UNIVERSIDADE DA BEIRA INTERIOR
Faculdade de Engenharia
Departamento de Informática

Mobile Applications Approaches using Near Field Communication Support

David Miguel Oliveira Bexiga Monteiro

Submitted to the University of Beira Interior in candidature for the Degree of Master of Science in Informatics Engineering

Supervised by Prof. Doutor Joel José Puga Coelho Rodrigues

Department of Informatics
University of Beira Interior
Covilhã, Portugal
http://www.di.ubi.pt

Acknowledgements

First of all, I would like to thank my supervisor Professor Joel José Puga Coelho Rodrigues for the opportunity to work with him in this ambitious project. Although my work life begins he never abandoned me and always gives me the guidance, support and encouragement to finish another important goal of my life.

I am most grateful to the University of Beira Interior, the Instituto de Telecomunicações and Next Generation Networks and Applications Group (NetGNA) for allowing me to work on this project with them.

A special thanks to my company TIMwe, my boss Paulo Salgado and all my team members, without them it was impossible to finish this task.

Many thanks to all my closest family for all support, encouragement and dedication.

And last a huge thanks to my girlfriend Carina Isabel Alçada Valério. She was the most important figure is this hard work. Thanks to very helpful hours that she ears me with my ideas and my algorithms without interest for her. And finally thanks for her love, support, dedication and encouragement.

My thanks to all the other friends not mentioned here.

Abstract

Nowadays, the society is constantly evolving technologically and new products and technologies appears every day. These technologies allow the well-being of societies and their populations.

Mobile gadgets evolution, mainly the smartphones, has always been at the forefront, everyday new devices appear and with them, more recent technologies. These technologies provide a better quality of life of everybody who uses them.

People need to have at their disposal a whole array of new features that make their life increasingly more easily. The use of gadgets to simplify the day-to-day is growing and for this people use all disposal types of devices, such as computers, laptops, file servers, smartphones, tablets, and among of others. With the need to use all these devices a problem appears, the data synchronization and a way to simplify the usage of smartphones. What is the advantage of having so much technology available if we need to concern about the interoperability between all devices?

There are some solutions to overcome these problems, but most often the advantage brought by these technologies has associated some setup configurations and time is money.

Near field communication (NFC) appeared in 2004 but only now has gained the market dominance and visibility, everybody wants to have a NFC based solution, like Google, Apple, Microsoft and other IT giants.

NFC is the best solution to overcome some problems like, file synchronization, content sharing, pairing devices, and launch applications without user interaction. NFC arises as a technology that was forgotten, but it has everything to win in every global solutions and markets.

In this dissertation two based solutions are presented, an application to transfer money using NFC and an application launcher. Both solutions are an innovation in market because there are nothing like these. A prototype of each application was build and tested. NFC Launcher is already in Android Market. NFC Launcher and Credit Transfer were built, evaluated and are ready for use.

Keywords

Near Field Communication, Trusted Service Manager, NFC Data Exchange Format, Record Type Definition, Peer-to-peer, Secure Element, Smartphone, Mobile Device, Chipset.

Contents

List of Figures

List of Tables

Acronyms

NDEF	NFC Data Exchange Format
RTD	Record Type Definition
LLCP	Logical Link Control Protocol
TSM	Trusted Service Manager
MNP	Mobile Network Provisioning
P2P	Peer-to-Peer
NFC	Near Field Communication
PCD	Proximity Coupling Device
PICC	Proximity Inductive Coupling Card
SWP	Single Wire Protocol
HIC	Host Controller Interface
IC	Integrated Circuit
EE	Execution Environment
UICC	UMTS Integrated Circuit Card
SE	Secure Element
SAM	Secure Access Module
OTA	Over the air
PKI	Public Key Infrastructure
EAL	Evaluation Assurance Level
CC	Common Criteria
JCOP	Java Card Open Platform

LLCP	Logical Link Control Protocol
APDU	Application Protocol Data Units
SCWS	Smart Card Web Server
SAT	SIM Application Toolkit
BIP	Bearer Independent Protocol

1. Introduction

1.1. Focus

Near Field Communication (NFC) [1, 2] is a new short-range wireless connectivity technology [3] with high expectations for innovative information services that emerged from the combination of contactless identification (RFID - Radio Frequency Identification [4]) and cell phones.

NFC [5] was launched in 2004 by Philips (NXP), Sony, and Nokia. NFC can be used with a large variety of devices for touching connectivity: consumer electronics, mobile devices, locks, objects, printers, TV and PCs [6]. Consumers will be able to easily access a variety of services [7, 8] (m-payment, transport, travel, infotainment, culture...) and conveniently exchange information with a simple touch gesture utilizing NFC technology.

The NFC technology is revolutionizing the mobile services world, and people have to know its technical aspects to exploit it today in their projects. The paper ticket or coupon is now a thing of the past. Our cell phone can be used to store virtual vouchers [9], transport ticket, payment [10], coupons, etc. Sky is the limit in terms of innovative content and services of our digital future where they will be location-based and touch-based.

Near Field communication is a short-range (< 5cm) wireless communication technology.

NFC technology has two operation modes. NFC chips [11] or tags [12] can perform as passive or active [13]. If the device has active state, it has own energy and needs a battery giving it energy [14], Otherwise, if is a NFC tag, it works in passive mode, it does not have own energy, and it only works when it receives energy from another NFC device. When it receives this energy it passes from the passive mode to active mode.

NFC technology allows three modes of communication, read / write [15], peer-to-peer [16] , and card emulation mode [17]. So a NFC device can act as NFC tag emulator or a tag reader. The first mode, read / write, gives ability to read and write a passive NFC tag using a NFC enabled device. Common example with the utility of this mode is smart poster which can get more information from those posters by touching with the device over it. In the second mode, peer-to-peer, the communication is performed between two NFC enabled devices [18]. Using peer-to-peer mode, devices can exchange data between each other like virtual business cards or photos. The card emulation mode is the last. In this mode, devices can emulate an existing contactless card, which gives a possibility to communicate with contactless reader. For example, it maybe use to make a payment by touching with a device over a payment terminal. Unfortunately these operating modes depend on device implementation and not every device has all the three modes.

NFC protocol distinguishes between the initiator and the target of a communication. Any device may be either an initiator or a target [19]. The initiator is the device that initiates and controls all the communication. The target is the device that answers a request from the initiator.

NFC tags are passive devices that can be used to communicate with active NFC devices. NFC tags can be used within applications such as posters, and other areas where small amounts of data can be stored and transferred to an active NFC devices. Within the poster the live area can be used as a touch point for the active NFC device.

The stored data on the NFC tag may contain any kind of data, but common applications are for storing URLs from where the NFC device may find further information. In view of this only small amounts of data may be required. NFC tags may also be used.

In order that the communication between the active NFC reader/writer and the passive NFC tag was defined [20]. The NFC forum introduced their first standard technology architecture and standards for NFC compliant devices in June 2006. This included the NFC Data Exchange Format, NDEF [21], and three Record Type Definitions, RTD [22]. These are for smart poster, text, and Internet resource reading applications.

To ensure the safety of stored data relating to applications for payment and ticketing, NFC standards advise the creation of a secure element [23].

Secure element [24, 25] is the area less defined in the NFC technology and currently there are three places used to store this information: a SIM card [26], a memory card, and an embedded chip in the phone.

1.2. Objectives

The main objective of this dissertation is the design, constructing, deployment, and performance evaluation of a mobile system solution for Android to turn more easy transfer money between two mobile devices and to launch any application in a mobile phone without human interaction between the man and the machine.

The Android applications will use the new NFC communication technology. Both applications should works over this technology. A solution for credit transfer will be implemented to turn easy transfer money between two devices without sending any messages to operator. The user only needs to insert the value and transfer it. NFC Launcher is an easy way to launch applications without user action. The user only needs to touch a

NFC with his device and the application automatically is launched. The user only needs to configure a smartcode in a Web portal before.

To reach these main objectives the following intermediate objectives were defined:

- Study of the related work, about NFC technology, user interaction between device and Android system;
- Detailed analysis of the NFC technology communications;
- Proposal, implementation and validation of the system based on requirement analysis;
- Construction and deployment of a mobile application on Android OS;
- Performance evaluation and system validation on Android based systems.

The research work and engineering is expected to produce not only the Android applications, but also to provide the opportunity to disseminate the knowledge and software through, at least, a conference paper.

Prototyping will be used as research methodology. For demonstration of NFC technology it will be built a real prototype of each application. These prototypes will be evaluated, demonstrated, and validated.

1.3. Main Contributions

This section is dedicated to the scientific contributions of this dissertation to the state-of-the-art on NFC technology and Android development. The main contribution is a mobile and secure NFC application for credit transfer between mobile phones. This proposal was submitted to a major international conference of the IEEE. Other two contributions are the innovations performed by the NFC Launcher and the presentation of a tutorial about NFC technology and its potential. These contributions will be

submitted to a major international conference and an international journal, respectively.

1.4. Dissertation Structure

This dissertation is organized in six chapters. This chapter, the first, presents the context of the dissertation, focusing on the topic under study, the objectives, the main contributions and the dissertation structure and its main contribution.

Chapter two reviews the literature about the NFC technology and all the related background, including communication protocols, NFC tags, security mechanisms, and mode of operation.

In chapter three it is presented the requirement analysis where it was analyzed all applications requirements.

Chapter four focuses on the system demonstration and validation of the credit transfer application. The conceptual design, used technologies, and performance tests done to Credit transfer are shown.

The system demonstration and validation of NFC Launcher application is presented in Chapter five. The conceptual design, used technologies, and performance tests performed with the NFC Launcher are presented.

Conclusion and future works are available in the Chapter six.

2. Related Work

This chapter addresses the current state of art on Near Field Communication [27]. The first section describes the NFC operation modes and communication modes. The second and third section describe the communication protocols and the existing NFC tags.

2.1. NFC Operation Modes

NFC is a half-duplex [28] communication which means that it has two components: a sender and a receiver. Any one of them can be the sender or the receiver, but only it is possible one communication at the same time.

The first device that sends a signal to another is the sender device and the other one is the receiver. In NFC [29] the sender always begins the communication and the receiver is the target of communication. The target answer initiator's request.

In NFC technology there are two operation modes: active mode and passive mode [30]. Any device with battery can work in one of these two modes. If the device does not have battery it only works in passive mode, a

good example are the NFC tags. In active mode both devices create their own magnetic field.

Table 1 shows the coding used in active mode. Manchester and modified Miller are the preferred. They are used by both, the initiator and the target. If an active device transfers data at 106 kbit/s, a modified Miller coding with 100% modulation is used. In all other cases Manchester coding is used with a modulation ratio of 8% to 30%.

Table 1 - NFC active mode coding to data transfer.

Standard	Initiator to target	Target to initiator
Active Mode at 106 kbps (fc / 128)		
ISO 18092	ASK 100% Modified Miller	ASK 100% Modified Miller
Active Mode at 212 kbps (fc / 64) and 424 kbps (fc / 32)		
ISO 18092	ASK 8-30% Manchester	ASK 8-30% Manchester

In passive mode, only one of the devices creates a magnetic field. This field sends energy to the passive element to establish the communication between two devices.

Table 2 shows the coding used in passive mode. Manchester, modified Miller, and NRZ are used. If an active device transfers data at 106 kbit/s, modified Miller with 100% modulation and NRZ with 8% to 14% coding are used. In all other cases Manchester coding is used with a modulation ratio of 8% to 30%.

Table 2 - NFC passive mode coding to data transfer.

Standard	Initiator to target	Target to initiator
Passive Mode at 106 kbps (fc / 128)		
ISO 18092 (ISO14443-A)	ASK 100% Modified Miller	Load Modulation Subcarrier (847,5 kHZ) Manchester
ISO 14443-B	ASK 8-14% NRZ	Load Modulation Subcarrier (847,5 kHZ) BPSK-NRZ
Passive Mode at 212 kbps (fc / 64) and 424 kbps (fc / 32)		
ISO 18092 (FeliCa)	ASK 8-30% Manchester	Load Modulation Manchester

2.2. NFC Communication Modes

Operating at 13.56 MHz and transferring data at up to 424 Kbits/second, NFC provides intuitive, simple, and safe communication between electronic devices [31]. NFC can be a “read” and “write” technology. Communication between two NFC-compatible devices occurs when one of them stay close from the other. An NFC device can offer three types of communication mode: Reader/Writer, Card Emulation [32], and Peer-to-Peer [33]. Table 3 shows all NFC communication modes.

Table 3 - NFC Communication modes.

<table>
<tr><th>Peer-to-Peer</th><th>Read / Write</th><th>Card Emulation</th></tr>
<tr><td colspan="3">Applications</td></tr>
<tr><td>NFC Forum protocol bindings
IP, OBEX, ...</td><td>RTDs & NDEF</td><td rowspan="2">Card Emulation</td></tr>
<tr><td>P2P -LLCP</td><td>NFC Tag Type 1, 2, 3, 4</td></tr>
<tr><td colspan="3">RF transport layer
(ISO 18092, ISO 14443-4, FeliCa, Proprietary Protocol)</td></tr>
<tr><td colspan="3">RF physical layer / RF link layer / data frames and anti - collision layer
(RF Layer ISO 18092, ISO 14443A&B, FeliCa)</td></tr>
</table>

Operating in Reader/Writer mode, the NFC device can read and alter data stored in NFC compliant passive (without battery) transponders. Such tags can be found on SmartPosters [34] e. g., allowing the user to retrieve additional information by reading the tag with the NFC device. Depending on the data stored on the tag, the NFC device takes an appropriate action without any user interaction. If a URI was found on the tag, the handset would open a web browser for example. In Figure 1 a NFC Reader / Writer mode is shown. Mobile device read a tag using the embedded NFC chip. ISO 14443 and ISO 15693 are the standards for data transmission used in this type of communication between tags / smartcards and a NFC based device.

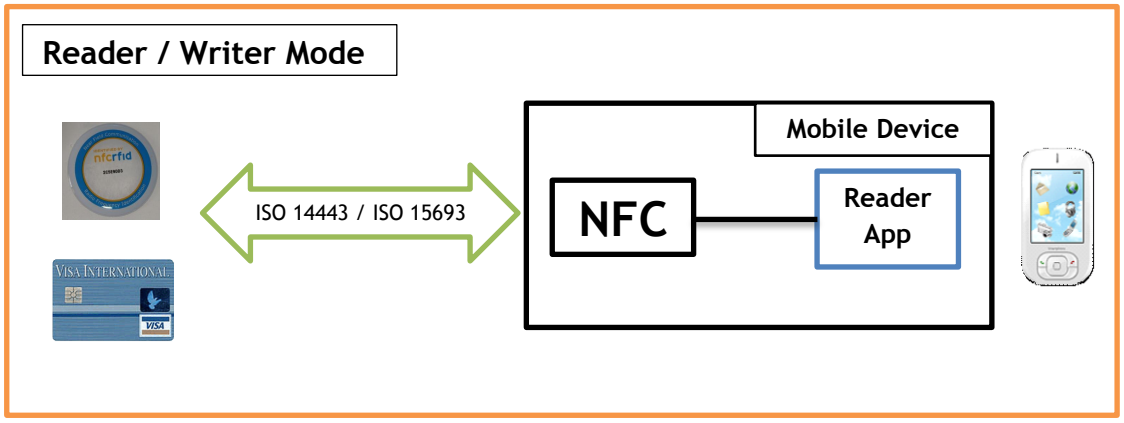


Figure 1 - Reader / Writer mode.

One NFC device can act as smart card (ISO 14443) after being switched into card emulation mode. In this case an external reader cannot distinguish between a smart card and an emulated NFC device. This mode is useful for contactless payment and ticketing applications for example. Actually, an NFC enable handset is capable of storing different contactless smartcard applications in one device. Figure 2 represents the NFC card emulation mode. ISO 14443 is the standard used to establish the communication between NFC chip with the emulated card and the contactless reader.

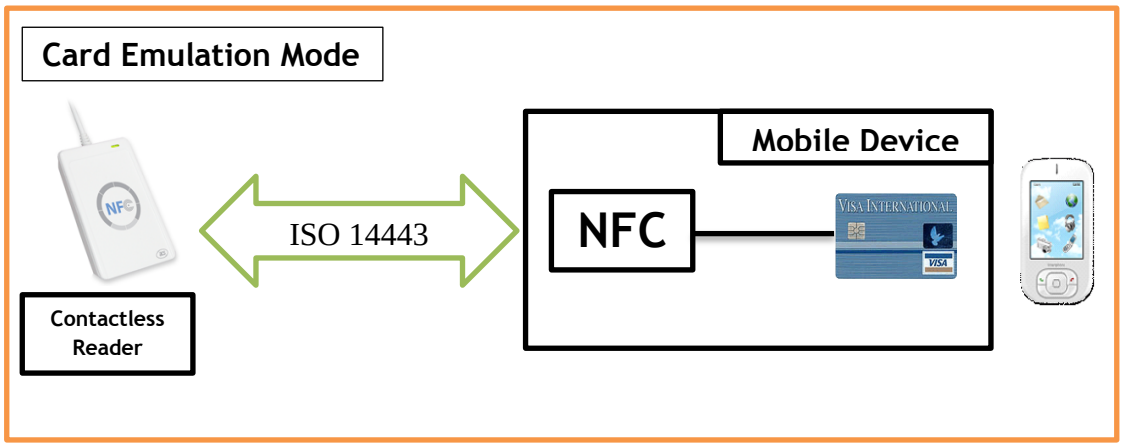


Figure 2 - Card Emulation mode.

NFC peer-to-peer mode (ISO 18092) allows two NFC enabled devices to establish a bidirectional connection to exchange contacts, Bluetooth pairing information or any other kind of data. Cumbersome pairing processes are a thing of the past thanks to NFC technology. To establish a connection, a client (NFC peer-to-peer initiator) is searching for a host (NFC peer-to-peer target) to setup a connection. Then the NDEF (NFC Data Exchange Format) is used to transmit the data. Figure 3 is a Peer-to-Peer NFC mode diagram. It is shown a communication between two NFC enabled devices using ISO 18092.

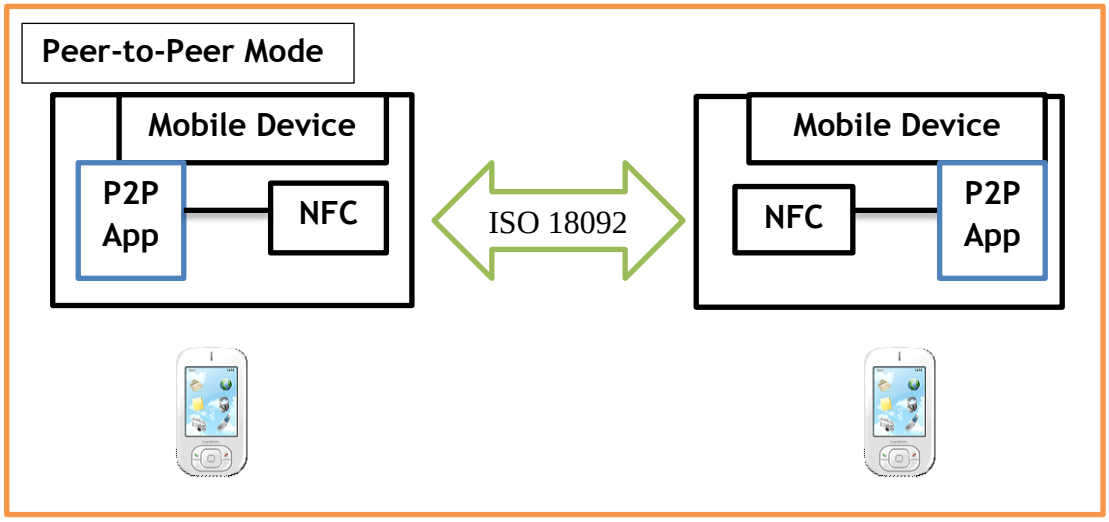


Figure 3 - Peer-to-Peer mode.

2.3. Communication Protocols

2.3.1. NFC Logical Link Control Protocol (LLCP)

Defined at OSI layer-2 protocol to support peer-to-peer communication between two NFC-enabled devices, which is essential for any NFC applications that involve bi-directional communications [35]. The specification defines two service types, connectionless and connection-oriented, organized into three link service classes: connectionless service only, connection-oriented [36] service only, and both connectionless [37] and connection-oriented service [38]. The connectionless service offers minimal setup with no reliability or flow-control guarantees (deferring these issues to applications and to the reliability guarantees offered by ISO/IEC 18092 and ISO/IEC 14443 MAC layers [39]). The connection-oriented service adds in-order, reliable delivery, flow-control, and session-based service layer multiplexing.

LLCP is a compact protocol, based on the industry standard IEEE 802.2 [40], designed to support either small applications with limited data transport requirements, such as minor file transfers, or network protocols, such as OBEX and TCP/IP, which in turn provide a more robust service environment for applications. The NFC LLCP thus delivers a solid foundation for peer-to-peer applications, enhancing the basic functionality offered by ISO/IEC 18092 [41], but without impacting the interoperability of legacy NFC applications or chipsets.

2.3.2. NFC Data Exchange Format (NDEF)

NFC Data Exchange Format (NDEF) [42] specification defines a data format to exchange information between two NFC enabled devices [43]. NDEF is a lightweight, binary message format that can be used to encapsulate one or more application-defined payloads of arbitrary type and size into a single message construct. Each payload is described by a type, a length, and an optional identifier. Type identifiers may be URIs, MIME media types [44], or NFC-specific types.

An example of using of NDEF is when two NFC Forum Devices are in proximity, an NDEF message is exchanged over the NFC Forum LLCP protocol. When an NFC Forum Device is in proximity of an NFC Forum Tag, an NDEF message is retrieved from the NFC Forum Tag by means of the NFC Forum Tag protocols [45]. Figure 4 shows a NDEF message with more than one record. Each message can contain as many records as the available space.

NDEF MESSAGE		
Email	**Picture**	**Picture2**

Figure 4 - NDEF email message with two attaches.

Figure 5 is another example of a NDEF message. This message has a smartposter and an application inside it.

NDEF Message				
Smartposter				application/vCard
URI	Text	Action	Configuration	vCard data

Figure 5 - NDEF message with data.

2.4. Record Type Definition (RTD)

RTD (Record Type Definition) [46, 47] are specifications by NFC Forum to write on NFC tags. "Record type names are used by NDEF [48] applications to identify the semantics and structure of the record content". NFC Forum suggests use of their RTDs and NDEF messages for simplify use of NFC Communications.

2.4.1. NFC Forum Record Type Definitions

All RTDs [49] should be MIME media types, URIs, NFC Forum external types or NFC types names [50]. This RTDs can be defined by NFC Forum or other entities, but should be used NFC Forum RTDs. The RTDs can be used when content to be written on tag is not MIME type, URI or when the message is bigger than tag memory size.

RTDs [51] recognized by NFC Forum are identified on NDEF header message inside TNF. TNF size is 3 bits, the values of this field are:

• 0x00 – "Empty", used when needs to close one message

• 0x01 – "NFC Forum well-known type", used when the message is one NFC Forum RTD

• 0x02 – "Media type as defined RFC 2046", used when tag content is one media type defined on specification RFC 2046

• 0x03 – “Absolute URI as defined in RFC 3986”, used when content is one URI specified on RFC 3986

• 0x04 – “NFC Forum external type”, identified tag content is defined by other entity

• 0x05 – “Unknown”, used when content is not recognized by any entity

• 0x06 – “Unchanged”, used when content is fragmented

• 0x07 – “Reserved”, not used

2.4.2. NFC Forum well-known type (0x01)

These types are recognized by NFC Forum and are used when tag content is not MIME type and message is larger than tag memory size. These types are URN (Uniform Resource Name) defined on specification RFC 2141 with namespace identifier (NID) “nfc” and is used “wkt” prefix, but the content should be written on tag as relative URI (specification 3896), omitted NID and prefix. For example the well-known type “urn:nfc:wkt:sample” should be written as “sample”

There are two types of NFC Forum well-known types: NFC Forum Global type and NFC Forum Local type. These types and external type will be described on next chapter.

Text

This RTD is used when tag content a text. It can be written on same tag in many languages. The NFC RTD for text is character “T”. The text can be encoded on UTF-8 or UTF-16, the MIME type used must be “text/xhtml;format=fixed”. Line breaks should represent with CRLF format, white spaces and new lines must be collapsed. If text describes one element, this element must be previously than description. Language codes must be use specification RFC 3066. Table 4 shows the Text RTD structure. It is shown all the bytes, their size and the content.

Table 4 - RTD Text Structure.

Offset (bytes)	Size (bytes)	Content
0	1	Status Byte
1	n	Language Code (encoding US-ASCII)
n+1	m	Text with encoding UTF-8 or UTF-16

Table 5 shows the status byte of a RTD text type.

Table 5 - Status Byte.

Bit number ...	Size (bytes)
7	0 – Text encoded with UTF-8 1 – Text encoded with UTF-16
6	RFU
5..., 0	Length of IANA language code

Length of IANA language code is calculated by:

Equation 1

m = length of payload – length of language code - 1

URI

Used when the tag content is an URI, the NFC RTD is character “U”, this RTD is useful when is needed shortly the URI. URI has two groups of bytes: the byte 0 is the URI identifier code and the other bytes are the URI’s content. Table 6 shows all URI RTD identifier codes. Each code represents an http protocol. For example if the user needs to write a tag with http internet protocol he only needs to put the code 0x03 in the tag’s prefix.

Table 6 - URI RTD identifier codes.

Decimal	Hex	Protocol
0	0x00	N/A prepending is done, and the URI field contains the unabridged URI
1	0x01	http://www.
2	0x02	https://www.
3	0x03	http://
4	0x04	https://
5	0x05	tel:
6	0x06	mailto:
7	0x07	ftp://anonymous:anonymous@
8	0x08	ftp://ftp.
9	0x09	ftps://
10	0x0A	sftp://
11	0x0B	smb://
12	0x0C	nfs://
13	0x0D	ftp://
14	0x0E	dav://
15	0x0F	news:
16	0x10	telnet://
17	0x11	imap:
18	0x12	rtsp://
19	0x13	urn:
20	0x14	pop:
21	0x15	sip:
22	0x16	sips:
23	0x17	tftp:

24	0x18	btspp://
25	0x19	btl2cap://
26	0x1A	btgoep://
27	0x1B	tcpobex://
28	0x1C	irdaobex://
29	0x1D	file://
30	0x1E	urn:epc:id:
31	0x1F	urn:epc:tag:
32	0x20	urn:epc:pat:
33	0x21	urn:epc:raw:
34	0x22	urn:epc:
35	0x23	urn:nfc:
36...255	0x24…0xFF	Reserved for Future Use

Smartposter

This RTD is used for show more information in a NFC tag. The NFC RTD character is “Sp”. This RTD uses other RTDs, like text RTD, URI RTD. Smartposter [52, 53] has a title, a URI, a recommended action, an icon and the size.

- Title – should be use RTD text, and is impossible to have two or more titles with same language.
- URI – should be use URI RTD and must be exist only one URI per tag
- Recommended action – is one local type, and is the suggest action to device execute after read tag, NFC Local Type character is “act”
- Icon – this field can contain two things, one image with MIME type “image/jpeg”, “image/png” or other, or one video with MIME type “video/mpeg” or other. The reader device chooses what icon show.
- Size – this field is the size of the content, is used to device choose make download or not

Generic Control RTD

Used when is needed to execute anything specific on reader device. The NFC RTD character is “Gc”. It is designed with a philosophy to allow:

- Access to functions or applications which are not covered by MIME type records or other NFC Forum Global Type records.
- The source device to explicitly indicate a certain function or certain application on the destination device to perform a certain action.

A MIME type record in a NDEF message provides indirect access to an associated function or application. When there are multiple functionalities and/or applications on a destination device and some of them share the same MIME type, only one of them is chosen for processing data contained in the record. Association between data and function/application is determined by the destination device. This may lead to a different result than the source device expects.

Other NFC Forum Global Type records implicitly assume a dedicated application. This does not resolve all the limitations of MIME type records because only a limited number of popular types are defined as NFC Forum Global Types.

To resolve the issue above, a Generic Control [54] is capable of requesting any application on the destination device. It allows the Generic Control to explicitly specify a certain function or a certain application to be accessed.

This RTD uses one NDEF message with several NDEF sub-records. Each sub-record with a different RTD.

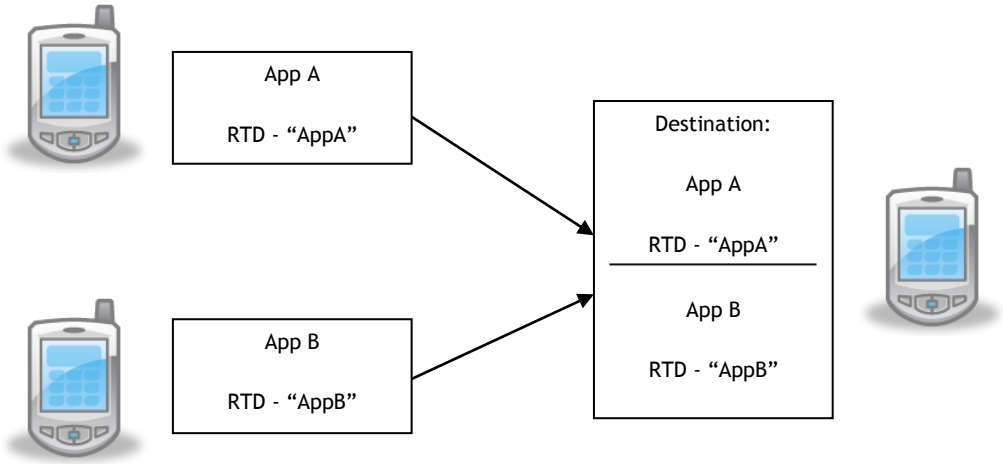


Figure 6 - Generic Control RTD mechanism.

The structure of a Generic Control record is shown in Table 7.

Table 7 - Structure of a Generic Control Record.

"Gc"	Config Byte	"t"	Target Identifier	"a"	Action Flag Byte	Action Identifier	"d"	Data
		Type Name	Payload	Type Name	Payload		Type Name	Payload
Type Name	Payload							

2.4.3. NFC External Type (0x04)

Used when other entities need to used RTDs proprietaries, is equal to URN [55], with NID “nfc” but with NSS “ext”, should be have the name of company and then the content. Example: “urn:nfc:ext:sample.com:this is a sample”. Like the others “urn:nfc:ext:” must be omitted.

2.5. NFC Tags

There are many NFC tag [56] formats with different shapes and memory sizes. Tags can be divided into 4 types and can be written and read. In Figure 7 a lot of NFC tags are shown. Each tag has unique, has different features and are used for a wide of different objectives.

Figure 7 - NFC Tags.

2.5.1. Type 1 tag

Type 1 is based on ISO 14443 A and is currently available exclusively from Innovision Research & Technology (Topaz™). It has a 96-byte memory capacity, which makes it a very cost-efficient tag for a wide range of NFC applications. The NFC Forum Type 1 tag utilizes a simple memory model. There shall be two memory model mappings depending on the memory size of the tag:

1. Static memory structure applies for a tag with physical memory size equal to 120 bytes,

2. Dynamic memory model applies for a tag with physical memory size larger than 120 bytes.

The memory shall be considered as being divided into blocks containing 8 bytes each. Each block is numbered from 0 to 15 (Eh) for static memory structure or from 0 to k for dynamic memory structure. The number associated to a block is called the ‗block number'. The 8 bytes inside each block are numbered from 0 to 7, where byte 0 is the Lost Significant Byte (LSB) and byte 7 is the Most Significant Byte (MSB) of the block. For the complete tag address space then, byte 0 of block 0 corresponds to ByteAddr = 0 as the LSB. Byte 7 of block Eh for static memory structure or byte 7 of block k for dynamic memory structure indicates the very MSB. Figure 8 shows the memory map of a type 1 NFC tag, all

EEPROM Memory map										
Type	**Block no.**	**byte-0 (LSB)**	**Byte-1**	**Byte-2**	**Byte-3**	**Byte-4**	**Byte-5**	**Byte-6**	**Byte-7**	**Lockable**
UID	0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	UID-6		Locked
Data	1	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Yes
Data	2	data 8	data 9	data 10	data 11	data 12	data 13	data 14	data 15	Yes
Data	3	Data 16	Data 17	Data 18	Data 19	Data 20	Data 21	Data 22	Data 23	Yes
Data	4	Data 24	Data 25	Data 26	Data 27	Data 28	Data 29	Data 30	Data 31	Yes
Data	5	Data 32	Data 33	Data 34	Data 35	Data 36	Data 37	Data 38	Data 39	Yes
Data	6	Data 40	Data 41	Data 42	Data 43	Data 44	Data 45	Data 46	Data 47	Yes
Data	7	Data 48	Data 49	Data 50	Data 51	Data 52	Data 53	Data 54	Data 55	Yes
Data	8	Data 56	Data 57	Data 58	Data 59	Data 60	Data 61	Data 62	Data 63	Yes
Data	9	Data 64	Data 65	Data 66	Data 67	Data 68	Data 69	Data 70	Data 71	Yes
Data	A	Data 72	Data 73	Data 74	Data 75	Data 76	Data 77	Data 78	Data 79	Yes
Data	B	Data 80	Data 81	Data 82	Data 83	Data 84	Data 85	Data 86	Data 87	Yes
Data	C	Data 88	Data 89	Data 90	Data 91	Data 92	Data 93	Data 94	Data 95	Yes
Reserved	D									
Lock / Reserved	E	LOCK-0	LOCK-1	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	

Figure 8 - EEPROM memory map.

UID Format

Block 0 is reserved for the read-only Unique Identification (UID) number. Byte 7 is reserved for future use. Byte 6 is the manufacturer‘s identification code. Bytes 5, 4, 3, 2, 1, 0 are the unique number.

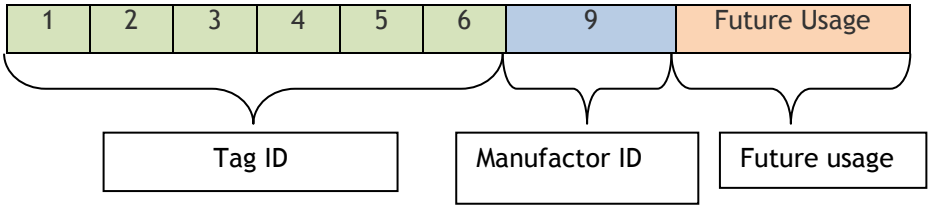


Figure 9 - UID format.

Main Read / Write Memory Format

The 12 blocks numbered as 1h to Ch, contain the 96 bytes of general read/write memory. Each block is individually lockable to become read-only by use of the relevant bits within the lock control bytes.

Block D

The block numbered as D is read-only and reserved for internal use.

Lock Control / Status Bytes

Bytes 0 & 1 of block Eh function as the lock controls for the various memory blocks.

2.5.2. Type 2 tag

A type 2 tag platform is based on a particular memory chip with a certain memory size and space for data. The following sections describe the details of such memory chip and in particular its memory structure and management.

Static Memory Structure

This memory structure is used by Type 2 tag platform with a physical memory size equal to 64 bytes. Figure 10 shows the memory layout of such tag. It is composed of different fields: UID (Unique identifier), Internal is reserved bytes for manufacturing usage, Lock (static lock bytes to switch the tag from READ/WRITE state to READ-ONLY state) CC (Capability Container bytes) and Data (bytes used to store information).

Static Memory Structure					
Byte number	**0**	**1**	**2**	**3**	**Block**
UID / Internal	UID 0	UID 1	UID 2	Internal 0	0
Serial Number	UID 3	UID 4	UID 5	UID 6	1
Internal / Lock	Internal 1	Internal 2	Lock 0	Lock 1	2
CC	CC 0	CC 1	CC 2	CC 3	3
Data	Data 0	Data 1	Data 2	Data 3	4
Data	Data 4	Data 5	Data 6	Data 7	5
Data	Data 8	Data 9	Data 10	Data 11	6
Data					.
Data					.
Data					.
Data					.
Data					.
Data					15

Figure 10 - Static memory structure.

The 7 bytes unique identifier (UID0-6, see Figure 10) are contained in bytes 1-3 of block 0, and the 4 bytes of block 1. The UID bytes are write-protected after having been programmed by the IC manufacturer. Figure 11 shows the UID coding of a byte.

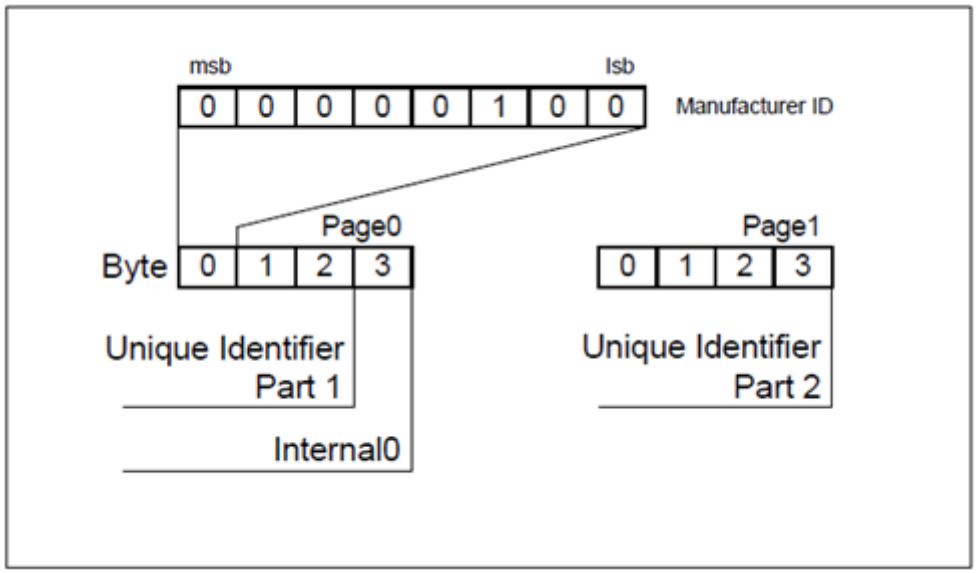


Figure 11 - UID Coding.

Internal Bytes

These bytes are reserved for manufacturing use. The NFC Forum device shall not use them to store information data.

Static Lock Bytes

The bits of byte 2 and 3 of block 2 represent the field-programmable read-only locking mechanism called static lock bytes. Depending on the value of the bits of the static lock bytes two configurations are possible:

• All bits are set to 0b, the CC area and the data area of the tag can be read and written.

• All bits are set to 1b, the CC area and the data area of the tag can be only read.

The locking bits are set to 1b via a standard write command to block 2. To set all static lock bits to 1b, the NFC Forum device SHALL set bytes D2 and D3 of the WRITE command to FFh and set the remaining two bytes D0 and D1 to any value.

This process is irreversible: if one bit of the lock bytes is set to 1b, it cannot be changed back to 0b.

Capability Container

The Capability Container (CC) manages the information of the Type 2 tag platform. The four bytes of block 3 contain the so called CC.

Data Area for Static Memory Structure

Block 4 to 15 is the available data area for information storage. The NFC Forum device SHALL write the data area consecutively in order starting from byte 0 of block 4 up to byte 3 of block 15. For static memory structure the data area size is equal to 48 bytes.

2.5.3. Type 3 tag

Blocks

The basic unit of information used in memory management is called a block. Each block has a fixed size of 16 bytes. The number of memory blocks available depends on the chip hardware. Memory blocks are not addressed directly but relative to the Service they belong to.

System Information

Each Type 3 Tag contains management data, called System Information. The system information of a Type 3 Tag consists of the following parts: Manufacturer ID Information, System Definition Information, and Service Definition Information. Manufacturer ID Information and System Definition Information are pre-assigned by the Type 3 Tag manufacturer.

Manufacturer ID Information

Manufacturer ID, shown in Figure 12 Information consists of the Manufacturer ID (IDm) and the Manufacturer Parameter (PMm). The Manufacturer ID Information cannot be deleted or re-written by users.

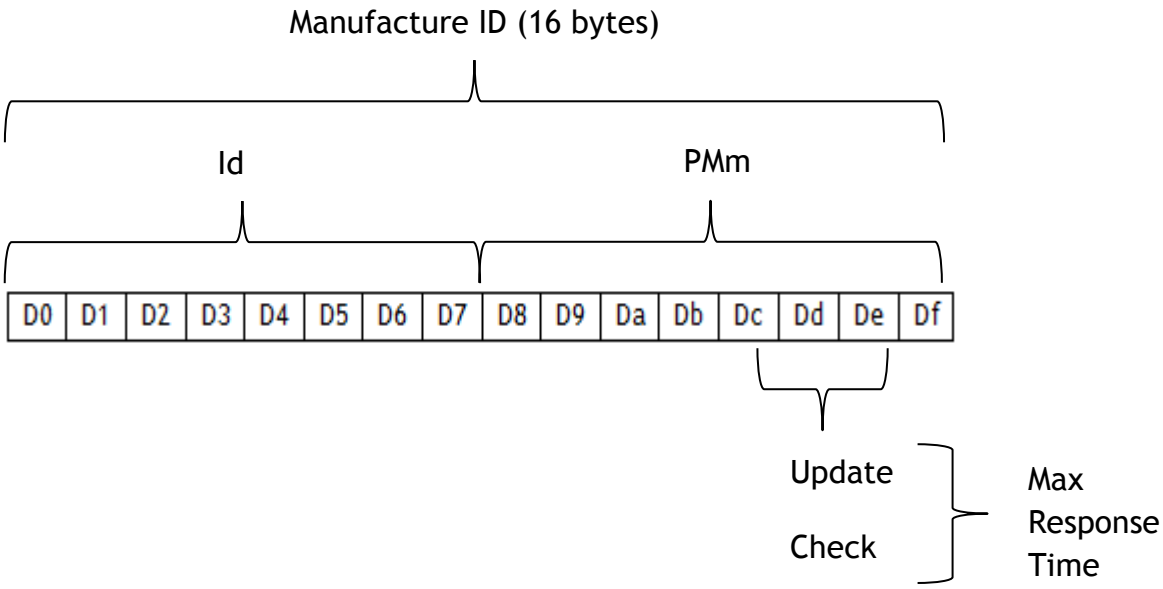


Figure 12 - Manufacturer ID.

The Manufacturer Parameter (PMm) contains the Maximum Response Time parameter. The Maximum Response Time Parameter is used to inform the NFC Forum Device about the maximum time needed by the Type 3 Tag to execute Check and Update commands. It has a length of 6 bytes. The NFC Forum Device shall interpret the 4th byte (Dd) to contain the maximum response time (Figure 13) information for the Check command and the 5th byte (De) to contain the maximum response time information for the Update command. The byte format for the 4th and 5th byte shall be interpreted as shown in the following diagram:

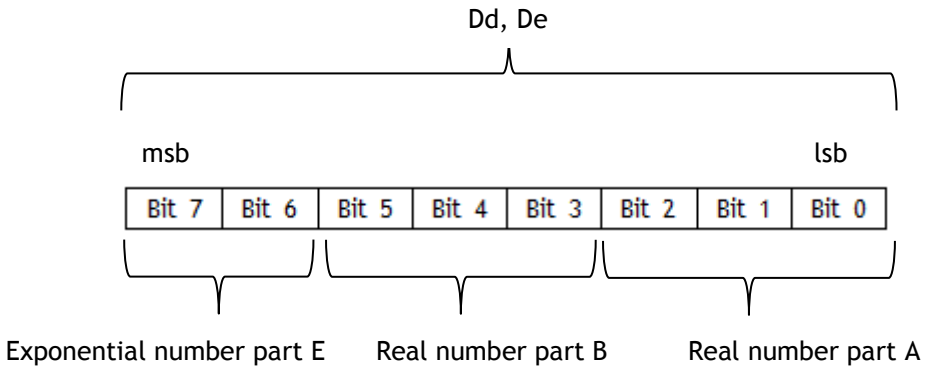


Figure 13 - Maximum Response Time Parameter.

The Maximum Response Time is defined by the chip manufacturer considering the maximum transaction time the Type 3 Tag needs to process the command. The response time, which is calculated using this parameter, is longer than the actual time needed for the processing of the corresponding command. The following formula SHALL be used for the calculation of the maximum response time of the Check and Update commands:

Maximum Response Time = $T.[(B+1).n+(A+1)].4^E$

T: 0.302 ms

A: Real number part A

B: Real number part B

E: Exponential number part E

N: Number of blocks used in the Check/Update command

System Definition Information

The System Definition Information consists of the System Code. The System Code is a 2 bytes number. NFC Forum Devices can use the

corresponding parameter in the Polling command to poll targets having a specific system code. The System Code is coded in big endian order.

Service Definition Information

A Service Definition Information is present for each Service existing on a Type 3 Tag. It consists of the Service Code and the Number of Blocks for the Service. The Service Code is uniquely identifying the Service on a Type 3 Tag. It has a length of 2 bytes. The format is little endian. The Service Code consists of a Service Number and an Access Attribute. The Service Number has a length of 10 bits (the 10 most significant bits of the 2 bytes) and is unique for each Service of a Type 3 Tag. The Access Attribute has a length of 6 bits (the 6 least significant bytes) and specifies the permissions for accessing the associated memory blocks. The Number of Blocks is a 2 byte number specifying the number of memory blocks associated with this Service. Each Service Definition Information usually references a number of memory blocks that are exclusively used by this Service. The only exceptions are Overlap Services. Overlap Services share the same memory blocks but have different Access Attributes (e.g. read only, read/write).

2.5.4. Type 4 tag

The Type 4 Tag Platform contains at least the NDEF Tag Application. The NDEF Tag Application contains the NDEF messages on a Type 4 Tag Platform that provides a file system composed of at least two EF files (see [ISO/IEC_7816-4]): the Capability Container file (CC file), and the NDEF file (NDEF file). Concerning the EF files, the byte with offset value equal to zero is the Most Significant Byte (MSB) and the byte with the highest offset value is the Lost Significant Byte (LSB). As defined by this document, if not

otherwise specified, the bit and byte ordering when defining packets and messages follows the big-endian byte order.

2.6. Secure element

Secure element [57, 58] is a secure storage in NFC device capable to store card applications and information that require a high level of security. The Secure Element could be integrated in various form factors: SIM Cards, chip embedded in the handset or SD Card.

2.6.1. Secure element management

In an environment where multiple types of Secure Elements [59] exist, as well as multiple bearers based on handset and SIM capability (BIP, SMS, midlet proxy, point of sale terminals), it is important to have information about what capabilities are valid for the current subscriber. In order to select the best bearer for an OTA [60] download, the SIM OTA [61] platform needs to have this information available.

A Delegated Management (DM) token (delegated OTA key) is issued for the NFC service provider or a TSM executing the NFC application management on the SIM card. The mobile network operator's SIM OTA platform will have the capability to issue these tokens.

In practice this means that the mobile network operator operates the SIM OTA platform. They then provide a remote OTA module via which a third party is able to control a leased area (their SSD) on the SIM. The remote OTA module is deployed either at the NFC service provider's premises, the Trusted Service Manager (TSM) or at a separate, secured, location within the mobile operator. The communication of the remote OTA module to the SIM goes via the mobile network operator's SIM OTA platform.

2.6.2. Secure element types

Figure 14 shows all NFC secure element types, the most usual is the simcards controlled by network operators. To access them it is need to have the javacard keys. Last month Google launch its new service Google Wallet that use the SDCard as secure element. In the future Google will give access to embedded chip to developers. SDCard is the simplest way to use a secure element with NFC because there is not any type of access key.

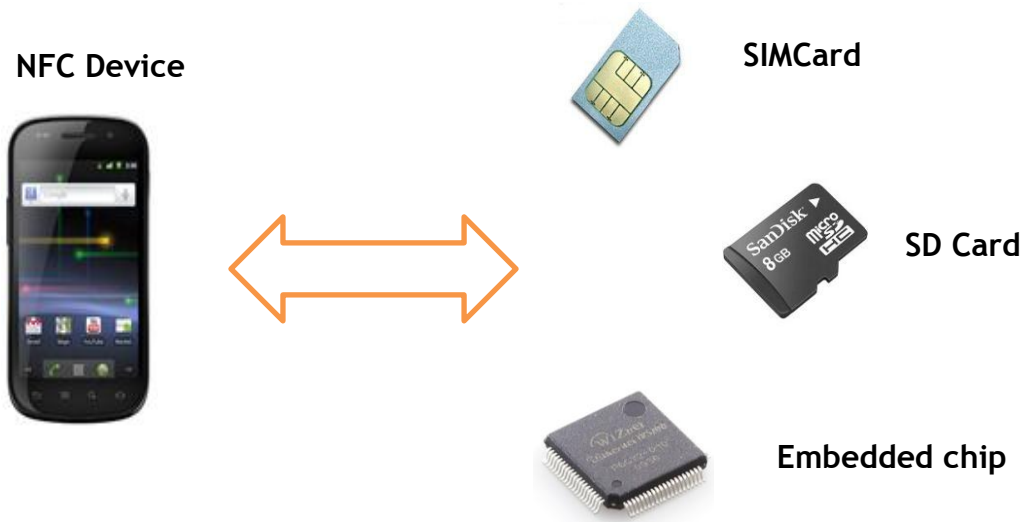


Figure 14 - Secure Element types.

2.6.3. Embedded Chip

The secure element is soldered on the PCB of the phone and cannot be removed without special equipment. SmartMX is a microprocessor based card with a dedicated operating system to fast cryptographic computations. This card is capable of executing complex operations that are as secure and fast as operations on contact based cards. These cards are capable of supporting a range of both proprietary and open operating systems (Java Card). The hardware of the SmartMX is Common Criteria certified at EAL5+ by the “Bundesamt für Sicherheit in der Informationstechnik”, BSI, which

means that it is highly resistant to tampering such as, for instance, reverse engineering attacks, fault/glitch attacks, or power analysis attacks.

To meet the needs of the differing handset manufacturers, the PN544 has been designed to support the three main architectures which are used to secure NFC transactions, including the Secure Element within the Universal Integrated Circuit Card (UICC or SIM), within the SD card and within the mobile handset (embedded Secure Element: PN544 plus Smart MX security chip in a pin to pin compliant solution).

2.6.4. Secure Memory Card

Tyfone solution named SideTap as known as u4ia (pronounced 'euphoria'), is the world's first standard MicroSD card to fully integrate onboard controller that manages over-the-air (OTA) access of a secure element, NXP's SmartMX SmartCard controller secure element, miniature 13.56MHz contactless antenna connected to the secure element, and still includes space for virtually any memory capacity for consumers to store documents, photos, videos, music or other files.

All secure information is stored inside the smartcard chip and not in the memory of the memory card.

Tyfone's u4ia (pronounced 'euphoria') has the potential to break the current handset availability problem by allowing any device with an SD memory card slot to have near field communication functionality added to it just by inserting a card.

The device also offers the potential for financial service providers — and others — to gain their own 'real estate' on a handset without needing to deal with a customer's mobile network operator (MNO).

The CardWizard Perso-to-Go platform personalizes — at the branch level or back office — cardholder information onto the chip embedded in Tyfone's patented MicroSD secure memory card. Tyfone's MicroSD card can then operate in any standard memory card slot within mobile phones, across all operating systems. No over-the-air (OTA) personalization is

required, and CardWizard Perso-to-Go provides instant activation capabilities so the device can be immediately used for contactless point-of-sale purchases, including open-loop channels or closed-loop retail applications.

The CardWizard Perso-to-Go platform can securely personalize any NFC payment form factor at the branch level. This includes embedded NFC chips in mobile phones, various contactless stickers, key fobs, and MicroSD cards. The technology meets worldwide security requirements for both EMV and MSD contactless specifications and complies fully with all Visa® and MasterCard® security recommendations.

2.6.5. Subscriber Identity Module

The MNO keeps the overall responsibility and the management is made via OTA platform. SIM card, which already plays a key role on handsets by identifying the subscriber and related account, should be the SE of choice for mobile payment. At first, this makes sense as from the technical perspective, the SIM card is very secure, and it has a secure channel between the SIM card and the NFC chip over Single Wire Protocol (SWP)

3. Requirement Analysis

In this chapter is shown the requirements to build a real demonstration of NFC usage. It will be presented two android applications: a p2p based NFC application called credit transfer and a typical NFC application with a NFC chipset, in this case, the smartphone Nexus S and NFC tags called NFC Launcher. It was chosen the android operating system because when the demo was built, was the only OS with a smartphone in the market. Before shows applications features and functionalities it is need to describe both the applications.

3.1. Credit Transfer

Credit Transfer is a peer-to-peer (P2P) NFC based application. The main goal of this application is to transfer money between two mobile phone simcards. This application will allow mobile operators change the usual call center calls or auto attendant to a nice look mobile phone application. The service already exists in most of the network operators over the world but this is a way to beautify and simplify the credit change service. The

solution was applied to this type of services but it is possible to apply it to another.

3.1.1. Architecture

It is shown in Credit Transfer architecture (Figure 15) two devices with a NFC chip and a Bluetooth adapter [62, 63]. Both transmit a message between devices, but each message has a specifically function. First message (NFC chipset) is to notify the receiver device to turn on the Bluetooth adapter and the second message (Bluetooth adapter) is to send all other information of the credit transfer. This application is based on a client-client system, it is not necessary to have a server side. The money sender communicates to his network operator the value that he want to send and after receive a positive answer it will notify the receiver.

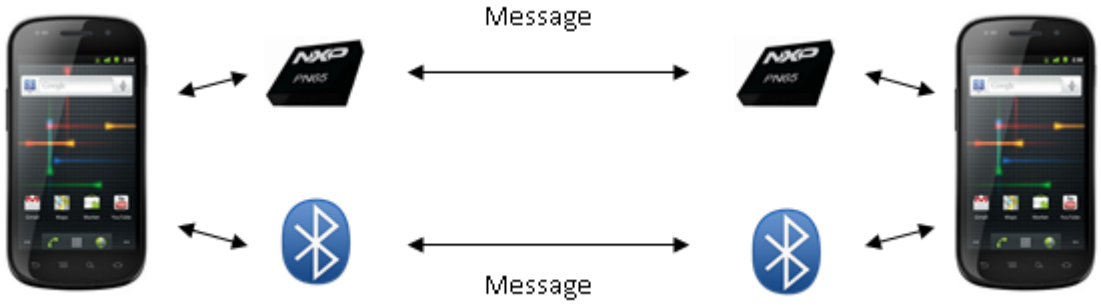


Figure 15 - Credit Transfer Architecture.

3.1.2. Application Use Cases

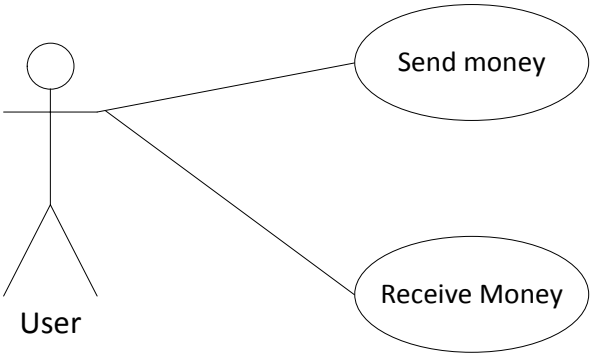


Figure 16 - Credit transfer application use cases.

3.1.3. Android Application

This application only was built to android operating system, because as already said it was the only with a NFC device in the market. Today there are other operating systems where it is possible to develop a similar solution. At the moment Symbian Anna, Bada 2.0 and blackberry 7.0 also have solutions to develop a Credit Transfer solution. Figure 17 shows the main screen of Credit Transfer. Each step of Credit Transfer functionalities will explain in the next chapter, Demonstration of Credit transfer.

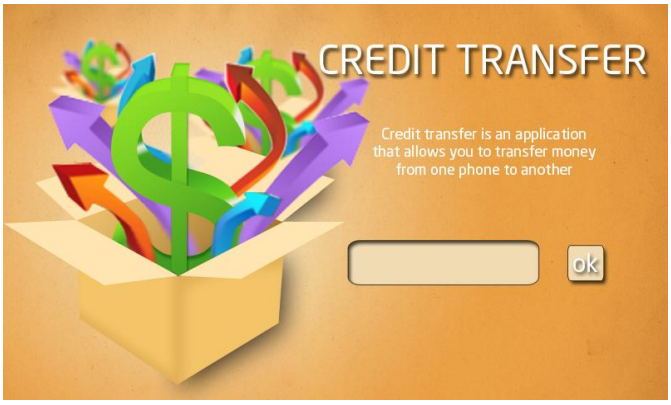


Figure 17 - Credit Transfer main screen.

3.2. NFC Launcher

NFC Launcher is another NFC application, but unlike Credit Transfer the NFC Launcher is simpler to develop. This is a common NFC application that uses a NFC device, in this case, an android smartphone and NFC tags. The main purpose of this application is to launch another already installed application in the phone, instead of launching it using the application icon. NFC launcher is a brilliant application because besides phone native application it has a webportal where is possible to configure NFC tags. In the portal it is possible to add, remove or edit as many applications as we want and put them in NFC tags. The guilty of this fantastic feature is a smartcode created in real time. The smartcode is unique for each application and this allow putting the same application in many NFC tags.

3.2.1. Architecture

Figure 18 shows the NFC Launcher Android application diagram. NFC Launcher needs a based NFC device to work. With NFC Launcher it is possible to read or write a smartcode. A smartcode is an intelligent code that identifies each application. It can be configured in NFC webportal that

works as an admin page for NFC Launcher. A smartcode contains any associated type of information. When the device read a smartcode it connects to the NFC Launcher server and receives the associated content in the response. All requests are made using an http connection, so the device needs to have a data account configured, WiFi or network operator data. Each smartcode is associated with an Android market application. If a code is written in a NFC tag the device will read it and launch the application. If the application is already installed it is launched, otherwise if it is not installed android market is launched and the application download begins. After download the application it is launched. The process is always the same. After launch an application the user can touch in the tag again, if the application still running it will resume, otherwise Android package manager launch it again.

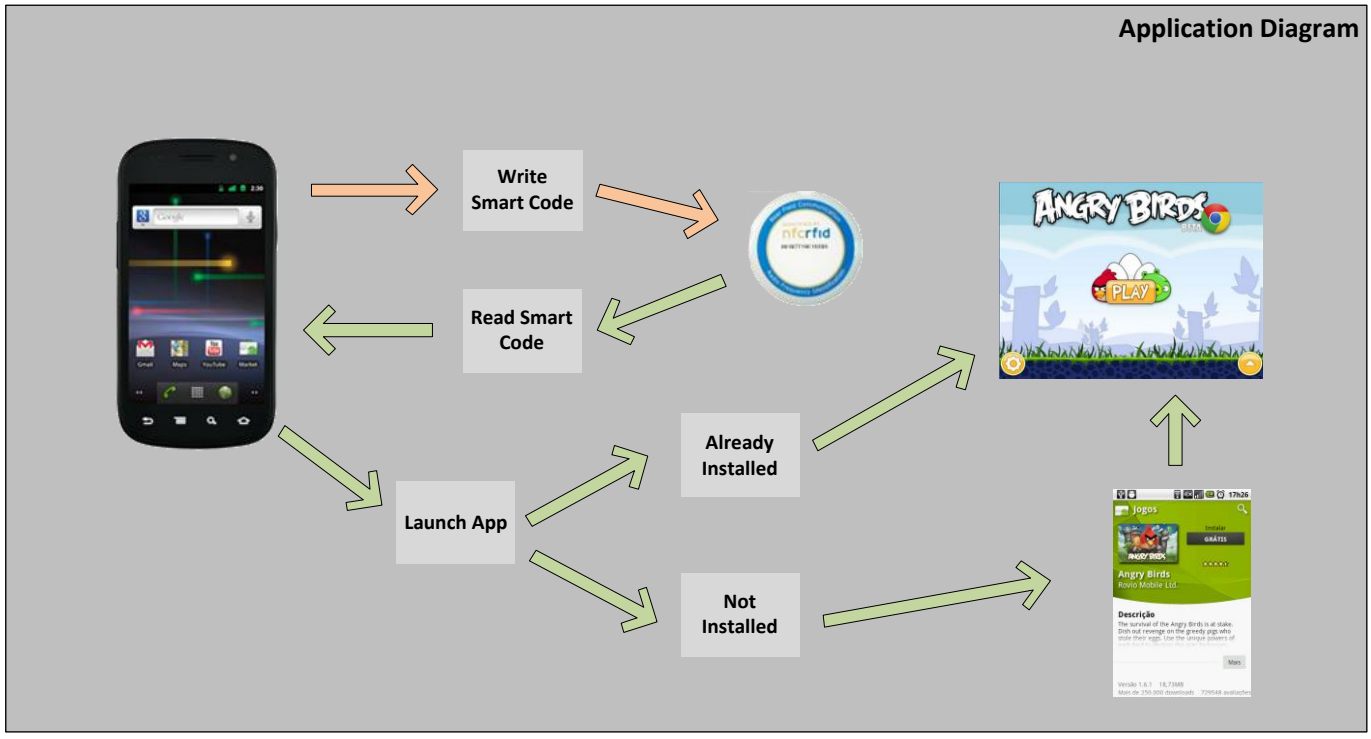
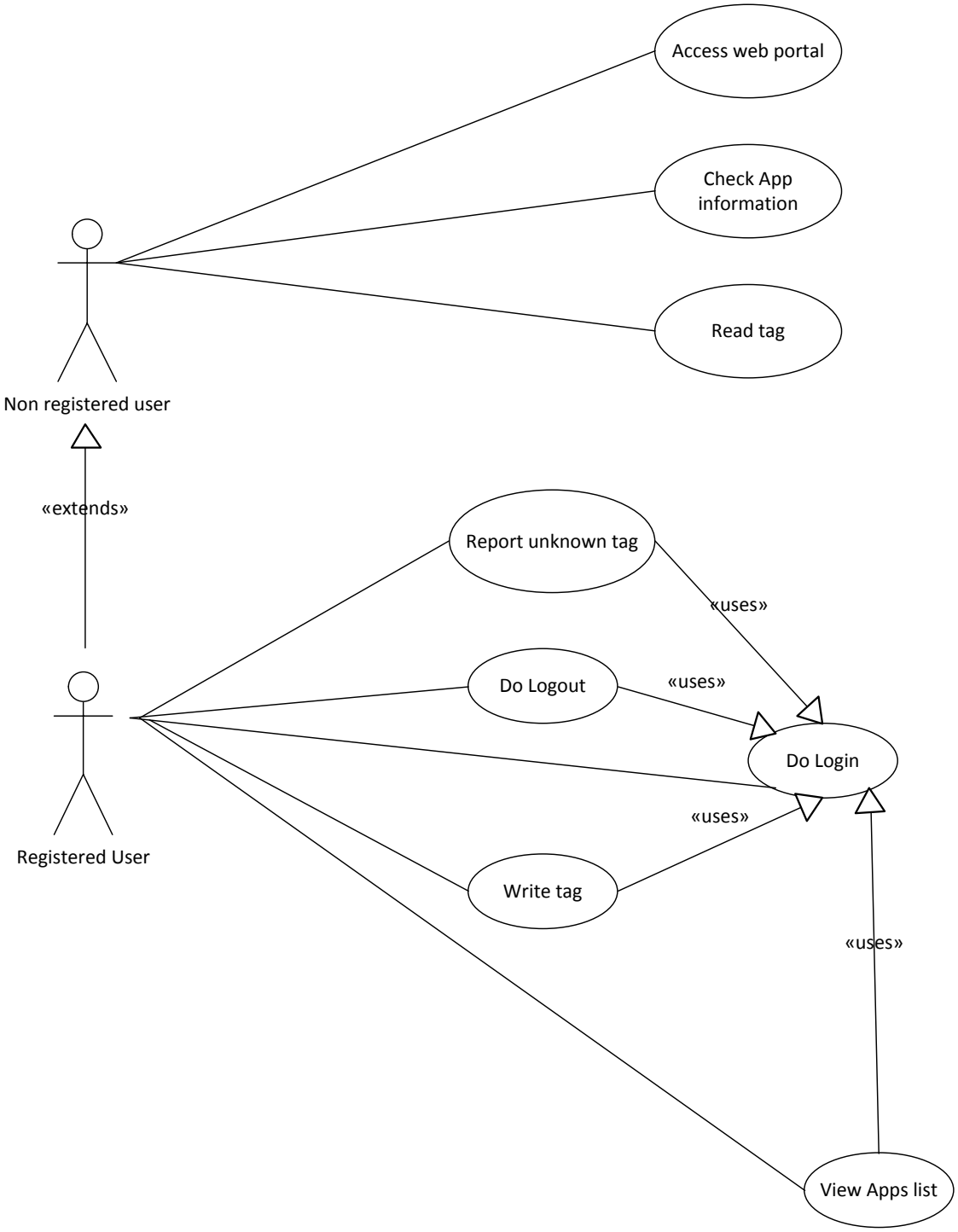
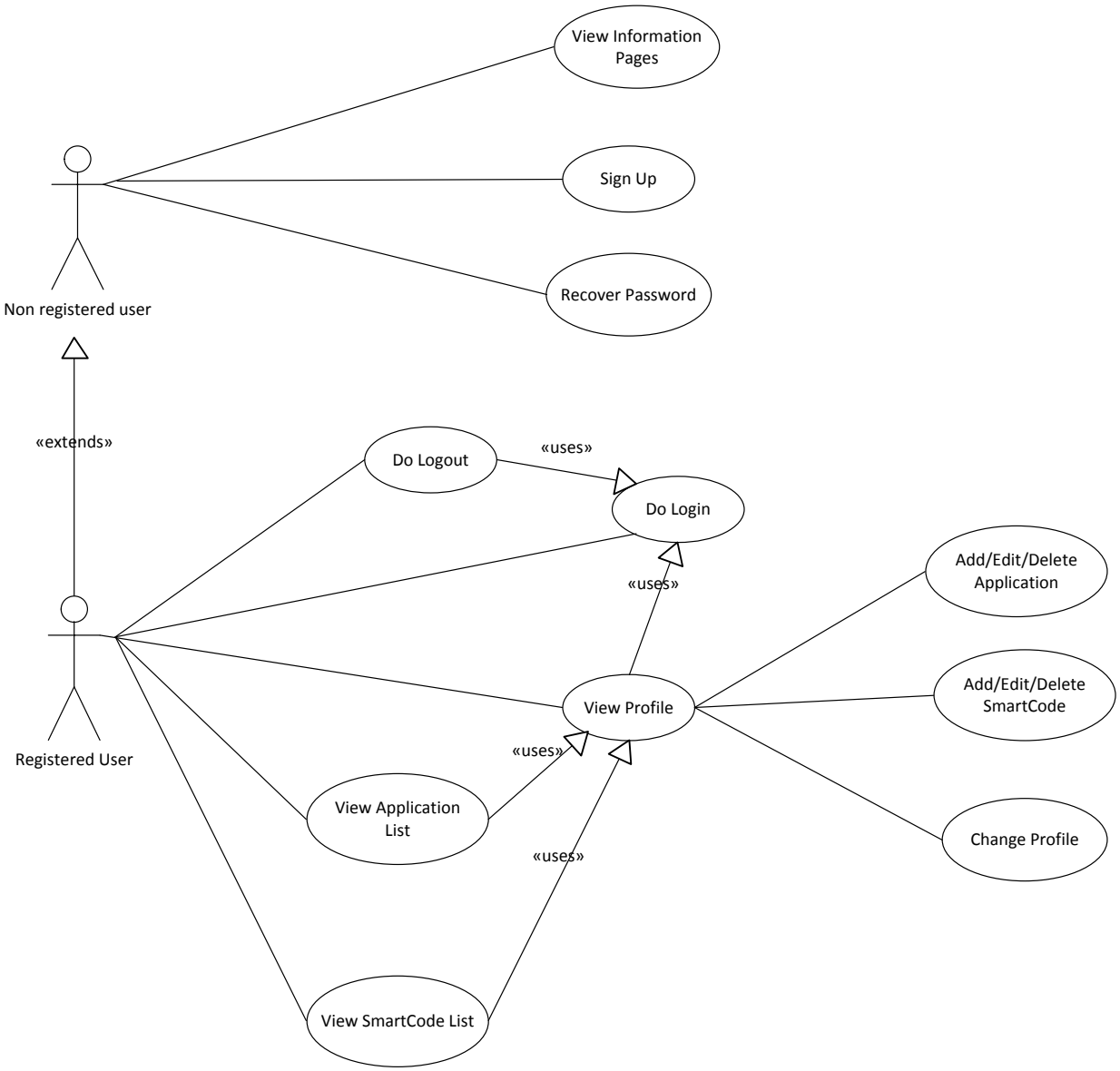


Figure 18 - NFC Launcher Android Application Architecture.

3.2.2. Android Application Use Cases



3.2.3. Portal Use Cases



3.3. Used Technologies

3.3.1. JAVA

Nowadays Java is one of the most used programming languages for mobile application development. Java is a generic language, platform independent and hardware independent too. Java applications are very portable for many operating systems. Inside mobile world development, this language is used for many companies: Google with its operating system Android, RIM from blackberry also uses Java for develop their applications. Google and Blackberry use Java as a native language to develop but there are other companies work in mobile development with Java. They work in a layer above, and they have their own IDEs to develop. Java is the main reason of this, because it is very adaptive.

Java has three essential elements: the programming language, the JVM (Java virtual Machine) and the APIs (Application Programming Interfaces) provided by other. There are two important steps to execute a Java application in a compatible device, compilation of code and after the interpretation to machine language. After that the virtual machine needs to launch the provided application.

If a device has a JVM the developer only need to build the application and send it to the JVM, because this virtual machine can launch any Java application.

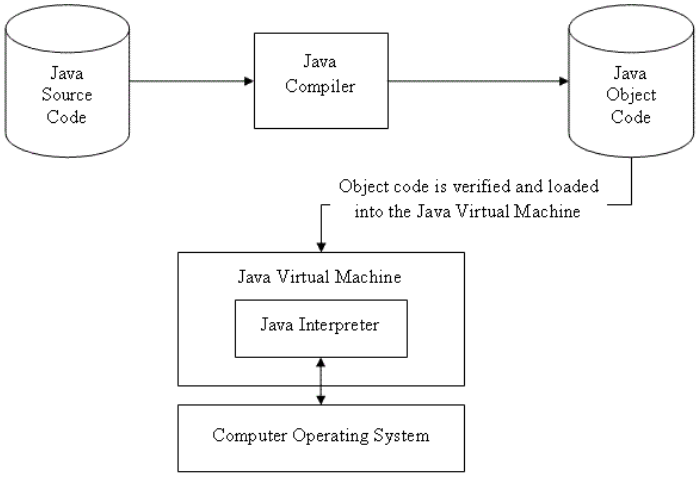


Figure 19 - Java development architecture.

3.3.2. Android

Android is Google mobile operating for mobile devices system such smartphones and tablets. It is launched in 2007 and it steal market share from the other competitors, like Apple and Microsoft. Android is based on Linux Kernel, and it is open source software, this is very useful for developers. At this moment android has two versions: Gingerbread (latest version: 2.3.5) for smartphones and Honeycomb (latest version: 3.2) for tablet computers. In a near future Android will receive an update to Ice cream Sandwich (4.0) that will work in both type of devices: smartphones and tablets. This is a good for developers because they only need to develop for only one platform with the same features.

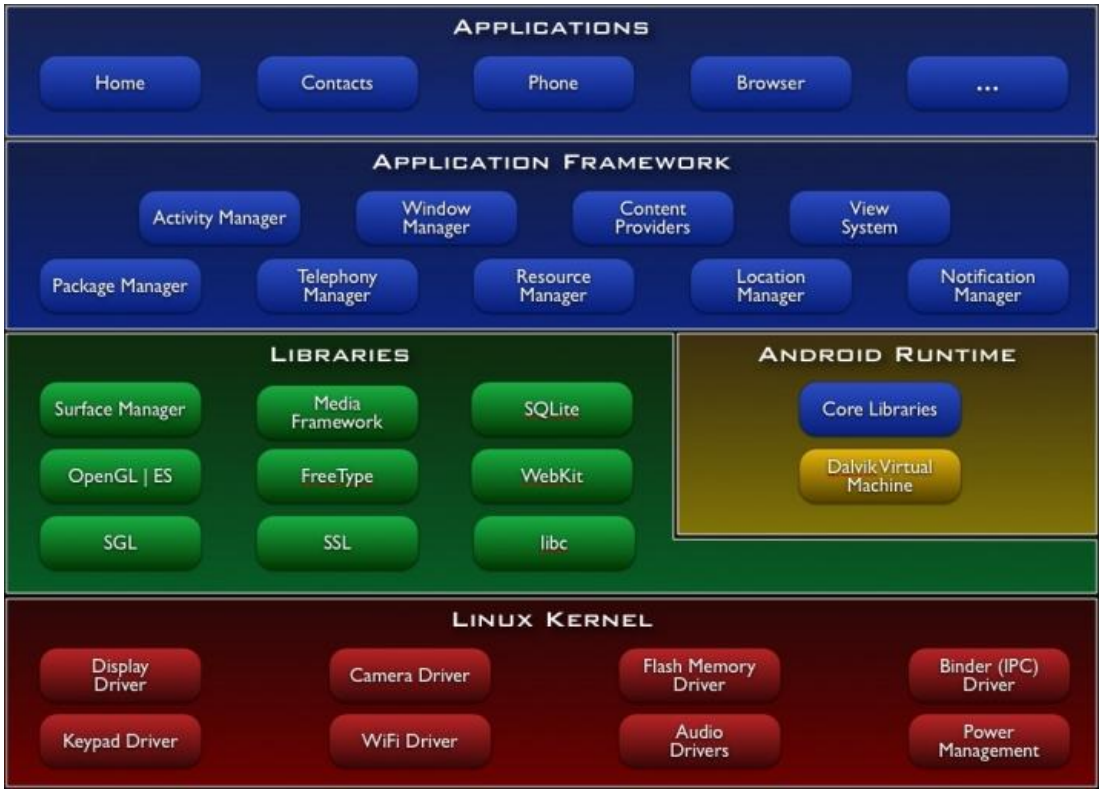


Figure 20 - Android Architecture.

Android architecture has the following components:

Applications – Java applications like a calendar, email client, SMS application, phone dialer, navigation, calculator, and others. This layer is used by the common user. Other layers are used by Google developers and hardware manufacturers.

Application Framework – This framework is followed for android developers. All developers can access to frameworks and APIs to manage phone's basic functions like, resources allocations, sensors, application manager, phone hardware and many other functionalities. The architecture is well designed to simplify the reuse of components. The application framework is as a set of basic tools with which a developer can build much more complex tools.

Libraries – All libraries in this layer are written in c and c++ the android's native language. These libraries tell to device how to handle with different kinds of data and are exposed to Android developers via Android Application Framework. Libraries examples are media library, graphics, 3D, etc.

Android Runtime – This layer includes set of base libraries that are required for java libraries. Every Android application gets its own instance of Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently and it executes files in executable (.Dex) optimized for minimum memory.

Linux Kernel - This layer includes Android's memory management programs, security settings, power management software and several drivers for hardware, file system access, networking and inter process communication. The kernel also acts as an abstraction layer between hardware and the rest of the software stack.

3.3.3. Android SDK

Android SDK is a platform provided by Google for Android application development. Android SDK has a wide range of useful tools to develop for Android system. These include a debugger, libraries, a handset emulator, documentation, sample code and many tutorials. SDK works almost in all operating systems like Windows, Mac OS X and Linux. The official IDE (Integrated Development Environment) is Eclipse using the Android Development Tools (ADT) plugin. The SDK also support the older versions so it is possible to develop for older devices. This tool is free and it is downloadable from Android Developers site.

3.3.4. Eclipse Software

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written in Java and can be used to develop applications in Java and, by means of various plug-ins, other programming languages like C++, C, COBOL, etc.

Android Development Tools (ADT) is a plugin for eclipse IDE that is designed to give a powerful integrated environment in which to build Android applications. ADT extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, add components based on Android Framework API, debug Android Applications and export signed .apk files to distribute applications in Android market.

4. System Demonstration and Validation – Credit Transfer

This chapter focuses on the performance evaluation and validation of Credit Transfer. Firstly it presents the application demonstration, through the windows that launch the program, insertion money screen and receiving money screen.

4.1. Application Demonstration

As mentioned in the last chapter, Credit Transfer is an easy way to transfer money between two mobile devices. The application is very simple to use. It is only necessary introduce the amount of money and press a button to begin the transaction. After that it needs to touch the other device and the transaction begins.

Figure 21 shows the first screen of Credit Transfer. In this screen the user introduces the amount to transfer and press button “ok”. After press it

screen waiting appears like in figure 22. This simple way to transfer money give the possibility to user avoid a lot of boring steps. Today for money transaction between two mobile devices a user needs to send a message or make a call to his operator. Credit transfer is a very well choice to substitute the actual mechanism.

Some world mobile network operators are trying to acquire Credit Transfer to simplify their money transfer mechanisms.

Figure 21 - Credit Transfer home screen.

While the “Waiting screen” is present on the device screen the amount of money was already introduced. Figure 22 represents a little group of steps. After insert money and press ok the application send to NFC chip all the information needed to communicate with the other device. At the same time the sender device turn on its Bluetooth adapter, because after touching with NFC the communication are made with Bluetooth communications.

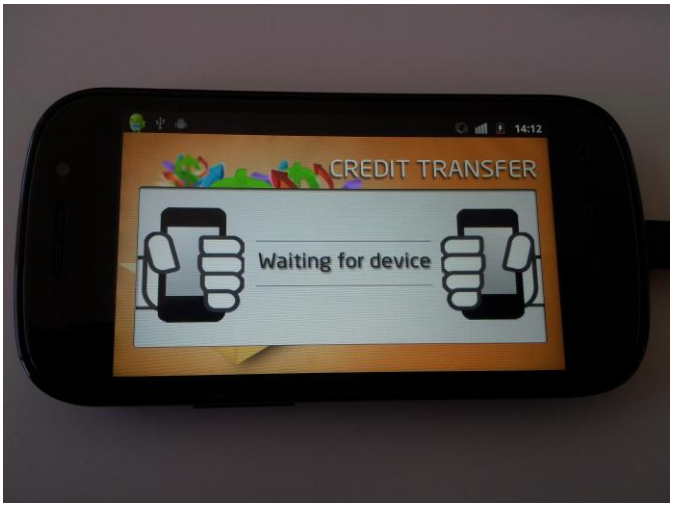


Figure 22 - Credit Transfer Waiting Screen.

Figure 23 appears after the touch between two devices. All devices that have a copy of Credit Transfer installed are registered in Android intent list to receive a notification if another credit transfer touches them. After the sender touches the receiver device a popup message appear to receiver. The message is to receiver allow the transfer. Every data stored in sender NFC chip is transmitted to receiver NFC chip. If receiver accepts the transaction Credit Transfer application will operate on received data.

Figure 23 - Credit Transfer Accept Screen.

Figure 24 shows the next step. In this step receiver Credit Transfer work with data received before. The operations made in this step are that there are today, but instead of user needs to send a message or make a call to begin the transaction he only needs Credit Transfer. All request to the operator are made in this step, if the sender does not have enough money he receive a message with the information and the operation is canceled. Receiver Bluetooth adapter is turned on before this step.

Figure 24 - Credit Transfer Transaction Screen.

Figure 23 and Figure 24 shows one of the most important steps, because it is here where the requests for operator are made. In this step a request is sent to the user's network operator to ask for permission to send money to other device. There are to answers from operator. If the user has money the operation begins, otherwise the network operator sends a message to sender user with a negative answer. All necessary data exchanged between both devices is sent by Bluetooth previously enabled by NFC touch event.

If the operation is successful a screen like in Figure 25 appears. All transactions are already done and the receiver has the money in his mobile phone. This application gives simplicity to user to do a simple credit transaction operation. All Bluetooth communications are turned off after all operations are done.

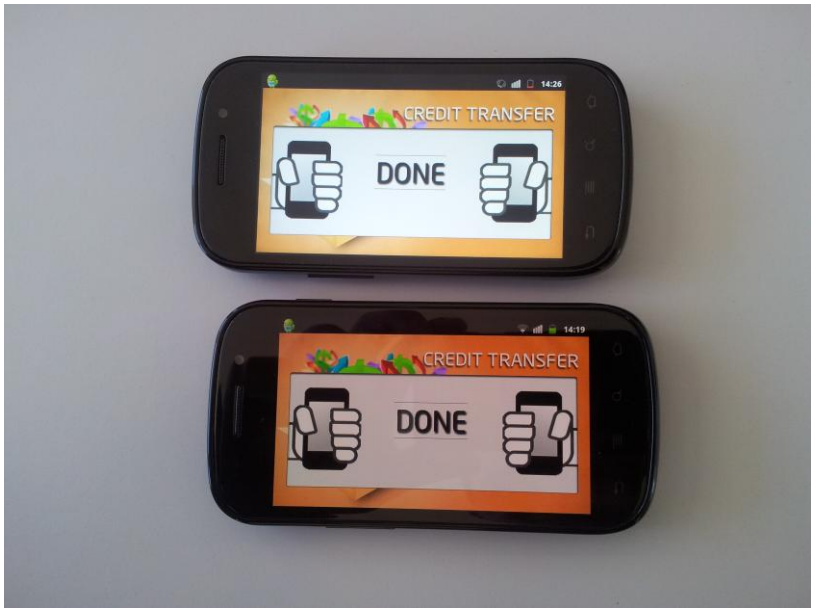


Figure 25 - Credit Transfer Operation Done Screen.

4.2. Application Validation

The performance evaluation and real deployment of Credit Transfer is presented on this section. The application validation was performed through exhaustive running experiments. Real devices were used in all the performed tests, as may be seen in all figures shown in application demonstration. Credit transfer was deployed in the only NFC android NFC based system until now, the Nexus S. Every application functionalities were tested using two devices and all functionalities were tested (Server Communications, Bluetooth enabling time, transfer time). These experiments enabled various debugging operations.

In order to validate the application, some measurements are used and were compared with similar technologies.

4.2.1. Comparison of Wireless Communication Standards

Table 8 shows a comparison between NFC and some other wireless technologies. As it shown in the table, the NFC technology is definitely not suited to transferring large amounts of data over long distances. This is a job for other wireless communication protocols such as any the flavours of Wi-Fi. However the NFC technology is the best way to transfer a small amount of information in a very short amount of time.

Table 8 - A Comparison of Wireless Communication Standards.

	International Standard	Operating Frequency	Maximum Data rate	Maximum Distance	Power Consumption Rate
Bluetooth 3.0	IEEE 802.15.1	2.4GHz	22 Mb/s	100 m	hours / days
Bluetooth 2.1	IEEE 802.15.1	2.4GHz	3 Mb/s	100 m	days
NFC	ISO 14443	13.56 MHz	0.42 Mb/s	10 cm	∞
Wi-Fi B	IEEE 802.11b	2.4GHz	11 Mb/s	100 m	hours / days
Wi-Fi G	IEEE 802.11g	2.4GHz	54 Mb/s	100 m	hours / days
Wi-Fi N	IEEE 802.11n	5 / 2.4GHz	144 Mb/s	100 m	hours / days
ZigBee	IEEE 802.15.4	2.4GHz	0.25 Mb/s	100-1200 m	months / years

The maximum distance supported by NFC is not a disadvantage. This short distance is seen like a security mechanism. While all other wireless technologies need security protocols and mechanism to secure the connection the short distance used by NFC is one of the best security mechanism. Another NFC advantage is the power consumption. It is almost unlimited because passive NFC does not need energy to work. This energy is given by the other active device.

4.2.2. Comparison between Bluetooth and NFC

NFC and Bluetooth are two transmission technologies. One and other are advantages and disadvantages. Some tests were made between them. Three measurements are chosen: setup time, data speed and range. In credit transfer is possible to test all measurements.

Setup speed is a very important feature of a system. This feature affects all the system because the time to begin the transfer is very important. NFC setup time is less than 0.1 seconds and the time for Bluetooth devices are average 6 seconds. NFC devices can setup or pair with other devices faster than Bluetooth. NFC is well suited for payment and ticketing applications. Figure 26 shows the setup time of two technologies.

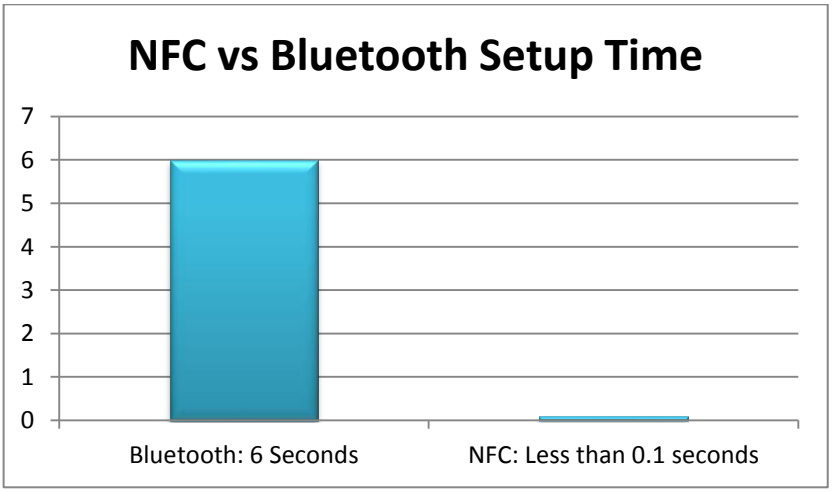


Figure 26 - NFC vs Bluetooth Setup Time.

Figure 27 shows a chart with a range comparison between NFC and Bluetooth. NFC range is less than 0.2 meters and Bluetooth range is about 10 meters. NFC is more suited to point to point transactions. The close range for NFC transactions reduces interference from other always on network devices. NFC works with devices is not powered by a battery, including NFC embedded phone covers, smartposters, and contactless smart

card. NFC is well suit for applications that need security mechanisms. The proximity is one of best security mechanism.

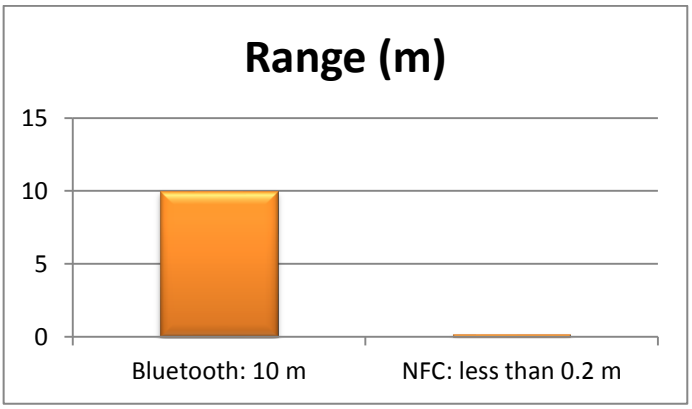


Figure 27 - NFC vs Bluetooth Maximum Range.

Figure 28 shows a comparison data speed comparison between Bluetooth and NFC technologies. NFC data speed is about 424 kbits / second and Bluetooth data speed is about 2.1 mbit / second. Bluetooth can sen data faster than NFC. Bluetooth is faster at sending small multimedia files than NFC.

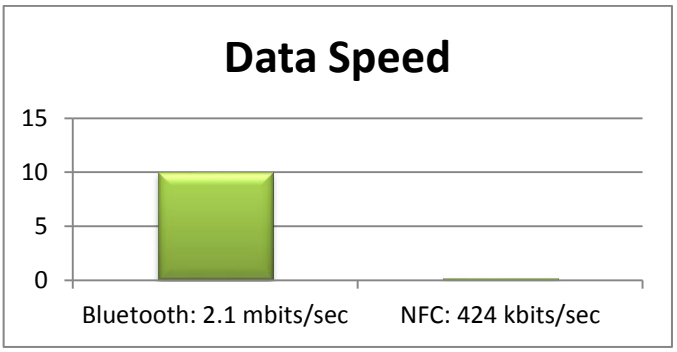


Figure 28 - NFC vs Bluetooth Data Speed.

5. System Demonstration and Validation – NFC Launcher

This chapter focuses on the performance evaluation and validation of NFC Launcher. Firstly it presents the application demonstration, through the windows that launch the program, starting in home screen, reader screen, writer screen, register screen and information screens.

5.1. Application Demonstration

As mentioned in the last chapter, NFC Launcher is an easy way to launch application with a simple touch gesture [64]. The interface is very simple. It has large buttons for fingers usage. The application has a web portal where everything is managed, since smartcodes (code used in NFC Launcher) to application packages. Everything is managed by user in admin portal.

Figure 29 shows the webportal home page. In this page the user can view all information about NFC Launcher and register as a new costumer. The usage of NFC is explained in this page as it seen in the figure. The home screen page has a small tutorial about NFC Launcher usage. It can be consulted inside mobile application too. NFC launcher portal is one of the most important parts of this project because inside it the user manages all necessary NFC Launcher configurations.

Figure 29 - NFC Launcher Web Portal.

Figure 30 shows the NFC Launcher's administration page. At the top is available the shortcuts of webportal. There are tutorials for all NFC Launcher functionalities and usages. The user can consult the writer and reader tutorial, and have access to technical support page. View profile shows all information about the user, the username, email, company name, phone contact, adress, zip-code, city and country. All information can be edited by user. Companies or single users can use NFC Launcher. They only need to register and configure all the system, install the application on device and use it. The page also has the smartcodes and applications management menus. Here it is possible to add a new smartcode or

application. Smartcodes are the most important feature in NFC Launcher, because each smartcode has a different task to execute. Each smartcode only has one application. On the other hand an application can match one or more smartcodes. An application does not be in the android market. Every downloadable application can be used in NFC Launcher.

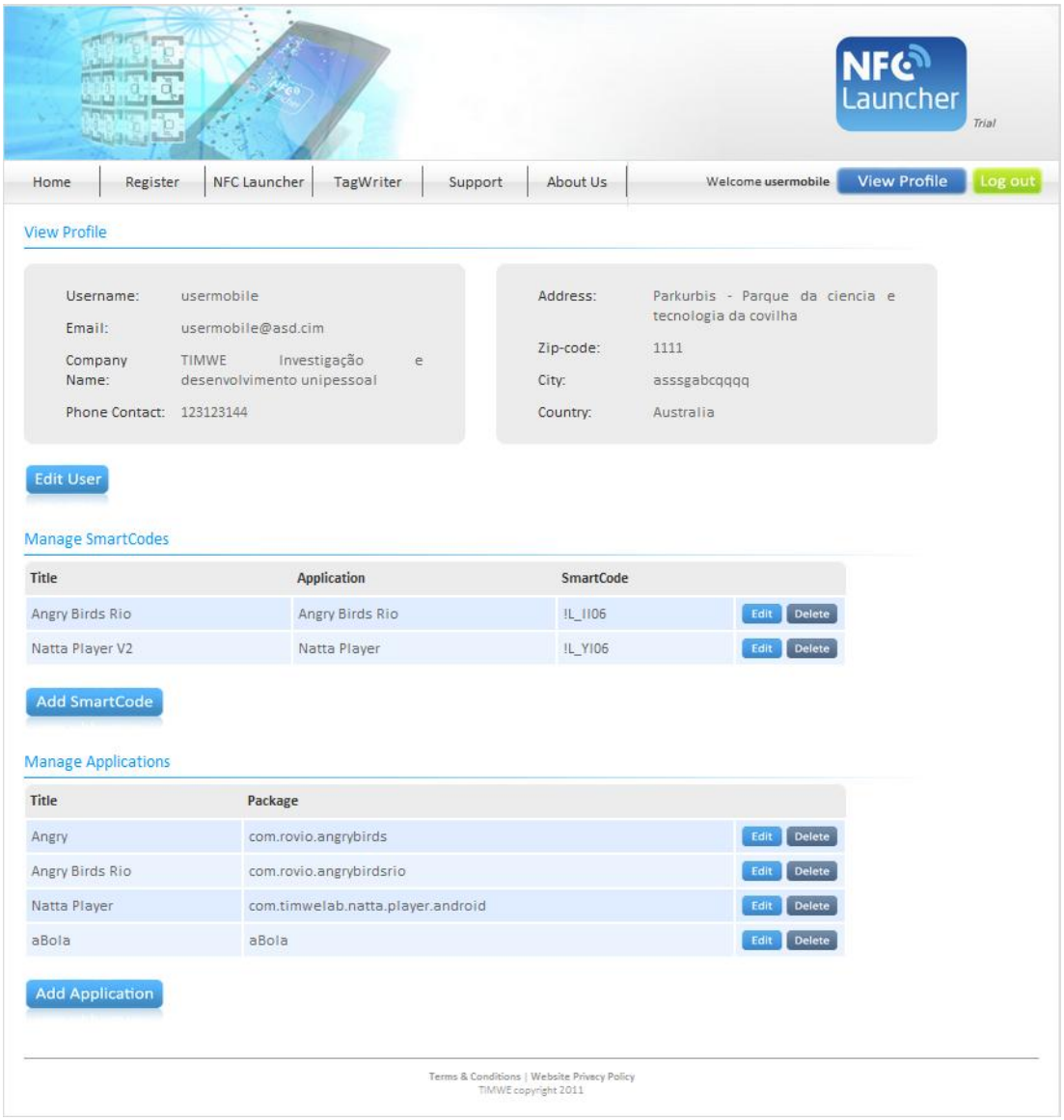


Figure 30 - NFC Launcher Web Portal Admin Page.

If button “add smartcode” is clicked a screen like figure 34 appears. Here a name for smartcode is necessary, then the user need to choose the application that will be associated to smartcode and if he wants cache in the smartcode. Smartcodes are given randomly by TIMwe. To attribute an application to a smartcode, it needs to be added before. The cache time is used to send the information about cache device. If the user wants cache in a smartcode he needs to insert how much cache he wants in hours. If he does not want any cache time he needs to inside the value 0.

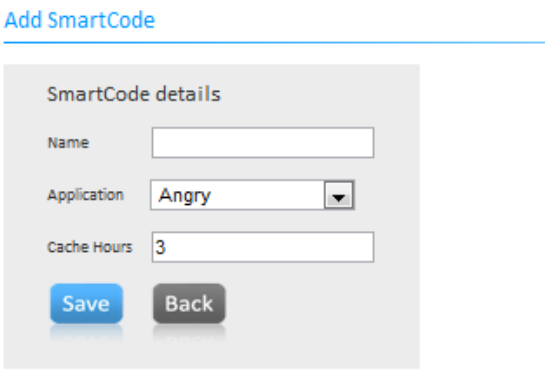


Figure 31 - NFC Launcher "Add SmartCode".

If button “add application” is clicked a screen like in figure 32 appears. In this form it is necessary a name for the application. This is the name that will appear to the mobile application user. The application platform Android, Blackberry, Bada or Symbian can be chosen, and finally the most important the application package. Finally if the application is available on operating system market.

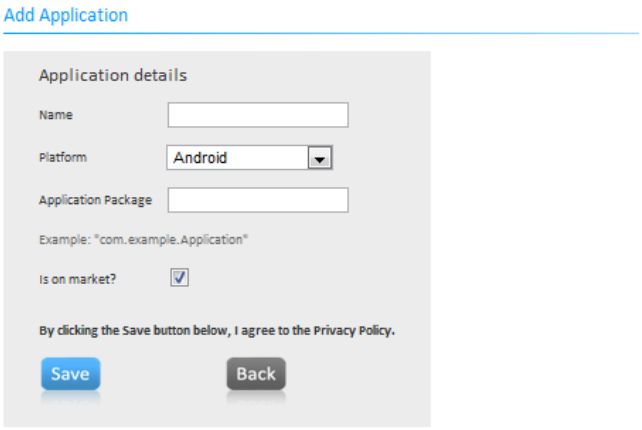


Figure 32 - NFC Launcher "Add Application".

Figure 33 shows the mobile application in Android operating system running in a Nexus S with Android Gingerbread 3.3.4. It is also shown two NFC tags used for the demonstration. The device has the NFC Launcher icon in the screen. NFC Launcher needs a NFC based device. The user needs to turn on NFC chipset in system definitions. Today the only Android device with NFC is Nexus, same like the mobile used in demonstration, but Samsung prepare the launch of second version of Nexus S, the Nexus Prime, another NFC based system. In next years a boom of NFC will occurs, because all manufacturers are making new devices with NFC technology. In the figure it is shown two tags used in demonstration. In next section there is a table with other types of tags and their features.

Figure 33 - NFC Launcher android application and NFC tags.

After launch the application the home screen appears like in Figure 34. This a very simple and well look screens. It has the application logo and a little description about NFC Launcher. Below logo bar there are four touchable buttons. A reader button, where NFC Launcher can read NFC tags, the writer button where the user can write NFC Launcher supported tags, a button with direct access to NFC Launcher mobile portal and the last button with more information about NFC Launcher, like tutorials, terms and conditions, and TIMwe company information.

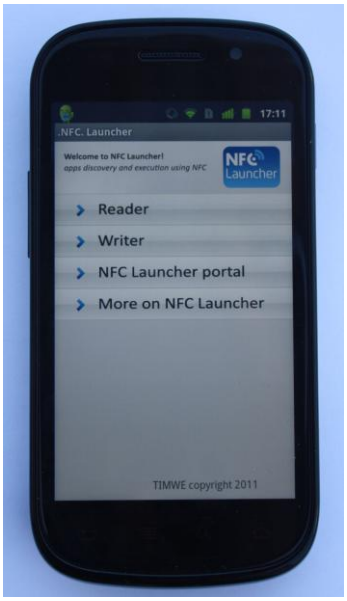


Figure 34 - NFC Launcher Home Screen.

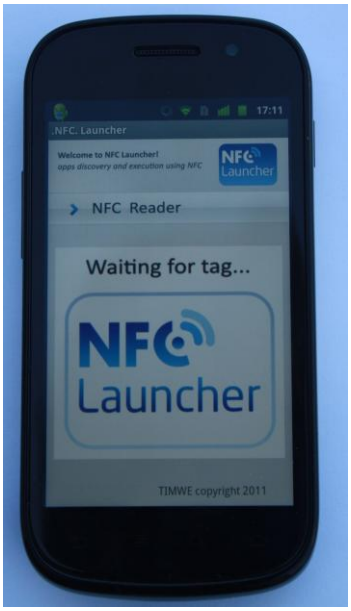


Figure 35 - NFC Launcher Reader Screen.

If Reader button is pressed a screen like figure 35 appears. In this screen if the user touches a NFC tag with the mobile phone the application saved in the tag is launched. Every types used by NFC Forum are recognized

like text, uri, smartposter, digital signature, etc. NFC Launcher can read the common TNF used by NFC community, so if there is more than one NFC application in the device the NFC Launcher is launched first it is in the front of the queue.

The other functionality of NFC Launcher is the writer mode. This feature allows the user to write own NFC tags with an applications previously created in web portal. To access this functionality the user needs to be registered in the NFC Launcher web portal. Figure 36 shows the login page inside application and figure 37 shows the recovery password screen. Only NFC Launcher registered user have access to writer mode. The registration is made in web portal. The main reason for it is the writer logs. Every tags written with NFC Launcher are logged in TIMwe servers. If a tag is corrupted or broken it is very easy to know what entity read it. Login also allows TIMwe to filter applications by entity. Each entity only have access to their added applications.

Figure 36 - NFC Launcher Writer Login Screen.

Figure 37 - NFC Launcher Writer Recovery Password.

Figure 38 shows the writer screen. In this screen there is a combo box (Figure 39) where the user can choose the application that he wants to write in the tag. If the user did not create an application in web portal it is possible to write a smartcode manually. In writer screen there a button to go to manual writer. Manual writer is shown in figure 40. Here the user need to know what is the smartcode to launch the application. Manual writer only accept codes with prefix "!L_". This prefix identifies tag content as a NFC Launcher tag. The only difference between the automatic writer and the manual writer is the previously insertion of application in NFC Launcher portal.

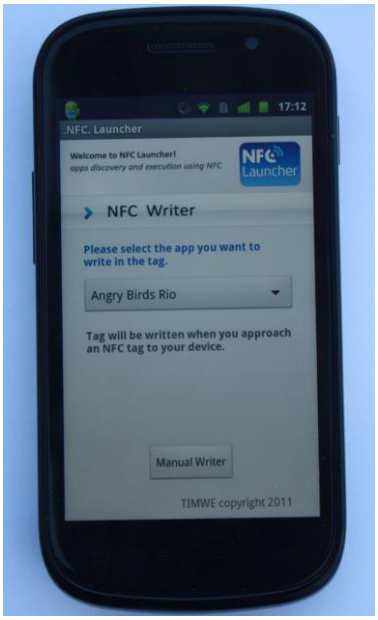


Figure 38 - NFC Launcher Writer Screen.

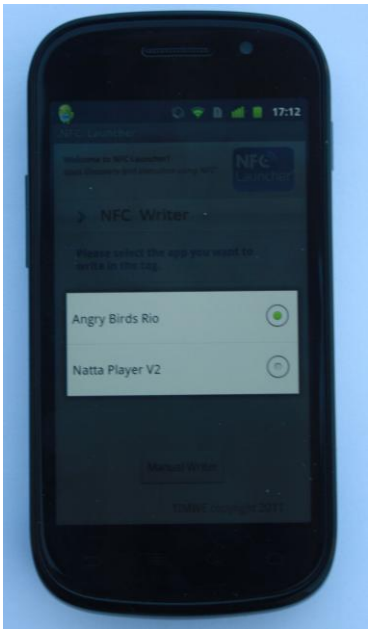


Figure 39 - NFC Launcher Choose Application Screen.

Figure 40 - NFC Launcher Manual Writer Screen.

It is possible to go to the mobile version web portal. The next two figures (Figure 41 and Figure 42) show the “how it works”, “about us” and “terms and conditions” screen. How it works menu has a little explanation about usage of NFC Launcher, how to write a tag or how to read a tag. For more information and a more detailed tutorial the user needs to go to web portal. In about us screen the TIMwe information is presented and the user can view all support contacts.

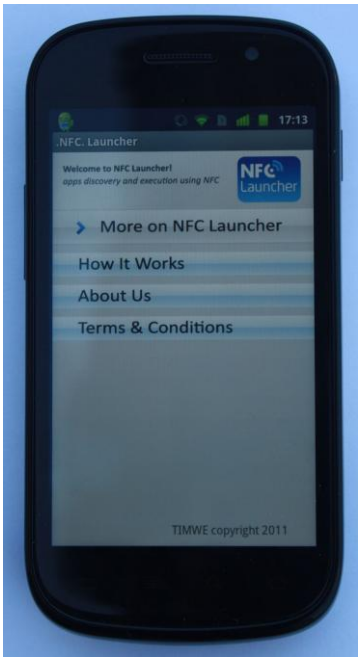


Figure 41 - NFC Launcher "More Options".

Figure 42 - NFC Launcher "How it works".

Figure 39 shows a tag with a smartcode. The NFC smartcode is chosen and written in the tag and a message like Figure 43 appears. Appears of a dialog with successful information means that the smartcode was written in the NFC tag. All the data contained in the NFC chip was transmitted to a tag. NFC Launcher writes tags with text RTD.

Figure 43 - NFC Launcher Write TAG success message.

After write the smartcode, the only thing needed is touch the tag with device. If the application written in tag is already installed in the device the application is launched, instead an Android market page appears and the application download begins. When the download finishes and the application is installed it is possible to launch the application. Next two figures (Figure 44 and Figure 45) show a device reading a tag and launch an application, in this case the game Angry Birds RIO.

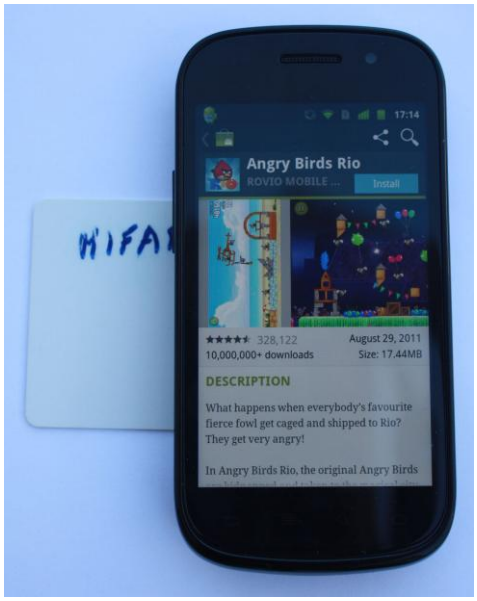


Figure 44 - NFC Launcher reading example without application installed.

Figure 45 - NFC Launcher reading example with application installed.

5.2. Application Validation

The performance evaluation and real deployment of NFC Launcher is presented on this section. The application validation was performed through exhaustive running experiments. Real devices were used in all the performed tests, as may be seen in all figures shown in application demonstration. NFC Launcher was deployed in the only NFC android NFC based system until now, the Nexus S. All application functionalities were tested using the portal and a Nexus S. The codes were configured in portal and tested in a real device.

In order to validate the application, many NFC tags were used. There were differences between every tag, like read distance, memory size, etc. In next topics it is shown table and charts with these features.

Every tag are different behaviors, response time was the measurement in focus in NFC Launcher application. Next pages show some charts with a comparison between them. Table 6 and table 7 have a lot of features of each tag.

Table 9 - Comparison between NFC tags for identification.

	Trikker BL38	Trikker-1k CL42	Trikker BT43	Trikker-1k CT50
				
Operational characteristics				
Item and person identification	Yes	Yes	Yes	Yes
Environment	dry indoor	dry indoor	outdoor / indoor	outdoor / indoor
Protection class	IP 50	IP 50	IP 65	IP 65
Temperature	5 ◦C - + 40 ◦C	5 ◦C - + 40 ◦C	-40 ◦C - + 70 ◦C	-40 ◦C - + 70 ◦C
Physical characteristics				
Delivery format	line or reel	line or reel	sheet	sheet
Type	label	label	hard tag	hard tag
Diameter	38 mm	42 mm	43 mm	50 mm
Thickness	0,2 mm (average)	0,3 mm (average)	2 mm (average)	2 mm (average)
Weight	0,5 g	0,5 g	3 g	3,5 g
Surface	synthetic Polyprint	synthetic Polyprint	3D	3D
Adhesive	permanent	permanent	permanent	permanent
Colour	white	3 colour FFS layout	white	3 colour FFS layout
Serial number	printed in black	printed in black	printed in black	printed in black
Electrical characteristics				
IC	NXP Mifare Ultra Light	NXP Mifare Standard 1k	NXP Mifare Ultra Light	NXP Mifare Standard 1k
Standard	ISO14443-A	ISO14443-A	ISO14443-A	ISO14443-A
Memory	512 bits	1024 bytes	512 bits	1024 bytes
Writable memory	48 characters	720 characters	48 characters	720 characters
Frequency	13,56 MHz	13,56 MHz	13,56 MHz	13,56 MHz
Reading distance	5 cm	5 cm	5 cm	5 cm

Table 10 - Comparison between NFC tags for payment and ticketing.

	Trikker-1k BC	Trikker-DESfire BC
		
Operational characteristics		
Usage	No (Nokia Field Force Solution applications)	Payment, ticketing and access control solutions
Environment	outdoor / indoor	outdoor / indoor
Temperature range	-20 ◦C - + 70 ◦C	-20 ◦C - + 70 ◦C
Delivery format	card	card
Physical characteristics		
Size	54 mm x 85,6 mm	54 mm x 85,6 mm
Thickness	0,8 mm	0,8 mm
Weight	approx. 2 g	approx. 6 g
Surface	PVC	PVC
Colour	blank	blank
Printed serial number	on request	on request
Printed personalisation	on request	on request
Id card encoding	on request	on request
Electrical characteristics		
IC	NXP Mifare Standard 1k	Mifare DESfire
Standard	ISO14443-A	ISO14443-A
Memory	1024 bytes	-
Writable memory	720 characters	4 kbytes
Frequency	13,56 MHz	13,56 MHz
Reading distance	10 cm	10 cm
Unique Identifier (UID)	-	7 bytes

5.2.1. Comparison of response time between NFC tags

Every tag are different behaviours, distance and response time were two measurements in focus. Next pages show some charts with a comparison between them. Table 9 and Table 10 have a lot of features of each tested tag. A very important feature in NFC technology is the response time. Each tag has it response time. This feature was validating using timers inside source code.

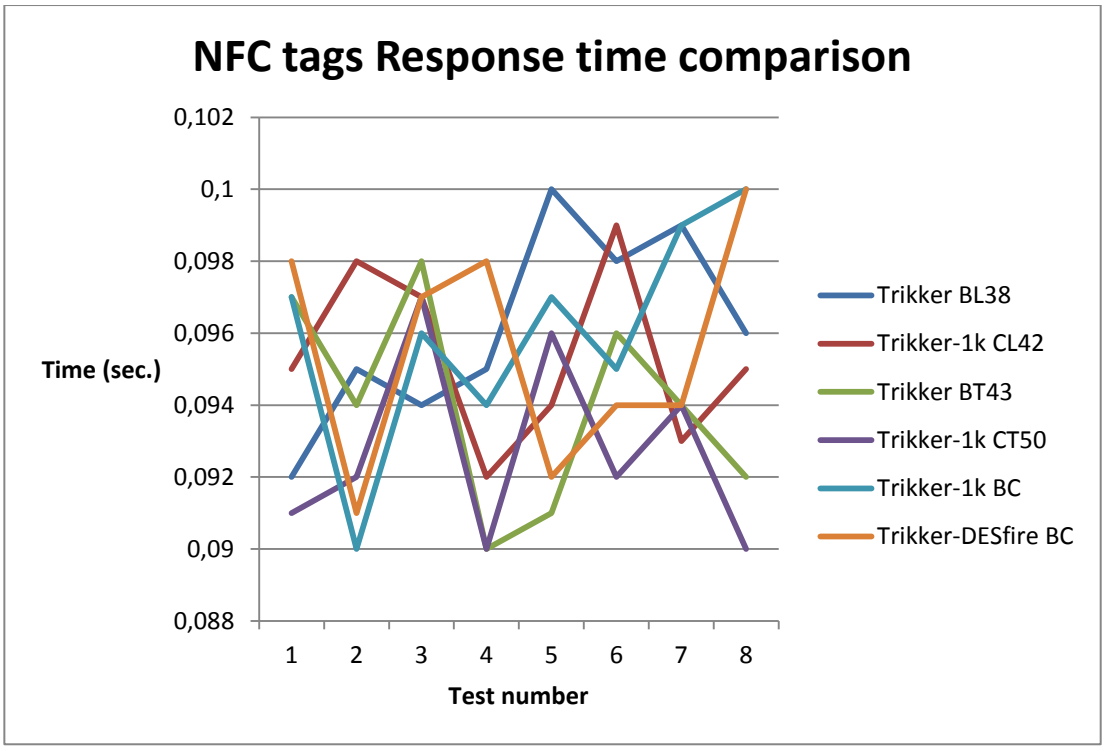


Figure 46 - NFC tags Response Time Comparison.

Figure 46 shows a comparison between tags response time. It is shown that every tag have similar response times between 0.09 and 0.1 seconds. It is possible to say that do not have a better type of tag in response time.

The response time is randomly and it not depends of tag.

6. Conclusions and Future Work

6.1. Conclusions

This chapter presents a synthesis of the main achievement and points to several directions for future work. The main objective of this dissertation was the development of NFC Android based applications. It was carried out with the building of an android mobile application providing utilities came from NFC technology. The main goal of the application is to turn easier the interaction between the man and the machine. Therefore all the dissertation objectives were successfully accomplished and all intermediate goals were successfully achieved.

After introducing and presenting the topic of this dissertation and define its objectives, Chapter two presents the revision on NFC technology literature.

Chapter three presented the requirement analysis for the construction of android NFC based android applications. Detailing the essentials requirements for all development process and the application diagrams.

In chapter four and chapter five were presented the demonstration and validation of two applications, Credit transfer and NFC Launcher. The first

is an android application that allows a user to transfer money between two devices. This is similar to actual mechanism but simpler. The user do not need to send any type of message to network operator, only need to introduce the money amount and transfer it. NFC Launcher is the second NFC based application demonstrated in this dissertation. This main goal of this application is launch other applications without user interaction. The mobile user only needs to touch a NFC tag for use the application. NFC launcher processes the code data and launches the previously recorded application. The user only needs to configure all NFC tags with a smartcode in a webportal.

It was also submitted a paper with some contributions for an international conference and more two will be submitted in the near future.

6.2. Future Work

To conclude this work, it just remains to suggest future research directions based on current work:

- Increase the number of mobile platforms where the application can be executed. Bada, RIM, iOS, Windows Phone 7 and Symbian are the main objectives.
- Include a better security mechanism in NFC applications. Possibility of development of a new communication protocol with another mechanism of security.
- Creation of innovative NFC based applications using other operating modes.

References

[1] R. Want, "Near Field Communication," *IEEE Pervasive Computing*, July 2011, pp. 4-7.

[2] E. Siira and V. Törmänen, "The Impact of NFC on Multimodal Social Media Application," *in Proceedings of the 2010 Second International Workshop on Near Field Communication*, April 2010, pp. 51-56.

[3] S. Dhawan, "Analogy of Promising Wireless Technologies on Different Frequencies: Bluetooth, WiFi, and WiMAX" *in Proceedings of the The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications*, August 2007, pp. 14.

[4] M. Langheinrich, "A survey of RFID privacy approaches" *Personal Ubiquitous Computing*, vol. 13, August 2009, pp. 413-421.

[5] G. Platform, "GlobalPlatform's Proposition for NFC Mobile: Secure Element Management and Messaging," *White Paper,* April 2009.

[6] V. Klos, M. O. v. Deventer, M. v. Staalduinen, and F. d. Hartog, "Mobile touch: NFC-like interaction with yesterday's phones" *in Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*, Las Vegas, NV, USA, 2009, pp. 13-14.

[7] R. Steffen, J. Preißinger, T. Schöllermann, A. Müller, and I. Schnabel, "Near Field Communication (NFC) in an Automotive Environment" *in Proceedings of the 2010 Second International Workshop on Near Field Communication*, Washington, DC, USA, 2010, pp. 15-20.

[8] A. Marcus, G. Davidzon, D. Law, N. Verma, R. Fletcher, A. Khan, and L. Sarmenta, "Using NFC-Enabled Mobile Phones for Public Health in Developing Countries" *in Proceedings of the 2009 First International*

Workshop on Near Field Communication, Washington, DC, USA, 2009, pp. 30-35.

[9] G. V. Damme, K. M. Wouters, H. Karahan, and B. Preneel, "Offline NFC payments with electronic vouchers" *in Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*, Barcelona, Spain, 2009, pp. 25-30.

[10] K. S. Kadambi, J. Li, and A. H. Karp, "Near-field communication-based secure mobile payment service" *in Proceedings of the 11th International Conference on Electronic Commerce*, Taipei, Taiwan, 2009, pp. 142-151.

[11] M. Zhao, S. Zhang, and X. Lin, "A chip solution for UWB-NFC receiver in CMOS 0.18um technology" *in Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, Caen, France, 2010, pp. 839-842.

[12] G. Broll and D. Hausen, "Mobile and physical user interfaces for NFC-based mobile interaction with multiple tags" *in Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, Lisbon, Portugal, 2010, pp. 133-142.

[13] H. Mika, H. Mikko, and Y.-o. Arto, "Practical Implementations of Passive and Semi-passive NFC Enabled Sensors" *in Proceedings of the 2009 First International Workshop on Near Field Communication*, 2009, pp. 69-74.

[14] S. Bhattacharya, "Enriching location information: an energy-efficient approach" *in Proceedings of the 13th international conference on Ubiquitous computing*, Beijing, China, 2011, pp. 519-522.

[15] G. Broll, S. Keck, P. Holleis, and A. Butz, "Improving the accessibility of NFC/RFID-based mobile interaction through learnability and guidance" *in Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, Bonn, Germany, 2009, pp. 1-10.

[16] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Practical NFC peer-to-peer relay attack using mobile phones" *in Proceedings of the 6th international conference on Radio frequency identification: security and privacy issues*, Istanbul, Turkey, 2010, pp. 35-49.

[17] G. Madlmayr, O. Dillinger, J. Langer, and J. Scharinger, "Management of Multiple Cards in NFC-Devices" *in Proceedings of the 8th IFIP WG 8.8/11.2 international conference on Smart Card Research and Advanced Applications*, London, UK, 2008, pp. 149-161.

[18] A. Nandwani, P. Coulton, and R. Edwards, "NFC Mobile Parlor Games Enabling Direct Player to Player Interaction" *in Proceedings of the*

2011 Third International Workshop on Near Field Communication, 2011, pp. 21-25.

[19] R. Kelkka, T. Kallonen, and J. Ikonen, "Remote identification and information processing with a near field communication compatible mobile phone" *in Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*, Ruse, Bulgaria, 2009, pp. 1-6.

[20] E. Siira, T. Tuikka, and V. Törmänen, "Location-Based Mobile Wiki Using NFC Tag Infrastructure" *in Proceedings of the 2009 First International Workshop on Near Field Communication*, 2009, pp. 56-60.

[21] M. Roland, J. Langer, and J. Scharinger, "Security Vulnerabilities of the NDEF Signature Record Type" *in Proceedings of the 2011 Third International Workshop on Near Field Communication*, 2011, pp. 65-70.

[22] M. Roland and J. Langer, "Digital Signature Records for the NFC Data Exchange Format" *in Proceedings of the 2010 Second International Workshop on Near Field Communication*, 2010, pp. 71-76.

[23] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "On the security issues of NFC enabled mobile phones" *Int. J. Internet Technol. Secur. Syst.*, vol. 2, 2010, pp. 336-356.

[24] M. Reveilhac and M. Pasquet, "Promising Secure Element Alternatives for NFC Technology" *in Proceedings of the 2009 First International Workshop on Near Field Communication*, 2009, pp. 75-80.

[25] H. Franssila, "User Experiences and Acceptance Scenarios of NFC Applications in Security Service Field Work" *in Proceedings of the 2010 Second International Workshop on Near Field Communication*, 2010, pp. 39-44.

[26] G. Madlmayr, O. Dillinger, J. Langer, and C. Schaffer, "The benefit of using SIM application toolkit in the context of near field communication applications" *in Proceedings of the International Conference on the Management of Mobile Business*, 2007, pp. 5.

[27] G. Madlmayr, J. Langer, and J. Scharinger, "Managing an NFC Ecosystem" *in Proceedings of the 2008 7th International Conference on Mobile Business*, 2008, pp. 95-101.

[28] R. Pellerin, C. Yan, J. Cordry, and E. Gressier-Soudan, "Player profile management on NFC smart card for multiplayer ubiquitous games" *Int. J. Comput. Games Technol.*, vol. 2009, 2009, pp. 1-9.

[29] W. Chen, G. P. Hancke, K. E. Mayes, Y. Lien, and J.-H. Chiu, "NFC Mobile Transactions and Authentication Based on GSM Network" *in*

Proceedings of the 2010 Second International Workshop on Near Field Communication, 2010, pp. 83-89.

[30] S. Miranda and N. Pastorelly, "NFC Mobiquitous Information Service Prototyping at the University of Nice Sophia Antipolis and Multi-mode NFC Application Proposal" *in Proceedings of the 2011 Third International Workshop on Near Field Communication*, 2011, pp. 3-8.

[31] F. Michahelles, F. Thiesse, A. Schmidt, and J. R. Williams, "Pervasive RFID and Near Field Communication Technology" *IEEE Pervasive Computing,* vol. 6, 2007, pp. 94-96, c3.

[32] S. Chaumette, D. Dubernet, J. Ouoba, E. Siira, and T. Tuikka, "Architecture and comparison of two different user-centric NFC-enabled event ticketing approaches" *in Proceedings of the 11th international conference and 4th international conference on Smart spaces and next generation wired/wireless networking*, St. Petersburg, Russia, 2011, pp. 165-177.

[33] Z. Antoniou and D. N. Kalofonos, "User-centered design of a secure P2P personal and social networking platform" *in Proceedings of the Third IASTED International Conference on Human Computer Interaction,* Innsbruck, Austria, 2008, pp. 186-191.

[34] H. Ailisto, M. Isomursu, T. Tuikka, and J. Häikiö, "Experiences from interaction design for NFC applications" *J. Ambient Intell. Smart Environ.,* vol. 1, 2009, pp. 351-364.

[35] J. Ylinen, M. Koskela, L. Iso-Anttila, and P. Loula, "Near Field Communication Network Services" *in Proceedings of the 2009 Third International Conference on Digital Society,* 2009, pp. 89-93.

[36] M. Gamage, M. Hayasaka, and T. Miki, "A connection-oriented network architecture with guaranteed QoS for future real-time applications over the Internet" *Comput. Netw.,* vol. 50, 2006, pp. 1130-1144.

[37] Y. H. Ho, A. H. Ho, and K. A. Hua, "Connectionless protocol: a localised approach to wireless ad hoc networks" *Int. J. Ad Hoc Ubiquitous Comput.,* vol. 2, 2007, pp. 21-35.

[38] H. Ailisto, L. Pohjanheimo, P. Välkkynen, E. Strömmer, T. Tuomisto, and I. Korhonen, "Bridging the physical and virtual worlds by local connectivity-based physical selection" *Personal Ubiquitous Comput.,* vol. 10, 2006, pp. 333-344.

[39] M. Khabbazian, F. Kuhn, D. R. Kowalski, and N. Lynch, "Decomposing broadcast algorithms using abstract MAC layers" *in Proceedings of the 6th International Workshop on Foundations of Mobile Computing*, Cambridge, Massachusetts, 2010, pp. 13-22.

[40] H.-J. Moon, S. Y. Moon, and W. H. Kwon, "Parameter Region for the Proper Operation of the IEEE 802.2 LLC Type 3 Protocol: A Petri Net

Approach" *in Application of Petri Nets to Communication Networks, Advances in Petri Nets*, 1999, pp. 131-149.

[41] S. Kobayashi, K. Sakamura, and T. Morokuma, "A Dynamic Retargettable Multi-Protocol RFID Reader/Writer" *in Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops - Volume 02*, 2007, pp. 340-346.

[42] NFC Data Exchange Format, NFC Forum Technical Specification, NDEF 1.0, July 2006.

[43] F. KneiBl, R. Röttger, U. Sandner, J. M. Leimeister, and H. Krcmar, "All-I-Touch as Combination of NFC and Lifestyle" *in Proceedings of the 2009 First International Workshop on Near Field Communication*, 2009, pp. 51-55.

[44] I. Sánchez, J. Riekki, J. Rousu, and S. Pirttikangas, "Touch & Share: RFID based ubiquitous file containers" *in Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia*, Umea, Sweden, 2008, pp. 57-63.

[45] K. Peternel, M. Pogacnik, J. Bester, L. Zebec, M. Pustisek, and A. Kos, "Touch to Communicate Using NGN Open Interfaces" *in Proceedings of the 2011 Ninth Annual Communication Networks and Services Research Conference*, 2011, pp. 130-136.

[46] F. Borrego-Jaraba, I. L. Ruiz, and M. Á. Gómez-Nieto, "NFC solution for the development of smart scenarios supporting tourism applications and surfing in urban environments" *in Proceedings of the 23rd international conference on Industrial engineering and other applications of applied intelligent systems - Volume Part III*, Cordoba, Spain, 2010, pp. 229-238.

[47] NFC Record Type Definition, NFC Forum Technical Specification, RTD 1.0, July 2006.

[48] F. Borrego-Jaraba, I. L. Ruiz, and Miguel Ángel Gómez-Nieto, "A NFC-based pervasive solution for city touristic surfing" *Personal Ubiquitous Comput.*, vol. 15, 2011, pp. 731-742.

[49] J. Zou, C. Zhang, C. Dong, C. Fan, and Z. Wen, "Mobile Payment based on RFID-SIM Card" *in Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology*, 2010, pp. 2052-2054.

[50] R. Verdult and F. Kooman, "Practical Attacks on NFC Enabled Cell Phones" *in Proceedings of the 2011 Third International Workshop on Near Field Communication*, 2011, pp. 77-82.

[51] P. C. Garrido, G. M. Miraz, I. L. Ruiz, and M. A. Gomez-Nieto, "Use of NFC-based Pervasive Games for Encouraging Learning and Student Motivation" *in Proceedings of the 2011 Third International Workshop on Near Field Communication*, 2011, pp. 32-37.

[52] Z. Lou, "NFC Enabled Smart Postal System" *in Proceedings of the 2010 Second International Workshop on Near Field Communication*, 2010, pp. 33-38.

[53] H. Aziza, "NFC Technology in Mobile Phone Next-Generation Services" *in Proceedings of the 2010 Second International Workshop on Near Field Communication*, 2010, pp. 21-26.

[54] B. Ozdenizci, K. Ok, V. Coskun, and M. N. Aydin, "Development of an Indoor Navigation System Using NFC Technology" *in Proceedings of the 2011 Fourth International Conference on Information and Computing*, 2011, pp. 11-14.

[55] J. Fontecha, R. Hervas, J. Bravo, and V. Villarreal, "An NFC Approach for Nursing Care Training" *in Proceedings of the 2011 Third International Workshop on Near Field Communication*, 2011, pp. 38-43.

[56] G. Broll, R. Graebsch, M. Scherr, S. Boring, P. Holleis, and M. Wagner, "Touch to Play -- Exploring Touch-Based Mobile Interaction with Public Displays" *in Proceedings of the 2011 Third International Workshop on Near Field Communication*, 2011, pp. 15-20.

[57] G. Madlmayr, P. Kleebauer, J. Langer, and J. Scharinger, "Secure Communication between Web Browsers and NFC Targets by the Example of an e-Ticketing System" *in Proceedings of the 9th international conference on E-Commerce and Web Technologies*, Turin, Italy, 2008, pp. 1-10.

[58] M. Hutter and R. Toegl, "A Trusted Platform Module for Near Field Communication" *in Proceedings of the 2010 Fifth International Conference on Systems and Networks Communications*, 2010, pp. 136-141.

[59] G. P. Hancke, "Practical eavesdropping and skimming attacks on high-frequency RFID tokens" *J. Comput. Secur.*, vol. 19, 2011, pp. 259-288.

[60] H. Barthélemy, S. Meillère, J. Gaubert, N. Dehaese, and S. Bourdel, "OTA based on CMOS inverters and application in the design of tunable bandpass filter" *Analog Integr. Circuits Signal Process.*, vol. 57, 2008, pp. 169-178.

[61] S. Sabetghadam, M. Niamanesh, and J. Esmaeili, "A Model for Assured Software Download on Mobile Terminals" *in Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing - Volume 02*, 2009, pp. 432-436.

[62] M. F. A. Karim and R. Muhamad, "Integration of near field communication (NFC) and Bluetooth technology for medical data acquisition system" *in Proceedings of the 6th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*, Elounda, Greece, 2006, pp. 147-152.

[63] N. Erasala and D. C. Yen, "Bluetooth technology: a strategic analysis of its role in global 3G wireless communication era" *Comput. Stand. Interfaces,* vol. 24, 2002, pp. 193-206.

[64] E. O'Neill, P. Thompson, S. Garzonis, and A. Warr, "Reach out and touch: using NFC and 2D barcodes for service discovery and interaction with mobile devices" *in Proceedings of the 5th international conference on Pervasive computing*, Toronto, Canada, 2007, pp. 19-36.