Masterarbeit

Entwurf und Implementierung einer NFC-Anwendung für den Einsatz im eCommerce-Bereich

Ralph Weissnegger

Institut für Technische Informatik
Technische Universität Graz
Vorstand: Univ.-Prof. Dipl.-Inform. Dr.sc.ETH Kay Uwe Römer

Begutachter: Ass.Prof. Dipl.-Ing. Dr.techn. Christian Steger
Betreuer: Ass.Prof. Dipl.-Ing. Dr.techn. Christian Steger

Graz, im Oktober 2013

Kurzfassung

In unserer heutigen Gesellschaft gewinnt das Mobiltelefon immer mehr an Bedeutung. Die Zahl der Nutzer die ein Smartphone im täglichen Leben gebrauchen, steigt rapide. Dies führt dazu, dass der Umgang mit diesen neuen Geräten für den Nutzer immer leichter wird und die Grenzen zwischen Technologie und Alltag verschwimmen. Eine Technologie, die bei Mobiltelefonen immer weiter in den Vordergrund rückt, ist NFC (Near Field Communication). NFC hat sich schon in einigen Anwendungsgebieten, wie im Bezahlvorgang, im Gesundheitswesen oder in der Paarung von Geräten etabliert. In dieser Arbeit kommt NFC zum Einlesen von passiven Tags, über sogenannte Smart Poster mit einem mobilen Endgerät zur Anwendung. Besonders auf die Erweiterung von Applikationen durch NFC wird hier Fokus genommen. Anwendungsgebiete von Applikationen die NFC unterstützen, reichen von Logistikanwendungen bis hin zu Online-Spielen. Durch die Verbindung von Applikation und physischen Kontakt, werden den Benutzern neue Erfahrungen ermöglicht. Gerade im eCommerce-Bereich ergeben sich dadurch neue Perspektiven. Bei der Implementierung von NFC in ein bestehendes System müssen aber einige Designrichtlinien und Protokolle beachtet werden. Auch der Gebrauch der Applikation ohne NFC-fähiges Mobiltelefon soll betrachtet werden.

Diese Masterarbeit beschäftigt sich mit dem Design und der Implementierung von NFC in ein bestehendes System. Aufbauend auf den Erkenntnissen aus dem theoretischen Teil wurde ein Prototyp entwickelt, der auf einem Bonusprogramm inklusive Schnitzeljagd aufbaut. Dieser spielerische Ansatz zeigt, dass der Benutzer sehr leicht mit der physischen Welt interagieren kann. Dadurch wird die Benutzerfreundlichkeit erhöht und die Kundenbindung gestärkt.

Abstract

Smartphones are becoming widely popular in contemporary society. The number of people using smartphones in daily life has reached a significant level which leads to much faster technology adoption as ever before. Also the handling of these new gadgets is getting easier. One of the most extended technologies used in mobile phones nowadays is NFC (Near Field Communication). This technology has set new standards in the application field of payment, health care or pairing of devices. In this thesis, NFC is used to scan passive Tags contactless with a mobile phone over short distances from a so called smart poster. Especially the leveraging of existing applications by adding NFC functionality for various use-cases is focused. The range of applications that benefit from NFC integration reaches from business centric logistics to eCommerce applications and online games. By connecting the application to real world interactions the user-engagement can be highly improved. Throughout the implementation of NFC in existing applications, various design patterns and protocols have to be considered. Moreover, the usage of mobile phones without NFC-support must also be guaranteed.

This thesis discusses the design and implementation of NFC into an existing system. With the findings of the theoretical part a proof of concept has been implemented which extends a loyalty program application with a real world paper chase game. This gamification prototype shows, that customers can easily perform various interactions with NFC, which rely on their physical presence at a given place. This improves the usability and strengthens the customer's loyalty.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz,am ....

(Unterschrift)

Danksagung

Diese Diplomarbeit wurde am Institut für Technische Informatik an der Technischen Universität Graz in Zusammenarbeit mit den Firmen Bergfex und Blue-Tomato durchgeführt wurde.

Als erstes möchte ich mich für die professionelle Unterstützung von Herrn Ass.Prof. Dipl.-Ing. Dr.techn. Steger bedanken. Ohne seine Mitwirkung hätte die Zusammenarbeit zwischen dem Institut und den Firmen nicht so hervorragend funktioniert. Seine jahrelange Erfahrung in dieser Thematik war für den Erfolg des Projektes ausschlaggebend. Danke an die Firma Bergfex, die mir ermöglichte an meiner Masterarbeit in ihrem Unternehmen zu arbeiten und vor allem danke für ihre großartige Unterstützung über den gesamten Zeitraum. Besonders ein großes Dankeschön der Firma Blue-Tomato, im Speziellen Mag. Kreimer und Mag.Zezula und allen Mitarbeitern, die mir mit Rat und Tat zur Seite gestanden sind. Weiteres möchte ich mich bei der FFG (Forschungsförderungsgesellschaft) bedanken, durch die das Projekt gefördert wurde.

Das Jahr 2013 war ein Jahr mit Höhen und Tiefen. Das prägende Erlebnis am 10. Februar und die darauf folgenden Monate waren für meine Umgebung und mich sicher keine leichte Zeit. Ohne die Hilfe meiner Familie, Freunde/Freundin und Verwandten würde ich nicht heute, am 7.Oktober, die letzten Zeilen meiner Masterarbeit tippen. Ich darf mich glücklich schätzen Eltern zu haben, die sich um jemanden so sorgen wie meine. Danke an meine Freundin Catrin die in dieser schlimmen Zeit für mich da war und mir Kraft gegeben hat das Erlebte durchzustehen. Vielen Dank an meine Freunde, die mir wieder den Weg ins normale Leben gezeigt haben. Erst in Zeiten wie diesen, weiß man es zu schätzen, Menschen zu kennen wie euch.

Ich möchte mich nochmal bei meinen Eltern bedanken, die mir das Studium in Graz ermöglicht haben. Ich bin froh diesen Schritt gewagt zu haben und bereue die Zeit in Graz in keinster Weise. Die vielen Erlebnisse und Personen die ich kennengelernt habe, von denen manche zu Freunde geworden sind, möchte ich nicht missen.

Graz, September 2013 Ralph Weissnegger

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

NFC (Near Field Communication) ist keine Technologie die erst seit kurzem auf dem Markt vertreten ist. Sie leitet sich von der allbekannten Technologie RFID (Radio-Frequency Identification) ab, die mittlerweile in vielen Bereichen Einzug gehalten hat. NFC ist bereits seit dem Jahre 2006 in mobilen Geräten vorhanden, doch erst jetzt gewinnt sie immer mehr an Bedeutung. Nicht zuletzt durch die Möglichkeit NFC in den Bezahlvorgang mittels Kredit- oder Bankomatkarte zu integrieren und der immer häufiger auftretenden Mobiltelefone mit NFC-Lesefunktion. Neben den kontaktlosen Bezahlfunktionen gibt es noch weitere Anwendungsmöglichkeiten, wie der Verkauf von Fahrkarten, im Gesundheitswesen, bei Zutrittskontrollen oder als Ergänzung zu anderen Funktechnologien. Sogenannte SmartPoster bieten eine perfekte Möglichkeit, Technologie und Marketing zu vereinen, um Informationen an den Kunden weiterzugeben. Passanten müssen oft längere Zeit an Orten verweilen, wie Bushaltestellen oder Warteschlangen an Kassen. Hier bietet es sich besonders an, intelligente Poster aufzustellen, um den Benutzer auf Aktionen aufmerksam zu machen. Auch in Verbindung mit Bonusprogrammen und Kundekarten bietet NFC eine hervorragende Ergänzung zur Kundenbindung.

1.1 Motivation

Die Firma *Bergfex* [Ber13] besitzt neben ihrer Webseite auch seit längerem eine mobile Android-Applikation, die von sehr vielen ihrer Kunden benutzt wird. Mit dieser Applikation können Benutzer Schneeberichte aber auch Informationen über Skigebiete oder Webkameras abrufen. Gerade in Skigebieten, wo mit Wartezeiten am Lift und an Kassen zu rechnen ist, müssen Informationen schnell abrufbar sein, um den Ablauf nicht zu stören. Dem Benutzer soll die Bedienung der Applikation daher so leicht wie möglich gemacht werden. Neben der Benutzerfreundlichkeit gibt es noch weitere Aspekte, wie Kundenbindung die durch diesen Ansatz gestärkt werden soll. In diesem Falle bestand die Anforderung, ein Bonuspunkteprogramm zu schaffen um die Attraktivität der Applikation zu erhöhen, den Kunden zu motivieren und ihn dadurch stärker an die Firma zu binden. *Bergfex* ist ein jahrelanger Affiliate-Partner

von *Blue-Tomato* (eine internationale Vertriebsfirma für Sportgeräte und Bekleidung mit stationären Shops und Onlineverkauf) und verkauft deren Artikel über ihr Portal [bt]. *Blue-Tomato* ist der ideale Partner um ein Bonuspunkteprogramm zu realisieren, da die Firma eine große Community besitzt und seit längerem in Richtung Omni-Channel geht. Mittels der immer populärer werdenden NFC-Technologie sollen diese Ziele erreicht werden und genau deshalb bestand die Anforderung, NFC in ein bestehendes System zu Integrieren und deren Vielzahl an Anwendungsmöglichkeiten aufzuzeigen.

1.2 Inhaltsübersicht

Diese Arbeit gliedert sich in 2 Teile, einem theoretischen Teil und der praktischen Implementierung des Prototypen. Der theoretische Teil besteht aus Kapitel 2, 3 und 4 und der praktische Teil startet mit Kapitel 5, beginnend mit der Definition der Anforderungen und endet mit dem fertigen Prototypen. Die Inhalte der Kapitel werden hier noch einmal genauer im Detail erklärt.

Die Einleitung wird noch mit einem historischen Rückblick über die Entstehung von NFC ergänzt. Zusätzlich werden noch andere gängige Technologien mit NFC verglichen.

Kapitel 2 geht näher in die Thematik NFC ein und erklärt, welche Betriebsarten und Anwendungsgebiete diese Technologie besitzt. Weiteres wird erklärt, wer sich hinter dem NFC-Forum verbirgt und welche Standards durch sie eingeführt wurden.

Kapitel 3 erklärt die verschiedenen Datenformate und wie das NDEF-Format aufgebaut ist. Anschließend wird auf die verschiedenen Typen eingegangen, die ein NFC-Tag haben kann. Speziell der Anwendungsfall Smart Poster wird in diesem Kapitel behandelt.

Kapitel 4 beschäftigt sich mit Arbeiten, die in diesem Bereich bereits veröffentlicht wurden. Die Publikationen geben Kenntnisse darüber, in welchen Gebieten NFC bereits Anwendung findet, dass Smart Poster bereits unseren Alltag erreicht haben und einen Ansatz, wie NFC in ein bestehendes Ticketing-System integriert werden kann.

Kapitel 5 beginnt mit der Beschreibung des Problems und die Anforderungen werden im Detail erklärt. Aufbauend auf den bisher erlangten Kenntnissen wurde ein Konzept für den Prototypen entwickelt und das Design bzw. die Architektur wird beschrieben.

Kapitel 6 beschäftigt sich mit der Implementierung des Prototypen im Detail und

welche Punkte beachtet werden müssen, um NFC in ein bestehendes System integrieren zu können. Anschließend wird auf Basis von Webzugriffen von *bergfex.com* analysiert, wieviel NFC-fähige Mobiltelefone sich bereits auf dem Markt befinden.

Kapitel 7 beschreibt den fertigen Prototypen, welche Geräte und Tags dafür verwendet wurden und die Entwicklungsumgebung. Ein Anwendungsfall wird aufgezeigt, bei dem der Benutzer 3 blaue Tomaten über NFC sammeln muss und diese über ein Bonuspunkteprogramm bei Blue-Tomato einlösen kann.

1.3 Historisches

NFC (Near Field Communication) ist eine kontaktlose Technologie um Daten über eine kurze Distanz zu übertragen. Sie entstand durch die Kooperation von NXP und Sony im Jahre 2002. NFC kann als eine Erweiterung von zwei jahrelang erprobten Technologien angesehen werden, RFID (Radio Frequency Identification) und Smartcard. Um NFC wirklich zu verstehen, müssen beide Technologien näher betrachtet werden.

1.3.1 RFID Technologie

RFID ist eine Technologie die Radiowellen verwendet, um Daten zwischen einem RFID Lesegerät und einem RFID-Tag zu übertragen. Generell werden diese Tags an Objekten angebracht, um sie zu überwachen oder identifizieren. Die Datenrate hängt von der verwendeten Frequenz und dem magnetischen Feld ab. Das erste Mal wurden RFID Tags im zweiten Weltkrieg verwendet, um eigene Fahrzeuge oder Flugzeuge von feindlichen zu unterscheiden. Erst 1970 fanden sie den Weg in unsere Kaufhäuser zur Warensicherung. Ab dem Jahre 1980 wurden RFID-Tags in der Tieridentifizierung und in Mautsystemen eingesetzt, 10 Jahre später in Zutrittskontrollsystemen, Skipässen, elektronischen Tickets, Tankkarten oder Bezahlkarten. Heutzutage finden wir RFID Systeme in vielen weiteren Anwendungen wie Kreditkarten, Reisepässe, in der Lagerverwaltung oder Zutrittssystemen wieder.

Auf einem RFID-Tag sind ICs (Integrated Circuit) angebracht, auf denen kleine Programme laufen oder wenige Daten abgespeichert sind. Man unterscheidet zwischen passiven und aktiven Tags. Ein passiver Tag besteht aus einem IC und einer Antenne. Er besitzt keine eigene Stromversorgung, die Energie wird zur Gänze vom Lesegerät induziert. Die Reichweite von passiven Tags reicht von etwa 10 Zentimetern bis zu einigen Metern. Aktive Tags haben hingegen eine eigene Stromversorgung, dies hat den Vorteil, dass sie eine höhere Reichweite haben und die Verbindung seltener abbricht. Ein Nachteil von RFID Tags verglichen mit Barcodes, ist der relativ hohe Preis. Deshalb ist es noch nicht kosteneffizient jedes Produkt in Geschäften mit einem RFID Tag zu bestücken [LR10, Kapitel1].

1.3.2 SmartCard

Eine SmartCard ist eine Karte mit einem integrierten Schaltkreis, der über einen Speicher verfügt und in den meisten Fällen über einen Mikrokontroller. Es gibt drei unterschiedliche Gruppen von SmartCards: kontaktbehaftete Karten, kontaktlose Karten und Hybrid Modelle. Sie verfügen über keine eigene Stromversorgung sondern werden über das Lesegerät oder einem externen Gerät mit Energie versorgt. Kontaktbehaftete Karten kommunizieren mit dem Lesegerät über den direkten Kontakt und versorgen sich so mit Energie. Sie verfügen über ein Mikro-Modul welches einen einzigen Silikon-IC mit integrierten Speicher und Mikroprozessor besitzt. Wenn die kontaktbehaftete Karte in das Lesegerät eingeführt wird, entsteht eine elektrische Verbindung über die leitenden Kontaktplatten der Karte. So können Befehle, Daten und Statusinformationen über die Karte abgerufen werden. Bei kontaktlosen Karten kommt es erst zu einer Verbindung, wenn sich die Karte in einer bestimmten Reichweite befindet. Dies hat den Vorteil, dass sich so die Sicherheit der Übertragung und der Datenfluss erhöht. Wenn sich eine Karte dem elektromagnetischem Feld eines Lesegeräts nähert, wird über die Antenne eine Spannung induziert. Durch die übertragene Energie, kann die Karte auf die Anfrage des Lesegeräts antworten. Die 3 häufigsten SmartCards sind:

- ISO/IEC 10536 Close Coupling Smart Cards (Reichweite bis zu 1 cm)
- ISO/IEC 14443 Proximity Coupling Smart Cards (Reichweite bis zu 10 cm)
- ISO/IEC 15693 Vicinity Coupling Smart Cards (Reichweite bis zu 1 m)

Die populärste Chipkarte ist die Proximity Coupling Smart Card, da sie in vielen Gebieten, vom Gesundheitswesen bis Unterhaltung zur Anwendung kommt. Nur wenige Proximity Cards haben den ISO/IEC 14443 Standard, den auch NFC verwendet. Dazu gehören MIFARE, Calypso und FeliCa.

Das erste Mal tauchten Chipkarten als bargeldloses Bezahlmedium durch das Kreditunternehmen Diners Club auf. Es folgten Mastercard und Visa, die, die Verbreitung der Kreditkarte für den Zahlungsverkehr vorantrieben. Um die Sicherheit zu erhöhen, wurde zuerst ein Magnetstreifen auf der Rückseite von Chipkarten verwendet, welcher digitalisierte Daten zur Datenweiterverarbeitung beinhaltete. Da aber die Daten auf einem Magnetstreifen beliebig oft kopiert werden können, wurden sichere Daten wie PIN (Personal Identification Number) niemals darauf gespeichert. Die Überprüfung einer PIN erfolgt über eine Online-Verbindung. Die ersten Karten mit integriertem Schaltkreis waren Telefonkarten. Sie setzten sich gegen Karten mit Magnetstreifen und optischer Speicherung durch. Aber erst durch die Integration von Mikroprozessoren und Kryptoprozessoren, konnten Chipkarten den Durchbruch schaffen. 1995 wurde in Österreich die erste Karte mit QUICK-Funktion eingeführt und war somit das erste Land mit flächendeckenden elektronischen Geldbörsensystem [RE96, S. 14-18].

1.3.3 NFC

NFC ist ein internationaler Übertragungsstandard, der den kontaktlosen Austausch von Daten über kurze Distanzen ermöglicht. Wie RFID, arbeitet NFC auf der gleichen und freien Frequenz von 13,56 MHz, einem ISO/IEC 18000-3 Standard. Die Übertragungsgeschwindigkeit beträgt 106, 212 und 424 kbps. NFC hat drei verschiedene Betriebsarten, Peer-to-peer, Reader/Writer und CardEmulation. Im Reader/Writer oder CardEmulation Modus gibt es immer eine passive (Target) und eine aktive (Initiator) Komponente. Die Aktive versorgt die Passive mit Energie, um eine Verbindung herzustellen. Passive Elemente gibt es in vielen verschiedenen Formen wie Stickers, Tags, Karten oder andere. Beim Peer-to-peer-Modus haben beide Komponenten ihre eigene Stromversorgung. Ein großer Vorteil liegt in der kurzen Distanz zwischen den Geräten, da dadurch das Signal schlecht manipuliert oder abgehört werden kann. Ein weiterer Vorteil von NFC ist die automatische Paarung von Geräten, so können Applikationen auf Smartphones sofort gestartet werden [For11].

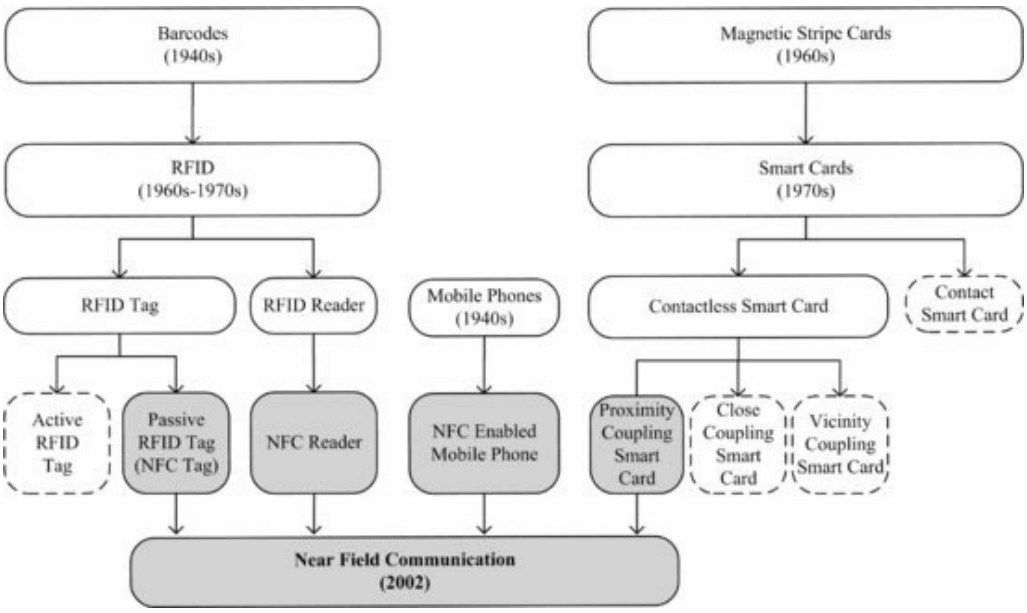

Abbildung 1.1: Entwicklung von NFC [COO12, Chapter 2.1]

Historie

1983	Das erste Patent welches mit RFID in Verbindung gebracht wurde, stammte von Charles Walton [Wal83]
1995	Tauchte das erste mal der Begriff *Wallet paying* auf
2004	Gründung des NFC Forums durch Nokia, NXP und Sony [NF03]
2006	Spezifikationen der Standards durch das NFC Forum [NF06b]
2006	Spezifikationen von SmartPoster- Records [NF06a]
2006	Das Nokia 6131 war das erste NFC-Mobiltelefon [Nok13]
2006	Feldversuch des NFC Research Lab Hagenberg
2009	Ersten Peer-to-peer Standards um Kontakte zu übermitteln
2010	Erste Android SmartPhone mit NFC
2010	In der Stadt Nizza in Frankreich wurde das erste NFC-Pilotprojekt getestet. Den Einwohnern wurden NFC-Mobiltelefone und Bankomatkarten zu Verfügung gestellt [inv10]
2011	In Australien entstand das erste NFC-Marketing-Unternehmen (Tapit Media)
2011	Google präsentierte auf der Google I/O das erste Spiel mit NFC und wie es möglich ist, Kontakte zu senden, Videos oder Applikationen starten[Goo]
2011	Symbian unterstützt ab sofort NFC [nfc11]
2011	Research in Motion ist das erste Unternehmen, die Geräte herausbringt die von MasterCard zertifiziert sind(PayPass) [mob11]
2012	Eine Restaurantkette und ein Mobilfunkanbieter erschaffen die erste SmartPoster Kampagne. Eine eigene Applikation auf einem Smartphone reagiert durch die Annäherung an das Plakat
2012	Sony liefert jedes Xperia P Smartphone mit Tags (SmartTags) aus [Son13] Mit diesen Tags können Context- basierte Aktionen ausgeführt werden.
2012	Samsung präsentiert TecTile. TecTile sind MIFARE NFC- Stickers die durch eine Android- Applikation gelesen und beschrieben werden können [Sam12]
2013	Samsung und Visa werden Partner um das Thema *mobile payment* voranzutreiben [inv]

Tabelle 1.1: Historische Entwicklung

1.4 Vergleich zu anderen drahtlosen Funktechniken

NFC ist selber eine Funktechnologie, kann aber auch andere Funktechnologien ergänzen, indem es Setup-Daten auf Tags speichert, um damit automatisch eine Verbindung zu Bluetooth oder Wi-Fi aufzubauen. Erst wenn beide Geräte in einer Reichweite von wenigen Zentimetern sind, wird die Kommunikation ermöglicht. Abb. 1.2 und 1.3 zeigen, wie sich NFC im Vergleich zu anderen Funktechnologien in Bezug auf Reichweite und Datenrate verhält [For11].

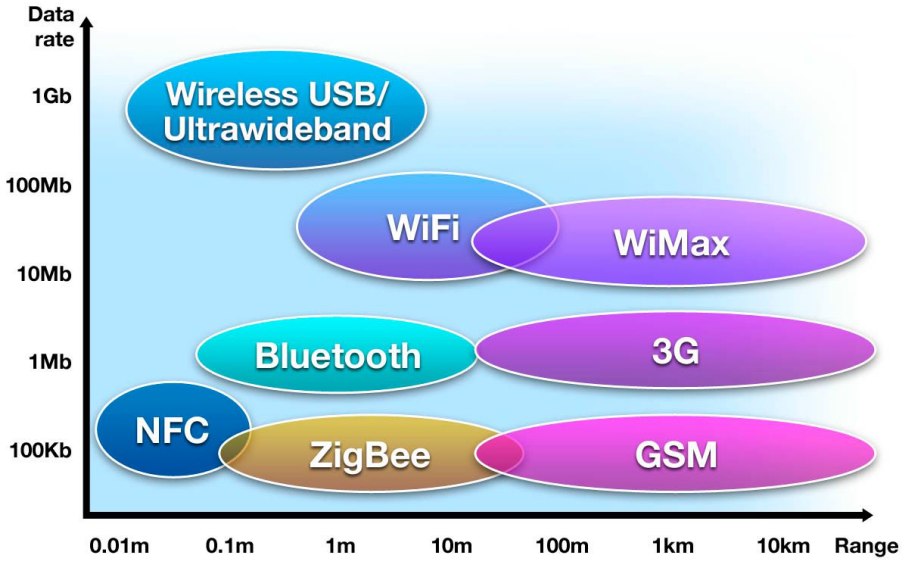

Abbildung 1.2: Vergleich von Datenrate und Reichweite

	BLE	NFC	Zigbee	Ant+	6LoWPAN
Bit-Rate (kbit/s)	200	424	250	1000	20 - 250
Frequency (MHz)	2400-2500	13,56	868/915/2400	2400	868/915/2400
Range (meter)	50 – 100	0,2	10 – 100	1 – 30	
Network Type	Peer to peer / Star	Point to Point	Mesh/Star/Peer to peer	Mesh/Star/Peer to peer	Mesh/Star/Peer to peer
Network Standard	802.15.1	ISO 13157	802.15.4		802.15.4

Abbildung 1.3: Kennzahlen

Kapitel 2

NFC

2.1 NFC Forum

NFC hat es in den letzten Jahren zu einer weit verbreiteten Technologie geschafft, die in vielen Gebieten Anklang findet. Damit NFC noch weiter aufblüht, braucht es eine stärkere Zusammenarbeit von kundenorientierten Firmen. In diesem Sinne wurde das NFC Forum 2004 aus der Zusammenarbeit von nur 3 Firmen gegründet. Mittlerweile zählen zu diesem Forum mehr als 175 Mitglieder aus den verschiedensten Ländern des gesamten Globus. Die Mitglieder kommen aus den verschiedensten Sparten wie:

- Chiphersteller
- Mobiltelefonhersteller
- SIM-Karten-Hersteller
- Banken
- Kreditkartenunternehmen
- Mobilfunkbetreiber
- Forschungsinstitute
- und Non-Profit Organisationen

Durch die Zusammenschließung des Forums wurde es möglich, die Technologie im Bereich Mobile und Personal Computer voranzutreiben und ein Framework zu schaffen, welches komplex, kompatibel und sicher ist. Die Ziele des Forums sind:

- Die Entwicklung von Standards der NFC-Technologie, um eine modulare Architektur für mobile NFC-Geräte und Protokolle zu garantieren.

- Die Förderung und Entwicklung von Produkten, die die NFC Forum Spezifikationen einhalten.
- Erfüllung der Ansprüche an NFC-Produkte und die Zusammenarbeit zwischen den Unternehmen.
- Weitergabe von Informationen an Konsumenten, Endverbraucher und Unternehmen.

Durch die Möglichkeit NFC auch mit anderen Wireless Technologien zu verwenden, erschließen sich neue Anwendungen, wie Zugriff zu Kontent und Diensten, Bezahlvorgänge für Produkte, Zugriffsbeschränkungen in Gebäuden und das Abrufen von digitalen Diensten überall und zu jeder Zeit.

Die Mitglieder des Forums erstrecken sich über die ganze Industrie, von Elektronik bis zur Wirtschaft und darüber hinaus. Die Erfahrungen aus den verschiedenen Sparten wurden dazu genutzt, um NFC noch weiter zu verbessern. Durch die gute Zusammenarbeit wurden Protokolle und Spezifikationen erschaffen, durch die viele Anwendungen profitieren. Alle Entscheidungen wie z.B. Standardisierung von Datenstrukturen und Formaten, Entwicklung gemeinsamer Protokolle oder Spezifikationen von geräteunabhängigen Services werden zuvor von allen stimmberechtigten Mitgliedern geprüft.

Das NFC Forum besteht aus mehreren Ausschüssen und Arbeitsgruppen, die sich mehrmals pro Jahr beim NFC Forum Meeting treffen. Wie in Abb. 2.1 zu sehen ist, gibt es 3 Ausschüsse, das Marketing-, Compliance- und Technical Committee.

Im **Marketing Committee** werden die Maßnahmen für Marketing und externe Kommunikation beschlossen. Ihre Aufgabe ist, den Markt über die neuesten Fortschritte zu informieren, die Vorteile von NFC aufzuzeigen und neue Mitglieder in das NFC Forum aufzunehmen. Es besteht aus 3 Gruppen: Events, Marketing und Unterstützung der Entwickler- Gemeinde.

Das **Compliance Committee** hat die Aufgabe ein Produkt zu erschaffen, welches ein Logo besitzt, das eine gute Wiedererkennung hat und dem Kunden Sicherheit vermittelt. Es ist in 3 Arbeitsgruppen unterteilt: Compliance Program, Minimum-Level of Interoperability and Testing.

Das **Technical Committee** besteht weitläufig aus technischen Arbeitsgruppen, in denen technische Details diskutiert werden. Hier entstehen die NFC-Standards für Geräte und Dienste.

Das **Privacy Advisory Council** ist für die Schulung von NFC Forum-Mitgliedern,

Entwicklern und Datenschutzbeauftragten zuständig. Auch die Öffentlichkeit hinsichtlich Datenschutz und deren Sicherheitslücken fällt in ihren Bereich.[For07c].

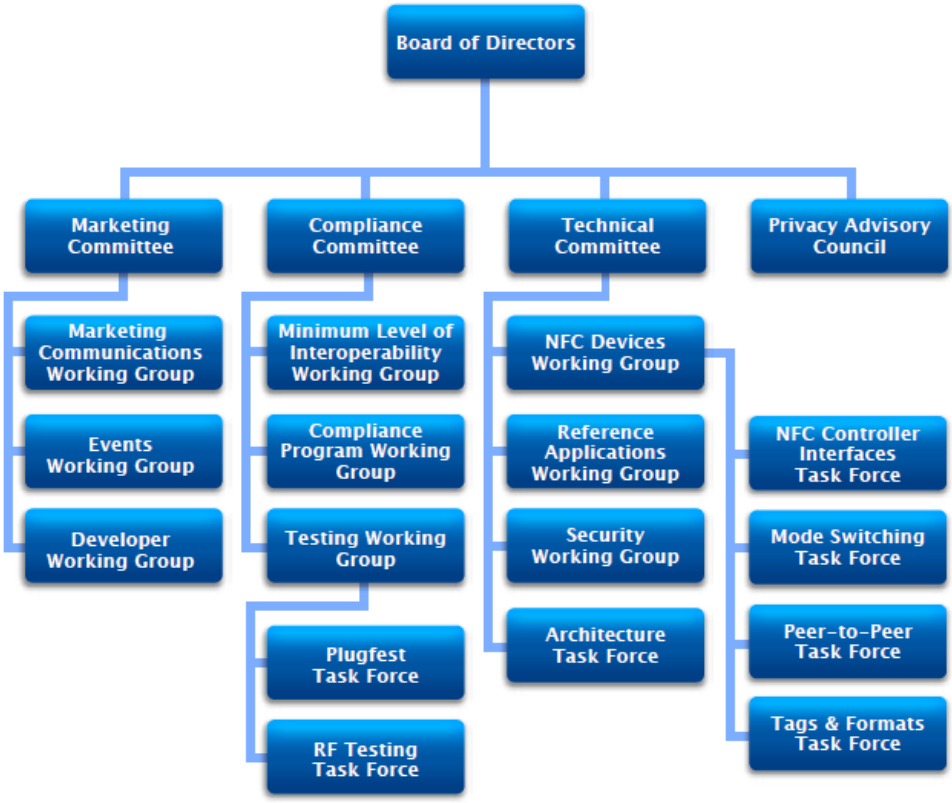

Abbildung 2.1: Ausschüsse und Arbeitsgruppen des NFC Forum

Um ein Mitglied im NFC Forum zu werden muss ein Mitgliedsbeitrag eingezahlt werden. Die Mitgliedschaften werden in 5 Kategorien unterteilt:

- Sponsor
- Principal
- Associate
- Implementer
- Non-Profit-Organisation

Je nach Höhe der Beiträge werden die Stimmrechte an die Firmen vergeben. Als Sponsor hat man die meisten Rechte und einen Sitz im Board of Directors. Auch als Principal hat man Stimmrechte und kann abstimmen, wenn Standards verabschiedet werden. Alle anderen haben kein Stimmrecht, dürfen aber in den Arbeitsgruppen mitarbeiten [LR10, S. 88-89].

2.2 NFC Standards

Um neue NFC Geräte zu produzieren oder Software zu entwickeln, müssen die NFC-Standards eingehalten werden. Die Standards stellen sicher, dass jede Art von Near Field Technologie mit NFC kompatiblen Geräten interagieren kann und es auch in Zukunft keine Probleme mit neueren Geräten geben wird. Zwei wichtige Standards sind in diesem Zusammenhang NFCIP-1 und NFCIP-2.

NFCIP-1 (Near Field Communication Interface and Protocol-1) ist eine Kombination aus existierenden Standards wie MIFARE (ISO/IEC 14443 Typ A) und FeliCa (JIS X 63 19-4). Hier sind drei Übertragungsgeschwindigkeiten definiert: 106kbit/s (MIFARE), 212 bzw. 424 kbit/s (FeliCa). NFCIP-1 sowie ISO/IEC 14443 und ISO/IEC 15693 Standards definieren das RF Interface, Initialisierung, Antikollision und das drahtlose Übetragungsprotokoll für induktiv gekoppelte RFID-Systeme im Frequenzbereich von 13,56 MHz [Int13].

NFCIP-2 (Near Field Communication Interface and Protocol-2) erweitert NFC um 2 Kommunikationsmodi: PCD (Proximity Coupling Device) und VCD (Vicinity Coupling Device). PCD hat den ISO/IEC Standard 14443 und VCD den Standard 15693. NFCIP-2 definiert den Mechanismus, der erkennt, welcher der 3 Modi ausgewählt werden muss. In Abb. 2.2 wird der Zusammenhang graphisch dargestellt [Int13].

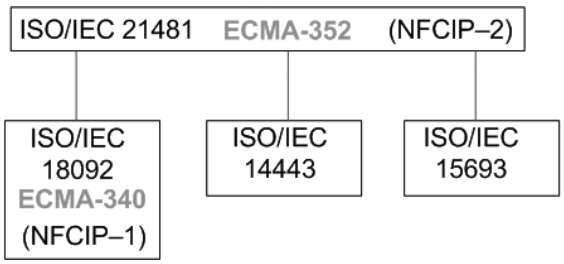

Abbildung 2.2: NFC Standards

Klassische RFID-Systeme haben einen aktiven und oder mehrere passive Komponenten. NFC-Geräte unterscheiden sich hier, denn sie können sowohl die Rolle der steuernden Komponente (Initiator) als auch der gesteuerten Komponente (Target) übernehmen. Bei der Energie und Datenübertragung gibt es auch Unterschiede. Im ersten Fall erzeugt der Initiator das Trägersignal für die Übertragung, dieser wird passiver Modus genannt und ist kompatibel zu MIFARE und FeliCa. Im zweiten Fall wechseln sich die Komponenten (Initator, Träger) ab, um das Trägersignal für die Daten und Energieübertragung zu erzeugen. Dieser Modus von NFCIP-1 wird aktiver Modus genannt.

Die NFC Forum Spezifikationen bauen auf dem NFC Protokoll Stapel auf, der es Geräten möglich macht, Tags zu lesen und zu beschreiben oder auch eine **Peer-to-Peer** Verbindung aufzubauen. Der NFC Forum Protokoll Stapel unterstützt aber auch die Funktion **Card Emulation**, die es NFC Geräten erlaubt, als gewöhnliche SmartCard zu agieren.

Die NFC Forum Spezifikationen sind analog zum OSI Modell, Schichte 2, 4 und darüber hinaus. Schichte 1 und 2 (analoge und digitale Schichte) werden durch die Analogue and Digital Protokoll Spezifikationen bzw. durch die Activity Spezifikationen unterstützt. Diese unterstützen die Peer-to-Peer und Reader/Writer Betriebsart. Die Logical Link Control Protocol (LLCP) Spezifikationen stellen ein Standard Interface für NFC Applikationen bereit. Wie die Daten für die Übetragung zu einem NFC-Gerät formatiert werden müssen, werden in den NDEF und RTD Spezifikationen festgelegt.

Seit 2010 stellt das NFC Forum ein Zertifizierungs-Programm [For07d] bereit, bei dem Geräte und Software mit Hilfe von Test-Cases geprüft werden, ob sie den NFC-Standards entsprechen. Wenn das Gerät die Prüfungen besteht, erhält es das Near Field Communication Zertifizierungs Zeichen wie in Abb. 2.3.

Abbildung 2.3: NFC Zertifikation

2.3 Betriebsarten

Das einzigartige an NFC-Geräten ist, dass sie 3 verschiedene Betriebsarten unterstützen:

- Reader/Writer
- Peer-to-Peer
- Card Emulation

Diese 3 Betriebsarten basieren auf den ISO/IEC 18092 NFC IP-1 und ISO/IEC 14443 Smartcard Standards.

Im Reader/Writer Modus ist es möglich, mit sogenannten NFC-Tags zu interagieren. Es können Daten auf den Chip geschrieben, aber auch von ihm gelesen werden. Die Daten sollten in einem bestimmten Format auf dem Tag abgespeichert werden und zwar in einem vom NFC Forum definierten Format genannt NDEF. NDEF (NFC Data Exchange Format) ist ein leichtes Binärformat, welches die Daten in einer bestimmten Form kapselt. Im Reader/Writer Modus können Geräte aber auch mit anderen Karten kommunizieren, die kein NDEF Format unterstützen. Im Peer-to-Peer-Modus können zwei NFC-Geräte miteinander kommunizieren. Hier wird ein NFC-spezifisches Protokoll für den Datenaustausch eingesetzt. Der Card Emulation Betriebsmodus ermöglicht es dem Gerät, wie eine SmartCard zu agieren. Zusätzlich kann die emulierte Karte auch von jedem RFID Lesegerät erkannt werden, um auf diesem Wege Daten zu lesen. Die Grundlagen für den Reader/Writer und Card Emulation Modus bilden die NFCIP-1 und RFID-Normen.

2.3.1 Reader/Writer Mode

Der Reader/Writer Modus besteht aus 2 Betriebsarten, dem Reader und dem Writer Modus. Im Reader Modus agiert das NFC-fähige Smartphone wie ein Lesegerät und kann die Daten, wenn es die NFC Forum-Architektur unterstützt, im NDEF Format von Tags einlesen. Wie die Kommunikation funktioniert wird, in Abb. 2.4 beschrieben. NFC Forum-Tags sind passive Transponder auf denen die Daten in einem vorgegeben Format abgespeichert sind. Ob es zu einer Peer-to-Peer oder Reader/Writer Verbindung kommt, wird bei der Anitkollision entschieden. Je nach dem wie der Kommunikationspartner auf das Signal antwortet, wird einer der beiden Betriebsarten gewählt. Ohne die vom NFCIP-1 (ISO/IEC 18092) Standard vordefinierten Bitübertragunsschichten und Antikollisionsverfahren würde keine Übertragung zustande kommen. Die Daten auf einer SmartCard müssen aber nicht zwingend im NDEF Format vorliegen, sondern das NFC-Gerät unterstützt auch andere Formate wie z.B. den RFID-Standard. Um das ganze Spektrum abzudecken, müssten die Tags den Standard ISO/IEC 14443 unterstützen.

Im Writer Modus kann das NFC-Gerät Daten auf den NFC-Tag schreiben. Wenn sich schon Daten auf dem Tag befinden und nicht schreibgeschützt sind, können diese modifiziert oder überschrieben werden [COO12, Chapter 4.3].

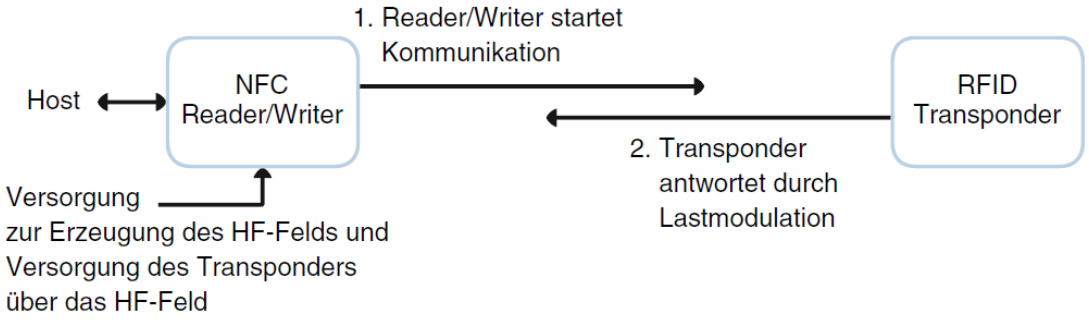

Abbildung 2.4: Reader/ Writer Modus [LR10, S. 120-128]

Der Reader/Writer Modus hat sehr viele nützliche Anwendungen. Besonders im SmartPhone-Bereich durch Unterstützung von Audio/ Video oder auch dem Internet, kann der Informationsfluss unterstützt werden. Je nachdem, welchen Typ der Record (RDT) auf dem Tag hat, können bestimmte Aktionen eingeleitet werden, wie z.B. das Öffnen einer bestimmten Anwendung oder das Weiterleiten an eine Webseite. Über diese Schnittstellen kann der User dann direkt mit dem Service Provider interagieren und sich Tickets für den nächsten Film oder den dazugehörigen Trailer ansehen. Diese Art von Tags werden meistens in Plakate eingebunden und heißen SmartPosters. Ein weiteres Anwendungsgebiet ist die Unterstützung von drahtlosen Kommunikationsschnittstellen. So können Zugangsdaten von Bluetooth oder WLAN am Tag abgespeichert werden um so eine Verbindung aufzubauen, ohne zusätzliche Daten einzugeben.

2.3.2 Peer-to-Peer

Der Peer-to-peer Modus erlaubt es zwei NFC Geräten Informationen wie Kontakte, Textnachrichten, oder andere, über eine drahtlose Verbindung auszutauschen. Dieser Modus hat zwei Standardisierungen, NFCIP-1 und LLCP. Bei Peer-to-peer gibt es zwei Kommunikationsmodi, einen Aktiven und einen Passiven. Wer der Initiator und wer der Target ist, wird am Anfang der Kommunikation nach der Handshake-Methode entschieden. Am Anfang der Kommunikation sind beide Geräte im aktiven Modus. Die Daten werden über einen bidirektionalen Half-Duplex-Kanal gesendet, d.h. wenn ein Gerät sendet, muss das andere Gerät zuhören, bis derjenige seine Daten übermittelt hat. Erst dann darf dass zuhörende Gerät seine Daten senden. Die maximale Datenrate beträgt hier 424 kbps. Die Bitübertragunsschicht von NFCIP-1 bildet die unterste Schicht des Protokollstapels wie in Abb. 2.6 zu sehen ist. Darüber

liegt der Media-Access-Control-Layer (MAC) und der Logical-Link-Control-Layer (LLC). Der MAC-Layer ist für den Verbindungsaufbau, die Initialisierung und dem Datenaustausch zuständig und ist nach ISO/IEC 18092 normiert. Das Logical-Link-Control-Protokoll (LLCP) ermöglicht die Kommunikation von 2 NFC- Geräten und deren Datenaustausch und ist durch das NFC Forum spezifiziert. Zusammen bilden der MAC-Layer und LLC-Layer die Datensicherungsschicht [COO12, Chapter 4.4].

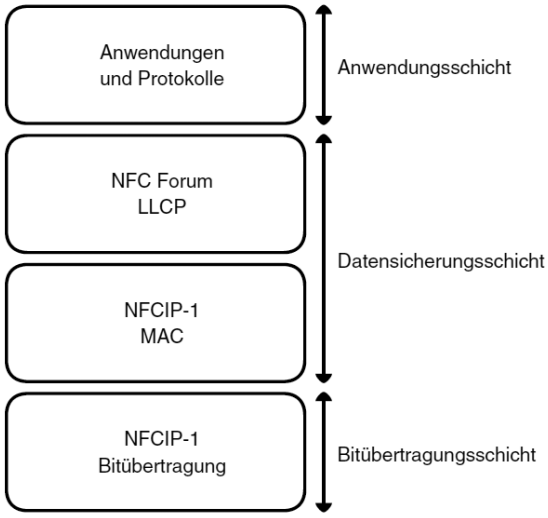

Abbildung 2.5: Peer-to-peer Protokollstapel [LR10, S. 120-128]

Der Peer-to-peer Modus schafft es eine Verbindung zwischen zwei NFC fähigen Mobiltelefonen aufzubauen, ohne das zusätzliche Einbinden eines Service Providers. Ein Anwendungsfall wäre der Austausch von privaten Daten. Informationen wie Kontaktdaten könnten über eine sichere Verbindung übertragen werden. Da es sich hier um eine Kommunikation handelt, deren Reichweite von ein paar Zentimetern nicht überschritten wird, fühlt sich der Benutzer relativ sicher seine Daten zu übertragen. Auch das Tauschen von Geldbeträgen über eine mobile Geldtasche oder von mobilen Tickets wäre ein gutes Anwendungsgebiet.

Passiver Kommunikationsmodus

Beim passiven Kommunikationsmodus hat der Initiator einen höheren Energieverbrauch, da er während der ganzen Kommuikation das hochfrequente Trägersignal erzeugt. Nicht nur um eine Anfrage an den Target zu senden, sondern auch für die Datenübertragung hat der Initiator ein höheres Energieaufkommen. Mittels ASK (Amplitude Shift Keying) werden die Daten auf das Trägersignal moduliert, der Target antwortet durch das Lastmodulationsverfahren. Bei diesem Verfahren hat der Target einen viel geringeren Energieverbrauch. Es muss nur Energie für die Verarbeitung des Peer-to-peer-Protokollstapels aufgewendet werden. Durch Entnahme

weiterer Energie aus dem Trägersignal, wäre sogar eine gänzliche Versorgung durch den Initiator möglich.

Aktiver Kommunikationsmodus

Beim aktiven Kommunikationsmodus hat der Initiator einen geringeren Energieaufwand wie beim passiven Modus, da der Energieverbrauch auf beide Kommunikationspartner gleichmäßig verteilt wird. Initiator und Target müssen Energie für ihr Sendesignal aufwenden. Der Initiator, wie auch der Target erzeugen das gleiche hochfrequente Trägersignal. Gleich wie bei dem passiven Kommunikationsmodus, werden die Daten mittels ASK auf das Trägersignal moduliert. Jedes NFC-Gerät ist standardmäßig als Target konfiguriert. Mittels Collision Avoidance kann getestet werden, ob ein externes Trägersignal in der näheren Umgebung vorhanden ist. Wenn innerhalb der Reichweite keine weiteren Signale erkannt wurden, schaltet das NFC-Gerät sein eigenes Sendesignal ein. Wird ein anderes Trägersignal gefunden, schaltet das Gerät in den Target und wartet auf Befehle des Initiators.

LLCP (Logical Link Control Protocol)

Bei einer typischen Peer-to-peer-Übertragung, gehen alle Befehle auf der MAC-Schicht vom Initator aus. Egal ob die Übertragung vom Initiator zum Target, oder vom Target zum Initiator ausgeht, der Initiator leitet einen Data Exchange Protocol Request ein. LLCP ermöglicht einen weiteren Kommunikationsmodus genannt Asynchronous Balanced Mode (ABM). Durch diesen Modus werden beide Service- Endpunkte gleichberechtigt. Mit ABM dürfen beide Endpunkte jederzeit initialisieren, überwachen und Informationen senden. LLCP spezifiziert weiteres wie NFC-Geräte innerhalb einer bestimmten Reichweite, kompatible LLCP-Geräte erkennen, eine LLCP-Verbindung aufbauen,überwachen und bei Gegebenheit wieder abbrechen. LLCP ist auch in der Lage mehrere Instanzen eines höheren Protokolls gleichzeitig unterzubringen (Protocol Multiplexing) [For07a].

Aufbau

LLCP kann in folgende logische Komponenten aufgeteilt werden (Abb. 2.6):

- MAC Mapping
- Link Management
- Connection-oriented Transport
- Connectionless Transport

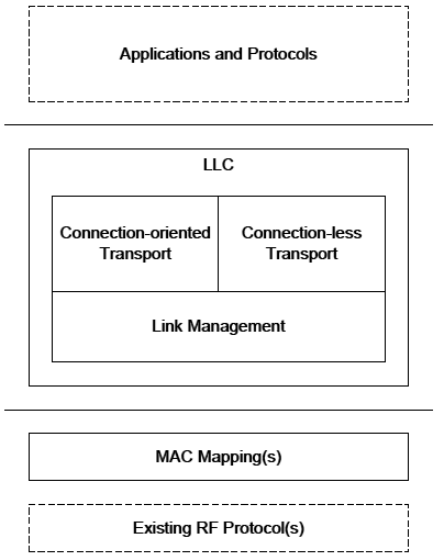

Abbildung 2.6: Logische Komponenten [For07a]

Verbindungsloser Transport

Bei dem verbindungslosen Transport ist es nicht notwendig eine Verbindung auf oder abzubauen. Dieser Modus kann verwendet werden, wenn darüber liegende Protokoll-Schichten eine Datenflusskontrolle implementiert haben und sich nicht auf den Link-Layer Flow Control verlassen müssen. Das bedeutet, es müssen weder die Paketreihenfolge eingehalten werden, noch kann garantiert werden, dass alle Pakete ankommen.

Verbindungsbasierter Transport

Der verbindungsbasierte Transport ermöglicht eine sichere Übertragung von Datenpaketen, bei dem jedes einzelne Paket durch eine eindeutige Sequenznummer gekennzeichnet ist. Der Datentransfer wird durch das Sliding Window Protokoll kontrolliert. Dieses Protokoll wird auch bei einer TCP Verbindung eingesetzt um möglichst effizient Daten zu übertragen. Beim verbingungsbasierten Transport ist es verpflichtend, eine Verbindung aufzubauen und eine Zuweisung von Ressourcen, solange die Verbindung bestehen bleibt. Das Übertragungsprotokoll garantiert das Einhalten der Paketreihenfolge und den Empfang aller übertragenen Pakete.

2.3.3 Card Emulation

Die Card Emulation-Betriebsart ermöglicht es, einem NFC-fähigem Mobiltelefon, wie eine kontaktlose SmartCard zu fungieren. Diese emulierte SmartCard kann dann von RFID- Lesegeräten erkannt werden. Mobile Endgeräte können auch mehrere SmartCard-Applikationen auf einer Karte speichern. Ein Beispiel von emulierten Karten wäre die Bankomatkarte, Kreditkarte oder Kundenkarte. Inwieweit der Funktionsumfang von NFC-Geräten im Card Emulation-Modus geht, ist vom NFC-Chipsatz abhängig. Der Card Emulation-Modus generiert kein eigenes RF- Feld sondern die Energie für die Datenübertragung wird vom NFC Lesegerät aufgebracht. Die Grundlage für die Übertragung bildet das Digital Protocol. Daher gibt es 3 Schnittstellen die den Card Emulation-Modus unterstützen, ISO/IEC 14443 Type A/B und FeliCa.

Es gibt zwei Möglichkeiten eine kontaktlose SmartCard zu emulieren, über die Software oder über ein *Secure Element* (SE). Ein Secure Element ist ein Mikrochip wie er auch bei SmartCards zum Einsatz kommt (Bankomatkarte, Kreditkarte). Dieser Mikrochip wird zum Speichern und Abarbeiten von sicheren Daten verwendet und umfasst meistens eine Java Card. Der Card Emulation Modus hat einige Vorteile gegenüber dem Peer-to-peer Modus, da er keinen mehrstufigen Protokollstapel für die Kommunikation oder zusätzliche Software benötigt. Dadurch kann auch der Chip bei ausgeschaltetem Mobiltelefon oder leerem Akku ausgelesen werden. Die Versorgung des Mirkochips übernimmt gänzlich das NFC-Lesegerät [COO12, Chapter 4.5].

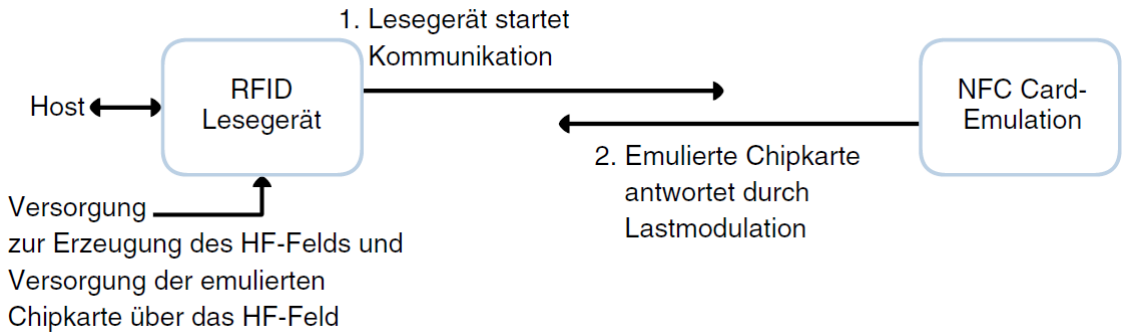

Abbildung 2.7: Card-Emulation-Modus [LR10, S. 120-128]

2.4 Anwendungsgebiete

2.4.1 Transit and Ticketing

Eines der größten Anwendungsgebiete von NFC ist der Verkauf von Tickets im Nah- und Fernverkehr. Kontaktlose Tickets ermöglichen eine schnellere Abwicklung beim Kauf von Tickets bei öffentlichen Verkehrsmitteln oder in Parkgaragen. Der Benutzer kann sich mit einem NFC-fähigen Mobiltelefon ein Ticket beim Schalter herunterladen oder das Ticket wird *over the air*(OTA) übertragen. Durch das Annähern des Smartphones an das Lesegerät wird das Ticket entwertet oder der Benutzer bekommt Zutritt z.B. zur U-Bahn. Der Kauf von Tickets wird dadurch beschleunigt, denn der Reisende verliert keine Zeit an langen Warteschlangen. Auch der Preis von Tickets wird durch den Gebrauch von NFC reduziert, da das Ticket keine Druck- und Papierkosten verursacht. Der Benutzer hat den Vorteil mehrere Tickets auf seinem Smartphone zu speichern und kann der Applikation das Managen seiner Tickets überlassen [For11].

2.4.2 Smart Poster

In der heutigen Zeit sind wir im Alltag umgeben von Werbung, angefangen von Displays in Auslagen bis hin zu Plakaten an Wänden. Manchmal ist aber die Information die wir über diese Medien erhalten unzureichend und der Benutzer würde gerne mehr über ein Produkt in der Auslage oder Preis und Verfügbarkeit eines Konzert-Tickets erfahren. Genau hier liegt die Stärke von NFC, denn es ist leicht für den Serviceanbieter wie auch den Benutzer, Informationen bereitzustellen und zu konsumieren. Ein SmartPoster kann in vielen Formen vorkommen, wie Plakate, Werbetafeln, Magazin- Seiten oder auch als dreidimensionale Objekte. Die Information wird als NDEF Nachricht auf dem NFC-Tag gespeichert und in das Smart Poster integriert. Durch das standardisierte N-Mark Zeichen ist es für jeden ersichtlich, wo sich der Chip befindet und in welchen Bereich man das Lesegerät halten soll. Der Benutzer entscheidet so selbständig welche Information er abrufen will. Auf einem NFC-Tag kann eine große Palette an Informationen gespeichert sein:

- Aktueller Klingelton einer Music-Band
- Fotos oder Hintergrundbilder für das Mobiltelefon
- Link zu einer Webseite
- Link zu einem Video
- Record zum Öffnen einer bestimmten Applikation am Mobiltelefon
- Link zu einer Applikation
- Telefonnummer der Firma für weitere Informationen

- Senden einer SMS an Freunde
- Verbreiten von Informationen über soziale Netzwerke

Der Serviceanbieter ist auch in der Lage, den Content den er bereitstellt, jederzeit zu aktualisieren und aktuell zu halten. Dies kann über ein Backend-System geschehen, indem nur ein Link zu einer statischen Webseite auf dem Tag gespeichert ist. Ein weiterer Vorteil ist es, dass Informationen über den Benutzer gleich in die Darstellung miteinfließen können. Durch die Einstellungen am SmartPhone kann der Text in der richtigen Sprache angezeigt werden. SmartPoster sind relativ günstig im Vergleich zu LCD Displays und es kann in großen Mengen implementiert werden. Das Update der Informationen über ein Backend-System gestaltet sich relativ leicht und Smart Poster sind flexibel in Größe und Anwendung. Ein NFC Tag kann an fast allen Objekten angebracht werden [For07b].

2.4.3 Payment

Da immer mehr Personen ein Smartphone mit sich tragen, liegt es nahe, den Bezahlvorgang auch als Funktion in Mobiltelefone zu integrieren. Bestehende Bezahlarten wie SMS oder WAP haben sich nicht durchgesetzt, da die Benutzerfreundlichkeit nicht gegeben war und sich längere Warteschlangen am Point of Sale gebildet haben. Durch den Gebrauch von NFC-fähigen Mobiltelfonen ist es möglich geworden, Bezahlvorgänge in Kaufhäusern oder Supermärkten durchzuführen. Die auf dem Smartphone installierte Applikation kann Informationen über Kreditkarten, Bankomatkarten oder Guthabenkarten speichern. Durch das Annähern des Gerätes an den Terminal/Lesegerät wird die Zahlung bestätigt. Dies hat den Vorteil, dass für die Zahlung keine Karten mehr in das Lesegerät eingeführt oder ein PIN abgefragt werden muss. Der Kunde spart sich dadurch Zeit an den Kassen und es entstehen keine langen Warteschlangen. Durch den NFC-Standard ist jedes NFC-fähige Mobiltelefon mit den Lesegeräten kompatibel. Da die Applikation passwortgeschützt und die Reichweite relativ gering ist, ist der Bezahlvorgang mit dem Smartphone sehr sicher. Falls ein Gerät abhanden kommen sollte, kann es aber auch von der Ferne gesperrt werden [For07c].

2.4.4 NFC und andere Funktechniken

Obwohl sich das Mobiltelefon in den letzten Jahren großartig entwickelt hat, war dies bei den Kommunikationstechnologien nicht der Fall. 13 Jahre nach dem Bluetooth entwickelt wurde, ist sie zwar eine Funktechnologie die von vielen verwendet wird, aber nicht allgegenwärtig. Sie wurde geschaffen um Kabelverbindungen von mobilen Geräten abzulösen und hat eine Datenrate von bis zu 3 Mbit/s. Ein Nachteil von Bluetooth ist es aber, dass bevor die Daten übertragen werden können, eine sicher Verbindung aufgebaut werden muss. Hierzu verwendet Bluetooth ein Passwort oder vergleicht die Verifikationsnummer. Auch das Auffinden von Geräten ist nicht

sehr benutzerfreundlich, da es meistens mehrere Bluetooth-Geräte in der näheren Umgebung gibt. Hier kommt NFC ins Spiel, denn es können die Einstellungen der Verbindung auf einem Tag gespeichert werden, um so zwei Bluetooth Geräte auf einem schnellen Wege zu paaren. Anwendungen finden sich hier im Verschicken von Fotos einer Kamera an einen Drucker, oder das Konfigurieren eines Heimnetzwerks über Wi-Fi [For07c].

2.4.5 Zutrittskontrolle

NFC kann auch für die Zutrittskontrolle von Gebäuden, Räumen oder Hotels verwendet werden. Es wird kein Schlüssel mehr benötigt, sondern durch das Annähern des NFC-Mobiltelefons oder einer Chipkarte an das Lesegerät, können die Rechte überprüft und ein Zutritt genehmigt werden. Ein Vorteil von diesem System ist es, dass mehrere verschiedene Schlüssel auf einer Karte gespeichert werden können. Auch die Administration kann über ein Backend-System laufen und jederzeit geändert werden, falls ein Benutzer andere Zutrittsrechte benötigt. Systeme mit MIFARE oder LEGIC werden schon seit einiger Zeit in Gebäuden zur Zutrittskontrolle verwendet. Da MIFARE zu NFC kompatibel ist und in vielen NFC-Mobiltelefonen bereits integriert ist, kann das Mobiltefon leicht in den Prozess integriert werden. Ein Mobiltelefon ist im Gegensatz von mehreren Karten leichter zu handhaben und wird weniger oft vergessen. Für die Zutrittskontrolle kommen verschiedene Sicherheitsstufen zur Anwendung. Sie reichen vom einfachen Auslesen von Seriennummern bis hin zu hochsicheren kryptografischen Verschlüsselungen [For07c].

2.4.6 Gesundheitswesen

Auch im medizinischen Bereich gibt es sehr viele Anwendungen für NFC. Die Vorteile liegen hier im Erfassen von automatisierten Daten oder zur Identifikation von Patienten. Mithilfe von Mobiltelefonen werden z.B. Daten über den Gesundheitszustand von Diabetes- oder Bluthochdruckpatienten regelmäßig an den Arzt oder die Ärztin weitergeleitet, dieser wiederum reagiert sofort bei auffälligen Werten. Weiteres können Patienteninformationen abgerufen werden, um richtige Medikation zu verabreichen oder um Unverträglichkeit von Medikamenten auszuschließen. Die Daten liegen auf zentralen Servern und werden von einem Lesegerät oder einer Applikation am Smartphone abgerufen. Die Werte können vom Patienten selber oder einer Hilfskraft bzw. direkt vom Arzt eingegeben werden [COO12, Chapter 1].

Kapitel 3

Datenformate

3.1 NFC Data Exchange Format (NDEF)

Die NFC Data Exchange Format (NDEF) Spezifikation definieren ein Format für den Aufbau einer Datenstruktur, um Informationen zwischen zwei NFC Forum-Geräten oder einem NFC Forum-Gerät und einem NFC Forum-Tag auszutauschen. Es ist ein einfaches binäres Datenformat, welches verwendet wird, um ein oder mehrere anwendungsspezifische Daten in ein Nachrichten-Konstrukt zu kapseln. Eine NDEF Message besteht aus einem oder mehreren NDEF Records. Jeder Record besitzt eine Payload mit einer Größe von bis zu 232-1 Oktett. Um eine längere Payload zu erreichen können Records zusammengekettet werden [LR10, S. 120-128].

Die Payload eines NDEF Records wird über 3 Parameter beschrieben :

- Payload Length
- Payload Type
- Payload Identifier

Die Sicherheit der Übertragung ist Aufgabe der darunterliegenden Übertragungsprotokollebenen bzw. der darüberliegenden Anwendungsebene und nicht des NDEF Formats.

3.1.1 NDEF Message

Eine NDEF Message besteht aus einem oder mehreren NDEF Records. Im ersten Record einer Message ist die Flag MB(Message Begin) und beim letzten Record ist die Flag ME(Message End) gesetzt (Abb. 3.1). Wenn in einer Message die MB- und ME- Flag gesetzt sind, besteht die Message nur aus einem Record, dies ist das kleinstmögliche Datenformat (Abb. 3.2).

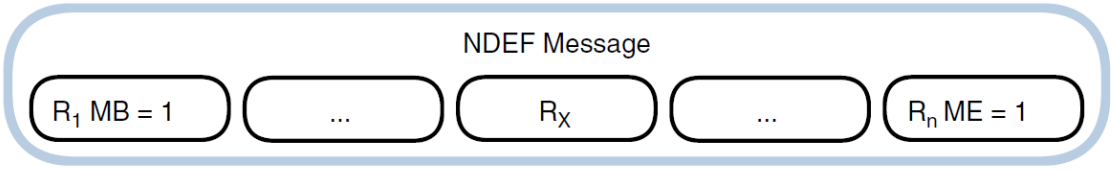

Abbildung 3.1: NDEF Message [LR10, S. 120-128]

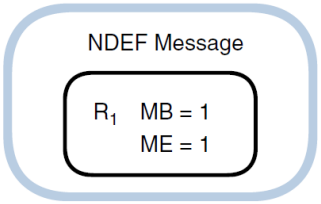

Abbildung 3.2: NDEF Message mit einem Record [LR10, S. 120-128]

3.1.2 NDEF Record

Wie schon weiter oben beschrieben, besteht ein Record aus 3 Parametern, Payload Length, Payload Type und einem optionalen Payload Identifier. Die Payload Length gibt die Anzahl der Oktetts in der Payload an. Der Payload Type gibt an, von welchem Typ die Payload ist. NDEF unterstützt URIs, MIME Media Typen und NFC-spezifische Formate. Der optionale Payload Identifier erlaubt Anwendungen, die tatsächliche Payload innerhalb eines NDEF-Records zu identifizieren.

Wie in Abb. 3.3 zu sehen ist, besteht der Header aus fünf Flags: Message Begin (MB), Message End (ME), Chunk Flag (CF), Short Record (SR) und ID Length Present (IL). CF gibt an, auf wieviele Records das Datenpaket aufgeteilt ist. SR kennzeichnet einen verkürzten Record, wenn die Payload von 32-Bit auf 8-Bit verkürzt ist. IL signalisiert ob der Record Identifikationsdaten beinhaltet. Type Name Format (TNF) gibt an, um welches Format des Felds Type es sich beim NDEF-Record handelt. Das TNF Feld besteht aus einem 3-Bit Feld mit unterschiedlichen Werten (Abb. 3.4). Das Feld TypeLength ist ein 8-Bit Integer-Wert der angibt, wie lang das Feld TYPE ist. Für bestimmte Werte des TNF-Feldes ist es immer Null. PayloadLength ist ein 32-Bit Wert und beschreibt die Länge des Payload-Felds. Das IDLength Feld ist ein 8-Bit Integer- Wert der die Länge des Feldes ID angibt. Nur wenn das IL Flag gesetzt ist, existiert das Feld ID. Im Type Feld wird das Format des Records angegeben laut Abb. 3.5. Wenn das Feld ID existiert, besteht es aus einer URI Referenz. Die Referenz kann absolut oder relativ sein. Im Payload Feld stehen die Daten.

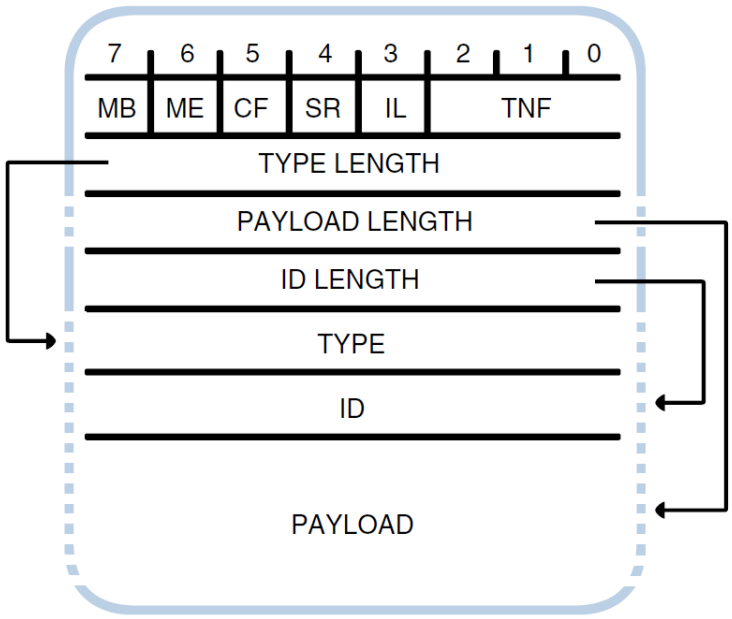

Abbildung 3.3: NDEF Record [LR10, S. 120-128]

3.2 MIME Media Type

MIME (Multiporpose Internet Mail Extension) dient zur Formatierung von Nachrichten mit verschiedenen Inhalten. Sie sind eine Erweiterung des Internetstandards RFC 822 und werden im Internet als auch zur Formatierung von Emails verwendet. MIME Types schaffen Kompatibilität für Umlaute und Multimedia Nachrichten. Um welchen Typ es sich dabei handelt, wird durch *Top-Level-Typ* und *Subtyp* gekennzeichnet. Beim Top-Level-Typ gibt es die Hauptkategorien wie Text, Image, Video, Audio oder Application. Der Subtyp gibt ein spezifischeres Format an, wie image/jpeg für den Typ Jpeg eines Bildes oder text/plain für einen Text ohne spezielle Formatierung. In Abb. 3.5 sind einige Beispiele für einen MIME Media Type aufgelistet [For06b].

3.3 RTD

Da das NDEF Format nicht definiert, wie NFC Geräte die Daten interpretieren oder darstellen müssen, wurde ein Standard geschaffen, namens RTD (Record Type Definition). Die RTD- Spezifikationen geben neben den grundlegenden Datenstrukturen die Richtlinien an, wie die Daten verarbeitet und dargestellt werden müssen[For06c].

Das Record Type String Feld eines NDEF Records beinhaltet den Namen des Record Typen (Record Type Name). Der Record Type Name wird von Anwendungen verwendet, um die Struktur und Semantik des Record- Inhaltes zu beschreiben. Ein

Type Name Format	Value
Empty	0x00
NFC Forum well-known type [NFC RTD]	0x01
Media-type as defined in RFC 2046 [RFC 2046]	0x02
Absolute URI as defined in RFC 3986 [RFC 3986]	0x03
NFC Forum external type [NFC RTD]	0x04
Unknown	0x05
Unchanged (see section 2.3.3)	0x06
Reserved	0x07

Abbildung 3.4: TNF [For06a]

Record Type Name kann in verschiedenen Formen vorkommen, wie MIME Media Types, absolute URIs, NFC Forum external Type Names oder Well- known NFC Type Names. Die Record Type Namen werden durch das NFC Forum und anderen dritten Parteien definiert.

Typen die vom NFC Forum spezifiziert wurden, werden **Well-known Types** genannt. Um Speicherplatz zu sparen, wird im Typ Feld nur der relative URN (Uniform Resource Name) eingetragen. Beim Well- known Type unterscheidet man zwischen lokalen und globalen Namen. Die globalen Typen fangen immer mit einem Großbuchstaben an. Sie werden vom NFC-Forum verwaltet und dürfen nicht beliebig definiert oder von den RTD- Spezifikationen abweichen. Die lokalen Typen hingegen fangen immer mit einem Kleinbuchstaben an und haben keine allgemein gültige Bedeutung. Lokale Typen sind anwendungsspezifisch und können in jeder Applikation eine andere Bedeutung haben. Auch das Datenformat kann sich von Anwendung zu Anwendung unterscheiden.

Es gibt aber auch Typen die nicht vom NFC Forum spezifiziert wurden, um auch anderen Organisationen die Definition von Record Typen zu ermöglichen, sie werden **NFC Forum External Types** genannt. Vor der URN wird eine Domäne hinzugefügt, die der Organisation zugehörig ist. Im Gegensatz zu Well-known Types, verfügen die External Types über keine Groß- und Kleinbuchstaben-Unterscheidung.

Ein sehr oft benutzter und wichtiger Record Type ist das Smart Poster. Die Idee dahinter ist, auf dem Tag zusätzliche Informationen zu speichern und ihn dadurch *smart* zu machen. Durch das Annähern eines NFC- Gerätes zu dem Tag, werden dem Benutzer die Informationen am Display dargestellt. Es können aber auch Aktionen festgelegt werden, die beim Empfang von Smart Poster Records auszuführen sind wie z.B. das Ausführen eines Webbrowsers um eine Webseite aufzurufen oder

application/acad	application/xml	application/x-tar	message/partial
application/applefile	application/x-bcpio	image/cis-cod	message/rfc822
application/astound	application/x-compress	image/cmu-raster	text/css
application/dsptype	application/x-cpio	image/fif	text/html
application/dxf	application/x-csh	image/gif	text/javascript
application/futuresplash	application/x-director	image/ief	text/plain
application/gzip	application/x-dvi	image/jpeg	text/richtext
application/listenup	application/x-envoy	image/png	text/rtf
application/mac-binhex40	application/x-gtar	image/tiff	text/tab-separated-values
application/mbedlet	application/x-hdf	image/vasa	text/vnd.wap.wml
application/mif	application/x-httpd-php	image/vnd.wap.wbmp	text/vnd.wap.wmlscript
application/msexcel	application/x-javascript	image/x-freehand	text/xml
application/mshelp	application/x-latex	image/x-icon	text/x-setext
application/mspowerpoint	application/x-macbinary	image/x-portable-anymap	text/x-sgml
application/msword	application/x-mif	image/x-portable-bitmap	text/x-speech
application/octet-stream	application/x-netcdf	image/x-portable-graymap	text/x-vcard
application/oda	application/x-nschat	image/x-portable-pixmap	text/xvcal
application/pdf	application/x-sh	image/x-rgb	video/mpeg
application/postscript	application/x-shar	image/x-windowdump	video/quicktime
application/rtc	application/x-sprite	image/x-xbitmap	video/vnd.vivo
application/rtf	application/x-stuffit	image/x-xpixmap	video/x-msvideo
application/studiom	application/x-supercard	message/external-body	
application/toolbook	application/x-sv4cpio	message/http	
application/xhtml+xml	application/x-sv4crc	message/news	

Abbildung 3.5: MIME Types

das Senden einer SMS um ein Ticket zu kaufen. Das gesamte Konzept von Smart Posters baut auf URIs (Uniform Resource Identifier) auf, die mittlerweile Standard im Referenzieren von Informationen im Internet geworden sind. Ein Smart Poster Record definiert eine Struktur, die eine URI mit verschiedenen Metadaten assoziiert.

Der Inhalt einer Smart Poster Payload ist eine NDEF-Message. Diese Message kann aus mehreren Records bestehen . Das Smart Poster kann aus null, einem oder mehreren dieser Komponenten bestehen:

- Title Record
- URI Record
- Action Record
- Icon Record
- Size Record
- Type Record

Ein Smart Poster könnte auch aus zusätzlichen Records bestehen, wie z.B. einer vCard mit angemessenen MIME Typen. Applikationen könnten diese zusätzlichen Records aber auch ignorieren.

Der Well-known Type eines Smart Posters wird mit **Sp** angegeben.

Abb. 7.12 zeigt einen Anwendungsfall für ein Smart Poster. Der Record besteht

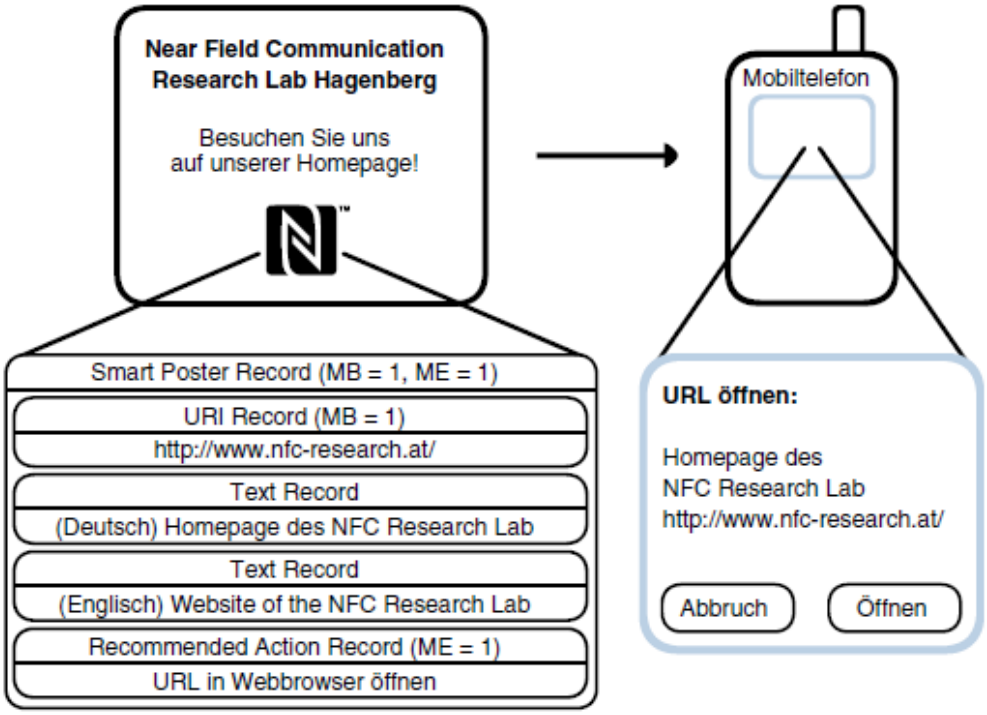

Abbildung 3.6: Smart Poster Anwendung [LR10, S. 120-128]

aus einer URL für eine Webseite, einem beschreibenden Text in zwei Sprachen und einer bevorzugten Aktion die ausgeführt werden soll. Wenn der Benutzer mit seinem NFC-fähigen Mobiltelefon in die Nähe des mit N-Mark gekennzeichneten Bereich kommt, wir der Smart Poster Record auf dieses übertragen. Das Smartphone startet eine für das Öffnen von URLs zuständige Applikation und stellt diese dar .

Weitere Record Types :

- URI Record Type
- Smart Poster Record Type
- Text Record Type
- Generic Control Record Type
- Singnature Record Type

Kapitel 4

Stand der Dinge

4.1 Use of NFC and QR Code Identification in an Electronic Ticket System for Public Transport [FT11]

In diesem Paper haben sich Ingenieure über den Bezahlvorgang bei öffentlichen Verkehrsmitteln Gedanken gemacht. Durch das Anstehen an Ticketschaltern vergeht oftmals wichtige Zeit und Passanten könnten so ihre Züge verpassen. Da beinahe jede Person in Ballungszentren ein mobiles Funkgerät besitzt und die Technologie dafür vorhanden ist, liegt es nahe, diese in den Bezahlvorgang mit einzubeziehen. Die 2 Technologien die dabei eingesetzt wurden sind QR(Quick Response) Codes und NFC (Near Field Communication).

Weiters wird es durch das Mitspeichern von Personendaten und Fahrverhalten möglich, eine weitere Technologie einzusetzen: AFC (Automated Fare Collection). Diese Technologie ermöglicht dem Verkehrsverbund seine Fahrpläne besser zu optimieren. Solche Informationen waren vorher nur durch mühevolle Kundenbefragungen zugänglich.

In diesem elektronischen Ticket-System müssen sich die Passagiere am Anfang und am Ende ihrer Reise über eine Android-Applikation ein- bzw. auschecken. Beide Technologien (RFID und QR Codes) wurden an den Stationen angebracht. Die RFID-Tags wurden über ein Nokia 6212 gelesen. Dieses mobile Gerät verfügt über NFC Technologie, um RFID Chips zu lesen und zu beschreiben. Da es noch nicht genug Handys mit NFC-Technologie gibt, wurden auch QR-Codes benutzt. QR ist ein zweidimensionaler Barcode der in seiner horizontalen wie auch in der vertikalen Ebene, Informationen enthält. Dies ermöglicht ihm 100mal so viel Informationen zu enthalten wie ein normaler Barcode. QR-Codes können mit jedem mobilen Gerät mit Kamera (Smartphones) abfotografiert werden. Dadurch erhält der Benutzer eine Nachricht oder wird zu einer Webseite weitergeleitet.

In Abb 4.1 sieht man einen typischen Usecase für diese Anwendung:

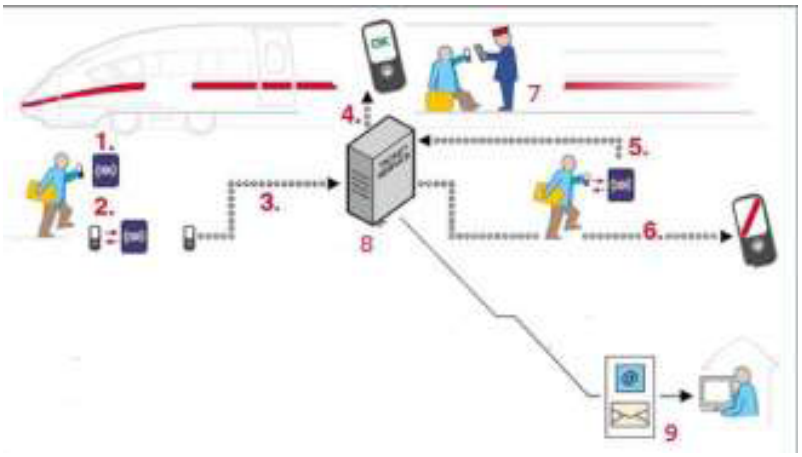

Abbildung 4.1: Public Transport

1. Benutzer platziert sein mobiles Gerät in der Nähe des RFID-Tags oder fotografiert den QR Code am Beginn seiner Reise
2. Die Applikation am Handy erkennt die Station
3. Die Applikation sendet die Daten zum Server um ein Ticket zu lösen
4. Der Server antwortet mit einem Ticket in Form eines QR-Codes oder einer Bestätigung am Ende der Reise
5. Der Name der Endstation wird über einen QRCode oder RFID-Tag gelesen und an den Server geschickt
6. Der Server beendet die Reise

4.2 NFC in Medical Applications with Wireless Sensors [ZL11]

Dies ist ein medizinischer Ansatz um mit Hilfe von NFC die Daten eines Patienten zu überwachen und aufzuzeichnen. Die Architektur besteht aus 4 Komponenten:

- Wireless medical Sensor, um die Daten aufzunehmen
- tinyOs und nesC um den Sensorknoten zu programmieren
- NFC für die Kommunikation
- und eine mobile Applikation basierend auf Android

Das Ziel ist es, die Vitalwerte mit Hilfe der verbindungslosen Sensoren zu lesen und in einer Datenbank abzuspeichern. Mit diesen Daten können Prognosen erstellt und effizient Medikamente verabreicht werden.

Die zwei eingesetzten Sensoren sind:

1. Pulsoximeter, um den Puls zu messen
2. Elektrokardiograph, um die Herzrate zu ermitteln

Diese Daten werden nun zu einer Basis-Station gesendet. In diesem Fall war die Basis-Station eine Zolertia Z1. Durch einen MSP430F2617 Mikrokontoller und einer starken 16-Bit RISC CPU @ 16MHz konnte ein robustes und energieeffizientes System garantiert werden. Jedes Smartphone mit GPRS, Wi-Fi und NFC, welches eine Android-Version höher oder gleich 2.3(Gingerbread) besitzt, könnte NFC unterstützen. Android 2.3 hat einen eigenen NFC Stack und eine API um NFC-Tags zu lesen.

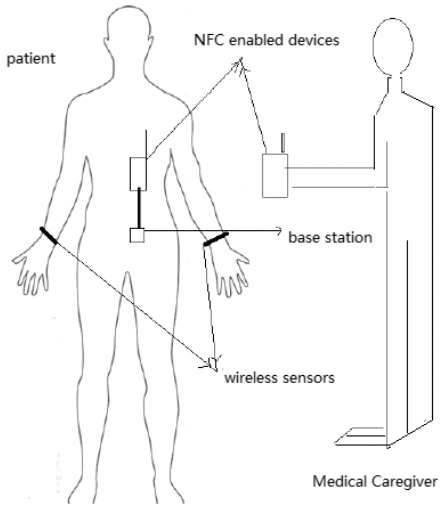

Abbildung 4.2: Überblick der Systemstruktur

Das große Ziel ist es, die Daten der Sensoren automatisch an die Basisstation zu senden. Hierzu kann man den Datenfluss in 3 Hauptsektoren teilen:

1. Datenfluss vom Sensor zum NFC Gerät
2. Kommunikation über NFC
3. Authentifikation des Empfängergerätes

Probleme die in Wireless Sensor Networks auftreten können:

Mehrere Empfänger: Ein Patient kann mehrere Ärzte oder auch Pflegepersonal haben, die auf die Patientendaten zugreifen. Es empfiehlt sich, dass die Netzwerkschicht Multicast Semantik unterstützt.
Mobilität des Gerätes: Patienten und Personal sind nicht an fixen Plätzen. Das Routing Protokoll muss sofort neue Routen finden, wenn der Arzt von einem Zimmer zum nächsten wandert.
Geographisches Routing in Sensor Netzen: Das Routing in Sensor Netzwerken unterscheidet sich vom Routing in drahtlosen Ad-hoc Netzwerken bzw. dem Internet. Routing Schemas die auf lokalen Informationen agieren sind besser geeignet als die auf geographischer Lage.
Sicherheit: Das WSN hat mehrere Sicherheitsvorkehrungen, wie öffentliche und private Schlüssel Kryptographie, welche teuer und komplex sind. Um auf die Daten zugreifen zu können, musste ein Pincode auf dem mobilen Endgerät eingegeben werden.

4.3 System Integration of NFC Ticketing into Existing Public Transport Infrastructure [WGSL12]

In einigen großen Städten hat sich die Chipkarte gegen die Papierkarte als Ticket bereits durchgesetzt wie die Octopus Card in Hongkong, die Oyster Card in London, die Ezlink Card in Singapore oder die Super Urban Intelligent Card in Tokio. Wenn man aber heutzutage in einer größeren Stadt von Punkt A nach B kommen will, kann es passieren, dass man verschiedene Verkehrsmittel benötigt, wie z.B. den Zug, die UBahn oder den Bus. Wenn nun jedes Verkehrsmittel von einem anderen Betreiber ist, benötigt man mehrere Tickets. Eine Applikation auf dem Mobiltelefon, mit der man Tickets für alle Verkehrsmittel kaufen und verwalten kann, wäre ein großer Vorteil für den Benutzer und die Verkehrsbetreiber. Einige Electronic Ticketing Standards in Europa haben sich bereits etabliert wie z.B. Calypso in Belgien, ITSO im Vereinigten Königreich und die VDV- Kernapplikation in Deutschland. In diesem Paper entschied man sich für die VDV- Kernapplikation, da sie sehr vielseitig ist und den Einsatz sowohl von kontaktlosen Chipkarten als auch von NFC-fähigen Mobiltelfonen mit Secure Element erlaubt.

Die VDV Smartcard Anwendung ist die Basis für das EFM System. Das EFM System ist eine Kombination aus bargeldlosen Bezahlen, elektronischen Ticket und automatischen Fahrtkosten Management. Die Anwendung muss sich aber nicht nur am NFC- fähigen Mobiltelefon befinden, sondern der Benutzer muss sich auch registrieren (*application distribution*). Im zweiten Schritt kann sich der Benutzer ein Ticket konfigurieren und sich auf sein Telefon mittels OTA (Over The Air) herunterladen (*ticket distribution*). Nachdem das Ticket vom Server heruntergeladen wurde, wird es in einer Liste von gekauften Tickets in der Applikation angezeigt (*displaying tickets*). Während der Fahrt kann das Ticket auf seine Aktualität und Regionalität überprüft werden (*inspection*). Zusätzlich muss die Applikation und das Ticket mit einer Blacklist verglichen werden, um gesperrte Tickets ausfindig zu machen. Falls dem Benutzer sein Mobiltelefon abhanden kommt oder seine Rechnungen nicht bezahlt, scheint die Applikation auf der Blacklist auf und wird für weitere Ticketkäufe gesperrt (*block request*). Wenn dies der Fall seien sollte, werden *block commands* generiert. Durch sogenannte *block removal requests* werden Sperren aufgehoben, die dann die Blacklist aktualisieren (*block removal commands*).

Das bereits existierende System baut auf der Architektur der Public-Transport-Association auf, genannt OÖVV. Diese Organisation besteht aus einem Netzwerk von mehr als 40 verschiedenen Verkehrsbetrieben.

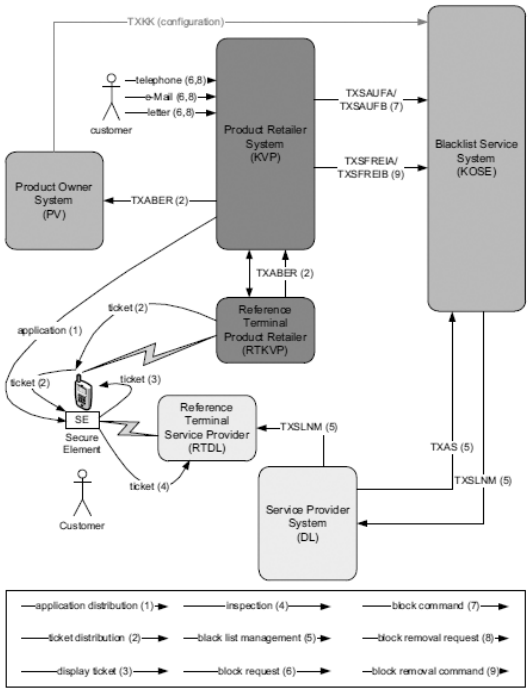

Abbildung 4.3: Überblick der Systemstruktur

Das standardisierte VDV besteht aus 4 Kernkomponenten:

- Care Module for Ticketing (UWP): Die Berechnung der Ticketpreise und die Abwicklung mit den verschiedenen Verkehrsbetrieben ist ein komplexer Prozess. UWP bietet eine effiziente Lösung in beiden Bereichen, mobilen Terminals und der Rechnungsstelle. Die Daten werden in einem geeigneten Format abgespeichert, dem Standard Clearing Data Record (SAD).

- Mobile Fare Management (MFGM): Das MFGM-System wird auf mobilen Geräten, Terminals und Ticketschaltern verwendet. Die OÖVV bietet bereits Geräte an, um elektronische Tickets zu validieren.

- Server Fare Management (SFGM): Das SFGM beschreibt die Serverseite des Management-Systems. Es erlaubt die Verbreitung von Systemakutalisierungen, von MFGM-Geräten und die Synchronisation der gesammelten Daten.

- Office Fare Management (OFGM): Im eigentlichen Sinne ist das OFGM-System ein Customer-Relationship-Management-System. Es wird verwendet um Benutzerdaten zu verwalten.

Implementierung

Das existierende Product Retailer System basiert auf Microsoft Dynamics und kann mit .NET-Modulen erweitert werden. Alle serverseitigen Implementationen wurden mit C- Sharp und die Benutzeroberfläche mit ASP.NET entwickelt. Die Daten werden in einer Microsoft SQL Server Datenbank verwaltet. Als mobiles Gerät wurde das Nokia 6131/ 6212 mit einem Java Card Applet verwendet. Im Hintegrund läuft ein serverseitiger Reference Terminal Product Retailer der es dem Benutzer ermöglicht, seine Tickets über ein Webinterface zu verwalten. Weiteres kann der Retailer über eine Applikation am Mobiltelefon genutzt werden, um Informationen über Tickets abzurufen. Intern benutzt das Webservice das UWP, um Preise alternative Routen oder Parameter wie Stationsnummern zu berechnen. Wenn der Benutzer seine Entscheidung getroffen hat, wird ein SAD Data Record generiert und an das Mobiltelefon übermittelt. Die Kommunikation zwischen Server und der Smartcard-Applikation im Secure Element findet über das HTTP Protokoll statt. Nachdem das Ticket erfolgreich an den Benutzer übermittelt wurde, wird eine TXABER Nachricht an das Product-Retailer-System als Bestätigung geschickt. Die gespeicherten Tickets können über ein J2ME MIDlet auf dem Secure Element mit einer graphischen Benutzeroberfläche abgerufen werden. Zusätzlich wird beim Terminal die Ticket-ID mit der Blacklist verglichen, um gesperrte Tickets auszuschließen. Dazu muss am Terminal zu jeder Zeit die aktuelle Kopie der Blacklist liegen. Der Blacklist Service ist eine alleinstehende Anwendung, um Blacklists für alle Partner zu berechnen und administrieren. Blockieranfragen von Benutzern inklusive Grund werden auch in der Blacklist eingetragen. Zuerst wird der Status der Tickets an das Product-Retailer-System gesendet und anschließend in der Blacklist aktualisiert. Es wurde auch eine benutzerfreundliche Android-Anwendung entwickelt, über die man mit jedem NFC-fähigen Mobiltelefon seine Tickets über eine kontaktlose Schnittstelle abrufen kann. Je nachdem, ob das das Ticket gültig ist oder nicht, scheint ein grüner oder roter Punkt am Terminal auf.

Herausforderungen

So wie in vielen anderen Projekten in dieser Sparte, fehlt es noch an aktuellen Geräten mit NFC, die den CardEmulation Modus unterstützen. Auch die Frage nach den Zugriffsrechten auf Secure Elementen bei einigen Geräten ist noch nicht geklärt. Kommunikationsfehler hat es mehr in der Übertragung des Tickets over-the-air gegeben als über die Übertragung mit NFC. Wenn das elektronische Bezahlen auch integriert werden soll, müssten Mechanismen eingebaut werden, die den Bazahlvorgang wieder rückgängig machen, falls dieser fehlschlägt.

Fazit

Das System wurde erfolgreich in die existierende Systemarchitektur der Verkehrbetriebe von Oberösterreich integriert. Es war auch ein Versuch die VDV Kernapplika-

tion in den Europäischen Standard zu integrieren. Heraus kam ein Proof-of-concept und ein Schritt in Nähe eines nationalem EFM-Systems.

4.4 Physical Poster Gateways to Context-aware Services for Mobile Devices [RSH04]

Die Kombination aus drahtlosen Netzwerken und leistungsfähigen mobilen Geräten, hat ein großes Potential für neue Dienste und Anwendungen. Heutige Mobiltelefone sind ausgestattet mit einer Vielzahl von verschiedenen Funktechnologien, einem großen Display, einer Kamera oder einem Browser. Viele Nutzer wissen aber nicht, wie sie diese Features effizient nutzen. Dies kann unterschiedliche Gründe haben:

- Benutzer sehen keinen Nutzen in manchen Diensten
- Der Dienst funktioniert nicht immer, bzw. ist es nicht klar, wieso ein Dienst in einem bestimmten Kontext nicht funktioniert
- Es ist zu kompliziert, den Dienst zu nutzen oder einzurichten
- Es zahlt sich nicht aus, den Dienst zu nutzen

Um SmartPoster effizient einzusetzen, muss man sich über folgende Punkte Gedanken machen:
Entdeckung von Diensten : Das Erkennen von Diensten wird dem Benutzer überlassen. Manchmal wissen Nutzer gar nicht, dass so ein Dienst überhaupt existiert.
Entdeckung von Netzwerken : Moderne Mobiltelefone sind mit einer Vielzahl von Netzwerk- Schnittstellen ausgestattet, wie GSM, UMTS, WLAN, Bluetooth oder NFC. Je nach Übertragungsgeschwindigkeit und Preis, muss sich der Benutzer für eine geeignete Funkttechnolgie entscheiden.
Kosten : Benutzer müssen für die Kosten der übertragenen Daten selber aufkommen, außer sie verfügen über eine Breitbandverbindung oder Flatrate. Um ein Poster als Gateway für das Erreichen von Informationen zu nutzen, muss man den Benutzer darauf aufmerksam machen. Dies wurde durch das Hinzufügen von Codes (maschinell oder visuell lesbar) erreicht. Das Kostenproblem von teuren Übertragungsraten, wurde durch das Nutzen eines lokalen Netzzugangs gelöst.

Um das Konzept zu veranschaulichen, wurden 2 Szenarien gewählt:
Ein **Kino Poster** welches an einer Bushaltestelle befestigt wurde. Der Benutzer kann sich dort über den aktuellen Film informieren und zusätzliche Informationen beschaffen. Über eine installierte Anwendung am Mobiltelefon (SPA) kann er den am Poster befindlichen Code abfotografieren. Die Anwendung baut sofort eine Bluetooth- Verbindung zwischen dem Telefon und einem an der Busstation befindlichen AccessPoint auf. Eine Webseite öffnet sich und der Benutzer hat Zugriff auf Videos, Titelmelodien oder dem Kinoprogramm.

Ein Poster einer **Autovermietung**, welches in einem Bahnhof hängt und über Verfügbarkeit und Preise von Autos informiert. Der Benutzer kann sich sofort nach der Ankunft über vermietbare Autos informieren und das zu jeder Zeit. Er spart sich auch Zeit an langen Warteschlangen in Büros von Autovermietern. Nachdem der Code am Plakat mit einer vorinstallierten Anwendung gescannt wurde, wird eine WLAN- Verbindung mit dem Access Point aufgebaut. Über eine Webseite kann sich der Benutzer ein Auto aussuchen und den Bezahlvorgang einleiten. Mit Hilfe der Applikation und Bluetooth kann ein Schließfach am Bahnhof geöffnet werden, indem sich Papiere und Schlüssel befinden. Über GPS wird dem Benutzer die Position des Autos gezeigt.

Kategorien von Postern

Heutzutage findet man an fast allen öffentlichen Plätzen, Plakate oder Anzeigen. In den meisten Fällen haben diese Anzeigen auch eine Beziehung zu dem Platz, an dem sie sich befinden. Speziell an Orten wie Flughäfen, Bahnhöfen oder Bushaltestellen findet man diese Art von Werbung. Aber auch dort wo Passanten sehr viel Zeit verbringen, wie Restaurants, Kinos, Kreuzungen oder Auslagen werden gerne Plakate platziert. Um mit Plakaten interagieren zu können, kommt es sehr darauf an, in welcher Entfernung sie sich befinden. Einerseits gibt es Plakate die sehr nahe an den Zielpersonen angebracht sind, ein Beispiel wären hier Busstationen, an denen die Personen eine längere Zeit verweilen. Andererseits gibt es Plakate, die nur von der Weite betrachtet werden können, der Benutzer hat keine Möglichkeit in die nähere Umgebung des Plakates zu kommen. Beispiele wären hier Plakate an Decken oder Hochhäusern. Es muss aber auch berücksichtigt werden, wie lange der Passant Zeit hat, ein Plakat zu betrachten. Im generellen kann man zwischen 4 Kategorien differenzieren(Abb.4.4):

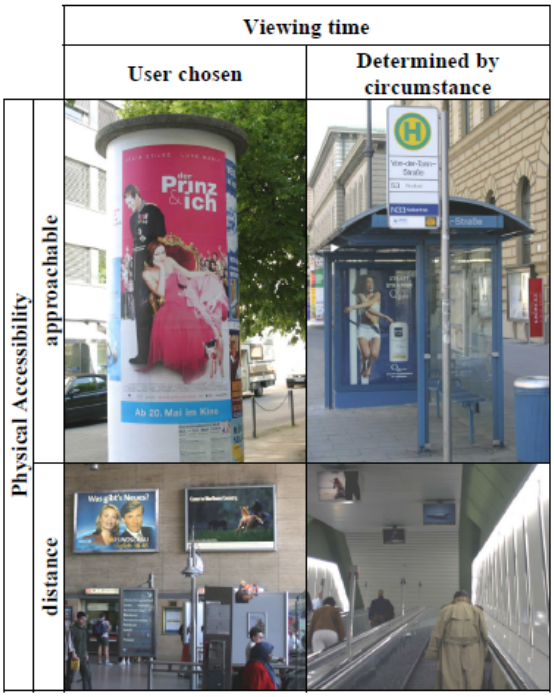

Abbildung 4.4: 4 Kategorien

Um herauszufinden wie Personen mit den Postern interagieren, wurden Aufzeichnungen an spezifischen Plätzen vorgenommen. Bushaltestellen und Bahnhöfe eignen sich großartig um das Verhalten von Passanten zu beobachten.
Im ersten Fall wurden 100 Personen an einer Bushaltestelle beobachtet. Es stellte sich heraus, dass fast ein Drittel der Passanten keine Zeit hatte um einen mobilen Dienst zu nutzen, da sie weniger als 60 Sekunden auf ihren Bus warteten. 44 Prozent der Beobachteten warteten länger als 3 Minuten und wären somit potentielle Nutzer. Im zweiten Fall wurde eine Haltestelle einer Straßenbahn gewählt. Hier betrug die durchschnittliche Wartezeit eines Passanten 4 Minuten und 37 Sekunden. Im dritten Fall handelte es sich um eine Haltestelle in einem Bahnhof. Man konnte beobachten, dass die Passanten im Durchschnitt 5 Minuten auf ihren Zug warteten. Personen die länger als 5 Minuten an der Haltestelle verbrachten, hatten etwas zum Lesen mit sich, Personen unter 5 Minuten meistens nicht. Im generellen konnte man beobachten, dass Personen die weniger Zeit an Haltestellen verbrachten, auch weniger zu tun hatten. Dies ist eine sehr vielversprechende Basis für die Nutzung von mobilen Diensten als *Killer Applikation*. Killing Time is a very important killer application [Nie00].
Es wurde auch eine Studie durchgeführt, welche Dienste ein Smart Poster bereitstellen sollte. Im Zusammenhang mit einer Autovermietung stellte sich heraus, dass 84 Prozent der Befragten mehr Informationen über Spezialangebote und Preise wünschten. Im Zusammenhang mit einem Reisebüro würden die Passanten gerne mehr über

Lastminute Flüge oder verbilligte Flüge erfahren (87 Prozent). Auch das Abrufen von standortbezogenen Informationen schnitt bei der Befragung sehr gut ab.

Architektur

In Abb. 4.5 ist die generelle Architektur des Systems zu sehen. Um Dienste auf der Serverseite anbieten zu können, wurden Markup Sprachen wie WAP, (X)HTML und Synchronized Multimedia Interchange Language (SMIL) verwendet. Auf dem Gerät wurde der Simplicity Personal Assistent (SPA) installiert, der als Schnittstelle zwischen dem Benutzer und den Diensten, Netzwerken und Geräten fungiert. Um eine Kommunikation zwischen dem Poster und dem mobilen Gerät herzustellen, entschied man sich für 4 verschiedene Sensoren: Kamera, Nahfeld Netzwerke, Lokalisierungsfunktionen von Mobilfunknetzen oder Eingabe durch den Benutzer. In

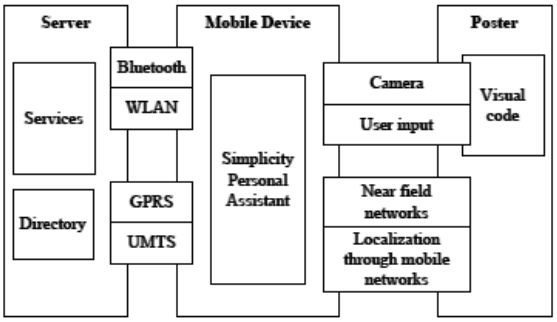

Abbildung 4.5: Architektur

Bezug auf die Bereitstellung von Netzwerken wird zwischen 3 Arten unterschieden:

- Das Netzwerk wird vom Mobilfunkanbieter bereitgestellt und die Werbeagentur zahlt
- Der Werber stellt ein lokales Netzwerk bereit (Bluetooth, WLAN)
- Das Netzwerk wird vom Mobilfunkanbieter bereitgestellt und der Benutzer zahlt die Gebühren

Für den Benutzer ist es natürlich von Vorteil, wenn er keine Gebühren zahlen muss. Die Konfiguration des Netzwerks sollte so einfach wie möglich sein, um die Benutzerfreundlichkeit zu erhöhen. Basierend auf dieser Architektur wurde ein Prototyp in Zusammenarbeit mit dem EU-Projekt *Simplicity* entwickelt.

4.5 An NFC-Based Solution for Discount and Loyalty Mobile Coupons [SSVARGN12]

Der Erfolg des Business- Models *Deal of the day* ist heutzutage kein Geheimnis mehr. In diesem Paper wird ein Ökosystem vorgestellt, welches den Gebrauch von Papier als Gutschein überflüssig macht. Das System heißt *WingBonus* und ist verantwortlich für die Verbreitung, Verteilung, Unterstützung, Validierung und Managen von Gutscheinen, Coupons oder Kundenkarten mit Hilfe von NFC. Durch den Gebrauch von NFC werden klassische Gutscheine oder Kupons durch M-Coupons ersetzt. Dies hat den Vorteil, dass sich Unternehmen viel Geld durch den Gebrauch von elektronischen Gutscheinen sparen und zusätzlich Erkenntnisse über das Verhalten ihrer Kunden sammeln. Außerdem wird die Kundenbindung gestärkt und der Umsatz erhöht. M-Coupons können über eine Internetverbindung, einem Bluetooth Server, einem Smart Poster oder einem NFC-Lesegerät übertragen werden und werden sicher am Mobiltelefon abgespeichert. Dadurch sind M-Coupons immer für den Benutzer verfügbar und sind gleich danach bereit zum Einlösen. Ein weiterer Vorteil dieses WingBonus Systems ist es, dass man auch ohne NFC, Kupons beziehen oder austauschen kann. Dies geschieht über QR-Codes, die über eine im Mobiltelefon integrierte Kamera abfotografiert werden können. Die Verschlüsselung der Daten am Mobiltelefon erfolgt über den Blowfish Algorithmus. Zusätzlich wird beim Einlösen des Gutscheins die Gültigkeit über den WingBonus Server geprüft.

System Architektur

Neben den klassischen Gutscheinen wurde bei WingBonus auch eine Klubkarten-Funktion implementiert. Jedesmal wenn der Benutzer einen Dienst eines Unternehmens in Anspruch nimmt, sammelt er damit Punkte. Wenn er eine bestimmte Anzahl von Punkten gesammelt hat, kann er sie gegen Produkte oder Dienste einlösen. Dadurch können Kunden an ein Unternehmen gebunden werden. Wie schon weiter oben beschrieben, kann der Benutzer über mehrere Wege einen Kupon beziehen. Die WingBonus-Architektur lässt eine Übertragung von einem SmartPoster oder einem Bluetooth Server zu. Der Kunde kann aber auch mit anderen Benutzern seine Kupons austauschen(1). Falls nicht alle notwendigen Daten auf dem Tag oder am Bluetooth Server gespeichert sind, werden die Daten an den Server gesendet(2). Nach der Validierung (3), erhält der Server alle notwendigen Informationen aus der Datenbank (4) und die Daten werden an die Mobile Anwendung weitergegeben (5). Alternativ können Gutscheine auch über die Mobile Anwendung heruntergeladen werden (6). Über eine Peer-to-peer Verbindung können die Gutscheine beim Händler eingelöst werden. Als erstes werden die Daten an das NFC-Lesegerät gesendet, anschließend gleicht der PC die Daten mit dem WingBonus Server ab. Wenn die Operation erfolgreich war, wird eine Nachricht am Mobiltelefon angezeigt. Durch den Einsatz einer Peer-to-peer Verbindung beim Einlösen des Gutscheins, können zusätzliche Informationen wie Standort durch GPS Daten gesammelt werden. In Ver-

bindung von weiteren nützlichen Daten wie Alter oder Beruf kann Marktforschung betrieben werden, die den Unternehmen hilft weiter zu wachsen.

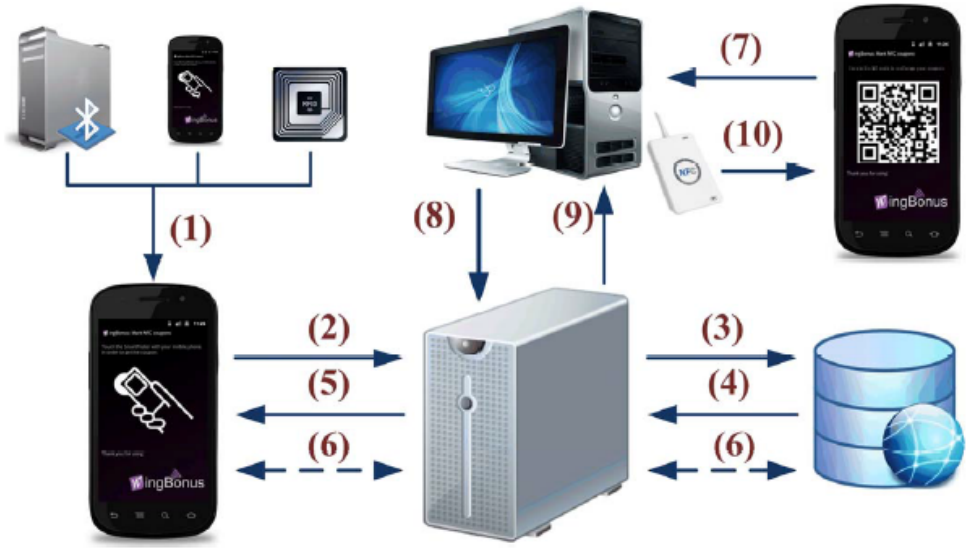

Abbildung 4.6: System-Architektur

Mobile Anwendung

Damit das System auch mobil anwendbar ist, wurde eine Android-Applikation entwickelt. Wenn der Benutzer das erste Mal die Applikation startet, wird ein Willkommensbildschirm ausgegeben, anschließend wird er gebeten, seine Daten einzugeben. Nachdem der Benutzer identifiziert ist, wird das Hauptmenü angezeigt. Das Managen oder das Herunterladen von Kupons ist über das Hauptmenü zugänglich. Zusätzlich wird dem Benutzer über eine Karte angezeigt, wo er seine Kupons in der näheren Umgebung einlösen kann. WingBonus nutzt hier die Google-Maps API um die schnellste Route zum nächsten Restaurant oder Händler zu finden. Neben der mobilen Anwedung, wurde auch eine Webseite entwickelt, die auf Content-ManagementsSysteme verzichtet und auf den Sprachen HTML5, CSS3 und Javascript basiert.

Synchronisation

Im Idealfall sollte es eine direkte Kommunikation von der Anwendung zum Datenbank Server geben. Da aber Android SQL Lite unterstützt, hat es keinen Zugriff auf eine externen Datenbank. Dies bedeutet, dass der Synchronisationsprozess ad-hoc zwischen der mobilen Anwendung und der Datenbank aufgebaut werden muss. Auf die Verwendung von Web-Services wurde hier vollkommen verzichtet, stattdessen verwendete man das SOAP-Protokoll oder das Parsen von XML-Dateien.

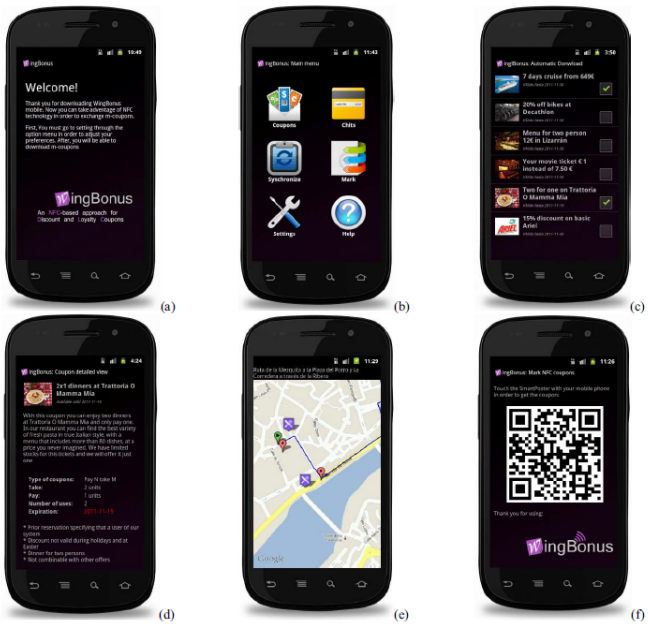

Abbildung 4.7: Mobile Anwendung

Durch die Unterstützung von NFC in mobilen Geräten wird dem Benutzer der Bezug, das Speichern, das Managen und der Gebrauch von M-Kupons immens erleichtert. Der gesamte Prozess wird dadurch schnell, sicher, effizient und transparent. Die Kupons sind zu jederzeit durch das Mobiltelefon verfügbar und können weder vergessen noch verloren werden. Partnerunternehmen von WingBonus bietet es große Vorteile wie:

- Reduktion von Kosten,
- Beseitigung von Papier,
- Erreichen von mehr potentiellen Kunden,
- Beseitigung von Fälschungen,
- Echtzeit-Tracking
- Marktanalyse,
- Trendforschung,
- Kundenbindung usw.

Diese Arbeit wurde durch das Ministry of Science and Innovation of Spain und FEDER unterstützt.

Kapitel 5

Design

5.1 Problembeschreibung

Das Ziel dieser Arbeit ist es, Near Field Communication in ein bestehendes System zu integrieren. Bei dem bestehenden System handelt es sich hier um die Bergfex Lite-Applikation die für Android entwickelt wurde. Sie ist derzeit im Google Play Store in der Version 1.4 veröffentlicht und in 18 Sprachen verfügbar. Die Bergfex Lite Version unterscheidet sich von der Pro Version darin, dass nicht alle Wetter- und Schneeberichte bzw. Webkameras verfügbar sind. Weiters können nur bis zu 3 Skigebiete zu den Favoriten gespeichert werden. Die Bergfex Lite Applikation weist folgende Funktionen auf:

- Tägliche Schneeberichte
- Wetter- und Schneevorhersagen
- Webkameras
- Karten der Skigebiete in hoher Auflösung
- Detaillierte Informationen über das Skigebiet
- Schneevorhesagen für die Alpenregion
- Anzeige der Skigebiete auf einer Karte
- Navigation zum Skigebiet (GoogleMaps)
- In 18 Sprachen verfügbar

Dem Benutzer soll es möglich sein, mit der NFC-fähigen Applikation, Informationen von SmartPostern schnell und einfach abzurufen. Ein Plakat an dem Tags angebracht werden, fungiert als SmartPoster. Die Tags sollen mittels dem NFC-Forum-Logo gekennzeichnet werden, um den Benutzer darauf aufmerksam zu machen. Durch die

Abbildung 5.1: Bergfex LITE Applikation

Annäherung des NFC-fähigen Mobiltelefons an ein NFC-Forum-Logo, werden die Daten vom Chip an das Gerät übertragen. Die Reichweite des Mobiltelefons um einen Tag zu lesen, beträgt um die 10 Zentimeter.

5.2 Zu klärende Punkte

Um NFC in eine bestehende Android Applikation zu integrieren, müssen folgende Fragen beantwortet werden:

- **Welche Android Versionen unterstützt NFC ?**
  Android ist derzeit in der Version 4.3 verfügbar. Nicht alle Mobiltelefone unterstützen die neueste Android Version. Dies hat damit zu tun, dass viele Geräte nicht mehr kompatibel zur neuesten Version sind und ein Update die

Performance sogar verschlechtern würde. Auch für die Hersteller ist es nicht rentabel für jedes Gerät ein Update auf die neueste Version zur Verfügung zu stellen.

- **Kann die Bergfex Lite Applikation auch ohne NFC bedient werden?** Da nicht alle Android Mobiltelefone NFC unterstützen, muss eine Lösung gefunden werden, damit alle bestehenden Bergfex Benutzer die Applikation weiterhin nutzen können. Ein Update auf die neueste Bergfex Version mit NFC soll keine Benutzer ausschließen. Auch ein automatisches Update durch den Google Play Store soll näher betrachtet werden.

- **Was passiert wenn das Gerät NFC nicht unterstützt?** Die Applikation soll weiterhin funktionieren, auch wenn der Benutzer kein NFC-fähiges Mobiltelefon besitzt. Die Applikation soll dem Benutzer im Google Play Store angezeigt werden, auch mit Integration von NFC.

- **Welche Vorteile werden durch die Integration von NFC erreicht?** NFC bietet wesentliche Vorteile in der Bedienung der Applikation. Es müssen keine Zwischenschritte erfolgen und der Benutzer hat sofort Zugriff auf den gewünschten Inhalt. Auch das Herunterladen der Applikation kann durch NFC beschleunigt werden, da der Benutzer weder Namen noch Ort der Applikation wissen muss. Auf dem NFC-Tag können zusätzliche Informationen gespeichert werden, um die Funktionalität der Applikation zu erweitern. Auch hinsichtlich Bedienung im Skigebiet sollen die Vorteile aufgezeigt werden.

- **Welche Rechte darf die Applikation besitzen?** Bevor eine Applikation aus dem Google Play Store geladen werden kann, muss der Benutzer informiert werden, auf welche Funktionen des Gerätes die Applikation zugreifen darf. Durch Ändern dieser Rechte, kann kein automatisches Update mehr durchgeführt werden.

- **Wie kann dem Benutzer die Anwendung von NFC erleichtert werden?** Viele Benutzer wissen gar nicht, dass ihr neues Mobiltelefon über diese Technologie verfügt. Durch Anleitungen und graphische Hilfeleistungen, soll dem Benutzer die Bedienung erleichtert werden. NFC-fähige Plakate (Smart-Poster) sollen sofort als diese erkannt werden.

- **Wie verhält sich die Dateigröße durch die Integration von NFC?** Die Integration von NFC soll die bestehende Bergfex Lite Applikation nicht in ihrer Größe verändern. Einbinden von großen Libraries ist nicht erwünscht.

5.3 Anforderungen

Darauf aufbauend soll ein Prototyp entwickelt werden, mit folgenden Funktionen:

Bergfex

- **Aufruf von Wetterdaten über ein Smart Poster**
  Über die vorhandenen Schnittstellen von Bergfex, soll der Aufruf von Wetterdaten direkt über NFC angeboten werden. Auf dem Tag werden Skigebietrelevante Daten gespeichert. Diese Daten werden über eine Webschnittstelle an den Bergfex-Server gesendet. Der Server wertet diese Daten aus und schickt eine Antwort an das mobile Gerät. Die Antwort beinhaltet die Wetterdaten, die in der Applikation graphisch dargestellt werden.

- **Anmeldung bei einem Newsletter**
  Der Benutzer soll sich über einen Newsletter von Bergfex über aktuelle Aktionen und Neuigkeiten informieren können. Der Benutzer muss nicht wissen wo er sich in der Applikation im Newsletter anmelden muss, sondern wird durch das Einlesen des Tags an das Newsletter-Formular weitergeleitet. Die Email-Adresse wird an den Bergfex-Server gesendet und in der Datenbank auf Duplikate überprüft.

Blue-Tomato

- **Bonuspunkte-Programm**
  Der Benutzer soll sich über die Applikation mit seinem Blue-Tomato-Benutzerkonto verbinden können. Falls der Benutzer noch kein Konto bei Blue-Tomato hat, muss er sich über einen Browser im Onlineshop (www.blue-tomato.com) anmelden. Seine Daten und Bonuspunkte sind in einer Datenbank am Server abgespeichert. Nach dem Anmelden kommt der Benutzer zu seinem Profil, wo seine bisherigen Bonuspunkte aufgelistet sind. Ein Ausbau des Bonuspunkte-Programms (Loyalty Program) ist in näherer Zukunft geplant.

- **Punkte sammeln mit Hilfe von NFC-Tags**
  Der Benutzer soll spielerisch animiert werden, seinen Punktestand zu erhöhen. Durch das Annähern des Mobiltelefons an das SmartPoster, werden Punkte(Tomaten) gesammelt. Je mehr Tomaten ein Benutzer sammelt, desto höher wird sein Punktestand. Die Smartposter auf denen die Tomaten abgebildet sind, werden im näheren Umkreis aufgestellt.

- **Anzeige der aktuellen Bonuspunkte**
  Je nachdem ob der Benutzer angemeldet ist oder nicht, werden am Startbildschirm die bisher gesammelten Punkte als *Bonus Points* angezeigt. Der

angemeldete Benutzer bekommt zu seinen gesammelten Tomaten den Online-Punktestand dazu gezählt. Es soll auch möglich sein, die Tomaten offline zu sammeln.

- **Hidden Spots**
  GPS Koordinaten werden auf einem Tag hinterlegt. Die Applikation liest diese Daten ein und stellt den geheimen Ort auf einer Karte dar.

- **Anzeige der Blue-Tomato Shops auf einer Karte**
  Alle Blue-Tomato Shops sollen auf einer Karte in der Applikation gekennzeichnet werden. Zusätzlich werden geheime Orte/Events auf der Karte angezeigt.

- **Verschlüsselung der Tags**
  Damit nur die Applikation die Daten auf den Tags auslesen kann, werden die Informationen verschlüsselt gespeichert.

- **Anmeldung beim Blue-Tomato Newsletter**
  Der Benutzer soll sich über einen Newsletter von Blue-Tomato über aktuelle Aktionen und Neuigkeiten informieren können. Der Benutzer muss nicht wissen, wo er sich in der Applikation im Newsletter anmelden muss, sondern wird durch das Einlesen des Tags an das Newsletter-Formular weitergeleitet. Die Email-Adresse wird an den Blue-Tomato-Server gesendet und in der Datenbank auf Duplikate überprüft.

5.4 Architektur

5.4.1 System-Architektur

Der Benutzer scannt mit seinem NFC-fähigen Mobiltelefon den durch ein NFC-Logo gekennzeichneten Tag ein. Der Tag ist in einem Smart Poster integriert, das durch ein durchsichtiges Kunststoffglas gegen Diebstahl und Schmutz geschützt ist. Beim ersten Einlesen des Tags wird die Applikation am Smartphone installiert, danach können weitere Tags eingelesen werden. Sammelt der Benutzer alle benötigten Tags (Tomaten) ein, kann er sie über das Bonuspunkteprogramm einlösen. Das Bonuspunkteprogramm besteht aus einem Webservice,einer Datenbank und einem Webinterface. In der Datenbank werden alle Benutzerdaten und Punkte gespeichert. Über das Webinterface können die gesammelten Punkte eingesehen und Einstellungen vorgenommen werden. Auch über die Anwendung (*Scoreboard*) kann der Benutzer seinen derzeitigen Punktestand einsehen. Für die Kommunikation zwischen Außenwelt und Bonuspunkteprogramm ist das Web-Service zuständig. Neben der Android-Applikation sollen auch andere Systeme auf das Bonuspunkteprogramm zugreifen können.

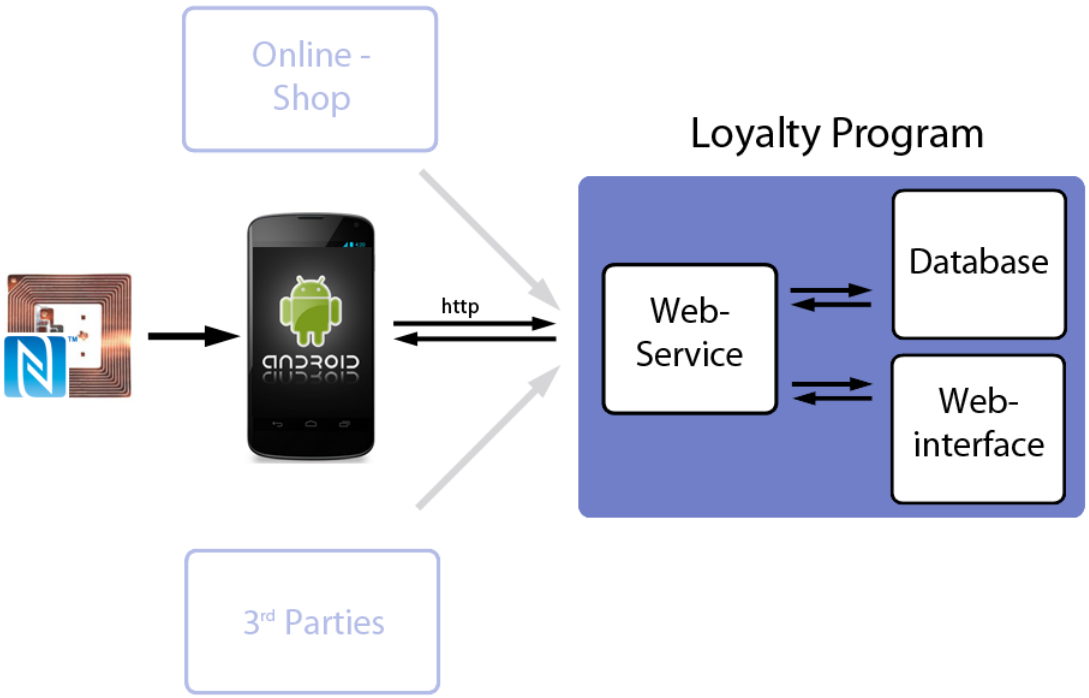

Abbildung 5.2: System-Architektur

5.4.2 Logische Sicht

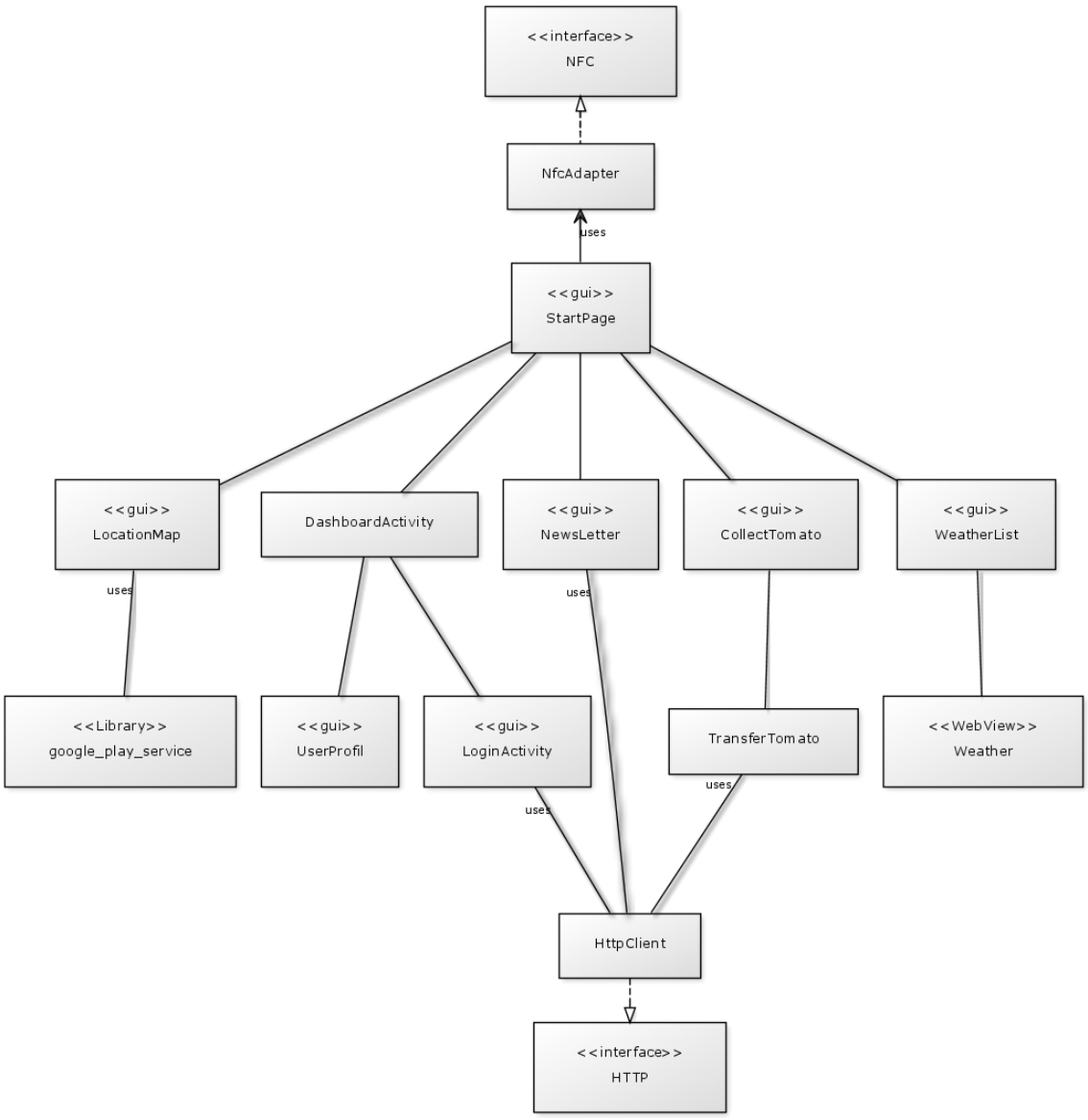

Abbildung 5.3: Klassendiagramm

StartPage

Als Ausgangspunkt des Klassendiagramms dient die Klasse StartPage. StartPage ist eine graphische Benutzeroberfläche, über die der Benutzer die Applikation bedienen kann. Sie wird aufgerufen sobald die Anwendung gestartet wird. Diese Klasse verwendet die Klasse NfcAdapter, die als Schnittstelle zu NFC dient. Alle Anfragen die über die NFC-Schnittstelle kommen, werden über die Klasse StartPage abgearbeitet.

LocationMap

Die Klasse LocationMap dient zur Darstellung der Tomaten auf einer Karte. Sie verwendet die Bibliothek google_play_service welche von Google bereitgestellt wird. Die Daten, die über die NFC-Schnittstelle kommen, werden über die Klasse StartPage an die Klasse LocationMap weitergeleitet.

DashboardActivity

Die Klasse DashboardActivity überprüft ob der Benutzer angemeldet ist oder nicht. Falls ja, wird die Klasse UserProfil aufgerufen, andernfalls die Klasse LoginActivity.

UserProfil

Die Benutzeroberfläche UserProfil zeigt die Informationen des Benutzers an.

LoginActivity

Die Klasse LoginActivity zeigt ein Anmeldeformular an. Nachdem der Benutzer seine Anmeldedaten eingegeben hat, werden die Daten über den HTTPClient an ein Webservice geschickt. Der HTTPClient verwendet die Schnittstelle HTTP.

Newsletter

In der Benutzeroberfläche Newsletter werden die Email-Adressen eingetragen und über den HTTPClient an das Webservice weitergeleitet.

CollectTomatos

Über die graphische Oberfläche CollectTomatos werden die Tomaten gesammelt. Die Informationen des NFC-Tags werden über die Schnittstelle NFC eingelesen und an die Klasse CollectTomatos weitergeleitet. Sobald der Benutzer alle Tomaten gesammelt hat, wird die Klasse TransferTomatos aufgerufen, die die Daten über den HTTPClient an das Webservice weiterschickt.

WeatherList

WeatherList zeigt eine Liste der verfügbaren Skigebiete an. Über diese Liste kann das gewünschte Wetter des Skigebiets abgerufen werden. Die Klasse Weather übergibt einen Parameter und zeigt das Wetter in einer WebView an.

5.4.3 Prozess Sicht

Collect Tomatos Aktivität

Um die Punkte(Tomaten) für das Bonussystem zu sammeln, ist es nicht zwingend notwendig die Applikation am Mobiltelefon zu öffnen. Die Applikation erkennt den Tag und entschlüsselt die auf dem Tag befindliche NDEF-Nachricht. Sobald der Benutzer alle Tomaten gesammelt hat um den Bonus einzulösen, wird er aufgefordert, falls er dies noch nicht gemacht hat, sich mit seinen Benutzerdaten anzumelden. Sobald sich der Benutzer angemeldet hat, kann er seine Punkte beim System bestätigen lassen. Diese werden vom Server kontrolliert und eine Bestätigunsnachricht wird in der Applikation angezeigt. Nach der Kontrolle am Server, werden die gesammelten Punkte zu seinem bisherigen Punktestand addiert.

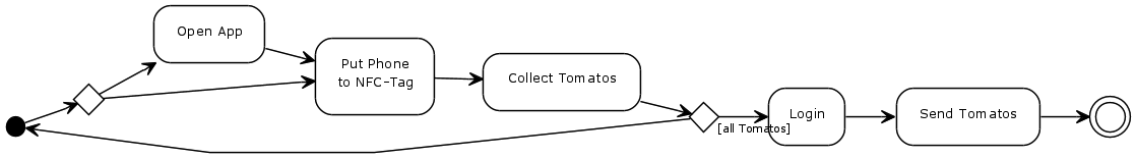

Abbildung 5.4: Collect Tomatos Activity

Login Aktivität

Dem Benutzer liegt es frei, ob er sich beim System anmeldet oder nicht. Das Sammeln der Tomaten ist auch ohne Anmeldung möglich. Ohne der Anmeldung hingegen, können dem Benutzer seine bisher gesammelten Punkte nicht angezeigt werden. Falls der Benutzer noch kein Profil bei Blue-Tomato hat, wird er zur Registrierung an blue-tomato.com geleitet. Nach der Anmeldung in der Applikation, kann er im UserProfil seine Benutzerdaten ändern und seinen bisherigen Punkteverlauf verfolgen.

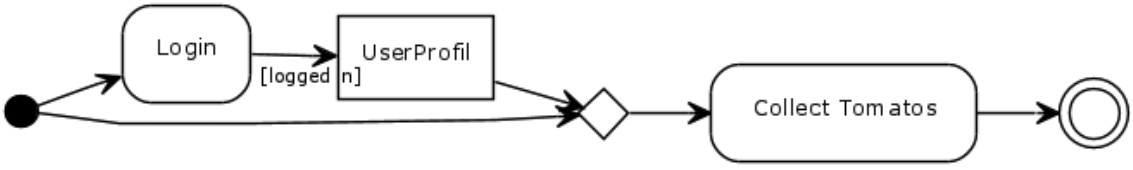

Abbildung 5.5: Login Activity

Newsletter Aktivität

Der Benutzer wird auf dem SmartPoster aufgefordert, sein Mobiltelefon an den NFC-Tag zu halten. Sobald die Applikation, die am NFC-Tag befindliche NDEF-Nachricht entschlüsselt hat, wird der Benutzer zum Newsletter-Formular geleitet. Nach Bestätigung der Email-Adresse wird die Adresse auf Duplikate und Richtigkeit geprüft. Dem Benutzer wird am Display eine Nachricht ausgegeben, ob die Email-Adresse korrekt eingegeben wurde oder nicht. Falls der Benutzer durch Anmeldung beim System bereits in der Newsletter Kartei gefunden wurde, wird der Button „Anmeldung beim Newsletter“ deaktiviert.

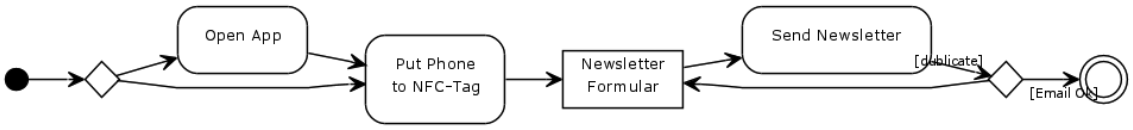

Abbildung 5.6: Newsletter Activity

5.4.4 Sequenzdiagramm

Da der Benutzer auch über andere Portale(Webshop, Kassen, dritte Anbieter) die Möglichkeit hat Bonuspunkte zu sammeln, muss ein Konzept entwickelt werden, dass vor unbefugten Zugriffen schützt. Bonuspunkte gelten als Zahlungsmittel und dürfen deshalb nicht leichtsinnig vergeben werden. Dem Webservice obliegt die Punkteverwaltung und es entscheidet, wer (Benutzer/Programm) und wieviel ein Punkte für seine Aktivität bekommt.

Login

Der Benutzer muss sich mit seiner Email-Adresse und seinem Passwort bei der Applikation anmelden. Voraussetzung ist, dass der Benutzer zuvor im System registriert ist. Das Webservice muss wissen, wer die Anfrage gestellt hat, deshalb wird zusätzlich der Name der Applikation mitgeschickt, um dadurch einen Hashwert zu generieren. Dieser Hashwert wird bei jeder Anmeldung neu erstellt und in der SQL-Datenbank abgelegt. Wenn der Hashwert abläuft, muss sich der Benutzer neu anmelden. Als Rückgabewert wird ein JSON-Objekt generiert das die Daten des Benutzers und den Hashwert beinhaltet. Die Daten werden anschließend in einer SQLite Tabelle in der Applikation hinterlegt.

Scoreboard

Damit der Benutzer weiß, wie sein Punktestand zustande kommt, werden ihm seine Punkte im Scoreboard angezeigt. Zur Abfrage reicht der Hashwert, da dieser vorher vom Webservice generiert wurde. Über diesen Hashwert werden die Punkte vom

Benutzer in der Tabelle abgefragt, um daraus ein JSON-Objekt zu generieren. Die Punkte werden nicht im Programm gespeichert, sondern jedesmal bei Bedarf, vom Webservice abgefragt.

Update

Damit immer der aktuelle Punktestand im Kundenkonto angezeigt wird, muss der Benutzer seine Punkte vom Webservice abfragen. Mit dem Hashwert wird die Summe alle Punkte von der Datenbank abgefragt und ein JSON-Objekt generiert. Die Informationen aus dem JSON-Objekt werden im Benutzerprofil in einer SQLite-Tabelle gespeichert.

Transfer Points

Nachdem der Benutzer alle seine Tomaten gesammelt hat, kann er diese über das Webservice einlösen. Das Webservice überprüft den Hashwert und die übergebene Aktivität. Abhängig von der Aktivität kann der Benutzer diese einmal oder auch mehrmals einlösen. Das Webservice übernimmt die Verwaltung der Aktivitäten. Falls eine falsche Aktivität angegeben wurde, oder das Programm nicht die Rechte besitzt diese Aktivität auszuführen, wird eine Fehlermeldung zurückgeschickt. Wenn die Aktivität zu oft ausgeführt wurde, wird ebenfalls eine Fehlermeldung ausgegeben. War die Anfrage korrekt, entscheidet das Webservice wieviel Punkte für diese Aktivität vergeben werden. Die Punkte werden anschließend in einer Punktetabelle gespeichert. Im JSON-Objekt wird eine Benachrichtigung für den Benutzer angezeigt und die Tomaten werden aus der Applikation gelöscht. Der Benutzer kann wieder von vorne beginnen die Tomaten zu sammeln.

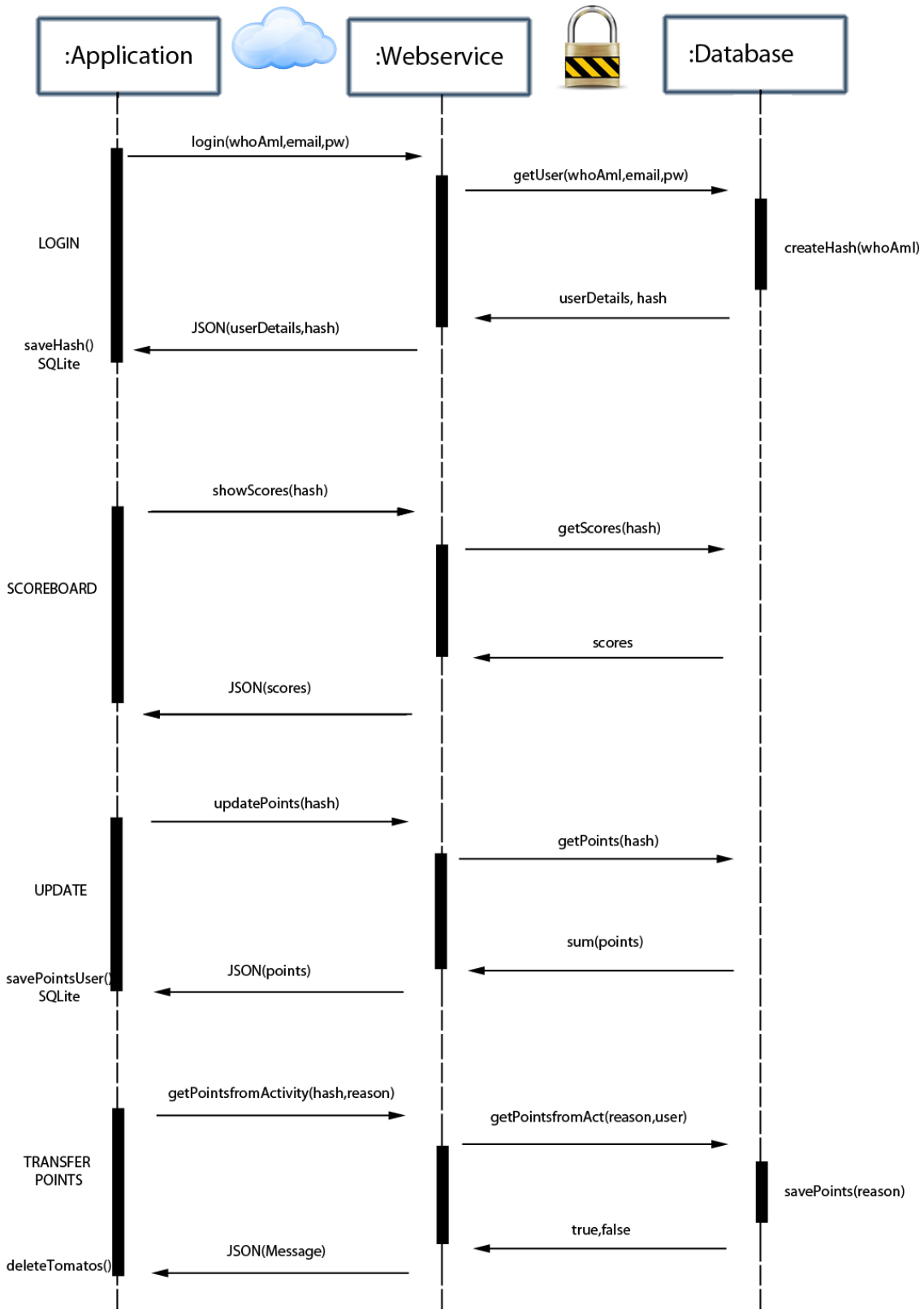

Abbildung 5.7: Sequenzdiagramm

Kapitel 6

Implementierung

6.1 Android Versionen

Googles Android ist ein Linux-basierendes Betriebssystem, welches hauptsächlich für mobile Endgeräte wie Smartphones und Tablets entwickelt wurde. Android ist Open Source und der Code steht unter der Apache Lizenz, was den Herstellern und Benutzern erlaubt, die Software zu verändern und zu verbreiten. Entwickelt wurde das Betriebssystem von der Open Handset Allicance, die neben Google, aus 84 anderen Mitgliedern (Stand July 2013) besteht [All].

Die Haupt-Versionen sind neben den Versionsnummern auch durch einen Namen gekennzeichnet, der sich nach dem Alphabet und einer Süßigkeit ableiten lässt [Goo13a].

Die erste Android-Version (1.0) wurde am 23. September 2008 veröffentlicht. Danach folgten die Versionen 1.5 (Cupcake), 1.6 (Donut), 2.0.x (Eclair), 2.2.x (Froyo), 2.3.x (Gingerbread), 3.x.x (Honeycomb), 4.0.x (Ice Cream Sandwich) und 4.1.x Jelly Bean. Die aktuelle Jelly Bean Version hat die Nummer 4.2.2 und wurde am 12. Februar 2013 veröffentlicht.

Abb. 6.1 [Goo13b] zeigt die relative Verteilung der Android Versionen von Android-Geräten im Allgemeinen. Die Grafik stellt sich aus Daten der Check-ins in den Google Play Store zusammen, die das Ökosystem von Android besser darstellen sollen.

In Abb. 6.2 sieht man den direkten Vergleich zu den Android-Versionen der Bergfex LITE Applikation.

Version	Codename	API	Distribution
1.6	Donut	4	0.1%
2.1	Eclair	7	1.4%
2.2	Froyo	8	3.1%
2.3.3 - 2.3.7	Gingerbread	10	34.1%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	23.3%
4.1.x	Jelly Bean	16	32.3%
4.2.x		17	5.6%

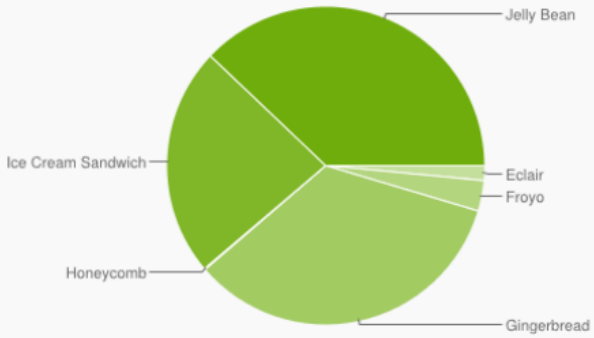

Abbildung 6.1: Verteilung von Android Versionen (Stand July,2013)

6.2 Android und NFC

Mit Einführung von Gingerbread am 6.Dezember 2010 in der Android Version 2.3 wurde das erste Mal NFC in das Betriebssystem integriert. Dem Benutzer wurde es dadurch möglich, NFC-Tags eingebettet in Postern, Stickern oder Werbeanzeigen zu lesen.

Wie schon weiter oben erwähnt, ist Gingerbread(API Level 9) die erste Version mit NFC-Unterstützung. API Level 9 unterstützt nur einzelne Funktionen von Tag Dispatch, wie z.B. ACTION_TAG_DISCOVERED. Hier war der Zugriff auf NDEF Nachrichten nur über EXTRA_NDEF_MESSAGE möglich und keine anderen Eigenschaften oder I/O Operationen waren zugänglich. Erst API Level 10 bekam volle Unterstützung für den NFC-Transceiver und förderte das Lesen und Schreiben sowie das NDEF foreground pushing[Goo13h].

Mit der Android-Version *Ice Cream Sandwich* konnten das erste Mal Kontakte, Daten oder Links von Smartphone zu Smartphone übertragen werden . Diese Technologie wurde auf den Namen „Android Beam“ getauft. Diese Version bietete einen einfacheren Weg um NDEF Nachrichten zu generieren und an andere Geräte zu senden. Darauf aufbauend, wurde in der API Version 16 „Android Beam 2.0“ vorgestellt [Goo13d].

```
<uses-sdk android:minSdkVersion="10"/>
```

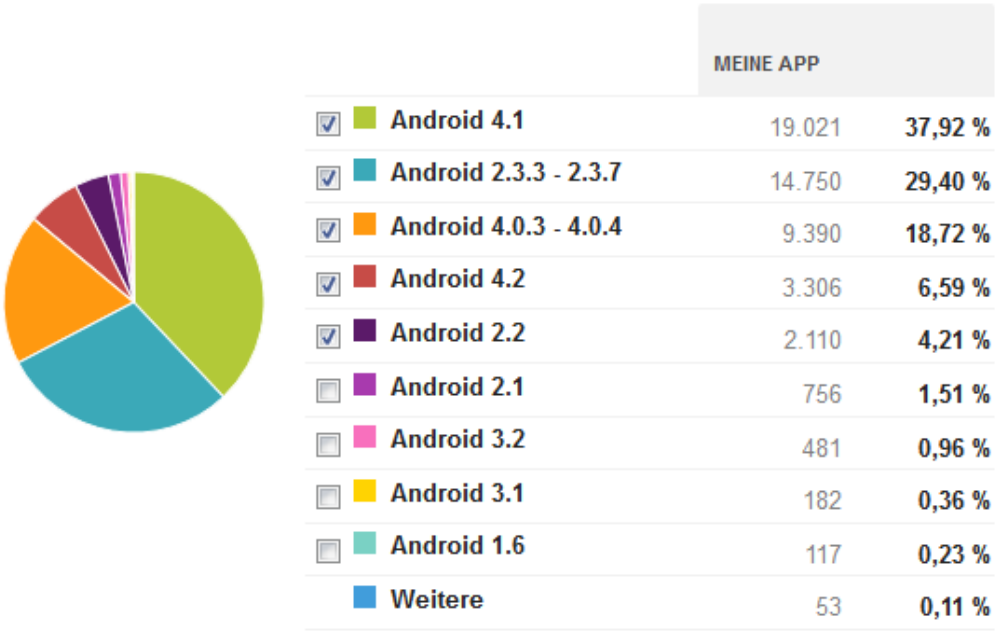

Abbildung 6.2: Verteilung von Android Versionen der Bergfex LITE Applikation (Stand July,2013)

6.2.1 Tag Dispatch System

Im Normalfall sucht ein Android-fähiges Mobiltelefon, sobald das Display entsperrt ist, nach NFC-Tags, außer NFC ist in den Einstellungen deaktiviert (6.3). Wichtig beim Lesen von NFC-Tags ist es, dass Android sofort erkennt, welche Applikation am besten damit umgehen kann, ohne den Benutzer um Eingabe oder Bestätigung zu bitten. Da die Funktechnologie einen relativ geringen Abstand zum Tag benötigt, müsste der Benutzer das Gerät für die Interaktion vom Tag bewegen und dies kann zu Verbindungsproblemen bis hin zum Verbindungsabbruch führen. Das Tag Dispatch System [Goo13h] ist dafür zuständig, dass Android sofort die richtige Aktion durchführt und keine Interaktion notwendig ist.

Sobald das Tag Dispatch System einen Intent aus den Informationen des Tags gebaut hat, wird der Intent an eine Applikation gesendet die damit umgehen kann. Wenn mehr als eine Applikation diesen Intent behandeln kann, wird der Activity Chooser (Abb.6.4) aufgerufen und eine Interakation durch den Benutzer ist von nöten.

Das Tag Dispatch System definiert 3 Intents:

1. ACTION_NDEF_DISCOVERD: Dieser Intent wird zum Starten einer Activity

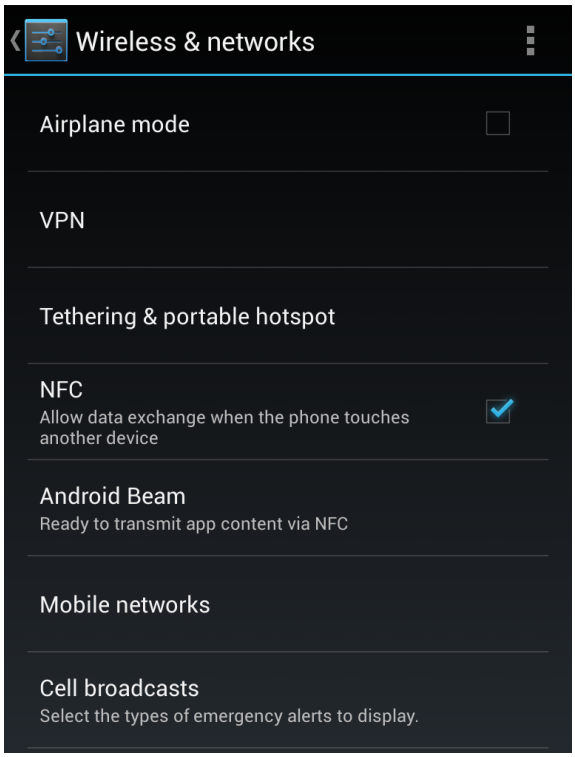

Abbildung 6.3: NFC Einstellungen

verwendet, wenn die NDEF Payload eines Tags von einem bekannten Typ ist. Dieser Intent besitzt die höchste Prioriät und das Tag Dispatch System versucht immer diesen Intent zu bevorzugen.

2. ACTION_TECH_DISCOVERED: Wenn keine Activities registriert sind um mit diesem Intent umzugehen, versucht das Tag Dispatch System eine Applikation damit zu starten. Dieser Intent wird auch direkt gestartet, falls die NDEF-Nachricht keinen MIME oder URI Typen zugeordnet werden kann, oder der Tag keine NDEF Nachricht besitzt, aber von einer bekannten Technologie ist.

3. ACTION_TAG_DISCOVERED: Dieser Intent wird gestartet, falls keine Acitvity mit ACTION_NDEF_DISCOVERD oder ACTION_TECH_DISCOVERED umgehen kann.

Vorgehensweise:

1. Versuche eine Activity zu starten, deren Intent beim Einlesen vom Tag vom Tag Dispatch System erzeugt wurde.

2. Falls sich keine Activities finden, die mit diesem Intent umgehen können, versuche eine Activity zu starten, deren Intent eine geringere Priorität besitzt

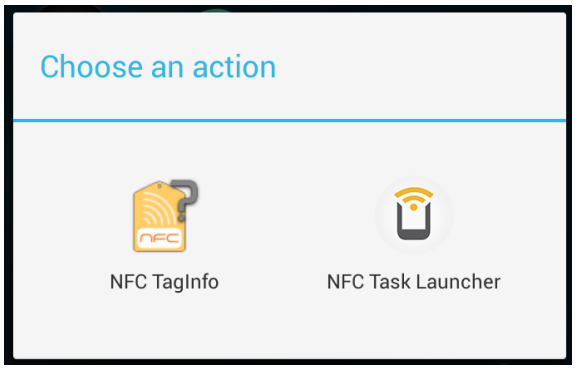

Abbildung 6.4: Activity Chooser

bis sich eine Applikation findet, oder das Tag Dispatch System alle Intents ausprobiert hat.

3. Falls sich keine Applikationen finden, höre auf zu suchen.

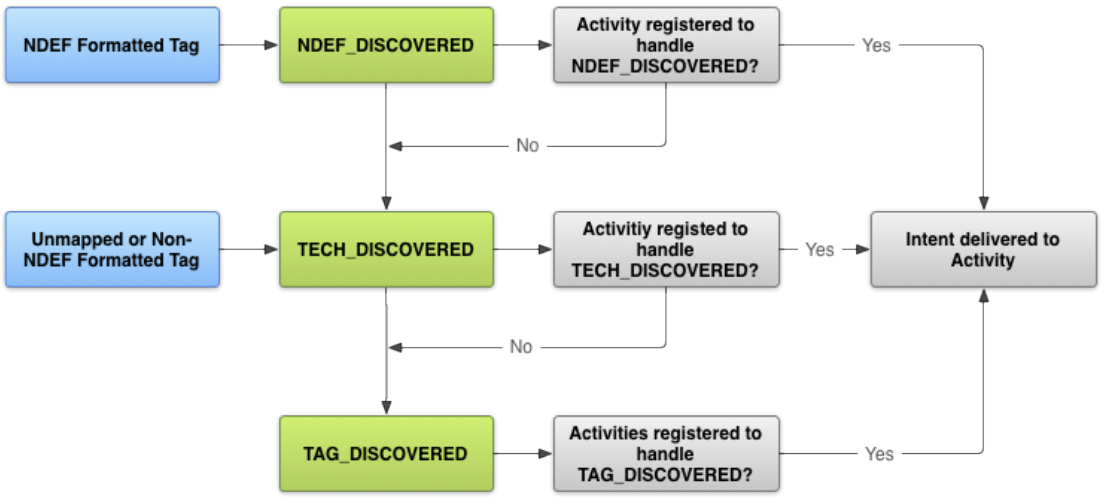

Abbildung 6.5: NFC Tag Dispatch System

Um Intents vom Typen ACTION_NDEF_DISCOVERED zu filtern, muss der Filter im Android Manifest deklariert werden. Beim Datentyp handelt es sich hier um einen MIME Typen:

```
<intent-filter>
  <action android:name="android.nfc.action.NDEF_DISCOVERED" />
  <category android:name="android.intent.category.DEFAULT" />
  <data android:mimeType="application/vnd.at.bergfex.proto" />
</intent-filter>
```

6.2.2 Rechte der Applikation

Damit die NFC-Applikation Zugriff auf die Hardware des Geräts erhält, müssen im Android Manifest (*AndroidManifest.xml*) die Rechte vergeben werden.

```
<uses-permission android:name="android.permission.NFC" />
```

Die Manifest XML[Goo13e], die jedes Android Programm besitzen muss, enthält die wichtigsten Informationen über die Applikation. Ohne diese Informationen, führt das Betriebssystem keinen Code der Applikation aus. Es beschreibt:

- den Namen des Java Pakets
- die Komponenten der Applikation
- den Prozess und welche Komponenten er besitzt
- welche Rechte die Applikation besitzen muss, um Zugriff auf die geschützten Teile der API zu erhalten
- welche Rechte andere Applikationen brauchen, um mit den Komponenten zu interagieren
- welches minimale Level der Android API die Applikation besitzen muss
- die Bibliotheken der Applikation

Im Manifest.xml der Applikation sind alle Rechte die benötigt werden, um bestimmte Teile der API in der Anwendung zu nutzen, festgehalten. Falls sich durch die Integration von NFC in ein bestehendes System, die Rechte der Applikation ändern, können keine automatischen Updates mehr durchgeführt werden. Erst wenn der Benutzer den neuen Richtlinien zustimmt, wird die Applikation installiert. Dies kann umgangen werden, indem statt der Rechtevergabe in der Manifest-Datei der NFC Adapter zur Laufzeit überprüft wird.

```
nfcAdapter = NfcAdapter.getDefaultAdapter(this);

if(nfcAdapter != null){

                Intent nfcintent = getIntent();
                String payload = getPayload(nfcintent);
                openTasks(payload);

}
```

6.2.3 Google Play Store

Der Google Play Store wurde das erste Mal unter dem Namen Android Market 2008 veröffentlicht. Es ist eine cloud-basierte Plattform und bietet Spiele, Musik, Bücher, Filme und Anwendungen an. Die Software ist standardmäßig auf den meisten Android Geräten vorinstalliert und ist für das Herunterladen und Installieren von Applikationen zuständig. Voraussetzung für die Verwendung des Google Play Stores ist ein bestehendes Google-Konto. Seit der Version 3.3.11 ist es möglich, dass alle Anwendungs-Updates automatisch eingespielt werden. Damit keine Kosten über den Netzbetreiber anfallen, kann eingestellt werden, dass nur Updates bei Verbindung mit einem WLAN heruntergeladen werden. Mittlerweile gibt es über 850.000 [Sta13] Applikationen im Play Store und damit mehr als beim Konkurrenten Apple(App Store).

In der Manifest-Datei jeder Applikation können bereits mehr als 160 Berechtigungen gesetzt werden[Goo13f]. Solange es keine Änderungen in der Manifest.xml gibt, können automatisch Updates über den Google Play Store durchgeführt werden. Dies würde zu einem Problem führen, falls die Applikation NFC für die Ausführung benötigt. Deshalb ist es für die Integration in ein bestehendes System von NFC als zusätzlicher Dienst nicht ratsam, die Rechte in der Manifest-Datei zu setzen, sondern zur Laufzeit auf die Existenz des NFCAdapter zu prüfen. Es gibt keine Einschränken und fast alle NFC-Funktionalitäten sind trotzdem verfügbar.

6.2.4 Android Application Record

Ein wichtiger Aspekt bei der Integration von NFC ist, dass das Betriebssystem wissen muss, welche Applikation es beim Einlesen von Tags ausführen soll. Ein sogenannter AAR (Android Application Record) schafft es, dass mit sehr hoher Sicherheit die richtige Applikation gestartet wird. Ein AAR beinhaltet den Paket-Namen der Applikation innerhalb eines NDEF Records. Da Android die komplette NDEF Nachricht nach einem AAR durchsucht, kann der AAR zu jedem beliebigen NDEF Record hinzugefügt werden [Goo13h]. Wenn das Betriebssystem einen AAR findet, wird die Applikation, die durch den Paket-Namen im AAR definiert wurde, ausgeführt. Falls das Betriebssystem keine Applikation mit dem Paket-Namen findet, wird der Google Play Store aufgerufen und nach der Applikation gesucht. Wenn die Applikation im Play Store gefunden wurde, kann sie mit einem Klick auf das Gerät heruntergeladen werden.

Ein großer Vorteil von AARs ist, dass es die Ausführung von anderen Applikationen verhindert, falls diese den gleichen MIME-Typen filtern. Wenn ein Tag einen AAR beinhaltet, wird nur die Applikation aufgerufen die den gleichen Paket-Namen besitzt. Da im Play Store keine Applikation den gleichen Paket-Namen haben darf, können Programme von dritten Parteien nicht auf die Tags zugreifen.

Beim Einlesen von Tags handelt das Dispatch System nach folgender Reihenfolge:

1. Suche nach dem Intent und starte die Aktivität. Falls die Aktivität durch den Intent dieselbe ist wie die des AAR, starte die Aktivität.

2. Wenn die Aktivität durch den Intent nicht dieselbe ist wie im AAR, wenn mehrere oder keine Aktivität mit dem Intent umgehen kann, starte die Applikation die im AAR angegeben ist.

3. Falls keine Applikation mit dem AAR gestartet werden kann, gehe in den Google Play Store und lade die Applikation herunter.

Wenn die Applikation mit selbst entwickelten Tags, aber auch mit Tags von dritten Parteien umgehen soll, können auch normale Intent Filter definiert werden. Dies ist sehr nützlich falls auch andere Tags eingelesen werden, die keinen AAR beinhalten.

Ein großer Nachteil ist es derzeit, dass AARs erst ab der Android Version 4.0 (API Level 15) erkannt werden können. Dadurch werden ältere Android Versionen leider nicht unterstützt und somit auch keine Applikationen ausgeführt. Durch die Kombination von AARs und MIME Typen kann dieses Problem behoben und die größte Palette an Android Geräten unterstützt werden. Durch den Einsatz von sehr spezifischen MIME-Typen können auch andere Geräte, die nicht von Android sind, unterstützt werden.

Wenn die NDEF-Nachricht aus mehr als einem Record besteht, ist es nicht von Vorteil den Andoid Application Record als ersten Eintrag zu speichern. Das Android Betriebssystem überprüft immer den ersten Eintrag einer NDEF-Nachricht auf MIME oder URI Typen und der darauf folgende MIME Typ würde dadurch nicht erkannt werden (Abb. 6.6).

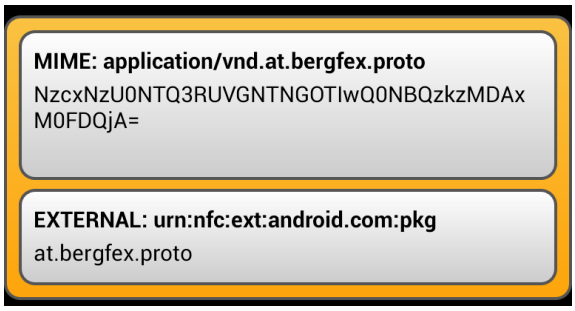

Abbildung 6.6: Android Applicatin Record

6.2.5 Foreground Dispatch

Das Foreground Dispatch System [Goo13g] erlaubt einer Aktivität den Intent zu unterbrechen und selber Anspruch auf eine Aktivität zu erheben, falls diese den gleichen Intent besitzt. Durch dieses System ist es möglich, dass der richtige Intent zur Applikation gelangt.

Als erstes muss ein PendingIntent Objekt erzeugt werden, damit das Android Betriebssystem über die Details des Tags Bescheid weiß.

```
PendingIntent pendingIntent = PendingIntent.getActivity( this , 0, new
Intent(this , getClass()).addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP), 0);
```

Danach muss der IntentFilter deklariert werden. Wenn dieser Filter mit dem Intent des Tags übereinstimmt, übernimmt die Applikation diesen Intent. Falls es keine Übereinstimmung gibt, fällt das System zurück auf das Dispatch System, wie oben weiter erwähnt. Hier wird explizit nur der MIME Type „application/vnd.at.bergfex.proto“ angegeben, um wirklich nur mit Tags von diesem Typen umgehen zu können. Falls kein Intent Filter angegeben wird, muss die Applikation mit allen Tags umgehen können, was hier nicht von Vorteil ist.

```
IntentFilter ndef = new IntentFilter(NfcAdapter.ACTION_NDEF_DISCOVERED);
            try {
                ndef.addDataType("application/vnd.at.bergfex.proto");
            }
            catch (MalformedMimeTypeException e) {
                throw new RuntimeException("fail", e);
            }

intentFiltersArray = new IntentFilter[] {ndef, };
```

Damit das System funktioniert, muss in einem Array angegeben werden, welche Technologien die Applikation unterstützt.

```
techListsArray = new String[][] {new String[]{NfcF.class.getName()}};
```

Weiteres müssen diese 3 Funktionen überschrieben werden. Diese Funktionen nennen sich *Lifecycle Callbacks* und werden aufgerufen, falls die Applikation unterbrochen oder wieder ausgeführt wird.

- onPause();
- onResume();
- onNewIntent();

Ein großer Nachteil dieses Systems ist es, dass es die NFC Rechte benötigt und damit auch nur auf NFC-fähigen Mobiltelefonen angewendet werden kann.

```
Requires the NFC permission.
```

6.2.6 Mehrfache APK-Unterstützung

Ein weiterer Ansatz, soviele Geräte wie möglich mit nur einer Applikation zu unterstützen, ist die Verwendung von mehreren APKs (Application Package File). Jede APK ist eine komplette und selbstständige Version einer Applikation, mehrere APKs werden aber unter dem gleichen Namen geführt. Das bedeutet, dass sie den selben Paket-Namen tragen und auch mit demselben Schlüssel signiert sind. Der große Vorteil an mehreren APKs ist, dass dadurch mehr Geräte unterstützt werden können, falls eine APK alleine nicht alle Geräte erreicht [Goo13l].

Durch den Einsatz von mehreren APKs ist es möglich, verschiedene:

- OpenGL-Formate zu verwenden
- Bildschirmgrößen und Rasterdichten zu unterstützten
- Platform-Versionen zu unterstützen
- CPU-Architekturen zu verwenden

Im Google Play Store wird nur ein Eintrag für die Applikation geführt, aber jedes Gerät könnte eine verschiedene Version der Applikation laden. Da die Applikation nur einmal im Play Store geführt wird, müssen auch die Details der Applikation nur einmal eingetragen werden. Das bedeutet auch, dass der Preis pro Version nicht variieren kann. Die Benutzer sehen die Applikation nur einmal im Play Store und werden nicht durch mehrere Versionen verwirrt. Auch die Bewertungen werden nur einmal pro Applikation abgegeben, es können keine Bewertungen pro Version abgegeben werden. Falls das Gerät ein System-Update auf eine höhere Version bekommt, wird die Version aus dem Play Store geladen, die für diese Version des Betriebssystems konzipiert wurde.

Um zwischen den APKs zu differenzieren wird die Filterfunktion von Google Play verwendet. Solang sich eine APK in den Kriterien OpenGL, Bildschirmgröße, Api-Level oder CPU-Architektur nicht unterscheidet, dürfen keine mehrfachen APKs veröffentlicht werden.

Regeln, die bei der Verwendung von mehreren APKs beachtet werden müssen:

- Alle veröffentlichten APKs müssen den selben Paketnamen besitzen und mit demselben Schlüssel signiert werden
- Jede APK muss eine andere Versionsnummer besitzen
- Falls sich APKs in der Unterstützung von Geräten überschneiden, nimm die APK mit der höheren Versionsnummer

- Es kann keine APK hochgeladen werden, die eine niedrigere Versionsnummer besitzt
- Eine APK mit höherem API-Level muss auch eine höhere Versionsnummer besitzen (Für Updates sehr wichtig)

6.2.7 Filtern in Google Play

Abhängig vom Filter werden dem Benutzer nur diejenigen Applikationen im Play Store angezeigt, für die sein Gerät geeignet sind. Jede Applikation benötigt je nach Typ, andere Anforderungen vom Gerät. Neben den Versionen kann auch auf Bildschirmgröße, Konfiguration, Hardware-Eigenschaft oder benötigte Bibliotheken gefiltert werden. Die Filterkriterien die eine Applikation benötigt, werden im Manifest deklariert.

Einen relativ großen Unterschied gibt es in der Vergabe der Rechte *„uses-permission“* und der Features *„uses-feature“*. Vergibt man in den Rechten die Erlaubnis auf eine Hardware (z.B. NFC) bedeutet es nicht zwingend, dass das Gerät die Hardware besitzen muss. Falls das Gerät auf die NFC-Hardware zugreifen muss, wird ihm der Zugriff gewährt. Der Google Play Store filtert keine Applikationen auf Grund seiner Rechte [Goo13j].

```
<uses-permission android:name="android.permission.NFC"/>
```

Setzt man hingegen in den Features die Anforderung für NFC, muss das Gerät diese Hardware besitzen, sonst wird die Applikation erst gar nicht im Play Store angezeigt.

```
<uses-feature android:name="android.hardware.nfc"/>
```

Ausnahmen gibt es hinsichtlich Bluetooth oder WiFi. Falls für diese Anforderungen die Rechte im Manifest deklariert wurden, schließt der Play Store automatisch auf die Existenz der Hardware. Ist diese Hardware am Gerät nicht vorhanden, wird die Applikation im Play Store nicht angezeigt.

Abb. 6.7 zeigt den Unterschied zwischen der Vergabe der Rechte und der Features. Werden die Rechte für NFC im Manifest vergeben, werden keine Geräte dadurch ausgeschlossen. Setzt man voraus, dass das Gerät die Hardware besitzen muss, werden dadurch viele Geräte ausgeschlossen. Eine Lösung wäre die Verwendung von mehreren APKs.

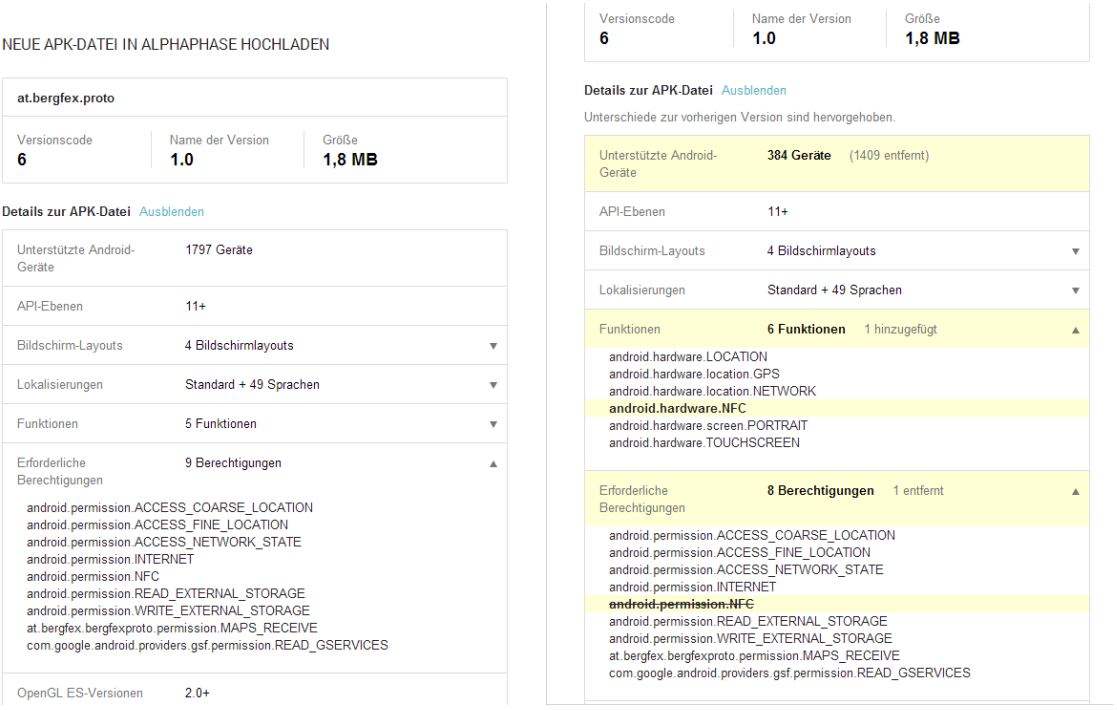

Abbildung 6.7: Unterschied Rechte und Features

6.3 Analyse

NFC-fähige Mobiltelefone

Insider behaupten, dass NFC in der Zukunft immer mehr an Bedeutung gewinnen wird. Dies macht sich auch in den Zahlen bemerkbar, da es immer mehr NFC-fähige Mobiltelefone am Markt gibt. Besonders durch die Fähigkeit, NFC als Zahlungsmittel einzusetzen, veranlasst Produkthersteller diese Technologie voranzutreiben. Mittlerweile gibt es von fast jedem Hersteller, Geräte die NFC integriert haben (Tabelle 6.1). Auch das Betriebssystem spielt hier eine große Rolle. Der Anteil an NFC-Geräten die das Betriebssystem Android nutzen, ist derzeit noch am höchsten.

Google brachte in ihrem ersten Mobiltelefon (Nexus S), welches von Samsung hergestellt wurde, NFC auf den Markt (Android). Kurz darauf brachte Samsung mit ihrem Galaxy S II eine Version heraus, die NFC-Funktionalität beinhaltete.

Gerät	**OS**
Samsung Galaxy S III	Android
Samsung Galaxy Note II	Android
Google Nexus 7	Android
Google Nexus 4	Android
Samsung Galaxy S IV	Android
HTC One X	Android
Sony Xperia Z	Android
LG Optimus L7	Android
Samsung Galaxy S II NFC	Android
Sony Xperia T	Android
Nokia Lumnia 920	WindowsPhone
LG Optimus 4X HD	Android
Nokia Lumnia 610	WindowsPhone
BlackBerry Z10	BlackBerry 10
Motorola DROID RAZR i	Android
HTC Desire C	Android
Google Nexus 10	Android
Blackberry Bold 9900	BlackBerry OS
Google Nexus S	Android
Samsung Galaxy Ace II	Android
Asus PadFonce	Android

Tabelle 6.1: Populäre NFC-Smartphones

In Abb. 6.8 kann man sehen, dass mittlerweile 14 Prozent der Geräte am Markt bereits die Technologie NFC integriert haben. In dieser Grafik wurden die Zugriffe der Webseite bergfex.com der letzten 2 Monate (Juli 2013, August 2013) analysiert. Diese Analyse beinhaltet alle mobilen Geräte, die über diesen Zeitraum die mobile Seite bergfex.com über ein Smartphone oder Tablet besucht haben.

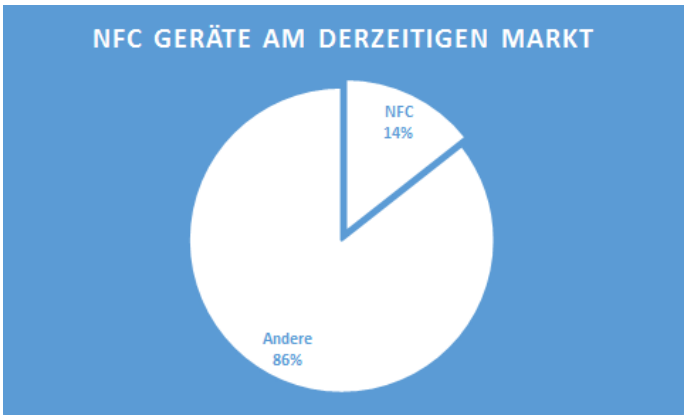

Abbildung 6.8: NFC-fähige Geräte am Markt

Bergfex Android Applikation

Betrachten wir nur die Bergfex LITE-Applikation für Anroid-Betriebssysteme können wir erkennen, dass bereits im letzten Halbjahr eine große Anzahl an NFC-fähigen Geräten hinzugekommen ist. Bedeutende Gerätehersteller wie Samsung und Google, aber auch HTC oder Sony setzten bereits in ihren neuesten Geräten auf diese Technologie. Analysiert man die derzeit installierten Applikationen auf den Android Geräten, kann man feststellen, dass bereits im Februar mehr als 26 Prozent der Geräte NFC integriert hatten (Abb. 6.9). Noch interessanter ist der Fakt, dass der Anteil an NFC-fähigen Geräten im letzten halben Jahr auf bis zu 33 Prozent angewachsen ist (Abb. 6.10). Das bedeutet, dass mittlerweile jedes dritte Android-Smartphone ein NFC-fähiges Gerät ist. Dies ist erklärbar durch die neuesten Geräte wie Samsung Galaxy 4, Nexus 4, Xperia Z aber auch HTC One X, die im Frühjahr 2013 auf den Markt kamen.

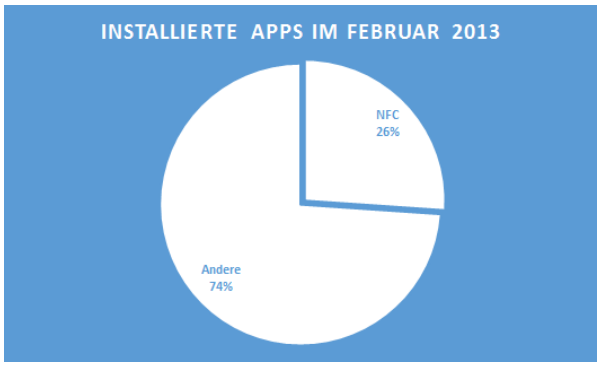

Abbildung 6.9: Anteil NFC im Februar 2013

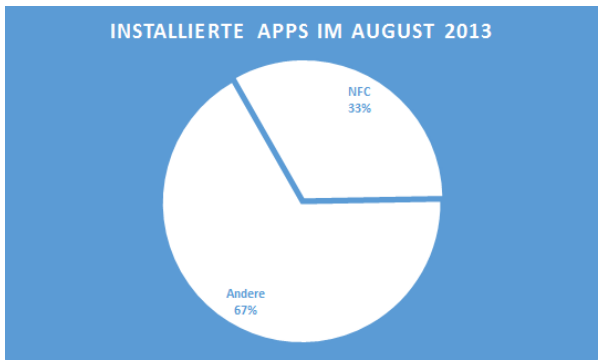

Abbildung 6.10: Anteil NFC im August 2013

Betriebssysteme

Der Gerätehersteller Apple mit seinem Betriebssystem iOS spielt im Bereich NFC noch eine untergeordnete Rolle. Obwohl der Anteil an Apple Geräten am Markt relativ hoch ist, es aber noch keine Geräte gibt, die diese Technologie unterstützen, werden sie aus dieser Analyse ausgeschlossen. In Abb. 6.11 kann man jedoch sehr schön sehen, wie hoch der Anteil der Betriebssysteme am derzeitigen Markt ist. In dieser Grafik wurden die Zugriffe von mobilen Geräten des letzten halben Jahres auf die Webseite bergfex.com dargestellt.

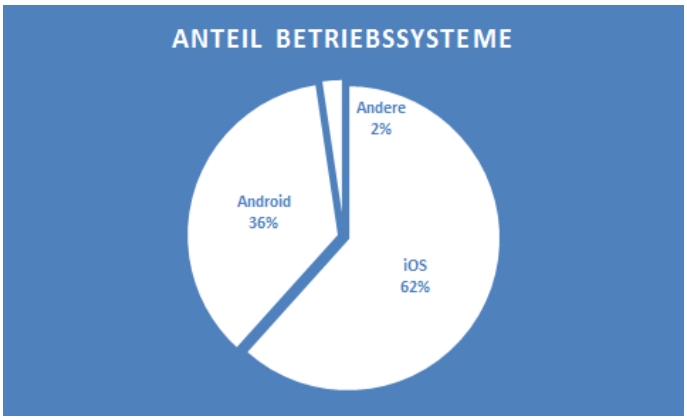

Abbildung 6.11: Anteil der Betriebssysteme am Markt

Kapitel 7

Vorstellung des Prototypen

7.1 Entwicklungsumgebung

7.1.1 Android Developer Tools (ADT)

Android bietet eine eigens entwickelte Entwicklungsumgebung an, genannt „Anroid Developer Tools (ADT)“. ADT ist ein Plugin für Eclipse und verfügt über die volle Java IDE um Applikationen zu bauen, testen, debuggen und zu bündeln. Die „Android Developer Tools“ sind open-source und laufen auf allen gängigen Betriebssystemen. Die ADT wird in der Version 22.0.5 verwendet [Goo13c].

7.1.2 Software Development Kit (SDK)

Die Android SDK Tools beinhalten alle API Bibliotheken um Applikationen zu bauen, testen und debuggen. Je nach unterstützter API Version, muss diese vorher heruntergeladen werden. Das SDK besteht aus mehreren Modulen die über den SDK Manager (Abb. 7.2) zur Verfügung gestellt werden [Goo13i].

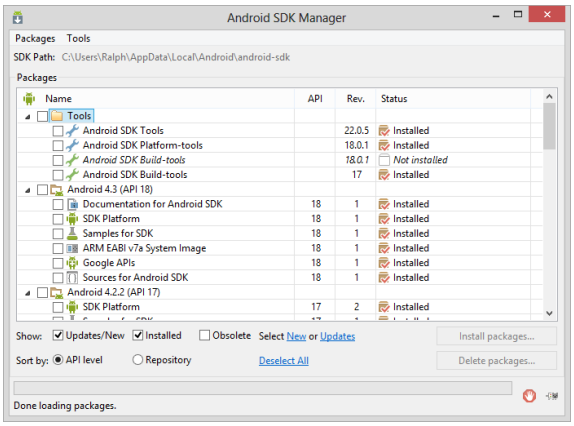

Abbildung 7.1: SDK Manager

7.1.3 Eclipse

Eclipse ist ein Editor zum Programmieren von Software. Eclipse ist open-source und unterstützt neben Java noch viele weitere Programmiersprachen. Es wird in der Version 4.2.2 verwendet [Ecl13].

7.1.4 Virtual Device

Anstatt dem mitgeliefertem Virtual Device von Google wird hier das Programm VirtualBox von Oracle [Ora13] verwendet. VirtualBox liefert bessere Leistung beim Testen von Applikationen. Dazu muss ein ISO-Abbild von Google Coder heruntergeladen und eingebunden werden. Die VirtualBox wird in der Version 4.2.14 verwendet. Das ISO-Abbild für Android hat die Version android-x86-4.2-20130228.

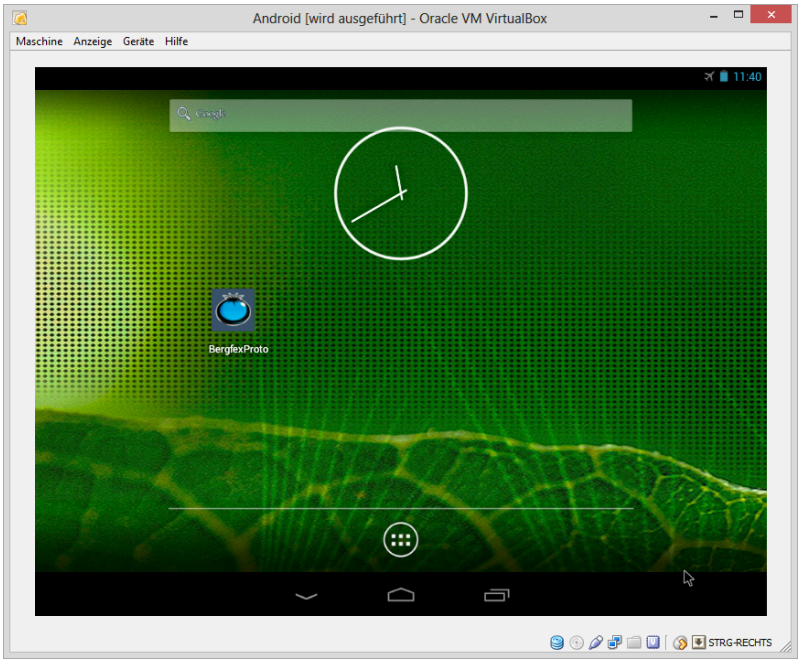

Abbildung 7.2: Virtual Device (VirtualBox)

7.1.5 Webservice

Als Webserver dient ein Apache in der Version 2.2.14 [Apa13].

7.1.6 Datenbank

Um die Daten auf dem Server zu speichern, dient eine MySQL Datenbank in der Ubuntu-Version 5.1.67 [MyS13].

7.2 Hardware

7.2.1 Test-Gerät

Für das Testen des Prototypen wird das Nexus 4 [Goo13m] von LG verwendet. Es wurde am 29.Oktober 2012 vorgestellt und entstand aus der Kooperation von Google und LG. Das Nexus 4 ist das erste Anroid Smartphone mit einem NFC Controller Chip von Broadcom (BCM20793). Alle bisherigen Controller Chips wurden von NXP hergestellt. Broadcom ist somit der bisher dritte Anbieter von NFC Controller Chips am Markt, neben NXP und Inside Secure. Ein Nachteil des Broadcom Controller Chips ist es, dass keine Mifare Classic NFC-Tags unterstützt werden. Dies bedeutet, dass es nicht möglich ist, mit einem Nexus 4, Mifare Classsic-Tags zu lesen oder zu beschreiben. Mifare Classic Tags machen von eigens entwickelten Protokollen Gebrauch und sind somit nicht NFC-Forum konform. Dies ist von Nachteil, da Mifare Classic Tags heutzutage sehr weit verbreitet sind.

Das Nexus 4 macht zusätzlich vom NFC Software Stack Gebrauch, der von Broadcom entwickelt wurde. Der Broadcom Stack ist herstellerunabhängig und ist kompatibel mit den NFC-Forum Controller Interface Spezifikationen. Er gehört zum Anroid Open Source Projekt und ist für alle Gerätehersteller frei verfügbar [Bro13].

Technische Daten Nexus 4:

Anzeige	4,7“ Diagonale 1280 x 768
Digitalkamera	8.0 Megapixel
Betriebssystem	Android 4.3
Prozessor	Qualcomm Snapdragon S4 Pro
RAM	2 GB
Interner Speicher	16 GB
Funkverbindungen	NFC, Miracast, Bluetooth, WLAN

Tabelle 7.1: Google Nexus 4 von LG

7.2.2 NFC-Tags

Um ein Smartposter zu simulieren werden Tags des Typ2 gemäß NFC-Forum verwendet. Es handelt sich hier um Tags der Sorte NXP Mifare Ultralight C und NXP NTAG 203.

Der NTAG203 wird von NXP Semiconductors entwickelt und entspricht den NFC Forum Type 2 Spezifikationen. Der Communication Layer entspricht dem ISO/IEC 14443A Standard. Er wird speziell für den Einsatz mit NFC-fähigen Geräten entwickelt. Anwendungsgebiete: Smart Advertisment, Bluetooth-pairing, WiFi proteced set-up, Call request, SMS, Device authentication [NXP11].

MIFARE ist eine bekannte Marke von NXP für eine große Auswahl an IC-Produkten. Die typische Reichweite für das Lesen und Schreiben von Mifare-Tags beträgt 10 cm. Einsatzgebiete für den Mifare Ultralight C: Public Transportation, Eventticketing, Prepaid Applications, Loyalty Cards, Toy and Amusement [NXP13].

Technische Daten:

Reichweite	1-30mm
Funktionen	7-byte Seriennummer, Lesen/ Schreiben, Schreibschutz, Wiederbeschreibbar
Form	Armband (ca. 23,5mm x 15mm (Chip: 28mm), Karte (85mm x 54mm), Key (30mm x 40mm), Sticker (30mm), Sticker (35mm x 35mm)
Material	Armband (Silikon), Aufkleber (Papier/ PET), Karte (ABS/ Kunststoff), Schlüsselanhänger (ABS/ Kunststoff), Schutzhülle (Papier/ abschirmende Aluminiumschicht)
Frequenz	HF 13.56MHz
Standards	ISO 14 443-2 A, ISO 14 443-3 A, NFC-Forum Typ 2
Formatierung	NDEF
Speicherkapazität	NXP Mifare Ultralight C - 192 Byte (144 Byte (NDEF: 137 Byte) nutzbar), NXP NTAG 203 - 168 Byte (144 Byte (NDEF: 137 Byte) nutzbar)
Chipsatz	NXP Mifare Ultralight C (MF0ICU2) (NFC-Forum Typ 2), NXP NTAG 203 (NTAG203)

Abbildung 7.3: Technische Daten NFC-Tag [nts13]

7.3 Prototyp

7.3.1 Beschreiben der Tags

Die Tags des Typen NXP NTAG 203 wurden ohne Daten geliefert und mussten vor der Integration in ein SmartPoster beschrieben werden. Dafür dient eine eigens entwickelte Applikation die auf dem Nexus 4 ausgeführt wird. Die Applikation ist mit dem Foreground Dispatch System ausgestattet und erkennt, falls ein Tag über die Applikation eingelesen wird. Die Applikation meldet mittels Vibration wenn sie den Tag erkannt hat. Der Benutzer hat die Möglichkeit zwei Textfelder zu beschreiben, MimeType und MimeData. Diese zwei Felder sind notwendig, um einen MimeType Record zu erstellen. Als MimeType dient der Name der Applikation und im Feld MimeData stehen die Daten bzw. die Payload. Zusätzlich wird ein Android Application Record in die NDEF-Nachricht geschrieben, um die Applikation über den Google Play Store zu laden. Der MimeType Record und der Application Record sind zu einer NDEF-Message gebündelt und werden durch Betätigung eines Buttons auf den Tag geschrieben. Die Methode createMime() ist ab dem API Level 14 und die Methode createApplicationRecord() ab dem API Level 16 verfügbar.

Verschlüsselung

Um den Tag vor unbefugten Lesezugriffen zu schützen, sind die Daten am Tag mit einer AES-Verschlüsselung versehen. Der Schlüssel hat eine Länge von 128-Bit, was den gängigen Sicherheitskriterien entspricht und wird vom Programm verwaltet.

Um den Tag zusätzlich gegen Vervielfältigung zu schützen, besteht der Schlüssel aus einer Buchstabenreihenfolge und der ID des Tags. Die ID des Tags ist einzigartig und kann nicht überschrieben werden. Die ID wird beim ersten Kontakt mit der Applikation, die den Tag beschreibt, ausgelesen. Anschließend wird die ID mit der Buchstabenreihenfolge verknüpft und als Schlüssel bereitgestellt. Nach bestätigen des Buttons werden die Daten mit dem Schlüssel signiert und auf den Tag geschrieben. Der Angreifer kann die Daten ohne dem Schlüssel weder benutzen, noch kann er Tags vervielfältigen.

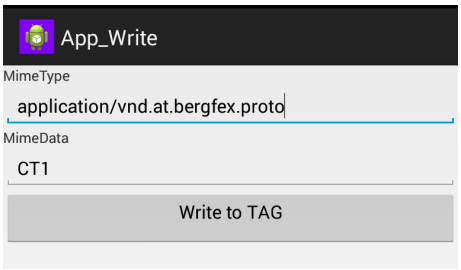

Abbildung 7.4: Applikation

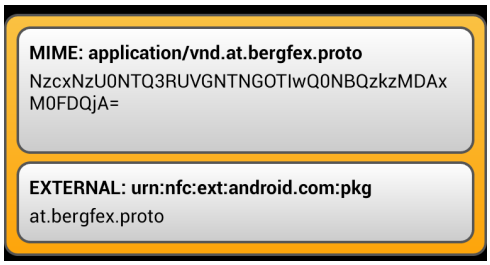

Abbildung 7.5: Daten auf dem Tag

7.3.2 Anmeldevorgang

Der Benutzer soll die Möglichkeit haben, mit aber auch ohne Anmeldung die Applikation zu bedienen. Er kann ohne Internetverbindung Punkte sammeln, jedoch können sie nicht eingelöst werden. Der Benutzer muss vor seiner Anmeldung im System registriert sein. Durch ein Anmeldeformular kann sich der Benutzer beim System anmelden. Die Daten werden über den HTTP-Client an das Webservice gesendet und dort auf Email-Adresse und Passwort überprüft. Das Passwort in der Datenbank ist verschlüsselt hinterlegt. Nach erfolgreicher Authentifizierung, wird ein JSON- Objekt an die Applikation gesendet. Das JSON-Objekt beinhaltet die Daten des Benutzers inklusive eines Hash-Wertes, der bei jeder Anmeldung neu generiert wird. Mittels diesem Hash-Wert können weitere Aktionen ausgeführt werden, ohne erneut auf Email-Adresse oder Passwort abzufragen.

Die Daten aus dem JSON-Objekt werden in eine eigene SQLite Datenbank geschrieben, welche die Applikation selber verwaltet. Erst wenn sich der Benutzer beim System abmeldet, wird er aus der SQLite Tabelle gelöscht, solange bleibt er angemeldet. Nach erfolgreicher Anmeldung wird ihm sein Punktestand im Menü angezeigt und sein Profil für Änderungen freigeschalten.

Im Menüpunkt *Scoreboard*, kann der Benutzer seine bisher gesammelten Punkte einsehen (Abb.7.6).

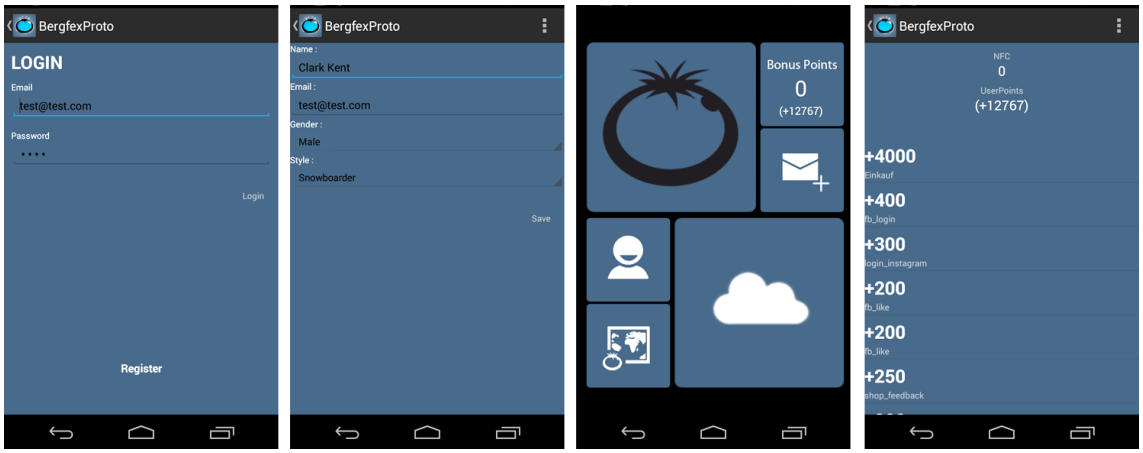

Abbildung 7.6: Anmeldevorgang

7.3.3 Anzeige der Tomaten auf einer Karte

Da die Tomaten im näheren Umkreis aufstellt sind und der Benutzer im vorhinein nicht weiß, wo sie sich befinden, bestand die Anforderung, die Tomaten auf einer Karte anzeigen zu lassen. Im Anfangszustand sind alle Tomaten mit einem schwarzen Rand versehen, ohne Füllung. Je nachdem wieviel Tomaten gesammelt wurden, werden die Tomaten in blau dargestellt.

Für die Darstellung in einer Karte wurde die Bibliothek von Google Play Services verwendet. Die Google Maps API ist ein kostenloser Google-Dienst, der für frei zugängliche Webseiten oder mobile Apps eingebettet werden kann. Die einzige Bedingung ist, dass es für Endnutzer kostenlos und öffentlich zugänglich sein muss. Sollte die Applikation nicht kostenlos angeboten werden, müsste eine Lizenz für Unternehmen bezogen werden. In der lizenzfreien Version sind bis zu 2500 Anfragen pro Tag möglich [Goo13k].

Durch das Einlesen des „geheimen“ Tags kann eine weitere Tomate in der Karte angezeigt werden. Auf diesem Tag sind die Koordinaten der versteckten Tomate verschlüsselt gespeichert. Diese werden ausgelesen, entschlüsselt und in der Karte angezeigt. Durch das Sammeln dieser Tomate können weitere Punkte verdient werden (Abb. 7.7).

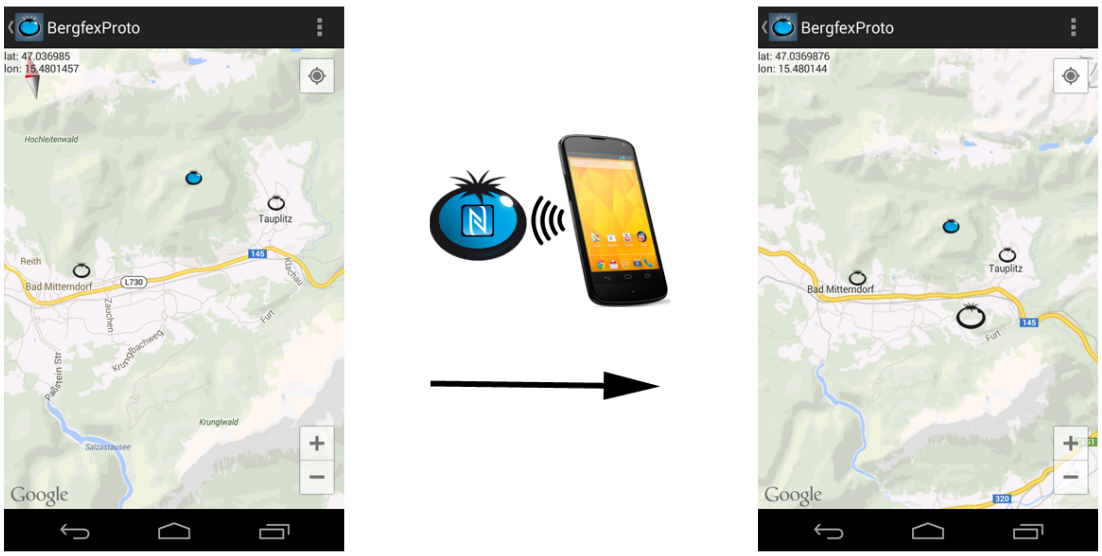

Abbildung 7.7: Versteckte Tomate

7.3.4 Sammeln der Tomaten

Das Ziel dieser Arbeit ist es, die Tags in ein Punktesystem zu integrieren. Durch das Sammeln der Tags in Form von Tomaten, kann der Benutzer Punkte für sein Konto gutschreiben. In einem weiteren Schritt, soll der Benutzer eine Vergütung für seine Punkte bekommen. Der Benutzer muss die Applikation nicht vorher gestartet haben, um die Tomaten zu sammeln, Voraussetzung ist aber die vorherige Installation der Applikation. Wie weiter oben beschrieben, erkennt das Betriebssystem die Applikation und startet sie von selbst wenn der Tag eingelesen wird. Wenn der MimeType erkannt und die richtige Applikation gestartet wird, muss die Payload aus dem Tag mittels Parser ermittelt werden.

```
String getPayload(Intent intent){

Tag myTag = (Tag) intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);

 if(myTag != null) {
        Ndef ndefTag = Ndef.get(myTag);
        // get NDEF message details
        NdefMessage ndefMesg = ndefTag.getCachedNdefMessage();
        byte[] payload = ndefMesg.getRecords()[0].getPayload();
        String st_payload = new String(payload);
        //Log.d("payload", st_payload);

         return st_payload;

        }
 return null;
}
```

Die dadurch ermittelte Payload wird entschlüsselt und an die zugehörige Aktivität gesendet. Abhängig davon, welcher Tag eingelesen wurde, wird die Tomate in den SharedPreferences der Applikation aktiv gesetzt und dem Benutzer graphisch angezeigt (Abb.7.11-2). Zusätzlich wird ein visueller Effekt beim Sammeln der Tomate angezeigt, um dem Benutzer zu animieren weitere Tomaten zu sammeln (Abb.7.11-1).

Wird eine der Tomaten gedrückt, bekommt der Benutzer einen Hinweis angezeigt, wo sich eine der nächsten Tomaten befinden könnte. Das Foreground Dispatch System verhindert, dass die Applikation neu gestartet werden muss und es können sofort weitere Tomaten gesammelt werden.

Der aktuelle Punktestand wird im Startmenü angezeigt oder kann im Punktemenü eingesehen werden. Derzeit werden pro gesammelte Tomate ein Punktewert von 1000 vergeben.

Erst wenn alle Tomaten gesammelt wurden, kann der Benutzer seine Tomaten einlösen (Abb.7.11-4). Mittels einem Button kann der Benutzer seine Tomaten an sein Kundenkonto übermitteln, jedoch muss er dafür beim System angemeldet sein (Abb.7.11-6). Ein Hashwert wird an den Server übermittelt und der Benutzer bekommt angezeigt, ob seine Transaktion erfolgreich durchgeführt wurde (Abb.7.11-4). Die Punkte werden aus der Applikation gelöscht und am Server gespeichert. Bevor die Punkte am Server eingetragen werden, wird überprüft, ob dem Benutzer die Punkte nicht schon gutgeschrieben wurden. Falls dies der Fall ist, wird eine Fehlermeldung ausgegeben (Abb.7.11-5).

7.3.5 Webservice

Das Webservice liefert pro Anfrage ein JSON-Objekt zurück. Im ersten Fall sehen wir das JSON-Objekt für den Anmeldevorgang. Es beinhaltet die Informationen des Benutzers und den generierten Hashwert der für die weiteren Operation benötigt wird.

```
{
        "tag": "login"
        "success": 1
        "error": 0
        "user":{
                "name":
                "email":
                "created_at":
                "updated_at":
                "blue_points":
                "hashcode":
        }
}
```

Dieses JSON-Objekt beinhaltet die Punkteliste des Benutzers. Diese Daten werden nach der Übermittlung im „Scoreboard“ angezeigt.

```
{
        "tag": "showscore"
        "success": 1
        "error": 0
        "user":[{
                "unique_id":
                "Points":
                "Reason":
                }
        ]
}
```

7.3.6 Wettervorhersage

Ein weiterer Dienst der Applikation ist die Anzeige der aktuellen Temperaturen über einen Tag. Der Tag weiß in welchem Skigebiet er sich befindet und übertragt seine Daten an die Applikation. Abhängig davon, in welchem Gebiet sich der Tag befindet, wird das regionale Wetter und eine Wettervorhersage angezeigt. Über den Parameter ID wird eine Webseite dynamisch erstellt und in einer WebView dargestellt. Die Wettervorhersage kann aber auch ohne scannen des Tages abgerufen werden. Dem Benutzer wird eine Liste der Skiregionen angezeigt und er kann aus diesen wählen. Abhängig von der Auswahl, werden die Wetterdaten abgerufen (Abb.7.8).

Der große Vorteil der durch die Integration von NFC erreicht wird, ist, dass der Benutzer keine Interaktion mit der Applikation benötigt. Durch den Tag lokalisiert die Applikation den Aufenthalt des Tags und zeigt die regionalen Wetterdaten an. Neben dem Skigebiet wird auch ein Token mitgegeben, damit die Applikation weiß um welchen Typ von Tag es sich handelt.

Abbildung 7.8: Ablauf Wettervorhersage

7.3.7 Datenbank

In Abb. 7.9 sieht man die Struktur der verwendeten Datenbank. Die Datenbank besteht aus 3 Tabellen: users, score und rewards.

users
Die Tabelle users verwaltet alle Details des Benutzers und deren Passwörter. Die Passwörter werden nicht als plain-text in die Tabelle geschrieben sondern verschlüsselt. Ein Salt-Wert dient zur Entschlüsselung. Eine vom Webservice generierte unique_id dient zur eindeutigen Identifizierung des Benutzers. Zusätzlich wird bei jeder Anmeldung ein Hashwert aus dem Namen der Applikation erstellt und dem Benutzer zugeordnet. Sobald der Hashwert abläuft wird er aus der Tabelle gelöscht und der Benutzer muss sich neu in der Applikation anmelden.

score
Die Tabelle score verwaltet den Punktestand des Benutzers. Zusätzlich zu den Punkten wird auch die Aktivität gespeichert, für die er die Punkte bekommen hat. Um die Punkte dem Benutzer zuordnen zu können, wird die unique_id des Benutzers verwendet.

rewards
Diese Tabelle verwaltet die Punktevergabe, sie entscheidet wieviel Punkte eine Aktivität wert ist. Bei jeder Anfrage werden die Punkte für eine Aktivität aus der rewards-Tabelle genommen und in die score-Tabelle geschrieben. Die rewards-Tabelle kann dynamisch erweitert werden.

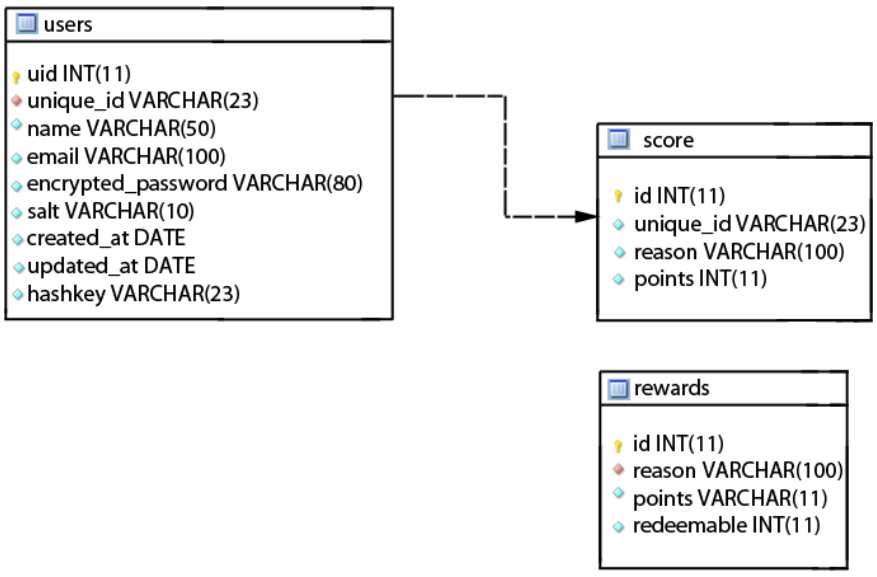

Abbildung 7.9: Datenbankmodell

Abbildung 7.10: Ablauf Tomaten sammeln

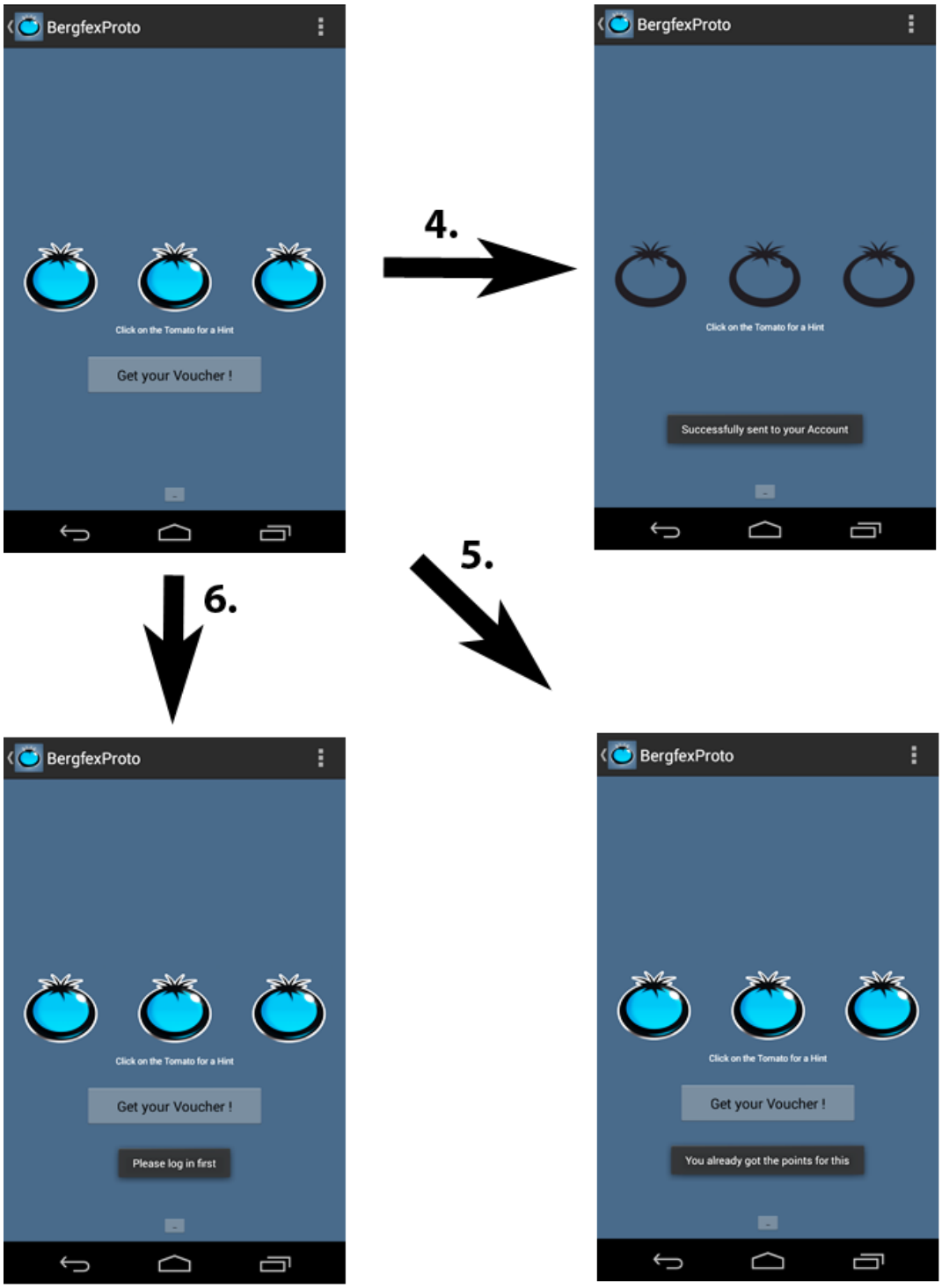

Abbildung 7.11: Ablauf Tomaten einlösen

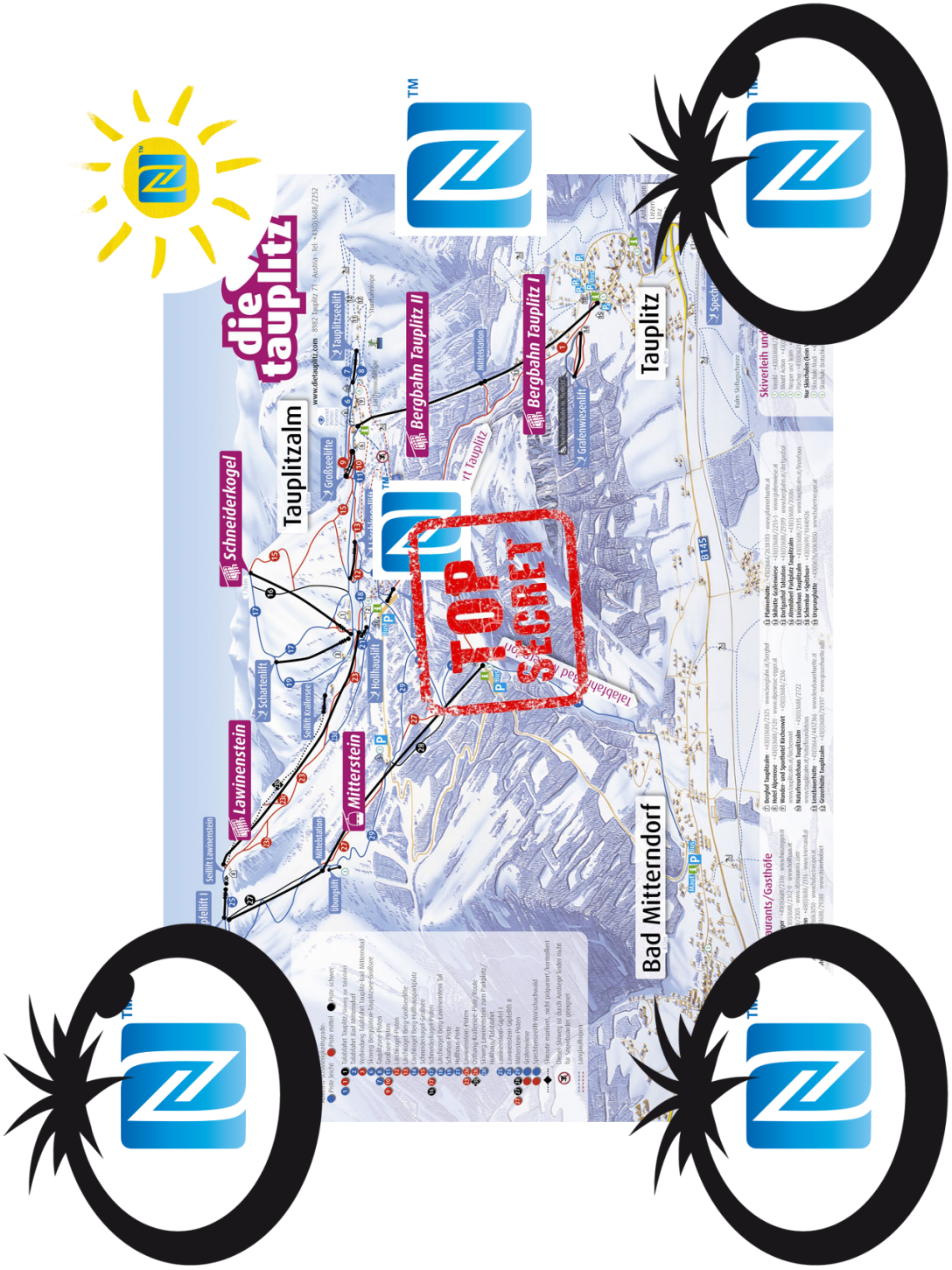

Abbildung 7.12: Smart Poster

Kapitel 8

Schlußbemerkung und Ausblick

Der Prototyp hat hervorragend gezeigt, wie durch Integration von NFC in ein bestehendes System, Interaktionen vereinfacht und Benutzerfreundlichkeit erhöht werden kann. Auch die Integration von NFC in ein Bonuspunkteprogramm hat gezeigt, dass sich in diesen Bereich eine Vielzahl von Anwendungen verbergen.

Es gibt einige Erweiterungen, von der die mobile Anwendung sehr profitieren würde. Einige davon sind Anregungen für weitere Projekte in diesem Bereich, andere sind notwendig um die Applikation kommerziell nutzen zu können.

Dynamische Erweiterung

Derzeit wurde der Prototyp als *Proof of Concept* entwickelt, um einen Einblick in die Thematik zu erlangen. Im Prototypen ist es möglich bis zu 3 Tomaten bzw. Wegpunkte zu sammeln. Wenn der Benutzer dies erreicht hat, muss er wieder von vorne beginnen. Damit die Applikation weiterhin attraktiv für den Benutzer bleibt, müssen durch Updates oder Webservices, die Wegpunkte erweitert werden. Der Nachteil von Updates ist, dass nach jeder Erweiterung, die Applikation über den Google Play Store neu geladen werden muss. Der Benutzer verliert dadurch das Interesse und deinstalliert im schlimmsten Falle die Applikation. Ein besserer Ansatz ist die Verwendung von Webservices, die über JSON-Objekte neue Wegpunkte an die Applikation liefern. Die Wegpunkte müssten eine eindeutige Kennung besitzen und in einer Datenbank eingetragen werden. Jedesmal wenn der Benutzer über die Applikation online geht, werden die Tomaten aktualisiert und sind bereit eingesammelt zu werden.

Events

Neben einfachen Wegpunkten können auch bei Veranstaltungen Tomaten gesammelt werden. Je mehr Events eine Person besucht, desto mehr Punkte bekommt sie für ihre Teilnahme.

Sicherheit

Der Tag wurde gegen unbefugte Lesezugriffe und Vervielfältigung geschützt. Die AES-Verschlüsselung ist für diesen Anwendungsfall ausreichend, jedoch könnten Verbesserungen durch andere Verschlüsselungs-Techniken erreicht werden. In die Thematik *Sicherheit* müsste noch näher eingegangen werden.

Erweiterung durch QR-Codes

Da der Anteil an NFC-fähigen Mobiltelefonen derzeit noch relativ gering ist, wird es angedacht die Applikation durch Verwendung von QR-Codes zu erweitern. QR (Quick Response)-Codes sind zweidimensionale Barcodes die gleich wir NFC-Tags überall angebracht werden können. Das einzige was dafür benötigt wird, ist ein Mobiltelefon mit integrierter Kamera, diese haben sich am Markt schon großteils etabliert. Der Nachteil an QR-Codes ist, dass für das Einlesen von Codes über die Kamera, die Applikation im Vorhinein geöffnet sein muss.

Literaturverzeichnis

[All] Open Handset Alliance. Open Handset Alliance, http://www.openhandsetalliance.com/.

[Apa13] Apache. The Apache Software Foundation, http://www.apache.org/, August 2013.

[Ber13] Bergfex. Bergfex, http://bergfex.at, September 2013.

[Bro13] Broadcom. BCM20793, http://www.broadcom.com/products/NFC/NFC-Solutions/BCM2079x-Family, August 2013.

[bt] blue tomato.com. Bluetomato onlineshop, http://blue-tomato.com.

[COO12] V. Coskun, K. Ok, and B. Ozdenizci. *Near Field Communication (NFC): From Theory to Practice*. Wiley, 2012.

[Ecl13] Eclipse. Eclipse Website, http://www.eclipse.org/, August 2013.

[For06a] NFC Forum. *NFC Data Exchange Format (NDEF), http://www.nfc-forum.org/resources/white_papers/NFCForum-TS-NDEF_1.0.pdf.* 401 Edgewater Place, Suite 600 Wakefield, MA01880, USA, 2006.

[For06b] NFC Forum. *NFC Forum Type Tags White Paper, http://www.nfc-forum.org/resources/white_papers/NXP_BV_Type_Tags_White_Paper-Apr_09.pdf.* 401 Edgewater Place, Suite 600 Wakefield, MA01880, USA, 2006.

[For06c] NFC Forum. *NFC Record Type Definition (RTD) Technical Specification, http://www.nfc-forum.org/specs/spec_license/.* 401 Edgewater Place, Suite 600 Wakefield, MA01880, USA, 2006.

[For07a] NFC Forum. *LLCP, http://www.nfc-forum.org/resources/white_papers/nfc_forum_marketing_white_paper.pdf.* 401 Edgewater Place, Suite 600 Wakefield, MA01880, USA, 2007.

[For07b] NFC Forum. *Smart Poster White Paper, http://www.nfc-forum.org/resources/white_papers/NFC_Smart_Posters_White_Paper.pdf.* 401 Edgewater Place, Suite 600 Wakefield, MA01880, USA, 2007.

[For07c] NFC Forum. *The Keys to Truly Interoperable Communications, http://www.nfc-forum.org/resources/white_papers/nfc_forum_marketing_white_paper.pdf.* 401 Edgewater Place, Suite 600 Wakefield, MA01880, USA, 2007.

[For07d] NFC Forum. *The NFC Forum N-Mark Brand Guide, http://www.nfc-forum.org/resources/N-Mark/brandguide.pdf.* 401 Edgewater Place, Suite 600 Wakefield, MA01880, USA, 2007.

[For11] NFC Forum. *NFC in Public Transport, http://www.nfc-forum.org/resources/white_papers/NFC_in_Public_Transport.pdf.* 401 Edgewater Place, Suite 600 Wakefield, MA01880, USA, 2011.

[FT11] L. Finzgar and M. Trebar. Use of NFC and QR code identification in an electronic ticket system for public transport. In *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pages 1 –6, sept. 2011.

[Goo] Google. How to nfc, http://www.google.com/events/io/2011/sessions/how-to-nfc.html.

[Goo13a] Google. Android Build Version, http://developer.android.com/reference/android/os/Build.VERSION_CODES.html, July 2013.

[Goo13b] Google. Android Dashboard, http://developer.android.com/about/dashboards/index.html, July 2013.

[Goo13c] Google. Android Developer Tool Eclipse ADT, http://developer.android.com/tools/sdk/eclipse-adt.html, August 2013.

[Goo13d] Google. Android Jelly Bean, http://www.android.com/about/jelly-bean/, July 2013.

[Goo13e] Google. Android Manifest, http://developer.android.com/guide/topics/manifest/manifest-intro.html, July 2013.

[Goo13f] Google. Android Manifest Permissions, http://developer.android.com/reference/android/Manifest.permission.html, July 2013.

[Goo13g] Google. Android NFC Foreground Dispatch, http://developer.android.com/guide/topics/connectivity/nfc/advanced-nfc.html#foreground-dispatch, July 2013.

[Goo13h] Google. Android NFC, http://developer.android.com/guide/topics/connectivity/nfc/nfc.html, July 2013.

[Goo13i] Google. Android SDK, http://developer.android.com/sdk/exploring.html#Packages, 2013.

[Goo13j] Google. Filter on Google Play, http://developer.android.com/google/play/filters.html, August 2013.

[Goo13k] Google. Google Maps API, https://developers.google.com/maps/licensing?hl=de, August 2013.

[Goo13l] Google. Multiple APK Support, http://developer.android.com/google/play/publishing/multiple-apks.html, August 2013.

[Goo13m] Google. Nexus 4, http://www.google.de/nexus/4/, August 2013.

[Int13] ECMA International. *TNear Field Communication Interface and Protocol -2 (NFCIP-2), http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-352.pdf.* Rue du Rhone 114, CH-1204 Geneva, 2013.

[inv] investincotedazur.com. Nfc: Major samsung visa partnership announced at the mwc 2013, http://investincotedazur.com/en/newsletter/nfc-major-samsung-visa-partnership-announced-at-the-mwc-2013&artid=act11035.

[inv10] investincotedazur.com. Nice City of contactless mobile - Official launch on May 21st, 2010, http://investincotedazur.com/en/newsletter/nice-city-of-contactless-mobile-official-launch-on-may-21st-2010?artid=act9257, May 2010.

[LR10] Josef Langer and Michael Roland. Anwendungen und Technik von Near Field Communication (NFC), 2010.

[mob11] mobilemarketingmagazine.com. RIM Scores MasterCard NFC Certification, http://mobilemarketingmagazine.com/content/rim-scores-mastercard-nfc-certification, September 2011.

[MyS13] MySql. Die populärste Open-Source-Datenbank der Welt, http://www.mysql.de/, August 2013.

[NF03] NFC-FORUM. Nokia, Philips And Sony Establish The Near Field Communication (NFC) Forum, http://www.nfc-forum.org/news/pr/view?item_key= d8968a33b4812e2509e5b74247d1366dc8ef91d8, 05 2003.

[NF06a] NFC-FORUM. NFC Forum Publishes Specification For SmartPoster Records, http://www.nfc-forum.org/news/pr/view?item_key= d58874aa69a4e57f7ce2314af283a41b372833e7, August 2006.

[NF06b] NFC-FORUM. NFC Forum Unveils Technology Architecture And Announces Initial Specifications And Mandatory Tag Format Support, http://www.nfc-forum.org/news/pr/view?item_key= 0b210bbd23e9c1a07cb3d975e6317d1d650ed51f, 06 2006.

[nfc11] nfcworld.com. Nokia releases Symbian Anna NFC update, http://www.nfcworld.com/2011/08/18/39164/nokia-releases-symbian-anna-nfc-update/, August 2011.

[Nie00] Dr. Jakob Nielsen. *Killing Time Is The Killer App, http://www.thefeaturearchives.com/8183.html*, 2000.

[Nok13] Nokia. *Nokia 6131 NFC, http://nds1.nokia.com/phones/files/guides/Nokia_6131_NFC_UG_de.pdf*, 08 2013.

[nts13] nfc-tag shop.de. Nexus 4 Starter Kit, http://www.nfc-tag-shop.de/nfc-starter-kits/android/26/nfc-google-nexus-4-samsung-galaxy-blackberry-starter-kit-big-18-artikel, August 2013.

[NXP11] NXP. *NFC Forum Type 2 Tag compliant IC with 144 bytes user memory, http://www.nxp.com/documents/short_data_sheet/NTAG203_SDS.pdf*, 2011.

[NXP13] NXP. *MIFARE Ultralight EV1 - contactless ticket IC, http://www.nxp.com/documents/short_data_sheet/MF0ICU2_SDS.pdf*, 2013.

[Ora13] Oracle. Virtual Box, https://www.virtualbox.org/, August 2013.

[RE96] W. Rankl and W. Effing. *Handbuch der Chipkarten.: Aufbau - Funktionsweise - Einsatz von Smart Cards.* Hanser Fachbuch, 1996.

[RSH04] E. Rukzio, A. Schmidt, and H. Hussmann. Physical posters as gateways to context-aware services for mobile devices. In *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*, pages 10–19, 2004.

[Sam12] Samsung. SAMSUNG Mobile Expands NFC Capabilities with TecTile Version 3.0, ttp://www.samsung.com/us/news/20301, October 2012.

[Son13] Sonymobile. Xperia SmartTags, http://www.sonymobile.com/at/products/accessories/xperia-smarttags/, September 2013.

[SSVARGN12] J.J. Sanchez-Silos, F.J. Velasco-Arjona, I.L. Ruiz, and M.A. Gomez-Nieto. An NFC-Based Solution for Discount and Loyalty Mobile Coupons. In *Near Field Communication (NFC), 2012 4th International Workshop on*, pages 45–50, 2012.

[Sta13] Statista. Anzahl der verügbaren Apps im Google Play Store bis 2013, http://de.statista.com/statistik/daten/studie/74368/umfrage/anzahl-der-verfuegbaren-apps-im-google-play-store/, July 2013.

[Wal83] Charles A. Walton. Portable Radio Frequency Emitting Identifier, 05 1983.

[WGSL12] R. Widmann, S. Grunberger, B. Stadlmann, and J. Langer. System Integration of NFC Ticketing into an Existing Public Transport Infrastructure. In *Near Field Communication (NFC), 2012 4th International Workshop on*, pages 13–18, 2012.

[ZL11] Huijuan Zhang and Junlin Li. NFC in medical applications with wireless sensors. In *Electrical and Control Engineering (ICECE), 2011 International Conference on*, pages 718 –721, sept. 2011.