys, it also provides access control. As a result, the most important benefits of card emulation mode applications are eliminating physical objects and providing access control.

We are able to eliminate the need for carrying physical objects by embedding that information into NFC enabled mobile phones. Currently credit cards, tickets, keys, and coupons are used with NFC enabled mobile phones, but there still exist many exciting usage options. We will be able to store so many objects such as identification cards, passports, fingerprints, and driver licenses on the mobile phone. Most of these future development options still have issues to solve, but they all have a high potential to be realized.

4.6 Overview on Benefits of Operating Modes

The benefits of each operating mode are given in each mode's section above. Table 4.3 summarizes the benefits of each operating mode.

Table 4.3 Benefits of each operating mode

Reader/Writer Mode	Peer-to-Peer Mode	Card Emulation Mode
- Increases mobility - Decreases physical effort - Ability to be adapted in many scenarios - Easy to implement	- Easy data exchange - Device pairing	- Physical object elimination - Access control

As a summary, the prominent mode of NFC is card emulation mode, because NFC yields two big improvements in this mode: elimination of a physical object and providing access control using a mobile device. Commercially available applications (payment, electronic key, ticketing, etc.) generally use the card emulation mode. The card emulation mode is currently the most promising mode of NFC technology [4].

Also an important point to be compared in operating modes is access to the service provider. Communication with the service provider is performed in different ways in each mode. In reader/writer and peer-to-peer modes, the mobile phone performs the connection using the mobile phone's Internet access capability, whereas it is performed by the NFC reader in the card emulation mode. Consequently, the user is not aware of being connected to the service provider in card emulation mode, since the connection is established by the reader seamlessly, and the user does not take any action to make the connection personally. However, in the other modes, the user is aware that she is getting a service, since the service is visualized in the mobile phone (in a browser, application, etc.) and mostly required an approval as well. A summary of interaction with the service providers is given in Table 4.4.

4.7 Case Studies

We present three case studies to illustrate all three operating modes and their usages. The first case is an NFC enabled shopping system which enables users to shop online without any geographical restrictions. Reader/writer mode is used in this case. The second case is an NFC based gossiping application which works in the same way as gossiping and disseminates information among peers using peer-to-peer mode. The third case is a movie ticketing application which also allows payment by making use of the card emulation mode.

For each case study, the description of the case is given first. Then, it is followed by use case diagrams, activity diagrams, and generic usage models. Application programming for the first two cases is given in Chapter 5. These cases are open NFC applications which means

Table 4.4 Interaction with the service provider

Operating Modes	Initiator	Target	Connecting to Service Provider	Awareness of Service Provider
Reader/writer	Mobile device	NFC tag	Through Internet	Yes
Card emulation	NFC reader	Mobile device	Through NFC reader	No
Peer-to-peer	Mobile device	Mobile device	Through Internet	Yes

that the cases do not need to include multiple actors and an NFC ecosystem. Users can use those applications after a successful implementation and installation. However, the third case study is a secure NFC application and involves many actors in its structure. We analyze this case study's ecosystem environment and business models at the end of Chapter 7 but do not provide its programming.

4.7.1 Reader/Writer Mode Case Study: NFC Shopping

(i) *Case introduction*

ROSCAXT is a supermarket chain that operates in many cities. They are currently offering both in-market shopping and online shopping via their website. ROSCAXT's main intention is to decrease in-market costs by increasing online shopping; thus they will be able to decrease their cost and increase revenue. However, they are dissatisfied with the users' low amount of penetration to the web market and they want to offer easier shopping experience to users with new and innovative technologies. Users should also have fun on the new online site, so that they will shop from it regularly.

(ii) *Integrating NFC technology into business*

By realizing that NFC is a new and innovative technology and provides an easy and enjoyable experience to users, the management thought that it would be nice to launch online shopping with NFC, so that the company would have the chance to overtake its competitors. They decided to develop an application for mobile devices to use NFC enabled online shopping. They have started a project named "NFC Shopping" and let their IT department build an NFC enabled system.

After the feasibility analysis, the IT department decided to provide NFC shopping to users by distributing NFC enabled shopping catalogs. In the catalog, products are printed with their descriptive information and price. Additionally, an NFC tag is integrated in the page for each product. Users will be able to shop online by touching tags on the catalog with their mobile phones and order online without any geographical restrictions.

(iii) *Development of the system*

In order to develop NFC Shopping the following steps are performed:

1. Description of the use case;
2. Use case diagram;
3. Activity diagram;
4. Data in the tag;
5. Applying generic usage model of reader/writer mode to use case.

(iv) *Description of the use case*

It was a busy day for Alice. She has to cook for her guests and needs to shop for ingredients but unfortunately she has no time to go shopping. She remembers that she already received the new market catalog. There was an NFC logo on the catalog so Alice decides to try this new service. She has the catalog and her mobile phone. On the first page, she reads the description on how to shop using an NFC enabled mobile phone. She finds the first item she needs to buy and touches her mobile phone to the product. Then the application is launched and the selected item is added to the basket. Alice touches her mobile phone to the goods one by one and adds them to the basket. There was only

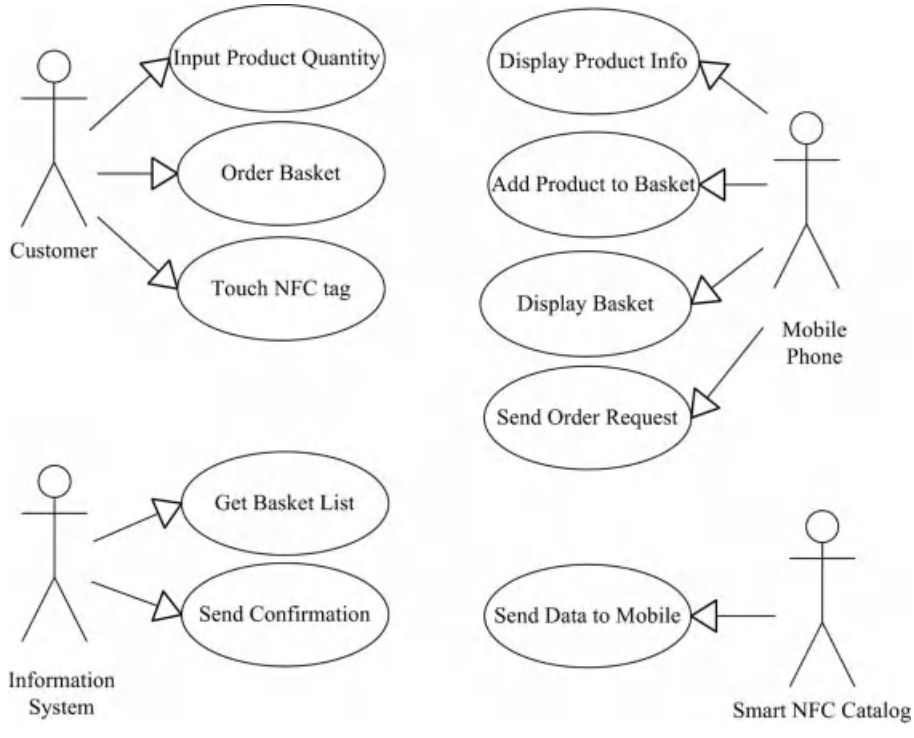


Figure 4.21 NFC Shopping use case diagram.

one remaining action to perform: ordering the items. She ordered her basket without exiting the application, and that was the end of the process.

(v) *Use case diagram*

In use case diagrams, we mainly show which functions of the application are performed by which actor. We include four actors in our specific system; namely, the customer, the customer's mobile phone, NFC catalog, and backend system. The customer is one of the main actors who performs communication using the mobile phone. She also selects the product by touching her mobile to the related tag, and enters the required quantity to the application. When the shopping basket is ready, the user orders it over the Internet. The mobile phone performs the actual processing as well as the communication with NFC tags. The information system performs the required backend server operations such as processing the basket and sending the confirmation response to the mobile phone. The NFC tag on the other hand only sends the stored data to the mobile device. The use case diagram is given in Figure 4.21.

(vi) *Activity diagram*

As stated above, four actors are involved in the current system. The activity diagram given in Figure 4.22 shows the step by step activities, actions of the four actors as well as the communication with other actors.

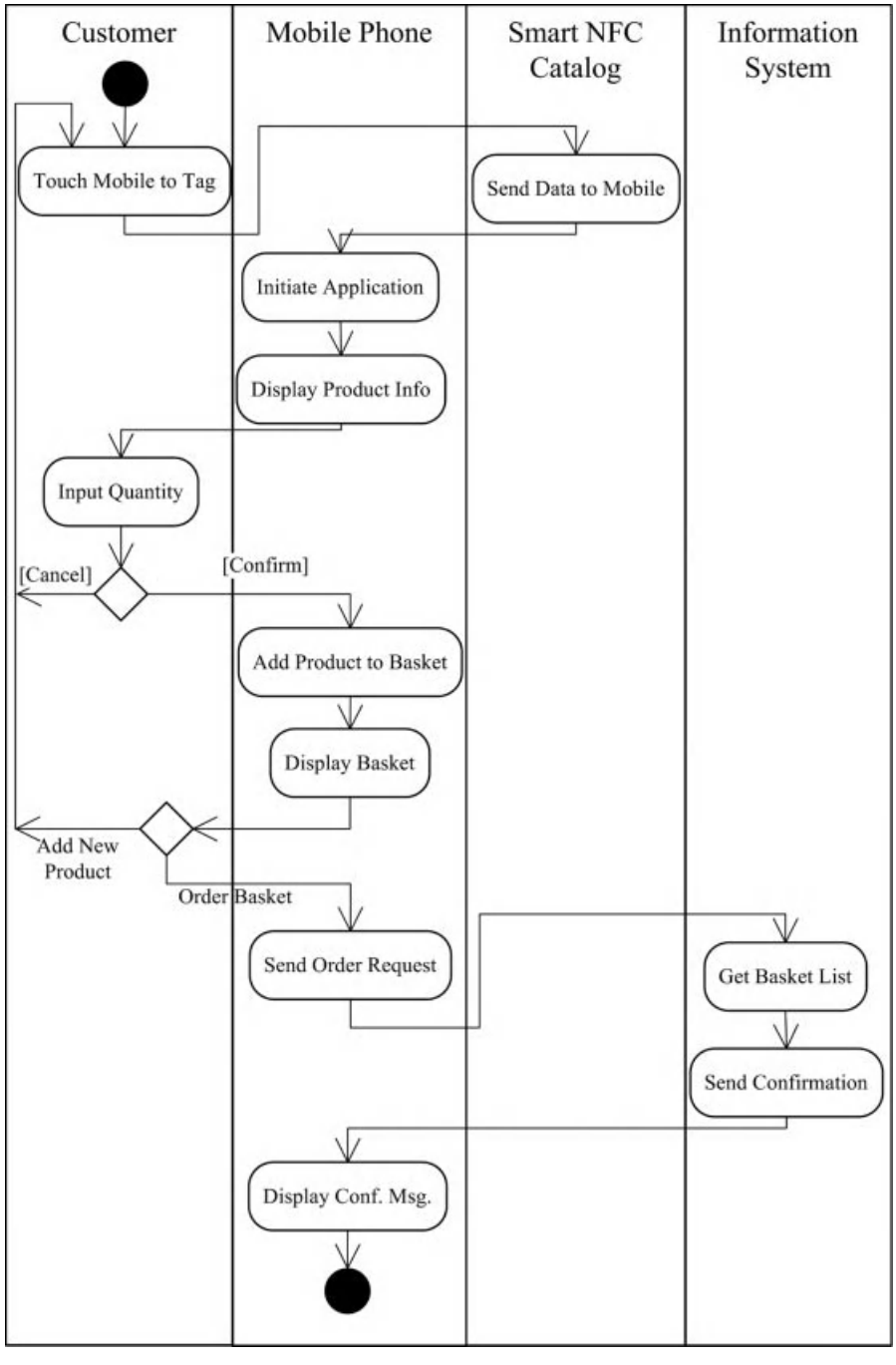


Figure 4.22 NFC Shopping case activity diagram.

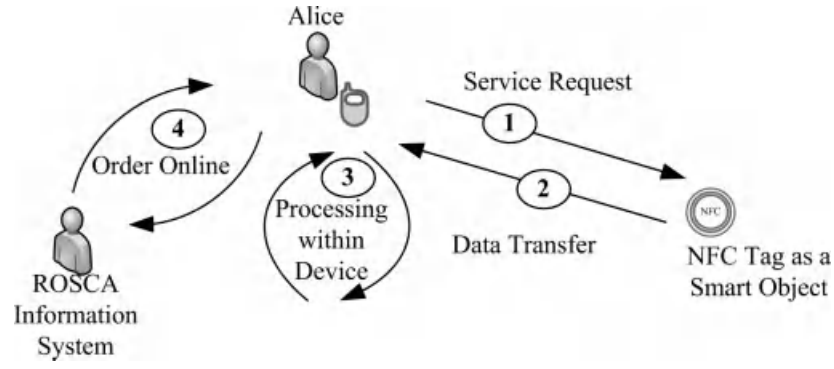


Figure 4.23 Usage model of NFC Shopping case.

(vii) *Data in the tag*

In order to implement the service, required data should be transferred from the NFC tag to the mobile device:

- *Push registry entry:* This is used to run the mobile application automatically when a user touches a tag.
- *Product identification:* This is the primary key for each product. The data will be sent to the backend information server after ordering the basket together with the desired quantities. The information system will be able to identify the desired products using the data. Note that the user will input the desired quantity of a product.
- *Product information:* This is used to display a short description of the product to the user.
- *Product price:* This is used to display the price of the product to the user.

(viii) *Generic usage model of NFC Shopping*

In this chapter, we have already introduced the generic usage models of the operating modes. The generic usage model of an operating mode provides a model that almost fits with the usage of all applications in the same operating mode. The NFC Shopping use case makes use of the reader/writer mode, and its usage under the generic usage model of the reader/writer mode is given below (see Figure 4.23).

1. *Service request:* Alice touches her mobile phone to the tag on the smart NFC catalog.
2. *Data transfer:* Push registry, product identification, product information, and product price are transferred to the mobile phone.
3. *Processing by the mobile device:* The application is started with push registry data (if not started already). Then, the mobile phone performs the required processing such as displaying product information to the user, asking for input from the user, adding a product to the basket, and so on.
4. *Order online:* Alice orders the current basket, and the mobile phone communicates with the supermarket's server through a mobile Internet service or a Wi-Fi service. The server confirms this request and sends a confirmation message about having received the required data successfully.

4.7.2 Peer-to-Peer Mode Case Study: NFC Gossiping

(i) *Case introduction*

Gossiping is a social interaction, namely idle talk mostly on private issues. It is one of the most common methods to share information between people. Although gossiping sometimes refers to sharing misinformation between people, in this case we refer to gossiping as idle talk to disseminate information.

The gossiping case's aim is to establish a digital gossiping service, so that people can disseminate information to others peer-to-peer. NFC technology here provides an important infrastructure, since the communication is performed over a short distance and requires an actual touch. As in classic gossiping, two people need to be close enough to each other in NFC gossiping. However, the information is not a tacit knowledge as in classic gossiping, instead it resides in a mobile phone's storage.

(ii) *Development of the system*

In order to develop the required NFC gossiping the following steps are performed:

1. Description of the use case;
2. Use case diagram;
3. Activity diagram;
4. Data structure of gossip data;
5. Applying generic usage model of peer-to-peer mode to use case.

(iii) *Description of the use case*

Alice was at the campus and sitting in the cafeteria. She remembered that she had heard of something that had happened between Joe and Jennifer yesterday. She created message by typing into the gossiping application and saved it. Later she saw Bob and wanted to share this gossip with him. She selected the message to share and touched her mobile phone to Bob's. Bob got the message, read it, and saved it to his mobile phone.

As seen in the use case, NFC gossiping is nearly identical to classic gossiping. New gossip can be created and gossip can be exchanged with others using NFC technology. Since this technology allows transferring data within a few centimeters, it fits with the idea of classical gossiping.

(iv) *Use case diagram*

Our system includes the user, user's mobile phone, and remote mobile phone as the actors. The user is the one who reads the incoming gossips and creates new gossips. The user's mobile phone performs the necessary NFC communication functions such as sharing the gossip with the remote mobile phone, displaying the received gossip and storing gossips. The use case diagram for NFC gossiping is given in Figure 4.24.

(v) *Activity diagram*

The activity diagram represents step by step activities and actions of the actors (see Figures 4.25 and 4.26).

(vi) *Data structure of gossip data*

In order to implement the required service, gossip data should be stored in the mobile device's storage. The following data need to be stored in order to perform the necessary functions:

- *Gossip message:* This is the actual message that will be sent to peers.
- *Group name:* This holds the defined group names. By default, "public" group is defined in the application. Further groups can be defined by the user.

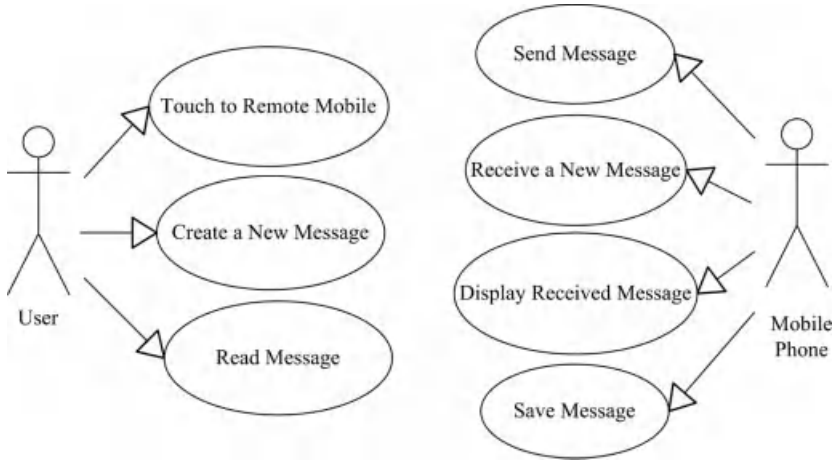


Figure 4.24 NFC gossiping use case diagram.

- *Message group:* This stores the corresponding group of each message. Since each message will be shared to the next person according to its group, the group data are required.

(vii) *Generic usage model of NFC gossiping*

The described use case fits with the generic usage model of peer-to-peer mode (see Figure 4.27). Although there is an optional service usage phase in the generic usage model of the peer-to-peer mode, it is not required in the gossiping case. There is no need for any additional service usage from the Internet, so message exchange will be made only locally.

1. *Gossiping message transfer:* Alice transfers the message to Bob that she wants to share.

4.7.3 Card Emulation Mode Case Study: NFC Ticketing

(i) *Case introduction*

ROSCAXINE is a cinema company operating in many cities. When the traditional method is used, a customer can buy a ticket via the kiosk by paying with a credit card. In order to pay, the customer swipes a credit card at the kiosk machine. The machine prints a paper ticket for the user. At the entrance of the cinema hall, a barcode reader reads the barcode that is embedded on the printed ticket and opens the turnstile. As another option, a customer may also buy tickets at the box office. The customer can pay either with cash or credit card. Then the ticket is printed by the clerk and given to the customer.

ROSCAXINE's main intention is to increase brand awareness by promoting new products to its users. They want to benefit from NFC's newly emerging technology.

After analysis, they have found out that NFC technology provides an easy and positive experience to users. They had no reservations about embedding NFC technology in their business projects.

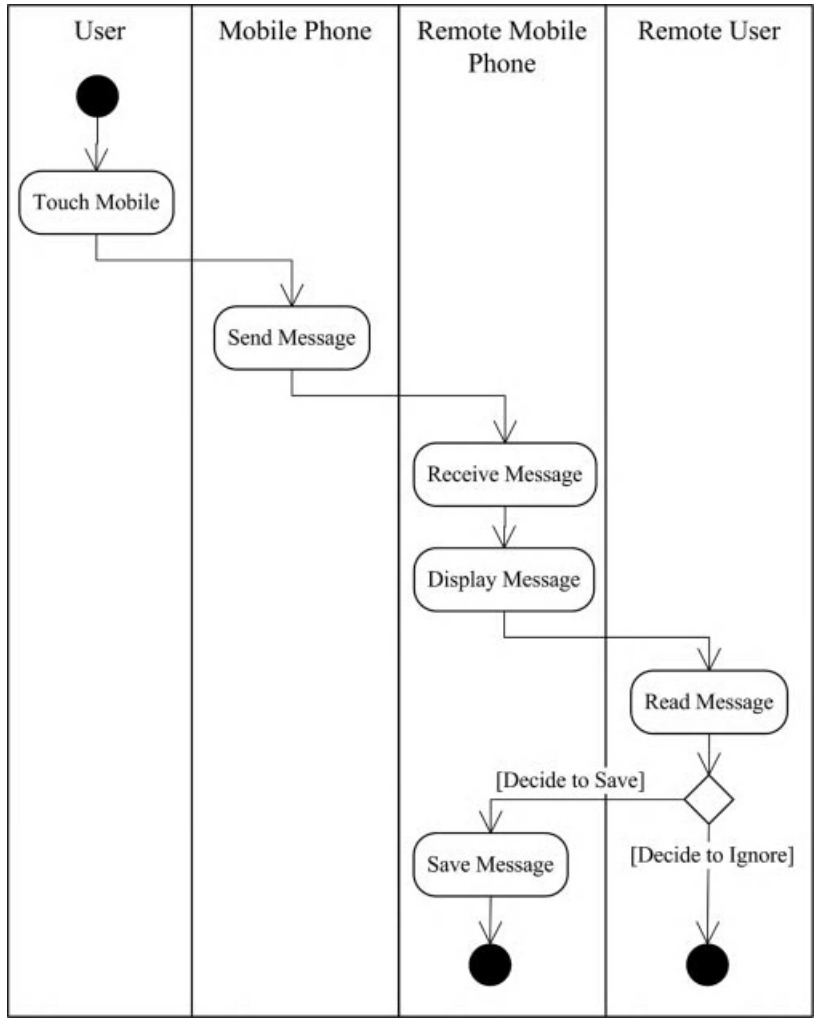


Figure 4.25 NFC gossiping activity diagram.

(ii) *Development of the system*

In order to develop the required NFC ticketing project, we will use the following steps:

1. Description of the use case;
2. Use case diagram;
3. Activity diagram;
4. Applying generic usage model of card emulation mode to use case.

(iii) *Description of the use case*

Bob was at home when his mobile phone rang. It was Alice, and she proposed going to a movie. Alice and Bob went to the theater to get their tickets. Alice approached a kiosk to buy tickets and found out that ROSCAXINE had just integrated NFC technology

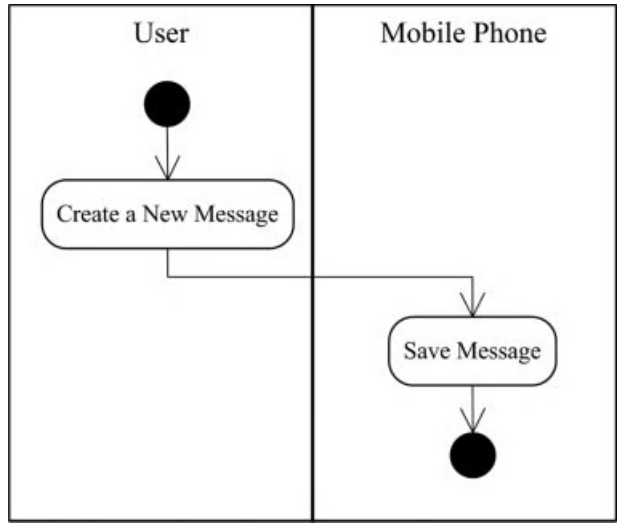


Figure 4.26 NFC gossiping activity diagram.

into the kiosk machines. She selected the movie, time, and seats; and then she paid for the tickets using her NFC enabled mobile phone by touching her mobile phone to the reader embedded at the kiosk. After the payment was confirmed, she touched another tag to transfer tickets to her mobile phone. It was a simple process. Then Alice and Bob approached the turnstile at the entrance of the cinema. Alice touched her mobile phone to the reader on the turnstile after which the turnstile opened.

(iv) *Use case diagram*

Our system includes the user, user's mobile phone, kiosk machine, turnstile, and remote backend systems as the actors. The user is the actor who initiates the process. The kiosk reads the credit card information on the mobile phone, and creates and sends the tickets via OTA. The reader at the turnstile reads the ticket and opens the turnstile. The backend system's process is required for payment and ticketing operations. The use case diagram for NFC ticketing is depicted in Figure 4.28.

(v) *Activity diagram*

The activity diagram represents the step by step activities and actions of the actors (see Figures 4.29 and 4.30).

(vi) *Generic usage model of NFC ticketing*

The generic usage model of the card emulation mode is described in earlier sections. The NFC ticketing use case includes two card emulation applications. Thus, the given

Figure 4.27 Usage model of NFC gossiping case.

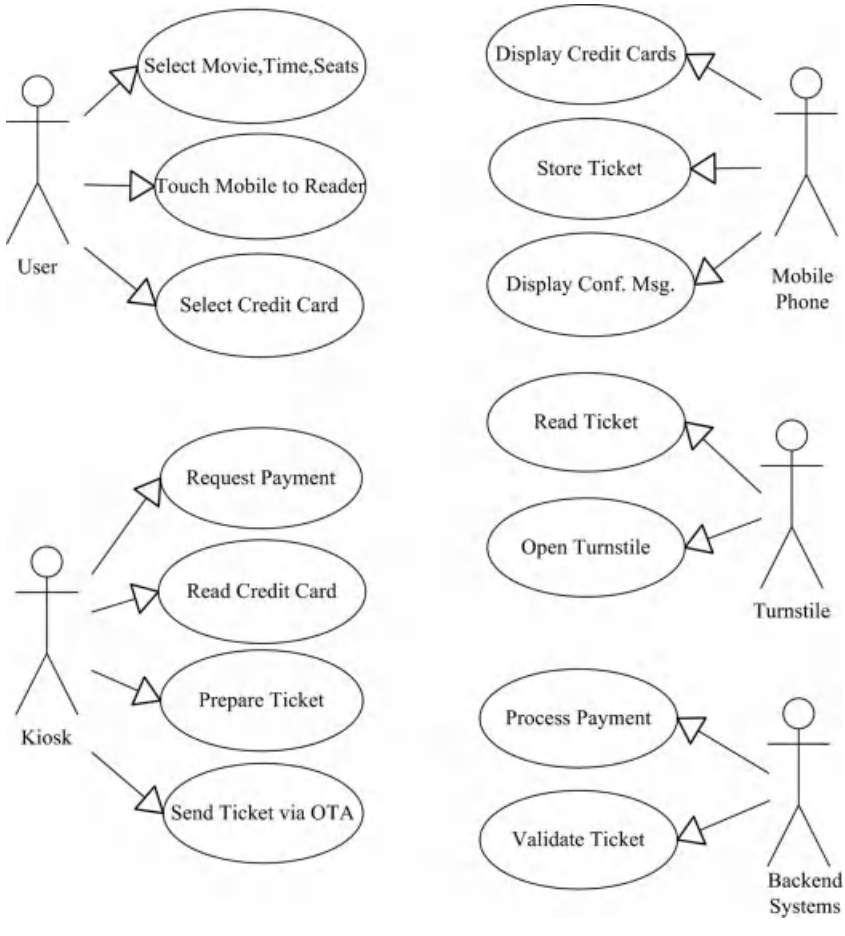


Figure 4.28 NFC ticketing use case diagram.

usage model of NFC ticketing includes the generic usage model twice. The NFC ticketing use case fits with the generic usage model of the card emulation mode (see Figure 4.31).

1. *Payment request:* Alice requests payment by touching her mobile phone to the reader on a kiosk machine. The NFC reader reads the required credit card data.
2. *Credit card authorization:* The reader sends the credit card information to the backend information system for credit card authorization.
3. *Payment confirmation:* The mobile phone is notified on the transaction and the ticket is sent to the device via OTA.
4. *Entrance request:* Alice requests entrance to the cinema by touching her mobile phone to the reader at the turnstile. The NFC reader reads the ticket and processes it.
5. *Ticket validation:* The NFC reader sends the ticket information to the backend system in order to validate it.
6. *Turnstile opening:* When the ticket is validated, the turnstile is opened.

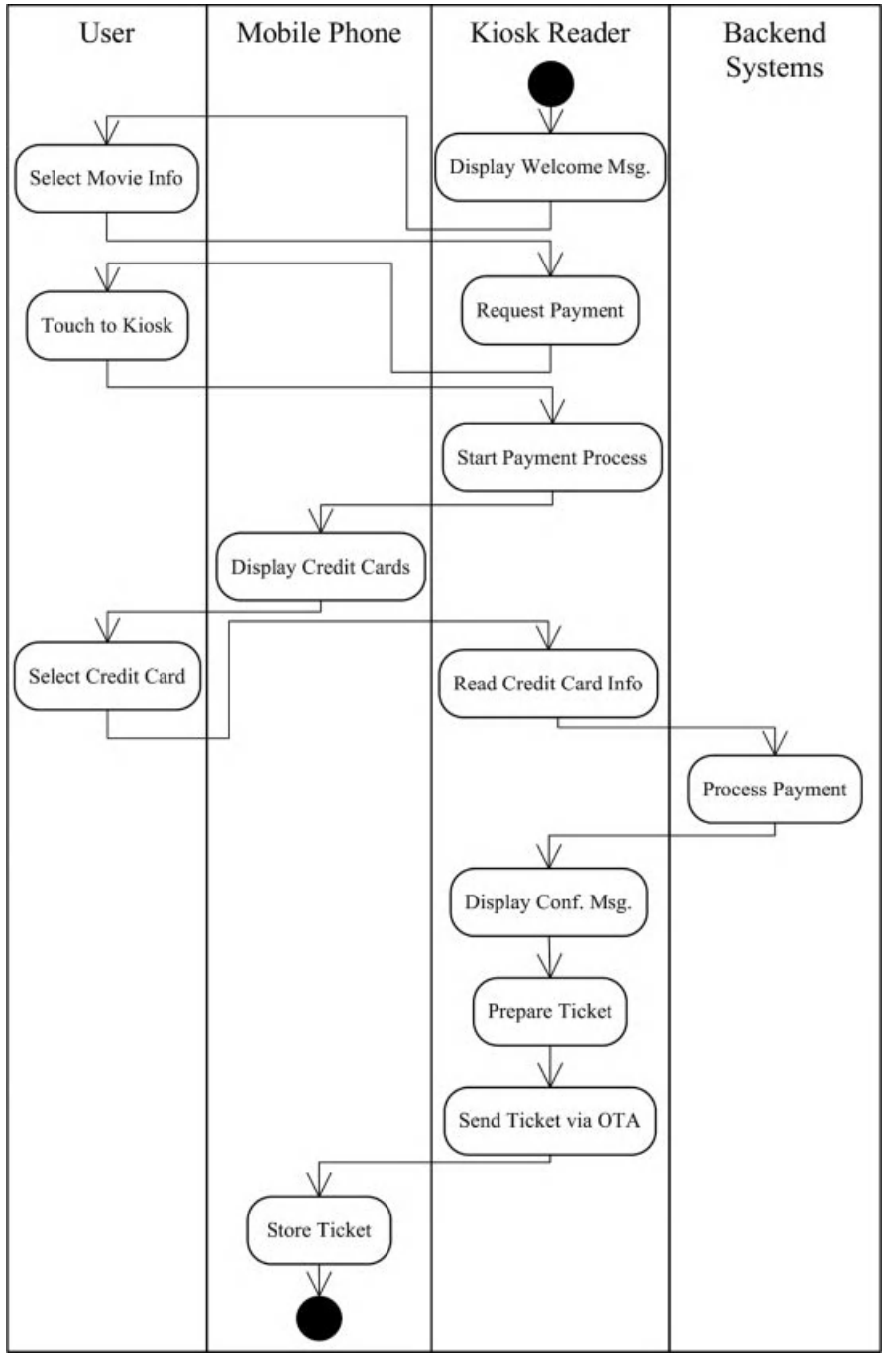


Figure 4.29 NFC ticketing activity diagram.

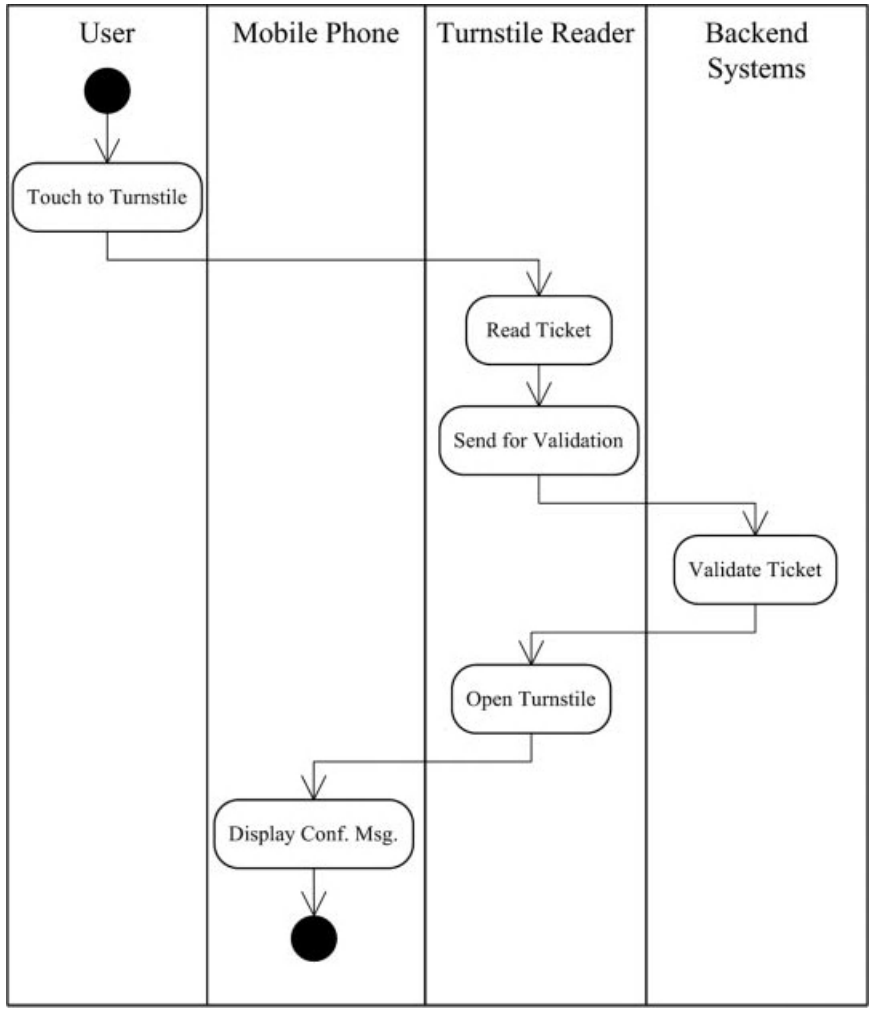


Figure 4.30 NFC ticketing activity diagram.

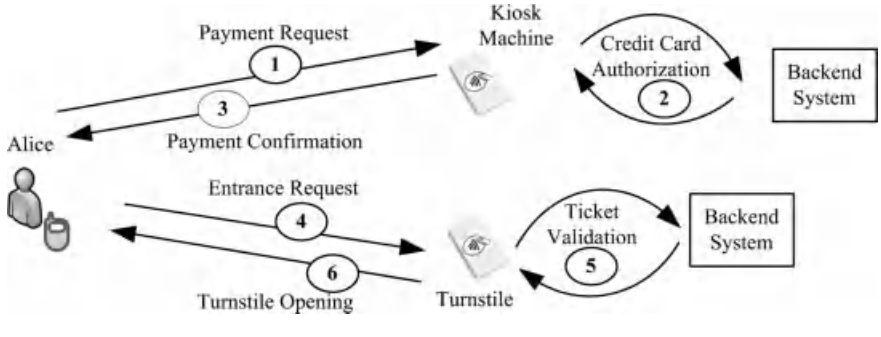


Figure 4.31 Usage model of NFC ticketing case.

4.8 Chapter Summary

There are three mobile interaction techniques in a mobile communication; namely touching, pointing, and scanning. NFC interaction is a touch based interaction technique since it occurs when two devices within a few centimeters are simply touched together. In order to perform an NFC interaction, a user needs to own an NFC enabled mobile phone and should interact with either an NFC tag, an NFC reader, or another NFC enabled mobile phone.

Three different operating modes are defined in NFC technology; reader/writer mode, peer-to-peer mode, and card emulation mode. Each mode's usage can be defined with generic usage models. Moreover each mode provides different benefits to users. Reader/writer mode applications can increase the user's mobility and decrease the physical effort. On the other hand, card emulation mode applications do not promote the mobility property of mobile phones; however, they tightly link users with mobile phones. This mode's benefits include eliminating physical objects and providing access control. Peer-to-peer mode applications are useful for easy data exchange between users and provide easy device pairing.

Designing smart posters is yet another important area that is highlighted in the chapter. Users should be aware of the touching point on the smart poster, otherwise they will need to search the smart poster which will decrease usability. Also, awareness of the tag content and the poster's trustworthiness are other important issues to consider.

This chapter considered NFC's use cases in terms of operating modes. Short use cases and generic usage models gave the basic design of applications. Detailed use cases included the requirements for application design and the characteristics of the operating modes.

Chapter Questions

1. What is the interaction technique of NFC? What are the basic properties of this technique?
2. What are the differences between active and passive devices?
3. How do initiator and target roles affect the communication mode of NFC?
4. Can a passive device be an initiator in NFC? Why?
5. Define a smart poster and explain its usage. Which operating mode do smart poster applications generally use?
6. Name and give details of one use case that uses reader/writer mode.
7. Name and give details of one use case that uses peer-to-peer mode.
8. Name and give details of one use case that uses card emulation mode.
9. Describe the underlying application benefits of reader/writer mode.
10. Describe the underlying application benefits of peer-to-peer mode.
11. Describe the underlying application benefits of card emulation mode.
12. Which NFC mode is most promising for financial organizations? Why?
13. Which NFC mode is most promising for mobile network operators? Why?
14. Which NFC mode is used for social networking applications? Give details.

References

[1] Rukzio, E., Callaghan, V., Leichtenstern, K., and Schmidt, A. (2006) *An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning*. Proceedings of Eighth International Conference on Ubiquitous Computing, CA, USA, 17–21 September 2006, pp. 7–104.

[2] QR Code, http://www.qrcode.com/ (accessed 10 July 2011).
[3] NFC Forum N-Mark for Tags and Media Frequently Asked Questions, http://www.nfc-forum.org/resources/N-Mark/ (accessed 10 July 2011).
[4] Ok, K., Coskun, V., Aydin, M.N., and Ozdenizci, B. (2010) *Current Benefits and Future Directions of NFC Services*. Proceedings of 2010 International Conference on Education and Management Technology (ICEMT), Cairo, Egypt, 2–4 November 2010, pp. 334–338.

5

Developing NFC Applications

This chapter is about developing NFC applications for mobile phones. Most properties of mobile phones differ from those of other devices such as monitor size, keyboard style, and so on. Hence, the codes must consider the properties of the target device in which the application will be installed and used. The application can be developed by considering the target device's Software Development Kit (SDK), but it requires further effort in order to test it irrespective of the target device. One option is transferring the application to the target device immediately after writing it once, which is not practical. No application can be developed error-free in the first time; this is especially true for mobile applications. If NFC mobile application is the issue, the transfer is even more impractical. Therefore, in addition to the SDK, a mobile simulator that supports the target device needs to be used for mobile application development. When the application is tested in the simulator, after all bugs are removed and it is error free, it can then be transferred to the target mobile phone.

There are various NFC development platforms and languages. For mobile phones with Android operating system, Android SDK [1] is used for NFC development. There is also another API (Application Programing Interface) in Qt SDK [2], which provides NFC technology support for Symbian^3 devices. Moreover, Java is the major development environment which can be used in most mobile phones. Java technology's NFC related APIs can also be used for the development of NFC applications. Other NFC application development platforms also exist and additional platforms may appear in the future as well.

In this chapter, we use Java technology to explain NFC application development. Readers are assumed to have a basic knowledge of Java technology as well as the capability of writing Java programs using, for example, J2SE. Additionally, some Java ME™ (Java™ 2 Micro Edition) programming content is given in Section 5.4 for those unfamiliar with it. The essence of this chapter, NFC application development, is given after the introduction to Java ME so readers with Java ME programming knowledge can skip to Section 5.5.

5.1 Initial Steps in NFC Application Development

Developing NFC applications is not trivial; it is one of the most important part of deploying NFC enabled services to customers. There are two types of NFC applications to consider at

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.
 Published 2012 by John Wiley & Sons, Ltd.

the development stage. One type is a GUI (Graphical User Interface) application which must be present for all operating mode applications and provides both a GUI for the user and the capability to read NFC components. The second type is a Secure Element (SE) application which is needed in order to provide a secure and trusted environment for applications requiring security such as payment, ticketing and authentication in SEs.

Reader/writer mode and peer-to-peer mode applications generally consist of only the GUI application since those operating modes do not require any secure operations. In Java language, MIDlets are the Java applications running on the mobile phone and provide the stated properties.

On the other hand, SE applications are used for card emulation mode applications. Card emulation mode applications consist of both GUI and SE components. These applications interact with NFC readers and MIDlets installed on a mobile phone. In Java language, Applets are JavaCard applications running on SEs or smart cards.

There are various development tools on the market and the user may choose the appropriate development tool for the targeted mobile phone, since applications are mobile phone dependent. For an Android mobile case for example, the application should be developed using Android SDK which can be downloaded from http://developer.android.com/. For a Symbianˆ3 mobile, the application should be developed with Qt SDK. This SDK can be downloaded from http://qt.nokia.com/.

SDKs also provide mobile phone simulators. Inside a simulator, tags can be created and edited. After the application development phase, the application should be transferred to actual mobile devices and should be tested in a real time environment. Applications can be installed on a mobile phone by connecting it to a computer using wired connection or Bluetooth or alternatively it can be installed online via an Internet resource.

Two JSRs (Java Specification Requests) are developed under the Java platform to enable NFC based applications. JSR 257 (Contactless Communication API) is for mainly reader/writer mode programming and JSR 177 (Security and Trust Services API) is for card emulation mode programming. JSR 257 is mainly concerned with discovering contactless targets in the proximity, notifying applications upon discovery, and performing tag operations. JSR 177 supports communication with smart card applications and also provides application level digital signature signing, user credential management, and cryptographic operations.

5.2 Why Java?

5.2.1 Why did we Choose Java?

To develop NFC applications easily in any platform, an NFC programming language should provide the fundementals of NFC programming and should serve as a basis role for readers. There are different NFC application development platforms currently available. New ones may also arise in the future.

We have chosen Java as the programming language; since it is widely used and is a well-known programming language. It also provided one of the first APIs in NFC technology. Nokia 6212 and Nokia 6131 NFC SDKs are development platforms that are able to work with JSR 257 and 177 which provide NFC programming in Java. Although those phones are outdated, the main function of this chapter is to give a basic knowledge of NFC programming. Learned NFC programming skills from Java technology will help users easily develop NFC

based applications in other programming languages for different mobile phones using different SDKs.

5.2.2 Why is Java the Favorite?

Why did Java become the favorite? The affirmative properties of Java definitely helped. Platform independency as well as being a common technology in all platforms made it favorable. When you want to write a standalone application on personal computers, you can use Java. If you want to export it to a web server, you can still use Java. When you want to implement programs on embedded hardware on refrigerators, Java can still be used. Mobile application development is still possible with Java. Hence, the same technology with some minor differences is enough on all platforms. Without Java, an application programmer has to learn different technologies for different environments.

Java™ is an object oriented programming technology which is primarily designed to be object based and platform independent. It was introduced by Sun Microsystems in 1995, and is a trademark of that company. Anyone wishing to implement a Java runtime environment must obtain permission from Sun and pass a comprehensive set of compatibility tests. The syntax and semantics of the Java language itself are described in a special document entitled The Java Language Specification (JLS). For more detail, please visit http://java.sun.com/docs/books/jls/index.html.

Java programs are written into files with .java extension. Java programs are not converted to machine codes, instead they are executed on interpreters online. The available interpreters vary according to the execution platforms. The converters on the personal computers are referred to as Java Virtual Machines (JVMs), whereas they are called Kilobyte Virtual Machines (KVMs) on mobile phones. Before running a Java program with .java extension, it should be compiled and converted to byte code with .class extension using a Java compiler. Java byte code programs are executed on top of the virtual machine.

The great advantage of this programming language is that Java programs are platform independent. This means that Java codes can be run on any hardware and any operating system as long as a compatible Java virtual machine is available.

The principal properties of the Java technology are as follows:

- *Platform independent and portable (write once, run everywhere):* Java byte code is interpreted by a virtual machine. Hence the Java application can be executed on any device that contains the required virtual machine, independent of the operating system and hardware. Nowadays, most mobile devices have the required JVM. When developing an application for a wide range of mobile devices, one development for many devices results in huge production savings.
- *Robust:* Even if a Java application crashes, it resides in the virtual machine and does not affect other applications or important data on the device. Additionally, a Java application has automatic memory management and garbage collection.
- *Secure:* Java technology provides a secure environment through a security manager and security APIs. Cryptographic functionality of the technology and public key infrastructure (PKI) integration enable the development of secure applications. Performing authentication and access control protects applications against unauthorized access.

- *Object oriented:* Java is an object oriented language in which the programming language model is organized around objects rather than actions.
- *Wide adoption at the backend:* Java clients can easily work with the backend Java application servers.
- *Multithreading capable:* A multithreaded application is essential in some specific problems; this is also provided by Java.
- *Network awareness:* Java applications are network aware (applications are dynamically downloaded over a network) and network transparent (can exchange data with a server using any technology such as GSM, CDMA, TDMA, and so on, and over any network protocol such as TCP/IP, WAP, i-mode, and so on).

However, technical merits are not the only factors that determine the success of a technology. The business values are just as important. In particular, the vendor's ability to address the concerns of the developer and user community are key to the technology's adoption. Now, let us look at how the Java™ community influences the evolution of the technology.

Java technology comprises the following three elements, the combination of which composes the Java platform:

- Java programming language standards which define the rules to write a Java program.
- JVM to execute Java programs which is written by satisfying the Java language standard.
- An extensive set of APIs that support a wide range of sources that a programmer needs.

The Java platform is designed to be available for a wide range of hardware, everything from mobile phones and smart cards to web application servers and enterprise servers. The Java platform comes in three forms (see Figure 5.1):

- *Java 2 Standard Edition (J2SE)* is designed for desktop computers [3].
- *Java 2 Enterprise Edition (J2EE)* is designed for a higher range of platforms such as multiuser or enterprise wide applications. It is based on J2SE and adds APIs for server side computing [4].
- *Java 2 Micro Edition (J2ME)* is designed for technologies and specifications used in small devices such as mobile phones and Personal Digital Assistants (PDAs) [5].

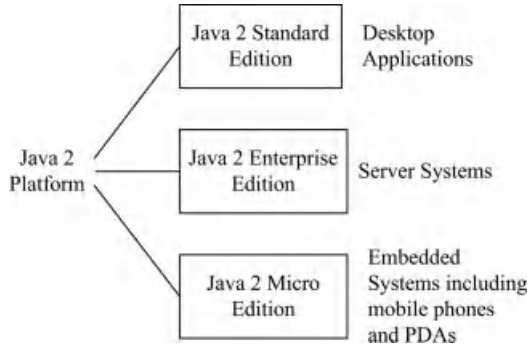


Figure 5.1 Java 2 platform.

Java technology's NFC related APIs are the main emphasis in this chapter. Java API specifications including Java ME configurations and profiles have been developed as part of the Java Community Process (JCP). The JCP tries to ensure that Java technology is developed according to community consensus to avoid industry fragmentation. The JCP brings together leading players in the relevant industries with the aim of agreeing on a common specification to which they can all design their products [6]. Each configuration or profile starts out as a JSR which describes the scope of the work that will be performed and an outline of the areas that will be covered [7]. An expert group is assembled to create the specification which is then subject to an internal ballot and revision before making the results available for public review. Following public review and a possible last revision, the final draft is produced and the JSR is completed. The current list of JSRs including those that have been completed can be found on the JCP website at http://jcp.org/jsr/all/.

5.3 Setting up the Environment for Java ME and NFC Programming

In this section, we will explain how to set up the environment for NFC application development that will be used throughout this book.

In order to develop Java applications, JDK (Java Development Kit) should be installed on the system.

A suitable environment to develop an NFC application requires three components:

- Java ME development environment.
- Simulator for the mobile device as well as NFC components.
- NFC extensions.

There are some available MIDP development environments as well as different simulators for different hardware platforms. Series 40 Nokia 6212 NFC SDK [8] is used for a mobile phone simulator with integrated NFC extensions. Series 40 Nokia 6212 NFC SDK provides an NFC development and testing environment for Nokia 6212 NFC devices. Also, an Integrated Development Environment (IDE) with the required add-ons for an NFC environment are required to write codes for mobile devices. There are various Java IDEs on the market and the user may choose any one that contains the required components. We have used the Eclipse software development environment in this book together with the Eclipse ME add-on to provide mobile application development support. The NFC SDK we have used is compatible with both Eclipse and Netbeans IDEs [8].

(i) *Series 40 Nokia 6212 NFC SDK*

NFC applications are developed for specific mobile phone platforms. We must remember that NFC is a relatively new technology, and not all mobiles are NFC capable yet. Some initial mobile phones have been introduced by the leading manufacturers such as Nokia, and the number of available models is increasing day by day. We use Nokia 6212 as the test bed for NFC applications in this book. Series 40 Nokia 6212 NFC SDK is the implementation of the Nokia 6212 classic device which also has NFC connectivity. This emulator software has an embedded NFC manager and Nokia 6212 classic device's simulator. The emulator enables developers to test and run MIDP applications

as well as NFC enabled applications. Using an NFC manager, new tags can be created and simulated; furthermore they can also be edited and saved for future use. A tag can be attached to the mobile phone simulator device and can communicate with it. Moreover, the interface allows supported NFC readers to easily connect to the emulator. SDK can be used as a standalone SDK or alternatively it can be integrated with a supported IDE such as Eclipse or Netbeans. Series 40 Nokia 6212 NFC SDK can be downloaded from http://www.forum.nokia.com/.

(ii) *Eclipse*

Eclipse projects are developed by the Eclipse Foundation which is an open source community. The Eclipse Foundation is a non-profit, member supported corporation that hosts Eclipse projects. Eclipse provides various IDEs that cover runtimes, static and dynamic languages, server side frameworks, modeling and business reporting, and embedded and mobile systems. We will use one of the IDEs of the Eclipse Foundation, namely Eclipse IDE for Java Developers for MIDlet application development in this book. The IDE can be downloaded freely from http://www.eclipse.org/downloads/. Also, for further information on the Eclipse Foundation and Eclipse IDEs, you can visit http://www.eclipse.org/ [9].

(iii) *Eclipse ME*

Eclipse ME is a plug-in for Eclipse IDE to develop MIDlets. The Eclipse ME plug-in makes MIDlet development possible by connecting to wireless toolkits.

In order to install Eclipse ME, you should have successfully installed both the Eclipse IDE and a suitable wireless toolkit which is designed for the target mobile phone. Series 40 Nokia 6212 NFC SDK contains a wireless toolkit, so you do not need to install a wireless toolkit separately. You can download the latest Eclipse ME plug-in from http://eclipseme.org/ [5, 10].

(iv) *Installation*

Installation of the platforms needs to be performed in following order:

1. *Installing Series 40 Nokia 6212 NFC SDK*
   a. Unzip the downloaded Series 40 Nokia 6212 NFC SDK archive to a folder.
   b. Run the installation file and follow the instructions. It will install Nokia 6212 NFC emulator, NFC manager and required files.
2. *Installing Eclipse IDE*

   Unzip the downloaded Eclipse IDE archive to a folder. Once you have unzipped, you will see a folder named “eclipse”. In that folder, you will find the eclipse executable file, “eclipse.exe”. The only remaining thing to do is to run the eclipse.exe file in order to start the Eclipse application. Obviously you can optionally create a shortcut to desktop to simplify the process.
3. *Installing Eclipse ME*
   a. Run Eclipse IDE.
   b. From “Help” menu on Eclipse, select “Install New Software”.
   c. Select “Add” and then “Archive” from the dialog (Figure 5.2).
   d. Select the downloaded Eclipse ME archive file and click “OK”.
   e. The Eclipse ME installation package will be listed in the dialog box. Click the checkbox next to it and then click “Next” (Figure 5.3).
   f. Read and accept the “License agreement” and click “Finish” and wait for Eclipse to install the package.

Figure 5.2 SDK installation screen-1.

4. *Configuring Series 40 Nokia 6212 NFC SDK with Eclipse*
   a. Run Eclipse IDE, if you have already shut it down.
   b. From "Window" menu, select "Preferences".
   c. On the left column, expand "Java" and click "Debug".
   d. Deselect all the options under "Suspend Execution" section.
   e. Under "Communication" section, set "Debugger timeout" and "Launch timeout" to 20 000 (ms).
   f. On the left column, expand "J2ME" and click "Device Management".
   g. Click "Import", and then select "Browse" (Figure 5.4).
   h. Select the root directory of the Series 40 Nokia 6212 NFC SDK installation folder, for example, "C:\Nokia\Devices\S40_Nokia_6212_NFC_SDK" and then click "OK".
   i. Click "Refresh" for Eclipse to search for devices in the selected folder (Figure 5.5).
   j. Once the Series 40 Nokia 6212 NFC SDK is found and displayed at "Devices", click the checkbox under "Import" and click "Finish" (Figure 5.6).
   k. Finally, click "OK" to apply settings and to close the "Preferences" window.

Once you have successfully installed and configured the applications, you are ready to develop Java ME MIDlets including NFC applications.

If you are running Windows Vista or Windows 7, then you should set the compatibility of "emulator.exe" in "C:\Nokia\Devices\S40_Nokia_6212_NFC_SDK\bin" to Windows XP.

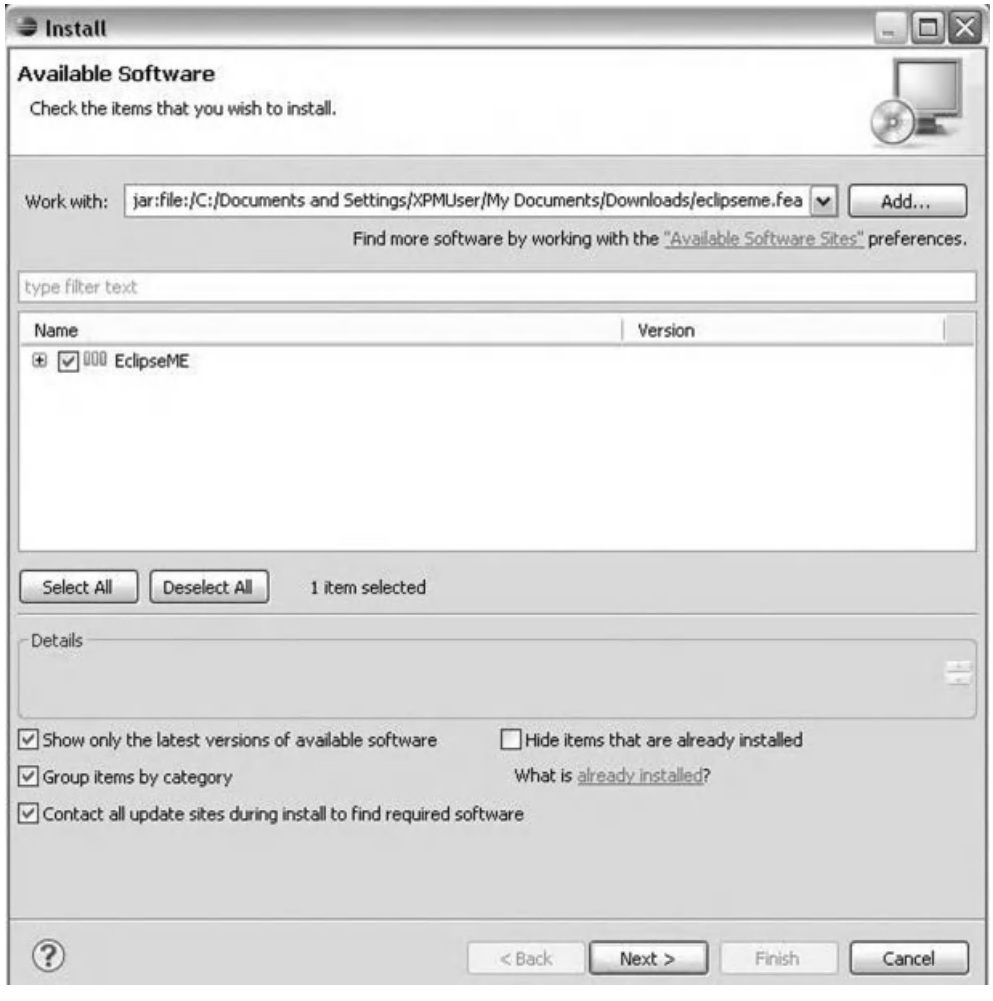


Figure 5.3 SDK installation screen-2.

Alternatively, you can use another wireless toolkit (e.g., Sun Java Wireless Toolkit) as a mobile emulation environment for MIDlet development. Series 40 Nokia 6212 NFC SDK is preferred, since it will be also used in NFC programming throughout this book.

5.4 Introduction to Mobile Programing

This section contains a brief introduction to Java ME for Java programmers who intend to write applications for mobile phones or PDAs. These devices have a common property; they have limited memory resources and processing power with respect to regular computers. Therefore, programs written for mobile devices are less capable with respect to input, output and processing. In order to overcome performance restrictions, mobile applications should be designed and implemented more carefully.

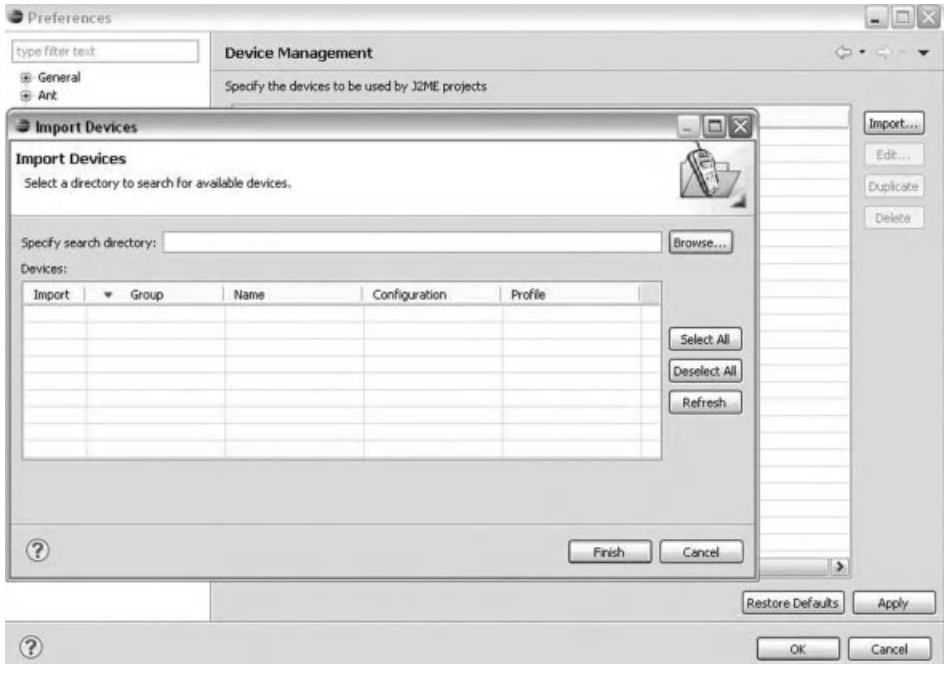


Figure 5.4 SDK installation screen-3.

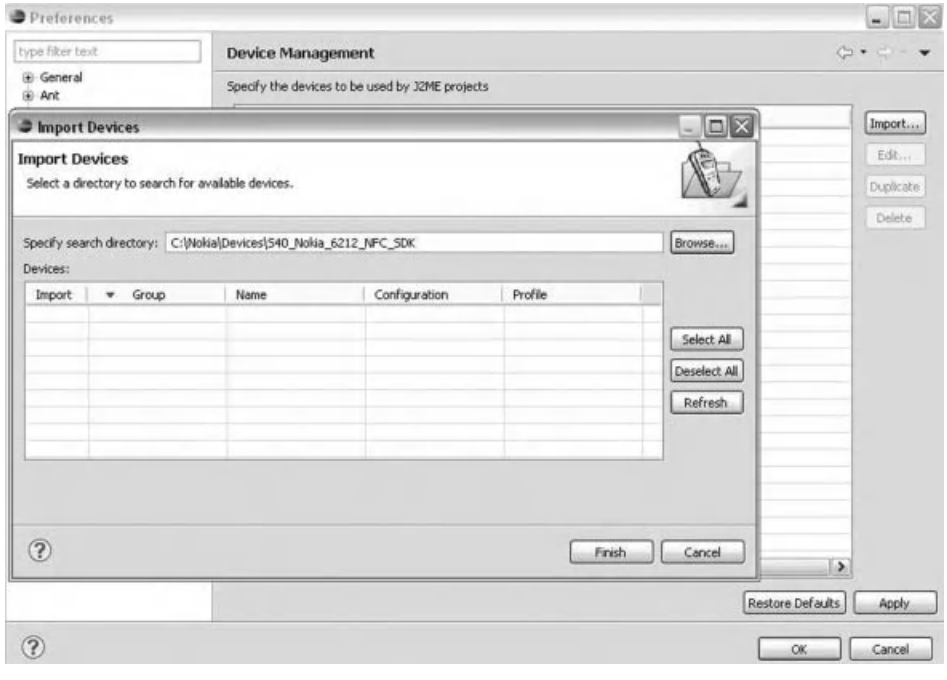


Figure 5.5 SDK installation screen-4.

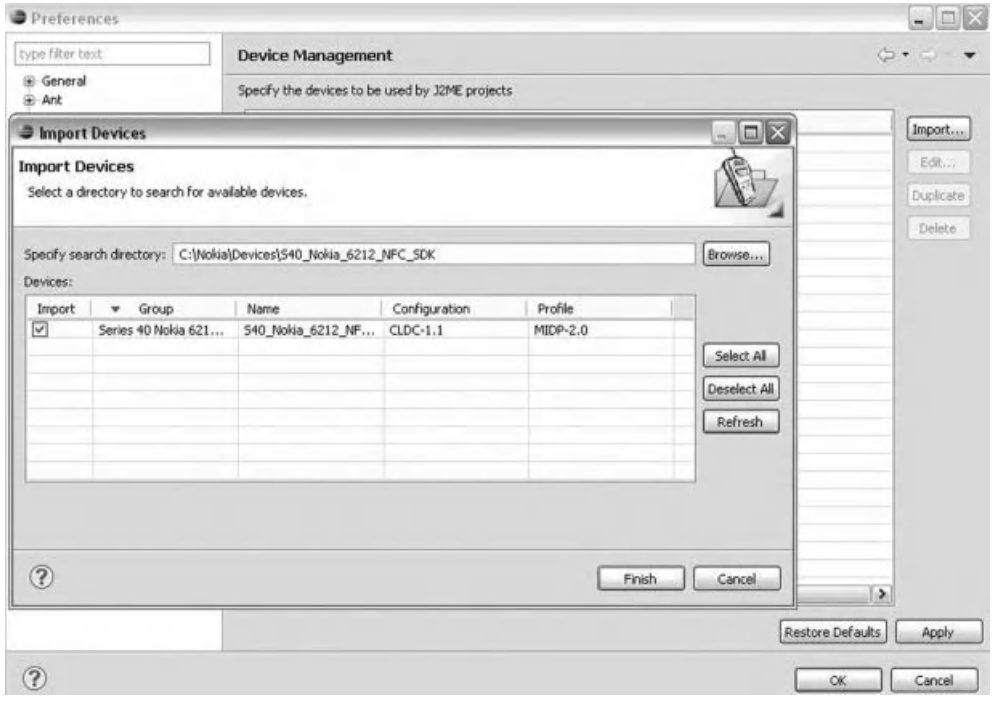


Figure 5.6 SDK installation screen-5.

5.4.1 Java ME Building Blocks

Java ME technology is based on configurations, profiles, and optional packages (see Figure 5.7) [5]:

- A *configuration* contains a base set of APIs and virtual machine capabilities that can be used with a device that has a certain hardware capacity.

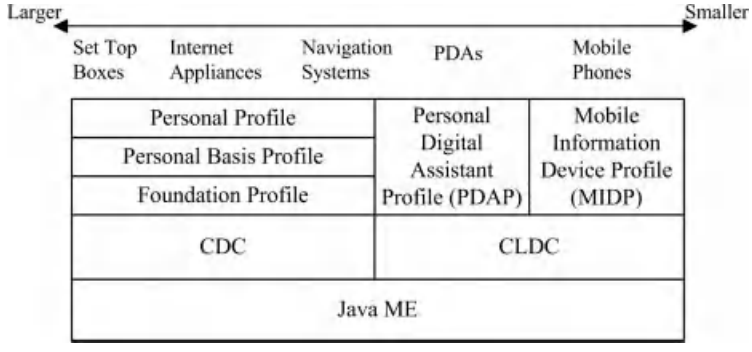


Figure 5.7 Java ME configurations and profiles.

Table 5.1 CLDC and CDC configurations

JSR No.	JSR Name	URL
30	Connected Limited Device Configuration (CLDC) 1.0	http://jcp.org/jsr/detail/30.jsp
139	Connected Limited Device Configuration (CLDC) 1.1	http://jcp.org/jsr/detail/139.jsp
36	Connected Device Configuration (CDC) 1.0.1	http://jcp.org/jsr/detail/36.jsp
218	Connected Device Configuration (CDC) 1.1	http://jcp.org/jsr/detail/36.jsp

- *Profiles* add more specific APIs than configurations. A profile is a set of APIs that support a narrower range of devices.
- An *optional package* provides functionality which is not included in a specific configuration or profile.

A configuration defines the basic Java ME runtime environment which includes:

- A virtual machine (which is more limited than J2SE VM);
- A set of core classes (which are derived primarily from J2SE classes).

Each configuration is designed for a specific family of devices with similar capabilities. The Java ME platform has been further divided into two configurations, one for relatively less capable mobile devices and the other for more capable mobile devices such as smart phones. The configuration for small devices like mobile phones is called the Connected Limited Device Configuration (CLDC) and the configuration for more capable devices is called the Connected Device Configuration (CDC). Some available configurations are given in Table 5.1.

The CLDC targets devices with limited resources such as limited processing power, memory, and graphics. On top of CLDC and CDC, different profiles are described which consist of several APIs and define complete application environments. The Mobile Information Device Profile (MIDP) is such a profile and is widely used in Java ME which is built upon CLDC. MIDP enables applications and services to be written for mobile devices including mobile phones. Those applications are downloadable and provide network connection capability.

A profile extends a configuration (which may be either CDC or CLDC as stated above), adding domain specific classes to provide functionality missing from the configuration used. Each device must support at least one configuration. Not every device supports every profile, and not every profile supports every device. Some available profiles are given in Table 5.2.

For MIDP development, which is a core development profile for mobile devices, initially a MIDlet needs to be created. MIDlets are also required in NFC application development since NFC applications run on mobile phones and are based on the MID profile. MIDlet examples are introduced and developed in the following sections.

5.4.2 MIDlets

In order to start programming MIDlets, we first need to be aware of the following definitions [11]:

Table 5.2 Java profiles

JSR No.	JSR Name	URL
37	Mobile Information Device Profile 1.0	http://jcp.org/jsr/detail/37.jsp
118	Mobile Information Device Profile 2.0	http://jcp.org/jsr/detail/118.jsp
271	Mobile Information Device Profile 3.0	http://jcp.org/jsr/detail/271.jsp
75	PDA Profile 1.0	http://jcp.org/jsr/detail/75.jsp
46	Foundation Profile 1.0	http://jcp.org/jsr/detail/46.jsp
219	Foundation Profile 1.1	http://jcp.org/jsr/detail/219.jsp
129	Personal Basis Profile 1.0	http://jcp.org/jsr/detail/129.jsp
217	Personal Basis Profile 1.1	http://jcp.org/jsr/detail/217.jsp
62	Personal Profile 1.0	http://jcp.org/jsr/detail/62.jsp

- *Application Management Software (AMS)* is part of the device's software operating environment that manages and controls the MIDlet. It retrieves the properties from the application descriptor and directs MIDlet through state changes.
- *MIDlet* is an MIDP application on the device which is defined earlier. It also sends a signal to the application management software about the process.
- *MIDlet states* are the states a MIDlet can have.

5.4.2.1 MIDlet States

In order to facilitate the MIDlet management, a MIDlet can pass into different states controlled by the AMS. Every MIDlet extends and overrides these states.

There are three valid MIDlet states (see Figure 5.8):

- *Paused state:* The MIDlet is paused in this state. It is not terminated; however, it should release the resources gained in the active state. A MIDlet is normally paused when the phone must execute a higher priority process such as an incoming call. A MIDlet can switch to the paused state by calling the `pauseApp` method when it is in the active state. Moreover

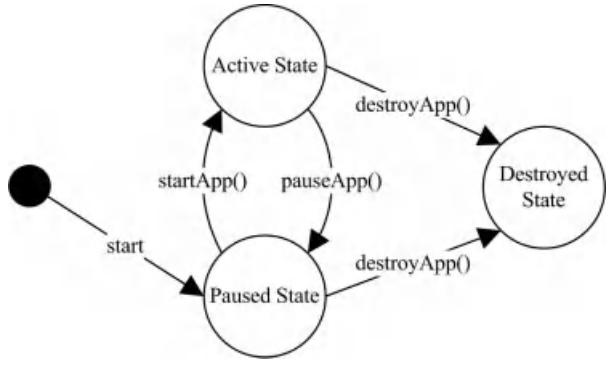


Figure 5.8 MIDlet states.

when a MIDlet starts its execution, it creates a new instance of the MIDlet, and the AMS places it in the paused state.
- *Active state:* The MIDlet is functioning normally in this state. A midlet can change its state to active by calling the `startApp` method if it is in the paused state.
- *Destroyed state:* The MIDlet is terminated in this state and it releases its resources. A MIDlet can switch to the destroyed state from either the paused or active state by calling the `destroyApp` method.

As we have described earlier, the application of the MIDP is a MIDlet. In Java ME, each MIDlet application must extend the abstract class `MIDlet` which resides in the `javax.microedition.midlet` package.

5.4.2.2 Class MIDlet

`MIDlet` class defines the MIDP applications and the interactions between the application and environment in which the application runs. The class resides in the `javax.microedition.midlet` package.

Applications must extend this class in order for AMS to control the MIDlet and to retrieve properties from the application descriptor, and request state changes to start, pause, or destroy a MIDlet.

A MIDlet must override the following three methods of the abstract MIDlet class; startApp, pauseApp and destroyApp. Each method enables a state change (see Figure 5.8).

- *`startApp()`:* This method signals the MIDlet that it has entered the active state. In the active state the MIDlet may hold some resources. The method can only be called when the MIDlet is in the paused state.
- *`pauseApp()`:* This method signals the MIDlet to enter the paused state. In the paused state the MIDlet must release shared resources. This method can only be called when the MIDlet is in the active state.
- *`destroyApp`(boolean unconditional):* This method signals the MIDlet to terminate the application. The MIDlet must release all resources and should save required data before terminating. This method can be called when the MIDlet is in either the paused or active state.

The boolean parameter in the `destroyApp` method is used in order to let the MIDlet request to continue if it does not want to be destroyed. If the value of the parameter is true, the MIDlet must release all resources that it holds. If the parameter is false, the MIDlet may continue to run by throwing an exception. One reason to request continuation for example is that the application is still saving data.

Other important methods in the `MIDlet` class are as follows:

- *notifyDestroyed():* The MIDlet sends notification to the AMS that it is switched to the destroyed state.
- *notifyPaused():* The MIDlet sends notification to the AMS that it is switched to the paused state.

- *resumeRequest():* The MIDlet asks the AMS to resume by starting again.
- *String getAppProperty(String key):* This method enables the application to receive the properties of the MIDlet from the application descriptor and the manifest file. The parameter, key, is the name of the property to be received.

5.4.3 Package javax.microedition.lcdui

`javax.microedition.lcdui` is an important package for MIDlet applications. This package provides implementation of the user interfaces. It is also an important package for NFC development, since user interfaces are required in NFC applications.

The main object of a user interface on a mobile phone is the screen. The screen emulates graphical objects and displays them to the user. Mobile applications can display one screen object at a time. When the current display is to be switched with another, the required arguments have to be passed to the next screen.

There are many useful classes defined in this package; some important classes are given below. A complete list of classes and their definitions can be accessed online via http://java.sun.com/javame.

5.4.3.1 Class Command

`Command` class represents a command that is to be executed in the application mostly by the user's action. Each command has a type that defines the intent of the command. As an example, the OK command directs the user to affirm a yes/no question, or submit an answer to a question. Available command types are listed in Table 5.3. Please note that actions of the commands are not automatically implemented and should be implemented in the `commandAction` method. Moreover, each command has a priority given by the user which describes the importance of the command. An application uses the priority of the commands in order to place them on the screen. Commands with a higher priority are mostly placed on the screen, thus the user

Table 5.3 Available Java ME command types

Command Type	Description
SCREEN	An unspecified command that applies to the screen's contents or to navigation among screens
BACK	Returns to the previous screen
CANCEL	Dismisses anything inputted to the current screen without taking any action. This command may also return to the previous or prior screen
OK	A confirmative answer to an object on the current screen
HELP	A help request by the user
STOP	A stop request to the current operation
EXIT	An exit request from the application
ITEM	Specific implementation to the items of the Screen or the elements of a Choice. May be used for creating context sensitive menu options

can directly choose to execute them; however commands with lower priority are placed on a menu which can be displayed after pressing the soft button and selected in the second action. In numbering the priorities, lower integers represent a higher importance and vice versa.

Each `Command` consists of a label, type, and priority. Type and priority are already described. Labels are displayed to the user to express the content of the command. A label should consist of only a few words since it needs to occupy a small area on the screen. A `Command` can be constructed as follows:

```
Command(String label, int commandType, int priority)
```

A sample command can be created as follows:

```
Command command = new Command("Back", Command.BACK, 0);
```

5.4.4 Creating a New MIDlet Project

In order to create a new mobile application using Eclipse IDE, click "New Project" on the "File" menu; after which the "New Project" wizard will be displayed. Select "J2ME Midlet Suite" and give a name for your project. If you are creating an NFC application, be sure that "S40 Nokia 6212 NFC SDK" is selected as the emulator device. After creating the project, create a new "J2ME Midlet" in your project.

(i) *Getting started with a simple MIDlet*

```
/*
 * A simple MIDlet that contains a Form and prints
 * a String to screen
 */
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class Hello extends MIDlet implements CommandListener
{
 private Form mainForm;
 private Command exitCommand;

 public Hello(){

  // instantiate the Form object
  mainForm = new Form("My MIDlet");

  // add a text message
  mainForm.append(new StringItem(null, "This is my first
     MIDlet"));

  // instantiate the exitCommand
  exitCommand=new Command("Exit", Command.EXIT, 0);

  // add exitCommand to the Form object
  mainForm.addCommand(exitCommand);
```

```
    /*
     * set up a command listener so that,
     * device listens for user commands
     */
    mainForm.setCommandListener(this);

  }

   public void startApp(){

    // display the Form on the screen
    Display.getDisplay(this).setCurrent(mainForm);
   }

   public void pauseApp(){

   }

   public void destroyApp(boolean unconditional){

    // notify on destroy operation
    notifyDestroyed();
   }

    public void commandAction(Command command, Displayable
       displayable){

      // if user hits the exitCommand
      if (command == exitCommand){
        destroyApp(true);
        // destroy the application
      }
    }
  } // end of MIDlet
```

(ii) *Running the MIDlet*

After implementing the MIDlet, right click on the project name in "Package Explorer", expand "J2ME" and click "Create Package". Then right click on the MIDlet and click "Run As - Emulated J2ME Midlet".

In this first example, a simple MIDlet is created which displays a Form object on the screen and gets an "Exit" command from the user. MIDlet's screenshot can be seen in Figure 5.9.

(iii) *Explanation of the code*

Two classes are imported to the MIDlet:

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
```

`javax.microedition.lcdui` is required to create screen objects, displaying them and getting commands from the user. `javax.microedition.midlet` is required

Figure 5.9 Hello MIDlet screen.

for all MIDlets, since it allows the application manager to control the MIDlet. Moreover, three methods to be implemented in this abstract class are; `startApp`, `pauseApp`, and `destroyApp`.

MIDlet implements the `CommandListener` interface from the `lcdui` package:

```
public class Hello extends MIDlet implements CommandListener { }
```

The `CommandListener` interface enables the application to listen to the commands from the user. When a command event occurs, it automatically calls `commandAction` method.

A Form object named `mainForm` and a Command object named `exitCommand` are created:

```
private Form mainForm;
private Command exitCommand;
```

A Form may contain either a single display item or a combination of different display items such as string items, text fields, and other display objects. When the user selects the `exitCommand`, MIDlet is destroyed, hence the application quits.

As Java programmers will remember, the constructor is executed when the corresponding object of that class is created. However, the application content will not be displayed on the mobile until the `startApp` method is invoked by the application manager. In the constructor part, as seen in the code below, initially a Form object is instantiated and a string item is added onto the form. Then a command is instantiated and added to the `mainForm` by the `addCommand` method. This method simply adds a command object to the Form object. Lastly, the `setCommandListener` method is implemented to the form object. The listener enables the form to listen to commands from the user. It allows the `commandAction` method to be called automatically, as soon as a user command is received.

```
public Hello() {
  mainForm = new Form("My MIDlet");
  mainForm.append("This is my first MIDlet");
  exitCommand = new Command("Exit", Command.EXIT, 0);
  mainForm.addCommand(exitCommand);
  mainForm.setCommandListener(this);
}
```

Then, the `startApp` method is implemented and the screen is set to display the Form object by the following line of code:

```
Display.getDisplay(this).setCurrent(mainForm);
```

`getDisplay` is a static method of the `Display` class, hence it can be invoked by using the name of the class, that is, Display, instead of an object reference name as the prefix. Calling the `setCurrent(mainForm)` sets `mainForm` as the current form to be displayed.

When the application is destroyed, the application manager is notified on the destroy operation by the `notifyDestroyed` method as seen in the following lines of code:

```
public void destroyApp(boolean unconditional) {
  notifyDestroyed();
}
```

The `commandAction` method is invoked by the application automatically when a command is received from the user:

```
public void commandAction(Command command, Displayable
displayable){ }
```

This method gets two arguments. `Command` gets the user's request, so that the application determines which operation to perform. In the following lines of code, the user's command is checked if the `exitCommand` is received. The application is destroyed and the application manager is notified about the destroy operation.

```
if (command == exitCommand){
  destroyApp(true);
  notifyDestroyed();
}
```

On the other hand, `displayable` gets the reference to the currently displayed object on the screen. If the same command is defined in several screen objects in the application, it can be handled within the `commandAction` method by using the `displayable` parameter.

5.4.5 Inside a MIDlet Suite (MIDlet Packaging)

In order to deploy a MIDlet to a mobile device, a MIDlet suite needs to be created that consists of:

- Java Archive File (JAR) containing:
  - Java classes for the MIDlets;
  - Resource files (images, etc.) used by the MIDlets;
  - A manifest file which describes the contents of the JAR file.
- Java Application Descriptor (JAD) File.

A JAR file primarily consists of the java classes that are needed in the MIDlets. JAR files are generally automatically created by the used IDE; hence user action is not required.

(i) *JAR manifest file*

A JAR manifest file defines the attributes which are used by AMS to identify the properties of the MIDlets. The JAD file may also include the same attributes. In this case, the attributes in the JAD file are used. Attributes in the JAR manifest file are only used when those attributes are not included in the JAD file. An attribute in the JAR manifest file must not appear more than once.

Most commonly used attributes for the JAD and JAR manifest files are as follows:

- *MIDlet-Name:* Name of the MIDlet suite.
- *MIDlet-Version:* Version number of the MIDlet suite. Version numbers need to be formatted as X.X[.X]. The [.X] part is not mandatory, and if it is left blank, it is set to zero as ".0" automatically. Each part of the version number may have a maximum of two digits from 0 to 99. If a MIDlet-Version tag is left blank, it is considered to be a newer version of the MIDlet suite.
- *MIDlet-Vendor:* Name of the organization that provides the MIDlet suite.
- *MIDlet-Icon:* An icon that represents the MIDlet suite. It should be a case-sensitive absolute name of a PNG image file in the JAR file.
- *MIDlet-Description:* Description of the MIDlet suite.
- *MIDlet-Info-URL:* A URL for further describing the MIDlet suite.
- *MIDlet-<n>:* Definition of the MIDlets in the suite. Three entries need to be included and to be separated by commas which are MIDlet name, MIDlet icon (if any), and MIDlet class. Number n starts from 1 and for each MIDlet it increases consecutively.
- *MIDlet-Jar-URL:* URL of the JAR file. Absolute and relative URLs can be used. The relative URL starts from the folder containing the application descriptor.
- *MIDlet-Jar-Size:* Size of the JAR file in bytes.
- *MIDlet-Data-Size:* Minimum number of bytes of persistent data required by the MIDlet.
- *MicroEdition-Profile:* J2ME profile (for example “MIDP-2.0”).
- *MicroEdition-Configuration:* J2ME Configuration (for example “CLDC-1.0”).

There are additional attributes as well. For details on all the attributes and additional information, please visit http://java.sun.com/javame.

There are three attributes that must be included in the JAR manifest file:

- MIDlet-Name
- MIDlet-Version
- MIDlet-Vendor

The following three attributes must be included in either the JAR manifest or the JAD file:

- MIDlet-<n> (for each MIDlet)
- MicroEdition-Profile

Table 5.4 Mandatory attributes for the JAD and JAR manifest files

Attribute Name	JAR Manifest	JAD	Either One of the Files
MIDlet-Name	Mandatory	Mandatory	
MIDlet-Version	Mandatory	Mandatory	
MIDlet-Vendor	Mandatory	Mandatory	
MIDlet-<n>			Mandatory
MIDlet-Jar-URL	Mandatory		
MIDlet-Jar-Size	Mandatory		
MicroEdition-Profile			Mandatory
MicroEdition-Configuration			Mandatory

- MicroEdition-Configuration

The manifest file may contain other attributes as well, but they are not mandatory.

(ii) *JAD file*

The JAD file is used by the MIDlet for configuration specific attributes. AMS manages MIDlets by using the JAD file together with the JAR manifest file. It also allows attributes to be supplied to the MIDlets without modifying the JAR file. The file extension of a JAD file is "jad".

The following five attributes must be included in the JAD file, however the JAD file may contain other optional attributes as well:

- MIDlet-Name
- MIDlet-Version
- MIDlet-Vendor
- MIDlet-Jar-URL
- MIDlet-Jar-Size

As can be seen from Table 5.4, there are three common mandatory attributes (MIDlet-Name, MIDlet-Version, MIDlet-Vendor) for both files. These attributes uniquely identify the application and must be duplicated in both files. If the values of these attributes are not the same in both files, then the JAR file cannot be installed. Other attributes are not required to be duplicated in both files. However if there is a duplicated one, remember again that the one that is in the JAD file will be used and the other one can be ignored.

Remember that the `getAppProperty` method receives the application properties from the JAD and JAR manifest files. This method first searches the JAD file. If it cannot find the specified attribute in this file, then it searches the JAR manifest file.

An example JAD file content is given as follows:

```
MIDlet-Name: Hello Midlet
MIDlet-Version: 1.0.0
MIDlet-Vendor: My Vendor Name
MIDlet-1: Hello,,Hello
MicroEdition-Profile: MIDP-2.0
MicroEdition-Configuration: CLDC-1.1
MIDlet-Jar-URL: Hello.jar
MIDlet-Jar-Size: 1219
```

Figure 5.10 UIMIDlet MIDlet screens.

5.4.6 A More Detailed User Interface MIDlet

We now present a MIDlet that is a more detailed example that shows how to use text boxes, lists and text fields. As seen in application screenshots (Figure 5.10), this application demonstrates the use of some user interfaces such as list menus, text boxes, text fields, and also command listeners. The upper-left screen in Figure 5.10 shows the main screen of the MIDlet. It consists of an implicit list menu which forwards the user to different menus by selections. "Form" selection displays to the user a form which consists of text fields. "TextBox" selection displays to the user a text box and finally "List" selection displays to the user a multiple list menu. Actions of the commands are only applied for displaying menus; further implementation is not made, since it is not in the scope of this MIDlet. Various actions will be given in Section 5.5.

(i) *Application source code*

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class UIMIDlet extends MIDlet implements
   CommandListener {
```

```
// create a display object
private Display screen;

/*
 * create two lists, one for main menu and one
 * for multiple choice menu
 */
private List listMenu, listMultiple;

// create a text box
private TextBox bodyText;

// create a form
private Form mainForm;

// create text fields
private TextField toField, subjectField,
   ccField, bccField;
/*
 * create commands for going back to main menu
 * and exiting MIDlet
 */

private Command backCommand = new
   Command("Back", Command.BACK, 0);
private Command exitCommand = new
   Command("Exit", Command.EXIT, 1);

public UIMIDlet(){

  // create a list for displaying on main menu
  listMenu = new List("My List",Choice.IMPLICIT);
  listMenu.append("Simple Form", null);
  listMenu.append("Simple TextBox", null);
  listMenu.append("Simple List", null);
  listMenu.addCommand(exitCommand);
  listMenu.setCommandListener(this);
}

public void startApp(){

  screen = Display.getDisplay(this);
  mainMenu();
}

public void pauseApp(){
}

public void destroyApp(boolean unconditional) {
```

```
      notifyDestroyed();
    }

    void mainMenu() {
      // set the current screen to the listMenu list
      screen.setCurrent(listMenu);
    }

    public void commandAction(Command command,
      Displayable displayable) {

      String label = command.getLabel();

      // destroy, if user hits Exit command
      if (label.equals("Exit")) {
        destroyApp(true);
      } else if (label.equals("Back")) {

      // show main menu if user hits Back command
      mainMenu();
      } else {

        /*
         *if nothing above happened, then user
         *chosen an item on main screen
         */
        switch(((List) screen.getCurrent())
           .getSelectedIndex()){

          case 0:
            emailHeader();
            break;
          case 1:
            emailBody();
            break;
          case 2:
            emailOptions();
            break;
       }
     }
  }

   /*
    * create a Form for composing the basic
    * header information of the e-mail and
    * display it
    */
   public void emailHeader() {

     mainForm = new Form("E-mail Header");
```

```
  toField = new TextField("To:", "", 32,
     TextField.EMAILADDR);

  subjectField = new TextField("Subject:", "",
     32, TextField.ANY);

  ccField = new TextField("Cc:", "", 32,
     TextField.EMAILADDR);

  bccField = new TextField("Bcc:", "", 32,
     TextField.EMAILADDR);
  mainForm.append(toField);
  mainForm.append(subjectField);
  mainForm.append(ccField);
  mainForm.append(bccField);
  mainForm.addCommand(backCommand);
  mainForm.addCommand(exitCommand);
  mainForm.setCommandListener(this);
  screen.setCurrent(mainForm);
}

/*
 *create a text box for composing the body of
 *the e-mail and display it
 */
public void emailBody() {

  bodyText = new TextBox("E-mail Body:", "",
     255, TextField.ANY);
  bodyText.addCommand(backCommand);
  bodyText.addCommand(exitCommand);
  bodyText.setCommandListener(this);
  screen.setCurrent(bodyText);
}

/*
 * create a multiple choice list for
 * selecting e-mail options and display it
 */
public void emailOptions() {

  listMultiple = new List("Choose E-mail
     Options", Choice.MULTIPLE);
  listMultiple.append("Request Read Receipt",
     null);
  listMultiple.append("Request Delivery
     Receipt", null);
  listMultiple.append("Save to Sent Items",
     null);
```

```
    listMultiple.addCommand(backCommand);
    listMultiple.addCommand(exitCommand);
    listMultiple.setCommandListener(this);
    screen.setCurrent(listMultiple);
   }
}
```

(ii) *Explanation of the code*

At the very beginning of the application, the following code lines create the corresponding user interface objects:

```
private Display screen;
private List listMenu, listMultiple;
private TextBox bodyText;
private Form mainForm;
private TextField toField,subjectField,ccField,bccField;
private Command backCommand = new Command("Back", Command.BACK,
0);
private Command exitCommand = new Command("Exit", Command.EXIT,
1);
```

These interface objects include:

- A *Display* object to handle the display of the screen.
- Two *List* objects which display lists on two different screens.
- A *TextBox* object to display a textbox to the user.
- A *Form* object which will include TextField objects and Commands.
- *TextField* objects to get input from the user inside the Form object.
- Two *commands* to get the user's command; one for going back to the main menu and one for exiting the application.

In the constructor part, `listMenu` List object is initiated with an IMPLICIT choice type. Three elements are added to the list and the user can choose one of them. Each element is given an index value automatically starting from 0. When the user selects an element, the `commandAction` method is invoked. There is no need to add an additional select command since the select key on mobile devices will be automatically set to select the element.

```
public UIMIDlet() {
  listMenu = new List("My List", Choice.IMPLICIT);
  listMenu.append("Simple Form", null);
  listMenu.append("Simple TextBox", null);
  listMenu.append("Simple List", null);
  listMenu.addCommand(exitCommand);
  listMenu.setCommandListener(this);
}
```

`List` is another class in the `javax.microedition.lcdui` package. When a `List` object is displayed, the user can interact with it by selecting one available option. A *select* operation on a list element differs in the various devices. In devices that have a dedicated select key, the select operation is implemented with that key.

Table 5.5 Available Java ME List types

List Type	Description
EXCLUSIVE	Only one element can be selected at a time
MULTIPLE	Arbitrary number of elements can be selected at a time
IMPLICIT	Currently focused element is selected when a Command is given

List objects may be created with Choice types of EXCLUSIVE, MULTIPLE, and IMPLICIT (see Table 5.5).

Inside the `startApp` method, the `mainMenu` method is called which will set the current screen to `listMenu` list object.

```
public void startApp(){
  screen = Display.getDisplay(this);
  mainMenu();
}

void mainMenu() {
  screen.setCurrent(listMenu);
}
```

`listMenu` object is the list that is displayed on the main screen. Since the main menu is called from many parts of the application, a specific method is created for displaying the main menu and it is called whenever the main menu needs to be displayed.

As we have seen earlier, when the user hits a button, the `commandAction` method is called. In this example, there are three options that a user can choose:

1. Exit the application.
2. Go back (to the main menu).
3. Select a choice from main menu.

In the `commandAction` method, the user's command is received by storing the command button's label to a string. Remember that a button is created with its label. `exitCommand's` label is set to "Exit" and `backCommand's` label is set to "Back". The application exits when the "label" string equals "Exit" and displays the main menu when the "label" string equals "Back".

```
public void commandAction(Command command, Displayable
displayable) {
  String label = command.getLabel();
  if (label.equals("Exit"))
  {
    destroyApp(true);
  }
  else if (label.equals("Back"))
  {
    mainMenu();
  }
  else
  {
    switch(((List) screen.getCurrent()).getSelectedIndex())
```

```
      {
        case 0: emailHeader();  break;
        case 1: emailBody();    break;
        case 2: emailOptions(); break;
      }
    }
  }
```

Finally if the selected command's label is not equal to "Back" or "Exit", then it means that the user selected an option from the main menu. The user's choice is obtained by the current screen's selected list index and the corresponding method is called by the MIDlet.

In the `emailHeader` method, a new form is created, and text fields related to the e-mail header and commands are added to the form.

In the `emailBody` method, a text box is created for the e-mail body, and commands are added to this display object.

In the `emailOptions` method, a new multiple choice list is created and list items are added to the list. Two commands are also added for going back to the main menu and exiting the application.

5.4.7 Push Registry

`PushRegistry` is a class defined in the `javax.microedition.io` package. Normally, MIDlets can be executed when the user triggers it from the mobile device's menu. As an alternative, a MIDlet may be invoked with the "Push" feature, without any user interaction. For example, an application can be triggered by a network package received by an incoming Short Messaging Service (SMS).

PushRegistry simply maintains a list of inbound connections and maps network connection strings to class names. AMS listens for inbound network activity and notifications. When a notification arrives at the AMS for a registered MIDlet, AMS runs the MIDlet via the `startApp` method.

PushRegistry can be implemented in two ways:

- *Static registration:* An application can register a push record with an entry in the JAD file.
- *Dynamic registration:* An application can register a push record dynamically by calling the `registerConnection` method in a MIDlet.

(i) *Static registration*

In order to declare a push connection statically, the MIDlet-Push-<n> attribute must be specified in the JAD file or the JAR manifest file.

```
MIDlet-Push-<n>: <ConnectionURL>, <MIDletClassName>,
<AllowedSender>
```

- *MIDlet-Push-<n>:* The attribute name for Push registration. <n> starts from 1 and increases consecutively.
- *ConnectionURL:* The connection string used in the `Connector.open` method. When this connection occurs, MIDlet will be pushed.

- *MIDletClassName:* The MIDlet that will be pushed. MIDlet should also be registered in a descriptor or manifest file. The required registration should be made via the MIDlet -<n> attribute.
- *AllowedSender:* A filter that enables or restricts senders pushing the MIDlet. Entries in this field should match the required addressing format. This field may contain:
  – Any IPv4 and IPv6 IP addresses within square brackets
  – "*" character to match any source
  – "?" character to match any single character

Wildcards in the `AllowedSender` part can be used as for example "193.255.146.*" to match a subnetwork. Port numbers cannot be filtered in this attribute.

If two MIDlet suites have the same static push connection and one of them is already installed in the mobile device, the user should uninstall the first one in order to install the second suite successfully, because more than one MIDlet cannot register the same push connection.

A sample static registration for push registry entry should look like the following:

```
MIDlet-1: MyGame, , example.games.MyGame
MIDlet-Push-1: datagram://:50000,  example.games.MyGame, *
```

(ii) *Dynamic registration*

As for static registration, three elements need to be set in order to implement a push registry entry. These elements should be set as parameters in the `PushRegistry.registerConnection` method.

The following example dynamically registers a push registry connection the same as for static registration:

```
PushRegistry.registerConnection("datagram://:50000", "
example.games.MyGame ", *)
```

In order to remove a dynamic push registry entry, either the related MIDlet suite should be removed or the entry should be unregistered with the `PushRegistry.unregisterConnection` method in the `PushRegistry` class. For further details of push registry please visit the related web page in http://java.sun.com/javame/.

Dynamic and static registrations have their own advantages and disadvantages. Static registration enables a push connection to register easily without user interaction. However, if there is a conflicting push registry entry in the device, the application cannot be installed. On the other hand, dynamic registration eliminates this issue and the application is installed independently from push registry entries in the mobile phone. However in dynamic registration, push record is saved in application's first run.

In this section, we have given a short introduction to Java ME. For more information, readers are encouraged to read specific books on Java ME and to use web resources such as:

- *General information on the Java ME*
  http://java.sun.com/javame/
- *Connected Limited Device Configuration*
  http://java.sun.com/products/cldc/
- *Mobile Information Device Profile*

http://java.sun.com/products/midp/
- *Java ME technical documentation*
  http://java.sun.com/j2me/docs/
- *Java developers page*
  http://developer.java.sun.com

5.5 NFC Application Development

In this section we start to talk about NFC programming. Two APIs, namely JSR 257 and JSR 177, are required to developed NFC applications. JSR 257 provides reader/writer mode application programming resources, whereas JSR 177 and some classes in JSR 257 provide access to SEs. For peer-to-peer mode programming, propriety APIs are required, since this mode is not supported by the standard Java APIs.

- *JSR 257 (Contactless Communication API):* This API describes contactless communication and the push registry in NFC. It provides an application programming interface that allows applications to access RFID tags, smart cards, and visual tags (barcodes). Different packages are defined for each target type and an application using this API can discover contactless targets and then can connect to the target by using the related package based on the target type [12].
- *JSR 177 (Security and Trust Services API, SATSA):* This API provides access to smart cards and provides security operations targeting SEs using APDU (Application Protocol Data Unit) and JavaCard RMI (Remote Method Invocation) protocols [13].

JSR 177 vs. JSR 257 (difference on accessing smart cards):

- Both allow applications to access smart cards.
- JSR257 allows access to a smart card with ISO14443Connection.
- JSR177 allows access to a smart card using an `APDUConnection` or `JavacardRMI-Connection`.

5.6 Reader/Writer Mode Programing

Contactless Communication API (JSR 257) [12] provides an application programming interface that allows applications to access RFID tags, smart cards, and visual tags. An application using this API can discover contactless targets and then connect to the target by using the related package based on the target type.

In reader/writer mode case, JSR 257 enables discovering contactless RFID tags and communicating them with NFC enabled mobile phones. Packages related to reader/writer mode are covered in this section. As an example, a user can open a web page by touching her mobile device to a tag containing the URL of the page on the smart poster. The application will first discover the target, then it will connect to the target with the related package, and it will read the contents of the target. Finally, the application may behave as it is programmed.

In order to communicate with a target, the mobile device first needs to register listeners for a specific target type (e.g., NFC Forum formatted tag types) using an instance of the `DiscoveryManager` class and `addTargetListener` method. When a specified target is discovered in the proximity, notification arrives at the `targetDetected` method from

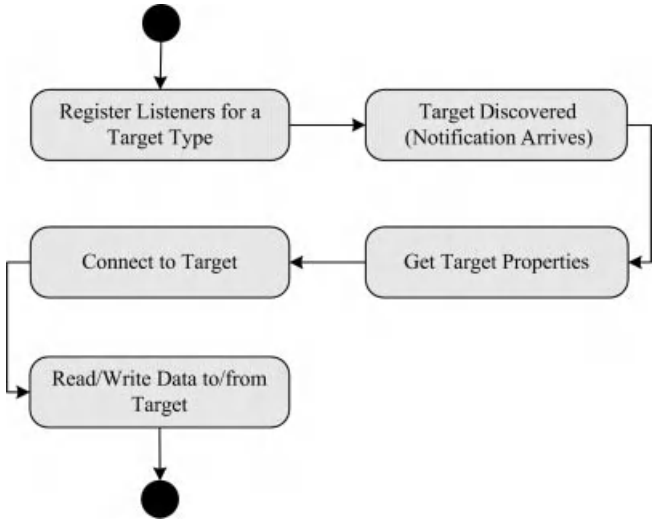


Figure 5.11 Process flow of contactless communication.

the `TargetListener` interface. The properties of the device, which are needed in order to establish the connection, are also sent to the `targetDetected` method. After making the connection to the target, the device may perform read and write operations on the target. The processes are depicted in Figure 5.11.

There are five packages defined in JSR 257 API. In order to work with this API, CLDC version 1.1 is required.

It can be seen from Table 5.6 that the packages are distinguished by different target types except `javax.microedition.contactless` package. This package is included in all

Table 5.6 JSR 257 packages

Package	Description	Supported Target
javax.microedition.contactless	Provides common functionalities to all contactless targets such as discovering the target	Common
javax.microedition.contactless.ndef	Provides functionality to exchange NFC Forum formatted data with RFID tags	NDEF_TAG
javax.microedition.contactless.rf	Enables communication with RFID tags that contain non-NFC Forum formatted data	RFID_TAG
javax.microedition.contactless.sc	Enables communication with external smart cards	ISO14443_CARD
javax.microedition.contactless.visual	Provides functionality to read data on visual tags and generate visual tag images	VISUAL_TAG

applications that implement Contactless Communication API irrespective of the target device. Additionally, one of the other packages is used in the application according to the target type.

NDEF_TAG target type defines a tag that contains NFC Forum formatted data. RFID_TAG is a general RFID tag that contains data in a proprietary format. ISO14443_CARD defines ISO14443-4 compliant smart cards that can be accessed with APDU commands defined in ISO7816-4. VISUAL_TAG, on the other hand, is a general visual tag.

In the following sections the common package for all target types, `javax.microedition.contactless`, is described first. Then the other two packages for NDEF_TAG and RFID_TAG types are described and code examples are given.

5.6.1 Package javax.microedition.contactless

This package includes common classes and interfaces that can be used along with all target types. It is also the starting point of JSR 257 API. The main function of the package is to discover contactless targets in proximity. The application can receive notifications when new targets are discovered in proximity and can make a target specific connection which is defined in target specific packages.

The package consists of three classes, one of which is an exception class, and three interfaces (see Table 5.7).

(i) *Class DiscoveryManager*

`DiscoveryManager` class is the heart of this package. It enables discovering targets at proximity, managing different listeners, and much more. It also returns the list of target types supported by the API implementation using the `getSupportedTargetTypes` method. Possible target types will be given in the `TargetType` class.

The `addTargetListener` method is used to add a listener for the specified target type to receive notification when the target is available within the proximity. Please note that, for each target type, only one listener can be registered. If another listener tries to be registered for a target type, `IllegalStateException` will be thrown. The `removeTargetListener` will remove the specified listener. The `getProperty` method receives the properties of the target after discovery.

(ii) *Class TargetType*

Another class in the `javax.microedition.contactless` package is `TargetType` which collects the supported target types. Available connection interfaces to the physical target can be gathered using the `getConnectionNames` method in the `TargetProperties` interface.

Table 5.7 javax.microedition.contactless package

Classes	DiscoveryManager TargetType
Interfaces	TagConnection TargetListener TargetProperties
Exception	ContactlessException

Table 5.8 Supported target types in JSR 257

Target Type	Description	Communication Interface
NDEF_TAG	NFC Forum formatted data containing tag	`NDEFTagConnection` from `javax.microedition.contactless.ndef` package
RFID_TAG	General RFID tag	`PlainTagConnection` from `javax.microedition.contactless.rf` package
ISO14443_CARD	ISO14443-4 compliant smart cards	`ISO14443Connection` from `javax.microedition.contactless.sc` package
VISUAL_TAG	General visual tag	`VisualTagConnection` from `javax.microedition.contactless.visual` package

Currently, four different target types are implemented in this API as shown in Table 5.8. Please note that there may be additional target type implementations in the future.

(iii) *Interface TagConnection*

`TagConnection` is a marker interface in this API which provides RFID tag connections, NDEF tag connections, and smart card connections. Please note that a marker interface does not actually define any components such as attributes or methods. Instead, it is used to mark classes. Connections other than visual tags need to extend this interface. The following three interfaces inherit the `TagConnection` interface:

- `NDEFTagConnection` for NDEF_TAG target types
- `PlainTagConnection` for RFID_TAG target types
- `ISO14443Connection` for ISO14443_CARD target types

(iv) *Interface TargetListener*

`TargetListener` is an important interface that all applications must implement in order to receive notification when a contactless target is discovered by a mobile device. The `targetDetected` method in this interface is called automatically when the contactless target is discovered.

If the application sets a listener using the `addTargetListener` method, it will receive a notification when the specified target is detected and it will receive all the information of the target.

(v) *Interface TargetProperties*

Using the `TargetProperties` interface, information on the detected target can be collected. The `getConnectionNames` method returns the name of the interface (e.g., NDEFTagConnection) which is needed to communicate with the target. The `getProperty` method enables the application to query specific properties of the target. The `getUid` method returns the unique identifier of the target. The `getUrl` method returns the URL in order to create a connection to a target. Finally, `hasTargetType` checks whether the discovered target is a specified target type.

(vi) *Interface TransactionListener*

This interface is described in Section 5.8.

(vii) *Exception ContactlessException*

This exception is thrown when an unsupported operation is attempted. Reasons to throw this exception are:

- Invoking unsupported method by the API;
- Invoking unsupported method by the target;
- An unsupported usage of the API.

The exception can be constructed with or without an exception detail message:

- `public ContactlessException()`
- `public ContactlessException(java.lang.String message)`

5.6.2 Package javax.microedition.contactless.ndef

This package provides functionality to exchange NDEF formatted data with contactless targets. In order to work with this package, the target needs to include NDEF formatted data. Moreover, the developer does not need to know the physical type of the target to exchange NDEF messages and records.

In order to receive a notification when a target is discovered with NDEF formatted data, `NDEFRecordListener` should be set by the `addNDEFRecordListener` method which resides in the `DiscoveryManager` class.

Classes and interfaces that reside in this package are given in Table 5.9.

(i) *Class NDEFMessage*

This class typically represents an NDEF message. An NDEF message consists of NDEF record(s) and the manipulation of the records (read, add, remove, update) is provided with this class. `getRecord` and `getRecords` methods allow record(s) to be read from the message. The `getNumberOfRecords` method returns the number of records in the message. The `insertRecord` method allows a record to be added to the message. The `removeRecord` method allows a record to be removed with the specified index.

Three different constructors can be used for this class:

- `public NDEFMessage ()`
- `public NDEFMessage (NDEFRecord [] records)`
- `public NDEFMessage (byte[] data, int offset)`

(ii) *Class NDEFRecord*

The `NDEFRecord` class represents an NDEF record and allows NDEF record operations. The `getId` method returns the record payload identifier. The `getPayload` method returns the record payload as a byte array. The `getRecordType` method returns

Table 5.9 javax.microedition.contactless.ndef package

Classes	NDEFMessage NDEFRecord NDEFRecordType
Interfaces	NDEFRecordListener NDEFTagConnection

the type of the record. The `appendPayload` method appends the specified payload to the end of the record payload.

There are two different constructors in this class:

- `public NDEFRecord (NDEFRecordType recordType, byte[] id, byte[] payload)`
- `public NDEFRecord (byte[] data, int offset)`

(iii) *Class NDEFRecordType*

The `NDEFRecordType` class represents the type of a record by storing its name and format. The record type name format, as its name suggests, specifies the structure and format of a record type name.

The `getFormat` method returns the format of the NDEF record type name. There are six possible record type name formats (see below). Another method in this class, `equals`, compares two record type names. The `getName` method returns the name of an NDEF record without any prefixes. Normally, NFC Forum RTD names and external record type names include prefixes (*urn:nfc:wkt:* and *urn:nfc:ext:*). However, the `getName` method does not return these prefixes.

- EMPTY
  Empty records are identified by this identifier. The EMPTY format record type's name can only be valid if it is null. Two EMPTY record type names are equal if both are null, otherwise they are not. The value of this record type name is 0.
- NFC_FORUM_RTD
  This is the record type name format identifier for NFC Forum RTDs such as NFC Text RTDs, NFC URI RTDs, and NFC Smart Poster RTDs. Two NFC_FORUM_RTD record type names are equal if all characters are equal and the comparison is case sensitive. The value of this record type name is 1.
- MIME
  This is the record type name format identifier for MIME types. Two MIME record type names are equal if all characters are equal in a case insensitive manner up to the first ";" character. The value of this record type name is 2.
- URI
  This is the record type name format identifier for URI types. Two URI record type names are equal if all characters are equal. Note that the comparison is case sensitive. Reserved characters must be encoded using percent-encoding. Hexadecimal digits should be upper case and host parts should be lower case. The value of this record type name is 3.
- EXTERNAL_RTD
  This is the specific record type name format identifier for NFC Forum external type names. Two EXTERNAL_RTD record type names are equal if all characters are equal. Comparison is case insensitive. The value of this record type name is 4.
- UNKNOWN
  If the type of payload is unknown, the record type name format identifier for this record is UNKNOWN. Two UNKNOWN record type names are equal if both are null, otherwise they are not equal. The value of this record type name is 5.

(iv) *Interface NDEFRecordListener*

This interface provides notification to the application when NDEF records are discovered in targets. However, the application can only read data (meaning that it has a read-only access) and cannot modify or delete it.

The `recordDetected` method is defined in this interface which is invoked automatically by the platform to notify the device that an NDEF record type is discovered from the target. The application performs the required actions and finally provides the data with a parameter of NDEFMessage.

(v) *Interface NDEFTagConnection*

The `NDEFTagConnection` interface inherits `TagConnection` interface in the `javax.microedition.contactless` package as described earlier.

This interface enables data to be exchanged with tags and contactless smart cards. The data are NFC forum formatted and need to be stored in an `NDEFMessage` object which consists of `NDEFRecords`. The application can read/write data from/to contactless target. It does not need to know the type of the physical target, but only needs to know if the target type consists of NFC Forum formatted data.

In order to open a connection to the contactless target, the target first needs to be discovered and notified using `targetDetected` notification from the `TargetListener` interface. After that, the URI that is used to open the connection can be obtained using the `TargetProperties` interface.

5.6.3 Package javax.microedition.contactless.rf

This package provides the `PlainTagConnection` interface for accessing RFID TAG targets. As described earlier, the `PlainTagConnection` interface inherits the `TagConnection` interface. It can be used with RFID targets that do not contain NFC Forum formatted data.

5.6.4 Package javax.microedition.contactless.sc

This package provides the `ISO14443Connection` interface for accessing ISO 14443-4 compliant contactless smart card targets. Remember that the ISO/IEC 14443 contactless proximity interface has already been described in Chapter 3. The `ISO14443Connection` interface inherits the `TagConnection` interface. The communication is based on APDU commands after target discovery. The slot number needed to open connection to the external smart card can be obtained using the `getProperty` method from the `TargetProperties` interface.

Up to now, we have described the reader/writer mode related parts of Contactless Communication API (JSR 257). There is another package in this API which is `javax.microedition.contactless.visual`. This package provides communication with visual tags (barcodes) but is outside the scope of this book. The card emulation mode related parts of this API are described in Section 5.8.

We have given only a short introduction of this API in this book. Please refer to Contactless Communication API in order to get a full specification of classes, interfaces and methods.

5.6.5 A Reader/Writer Mode Application

We provide an example that demonstrates how to read data records from NDEF formatted tags and how to write NDEF formatted data records to tags. This application includes two additional classes: a *Read* class to read from tags and a *Write* class to write data to tags. As

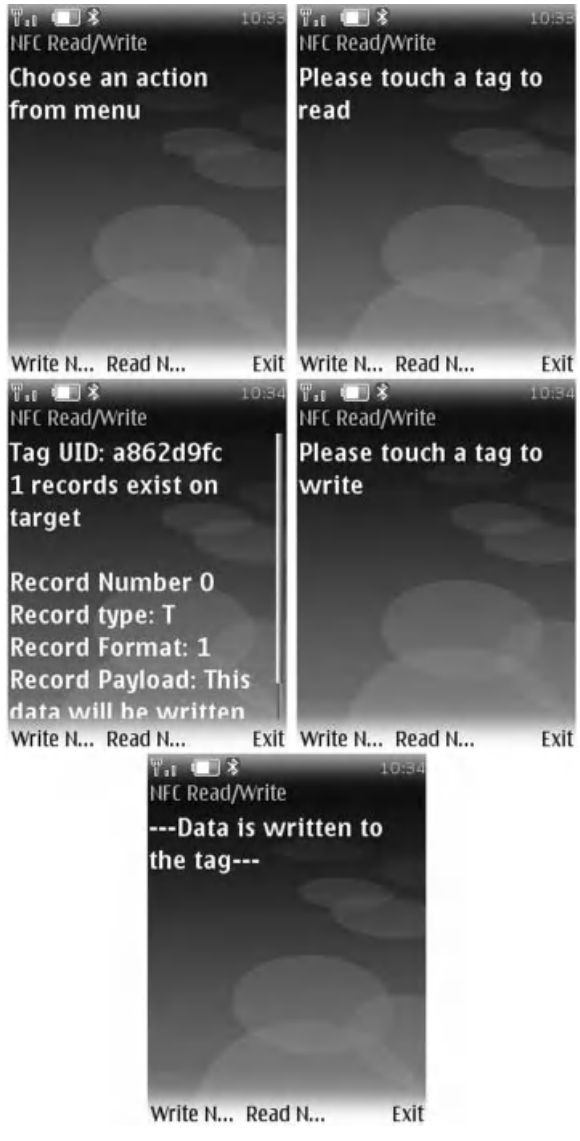


Figure 5.12 NFCReadWriteMIDlet MIDlet screens.

part of NFC programming, the `targetDetected` method is created which is automatically called when a new target is detected.

The application includes three buttons on the main screen. Two buttons are used to read and write data from/to the tag and the third one is used to exit the application. The main screen of the MIDlet is shown on the top left of Figure 5.12. When the "Read NDEF" button is pressed, the screen on the top right is displayed. When the application discovers the target in proximity the screen on the middle left of Figure 5.12 is displayed. If the user presses

the "Write NDEF" button the screen on the middle right is displayed. When the target is discovered, the application saves the desired data to the tag as shown on the bottom screen in Figure 5.12.

(i) *Application source code*

```
import java.io.IOException;
import javax.microedition.contactless.*;
import javax.microedition.contactless.ndef.*;
import javax.microedition.io.Connector;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class NFCReadWriteMIDlet extends MIDlet
  implements CommandListener, TargetListener {

  private Display screen;

  private Form mainForm;

  private StringItem screenText = new
     StringItem(null, "Choose an action from
     menu");
  private String screenString = "";

  /*
   *this data will be written to NFC tag by
   *converting it to byte array
   */
  private String payloadString = "This data will
     be written to tag";

  /*
   *two boolean variables that switches to true
   *with user command and is used to decide
   *whether to run Read or Write function
   */
  private boolean readTag = false;
  private boolean writeTag = false;

  /*
   *a boolean value which switches to true if the
   *target type is NDEF tag. It is used to inform
   *user if the target type is not NDEF tag
   */
  private boolean ndefMessage=false;

  private Command readCommand = new Command("Read
     NDEF", Command.OK, 1);
  private Command writeCommand = new
```

```
    Command("Write NDEF", Command.OK, 1);
private Command exitCommand = new
    Command("Exit", Command.EXIT, 0);

// connection object for NDEF tag types
private NDEFTagConnection conn = null;

/*
 *TargetProperties object is created to collect
 *the properties for contactless targets
 */
private TargetProperties[] target;

public NFCReadWriteMIDlet() {

  mainForm = new Form("NFC Read/Write");
  mainForm.append(screenText);
  mainForm.addCommand(readCommand);
  mainForm.addCommand(writeCommand);
  mainForm.addCommand(exitCommand);
  mainForm.setCommandListener(this);

  /*
   *DiscoveryManager class object is created to
   *discover targets for contactless
   * communication and instance of the class is
   * retrieved by getInstance() method
   */

  DiscoveryManager dm = DiscoveryManager.
     getInstance();

  // listener added for NDEF_TAG tag types
  try {
    dm.addTargetListener(this,
       TargetType.NDEF_TAG);
  }catch (ContactlessException ce) {
    screenText.setText("Error in adding
       listeners: " + ce.getMessage());
  }
}

protected void startApp() {

  screen = Display.getDisplay(this);
  screen.setCurrent(mainForm);
}
protected void destroyApp(boolean unconditional) {

  try {
```

```
      if (conn != null) {
        conn.close();
      }
    }catch (Exception e) {
      screenText.setText(" Error: " +
         e.getMessage());
    }

    notifyDestroyed();
  }

  protected void pauseApp() {

  }

  public void commandAction(Command command,
     Displayable displayable) {

    String label = command.getLabel();

    if (label.equals("Exit")) {
      destroyApp(true);
    }

    /*
     *if user hits the Read NDEF button,
     *readTag variable is set to true
     */
    if (label.equals("Read NDEF")){
      writeTag = false;
      readTag = true;
      screenText.setText("Please touch a tag to
         read");
    }

    /*
     *if user hit the Write NDEF buton,
     *writeTag variable is set to true
     */
    if (label.equals("Write NDEF")){
      readTag = false;
      writeTag = true;
      screenText.setText("Please touch a tag to
         write");
   }
  }

  /*
   *targetDetected method is automatically called
   *by the platform when a target is detected
```

```
 */
public void targetDetected(TargetProperties[]
   prop) {

  /*
   * target variable receives the properties of
   * the detected target
   */
  target = prop;

  /*
   *when target is detected and if user already
   *hit "Read NDEF" button, new thread is
   * created for Read method which will process
   * the read operation of the tag
   */
  if (readTag) {
    Read read = new Read();
    new Thread(read).start();
  }

  /*
   *when target is detected and if user already
   *hit "Write NDEF" button, new thread is
   * created for Write method which will process
   * write operation to the tag
   */
  else if (writeTag) {
    Write write = new Write();
    new Thread(write).start();
  } else {
    screenText.setText("Please choose an action
       from menu");
  }
}

/*
 * this method is used to close connection to
 * the target
 */

public void closeConnection(NDEFTagConnection
   conn){
  try {
    if (conn != null) {
      conn.close();
    }
  } catch (IOException e) {
 }
}
```

```
/*
 *Read thread runs when Read NDEF button is
 *hit and a new target is discovered
 */
public class Read implements Runnable {

 public void run() {

 /*
  *variable "ndefMessage" is set to false. If
  *the target type is an NDEF_TAG it is set to
  *true in following lines
  */
 ndefMessage = false;

 for (int i = 0; i < target.length; i++) {

  /*
   * if target type is NDEF_TAG, processing
   * continues
   */
  if (target[i].hasTargetType
     (TargetType.NDEF_TAG)) {

    ndefMessage = true;
    try {

      /*
       * get url for current NDEF_TAG target
       * to connect to it
       */
      String url = target[i].getUrl
       (Class.forName("javax.microedition.
        contactless.ndef.NDEFTagConnection"));

      conn = (NDEFTagConnection)
         Connector.open(url);

      // read formatted data from target
      NDEFMessage message = conn.readNDEF();

      if (message != null) {

        //save NDEF records to "records" array
        NDEFRecord[] records = message.
           getRecords();

        /*
         *save the "unique identifier of
         *target" and number of records to
```

```
          *"screenString" named string which
          *will be displayed on screen
          */
         screenString = "Tag UID: " +
            target[i].getUid() + "\n" +
            records.length + " records exist on
            target";

         for (int j=0; j<records.length; j++) {

           /*
            *for each record, save record
            *number, record name, record
            *format(value) and record payload
            *to "screenString"
            */
           screenString += "\n\nRecord Number "
              + j +"\nRecord type: " +
              records[j].getRecordType().
              getName()+ "\nRecord Format: " +
              records[j].getRecordType().
              getFormat()+ "\nRecord Payload: "
              + new String(records[j].
              getPayload());
         }

         // display the string on the screen
         screenText.setText(screenString);

       } else {
         screenText.setText("Tag UID: " +
            target[i].getUid() + "\nNo saved
            NDEF Messages in the target");
       }
      } catch (ContactlessException e) {
        screenText.setText("Error: " +
           e.getMessage());
      } catch (IOException e) {
        screenText.setText("Error: " +
           e.getMessage());
      } catch (Exception e) {
        screenText.setText("Error: " +
           e.getMessage());
      }
    } // end of if clause
  } // end of for loop

  if(!ndefMessage){
    screenText.setText("Target type is not an
       NDEF formatted tag");
```

```
    }

    closeConnection(conn);
  } // end of Read thread constructor
  } // end of Read thread

  /*
   *Write thread is run when Write NDEF button is
   *hit and a new target is discovered
   */
  public class Write implements Runnable {

    public void run() {

    ndefMessage = false;

    for (int i = 0; i < target.length; i++) {

      if (target[i].hasTargetType
         (TargetType.NDEF_TAG)) {

        ndefMessage = true;
        try {

          String url = target[i].getUrl
            (Class.forName("javax.microedition.
            contactless.ndef.NDEFTagConnection"));
          conn = (NDEFTagConnection)
             Connector.open(url);

         /*
          * Creates record type object formatted
          * with NFC_FORUM_RTD
          */
         NDEFRecordType recordType = new
         NDEFRecordType(NDEFRecordType.
         NFC_FORUM_RTD, "urn:nfc:wkt:T");

        /*
         *new NDEFRecord with specified
         * record type identifier and payload
         * are created
         */
        NDEFRecord newRecord = new NDEFRecord(
           recordType, null,
           payloadString.getBytes());

        /*
         *creates "newRecord" named
         *NDEFRecord to an NDEFRecord array.
```

```
            *In order to save more than one record
            *to tag, multiple records can be
            *created and saved to array
            */
          NDEFRecord[] newRecordArray = new
             NDEFRecord[]{newRecord};

          /*
           *Creates a new NDEF message with the
           * record(s) in NDEFRecord array and
           * saves it to tag
           */
          NDEFMessage newMessage = new
             NDEFMessage(newRecordArray);
             conn.writeNDEF(newMessage);
             screenText.setText("---Data is written
                to the tag---");
           } catch (ContactlessException e) {
             e.printStackTrace();
             screenText.setText("Error: " +
                e.getMessage());
           } catch (IOException e) {
             e.printStackTrace();
             screenText.setText("Error: " +
                e.getMessage());
           } catch (Exception e) {
             e.printStackTrace();
             screenText.setText("Error: " +
                e.getMessage());
           }
         } // end of if clause
       } // end of for loop

       if(!ndefMessage){
         screenText.setText("Target type is not an
            NDEF formatted tag");
       }

       closeConnection(conn);
     }// end of Write thread constructor
     } // end of Write thread
   } // end of MIDlet
```

(ii) *Explanation of the Code*

In a very basic explanation, the listener for targets is registered using the `addTargetListener` method when the application runs. The user may press the "`Read` NDEF" button; however, the mobile device cannot detect the target since the target is not in the proximity as seen in Figure 5.13.

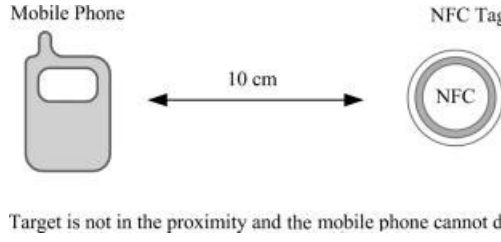


Figure 5.13 Tag not in proximity.

When the target gets into the proximity (Figure 5.14), the mobile phone detects the target. Then, the `targetDetected` method is called automatically by the platform. This method runs the Read thread. Inside the thread, connection with the target is established and data are read from the target.

Now, we will describe NFC programming parts of the MIDlet in detail.

As described earlier, `DiscoveryManager` is a class included in the `javax.microedition.contactless` package which enables the application to discover contactless targets in proximity. An object of type `DiscoveryManager` class is created as `dm`. The target listener for this object is added using the `addTargetListener` method in order to receive a notification when a contactless target is discovered in the proximity.

```
DiscoveryManager dm = DiscoveryManager.getInstance();
try {
    dm.addTargetListener(this, TargetType.NDEF_TAG);
}catch (ContactlessException ce) {
    screenText.setText("Error in adding listeners: " +
        ce.getMessage());
}
```

The `addTargetListener` method header is shown below:

```
addTargetListener(TargetListener listener, TargetType targetType)
```

As seen in the MIDlet code, the target listener is added to detect a specific target type NDEF_TAG.

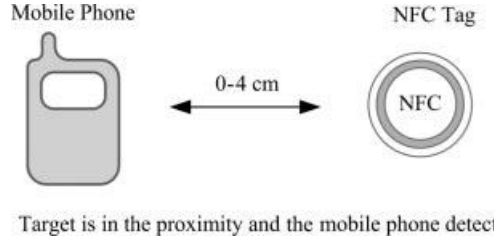


Figure 5.14 Tag within proximity.

The targetDetected method is called automatically when a target is discovered in the proximity. This method gets the parameter TargetProperties as seen in the code lines below and receives the properties of the detected targets. When a new target is detected, the properties of the target are passed to the targetDetected method and the properties are saved to the target array as defined earlier.

```
private TargetProperties[] target;

public void targetDetected(TargetProperties[] prop) {
  target = prop;
  ...
}
```

The If-else clause in the targetDetected method determines whether to run the Read or Write thread with the help of readTag and writeTag boolean variables. Those boolean variables are set in the commandAction method after the user pushes the "Read" or "Write" buttons. If the user selects the "Read NDEF" menu to read data from the tag, the readTag variable is set to true and the writeTag variable is set to false and vice versa in the"Write NDEF" menu. The Read class gets the NDEF message from the target and prints each record in the message to the screen. On the other hand, the Write class writes the predefined string to the tag as an NDEF message.

5.6.5.1 Read Class

Read class is used to read data from NDEF formatted tags. It first sets the ndefMessage boolean variable to false which will be set to true if the target is an NDEF_TAG. This variable is used later in this class to display a warning if the target type is not an NDEF_TAG.

The target is checked with the hasTargetType method, if it is an NDEF_TAG. This method is one of the methods defined in the TargetProperties interface.

```
if(target[i].hasTargetType(TargetType.NDEF_TAG))
```

When the target is identified as an NDEF_TAG, the application enters to the try-catch clause. Initially, a string named url is defined and set to the URL of the discovered target which is needed in order to connect to the target.

```
url = target[i].getUrl(Class.forName("javax.microedition.contactless.
                  ndef.NDEFTagConnection"));
```

The URL of the target is obtained by the getUrl method which is defined in the TargetProperties interface. Usage of the getUrl method is as follows:

```
public java.lang.String getUrl(java.lang.Class connectionName)
conn = (NDEFTagConnection) Connector.open(url);
```

Connection to the URL is opened using the open method which is defined in the javax.microedition.io.Connector class. As NDEF formatted data can be accessed with NDEFTagConnection, the parameter in the getUrl method is set to NDEFTagConnection.

After the connection is set up, an NDEFMessage is created and NDEF records are read from the tag using the readNDEF method which is defined in the NDEFTagConnection interface.

```
NDEFMessage message = conn.readNDEF();
```

If there is any NDEF message in the tag, then a new NDEFRecord array, records, is created and the records are saved to this array.

```
NDEFRecord[] records = message.getRecords();
```

There are two possible constructors for NDEFRecord (given in the previous section). On the other hand, the getRecords method in the NDEFMessage class returns all NDEF records in an NDEF message. The usage of the method is as follows:

```
public NDEFRecord[] getRecords()
```

When all NDEF records are saved, the only thing that remains is to process the data.

Since the application has already saved all NDEF records into the records array, the unique identifier of the discovered target is first printed, just to show some examples of API usage. The unique identifier is about the properties of the target and the getUid method defined in the TargetProperties interface returns the unique identifier of the discovered target. In the application, the unique identifier of the target together with the number of records are saved to a string as shown below:

```
screenString="Tag UID: " + target[i].getUid() + "\n" +
records.length + " records exist on target";
```

Now, it is time to display the records in the tag. In order to print all records, we need to print the records in a loop. We can do so many things using the records with the methods defined in the NDEFMessage and NDEFRecord classes.

In the given application example, the data shown in Table 5.10 are printed to the screen.

getRecordType is a method which is defined in the NDEFRecord class and returns an NDEFRecordType which contains the type and the type format information of the data.

The getRecordType().getName() returns the name of an NDEF record without any prefixes. On the other hand, getRecordType().getFormat() returns the format of the record type name. Available format options are defined in the NDEFRecordType class in the previous section. As an example, NFC_FORUM_RTD returns the value of 1.

In order to print the payload of the data, the getPayload method is used which is defined in the NDEFRecord class. This method returns the payload in the NDEF record as a byte array.

Table 5.10 Printed data in NFCReadWriteMIDlet

Data	Printed by
Record Number	j variable in the for loop
Record Type	getRecordType().getName()
Record Format	getRecordType().getFormat()
Record Payload	getPayload()

In the example, all data gathered from the tag are saved to a string named `screenString`. The last thing to do is to print the string to the screen. The `uid` of the tag, number of records in the tag, record number, record type, record format and payload of all records are to be displayed on the screen.

5.6.5.2 Write Class

`Write` class is used to write data to NDEF formatted tags. Everything is the same as with the `Read` class until reading the data.

After setting up the connection to the tag with the `Connector.open(url)`, an `NDEFRecordType` object is created which is named `recordType`. It includes the format and the name of the record. In our example, format and name values are NFC_FORUM_RTD and urn:nfc:wkt:T, respectively:

```
NDEFRecordType recordType = new
NDEFRecordType(NDEFRecordType.NFC_FORUM_RTD,"urn:nfc:wkt:T");
```

`NDEFRecordType` is then used to create a new `NDEFRecord` object named `newRecord` with specified record type, identifier and payload:

```
NDEFRecord newRecord= new NDEFRecord(recordType, null,
payloadString.getBytes());
```

The following constructor is used for the `NDEFRecord`:

```
NDEFRecord(NDEFRecordType recordType,byte[] id,byte[] payload)
```

As described in the previous section, an `NDEFMessage` can be created in three ways. We will create it with an `NDEFRecord` array as follows:

```
NDEFMessage(NDEFRecord[] records)
```

The created `NDEFRecord` with the name `newRecord` is saved to a new `NDEFRecord` array to construct an `NDEFMessage`:

```
NDEFRecord[] newRecordArray = new NDEFRecord[] { newRecord };
```

Then, a new `NDEFMessage` is created with the following code:

```
NDEFMessage newMessage = new NDEFMessage(newRecordArray);
```

In the sample MIDlet, one record is saved to the tag; however, multiple records can also be processed by saving multiple records to the `NDEFRecord` array as follows:

```
NDEFRecord[] newRecordArray = new NDEFRecord[] { newRecord1,
newRecord2,...};
```

The last thing to do is to write the `NDEFMessage` to the tag:

```
conn.writeNDEF(newMessage);
```

It is also important to close the connection to the contactless target after completing the required operations:

```
conn.close();
```

5.6.6 NFC Push Registry

Remember from earlier chapters that the push registry simply maintains a list of inbound connections and maps network connection strings to class names in order to run them automatically. In the case of NFC, push registry is an important topic; it aims to improve the usability of applications by launching applications automatically with an NFC connection.

For reader/writer mode, when a target contains the desired NDEF record, the application registered for this record can be automatically launched, so those users do not need to launch the application manually. In order to launch an application using push registry, targets must have NFC Forum formatted NDEF records.

The name and format of a record type name are used to match the desired application with a corresponding NDEF record. Note that one unique pair of name and format can only be matched with one application. Moreover, the first record on the target that matches with the push registry entry on the phone is used.

Assume that a contactless target is discovered in proximity. First, AMS checks the available push registry entries. If there is an entry with the matching name and format of record type name, the application in the entry is executed. Otherwise, if there is a listener for the NDEF record (`NDEFRecordListener`), notification is sent to this listener. When the application is started with the push registry entry, the application should automatically register the listener for NDEF records to further receive notifications on discovered NDEF records.

5.6.6.1 NDEF Push Record Format

The NDEF push record format needs to be formed as follows:

NDEF push record format = "ndef:" + record_type_format + "?name=" +record_type_string

- record_type_format variable may be assigned to one of the following four available values:
  record_type_format= "rtd" | "external_rtd" | "mime" | "uri"
- record_type_string variable can have multiple values:
  record_type_string= fully qualified name of the record type

Please see Table 5.11 for example NDEF push records.

As described earlier, push registry can be implemented in two ways: dynamic registration and static registration. `javax.microedition.io.PushRegistry`

Table 5.11 Example NDEF push records

NDEF push record	Description
ndef:rtd?name=urn:nfc:wkt:Sp	Record to register an application when a target containing Smart Poster RTD is discovered
ndef:mime?name=text/x-uri	Record to register an application to be started when a target containing an URL is discovered
ndef:ext?name=urn:nfc:ext:example.com:f	Record to register an application when a target with the specified data is discovered

`.registerConnection` method should be used for a dynamic registration. There are other useful methods in the PushRegistery class such as unregistering a connection or getting a list of registered push connections. For full specification, please refer to the `javax.microedition.io.PushRegistry` specification.

An example push registry connection from a mobile Java application is as follows:

```
PushRegistry.registerConnection("ndef:rtd?name=urn:nfc:wkt:T",
     "MyMobileApplication", "*");
```

5.7 Peer-to-Peer Mode Programing

It was stated earlier that the peer-to-peer mode performs device-to-device link-level communication. It is not supported by standard Java APIs; however, there are various extensions to Contactless Communication API that provide programming in peer-to-peer mode. These are generally mobile device specific APIs, namely proprietary APIs. These APIs can be found on developer websites for specific devices.

JSR 257 API extensions are provided for Nokia 6212 mobile devices [14]. These extensions provide additional functions including peer-to-peer mode support. A summary of Nokia 6212 JSR 257 API extensions is given in Table 5.12.

As seen in Table 5.12, the `com.nokia.nfc.llcp` package and `com.nokia.nfc.p2p` package provide interfaces for peer-to-peer mode programming using LLCP (Logical Link Control Protocol) and NFCIP-1 (Near Field Communication Interface and Protocol-1). Most of the functions of a peer-to-peer mode application cannot be tested along with the SDK since two mobile devices are needed, and SDK provides one instance of a mobile device. Applications should be loaded to mobile phones and testing should be performed on them for this reason.

5.7.1 Package com.nokia.nfc.p2p

This package contains one interface; NFCIPConnection which provides communication between two NFCIP devices.

Table 5.12 Nokia 6212 Classic JSR 257 implementation overview

Package Name	Description
com.innovision.rf	Provides an interface for accessing Innovision Jewel tags
com.nokia.nfc.llcp	Provides an interface for communicating with remote devices using LLCP
com.nokia.nfc.nxp.desfire	Provides a connection interface and utility classes to be used when accessing MIFARE DESFire cards
com.nokia.nfc.nxp.mfstd	Provides an interface for accessing a MIFARE Standard card
com.nokia.nfc.nxp.simpletag	Provides an interface for accessing a SimpleTag
com.nokia.nfc.p2p	Provides an interface for communicating with NFCIP-1 devices
com.sony.felica	Provides an interface for accessing a NFC Forum Type 3 tag

Table 5.13 Constant field values for NFCIPConnection

Name	Value
MAX_LEN	65 524
MODE_INITIATOR	1
MODE_TARGET	2

5.7.1.1 Interface NFCIPConnection

The `NFCIPConnection` interface enables communication between two devices using NFCIP-1. During the communication, one device becomes the initiator and the other device becomes the target. Communication is performed according to the initiator–target paradigm in which when one device is sending data, the other one needs to listen. The connection is opened using the `Connector.open` method and the connection mode of the devices is determined in this step.

The connection string needs to be formatted as follows:

NFCIP Connection String= "nfc:rf;type=nfcip;mode=" + target (+ ";timeout=" + timeout)

- The connection mode can be one of the two types:
  target = "initiator" | "target"
- Timeout is optional and defines the timeout value for a waiting target device in milliseconds (1000 ms = 1 s):
  timeout = an integer value

Methods `send` and `receive` are used in order to send and receive data to/from a remote device in bytes. The initiator device should first send data and then wait to receive. After receiving the data, it can send again, and so on. On the other hand, the target device should first wait to receive data, and then send.

Using the `getMode` method, the connection mode of the device can be received which returns either MODE_TARGET or MODE_INITIATOR. The constant field values of the target and initiator devices are given in Table 5.13. Moreover the `getUID` method returns the `uid` of the other NFC device in bytes. The MAX_LEN constant in Table 5.13 defines the maximum amount of bytes that can be sent in one call.

5.7.2 Package com.nokia.nfc.llcp

LLCP API provides an interface to MIDlets in order to communicate with remote devices using LLCP.

This package contains the `LLCPManager` class which is used for managing LLCP related listeners. The package also contains four interfaces as given in Table 5.14.

Both LLCP transport types are supported in this package. Remember that, Type 1 provides a connectionless data transport which is unreliable and sessionless. On the other hand, Type 2 provides a connection oriented data transport which is reliable and session based. In Type 2

Table 5.14 com.nokia.nfc.llcp package

Classes	LLCPManager
Interfaces	ErrorRecoveryListener LLCPConnection LLCPConnectionListener LLCPLinkListener

transport, connections have separate channels. However, in Type 1 transport, connections are specific for a service identifier and the first incoming data frame will open the connection. Both transport types are supported in this API. Listener is set up using the `startListening` method in the `LLCPManager` class and also the transport type is chosen here. Communication using a transport type is done with the `LLCPConnection` interface.

Listeners for activities on LLCP links are added and removed using the `LLCPManager` class and notifications are received by the `LLCPLinkListener` interface. After link establishment, communication to the target can be opened through the `Connector.open` method. When the connection is set up, data can be sent and received using the `send` and `receive` methods in the `LLCPConnection` interface.

An LLCP connection URI format is standard and it needs to be formatted as follows:

LLCP Connection String= "nfc:llcp;type=" + transport_type + ";sid=" + service_id + ";uid=" +uid

- The `Transport_type` variable can be one of the two types:
  transport_type = "1" / "2"
- The `service_id` variable can have different integer values between 0 and 62:
  service_id = integer (0–62)
- The `uid` variable can have different hexadecimal strings:
  uid = hexadecimal string of the uid of the remote LLCP device

(i) *Class LLCPManager*

The `LLCPManager` class is used for managing LLCP related listeners. Using this class's methods, devices can add and remove listeners, start and stop listening for incoming connections. It also implements listeners for LLCP error recovery related events.

The `addLinkListener` and `removeLinkListener` methods add and remove the `LLCPLinkListener`, so that the device can be notified on the LLCP links.

The `startListening` and `stopListening` methods are used to start and stop listening for connections. Once a connection has been opened, the connectionOpened method from the `LLCPConnectionListener` interface is called automatically.

Furthermore, two methods, `addErrorRecoveryListener` and `removeErrorRecoveryListener` add and remove an `ErrorRecoveryListener` in order to be notified about LLCP error recovery related events.

(ii) *Interface ErrorRecoveryListener*

This interface is used in order to send notification to the application on LLCP link error recovery. When a recovery status is changed, the `recoveryStatus(int)` is called

Table 5.15 Constant field values for recoveryStatus method

Name	Value
RECOVERY_STARTED	1
RECOVERY_FAILED	2
RECOVERY_SUCCESSFUL	3

by the platform to notify on the change. Possible status values for the recoveryStatus method are given in Table 5.15.

(iii) *Interface LLCPConnection*

This interface is used to communicate with another LLCP device. The main methods of the interface are `send` and `receive` which are used to send and receive data. Connection related values can be gathered by the methods of this class. `getTransportType` returns the transport type of the connection in bytes. The `getUID` method returns the `uid` of the remote device in string form.

As described earlier, one of two different transport types needs to be selected. Possible values of transport types and maximum data length that can be sent in one call are given in Table 5.16.

(iv) *Interface LLCPConnectionListener*

This interface is used in order to be notified when an LLCP connection is opened. The `connectionOpened` method is called automatically by the platform when the connection is opened.

(v) *Interface LLCPLinkListener*

This interface provides a notification mechanism when a link with the target is established or lost. The `linkEstablished` method, as its name suggests, is called when a link is established and the `linkLost` method is called when a link is lost. Both methods are called automatically and the `uid` of the remote device is sent to the `linkEstablished` method.

(vi) *Push Registry in LLCP*

The required application can be launched automatically with a push registry entry and by the trigger of the other device. However some parameters need to be given in push registration URI.

LLCP Push registry= "llcp-type" + transport_type + ":?sid=" +service_id

Table 5.16 Constant field values for LLCPConnection

Name	Value
MAX_DATA_LEN	65 521
TYPE_1	1
TYPE_2	2

As stated earlier, the peer-to-peer mode related packages are `com.nokia.nfc.llcp` and `com.nokia.nfc.p2p`. We have introduced these two packages in this section. Other packages introduce different types of connections related to smart cards and tags which are not discussed here. In order to learn about those packages and for more detailed information on the two packages discussed, please refer to the JSR 257 Nokia Extensions [14].

5.7.3 A Peer-to-Peer Mode Application

In this section, we provide a sample MIDlet that enables peer-to-peer communication between two mobile phones using the NFCIP-1 communication interface. In this application, two users can send and receive messages to each other.

When the MIDlet is first launched, it is in the read state, meaning that it is waiting for a connection from an initiator. Also a main screen form is displayed to the user as seen in the top-left screen in Figure 5.15. After the connection is set up with the initiator, it is ready to receive a message. Remember that in NFCIP communication a target must first receive a

Figure 5.15 P2PExample MIDlet screens.

message and then send and then again receive. If a user wants to send a message, she writes her message through a text box as seen in the top-left screen and then pushes the "Send Message" button. In this case, the state of the application is in the write state and the application is trying to make a connection with a target device (bottom screen in Figure 5.15). When it finds a device in a reading state the message exchange is performed.

MIDlet mainly consists of user interfaces, a thread and a state changer. The state changer provides the MIDlet to switch from the reading state to the writing state and vice versa. The thread performs the NFCIP-1 connection set-up and the message exchange. After the message exchange, both devices print the exchanged message to the screen.

(i) *Application source code*

```
import java.io.IOException;
import javax.microedition.io.Connector;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
import com.nokia.nfc.p2p.NFCIPConnection;

public class P2PExample extends MIDlet
  implements Runnable, CommandListener {

  // NFCIP Connection object
  private NFCIPConnection conn;

  /*
   *connection string to be used by initiator and
   * target devices
   */
  private static final String INITIATOR =
     "nfc:rf;type=nfcip;mode=initiator";
  private static final String TARGET =
     "nfc:rf;type=nfcip;mode=target";

  private StringItem screenText = new
     StringItem(null, "Write a new message to
     send\n");

  private Display screen;
  private Form mainForm;
  private TextBox chatTextBox;

  private Command writeCommand = new Command("New
     Message", Command.SCREEN, 1);
  private Command sendCommand=new Command("Send
     Message", Command.SCREEN, 1);
  private Command cancelCommand=new
     Command("Cancel", Command.CANCEL, 0);
  private Command exitCommand=new Command("Exit",
     Command.EXIT, 0);
```

```
/*
 *determines if the application is in reading
 *state or in writing state
 */
boolean readState = true;

public P2PExample() throws IOException {

  mainForm = new Form("Form");
  mainForm.append(screenText);
  mainForm.addCommand(writeCommand);
  mainForm.addCommand(exitCommand);
  mainForm.setCommandListener(this);
}

protected void startApp(){

  screen = Display.getDisplay(this);
  screen.setCurrent(mainForm);

  /*
   *a new thread is started at startup in order
   *to get incoming connections and messages at
   *main screen
   */

  Thread thread = new Thread(this);
  thread.start();
}

protected void pauseApp() {

}

protected void destroyApp(boolean unconditional){

  notifyDestroyed();
}

public void commandAction(Command command,
   Displayable displayable) {

  String label = command.getLabel();

  if (label.equals("Exit")) {
    destroyApp(true);
  }else if(label.equals("New Message")) {
    chatMenu();
```

```
  }else if(label.equals("Send Message")) {
    changeState();
  }else if (label.equals("Cancel")) {
    screen.setCurrent(mainForm);
  }
}
// displays a textbox to write a message
public void chatMenu() {

  chatTextBox = new TextBox("Text to write",
      null, 255, TextField.ANY);
  chatTextBox.addCommand(sendCommand);
  chatTextBox.addCommand(cancelCommand);
  chatTextBox.setCommandListener(this);
  screen.setCurrent(chatTextBox);
}
// method to change the readState value
public void changeState() {

  /*
   *if readState is true(MIDlet is in read
   *state) and MIDlet called this method (user
   *pushed "Send Message button"),readState is
   *set to false in order to switch to write
   *state and make the connection to a target.
   */
  if(readState){
    readState = false;

    Alert sendAlert = new Alert("Sending
       Message...", "Please Touch", null,
       AlertType.INFO);
    sendAlert.addCommand(cancelCommand);
    sendAlert.setTimeout(10000);
    sendAlert.setCommandListener(this);
    screen.setCurrent(sendAlert);
  }

  /*
   *if readState is false (user successfully
   *sent the message),it is set back to true
   *(reading state), so MIDlet set back to
   *reading state.
   */
  else{
    readState = true;
    screen.setCurrent(mainForm);
  }
```

```
    /*
     *finally a new thread started at the end of
     *the method in order to activate new
     *readState values
     */
    Thread thread = new Thread(this);
    thread.start();
  }
  /*
   *thread that performs the NFCIP connection
   *setup and the message Exchange
   */
  public void run() {

    /*
     *conn is set to null to reset the connection
     *in each thread run
     */
     conn = null;

     if (readState) {

     /*
      *if readState is true, device waits until
      *initiator opens a  connection
      */

     while (readState) {
       try {
         conn = (NFCIPConnection)
            Connector.open(TARGET);

         /*
          *when the connection is set up,
          *target receives the first message
          *from initiator
          */
          byte[] data = conn.receive();

          /*
           *target sends a meaningless message to
           *the initiator, since initiator is
           *waiting a reply
           */
          conn.send("-1".getBytes());

          // write received message to the screen
          mainForm.append("Received: " + new
             String(data) + "\n");
```

```
        // close the current connection
        conn.close();
      } catch (Exception e) {
        screenText.setText("Error in connection
           : " + e.getMessage());
      }
    }
  }else {

    /*
     *if readState is false, device tries to
     * open a connection to   a target device
     * and waits until a device is found or
     *canceled
     */
     while (!readState) {
       try {
         conn = (NFCIPConnection)
           Connector.open(INITIATOR);

     /*
      *when connection is set up, initiator
      *sends first message to target
      */
     conn.send(chatTextBox.getString().
        getBytes());

     /*
      *initiator waits and receives the
      *reply from target
      */
     conn.receive();

     // write sent message to the screen
     mainForm.append("Sent: " +
        chatTextBox.getString() +"\n");

     // display the main form at the screen
     screen.setCurrent(mainForm);

     // close the current connection
     conn.close();

    /*
     *call changeState() method in order to
     *change reading state to true
     */
     changeState();
   } catch (Exception e) {
```

```
            screenText.setText("Error in connection
               : " + e.getMessage());
          }
         }
        }
       }
      }
```

(ii) *Explanation of the code*

We will now describe the NFC programming parts of the MIDlet. Two final strings are defined as the INITIATOR and TARGET. These are connection strings that will be used for establishing a connection. As seen in the string, the modes of the devices (initiator or target) are also stated.

```
private static final String
    INITIATOR="nfc:rf;type=nfcip;mode=initiator";
private static final String
TARGET="nfc:rf;type=nfcip;mode=target";
```

The peer-to-peer operating mode uses half duplex transmission. Hence, one of the partners is in sending, and the other is in receiving mode at any given time. The sending device is in the initiator role for that communication, and the receving device is in the target role. In order to handle this situation, the `readState` boolean variable is defined. Since multithreaded programming is used in the example, the same code is executed by both devices. Hence, when `readState` is true for one device, it should be false for the other. The `changeState` method handles switching between the reading and writing states and thus it results in the device becoming an initiator or a target.

```
public void changeState(){
  ...
}
```

The `changeState` method is called twice in the MIDlet:

- The `commandAction` method consists of invocation of the `changeState` method. As the user writes the text in the textbox and presses the "Send Message" button in order to send the text to the target device, the `commandAction` method is invoked after which the device changes its state to the writing state in order to perform the sending action.
- After MIDlet sends the message, it needs to be in reading state in order to listen for incoming messages from the remote device. However it is currently in the writing state and the state change is performed by calling the `changeState` method.

Programing for peer-to-peer communication is performed within a thread. The `NFCIPConnection` object `conn` is set to null (which ensures setting up a new connection if the MIDlet is not running the thread first time) and then the thread is split into two parts, one for the reading state and one for the writing state.

```
public void run() {
  ...
}
```

Devices both in the read state and write state perform similar actions; however, the NFCIPConnection object is set up differently. If MIDlet is in the reading state, it is set to TARGET; otherwise it is set to INITIATOR.

```
conn = (NFCIPConnection) Connector.open(TARGET);
conn = (NFCIPConnection) Connector.open(INITIATOR);
```

MIDlet in the reading state first receives data using the receive method and sets the data byte array to the received data. Remember that receive is a method defined by the NFCIPConnection interface and used to receive data from the other NFC device.

```
byte[] data = conn.receive();
```

Then, the target device sends data to the initiator, since the initiator is waiting for a reply. In this example, the target sent "-1" to the initiator in order to notify on a receive operation. The application understands the meaning of "-1" since it is programmed in this way. The send method is also defined by the NFCIPConnection interface and used to send data to the other NFC device.

```
conn.send("-1".getBytes());
```

The only remaining thing to do for the target device is to print the received message onto the screen and close the connection afterwards. There is no need to change its state, since it is still in the reading state.

```
mainForm.append("Received: " + new String(data) + "\n");
conn.close();
```

On the other hand, after the connection is set up, the initiator device first sends the data to the remote device, and waits for a reply. After getting the reply, it displays the data on the screen after switching the screen to mainForm. Note that, the reason to change the screen is that an alert was still being displayed. Then, MIDlet closes the connection. Finally, the initiator device needs to change its state from the writing state to the reading state, and thus calls the changeState method for this purpose.

```
conn.send(chatTextBox.getString().getBytes());
conn.receive();
mainForm.append("Sent: " + chatTextBox.getString() + "\n");
screen.setCurrent(mainForm);
conn.close();
changeState();
```

This example demonstrated a peer-to-peer communication. We have showed how to set up communication with a remote device and how to send and receive messages using NFCIP-1 communication.

5.8 Card Emulation Mode Programing

Accessing SEs can be programmed using either JSR 257 or JSR 177.

5.8.1 Accessing Secure Element Using JSR 257

`APDUConnection` and `JavaCardRMIConnection` allow access to smart cards in JSR 177 API which will be described in the following sections. In JSR 257, the `TransactionListener` interface can be used to access SEs.

When an activity occurs between the SE and the external reader device, MIDlet receives the notification using the `TransactionListener` interface if it is listening to the SE. In order to receive notifications, applications need to implement this interface and also need to register a transaction listener using the `addTransactionListener` method of the `DiscoveryManager` class.

The `externalReaderDetected` method is also defined in this interface which is called automatically by the platform to notify the application that an event has happened on the SE. A possible cause is a change in the SE, and the application may take the required action (e.g., show remaining credits on screen).

5.8.2 Accessing Secure Element Using JSR 177

The Security and Trust Services API (SATSA) (JSR 177) [13] defines optional packages to support smart card communication and security operations. Digital certificates, digital signatures, and much more can be implemented using this API.

As described in earlier chapters, SEs can be in a variety of forms. The most important favorable SE option is the smart card and JSR 177 provides the programming environment. Smart card communication with this API can be implemented either using APDU protocol or JavaCard RMI protocol.

The card emulation mode is developed mainly for providing applications that impose high security constraints such as credit cards. This API handles the required security operations including digital signature services, user credential management, cryptographic operations, and cryptographic key storage.

5.8.2.1 Contents of Optional Packages

Four optional packages are defined in this API which consists of different packages and classes:

(i) *SATSA-APDU optional package*

The SATSA-APDU optional package is used for communication with smart cards using a protocol based on APDUs. This optional package contains the following:

- Exception class `UnsupportedOperationException` in `java.lang` package
- javax.microedition.apdu package

(ii) *SATSA-JCRMI optional package*

The SATSA-JCRMI optional package is used for communication with smart cards using the JavaCard RMI protocol. This optional package contains the following:

- `javax.microedition.jcrmi` package
- A subset of the `java.rmi` package
- A subset of the `javacard.framework` package

- A subset of the `javacard.framework.service` package
- A subset of the `javacard.security` package
- Exception class `UnsupportedOperationException` in the `java.lang` package

(iii) *SATSA-PKI optional package*

The SATSA-PKI optional package enables smart cards to manage digital signatures and certificates. This optional package contains the following:

- `javax.microedition.pki` package
- `javax.microedition.securityservice` package

(iv) *SATSA-CRYPTO optional package*

The SATSA-CRYPTO optional package provides cryptographic operations such as message digests and digital signatures to increase security. This optional package contains the following:

- `java.security` package
- `java.security.spec` package
- `javax.crypto` package
- `javax.crypto.spec` package
- Exception class `IllegalStateException` in the `java.lang` package

5.8.2.2 Details of Optional Packages

(i) *SATSA-APDU optional package*

This package enables applications to access smart cards using APDUs. An APDU is the communication unit (a message) represented by bytes. This package enables an application on the resident smart card to exchange these APDU messages with a card application. APDU commands should conform to the format specified in ISO7816-4.

There are two types of messages in an APDU: command APDUs and response APDUs. This means that an MIDP application can both send commands to the smart card application and receive commands from it.

Connections to a smart card are made using the Generic Connection Framework. Calling the `Connector.open` method will return the connection of the APDU connection which will be used to connect with the card application. A smart card application is identified by the Application Identifier (AID).

An APDU connection string needs to be formatted as follows:

APDU Connection String= "apdu:" + slot_number + ";target=" + target

- *slot_number* indicates the slot number that the card application will be communicated through.
- *target* is either an AID value or "SAT" string. If the communicated smart card application is (U)SIM Application Toolkit [(U)SAT], then the value should be a "SAT" string. Otherwise, the value should be an AID value. The AID is represented by 5–16 hexadeximal bytes and each byte value is separated by a ".". Also note that if the smart card application is a (U)SAT, then the communication needs to occur through channel 0.

For example, the following code will try to establish a connection with the target which has an AID of A0.0.0.67.4.7.1F.3.2C.3 from slot 0:

```
String url = "apdu:0;target=A0.0.0.67.4.7.1F.3.2C.3";
```

```
APDUConnection connection =
(APDUConnection)Connector.open(url);
```

If the specified slot, the card, or the card application in the specified slot does not exist, the connection will throw `ConnectionNotFoundException`.

Moreover, available card slots can be gathered by the `getProperty` method with a parameter of "microedition.smartcardslots". If the slot is cold-swappable, then "C", if the slot is hot-swappable then "H" is appended to the slot number.

The `exchangeAPDU` method is used to send and receive messages between MIDP and smart card applications. Messages are retrieved using a byte array. In the following code example, a byte array named `apdu` is sent to the card application and the response is saved to a `response` byte array when it is received:

```
byte[] response = connection.exchangeAPDU(apdu);
```

(ii) *SATSA-JCRMI optional package*

This package enables communication with a smart card using JavaCard Remote Method Invocation (JCRMI) protocol. It allows applications that are not located in the smart card to communicate with the applications on the smart card. In order to achieve this, the applications use stubs for communicationg with remote objects on a smart card. A stub can be seen as a proxy for a remote object. When an application invokes a method on the stub, this invoke operation is transferred to the smart card and the results are returned to the application after performing the required operations on the smart card.

Smart card slots can be discovered in the same way as it is described in the SATSA-APDU package. The connection in SATSA-JCRMI is also implemented similarly. A JCRMI connection string needs to be formatted as follows:

JCRMI Connection String= "jcrmi:" + slot_number + ";AID=" + AID

The following code shows an example of the JCRMI connection:

```
String url = "jcrmi:0;AID= A0.0.0.67.4.7.1F.3.2C.3";
JavaCardRMIConnection
  connection=(JavaCardRMIConnection)Connector.open(url);
Counter counter = (Counter) connection.getInitialReference();
```

After setting up the connection, a stub object for remote reference can be retrieved using the `getInitialReference` method. This reference allows the application to invoke methods on a remote object, by invoking it on the instance.

(iii) *SATSA-PKI optional package*

This package allows applications to manage digital signatures and certificates. Applications can sign messages with a private key in order to satisfy authentication and non-repudiation. The application can also perform certificate enrollment requests, and add/remove certificates to/from the certificate store using the SATSA-PKI optional package.

(iv) *SATSA-CRYPTO optional package*

This package allows applications to manage cryptographic functions. It includes message digests and digital signatures to satisfy the integrity of data, ciphers to encrypt and decrypt data. This package's main aim is to protect security and privacy of the data in

the SE. Note that the SATSA-CRYPTO package is a subset of the J2SE platform Java Cryptography Extension.

5.9 Reader/Writer Mode Case Study: NFC Shopping

At the end of Chapter 4, three use cases are described and presented with their design requirements and generic usage models. In this section we implement the first two use cases. The third use case's ecosystem environment and business models are analyzed at the end of Chapter 7.

As described in Chapter 4, the reader/writer mode provides so many opportunities for users and developers. Many real life scenarios can be implemented in this mode. In this use case we have implemented an online shopping scenario in order to show NFC in action. The application does not include complex operations since our aim is not to introduce advanced programming algorithms, but only to provide NFC programming and the idea behind the operating mode in a real life scenario. For production purposes, many features should be added to the application such as an easy to use and good-looking user interface, more shopping functions (deleting a product, reading comments about the product online, etc.) and other customer services.

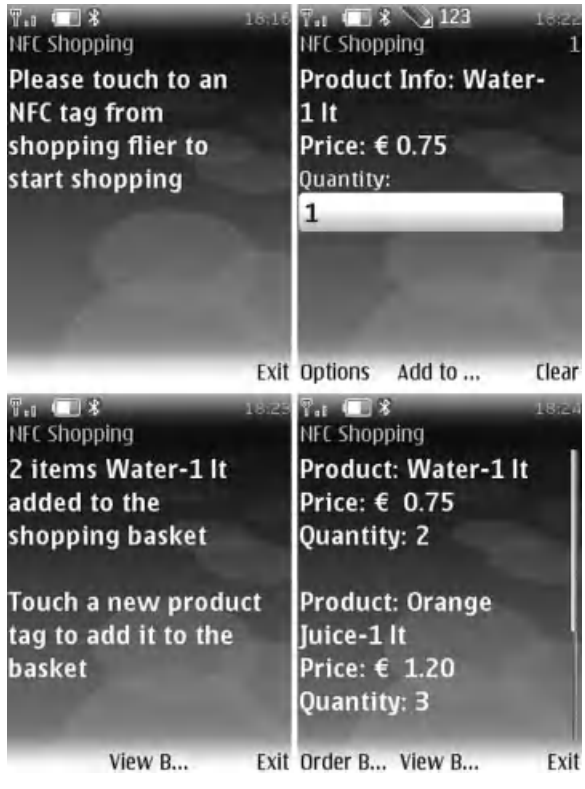


Figure 5.16 NFC shopping MIDlet screens.

Remember from the use case in Chapter 4 that NFC shopping provides a shopping opportunity to users by eliminating geographical restrictions. The user adds products to the basket simply by touching the phone to the tags on the NFC enabled shopping catalogs. Application screenshots are given in Figure 5.16.

When the user runs the application, the application waits for an NFC interaction to start as seen in the top-left screen in Figure 5.16. When the desired tag is discovered within the proximity, data in the tag are transferred to the mobile phone and product information is displayed to the user, and then the desired quantity is requested (top-right screen in Figure 5.16). When the user confirms adding product to the basket, the product is inserted to the basket and a confirmation message is displayed to the user (bottom-left screen in Figure 5.16). These steps are repeated until the user decides to order the basket. The user may also view the current basket by pressing the "View Basket" button (bottom-right screen in Figure 5.16).

(i) *Application source code*

```
import java.io.IOException;
import java.util.Vector;
import javax.microedition.contactless.*;
import javax.microedition.contactless.ndef.*;
import javax.microedition.io.Connector;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class ShoppingMain extends MIDlet
   implements Runnable, CommandListener,
   TargetListener {

   private NDEFTagConnection conn = null;

   private TargetProperties[] target;

   private Display screen;
   private Form mainForm, addProductForm ;

   private StringItem screenText = new
      StringItem(null,"Please touch to an NFC tag
      from shopping flier to start shopping");

   private String screenString,productinfo,
      productprice, productid;

   private int productquantity;
   private double totalbasketprice=0;

   private Vector basketprices = new Vector();
   private Vector basketinfo = new Vector();
   private Vector basketquantity = new Vector();
   private Vector basketids = new Vector();
```

```
private TextField quantityTextField;

private boolean ndefMessage=false;

private Command addToBasketCommand = new
   Command("Add to Basket", Command.SCREEN, 1);
private Command viewBasketCommand = new
   Command("View Basket", Command.SCREEN, 1);
private Command orderBasketCommand = new
   Command("Order Basket", Command.SCREEN, 1);
private Command cancelCommand = new
   Command("Cancel", Command.CANCEL, 0);
private Command exitCommand = new
   Command("Exit", Command.EXIT,0);

public ShoppingMain() {

  createMenu();

  DiscoveryManager dm = DiscoveryManager.
     getInstance();

  try {
    dm.addTargetListener(this,
       TargetType.NDEF_TAG);
  }catch (ContactlessException ce) {
    screenText.setText("Error in adding
       listeners: " + ce.getMessage());

  }
}

protected void startApp() {

  screen = Display.getDisplay(this);
  screen.setCurrent(mainForm);
}

protected void pauseApp() {

}

protected void destroyApp(boolean unconditional)
{

  try {
    if (conn != null) {
      conn.close();
    }
  } catch (Exception e) {
```

```
    screenText.setText(" Error: " +
       e.getMessage());

  }

  notifyDestroyed();
}

public void commandAction(Command command,
   Displayable displayable) {

  String label = command.getLabel();

  if (label.equals("Exit")) {
    destroyApp(true);
  }else if(label.equals("Cancel")) {
    screenText.setText("Please touch to an NFC
       tag from shopping flier to start shopping");
    screen.setCurrent(mainForm);
  }else if(label.equals("Add to Basket")) {
    addToBasket();
  }else if(label.equals("View Basket")) {
    viewBasket();
  }else if(label.equals("Order Basket")) {
    orderBasket();
  }
}

public void targetDetected(TargetProperties[] prop) {

  target = prop;

  Thread thread = new Thread(this);
  thread.start();
}

public void closeConnection(NDEFTagConnection conn){
  try {
     if (conn != null) {
       conn.close();
     }
  } catch (IOException e) {

  }
}

public void createMenu(){

  mainForm = new Form("NFC Shopping");
  mainForm.append(screenText);
```

```
    mainForm.addCommand(exitCommand);
    mainForm.setCommandListener(this);
  }

  public void addToBasket(){

    mainForm.addCommand(viewBasketCommand);
    productquantity = Integer.parseInt
       (quantityTextField.getString());
    basketinfo.addElement(productinfo);
    basketprices.addElement(productprice);
    basketquantity.addElement(
       Integer.toString(productquantity));
    basketids.addElement(productid);
    totalbasketprice += productquantity *
       Double.parseDouble(productprice);
    screenText.setText(productquantity + " items "
       + productinfo + "added to the shopping
       basket" + "\n\nTouch a new product tag
       to add it to the basket");
    screen.setCurrent(mainForm);
  }

  public void addProduct(){

    quantityTextField = new TextField("Quantity:",
        "1", 2, TextField.NUMERIC);
    addProductForm = new Form("NFC Shopping");
    addProductForm.append("Product Info: " +
       productinfo + "\nPrice: €  "+ productprice);
    addProductForm.append(quantityTextField);
    addProductForm.addCommand(addToBasketCommand);
    addProductForm.addCommand(cancelCommand);
    addProductForm.addCommand(exitCommand);
    addProductForm.setCommandListener(this);
    screen.setCurrent(addProductForm);
  }

  public void viewBasket(){

    screenString = "";
    for(int i=0;i<basketprices.size();i++){
      screenString += "Product: " + (String)
         basketinfo.elementAt(i);
      screenString += "\nPrice: €   " + (String)
         basketprices.elementAt(i);
    screenString += "\nQuantity: " + (String)
         basketquantity.elementAt(i) + "\n\n";
    }
    screenString+="Total Price: €  " +
```

```
      totalbasketprice + "\n\nTouch a new product
      tag to add it to the basket";
    screenText.setText(screenString);
    mainForm.addCommand(orderBasketCommand);
  }

  public void orderBasket(){

    /*
     * web service thread should be implemented
     * here
     */
  }

  public void run() {

    ndefMessage=false;
    for (int i = 0; i < target.length; i++) {

      if (target[i].hasTargetType
         (TargetType.NDEF_TAG)) {

        ndefMessage = true;
        try {

          String url = target[i].getUrl
            (Class.forName("javax.microedition.
            contactless.ndef.NDEFTagConnection"));
          conn = (NDEFTagConnection)
            Connector.open(url);
          NDEFMessage message = conn.readNDEF();
          if (message != null) {

            NDEFRecord[] records =
               message.getRecords();
            productid = new
               String(records[0].getPayload());
            productinfo = new
               String(records[1].getPayload());
            productprice = new
               String(records[2].getPayload());
            addProduct();
          } else {
            screenText.setText("There isn't any
               saved NDEF Messages in the target");
          }
        } catch (ContactlessException e) {
          screenText.setText("Error: " +
             e.getMessage());
        } catch (IOException e) {
```

```
            screenText.setText("Error: " +
               e.getMessage());
          } catch (Exception e) {
            screenText.setText("Error: " +
               e.getMessage());
          }
        } // end of if clause
      } // end of for loop

      if(!ndefMessage){
        screenText.setText("Target type is not an
           NDEF formatted tag");
      }

      closeConnection(conn);
    } // end of Read thread constructor
  } // end of MIDlet
```

(ii) *Understanding the code*

As described in the reader/writer mode example, a new `DiscoveryManager` object is created in the constructor and the target listener for NDEF_TAG (NFC Forum formatted data) is set up.

```
DiscoveryManager dm = DiscoveryManager.getInstance();
try {
  dm.addTargetListener(this, TargetType.NDEF_TAG);
}catch (ContactlessException ce) {
  screenText.setText("Error in adding listeners: " +
    ce.getMessage());
}
```

When the target is detected, properties of the detected target are saved to the `TargetProperties` object.

```
public void targetDetected(TargetProperties[] prop) {
  target = prop;
  Thread thread = new Thread(this);
  thread.start();
}
```

After an NFC tag has been discovered in the proximity and the records in the tag have been transferred to the mobile, the `addProduct` method is called. This method displays the product information that is transferred from the tag and asks the user to enter the desired quantity (top-right screen in Figure 5.16). The user can both input the quantity and confirm adding the product to the basket, or alternatively touch another NFC tag to cancel the operation.

```
public void addProduct(){
  quantityTextField = new TextField("Quantity:", "1", 2,
     TextField.NUMERIC);
  addProductForm = new Form("NFC Shopping");
```

```
  addProductForm.append("Product Info: " + productinfo +
     "\nPrice: €  " + productprice);
  addProductForm.append(quantityTextField);
  addProductForm.addCommand(addToBasketCommand);
  addProductForm.addCommand(cancelCommand);
  addProductForm.addCommand(exitCommand);
  addProductForm.setCommandListener(this);
  screen.setCurrent(addProductForm);
}
```

The `addToBasket` method adds the product to the basket which is confirmed by the `addProduct` method.

```
public void addToBasket(){
  mainForm.addCommand(viewBasketCommand);
  productquantity =
Integer.parseInt(quantityTextField.getString());
  basketinfo.addElement(productinfo);
  basketprices.addElement(productprice);
  basketquantity.addElement(Integer.toString(productquantity));
  basketids.addElement(productid);
  totalbasketprice += productquantity*
      Double.parseDouble(productprice);
  screenText.setText(productquantity + " items " + productinfo +
      " added to the shopping basket" + "\n\nTouch a new
product tag
      to add it to the basket");
  screen.setCurrent(mainForm);
}
```

In order to store the information of the products that are added to the basket, four vectors are created in the application:

- *basketPrices:* to keep each product's unit price;
- *basketInfo:* to keep each product's information;
- *basketQuantities:* to keep each product's ordered quantity;
- *basketIds:* to keep each product's identification.

These vectors, except the `basketIds`, are used to show information about the basket to the user. The `basketIds` are only needed to send the identication of the products in the basket to the supermarket's backend system. Also, the quantity of each product is both displayed to the user and sent to the supermarket. Moreover an integer variable, `totalBasketPrice`, keeps the total price of the basket.

The `viewBasket` method is used to display the current products in the basket together with their information, unit price, and quantity as seen in the bottom-right screen in Figure 5.16. The total price of the products in the current basket is displayed to the user. The user may order the basket using the "Order Basket" command or may still add other products to the basket by touching to the tag of a new product.

```
public void viewBasket(){
  screenString = "";
```

```
  for(int i=0; i<basketPrices.size(); i++){
  screenString += "Product: " +
(String)basketInfo.elementAt(i);
  screenString += "\nPrice: €   " +
(String)basketPrices.elementAt(i);
  screenString += "\nQuantity: " +
      (String)basketQuantities.elementAt(i) + "\n\n";
}
  screenString += "Total Price: €  " + totalBasketPrice +
     "\n\nTouch a new product tag to add it to the basket";
  screenText.setText(screenString);
  mainForm.addCommand(orderBasketCommand);
}
```

The `run` method starts when a thread is run in the current MIDlet. It is the main execution part of the NFC program. It simply makes the connection with the target using the `NDEFTagConnection`. The connection has already been described in Section 5.6.5, so we only briefly describe it here.

```
public void run() {
  ...
}
```

After the connection is established, the NDEF message in the tag is read and all records in the message are saved to the `records` array. Then the product identification, product information, and product price are saved to the corresponding variables. Note that the data in the tag are organized in the same order. The first record corresponds to product identification, the second record corresponds to product information, and the third record corresponds to product price. The last thing to do is to invoke the `addProduct` method in order to add related products to the basket.

The web service code in this case is left unimplemented for simplicity. We want to point out again that the purpose of this section is to show NFC in action, not to implement an industrial product.

5.10 Peer-to-Peer Mode Case Study: NFC Gossiping

In this application, the NFC gossiping case described at the end of Chapter 4 is implemented. Remember that the NFC gossiping case enables users to create new gossips or receive one from a friend. A user simply touches her mobile device to a friend's mobile device to exchange (send or receive) a gossip. Application screenshots are given in Figure 5.17.

When the user first runs the application, a main menu with two options is displayed: "Share a Gossip" and "Create New Gossip" as seen in the top-left screen in Figure 5.17. When the user wants to create a new gossip, a text box is displayed to the user for input (top-right screen in Figure 5.17). The user may save the gossip or cancel and return to the main screen. When the user enters the "Share a Gossip" menu from the main screen, all saved gossips are displayed on the screen. The user may share one by pushing the "Share" button (bottom-left and bottom-right screens).

Figure 5.17 NFC gossiping MIDlet screens.

When a remote device gets the gossip, it can save the gossip. We cannot give those screens here, since the emulator cannot manage a peer-to-peer communication. Thus those operations can be performed only on a real mobile phone.

(i) *Application source code*

```
import java.io.IOException;
import javax.microedition.io.Connector;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
import javax.microedition.rms.*;
import com.nokia.nfc.llcp.*;

public class NFCGossiping extends MIDlet
  implements CommandListener,
  LLCPConnectionListener, LLCPLinkListener {

  // LLCPManager object
  private LLCPManager llcpManager;
```

```
 // LLCP Connection objects
 private LLCPConnection inConnection;
 private LLCPConnection outConnection;

 /*
  * service id that to be used in
  * connection string
  */
private static final byte SID=11;

 /*
  * transport type to be used in
  * connection string
  */
 private static final byte
 TYPE=LLCPConnection.TYPE_2;

 /*
  *uid string that stores the hexadecimal string
  * of the UID of the remote LLCP device and will
  * be used in connection string
  */
 private String uid = "";

 /*
  *isReading string is used by the device that
  *is in read state. It replies with this string
  *in order to notify the sender device
  */
private String isReading = "-1";

private StringItem screenText = new
   StringItem(null,"");

/*
 *byte arrays that stores the gossip to be sent
 *and the gossip that is received
 */
private byte[] sendData = null,
   receiveData= null;

/*
 * determines if application is in reading or
 * writing state
 */
private boolean readState=true;

// record store that is used to store gossips
private RecordStore rs = null;
```

```
// string that will be used for the record store
private static final String
   RECORD_STORE = "GOSSIPS";

private Display screen;
private Form mainForm = null;
private List mainList = null, gossipList = null;
private TextBox newGossipTextBox = null;

private Command saveIncomingGossipCommand=new
   Command("Store Gossip", Command.SCREEN, 1);
private Command saveGossipCommand = new
   Command("Save Gossip", Command.SCREEN, 1);
private Command shareGossipCommand = new
   Command("Share Gossip", Command.SCREEN, 1);
private Command mainMenuCommand = new
   Command("Main Menu", Command.SCREEN, 1);
private Command cancelCommand = new
   Command("Cancel", Command.CANCEL,0);
private Command exitCommand = new
   Command("Exit", Command.EXIT, 0);

public NFCGossiping() throws IOException {

  /*
   *this function(implemented below) creates
   * the form and the list that will be used in
   * user interfaces
   */
  createMenu();
  screen = Display.getDisplay(this);

  // listeners added for the LLCPManager object
  llcpManager = LLCPManager.getInstance();
  llcpManager.addLinkListener(this);
  llcpManager.startListening(TYPE, SID, this);
}

protected void startApp() throws
   MIDletStateChangeException {

  displayMainMenu();
  /*
   *this function (implemented below) is called
   *at startup and displays mainList to user
   */
}

protected void pauseApp() {
```

```
}

protected void destroyApp(boolean unconditional) {

  /*
   *when application is destroyed, llcpmanager
   *stops listening and connections are closed
   */
  llcpManager.stopListening(TYPE, SID, this);
  llcpManager.removeLinkListener(this);
  closeLLCPConnection(inConnection);
  closeLLCPConnection(outConnection);
  inConnection = null;
  outConnection = null;
  notifyDestroyed();
}

public void commandAction(Command command,
   Displayable displayable) {

  String label = command.getLabel();
  if (label.equals("Exit")) {
    destroyApp(true);
  }else if(label.equals("Save Gossip")) {
    saveGossip();
  }else if (label.equals("Cancel")) {
    displayMainMenu();
  }else if (label.equals("Store Gossip")) {
    createNewGossip(new String(receiveData));
  }else if (label.equals("Main Menu")) {
    displayMainMenu();
  }else {
    List down = (List)screen.getCurrent();
    if(screen.getCurrent() == gossipList)
      sendMessage(getGossip(down.
         getSelectedIndex()+1));
    else{
      switch(down.getSelectedIndex()) {
        case 0:
          processGossips();
          break;

        case 1:
          createNewGossip("");
          break;
    }
  }
 }
}
```

```
public void displayMainMenu(){

  screen.setCurrent(mainList);
}

// a form and a list are created for later use
public void createMenu(){

  mainForm = new Form("NFC Gossiping");
  mainForm.append(screenText);
  mainForm.addCommand(mainMenuCommand);
  mainForm.addCommand(exitCommand);
  mainForm.setCommandListener(this);

  mainList = new List("NFC Gossiping",
     Choice.IMPLICIT);
  mainList.append("Share a Gossip", null);
  mainList.append("Create New Gossip", null);
  mainList.addCommand(exitCommand);
  mainList.setCommandListener(this);
}

/*
 *displays a text box to write a new gossip.
 *It is also used to edit and store an incoming
 *gossip using a String parameter newgossip
 */
public void createNewGossip(String newgossip) {

  newGossipTextBox = new TextBox("Text to
     write", newgossip, 255, TextField.ANY);
  newGossipTextBox.addCommand
     (saveGossipCommand);

newGossipTextBox.addCommand(cancelCommand);
    newGossipTextBox.setCommandListener(this);
    screen.setCurrent(newGossipTextBox);
}

public String getGossip(int index) {

  String gossip="";
  openRecordStore();
  try{

    byte[] recData = new
       byte[rs.getRecordSize(index)];
    int len = rs.getRecord(index, recData, 0);
    gossip= new String(recData, 0, len);
  }
```

```
    catch (Exception e){
      e.printStackTrace();
    }
    closeRecordStore();
    return gossip;
  }

  /*
   *Lists all records in the RecordStore.
   *Adds share option to share a gossip from list
   */
  public void processGossips() {

    openRecordStore();
    try{
      gossipList = new List("Gossips",
         Choice.IMPLICIT);
      for (int i=1; i <= rs.getNumRecords(); i++){
        byte[] recData = new
           byte[rs.getRecordSize(i)];
        int len = rs.getRecord(i, recData, 0);
          gossipList.append(new String(recData, 0,
             len), null);
      }
    }
    catch (Exception e){
      e.printStackTrace();
    }
    gossipList.setSelectCommand
       (shareGossipCommand);
    gossipList.addCommand(mainMenuCommand);
    gossipList.addCommand(exitCommand);
    gossipList.setCommandListener(this);
    screen.setCurrent(gossipList);
    closeRecordStore();
  }

  /*
   *Gets the gossip from the text box that is
   * created in createNewGossip method and saves
   * it to the record store.
   */
  public void saveGossip(){

    openRecordStore();
    byte[] record = newGossipTextBox.
       getString().getBytes();
    try{
       rs.addRecord(record, 0, record.length);
   }
```

```
 catch (Exception e){
   e.printStackTrace();
 }
 closeRecordStore();
 screenText.setText("Gossip is saved: " +
    newGossipTextBox.getString());
 screen.setCurrent(mainForm);
}

// opens connection to the RecordStore
public void openRecordStore(){
  try{
    rs = RecordStore.openRecordStore
        (RECORD_STORE, true );
  }
  catch (Exception e){
    e.printStackTrace();
  }
}

// closes connection to the RecordStore
public void closeRecordStore(){
  try{
    rs.closeRecordStore();
}

  catch (Exception e){
    e.printStackTrace();
  }
}

/*
 *this method saves gossip to the sendData byte
 *array that is wanted to be shared. It also
 *changes readState to false in order to
 * send the message. A new alert is created
 * warning user to touch to remote device
 */
public void sendMessage(String gossip) {

  readState = false;
  sendData = gossip.getBytes();

  Alert sendAlert = new Alert("Sending
     Gossip...", "Please Touch", null,
     AlertType.INFO);
  sendAlert.addCommand(cancelCommand);
  sendAlert.setTimeout(10000);
  sendAlert.setCommandListener(this);
  screen.setCurrent(sendAlert);
```

```
}

/*
 *when a device successfully sends a message,
 * it calls this method to change its state to
 * readState
 */
public void readMessage() {

  sendData = null;
  receiveData = null;
  readState = true;
}

/*
 *this method is called when a link is
 *established with the remote device.
 *Note that the device is not in read state
 */

public void linkEstablished(String uid) {

  receiveData = null;
  this.uid=uid;

  ProcessSend sendThread = new ProcessSend();
  new Thread(sendThread).start();
}

/*
 *method is called when a link is lost with the
 *remote device
 */
public void linkLost() {

  closeLLCPConnection(outConnection);
  closeLLCPConnection(inConnection);
  outConnection = null;
  inConnection = null;
}

/*
 *this method is called when the connection is
 *opened with the remote device. Note that the
 *device is in read state
 */

public void connectionOpened(LLCPConnection
   connection) {
```

```
    if (inConnection == null) {
     inConnection = connection;
     ProcessReceive receiveThread = new
        ProcessReceive();
     new Thread(receiveThread).start();
    }else
      closeLLCPConnection(connection);
  }

  // method that closes the LLCPConnection
  public void closeLLCPConnection(LLCPConnection
     connection) {

     if (connection != null) {
       try {
         connection.close();
       } catch (IOException e) {
         screenText.setText("Error closing
            connection : " + e.getMessage());
     }
   }
  }

  /*
   *when the connection opened with the
   *initiating device, the device in the reading
   *state calls this method.It sends "isReading"
   *(-1) string to the writer device and waits
   *for the actual incoming data.
   */

  public class ProcessReceive implements Runnable {

    public void run() {
      try {
        if(readState) {

inConnection.send(isReading.getBytes());
           byte[] data  = inConnection.receive();
           processReceivedData(data);
        }
    }
    catch (Exception e) {
      screenText.setText("Error handling
        incoming connection:" + e.getMessage());
    }
   }
  }

  /*
```

```
 *when data is received from the remote device,
 *it is processed in this method. If the device
 *is in read state,it prints the gossip to the
 *screen. If the device in not in read state,
 *(it means that the device is received "-1"
 *from remote device), then the message is sent
 */

private synchronized void
   processReceivedData(byte[] data) {

  if(receiveData == null) {
    receiveData = data;
  } else
    return;

  if(readState) {
    if(!new String(receiveData).
       equals(isReading)) {
      screenText.setText("Gossip is received: "
         + new String(receiveData) + "\n");
      mainForm.addCommand
         (saveIncomingGossipCommand);
      screen.setCurrent(mainForm);
  }
 } else {
   if(sendData != null) {
     screenText.setText("Gossip is successfully
        sent");
     screen.setCurrent(mainForm);
     readMessage();
  }
 }
}

/*
 *when the link is established with the remote
 *device, program calls this method. This
 *method opens a connection with the remote
 *device and sends the gossip data.
 */

public class ProcessSend implements Runnable {

   public void run() {
     try {
       outConnection = (LLCPConnection)
          Connector.open("nfc:llcp;type=" + TYPE
             + ";sid=" + SID + ";uid=" + uid);
       if(!readState) {
```

```
                outConnection.send(sendData);
                byte[] data = outConnection.receive();
                if(new String(data).equals(isReading))
                    processReceivedData(data);
            }
        }
        catch (IOException e) {
          screenText.setText("Error handling
             outgoing connection : " +
             e.getMessage());
        }
      }
     }
    }
```

(ii) *Explanation of the code*

Details of the application are now described. The application uses `RecordStore` to store gossips in a mobile device, however only the NFC part will be detailed here. For `RecordStore` APIs and descriptions, you can visit the related pages at http://java.sun.com/.

The following code, in the constructor part, creates a new instance of the LLCP manager. Then a link listener is added to be notified of LLCP link related events and a listener is added in order to start listening for connections. In order to start listening for connections, the transport type of the connection and the service identification need to be given as parameters. *TYPE* and *SID* are defined as static variables. Remember that there are two possible transportation types in LLCP communication.

```
llcpManager = LLCPManager.getInstance();
llcpManager.addLinkListener(this);
llcpManager.startListening(TYPE, SID, this);
```

NFC interaction starts when the user selects one of the existing gossips to be shared. In this case, the mobile device enters write state meaning that it is the one that will send the gossip. On the other hand, the NFC interaction can also be started by the remote device's initiation as well. In this case, the device is in read state and it is the one that will receive the gossip.

5.10.1.1 Writing Gossip

When a user wants to share a gossip, the `sendMessage` method is called with the parameter of a string. This parameter consists of the content of the gossip that will be shared. Shared gossip is obtained using the index value of the gossip list that is used to access to the record store. The index value of the gossip list is sent to the `getGossip` method and this method returns the corresponding gossip from the record store. In the `sendMessage` method, the `readState` of the device is set to false and the content of the gossip is saved to the `sendData` byte array as seen in the following code.

```
readState = false;
sendData = gossip.getBytes();
```

The device then starts waiting for a link to be established with the remote device. When the link is established, the `linkEstablished` method is called automatically. In this method, the `receiveData` byte array is set to null, in order to clear the previous gossip if there is any. Also, the `uid` of the remote LLCP device is saved to the `uid` string which will be used during the connection. Then a new thread is run in order to exchange the message as seen in the following code.

```
receiveData = null;
this.uid = uid;
ProcessSend sendThread = new ProcessSend();
new Thread(sendThread).start();
```

In the `ProcessSend` class, a new LLCP connection is opened with TYPE, SID and `uid` strings. Since the device's read state is false, it sends the `sendData` byte array to the remote device and waits for the incoming message. When the message from the remote device arrives, it checks if the received data equal the `isReading` string. Note that this string is initially set to "-1". If the received data equals to "-1", it calls the `processReceivedData` method as seen in the following code.

```
outConnection = (LLCPConnection)
Connector.open("nfc:llcp;type=" +
     TYPE + ";sid=" + SID + ";uid=" + uid);
if(!readState) {
  outConnection.send(sendData);
  byte[] data = outConnection.receive();
  if(new String(data).equals(isReading))
    processReceivedData(data);
}
```

When the `processReceivedData` method is called, since the device is in write state, it runs the `else` part of the code and sets the current screen's text. It also calls the `readMessage` method in order to switch to read state.

```
if(readState) {
  ...
} else {
  if(sendData != null) {
    screenText.setText("Gossip is successfully sent");
    screen.setCurrent(mainForm);
    readMessage();
  }
}
```

5.10.1.2 Reading Gossip

When the user wants to get a gossip, the remote device should be in write state and the user should touch her mobile device to the remote device. When this happens, the `connectionOpened` method is called automatically by the reader device (the device that is in read

state). In this method, a new thread from `ProcessReceive` class is created as seen in the following code.

```
if (inConnection == null) {
  inConnection = connection;
  ProcessReceive receiveThread = new ProcessReceive();
  new Thread(receiveThread).start();
}else
  closeLLCPConnection(connection);
```

The `ProcessReceive` class saves the incoming message to a byte array and sends "-1" to the remote device. In order to handle the saved incoming message, it sends the byte array to the `processReceivedData` method.

```
inConnection.send(isReading.getBytes());
byte[] data = inConnection.receive();
processReceivedData(data);
```

Since the device is in read state, it runs the `if` part of the code. In this part, it simply writes the incoming message to the screen. It also adds a command which enables the user to save the incoming gossip as seen in the following code.

```
if(readState) {
  if(!new String(receiveData).equals(isReading)) {
    screenText.setText("Gossip is received: " + new
       String(receiveData) + "\n");
    mainForm.addCommand(saveIncomingGossipCommand);
    screen.setCurrent(mainForm);
  }
} else {
  ...
}
```

When the user pushes the save command, the application calls the `createNewGossip` method. The application also displays the received message in a text box, so that the user can also edit it before saving.

As described earlier, this application enables the digital gossiping service so that people can disseminate gossips peer-to-peer. It can create, store, and share gossip messages. Note that advanced implementation is not discussed here since the idea behind these use cases is to show NFC in action rather than implementing an end user application.

5.11 Chapter Summary

This chapter described the development of NFC applications. Since Java is used widely and is well-known, we think that learning a new technology's application development with Java APIs helps readers understand it more clearly.

We have used Eclipse IDE as the basis for mobile programming. Series 40 Nokia 6212 NFC SDK is also used for NFC application development and the testing environment. Nokia 6212 NFC SDK is compatible with Eclipse IDE as well as Netbeans platform. However, there are

Table 5.17 JSR 257 API overview

Package	Description	Interfaces, Classes, Exceptions
javax.microedition.contactless	Provides common functionalities to all contactless targets such as discovering the target	TagConnection Interface TargetListener Interface TargetProperties Interface TransactionListener Interface DiscoveryManager Class TargetType Class ContactlessException
javax.microedition.contactless.ndef	Provides functionality to exchange NFC Forum formatted data with RFID tags	NDEFRecordListener Interface NDEFTagConnection Interface NDEFMessage Class NDEFRecord Class NDEFRecordType Class
javax.microedition.contactless.rf	Enables communication with RFID tags that contain non NFC Forum formatted data	PlainTagConnection Interface
javax.microedition.contactless.sc	Enables communication with ISO14443-4 smart cards	ISO14443Connection Interface

also other NFC development environments too. For Android devices, there is Android SDK and for Symbian^3 devices, Qt Connectivity API provides NFC development support. Other development environments exist as well.

First, introduction to J2ME is given, so that readers can gain a basic knowledge on mobile applications programming. Then development of NFC applications with two Java packages (JSR 257 and JSR 177 APIs) is described. JSR 257 mainly enables reader/writer mode programming; however some classes in the package are also used for accessing SEs. NFC related packages in JSR 257 API are summarized in Table 5.17.

With this API, an application can receive notifications when targets are discovered in proximity and can make a target specific connection. The `javax.microedition.contactless` package includes common classes and interfaces that can be used along with all target types. This package must be included in all applications that implement JSR 257. In addition to this package, a second package needs to be used based on the target type. The `javax.microedition.contactless.ndef` package enables mobile devices to exchange NDEF formatted data with contactless targets. The `javax.microedition.contactless.rf` package can be used for RFID targets that do not contain NFC Forum formatted data. Finally, the `javax.microedition.contactless.sc` package provides `ISO14443Connection` interface for accessing ISO 14443-4 compliant contactless smart card targets.

In order to listen to activity on SEs using JSR 257, applications need to implement the `TransactionListener` interface and also need to register a transaction listener using the `addTransactionListener` method of the `DiscoveryManager` class. The `externalReaderDetected` method is called automatically to notify the application that an event has happened on the SE.

On the other hand, JSR 177 API provides communication with SEs. The card emulation mode is developed mainly for applications that implement high security constraints. This API also enables security operations by providing digital certificates, digital signatures and cryptographic operations. Smart card communication with this API can be implemented either using APDU protocol or JavaCard RMI protocol.

For peer-to-peer mode programming, no standard APIs are provided, so that propriety packages need to be used. These propriety packages can be found in Nokia 6212 JSR 257 API Extensions. LLCP and NFCIP-1 are both covered for peer-to-peer mode programming.

Two use cases are implemented at the end of the chapter to show NFC in action and to illustrate the operating modes in a real life scenario.

Chapter Questions

1. What types of applications are needed for NFC?
2. What type of applications are needed for each operationg mode?
3. Which elements constitute the Java platform?
4. Define Application Management Software.
5. Describe MIDlet states, and methods that handle state changes. Which of these methods must be implemented in a MIDlet?
6. Which package enables a MIDlet to implement user interfaces?
7. Which files are contained in a MIDlet suite?
8. What are the attributes that must be included in JAD and JAR manifest files?
9. Define push registry. How can a MIDlet register a push record?
10. Give an example of static push registry entry.
11. List the JSRs that are developed for NFC programming.
12. Which target types are supported in Contactless Communication API?
13. Give the processes of a reader/writer mode application that reads data from an NFC tag.
14. How does a reader/writer mode application discover a target and how does it receive the notification using JSR 257 API?
15. What does an NDEF_TAG target type mean?
16. In order to implement a reader/writer mode application that communicates with an NDEF_TAG target type, which packages can be used from JSR 257?
17. Which class represents NDEF message in JSR 257?
18. List record type name formats and define three of them.
19. How can push registry be used in reader/writer mode?
20. Give an example of push registry connection entry which enables an application to run upon discovering the tag.
21. Which protocols can implement a peer-to-peer mode application?
22. Give examples of NFCIP and LLCP connection strings.
23. Describe the communication of two devices in NFCIP connection.
24. Describe the communication of two devices in LLCP connection.
25. What are the ways of accessing SEs with an application?
26. Which essential functions are provided by JSR 177?
27. Give examples of APDU and JCRMI connection strings.

References

[1] Android SDK, http://developer.android.com/sdk/ (accessed 10 July 2011).
[2] Qt SDK, http://qt.nokia.com/ (accessed 10 July 2011).
[3] Java Platform, Standard Edition (Java SE), http://www.oracle.com/technetwork/java/javase/ (accessed 10 July 2011).
[4] Java Platform, Enterprise Edition (Java EE), http://www.oracle.com/technetwork/java/javaee/ (accessed 10 July 2011).
[5] Java Platform, Micro Edition (Java ME), http://www.oracle.com/technetwork/java/javame/ (accessed 10 July 2011).
[6] Java Community Process (JCP), http://jcp.org/ (accessed 10 July 2011).
[7] Java Specification Requests (JSR), http://www.jcp.org/en/jsr/ (accessed 10 July 2011).
[8] Series 40 Nokia 6212 NFC SDK, http://www.developer.nokia.com/info/sw.nokia.com/id/5bcaee40-d2b2-4595-b5b5-4833d6a4cda1/S40_Nokia_6212_NFC_SDK.html (accessed 10 July 2011).
[9] The Eclipse Foundation open source community, http://www.eclipse.org/ (accessed 10 July 2011).
[10] Eclipse, http://eclipseme.org/ (accessed 10 July 2011).
[11] Mobile Information Device Profile (MIDP); JSR 118, http://java.sun.com/products/midp/ (accessed 10 July 2011).
[12] JSR 257: Contactless Communication API, http://www.jcp.org/en/jsr/detail?id=257 (accessed 10 July 2011).
[13] JSR 177: Security and Trust Services API for J2METM, http://jcp.org/en/jsr/detail?id=177 (accessed 10 July 2011).
[14] Nokia 6212 Classic JSR-257 Implementation, can be found in Series 40 Nokia 6212 NFC SDK, http://www.developer.nokia.com/info/sw.nokia.com/id/5bcaee40-d2b2-4595-b5b5-4833d6a4cda1/S40_Nokia_6212_NFC_SDK.html (accessed 10 July 2011).

6

NFC Security and Privacy

Although a mobile phone is almost identical to a Personal Computer (PC) in technical terms, it is different in that it is more a personal item and carried by people at all times. Users generally believe that their mobile phones are an important part of their lives and they generally have them under physical surveillance. However, our mobile phone is always subject to physical attacks such as theft, and technical wireless attacks using Bluetooth or Wi-Fi communication technologies. The integrated NFC capability imposes some additional risks to mobile phones as well.

This chapter contains detailed information regarding security and privacy in NFC technology. We begin by introducing general information and definitions on security, its principles, tools, and mechanisms and then continue by specifying details on NFC security, privacy issues, problems, and their solutions. Considering that the reader may not have a detailed knowledge on the subject, we especially try to give introductory knowledge on security and privacy.

6.1 Security in General

Security is the degree of protection against an intentional or accidental misuse or action. If the malicious activity somehow makes use of a weakness, it may cause harm to the system. The malicious activity may be triggered by an adversary whose purpose might be to gain some benefit, to hamper the proper functioning of the attacked system, or to cause the system to malfunction or to leak some information.

Secure access to a system requires applying basic processes of authentication, authorization, and non-repudiation. Authentication assures that the person who is attempting to access a system is really who the person claims to be. In the case of a customer trying to access an account using a bank's website for example, authentication is the process for proving the identity of the customer. Authentication essentially validates – or invalidates – the claimed identity of the user.

Authorization is permitting some set of actions to be carried out by the authenticated user. Hence, an authenticated user may perform any one from a set of allowed actions but may not perform any action that she is not allowed.

Non-repudiation is a stronger version of authentication. Considering the previous bank example, a web server may authenticate the customer using a password that is only known by

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.
 Published 2012 by John Wiley & Sons, Ltd.

the user and the server. However, the web server cannot prove the identity of the user unless the authentication data are known only by the user. The distinction between authentication and non-repudiation will be clarified later in this chapter.

Security mainly deals with considerations such as confidentiality (secrecy) as protection of data against unauthorized users and usage, integrity as protection of data against modification by the unauthorized users, and availability to enable access to the authorized users any time and in any condition.

Confidentiality is about keeping the content of the data accessible to only the authorized users and in an authorized manner. There are two subcategories of confidentiality, namely, secrecy and privacy. Secrecy assures that private information is not accessible, is not made available or disclosed to an unauthorized user. After an authorized user accesses the secret data, privacy is assuring that the data are not used for an unintended purpose by the owner. Privacy implicitly restricts subsequent transfer of data by the authorized user to an unauthorized user. Secrecy is about accessing data by authorized people only whereas privacy is about the misuse of data by authorized users. Secrecy is enabling access to the data by authenticated people only. Privacy is enabling usage of data for the intended purpose only once the data are accessed by an authenticated user, thus preventing the data from being used for unintended purposes.

Integrity is being sure that data are not modified or deleted by an unauthorized user or in an unintended manner. When talking about data on some storage device, integrity may be compromised by either technical problems or as a result of a malicious or accidental activity. Knowing that the technical problems are not an issue for security, malicious or accidental activities of all users must be appropriately dealt with. Integrity of data being transferred between a sender and a receiver on the Internet is also subject to the same problem. Data may corrupt during transfer, so that the sent and received messages may not match or the sent message may not even arrive at the receiver. When the reason for the problem is some malicious activity, security mechanisms are responsible for dealing with the issue.

Based on the described security concepts, providing security in the case of NFC is critical because NFC is mostly used for the purpose of payment and ticketing.

6.1.1 Why is Security Important?

When a user is eager to use a service such as an information management tool, her main purpose is to be free from problems associated with the use of the service. Her next motivation might be in getting as much performance as possible. Functionality and performance together is the sum of the acquired service.

Two negative parameters that potentially affect the amount of service usage are technical deficiencies and security related problems. The difference between security and technical deficiencies is that security takes into account the actions of people and smart machines attempting to cause destruction, whereas technical problems are not triggered by security problems.

Security became an important issue in the last decade for the following reasons:

- From the hacker's point of view, there are now more financial opportunities. A hacker can earn money from malicious activities.
- From the technical point of view:

  - The number of internet users is increasing exponentially; hence the media for malicious actions becomes more convenient. One hacker can try the same method to hack many potential victims.
  - It is hard to embed security measures in the new applications due to the high development cost in the IT area and the increasing requirement of new applications. Also, designing the rest of the application is much easier than embedding the security features. This is because more time and more money is required to design embedded security architecture.
- From the developer's point of view, potential buyers tend to appreciate functionality much more than security. Although noticing security features requires expertise, users easily notice other functionalities such as a user interface.

Security is an important issue in the NFC ecosystem as well. The major reasons may be listed as:

- NFC is a hot technology and integrated with mobile phones.
- Everybody owns a mobile, and people are concerned about things related to themselves.
- NFC is heavily pushed by service providers.
- NFC potentially has a big financial market, which is a demanding reason for the hackers.

6.1.2 Primary Goals of Security Measures

Each system has its own security requirements. Some require keeping the information available to only one or more people, while others require keeping the contents unchanged by illegal parties. In this section we will introduce important security requirements that are used in almost all systems. Not all systems require all security measures. Some only require authentication, while others require integrity instead. We will first list all measures, and use the definitions in further subsections.

6.1.2.1 Secrecy/Confidentiality

Secrecy is assuring that information is accessible only to the authorized party (person, process, or device). Secrecy requires hiding the content of the information, traditionally by encryption, so that only authorized parties can access it, with the help of the secret key.

6.1.2.2 Authentication

Authentication is confirming the identity of a person, a process, or a device. There are several ways to satisfy authentication. For example, assume that you have invited a friend for a meal, but you do not want to spoil your night by answering the door to somebody else when the door is knocked. For this reason, you may agree on a method for knocking the door so that your friend will knock your door three times as an example. Hence, the information that you have exchanged with your friend is the key that nobody else knows. Since you and your friend are the only pair of people knowing the secret information for authentication, you will be sure whether the person knocking your door is your friend or not. What we have used here is simply

a symmetric cipher which will be described in Section 6.2; the secret information is known by both partners, and by nobody else.

(i) *Authentication means*

Many methods can be used for satisfying authentication such as requesting a password, touching an RFID identity card to an RFID card reader, pressing fingers onto a scanner where the fingerprint is checked from a database, or applying some handwriting onto the same scanner. Different means of authentication can be classified as follows:

- Something she knows: password.
- Something she owns: smart card, OTP token, physical key or NFC handset.
- Something she is: physiological/static biometrics like fingerprint, voice, retina or iris.
- Something she does: behavioral/dynamic biometrics like handwriting, typing or walking rhythm.

(ii) *Current authentication schemes*

There is always a struggle between the hackers and the security providers. The aims of the security providers are to create security mechanisms to enable a strong degree of protection and enough level of functionality at the same time; the aim of the hackers is to pass over those mechanisms.

There may be trade-offs among different security mechanisms. For example, digital signature is currently the strongest way of providing authentication, but it is more costly to use a digital signature than a one-time password. Banks do not insist upon but only support the use of a digital signature; however, they insist on the use of a one-time password. The following are the currently used options for authentication by banks:

- *One-time password via SMS*: The bank server sends a password to the mobile phone that is registered previously. The user receives the password using her mobile phone and types into the web page. Hence, it is assumed that the current user is actually the intended user.
- *One-time password generator token*: The user invokes the token that she has received previously from the bank to generate a one-time password, and types it into the web page. Hence, it is assumed that the current user is actually the intended one.
- *One-time password generator software*: The case is similar to the previous one, only with the difference that the one-time password generator application is installed and invoked on the mobile phone instead of a token.
- *Mobile signature*: The user receives a digital signature from a trusted certificate authority, so that she can sign any transaction that she requests. Mobile signature also provides security mechanisms other than authentication as well.

(iii) *Biometric authentication*

A biometric system authenticates a user based on physical (biometric) features. The metrics may include static features such as a retina, a fingerprint, or dynamic features such as voice pattern. It is assumed that using a retina scanner or fingerprint analyzer provides a higher level of security than using a password. However, a secure password which is sufficiently long and complex that is updated frequently provides enough security. When compared with cryptography based schemes, biometric methods are not yet mature enough and are still too costly. Some of the reasons why expensive mechanisms are still developed are as follows:

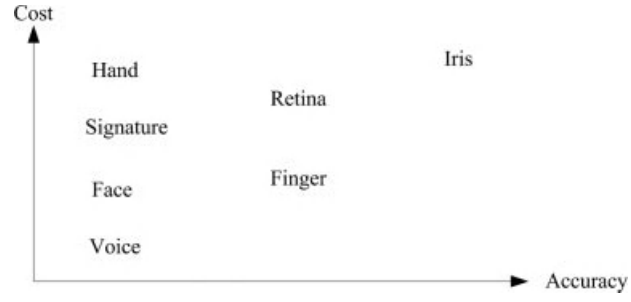


Figure 6.1 Relative cost and accuracy of a biometric system.

- It is hard to trust to the user because the user may insert some relevant data during creation of the password by only making some modification to, say, a birth date.
- It is hard to trust to the user even her password is complex because the probability that the user writes the password somewhere is very high.
- The layering principle dictates that we should insert different security check mechanisms on different points. A biometric password is something more different than a normal password, and checking a biometric information at one check point after a password at an earlier point provides better security than two password checking mechanisms.
- As the computing capability increases, hackers will be able to crack complex passwords in just a few seconds that a user can handle in her brain. So it is good to have a replacement for that future time.
- As the usage of biometric passwords increases, studies on improving their success will increase and they will become more successful.

The biometric system may be based on static as well as dynamic properties. Some static methodology examples are facial characteristics, fingerprint, DNA, hand geometry, iris and retinal pattern. Some dynamic methodology examples are typing rhythm, voice and signature. Figure 6.1 gives an indication of the relative cost and accuracy of a biometric system.

(iv) *How do biometric systems work?*

Let us consider photography. Old fashioned classic analog cameras used analog photographic films and stored exact information. Then digital photography appeared. Not all digital cameras use the same resolution though. Consider three digital cameras with 1, 2 and 3 mega pixels. When each one is used to record the same frame, they use different file sizes because of the different resolution they support. Even when they record the same area, the file size and content they generate will be different. Hence, they do not store the actual information, but some representation of it. Their color range might be different as well. If they use 16, 24, and 32 bit color schemes, they store different color information for each bit. Hence, the representation of a real pixel in the world will be different for each camera. When you compare three files of three pictures taken by different cameras, it is hard to decide whether the pictures taken represent the same real frame or not.

A biometric system works similarly. The biometric information stored in the enrollment phase is actually a digital representation of the real data such as information about the iris, hand geometry or fingerprint. Obviously, how to represent the actual data affects the success ratio of the overall algorithms. Hence, different proposed algorithms have different success ratios. The system first stores a digital representation of the actual data (enrollment phase) to the database, then the digital representation of the current user is compared with the stored data (verification phase) in the database. The steps are analogous to the password based authentication mechanism. The difference is that in password based authentication, there is no difference between the real password and the representation of it, since the process is performed immediately using the symbols. However, in the biometric system there may be even differences between the different recordings of the same information such as geometry information of the same hand due to the change in angle and light conditions or some differences in the algorithm. Hence, the similarity is determined by using some threshold; if the similarity is above a certain rate the user is authenticated, otherwise the user is rejected.

6.1.2.3 Mutual Authentication

Authentication only satisfies assurance of one party to the authenticator party. Consider a web page that requests a user's identification and password to enable further functionality. When the user enters this information, the server will authenticate, but the user will not be able to understand whether the web page belongs to the actual bank server or a hacker's computer that imitates a bank server application. As a result, the user cannot be sure whether the secret information has been sent to the bank server or a hacking website instead. This problem may be solved with a mutual authentication technique, in which both parties authenticate each other. In the case where phishing is a possibility, mutual authentication is essential.

6.1.2.4 Authorization

After authentication is provided, authorization allows different actions on the object (file, application, or machine) by the subject (user). For example, a professor must be authenticated before any insert, update or delete operations on student grades. Each student is authorized to view her grade. In contrast, no student's grade is visible to any other student. Similarly, one professor may be able to work on the grades of her courses, but not anyone else's. The authorization rights may even vary based on conditions, such as time. For example, the professor may not be able to modify the grades after a certain time such as the end of the semester.

6.1.2.5 Non-Repudiation

Non-repudiation is a stronger requirement than authentication. Authentication is based on some secret information, a key or a password (see Figure 6.2). When one of the partners uses the secret information, the counterparty will be able to authenticate the other. One limitation of authentication is that no party can prove the counterparty against a third party, which is satisfied only by non-repudiation techniques. Non-repudiation is harder to be provided. A

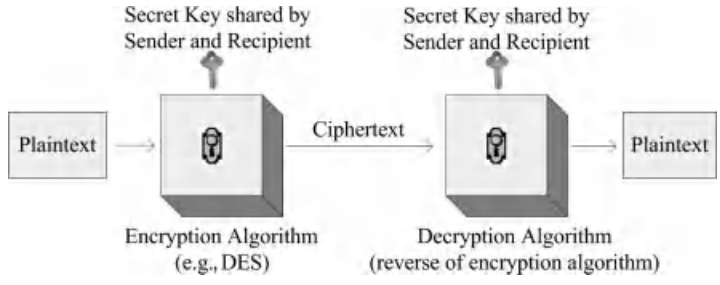


Figure 6.2 Authentication scheme.

secret key shared among a pair of partners is not enough, since the third party cannot be sure about which party has used the key. One additional requirement for non-repudiation is that the third party, in other words the referee, needs to know the key or some important information about it (see Figure 6.3).

Let us consider an example. A bank has a website, which requires users to enter their passwords. In this case, the password of the user is known both by the bank as well as the user. When the user is prompted to enter the password and enters the password correctly, the bank authenticates the user. Assume that the user has ordered a money transfer using the website and the transfer is performed by the bank web server. The bank is sure that the user has requested the transfer, since only the user and the bank know the password. However, the bank cannot prove this, because the bank itself also knows the key. A staff of the bank who has access to the password might have used the key. Hence, a totally different type of key design, generation, management and usage system are required. It should be only the user who knows the key. The only thing that the bank as well as the referee could do is to verify the key. It is an asymmetric or contemporary or public key that is required here. In symmetric cryptography, only the user owns the private key. The referee owns the public key that can be used to verify the usage of the private key. The owner of the public key (the bank and the referee) cannot access the account without using the private key. So that when a user has performed a money transfer in this case, she cannot deny the operation.

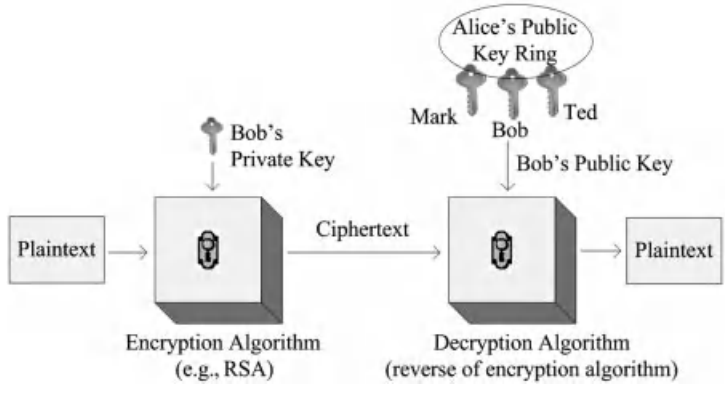


Figure 6.3 Non-repudiation scheme.

Table 6.1 Downtime measures

% Availability	9s	Downtime
90.0000	1	36.5 d
99.0000	2	3.65 d
99.9000	3	8.76 h
99.9900	4	52 min
99.9990	5	5 min
99.9999	6	31 s

6.1.2.6 Availability

Availability is about ensuring that the system responds correctly and completely to the requests of the authorized user at any given time. In a technical sense, availability is the ratio of the time that a system performs as expected to the total time. Consider a system that is used 1 h per day; this would have an availability of 23/24. Typical availability values are specified as a decimal (such as 0.9583). In order to express availability of applications, a metric known as nines is used. The nines correspond to the number of nines in the availability percentage. For example, "four nines" equals 0.9999 (or 99.99%) availability (see Table 6.1). Downtime is another notation, which shows the non-available period of the system in nines notation.

$$A = \text{MTBF}/(\text{MTBF} + \text{MTTR})$$

$$\text{MTBF} = \text{Mean Time Between Failures}$$

$$\text{MTTR} = \text{Mean Time To Repair}$$

6.1.2.7 Data Integrity

Integrity is the assurance that the received information is exactly the same as the information sent. This means that the information has not been accidentally or maliciously modified or deleted during any operation such as storage or transfer. Data that have integrity are identical to the original.

6.1.2.8 Accountability

The performed actions related to security must be able to traced back to the originator whenever it is required by the security officer or the application. Accountability actually brings other requirements such as non-repudiation and activity logging.

6.1.3 Vulnerability, Threat, Attack, and Risk

6.1.3.1 Vulnerability

Vulnerability is a security weakness which may be exploited by an attacker. For example, a system might be vulnerable to unauthorized data manipulation because it does not verify

Parameters	Availability	Confidentiality	Integrity
Hardware	Not available when the hardware or its components are stolen or broken		
Software	Not available when the software is modified or deleted so that it cannot function at all	Confidentiality is violated when software is modified to help hackers	Functions improperly generate unexpected results when the software is modified
Data	Not available when the data are modified or deleted	Confidentiality is violated when data are read by unauthorized parties	False data results in inappropriate results when existing data are modified, deleted or even fake data are generated
Packages	Not available when the packages are modified or deleted on the way to the receiver	Confidentiality is violated when packages are read by unauthorized parties	False data results in inappropriate results when a package content is a modified, deleted or even a fake package is generated

Figure 6.4 Threats analysis.

the user's identity properly before allowing access. The weakness may be a design, or an implementation error, or a deficiency in some procedure.

Vulnerability is the combination of three elements: a system flaw, access capability of the attacker in order to make use of the flaw, and attacker capability to exploit the flaw. If vulnerability exists in a system, then the attacker may initiate an attack to make use of the vulnerability. So, the risk is the (possible) harm.

6.1.3.2 Threat

A possible danger that has the potential to cause an unfair benefit to the unauthorized people or to cause harm by exploiting vulnerability is called a threat. Threats may be either intentional or unintentional. Intentional threats aim to purposefully gain unfair benefit or cause harm; unintentional threats generally occur accidentally and do not purposefully intend to cause harm. Examples of unintentional threats are human errors such as design of software, hardware and data entry errors, and technical errors such as computer system failures, and fires. Threats that damage availability, confidentiality and integrity are categorized in Figure 6.4.

6.1.3.3 Attack

An attack is an intentional attempt of intruders to read, modify, delete, disable or gain unauthorized access to information. An attack may be classified further as active or passive. Actions

attempting to read information without affecting the system's resources are called passive attacks whereas actions altering the system's resources are called active attacks.

Passive attacks are against confidentiality and common examples are covert channel, traffic flow analysis, reconnaissance, packet sniffing, eavesdropping, backdoor, and key logging. It is very difficult to detect a passive attack since there is no change in the data during the attack. Encryption is an obvious choice to prevent passive attacks. The emphasis in dealing with passive attacks is on prevention, without detection.

Active attacks are against integrity and common examples are masquerading, in other words spoofing, Denial of Service (DoS) attack, Distributed DoS (DDoS) attack, reflection attack, amplification attack, Man in the Middle (MIM) attack, replay attack, modification of message and salami attack. Attacks may be initiated by an insider who is an authorized user and can access system resources. However after she accesses the data, she uses them in a way not approved by those who granted the authorization. Also an attack may be initiated by an outsider who is an unauthorized or illegitimate person.

Various types of attacks exist. These attack types are developed by the hackers in parallel to the technology. Some of the major attacks are as follows:

(i) *Eavesdropping*

Eavesdropping, in other words sniffing, is the act of unauthorized real time secret interception of the private communication. The term eavesdrop derives from the practice of listening to a conversation within a physical vicinity.

(ii) *Social engineering*

Social engineering is manipulating people; the hacker is unauthorized and tries to get information. Some techniques involve fooling while others include harsh methods such as threatening the user.

(iii) *Phishing*

Phishing is the process of trying to acquire information such as usernames or passwords by masquerading. Phishing typically uses e-mail or instant messaging and it often convinces users to enter personal information such as a password to a fake website. Phishing can be classified as an example of the social engineering technique as well.

(iv) *Leakage*

Leakage, in other words covert channel, is collecting a small amount of data from each package and then combining these tiny pieces in a long run. Then, the collected data can be used by the intruder for some purpose which is against confidentiality. For example, consider a MNO that gives out bonus points based on the amount of talking period, which can further be used to get free items from a shop. Your friend is a customer of that MNO and allows you to use the bonus points of her mobile, and gives your name and other information to the MNO. Thereafter you are ready to earn and use the bonus points for each minutes that your friend spends by her mobile. After each monthly invoice of X euro, your friend receives X bonus points. The problem here is the leakage of information as you get your friend's bonus points and invoice, you have the information about your friend's talking period.

(v) *MIM attack*

By standing between two legitimate users of the system, an attacker can receive the package from the sender (see Figure 6.5). Then she transfers it to the receiver, and makes use of the information accessed. Assume that two ladies are talking to each other. When

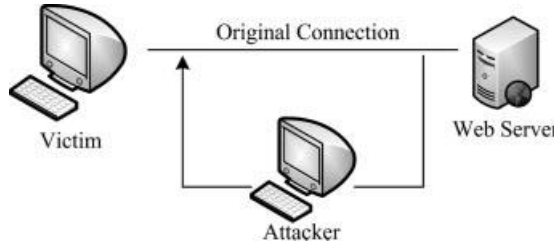


Figure 6.5 Man in the middle attack.

they are talking face to face, they are both sure about the identity of each other. If we insert another communication media for chatting such as a phone line, the speech data are possibly converted to digital format and carried by some mediator servers. In this case it is possible that a third party receives the data from the sender's server and sends it to the receiver's server, and both servers may be unaware of the server in the middle. This scenario is actually a case study of a well-known algorithm, called MIM attack. New prevention mechanisms are developed against MIM attacks; however, improved attacking mechanisms are then generated by the hackers.

(vi) *Replay attack*

A replay attack is when a valid data transmission is repeated maliciously after the original transmission occurs. This is carried out either by the originator or by an adversary who intercepts the data and retransmits them. An example of a dangerous replay attack is a money transfer transaction between two accounts.

(vii) *Relay attack*

A relay attack is about relaying a message that is to be exchanged between a sender and a receiver. It is related to MIM and replay attacks. The main way to avoid a relay attack is differentiating data exchange sessions. Several prevention examples exist:

- The sender inserts a time stamp to each message sent. The receiver notices the relayed message, since the time of reception of the message and the time stamp embedded in the message will be different. Time synchronization should be achieved using a secure protocol.
- The receiver sends a one-time token to the sender, and the legitimate sender uses a pre-set algorithm to transform the token and send the result to the receiver such as computing a hash function of the token.

(viii) *Pharming*

Pharming aims to redirect a website's traffic to a fake server. Some possible ways to perform a pharming attack are:

- Information can be illegally modified on a victim's computer. When the user tries to access the server, she is redirected to the fake server. A physical attempt to enter an office, access the computer and modify the address on a shortcut is one example. When you modify the address behind a bank website in the Favourites is one option. When the user selects the bank from the Favourites, the fake server is accessed. Another option is changing the operating system's local DNS entries. Moreover, malicious software can perform these actions automatically.

- Network components may be modified to redirect an incoming packet to the fake server. In this case there will be no trace on the victim's computer. Changing the routing table of a router on the way between the victim's computer and the actual web server is enough for this purpose.

(ix) *Spoofing attack*

A spoofing attack is masquerading as another, and therefore gaining an illegitimate advantage. Some examples are:

- *URL spoofing:* A legitimate web page is reproduced which seems identical to the original one. Users may be fooled and think that they are connected to a trusted site. This way hackers may receive usernames and passwords.
- *SMS spoofing:* Sending an SMS by masquerading as someone else.
- *E-mail spoofing:* Sending an e-mail by masquerading as someone else.
- *IP address spoofing:* Sending a packet by masquerading with a false IP address.

(x) *DoS attack*

A DoS attack is making a resource unavailable to its intended users. One common method involves accumulating vast amount of requests to the target machine, so that it cannot respond to legitimate traffic any more or responds so slowly that the service cannot be considered as effectively available. DoS attacks are implemented in order to force the targeted computer to shut down or to exhaust its resources, so that it can no longer provide its intended services.

6.1.3.4 Risk

Threat is a possible danger that can cause harm by exploiting vulnerability, whereas the potential harm that may arise after the realization of some threat is called the risk. A threat may have a low probability of occurrence, but may have serious consequences when it happens. For example, the possibility of an earthquake may be very low, but in the case of an automobile factory, an earthquake might completely destroy the factory and the company might even go bankrupt if it does not take measures prior to the hazard.

We cannot accept a big risk even if the probability of the realization of the risk is small. We have to enhance security up to a point where our risks are under control and tolerable. The solution is risk management.

Risk is defined as a function of three variables:

- The potential impact of the possible harm that the threat will cause (cost).
- The probability of existence of an attack (P_{attack}).
- The probability of success of an attack ($P_{success}$).

So that:

$$\text{Risk} = (\text{Cost})^*(P_{attack})^*(P_{success})$$

We can understand from the equation that the probability of the risk remains high when the cost is high even if the probabilities are low. A similar result may be calculated when the cost is moderate, but the probabilities are high. In any case, the risk manager should consider all components of the equation.

Considering a virus attack possibility, assume that the following values are given:

- The expected cost of the virus to the company is € 10 000.
- The probability of a virus attack is 40%.
- The probability that the attack will be successful is 80%.

We can calculate risk as:

$$\text{Risk} = 10\,000^{*}0.40^{*}0.80 = € 3200$$

To summarize, we can use a single example to show the relationship between vulnerability, threat, attack, and risk. When a window is left open in a house, the open window will be the vulnerability of the house. Considering that thieves may be trying to break into houses in the region, the threat is that thieves may enter the house. When a specific thief is trying to enter the house, the action of the thief is the attack. If the thief manages to enter the house, the risk is that all of the jewellery may be stolen.

6.1.4 Principles of Security

6.1.4.1 Compromise in Security: Security vs. Functionality and Performance

Functionality is what a product offers or provides to its user. Performance is the amount of the offered work. Functionality and performance together is the sum of what and how much an information system performs for the user or potential customer of the offered product.

Security is the degree of protection against malicious actions that potentially cause harm in some way. A secure system is one which performs the functionality of the system and prevents any actions of misuse at the same time.

A service should have a balance between ease of use and security. As a simple example, when a password is easy to remember, the user is happy since she does not spend too much effort when trying to remember it. However, this password does not provide much security. The security is improved when the password becomes more complex, but in that case the password becomes a problem for the user since remembering it becomes harder.

(i) *User's point of view for installing security components*

When definitions of functionality, performance, and security are evaluated together, it seems that security helps improve functionality and performance. However, it is not so trivial. Consider using antivirus software. If a virus is spread to your computer, the computer will be degraded. This means it will show less performance, less functionality or even no performance at all.

However, there are drawbacks in using antivirus software. The first one is the cost of the software. The next is the decrease in computer performance, because of the processor and memory usage of the software during execution. Hence, antivirus software is used to increase functionality and performance, but causes a decrease in both. What are the other alternatives?

- If you do not use antivirus software:
  - No successful attack occurs, which is the best case.
  - Some successful attack occurs by the virus. As a result, you will lose performance, useful data, installed applications or even your computer.

- If you use antivirus software:
  - No attack occurs but you have to purchase the antivirus software and your computer loses performance because of the execution of the software.
  - Some successful attack occurs by the virus, however the virus will be unsuccessful. You will save functionality, performance, data and potentially money since you have saved data, applications, the operating system and the computer itself.

(ii) *Developer's point of view for installing security components*

When a new application is developed, the company always tries to (and normally has to) present the final product as early as possible for marketing purposes. There is always the possibility that a competitor is about to offer a similar product to the market. But, embedding security components in a product is always a tiresome and costly process. Hence, when the decision is left to the company, it seems to postpone most of the security measurements to a later time which results in high risks.

From a broader viewpoint, the trade-off is between the level of security, the speed of the execution of the applications, the functionality of the system, and the additional cost of the security measures.

6.1.4.2 Is Perfect Security Possible?

Perfection is impossible in security. No system is 100% secure. It is necessary to achieve a balance between increasing the security of the system and the functionality of that system. Knowing that the user notices functionality more than security, it is hard to build a very secure system.

We can say that a system is completely secure, if all the computing power in the world is used forever, the system can not be broken. A system is computationally secure, even when limited time (but very high) and limited computer power (but very high) are used, the system cannot be broken. In this case we can also say that it is infeasible to break the system, since too much time would be used by the intruder. Consider attacking the security mechanism of a credit card. If the security system is computationally secure, it is infeasible to try to break the system because even if the card security is broken, it will take several centuries and the credit card will no longer be in use.

After accepting the impossibility of perfect security as a truth, it is still necessary to keep the security level of a system high. With the introduction of contemporary cryptosystems and proper implementations subsequently, it is possible to build systems which are sufficiently secure.

The security of a system relies on two parameters:

- The strength of the cryptographic algorithms which needs to be studied by the academicians and the security professionals, but not by the people who benefit from the provided security.
- We cannot ignore the role of the user. She is responsible for keeping the password or the key assigned to her. This is not a trivial matter. If the password is easy to remember, it is also easy for the hacker to find it in just a few seconds using publicly available programs from the Internet. Increasing the complexity of the password seems to be a solution to overcome the

hackers. However, embedding some complex letters into the password such as "?-;,*/&%+" makes the password hard to remember, and so it may be written somewhere by the user, which enables the hacker to find it.

Therefore, when you see on a bank's website the phrase "%100 secure", what they claim to offer is the cryptographic algorithm only. Total security always depends on the counterpart such as the building and keeping the password secret, so that a hacker or a hacking tool cannot reach it or generate it.

6.1.4.3 Hackers are Always One Step Ahead

Antivirus software prevents the access of new viruses only after the virus has been generated by a hacker. The antivirus is generated after the related virus emerges. Hackers are always one step ahead, which seems unfair but is unfortunately true. There are important reasons for this. Using the analogy of a castle:

- There is more than one weak wall of a castle and it is hard to fix them all. Even if we did, the earlier relatively strong parts become the weaker parts from then on. We need to strengthen those parts also. This will put us into an unending effort sequence.
- It is hard to guess how the hackers will attack the weaker parts. There are so many options that it is really not easy to technically cater for each possibility.
- Cost is also a consideration. It is costly to try to fix many parts of the wall.

6.1.4.4 Keep it Simple

Consider security mechanisms built for a large organization. The campus is huge, there are thousands of personnel and visitors, and it is not easy to provide security mechanisms. Immediate options that the security manager might consider would be to use lots of security staff, check points, access cards, passwords, comprehensive strict rules, and supporting booklets including directives to the security staff and personnel. Obviously, this might result in failure as when you embed high complexity to a system, you may also cause the system not to work properly. Security people for example, will find it hard to check details of rules from booklets; they will have disagreements on applying strict rules. Employees may also feel very uncomfortable with those rules and may complain.

Why complexity weakens security:

- The security discipline is inherently complex and there are many parameters to think about. If procedures become complex, then it is harder to discover their weaknesses.
- If security procedures are complex, it is harder to verify their correctness and completeness; hence there is a high probability that some gaps are already missed.
- When a problem is observed within the security mechanism, it is harder to identify the problem if the plan is complex.

The solution is simplicity. Simple rules with clear definitions allowing just enough initiative to be used by the authorized security people provide better security.

6.1.4.5 Strengthen the Weakest Link

A system will most probably fail at the weakest link and the attacker will attack the weakest link as well. When you intend to increase the security of a system, it is wise to concentrate on the weakest link. If you use resources on strengthening other points, it will not help as the system will be most probable break at the weakest link.

6.1.4.6 Do Not Trust Users

When people make choices, they tend to make the worst security decisions. The reason for this is obvious. Users always tend to prefer functionality and assume that security means spending time with nothing to show for it.

6.1.4.7 Assign Least Privilege

Only the necessary rights and privileges should be given to a subject (user or application) to perform the task with no additional and unnecessary permissions. Limiting a subject's privileges potentially limits the amount of damage that can be caused. This rule helps the organization to protect its resources.

The principle of least privilege requires that every subject is able to access only the objects that are definitely necessary for the subject's functionality. Consider a user that needs to backup data but not need to modify them. Hence, the user must only be able to run the backup, but must not be allowed to update the data. Hence, any other action that is not relevant to making backup needs to be blocked.

6.1.4.8 Defense in Depth

Defense in depth aims to delay the action of an attacker instead of using preventive mechanisms. Additional precautions may then be taken in the mean time and further damage by the attacker may be prevented. Rather than defeating an attacker with a single and strong defensive line, defense in depth aims to force the attacker to lose momentum over a period of time. Once an attacker has lost momentum, reactions can be mounted on the attacker's weak points.

6.1.4.9 Layered Security

The means of authentication is something that the user knows, has, is and does. Consider a firm that makes their staff show an identification card at its main entrance. When attempting to increase the security of the system, it is better to add other mechanisms such as a retina scanner or a password at the entrance of a department. Layered security is practicing different mechanisms to protect information technology resources and data. Hence, rather than

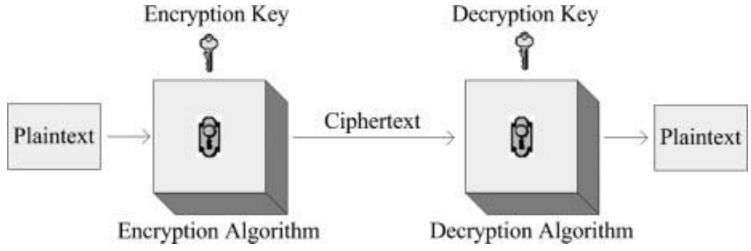


Figure 6.6 Encryption and decryption process.

defending against an attacker with methods using the same strategy, layered security uses different types of mechanisms. For example, an identification card could be used at the main entrance, biometric methods could be used at the headquarters entrance and a keypad requiring a password to be entered could be used at the computing center entrance.

6.2 Security Tools and Mechanisms

Most security mechanisms rely on cryptography. Cryptography is used for providing secure channels, storing password information within the hard disk, digitally signing financial transactions, and so on.

6.2.1 Cryptography

Cryptography is the practice of hiding information by changing its form into a non-obvious text. Basic services provided by cryptography are:

- The ability to store information in a form which is not easy to reveal the original information.
- The ability to exchange information between participants in a secure way.

Cryptography consists of encrypting the original data (message) by using an encryption key, storing or transferring the encrypted data (ciphertext) to the receiver and decrypting the ciphertext using the decryption key to the original message (see Figure 6.6).

The idea of cryptography is being able to send the message in a scrambled form called a ciphertext, so that the communication between the sender and the receiver is still possible, and made by using public channels such as the Internet. In the mean time, the intruder does not have any idea on the actual content, that is, the plaintext of the message.

Cryptography traditionally was performed using symmetric cryptography in which both the sender and the receiver share the same secret key. In 1976, a new form of cryptography, asymmetric cryptography, was introduced with the first example of an RSA algorithm in which the sender and the receiver own a different but matching pair of keys. Hashing, as another form of cryptographic algorithm, does not use any key. It satisfies mostly the integrity of the transferred message but not authentication or confidentiality. Hashing algorithms [e.g., Hash-based Message Authentication Code (HMAC)] that use keys provide authentication as well.

6.2.2 Symmetric Cryptography

Symmetric cryptography, also known as traditional cryptography and secret key cryptography, uses one common key for both the encryption and decryption process. It is referred to as:

- *Traditional cryptography* because it consists of a relatively older mechanism compared with modern cryptography.
- *Secret key cryptography* because the shared key between the sender and the receiver have to be kept secret from third parties.
- *Symmetric cryptography* because:.
  - The algorithms for encryption and decryption are identical (symmetric).
  - The same key is used for both encryption and decryption.

Symmetric key cryptography is used to satisfy secrecy, authentication and integrity, which means the key has to be kept secret. Since the same key is used for both encryption and decryption, the key is symbolized with K without using any subscript. Two important requirements for the secure use of symmetric encryption are:

- A strong encryption algorithm is required.
- The secret key should be kept secret by the sender and receiver. Disclosing the key to any third party results in failure to satisfy the security requirements such as authentication.

Symmetric cryptosystem algorithms are mostly based on substitution, permutation and hybrid schemes. Substitution involves replacing parts of the plaintext with some data from the alphabet. Permutation involves switching parts of the plaintext among themselves. Hybrid schemes consist of performing both methods one after the other.

Earlier examples of the symmetric cryptosystem, for example, Caesar algorithm, transfer the plaintext one letter at a time that are called stream ciphers. More recent symmetric algorithms such as Data Encryption Standard (DES), Triple DES (3DES) and Advanced Encryption Standard (AES) encrypt plaintext in certain lengths and are called block ciphers.

(i) *DES*

DES is a symmetric block cipher that uses shared key and was accepted as a standard in 1976. The algorithm was initially controversial with design elements which were not made public, a relatively short key length, and possibility of a National Security Agency (NSA) backdoor. DES is nowadays considered to be insecure for many applications. This is mainly due to the fact that the 56-bit key size is too small, and many have already cracked the system.

(ii) *3DES*

After introduction of DES, computational power has been increased worldwide and attackers making use of new computers have been successful in brute force attacks. One major reason was the key size. 3DES provides a method for increasing the key size virtually without designing a new algorithm. Hence, the DES algorithm is believed to be relatively secure only in the form of 3DES. It is a modified version of DES where the DES cipher algorithm is applied three times to each data block.

(iii) *AES*

In recent years, the 3DES cipher has been superseded by AES.

6.2.3 Asymmetric Cryptography

In asymmetric cryptography, two different keys are used: the public key which is used for encryption and the private key which is used for decryption and vice versa. The public encryption key is made public to everyone or to every member of a closed group, whereas the private decryption key is known only by the recipient. Messages are encrypted with the public key and can only be decrypted with the private key. Knowledge of the private key does not help in creating the corresponding public key and vice versa. The private key is denoted by K_R and the public key by K_U.

Asymmetric cryptography is referred to as:

- *Modern cryptography* because it consists of a relatively newer mechanism compared with traditional cryptography.
- *Public key cryptography* because the encryption (and sometimes decryption) key is made public and there is no need to keep it secret.
- *Asymmetric cryptography* because:
  - The algorithms for encryption and decryption are different (not symmetric).
  - The encryption key cannot be used for decryption and vice versa.

Symmetric cryptography requires exchanging keys between the parties before encryption occurs. Physical transfer of the key seems secure enough. If the key is kept by both parties absolutely secret, it could then be used to exchange encrypted messages. It is obvious that exchanging keys physically is not a trivial process. Public key cryptography addresses this issue and other drawbacks.

Public key cryptography can also be used for secure exchange of symmetric cryptography keys between two parties. One distinguished example is known as Diffie–Hellman key exchange. This was the first method for establishing a secret key over a public channel.

Asymmetric cryptography can be used to satisfy different security requirements:

- *Public key encryption:* A message encrypted with a public key of a pre-defined receiver can be decrypted only by the legitimate receiver, but nobody else. This case is used for confidentiality.
- *Digital signature:* A message signed with a private key can be verified by anyone who has the public key. If using the public key results in a successful decryption, it proves that the encrypted document has been created by the owner of the private key. This essentially requires keeping the sender's private key secret of course.

Until 1970s, only symmetric algorithms were known and consequently only some requirements could be satisfied such as secrecy and data integrity. On the other hand, extremely important requirements such as non-repudiation could not be provided. The discovery of public key algorithms revolutionized cryptography. If public key cryptography had not been introduced,

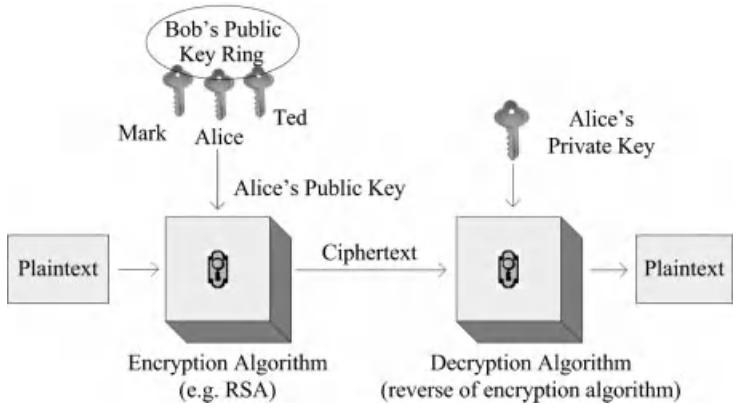


Figure 6.7 RSA Scheme A: using public key for encryption, and private key for decryption.

current usage of bank websites, digital money transfer and so on would not be possible. Most important algorithms are:

(i) *RSA*

The RSA algorithm is the first publicly known public key cryptography (see Figures 6.7 and 6.8). It was a major improvement in cryptography. Currently, RSA is widely used in electronic commerce protocols. It is believed to be secure when sufficiently long keys are used and the usage protocol is set to be perfect.

(ii) *Elliptic curve cryptography*

The vast majority of services that make use of public key cryptography for encryption and digital signatures use the RSA algorithm. Currently a competing system has emerged to challenge RSA called Elliptic Curve Cryptography (ECC). When compared with RSA, the principal advantage of ECC is that it requires smaller bit size for the same security level. Hence, ECC reduces the processing overhead. ECC is fundamentally more difficult

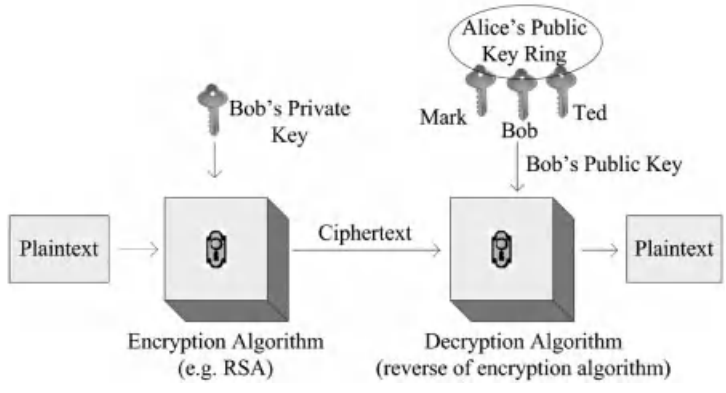


Figure 6.8 RSA Scheme B: using private key for encryption, and public key for decryption.

to explain and to process than RSA. A mathematical technique is required based on the use of a mathematical construct known as an elliptic curve.

6.2.4 Hashing

Hashing provides integrity of the information by creating a digest value of the input. The integrity provided by hashing is not as strong as the integrity that is provided by digital signature though. First of all, not all hashing algorithms provide an equal level of confidence. Consider a hashing algorithm using 1 bit of the hash value, so that the hash value of any message is either 0 or 1. In this case, a brute force attack has a 50% possibility of success even in the first attempt. Apparently, this is not a trustable hashing algorithm at all. As the size of the hash value increases, the robustness of the system increases. The algorithm used for this purpose is of course important and alters the confidence level. We will not give details on this issue, but it is important to note that a poor hashing algorithm disappoints the user.

6.2.5 Message Authentication Code (MAC) and HMAC

In order to be sure about the identity of the sender of a message and the content of it, one option is to encrypt the whole message and then send it. According to the type of cryptographic algorithm used (i.e., symmetric or asymmetric), confidentiality, data integrity, authentication and so on may be satisfied. The problem is that the encryption and decryption phases take time which is not preferable in some cases. A MAC can be generated using some hashing algorithm faster than encryption for this purpose. A MAC algorithm inputs a secret key and a message and then outputs a MAC. The MAC value may be used to provide data integrity as well as its authenticity of the message afterwards.

HMAC is an improved version of a MAC algorithm. MAC does not use any key whereas HMAC uses a secret key shared by both the sender and the receiver. MAC provides data integrity but not authenticity, since the MAC algorithm is public. In contrast, HMAC provides authentication as well, since only the sender and the receiver know the secret key. It should also be noted that since both users own the key, non-repudiation is not satisfied. Currently available common HMAC algorithms are MD5, SHA-1, and SHA-2. The security of HMAC depends on the success of the cryptographic algorithm used, the size of output, and the size and the quality of the cryptographic key.

6.2.6 Digital Signature and Mobile Signature

Digital signature provides authentication and non-repudiation of the user, and integrity of the document. Let us analyze the underlying reasons. First, remember that asymmetric cryptography may be applied in two scenarios. In the first scenario, anybody can use the public key to encrypt a message and send it to the receiver. Then, the receiver who holds the inclusive content of the private key uses that key to decrypt the ciphertext. In the second scenario, the private key is known only by the sender and the sender uses that key to create the ciphertext. The public key is known by everybody and can be used by anyone to decrypt the ciphertext. The

first scenario provides integrity and secrecy, whereas the second scenario provides integrity, authentication and non-repudiation.

Scenario 1 provides:

- Integrity, because if the ciphertext is modified somehow on the way, the modified version of the ciphertext cannot be decrypted by using the private key afterwards.
- Secrecy, because nobody other than the receiver can decrypt the ciphertext since the private key is known only by the receiver.

Scenario 2 provides:

- Integrity, because if the ciphertext is modified somehow on the way, the modified version of the ciphertext cannot be decrypted by using the public key afterwards.
- Authentication, because nobody other than the sender can create the ciphertext, since the private key is known only by the sender and decryption of the ciphertext assures the receiver about this fact.
- Non-repudiation, because encrypting the message by the private key proves that only the sender and nobody else is able to send the encrypted version of the message.

It may be confusing to see that both digital signature and MAC provide integrity. The user may be puzzled about which one to choose when she wants to provide integrity for the data. We have already noted that not all hashing algorithms or MACs are equally promising. There is a difference between hashing and digital signature algorithms. Basically, digital signature algorithms provide better integrity than hashing algorithms. The reason lies in the fact that each specific hash value corresponds to more than one possible input value and the hacker always has a chance of gaining access. It is not true in digital signatures. It is only accidental and rare for a signature to represent any message content.

6.2.7 Comparing Security Mechanisms

The simplest error checking algorithm is parity bit encoding, since it uses 1 bit to check errors which is the smallest amount of data it can store. It is trivial to say that it is the least successful option. Let us consider the RSA algorithm. When shorter key size is used such as 256 or 512 bits, the security is smaller when compared with longer keys such as 1024 or 2048 bits.

Remember that we have security requirements, and cryptographic tools to satisfy them. It might be simpler if there were one mechanism for each consideration but unfortunately this is not the case. Hashing may be used for document integrity; MAC may be used for document integrity and sender authentication; symmetric cryptography may be used for sender and receiver authentication as well as document integrity; digital signature may be used for sender authentication, non-repudiation and document integrity. Unfamiliarity with the details of the algorithms may lead to confusion. In order to select the appropriate one, we should analyze them in detail.

Hashing uses one way functions thus creating a smaller size hash value when compared with the message. The obvious advantages of hashing and MAC when compared with symmetric and asymmetric algorithms is that the hashing algorithms run much faster and have smaller

size which results in better storage. The disadvantage is that the well-known birthday paradox is a vulnerability of this kind of algorithms, thus they are less secure than the other options.

Symmetric cryptography provides confidentiality and message authentication (sender authentication together with document integrity). Symmetric algorithms run faster than hashing based algorithms, but slower than asymmetric algorithms.

Asymmetric cryptography with private key encryption provides confidentiality, message authentication and non-repudiation of the sender and the receiver. When the public key is used for encryption, only message authentication and non-repudiation of the receiver are satisfied. However confidentiality is not satisfied because the public key is known by everybody.

Asymmetric cryptography runs much slower than symmetric algorithms and is not suitable for bulk encryption. Therefore, in cases where high performance is required, some hybrid schemes have to be used. For example, in order to provide a secure channel, first the authentication has to be provided using asymmetric cryptography and then symmetric cryptography has to be performed using the session key which is generated only for that session.

6.2.8 Digital Certificates and Certificate Authority

A digital certificate is actually a digital signature of the corresponding person or organization (e.g., identity, validity period) with an integrated public key. The certificate can be used to verify that the public key belongs to the owner of the certificate. The signature is traditionally issued by a trusted entity that is, a Certificate Authority (CA).

6.2.9 Do Not Keep Cryptographic Algorithms Secret

Assume that you have developed a cryptographic algorithm which can be used to provide some security requirements. It seems obvious that keeping the algorithm secret to yourself makes the algorithm secret. At first sight, the cryptographic system that uses your algorithm seems secure enough. This is not the reality though. It is customary to publish cryptographic algorithms, mostly in an academic environment such as at conferences or in journals, so that interested parties can immediately start to analyze the algorithm in the hope of finding some weakness and then publishing the result of their study.

There are actually three cases for consideration:

- A big gap is found, so that it is impossible to fill the gap and consequently the algorithm is discarded.
- One or more smaller gaps are found which can be fixed by the inventor of the algorithm or repairs might even be suggested by the academician who found the gap.
- No weakness is found, so that the algorithm is perfect enough to be used in providing some security considerations.

As all algorithms are made public, the strong algorithms stand up to scrutiny and the weak algorithms are pushed aside. Using the expertise of academicians, practitioners and professionals will eventually result in either discarding the algorithm or fine tuning it until the algorithm becomes secure enough. Sometimes an encryption system lasts for decades. Private or proprietary algorithms do not help advance security. Often those who analyze proprietary cryptographic systems are the same people who designed them and it is in their best interests not to find a flaw.

Professional cryptographers and amateurs review an algorithm from different viewpoints and this is a sure way to discover whether an encryption algorithm is trustworthy. Manufacturers who do not use a peer review system usually are marginalized and lose business as the public do not trust them. Making the algorithm public deserves help from the security world. Private or proprietary algorithms are mostly not secure enough, because only a few people have contributed to the algorithm. However, it is important to state that the security of an encryption system should be based on key security rather than algorithm security.

6.2.10 Key Types: Symmetric Key, Private Key, Public Key, Master Key, and Session Key

Keys are named according to the algorithms they involve or the purpose they are used for. The keys used in symmetric cryptography are referred to as symmetric (denoted by K). Asymmetric algorithms use private and public key pairs (denoted by K_R and K_U, respectively).

When bulk data are to be encrypted, first a symmetric key is generated and then a symmetric algorithm is used for encryption and decryption. A symmetric algorithm is preferred because of the higher speed of encryption and decryption. If a shared secret key is used in all sessions, then an attacker has the possibility of extracting the key due to the high amount of same key usage. For this reason, a symmetric key is generated where the private and public key pairs are used to increase the security of the algorithm.

6.2.11 Key Management and its Importance

It may be surprising that all of the authentication and digital signature schemes which depend on "what a person knows" algorithms rely on one password. The complete security scheme can be simply broken by inappropriate extraction of the password by unauthorized people. For example, when you receive a password from your bank to access your bank account, all your money may be transferred to another account by an unauthorized party if you are careless and note down your password. Layered security tends to overcome this problem by introducing additional authentication points but it does not trivialize the importance of securing keys or passwords.

6.2.12 WEP (Wired Equivalent Privacy) and WPA (Wi-Fi Protected Access)

WEP was introduced as part of 802.11 protocol in 1997 to provide confidentiality but experience has shown that it is susceptible to eavesdropping. Several weaknesses have been identified. Later, the Wi-Fi Alliance introduced WPA and subsequently WPA2 to overcome the problems encountered.

6.2.13 Other Security Components

(i) *Security policy*

Security policy consists of details on procedures and configuration parameters of the organization in order to satisfy the requirements, possibly set by the management and IT people.

(ii) *Firewall*

A firewall is traditionally used to block unauthorized access whilst allowing authorized communication. The firewall uses rules to decide how to behave on any kind of communication request. Firewalls can be either software installed on a computer or can be integrated to hardware.

(iii) *Intrusion Detection and Prevention System (IDPS)*

An Intrusion Detection System (IDS) is an integrated device or software application that monitors network activities for malicious actions or policy violations. IDPS is mainly focused on detecting possible incidents, logging related information, trying to stop them and reporting them.

(iv) *Physical security*

Physical security consists of measures to prevent attackers from accessing a resource and consists of mechanisms to resist malicious actions. Using locks, fences, security guards, turnstiles are examples of such mechanisms.

(v) *Secure channel*

A secure channel is a communication media between two parties so that security requirements such as secrecy and data integrity are satisfied during data exchange. The obvious technique for creating a secure channel is using cryptographic measures. Establishing a secure channel between two devices is the best approach to protect against an attack. A secure channel can be built using digital signatures and a public key infrastructure. Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS) are methods currently used for building secure channels.

6.3 NFC Security Framework

As same with all information systems, NFC based systems are subject to attacks that threaten system security and user privacy. As discussed in Chapter 3, different NFC operating modes use different communication protocols. Some security threats, services and mechanisms are similar whereas some issues are unique for each operating mode. Here we will examine the security issues related to NFC based systems depending on their operating mode. The reader/writer mode mainly consists of threats and services that are based on RFID infrastructure whereas the peer-to-peer mode is different. The card emulation mode consists of threats and services similar to the contactless smart card systems as well as other operating mode issues.

When NFC based systems are analyzed from a security point of view, the components of the system considering all three operating modes can be listed as (see Figure 6.9):

- Security concerns related to the NFC tag (Section 6.3.1);
- Security concerns related to the NFC reader (Section 6.3.2);
- Security concerns related to a smart card (Section 6.3.3);
- Security concerns related to communication (Section 6.3.4);
- Security concerns related to middleware and backend systems (Section 6.3.5);
- Standardized security protocols (Section 6.3.6).

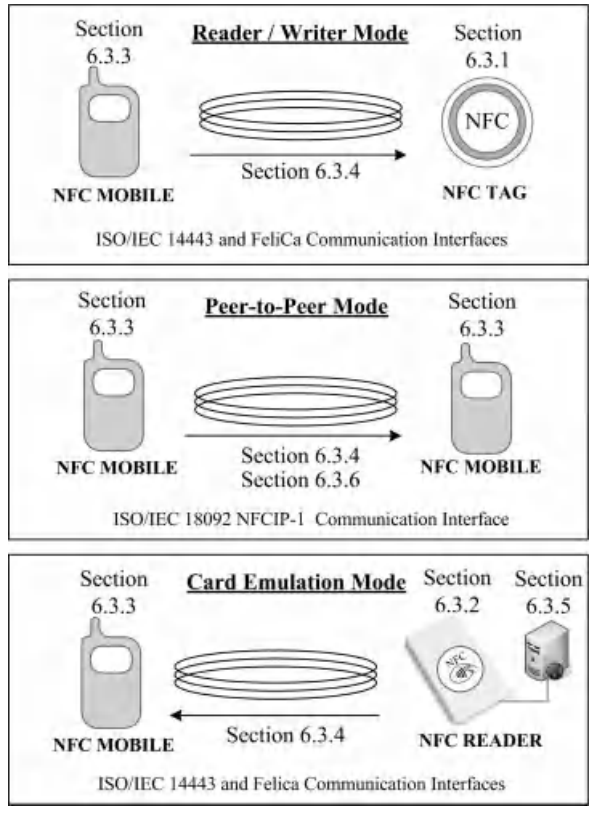


Figure 6.9 NFC security framework.

6.3.1 Security Issues on NFC Tag

Two NFC devices that are involved in the reader/writer mode are the NFC tag on one side and an NFC mobile on the other. Hence, providing security for the NFC system should consider the security of the NFC tag as well as the security of the communication between NFC devices. Remember again that an NFC tag is actually an RFID tag with the corresponding security concerns. Hence we should discuss the security issues and the possible attacks on passive RFID tags first.

6.3.1.1 Attacking NFC Tags

In this section, we will analyze the NFC tag when it is in standby mode. The attacks which occur during communication between an NFC tag and NFC mobile will be analyzed in Section 6.3.4.

The common attacks against NFC tags can be categorized as follows:

- Tag cloning and tag impersonation;
- Tag content changes;
- Tag replacement and tag hiding.

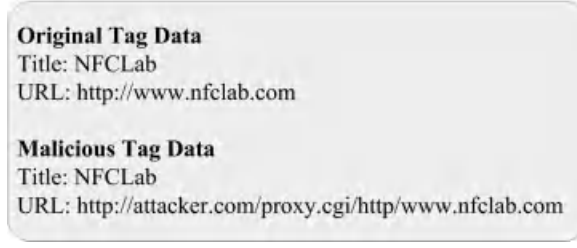


Figure 6.10 Example of malicious tag.

(i) *Tag cloning and tag impersonation*

From the RFID technology point of view, the most challenging security threats in commercial RFID applications are tag cloning and tag impersonation. The research community addresses these threats primarily by trying to make tag cloning harder by using cryptographic tag authentication protocols. The fundamental difficulties of this research revolve around the trade-offs between tag cost, level of security and performance in terms of reading speed and distance. It is currently challenging to protect a passive RFID tag from cloning.

Tag cloning has the same impact as relay attacks with the big advantage to the attacker that the RFID tag can be reused without the need to use the victim for each individual attack. Note that a relay attack is performed during an NFC communication, whereas cloning is performed before the communication starts. Hence, it technically seems easier to perform tag cloning.

(ii) *Tag content changes*

Tag content can be changed by various attacks:

- *Spoofing attacks*

A spoofing attack is providing false information to the user that seems valid. Spoofing attacks typically involve a fake domain name, telephone number or false information about the identification of some person, item, or activity [8]. Broadcasting an incorrect Electronic Product Code™ (EPC™) is a spoofing example in an RFID system. Some of the possible spoofing attacks are:

– *URI spoofing:* Basically this attack abuses smart posters and hides real URI to take advantage of the weakness in the GUI of mobile phones. It tricks users to perform harmful operations.

– *URL spoofing:* A fake, innocent looking URL is stored on a smart poster title while the actual and malicious URL is stored in the NFC tag (see Figure 6.10). Users generally cannot notice the actual URI after reading the tag. They assume the URL on the smart poster is being used.

– *Phone call spoofing:* A fake phone number is stored in a smart poster while a premium rate number such as 0900xxxxxxx is stored in the URI record. Thus, the possible impact of the attack is high. This kind of attack is probable since the attacker gains financial benefit out of it.

– *SMS spoofing:* A fake phone number and a message are stored in a smart poster while another service number is mentioned; for example, to download a ring tone

which have some cost. This is less likely to happen since the user has to confirm the SMS in a clear screen.

- *Manipulating tag data*

  Depending on the nature of the tag, the price, stock number and any other data on the tag can be modified or manipulated. For example, by changing price data on the tag, a hacker can obtain a dramatic discount. Other changes to a tag's data can also be used to fool the user.

- *DOS attack*

  DoS attacks can be used to frustrate the relationship between the customer and service provider. For example, a malicious tag containing a malformed NDEF message that is used for a service can cause mobile phones to crash and to reboot in every time the service is used. Users will eventually stop using the service to avoid the crash.

(iii) *Tag replacement and tag hiding*

Sticking a malicious tag on top of the original tag or replacing the original tag with a malicious tag is enough to let the system work as the attacker desires. In the case of sticking on a new tag, it is possible to disable the old tag. Another method to attack passive tags is to break the write protection of the tag and overwrite it with malicious data.

6.3.1.2 Defense Mechanisms

NFC tags can be protected by signing them using signature techniques such as encryption. However only signing the tag data still does not prevent cloning.

6.3.2 Security Issues on NFC Reader

An NFC reader is an important NFC device which mainly enables card emulation mode applications consisting of an NFC enabled mobile phone on one side and a reader on the other. An NFC reader is similar to an RFID reader so the security concerns are the same. The major attacking methods on NFC readers are their removal or destruction and impersonation [1]:

- *Removal or destruction of NFC readers:* As with RFID readers, NFC readers can also be subject to destruction or removal. NFC readers can be stolen particularly when they are placed in unattended places. An NFC reader can include critical information such as cryptographic keys which can be the target of an attacker. The impact of a stolen NFC reader is considerable since its potential manipulation could enable malicious attackers to gain access not only to NFC enabled mobile phones, but also to the backend system where possible data manipulation can be facilitated.
- *Impersonation:* When NFC communication is unauthenticated, attackers can easily counterfeit the identity of a legitimate reader in order to elicit sensitive information and also modify data on tags. The feasibility of these attacks depends on the security measures for authenticating the reader. For instance, if credentials are stored on the reader, a stolen reader may provide the necessary credentials for gaining access to RFID tags and backend systems.

6.3.3 Security Issues on Smart Card

As already mentioned, smart cards are generally used for Secure Element (SE) on NFC enabled mobile devices. As usual, there can be vulnerabilities on smart cards of NFC mobiles. In this section we describe attacks and countermeasures performed on smart cards. The attacks are examined in two groups: invasive attacks and side channel attacks. The attacks that will be described here are valid in both contact smart cards as well as contactless smart cards.

(i) *Invasive attacks*

Invasive attacks require removal of or attack on the microprocessor of a smart card physically. These attacks typically require very expensive equipment, great expertise and a large investment. The countermeasures that can be implemented to prevent invasive attacks include [2]:

- *Design:* The integrated circuit design can include such countermeasures as glue logic, obfuscated logic and buried buses which make reverse engineering harder. Non-volatile memory, buses and logic can be scrambled to prevent reverse engineering of embedded software or chip design techniques through probing.
- *Silicon features:* Some integrated circuits include a shield that is an extra metal layer above the functional metal layers which acts to prevent visual and physical access to the surface of the chip. Features of this nature can be used over the entire chip or specific sensitive parts of the chip.
- *Anomaly detectors:* There are usually different types of anomaly detectors in a smart card. These are used to detect unusual environmental conditions such as events in the voltage and clock supplied to the card. A smart card will typically reset or execute an infinite loop until the abnormal condition is removed.

(ii) *Side channel attacks*

One method of attacking smart cards is to observe a side channel while information is being processed. This means that an attacker seeks to derive information by observing how the characteristic of a smart card change as it processes different information. Some examples of side channel attacks are as follows [2]:

- *Timing analysis:* The simplest form of side channel analysis is to simply observe how long a given process takes to execute and make inferences from these observations. The time length of a process can leak information about the data being processed. For instance; if the digits of a Personal Identification Number (PIN) are checked individually and a negative result returned when a wrong digit is encountered, an attacker could use the time length to determine how many digits of a guessed PIN are correct.
- *Simple side channel analysis:* Another way is to observe the power consumption of a smart card over time by simply observing the voltage change over a resistor in series. The amount of consumed power is dependent on the type of instruction being executed and the data being manipulated.
- *Fault induction attack:* Another method of attacking a smart card is to attempt to inject a fault during its normal functioning such as changing a ciphertext produced by a cryptographic algorithm. This would allow information to be derived on the key being used.

The countermeasures that can be implemented to prevent side channel attacks include:

- *Constant execution:* Algorithms can be implemented such that the same operations will be conducted in the same order irrespective of data and key values being used. This prevents an attacker from conducting timing and simple side channel analysis.
- *Random delays:* Differential side channel analysis requires the same operations to be conducted in the same order irrespective of data and key values. Functions that do nothing but loop for a random length of time can be included in implementations, so that an attacker is required to synchronize acquisitions a posteriori.
- *Randomization:* Differential side channel analysis also requires that there is a correlation between data being manipulated and an observed side channel. This is achieved by manipulating the data in such a way that the value presented in the memory is always masked with a random value. Then, this mask is removed at the end of the algorithm to produce the ciphertext.

6.3.4 Security Issues on Communication

In all operating modes of NFC technology, it is obvious that a short range communication is used. Attackers with enhanced radio devices can communicate with the contactless smart cards within several meters. Therefore, attacks and threats during the communication are valid in all modes.

6.3.4.1 Attacks against Communication

(i) *Eavesdropping*

In eavesdropping an unauthorized individual uses an antenna in order to record communication between legitimate devices [7]. The wireless nature of RFID makes eavesdropping one of the most serious threats. This type of attack can be performed in both tag to reader and reader to tag directions. The signal that will be eavesdropped is also subject to the location of the eavesdropper regarding the RFID tag and the reader as well as the possible countermeasures employed for deteriorating the radio signal.

- *Eavesdropping range*

The communication in NFC is always performed between two devices in close proximity. The main question is how close an attacker needs to be in order to retrieve a usable RF signal. Unfortunately, there is no exact answer to this question. The reason for this results from the huge number of parameters that affect the problem.

Additionally, the operating mode in which the communication is performed is important. This means whether the sender is generating its own RF field or whether the sender is using the RF field generated by another device. It is much harder to eavesdrop on a passive device that uses an RF field generated by an active device. The reason is obvious; an active device generates a signal with a relatively higher range.

(ii) *Data corruption*

An attacker may also try to modify NFC data. In the simplest case, the attacker may try to generate a DoS attack as explained earlier.

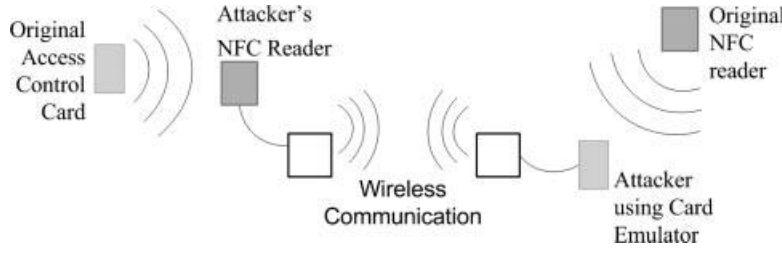


Figure 6.11 Relay attack.

(iii) *Data modification*

In data modification the attacker wants to modify or delete valuable information by intercepting the communication.

(iv) *Data insertion*

The attacker may also wish to insert some code into the exchanged data. This is only possible in the case where the responding device waits a very long time to answer. Then the attacker could send the data back to the sender earlier than the valid receiver. The insertion will be successful if the inserted data are transmitted before the original device starts to respond. If both data streams overlap, the data will be corrupted.

(v) *MIM attack*

MIM attack takes advantage of the mutual trust of a third party or the simultaneous impersonation of both sides of a two-way trust as already shown in Figure 6.5. MIM attackers are unknown parties in a communication who relay information back and forth giving the simultaneous appearance of being the other party.

Since RFID and NFC use wireless media, it becomes easier to perform a MIM attack which forces the security people to create stronger preventative measures. Physical and administrative measures are one set of options and another is a cryptographic solution. The parameters that need to be considered in NFC case are as follows:

- The near field property of communication;
- The generated RF power;
- The range to the NFC tag.

(vi) *Relay attack*

ISO/IEC 14443 compatible cards are vulnerable to relay attacks. The victim is unaware of this attack if appropriate precautions are not followed. Figure 6.11 shows a simple set-up for a relay attack. The attacker uses wireless communication to borrow the data from the victim's tag into another tag. This means that the attacker inserts messages into the exchanged data between two devices. This is only possible if the responding device needs a period of time before sending the request so that the relay action may be performed during that interval. If both data streams overlap, the data will be corrupted.

(vii) *Replay attack*

Intercepting a valid NFC signal, recording it for a later use and transmitting to a reader afterwards is called a replay attack. Since the data appear valid, the reader most probably accepts them. There are some prevention mechanisms to prevent a replay attack. An example is to use a sequence number or timing constraints. Note, a relay attack is performed online whilst a replay attack is performed offline.

6.3.4.2 Defending the Communication

(i) *Eavesdropping*
Data transmitted in passive mode are significantly harder to be eavesdropped when compared with active modes. Just using the passive mode is probably not sufficient for most applications which transmit sensitive data. The only actual solution to prevent an eavesdropping attack is to establish a secure channel.

(ii) *Data corruption*
NFC devices can counter this attack because they can check the RF field while they are transmitting data. NFC device can easily detect this type of attack easily since the required power to corrupt the data is significantly high.

(iii) *Data modification*
Protection against data modification can be achieved in various ways. By using 106 kbaud in active mode, it becomes impossible for an attacker to modify all the data transmitted via the RF link. This means that in both directions, an active communication mode needs to prevent data modification. Also, the protection against modification is not perfect, since even some bits can be modified at 106 kbaud.

(iv) *Data insertion*
There are three possible countermeasures for data insertion. One is that the responder answers with no delay. In this case the attacker cannot be faster than the valid responder. The attacker can be as fast as the valid device, however if two devices answer at the same time, no correct data are received. The second possible countermeasure is that the valid responder can listen to the channel. Then the device could detect an attacker who wants to insert data. The third option is building a secure channel between the two devices.

(v) *MIM attack*
In order to prevent a MIM attack, the active party should listen to the RF field while sending data to be able to detect any disturbances caused by a potential attacker.

6.3.5 Middleware and Backend System Security

An NFC based system contains NFC readers, NFC mobiles and NFC tags in technical terms. However, a complete NFC system includes servers to store and manage data such as banking servers, credit card middleware, authentication subsystems, and so on. Hence, security of an NFC system is not complete unless the security of all components of the complete system is provided. Security of the middleware and the backend systems is not within scope of this book. The reader should be aware that the middleware and backend systems should be secure.

Databases may be extremely sensitive if they contain valuable information such as credit card numbers. Companies may even lose the confidence of consumers unless they prevent the damage or quickly correct it. There have been many reports of companies suffering major setbacks by losing customers due to an IT related failure.

6.3.6 Standardized NFC Security Protocols

Security protocols of NFCIP-1 are standardized in ECMA 385 as NFC-SEC and ECMA 386 as NFC-SEC-01. These security protocols are used in peer-to-peer operating mode.

Table 6.2 Summary of provided security services [3]

Protocol	Security Services
NFC-SEC	Eavesdropping, Data modification
NFC-SEC-01	SSE - Elliptic Curve Diffie–Hellman (ECDH) Key exchange (192 bit) - Key derivation and confirmation (AES 128 bit) SCH - Data encryption (AES 128 bit) - Data integrity (AES 128 bit)

6.3.6.1 NFC-SEC: NFCIP-1 Security Services and Protocol – ECMA 385

NFC-SEC provides security standard for peer-to-peer NFC communication, which does not include reader/writer and card emulation mode. Please remember that increased security services always decreases the performance provided to the user. Nonetheless, NFC-SEC provides a good balance between security and performance.

NFCIP-1 provides no security at all. On top of NFCIP-1, NFC-SEC is promoted to provide security capabilities to it. Protocols that are included within NFC-SEC are defined to be used on top of NFCIP-1 protocol. NFC-SEC defines a protocol stack that enables application independent encryption functions on the data link layer. Applications using peer-to-peer mode do not require application specific encryption mechanisms for the security services that are provided by NFC components. This case is similar to using IPsec and ignoring security services on higher layers.

NFC-SEC consists of two different protocols (see Table 6.2). Basic NFC-SEC is the Security Services and Protocol standard. It is specified in ECMA-385 [3]. Its main services are protection against eavesdropping and data modification. The common framework is complemented by ECMA-386 NFC-SEC-01; NFC-SEC Cryptography Standard using ECDH and AES [4]. It specifies cryptographic mechanisms that use ECDH protocol for key agreement and AES algorithm for data encryption and integrity. More cryptography standards may follow in the future, each of them is to be identified by a Protocol Identifier (PID) (see Figure 6.12).

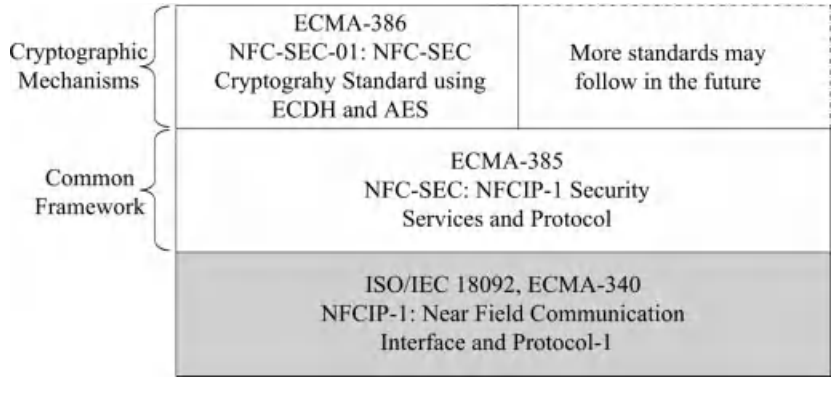


Figure 6.12 NFC-SEC standards architecture [5].

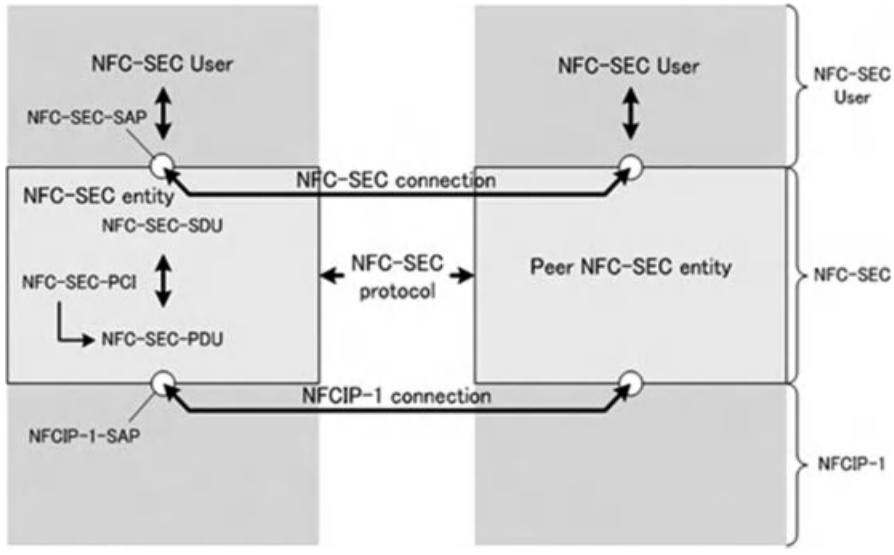


Figure 6.13 NFC-SEC architecture [3].

6.3.6.2 NFC-SEC and its Architecture

NFC-SEC architecture uses OSI reference model specified in ISO/IEC 7498-1 and the security architecture of ISO 7498-2 (see Figure 6.13). NFC-SEC users invoke and access NFC-SEC services through NFC-SEC Service Access Points (NFC-SEC-SAPs). NFC-SEC entities obtain NFC-SEC Service Data Units (NFC-SEC-SDUs) which are referred to as requests from NFC-SEC users and return NFC-SEC-SDUs (confirmations) to them. NFC-SEC entities that are peered exchange NFC-SEC-PDUs (Protocol Data Units) by conforming to the NFC-SEC protocol. NFC-SEC entities that are peered communicate with each other by accessing NFCIP-1 data service through NFCIP-1 Service Access Points (NFCIP-1-SAPs). NFC-SEC entities can send and receive NFC-SEC-PDUs. A typical NFC-SEC-PDU includes NFC-SEC Protocol Control Information (NFC-SEC-PCI) and a single NFC-SEC-SDU.

The structure of an NFC-SEC-PDU is seen in Figure 6.14. It includes a Secure Exchange Protocol (SEP), a PID, and an NFC-SEC Payload.

Table 6.3 shows possible NFC-SEC-PDU options with their codes and descriptions. Four basic steps are defined, as illustrated in Figure 6.15. Steps of "key agreement" followed by "key confirmation" are the key establishment phases and they are required for both SSE and SCH. "PDU security" provides the actual data encryption and protection, required only by SCH. Finally both SSE and SCH end with a "Termination" protocol step.

Figure 6.14 Structure of NFC-SEC PDU [3].

Table 6.3 PDU types and description [3]

NFC-SEC-PDU		
Name	Code	Description
ACT_REQ	0000	Activation request to request a new service
ACT_RES	0001	Activation response to accept service request
VFY_REQ	0010	Verification request to submit check values for verification of the sender's shared secret
VFY_RES	0011	Verification response to submit check values for verification of the recipient's shared secret
ENC	0100	Encrypts packet for secure data exchange
TMN	0110	Terminate request to terminate a service
ERROR	1111	Indication of an error

- *Key agreement:* Peered NFC-SEC entities establish a shared secret using ACT_REQ (activation request PDU) and ACT_RES (activation response PDU) afterwards.
- *Key confirmation:* NFC-SEC entities verify their agreed shared secret using VFY_REQ (verification request PDU) and VFY_RES (verification response PDU).
- *PDU security:* Peered NFC-SEC entities protect their data exchange mechanisms by using ENC (encrypted packet PDU). The protocol also provides sequence integrity based on

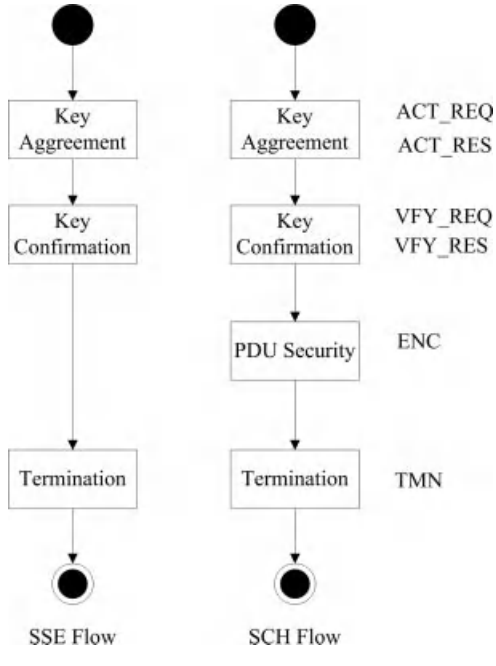


Figure 6.15 Flow of NFC-SEC services [3].

protected sequence numbers. Additionally the protocol enables confidentiality, data integrity and origin authentication as well.
- *Termination:* Peered NFC-SEC entities can terminate their session by using TMN (terminate PDU). When NFCIP-1 is released or deselected, or the NFC device is powered off, the session is to be terminated as well.

6.3.6.3 NFC-SEC-01 Mechanisms using ECDH and AES

To enable secure communication between NFC devices that do not share any common secret data before they start communicating with each other, public key cryptography is used to establish a shared secret between these devices, and more specifically an ECDH key exchange scheme. This shared secret is used to establish SSE and SCH.

NFC-SEC-01 is standardized in ECMA 386 which specifies cryptographic mechanisms that use the ECDH protocol for key agreement and the AES algorithm for data encryption and integrity [4]. ECMA 386 standard specifies mechanisms for SSE and SCH which are defined in ECMA 385. NFC-SEC-01 provides mainly the following functions:

- Message contents with concatenation rules for keys and other fields;
- Key primitives;
- Random number requirements;
- Conversion and transformation rules;
- Cryptographic algorithms and methods.

The specified Key Derivation Function (KDF), key confirmation, data encryption, and data integrity functions are based on AES. Data confidentiality is ensured by AES with 128 bit key length in CTR mode which is secure and suitable for restricted communication bandwidth because no padding is required.

Two KDFs are specified in ECMA 386. The first one is for SSE which is the master key MKSSE that is for the key verification for the secure channel. The other function is for SCH which derives three keys: the master key for SSE (MKSCH); the encryption key for SCH (KESCH); and the integrity protection key for SCH (KISCH). The usage of keys is described in Table 6.4.

When a key is derived after the KDF processes, NFC-SEC entities check if they indeed have the same key. To perform this key confirmation process, each entity generates a key

Table 6.4 Key usage summary [4]

Key	Key Usage
MKSSE	Key verification for the secure channel
MKSCH	Master key for SSE used as a shared secret to be passed to the upper layer and as key verification
KESCH	Encryption of the data packets sent through SCH
KISCH	Integrity protection of data packets sent through SCH

Table 6.5 Security protocols and their reference standards [5]

Protocol	Protocol Parts	Reference Standards
NFC-SEC	Framework	ISO/IEC 11770-1
	Basic model	ISO/IEC 7498-1
	Security architecture	ISO 7498-2
	Conventions for the definition of OSI services	ISO/IEC 10731
NFC-SEC-01	General specifications	ISO/IEC 15946-1
	Key management using asymmetric technique	ISO/IEC 11770-3
	Block ciphers	ISO/IEC 18033-3 and ISO/IEC 10116
	Public key cryptography	IEEE 1363 and FIPS 186-2
	Random number bit generation	ISO/IEC 18031

confirmation tag (called MacTag) and then sends it to its peer entity. Entities verify the key confirmation tag after the reception of the key.

6.3.6.4 Entity Authentication and MIM Attacks

The mechanism does not protect against MIM attacks because no entity authentication can be provided when the peer NFC devices do not share any secret beforehand [4]. The practical risk of MIM attacks is regarded as low in the targeted use cases due to the short operating distance and the specific RF characteristics of NFC. Users should be aware and carefully evaluate the potential vulnerability of planned implementations.

6.3.6.5 Summary of NFC Security Standards

NFC-SEC and NFC-SEC-01 protocols are based on some international standards as seen in Table 6.5.

6.4 Privacy, Legal, and Ethical Aspects

Privacy is appropriate use of information. When some information is private to a person, it means that the information is personally special. Privacy is a broader term than security meaning that privacy may be violated even when many security requirements are satisfied.

Let us consider authentication for a while. When somebody, say Alice, gives her phone number to Bob, she assumes that Bob will keep the number secret. However Bob gives out Alice's phone number to Liza without requesting Alice's permission. He may have created an awkward situation. The reason is that Alice simply may not be willing to give out her phone number to Liza. This is indeed a very common example of information misuse.

Privacy has several dimensions. One is protection of personal data and is described as informational privacy or data protection. As individuals, we do not want information about ourselves to be made available to others, even if the initial disclosure is performed with our

Figure 6.16 Major privacy issues.

approval. It is similar with the private information that we have shared with our close friends with the initial warning: "I will tell you something, but you must promise me not to tell anybody else first!"

Disclosure of some information about people may be considered as an invasion of privacy in some countries, but may be more acceptable in others. Privacy may also be voluntarily sacrificed, mostly for perceived benefits. Constitutes or laws of different countries have different levels of support in privacy issues, and there may also be laws limiting the degree of privacy for the sake of public interest. One basic support of privacy can be in keeping conversations made by people using mobile phones confidential.

Privacy issues can be further classified into four categories: unobservability, unlinkability, pseudonymity, and anonymity (see Figure 6.16):

- According to ISO 99, unobservability is the state of items of interest being indistinguishable from any other items of interest (e.g., subjects, messages, events, actions).
- According to ISO 99, unlinkability is when two or more items of interest within a system, from the attacker's perspective, are no more and no less related after observation than they are related before the attacker's knowledge.
- Pseudonymity means an unknown or undeclared source which is the state of mistaken disguised identity.
- According to ISO 99, anonymity is the state of being not identifiable and not noticed.

The people that have immediate or subsequent access to private information may not be trusted; privacy is provided with the help of anonymity. Using pseudo identities, unlinkable information and unobservable behaviors and relations are the most used methods to provide anonymity and consequently privacy.

6.4.1 It is a Different World

A few decades ago the world was regarded to be in the age of information. Now, it is regarded to be in the age of information management. The reason for this difference is obvious: data and information are currently everywhere. It is more important to retrieve the useful information than to create it.

Data will be collected from all possible points using all possible sensor and data collector devices. Wireless sensor nodes, NFC tags, and cameras are only a few examples of data collecting devices. When you call somebody or even carry your mobile phone without using it, even buy something using cash or click on a website on your browser, a vast amount of data are recorded.

6.4.2 Some Examples on Privacy Issues

The easiest solution to physical access control is to have people as guards at the doors of all sensitive areas. However, guards are expensive, make mistakes, and do not like to keep audit trails. Access cards in the form of magnetic stripe cards have been introduced. These systems use a computer driven backend, so that cards can be revoked and removed from the system, and identity, location and time logs are kept. However, illegally captured stripe cards may be used by a hacker and unfortunately illegal access will be easier than using guards in this case.

RFID and contactless proximity smart card technologies were used to solve these problems. These cards are active RFID implementations, meaning that they have an embedded power source.

Consider the following theoretical situation which makes use of tags and violates user privacy. You buy a sweater that contains an RFID tag. When you go through the checkout stand, the item is scanned and you pay for it with your credit card. A few weeks later you wear the sweater to the same store where you purchased it. Provided the tag still works, when you enter the store, the reader in the door recognizes the identification number and matches it to your name and credit card information.

There are various RFID cases all over the world that highlight the issues of security and privacy:

(i) *Benetton case*

One example that caused a negative public reaction was the Benetton case which led the company to withdraw plans for embedding RFID tags in their items. In 2003, Philips announced that it would supply RFID tags to Benetton. After the privacy advocates heard of the plan, they called for a boycott of the company. In the face of this public outcry, Benetton backed down and announced that individual items of clothing would not carry RFID tags.

(ii) *Metro Group*

Metro Group established an experimental outlet, named Future Store, in order to test new technologies and concepts including RFID technology. RFID tags were attached to each product wrap. These tags contained a unique identification which was programmed at the factory during the manufacturing phase. The readers that were attached to the shelves monitored the current amount of items for each brand.

To satisfy concerns on privacy, the store provided deactivation kiosks to deactivate the tags when the customer wanted to do so before purchasing the product. The company included RFID tags in its loyalty cards, but did not inform the customer about this. This led to protests being made and the Metro Group subsequently stopped the use of RFID tags in the loyalty cards. The Metro Group is continuing to move forward with its Future Store initiative but is dealing with the privacy issues much more.

(iii) *Wal-Mart*

Wal-Mart is a big market chain that aimed to encourage their suppliers to integrate RFID into their supply chains. The purpose was to automatically scan a pallet of goods as it enters or leaves the warehouse, thus saving time and other resources.

(iv) *RFID passports*

RFID tags have started to be used in new passports for authentication and integrity purposes. An RFID enabled password is also one use case of NFC technology. Security and privacy issues related to RFID passports also exist for the NFC enabled password case.

Many countries, including the USA, Turkey, and some other European countries have actually implemented the tags in recent years. RFID tags on passports are used to update security and to protect against counterfeit. However, this addition to the passports has caused a huge debate among security and privacy experts, and national security advocates.

The new passport design integrates an RFID tag into the passport according to the ISO 14443A and 14443B format specifications. Passports are readable within 10 cm and the tag contains an identical copy of the information that is already printed in the passport including the photo and other biometric information such as fingerprints and signatures. With this addition, altering stolen or lost passports became much harder. The chip stores a person's biometric information which increases the ability for border guards and issuing agencies to confirm someone's passport.

One major concern with RFID passports is skimming which is the ability to read information on a passport. The fear is that criminals might be able to pick specific people out of a crowd and possibly target them for kidnaping or robbery.

Even if the information is encrypted, a passport can still be identified according to the issuing country. To prevent the problems where more than one tag is in range of a reader, every tag has a collision avoidance identifier. This unique identifier allows the reader to distinguish one tag from another. Having RFID in passports also solves a standards compliance problem and a political issue concerning the perceived need to increase passport security. Extra care should be taken when a security device is used in something as important as a passport.

6.4.3 Summary on Privacy and Countermeasures

Some problems exist with RFID technology and there is also some exaggerated and misleading information. A lot of information regarding RFID has been published with respect to its possible covert tracking possibilities. This speculation and misinformation has led people to be wary of RFID.

It is hard to gage just how widespread consumer concerns are. A small number of consumers that are passionate about a particular issue can have a large voice. Even a small number of people whose concerns are not satisfactorily addressed can bring an RFID initiative to a halt. The following lessons can be learned from the previous examples:

- As a first step, there is a need to understand the consumer's point of view regarding RFID tags. Try to get feedback from consumers.
- Demonstrate the steps being taken to protect consumer privacy and put control into the consumer's hands. Make sure that the message gets out.

6.4.4 Some Proposals for Providing Privacy on Tags

Here we will present some privacy mechanisms about the use of NFC tags.

(i) *Kill switch*

The content of NFC tags may be cleared, so that it cannot be used anymore. A problem occurs when the data on the NFC tags are required after leaving the store. If the customer wants to return the item to the store for some reason, it would be useful if the tag would be still readable. Some potential problems [6] are:

- Stores may wish products to have tags readable in case the products are returned as defective.
- Products may need to be read, so that they may be categorized for recycling purposes after the usage of the item for some period by the customer.
- Stores may issue receipts with embedded tags, so that they can confirm purchase details when a product is returned.
- Collectibles such as baseball cards and CDs may have RFID tags to enable owners to manage their inventory better.
- A refrigerator or pantry shelf may be able to tell when some food or drug product has passed its expiry date.

(ii) *Faraday Cage approach*

An NFC tag may be protected by using a container made of metal that is impenetrable by radio signals; this is called Faraday Cage.

(iii) *Active jamming*

It is possible that a customer can carry a device that broadcasts radio signals. It can block or disrupt the operation of any nearby RFID readers. This is a high cost solution.

(iv) *Smart RFID approach*

NFC tags can be made smarter by the motivation to protect privacy. This would obviously require using cryptographic methods and involve higher cost than regular tags. Three instances of the smart tag approach that have been proposed are:

- Hash lock method;
- Re-encryption method (in several forms);
- Silent tree walking.

6.4.5 What to do for Protecting Privacy

The public confronts RFID systems since they use relatively new and unfamiliar technology whose functionality, limits and risks are not completely understood. As in other new technologies, RFID meets with fear and rejection.

Using RFID technology has caused considerable objections by consumer privacy advocates. This is especially true when the owner of an item is not informed about the existence of a tag, which can be read at a distance without being noticed. It is possible to leak sensitive data if an individual does not have the appropriate knowledge.

In the case of NFC, with the widespread use of NFC mobiles and NFC applications, the collection of data as well as the storage of valuable private data (e.g., credit card information, identity information, access data) will increase. It will become more complicated for the consumer to manage and oversee the collected and stored data. Thus, privacy gains special importance.

Users of NFC applications should be convinced that these applications will not misuse their personal data and privacy. In order to achieve the trust of users there is a clear need for effective tools that support users in protecting their privacy. Supporting data protection legislations, other legal measurements and auditing mechanisms can be required too.

6.5 Chapter Summary

NFC does not only impose new vulnerabilities, but also leads to privacy concerns of the users as well. Consider the case where personal health data are saved to hospital servers, so that

the hospital management as well as insurance firms can use the information. You may have received notices from your insurance firm, informing you about the cancellation of some portion of a policy due to some illness you have experienced. This situation is definitely for the benefit of the insurance companies. If the health data are to be stored in a local database on an NFC enabled personal mobile phone and are to be removed from the hospital database, the privacy concerns of the user can be satisfied more.

NFC is a new emerging industry and there is much work to be done in this field. There is a need for highly standardized, interoperable mechanisms worldwide. Since NFC is a wireless communication technology and involves storing and exchanging private and important data, security and privacy issues need to be clearly handled.

Chapter Questions

1. Explain confidentiality, data integrity, authentication and authorization.
2. Why is perfect security not possible?
3. Why are hackers one step ahead?
4. Why does complexity weaken security?
5. How can you apply a pharming attack to a user? Give two examples.
6. Explain vulnerability, threat, attack, and risk by giving an example.
7. What are the differences between symmetric and asymmetric cryptography security?
8. Explain tag cloning attacks against NFC tags giving examples.
9. What is an eavesdropping attack in terms of NFC security?
10. What is a man in the middle attack in terms of NFC security?
11. What is the difference between relay and replay attacks?
12. What are NFC-SEC and NFC-SEC-01? Explain the security services provided by these protocols.
13. What are the major privacy concerns that need to be handled in designing NFC applications?

References

[1] Mitrokotsa, A. *et al*. (2009) Classification of RFID attacks. *Information Systems Frontiers*, **12**(5), 491–505

[2] Leng, X. (2009) Smart card applications and security. *Information Security Technical Report*, **14**(2), 36–45

[3] ECMA International (2010) *ECMA 385: NFC-SEC: NFCIP-1 Security Services and Protocol*, June 2010, http://www.ecma-international.org/memento/TC47-M.htm (accessed 10 July 2011).

[4] ECMA International (2010) *ECMA 386: NFC-SEC-01: NFC-SEC Cryptographic Standard using ECDH and AES*, June 2010, http://www.ecma-international.org/memento/TC47-M.htm (accessed 10 July 2011).

[5] ECMA International (2008) *NFC-SEC*, White Paper, December 2008. Available at: http://www.ecma-international.org/activities/Communications/tc47-2008-089.pdf (accessed 10 July 2011).

[6] Juels, A. *et al*. *The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy.* Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.7036&rep=rep1&type=pdf (accessed 10 July 2011).

[7] Haselsteiner, E. and Breitfuß, K. *Security in Near Field Communication (NFC)*, Philips Semiconductors, Available at: http://events.iaik.tugraz.at/RFIDSec06/Program/papers/002%20-%20Security%20in%20NFC.pdf (accessed 10 July 2011).

[8] Mulliner, C. (2008) *Attacking NFC Mobile Phones*, EUSecWest 2008. Available at: http://www.mulliner.org/nfc/feed/collin_mulliner_eusecwest08_attacking_nfc_phones.pdf (accessed 10 July 2011).

7

NFC Business Ecosystem

This chapter aims to establish a solid foundation for the business aspect of NFC. It is a fact that studies in NFC literature are scarce and the terminology used for discussion on business matters is currently too immature. There is a need to clarify the essential notions characterizing NFC and its business environment. In this chapter, those characterizing notions are articulated, and a conceptual ground for NFC is developed in the ecosystem context.

First, this chapter seeks answers for the questions "What does an ecosystem mean in general?" and "What does an ecosystem mean within an NFC context?". Then a comprehensive analysis of the NFC business environment is performed. Dominating stakeholders, responsibilities and generic roles which can serve as a reference point for presenting business models are examined by considering the viewpoints of different associations.

The business model development deserves an in depth discussion in terms of business drivers and challenges, critical issues affecting business modeling, and viable business scenarios with industry insights. This chapter seeks to represent the main driving factors of business models, proposes three business model alternatives, and explains how and why NFC based systems are complex. Also, at the end of this chapter, the NFC ticketing use case discussed in Chapter 4 is studied from an ecosystem perspective, and also a business model proposal from one of the industry pioneers is presented.

7.1 Business Ecosystem

The notion of an ecosystem is rooted in various sciences such as biology, sociology, management, and economy. One can give its meaning as a community of living organisms in a particular area with air, water, sunlight and other resources. Actually this definition describes an ecosystem in the biological context. Depending on the context in which the ecosystem is used, several definitions and expressions have been introduced in the literature. Some of the ecosystem examples and ideas presented here have been discussed earlier [1–3].

For example, an industrial ecosystem is proposed to emphasize the importance of sustainable development in all kinds of industrial operations. Another example is the social ecosystem which is defined as "each organization is a fully participating party which both influences and is influenced by the social ecosystem made up of all related businesses, consumers, and

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.
 Published 2012 by John Wiley & Sons, Ltd.

suppliers, as well as economic, cultural, and legal institutions" [1]. Actually the interdependence among entities is highlighted in [1] as one of the major characteristics of a social ecosystem.

Our final example is the business ecosystem which is essentially the focal point of this chapter. Various contributors have extensive suggestions and expressions for the business ecosystem. It can simply be identified as an environment of interacting organizations and individuals. According to James F. Moore's business ecosystem suggestions, a business ecosystem includes customers, lead producers, competitors, and other stakeholders [2]. The key entities of a business ecosystem are the leadership companies who are "keystone species" and have a strong influence over the co-evolutionary processes.

Moore describes the life cycle of a business ecosystem in four stages: birth, expansion, leadership, and self-renewal, or death. In the birth stage, the business ecosystem needs to do more than just satisfying customer expectations. In the expansion stage, the scale-up potential of the business concept is tested. In the leadership stage, the business ecosystem reaches stability and high profitability. The final stage is caused by the threat of rising new ecosystems. Another important point indicated by Moore is that the major difference between ecological and social systems is the role of conscious choice. In ecological systems, animals can choose their habitats, mates, and behavior. In economic systems, policy-makers, managers, and investors spend a lot of time understanding the situation, giving decisions and contemplating the possible outcomes of different choices.

On the other hand, a business ecosystem can also be evaluated from a biological environment perspective. Let us consider a biological ecosystem in depth in which each species has an important role. If any considerable change in the amount of any species occurs, the ecosystem will possibly experience serious problems. To make all species live in harmony, as new species enter into an ecosystem, they need to adapt themselves to the physical conditions within the environment. Moreover, the environment must have the right kind of and enough food for all existing species so they can grow and reproduce within the ecosystem. Another important point is that the amount of deaths should not exceed the number of births in the long run otherwise it may lead to the elimination of the species. This indeed is required for the survival of the ecosystem. When the deaths exceed births, the species in the ecosystem starts to be exhausted over time. This can occur also when a new species enters the ecosystem; a new species may start to eat some species in an uncontrolled way. In short, the balance within the ecosystem is affected, and if the ecosystem is healthy, all species can thrive; if the ecosystem is unhealthy, all species suffer deeply.

Similarly, in a business ecosystem all participating actors need to adapt themselves to market situations and circumstances. The entities that cannot adapt themselves and cannot cope with competitors are excluded and sidelined from the ecosystem in the short run. Also, the actors involved in the ecosystem need to produce with the help of resource suppliers and deliver value to the customers continuously. The competition between the existing actors must continue when new partners or actors enter the business ecosystem. The business ecosystem can also be appreciated and realized more from such a biological ecosystem perspective.

Another study focuses on the critical success factors of a business ecosystem which can also provide useful understanding for the NFC business ecosystem. The first factor is productivity which is a very basic factor and defines the success of any kind of business. In the NFC context, this may be referred to as successful roll-out of NFC technology and applications as it will be

explained later on. The second factor is the robustness of the business ecosystem. Robustness in the natural ecosystem expresses the capabilities of surviving when shocks from inside or outside the ecosystem threaten (e.g., unprecedented moves in the NFC market or introduction of better technologies into the market) to destroy it. Actually robustness in business life means drawing competitive advantage from many sources and having the ability to adapt when the environment changes. Finally, a business ecosystem should have the ability to create niches and opportunities for new firms (e.g., transportation authority playing the Trusted Service Manager (TSM) as a neutral key role). This requires a change in attitudes from protectionist to cooperative of the stakeholders and other players in the ecosystem.

7.1.1 Generic Features of a Business Ecosystem

In accordance with [1–3], some major features of a typical business ecosystem are presented. It is shown that a business ecosystem develops through complexity, self-organization, emergence, co-evolution, and adaptation, which are also relevant with an NFC business ecosystem. These terms are briefly explained below:

- *Complexity:* Business ecosystem in itself actually is a complex environment. Complexity can be defined in various perspectives. According to [1], a complex system is one whose properties are not fully explained by an understanding of its parts. Understanding dynamics of business ecosystem, requirements, participating entities and their roles in the ecosystem is really a complex issue that needs to be handled.
- *Self-organization:* Self-organization is a process in which "novel structures or features arise in a system without the intervention of an outside or inside controller". Self-organization is an ongoing process, since it will never complete its final outcome. It is the "self" that organizes. The lack of an outside or inside controller is the key to self-organization. The formation of a business ecosystem is a process where participants are gathered voluntarily and without an external or internal leader. Goals are set in local interactions, where organizations negotiate and create new order.
- *Emergence:* In accordance with [1, 3], emergence is the process that creates new order together with self-organization. The potential to create new order is the most important feature of complex evolving systems. Emergence is also identified as a "surprise-generating mechanism dependent on connectivity" by Casti [3]. It is important to remember that a business ecosystem is always more than the sum of its parts. It means that the result of interactions between different units is something that cannot be produced by only one entity.
- *Co-evolution:* Co-evolution in business ecosystems can be described as the evolution of one company affecting the development of other companies. Simply, it is the mutual changes of involving organizations in the ecosystem. For example, let us consider the NFC industry. When NFC chip manufacturers produce more efficient chips, NFC enabled reader producers quickly make use of the new opportunities. The strategic movements of a company affect other companies' movements and attitudes in the market.
- *Adaptation:* A business ecosystem should also adapt itself to external environmental changes and constraints, for instance to governmental and legal restrictions on NFC enabled mobile payment procedures. When the environment changes, a business ecosystem adapts to changed conditions by emergence, co-evolution and self-organization.

7.1.2 Business Ecosystem of NFC

From the ecosystem point of view, the NFC industry is a new emerging business environment and large value chain including several industries and organizations. The main industries that have important roles in the NFC ecosystem are Mobile Network Operators (MNOs), banking and payment companies, semiconductors and electronic appliances including mobile handset makers, software developers, and other merchants including transport operators and retailers.

The potential of NFC technology in business opportunities (especially in the mobile financial services industry) has impressed many organizations. Since NFC technology is made up of several components, it cuts across the boundaries of many organizations from diverse business sectors. All parties have already agreed that NFC technology cannot be provided by a single firm that could develop all the technology including infrastructural and service requirements.

NFC Forum, currently including 900+ members, is one of the major organizations in the world who aims to coordinate all participating institutions in the development of NFC based projects by offering interoperable technological specifications. According to participants and observers of the technology, NFC take-off has been slower than expected. The main reason for this slow take-off is strongly related to the lack of formation of a common understanding and vision in NFC technology among participating organizations and industries. Thus a mutually beneficial business model could not have been sustained yet. The main reasons for this lack of common understanding and vision are summarized below:

- The profit that will be shared is enormous.
- Participating organizations are powerful companies, hence they think that all other parties must follow their needs.
- Different technical solutions and infrastructures exist for each NFC enabled service. Hence, each actor may propose a different model which brings more advantage to them than others. For example, MNOs propose SIM based models, since they can control the UICC cards and hence can receive more profit if this model is used.

To achieve a good business model, interoperability, compatibility and standardization of the accepted NFC technology model is essential. It is also important to drive the cooperation of partners in the ecosystem and also to enable customer acceptance. Currently there is a tremendous amount of work being done on organizing the contributions and interests of all entities, and better governance of the overall ecosystem.

This chapter will concisely cover NFC business ecosystem stakeholders, and the business models. This will be beneficial for the reader to see the big picture, and understand the complexity of the technology in the meantime.

7.2 Stakeholders in NFC Ecosystem

NFC has a wide range of stakeholders or actors, depending on the type of provided services such as smart poster, payment, ticketing, and so on. The type of NFC enabled service defines the complexity in terms of which business model is applied, which stakeholders are involved, the appropriate collaboration model, the revenue model among players, and so on. NFC is a new technology and it is essential to perform more than justifying customer needs and expectations.

Figure 7.1 NFC ecosystem view by NFC Forum. Reproduced by permission of NFC Forum.

Currently, mobile financial services are the most promising opportunities in the NFC context. Also, they have highest complexity from both technological and business aspects with respect to other NFC services such as smart poster services, social networking, games, and so on. Various NFC ecosystem views are defined by standardization organizations. We provide two examples of those views. NFC Forum identifies the NFC ecosystem depending on and relating to its member portfolio (see Figure 7.1). On the other hand, GSMA describes and visualizes the ecosystem from the subjective point of view of MNOs (see Figure 7.2). However, the key players in an NFC ecosystem are mostly the same in both models.

Depending on the different approaches provided by organizations, this section analyzes and identifies the main stakeholders (see Figure 7.3), their roles, and responsibilities to give an insight into the NFC industry.

7.2.1 Standardization Bodies and Other Contributors

As already mentioned in Chapter 3, there are various standardization bodies and organizations in the NFC ecosystem with relatively different missions, business goals, and interests such as NFC Forum, GlobalPlatform, GSMA and EMVCo. Some of those business goals match each other, but there is a large number of conflicting aims as well. The major standardization bodies of the NFC ecosystem have already been presented in Chapter 3. Those bodies aim to develop

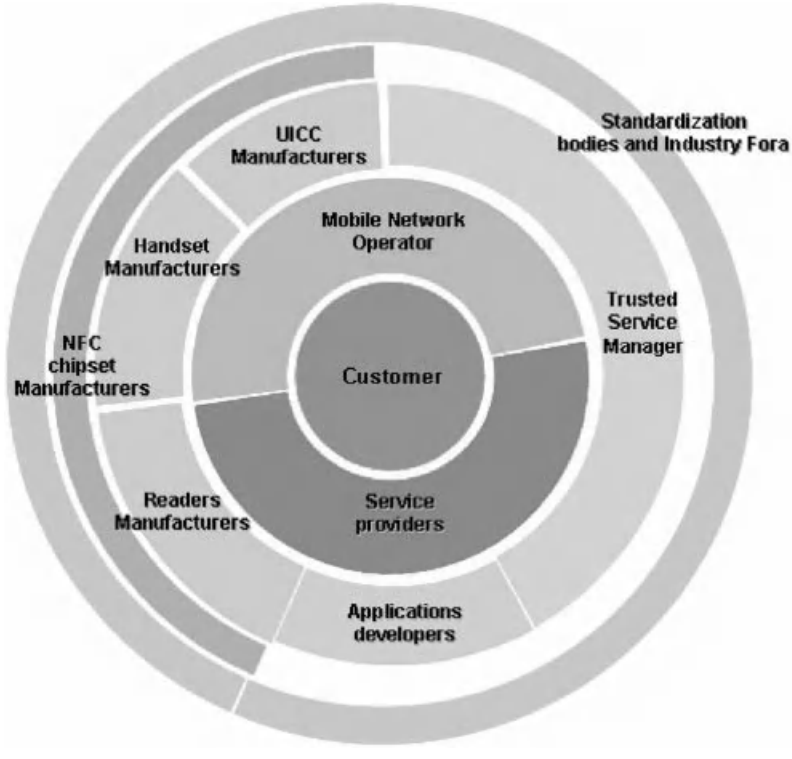


Figure 7.2 NFC ecosystem view by GSMA. Reproduced by permission of GSMA. All rights reserved.

global and interoperable standards for NFC, its dependent technologies such as smart cards, and mobile phones. Table 7.1 describes other minor associations which also have beneficial contributions to the development of NFC.

7.2.2 NFC Chip Set Manufacturers and Suppliers

NFC chip set manufacturers and suppliers provide standards conforming NFC chip sets and other related hardware to integrate NFC technology into the mobile handsets. NXP Semiconductors, as the co-inventor of NFC technology along with Sony, is still an important NFC chip set manufacturer and supplier in the NFC business ecosystem. It was originally founded by Philips and sold to a consortium in 2006. The company manufactures and sells NFC chip sets as well as MIFARE contactless smart cards which are commonly used by various transit systems all over the world.

7.2.3 Secure Element Manufacturers and Suppliers

As already described in Chapter 3, the Secure Element (SE) has a central role in NFC based systems, especially applications that use card emulation mode. SE manufacturers and suppliers

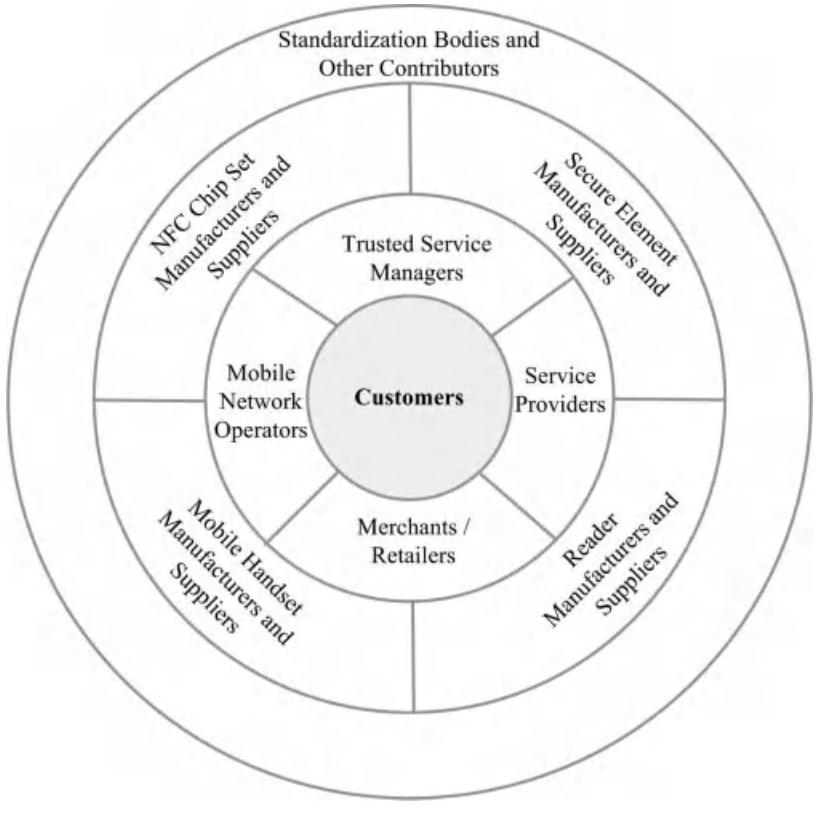


Figure 7.3 Stakeholders in the NFC ecosystem.

Table 7.1 Other active contributors to NFC technology

Organization Name	Responsibility
NXP Semiconductors	Co-inventor of NFC technology along with Sony, participates in and supports development of NFC
Mobey Forum	Non-profit, global, financial industry-driven forum, who encourages the use of mobile technology in financial services
Payment Firms (American Express, Discover, MasterCard, Visa)	Provides specifications on mobile payments as well as security and functionality qualification issues for the applications
Smart Card Alliance	Non-profit association who works to stimulate the understanding, adoption and widespread application of smart card technology
Stolpan	Association who contributes to the establishment of an open, interoperable NFC service environment

need to provide standards conforming SEs. Embedded hardware and Secure Memory Card (SMC) based SEs are manufactured for mobile handset manufacturers or other retailers, and UICC based SEs for MNOs. UICC manufacturers and suppliers are contracted and forced to provide the required UICC hardware to the MNOs. They have a direct relationship with the MNOs, who also define the requirements for the UICCs and issue the SEs. Thus generally NFC based systems using UICC based SEs are more MNO centric business models.

7.2.4 Mobile Handset Manufacturers and Suppliers

The role of mobile handset manufacturers and suppliers is to provide standardized NFC enabled mobile handsets by incorporating an NFC chip set and SEs for NFC enabled applications and services. Mobile handset manufacturers obviously wish to make their mobile phone more attractive for the customers. Moreover mobile handset manufacturers can gain a competitive advantage by offering mobile phones that support payment and other attractive applications. Innovative mobile applications provide an opportunity for manufacturers to attract new customers and create additional business partnerships.

7.2.5 Reader Manufacturers and Suppliers

The role of the reader manufacturer is to provide NFC enabled readers for the merchants, retailers and other parties who want to apply NFC enabled applications in their stores such as payment, ticketing, and transport applications. These readers are similar to Point of Sale (POS) terminals used in contactless payment transactions done by contactless payment cards.

7.2.6 Mobile Network Operators

MNOs traditionally provide communication and data network to the mobile phone owners. They are actually currently responsible for – indeed have the privilege of – providing all kinds of mobile services to their subscribers. MNOs can provide and maintain the network infrastructure that enables the secure Over the Air (OTA) solutions to provide remote management and maintenance of applications stored on SEs.

MNOs have an important role in defining the business models. The issues on SE ownership and provision of OTA platforms have direct impacts on NFC business models. Today, MNOs are trying to get a big market share, and up to now MNOs are at the center of the business environment in most trials and projects, control the life cycle of SEs, and manage SE platforms by their own OTA solutions. As usual, in these trials, UICC based SEs are mostly used. Hence, the customers of that business model need to be a subscriber of that MNO. Such an environment creates dependency on a single MNO since that service provider has no collaboration with any other MNO.

7.2.7 Trusted Service Managers

A TSM is required to create and manage a trusted environment among actors of the NFC ecosystem, mainly between service providers and MNOs. Integrating TSM into the ecosystem

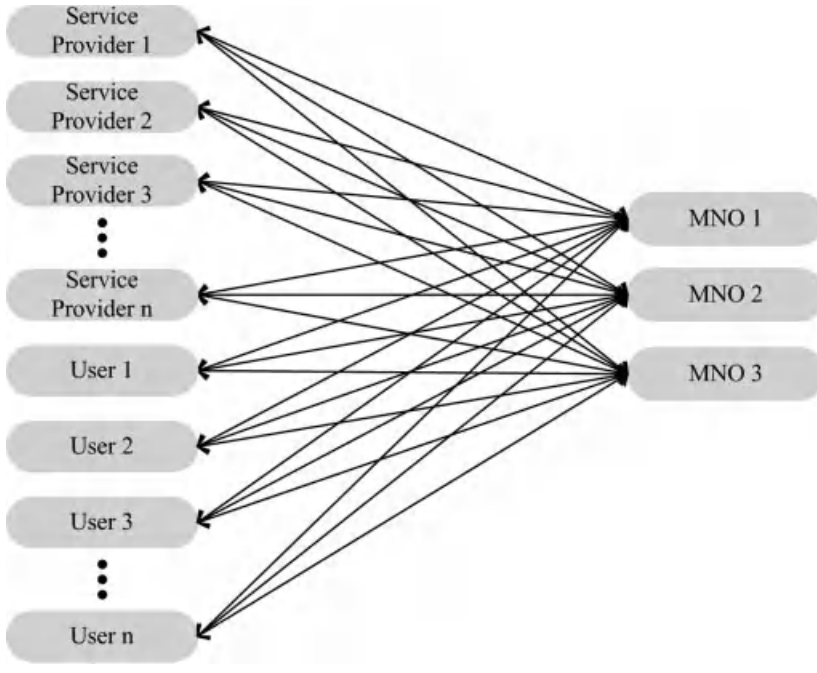


Figure 7.4 Chaotic nature of a business environment without TSM.

enables secure communication and interest protection of each entity, and also reduces the complexity of business models.

On the other hand in order to provide simplicity,, the players need to be in touch with each other which creates a complex communication environment when no TSM is used (see Figure 7.4). As an alternative, TSM plays a central authority role in such a system and eliminates the complexities (see Figure 7.5). TSM generally offers a single point of contact with MNOs for service providers such as financial institutions, banks, transit authorities, retailers and others who want to provide NFC enabled payment, ticketing, loyalty services and so on to customers.

Also, TSM is able to provide its own OTA platform and business solutions. This allows sending and loading of NFC enabled applications via its own OTA link to the SE and managing the life cycle of the SE at the same time. Especially in large NFC based systems, using a TSM as a central and neutral trusted entity is beneficial.

This is neither a healthy nor sustainable situation though. The roles of the TSM need to be defined and agreed between the service providers and MNOs in order to provide efficient mobile NFC services to the customers. The TSM's role becomes important especially in mobile financial service usage cases. MasterCard and Visa, being major payment brands and hence service providers, have strict requirements for the organizations that wish to act as TSMs. Many specifications and standards are already published about the TSM's role in mobile financial services.

CASSIS International, as one of the important TSM infrastructure and platform providers, was established by a team of visionary entrepreneurs and smart card veterans. CASSIS OTA

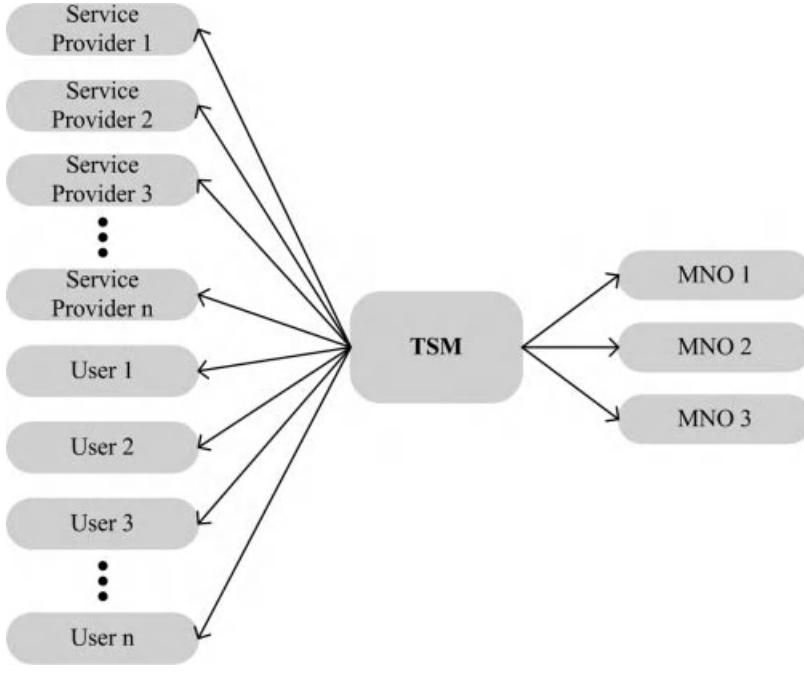


Figure 7.5 TSM role.

platforms provide the management of SEs, keys for all kinds of NFC enabled applications, and payment services. CASSIS has strong business partnerships with many leading banks, MNOs and transit operators in areas like Asia Pacific, Europe and the USA. A good example from CASSIS is the business partnership with Interbank Card Center (BKM), the organization that enables credit card transactions across different banks in Turkey. Currently BKM is trying to be the TSM of the Turkish banking industry in the NFC enabled mobile financial services domain and invites all Turkish banks who want to deploy NFC enabled payment services to create a common vision. CASSIS provides TSM business solutions and the infrastructure to enable BKM to be the first inter-banking domestic TSM in Turkey. BKM has agreed with some banks and MNOs to provide this service.

7.2.8 Service Providers

A service provider is the entity who wishes to deploy a service to the customer's mobile device as well as managing it. A service provider may be a financial institution, bank, transport authority or other organization. A service provider may concurrently take the responsibility of an application developer, owner and provider as well.

Due to the current attractiveness of mobile payment services, financial institutions or banks (e.g., Visa, MasterCard, HSBC, ING Bank) are the core service providers in the NFC business ecosystem. These service providers are responsible for offering new and differentiated trusted payment services to their customers, and increasing the credit and debit transaction volumes

as well as improving their brands. From their point of view, it is believed that NFC technology as a new form of payment method can increase the amount of payments made by credit cards with the widespread usage of NFC enabled mobile phones.

7.2.9 Merchants/Retailers

In the NFC ecosystem context, merchants and retailers are the stakeholders who are accepting contactless proximity mobile payment services. Merchants have the opportunity to speed up the payments at the POS. They own a contactless POS device system and financial networks for a specific bank or financial institution. A micropayment offers immediate benefits to merchants such as operational efficiencies through faster transactions and fewer requirements to handle cash, lower labor costs, and increased customer convenience at the same time. Micropayment can also help merchants establish stronger customer relationships and customer loyalty.

Merchants, like banks, can offer their own loyalty services to their customers. Also, they can offer their gift cards as well as loyalty programs since customers' credit and debit cards are always available on their mobile phones. Furthermore, merchants can deliver advanced mobile marketing and promotion programs by delivering messages to customers or members influencing their behavior to use mobile payment. According to the results of pilots and other studies of some NFC enabled mobile payment pilots, it has been observed that mobile promotions and couponing were attractive options and have a positive influence on consumer behavior.

7.2.10 Customers

A customer is the initiator of an NFC service by touching her mobile phone to a smart NFC object. Customers are always the principal stakeholders in any business and are the focus of the service providers. The reason for this is obvious. Customers will sustain the business only if their needs are met. All marketing activities always focus on them. Consumers at retail stores, banks, airports, theatres, hotels, hospitals or using public transportation are the core of this industry who want to financially benefit from NFC applications and services.

7.3 Business Models

When a new service is to be promoted, it is obvious that there are many issues to be handled and problems to be solved. Most issues are business related with respect to relatively few technology and infrastructure items involved. In order to propose a new service, it is very important to work out the necessary and suitable business models and processes. In business models it is important to deliver value to all stakeholders. Some business models do not encourage cooperation between all bodies; however, in NFC this is essential.

New service channels and concepts lead to new business requirements as in NFC. For the construction of an efficient business model, a proper analysis of the customers and industry is required. A good example for NFC business requirement analysis is provided by the Mobey Forum [4]. Mobey Forum proposes a number of requirements (i.e., initial, operational, usability, and external requirements) that need to be considered when forming business models.

Figure 7.6 Business model requirements [5]. Reproduced by permission of Mobey Forum.

Figure 7.6 summarizes these requirements in four categories. At first, the requirements were identified separately for consumers and for each industry in the NFC ecosystem. Then it was seen that the requirements of different sectors are indeed very similar, and a list of common requirements was created (see Figure 7.6).

Let us consider the business case requirement in business model creation. Both consumers and industries need to have their own reasons and objectives to initiate an NFC service or project, or participate in an NFC business environment. In terms of an industry, for instance transportation service providers, new business opportunities and cost savings are good business cases to participate in an NFC environment. With the replacement of current processes in transportation, new revenue streams can emerge for MNOs, customers and other service providers. In the case of customers, each consumer entering this ecosystem and starting to use NFC technology can be affected by some motivating factors and drivers. The business case behind customers enables them to accept this new technology and to use it.

The NFC ecosystem includes a wide range of applications and services from a vast amount of service providers in many sectors. All participants in the ecosystem are representing diverse business requirements and interests. Thus, most NFC services result in different collaboration models in a business context. Various real collaboration cases are presented in Chapter 9.

Currently, there is a pervasive amount of uncertainty on which business model is the best, which firm will perform exactly which activity, and who will pay who for which service, as well as how much profit is to be earned and shared by any stakeholder. Due to the novelty of NFC technology, there is still no agreement on a business model that satisfies sufficiently all stakeholders. Many business model proposals and specifications have been published and various projects and trials have been performed throughout the world, especially in mobile financial services due to the high degree of complexity in the ecosystem and the technological infrastructure. NFC Forum, GlobalPlatform, GSMA, and EMVCo are some of the important associations who intensively work on the NFC ecosystem, business models and underlying technological infrastructure.

In short, it is essential to harmonize the interests of all participants in forming business models. Without this, conflicting and not interoperable solutions will be performed, and hence the technology will not have a chance to be improved. In the next section, the main technical

issues and concerns that shape business models and the responsibilities of the stakeholders are presented and described.

7.3.1 Key Indicators in NFC Business Models

Up to now the market has adopted several SE options which are supported in different market segments and in different handset models. Currently there are three popular models (i.e., embedded hardware, SMC and UICC). These three SE alternatives are discussed and presented in technical detail in Chapter 3. As already known, mobile handsets need a SE to securely store and process sensitive data and facilitate privacy required applications. Some questions that are important for NFC business models are:

- Who will issue and own the control of the SE?
- Who will manage the life cycle of the SE platform?
- Whose OTA platform will be used for management of the SE platform?

Relating to these questions, in this section we identify three main issues that determine business model alternatives for NFC: SE issuer, platform manager, and OTA provider. These issues can also be referred to as functional roles and responsibilities that need to be handled by a single entity or multiple entities in the NFC business model.

(i) *SE Issuer*

The SE issuer is the entity who issues and owns the control of the SE that is supplied to the end user. The SE issuer's role is held by different actors based on the type of SE used. In an ideal case, the SE issuer should be an independent actor, but this is not so in real cases or implementations. Currently, it is not surprising to see that MNOs or service providers (e.g., banks) concurrently play the role of SE Issuer.

Figure 7.7 illustrates the SE issuer alternatives depending on the SE models. If a UICC based SE is used in a business model, the MNO holds the responsibility of the SE issuer, since MNOs distribute UICC. If an embedded hardware based SE is used, the actor who gives the mobile phone to the user plays the role. If the mobile phone is supplied to the current customer by an MNO as part of a campaign for example, the MNO is the card issuer. If the mobile is purchased by the user from a retailer or from a service provider, the card issuer can theoretically be any independent partner such as the retailer, service provider or other. If an SMC based SE is used, it can be issued either by an MNO, a service provider or a retailer.

(ii) *Platform Manager*

The second important issue is controlling and managing the SE platform. The platform manager owns the cryptographic keys used to control the SE in its life cycle. All SEs own a master key which is generated during the SE personalization process. The master key is a unique value in order to provide security mechanisms such as secrecy, authentication, data integrity, and non-repudiation. The actor who owns the master key has full control over the SE and this key can only be changed by the platform manager. The platform manager allows authorized service providers to install an application on the SE, preferably using an OTA infrastructure.

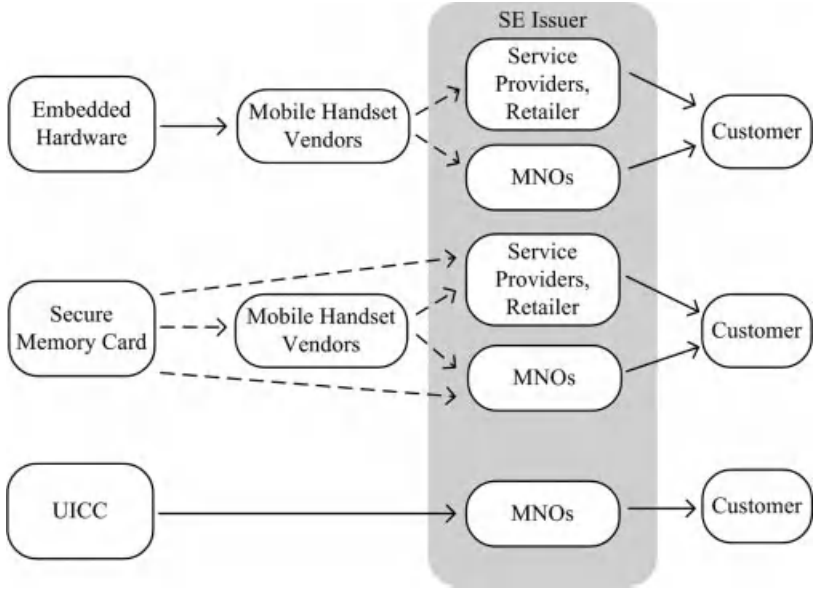


Figure 7.7 Secure element issuer models [6]. Reproduced by permission of Mobey Forum.

The platform manager may use the master key to divide storage platform (i.e., SE) into secure domains as required, and delegate control of each secure domain to the related service provider, thus enabling the provider to manage the application's life cycle. Service providers may manage life cycles of their respective applications. The platform manager can also disable a platform or part of the platform (i.e., security domain) when necessary. The platform manager has no access to application specific information, because after the reservation of a specific secure domain to a service provider, the provider has full control of it.

The platform manager role can be played by the SE issuer or another trusted, independent entity. Remember that currently different models are possible here and the platform manager role can be played by different organizations. Let us consider a case where a UICC based SE is used and the MNO plays normally the SE issuer role. The platform manager role can be carried out by that MNO at the same time, and the MNO owns the master key and hence controls the life cycle of the UICC. Service providers need to make business agreements with the MNO to deploy applications and the business model becomes completely MNO centric.

Actually the business model is simpler when the SE issuer and platform manager are the same entity. Delegating the platform manager role (i.e., transferring the master key of the SE) to some other party than the SE issuer is yet another option. This entity can be a TSM as an independent neutral party. In order to realize this option, the TSM and SE issuer should previously agree on the business model. The agreement would possibly consist of details from technical infrastructure to revenue sharing. The TSM can handle all SE management functions by using its own OTA platform or MNO OTA link.

Mobey Forum has proposed three business models depending on the ownership of UICC based SEs. It can be owned by one single organization or co-owned, issued by one single organization or co-issued, managed by one single organization or co-managed. These models are: the hotel concept model, the rental building concept model and the ownership concept model. In the hotel concept model, SE and all related facilities such as keys and security domain management are owned, controlled, and managed by one entity such as the hotel owner who is the SE issuer at the same time. In the case of the rental building concept, the SE is issued and owned by one organization but managed by a third party such as a TSM. The SE issuer as a building owner may choose to hire a third party to manage and organize services in the building (i.e., SE). In the ownership concept model, one or more organizations control the issuing process of the SE and each stakeholder that has a space on the SE can participate in management of the SE life cycle. The details about the operational and business models of Mobey Forum can be found elsewhere [4].

(iii) *OTA Provider*

The final issue is provision of the OTA platform. Providing a flexible and interoperable OTA solution is a key requirement in an NFC ecosystem. The technical details of OTA technology are presented in Chapter 8. It enables secure wireless communication between two end parties, and provides transmission and reception of application related information in a wireless communication system. OTA technology enables remote download, installation and management of applications such as updating, activating or deactivating applications stored on the SE. Thus OTA providers also play a crucial role.

Currently, some MNOs worldwide are capable of providing OTA solutions using their current technology infrastructure. When MNOs act as platform managers, they use their own OTA link to manage the life cycle of SEs.

However, it is also possible for the required infrastructure to be set-up by other entities (e.g., service providers, TSM). Those entities can provide OTA service independently from SE issuers or platform managers. The ideal and most appropriate case is to use an OTA solution of a TSM and make the TSM the platform leader. The TSM, as a neutral party between MNOs and service providers, handles the SE management fully using its own OTA link. Revenue sharing will become fair and interests will be harmonized.

7.3.2 Business Model Alternatives

This section gives a basic knowledge and understanding of business models proposed and applied in the industry so far. We present some collaboration models that have been discussed, studied in the literature or applied in practice. Currently available models are MNO centric, distributed, and TSM centric business models. In Section 7.5 a successful project example from GSMA is presented together with its business models.

(i) *MNO centric business model*

In the MNO centric business model, MNO issues SEs and acts as SE issuer, platform manager, and OTA provider. There is no independent trusted entity. Thus, the MNO performs all functions of the TSM; it owns and manages loading, installation, and

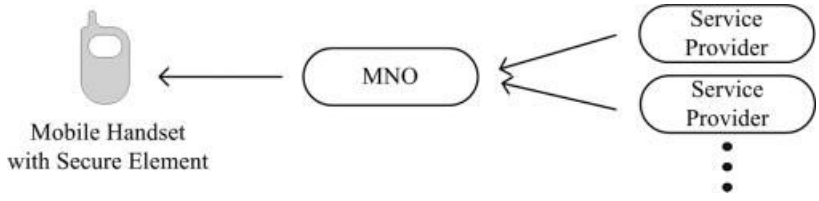


Figure 7.8 MNO centric business model.

personalization processes, as well as security domain creation on the SE. Service providers have to share their personalization data with the MNO.

Figure 7.8 shows the case where there exists only one MNO in the market for a specific service such as payment. All service providers (i.e., banks) sign business agreements with the unique MNO in the market. Actually the number of MNOs involved can increase for that specific service environment. A service provider has to work with other MNOs and their issued SEs. When a new MNO enters the industry for that service, obviously the service provider can sign a business agreement with that MNO in order to reach all potential customers in the market.

(ii) *Distributed business model*

In the distributed business model, the platform management services are distributed among SE issuers and service providers. There can be a separate TSM actor, and using the MNO OTA infrastructure is still another option as well.

If a service provider has no TSM capabilities, it needs to make an agreement with an existing TSM. Also the service provider has to share application related data with the TSM. Currently, service providers prefer to collaborate with trusted third parties rather than making high investments for building a TSM platform within their organizations. Thus, in a distributed business model we assume that each makes agreement with an existing TSM, and this Trusted Third Party (TTP) performs all TSM capabilities on behalf of the service provider.

In this model, the SE issuer role is mostly performed by the MNO who performs application loading and installation on the used SE. The used SE alternative does not have to be only a UICC based SE. The MNO also can perform lock, unlock, and delete processes as well as security domain creation on the card. For these processes the MNO uses its own OTA platform infrastructure. The service provider performs data preparation and personalization processes on the SE platform. The service provider can use the MNO's OTA or the TSM's OTA infrastructure. As a matter of fact, in this model the OTA provider role changes depending on the business agreement between the MNO and the service provider. However, in both cases, the SE platform is managed by two entities as illustrated in Figure 7.9.

The distributed business model is the most preferred, but not the most suitable business solution. This model actually creates a win–win situation in the ecosystem. However, it has some limitations on the customer side, depending on the SE option. In the case of UICC based SEs, the NFC service can be offered only to a limited number of service providers' customers who need to be subscribers of that MNO at the same time. To reach more customers, the service provider needs to sign and agree with other MNOs in the market as well, which creates other complexities.

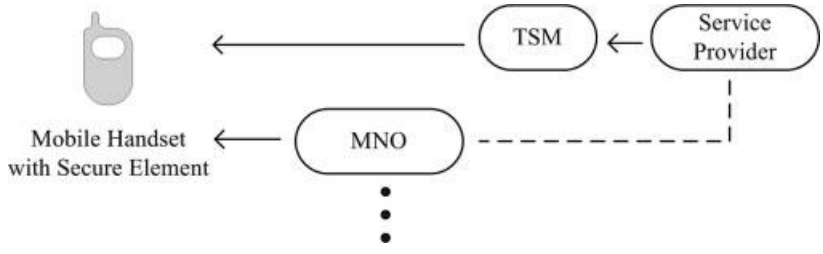


Figure 7.9 Distributed business model.

(iii) *TSM centric business model*

For an NFC service, a single TSM centric business model is actually the best option and less complex at the same time (Figure 7.10). The involvement of a TTP as a TSM for a specific NFC service reduces the complexity of the environment, when compared with other models. It also provides fair revenue distribution among MNOs and service providers. In the meantime, a wide range of users benefit from the NFC services which are provided by different service providers.

The number of TSMs may increase depending on the available services and the agreements of the actors in the NFC ecosystem. For instance, NFC enabled payment and transportation services may use the same as well as different TSM platforms. MNOs and service providers that want to participate in the NFC ecosystem need to sign and agree with the TSM. All entities should share required data with the authorized TSM in the ecosystem. The TSM performs the platform manager's role completely on behalf of the service providers by realizing loading, installation, and personalization processes via its own OTA platform.

(iv) *Summary of Business Models*

Figure 7.11 summarizes all the possible combinations of the three business models discussed above. In all defined business models, all SE options – UICC, SMC and embedded hardware based SEs – can be issued and managed. In the case of the MNO centric business model, all key indicators are handled by the MNO. In the case of distributed business model, the platform management is performed by the MNO and the service provider. During the platform management, the OTA platform used may differ depending on the business agreement between the service provider and the MNO.

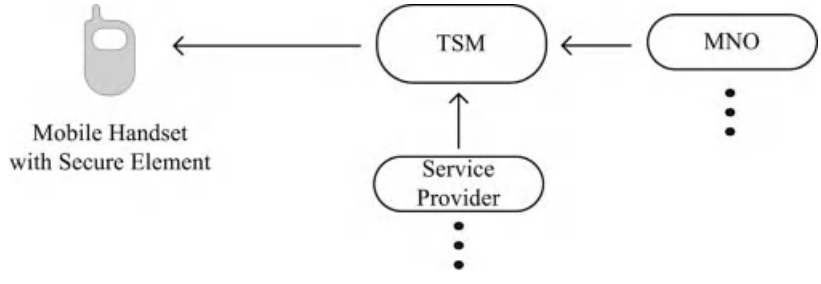


Figure 7.10 TSM centric business model.

SE Option	SE Issued By	Platform Management				Business Model
		Application Loading and Installation		Data Preparation and Personalization		
		Installation	OTA Provider	Personalization	OTA Provider	
All*	MNO	MNO	MNO	MNO	MNO	MNO Centric
All*	MNO	MNO	MNO	Service Provider	MNO	Distributed
All*	MNO	MNO	MNO	Service Provider	TSM	Distributed
All*	MNO	Service Provider	TSM	Service Provider	TSM	TSM Centric
SMC, Embedded Hardware	Service Provider or Retailer	Service Provider	TSM	Service Provider	TSM	TSM Centric

*"All" refers to embedded hardware, SMC or UICC based SEs

Figure 7.11 Summary of business model alternatives.

In the TSM centric business model, the TSM has full authorization over platform management and OTA provision in all phases. Actually the identity of the issuing party of the SE and the used SE option is not important. Authorization of SE management is given to the central TSM.

As already mentioned, service providers and other retailers can only issue SMCs and embedded hardware within mobile phones. An example is that of Akbank in Turkey, who deployed an NFC enabled Visa payWave micropayment service without dependency on an MNO in the market with a well qualified TSM.

According to Mobey Forum, it seems unlikely for MNOs to allow a third party to take over the platform management role of the UICC based SEs. Actually for quite a while, MNOs will definitely fully or partially take the platform management (i.e., management of SE life cycle) role in business models where UICC based SEs are used, and MNOs will be the only entities who issue them. Since SMC based SEs are more MNO independent and can be applied in all business models, we believe that they will be preferred and used more by service providers in the near future.

In some scenarios such as ticketing and couponing which involve additional service providers other than banks and financial institutions, the complexity of the ecosystem increases. In an optimal ecosystem, each service provider and MNO needs to sign a business agreement with one or more centralized platform managers in order to have its application uploaded onto SEs. This allows consumers to have access to all available services, and they can switch on and off any service that they choose, they can also switch to another operator at any time and reach that specific service easily.

7.3.3 General Revenue/Expenditure Flow Model

In terms of the revenue and expenditure aspect of the business model, Mobey Forum has performed an overall analysis of the revenue model in [4]. In order to extend this

Stakeholders	Revenue	Expenditure
Customer	•Gaining coupons and other benefits from financial and loyalty services	• NFC enabled mobile phone • Removable secure element; UICC, SMC • Monthly subscription fees to service providers, and other bills
Service Providers	•Monthly fees paid by customers as well as merchants, retailers	• Application development and other application related backend services • Maintenance services for applications • Customer care services
Mobile Network Operators	•Monthly fees paid by its subscribers	• New UICC issuances • Mobile network services • OTA management services depending on business model • Billing subscribers • Customer care services
Merchants / Retailers	•Increased sales (due to new customers or customer retention)	• Customer care services • Bank fees and other
Trusted Service Managers	•Fees paid by MNOs, service providers depending on business model	• TSM infrastructure services • OTA management solutions depending on business model • Customer care services

Figure 7.12 General revenue/expenditure flow between NFC stakeholders.

perspective, a useful understanding of simple cash flow among NFC stakeholders is summarized in Figure 7.12.

7.4 Case Study: NFC Ticketing

This section aims to present and describe a use case study of an NFC enabled ticketing service at the movie theater. This use case is also discussed and evaluated in Chapter 4 from a different aspect. As already stated, ticketing is a card emulation operating mode service. The operating issues and working principles of the movie ticketing service is presented and illustrated briefly in Chapter 4.

This section expresses and evaluates an NFC ticketing service's business environment depending on the proposed business models in Section 7.3. The services operating in reader/writer and peer-to-peer modes have a more distinct business environment than a service in the card emulation operating mode. Reader/writer mode and peer-to-peer mode services generally do

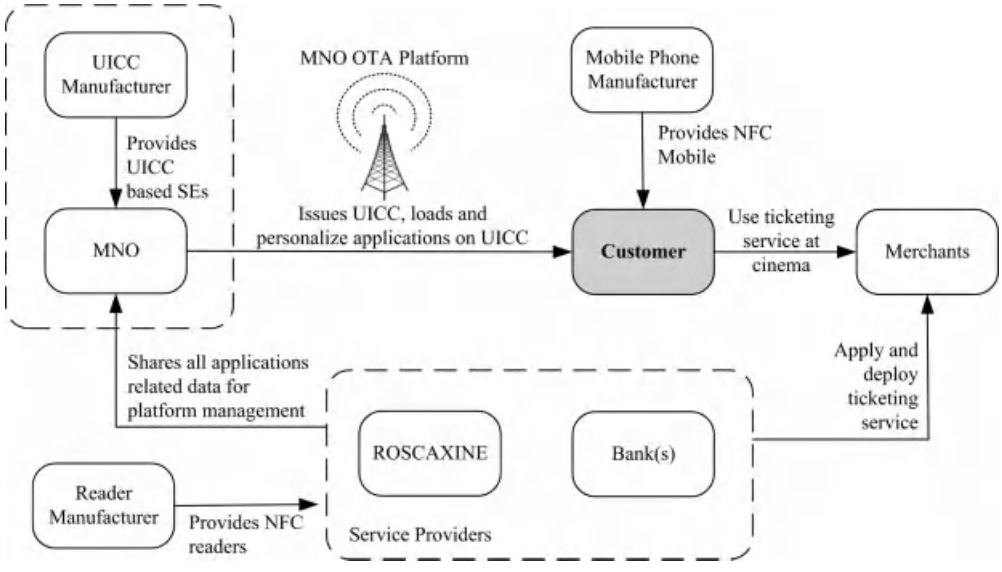


Figure 7.13 Business environment of MNO centric NFC ticketing case.

not need a complex and secure environment to be applied on a user's NFC enabled mobile phone. Applications operating in card emulation mode include a wide range of technical infrastructure alternatives, these applications need to be securely downloaded to the SE of a user's mobile phone, and security domains and other keys need to be generated on the SE. Then the life-cycle management of the SE platform and applications on it need to be continuously performed by the authorized actors via the appropriate OTA solution. The business environments of such services involve diverse stakeholders and business collaborations.

Remember that in our use case, the cinema company ROSCAXINE decides to deploy an NFC ticketing service using box offices in its entertainment zones. Users can select the movie and make seat arrangements by informing the ticket clerk at the box office. The user initially touches to the contactless NFC POS reader first to pay for the tickets, and second to obtain the tickets. The tickets are stored on her mobile device, so that when the user touches to the NFC readers at the turnstile, she can enter the cinema hall easily.

This case study applies NFC ticketing service using three different business model alternatives. Each alternative differs in the selected SE option and the key indicators defined in Section 7.3.1. Most of the discussed models in Figure 7.11 are visualized more with this case study, and the strengths and weaknesses of the stakeholders are presented briefly.

(i) *Case 1: MNO centric business model*

Figure 7.13 shows the business environment of the MNO centric NFC ticketing case and the relationships of actors involved in the ecosystem. In a typical ticketing case there are actually two major phases: the payment and the ticketing activities. As usual, the service provider is the bank who provides the NFC enabled payment service for getting a ticket, and the ticketing company such as ROSCAXINE who provides the e-ticket for the user.

Let us consider that ROSCAXINE decides to perform the NFC ticketing service in cooperation with a specific MNO in the market and uses UICC based SE to enable this service. ROSCAXINE agrees and signs business agreements with the MNO. The MNO may also have signed agreements with banks which provide payment services. Since the UICC based SE is used, the owner and issuer of the card is the MNO.

In the business agreement phase, all entities decide on the technical infrastructure of the ticketing service that will be deployed on the UICC card for the customers as well as decide on the revenue that will be shared.

ROSCAXINE's ticketing and the bank's payment applications are stored on the UICC of the customer. As the MNO plays the management role of entire platform, the application loading and personalization processes are performed by the MNO using its own OTA link. So, in order to perform OTA services there is no need for additional investment for service providers; the MNO can use its existing infrastructure.

In such a business model, the revenue is shared mainly between the MNO and the service providers. The main constraint of this service is the limited target audience who benefits from this service. Only the subscribers of that MNO and members of the agreed banks can benefit from this service without an additional cost (e.g., costs for memberships, etc.). If ROSCAXINE wants to reach more customers, it should sign business agreements with other MNOs in the market and use their UICC based SEs and OTA network. Then other operators' subscribers who are already members of the agreed bank can benefit from this service as well.

(ii) *Case 2: Distributed business model*

Let us assume again that ROSCAXINE decides to perform the NFC ticketing service together with a specific MNO in the market and uses UICC based SE to enable this service. As usual, UICC based SEs are issued and owned only by an MNO for application loading and installation in this scenario. However, the platform management process is shared by the MNO and the service providers which shape the distributed business model.

Figure 7.14 shows the relationships between the involved actors in the distributed business model of the NFC ticketing case when the UICC based SE alternative is used. First, ROSCAXINE agrees and signs business agreements with the MNO. It selects the technical infrastructure of the ticketing service when the UICC based SE and the distributed platform management model is used. The ticketing and payment applications' loading and installation phases are performed by the MNO on the SE via the MNO OTA channels. The service provider is responsible for the application personalization phase on the UICC. They usually do not have any existing TSM business solutions. Thus they should make business partnerships with a TSM within the industry at the same time.

To sum up, this model is MNO dependent as well. The revenue is shared mostly between the MNO and the service providers. The major benefit is acquired by service providers. This distributed model provides more control for service providers on their applications on the user's SE and they perform the service with their TSM OTA platform at the same time. The main constraint of this service is again the limited number of target customer who benefits from this service which is discussed in the first use case. Only the subscribers of that MNO and at the same time members of the agreed bank(s) can benefit from this service without any additional cost.

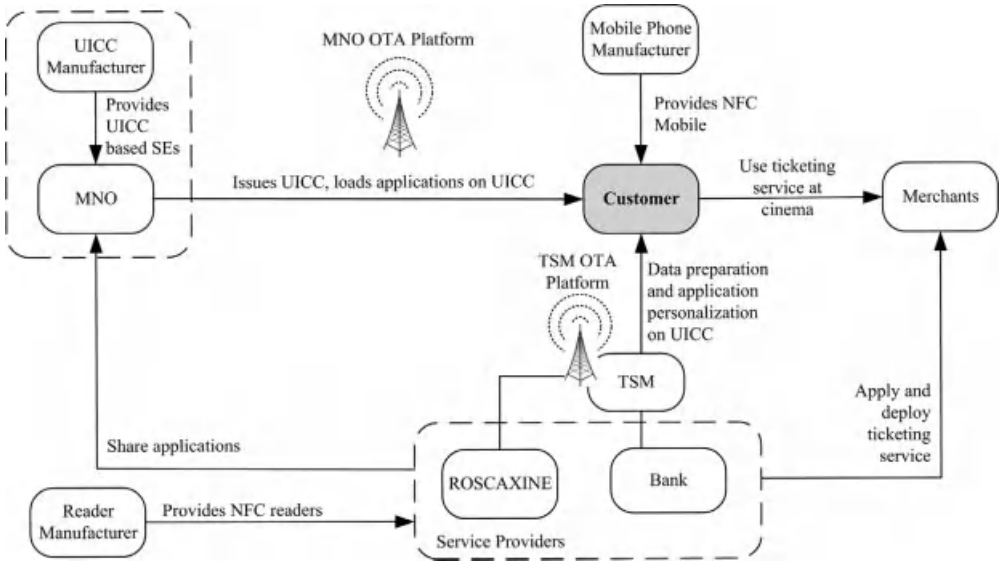


Figure 7.14 Business environment of distributed NFC ticketing case.

(iii) *Case 3: TSM centric business model*

Let us assume that SMC based SE is used in a user's NFC enabled mobile phone in this final scenario (see Figure 7.15). Service providers load the ticketing application and payment services on SMC. ROSCAXINE and the bank use a single TSM OTA platform and perform application loading, installation and personalization processes completely by using the TSM OTA infrastructure which provides independence from MNOs. Platform management is performed by the TSM.

This model's benefit is its independence from MNOs. All bank customers can use this service by obtaining the issued SMC. If ROSCAXINE wants to reach more customers, it can agree and work with other banks who want to deploy payment service on SMC based SEs for NFC ticketing at entertainment zones of ROSCAXINE.

7.5 Additional Reading: Pay-Buy-Mobile Project by GSMA

The Pay-Buy-Mobile project was launched by GSMA in February 2007 during the Mobile World Congress. This project was also published as a white paper called "Pay-Buy-Mobile Business Opportunity Analysis" that is publicly available on GSMA's official website [7]. The project's main objective is to create a common global vision for NFC enabled mobile payment services and achieve interoperability. More than 60 MNOs (Vodafone, Orange, Turkcell, etc.) are supporting this project, thus the proposed business models are obviously more MNO centric. Many Pay-Buy-Mobile NFC pilots and trials have been performed to enhance the NFC market partners. Trials have been carried out in many countries such as Australia, Canada, France, Japan, Korea, Malaysia, Norway, Philippines, Spain, Taiwan, Turkey and USA.

Pay-Buy-Mobile is an NFC enabled mobile payment service that is performed on UICC based SEs. Needless to say, GSMA promotes the idea that UICC is the most appropriate SE for NFC enabled mobile phones. They are universal (i.e., globally involved in more than 80% of

Table 7.2 Business models and TSM role in Pay-Buy-Mobile

Business Model	TSM Platform Provided By
MNO centric model	MNO
Independent entity model	Trusted third party
CIB centric model	CIB
Combination model	MNO-CIB-trusted third party

mobile phones), portable, standardized, and they enable dynamic remote management through OTA capability as explained in Chapter 8.

The main actors of Pay-Buy-Mobile are the TSM, MNOs, and Card Issuing Banks (CIBs). TSM is the link between the CIBs and the MNOs. CIBs as service providers enable the overall payment service and are responsible for customer care, issuance of payment application and the customer's personalization data, as well as establishing a formal agreement with the customer for the NFC enabled payment service.

It does not only link a single CIB to a single MNO but it is also an integrated point between multiple CIBs and multiple MNOs. It is also possible to have multiple TSMs within a Pay-Buy-Mobile ecosystem as well as having a combination of MNOs and CIBs performing the role of TSMs in the market. Four business models are proposed by Pay-Buy-Mobile project as shown in Table 7.2. Participating entities of the Pay-Buy-Mobile ecosystem can choose the most suitable business model to satisfy its market needs.

(i) *MNO centric model*

In this model, MNO forms and integrates TSM capabilities within its network infrastructure and offers a secure and open TSM platform to CIBs as seen in Figure 7.16. MNO can provide flexibility in customizing service provisioning for NFC enabled mobile services.

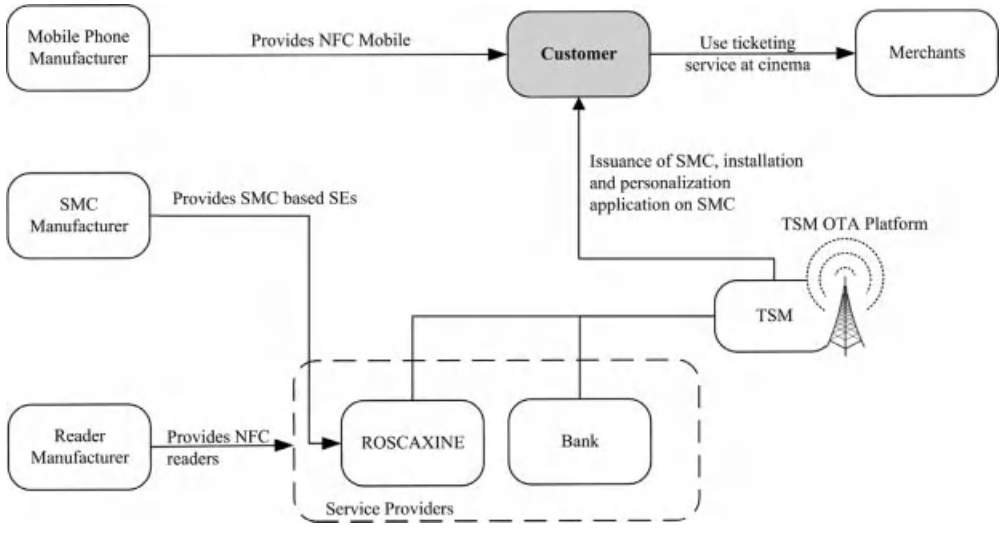


Figure 7.15 Business environment of TSM centric NFC ticketing case.

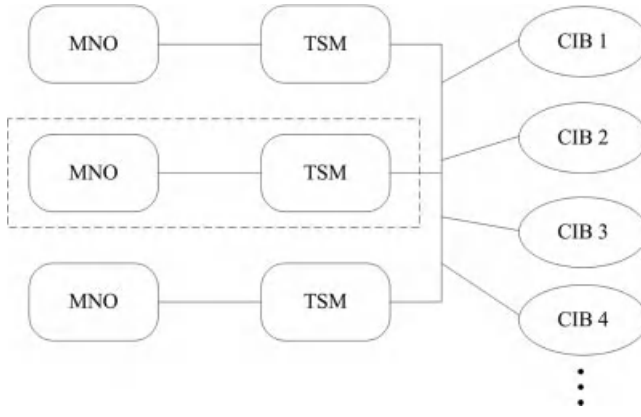


Figure 7.16 MNO centric model. Reproduced by permission of GSMA. All rights reserved.

MNOs should make a large investment in establishing a TSM infrastructure, if they do not have one already. On the other hand, CIBs should provide an NFC enabled payment infrastructure to enable Pay-Buy-Mobile. When compared with the other three business models in Pay-Buy-Mobile, CIBs need to make a relatively small investment in building the Pay-Buy-Mobile infrastructure, a significant part of the job is handled by the MNO. However, each CIB in the market needs to make a business agreement as well as it has to collaborate with each MNO to perform Pay-Buy-Mobile.

(ii) *Independent entity model*

In this model, the TTP is separate from all other actors, thus it is independent of all other players and performs TSM capabilities. The third party is the organization providing only TSM business solutions in the market who acts as the single, and neutral point of contact between MNOs and CIBs.

The TSM performs overall management of security domains on the UICC based SE of the user. This model reduces overall complexity in the Pay-Buy-Mobile ecosystem by providing an integrated interface between MNOs and CIBs. This model eliminates the need to make an investment in TSM infrastructure for MNOs as well as CIBs. To facilitate this model, each MNO and CIB in the market needs to make a business agreement with the third party as illustrated in Figure 7.17.

(iii) *CIB centric model*

In the CIB centric model, CIB performs all TSM capabilities and collaborates with participating MNOs in the market as illustrated in Figure 7.18. CIB needs to make a big investment in building the TSM platform, because it has additional high costs for the CIB. For small sized CIBs, this investment can be a barrier to enter the NFC enabled mobile payment market. From the MNOs viewpoint, they still provide UICC based SEs.

(iv) *Combination model*.

In the combination model, the role of the TSM can be played by any combination of MNO, CIB, and independent TTP entity. For example, as illustrated in Figure 7.19, participating MNOs and CIBs can jointly form a TSM infrastructure to serve their customers. Investments are shared between the initial founders of the TSM infrastructure.

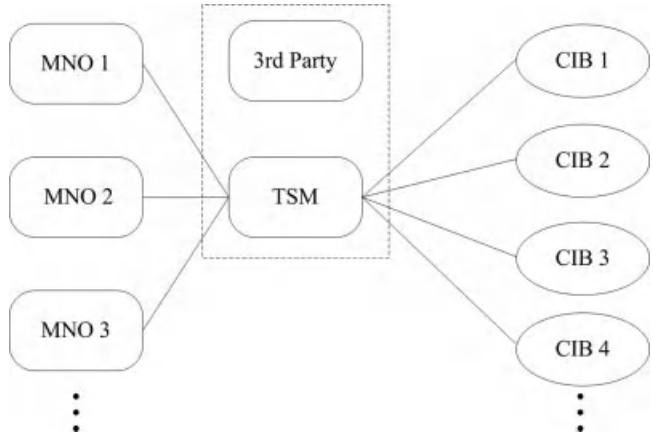


Figure 7.17 Independent entity model. Reproduced by permission of GSMA. All rights reserved.

After forming the TSM platform, new partners such as MNOs or CIBs can join the business network.

In order to create a sustainable model, it is really important to create a win–win business model for all stakeholders in the market with many additional revenue and marketing opportunities. The Pay-Buy-Mobile project triggers many UICC based NFC enabled mobile services. According to GSMA, the UICC is the main component in the payment business solution. It provides a secure and flexible channel for carrying out NFC enabled mobile payment transactions.

Another important component is the banking companies. GSMA has invited financial service communities to work with MNOs in order to achieve a common vision in the Pay-Buy-Mobile project. MNOs and the banks need to agree on a mutually beneficial and acceptable business models. In order to move forward successfully with all kind of business models, understanding

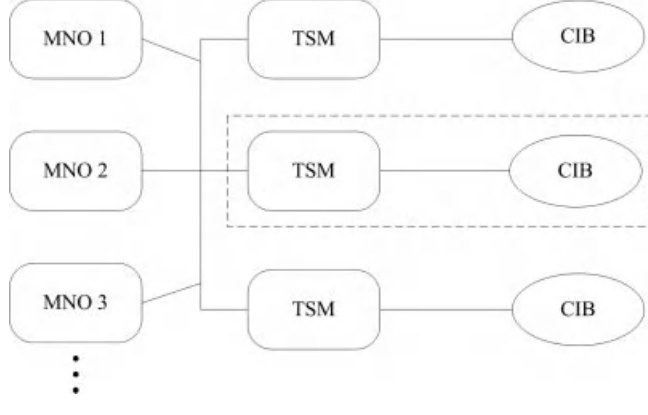


Figure 7.18 CIB centric model. Reproduced by permission of GSMA. All rights reserved.

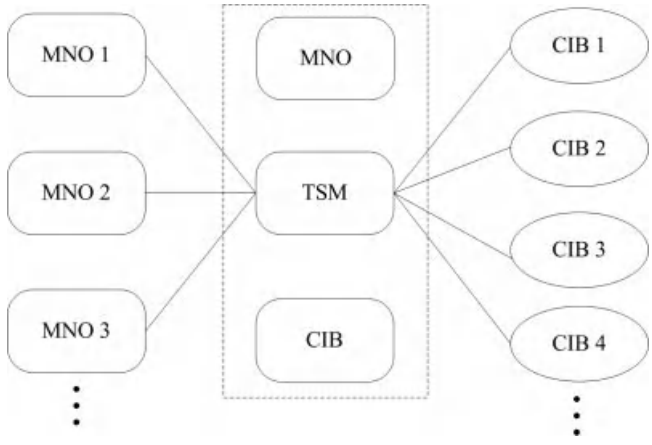


Figure 7.19 Combination model. Reproduced by permission of GSMA. All rights reserved.

the MNO's and the bank's business requirements, and establishing trust between them is required.

7.6 Chapter Summary

The term ecosystem has various meanings in different areas such as biology, sociology, economics, and business. One of the best descriptions for the business ecosystem is given by James F. Moore who defines it as an environment of interacting organizations and individuals which has its own life cycle. A business ecosystem can be also evaluated from a biological environment perspective too. In a business ecosystem all participating actors need to adapt themselves to the market situations and circumstances. The entities that cannot adapt and cannot cope with the competitors are excluded from the ecosystem within a short time. The continuous production and delivery of value to the customers is required for the survival of the ecosystem. If the ecosystem is healthy, all actors can succeed.

One important aspect that needs to be considered in understanding NFC technology is its business ecosystem. The NFC industry has a new emerging business ecosystem and value chain including several industries and organizations. The players are mainly standardization bodies and other contributors for NFC technology's development such as NFC chip set manufacturers and suppliers, mobile handset manufacturers and suppliers, reader manufacturers and suppliers, secure element manufacturers and suppliers, MNOs, TSMs, service providers, merchants or retailers, and finally customers.

In order to implement an ecosystem, it is very important to develop and process suitable business models. Some business models do not encourage cooperation of all bodies, however an NFC ecosystem needs various stakeholders and industries to collaborate. Business models must be developed to deliver profit to all stakeholders. Three main collaboration models are possible: MNO centric, distributed and TSM centric business models. In defining NFC business models, three key issues need to be handled: who will be the SE issuer depending on

the SE alternative, who will manage the life cycle of the SE and be a platform manager and, finally, who will be the OTA provider.

Many business model proposals and specifications have been published and various projects have been implemented through trials throughout the world, especially for mobile financial services due to the high degree of ecosystem and technological infrastructure complexity. NFC Forum, GlobalPlatform, GSMA, and EMVCo are some important associations. GSMA has carried out a valuable project called Pay-Buy-Mobile which has 60+ participating and supporting MNOs. Many Pay-Buy-Mobile NFC pilots and trials have been performed to enhance the NFC market in many countries. This project proposes four business models where participating entities can choose the most suitable one. The four models are: MNO centric model, independent entity centric model, CIB centric model, and combination model.

Due to the novelty of the NFC technology, there is still no common vision for a business model that satisfies all stakeholders. There is a pervasive uncertainty about which actor will perform exactly what, who will pay whom, as well as how much profit is to be earned by any stakeholder. It is essential that the technological infrastructure and secure element used in mobile handsets, OTA platform provider, and TSM infrastructure shape business models and responsibilities of the stakeholders in the NFC ecosystem. Understanding all stakeholders' business requirements and market needs as well as building trust among the stakeholders are essential issues for the development of the NFC ecosystem.

Chapter Questions

1. What is a business ecosystem? Explain.
2. List and explain the generic features of a business ecosystem in general.
3. List the major stakeholders of an NFC business ecosystem.
4. What is the importance of the TSM in an NFC business ecosystem?
5. Do you suggest using a central TSM in an NFC ecosystem? What are the comparative advantages?
6. What are the major indicators of business modeling in an NFC ecosystem?
7. Compare advantages and disadvantages of the MNO centric business model in terms of revenue and expenditure.
8. Compare advantages and disadvantages of the distributed business model in terms of revenue and expenditure.
9. Compare advantages and disadvantages of the TSM centric business model in terms of revenue and expenditure.
10. What is the most critical decision in the NFC ecosystem regarding the NFC card emulation mode?

References

[1] Peltoniemi, M. and Vuori, E. (2004) *Business Ecosystem as the New Approach to Complex Adaptive Business Environments*. Proceedings of Frontiers of E-Business Research, 2004.

[2] Moore, J. (1993) Predators and prey: The new ecology of competition *Harvard Business Review*, **71**(3), 75–86

[3] Casti, J.L. (1997) *Would-be Worlds: How Simulation Is Changing the Frontiers of Science*, John Wiley & Sons, Ltd, New York, ISBN: 978-0471196938

[4] Mobey Forum Enrollment Task Force (2008) *Best Practices for Mobile Financial Services*, MOBEY Forum, 2008. Available at: http://www.mobeyforum.org/content/download/460/2768/file/Best%20Practices%20for%20MFS%20Enrolment%20Business%20model%20analysis%20final.pdf (accessed 10 July 2011).

[5] Mobey Forum Enrollment Task Force (2008) *Best Practices for Mobile Financial Services*, Table 4.1, MOBEY Forum, 2008. Available at: http://www.mobeyforum.org/content/download/460/2768/file/Best%20Practices%20for%20MFS%20Enrolment%20Business%20model%20analysis%20final.pdf (accessed 10 July 2011).

[6] Mobey Forum Enrollment Task Force (2008) *Best Practices for Mobile Financial Services*, Figure 6.2, MOBEY Forum, 2008. Available at: http://www.mobeyforum.org/content/download/460/2768/file/Best%20Practices%20for%20MFS%20Enrolment%20Business%20model%20analysis%20final.pdf (accessed 10 July 2011).

[7] GSMA (2007) *Pay-Buy Mobile Business Opportunity Analysis*, Version 1.0, November 2007, White Paper. Available at: http://www.gsmworld.com/documents/gsma_nfc_tech_guide_vs1.pdf (accessed 10 July 2011).

8

Secure Element Management

The NFC ecosystem is tremendously dynamic. NFC enabled applications are frequently created, downloaded by the NFC enabled mobile phone, installed either on mobile phones or the Secure Element (SE), configured and maintained thereafter by the related actors of the ecosystem. As discussed in Chapter 3, currently the best alternative for managing the SE, including application installation and remote management, is by use of the Universal Integrated Circuit Card (UICC). UICC provides appropriate card structure based on GlobalPlatform's card specification that allows multiple security domains for different applications on the same card. Each security domain is actually a secure space for an NFC application where the unique security key of the application is also stored in order to control access to the secure domain by the authorized party. NFC enabled mobile handsets with appropriate SEs can be accessed by Over-the-Air (OTA) technology which is a promising and effective way to reach the phone regardless of the location of the user. Usage of OTA enables remote installation, personalization, and life cycle management of NFC applications in SE. This means the user does not need to physically touch her NFC enabled mobile device to a system in order to perform any of the processes mentioned. This section focuses on SE management and the role of OTA technology. OTA deployment models on UICC based SEs and management of multiple SE environments in accordance with GlobalPlatform specifications are also presented.

8.1 Introduction to OTA Technology

The operating system controlling the SE within the mobile device must allow installation, personalization and management of multiple applications – each of which is possibly issued by different providers – via OTA technology. OTA is a standard for transmission and reception of applications and application related information through wireless communications media. Using an OTA platform, new services can be introduced and the content of the SEs can be modified in a rapid and cost effective way. The OTA service can be provided by a Mobile Network Operator (MNO) or another trusted entity, depending on the agreed ecosystem.

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.
 Published 2012 by John Wiley & Sons, Ltd.

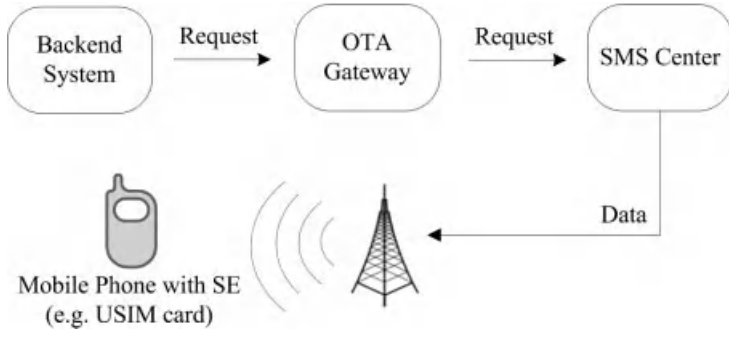


Figure 8.1 Basic OTA architecture.

8.1.1 OTA Technology and Mobile Device Management

Currently, Universal Subscriber Identity Module (USIM) cards are submitted to the user for mobile telephony services by MNOs who own them. MNOs control and manage the USIM, and some of them use OTA technology to manage the USIM as well. Today the MNOs that have set up the required OTA infrastructure have the ability to remotely configure a single mobile device, an entire fleet of mobile devices, or any IT defined set of mobile devices. They can send software and Operating System (OS) updates; remotely lock a device to protect the application and the data when it is for example lost or stolen; remotely troubleshoot the device; and perform additional similar services.

The basic architecture of OTA technology is depicted in Figure 8.1. Typically, OTA is based on a client – server based architecture consisting of a backend system as the server, and a smart card within the mobile phone as the client. The backend system sends requests which can be a customer care operator to a billing system, a content provider, and so on. These requests can be actions such as activate, deactivate, load and modify. The requests are sent to the OTA gateway which transforms the requests into short messages. Then these short messages are sent to a Short Messaging Service (SMS) center by the OTA provider which forwards them to the corresponding smart card. The OTA commands are generally sent as SMS messages. These commands are executed within the smart card.

Smart cards provide secure user access and are mainly used as Subscriber Identification Module (SIM) cards in GSM standard. The SIM is the major component of the GSM ecosystem, paving the way to value added services. SIM cards offer enhanced functionalities such as sophisticated menus, previously recorded numbers for speed dialing, and the ability to send SMSs to query a database, perform secure transactions, and so on. With the development of SIM cards, UICCs became valuable smart cards to enable enhanced services such as NFC technology and OTA management.

OTA technology has various benefits for stakeholders in the market. In terms of MNOs, as they are also SIM issuers, OTA technology provides increased cost effectiveness, remote management of SIM, and hence increases the value of the investment. Users do not need to physically go anywhere to activate or update any services. Smart cards in mobile devices are no longer static components. They can stock more information and they have more value for the user.

Making use of the advantages, a tremendous number of concurrent value added services can be offered to users by OTA. Furthermore OTA technology allows authorized third parties to keep control of the information downloaded on smart cards. OTA technology enables operators to choose the services which are at the customer's disposal. They own the content, thus they own the customer.

From the users' point of view, they are able to select the service(s) they want on their own personalized phone. Moreover they are free to do so whatever, wherever, and whenever. They can download the services and update their smart cards while they are mobile. Applications (i.e., recharging prepaid subscriptions, changing subscription parameter set) can be performed with OTA while they are mobile, instead of getting their phone to the customer service. This increased service quality provides convenience for the user and enables control over her subscription parameters.

8.1.2 OTA Technology and UICC Based SEs

As already mentioned, UICC has become the most popular way of promoting NFC based systems by using OTA technology. With OTA technology, new NFC applications can be delivered to a UICC and managed easily afterwards. OTA technology contributes to NFC by making the NFC application management on SEs more flexible and easier. New entrants can have their applications launched seamlessly, and existing participants can update their applications as the new versions become available. Since the management is performed via OTA, the consumer does not need to visit a shop physically or connect their handset to a docking station to update available applications.

In terms of NFC technology, the necessary OTA services are activation and deactivation of SEs, remote service management, remote life cycle management of NFC applications on the SEs.

With the emerging new NFC enabled services, accommodation and remote management of the applications on the mobile devices has become an important concern. Various associations are intensively working on remote service management (i.e., installation, personalization, update, termination) and remote life cycle management (i.e., card block, unblock, re-issuance, PIN reset, change, parameters update) of SEs via OTA.

In NFC systems where UICC has been selected as the SE, actors of the ecosystem have the advantage of being able to use an already existing infrastructure available for SIM management. MNOs already achieve maintenance of UICC procurement, distribution, configuration, management and update.

Consider the case that after a UICC has been issued to the user, a new NFC application is requested by the user. If OTA technology will be used in this situation, the SE issuer creates a secure space (secure domain) on the UICC for the new NFC application and assigns a unique security key to the security domain. This process can be performed easily and securely by the MNO using its OTA platform. The application and its data can be downloaded using OTA or alternately can be copied from a pre-loaded instance of the application on the UICC as well.

Furthermore, it is possible to lock or remove NFC enabled services on the UICC when it is required to do so for some reason. This is a vital function especially for credit card applications when a mobile has been lost or stolen. MNO can lock or delete all services on behalf of the NFC service providers in this way.

8.2 GlobalPlatform Specifications

As already mentioned in Chapter 3, GlobalPlatform is a cross industry membership organization currently comprising 50+ organizations, ranging from payments and communications industries to government sector and the vendor community. The companies involved are the pioneers for promoting a global infrastructure for smart card implementation across all possible industries. They feel responsible for specifically promoting interoperable technical specifications necessary to support smart cards and appropriate smart card system management for card issuers. GlobalPlatform provides a card specification which is currently accepted as a proper card model and offers secure and flexible multi-application card content management (i.e., loading, installation, extradition, registry update, and removal of card content) functionality during a card's life cycle.

8.2.1 GlobalPlatform Card Specification

This section aims to highlight the main properties of the GlobalPlatform card specification [1]. According to the GlobalPlatform card specification, the smart card is comprised of a number of logical and physical components that aim to provide application interoperability and security in an issuer controlled environment. Smart card specification is depicted in Figure 8.2. The Run Time Environment (RTE), the GP Environment (OPEN) and GP Trusted Framework, as

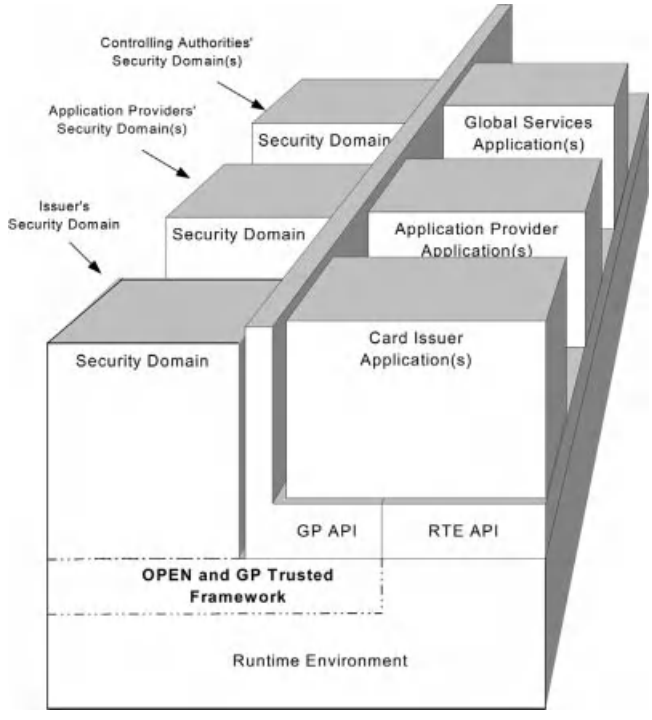


Figure 8.2 GlobalPlatform card architecture. Reproduced by permission of GlobalPlatform.

the communication components, are located over the smart card microprocessor. The rest of the smart card is divided into one or more services of the card issuer, application provider, and global services as well as into their security domains.

GlobalPlatform is intended to run on top of any secure multi-application card RTE; in this manner, GlobalPlatform does not mandate a specific RTE technology. RTE consists of three main components: Smart Card Operating System (SCOS), virtual machine (VM) and application programming interface (API). RTE is responsible for providing a hardware-neutral API for applications as well as a secure storage and execution space for applications. The VM acts as the interpreter between the SCOS and the applications. The RTE provides RTE API which involves a set of services utilized by the smart card applications such as inter-application communication and secure application isolation. GlobalPlatform cards may contain one or more trusted frameworks, which provide inter-application communication services between applications. Trusted frameworks have a special status; they are part of or have extensions to the card's RTE.

OPEN provides GP API for the applications. GP API is responsible for command dispatch, application selection, (optional) logical channel management and card content management services (e.g., locking, updating). GlobalPlatform provides specifications of API for MULTOS as well as JavaCard. The details of GP API for JavaCard and MULTOS card can be found in GlobalPlatform specification.

Security domains are extremely important for the GlobalPlatform based cards. Security domains act as the on-card representatives of off-card authorities. They enable secure management of applications by ensuring a complete separation of cryptographic keys as well as the owned applications and data between the card issuers and remaining security domains which are owned by the related application providers. There are three main categories of security domains reflecting different types of off-card authority recognized by a card:

- The Issuer Security Domain (ISD) is the primary and mandatory on-card representative of the card issuer. This security domain represents the issuer's territory on the card and controls the issuer's applications.
- Supplementary Security Domain (SSD) or Application Provider Security Domain (APSD) is the on-card representative of application providers and card issuers or their agents. This mechanism allows application providers to share and utilize some space on the card without the risk of compromising management of the card or any application on the card. This security domain delegates some content management capabilities to trusted application providers. This saves time for the card issuer with respect to additional administrative jobs.
- Controlling Authority Security Domain (CASD) is a special type of SSD. The role of the controlling authority is to enforce security policy to all applications that aim to be or are already loaded to the card. The controlling authority also uses this type of security domain as its on-card representative.

Security domains support various security services such as key handling, encryption, decryption, digital signature generation and verification for their providers' (i.e., card issuer or application provider or controlling authority) applications. Each security domain is established on behalf of a specific card issuer, an application provider or a controlling authority. When these off-card entities use the keys specific to a domain which are completely isolated from each other, applications interact with their related security domain via the GP API.

Card manager is the primary GlobalPlatform card component that acts as the central administrator for a GlobalPlatform card. It can be viewed as a combination of three entities: the OPEN, the ISD and Cardholder Verification Method Services which is one of the important global services applications.

GlobalPlatform's main goal is to ensure security and integrity of the RTE, OPEN, ISD, SSD and applications for the life cycle management of the smart card. Data integrity, resource availability, confidentiality and authentication mechanisms are completely supported on the GlobalPlatform based cards via the specification content.

GlobalPlatform is designed to provide maximum flexibility to the card issuer as well as its business partners regarding card content management. Due to this flexibility, the card issuer can delegate or authorize card content management functions to a service provider as will be discussed in the following section. Details about the card specification can be found on GlobalPlatform's website.

8.2.2 GlobalPlatform Messaging Specification

The GlobalPlatform messaging specification defines a set of standard messages for data exchange between entities in a smart card environment [2]. This specification is also applicable to all SEs (i.e., embedded hardware, SMC and UICC) on NFC mobiles. The specification is used to provide a standard infrastructure and to avoid proprietary solutions and therefore to exchange information seamlessly, promote standard interactions between players, as well as facilitate easy integration of new players in a smart card ecosystem. It brings interoperability at three major levels; business data (i.e., common vocabulary), business process (i.e., role, responsibilities, process and constraints) and data exchange (i.e., data structure, integrity and security of information, error handling, etc.). Details about the messaging specification [3] can be found on GlobalPlatform's website.

8.3 Life Cycle Management of SEs

The life cycle management of an SE within an NFC mobile starts after issuance of the SE to the user. The life cycle can be identified by two phases, namely, the *installation and personalization process* and the *remote management process*. The remote management process starts after installation and personalization of the NFC application. Initially, the card holder – as the user – downloads an NFC application, the related security domain together with its keys are created, and the application is installed on the SE. The remote personalization process can be performed and the new application can be managed (i.e., updated, removed, renewed) remotely thereafter.

Since installation, personalization, and remote management processes can be performed on all SE options (i.e., embedded hardware, secure memory card, UICC), the establishment of new security domains on each type of SE requires a different technological infrastructure. Currently the embedded hardware and secure memory card (SMC) based SEs can only be accessed via OTA through midlet proxy. This connection requires that the communication is initiated on the mobile handset side and secure http connection is established using GPRS or 3G communication.

UICC based SEs provide a great opportunity for extensive customer support via OTA as already mentioned. In this chapter the life cycle management of UICC based SEs is mainly

covered. GlobalPlatform specifies three models for the installation and personalization process of NFC applications on UICC; simple mode, delegated mode, and authorized mode. These models can be applied for the other SE options with different infrastructures and requirements as well. Other details on card content management and UICC configuration can be found in the GlobalPlatform card specification. This section briefly covers the required background knowledge such as importance of Trusted Service Managers (TSMs), roles and actors defined by GlobalPlatform, structure of UICC depending on GlobalPlatform, as well as GlobalPlatform management models for the application installation and personalization process.

8.3.1 TSM in NFC Environment

The NFC ecosystem is inherently a complex environment. It consists of various players ranging from NFC component manufacturers to consumers. A detailed business analysis of the NFC ecosystem has already been discussed in Chapter 7. As detailed there, the main actors of the ecosystem are hardware vendors, service providers (e.g., banking financial entities, transit authorities, retailers who wish to deploy and manage services on users' SEs), card issuers (or SE issuers), MNOs, and OTA platform providers. In most of the NFC service prototypes and trials, MNOs are issuing the SEs and some of them are also owning and managing the OTA platform. However, it is also possible that SE issuance as well as providing the OTA can be performed by trusted entities other than MNOs.

In order to deploy NFC applications and services to the users' SEs, service providers and MNOs have distinct requirements. To create and manage a trusted environment and to allow actors to communicate among themselves securely, an additional trusted actor is required. TSM is an independent party serving other actors of the NFC ecosystem as required. It ensures a level of trust and security between major actors of the system such as service providers and MNOs during life cycle management of applications.

If TSM does not exist in the ecosystem, each player needs to communicate directly with each other player to perform efficient card content management, which creates a complex and insecure system. TSM plays a central authority role in the system instead. Alternatively, card content management functions can be established by either the card issuer centric approach which uses the card issuer OTA platform or the independent-TSM centric approach which uses the TSM OTA platform. The architectural and technical features of the independent OTA deployment and card content management according to this scenario will be discussed in the next section.

Now let us consider NFC enabled payment applications running on a UICC. This is a very complex and dynamic world by nature. Currently there are various collaboration models throughout the world in the mobile payment ecosystem. In this model, collaboration among financial institutions, MNOs and other stakeholders in the mobile payment ecosystem, and also a TSM who needs to play a central role between the MNO and the service providers is required. For instance, TSM can deliver the user's payment account information after it is received from the financial institution via OTA to the UICC; this means the user can use her mobile phone as a virtual credit card or debit card (see Figure 8.3). The TSM's or MNO's OTA platform can be used depending on the deployment model as presented in the next section.

As already mentioned, the primary function of the TSM is to ensure security among entities of the ecosystem. GlobalPlatform defined a secure channel protocol (SCP) which provides secure functionality of communication and storage of sensitive data between the TSM and SE of

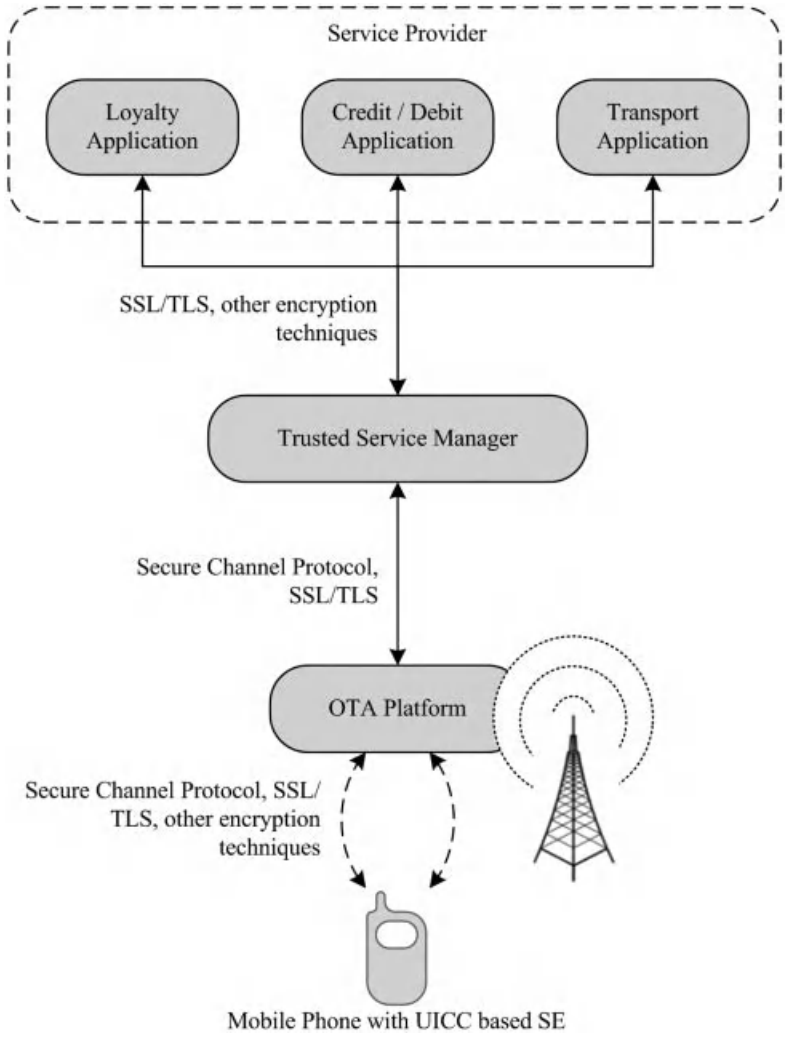


Figure 8.3 NFC environment and security mechanisms.

the mobile phone. Various types of SCP (e.g., SCP01, SCP02) are defined in the GlobalPlatform card specification [1]. SCP is used by the TSM when it tries to reach security domains on UICC via OTA as presented in the deployment models in the next section. TSM also uses other security mechanisms such as Public Key Infrastructure (PKI), Virtual Private Network (VPN), and Secure Sockets Layer (SSL)/Transport Layer Security (TLS) encryption techniques.

8.3.2 Actors and Their Functional Roles in GlobalPlatform

GlobalPlatform defines five major actors in NFC ecosystem: service provider, SIM vendor, MNO, TSM, and controlling authority. GlobalPlatform also defines various roles which are

adopted by these actors. Each actor may perform one or more roles. Roles defined by the GlobalPlatform [2] are as follows:

- *Application owner* is the one who maintains application related specifications.
- *Application developer* is the one who designs and implements application code.
- *Application provider* is the one who provides the component to load an application (i.e., application code, application data, application keys and/or certificates and data belonging to a specific cardholder) onto a card.
- *Card enabler* performs pre-personalization functions; the loading of the initial issuer security domain, a controlling authority security domain, and if any, additional application provider security domains.
- *Card manufacturer* is the entity that manufactures cards that follows the requirements of the card issuer.
- *Integrated Circuit (IC) manufacturer* is the entity that fabricates wafers containing chips with a specified ROM configuration.
- *Platform developer* is the one who is responsible for the development of GlobalPlatform cards in accordance with the specifications provided by the GlobalPlatform consortium.
- *Platform owner* defines and maintains the card platform operating system specifications.
- *Card issuer* is the one who holds ultimate responsibility of the SE. It is the only authority to allow loading, installation, deletion, extradition and personalization of the applications. Also, the card issuer may delegate these activities to a third party such as an application provider, via the SSD manager.
- *Loader* loads applications and personalization data to the card issuer specific cards according to the instructions of the application provider, complying with security policies and procedures set by the card issuer.
- *SSD manager* manages instantiated security domains on a card. It holds the SCP (i.e., SCP02 and SCP80) keys and certificates belonging to the secure domain and allows secure communication to the secure domain that it is in charge of. SCP02 is a symmetric secure channel protocol allowing script messaging that can be received directly through contact and contactless interfaces. SCP80 is an OTA secure channel protocol enabling direct access to the security domain from an OTA platform through the GSM network. SSD manager has the capability to load, install, extradite and personalize applications on behalf of the service provider. Its role is not managing the applications on the smart card but to provide and to intermediate secure domain to manage card content and to perform cryptographic services.
- *Collator and decollator* is the one who performs the collation (data aggregation) of personalization data from all service providers for the card holder after the loader has performed application loading. It performs the decollation function to return information to the card issuer and appropriate application providers.
- *Controlling authority* is the most central, important, and required actor for confidential application loading and personalization. The entity can either be a certificate authority or a SIM Vendor. The controlling authority keys and certificates are loaded into the Controlling Authority Secure Domain (CASD) on the UICC card by the card manufacturer during the UICC manufacturing process. The controlling authority provides load file data block signatures according to its own security policy for integrity and source authenticity. The controlling authority entity needs to be a trusted party to the entity who operates the OTA platform, as well as to the entity performing application personalization on the UICC.

8.3.3 UICC Based SE: Security Domains and Hierarchy

As already mentioned, currently a UICC based SE uses GlobalPlatform based card structure as reference architecture to support NFC based systems. The UICC can host a number of different applications owned by the card issuer or other parties, each one defining and controlling its own application. The UICC has a separate security domain for each application or for each application provider administered by the application issuer and based on the use of secret administrative keys. SCOS on the card implements a firewall that prevents applications from accessing or sharing data among themselves.

Any GlobalPlatform compliant UICC based SE comes with one ISD and the option for single or multiple SSDs. These SSDs can be TSM security domains, or service provider security domains such as credit card, transport, ticket, and loyalty applications. Application and data storage areas of TSMs and service providers are separated and isolated from each other. Additionally, each UICC based SE has only one CASD. This architecture enables card issuers, service providers, and TSMs to perform key management and application verification during application loading and personalization processes.

On the ISD area of the UICC, the card issuer (mostly a MNO) stores the keys for OTA provisioning, card content management, and security domain management. The ISD is created during the manufacturing process, and the key for card content management is securely transferred from the manufacturer to the MNO thereafter. According to GlobalPlatform specifications, the ISD must authorize the creation of any SSDs, so that only the ISD has the privileges to create an SSD. Also, only an ISD can assign authorized or delegated management privileges.

8.3.4 UICC Management Models

According to GlobalPlatform specification [2], the card content management process includes the following activities: confidential loading of initial key set of third party's security domain, application loading and personalization. GlobalPlatform proposes three card content management models as simple mode, delegated mode, and authorized mode for the SE as seen in Figure 8.4. These models cover application loading and personalization processes on the UICC cards. The simple mode is the card issuer centric model, whereas the delegated mode and authorized mode are more TSM centric models (see Chapter 7). GlobalPlatform messaging specification supports all deployment models [2, 3]. The essential technical and architectural details of these modes are given below.

(i) *Simple mode using MNO OTA platform*

In simple mode, the service provider delegates full management of its NFC enabled application to a TSM. The TSM manages the security domain on behalf of the service provider. The MNO is authorized to perform the card content management functions, that is, loading, installing, activating and removing the application on the SE. The TSM only manages the application lock, unlock and personalization processes using its own OTA server and network of MNOs.

As seen in Figure 8.5, the MNO has the loader role. However, the TSM can monitor the application loading by providing the Data Authentication Pattern (DAP) signature to the MNO. DAP verification is used to authenticate and check integrity of an application loaded on a UICC.

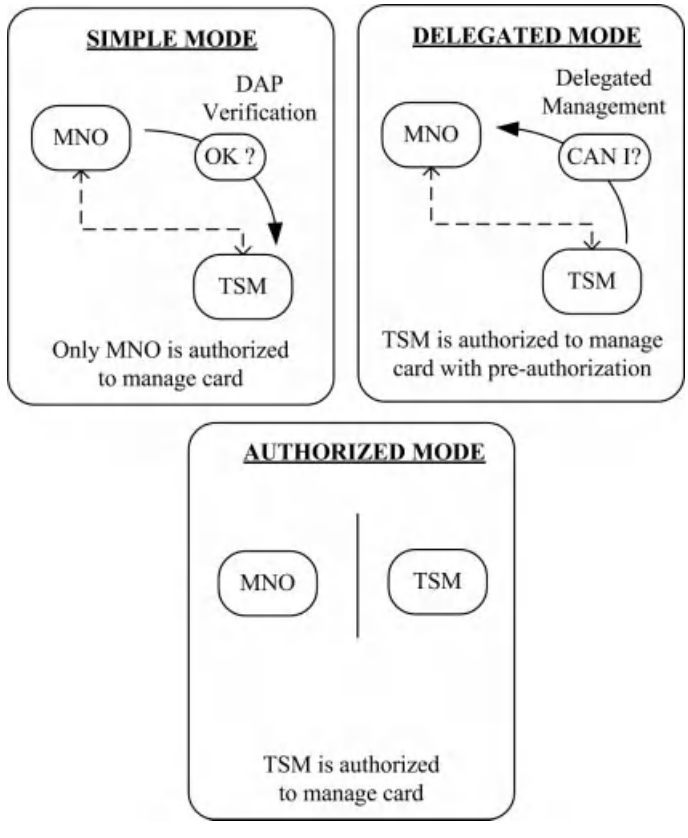


Figure 8.4 Application management models. Reproduced by permission of GlobalPlatform.

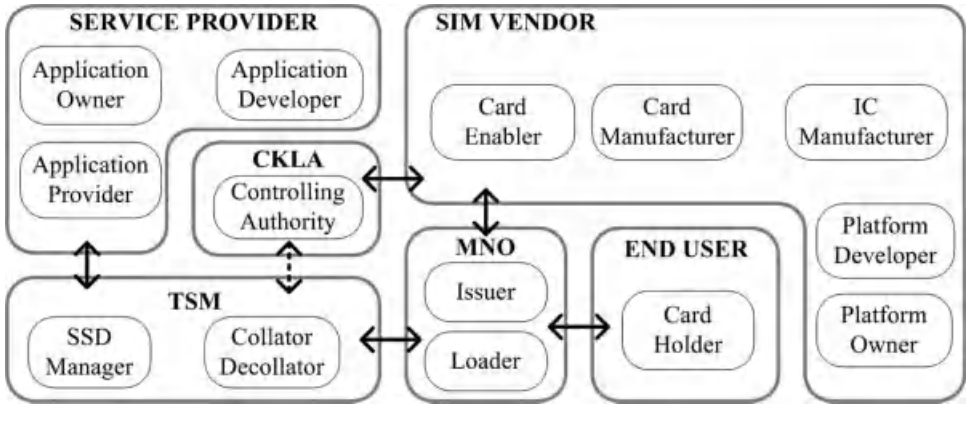


Figure 8.5 Roles of actors in simple mode using MNO OTA platform. Reproduced by permission of GlobalPlatform.

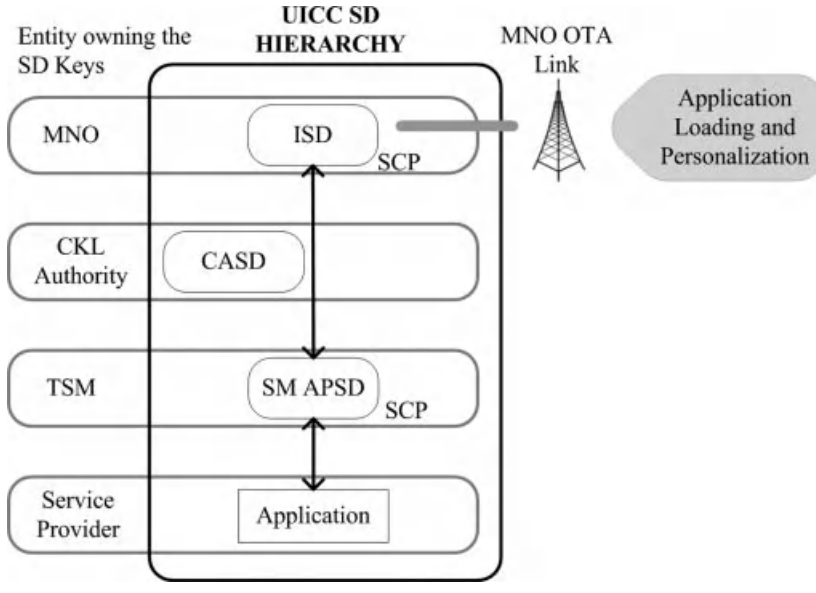


Figure 8.6 Simple mode using MNO OTA platform on UICC. Reproduced by permission of GlobalPlatform.

As illustrated in Figure 8.6, each actor has its own security domain on the UICC to support security services such as key handling, encryption, decryption, digital signature generation and verification for applications owned by that actor. The content loading process allows an off-card entity (with on-card representation through a secure domain) to add content to the card.

The MNO has an ISD on the secure UICC; the Confidential Key Loading Authority (CKLA) has the controlling authority secure domain (CASD). TSM enables the link between the service provider and the MNO via SCP on the card, and has simple mode application provider security domain (SM APSD) which establishes the application of the service provider.

TSM uses the MNO OTA platform for the creation of SM APSD. SM APSD keys are created and retrieved by the TSM for the management of applications in a confidential way. The TSM OTA platform is not completely involved in simple mode.

(ii) *Delegated mode with full delegation to the TSM*

The delegated management case can be described as TSM centric loading. In this case, the MNO is no longer in charge of loading, installing, activating or removing the application (see Figure 8.7). Card content management is performed by the TSM with a pre-authorization from the MNO. In some cases, the service provider may need to manage its own application personalization process to prevent any third party manipulation of application keys or application data which are valuable for their customers.

Delegated management privilege allows delegated loading, installation, extradition, update and deletion. The service provider delegates full management of its application to a TSM, and the TSM is responsible for the creation of its APSD and the management of its

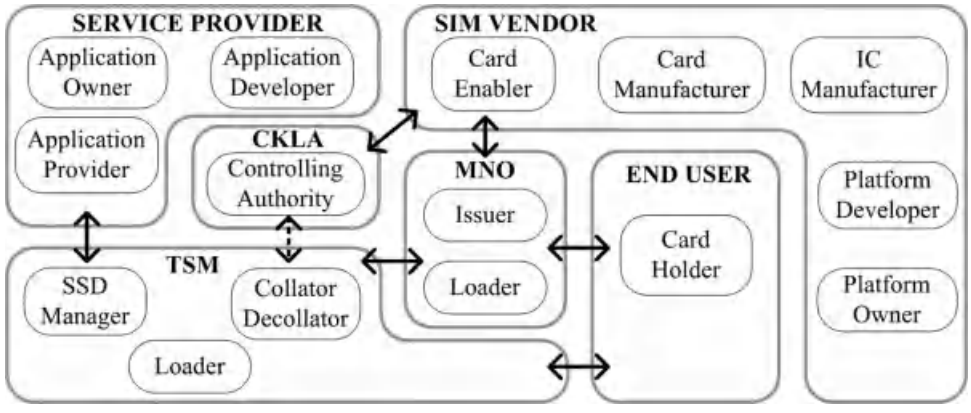


Figure 8.7 Roles of actors in delegated mode with full delegation to TSM. Reproduced by permission of GlobalPlatform.

application loading and personalization process. The TSM will use its own OTA platform. The MNO needs to deliver a management token to the TSM for a pre-authorized card content management action. This management token can be also identified as a *load token* which is a digital signature performed only by the card issuer. In the case of the application lock/unlock and personalization, there is no need for the delegated management token; the TSM can perform without a token. Similarly the APSD keys are created and retrieved by the TSM in a confidential way.

Delegated management empowers the card issuers with a preauthorization mechanism. In this way, card issuers are protected from unauthorized changes taking place and they dismiss responsibility for managing applications outside their direct interests. At the same time, service providers are provided with the flexibility for managing their own applications.

As seen in Figure 8.8, the MNO has its own security domain as the ISD on the UICC card and CKLA has the CASD. The TSM enables a link to be established between the service provider and the MNO via the SCP on the UICC. In delegated mode, the TSM creates its own TSM security domain (TSD) on the UICC with the preauthorization provided by the card issuer. The service provider's application can be loaded in two ways: it can be loaded directly on the delegated mode TSD (DM TSD) or on the dedicated simple mode APSD (SM APSD) which is assigned per application.

(iii) *Authorized mode with full delegation to TSM*

The authorized management deployment is completely organized around the TSM centric loading option. The TSM has service provider applications and is able to perform card content management without authorization (or being forced to use a token) from the MNO. As in the delegated mode, the service provider can manage its own application personalization instead of delegating it to the TSM. The MNO has no linkage with the end user as well as it does not perform the loader role any more as seen in Figure 8.9.

In this mode, the service provider delegates the card content management of its application to the TSM and the TSM uses its own OTA platform. Hence, the TSM has full flexibility to create a security domain as the authorized mode TSD (AM TSD) and load

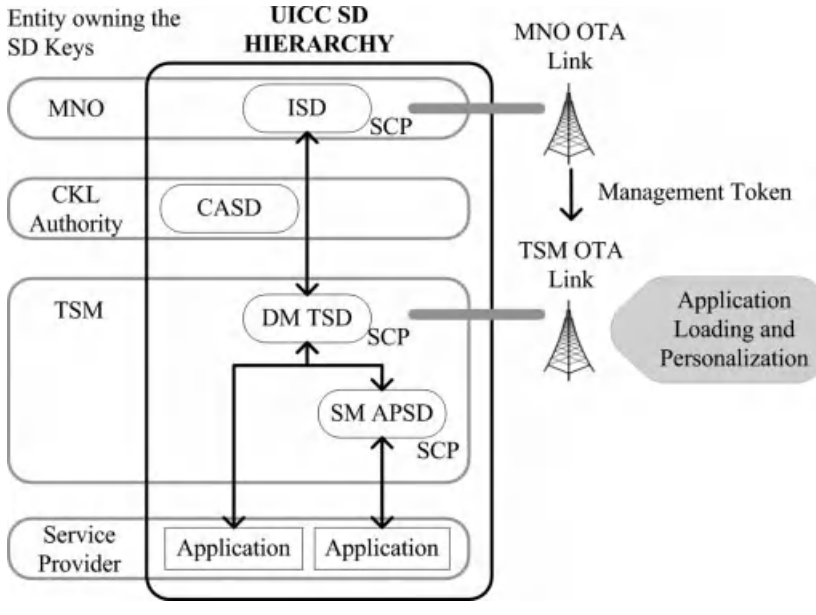


Figure 8.8 Delegated mode with full delegation to TSM on UICC. Reproduced by permission of GlobalPlatform.

the application without MNO authorization. The APSD keys are created and retrieved by the TSM in a confidential way.

As seen in Figure 8.10, the TSM creates its own security domain on the UICC. The service provider's application can be loaded directly under AM TSD or alternatively a dedicated and separate simple mode APSD (SM APSD) can be assigned for each application. During the application loading and personalization processes, there is no connection or communication link with the MNO and its OTA platform.

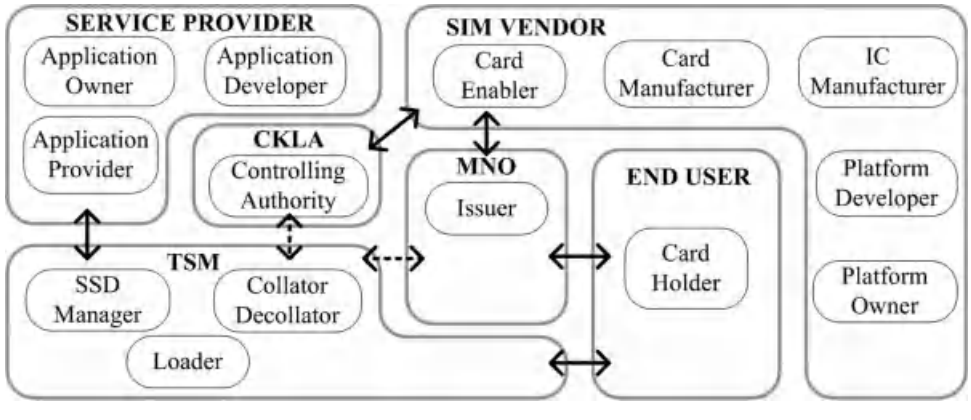


Figure 8.9 Roles of actors in authorized mode with full delegation to TSM. Reproduced by permission of GlobalPlatform.

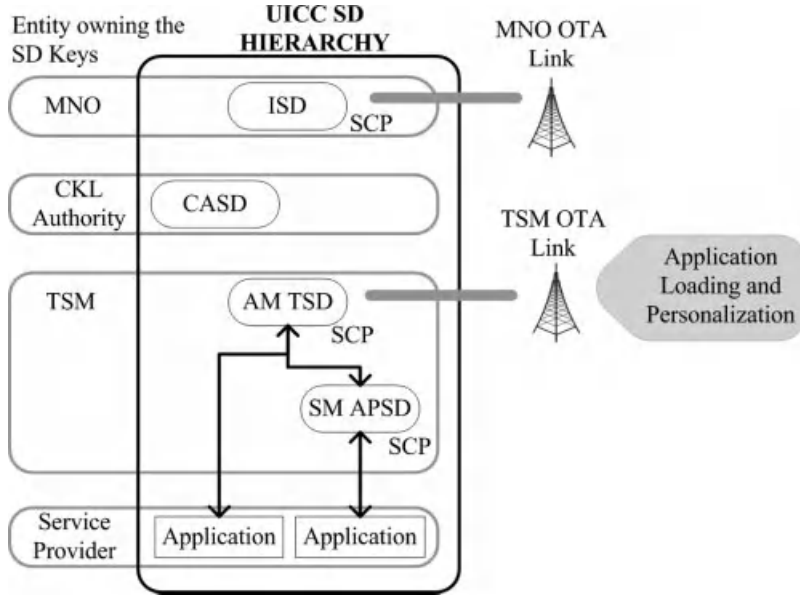


Figure 8.10 Authorized mode with full delegation to TSM on UICC. Reproduced by permission of GlobalPlatform.

8.4 Multiple SE Environments

It is also possible to see that an NFC enabled mobile phone can host multiple SEs. For example, a mobile phone may contain embedded hardware or SMC together with a UICC based SE that is embedded by the MNO before handing the mobile phone to the customer. So, each SE issuer may offer a portfolio of applications and services provided by different service providers on the corresponding SE.

Many problems may emerge when multiple SEs reside on the same NFC enabled mobile handset; especially the most important one can occur when implementing card emulation mode applications. The NFC reader needs to initiate communication with only one SE on the NFC enabled mobile phone. Another problem is the management of multiple SEs at the same time. GlobalPlatform has performed valuable work on the potential implications of managing multiple SEs in the same mobile handset. A GlobalPlatform document [4] gives the details of managing multiple SEs in a single mobile handset to deliver NFC applications and services. To summarize the document [4], two business models can be performed: architecture without aggregation and architecture with aggregation. Each model has its own requirements; however these business models need to be considered more in terms of compatibility and interoperability.

8.4.1 Architecture without Aggregation

According to this model, if multiple SEs are hosted on the same NFC enabled mobile phone, the service availability between applications located on different SEs can differ depending on

the capability of the SEs such as infrastructure and security level of the SE, power off capacity, and so on. The service availability can also differ depending on the service level of customer support provided by the SE issuers and service providers.

In this model, the NFC controller can be used by only one SE at a given time. For this reason, only one SE can be active at a time. When the "without aggregation" model is in use, the SE is activated by the user so that the activated SE is able to perform NFC enabled contactless transactions. The selection of the SE depends on the service to be used of course, and the SE that contains the related application has to be selected; the user is responsible for selecting the correct SE in this mode. If the user wants to use an NFC application hosted in a currently idle SE, she needs to switch the active SE. Another important requirement is the firewall mechanism between SEs. The firewall ensures that information from an SE cannot be retrieved by another SE without authorization of the SE issuer.

The selection of an active SE can be done through the mobile phone's menu by the consent of the user. The menu displays information related only to the SE and no information about applications in the SE is displayed. After selection of the SE, the list of applications can be displayed. This interface needs to be so dynamic that when a new application is loaded on the SE or an application is removed, the list of all current applications on the SE should be retrieved from the SE and displayed to user.

Furthermore, the status of NFC applications (i.e., activate or deactivate) on an active SE can also be controlled by the user. For instance, one or multiple NFC enabled credit card applications on the SE can be activated by the user. Additionally, the user can activate and prioritize these applications located on an SE prior to presenting the mobile phone to the reader.

8.4.2 Architecture with Aggregation

In the "with aggregation" model, all SEs hosted in the NFC enabled mobile phone are active at the same time. Any application on any SE can perform contactless NFC transactions at any given time. So, a list of all NFC applications on all SEs in a mobile handset should be displayed to the user at once. Obviously each SE can contain one or more applications.

It is still possible to control the status of NFC applications on a single SE or across multiple SEs. The status of an NFC application is controlled by the user and the application should be selectable by an external reader. The external reader sends a selection request to an active application and this selection request is directly forwarded to the appropriate SE where the active application is located. The reader receives a single response from the corresponding application.

8.5 Alternative TSM Based OTA Management Model

Currently in the NFC industry, there is a wide range of competition between business partners, thus the business models they are using differ depending on the business opportunities. MNOs play an important role as the UICC issuer and even they may act like a card manager. The MNO holds the essential keys on the card. Thus service providers and other entities such as the TSM are mainly dependent on the card issuers. This section provides an alternative model where none of the actors are dependent on one actor, and which benefits from the trustworthiness role of the TSM. This model actually is an extension of GlobalPlatform's authorized model where

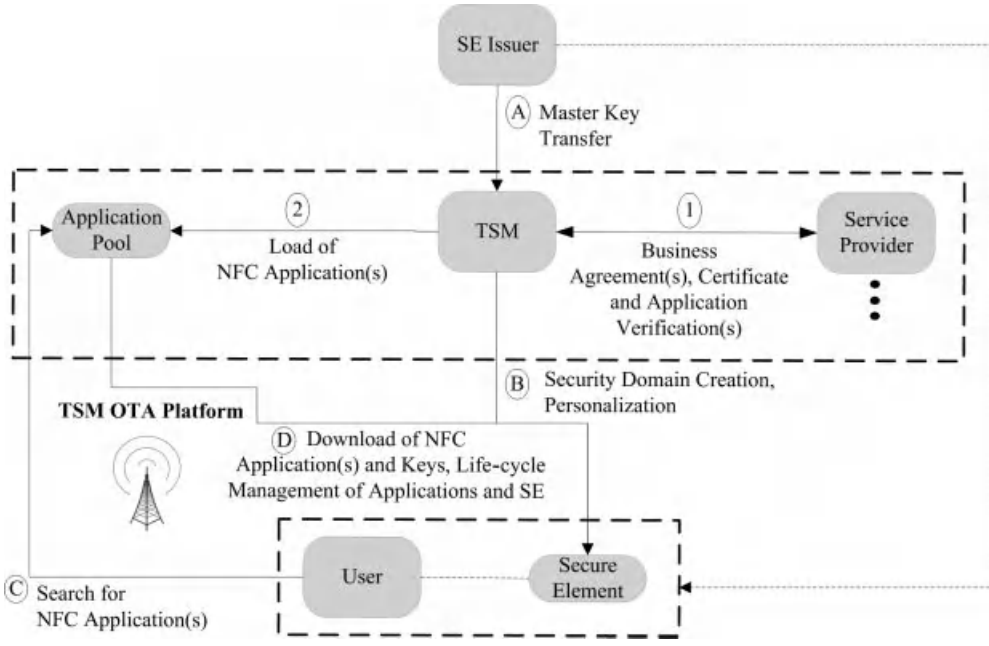


Figure 8.11 Alternative TSM based OTA management of NFC applications.

full delegation is given to the TSM, but also can be applied to all SE options as embedded hardware, SMC, and UICC.

A TSM has the ability to create and manage a trusted environment by using its own OTA platform for both users and service providers. Providing an interoperable and flexible OTA platform is one of the key requirements in NFC systems. In order to provide more secure and efficient life cycle management of NFC applications and SEs, a new infrastructure is proposed; the application pool infrastructure which is controlled by the TSM. The proposed application pool provides a central location or database for application housing; it can be an online application store as well. Applications can be loaded by service providers to this application pool and they can be offered to users any time.

As illustrated in Figure 8.11, when service providers decide to distribute an NFC enabled application, they need to initially make business agreements with the TSM in the market. The TSM then checks and, if valid, approves the application together with its certificates (Step 1), after which applications can be loaded and published in the TSM's application pool (Step 2).

While TSM is handling service providers and market requirements, it can also control and manage the life cycle of the SE securely. As seen in Figure 8.11, the TSM requests the SE's master key from the SE issuer (Step A). As already mentioned, the master key is a unique key and exists in all SEs. It enables the owner to create security domains on the SE, control each security domain and personalize the SE (Step B). In the proposed model, the TSM acts as the owner of the master key as the trusted party between actors. The TSM requests this key to handle SE functions from the SE issuer who can be a MNO or another entity depending on the SE option (as discussed in Chapter 7).

A user who wants to use one of those NFC applications can search the application from the TSM's application database (Step C). The TSM can easily manage these applications; it can update and personalize them via OTA on the user's UICC (Step D). Also, such a mechanism enables prevention of spam-like applications, and private and sensitive information are secured on the user's UICC.

8.6 Chapter Summary

UICC cards have become the most popular way of promoting NFC based systems that support OTA technology. OTA technology is always available and manageable to offer value added services to the customers. With OTA technology, NFC enabled applications can be delivered to a UICC, and they can be updated and managed easily. On the user side, OTA provides many advantages to the users as well as to their UICCs. On the business and technical side, there is a complex structure. GlobalPlatform tried to provide standards and specifications for appropriate, interoperable smart card system management and has offered secure and flexible multi-application card content management functionality during a card's life cycle.

GlobalPlatform has offered card architecture and OTA deployment models on this card architecture, which is also applicable to the UICC based SE. According to the smart card architecture, the card is comprised of a number of logical and physical components: microprocessors, RTE, OPEN, GP Trusted Framework, APIs, areas consisting of one or more services from the card issuer, application provider and global services and security domains (i.e., ISD, APSD, and CASD). Security domains are the important part of the card which support security services such as key handling, encryption, decryption, digital signature generation and verification for their providers' applications. Each security domain is established on behalf of a card issuer or an application provider or a controlling authority.

Any GlobalPlatform compliant UICC comes with one ISD and the option for multiple TSM security domains or application provider (or service provider) security domains (such as credit card, transport, ticket, loyalty applications). Each service provider's and TSM's data storage areas are isolated from each other as well as their applications. Also, each UICC based SE has only one CASD. This architecture enables card issuers, service providers, and TSMs to perform key management and application verification during application loading and personalization processes.

According to GlobalPlatform, confidential card content management is initial key set of third party's security domain loading, application loading and personalization. GlobalPlatform proposes three card content management models on UICC cards: the simple mode which is a completely card issuer centric model; the delegated mode and authorized mode are more TSM centric models.

In simple mode, the service provider delegates full management of its NFC enabled application to a TSM. TheTSM manages the application lock, unlock and personalization processes using its own OTA server but the network of the MNO. The MNO is authorized to perform card content management functions. In the delegated mode, the MNO needs to deliver a digital signature as a token to the TSM for a pre-authorized card content management action. However, the TSM can still perform the application lock/unlock and personalization process without a token. In authorized mode, deployment is completely organized around the TSM. The TSM has the service provider applications, and is able to perform card content management with its own OTA platform, without authorization or a token from the MNO.

Another important issue is the multiple SE environments on a single NFC enabled mobile device. Each SE hosts one or more NFC applications which are provided by different service providers. Two issues need to be considered while multiple SEs reside in the same handset; the external reader's operation with an application on a SE and management of multiple SEs at the same time. GlobalPlatform's Mobile Task Force has performed an analysis on the potential implications of managing multiple SEs in the same handset; two business models can be followed which are architecture without aggregation and architecture with aggregation. In the case of the architecture without aggregation model, only one SE is activated by the user and is visible to the reader. Hence, the activated SE is able to perform contactless transactions. In the case of architecture with aggregation model, all SEs hosted in an NFC enabled mobile phone are active at the same time. Each model with distinct requirements needs to be considered more in terms of compatibility and interoperability.

With the development of NFC and UICC technology as well as with the new entrants to the NFC ecosystem and collaboration models, new deployment models may emerge. An improved authorized model may come and TSMs may dominate on NFC based systems with their own flexible and interoperable OTA platforms, and will eliminate the dependencies on card issuers and network operators. On the other hand, this may cause TSMs to be the only authority that publish or vendor UICC cards in the future.

Chapter Questions

1. What is the role of OTA technology in mobile device management?
2. What is the importance of OTA technology in the NFC ecosystem?
3. What is the role and importance of the TSM in the NFC ecosystem?
4. What is a security domain? Explain the architecture of an UICC based secure element.
5. Explain the relationship between card content management and OTA deployment.
6. Explain the importance of GlobalPlatform's specifications and OTA deployment models.
7. Give details of the simple mode using the MNO OTA platform.
8. Give details of the delegated mode with full delegation to the TSM.
9. Give details of the authorized mode with full delegation to the TSM.
10. Is it possible to host multiple secure elements on one NFC mobile phone? How?

References

[1] GlobalPlatform (2006) *GlobalPlatform Card Specification Version 2.2*, March 2006, http://www.globalplatform.org/specificationscard.asp (accessed 10 July 2011).

[2] GlobalPlatform (2009) *GlobalPlatform's Proposition for NFC Mobile: Secure Element Management and Messaging*, White Paper, April 2009. Available at: http://www.globalplatform.org/documents/GlobalPlatform_NFC_Mobile_White_Paper.pdf (accessed 10 July 2011).

[3] GlobalPlatform (2011) *GlobalPlatform System, Messaging Specification for Management of Mobile NFC-Services, Version 1.0*, February 2011, http://www.globalplatform.org/specificationscard.asp (accessed 10 July 2011).

[4] GlobalPlatform Mobile Task Force (2010) *Requirements for NFC Mobile: Management of Multiple Secure Elements Version 1.0*, GlobalPlatform, Februrary 2010. Available at: http://www.globalplatform.org/documents/whitepapers/GlobalPlatform_Requirements_Secure_Elements.pdf (accessed 10 July 2011).

9

NFC Cities and Trials

There have been many attempts that realize and make use of NFC technology. Some models have been developed by universities, others by companies, and even more as a joint effort between universities and companies. Many models are only theoretical, some have been implemented but cannot be used because of missing elements, and others have been fully developed (e.g., so that bank accounts can be processed). An NFC city typically describes a city where one or more NFC applications are being used. The purpose of an NFC city is either to test or use one or more implementations. NFC cities are important for NFC technology's improvement, since they are the actual arena of a moderate sized usage media. In NFC trials and projects, an application or an NFC ecosystem is tested; thus usability issues together with problems residing in the technology can be obtained through tests in the initial phase of NFC cities. During the testing period, one of the purposes is to evaluate the applicability and use of the NFC technology. One major aim is to test NFC ecosystem issues. NFC ecosystem usability is at least as important as the technical appropriateness, since the model cannot be used when there is disagreement between the parties involved.

9.1 NFC Cities

We have analyzed three leading NFC cities, namely Oulu, Nice, and Smart Urban Spaces.

9.1.1 City of Oulu

Oulu was used as a test bed for the SmartTouch project. The SmartTouch project had 22 partners from 8 European countries. The partners included 9 large industrial organizations, 5 companies, 4 research and 2 public organizations. The project was coordinated by the VTT Technical Research Center of Finland. With this project, Oulu became a pioneer city that then became a member of NFC Forum.

The project ran from 2006 to the end of 2008 and examined the role of NFC in city life, the home, wellness and health, entertainment, technological and business building blocks. Citizens living in Oulu had the opportunity to test commercial and public NFC services as the

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.

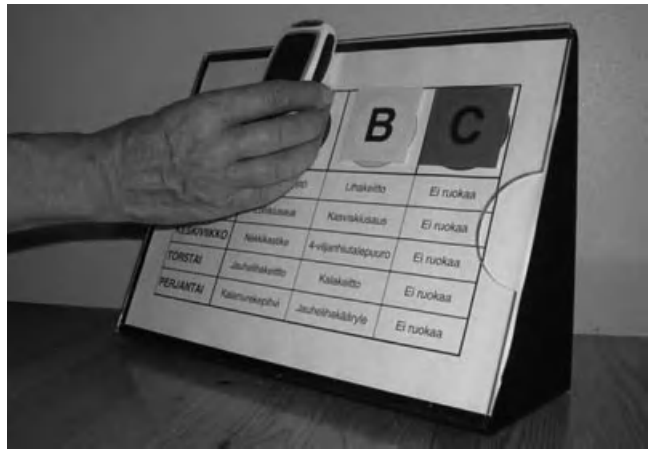


Figure 9.1 Meal ordering for elderly [1].

first users of the technology covering this broad array. The applications, trials and pilots that were tested in this project are as follows [1, 2]:

(i) *Meal ordering for elderly*

This service allowed elderly clients to order meals by simply touching tags of the meals, instead of making phone calls. The motivation for the project was to reduce people's efforts.

During the pilot phase, users touched tags of the preferred items on the daily menu using their NFC enabled mobile phones. The scenario is depicted in Figure 9.1. The meals were then delivered to the home by the food company staff. After the user touched to a tag, the Oulu Meal Service received the order and prepared the meal. A logistics service then delivered the food. Logistics staff were also using NFC enabled mobile phones and they reported the status of the meal at the beginning of the delivery round, upon successful delivery and finally on completion of the whole process. Backend staff controlled the overall process by gathering real time information on the status of the meal, since the information was sent online to the backend servers.

Nine users participated in the pilot phase; the average age of the participants was 76.6. The testing implementation showed that interaction with NFC technology was fairly easy for elderly people, but some of them experienced problems in using their mobile phone. Specifically, the use of a mobile phone's keypad was difficult for some of the participants due to decline in their vision and motor skills. Some of the participants complained about the small size of the keys on the keypads: this is actually a common problem of mobile phones and not directly related to the NFC technology.

Among several menu options, the right touch point was required to order the correct item. Experience has shown that this was fairly easy for the users. As a result, the touch-based interaction technique seems to be easily adaptable for all users.

The most important finding of the trial was that five of the nine participants declared that they would prefer NFC based meal ordering rather than making a phone call to order meals from a distance. The remaining participants still preferred the old

Figure 9.2 A student logs into class [1]. Photographed by Juha Sarkkinen, The City of Oulu, Smart Touch Project.

fashioned method [3]. For a demonstration video, the readers are encouraged to visit http://www.ouka.fi/video/smart/elderly/.

(ii) *NFC attendance supervision for secondary school*

The project's aim was to monitor the attendance of students in secondary schools for their parents. It eliminated the need for teachers to mark absences, which saves time allowing more time for teaching.

As a student arrives at the class, she touches the tag which is attached to the teacher's desk; the database on the backend server is updated immediately about the presence of the user student. Consequently, a student is marked absent if no login action occurs before a certain deadline. In the case of a late arrival, the system records the situation accordingly. In the case of a reasonable excuse for the absence such as sickness, teachers update the data for this situation (see Figures 9.2 and 9.3).

Another important property of the system was to inform parents and administrators on students' attendance which is performed in real-time fashion. The pilot was performed with 113 pupils from 5 classes of 8th graders in 2008 and lasted for 8 weeks [4]. For a demonstration video, the readers are encouraged to visit http://www.ouka.fi/video/smart/school/.

(iii) *Attendance supervision for primary school using NFC*

The usage cases of this project are fairly similar to the previous one. In this project, the attendance of primary school students is recorded in order to give parents real-time information on their children. In a class, each student has a smart card in which the pupil's name and identification are saved. The teacher has an NFC enabled mobile phone with an installed attendance application. When a pupil touches her card to the teacher's mobile phone, the entrance or exiting action, time stamp, and pupil identification is saved to the backend system. Parents can monitor their children's attendance online or receive notifications via SMS according to their preferences.

The pilot of this project was started in 2008 and lasted for 14 weeks. A total of 27 students were used in the pilot aged between 6 and 7 years old. One class consisted of

Figure 9.3 Accepted presence information [1]. Photographed by Juha Sarkkinen, The City of Oulu, Smart Touch Project.

20 pupils and an active reader was placed in the classroom. The second class consisted of seven pupils and a teacher's NFC enabled mobile phone was used as the NFC reader (see Figure 9.4). Usability of NFC technology and security aspects of attendance control among children were also tested in the pilot [4]. For a demonstration video, the readers are encouraged to visit http://www.ouka.fi/video/smart/school2/.

(iv) *Amazing NFC*

Amazing NFC is a city orienteering project for schools conducted in Oulu. Its aim is to teach pupils about the essentials for the school and everyday life such as information on student loans or information about the city's history. Two different types of the orienteering is performed: survival track and cultural/historical track. The survival track

Figure 9.4 A teacher logs-in students using an NFC enabled mobile phone [1]. Photographed by Juha Sarkkinen, The City of Oulu, Smart Touch Project.

Figure 9.5 A student performing orienteering [1]. Photographed by Juha Sarkkinen, The City of Oulu, Smart Touch Project.

enables pupils to learn about Oulu's offices and institutions. The cultural track helps students get information about Oulu's culture and history.

When a student starts the orienteering, a departure point is defined. When the student reaches this departure point, she touches a tag that she finds and the student is asked a question via a mobile website. The question can be asked using a text, a video, or an audio. When the student answers correctly, she receives the next control point (see Figure 9.5).

The pilot phase of this project was launched in 2008 with survival track, and then cultural track. About 400 students were used in the pilot phase. However, the cultural track was limited by poor availability of the NFC terminal devices from the supplier [4]. For a demonstration video, please visit http://www.ouka.fi/video/smart/amazing/.

(v) *Smart parking with NFC*

This project enabled parking control with NFC technology by eliminating coins and parking tickets. Moreover, only the actual parking time instead of pre-defined periods was paid for and the system which decreased parking fees.

In order to initiate the parking registration and payment, the user first touches her mobile phone to the NFC tag on the windshield then a tag at the parking zone. Alternatively, she may choose the parking zone from the application in the mobile phone. After parking ends, the user only touches the tag placed on the windshield or at the parking zone as seen in Figure 9.6. Traffic staff were also equipped with NFC enabled mobile phones and when the staff touched a tag on a parked car, the application on the mobile phone checked the status of the parked car.

The pilot of this study was started in late 2007 and lasted for about 3 months. There were about 55 randomly selected participants in the pilot phase [5]. For a demonstration video, please visit http://www.ouka.fi/video/smart/parking/.

(vi) *NFC at the theater*

The Oulu NFC at the Theater project was planned by the Oulu Theater and Kanresta, the restaurant service provider. TeliaSonera and MSG Software provided technical solutions. Users received theater tickets from the theater sales point using their NFC enabled mobile

Figure 9.6 Smart parking in action [1]. Photographed by Juha Sarkkinen, The City of Oulu, Smart Touch Project.

phones or wirelessly via TeliaSonera's backend system. In the theater restaurant, users were able to retrieve news via smart posters. Users were also able to download videos to their mobile phones via smart posters (see Figure 9.7).

The pilot of this project was carried out in 2007 for about 6 weeks. Overall, 158 people attended to the pilot studies and the service was piloted during nine performances. The aim of the pilot was to test how NFC technology can be used in entertainment events [6]. For a demonstration video, the readers are encouraged to visit http://www.ouka.fi/video/smart/theatre/.

(vii) *NFC at the restaurant*

In this project, users can order their lunch quickly via their NFC enabled mobile phones. A customer first touches an NFC tag placed on the table and then touches one of the tags on the menu list, and submits her order. Then TeliaSonera's mediator system receives the order and delivers it to the restaurant's cash system and kitchen.

Figure 9.7 NFC smart poster [1]. Photographed by Juha Sarkkinen, The City of Oulu, Smart Touch Project.

A couponing system is also integrated to the system, and customers may use electronic lunch coupons. When an order is given by using a coupon, the coupon is removed from the device immediately. The project aims to provide more throughput and revenue during busy lunch times. It also eliminates paper coupons. Information about the foods and beverages on the menu was also accessible through tags on posters and tables [2].

Other projects implemented in the SmartTouch Project [1] are:

- Bus ticketing;
- Work time management and drivers' diary;
- Lock management in public sport hall;
- Information tags in city environment;
- Future shop concept;
- NFC enabled blood glucose meter.

For a full list of projects and their details, the readers are encouraged to visit http://www.vtt.fi/inf/pdf/tiedotteet/2009/T2492.pdf.

9.1.2 City of Nice

City of Nice is assumed to be the first NFC city in Europe with a commercial NFC roll out by every French Mobile Network Operator (MNO). Three thousand NFC phones were marketed in 2010 with a primary focus on transport and payment services. The success of City of Nice was achieved through the works of MBDS, the graduate computer science degree and innovation laboratory of the University of Nice Sophia Antipolis in France (www.mbds-fr.org) with its international partners. The projects were conducted mainly on travel, m-tourism, healthcare, assisted living, m-payment, m-culture, m-government, m-education and fair trade.

Implemented projects in City of Nice consist of all operating modes (reader/writer, peer-to-peer and card emulation modes). Applications are stored and implemented on a single SIM card. The projects are developed in France, Morocco, Russia and Haiti with different partnership and contracts afterwards.

These projects combined with the University of Nice to host an important NFC pilot project (Nice Future Campus) in 2010–2011. Nice Future Campus project's aim is to replace physical student identification cards with NFC enabled mobile phones. Implemented projects in City of Nice are presented below with brief descriptions [13].

(i) *Nice Future Campus*

Nice Future Campus project was implemented in France in 2010–2011. Some of the partners involved were Extelia, Orange, Mobile Distillery and University of Nice Sophia Antipolis.

The project's primary aim is to replace physical student identification cards with NFC enabled mobile phones and to enable multiple applications in a single SIM card. All three operating modes are used in this project, and two pilots are implemented. The project consists of multiple applications from different categories including payment, transport, library, administration, ticketing and social student life. A student in this

project was able to pay, manage tickets and coupons, share an opinion regarding a book, get contextual information, communicate with her friends and much more with her NFC enabled mobile phone [13].

In the Nice Future Campus project, an NFC enabled university card can be used for:

- Payment;
- University restaurant ticket;
- Library access;
- Access control.

In the Nice Future Campus project, in the context of location based services, a student can:

- See the path to the next course;
- Find a given place;
- Find the location of a book in the library.

In the context of social network related services, a student can:

- Leave a contextual message to Facebook;
- Leave a message about a book on Facebook.

In the context of information services, a student can:

- Get contextual information regarding the campus;
- Get information on students advertisements, job offers, and contact details for a topic.

In the context of library services, a student can:

- See teachers' recommendations on a book;
- See ratings of a book provided by students;
- Read other students' comments on a book;
- Leave a comment on a book;
- Rate a book.

A web based platform is developed, and a student can check into the events using this platform, buy tickets and share events on social networks. Also using an NFC enabled mobile phone, a student can:

- See tickets;
- Use tickets.

(ii) *REVE*

The REVE project was implemented as part of the Nice Future Campus in Astrakhan in 2010. Students were able to get recommendations on books by touching their mobile phones to the NFC tagged books in the library. Social networking platforms such as Facebook and Twitter were also integrated to the system to build a "friends or followers" of a degree, a book, or a professor.

(iii) *TAP'nFlouss*

This project was implemented in Morocco in 2009–2011 and enables users to withdraw money from ATM machines with their NFC enabled mobile phones. The project was developed with Omnidata, NCR and M2T partnerships and NFC add-ons were used to provide NFC capability to Bluetooth phones [13].

(iv) *MBDS*

The project was implemented in Nice, Grasse, Spain, Tunisia, Egypt and Astrakhan in 2009–2010. The project enables the transformation of cultural places in cities into museums. NFC and 2D tagging is used and participants used their mobile phones to explore the cultural, thematic, family and historical paths in the cities [13].

Applications are developed for GPS based phones which have an Android operating system and NFC-sticker to gain NFC capability.

Implementations are:

- The science park of Sophia Antipolis (France) in 2008;
- The historical Valrose Campus of the University of Nice (France) in 2009;
- Menton and Grasse (France), Sidi Bou Said (Tunisia), Astrakhan Digital Kremlin Museum (Russia) and Sophia Antipolis science park as a part of the Sophia Zen project in 2011.

(v) *TICKET TAP*

The project was implemented in France in 2008. Virtual coupons and tickets are managed with the Cassis OTA (Over the Air) platform. This project was also used in the Nice Future Campus project in 2010 and the IMAJEANS project for street marketing and point of sale marketing to be deployed in Nice [13].

(vi) *CAMPUS NOVA*

The project was implemented in France in 2008 with the Credit Agricole bank. This was the first NFC French m-payment project which enabled 30 students to use their mobile phones for payment. Students were able to pay with their mobile phones as well as use their mobile phones for access control in classrooms. This project also helped new payment projects to be implemented in France such as Nice Future Campus [13].

(vii) *CAP ROUGE NIMERIK*

The project was implemented in Haiti in 2008 with an aim to track coffee bags for management purposes from rural areas in Haiti to the factory in Nice. For this purpose NFC tags were placed in the coffee bags. The project was implemented with big partners such as Malango (Coffee manufacturer from France), Solutions (software house in Haiti), Voila (Haiti MNO) and Alcatel Lucent (for placing WiMAX) [13].

(viii) *RoboDOMO*

The project was implemented in France in 2006 and runs to 2012 by Microsoft, Covea Tech and the Hospital of Nice. A robot from Robosoft Company was built around communication objects to monitor elderly people at home. Medicine delivery control, management of people working with elderly people at home, and home care visits are some of the implemented services [13].

(ix) *MOSTRA*

The MOSTRA project was implemented in 2005 and 2006 with partnerships with Amadeus and Renault. This project's aim is to use NFC enabled mobile phones as an access control object in cars and hotels and also to store virtual vouchers in the mobile device. This project also enabled an experiment to be carried out on the usage of a virtual NFC boarding pass at Nice Airport in 2009 [13].

9.1.3 Smart Urban Spaces

Smart Urban Spaces is a collaborative European project focusing on designing and adopting context aware services and e-city services based on NFC with the latest mobile and ubiquitous technologies. Within this scope, NFC applications and services are being prototyped and interoperability issues are investigated.

Smart Urban Spaces is a 3-year project from mid 2009 to mid 2012 and involves Finland, France, Spain, and Greece. Oulu, Helsinki, Caen, Valencia, and Seville are some of the other

cities involved. The project involves many organizations from the four participating countries. These organizations include service providers, software developers, universities, chip and SIM providers, MNOs, and system providers. The project is one of the largest efforts in Europe in creating EU wide city services [8–11].

The project's main purposes are:

- Framework for adopting e-city services;
- Technical and operational service analysis;
- Pilots and trials to test applications and services;
- Interoperability analysis;
- Identifying a set of European Urban Recommendations and Standards;
- Build a network of European cities to share experiences in mobile and context-aware services.

E-city services included in the project can be categorized as follows:

- Transport;
- Family and community;
- Leisure, culture and sports;
- Utility tools;
- Education and learning;
- NFC city ecosystem;
- Other specific services.

Some of the projects are considered briefly below.

(i) *Mobile ticketing in public transportation*

In this service, ticketing services will be transformed to NFC ticketing. NFC phones will be included in the overall development plan for the Helsinki metropolitan area fare and ticketing system 2014 by Helsinki Region Transport. As a first step, 700 000 travel cards have been changed to standard RFID based cards. Card readers in the vehicles have been transformed to NFC phone compliant readers.

(ii) *NFC technology in day care centers*

With this project, day care staff can monitor children's attendance including arrival and departure times using NFC technology. They also order meals using the attending children list.

(iii) *Oulu city card services*

Oulu city cards are NFC-compatible and equipped with two-dimensional codes. With this project, tickets can be checked from the card using a mobile phone.

(iv) *Intelligent signs in Hernesaari*

In Hernesaari, tags guide the users in and out of Hernesaari using intelligent signs. Touching a tag opens information in text, picture, audio or video format and then a user can be guided using this information.

(v) *City paths*

In this project, NFC tags are used for tourists to have a good experience in the city in a limited time. Digital tagging is used to guide users with a tourist map on the mobile handset.

(vi) *Bouquet of services*

The aim of this service is to provide multiple services to users within a single platform. For this purpose, the Oulu city card is used in different service entities to provide multiple services such as payment, event ticketing and transportation ticketing.

(vii) *Learning*

In order to support learning, several projects are being conducted. The aims of these projects cover:

- Improving learning through growth in motivation at school;
- Adding a sense of community through sharing and commenting on learned information;
- Improving communication between school and parents.

There are also other projects being conducted including smart sports, mobile math, multipurpose tagged cities, interoperable student cards, social networking and NFC solutions for mobile workers in the public sector. The projects also implement the following solutions to technical problems:

- Subscription to services;
- Unsubscribing to services;
- Application provisioning;
- Credit card payment;
- Multiple payment cards;
- E-purse payment;
- Access control and ticketing;
- Exchange tickets;
- Locking and unlocking an application;
- Deleting an application;
- Losing phone;
- Profile creation and edition.

9.2 NFC Trials and Projects

As already mentioned, there have been various NFC trials and projects throughout the world. Payment and ticketing applications are possibly the most well-known and promising everyday applications of NFC technology, and are the most complex from the ecosystem aspect as well. Thus most of the tests and trial projects are implemented in this application domain. In addition to the NFC cities, this section tries to present and illustrate different NFC projects in different countries. Some of these projects have been completed, or expanded into different application domains with growing participating entities, or still continue.

9.2.1 Contactless Payment Trials

(i) *Visa payWave payment pilot in Malaysia*

A short, but successful case comes from Malaysia that was launched at April 2006. The MNO Maxis Communications, Maybank which is the largest bank and financial group in Malaysia, and Visa collaborated to implement an NFC enabled payment in Malaysia. This is the world's first mobile Visa payWave payment pilot. It was implemented in the capital of Malaysia, Kuala Lumpur, involving about 2000 merchants and

200 participants. Participants were selected from Maybank Visa cardholders and Maxis mobile network subscribers. Other vendors that took part in this pilot were Vivotech for NFC enabled contactless terminals or readers, and NXP Semiconductors for NFC chip set and embedded hardware based secure elements (SEs) for mobile handsets. This payment trial had a big part to play in the commercial roll out of NFC services in Malaysia in April 2009 [7,14].

(ii) *HSBC trials in USA*

Another case comes from a global banking and financial services company. HSBC launched an NFC enabled mobile payment pilot in partnership with MasterCard in January 2007 in the USA. This pilot lasted for 6 months, and tested the use of NFC enabled mobile handsets in payment. The payment service was used where payment by contactless credit card and MasterCard PayPass was accepted. About 36 000 merchants accepted the MasterCard PayPass payment option at that time. More than 200 bank employees in New York, Chicago and several other large US cities used the OTA installation and personalization process to download HSBC credit card information on their NFC enabled mobile phones. The application itself and TSM platform was provided by Vivotech. Nokia 3220 NFC enabled mobile handsets were used with embedded hardware based SEs. These SEs and NFC chip sets were supplied by NXP Semiconductors [7,14].

(iii) *Payez Mobile project*

The Payez Mobile project is a joint initiative launched in November 2007 [12, 13]. It is a wide mobile payment service pilot implemented with about 1000 testers and 500 retailers in Caen and Strasbourg. It involves four MNOs (i.e., Bouygues Telecom, NRJ Mobile, Orange, and SFR), eight leading French banks (i.e., BNP Paribas, Crédit Agricole, LCL, Crédit Mutuel, CIC, Groupe Caisse d'Epargne, La Banque Postale, and Société Générale), Visa and MasterCard. This project combines the capabilities of the named organizations and also makes use of technologies developed in the ITEA SmartTouch project.

The contactless payment service provided by Payez Mobile is fully compatible with the existing Visa and MasterCard international specifications. Moreover, MasterCard and Visa applications can be hosted simultaneously in the same SE. The global objective of the participants in the Payez Mobile trial is to create a common vision, namely, a business solution for banks and MNOs in the contactless payment application domain.

Technically, the Payez Mobile experiment relies on the combination of four issues. The first issue is that payment applications provided by banks are installed on multi-application enabled UICC based SE of the mobile phone. The second issue is that NFC technology is used only in handling the payment service from the NFC enabled mobile device to the merchant terminal. The third issue is that Single Wire Protocol (SWP) is used for managing communication to UICC based SE from the NFC interface. Finally, advanced OTA mechanisms are required to deploy multiple applications remotely. The providers of UICC based SEs and secure OTA platforms are Gemalto and Obertur Card Systems. In addition, the NFC enabled mobile phones used in the project are the Motorola L7, Sagem My700, and LG L600V.

This contactless mobile payment service uses the existing bank card infrastructure. The payment methods are defined by Payez Mobile. The first method is for amounts less than 20 euros, so that if customers wish, they can pay without using a PIN. The second

method is for amounts exceeding 20 euros where a PIN is required. For these methods, each bank can deploy one or more NFC enabled payment applications and these can be loaded and installed on the customer's UICC of the NFC enabled mobile phone.

This payment solution is suitable for all amounts, and has been tested with people of all ages. According to the Association Europeenne Payez Mobile, by the end of 2008, surveys have shown that most customers consider this payment solution as "highly practical", "easy" and "attractive". More than 90% of all customers expressed "satisfaction" with this solution, and 59% stated that they would use this service once it becomes available. In addition, 82% of the merchants are satisfied with this payment service, and appreciate the speed of the contactless mobile payment service. For more information about the Payez Mobile project, specifications and demonstrations, the readers are encouraged to visit http://www.payezmobile.com [12].

(iv) *C1000 NFC pilot with Rabo Mobile in the Netherlands*

The Dutch based Rabobank has become the first bank in Europe to introduce mobile banking and low-cost calling services in a different way with Rabo Mobile. It is a Mobile Virtual Network Operator (MVNO) that is fully owned by Rabobank. This new service changes the current strategy from "going to the bank during opening hours" to "the bank that is accessible anytime, anywhere". Rabobank offers its customers transparent and efficient customer support with Rabo Mobile. The partners within this scheme are the multimedia company Talpa and the MNO Orange providing the mobile network service [7].

Initially Rabo Mobile is focused on Rabobank's Internet banking customers. Also various products and services are developed for different customer bases such as small and medium sized enterprises. This is a new and innovative way that combines traditional banking and internet banking with the fast paced mobile operator business.

In August 2007, Rabo Mobile initiated a new NFC pilot called "Pay with your mobile phone at C1000" in the Netherlands. C1000 is one of the largest Dutch based supermarket chains. A number of NFC enabled applications in C1000 retail stores including mobile payment and loyalty services were implemented during a 6-month period. About 100 customers of the C1000 supermarket and Rabobank participated in the pilot in the small town of Molenaarsgraaf. They used NFC enabled Samsung mobile phones for their shopping, and paid for their purchases by simply touching their phones to the NFC readers at the checkout. Then they entered their PIN code to the POS terminal and completed the transaction. This is a good example of a contactless online debit application in Europe that enables the use of the existing PIN system with a mobile phone. Moreover in this trial customers can return their empty bottles and gain discount receipts to be used at the checkout from bottle machines which are located within the supermarket or they can have a refund credited to their Rabobank accounts.

According to the results obtained by the C1000 retailer and trial co-organizer Schuitema, about 68% of the users said that they prefer mobile payment. The remainder said that it did not matter what form of payment they used. Furthermore, about half of the users said they would buy an NFC enabled mobile phone if more NFC services were widely available including payment. More than 70% of the users said that they wanted to enter a PIN code to complete transactions even in low valued purchases. However, about 78% of the users do not want to pay a few euro cents extra per transaction for an NFC payment [7].

(v) *EZ-Link and StarHub trials in Singapore*

A subsidiary of the land transport authority, EZ-Link, processes more than 4 million financial transactions daily in Singapore and issues millions of cards. EZ-Link launched an NFC enabled ticketing trial with the MNO StarHub in October 2007 which lasted for 6 months. Throughout the trial, about 20 000 terminals were used which accept Singapore's transit purse, EZ-Link. These terminals were generally on trains and buses, however they can also be found in some retail stores, restaurants, and vending machines [7].

The StarHub and EZ-Link Singapore trial was the world's first NFC FeliCa trial providing ticketing as well as smart poster services. Users can touch tags located on smart posters to download URLs, coupons or other promotional content. Users can also check their purse balance, transaction history and other details from a mobile device.

The other partners of this transit ticketing trial were Vivotech as the NFC reader supplier and NXP Semiconductors as the provider of NFC chip set for mobile phones. The trial was implemented with more than 800 users who had used specially designed NFC enabled mobile phones provided by Sony with embedded FeliCa SEs.

According to the trial's results, if the technology is offered, 23% of post-trial survey respondents said they would be "very likely" to adopt the technology. Another 45% of the respondents said that they would be "likely" to adopt the technology. Another important result is that about 83% of users used their phones during the trial to pay for transit fares, and about 70% of users checked their transaction histories [7].

(vi) *Pay-Buy-Mobile trial in Australia*

A Pay-Buy-Mobile (see Chapter 7) pilot was performed in Australia with participation of Australia's largest MNO Telstra, National Australia Bank and Visa (payWave). The pilot was launched in August 2008 and lasted for 3 months. About 500 users and 12 merchants participated in this small pilot. The users were generally from the Telstra and National Australia Bank staff.

Telstra provided and issued SIM based SEs for users. Payment application of the National Australia Bank and Visa was installed on SIMs of Sagem my700X NFC enabled mobile phones. The other vendors were Inside Contactless for NFC chip set in mobile phones, Vivotech for NFC readers in the merchant stores and Cassis International for TSM business solutions. With this pilot, users could touch to the NFC readers for purchases of $35 or less. According to the results of Telstra, about 95% of trial participants said that they would be "likely" or "extremely likely" to use the NFC technology in the future [7].

(vii) *ING Bank trial in Romania*

The Dutch bank ING and MasterCard (PayPass) tested the viability of NFC technology in mobile payment systems for low value purchases in Romania. The trial was launched in November 2008 and lasted for 7 months. The trial involved many MNOs in Romania and only one TSM platform provided by Venyon.

This trial was one of the first which enabled users to top up and check their contactless MasterCard PayPass account balances via OTA with a special code. This trial was also the first for ING Bank and MasterCard in Romania. They worked with the payments technology vendor Collis, OTA service platform provider for NFC payments Venyon and Taiwan based NFC outfit Toro on the project [7].

NFC enabled payment terminals were established at about 11 stores including fast food restaurants, cinemas, and newsagents. About 360 bank customers participated in the trial with Nokia 6212 NFC enabled mobile phones which have embedded SEs where the payment application was installed.

(viii) *Cep-T Cuzdan Launch in Turkey*.

Another good NFC launch case comes from Turkey. Garanti Bank, Yapı Kredi Bank and the MNO Turkcell collaborated in the NFC enabled mobile-wallet service. The service is launched commercially in 2011 [7].

Turkcell brought the NFC enabled Android phone (U8650NFC Sonic) to the market as Turkcell T20. Turkcell preloads the mobile wallet software to the mobile phone which also supports more than one bank-issued application. The phone supports the single wire protocol standard, enabling secure applications to be stored on SIM cards. Turkcell also supports its mobile wallet flexible antennas for the mobile phones those do not have NFC capability.

Turkcell is also serving as TSM to download and manage secure applications in its wallet. The TSM platform is built in-house. Two payment applications from two of Turkey's largest privately owned banks are available currently.

9.2.2 Transport or Other Ticketing Trials

(i) *NFC public transport ticketing with RMV in Hanau*

RMV (Rhein-Main-Verkehrsverbund) is one of the largest regional public transport authorities in Europe that provides transportation service for five million inhabitants in the state of Hess, Germany. Nokia, and Vodafone as one of the largest MNOs together with a public transport authority for Frankfurt's greater area, RMV, performed a joint project in the NFC enabled transport ticketing service domain. The trial started in early 2005 with about 200 users [7].

In this trial, RMV customers used Nokia 3220 NFC enabled mobile phones which have a smart NFC shell where tickets are stored to access a local bus network in Hanau, a city near Frankfurt. The RMV electronic ticketing application is securely stored on an integrated smart card controller in the mobile phone.

According to the first survey results, NFC enabled mobile phones are seen as more attractive and innovative than smart cards. Many pilots have been done with other operators and third parties. The service expanded into broader applications, including information, loyalty, and payment application domains with growing number of users. Currently, NFC ticketing with RMV is commercially available with loyalty programs. For example, Nokia NFC enabled mobile phones can also be used as a bonus card called the "RMV ErlebnisCard".

(ii) *NFC stadium experience in Manchester*

The MNO Orange UK performed a trial of contactless ticketing services with Manchester City Football Club. This small trial was launched in August 2006 with about 20 users who held valid season tickets. Nokia 3220 NFC enabled mobile phones with embedded hardware based SEs were used. Manchester City Football Club and Orange UK provided a ticketing application for these devices. The participating fans were allowed to use their NFC enabled mobile phones to touch to the NFC readers at the gates of the Manchester City football ground and enter the turnstiles easily [7].

(iii) *Bouygues Telecom trials in Paris*

France's major MNO Bouygues Telecom performed a 3-month NFC enabled transit ticketing trial in Paris. This trial was launched in November 2006. About 50 users participated in this trial. The main service providers were RATP (Régie Autonome des Transports Parisiens) and SNCF (Société Nationale des Chemins de fer Français) who are the providers of Navigo contactless transit fare cards. They provided an NFC enabled ticketing application which was loaded and installed on the user's SIM based SE. This trial's aim is to enable users to pay for fares at gates or at readers on buses which accept the Navigo ticketing application using their NFC enabled mobile phones.

Users also could recharge their cards over the mobile internet service. Bouygues Telecom provided an NFC enabled mobile phone from a Japanese mobile handset manufacturer, NEC, which was specially designed for the trial. Other vendors involved were Axalto/Gemalto as the SIM based SE provider and Inside Contactless as the NFC chip set provider. This first trial by Bouygues Telecom led to subsequent transit ticketing and payment pilots in France [7].

(iv) *O2 Wallet*

Telefonica O2, as one of the largest MNOs, announced O2 Wallet in November 2007 and performed a 6-month trial in conjunction with various service providers, namely, TfL (Transport for London), TranSys who operates the Oyster smart card for TfL, Venyon for TSM business solutions, Barclaycard, Visa Europe (payWave), Nokia, Giesecke & Devrient for data management, Innovision for smart poster tags, NXP Semiconductors, Inside Contactless, Consult Hyperion for consulting, and AEG Europe. O2 Wallet included many services from transport ticketing to smart poster applications, and was the UK's first large scale NFC pilot [7,15].

O2 Wallet pilot paved the way for the large usage of mobile phones as Oyster cards for travel around London, paying for purchases by Barclaycard and accessing events. About 500 O2 mobile network subscribers participated in the pilot and they were equipped with Nokia 6131 NFC enabled mobile phones. Embedded hardware based SEs were used in the pilot. The user only needs to store an NFC enabled Oyster application on her mobile phone and preload her application. This application eliminates the need for users to carry Oyster smart cards in their wallets. Users can pay for their travel expenses through the Oyster application by simply touching their mobile phones to the Oyster NFC readers at London underground tube stations, and on buses and trams (see Figure 9.8). If the user's phone rings while making a transaction, she can still answer the call. A call or text message does not interfere with the NFC service.

In addition to the preloaded ticketing application in O2 Wallet, payment, smart poster and access control applications are enabled by the service providers. The payment service is provided by Barclaycard who introduced the first credit card in the UK and Visa payWave for O2 Wallet. Users were able to make payments with their Barclaycard payment application installed on their mobile phones. This payment application can be used at about 5000 merchants including Books Etc., Chop'd, Coffee Republic, EAT, and Krispy Kreme.

Users could also touch to smart posters to gather information on restaurants and other locations. The tags on smart posters serve as shortcuts for services enabled through the mobile phone. For example, when a user touches a tag on a smart poster, she can automatically dial a number, send a text message or view a website containing information

Figure 9.8 O2 Wallet [1]. Photographed by Juha Sarkkinen, The City of Oulu, Smart Touch Project.

about an event, and so on. Another important service of O2 Wallet is access control. Users can access and enter the VIP area of O2 entertainment venues using their NFC enabled mobile phone.

According to the results from conducted surveys after the O2 Wallet pilot, more than half of the participants said that if contactless services were available, they would use them on their mobile phones. About 90% of the participants were satisfied with NFC technology, and again a large majority said that touching mobile phones is more convenient than using Oyster smart cards. Currently, there is a lot of work being done to expand the services of O2 Wallet. The pilots and trials are leading O2 Wallet to a commercial launch.

(v) *Orange trials in Spain*

Another transit ticketing trial was launched in April 2008 in Malagna. This was a small trial that tested fare collection on buses with about 50 users. The partners in this pilot were the EMT as the transit fare collection provider, and France Telecom-Orange Spain as the MNO. The NFC enabled mobile phones supplied by Sony were the Sony Ericsson Z750i. The application was stored on SIM based SEs and was provided by Oberthur Technologies. This was Orange Spain's first NFC pilot in the NFC based ticketing application domain [7].

9.2.3 Other Trials

(i) *London Fashion Week trials*

An attractive and innovative case was implemented by one of the largest MNOs, Telefonica O2, in the UK. O2 organized and performed a small trial in February 2008 at London Fashion Week which is the key event for designers in London to show their designs to fashion buyers from around the world. The aim of this trial was to provide fashion buyers the opportunity to give instant feedback to the collection of Emilio de la Morena. This NFC enabled messaging trial was performed with a limited number of buyers or users.

In order to enable this trial, smart posters with NFC tags were used to send a text message automatically to Emilio de la Morena. Also, fashion buyers were equipped with Nokia 6131 NFC enabled mobile phones, and when a buyer touched NFC tags located on smart posters with her mobile phone, she registered her feedbacks and interests in a certain Emilio de la Morena's Autum/Winter collection design. The text message was then automatically sent to the designer [7].

(ii) *Pass and Fly pilot at Nice Airport*

Pass and Fly was a joint project of Nice Cote d'Azur Airport and Air France in partnership with Amadeus and IER. This pilot was launched in April 2009 at Nice Cote d'Azur Airport and lasted for 6 months. The aim of the pilot was to enable passengers to download digital boarding passes to their mobile phones using NFC technology. Amadeus developed the applications for mobile phones, the departure control system and airport readers to share and display information relevant to the passenger boarding process. IER provided NFC booths and readers integrated into the airport's infrastructure and connected to Air France's passenger management system. Air France provided the electronic boarding passes [7].

Members of Nice Airport's Club Airport Premier (CAP) program and Air France's frequent flyer program participated in the Pass and Fly pilot. They used Nokia 6212 NFC enabled mobile phones or NFC stickers to get their electronic boarding pass and collect loyalty points. Applications were loaded on the Nokia 6212 NFC enabled mobile phones which uses an embedded hardware based SE.

Boarding passes complied with the International Air Transport Association format. The NFC application also allowed users to earn points in the airport's frequent flyer loyalty program. To enable the download, the passenger first needed to check-in over the Internet before going to the airport.

After the check-in process was completed, the user touched her mobile phone to the Pass and Fly NFC reader. The machine then identified the passenger and found her frequent flyer and boarding pass information. The digital boarding pass in IATA format was sent to the user's mobile phone. Also, the Nice Airport CAP frequent flyer points were automatically loaded. Hence, members of the loyalty program did not need to go to a separate kiosk to obtain their credits.

Then the user again touched her mobile phone to a second NFC reader at the security inspection point. This terminal displayed the boarding pass electronically for security staff. At the boarding gate, airline staff only needed to check the passenger's identification information. The NFC reader checked the boarding pass for a third time, and the terminal printed a coupon with a seat assignment. Pass and Fly is the one of the first, innovative pilots in the airline industry. This new NFC enabled infrastructure and information flow speeds up the traveler's airport experience with time and cost savings.

(iii) *Student attendance at a London college*

NFC enabled payment and ticketing applications are currently very popular. Additionally, attendance and other workforce management applications can be implemented as one of the NFC services through mobile phones. A good trial is taking place at Newham College in the UK. This trial started in January 2010 and is still in progress [7].

In accordance with the trial, Newham College distributed Nokia 6212 NFC enabled mobile phones to four teachers, who touch these mobile phones to the identification cards of about 120 students in place of taking the register each day. The teachers log in

themselves by touching their phones to the tags when they arrive at their classrooms. The attendance data are captured and sent from NFC enabled mobile phones to a "time-and-attendance system" provided by Reslink, a Finnish software company.

The college uses the "time-and-attendance system" to save time and paperwork costs for taking attendance at the beginning of each class and to reduce paperwork for administrators. Student identification cards for the pilot have been modified with contactless stickers containing identification numbers, so that when teachers touch to them, the student identification is transferred to the teacher's mobile phone. If this trial becomes successful, it could lead to roll out within the college to 300 teachers and 16 000 students.

9.3 Chapter Summary

There are currently many trials and projects to realize NFC technology. Some models are developed by universities, companies and also as a joint effort between universities and companies. With these trials and projects, the participating entities try to understand the nature of the NFC technologies from both their social and business aspects.

NFC cities are the most popular implementations of NFC technology. The purpose of an NFC city may be either to test an implementation, or even to actually use one in a defined arena. NFC cities are used mostly to test the social aspect of the NFC technology compared with trials and projects. The usability issues together with the problems residing in the technology are obtained continuously through tests easily in NFC cities. In the case of NFC trials and projects, the business aspect of an application or an NFC ecosystem is tested more than the social aspect. In this chapter, we have briefly presented all NFC cities (City of Oulu, City of Nice and Smart Urban Spaces) and some important implementations of NFC technology throughout the world.

References

[1] Tuikka, T. and Isomursu, M. (eds) (2009) *Touch the Future with a Smart Touch*, VTT Tiedotteita – Research Notes 2492, Espoo, Finland, 2009. Available at: www.vtt.fi/inf/pdf/tiedotteet/2009/T2492.pdf (accessed 10 July 2011).

[2] Smart Touch City of Oulu – Services through NFC Technology. Available at: ttuki.vtt.fi/smarttouch/www/kuvat/December08_Newsletter.pdf (accessed 10 July 2011).

[3] Häikiö, J., Wallin, A., and Isomursu, M. (2009) *Touch the Future with a Smart Touch*, Section 5.2.2, VTT Tiedotteita – Research Notes 2492, Editors: Tuikka, T., Isomursu, M., Espoo, Finland, 2009. Available at: www.vtt.fi/inf/pdf/tiedotteet/2009/T2492.pdf (accessed 10 July 2011).

[4] Rouru-Kuivala, O. (2009) *Touch the Future with a Smart Touch*, Section 5.2.3, VTT Tiedotteita – Research Notes 2492, Editors: Tuikka, T., Isomursu, M., Espoo, Finland, 2009. Available at: www.vtt.fi/inf/pdf/tiedotteet/2009/T2492.pdf (accessed 10 July 2011).

[5] Rouru-Kuivala, O. (2009) *Touch the Future with a Smart Touch*, Section 5.1.2, VTT Tiedotteita – Research Notes 2492, Editors: Tuikka, T., Isomursu, M., Espoo, Finland, 2009. Available at: www.vtt.fi/inf/pdf/tiedotteet/2009/T2492.pdf (accessed 10 July 2011).

[6] Rouru-Kuivala, O. (2009) *Touch the Future with a Smart Touch*, Section 5.4.1, VTT Tiedotteita – Research Notes 2492, Editors: Tuikka, T., Isomursu, M., Espoo, Finland, 2009. Available at: www.vtt.fi/inf/pdf/tiedotteet/2009/T2492.pdf (accessed 10 July 2011).

[7] NFC Times, Project Map, http://www.nfctimes.com/nfc-projects (accessed 10 July 2011).

[8] Smart Urban Spaces, http://www.smarturbanspaces.org/ (accessed 10 July 2011).

[9] Smart Urban Spaces – City of Oulu, http://www.ouka.fi/sus/english/ (accessed 10 July 2011).

[10] Research Notes, 2011 Third International Workshop on Near Field Communication, Hagenberg, Austria, 22 February 2011.
[11] Research Notes, WIMA NFC 2011 – 5th Global NFC Applications Products & Services Congress, Monaco, France, 19–21 April 2011.
[12] Payez Mobile, http://www.payezmobile.com (accessed 10 July 2011).
[13] Miranda, S. and Pastorelly, N. (2011) *NFC Mobiquitous Information Service Prototyping at the University of Nice Sophia Antipolis and Multi-mode NFC Application Proposal*. Proceedings of 2011 Third International Workshop on Near Field Communication, Hagenberg, Austria, 22 February 2011, pp. 3–8.
[14] Mobey Forum Enrollment Task Force (2008) *Best Practices for Mobile Financial Services*, Mobey Forum, 2008, Available at: http://www.mobeyforum.org/content/download/460/2768/file/Best%20Practices%20for%20MFS%20Enrolment%20Business%20model%20analysis%20final.pdf (accessed 10 July 2011).
[15] O2 News Centre, http://mediacentre.o2.co.uk (accessed 10 July 2011).

Index

Near Field Communication: From Theory to Practice, First Edition. Vedat Coskun, Kerem Ok and Busra Ozdenizci.
 Published 2012 by John Wiley & Sons, Ltd.