NFC Prototype Bonanza

Øyvind Holmstad, Tor Kreutzer
Department of Computer Science
Faculty of Science and Technology
University of Tromsø, Norway
Email: nfc-city@cs.uit.no

July 2011

Supported by the Norwegian research council, VERDIKT project number 201377.

Contents

Chapter 1

Introduction

This report presents the results of the research conducted by Tor Kreutzer and Øyvind Holmstad during the summer of 2011 at the University of Tromsø. The goal of the project was to gain practical experience with Near Field Communication (NFC) technology by exploring its properties through hands-on development of applications and services.

To explore the applicability and limitations of NFC we have developed a range of different prototype applications and services that utilize the technology in different ways. The applications are mainly implemented on the Android mobile platform, but many of them communicate with servers running on traditional computers.

The applications vary widely, from the simple *Tagger* which can read and write NFC tags, to *NFC Presenter* which uses NFC to simplify the process of starting presentations from your mobile device. *NFC Safari* uses the user's location to identify the closest sightseeing spot, and from there on takes him on a city safari. Applications like *PartyShare* and *Are You the One?* explores how powerful NFC can be in a social setting. Detailed descriptions and implementation details of all applications can be found in chapter 4.

1.1 Context

This project is part of the NFC City-project, which vision of is to provide value by simplifying everyday tasks through use of NFC services. The underlying idea is to expose users to a range of NFC services within a limited geographical area - a NFC city. The NFC city is chosen to be Tromsø.

By utilising existing technologies to provide NFC services within ticketing, payment and physical access, the project will create an arena where services from different providers can run independently and with good usability on the same mobile handsets. As ticketing, payment and physical access is being looked into by other project parties, we will look at other areas where NFC technology can be helpful.

The objective is to gain user experience on a pilot solution and knowledge on the solution's applicability. When the project is finished after three years, it should have released some of the potential of NFC services in terms of increased number of handsets, users and services offered by a value system of service providers, mobile operators and financial institutions.

Chapter 2

Near Field Communication

In this section we will give a brief overview of what Near Field Communication (NFC) is, how it works, and what is can be used for.

2.1 Overview

NFC-Forum, a forum formed to advance the use of Near Field Communication technology by developing specifications, ensuring interoperability among devices and services, and educating the market about NFC technology, defines the technology as follows:

> *Near Field Communication (NFC) technology makes life easier and more convenient for consumers around the world by making it simpler to make transactions, exchange digital content, and connect electronic devices with a touch.*

NFC is a standards-based connectivity technology for short range communication. It tries to harmonize today's diverse contact-less technologies, enabling current and future solutions involving access control, consumer electronics, information collection and exchange, loyalty and coupons, payments and transport. In other words, NFC can make it possible to create services like touching your mobile phone to a lock to open a door, or touch it to a terminal in a shop to pay for your groceries.

NFC can be implemented in all types of devices, including your smart phone, your tablet, your computer and your car. However, the number of NFC-enabled devices today are quite limited. There has been a deadlock situation where handset vendors, service providers and users are waiting for the others to make a move. The lack of NFC handsets hinder service development and lack of services is no good incitement for developing new handsets. However, as the option to pay for your goods through NFC will become more widespread, the number of devices supporting the technology is likely to explode.

2.2 Technical details

In short, NFC is a set of short-range wireless technologies. Where Bluetooth works within a range of 10 meters with a maximum bandwidth of 2.1 Mbit/s, NFC is limited to only work within distances as short as 4 cm, and with a bandwidth of less then 1000kbit/s. NFC operates at 13.56 MHz on the ISO/IEC 18000-3 air interface.

NFC always involves an initiator and a target; the initiator actively generates an RF field that can power a passive target. This enables NFC targets to take very simple form factors such as tags, stickers, key fobs, or cards that do not require batteries. NFC peer-to-peer communication, where both devices are powered, is also possible. However, the limited bandwidth makes it inferior to Bluetooth for big data transfers.

NFC tags contain data, and may be configured to be both read-only and rewritable. They can be custom-encoded by their manufacturers or use the specifications provided by the NFC Forum, including the Type 2 Tag format[1] for the tag header and the NDEF data format[2] for the payload. The tags can securely store personal data such as debit and credit card information, loyalty program data, PINs and networking contacts, among other information. The NFC Forum defines four types of tags which provide different communication speeds and capabilities in terms of configurability, memory, security, data retention and write endurance.

2.2.1 NDEF

The NFC Data Exchange Format (NDEF)[2] specification defines a message encapsulation format to exchange information between NFC Forum Devices and NFC Forum Tags. In the specification it is defined as follows:

> 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.

The type identifiers may be URIs, MIME media types, or NFC-specific types. NFC-specific types are well defined, and permits compact identification of well-known types commonly used in NFC applications. These types include URIs(Urls,, telephone numbers, e-mail adresses, vCards (for contact information), and standard text. In addition there is allocated space for organizations that wish to use it for their own NFC-specific purposes.

The NDEF format permits messages of different sizes, and provides the possibility to stream data from one device to another by chopping the payload into chunks. This ensures that NDEF messages of unknown length at the time the data is generated can be delivered continuously.

The specification notes that NDEF is strictly a message format, which provides no concept of a connection or of a logical circuit, nor does it address head-of-line problems.

Chapter 3

Development

In this section we will describe challenges and solutions for the development process of NFC applications. We will start by giving a description of the equipment used in the development process, then proceed to give a brief outline of android development in the context of NFC. Finally we will end this chapter by describing development for desktop with USB devices and Libnfc[1].

3.1 Equipment

This section describes the hardware equipment and software environment used in the project.

3.1.1 Programming environment

All Android applications have been developed and debugged using the Eclipse[2] IDE. This IDE is recommended by Google Android[3] for Android application development. We have used the Android SDK Tools, Revision 11, which includes the Android SDK and the Android Development Tools (ADT).

Development in C has also been done using Eclipse with its C/C++ plug-in, and GNU C Compiler from GCC version 4.5.2. We have used Libnfc development branch Revision 1120 for this development.

[1]http://libnfc.org/

[2]http://eclipse.org/

[3]http://developer.android.com/

3.1.2 Handset

All applications developed on the Android platform have been tested on a Samsung Nexus S. The Nexus S runs Android version 2.3.4, and is the first modern NFC-enabled smart phone.

Figure 3.1: Nexus S

3.1.3 USB Device

The desktop NFC development have been on a SCM Microsystems SCL3711. This is a USB device with a PN532 chip, which supports emulation of all NFC Forum type tags and many other non-standard tags. It also supports read/write communication with the most common tag types.

It has been tested, and is supported by Libnfc.

Figure 3.2: Dingle Dongle

3.1.4 NFC Tags

The NFC cards and stickers used are MiFare Ultralight tags. These fulfil the NFC Forum type 2 tag specification[1]. This makes them rewritable, and possible for write protection. The size of our tags are 64 bytes, giving a total payload size of 48 bytes.

3.2 Android

Android application development is driven around creating *Activities*. An activity is a single, focused entity that the user can interact with. The Activity class takes care of creating a window for placing UIs.

When starting an activity, the parent activity must start it with an *Intent*. The Intents doesn't need to contain anything, but can be filled with data that the newly started activity can use as input data.

Intents can also be generated when the device tries to handle user input events, such as NFC-tags coming inside range of detection. The intent is then handled by the Activity Manager, or the foreground Activity, depending on the application.

When developing NFC applications, the application must be programmed to handle the new NFC Intents. This is done using the Android API package *android.nfc*. We suggest the Android documentation for more details[3].

3.2.1 EasyNfc

To create a simpler development process we have developed EasyNfc. EasyNfc is an Eclipse library project that simplifies the process of developing NFC applications on the Android platform. The library project contains extendible Activity classes for reading and writing to tags, P2P, and Bluetooth pairing of Android devices.

It elegantly reduces the code required when creating NFC applications by hiding much of the boiler plate code from the application developer. This is done by creating EasyNfc as an extension of the original Activity-class. The application developer can simply extend one of the EasyNfc classes to gain access to a powerful set of NFC tools.

The library was written in a relatively late stage of the research period, and is thus used for some, but not all of our prototype applications. To help future Android NFC Application developers, EasyNFC is delivered with solid documentation.

3.3 Desktop

Libnfc[4] is an open source SDK and programmers API for working with NFC on traditional desktop computers. We have used it to work with SCL3711, our dedicated NFC USB device developed by SCM Microsystems[5]. As libnfc is a low level API, our focus has been on developing higher level APIs for use in future application development processes.

The SCL3711 USB device makes it possible to read external tags and to be read by external readers, in other words it can both read and emulate tags. Our API include functionality for emulating and reading Type 2 Mifare Ultralight tags. It also simplifies the process of creating and parsing NDEF messages. The implementation of these APIs are done in C, but we have implemented a python wrapper to make it even easier to work with our device in future projects.

The advantage of emulating an NFC tag with an USB device compared to using a static tag is the possibility for varying responses. As the tag is emulated in software the application can give different responses dependent on context. The tag data may vary over several parameters, like what time of the day it is or which user is reading it. This may prove useful in many areas, including permission control and security.

[4]http://libnfc.org/

[5]http://www.scmmicro.com/

Chapter 4

Applications

This section will describe our prototype applications and how they were implemented.

4.1 Tagger

In the infancy of our exploration of the NFC technology on the Android platform we decided to implement the simplest NFC application we could think of. The goal was to learn the basics of Android programming and understand NFCs place on the Android platform. We intended to build on the experiences gained, both theoretically and practically, to build more advanced applications later on in the project.

Our first prototype application is named *Tagger*. *Tagger* is a standard NFC tag reader and writer. The application utilizes the NFC capabilities of the mobile device to read Type 2[1] NDEF[2] tags and view them in plain text to the user. The reader is capable of reading tags with Plain Text[4] and URI[5] payload records. *Tagger* can also write to tags, but this is limited to Plain Text records[4]. As *Tagger* is implemented complying to the Type 2 and NDEF standards, the tags written with the application can be read by all standard compliant readers.

The work done on *Tagger* is the foundation on which we build the other Android applications.

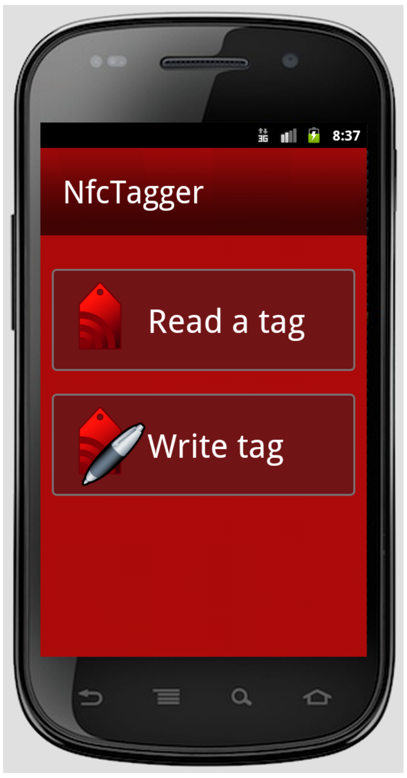

Figure 4.1: Main screen of Tagger

4.2 NfcPresenter

NfcPresenter is an Android application that simplifies the user-computer interaction by creating a more intuitive way to present slide-show presentations.

Traditionally, starting a presentation is a cumbersome process. The person doing the presentation usually has two alternatives for getting started; either connecting his own laptop to the projector, or transfer the presentation with a USB memory stick to a local computer which is already connected. Both routines are unnecessary tedious. Haven't we all experienced trouble when connecting a laptop to a new projector? The alternative, transferring the presentation to a USB memory stick, connecting the USB device to the local computer, navigating to the relevant file, opening it, and starting the presentation, isn't any more tempting.

With *NfcPresenter* we aim to create a smooth and painless presentation experience by utilizing NFC technology. Starting a presentation can be done by selecting the appropriate presentation file on the mobile device, and touching it to a NFC tag present at the presentation location. Simple as that.

When the presentation has started, the Android device serves as a remote control. With a swipe you can show the next or previous slide. In addition each slide is shown on the screen of the mobile device for better control.

NfcPresenter also lets the user download the currently active presentation to a mobile device. In other words, a spectator of the presentation can use the application to download the presentation to his own device by scanning a tag. This is a useful feature for distributing presentation material.

As explained, *NfcPresenter* utilizes NFC to collect the url from a tag present at presentation location. The url directs to a server running on the local presentation computer, which is already connected to the projector. The application uses the recently acquired hostname to upload the presentation file to the server. The server starts the presentation, and can now answer to other commands to control it remotely.

In this prototype implementation of the application the server only accepts PDF uploads. The viewer is implemented using Pygame[1], after first having used pyPdf[2] to convert the PDF into Pygame-compliant images. Communication is done over HTTP, using web.py[3] server-side to handle the connections.

We predict that the way presenting is handled in NfcPresenter is a pointer to how presenting will be handled in the future. Including NfcPresenter-like server functionality into popular end-user products like Microsoft Powerpoint or Apple Keynote would be a simple and efficient way to distribute this kind of features. We believe it would be a great enhancement to the current presenting experience.

[1]http://pygame.org

[2]http://pybrary.net/pyPdf/

[3]http://webpy.org

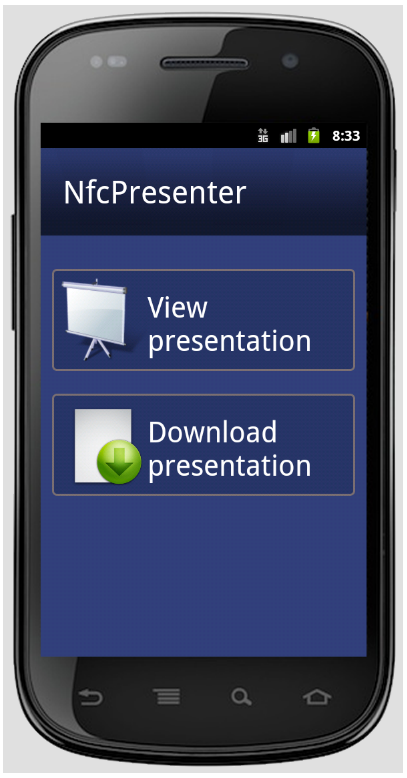

Figure 4.2: The startup screen of NfcPresenter

4.3 NfcSafari

NfcSafari is a tour guide application that aims to help people discover, experience and remember interesting locations and spots. The general application area is giving tourists visiting unknown cities and locations a helping hand by guiding them to interesting sightseeing spots. The overall goal of creating the application is to explore how NFC can be utilized in pervasive applications.

The application initiates the tour by using the current location of the user to find nearby spots. The user is given a choice between the three closest spots, and the user will start the *Safari* by choosing one of them. The next screen shows a map, guiding the user from his current location to the spot he chose.

When the user finds a spot, he needs to touch the nearby NFC tag to register his presence. By reading the ID of the spot, the application knows what web service to contact to download a description of the spot. If the user is logged in, we can now choose to share information about the spot through Facebook. It is also possible to give the spot a rating from 1 to 5 stars. When the user is happy with the experience he can choose to continue the tour. The next spot is shown on the map, and the process starts from the top. All visited spots are stored in a local database.

In *NfcSafari* all sightseeing spots are predefined on the server. Spots are grouped together by *Safari*. The idea is to let spots in close vicinity to each other, or spots with the same theme, to be placed in the same tour. All tours are defined in XML files, where each file describes a single *Safari*. The XML below shows how a *Safari* is defined:

```
<SafariRoute id="1" name="University Safari">
    <Spots>
        <Spot id="1"
              name="Gandhi bust"
              longtitude="18975154"
              latitude="69681120"/>
        </Spot>
    </Spots>
</SafariRoute>
```

Each Safari has an ID and a name, in addition to having a list of spots with names, IDs and location. When the server starts it parses all the *Safari* xmls for spots and locations, which in turn is used when the Android application requests the closest spot. When this happens, the server computes the distance from the user to all active spots (from all Safaris) and returns a list of the three closest of them. When the user chooses one of them, the *Safari* it belongs to is chosen as the active tour, and the corresponding XML i downloaded. When the user is done with the current spot, the next spot is the next spot in the current Safari.

Communication between the server and the Android client is done over HTTP, and the server is implemented using web.py[4]. It responds with both static and dynamic content. NfcSafari uses a substantial amount of data traffic, where the map data is the biggest contributor.

NfcSafari could easily be modified to create other pervasive applications. An idea we have discussed is a "New Student" application for the University of Troms, where the spots are important locations on campus selected on the basis of what you study, and the application guides you through them all.

4.4 PartyShare

The goal of PartyShare is to bring multimedia from handheld devices to stationary devices in a seamless manner. It consists of a client running on

[4]http://webpy.org

a NFC enabled Android device, and a server running on a computer. The application lets you choose to share music or images from the android device to the computer in real-time. The transfer is done simply by touching the android device to the stationary device. As a result, the images are viewed and the music is played immediatly.

In principle, PartyShare uses the HTTP protocol to transfer the multimedia from the android device to the computer. This implies that the application could've been built without NFC technology. However, configuring such an application would be tedious. For instance, moving from one space to another would require the user to enter a new IP-address in order to interact with the current stationary device.

We all know that entering an IP-address or hostname on a handheld device is both time consuming and cumbersome. By taken advantage of NFC technology we can remove the configuration step completely, and as a result get a smart and intuitive sharing application. The trick is to store the relevant hostname in the NFC tag. This way the android application can read the relevant adress directly from the relevant device, and get immediate feedback as his image is viewed, or the track is played.

The server is implemented in python, using a share of open source python modules. The web server utilizes web.py[5], the ImageViewer which displays the latest image is created using Pygame[6], and the music sharing is implemented using a self made libspotify-wrapper (spotipy). The web server responds to HTTP post requests on the uris "/upload/image" and "/upload/track".

On the client side we use internal android activities to handle the camera and the gallery to choose image, and the spotify meta data api to choose music. The music sharing works by transfering a Spotify URI which is then used by spotipy to play the song.

[5]webpy.org

[6]pygame.org

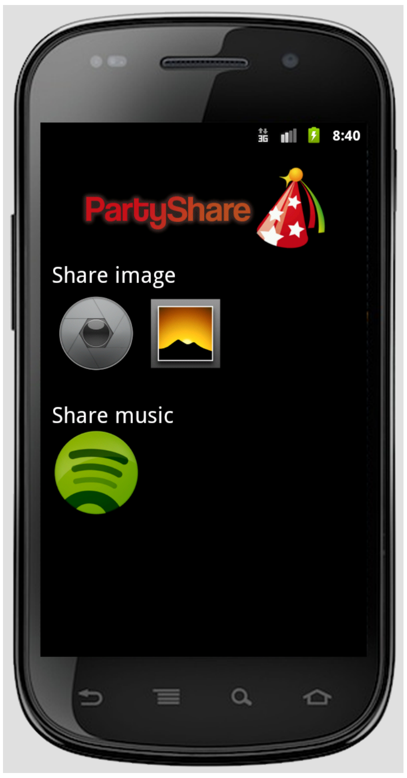

Figure 4.3: Main menu of PartyShare

4.5 Are You The One?

Are You The One is a quirky "love tester" that calculates the compatibility between two persons. The goal of the application is to illustrate how NFC can be used in social scenarios. The application works by having the user enters his or her name, and then continue to touch another handset with the same application. As a result their "love match", or compatibility, is visualized on screen as a percentage.

The compatibility is calculated deterministically using their names, and obviously has no basis in reality. Nevertheless, it is a fun example of a small application which delivers something new by incorporating NFC as a vital component.

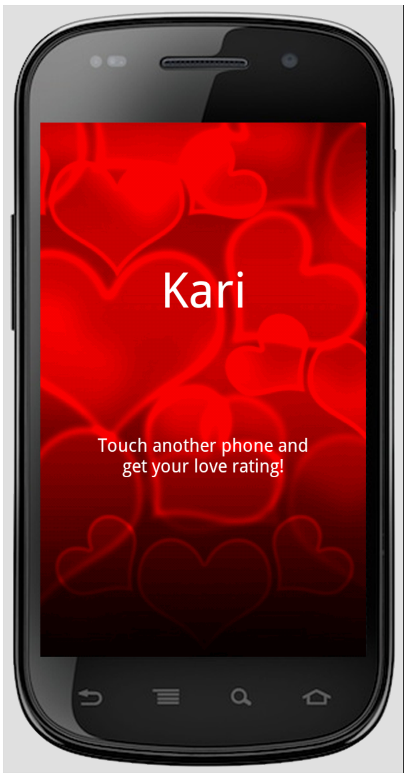

Figure 4.4: Screenshot from Are You The One?

4.6 Collector

Collector is a prototype of a concept that will increase, or add social interaction between mobile gamers.

The idea is that instead of putting NFC in the front seat, it can instead be the cherry on top, making games that are generally not made for social interaction social.

In the prototyped example, the setting is a Pokmon like game, where the goal is to collect many collectables and use them in you game. With NFC, these collectables can be shared among friends giving a social experience to an otherwise not social game.

4.7 BluetoothChat

This application is mostly an example application to show the use of Bluetooth connections from NFC. It is a simple chat client which enables users to set up a connection by touching each others phone and chat via Bluetooth.

Chapter 5

Discussion

Here we will discuss the current state of NFC from our experience; Its possibilities and limitations. We will also mention what role we think NFC will have for future applications.

5.1 Current State

NFC is a relatively new technology, and does not have the infrastructure of other older technologies. Because of this, there is less support available for development of NFC services and applications. However, as the popularity of the technology is rising, we conjecture that it will not stay this way for long.

Developing NFC application on Android works quite well. Although Androids NFC implementations can be unstable during testing, they have provided a thorough implementation and documentation of both new and legacy NFC standards.

Developing with Libnfc is harder. Not only because its a low level API, but also because the documentation is a bit weak, and unforgiving to newcomers. To be able to use Libnfc, and to fully understand it, you will have to do a lot of of research and understand a range of technical specifications. With few other alternatives, this is a challenge.

The MiFare Ultralight tags we used in this project had some limitations worth noting. The mobile device was unable to read the tags if they were to close to other metal. This is definitely worth remembering when placing tag stickers.

5.2 Future Work

In general, there is a lot more to explore regarding NFC technology. While we have investigated possible application and service areas, traditional NFC research areas, such as mobile payment, security and authentication, has been left untouched. This was done consciously, as other NFC City partners are currently exploring these fields to great extent.

While we have implemented a number of prototype applications utilizing NFC technology on mobile platforms, we have not developed any desktop prototypes using the USB devices. An advantage of the USB device is the possibility to create tags dependent on context. Although this is achievable by utilizing static tags combined with URLs and web access, USB may be a better option as it delivers the same context dependent experience without relying on a WI-FI or 3G/EDGE connection for the mobile device.

5.3 Evaluation

NFC is a relatively fresh technology that is conjectured to explode in the global market in the near future. It is useful in many areas, such as payment, security, configuration, and social interaction. It acts as a link between the physical and virtual world, giving us a simpler and more intuitive way of sharing information.

One of the properties of NFC is that context is implicitly defined. This stems from the very short range of the tags. This can create simplifications in many aspects of development, such as security, and configuration.

Apart from payment, we feel that NFC for mobile devices are most useful where it can simplify the user-interactions of services or applications. The best example is setting up connections for WI-FI or Bluetooth, as exemplified in BluetoothChat (4.6). A task that can be tedious is replaced by an intuitive gesture that connects touching devices. We think that NFC is not bringing something entirely new to the table, but it can certainly create more convenience to already established applications and services.

A good example of this is *NfcSafari* (4.2), which does not accomplish something we could not have done using QR codes or the GPS more extensively. However, the GPS would limit us to outdoor spots, and scanning QR codes is slightly less convenient as it takes more time and can be hard in the dark. If this application was to be launched in todays market it would have been wise to incorporate both QR and NFC.

Applications like *Are You The One?* (4.4) illustrates how an apparently exhausted concept gets a fresh feel by incorporating NFC. The act of touching two devices to get a love match (instead of typing in the names on one device) give a greater connection between the two participants.

NfcPresenter (4.1) underlines the convenience of NFC enabled applications. To configure the upload and download hostname by a physical touch instead of entering it manually makes a mobile presentation application go from something a few people will use to something everyone will want to have.

PartyShare (4.3) illustrates how NFC can enhance a social event by giving everyone the ability to participate. This type of crowd sourcing has been possible before NFC, but the threshold to join the fun has been significantly lowered by making the act of sharing something come down to a simple touch.

Chapter 6

Conclusion

During our work on the NFC-City project we have explored NFC technology both in design of new services and in the development of actual applications.

In the process of gathering experience in NFC technology for the University of Tromsø, we have also created a baseline for future research with the implementations and the documentation produced.

Bibliography

[1] *Type 2 Tag Operation Technical Specification*, NFC Forum Std., July 2007. [Online]. Available: http://www.nfc-forum.org

[2] *NFC Data Exchange Format (NDEF) Technical Specification*, NFC Forum Std., July 2006. [Online]. Available: http://www.nfc-forum.org/

[3] "Android documentation." [Online]. Available: http://developer.android.com/

[4] *Text Record Type Definition Technical Specification*, NFC Forum Std., July 2006. [Online]. Available: http://www.nfc-forum.org/

[5] *URI Record Type Definition Technical Specification*, NFC Forum Std., July 2006. [Online]. Available: http://www.nfc-forum.org/