FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Unterstützung von Facility-Management Prozessen mit NFC-Smartphones am Beispiel eines Gebäudereinigungsdienstes

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Sebastian Paral
Matrikelnummer 0225674

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gerald Futschek

Wien, 02.10.2014

(Unterschrift Verfasser/in) (Unterschrift Betreuer)

Erklärung zur Verfassung der Arbeit

Sebastian Paral
Hauptstraße 45/2/13, 3001 Mauerbach

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser/in)

Danksagung

Mein Dank gilt meinen Eltern *Johann und Christa Paral*, welche mir die Möglichkeit gaben zu studieren und somit zum Abschluss dieses Studiums verhalfen.

Meinem Betreuer, *Prof. Dr. Gerald Futschek*, welcher mir die Möglichkeit zur Erarbeitung dieses Themas gab, möchte ich ebenfalls herzlichst danken. Er hat mir in seinen Sprechstunden und Seminaren stets konstruktive Kritik gebracht.

Das gemeinsame Lernen und das zuverlässige Erarbeiten von Aufgabenstellungen in den Lehrveranstaltungen dieses Studiums durfte ich mit meinem langjährigen Freund und Studienkollegen *Christian Behon* genießen. Danke auch hierfür!

Ich möchte mich auch bei *Oliver Weidel* dafür bedanken, dass er mir beim Korrekturlesen der Arbeit stets mit wertvollen Ratschlägen zur Seite stand.

Auch gilt mein Dank allen Personen, die sich zum Usability-Test bereitgestellt haben. Auch jenen, die es nicht freiwillig gemacht haben :-)

Danke auch an meine Freundin, *Sandra Schreiblehner*, welche auf viel gemeinsame Zeit verzichtet und immer an mich geglaubt hat.

Kurzfassung

Mit Integration der *Near Field Communication (NFC)* Technologie zur kontaktlosen Erfassung von Daten auf Smartphones ergeben sich neue Anwendungsmöglichkeiten für Industrie und Privatanwender. Diese Forschungsarbeit widmet sich der mobilen Unterstützung eines im infrastrukturellen Gebäudemanagement beheimateten Facility-Management (FM) Prozesses, dem eines Gebäudereinigungsdienstes.

Die Dokumentenerfassung zur Unterstützung der Prozesse wurde bislang mit Hilfe stationärer CAFM-Systeme bewerkstelligt. Dabei handelt es sich um Systeme zur Erfassung und zur Verwaltung von FM-Daten.

Mobile Systeme sind gerade in dieser Branche begehrt, besonders zur Arbeitszeit- und Leistungskontrolle. Es soll die Frage beantwortet werden, inwieweit es möglich ist mit Hilfe eines Smartphones und dem Einsatz der NFC-Technologie den FM-Prozess eines Gebäudereinigungsdienstes geeignet zu unterstützen, zu optimieren und alle papierbasierten Prozesse zu ersetzen. Kann der gesamte Dokumentationsprozess ohne Einschränkung der Benutzerfreundlichkeit auf einem Smartphone stattfinden?

Leistungserbringer, *Auftragnehmer* und *Endnutzer* sind jene drei Rollen, welche am Geschäftsprozess aktiv teilnehmen. Es wurde untersucht, welche mobilen Softwarelösungen bereits existieren und welche Technologien zur mobilen Datenerfassung im FM bereits eingesetzt werden. Auf Basis der untersuchten Lösungen entstand ein Konzept für ein auf NFC basierendes mobiles System für Smartphones zur Unterstützung der FM-Prozesse. Dabei wurde die Rolle des Endnutzers in die Nutzung der NFC-Infrastruktur aktiv miteinbezogen. Auftragnehmer werden durch den Einsatz des Systems beim Qualitätssicherungsprozess unterstützt, und Leistungserbringern wurde zusätzlich zur Protokollierung auch die Möglichkeit einer Selbstkontrolle auf Basis der Bewertungsrichtlinien gegeben.

Das Konzept basiert auf der Vision des *Internet of Things*, welches sich mit der Integration von realen Objekten in das Internet befasst. Die Existenz von drei NFC-Tag Gruppen in Zusammenarbeit mit einem Webservice, welcher die Anbindung an das CAFM-System simuliert, ermöglichte die Realisierung eines Smartphone Prototyps für das *Android*-Betriebssystem.

Durch die notwendige Bedingung NFC-Tags zu schützen konnte in Verbindung mit einer Authentifizierungsprozedur auf NDEF-formatiereten NFC-Tags diese Problematik im *Reader/Writer* Modus gelöst werden. Personal-Tags werden für die Anmeldung verwendet, Raum-Tags werden zur Identifikation von Räumlichkeiten eingesetzt und Aktions-Tags sind notwendig um Arbeitsaktionen tätigen zu können.

Da die Benutzerfreundlichkeit eines solchen Systems eine wichtige Rolle spielt, wurde das erarbeitete Konzept und somit der umgesetzte Prototyp Probanden zum Usability-Test vorgelegt. Hiermit sollten Mängel erfasst werden um das Konzept und den Prototyp entsprechend zu überarbeiten.

Die zentralen Ergebnisse dieser Forschungsarbeit waren, dass die Probanden durchwegs der Meinung waren, die Nutzung eines solchen Systems jedenfalls der papierbasierten Variante zu bevorzugen.

Abstract

While the integration of *Near Field Communication (NFC)* technology for contactless detection of data on smartphones arises, a new scope of applications for industry and consumers can be established. This research thesis is dedicated to aiding infrastructural building management processes in the context of facility management (FM) of a building cleaning service.

Until now the collection of documents to aid the processes has been accomplished with stationary CAFM systems, which are systems to gather and manage FM data.

Mobile systems are in a high demand, especially in the FM industry. They can be used for working controls and quality assurance. This thesis should answer the question what extent it is possible with the help of a smartphone and the use of the NFC technology to optimize, assist and replace all paper based processes. Can the entire documentation process take place on a smartphone without restricting its usability?

Service providers, *contractors* and *endusers* are the three roles that actively participate in the business process. It was examined which mobile software solutions already exist and which technologies are being used for mobile data collection in FM. Based on the reviewed solutions, a concept for a mobile NFC based smartphone system has been formed to support the FM processes. The role of the enduser has been actively involved in the use of the NFC infrastructure. Contractors will be supported by the use of the system during the quality assurance process and providers have the possibility of self-control based on valuation guidelines in addition to log their work.

The concept is based on the vision of the *Internet of Things* which deals with the integration of real objects into the Internet. The existence of three NFC tag groups in cooperation with a web service which simulates the connection to the CAFM system, allowed the realization of a smartphone prototype for the *Android* operating system.

The essential condition to protect NFC tags could be achieved in conjunction with an authentication procedure on NDEF formatted NFC tags. For this purpose the reader/writer mode was used. Personal-tags are used for registration, room-tags are used for identification of premises and action-tags are necessary to be able to make work actions.

Since the usability of such a system plays an important role, the developed concept and thus the implemented prototype were submitted to a usability test with probands. Conceptual and prototyped issues should be covered to revise them accordingly.

The main results of this research show that the probands consistently favoured the use of such a system over paper based systems.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

AQL	Accepted Quality Level	19
ART	Android Runtime	67
CAFM	Computer-Aided Facility Management	1
DVM	Dalvik Virtual Machine	66
FM	Facility-Management	1
IoT	Internet of Things	33
JSON	JavaScript Object Notation	54
JVM	Java Virtual Machine	66
MVC	Model View Controller	68
NDEF	NFC Data Exchange Format	8
NFC	Near Field Communication	5
ORM	Object Relational Mapping	52
PCD	Proximity Coupling Device	7
PICC	Proximity Integrated Circuit Card	7
REST	Representational State Transfer	54
RFID	Radio Frequency Identification	5
TNF	Type Name Format	12
UID	Unique Identification Number	11
WSDL	Web Service Description Language	53
WSGI	Web Server Gateway Interface	52

KAPITEL 1

Einleitung

Durch die zunehmende Anzahl an Smartphones am Markt, welche mit Near Field Communication (NFC) Technologie ausgestattet werden, eröffnen sich neue Möglichkeiten für Industrie und Privatanwender. NFC ist ein Standard zum kontaktlosen Austausch von Daten über kurze Strecken mittels Funktechnik.

Diese Technologie kann beispielsweise im *Facility-Management (FM)*, welches sich mit der Verwaltung von Liegenschaften, Anlagen und Einrichtungen befasst, zum Einsatz kommen.

Computer-Aided Facility Management (CAFM) bezeichnet die Ausführung von FM-Prozessen, welche durch den Einsatz der Informationstechnologie (IT) unterstützt werden. Mobile Technologien sind begehrt und können effizient in CAFM integriert werden [L13]. Die vorliegende Arbeit behandelt einen Kernbereich des infrastrukturellen Gebäudemanagements, den FM-Prozess eines Reinigungsdienstes, um die Frage zu behandeln, inwieweit es möglich ist, diese Prozesse mit Smartphones und NFC zu unterstützen und zu optimieren. Hierfür sollen bestehende Lösungen analysiert werden und auf dessen Basis ein innovatives Konzept für einen Smartphone-Prototyp entstehen um ein neuartiges mobiles System vorzustellen.

1.1 Motivation und Problemstellung

Im infrastrukturellen Gebäudemanagement werden Reinigungsprozesse und dessen Qualitätssicherung im sogenannten Standardleistungsverzeichnis (Standard-LV) zusammengetragen. Die abzuarbeitenden Prozesse und Dienstleistungen wurden bisher größtenteils durch zumeist zeitaufwändige und teure Verwaltung in Form von Papier aus dem Standard-LV und einem zugrundeliegenden CAFM-System erfasst. Diese Dokumente umfassten Informationen betreffend der anstehenden Tätigkeiten und sind nach Vergabe an die Dienstleister durch die Tatsache, dass sie auf Papier vorliegen, nicht mehr änderbar oder erweiterbar. Aufgrund der großen Relevanz, dass Reinigungsdienstleistungen und dessen Qualitätssicherung möglichst nebenläufig und somit auch unbemerkt vonstatten gehen sollen, stellen sich kurzfristig eintretende Ausnahmen wie z.B. die Belegung eines Konferenzraums zu einem bestimmten Zeitraum als problematisch

heraus und müssen somit anderweitig kommuniziert werden. In den Protokollen und Dokumenten sollen zusätzlich zu den ausgeführten Tätigkeiten auch Auffälligkeiten wie etwa Beschädigungen und andere besondere Vorkommnisse vermerkt und entsprechend protokolliert werden. Auch ein nachträgliches Archivieren des somit entstandenen Protokolls auf elektronischen Datenträgern ist nur mit zusätzlicher, meist fehleranfälliger Arbeit verbunden.

Mittels sogenannten NFC-Tags können jegliche Arten von Objekten – sei es ein Raum, eine Anlage, aber auch Dienstkräfte sind nicht ausgeschlossen – versehen werden um eine eindeutige Identifikation und ggf. eine Bestimmung des aktuellen Standortes der Dienstkraft mittels des Lesegerätes bewerkstelligen zu können. Beim Scannen eines Objektes können durch einen zentralen Server Informationen bereitgestellt und übermittelt werden [L15]. Beispielsweise kann ein fotografischer Nachweis einer Beschädigung eines Objektes durch das Lesegerät in Echtzeit übermittelt werden und somit automatisiert ein Reparaturauftrag angefertigt werden. Auch der genaue Arbeitsablauf kann mit NFC unterstützt und auch unverfälscht dokumentiert werden [L13].

Ein dynamisches, in Echtzeit einsehbares Leistungsverzeichnis, welches Informationen über das betreffende Objekt und dem Arbeitsprozess bereitstellen soll und zusätzlich ein Protokoll der Tätigkeiten und dessen Ausnahmezustände speichern und archivieren kann, würde den besagten FM-Prozess hinsichtlich den vorgestellten Beispielen geeignet unterstützen und optimieren. Das Ziel ist es, nicht nur FM-Daten abzurufen, sondern interaktiven Zugriff zu ermöglichen.

Ziel dieser Forschungsarbeit ist es, die Frage zu beantworten, inwieweit es möglich ist mit Hilfe eines Smartphones unter Einsatz der NFC-Technologie den FM-Prozess am Beispiel eines Gebäudereinigungsdienstes geeignet zu unterstützen und zu optimieren. Der Benutzer soll vom eigentlichen Aufgabenziel durch die Systembenutzung nicht abgelenkt werden. Kann durch den Einsatz eines solchen mobilen Systems erreicht werden, dass die zu erledigenden Aufgaben ohne schriftliche Protokollierung und somit durch weniger Aufwand mit mehr Komfort erledigt werden können als zuvor, und kann ein solcher NFC-gestützter Lösungsansatz den gesamten Dokumentationsprozess ohne Einschränkung der Benutzerfreundlichkeit auf mobilen Geräten ersetzen?

Da es noch sehr wenig Praxiserfahrung zur Integration von Smartphones für die Unterstützung der FM-Prozess gibt, ist es die Herausforderung dieser Arbeit, zu zeigen, dass diese mobilen Systeme auch in das Facility-Management sinnvoll integriert werden können.

1.2 Vorgehensweise und Forschungsmethodik

Als Ausgangsbasis und zur Vermittlung des Grundwissens werden im Abschnitt 2.1 die Funktionsweise und der Ursprung, der in dieser Arbeit relevanten NFC-Technologie dem Leser näher gebracht. Hierbei gilt es auch die Grenzen der Technologie klarzustellen. Auf Basis der im Abschnitt 2.2 vermittelten Grundlagen der Reinigungsorganisation können in weiterer Folge die verschiedenen Rollen und Abläufe, welche am Geschäftsprozess teilnehmen, präsentiert werden. Mit Klarstellung der Relevanz mobiler Systeme im behandelnden FM-Prozess sollen verschiedene Technologien zur mobilen Datenerfassung, die im FM bereits zum Einsatz kommen, beschrieben werden (2.3).

Durch Recherche sollen im Abschnitt 2.3 aktuelle Lösungen identifiziert werden, welche sich dem Problem annehmen um in weiterer Folge eine Anforderungsliste erstellen zu können. Eine Sammlung von Anforderungen für das Eigensystem soll entstehen, indem alle relevanten Kriterien zusammengetragen werden um anschließend eine Gegenüberstellung der Systeme durchführen zu können. Diese Anforderungen bilden sich aus Eigenschaften der Systeme, welche durch einen hohen Nutzungsgrad herausstechen und/oder Charakteristiken eines neuartigen Systems enthalten.

Das zu erstellende innovative Konzept, welches im darauffolgenden Kapitel 3 beschrieben wird, soll den vollständigen FM-Prozess abbilden um eine Smartphone-Applikation erstellen zu können. Der praktischen Realisierung und Umsetzung dieses Konzeptes auf einem *Android*-Smartphone soll sich Kapitel 4 widmen. Es werden die Technologien und die verschiedenen Komponenten für das Gesamtsystem sowie dessen Herausforderungen aufgegriffen.

Letztendlich soll ein Usability-Test darüber Auskunft geben, ob der umgesetzte Prototyp auf Akzeptanz trifft und ob eine optimale Bedienung durch Probanden bestätigt werden kann. Mängel sollen hierbei durch die Bewältigung verschiedener Aufgabenstellungen und Fragebogen für Probanden identifiziert werden. Anschließend sollen diese einer Diskussion unterzogen werden um weiters eventuell notwendige Verbesserung zur Implementierung oder Anpassungen am Konzept durchführen zu können.

Das abschließende Kapitel 6 befasst sich zu guter Letzt mit noch notwendigen Schritten zur Erlangung einer Marktreife des Prototyps, sowie die in dieser Arbeit relevante Zusammentragung von Ergebnissen zur Beantwortung der Forschungsfrage.

KAPITEL 2

Grundlagen

Das nachfolgende Kapitel soll den aktuellen Stand der Themengebiete dieser Arbeit vermitteln.

Mit einem Einblick in den Ursprung von NFC, welcher stark mit jenem von RFID zusammenhängt, sollen einerseits deren Funktionsweisen und Standards näher beschrieben werden, andererseits soll auch der klare Unterschied zwischen diesen beiden Technologien vermittelt werden.

In weiterer Folge wird detaillierter auf NFC, dessen Einsatzmöglichkeiten, Formate und Grenzen eingegangen. Im Zusammenhang mit Smartphones soll dem Leser vermittelt werden, warum sich NFC zunehmend an Beliebtheit erfreut und warum diese Technologie für die vorliegende Arbeit gewählt wird.

Dafür wird zunächst auf den Dienstleistungsprozess der Reinigung im Kontext des Facility-Managements eingegangen und ein Überblick der aktuellen Gegebenheiten und dessen Entwicklungen präsentiert, indem die IST-Situation aus Sicht der verschiedenen Rollen beschrieben wird.

Darauffolgend wird der wissenschaftliche Aspekt dieser Arbeit erläutert, und anschließend werden bestehende mobile Systeme vorgestellt, welche im Reinigungsmanagement bereits zum Einsatz kommen. Durch Evaluierung dieser soll letztendlich eine Funktionalitäten-Liste für das Eigensystem erarbeitet werden, welches im Kapitel 3 konzeptioniert und im Kapitel 4 implementiert werden soll.

2.1 RFID & NFC

In den letzten Jahren ist im Zusammenhang mit *Radio Frequency Identification (RFID)* ein neuer Begriff aufgetaucht. *Near Field Communication (NFC)* und RFID werden oft in demselben Atemzug als dieselbe Technologie dargestellt. Die Wurzeln von NFC liegen bei jenen von RFID [O18]. Mit NFC ist es wie bei RFID möglich Tags zu lesen und zu beschreiben; NFC kann aber noch mehr. Es kann als eine Erweiterung von RFID angesehen werden, welche auf dessen Standards aufbaut und eine breitere Plattform zum Datenaustausch bietet [43].

RFID ist dabei keine Kommunikationstechnologie bei welcher sinnvolle Nachrichten ausgetauscht werden, es wurde vielmehr dazu entwickelt um Objekte zu identifizieren. RFID Tags beinhalten kleine Daten und mit RFID-Geräten ist es möglich diese zu lesen und auch zu verändern. Mit NFC-Geräten ist es ebenfalls möglich (passive) RFID-Tags zu lesen und zu beschreiben. Das Beschreiben ist aber mit Hilfe eines standardisierten Formates (2.1), unabhängig vom Tag-Typ möglich. NFC wurde für Kurzstreckenkommunikation im Bereich von bis zu 4 cm entworfen und ist daher nicht dafür gedacht eine langzeitliche Kommunikation aufzubauen. Durch die Tatsache, dass NFC *low-powered* ist, was Interferenzen mit anderen Radiobändern verhindert, ist NFC eine langsamere Technologie verglichen mit WiFi oder Bluetooth. Die angedachten Anwendungsfälle erfordern aber keinen möglichst schnellen langfristigen Datenaustausch, sondern sind vielmehr für den Austausch von Kurznachrichten angedacht. NFC-Geräte können untereinander kommunizieren und im Zwei-Wege- oder Duplex-Modus Daten austauschen, oder über andere Technologien eine langfristige Kommunikation durch Austausch von Berechtigungen aufbauen [43].

Funktionsweise von RFID

Bei RFID wird mit Hilfe eines Lese-/Schreibgerätes (Initiator) das sogenannte Target (oder auch Transponder) ausgelesen. Der Initiator erzeugt dafür ein elektromagnetisches Wechselfeld und wartet auf Antworten vom Transponder. Diese Transponder werden durch das erzeugte Feld angeregt ihre *Unique Identification Number* zum Initiator zu senden.

Bei RFID gibt es einen passiven und einen aktiven Kommunikationsmodus. Passive Transponder sind nicht-energieversorgte Einheiten (Tags) und werden erst durch das elektromagnetische Feld mit Energie versorgt und angeregt. Aktive Transponder werden durch externe Energiequellen wie etwa Batterien versorgt um eine größere Reichweite zwischen Lesegerät und Transponder zu realisieren. Aktive Tags werden oft in Kombination mit Sensoren verwendet um die gebundenen Informationen an das Lesegerät zu übermitteln.

Die enthaltenen Daten (für gewöhnlich weniger als 1 KB), welche auf dem Transponder hinterlegt sind, können vom Lesegerät ausgewertet und weiterverarbeitet werden. Nicht nur ein Auslesen der Daten ist möglich, sondern auch das Verändern von Datenblöcken kann bewerkstelligt werden. Da die meisten RFID-Systeme an ein Netzwerk gebunden sind, ist der generelle Ansatz einer index-basierten Relation (UID) gebräuchlich. Hierfür werden in einer Datenbank zur ausgelesenen UID weitere Informationen hinterlegt, welche vom Lesegerät abgerufen werden können.

Typische Anwendungsbeispiele, in denen RFID in unserem Alltag bereits Anwendung gefunden hat, sind Diebstahlschutzeinrichtungen in Kaufhäusern. Die RFID Transponder werden hierbei auf Waren angebracht und werden bei nicht sachgemäßer „Entwertung“ durch ein stationäres Lesegerät beim Verlassen des Kaufhauses auf einen korrekten Status geprüft. Sollte dieser nicht gegeben sein, wird das Lesegerät ein akustisches Signal ausgeben.

Die Technologie kann nicht nur zur Signalisierung von illegal entwendeten Gegenständen verwendet werden. Auch eine Identifikation von Lebewesen, wie etwa die von Haustieren, kann durchgeführt werden. Der RFID-Transponder wird dem Tier unter die Haut implantiert und kann durch einen Tierarzt mit dem entsprechenden Lesegerät ausgelesen werden. Eine Zuordnung zum Besitzer bei entlaufenen Tieren oder zur Krankenakte ist hiermit leicht gegeben.

Funktionsweise von NFC

Wie bereits erwähnt, handelt es sich bei NFC um eine Erweiterung von RFID, bei welcher ebenfalls ein Initiator und ein Target involviert sein müssen. Beide Einheiten teilen sich ein Radiofrequenzband auf dem sie Informationen übertragen oder empfangen (half-duplex) [O22]. Der wesentliche Unterschied zu RFID ist, dass Targets nicht nur Tags, sondern auch programmierbare Endgeräte wie Smartphones sein können. Damit ist es möglich, nicht nur statischen Inhalt an den Initiator zu senden, sondern es kann entschieden werden, welcher Inhalt zu welchem Initiator gesendet werden darf.

Bei NFC funktioniert die Art der Verbindung wie bei RFID, nur ist die Reichweite zur Versorgung der Dateneinheiten beabsichtigt gering gehalten. Das NFC Protokoll agiert auf einer Bandbreite von 13.56 MHz mit einer maximalen Datenübertragungsrate von 424 Kbits pro Sekunde [L11].

NFC-Geräte haben wie bei RFID zwei Kommunikationsmodi. Passive Kommunikation findest statt, wenn das Target durch das vom Initiator erzeugte Wechselfeld versorgt wird, aktive Kommunikation hingegen setzt an beiden Enden eine externe Energieversorgung voraus. Die Kommunikation zwischen zwei NFC-Geräten wird somit als *aktiv* bezeichnet. Das NFC-Forum [O23], welches die Implementierung und Standardisierung sowie die Kompatibilität der Geräte sicherstellen soll [O18], definiert drei verschiedene Modi, in welchen die Kommunikation stattfinden kann. Diese Modi werden im Abschnitt 2.1 näher beschrieben.

In aktuellen Dokumentationen und Manuals des ISO/IEC 14443 Standard wird die lesende Einheit als *Proximity Coupling Device (PCD)* und die RFID-Karte als *Proximity Integrated Circuit Card (PICC)* bezeichnet.

RFID & NFC Standards

RFID bietet kein universell einsetzbares Protokoll. Im Gegenteil, es gibt Dutzende davon, welche von der *International Standards Organization (ISO)* [1] in Zusammenarbeit von wesentlich in der RFID-Branche tätigen Firmen entwickelt werden. Auch wenn diese Firmen gegenseitige Konkurrenten darstellen, agiert die ISO als Vermittler um dennoch interoperable Standards entwerfen zu können. In diesen Standards werden Frequenzbereiche, Transferraten , Datenformate und weitere Informationen definiert [43].

Als Beispiel sei die zuvor beschriebene Identifikation von Haustieren zu erwähnen, welche im ISO-11784 standardisiert wurde und im Frequenzbereich zwischen 129 und 139.4 kHz agiert. Die Datenfelder bieten hierbei Informationen um das Tier zu beschreiben.

Der ISO-14443 Standard wurde entworfen um Zahlungssysteme mit *Smart Cards* zu verbinden, welche auf einem Frequenzband von 13.56 MHz agieren. Als wesentliche Merkmale sind hierbei Funktionalitäten für das Zählen (Increment und Decrement) von Werten und das Verschlüsseln von Daten zu erwähnen. Die 14443 Familie bietet verschiedene Formate wie etwa von *Philips* und *NXP Mifare*, *Sony FeliCa* und *NXP DESFire* [43]. Diese ISO-14443A Tags sind mit NFC kompatibel und werden im Abschnitt 2.1 nochmals separat im Zusammenhang mit den verschiedenen Tag-Typen behandelt.

[1] `http://www.iso.org`

Der Datenaustausch bei NFC kann durch ein einheitliches Format stattfinden. Das *NFC Data Exchange Format (NDEF)*, welches im Abschnitt 2.1 genauer beschrieben wird, ist eines der wesentlichen Erweiterungen, welche NFC gegenüber RFID bietet. Es handelt sich dabei um ein allgemeines Datenformat, welches für jedes NFC-Gerät unabhängig von der zugrundeliegenden Geräte- und Tag-Technologie eingesetzt werden kann.

NFC Operationsmodi

Wie im Abschnitt 2.1 erwähnt, kann die Kommunikation eines NFC-Gerätes in verschiedenen Modi stattfinden. Diese werden nachfolgend näher erläutert.

Reader/Writer (R/W)

In diesem Modus kann das NFC-Gerät (Initiator) Daten von einem Tag (Target) lesen oder zu selbigem schreiben. Tags sind nicht-energieversorgte Einheiten, die erst durch ein vom NFC-Gerät erzeugtes elektromagnetisches Induktionsfeld angeregt werden. Nachdem die passive Einheit mit Energie versorgt wurde, kann die Kommunikation beginnen. Die übertragenen Daten können jegliche Art von Text sein wie etwa eine Internetadresse, ereignisbezogene Daten oder andere Datensätze. Die ausgelesenen Daten können in weiterer Folge auf viele verschiedene Arten weiterverarbeitet werden. Für gewöhnlich werden sie dem Benutzer auf dem Bildschirm präsentiert bis dieser die Informationen bestätigt oder schließt [L10]. Das Hinterlegen von persönlichen Daten wie etwa einer elektronischen Visitenkarte ist in diesem Modus ein gebräuchlicher Anwendungsfall. Die Datenübertragungsrate erreicht ihr Maximum bei 106 Kbit/sec [O23].

Card Emulation (CE)

Hierbei entfällt im Gegensatz zum Reader/Writer Modus die Notwendigkeit einer passiven Einheit. Das NFC-Gerät selbst kann als Tag (Target) agieren und andere NFC-Geräte (Initiatoren) können Daten auslesen. Das Charakteristische an diesem Modus ist jene Tatsache, dass auf physische Objekte verzichtet werden kann. Im Micropayment beispielsweise kann ein Smartphone als Ersatz für Kreditkarte, Smartcard und in Folge dessen auch für Bargeld eingesetzt werden. Auch kann auf die physische Existenz eines herkömmlichen Schlüssels verzichtet werden, da elektronische Schließsysteme NFC in diesem Modus auch für Zutrittssysteme und in Folge dessen auch zur Anwesenheitskontrolle verwenden können.

Peer-to-Peer (P2P)

Im Vergleich zu den beiden oben erwähnten Modellen ist die Peer-to-peer Kommunikation zwischen zwei NFC-Geräten jener Modus, welcher am seltensten genutzt wird. [L10] bezeichnet diesen Modus als „albern“ für das Geräte-Pairing, netzwerkbezogene Anwendungen und Datenübertagung, da hierfür geeignetere Protokolle wie etwa WLAN existieren. Dennoch sind die erwähnten Szenarien durchaus zu bewerkstelligen. Der bereits angeführte Anwendungsfall bezüglich elektronischer Visitenkarten kann hier jedoch auf den gegenseitigen Austausch der Daten erweitert werden, indem die beiden NFC-fähigen Geräte aneinander gehalten werden um

gegenseitig persönliche Daten auszutauschen. Die Übertragungsgeschwindigkeit reicht hierbei bis zu 424 Kbit/sec [O23].

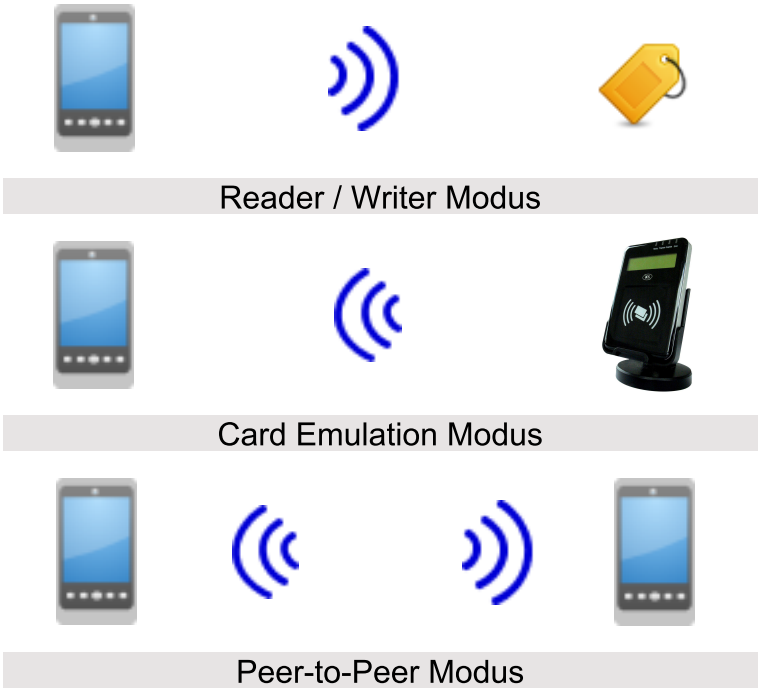


Abbildung 2.1: NFC Modi

Abbildung 2.1 veranschaulicht nochmals die verschiedenen NFC Betriebsmodi.

Typen von NFC-Tags

NFC-Tags besitzen integrierte Schaltkreise und können Daten speichern, welche von NFC-Geräten ausgelesen werden können. Um die Interoperabilität zwischen Tags und NFC-Geräten zu erhalten hat das NFC-Forum [O23] vier Typen von Tags spezifiziert. Jeder Tag-Typ kann für bestimmte Aufgaben abhängig vom jeweiligen Anwendungsfall eingesetzt werden. Die verschiedenen Typen unterscheiden sich vor allem in ihrer Speicherkapazität, sowie Verschlüsselung und in ihrem Schreibschutz. Typ 1, 2 und 4 basieren auf ISO-14443A und Typ 3 auf ISO-18092. [O6] und [O8] beschreiben diese Typen hinsichtlich ihren Eigenschaften und Unterschieden genauer:

NFC Typ-1

Dieser Tag-Typ nutzt ein einfaches Speichermodell. Es wird ein proprietäres Verbindungsprotokoll mit ISO-14443A Modulationstechnik eingesetzt. Tags dieses Typs sind für Lese-, Schreib-

und wiederbeschreibbaren Zugriff geeignet und können auch als „read-only“ konfiguriert werden. Die Speichergröße reicht von 96 Bytes bis max. 2 Kilobytes und Lese- bzw. Schreiboperationen werden mit bis zu 106 KB/s ausgeführt. Anti-Kollision ist bei dem Tag-Typus nicht gegeben. Er ist elektronisch recht einfach aufgebaut, kann daher sehr kostengünstig hergestellt werden und eignet sich damit ideal für viele NFC Anwendungen. Hersteller sind *Innovision Research & Technology TOPAZ* und *Broadcom*.

NFC Typ-2

Dieser Tag-Typ wird nur von *NXP* [2] hergestellt. Wie bei den Typ-1 Tags wird ein proprietäres Verbindungsprotokoll mit ISO-14443A Modulationstechnik eingesetzt. Handelsnamen sind etwa *MIFARE Ultralight* und *MIFARE Ultralight C*. Sie können gelesen, beschrieben, schreibgeschützt und als „read-only“ konfiguriert werden. *Mifare Ultralight* Tags haben üblicherweise nur eine Speichergröße von 48 Bytes. Sie eignen sich daher nur zur Speicherung sehr kleiner Daten, wie etwa einer URL. Diese Tags wurden ausgiebig getestet und werden als zuverlässig, sicher und als kosteneffizient angesehen. Anti-Kollisionsunterstützung ist gegeben und die Schreib/Lesegeschwindigkeit liegt bei 106 KB/s.

NFC Typ-3

Dieser Tag-Typ wird von *Sony* unter dem Namen „FeliCa“ (ISO-18092 und JIS-X-6319-4) vertrieben. Sie basieren auf dem Japanischen Industriestandard und sind vom Hersteller entweder lese-, schreib- und wiederbeschreibbar oder als schreibgeschützt vorkonfiguriert. Die Speicherkapazität reicht bis zu 1 MB, Kommunikationsgeschwindigkeit bis zu 212 oder 424 KB/s und Anti-Kollisionsunterstützung ist vorhanden.

NFC Typ-4

Diese Tags sind voll kompatibel zu ISO-14443A und ebenfalls vom Hersteller entweder lese-, schreib- und wiederbeschreibbar oder als schreibgeschützt vorkonfiguriert. Typ-4 Tags werden von *NXP* gebaut und unterstützen Anti-Kollision. Handelsnamen dieser Tags sind *NXP DESFire*, *NXP DESFire EV1* und *NXP SmartMX-JCOP*. Diese Tags sind verhältnismäßig teuer. Sie bieten ein flexibles Dateisystem mit einer Speicherverfügbarkeit bis zu 32 KB für verschiedene Daten und Zugriffsarten sowie Datenintegritätsprüfungen und Verschlüsselungsoptionen. Sie haben eine hohe Speicherkapazität, meist 4 KB oder mehr und lassen sich verschlüsseln. Die Verschlüsselung des *Mifare DESfire* Tags wurde jedoch 2011 geknackt [L12]. Derzeit ist der *Mifare DESfire EV1* die sicherste Variante um etwa Zugangsbeschränkungen zu realisieren. Kommunikation wird durch die *Application Protocol Data Unit (APDU)* Kommunikationseinheit nach ISO/IEC 7816-4 bewerkstelligt und findet mit einer Geschwindigkeit bis zu 424 KB/s statt.

Des weiteren gibt es noch von *NXP Semiconductors* proprietäre Tags, welche nicht durch das

[2] `http://www.nxp.com/campaigns/nfc/`

NFC-Forum standardisiert worden sind wie etwa der *Mifare Classic Tag*, welcher eine Speicherkapazität von bis zu 192/768 oder 3,584 Bytes und eine Kommunikationsgeschwindigkeit mit 106 KB/s bietet. Chips dieser Art sind jedoch häufig anzutreffen. Bis heute wurden davon 2 Milliarden Tags und 25 Millionen Kartenlesegeräte ausgegeben [O8]. Die Verschlüsselung des oft eingesetzten *MIFARE-Classic*-Chips basiert auf einer proprietären Stromchiffre. Dieser Algorithmus konnte von Forschern durch *Reverse Engineering* rekonstruiert werden. Es wurde ein systematischer Fehler gefunden, der die Verschlüsselung praktisch nutzlos macht [38]. Diese Tags unterstützen ebenfalls Anti-Kollision und als Beispiele sind *NXP Mifare Classic 1K/4K/Mini* zu erwähnen.

	Typ 1	Typ 2	Typ 3	Typ 4
Handelsnamen	Broadcom Topaz	NXP Mifare Ultralight, NXP Mifare Ultralight C, NXP NTAG203	Sony FeliCa	NXP DESFire, NXP SmartMX-JCOP
Typische Speicherkapazitäten	96 Bytes	48/144 Bytes	1/4/9 KB	4/32 KB
Daten Schreibschutz	R/W od. nur Lesen	R/W od. nur Lesen	R/W od. nur Lesen	R/W od. nur Lesen
Aktive Inhalte	-	-	-	NXP DESFire: Nein NXP SmartMX-JCOP: Ja
Antikollision	-	X	X	X
Einzelpreis	Niedrig	Niedrig	Hoch	Mittel bis Hoch

Tabelle 2.1: Vergleich der NFC-Tag Typen [O8]

Tag-UIDs

NFC-Tags besitzen je nach Tag-Typ vom Werk aus verschiedene Arten von *Unique Identification Numbers (UIDs)*. Dabei handelt es sich nicht um eine Seriennummer, sondern um einen eindeutigen Identifikator. UID ist ein allgemeiner Ausdruck der im ISO/IEC 14443-3 Standard verwendet wird. Sie sind in verschiedenen *Sizes* verfügbar und es existieren auch UIDs die nicht eindeutig vergeben werden, sondern etwa durch Wiederverwendung öfter im Umlauf gebracht werden können.

Single Size UIDs sind einzigartig, 4 Byte groß und beinhalten keinen Herstellercode. Während diese Art von UID durch die 4 Byte-Size und mit einer maximalen Anzahl von etwa 3.7 Millionen bald an ein Ende gelangt, will die *Double Size UID* (7 Byte) diese Limitierung durch Verwendung eines Herstellercodes mit einer Gesamtanzahl von $2.8 * 10^{14}$ für jeden Hersteller ausgleichen. *Triple Size UIDs* codieren ebenfalls einen Herstellercode und sind 10 Byte groß, werden aber aktuell von keiner PICC implementiert. Nichtsdestotrotz wird laut Norm verlangt, dass jeder PCD *Tripe Size UIDs* unterstützt.

UIDs werden primär für Anti-Kollisions- und Tag-Auswahl Prozeduren verwendet [O27]. Für eine vollständige Liste von UIDs und allen Unterarten sei auf die jeweiligen Herstellerdokumentationen des zu verwendenden Tags verwiesen .

NFC Data Exchange Format (NDEF)

Im Abschnitt 2.1 wurde bereits das NDEF-Format erwähnt. Es handelt sich dabei um ein binäres Datenformat, welches unabhängig von der zugrundeliegenden Geräte- und Tag-Technologie ist

und eine der wesentlichen Erweiterungen von NFC gegenüber RFID darstellt. Es wird dazu verwendet die zu transferierenden Applikationsdaten zu kapseln und als Ganzes zu übertragen. Für alle vier Tag-Typen (2.1) stellt das NFC-Forum Spezifikationen bereit um zu beschreiben wie *NDEF-Messages* gespeichert werden können. Applikationen, welche das NDEF-Format zum Speichern der Daten verwenden, können somit alle vier Tag-Typen einfach und austauschbar einsetzen [O31].

Das Format enthält *Messages*, welche wiederum einen oder mehrere *Records* enthält. Jeder *Record* ist aus einem *header* und einer *payload* zusammengesetzt. Letztere enthält den tatsächlich relevanten Dateninhalt, welcher übermittelt werden soll. Abb. 2.2 veranschaulicht die grobe Struktur einer *NDEF-Message* mit drei *Records* etwa für einen Kontakteintrag (Name, Telefonnummer, Adresse) [43].

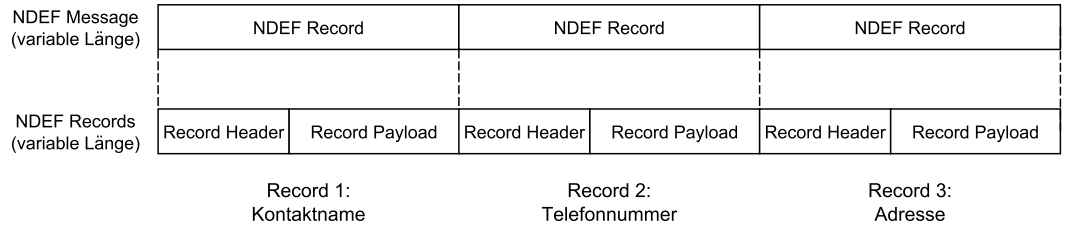


Abbildung 2.2: Grobe Struktur einer *NDEF-Message* nach [43]

NFC Transaktionen werden in der Regel kurz gehalten und jeder Tag beinhaltet generell eine *Message*, welche transferiert werden kann. Es kann sich bei der Payload natürlich um verschiedene Datentypen handeln, und NFC-Endgeräte sollten wissen, wie sie diese Typen handhaben sollen. Um diesen korrekt interpretieren zu können, muss der Header verschiedene Metadaten beinhalten. Diese Daten sind u.a. ein *Type Name Format*, eine Angabe des Payload-Typs und eine eindeutige (optionale) ID.

Type Name Format & Payload-Typ

Das *Type Name Format (TNF)* bestimmt, wie der Payload-Typ interpretiert werden soll. Letzterer ist ein NFC-spezifischer Typ, MIME-Mediatyp, ein URI oder ein externer Typ, welcher angibt, wie die Payload selbst zu interpretieren ist [43]. Während das TNF das Format des Payload-Typen beschreibt, soll der Payload-Typ (oder auch *Record-Type* genannt) den Dateninhalt der Payload selbst etwas spezifischer beschreiben. Abb. 2.3 veranschaulicht den detaillierten Aufbau einer *NDEF-Message* mit Visualisierung der Header-Struktur.

Es gibt sieben vordefinierte TNF, welche vom NFC-Forum standardisiert wurden, darunter einige *well-known*-Typen, welche in der NDEF *Record Type Definition (RTD)* beschrieben sind. Diese TNF werden in Tabelle 2.2 zusammengefasst.

Tabelle 2.3 veranschaulicht einige Beispiele für *NDEF-Records* mit TNF und Payload-Typangaben.

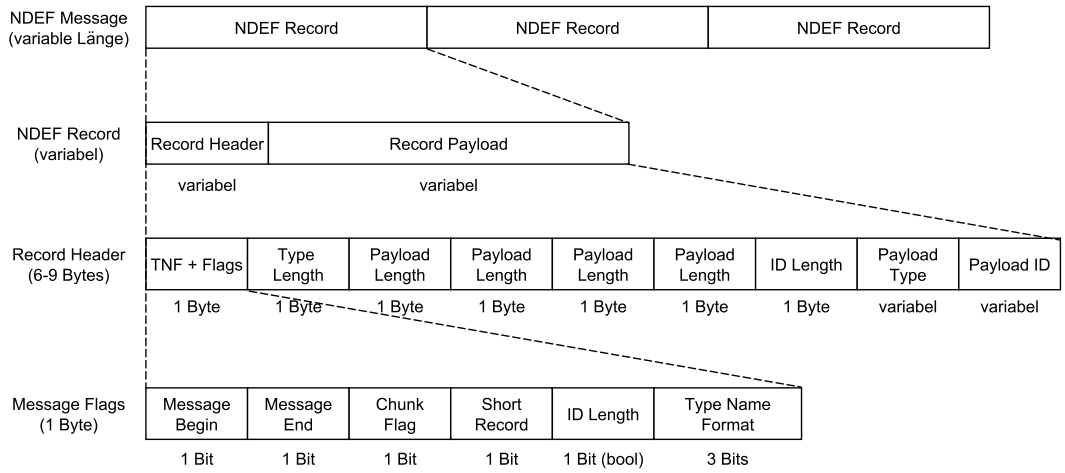


Abbildung 2.3: Detaillierte *NDEF-Message* Struktur mit Beschreibung der Header-Metadaten [43]

Type Name Format (TNF)	**Beschreibung**	**Wert**
Empty	Leerer Record ohne Payload-Typ	0x00
Well-Known	Vordefinierte Typen nach NFC-Forum RTD	0x01
MIME media-type	MIME-Type nach RFC 2046	0x02
Absolute URI	URI nach RFC 3986	0x03
External	Externer benutzerdefinierter Typ nach NFC-Forum RTD	0x04
Unknown	Unbekannter Typ mit Typ-Länge 0	0x05
Unchanged	Für mittlere und letzte gestückelte Payloads mit Typ-Länge 0.	0x06
Reserved	Durch das NFC-Forum für zukünftige Nutzung reserviert	0x07

Tabelle 2.2: Sieben TNF zum Interpretieren des Payload-Typ

TNF	**Payload-Type**	**Beschreibung**
0x01	T	Einfache Text-Message
	U	URI-Message
	Sp	Smart Poster
0x02	text/html	HTML Text
	text/json	JSON String
	image/gif	GIF-Image
0x03	http://nfc-forum.org/	URI zur NFC-Forum Webpräsenz
0x04	android.com:pkg	Definition des Package-Namen einer Android Anwendung

Tabelle 2.3: Beispiele für Payload-Typen

Die zuvor vermittelte kurze Einführung zum NDEF-Format soll dem Leser nur einen Über-

blick über das standardisierte Format geben. Für weiterführende und vollständige Spezifikationen soll auf die Dokumente des NFC-Forums verwiesen werden, welche nähere und vollständige Details enthalten. Diese Dokumente können nur mit einem gültigen Benutzerkonto bezogen werden [O12]. Im Kapitel 4 soll das NDEF-Format zur Speicherung der Daten auf den NFC-Tag zum Einsatz kommen. Genauere, und möglicherweise hier noch etwas unklare Einzelheiten sollen im besagten Kapitel beseitigt werden.

Smartphones

Eine genaue Definition was ein Smartphone ist gibt es derzeit nicht. Es ist eine Erweiterung des klassischen Mobiltelefons, welche die gewohnten Kommunikationsmöglichkeiten wie das Telefonieren oder das Versenden von Kurzmitteilungen durch multimediale Funktionalitäten und dafür abgestimmte Softwarekomponenten erweitert. Smartphones sind weiter im Vormarsch. Wurden im Jahr 2010 weltweit rund 300 Millionen ausgeliefert, waren es im Jahr 2011 schon mehr als 490 Millionen. In den ersten drei Quartalen 2012 belief sich der Smartphone-Absatz bereits auf fast 490 Millionen Geräte. Alleine die beiden Marktführer Samsung und Apple konnten in diesem Zeitraum knapp 240 Millionen Smartphones absetzen. Bis zum Jahresende 2012 summierte sich der weltweite Smartphone-Absatz auf mehr als 720 Millionen Einheiten. Laut einer Prognose der *International Data Corporation (IDC)* soll sich das Wachstum auch in den kommenden Jahren fortsetzen [O33].

Integration von NFC

Die Integration von NFC setzt sich als zusätzliches Feature in immer mehr Smartphones durch und die Liste an NFC fähigen Mobiltelefonen wächst rasant. So befinden sich derzeit bereits mehr als 200 Millionen Geräte im Einsatz [O17]. NFC wurde bereits in einige Business-Szenarios integriert und findet Einsatz im kontaktlosen Zahlungsverkehr (Micropayment) [L17] oder auch in Ticketsystemen für Nah- und Fernverkehr [L14]. Die NFC-Technologie für Smartphones bietet den großen Vorteil, dass mit Hilfe des Gerätes nicht nur Informationen ausgelesen, auf dem Display dargestellt und anschließend weiterverarbeitet werden können, sondern auch das Gerät selbst kann als Datenspeicher für digitale Informationen wie virtuelle Tickets oder Kreditkarteninformationen verwendet werden.

Mobile Betriebssysteme mit NFC Unterstützung

Laut einer Studie [O37] sollen im Jahr 2014 weltweit 416 Millionen Geräte mit NFC-Technik ausgeliefert werden, welche zu 93 Prozent auf das *Android*-Betriebssystem setzen. Bis 2018 soll das Wachstum um 375 Prozent auf 1,2 Milliarden Smartphones mit NFC steigen. *Android* dürfte dann nur noch 75 Prozent ausmachen, da auch andere Hersteller verstärkt NFC integrieren werden. Immerhin wären dies noch 844 Millionen verkaufte *Android*-Geräte mit NFC.

Der wichtigste offene Punkt der Studie ist, ob Apple in seine iPhones NFC in naher Zukunft integrieren wird. Dies könnte die Akzeptanz der Technik deutlich beschleunigen. Während Android-Hersteller eher bereit sind neue Technik zu verbauen, zögert Apple bislang noch als einziger großer Smartphone-Hersteller, der keines seiner Geräte mit NFC-Funktionalität aus-

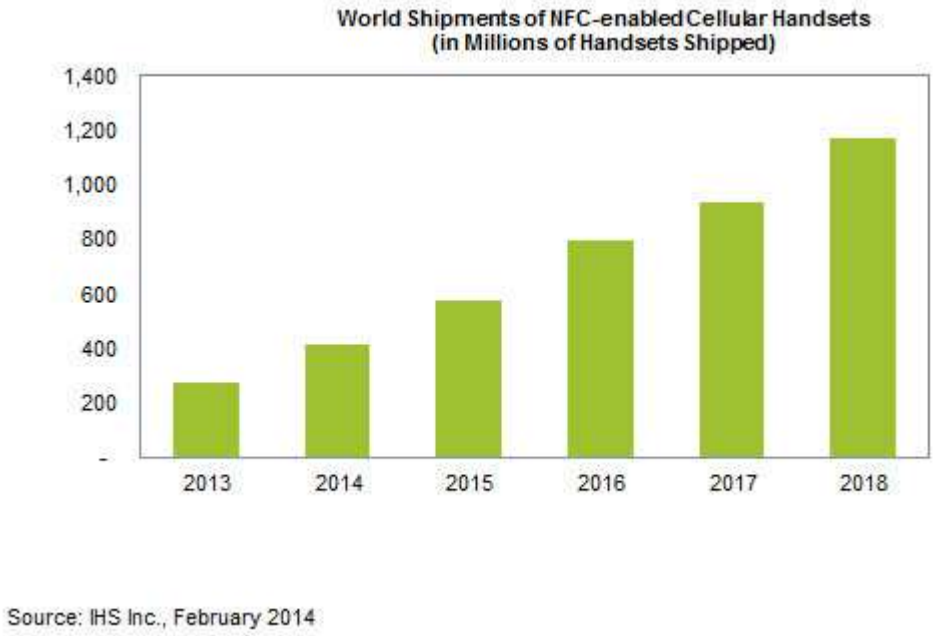


Abbildung 2.4: NFC-Smartphone-Prognose bis 2018

gerüstet hat. Der im Kapitel 4 zu implementierende Prototyp legt somit seinen Schwerpunkt auf die Inbetriebnahme auf einem *Android*-Betriebssystemen.

2.2 Reinigung

In nachfolgendem Abschnitt wird die Bedeutung der Reinigung im Zusammenhang mit Facility-Management beschrieben. Um in weiterer Folge den Reinigungsprozess aus Sicht der verschiedenen Rollen aufzeigen zu können, müssen zuerst einige Begriffsklärungen folgen.

Reinigung im Kontext des Facility-Managements

In aktueller Literatur herrscht unter dem Begriff „Facility-Management“ keine Einigung. Reinigungsfirmen vermarkten ihre Leistungen zunehmend unter dem Begriff *FM*. In den USA wurde in den 1970er Jahren erstmals die Koordination von Mitarbeitern, Prozessen und Räumen als FM bezeichnet. Es ist eine Schnittstelle zwischen dem, was Leute tun und wo sie es tun [41].
Die Österreichische Norm [NS4] definiert Facility-Management wie folgt:

> *„Facility-Management ist ganzheitliches Management der Immobilien und Infrastruktur einer Organisation mit dem Ziel der Verbesserung der Produktivität des Kerngeschäfts.“*

Die Kernaussage ist eine effiziente Gebäudebewirtschaftung und somit Unterstützung der Kern- und Wertschöpfungsprozesse der Nutzerinnen und Nutzer. Die Reinigung stellt in der Regel für den FM-eister den Kernprozess dar, und für den FM-Kunden ist die Reinigung ein Unterstützungsprozess [42]. In der Europäischen Norm für Facility-Management [NS2] wird Reinigung als solche definiert:

> *„Der Bedarf des Auftraggebers nach Hygiene und Sauberkeit wird durch Dienstleistungen erfüllt, die zu einem einwandfreien Zustand der Arbeitsumgebung führen sowie dazu beitragen, die Vermögenswerte in einem guten Zustand zu halten. Beispiele für diesen Bedarf entsprechende Dienstleistungen sind: Hygienedienstleistungen, Unterhaltsreinigung, Maschinenreinigung, Baureinigung und Glasreinigung, Bereitstellung und Instandhaltung von Reinigungsgeräten, Außenanlagenreinigung und Winterdienste, ...“,*

Selbige Norm unterscheidet auch die Rollen im Dienstleistungsprozess:

Kunde kauft die Reinigungsdienstleistungen, welche beim Auftraggeber gedeckt werden müssen

Auftraggeber ist jene Partei, welche die FM-Dienstleistungen schlussendlich empfängt

Auftragnehmer trägt die Verantwortung für die Leistungserbringung

Leistungserbringer führt direkte Dienstleistungen aus

Endnutzer sind jene Personen, welche die Dienstleistungen empfangen, also z.B. Mitarbeiter, Lieferanten, Besucher, ...

Auftragnehmer und Leistungserbringer können hierbei entweder Einheiten derselbigen Organisation der Auftragsgeberseite sein, oder aber auch als jeweils externe Dienstleister agieren. Auf Leistungserbringer-Seite findet auch zunehmend der Ansatz eines Subunternehmens an Beliebtheit, da es sich hierbei um eine rechtlich selbstständige Erfüllung des Vertrages handelt.

Reinigungsorganisation

Leistungsbeschreibung

Um die Anforderungen an Reinigungsprozesse geeignet organisieren zu können wird zunächst eine Leistungsbeschreibung benötigt. Sie beinhaltet sämtliche zu erledigenden Reinigungsarbeiten und Zusatzleistungen je Raumgruppe oder Raum und bildet die Basis für die Arbeitsteilung bzw. Schnittstellengestaltung, das Leistungsverzeichnis, die Pflegepläne und Beschaffung der Betriebsmittel [42].

Leistungsbeschreibungen können auf zwei verschiedene Arten implementiert werden. Die raumgruppenorientierte Leistungsbeschreibung definiert Tätigkeiten entsprechend einer Gruppe von Räumen, die objektorientierte Variante hingegen ordnet die Tätigkeiten den entsprechenden Objekten zu.

Leistungsverzeichnis

Das Leistungsverzeichnis (LV) beinhaltet den Reinigungsbedarf und ist Bestandteil der Leistungsbeschreibung. Es ist Voraussetzung für die Kooperation zwischen Auftraggeber und Leistungserbringer, denn es werden die Kundenbedürfnisse und die Anforderungen aus den Hauptaktivitäten einer Organisation in Reinigungsaktivitäten übersetzt. Das Ziel eines jeden LV ist es, Minimalstandards für die Reinigungsleistung zu definieren und sie sind unverzichtbar für die Steuerung des Dienstleistungsprozesses. Ein umfangreiches Leistungsverzeichnis enthält lt. [44] folgende Informationen:

- Objektbeschreibung
  - Art und Lage des Objektes
  - Raum- u. Flächenverzeichnis
  - Raumgruppen bzw. Nutzungsarten der Räume
  - Bodenbeläge, Raumausstattung
  - Fensterfläche und Fensterart
- Leistungsumfang
  - Anzahl der Räume und Gesamtfläche der jeweiligen Raumgruppe
  - Art und Anzahl des zu reinigenden Inventars
  - Reinigungsturnus, -frequenz
- Leistungsart
  - Reinigungsart: Unterhalts-, Glas-, Grundreinigung
  - Reinigungsmethoden
- Leistungsrahmen
  - Zeitspanne der Reinigung
  - Anforderungen an Personal und Ausrüstung
  - Kosten

In der Praxis sind Leistungsverzeichnisse jedoch nicht so umfangreich und enthalten lediglich eine Leistungsbeschreibung, Frequenz oder das Ergebnis je Raumgruppe oder Raum. Auch bei Leistungsverzeichnissen gibt es verschiedene Varianten [42]. Neben der Leistungsbeschreibung der Qualitäten, das jedes LV zwingend beinhalten muss, werden im tätigkeitsorientierten (inputorientiert) LV zusätzlich auch Quantitäten angeführt. Die tatsächlich vor Ort vorkommenden Verunreinigungen werden nicht berücksichtigt, sondern es zählen nur die Regelmäßigkeit sowie die Methodik der Ausführung. Es handelt sich um eine Arbeit nach Plan, auch wenn theoretisch keine Verschmutzungen erkenntlich sind, muss gereinigt werden. Sollten nicht planmäßige Reinigungsarbeiten notwendig sein, muss für diese ein separater Auftrag erteilt werden, welcher auch gesondert verrechnet wird. Qualitätskontrollen stellen sich als schwierig heraus, müsste eine Person die Prozesse vor Ort beobachten.

Das ergebnisorientierte (outputorientiert) LV will diese Mängel beheben und verzichtet auf die Angabe von Frequenzen. Es beschreibt stattdessen wie das gewünschte Ergebnis während einer bestimmten Zeitspanne auszusehen hat und fordert nicht die Anwesenheit einer Reinigungskraft zu einem bestimmten Zeitpunkt. Die Wahl der Reinigungsmethoden und die Bestimmung der Frequenzen liegt bei den Verantwortlichen für die Reinigung einerseits auf Managementebene und andererseits auf der ausführenden Ebene. Da es schwierig ist einen gleichbleibenden Reinigungsstandard zu halten, wird, um die Werterhaltung eines Gebäudes aufrecht zu erhalten, auf Eigenverantwortung in regelmäßigen Abständen (ca. einmal wöchentlich) eine vollflächige Reinigung durchgeführt. Als eine Unterart der ergebnisorientierten Variante gibt es außerdem die Reinigung auf Bestellung durch den Raumnutzer selbst oder einen intelligenten Raum, welcher mit einem *Wireless Sensor Network (WSN)* verbunden ist [L8]. Qualitätssicherung ist für ein Gelingen der ergebnisorientierten Leistungsvereinbarung unverzichtbar, damit der Auftraggeber Vertrauen in die Arbeit der Reinigungsdienstleister entwickeln kann. [42] definiert auch Parameter, welche die beiden verschiedenen Arten von Leistungsverzeichnissen unterscheiden:

- Tätigkeitsorientiertes Leistungsverzeichnis
  - Beschreibung der zu reinigenden Raumgruppen (bzw. Flächen und Räume oder Objekte)
  - Beschreibung der Reinigungsaktivitäten und -methoden
  - Reinigungsfrequenzen und ev. Zeitpunkt der Reinigung
- Ergebnisorientiertes Leistungsverzeichnis
  - Rahmenbedingungen und Ziele
  - Beschreibung der zu reinigenden Flächen und Räume
  - Beschreibung des Ergebnisses bzw. des gewünschten Zustands
  - Messbare Indikatoren und Mindestleistungsanforderungen, Messmethoden und Nachweise
  - Nennung der Zeitspanne bzw. des Zeitpunktes, wann das Ergebnis erwartet wird

Qualitätssicherung

Betrachtet man bei der ergebnisorientierten Reinigung die Tatsache, dass Qualitätssicherungen notwendig sind, damit diese Leistungsvereinbarungen gelingen, muss sie zwingend an ein Qualitätsmesssystem (QMS) gekoppelt werden [42]. Ein QMS in der Gebäudereinigung wird nach DIN EN 13549 [NS5] wie folgt definiert:

> *„Ein Qualitätsmesssystem ist ein Prüfsystem, das überwacht, ob die Dienstleistungen, die zwischen zwei Parteien definiert und vereinbart wurden, das Qualitätsniveau erfüllen, das nach Anforderungen, die implizit oder explizit in der Vereinbarung ausgedrückt werden, vorgesehen ist.“*

Ein Prozess muss ausgehend vom Mikro- bis zum Makroniveau mit Hilfe von Regelkreisen überwacht werden. Durch das nach DIN EN 13 549 vorgegebene Prüfsystem werden sowohl ein ausgeglichener Reinigungsprozess als auch eine weniger mit Fehlern behaftete (Nicht-konformitäten) Ausführung gewährleistet. Es handelt sich dabei kurz gesagt um die Lenkung und Verbesserung des Reinigungsprozesses. Mit dem System wird überprüft, ob das Ergebnis im „absoluten“ Sinne der festgelegten Untergrenze entspricht, und ermöglicht ebenfalls Erkenntnisse zur erreichten Bandbreite, innerhalb derer sich die erbrachten Reinigungsdienstleistungen bewegen [O25]. Hierfür hat das Europäische Normungsinstitut CEN [3] auch Anforderungen an ein Qualitätsmesssystem entwickelt und publiziert [NS5].

- *Das Qualitätsmesssystem soll leicht zu handhaben und zu verstehen sein.*
- *Die Kosten der Prüfungen sollen niedrig sein.*
- *Das System soll auf alle Arten von Gebäuden und Objekten anzuwenden sein.*
- *Sowohl Arbeitnehmer als auch Arbeitgeber und Kunden sollen in der Lage sein, ohne spezielles Training Qualitätsprüfungen auszuführen.*
- *Das System soll objektiv sein, oder mehrere Level von Sauberkeit beschreiben.*
- *Qualitätsprüfungen sollen anhand einer bestimmten Anzahl objektiver Kriterien vorgenommen werden, die visuell erfasst werden und ggf. durch objektive Prüfmethoden ergänzt werden können.*
- *Das System und die Berichterstattung sollen leicht zu dokumentieren sein.*
- *Das System soll eine gewisse Anzahl an Nicht-Konformitäten (Fehler) zulassen.*

Auf Basis dieser Anforderungen und der vorliegenden Normierung wird im nächsten Abschnitt der Geschäftsprozess näher beschrieben.

Analyse der Geschäftsprozesse aus Sicht der Rollen

Ablauf Auftragnehmer

Eine Führungskraft der Auftragsnehmerseite (Objektleiter) hat die Verantwortung für die Durchführung der Qualitätssicherung zur Feststellung, ob das vereinbarte Reinigungsergebnis und somit das *Accepted Quality Level (AQL)* [4] erreicht wurde. Anhand der Gesamtzahl der zu reinigenden Räume wird eine Stichprobe und ein Startraum ausgewählt, in der die Qualitätsprüfung durchgeführt wird. In Abhängigkeit vom AQL und der Gesamtgröße des Objektes kann aus entsprechenden statistischen Tabellen die notwendige Anzahl zu prüfender Räume bestimmt werden. Die Stichprobe wird für gewöhnlich zufällig gewählt, beispielsweise durch ein automatisiertes System auf Basis der ISO 2859 Norm. Auf diese Weise ist auch gewährleistet, dass die Auswahl der Räume zufallsgesteuert erfolgt und von subjektiven Einflüssen unverfälscht bleibt [O14].

Vor Durchführung der Qualitätsprüfung, welche auch im Beisein des Auftraggebers abgewickelt werden kann, müssen die Prüfformulare zur Erfassung der Ergebnisse in entsprechender

[3] http://www.cen.eu

[4] Qualitätslage, der eine bestimmte Annahmewahrscheinlichkeit zugeordnet ist (NS1)

Anzahl vorbereitet werden. Hierfür ist von Nöten, dass das Profil für das vertraglich festgelegte Qualitätsniveau des Prüfungsobjektes vorliegt. Diese Informationen werden für gewöhnlich aus der Leistungsbeschreibung (bzw. dessen Anhang) bezogen. Tabelle 2.4 veranschaulicht, wie ein solches Formular aussehen kann.

Prüfung / Inspektion

Objekt: Gebäudeart:	Raumgröße: Raumart: Raumnr.:		Datum: Uhrzeit: Name:	
Anzahl d. Verunreinigung ▷ Raumkomponentengruppen ▽	**Abfall**	**Nicht haftende Verschmutzungen**	**Haftende Verschmutzungen**	**Reinigungsassoziierte Dienstleistungen**
Hauptnutzungskomponenten				
Restliches Inventar				
Wände/Decken				
Boden				
Schwer einsehbare Bereiche				
Bemerkungen				

Tabelle 2.4: Prüfformular zur Qualitätssicherung [O14]

Außerdem muss im Vorfeld auch ermittelt werden, ob aus vorhergehenden Prüfungen Aufträge für Nachprüfungen aufgrund von Nacharbeit durchzuführen sind, welche zeitlich mit der aktuellen Prüfung zu koordinieren sind. Falls dies erwünscht wird, muss festgelegt werden, wie mit den relevanten Räumen vorgegangen werden soll [O14].

Zur Durchführung der Prüfung werden auf den Formularen die prüfungs- sowie die raumbezogenen Angaben eingetragen. Bei Betreten des Raums ist – insbesondere bei geringem Tageslicht – zunächst darauf zu achten, dass alle vorhandenen Beleuchtungsmöglichkeiten eingeschaltet sind. Da die Prüfungen unter dem Aspekt der Nutzerzufriedenheit erfolgen sollen, wird bei der Begutachtung nach Möglichkeit eine Prüfposition eingenommen, die der Blickrichtung des Raumnutzers entspricht.

Daraufhin erfolgt eine kurze Orientierung bezüglich der vorhandenen Einrichtungsgegenstände und der Raumeinteilung. Der Prüfer ordnet alle Bereiche den Raumkomponenten zu (Tabelle 2.4). Die Prüfung erfolgt anhand einer visuellen Kontrolle aller Räume der Stichprobe. In jedem Raum werden fünf Oberflächengruppen in folgender Reihenfolge geprüft:

Hauptnutzungskomponenten Inventarbestandteile, die vorwiegend und regelmäßig, z. B. als Arbeitsplatz, genutzt werden (z. B. Schreibtisch, -Lampe, Bestuhlung, Telefon, Papierkorb)

Restliches Inventar Alle übrigen Inventarbestandteile (z. B. Sideboards, Garderobenständer, Schränke, Bücherregale)

Wände/Decke Decken, Wände und fest mit ihnen verbundene Gegenstände (z. B. Heizkörper, Fensterbank, Bilder, Türen, Spiegel)

Boden Bodenbeläge, Sockelleisten, Türschwellen, Schmutzfangmatten

Schwer einsehbare Bereiche Flächen, die bei üblicher Nutzung eines Raumes nicht direkt einsehbar sind (z. B. Fliesen hinter Heizkörpern, Boden unter niedrigen Schränken)

Die Vorgabe der Reihenfolge soll zu einer gewissen Systematik der Prüfung verhelfen. Auf dieses Weise wird der Raum „von innen nach außen“ bzw. vom wichtigsten Bestandteil ausgehend untersucht. Die Möglichkeit, Raumteile zu übersehen, ist somit geringer. Anschließend erfolgt eine visuelle Prüfung hinsichtlich der Verschmutzungsarten Abfall, nicht-haftende und haftende Verschmutzungen.

Bei der Bewertung des Reinigungsergebnisses hat die begutachtete Fläche frei von den Verschmutzungsarten zu sein. Ist dieses nicht der Fall, wird jede einzelne festgestellte Verschmutzung gezählt und gibt somit in der Summe die Anzahl der Fehler [5] an. Bei jedem festgestellten Fehler wird dieser in der jeweiligen Kategorie auf dem Prüfformular per Strich bzw. die Anzahl der Fehler unter der jeweiligen Verschmutzungskategorie eingetragen. Für sonstige Feststellungen (z.B. außergewöhnliche Verschmutzungen; Verschmutzungen, die materialbedingt oder nicht entfernbar sind; etc.) können Eintragungen im Bemerkungsfeld auf dem Prüfformular vorgenommen werden. Insbesondere sollen in diesem Feld auch methodische Fehler festgehalten werden. Dies dient der späteren Schulung des Reinigungspersonal um methodische Fehler wie Reinigungsmittelrückstände oder Wischspuren in Zukunft zu vermeiden [O14].

Für die anschließende Auswertung kommen nun die eingangs erwähnten Qualitätsniveaus (QN) zum Einsatz. Hierfür ist ein Auswertungsformular notwendig um Abweichungen vom Soll-Zustand zu ermitteln und ggf. Nacharbeiten anzufordern. Zunächst muss die Anzahl der Fehler je Raumkomponentengruppe eingetragen werden, welche in weiterer Folge summiert werden. Entsprechend einer Tabelle zur Ermittlung des IST-QN wird dieser ebenfalls vermerkt. Wurden alle diese Werte ermittelt, kann der IST-QN dem SOLL-QN gegenübergestellt werden um schließlich die Abweichung zu erhalten. Räume, welche eine negative Abweichung aufweisen, sind in der Prüfung durchgefallen. Tabelle 2.5 zeigt eine Raumauswertung, welche negativ ausgefallen ist und dementsprechende Nacharbeit oder anderwertige in der Leistungsbeschreibung vereinbarte Sanktionen abverlangt.

In Abhängigkeit von der Stichprobengröße darf eine bestimmte Anzahl von durchgefallenen Räumen nicht überschritten werden, dabei werden durchgefallene Räume nicht gegen „zu gut“ gereinigte Räume aufgerechnet [O14]. Sind alle Räume geprüft, werden jene, die das Reinigungsergebnis erreicht haben, denen die es nicht erreicht haben, gegenübergestellt. Mithilfe des vereinbarten AQL-Wertes (annehmbare Qualitätsgrenze) kann ermittelt werden, ob das Reinigungsziel als Gesamtes erreicht wurde. Sollte dies nicht der Fall sein, muss der Objektleiter den verantwortlichen Leistungserbringer dies auf geeignete Weise kommunizieren und ggf. Nacharbeit vereinbaren. Abb. 2.5 illustriert den Gesamtablauf der Qualitätssicherung.

[5] Die Fehlerbewertung wird für gewöhnlich detailliert in der Leistungsbeschreibung festgehalten. Beispielsweise kann ein Bereich von angefangenen 1 m x 1 m bei allen Verschmutzungsarten als jeweils ein Fehler gewertet werden. Bei nicht in Quadratmetern zu messenden Bereichen kann wie folgt bewertet werden: Bei zusammenhängenden Bereichen, wie Fußleisten, wird eine Strecke von jeweils 5 m als ein Fehler bewertet [NS5].

Anzahl d. Verunreinigungen ▷ Raumkomponentengruppen ▽	Abfall	Nicht haftende Verschmutzungen	Haftende Verschmutzungen	Summe	QN	Soll-QN	Abweichung
Hauptnutzungskomponenten	1	0	1	2	2	4	-2
Restl. Inventar	0	0	1	1	4	4	±0
Wände/Decke	0	0	0	0	5	4	+1
Boden	0	0	1	1	4	4	±0
Schwer einsehbare Bereiche	0	1	0	1	4	4	±0
Summe d. QN-Abweichungen:	Positiv: +1, Negativ: -2						

Tabelle 2.5: Beispiel einer negativen Raumauswertung

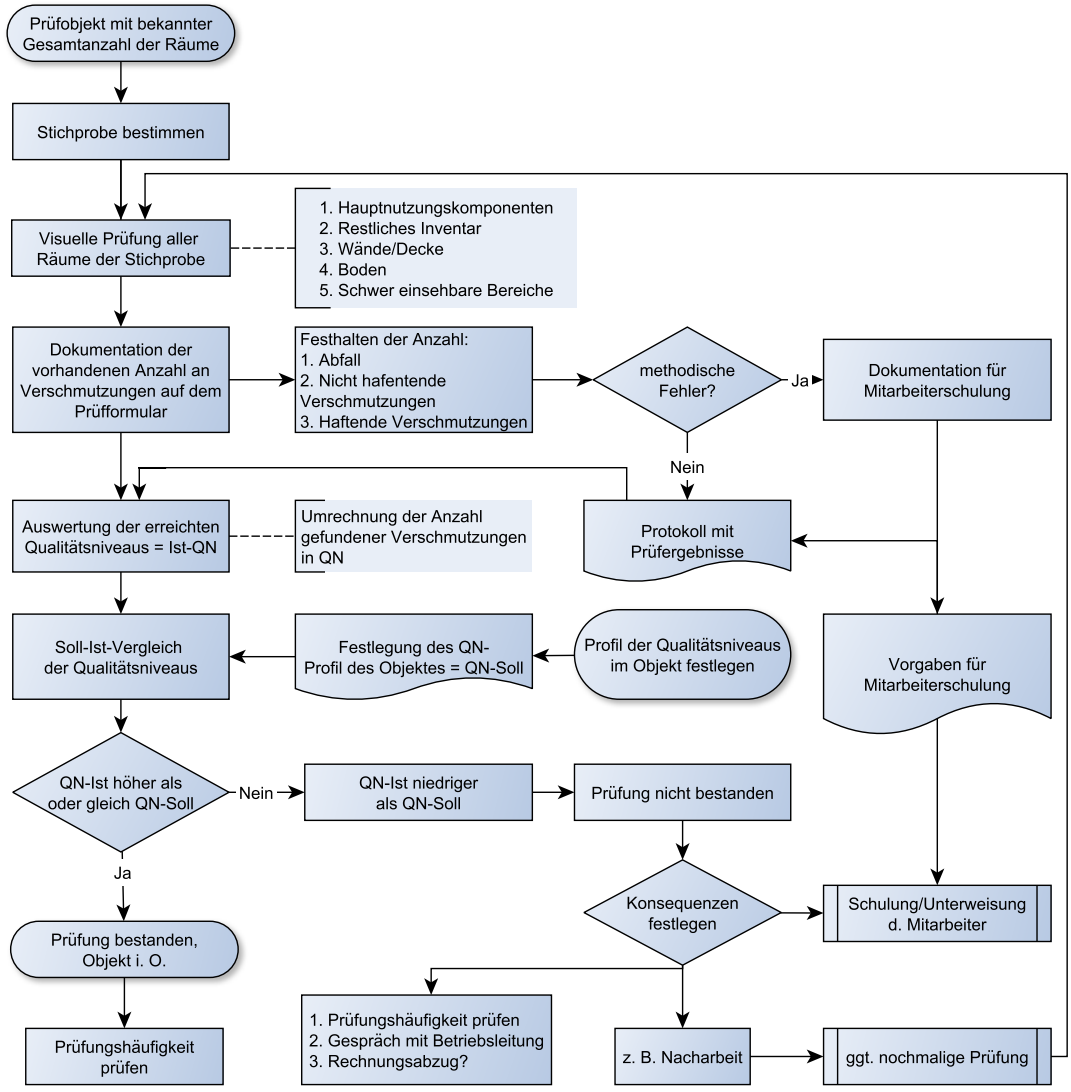


Abbildung 2.5: Ablauf der Qualitätssicherung [O14]

Ablauf Leistungserbringer

Die Reinigungsdienstleister werden versuchen, das Endergebnis der Reinigung mit der geplanten Qualität in Einklang zu bringen. Dafür müssen sie das geforderte Endergebnis kennen. Abhängig von der Organisationsform im Objekt bekommt die Reinigungskraft die Weisungen vom Objektleiter (zentral) oder von verschiedenen Einheiten und Abteilungen (dezentral).

Qualitätsniveau muss jeweils an folgenden Zeitpunkten erreicht sein: **Mo - Fr 08:00 Uhr** Vereinbartes Qualitätsniveau (bezogen auf die Raumkomponenten): **QN 3**	
Abweichende Ergänzungen	
Decken: Boden:	QN 0 Bei Teppichboden entfällt das Entfernen von „haftenden Verschmutzungen“
Achtung: In diesen Räumen können sowohl interne als auch externe Veranstaltungen stattfinden. Die Reinigung ist an die aktuellen Veranstaltungen anzupassen.	
Reinigungsassoziierte Dienstleistungen, die jeweils zum Prüfungszeitpunkt erreicht sein müssen	
1 2 3 4	Papier- und Abfallbehälter sind geleert und mit einem Beutel versehen. Entfernen von Spinnweben und Staubfäden von den Decken Fenster und Türen verschlossen, Licht ausgeschaltet Alle Flaschen, Gläser und sonstiges Geschirr ist zur Abholung durch das Cateringunternehmen auf einem Tisch bereit zu stellen.
Raumkomponenten	Inventar
Hauptnutzungs-komponente (HK) Restliches Inventar (RI) Wände / Decken (W/D) Boden (B) Schwer einsehbarer Bereich (SEB)	Tische, Stühle, Rednerpult, Touch Panel, Flipcharts, Moderationsgegenstände inkl. dazugehörigen Tischen und Ablagen Sideboards, Stehlampen, Garderobenständer und alle fest verbundenen Gegenstände, wie z.B. Heizkörper, Fensterbänke, Steckdosen, Schalter, Türen, Türrahmen Bodenbeläge, Schmutzfangmatten, Sockelleisten, Türschwellen Flächen, die nicht direkt einsehbar sind. Boden unter niedrigen Schränken, Bilderschienen, Schrankflächen über 2,15 cm
Ausstattung:	Türflächen lackiert oder beschichtet: 61 qm, Radialvektoren: 23 St.
Anmerkung: Die Aufzählung des Inventars ist nicht abschließend, sondern soll lediglich die Schwerpunkte (insbesondere bezogen auf die Hauptnutzungskomponenten) darstellen.	

Tabelle 2.6: Beispielauszug LV der Seminarraumgruppe
Quelle: http://www.caesar.de

Bei Arbeitsbeginn wird dem Leistungserbringer das Leistungsverzeichnis (siehe Teilauszug Tabelle 2.6), ein Raumbelegungsplan sowie ein Reinigungsplan ausgehändigt. Außerdem wird bekanntgegeben, ob im Zuge eines Qualitätssicherungsprozesses der Mitarbeiter Nacharbeiten leisten muss. Sollte dies der Fall sein, wird hierfür der Reinigungsplan in einer entsprechend erweiterten Form übergeben.

Der korrekte und vollständige Bezug der Betriebsmittel, welche verwendet werden sollen,

kann durch eine Checkliste, welche im Lagerraum angebracht ist, kontrolliert werden. Beim Durchführen der Reinigungsdienstleistung ist insbesondere darauf zu achten, dass ein Reinigungsprotokoll geführt wird. Auf einem solchen Dokument ist zumindest das Datum, die Zeitspanne der Dienstleistung je Raum, der Bereich (z.B. in Form einer Raumnummer) und eine Unterschrift des Mitarbeiters notwendig. Falls notwendig, kann auf zusätzlichen Bemerkungsfeldern Besonderheiten wie Defekte oder andere Auffälligkeiten wie Mängel oder Fehlen von Inventar vermerkt werden. Hierfür wird eine andere Instanz ggf. Reparaturaufträge oder Ersatzanforderung beantragen.

Im ausgehändigten Leistungsverzeichnis wird das vereinbarte Qualitätsniveau je Raumgruppe angeführt. Außerdem ist für die Reinigungskraft genau ersichtlich, zu welchen Zeitpunkten dieses erreicht sein sollte. Abweichende Ergänzungen informieren den Mitarbeiter beispielsweise über spezielle Fehlerbehandlungen oder über besondere Gegebenheiten wie Verweise auf Veranstaltungen oder unregelmäßige Raumbelegungen. Bei individuell festgelegten Bewertungskriterien für reinigungsassoziierte Dienstleistungen werden diese separat angeführt. Die Aufzählung des Inventars, das den verschiedenen Raumkomponenten zugeordnet wird, ist nicht abschließend, sondern soll lediglich die Schwerpunkte darstellen.

Reinigungskräfte, welche eigenmotiviert sind sich selbst zu überprüfen, sollten auch wissen, was und wie überprüft wird. Nach Abschluss der notwendigen und geforderten Tätigkeiten je Raum kann der Mitarbeiter eine Selbstkontrolle basierend auf den Qualitätssicherungskonzepten durchführen.

Nach Abschluss aller Tätigkeiten und Rückgabe der Betriebsmittel im Objekt wird das schriftliche Protokoll dem Objektleiter übergeben.

Ablauf Endnutzer

Endnutzer sind jene Personen, welche die Dienstleistungen empfangen. Sie haben die Möglichkeit, unsachgemäß vorliegende Zustände oder Beschädigungen sowie Fehlen von Inventar zu melden. Hierfür liegen in der Regel bei Eingangsbereichen Verweise und Kontaktmöglichkeiten zu den verantwortlichen Facility-Management Einrichtungen vor. Dringende Meldungen können auch telefonisch zum Objektleiter kommuniziert werden, welcher entsprechende Maßnahmen einleiten sollte.

Forschungskontext

Mit dem Fortschritt der Informatik entwickelt sich auch die Software für das Reinigungsmanagement weiter. Vernetzung, Einsatzbreite und Mobilität der Anwendung spielen eine immer wichtigere Rolle [L6]. Stationäre Systeme sind normalerweise nicht vor Ort direkt im verwalteten Gebäude nutzbar, weil dort in der Regel kein Zugriff auf das CAFM-System besteht. Ein Zugriff über das Internet scheidet meistens aus, da die Kunden, die das Gebäude nutzen, die Verwendung ihrer PCs zu derartigen Zwecken meist nicht gestatten. Um CAFM wirklich mobil im verwalteten Gebäude nutzen zu können, ist daher ein System notwendig, welches den Zugriff zu den Daten über mobile Geräte wie ein Smartphone gestattet [L2].

Mobile Lösungen werden vor allem für Arbeitszeit- und Leistungskontrolle immer gefragter. Die passende Reinigungssoftware *von der Stange* gibt es nicht. Anpassungen an die individuelle

betriebliche Situation und an Kundenwünsche sind unumgänglich. Dabei wird das *Wünschbare* durch die Fortschritte der Informatik und durch die Zunahme der Systemanwendungen immer mehr zum *Machbaren* [42].

Die im nächsten Abschnitt vorgestellten bestehenden Systeme zur Unterstützung des FM-Prozesses aus Sicht der Rollen *Auftragnehmer*, *Leistungserbringer* und *Endnutzer* basieren teilweise auf veralteten Technologien oder nutzen keine oder ungeeignete Auto-ID Verfahren. Außerdem wird oft die Rolle *Endnutzer* vernachlässigt, da diese in deren Systeme aus technischen Gründen nicht einbeziehbar ist. Des weiteren wird der Anwendungsfall *Qualitätssicherung* als eigenständig behandelt und versteckt vom Dienstleister in separate Anwendungen implementiert, sodass dieser nicht weiß, wie bei solch einem Prozess eigentlich vorgegangen wird. Aus Sicht des Leistungserbringers stellt sich somit auch die Eigenkontrolle als schwierige Aufgabe heraus.

2.3 Softwarelösungen für das Reinigungsmanagement

Arten von Softwarelösungen

Reinigungsverantwortliche setzen für die Reinigungsorganisation im Gegensatz zu einer Systemsoftware eine Anwendungssoftware ein. Dabei werden in der Reinigungsbranche vier Typen unterschieden [42]:

- Spezifische Reinigungslösungen, welche speziell für eine Anwendung in der Gebäudereinigung entwickelt wurden
- Module Reinigungs- und Flächenmanagement in CAFM-Sytemen
- Standardsoftwarelösungen wie Tabellenkalkulation oder Datenbank
- Businesssoftware/ERP-Lösungen (*Enterprise Resource Planning*), welche komplexe Lösungen zur Unterstützung der Ressourcenplanung eines gesamten Unternehmens sind

Die im ersten Punkt erwähnten spezifischen Softwarelösungen für das Reinigungsmanagement basieren in der Regel auf im Punkt vier erwähnten Programmen wie Tabellenkalkulationssoftware oder Datenbanksystemen. CAFM-Systeme sind modular aufgebaut und können speziell auf die Kundenbedürfnisse um bestehende Module erweitert werden. Diese Systeme verwenden eine zentrale Datenbank, welche Grundlageninformationen wie Flächenangaben, Bodenbeläge, Inventar, Raumnutzung, etc. enthält und als Informationsbasis für sämtliche Anwendungen zur Verfügung stellen. ERP-Lösungen können speziell auf Gebäudereinigungsunternehmen ausgerichtet sein, müssen es aber nicht. Die wohl bekannteste ERP-Lösung in Österreich ist *SAP* [6] [42].

Laut einer Marktbefragung [40] war schon 2005 absehbar, dass die Nachfrage nach mobilen Lösungen insbesondere für Arbeitszeit- und Leistungskontrolle immer gefragter ist. Software zur Unterstützung der FM-Prozesse aus Sicht der beschriebenen Rollen 2.2 können auf folgender (mobiler) Hardware implementiert werden:

[6] `http://www.sap.com/pc/bp/erp.html`

- tragbare Computer mit PIM[7]-Funktionalität
- Smartphones
- Tablets
- Notebooks
- Proprietäre Hardware

Viele mobile Reinigungsmanagementsysteme verwenden Technologien um Daten auf geeignete Weise zu erfassen. Im Facility-Management haben sich abseits der manuellen Identifikation zwei Auto-ID Technologien durchgesetzt, welche im nächsten Abschnitt näher beschrieben werden.

Mobile Datenerfassung und automatische Identifikation

Durch Mobile Datenerfassung (MDE) mit automatischer Identifizierung der Flächen/Objekte (Auto-ID) können Daten dort erhoben werden wo sie entstehen. Abseits des stationären PC-Arbeitsplatzes, in der hintersten Ecke des Gebäudes oder unterwegs im Außenbereich. Die gewonnenen Daten sind elektronisch sofort verfügbar und können in Echtzeit oder zeitversetzt vom übergeordneten System weiter verarbeitet werden [O19]. Dabei haben sich im FM zwei Auto-ID Technologien durchgesetzt, welche auf dem mobilen System als Leseeinheit bereitgestellt werden muss. Im folgenden Abschnitt werden die beiden Auto-ID Technologien vorgestellt und gleichzeitig auf deren Relevanz für den Einsatz im FM eingegangen.

Bar-/Strichcodes

Barcode-basierte Systeme werden in die Kategorie der kontaktlosen Auto-ID Systeme eingeordnet. Die Global Standards One (GS1) [8] Organisation ist auf die Entwicklung und Standardisierung von Barcodes und den Identifikationssystemen für selbige fokussiert. Barcodes finden beispielsweise in Kaufhäusern durch Kassiere für die Produktidentifikation und in weiterer Folge für die Preiszuordnung ihren Einsatz.

Das Prinzip der maschinellen Lesbarkeit des Barcodes basiert auf der optischen Abtastung des Codes mittels Laserstrahl oder anderen Bilderfassungsmechanismen (z.B. Kamera). Die Basis für die Codeerkennung ist ein zuvor vereinbartes Kodierungsschema. Durch Reflektionen können Unterschiede zwischen hellen und dunklen Stellen des Codes ermittelt und anhand definierter Kodierungsvorschriften erkannt werden. Aktuelle Lesegeräte arbeiten, je nach Hersteller, im Bereich von einigen Zentimetern bis zu 10 Metern [O7].

Durch die Einfachheit der Anwendung von Barcodes ist dieser Technologie eine große Akzeptanz zuzuschreiben. Der kostengünstigen Herstellung durch Aufdrucken und der zuverlässigen Funktionsweise stehen dennoch einige Nachteile gegenüber. Barcodes können Probleme bereiten, wenn sie zerkratzt oder verschmutzt sind, da sie bei der Verwendung für das Lesegerät eine Sichtlinie bereitstellen müssen [L7]. Außerdem ist beim Lesevorgang darauf zu achten, dass ein bestimmter Winkel des zu identifizierenden Objektes zum Lesegerät einzuhalten

[7] Personal Information Manager
[8] `http://www.gs1.org`

ist [O7]. Durch die erwähnte Einfachheit ergeben sich noch weitere Nachteile dieser Technologie. Barcodes können durch Fotografie oder durch Kopie einfach vervielfältigt werden, was den Einsatz in sicherheitskritischen Bereichen nicht fördert. Ein Unkenntlichmachen oder gar ein Entfernen des aufgedruckten (Barcode-)Musters führt dazu, dass eine Identifikation durch das Lesegerät nicht mehr möglich ist. Lesegeräte, welche eine Digitalkamera zur Bilderfassung verwenden, haben außerdem einen hohen Akkuverbrauch und sind durch die Tatsache, dass ein automatischer Fokus verwendet werden muss, oftmals nicht so schnell wie erwartet.

Im Facility-Management findet der Barcode heute bereits rege Anwendung. Als ein Beispiel sei die Inventarisierung von Einrichtungsgegenständen genannt [O7].

Radio Frequency Identification

Radio Frequency Identification (RFID) wird ebenfalls als kontaktlose Auto-ID Technologie bezeichnet, da eine Möglichkeit zur maschinellen und automatischen Identifikation beliebiger realer Objekte bewerkstelligt werden kann. Im Abschnitt 2.1 wurde bereits auf die RFID-Technologie und einige Einsatzszenarien eingegangen. In der Logistik sowie im Einzelhandel wird diese Technologie als Ablösung des Barcodes, welcher im Abschnitt 2.3 beschrieben wurde, betrachtet. Das Besondere gegenüber anderen Auto-ID Systemen ist, dass die Identifikation mittels funkbasierter Kommunikationsmethoden realisiert wird [O7].

Mehr Flexibilität, kürzere Erfassungszeiten und damit verbunden eine erhöhte Effizienz sind hierbei die vorherrschenden Argumente der Industrie. Anders als bei der Verwendung von Barcodes können hierbei nicht nur Artikelgruppen, sondern auch das einzelne Produkt identifiziert werden. Ermöglicht wird dies durch die Verwendung elektronischer Speicher, die während der Identifikation ausgelesen werden [O7].

Im Facility-Management sind grundsätzlich zwei große Einsatzgebiete für RFID Systeme denkbar: Unterstützung der Inventarerfassung und -verwaltung und Wartung und Reparatur von Anlagen und Prozessen.

Evaluierung mobiler Systeme

Kärcher

Die Firma *Alfred Kärcher GmbH* [9] bietet mit seinem *ECO!Cleaning* und *ECO!Manager* ein System für Leistungserbringer und Qualitätssicherung an, das es Mitarbeitern und Objektleitern ermöglicht, Arbeitsschritte und Arbeitszeiten mobil mit Hilfe eines sogenannten *Scanner ECO!* zu dokumentieren. Dieser Laser-Scanner ist ein kleines akkubetriebenes Gerät, mit welchem Barcodes gelesen werden können. Er verfügt über einen Speicher, welcher nach Abschluss der Arbeit dessen hinterlegte Datensätze kabelgebunden in ein (Online-)System übertragen kann. Dies wird für gewöhnlich vom Objektleiter nach der Durchführung der Reinigungsdienste vorgenommen.

Der Scanner kommt bereits bei Arbeitsbeginn zum Einsatz. Der Mitarbeiter scannt hierfür einen entsprechenden Barcode um den Zeitpunkt des Arbeitsbeginns zu speichern. Analog wird beim Beenden der Arbeit ein entsprechender Barcode gelesen. Während der Arbeitszeit scannt

[9] `http://www.kaercher.at`

der Mitarbeiter beim Betreten und beim Verlassen eines Raumes einen Barcode am Türfalz. Zeiten und Leistungen sind somit räumlich zugeordnet. Außerdem befinden sich am Reinigungswagen des Mitarbeiters weitere Barcodes, welche für spezielle Ausnahmefälle wie Beschädigungen oder dergleichen genutzt werden können. Ist ein Raum in einem so desolaten Zustand, dass der Mitarbeiter nicht die geeigneten Reinigungsmittel mitführt, steht ihm ein Funkgerät zur Verfügung, mit welchem er entsprechende Anforderungen zum Objektleiter kommunizieren kann.

Zur Qualitätssicherung durch den Objektleiter verwendet dieser eine App, mit welcher durch die integrierte Digitalkamera der Barcode erfasst werden kann um Daten abzurufen und weiterzuverarbeiten.

Der FM-Prozess wird in diesem Szenario durch eine Vielzahl von eingesetzter Hardware (Scanner, Funkgerät, Smartphone) unterstützt. Unternehmen, welche diese Lösung produktiv einsetzen wollen, müssen mit einer kostenspieligen Anschaffung [10] rechnen. Als Nachteil des von Kärcher gewählten Auto-ID Verfahrens ist außerdem zu erwähnen, dass der Prozess des Arbeitsbeginns (und weitere) nicht zwangsläufig an der vom Objektmanager vorgesehenen Stelle (z.B bei der Betriebsmittelausgabe) stattfinden muss. Die im Abschnitt 2.3 erwähnten Nachteile von Barcode-Identifikationen kann zu Betrugsversuchen führen, indem ein Duplikat vom Mitarbeiter direkt aus einer Fotografie gescannt wird um somit die Dokumentation eines tatsächlich nicht ausgeführten Arbeitsschrittes zu manipulieren.

speedikon FM

Die Firma *speedikon FM* [11] bietet ein auf einem *Personal Digital Assistant (PDA*) betriebenes System zur Unterstützung der Reinigungskontrolle für die Qualitätssicherung an. Im Vorfeld müssen die Informationen über die Flächen und das Leistungsverzeichnis zusammengestellt und anschließend auf einen PDA übertragen werden. Der für die Kontrolle zuständige Mitarbeiter arbeitet bei seiner Begehung die Liste der Flächen ab. Für jede Fläche wird einzeln dokumentiert, welche Tätigkeiten von der Reinigungsfirma ordnungsgemäß durchgeführt wurden und welche zu beanstanden sind.

Wurde die Tour abgearbeitet, werden die Daten in das FM System übertragen und ausgewertet. Es werden Statistiken aus verschiedenen Sichten zur Verfügung gestellt. Auf Basis der Beanstandungen wird dokumentiert, ob hierzu eine Nachbesserung angefordert wurde oder bei der nächsten Rechnung eine Minderung erfolgt. Mit dieser IT-technischen Unterstützung erfolgt die Reinigungskontrolle zielgerichtet und standardisiert. Eine durchgeführte Begehung kann in regelmäßigen Zeitabständen wiederholt werden, so dass eine objektive Bewertung des Dienstleisters möglich ist und die Bewertungskriterien einfach offengelegt werden können [O11].

easyBird Cleaning Cloud

Die *easyBird Cleaning Cloud* [12] ist ein System, mit dessen Hilfe die Qualitätssicherung und das Mängelmanagement im Bereich der Gebäudereinigung vereinfacht werden soll. Objektleiter erhalten einen schnellen Überblick über den Stand der Dienstleistung und können über die

[10] `http://kaercher-katalog.sbr24.de/2013/2013_Kaercher_Professional_160.pdf`
[11] `https://www.speedikonfm.com`
[12] `https://www.sc-protec.de`

Kommunikations- und Dokumentations-Plattform das Auftragsmanagement übersichtlich abwickeln. Das angebotene System verspricht, dass die Nachweisführung des gesamten Tätigkeitsprofils gegenüber Auftraggebern oder Eigentümern von Objekten einfacher werden soll. Zur Prozessabwickling werden mit mobilen oder stationären Barcode- oder RFID-Lesegeräten Daten erfasst und per UMTS mit einer Datenbank verknüpft.

Die Software selbst ermöglicht es, Arbeitspläne zu erstellen, Routen festzulegen, Tätigkeiten zu dokumentieren, Personal zu verwalten. Mängelanzeigen helfen den erwünschten Standard der Dienstleistung zu halten. Aufgetretene Mängel können nach erfolgreicher Erledigung über die Internet-Plattform des Systems oder durch erneutes Scannen des Raumes als erledigt vermerkt werden [O2].

Genauere Informationen bezüglich der eingesetzten mobilen Plattformen oder Betriebssysteme sind auf der Website nicht beschrieben.

Capitano

Die *PREVERA Consulting GmbH* [13] bietet mit *Capitano* ein webbasiertes mobiles System zur Leistungsdokumentation und Qualitätskontrolle für Facility-Manager und Gebäude- bzw. Hausverwalter und Gebäudereiniger an.

In [O16] wird beschrieben, dass Dienstleister über die WebApp alle durchgeführten Tätigkeiten sowie Material, Maschinen und Geräte erfassen können. Mängel und Schäden können zusätzlich mit Fotos dokumentiert und verarbeitet werden. GPS-Standortüberprüfung beim Check-In, Erinnerungs-Mails und Check-Funktionen für periodische Arbeiten sowie QR-Code Unterstützung runden die Funktionen von *Capitano* ab. Das System ist modular aufgebaut, kann somit individuell auf den Kunden angepasst werden und bietet parallelen Zugriff durch Echtzeit-Datenerfassungen aus dem Internet.

Das Basismodul, mit welchem das Dokumentieren und Kontrollieren (nach DIN 13549) durchgeführter Reinigungsarbeiten und Dienstleistungen sowie deren Qualität am Objekt möglich ist, beinhaltet auch die Verwaltung der Objekte, Checklisten und Basisfunktionen. Ein Modul zur Dokumentation und Kontrolle der Anwesenheit und der Tätigkeiten aller Dienstleister im Objekt ist ebenfalls erhältlich.

Capitano bietet auch als eines der wenigen Systeme auf dem Markt ein Feature für Endkunden in Form von einer Kundenbefragung, welche individuell gestaltet und ausgewertet werden kann.

pit-Mobile

In [O15] beschreibt die Firma *pit - up GmbH* [14] einen mobilen Smartphone Client zur Zeit- und Leistungserfassung sowie für Datenerfassung bei Kontrollgängen in der Reinigungskontrolle und anderen Einsatzgebieten.

Der Dienstleister wählt aus einer Liste mit offenen Aufträgen den aktuell anstehenden Auftrag aus und entnimmt dem System die dafür notwendigen Informationen. Anschließend arbeitet

[13] `http://www.prevera.at/`
[14] `http://www.pit.de`

er die zu erbringenden Arbeitsschritte nacheinander ab und markiert den Status der erbrachten Leistungen. Auf diese Weise können die Aufträge direkt am Gerät abgearbeitet werden [O15].

Durch den Online/Offline-Modus ist auch ein problemloses Arbeiten in Bereichen ohne Internet- oder Telefonnetz möglich. Vor Ort erstellte Dokumente, Fotos oder Sprachnotizen werden beim nächsten Synchronisationsvorgang mit den entsprechenden Objekten am Applikationsserver verlinkt und können später weiterbearbeitet werden [O3].

Durch die sichere Authentifizierung über Benutzername und Kennwort ist gewährleistet, dass jedem Mitarbeiter nur seine ihm zugewiesenen Daten und Aufgaben übersichtlich gefiltert angeboten werden [O3].

Die Digitalkamera oder externe Barcode- und RFID-Scanner können über Bluetooth gekoppelt werden um Auto-ID zu verwenden. Die Dokumentation im Mängelmanagement ist durch Fotonachweise oder durch Sprachnotizen zu bewerkstelligen. Für *Outdoor*-Tätigketen verwendet die App auch den GPS-Empfänger zur Standortprotokollierung. Bei Arbeitsende wird dem Dienstleister ein Signaturfeld angezeigt, in welchem dieser mit seiner Unterschrift auf dem Gerät die geleistete Arbeit bestätigen muss, und in weiterer Folge bekommt dieser eine Bestätigung des Arbeitsaufwands per E-Mail.

Überblick und Zusammenfassung

Im Abschnitt 2.3 wurden bestehende mobile Reinigungsmanagementsysteme evaluiert. Um nun in weiterer Folge eine Anforderungsliste für das Eigensystem erstellen zu können, werden die evaluierten mobilen Systeme hinsichtlich der Teilschritte des Geschäftsprozesses aus Abschnitt 2.2 gegenübergestellt. Durch diese Gegenüberstellung soll sich letztendlich eine Liste mit Funktionalitäten für das zu entwickelnde Eigensystem ergeben.

ID	Beschreibung
A	An- u. Abmeldung mit Arbeitszeiten Dokumentation
B	Informationen zu Raumbelegung
C	Checkliste zur Fläche nach DIN EN 13549
D	Reinigungsprotokollierung
E	Nacharbeiten-Management
F	Betriebsmittel Checkliste
G	Besonderheiten/Mängel-Protokoll
H	Selbstkontrolle
I	Protokollabschluss und Übergabe
J	Einbezug des Endnutzers
K	QMS nach DIN EN 13549

Tabelle 2.7: Teilschritte des Geschäftsprozesses

	Leistungserbringer									Auftragnehmer	Endnutzer	Auto-ID
Teilschritt ID (2.7)	A	B	C	D	E	F	G	H	I	J	K	
Kärcher	X	-	-	-	-	-	X	-	X	-	X	X
speedkonFM	X	-	-	X	X	-	-	-	X	-	-	-
easyBird Cleaning	X	-	-	X	-	-	X	-	X	-	-	X
Capitano	X	-	X	X	-	X	X	-	X	X	X	X
pit-Mobile	X	-	-	X	-	-	X	-	X	-	-	X

Tabelle 2.8: Vergleich der Teilschritte im Geschäftsprozess mit den evaluierten mobilen Systemen mit Auto-ID Einsatz

Resultierende Funktionalitätenliste

Betrachtet man in Tabelle 2.8 die Gegenüberstellung der evaluierten Systeme so ist ersichtlich, dass keine untersuchte Softwarelösung den Geschäftsprozess vollständig abbildet. Durch Kombination der Funktionalitäten ergibt sich eine vollständige Liste an Merkmalen für das Eigensystem.

- Auftragnehmer
  - Personalisierter Login mit Rechte für Auftragnehmer (Objektleiter)
  - Zugriff Objekt- und Stammdaten, Leistungsverzeichnis je Raumgruppe
  - Qualitätssicherung nach DIN EN 13549 (Ablaufdiagramm siehe Abb. 2.5)
  - Zugriff auf das Mängelmanagement
  - Erweiterung/Bearbeitung der Auto-ID Infrastruktur
- Leistungserbringer
  - Personalisierter Login mit Rechten für Leistungserbringer
  - Zugriff auf Leistungsverzeichnis je Raumgruppe
  - Reinigungsprotokoll je Fläche
  - Checkliste für Betriebsmittel vor Beginn der Reinigungstätigkeiten
  - Handhabung von Nacharbeiten
  - Mängel-Management und Protokollierung
  - Berücksichtigung des Raumbelegungsplanes
  - Selbstkontrolle nach DIN EN 13549
  - Arbeitsabschluss mit Protokollübermittlung
- Endnutzer
  - Vorort-Zugriff auf eingeschränkte Flächendaten
  - Übermittlung von Nachrichten für den Objektleiter
  - Reinigung auf Bestellung

KAPITEL 3

Konzept des Prototyps

Im Kapitel 2 wurden die Grundlagen von NFC und die aktuellen Gegebenheiten im (mobilen) Reinigungsmanagement beschrieben. Auch ist durch die Evaluierung von bereits bestehenden Systemen eine Funktionalitätenliste (2.3) für das Eigensystem entstanden.

So wird im nachfolgenden Kapitel das Grundkonzept des zu entwickelnden Prototyps vorgestellt, welcher im Kapitel 4 auf Basis dieses Konzeptes implementiert werden soll.

Hierfür wird zunächst die Art des Systems im Zusammenhang mit dem Begriff *Internet of Things* beschrieben und auf die wesentlichen Prozessunterstützungen mit Hilfe von NFC eingegangen. Die Software-Architektur bestehend aus Smartphone-Client, Webservice und Endnutzer-Website wird gemeinsam mit der im realen Objekt eingesetzten NFC-Tag Infrastruktur näher erläutert.

3.1 Grundkonzept

Systemcharakter

Der Begriff *Internet of Things (IoT)* (zu Deutsch „Internet der Dinge“) erschien erstmals 1999 als Titel einer Präsentation von *Kevin Ashton* [O4]. Es will die Integration von echten, sich im realen Umfeld befindlichen *Dingen*, mit dem *Internet* ausdrücken um somit die Informationslücke zwischen realer und virtueller Welt zu minimieren. Dabei spielt es keine Rolle, ob das Ding von technischer Natur ist oder nicht [L4] und somit können auch Objekte ohne Kommunikationseinheit in das IoT integriert werden. Mit Datenkommunikationsgeräten, welche mit dem Internet verbunden sind, können diese *nicht-technischen* Dinge etwa mit Hilfe von RFID-Tags oder Barcodes gelesen werden. Diese *Smart Things* können für jegliche Art von Objekten auf der Welt stehen und als Kommunikationsknoten eingesetzt werden. Betrachtet man den Begriff *Internet of Things* aus syntaktischer Sicht, so ist zu erkennen, dass sich mit *Internet* und *Things* zwei Teilbereiche ergeben. Diese globale Informationsarchitektur spaltet sich in eine *Ding-orientierte* und eine *Internet-orientierte* Version auf. Erstere beschäftigt sich mit dem Problem, diesen Dingen einen (Speicher-)Status, eine Sichtbarkeit und einen Standort zu verleihen

und diesen auch erkennen zu können. Die *Internet-orientierte* Version hingegen beschäftigt sich damit, diese realen Objekte in das Internet zu integrieren. Dabei handelt es sich um das sogenannte *Web of Things*, einem Ansatz um physikalische Objekte auf einfache Weise mit virtuellen oder physischen Ressourcen zu kombinieren [L16].

Die Integration der Dinge in das Internet soll im Kontext dieser Arbeit in Form von Ressourcen, welche durch einen Webservice gestützt sind, erreicht werden. Dabei soll dieser Webservice ein virtuelles Alter Ego des Dings repräsentieren. Der Status des physischen Objektes, welcher mit NFC-Tags gespeichert wird, beeinflusst dabei den Webservice und umgekehrt [L1].

Zur Unterstützung der Reinigungsprozesse aus Sicht der verschiedenen Rollen beschrieben im Abschnitt 2.2 soll ein Konzept für einen Prototyp, welcher die Funktionalitätenliste aus Abschnitt 2.3 zusammenfasst, erarbeitet werden. Da es sich in den Geschäftsprozessen der Reinigung um keine sitzende Bürotätigkeit handelt, sondern aktive Vorort-Dienstleistungen notwendig sind, soll eine mobil einsetzbare Anwendung auf einem Smartphone konzeptioniert werden.

Das Grundkonzept sieht somit primär eine Interaktion zwischen Nutzern und zu identifizierenden Objekten vor um das *Internet of Things* mit den Anwenderkonzepten von Smartphones vereinen zu können.

Prozessunterstützung durch NFC

Die *Ding-orientierte* Version des *Internet of Things* benötigt einige Komponenten um Kommunikation zwischen Endgeräten und Dingen zu ermöglichen. Im Kontext dieser Arbeit sollen diese *Dinge* mit einer Auto-ID Technologie versehen werden um identifiziert werden zu können [O34]. Im Abschnitt 2.3 wurden zwei dieser Technologien vorgestellt, welche im FM bereits Anwendung finden. Da der Software-Prototyp auf einem Smartphone implementiert werden soll, sind mit aktuellen *Android*-Geräten grundsätzlich beide Technologien getrennt oder auch in Kombination einsetzbar. Das mobile Betriebssystem stellt hierfür Schnittstellen zu Technologien zur Verfügung um einerseits Barcodes durch die integrierte Kamera zu verarbeiten, oder stellt auch NFC fähigen Geräten die Möglichkeit als RFID-Lese/Schreibgerät (PCD) agieren zu können in den Raum. Die im Abschnitt 2.3 erwähnten Nachteile der automatischen Identifikation per Barcode sollen durch den Einsatz von NFC im zu entwerfenden Prototyp ausgeglichen werden.

Konzept für Rollen-, Flächen und Aktionsidentifikation

Die zu erarbeitende Software- und Hardware-Infrastruktur sieht ein Rollenkonzept vor. Jede Rolle, sei es ein *Auftragnehmer*, *Leistungserbringer* oder ein *Endnutzer*, soll Teile oder die komplette Infrastruktur nutzen können. Der Prototyp soll hierfür nicht mit einem herkömmlichen, durch ein Benutzername/Passwort basiertes Authentifizierungsverfahren ausgestattet werden, sondern die Identifikation, und somit auch die Rollendefinition, soll durch NFC-Tags bewerkstelligt werden. Abhängig vom, nach der Anmeldung zugrundeliegenden Status, soll der Nutzer verschiedene Möglichkeiten haben um mit der Software zu interagieren.

Aus Sicht des *Auftragnehmers* soll es möglich sein, den Benutzer durch den Qualitätssicherungsprozess zu leiten und diesen geeignet auszuwerten. Das Prüfsystem soll nach DIN EN 13549 [NS5] implementiert werden. Dabei soll es Aktions-Tags geben, welche es möglich ma-

chen, Qualitätssicherungskontrollen und darin befindliche Stichproben zu erzeugen. Außerdem sollen der Fläche entsprechend Informationen abrufbar sein und Bewertungsmasken zur Verfügung stehen. Die Identifikation dieser Flächen, zu welcher alle Arten von Räumlichkeiten im Gebäude zählen, soll ebenfalls per NFC realisiert werden. Auch soll es dieser Benutzergruppe möglich sein die bestehende NFC-Infrastruktur zu erweitern und zu bearbeiten. Das sieht vor, dass auch Zugriff auf die Datenbank besteht um neue Datensätze in dieser anzulegen, und mit dem mobilen Client NFC-Tags zur Identifikation von Personen, Räumlichkeiten und Aktionen zu programmieren sind.

Der *Leistungserbringer* hingegen wird durch Einsatz des Systems hauptsächlich bei der Protokollierung und der Arbeitszeiterfassung unterstützt. Durch das Scannen von Aktions-Tags für Arbeitsbeginn und Arbeitsende ist eine genaue Erfassung der Arbeitszeit möglich. Er soll zusätzlich die Möglichkeit haben, sich durch Unterstützung und dem „Wissen“ der Software selbst einer Kontrolle zu unterziehen. Diese Funktionalität wird nicht jene sein, die eine rege Alltagsanwendung erwartet, durch immer höhere Fluktuationsraten im Reinigungspersonal – speziell auf Seite des Leistungserbringers – jedoch eine wichtige Trainings- und Lernfunktionalität darstellen wird. Auch soll durch den in der Datenbank hinterlegten Raumbelegungsplan ein System entstehen, das weiß, ob eine Fläche tatsächlich gereinigt werden muss und dies auch zum aktuellen Zeitpunkt möglich ist. Das Mängel-Management sowie das Auftreten von besonderen Ereignissen soll ebenfalls erst nach Scannen eines Tags mit dem Client dokumentierbar und in Echtzeit raumgebunden übertragbar sein.

Da die Rolle des *Endnutzers* letztendlich nicht für den Bezug der mobilen Software vorgesehen ist, soll hierfür ein spezielles Interaktionskonzept basierend auf NFC oder manuellen Eingaben erarbeitet werden. Im ersteren Fall (Auto-ID) wird vorausgesetzt, dass Endnutzer ihr privates mobiles NFC-Smartphone einsetzen können. Jedoch müssen sie dies nicht, da auch eine manuelle Variante nutzbar sein soll. Endnutzer sind jene Personen, die Reinigung – und im besten Fall auch die dadurch resultierende Sauberkeit – empfangen. Diese Rollendefinition sieht vor, dass Reinigung in geeigneter Weise bewertet, oder in speziellen Fällen eine Reinigung auf Bestellung geordert werden kann.

Alle im Geschäftsprozess teilnehmenden Rollen sollen die NFC-Infrastruktur nutzen, in einigen Anwendungsbereichen soll wie erwähnt, je nach Rollenstatus, auch bewusst darauf verzichtet werden können.

Nutzen und Vorteile

Für Leistungserbringer ergibt sich der wesentliche Vorteil durch den Einsatz des Systems, dass der komplette Verzicht auf jegliche Art der Protokollführung in Form von Papierdokumenten wegfällt. Die Möglichkeit einer Selbstkontrolle auf Basis der Fehlerhandhabung des QM-Systems wird ebenfalls als Vorteil gesehen.

Aus Sicht des Auftragnehmers ist es auch möglich den Gesamtprozess in einer für ihn geeigneten Art und Weise überschaubar zu verwalten und auf eine Art Live-Übersicht über alle aktiven Tätigkeiten im Objekt zuzugreifen. Die Ersparnis von Nacharbeiten zur digitalen Erfassung von Protokollen fällt durch den Charakter eines Echtzeit-basierten Client-Server Systems ebenfalls weg. Die Ausgabe und Archivierung von Checklisten, Leistungsverzeichnissen, Belegungsplä-

nen und Mängelprotokollen, sowie Auswertungsbögen bei der Qualitätssicherung sollen digital, raumbezogen und immer aktuell verfügbar sein.

Durch den möglichen Miteinbezug der Raumnutzer ist diesem ebenfalls ein Einfluss auf die von ihm gewünschte Sauberkeit im genutzten Raum möglich.

3.2 Systemkomponenten und deren Anforderungen

Im Abschnitt 2.3 wurden verschiedene Arten von Softwarelösungen für das Reinigungsmanagement vorgestellt. Für diese Arbeit wird vorausgesetzt, dass bereits ein CAFM-System mit einer zugrundeliegenden Datenbank als Anwendungssoftware durch den verantwortlichen FM-Service eingesetzt wird.

Diese zentrale Datenbank kann durch Objektleiter und zuständige *Facility-Services* verwaltet werden. Durch den Einsatz des mobilen Clients soll diese Datenbank lediglich genutzt werden um autorisierten Personal Zugriff auf dessen hinterlegte Grundlageninformationen wie Flächen, Raumbelegungen, etc. zu gestatten. Diese Informationen sollen in weiterer Folge bei der Durchführung von Dienstleistungen als Hilfestellung zur Abarbeitung und Protokollierung von Reinigungsprozessen und dessen Qualitätssicherung dienen.

Zur Identifikation von Personal und Flächen sowie zur Erzeugung von Aktionen wie Arbeits- und Kontrollbeginn muss eine NFC-Tag Infrastruktur konzeptioniert werden, wie sie zum Teil bereits im Abschnitt 3.1 vorgestellt wurde. Diese Tags werden im Falle vom Personal einerseits mit sich getragen und andererseits auf realen Objekten zur Identifikation angebracht. Ausgehend von den verschiedenen Typen von NFC-Tags, welche im Abschnitt 2.1 vorgestellt wurden, ist zu entscheiden welche Art von Tags auf welche Weise angebracht werden soll und wie das Speichermanagement zur Hinterlegung der Datensätze organisiert werden soll. Hierfür ist auch zu berücksichtigen, dass Tags nicht unberechtigt manipuliert werden dürfen. Hinsichtlich Anschaffungskosten muss auch auf wirtschaftlicher Ebene entschieden werden.

Um Kommunikation zwischen Clients und dem CAFM-System zu ermöglichen, muss eine Server-Schnittstelle entwickelt werden um Anfragen geeignet zu handhaben. Das CAFM-System soll hierdurch eine Anbindung an das Internet bekommen um dessen hinterlegte Datensätze in geeigneter Weise an die mobilen Clients zu servicieren.

Die zu entwickelnde Applikation für das Smartphone soll die bereits erwähnten Systemkomponenten hinsichtlich ihrer Anwendung vereinen. Einerseits wird durch die automatische Identifikation von Personen/Objekten/Aktionen durch NFC eine Möglichkeit genutzt, dem Benutzer den Arbeitsaufwand bezüglich manueller Eingaben und einer dadurch vorhandenen Fehlerquelle gering zu halten. Andererseits bietet das Smartphone die Möglichkeit über Internetprotokolle seinen mobilen Charakter zu bewahren und über Web-Schnittstellen auf das CAFM-System zurückzugreifen. Diese Art von Schnittstelle ist besser bekannt als *Webservice* und wird im folgenden Abschnitt näher beschrieben.

Endnutzer sind nicht für den direkten Bezug der mobilen Anwendung vorgesehen, dennoch soll, wie in den Kapiteln zuvor erarbeitet, diese Rolle nicht vernachlässigt werden. Der Anwendungsfall sieht vor, dass Endnutzer die NFC-Tag Infrastruktur in einem für sie „offenen“ Zustand nutzen können um etwa Reinigung auf Bestellung oder eine Art Zustandsbewertung hinterlassen zu können. Für diese Funktionalität ist es von Nöten, dass eine Website entwickelt

wird, welche dem Endnutzer beim Scannen eines Tags automatisch präsentiert wird um Eingaben tätigen zu können. Hierfür soll im Kapitel 4, das sich mit der Implementierung befasst, das NDEF-Format auf NFC-Tags zum Einsatz kommen, welches bereits im Abschnitt 2.1 vorgestellt wurde. Da der Endnutzer nicht zwingend NFC für diesen Anwendungsfall einsetzen muss, was ein privates NFC-fähiges Smartphone voraussetzen würde, muss auch eine manuelle Variante bereitgestellt werden. Hierfür sollen Konzepte des *Responsive Webdesign* zum Einsatz kommen um den grafischen Aufbau einer Website entsprechend dem darstellenden Endgerät anzupassen.

In den folgenden Abschnitten sollen die Systemkomponenten als einzelne jeweils genauer beschrieben werden.

Webservice

Der Webservice ist die zentrale Anlaufstelle für alle mobilen Clients sowie die Endnutzer Website. Im Kontext des *Internet of Things* soll er die *Internet-orientierte* Version darstellen um die realen *Dinge* mit virtuellen Komponenten zu kombinieren. Das Ziel ist es Daten aus dem CAFM-System zu erhalten und Schnittstellen für Business-Logiken der Prozessabwicklung bereitzustellen. Der zu entwickelnde Service läuft auf einer separaten Server-Umgebung. Sicherzustellen ist nur, dass auf das Datenbankmanagementsystem des CAFM-Systems zugegriffen werden kann und darf. Mit Hilfe der Schnittstelle soll ein standardisierter Zugriff durch alle Clients auf die Grundlagendaten des Objektes erfolgen. Würde diese Zwischenschicht nicht existieren, wäre es Angreifern durch den Einsatz von Decompilern möglich Informationen über die Datenstruktur des CAFM-Systems zu erlangen. Außerdem wird durch den Einsatz eines Webservices sichergestellt, dass im Falle von Designänderungen der CAFM-Datenbank oder der Serverumgebung nicht jeder Client einem Update unterzogen werden muss.

Wie bereits eingangs erwähnt wurde, wird die Annahme getroffen, dass bereits ein solches stationäres CAFM-System im Einsatz ist und u.a. folgende Daten beinhaltet.

- Personaldaten
- Objekt- und Stammdaten
- Raum- und Raumgruppenverzeichnis
- Flächendaten
- Raumbelegungsplan
- Inventar

Der Webservice soll in weiterer Folge diese Daten für die Entwicklung des Prototyps simulieren. Dies impliziert, dass die Server-Umgebung des Webservices selbst eine Datenbank implementiert und seine Daten direkt auf derselben Instanz zur Verfügung stellt wie jene, die vom CAFM-System bereitgestellt werden würden.

Da der Webservice nicht nur auf die CAFM-Daten zugreifen kann, sondern auch eigene Funktionalitäten und Logiken bereitstellen soll, müssen insgesamt folgende Schnittstellen entsprechend der erarbeiteten Funktionsliste aus Abschnitt 2.3 zur Verfügung gestellt werden um die Anwendungsfälle abzudecken:

- Auftragnehmer

  - Benutzer Login
  - Abruf von Objekt- mit Stammdaten
  - Abruf einer Stichprobe zur Qualitätssicherung
  - Abruf des Leistungsverzeichnisses mit Flächendaten je Raumgruppe/Raum
  - Übermittlung der Auswertung einer Qualitätssicherung je Fläche
  - Abruf von Mängel im Raum
  - Hinzufügen einer Mängelbeschreibung
  - Abruf von neuen Raum-Tags
  - Abruf von neuen Personal-Tags
  - Registrieren eines Raum-Tags
  - Registrieren eines Personal-Tags
- Leistungserbringer
  - Benutzer Login
  - Arbeitsbeginn
  - Arbeitsende
  - Abruf der Betriebsmittel-Checkliste
  - Abruf des Leistungsverzeichnisses mit Flächendaten je Raumgruppe/Raum
  - Abruf von Nacharbeiten-Aufträgen
  - Hinzufügen des Reinigungsprotokolls je Fläche
  - Hinzufügen einer Mängelbeschreibung
  - Abruf des Raumbelegungsplanes
- Endnutzer
  - Abruf von Raumdaten
  - Übermittlung eines Kommentars

NFC-Infrastruktur

Laut Vision der *Internet of Things* Informationsarchitektur ist es möglich reale *Dinge* mit einem virtuellen System zu verbinden [L16]. Für die Entwicklung des Prototypen werden nachfolgend verschiedene NFC-Tag Gruppen hinsichtlich ihrem tatsächlichen Anwendungsfall näher beschrieben. Bei diesen Gruppen handelt es sich im IoT-Kontext um die *Ding-orientierte* Version, welche den Status repräsentiert, der durch die *Internet-orientierte* Version (dem Webservice) verarbeitbar ist. Jeder Tag muss einer Gruppe von *Dingen* zugeordnet werden um später die entsprechende Schnittstelle des Webservices nutzen zu können.

Alle im Geschäftsprozess eingesetzten NFC-Tags müssen auf individuelle Art und Weise in das Gesamtsystem integriert werden. Hierfür gilt, dass alle Tags durch ein geeignetes Authentifizierungsverfahren gesichert werden können um Datenmanipulation durch unberechtigte Dritte

zu verhindern. Nur durch autorisiertes Personal soll es möglich sein die hinterlegten Daten zu verändern. Für eine solche Sicherung und Schreibbefugnisse soll die mobile Anwendung mit Benutzerrechten für Auftragnehmer (Objektleiter) zuständig sein, auf welche im Abschnitt 3.2 eingegangen wird.

Raum-Tags

Raum-Tags repräsentieren im Gebäude öffentlich verfügbare Identifikationsperipherie. Da im vorgestellten Rollenkonzept aus Abschnitt 3.1 auch Rücksicht auf Endnutzer genommen wird, kann die Anbringung von solchen Tags nicht auf eine möglichst unscheinbare Art und Weise stattfinden. Dennoch muss darauf geachtet werden, dass Tags dieser Gruppe nicht bzw. nur schwer oder mit anderen technischen Hilfsmitteln – etwa für einen Austausch – entfernt werden können. Die Montage sollte in unmittelbarer Nähe der zu identifizierbaren Räumlichkeit, etwa direkt an oder neben der Türe (Abb. 3.1), durchgeführt werden. Im Falle von Raum-Tags ist es nicht von entscheidender Bedeutung, dass nur angestelltes Personal die Daten auf den Tags auslesen kann, da sie auch für Endnutzer zum Einsatz kommen können. Idealerweise soll auf Tags

Abbildung 3.1: Beispielhafte Montage eines Raum-Tags
Quelle: www.allviewmobile.de

dieser Gruppe eine Identifikationsnummer für die Räumlichkeit hinterlegt werden, welche auch im CAFM-System registriert ist. Diese Identifikation soll gemeinsam mit aktionsspezifischen Parametern an den Webservice zur weiteren Verarbeitung übermittelt werden können. Außerdem ist es von entscheidender Bedeutung, dass auch Endnutzer durch eine hinterlegte URL Daten im Smartphone-Browser abrufen können.

Personal-Tags

Personal-Tags sind an Dienstleister gebundene Tags. Sie sollen durch Leistungserbringer und Auftragnehmer (Objektleiter) nutzbar sein und befinden sich in deren direktem Besitz. Sie sollten zu jeder (Arbeits-)Zeit mit sich getragen werden und können je nach Bauart entweder in Form eines Schlüsselanhängers oder als Scheckkartenformat in der Brieftasche transportiert

werden (Abb. 3.2). Ihr Nutzen ist es, sich durch die hinterlegten Daten in der mobilen Applikation zu authentifizieren. Eine Authentifizierung ist notwendig um die Rolle (Benutzergruppe) zu bestimmen, in welcher die Applikation Anwendung finden soll. Ohne aktive Nutzung dieser Tags soll die Applikation nicht verwendet werden können. Das stellt sicher, dass nach einer möglichen Entwendung des NFC-Smartphones kein Zugriff durch unberechtigte Dritte mittels ausgespähter Logindaten möglich wäre. Die am Tag gespeicherten Daten sollen nur mithilfe

Abbildung 3.2: Portable Personal-Tags
Quelle: www.amazon.de

der Applikation lesbar sein und sind nicht für die Öffentlichkeit bestimmt. Für das Personal ist es von entscheidender Bedeutung, dass nach einem Verlust des eigenen Tags dieser durch den Objektleiter im System als ungültig markiert werden kann. In einem solchen Fall kann durch berechtigtes Personal mit der mobilen Anwendung ein Ersatztag programmiert werden. Die im Abschnitt 2.1 beschriebenen Tag-UIDs sollen hierfür im Zusammenhang mit der Komponente *Webservice* den eigentlichen Nutzen erweitern um schließlich Tags auch online verwalten zu können (4.4).

Auf diesen Tags soll idealerweise eine Personal-ID hinterlegt werden, welche in weiterer Folge durch das Webservice entgegengenommen wird um weitere Entscheidungen betreffend Anwendungsfunktionalität in der Applikationsoberfläche treffen zu können.

Aktions-Tags

Aktions-Tags sollen durch Auftragnehmer und Leistungserbringer nutzbar sein. In dieser Tag-Gruppe sollen insgesamt nur zwei Tags zum Einsatz kommen. Je nach Rollenstatus ergeben sich jedoch zwei getrennte Anwendungsfälle.

Im Falle eines Leistungserbringers soll es nach dem Benutzer-Login möglich sein den Arbeitsbeginn durch Scannen eines entsprechenden Tags mit Hilfe des Webservices im Online System zu registrieren. Dieser Anstoß des Arbeitsbeginns soll zeitgleich auch den Reinigungsplan sowie die notwendigen möglichen Nacharbeiten zurückliefern. Beim Beenden der Arbeit soll analog ein Tag verwendet werden, welcher den Zeitpunkt des Arbeitsendes an das System bekanntgibt und den Abschluss der an diesem Tag fälligen Tätigkeiten übermittelt. Erst nach

erfolgreichem Arbeitsende soll ein erneuter Arbeitsbeginn und ein damit erzeugter aktueller Reinigungsplan zurückgeliefert werden können.

Im Falle eines Objektleiters (Auftragnehmers) soll nach dem Benutzer-Login mit Hilfe des Aktionsbeginn-Tags eine Stichprobe mit den zu prüfenden Räumlichkeiten entsprechend dem Objekt erzeugt werden können. Das Scannen des analogen Tags zum Abschluss der Qualitätssicherung nach Überprüfung aller Räumlichkeiten soll die Auswertung des Gesamtergebnisses einleiten.

Diese beiden Tags sollen nach Möglichkeit im Aufenthaltsraum des Objektleiters angebracht werden. Das hat einerseits den Nutzen, dass dieser noch vor Arbeitsbeginn mit dem Leistungserbringer Absprache halten kann falls Nacharbeiten notwendig sind. Andererseits ist durch die Existenz von zwei stationär angebrachten Aktions-Tags zur Arbeitszeiterfassung die Kontrolle darüber gegeben, dass diese nicht an einem beliebigen Ort gescannt werden können. Damit wird der Manipulation von Zeiterfassung entgegengewirkt, welche bei Verwendung durch benutzergebundene Tags (Personal-Tags) möglich wäre. Ähnlich wie bei Personal-Tags sollten Aktions-Tags nur durch die mobile Anwendung lesbar sein.

Für die Programmierung solcher Tags ist eine Aktions-ID notwendig um zwischen diesen zu unterscheiden um mit dem Webservice entsprechend kommunizieren zu können. Außerdem befinden sich Aktions-Tags in genau einer Liegenschaft, was die Speicherung einer zusätzlichen Objekt-ID notwendig macht.

Wahl des Tag-Typs

Ausgehend von den im Abschnitt 2.1 vorgestellten NFC-Tag Typen und der im Abschnitt 3.2 resultierenden Anforderungen an die NFC-Tag-Infrastruktur muss entschiedenen werden, welcher Chip-Typ für die Entwicklung des Prototyp zum Einsatz kommen soll.

Tabelle 3.1 fasst die Anforderungen an die NFC-Tag-Infrastruktur nochmals zusammen:

Tag-Gruppe ▷ Eigenschaft ▽	Raum Tag	Personal Tag	Aktions Tag
Anwendung	Raumidentifikation	Personalidentifikation, Login	Zeiterfassung, Stichproben
Nutzbarkeit	AN, LE, EN	AN, LE	AN, LE
Lesezugriff	Öffentlich	App: AN & LE	App: AN & LE
Schreibzugriff	App: AN	App: AN	App: AN
Datenspeicher	Raum ID	Personal ID	Aktions ID
Verfügbarkeit	fixe Montage	portabel	fixe Montage
Anzahl	für jede Fläche im Reinigungsprozess	pro AN, LE	2

Tabelle 3.1: Eigenschaften der NFC-Tag Infrastruktur (AN: Arbeitnehmer, LE: Leistungserbringer, EN: Endnutzer)

Die Anzahl der Raum-Tags ist vom Objekt abhängig, welches den Geschäftsprozessen der Reinigung unterzogen werden soll. Dies kann je nach Objekt und Flächen zu einer hohen Anzahl

an Raum-Tags führen. Die Summe der Personal-Tags hingehen ist an die Anzahl der Dienstleister gebunden und sollte für die Entscheidung hinsichtlich des Kostenfaktors nicht ausschlaggebend sein. Die Anzahl der Aktions-Tags ist am geringsten, kann jedoch je nach Anwendungsfällen in Zukunft beliebig erweitert werden.

Da es sich beim vorliegenden Konzept nicht etwa wie bei einem kontaktlosen Zahlungssystem um ein ausgesprochen sicherheitskritisches handelt, jedoch dieser Aspekt nicht ganz außer Betracht gelassen werden kann, muss die Entscheidung hinsichtlich der Parameter Kosteneffizienz und Sicherheit getroffen werden. Außerdem sieht das Konzept vor, dass keine großen Datenmengen auf Tags geschrieben werden müssen. Tags des Typ 2 wurden ausgiebig getestet und werden als zuverlässig, sicher, kosteneffizient und als ausreichend hinsichtlich ihrer Speicherkapazität angesehen und sollen in weiterer Folge für den Prototyp zum Einsatz kommen. Durch die Verwendung von *Mifare Ultralight C* Tags, welche Typ-2 kompatibel sind, ist die Möglichkeit einer 3DES-Authentifizierung und somit das notwendige Maß an Sicherheit für den weiteren Projektverlauf gegeben. Auf die Implementierung und die Umsetzung dieses Authentifizierungsverfahrens wird im Kapitel 4 genauer eingegangen.

Smartphone-Client

Die System-Architektur (3.2) sieht einen Smartphone-Client als Hauptkomponente zur Unterstützung von Reinigungsprozessen und dessen Qualitätssicherung im Gesamtsystems vor. Ähnlich wie bei den durch NFC-Tags repräsentierten Objekten handelt es sich hierbei ebenfalls um ein reales *Ding*. Dieses ist jedoch nicht wie die zu identifizierenden Objekte keiner technischen Natur sondern, besitzt selbst die Möglichkeit sich mit dem Internet zu verbinden um Daten abzurufen und zu verarbeiten. Wie eingangs im Abschnitt 3.1 erwähnt handelt es sich dabei um ein Datenkommunikationsgerät, welches als *Ding* in das Internet integriert ist um mit der *Internet-orientierten* Version (Webservice) zu interagieren.

Dieser mobile Client soll durch Auftragnehmer und Leistungserbringer nutzbar sein. Auf dessen Implementierung wird im Kapitel 4 genauer eingegangen. Nachfolgend soll die Client-Applikation hinsichtlich der geforderten Funktionalitäten aus Abschnitt 2.3 genauer beschrieben werden.

App-Start und Übersichtsseite

Beim Starten der Applikation soll dem Benutzer der Status vermittelt werden, ob das Smartphone mit der aktuellen Konfiguration bereit ist eingesetzt zu werden. Das Rollenkonzept aus 3.1 sieht vor, dass Personal sich nur mit Hilfe eines entsprechenden NFC-Tags (Personal-Tag siehe 3.2) am Online-System anmelden kann. Hierfür soll es dem Benutzer nach Applikationsstart möglich sein den persönlichen Tag am Gerät anzuhalten um sich mit dessen hinterlegten Daten am System anzumelden. Die Anmelden-Funktionalität muss bei jedem Start der Applikation ausgeführt werden und hält nur solange der Benutzer als eingeloggt markiert ist.

Nachdem der Benutzer erfolgreich eingeloggt wurde, werden die für ihn bestimmten Funktionalitäten entsprechend dem Rollenstatus freigeschaltet und auf der Übersichtsseite zusammengefasst. Je nach Anwendungsfall und Rolle soll die Initiierung einer Funktionalität durch

Scannen eines Tags oder durch manuelle Eingaben stattfinden. In den folgenden Abschnitten wird auf diese Gegebenheit genauer eingegangen.

Auftragnehmer Funktionalität

Der Rolle eines Objektleiters sollen nach erfolgreichem Login die Hauptfunktionalitäten zur Qualitätssicherung nach DIN EN 13549 (2.2) sowie die Möglichkeit der Erweiterung und Bearbeitung der NFC-Tag-Infrastruktur zur Verfügung stehen.

Nachdem der Benutzer mit dem persönlichen Tag angemeldet wurde, soll entweder automatisiert eine aktive, noch nicht abgeschlossene Qualitätssicherung fortgesetzt werden können, oder die Applikation soll dazu bereit sein eine neue Stichprobe durch Scannen eines Aktions-Tags (Kontrollbeginn) zu erzeugen. Durch Weiterführung oder Neubeginn dieser Kontrolle soll der Applikation für alle weiteren Aktionen die Objekt-ID bekannt gemacht werden. Bei Objekten handelt es sich in diesem Kontext um verschiedene Liegenschaften.

Die besagte Stichprobe beinhaltet eine Liste von zu prüfenden Flächen, welche durch eine Identifikationsnummer je Räumlichkeit aufzulisten sind. Der mobile Smartphone-Client sollte an dieser Stelle bereit sein um Raum-Tags (3.2) zu erfassen. Werden in der Stichprobe nicht enthaltene Raum-Tags gescannt, soll das System diesen Umstand erkennen um entsprechend reagieren zu können. Der Qualitätssicherungsprozess soll hierbei nicht beendet werden, sondern es soll auch währenddessen möglich sein Raumdaten und gegebenenfalls Mängel zu erfassen. Nachdem ein Raum-Tag, welcher in der Stichprobe enthalten ist, gescannt wurde, soll die Applikation alle notwendigen Daten zur Prüfung präsentieren. Hierbei soll eine abgewandelte Form des in Tabelle 2.4 gezeigten Prüfformulars zur Qualitätssicherung zum Einsatz kommen. In dieser Eingabemaske sollen die Anzahl der Verunreinigungen (Fehler) je Raumkomponentengruppe eingetragen werden. Auch soll es in diesem Status möglich sein, Bemerkungen jeglicher Art zum geprüften Raum im System zu hinterlegen. Andere in dieser Abbildung dargestellte zu erfassende Daten werden durch das Scannen des Raum-Tags oder durch das Smartphone selbst gehandhabt und an das Backend übermittelt. Nachdem alle Eingaben für die jeweilige Fläche getätigt wurden, soll das System eine Raumauswertung errechnen und darstellen. Diese Auswertung basiert auf dem in Tabelle 2.5 veranschaulichten Auswertungsformular. Die mobile Applikation soll diese Daten automatisch entsprechend den zuvor getätigten Eingaben errechnen und dem Benutzer dessen Resultat bekanntgeben. Dieses Resultat soll in weiterer Folge als einzelnes in die Gesamtbewertung der Stichprobe einfließen. Nachdem ein Raum geprüft wurde, muss der Prüfer den nächsten in der Stichprobe enthaltenen Raum aufsuchen und denselben Prozess nochmals durchlaufen bis alle zu prüfenden Flächen abgearbeitet sind. Wurde das erreicht, soll die bereits erwähnte Gesamtauswertung erfolgen. Hierfür werden jene Räumlichkeiten, die das erforderte Reinigungsergebnis erreicht haben denen die es nicht erreicht haben gegenübergestellt, und die Auswertung soll in Form einer Tabelle und dem klar definierten Gesamtergebnis dargestellt werden.

Für den Anwendungsfall, dass der Objektleiter die NFC-Tag-Infrastruktur bearbeiten bzw. erweitern können soll, sollte die mobile Anwendung nach dem Benutzer-Login entsprechende Funktionalitäten bereitstellen. Für die Programmierung der vom Objekt abhängigen Aktions- und Raum-Tags sollte zuvor durch manuelle Eingabe entschieden werden in welcher Liegenschaft (Objekt) die zu erstellenden Tags eingesetzt werden sollen. Nachdem das Objekt gewählt

wurde, sollte das Benutzerinterface die Möglichkeit bereitstellen, die beiden Aktions-Tags entsprechend der Liegenschaft zu programmieren. Außerdem soll eine Funktionalität freigeschaltet werden, welche die Auswahl der Räumlichkeiten darstellt um auch Raum-Tags entsprechend der Liegenschaft beschreiben zu können. Das System soll in beiden Fällen erkennen, welche Tags noch nicht im System registriert sind und daher noch nicht auf einem realen Tag geschrieben wurden. Nach Abruf und Auswahl der zu programmierenden Tag-Daten soll ein Dialog angezeigt werden, der dem Benutzer vermittelt, dass der Tag nun angehalten und programmiert werden kann. Nach Programmierung soll dieser Dialog automatisch verschwinden und dem Benutzer eine Nachricht über den Status anzeigen.

Analog sollte diese Funktionalität für Personal-Tags einsetzbar sein. Diese sind jedoch nicht von einem Objekt abhängig und sollen daher unabhängig von der Liegenschaft programmierbar sein. Auch hier gilt es, dass das System automatisch bereits programmierte Personal-Tags nicht mehr zur Auswahl stellt. Bei den zu erzeugenden Tags muss beachtet werden, dass der Smartphone-Client diese auch vor unberechtigten Manipulationen schützt, indem der Tag nur durch eine zuvor durchgeführte Authentifizierung beschrieben werden kann. Je nach Tag-Gruppe soll es auch möglich sein den beschriebenen Tag auch als lesegeschützt zu deklarieren. Dies gilt laut Konzept für Personal- und Aktions-Tags.

Leistungserbringer Funktionalität

Leistungserbringer sollen zusätzlich nach dem erforderlichen Login den Beginn der Arbeitszeit erfassen können. Hierfür soll die Applikation auf der Übersichtsseite bereit zum Scannen des ersten Aktions-Tags (Arbeitsbeginn – 3.2) sein. Erst nach dem erfolgreichen Scannen sollen dem Leistungserbringer die für den Geschäftsprozess zur Verfügung gestellten Funktionalitäten freigeschaltet und der erfolgreiche Start der Arbeitszeit auf der Übersichtsseite mit dessen Zeitpunkt vermerkt werden. Zeitgleich mit der Erfassung des Arbeitsbeginns soll der Reinigungsplan mit den abzuarbeitenden Räumlichkeiten bezogen und in einer Liste dargestellt werden. Sollte eine Arbeit noch nicht abgeschlossen worden sein, so muss dies die Applikation erkennen und die noch zu erledigenden Räumlichkeiten entsprechend den bereits abgearbeiteten Positionen zurückliefern.

Zusätzlich zu der abzuarbeitenden Liste an Räumlichkeiten sollen Informationen über mögliche Aufträge zur Nacharbeit angezeigt werden. Letztere sollen vom Benutzer durch Auswahl genauer betrachtet werden können. Hierfür sollen ein Grund des Nichtbestehens des vereinbarten Qualitätsniveaus sowie die betreffende Räumlichkeit und Information über Handlungsbedarf ersichtlich sein. Die abzuarbeitende Liste an Räumlichkeiten soll in solch einem Fall entsprechend erweitert werden.

Der mobile Smartphone-Client muss an dieser Stelle bereit sein, um Raum-Tags (3.2) zu erfassen. Abgesehen davon sollen dem Benutzer weitere Funktionalitäten unabhängig von raumbezogenen Daten bereitstehen. Dies sind einerseits ein Abruf der Betriebsmittel-Checkliste entsprechend der zu prüfenden Raumliste und andererseits die Möglichkeit den Raumbelegungsplan als Ganzes zu betrachten.

Werden in der abzuarbeitenden Liste nicht enthaltene Raum-Tags gescannt, soll das System diesen Umstand erkennen um entsprechend reagieren zu können. Es soll dem Dienstleister dennoch zu jedem Zeitpunkt möglich sein Raumdaten und gegebenenfalls Mängel zu erfas-

sen und um nicht-planmäßige Reinigung zu dokumentieren. Die mobile Anwendung soll durch diese raumgebundenen Aktionen automatisiert ein zeitlich basiertes Protokoll führen. Der Objektleiter soll dadurch in weitere Folge im Online-System Statistiken bezüglich der durchgeführten Dienstleistungen personen- und raumbezogen darstellen können. Die Dokumentation eines Mangels soll raum- bzw. flächengebunden möglich sein. Nachdem ein planmäßig anstehender Raum-Tag gescannt wurde, soll die Applikation alle notwendigen Daten präsentieren um eine elektronische Form des in Tabelle 2.6 gezeigten Auszuges des Leistungsverzeichnisses entsprechend den Raumdaten darstellen zu können. In diesem Status soll es dem Benutzer auch möglich sein einen Mangel zu dokumentieren. Hierfür muss (raumgebunden) eine eigene Funktionalität aufgerufen werden um über Eingabefelder eine Beschreibung des Mangels tätigen zu können. Nachdem Informationen über einen Raum bezogen wurden und ggf. eine Mangelbeschreibung erfasst wurde, soll die mobile Applikation die Daten an das Backend übermitteln.

Die Möglichkeit der Eigenkontrolle soll dem Benutzer durch die Eingabe von Fehlern hinsichtlich der, den Raumgruppenkomponenten zugeordneten, Verunreinigungen nach Tabelle 2.5 möglich sein. Der Dienstleister soll verschiedene Kombinationen von Fehlern probieren können, oder sich selbst einer Bewertung unterziehen um zu überprüfen, ob das vereinbarte Qualitätsniveau erreicht wurde oder nicht. Dies soll in Form einer Tabelle geschehen, welche nach jeder Veränderung den aktuellen Status errechnet und das Raumergebnis präsentiert.

Nach Abarbeitung aller in der Liste stehenden Räumlichkeiten, soll der Mitarbeiter auf der Übersichtsseite einen Arbeitsabschluss durch Scannen des Aktions-Tags für das Arbeitsende tätigen können. Die Applikation wird dadurch angestoßen diesen Zeitpunkt an das Backend zu übermitteln und ist in weiterer Folge dazu bereit erneute Anweisungen entgegenzunehmen.

Abschließend soll nun nochmals in Abb. 3.3 der grobe NFC-Ablauf im Smartphone-Client illustriert werden:

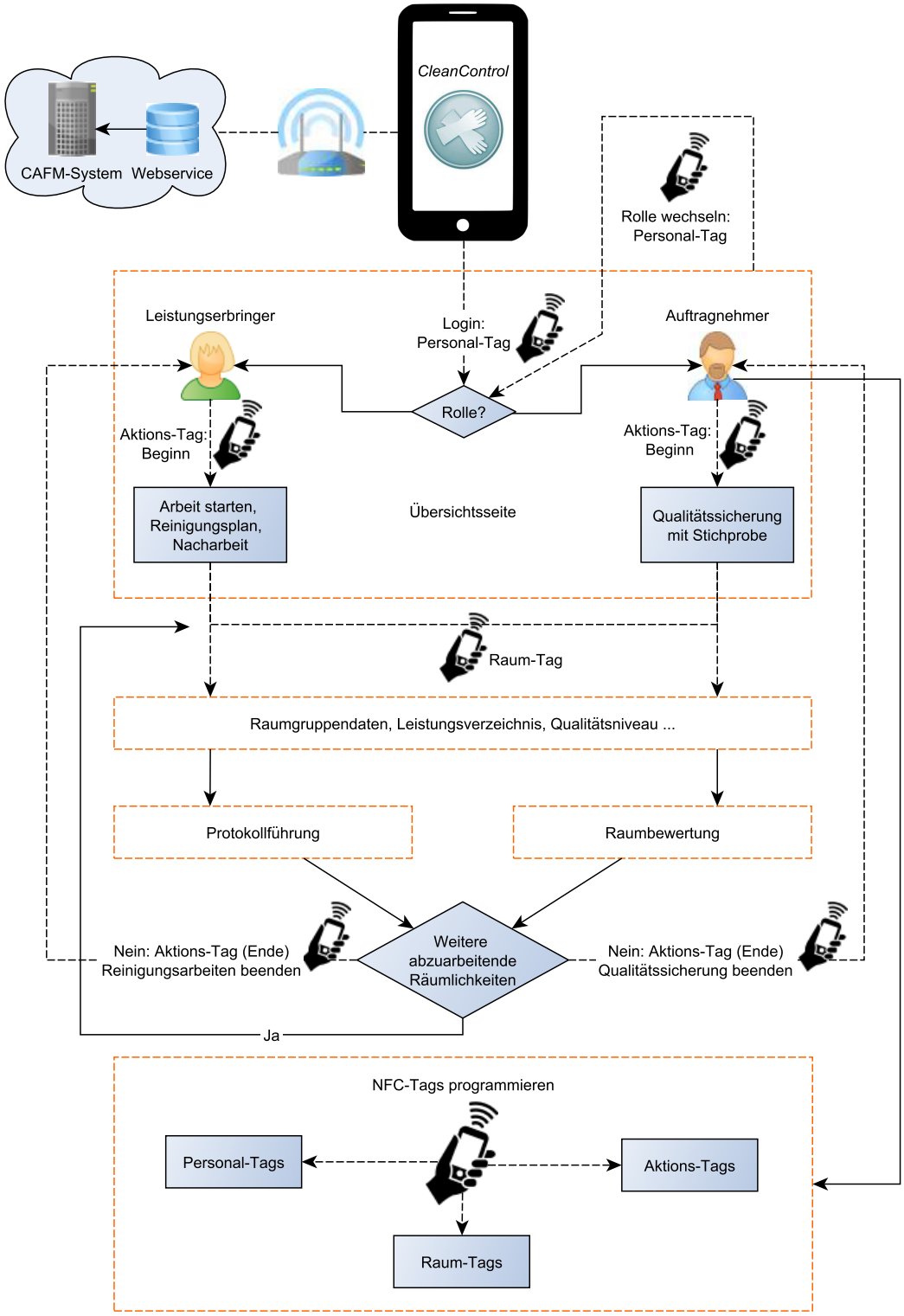


Abbildung 3.3: Ablauf der NFC-Funktionalitäten im Smartphone-Client

Website

Als letzte Systemkomponente sieht die Architektur (3.2) eine Website zur Bewertung oder Bestellung von Reinigung durch Endnutzer vor. Auf dessen Implementierung wird im Kapitel 4 genauer eingegangen. Hinsichtlich der im Abschnitt 2.3 erarbeiteten Funktionalitätenliste soll es die primäre Funktion sein, dem Benutzer in Form einer Eingabemaske eine textuelle Eingabe zu erlauben.

Die Website soll nur sehr eingeschränkten und nur netzwerkinternen (Intranet) Zugriff auf die CAFM-Daten bieten. Sie soll durch einen Parameter, welcher eine Raum-ID repräsentiert, aufrufbar sein. Der Bezug der Daten, und damit die Nutzung der Website, kann auf zwei verschiedene Arten bewerkstelligt werden.

Mobil durch NFC Der notwendige Parameter soll durch einen Raum-Tag (3.2) bezogen werden, was den Einsatz eines NFC-fähigen Smartphones voraussetzt. Nach dem Scannen eines Raum-Tags soll der mobile Webbrowser automatisch geöffnet werden. Die Website soll für eine, dem Endgerät optimierte Darstellung angezeigt werden.

Manuelle Eingabe Durch eine vom Benutzer manuell getätigte Eingabe im Webbrowser soll der Parameter zur Identifikation des Raumes bezogen werden. Dieser wird in weiterer Folge dazu verwendet die betreffenden Informationen abzurufen und darzustellen. Die Variante der manuellen Eingabe kann sowohl im Smartphone-Webbrowser als auch auf stationären Endgeräten benutzt werden.

Daten bezogen auf Raum- und Leistungsverzeichnis sollen in einer schlankeren Form des in Tabelle 2.6 gezeigten Beispiels präsentiert werden. Für den Endnutzer soll klar ersichtlich sein, wann welches vereinbarte Qualitätsniveau erreicht werden soll. Nachdem alle Eingaben getätigt wurden, soll die (mobile) Website die Daten an das Backend übermitteln, und Objektleiter sollen benachrichtigt werden um entsprechenden Handlungsbedarf setzen zu können. Abb. 3.4 veranschaulicht diesen Ablauf.

Abbildung 3.4: Ablauf von Bewertung/Bestellung von Reinigung für Endnutzer mittels NFC

KAPITEL 4

Implementierung

Im Kapitel 3 wurde ein Konzept für einen mobilen Smartphone-Prototyp zur Unterstützung der FM-Prozesse aus Sicht der verschiedenen Rollen (2.2) auf Basis einer evaluierten Funktionalitätenliste (2.3) erarbeitet.

Im nachfolgenden Kapitel soll nun auch dieser Prototyp für das *Android*-Betriebssystem entwickelt werden. Zuvor ist es allerdings notwendig das Vorgehensmodell zur Entwicklung der Software zu beschreiben, im Zuge dessen sogenannte *User Stories* verwendet werden um die Anforderungen des Prototyps aus Sicht der Rollen hinsichtlich den Kriterien für die Implementierung zu beschreiben.

Außerdem wird auf client- und serverseitige Komponenten und deren praktische Implementierung genauer eingegangen. Diese Systemkomponenten aus Abschnitt 3.2 werden hinsichtlich des technischen Aufbaus, der Architektur und der Benutzerschnittstellen genauer beschrieben. Dabei muss auch klargestellt werden, in welchem NFC-Anwendermodus die Applikation arbeiten soll.

Der Begriff der Systemkomponente *Webservice* ist nicht eindeutig definiert. Das im Kapitel 3 erwähnte *Internet of Things* und der damit notwendige Miteinbezug von *Dingen* in das Internet verlangte es, vor der Implementierung zwischen den beiden Architekturstilen *WS-** und *REST* eine Auswahl zu treffen.

Der in diesem Abschnitt resultierende Prototyp soll im Kapitel 5 einem *Usability-Test* unterzogen werden, um dessen Funktionalitäten und Neuheitswert an einem ausgewählten Personenkreis zu erproben und zu untermauern.

4.1 Vorgehensmodell

Die Vorgehensweise zur schrittweisen Implementierung des Smartphone-Prototyps wurde auf Basis eines agilen Entwicklungsmodells definiert, das durch *User Stories* gestützt wird. Sie sollen die Sinnhaftigkeit (*Wer?, Was?* und *Wozu?*) einer Funktionalität zwischen Kunde und Entwicklern in Form von kurzen textlichen Beschreibungen aus Perspektive der Rolle, welche diese

Funktionalität verwenden soll, formulieren. Dabei gelten diese als ein wichtiger Planungsgegenstand bei der Softwareherstellung, welcher anstelle von umfassenden Anforderungsdokumenten verwendet werden kann. Es soll eine schrittweise Abarbeitung der Funktionalitäten im Gegensatz zur traditionellen Vorgehensweise möglich sein und außerdem bei Erstellung von Akzeptanztests helfen. In der Praxis wird auch der Nutzen verfolgt, es Entwicklern zu vereinfachen Abschätzungen betreffend der Implementierungszeit zu treffen [L18, O36].

User Stories werden in agilen Entwicklungsmodellen wie *Extreme Programming* (XP), *Crystal Methods*, *Scrum* oder *Feature Driven Development* (FDD) angewandt und folgen einem Beschreibungsmuster:

> *Als <Rolle> (Wer?)*
> *möchte ich <Funktionalität> (Was?),*
> *damit ich <Nutzen>. (Wozu?)*

In Forschungsarbeiten, welche sich agilen Entwicklungsmodellen und *User Stories* widmen, wurden bereits abgewandelte Vorlagen erarbeitet, um für objektorientierte Applikationen automatisiert Software-Artefakte zu erzeugen [L18] oder *User Stories* mit Code zu verbinden [L5].

Für die in dieser Arbeit formulierten *User Stories* wurde das oben angeführte Muster angewandt und von der Implementierung automatisierter Tests abgesehen, da diese nicht Teil der Forschungsarbeit waren. Das Testen der Funktionalitäten erfolgte inkrementell auf manuelle Weise. Im Zuge der schrittweisen Implementierung entsprechend den im folgenden Abschnitt erarbeiteten *User Stories* mussten parallel auch die Webservice Schnittstellen (4.2) entwickelt werden.

User Stories

Als *Auftragnehmer* möchte ich ...

- ... mich mit einen Personal-Tag am System anmelden können um ggf. nicht abgeschlossene Arbeiten weiterzuführen.
- ... mit einem Aktions-Tag eine Stichprobe zur Qualitätssicherung erzeugen können.
- ... alle Raumreservierungen im Objekt abrufen können.
- ... Raumgruppendaten einer Räumlichkeit (auch einer, die nicht in der Stichprobe angeführt ist) abrufen können.
- ... beim Anzeigen der Raumgruppendaten einer Räumlichkeit Raumreservierungen dieser betrachten können.
- ... beim Anzeigen der Raumgruppendaten das Hinzufügen einer Raumbewertung und eines Defektes erledigen können.
- ... bei einer Raumbewertung Daten hinsichtlich der Qualitätssicherungskriterien sowie eine Nachricht und zusätzliche Informationen bezüglich der Raumkomponenten eingeben und übermitteln können.

- ... zu jeder Zeit die aktuelle Gesamtauswertung des laufenden Qualitätssicherungsprozesses betrachten können.
- ... nach Abarbeitung aller Stichproben im Qualitätssicherungsprozess diesen abschließen.
- ... Personaldaten abrufen können, um die NFC-Infrastruktur durch Erstellung eines zusätzlichen Personal-Tags erweitern zu können.
- ... Raumdaten abrufen können, um die NFC-Infrastruktur durch Erstellung eines zusätzlichen Raum-Tags erweitern zu können.
- ... Objektdaten abrufen können, um die NFC-Infrastruktur durch Erstellung von zusätzlichen Aktions-Tags erweitern zu können.
- ... meine Benutzerdaten bereinigen können (Logout).

Als *Leistungserbringer* möchte ich ...

- ... mich mit einen Personal-Tag am System anmelden können um ggf. nicht abgeschlossene Arbeiten weiterzuführen.
- ... mit einem Aktions-Tag den Reinigungsplan und notwendige Nacharbeiten sowie den Arbeitsbeginn abrufen können.
- ... alle Raumreservierungen im Objekt abrufen können.
- ... alle Betriebsmittel entsprechend den aktuellen Reinigungsplan abrufen können.
- ... Raumgruppendaten einer Räumlichkeit (auch einer, die nicht in der Stichprobe angeführt ist) abrufen können.
- ... beim Anzeigen der Raumgruppendaten einer Räumlichkeit Raumreservierungen dieser betrachten können.
- ... beim Anzeigen der Raumgruppendaten das Hinzufügen eines Protokolls und eines Defektes erledigen können.
- ... bei einer Protokollierung eine Nachricht eingeben und übermitteln können.
- ... beim Anzeigen der Raumgruppendaten die Möglichkeit zur Selbstkontrolle entsprechend den Qualitätssicherungskriterien haben.
- ... wissen, ob und warum Nacharbeiten notwendig sind. Hierzu möchte ich die Eingaben der Qualitätssicherung betrachten können.
- ... nach Abarbeitung aller Räumlichkeiten im Reinigungsplan die Arbeit abschließen können.
- ... meine Benutzerdaten bereinigen können (Logout).

Als *Endnutzer* möchte ich ...

- ... mit einem Raum-Tag raumgruppenspezifische Daten im Smartphone-Browser abrufen können.
- ... im Browser alle Räumlichkeiten des Objektes betrachten können um ein spezifisches auszuwählen.
- ... einen raumbezogenen Kommentar abgeben können, um Reinigung zu bestellen oder zu bewerten.

4.2 Webservice

Im Abschnitt 3.2 wurde bereits das Konzept und die Grundidee für den Webservice vorgestellt. Dort wurde beschrieben, dass es das Ziel dieses Services ist, Daten aus dem CAFM-System zu erhalten und als Schnittstelle für Business-Logiken der Clients zur Hilfestellung bei Prozessabwicklungen zur Verfügung zu stellen. Dabei spielt es keine Rolle, ob diese Daten durch Interaktion mit *Dingen* (im Kontext des IoTs) abgerufen werden, oder ob der Benutzer selbst der Auslöser ist.

Der Webservice wurde absichtlich auf einem einfachen Level gehalten und auch wie im Kapitel zuvor erwähnt, gibt es keine tatsächliche Anbindung an ein (echtes) CAFM-System. Diese Daten werden vielmehr durch den entwickelten Webservice simuliert.

Verwendete Technologien

Als Programmiersprache für den Webservice wurde *Python* [1] in der Version 2.7.6 eingesetzt. Sie ist eine freie Sprache und für die meisten gängigen Betriebssysteme erhältlich. *Python* bietet nicht nur die Möglichkeit sich mit Webservern wie etwa dem der *Apache Foundation* [2] zu verbinden, sondern kann auch mittels des *Web Server Gateway Interface (WSGI)* als eigenständiger Webserver genutzt werden. Diese Schnittstellen-Spezifikation ermöglicht es, die Komponente *Webservice* auf dem besagten einfachen Level zu halten und auch für die prototypische Implementierung von komplexen Webserver-Konfigurationen vorerst Abstand zu halten.

Auch auf Datenbankseite wurde bewusst *SQLite* [3] als eine schlanke Programmbibliothek eingesetzt, welche ein relationales Datenbankmodell in Form einer einzelnen Datei enthält. Für schnell wachsende und stark belastete Datenbanken sollte dieses System für den produktiven Einsatz ersetzt werden. Mittels eines Frameworks für das *Object Relational Mapping (ORM)*, das mit *Python* verwendet wird, ist die Datenbankschicht einfach ersetzbar.

Für die Realisierung des eigenständigen Webservers und der Anbindung an die Datenbank wurde auf in *Python* programmierte Frameworks zurückgegriffen. *Flask* [4] kommuniziert über die bereits erwähnte WSGI-Schnittstelle und bietet dem Programmierer auch Erweiterungen für

[1] `https://www.python.org`
[2] `http://www.apache.org`
[3] `https://sqlite.org`
[4] `http://flask.pocoo.org`

die gängigen Funktionalitäten wie Authentifizierung, Sessions, Internationalisierung etc. Auch bietet Flask eine Erweiterung für das ORM-Framework *SQLAlchemy* [5], mit welchem eine Abstraktionsschicht für Datenbanken erzeugt werden kann, um dynamisch SQL zu generieren. Kompatibilität zu vielen Datenbankmanagementsystemen kann dadurch realisiert werden, und ein besagter möglicher Austausch der Datenbankschicht ist dadurch auf einfache Art und Weise möglich.

Da Sicherheit, Skalierbarkeit und Performance nicht Hauptbestandteil dieser Forschungsarbeit sind, wurde diesen Systemeigenschaften im Gesamten weniger Aufmerksamkeit geschenkt. Dennoch wird im Kapitel 6 im Zusammenhang mit den noch notwendigen Schritten für einen möglichen Produktiveinsatz näher darauf eingegangen. So werden die im Prototyp eingesetzten Technologien im möglichen Szenario eines Produktiveinsatzes kurz einer Kritik unterzogen.

Architektur

Das *World Wide Web Consortium (W3C)* bezeichnet *Webservice* als einen Begriff, welcher viele Dinge auf der Welt bezeichnen kann. Für die Zwecke dieser Arbeitsgruppe, und der unter dem Begriff *Webservice* beschriebenen Architektur, definiert das *W3C* wie folgt [O1]:

> *„A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.“*

Webservices bieten einen Standard für *Machine-to-Machine (M2M)* Kommunikation im Internet. Dabei ist es nicht von Bedeutung, dass diese Kommunikation vom Menschen verstanden werden kann. Interoperable Webservices können auf zwei verschiedenen Herangehensweisen implementiert werden: *WS-** und *RESTful* Webservices. Diese beiden service-orientierten Architekturen wurden schon hinsichtlich Performance Funktionalitäten und in Studien aus Sicht von Entwicklern verglichen und evaluiert [L3].

WS-*

In der *Webservice*-Definition des *W3C* geht es hauptsächlich um Services, welche ihre Funktionalitäten und Schnittstellen in einer *Web Service Description Language (WSDL)* definieren. *Client-Request*- und *Server-Response*-Objekte sind in einem Netzwerkprotokoll namens SOAP strukturiert. SOAP ist ein industrieller Standard, welcher vom *W3C* verabschiedet wurde und ursprünglich für *Simple Object Access Protocol* stand, was seit Version 1.2 nicht mehr gebräuchlich ist. Die WSDL-Datei, welche von Clients verwendet wird, um zu erfahren wie mit dem Webservice kommuniziert werden kann, sowie die tatsächlich zu sendenden Daten basieren auf

[5] `http://www.sqlalchemy.org`

XML und werden für gewöhnlich über HTTP transferiert. Nicht nur die WSDL spielt eine wichtige Rolle, sondern das Modell involviert die drei Aktivitäten *publish*, *find* und *bind*. Dabei können die Services im *Universal Description Discovery and Integration (UDDI)* Verzeichnisdienst, welcher wiederum ein Webservice ist, hinterlegt werden, um sie zu beschreiben.

*WS-** wurde ursprünglich entwickelt, um Interoperabilität für Enterprise-Applications zu erreichen [L3].

Da nativer *WS-** Support für das *Android*-Betriebssystem nicht unterstützt wird und somit auf externe Bibliotheken zurückgegriffen werden müsste, fiel die Entscheidung auf ein rein über HTTP kommunizierendes Programmierparadigma, welches auch von vielen *Web 2.0* Plattformen eingesetzt wird.

REST

Representational State Transfer (REST) ist im Gegensatz zu *WS-** kein Protokoll, sondern vielmehr ein Programmierparadigma, das von *Roy Fielding* in dessen Dissertation [39] definiert wurde.

Kernpunkt der REST-Architektur sind *resources*, welche eindeutig durch *Uniform Resource Identifiers (URIs)* identifiziert sind. Als ein gutes Beispiel einer *RESTful* Implementierung gilt das Web, welches dessen Ressourcen durch URLs identifiziert und HTTP als Serviceinterface verwendet. Ressourcen können dabei durch verschiedene Formate wie HTML oder JSON [6] repräsentiert werden, welche zur Laufzeit vereinbart werden. In einem typischen *REST-Request* im Sinne des Webs wird durch einen Client die URL des Services aufgerufen, um dessen HTML-Inhalt abzurufen. Dabei kann dieser Client durch Angabe einer HTTP-Request-Methode (`GET`, `POST`, `PUT`, `DELETE`) verschiedenen Optionen und einer Payload im vereinbarten Format eine HTTP-Anfrage absenden.

REST bietet Entwicklern im Gegensatz zu SOAP keine maschinenlesbare Beschreibung, sondern verlangt es, die Dokumentation der API in Form von textuellen Beschreibungen offen zu legen. Abb. 4.1 veranschaulicht die SOAP- sowie die REST-Architektur mit deren Komponenten in Form einer Gegenüberstellung.

In etlichen Forschungsarbeiten und Studien im Zusammenhang mit dem IoT hat sich REST als der bevorzugte Architekturstil exponiert, um *Smart Things* in das Internet zu integrieren. Dies ist heutzutage als das *Web of Things* bekannt. REST hat sich dabei nicht nur als einfacher zu erlernen, sondern auch als flexibler, intuitiver und schlanker hinsichtlich der übertragenen Daten herausgestellt [L3].

Für den entwickelten Smartphone-Client wurde ein Webservice auf Basis der REST-Architektur entwickelt. Wie eingangs beschrieben können Ressourcen durch verschiedene Formate repräsentiert werden. Während die API (4.2) eine Schlüsselkomponente des Webservices darstellt, um auf Daten mit *RESTful* Methoden zurückzugreifen, stellt das Format eine ebenso wichtige Rolle dar. Dabei ist die *JavaScript Object Notation (JSON)* im Gegensatz zur *Extensible Markup Language (XML)* weniger wortreich, beinhaltet keine detaillierten Prozessinstruktionen und stellt somit ein *leichtgewichtiges* Format zum Datenaustausch dar. Mit der steigenden Popularität von REST gegenüber SOAP im Zusammenhang mit dem *Internet of Things* wird auch der

[6] `http://www.json.org`

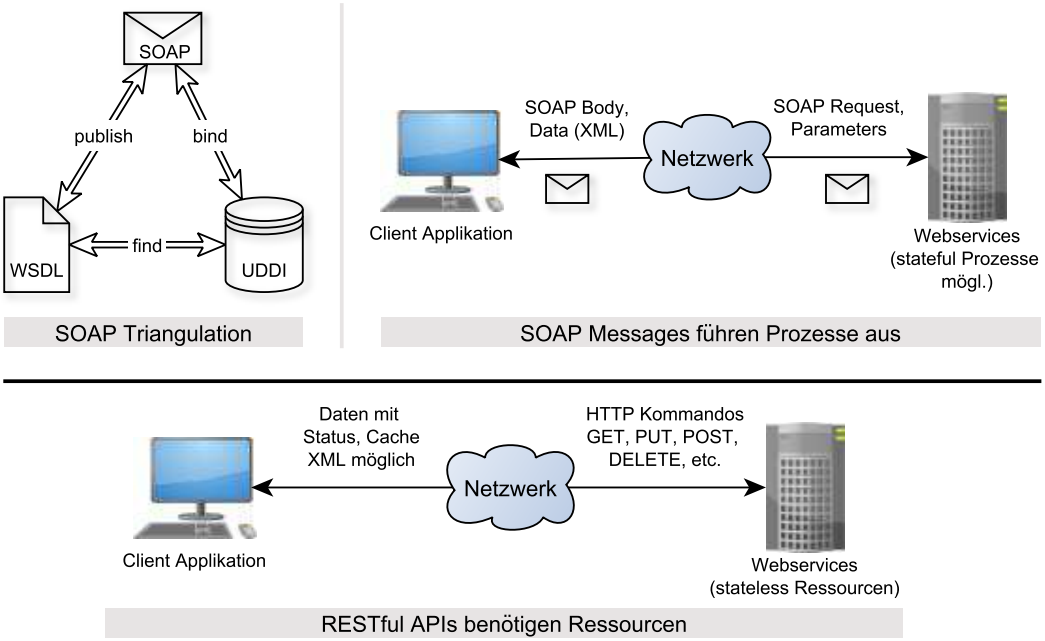


Abbildung 4.1: SOAP vs. REST
Quelle: www.computerworld.com

Einsatz von JSON als bevorzugtes Datenaustauschformat gefördert [O35]. Daten können hiermit zur Übermittlung serialisiert und anschließend am Client auf analoge Weise einer Deserialisierung unterzogen werden, um selbige auf POJOs zur weiteren Verarbeitung im RAM abzubilden.

Das *Android*-Framework stellt außerdem seinen Benutzern eine integrierte Programmbibliothek zur Verarbeitung von JSON-Daten zur Verfügung, welche im Prototyp auch zum Einsatz kommt. Im folgenden Abschnitt wird auf die vom Webservice zur Verfügung gestellte API eingegangen und näher beschrieben.

Webservice API

Eine Liste der Schnittstellen, die vom Smartphone-Client genutzt werden, ist in Tabelle 4.1 ersichtlich. Dabei setzen sich die Endpunkt-URLs so zusammen, dass zu Beginn die Angabe (Alias) einer Datenbank-Entität folgt. Diese Vorgehensweise erleichtert es dem Nutzer der API eine kontextuelle Zuordnung zur auszuführenden Aktion auf eine einfache Weise herzustellen. Auf dessen Basis wird eine Ressource abgerufen, erzeugt oder manipuliert. Außerdem wird durch die Angabe von verschiedenen Arten von URL-Parametern die Rückgabe oder die Manipulation eines eindeutig identifizierten Objektes bewerkstelligt. Sollten diese Informationen nicht ausreichen, können im `POST`-*Body* der Anfrage zusätzliche Parameter, welche im JSON-Format serialisiert werden müssen, angegeben.

Außerdem wurde das HTTP-Protokoll um die Angabe eines UID-Header Feldes erweitert.

Da dieses Header-Feld im Zusammenhang mit der NFC-Funktionalität und der auf Tags hinterlegten UIDs steht, wurde diese Funktionsweise im Abschnitt 4.4 näher beschrieben.

Methode	Endpunkt	JSON-Parameter	Header
GET	/user/<int:user_id>/login	-	uid
GET	/user/programlist	-	-
GET	/user/<int:user_id>/ratings/negative	-	-
			-
GET	/object/<int:object_id>/room/programlist	-	-
GET	/object/<int:object_id>/reservation/list	-	-
			-
GET	/room/<int:room_id>	-	-
POST	/room/<int:room_id>/defect	text	-
GET	/room/<int:room_id>/reservation/list	-	-
GET	/room/<int:room_id>/reservation/list/now	-	-
POST	/room/<int:room_id>/protocol	work_id, message	-
			-
POST	/tag/program	uid, type_id, payload, id	-
			-
POST	/work/begin	user_id, object_id	uid
POST	/work/end	user_id	uid
GET	/work/<int:work_id>/room/list	-	-
GET	/work/<int:work_id>/resources/list	-	-
			-
POST	/control/begin	user_id, object_id	uid
POST	/control/end	user_id	uid
POST	/control/<int:control_id>/rating	room_id, mistakes, remark, negative, positive, passed	
GET	/control/<int:control_id>/room/list	-	-
GET	/control/<int:control_id>/rating/list	-	-
			-
GET	/group/<int:group_id>	-	-

Tabelle 4.1: Implementierte Endpunkte der *Webservice*-Komponente

Als Beispiele folgen eine GET-Anfrage zum Abrufen von Räumen in einem aktiven Qualitätssi-

cherungsprozess auf Seite des Objektleiters, sowie eine `POST`-Anfrage mit näher spezifizierten JSON-Parametern zum Erzeugen eines neuen Raumprotokolls auf Seite eines Leistungserbringers.

```
$ curl -i -X GET http://127.0.0.1:5001/control/1/room/list
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 133
Server: Werkzeug/0.9.4 Python/2.7.6
Date: Wed, 04 Jun 2014 11:18:42 GMT

[
    {
        "group_id": 2,
        "id": 2,
        "name": "Room 2",
        "object_id": 1,
        "size": "45.0"
    }
]

$ curl -i -X POST -H "Content-Type: application/json"
-d '{"work_id":1, "message":"Ein Demo-Protokolleintrag"}'
http://127.0.0.1:5001/room/1/protocol
HTTP/1.0 201 CREATED
Content-Type: application/json
Content-Length: 236
Server: Werkzeug/0.9.4 Python/2.7.6
Date: Wed, 04 Jun 2014 11:37:20 GMT

{
    "message": "Protokoll erfolgreich erstellt.",
    "protocol": {
        "date": "2014-06-04 13:37:20.353000",
        "id": 1,
        "message": "Ein Demo-Protokolleintrag",
        "room_id": 1,
        "work_id": 1
    }
}
```

Listing 4.1: Abrufen einer Raumliste und Erzeugen eines Protokolls am Webservice

Alle Anfragen im implementierten Webservice folgen diesem Muster. Für `GET`-Anfragen werden JSON-Arrays oder JSON-Objekte ggf. mit einer Nachricht, die am Client ausgegeben werden kann, zurückgeliefert. Für erfolgreich durchgeführte Anfragen wird der HTTP-Statuscode `200` vom Server zurückgeliefert. Sollte eine speziell angesprochene Entität oder

ein Endpunkt nicht existieren, so werden entsprechende Statuscodes ungleich `2XX` zurückgeliefert. Beim Erzeugen oder Manipulieren von Daten wird bei Erfolg der Statuscode `201` sowie das betroffene serialisierte Objekt (bei Bedarf mit Zusatzinformationen) zurückgeliefert. Wird durch eine `POST`-Anfrage eine mögliche Inkonsistenz am Server festgestellt – etwa das nochmalige Erstellen eines bereits vorhandenen Protokolls – so wird entsprechend darauf reagiert, der korrekte Statuscode erzeugt und keine Daten verändert. Der Smartphone-Client benutzt diese Informationen, um den Benutzer über Fehler und Erfolg zu informieren.

Auf Serverseite wurde für jeden API-Endpunkt eine Klasse angelegt, welche die jeweils zu verarbeitende Anfrage handhabt. Hierfür wurden die Klasse und der Endpunkt in der *Flask*-API registriert, welche das korrekte HTTP-*Routing* verwalten. Jede API-Klasse folgt diesem Schema, wobei die *Python*-Methoden `get` und `post` entsprechend den zu tätigenden Aktionen implementiert wurden. Als Beispiel folgt ein Ausschnitt der Klasse, welche Anfragen für Defekte eines Raums verwaltet.

```
#API-Endpunkt in Flask registrieren
api.add_resource(room.RoomDefectAPI, '/room/<int:room_id>/defect')

class RoomDefectAPI(Resource):

    defect_fields = {
        'id': fields.Integer,
        'text': fields.String,
        'date': fields.String,
        'room_id': fields.Integer
    }

    @marshal_with(defect_fields)
    def post(self, room_id):
        if not request.json:
                abort(400)

        #Abrufen der POST-Parameter
        args = self.reqparse.parse_args()

        #Existiert der betreffende Raum?
        models.Room.query.get_or_404(room_id)

        #neuen Defekt in die Datenbank eintragen
        defect = models.Defect(text=args['text'],
                               room_id=room_id, date=datetime.now())
        db.session.add(defect)
        db.session.commit()
        return defect, 201        #Defekt erzeugt

    @marshal_with(defect_fields)
    def get(self, room_id):
    #Abrufen von Defekten im Raum
```

```
34          r = models.Room.query.get_or_404(int(room_id))
35          return models.Defect.query.filter_by(room_id=r.id).all()
```

Listing 4.2: Webservice Klasse zum Abrufen und Hinzufügen von Defekten eines Raumes

Datenbank

Der Webservice stellt relevante Funktionalitäten zur Unterstützung der Reinigungsprozesse auf Serverseite zur Verfügung. Auch wurde die Simulation von CAFM-Daten implementiert. In beiden Fällen musste ein Datenbankmodell entworfen werden. Für den Prototyp wurde die CAFM-Datenbank mit der des Webservices vereint und in einem Entitätenmodell (4.2) abgebildet. Als eine der wichtigsten Entitäten im Modell wird eine Räumlichkeit (`Room`) angesehen. Beim Arbeitsbeginn in der Rolle des Leistungserbringers wird der aktuelle Reinigungsplan in Form von einer Raumliste erzeugt. Ähnlich wird auch beim Qualitätssicherungsbeginn in der Rolle des Objektleiters eine Stichprobe ebenfalls in Form einer Raumliste generiert. Jeder Benutzer kann nach Bezug dieser Liste die eigentliche Arbeit beginnen. Der Leistungserbringer erzeugt Raumprotokolle (`Protocol`) und der Objektleiter erzeugt Raumbewertungen (`Ratings`). Auch diese beiden Entitäten sind an die Raumtabelle gebunden. Räumlichkeiten sind außerdem einer Liegenschaft (`Object`) zugeordnet und befinden sich in einer Raumgruppe (`Roomgroup`), welche Informationen zu den Raumkomponenten (`Inventary`), reinigungsassoziierten Dienstleistungen (`Assocservice`) und abweichenden Ergänzungen (`Additionals`) gemäß den im Abschnitt 2.2 vorgestellten Qualitätssicherungsparametern laut Norm enthält. Der Raumgruppe sind auch Betriebsmittel entsprechend den Raumkomponenten zugeordnet, welche vom Leistungserbringer als Liste zur Verfügung gestellt werden kann. Lose an einen Raum gekoppelte Tabellen sind Raumreservierungen (`Reservation`) und Defekte (`Defect`). Letztere können von beiden Rollen erzeugt und auch eingesehen werden.

Eine weitere wichtige Tabelle ist die der Benutzer (`User`). Sie speichert u.a. den Status der Rolle, welcher wichtig ist, um zu entscheiden, welche Funktionalitäten im Smartphone-Client freigeschaltet werden sollen. Im Falle des Objektleiters ist es jene Hauptfunktionalität, die den Beginn von Qualitätskontrollen (`Control`) zulässt. Beim Arbeitsablauf des Leistungserbringers steht der Beginn einer Arbeit (`Work`) mit entsprechenden Funktionalitäten im Vordergrund.

Die beiden Tabellen zur Speicherung von Benutzern und Räumlichkeiten sind jene, welche Daten von real existierenden *Objekten* darstellen und im vorgestellten Konzept (3.2) wesentlicher Bestandteil der NFC-Infrastruktur sind und somit an NFC-Tags gebunden sein müssen. Die drei Tabellen `Usertag`, `Roomtag`, `Tag` implementieren das im Abschnitt 3.1 beschriebene Alter Ego Konzept des zu repräsentierenden *Dings* im IoT -Kontext.

Abschnitt 4.4 beschreibt die Implementierung des Smartphone-Clients, welcher auf die Nutzung dieser Tabellen im Zusammenhang mit der NFC-Funktionalität näher eingehen wird.

Die Erzeugung der Datenbanktabellen wurde in *Python* mit Hilfe von *Flask-SQLAlchemy* realisiert. Für jede erzeugte Tabelle musste eine Klasse entwickelt werden, und ähnlich wie beim Generieren der Tabellen mit SQL, mussten auch in diesem Fall Relationen angegeben werden. Im Falle von *n:m* Beziehungen wurde automatisiert eine weitere Tabelle angelegt. Als Beispiel sei die Beziehung zwischen den Entitäten `Room` und `Control` angegeben. Letztere

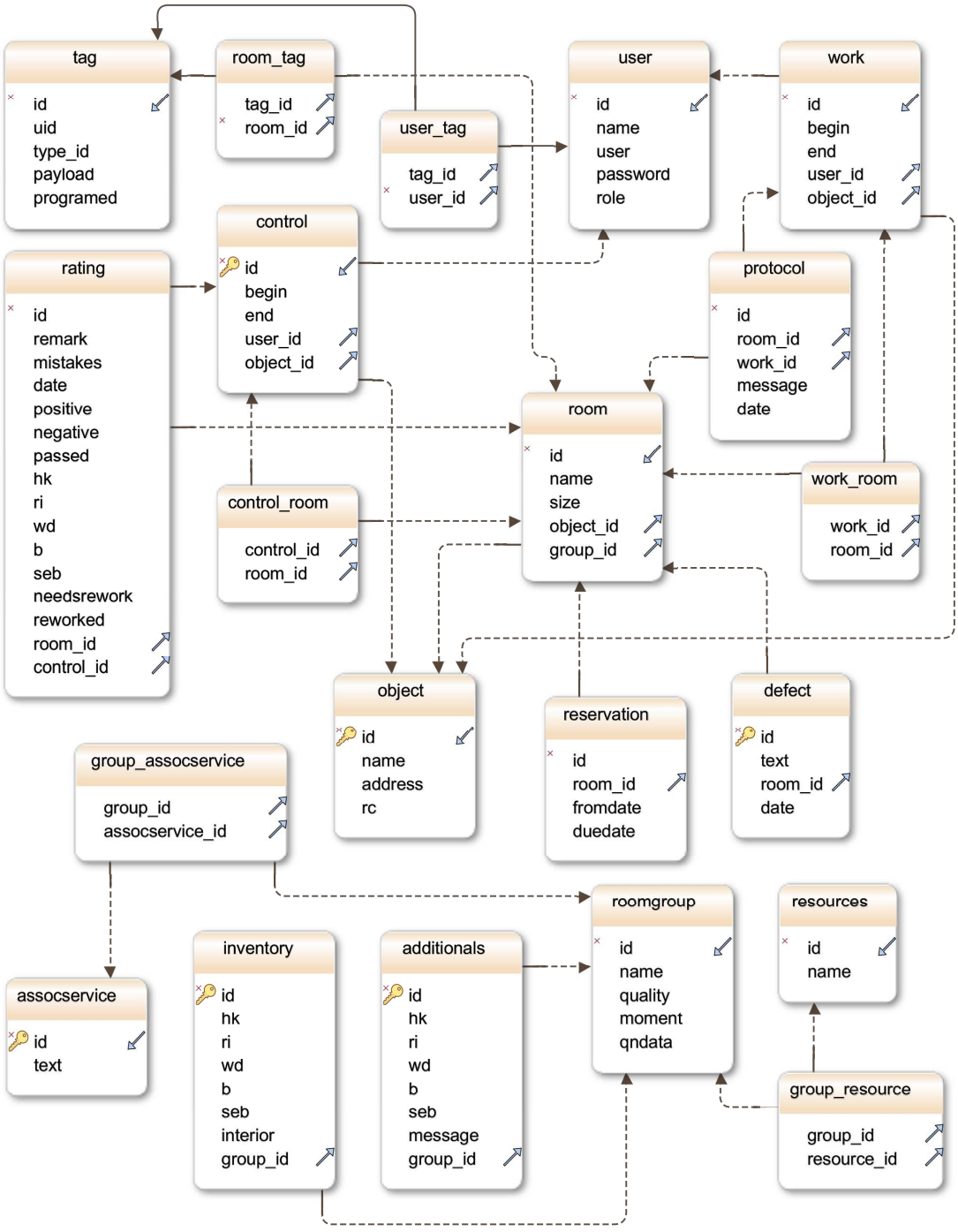


Abbildung 4.2: Webservice Datenbankmodell mit CAFM-DB Integration (DbSchema.com)

Tabelle speichert beim Beginnen einer Qualitätssicherung relevante Daten wie etwa Beginnzeitpunkt, Benutzer- und Objekt-ID. Außerdem wird automatisiert entsprechend dem Objekt, in

welchem die Qualitätssicherung stattfinden soll, eine Stichprobe der zu prüfenden Räumlichkeiten erzeugt. Die daraus resultierenden Daten stehen dadurch in einer *n:m* Beziehung (Tabelle `control_room`).

```
class Room(db.Model):
    id = db.Column(db.Integer, primary_key = True)
    name = db.Column(db.String(64), unique = True)
    size = db.Column(db.Float)
    object_id = db.Column(db.Integer, db.ForeignKey('object.id'))
    group_id = db.Column(db.Integer, db.ForeignKey('roomgroup.id'))

class Control(db.Model):
    id = db.Column(db.Integer, primary_key = True)
    begin = db.Column(db.DateTime)
    end = db.Column(db.DateTime, nullable = True)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'))
    object_id = db.Column(db.Integer, db.ForeignKey('object.id'))

    rooms = db.relationship('Room', secondary=control_room,
    backref=db.backref('control'), lazy='dynamic')

#Relationstabelle
control_room = db.Table('control_room',
    db.Column('control_id', db.Integer,
                                    db.ForeignKey('control.id')),
    db.Column('room_id', db.Integer, db.ForeignKey('room.id'))
)
```

Listing 4.3: n:m-Beziehung zwischen Räumen und Kontrollgängen im Qualitätsicherungsprozess

Bei der Programmierung der Datenbanktabellen sowie dessen Relationen zueinander wurde für alle Entitäten nach diesem Implementierungsschema vorgegangen.

4.3 NFC Infrastruktur

Im Abschnitt 3.2, welcher das Konzept der NFC-Infrastruktur vorstellt, wurden die drei Tag-Gruppen zum Lösungsansatz der Problemstellung dieser Arbeit beschrieben. Außerdem wurde entschieden, dass der Prototyp mit Typ-2 *Mifare Ultralight C* Tags entwickelt werden soll. Auch wurde ein grober Überblick darüber verschafft, welche Daten auf den jeweiligen Tags zu speichern sind. Nachfolgend soll nochmals eine Übersicht darüber gegeben werden:

Personal-Tags Auf diesen Tags soll idealerweise eine Personal-ID hinterlegt werden, welche in weiterer Folge durch das Webservice entgegengenommen wird, um weitere Entscheidungen betreffend Anwendungsfunktionalität in der Applikationsoberfläche treffen zu können.

Raum-Tags Idealerweise soll auf Tags dieser Gruppe eine Identifikationsnummer für die Räumlichkeit hinterlegt werden, welche auch im CAFM-System registriert ist. Diese Identifikation soll gemeinsam mit aktionsspezifischen Parametern an den Webservice zur weiteren Verarbeitung übermittelt werden können. Außerdem ist es von entscheidender Bedeutung, dass auch Endnutzer durch eine hinterlegte URL Daten im Smartphone-Browser abrufen können.

Aktions-Tag Eine Aktions-ID ist notwendig, um zwischen diesen zu unterscheiden und mit dem Webservice entsprechend kommunizieren zu können. Außerdem befinden sich Aktions-Tags in genau einer Liegenschaft, was die Speicherung einer zusätzlichen Objekt-ID notwendig macht.

Dateninhalt der Tag-Gruppen

Jeder Tag-Gruppe wurde eine eindeutige Gruppen-ID zugeordnet, welche am Tag gespeichert werden muss. Mit dieser ID ist es am mobilen Client sowie auf Serverseite möglich zu entscheiden, welche Aktion durchgeführt werden kann. Die Gruppen-IDs wurden in 10er Schritten vergeben womit für mögliche zukünftige Erweiterungen entsprechend dem Tag-Kontext noch Platz wäre. Etwa könnte in Zukunft die Gruppen ID *21* für Personal vergeben werden, welches mit dem NFC-Tag zusätzlich ein hausinternes Schließsystem nutzen könnte. Für zusätzliche Aktions-Tags mit ID *32* wäre ein weiterer möglicher Anwendungsfall Arbeitszeiten zu pausieren und wiederaufzunehmen.

Weiter war es notwendig, einen zusätzlichen Datensatz, in dieser Arbeit als *Payload* bezeichnet, zu speichern. Die *Payload* ist jener Datenhalter, welcher IDs entsprechend der Tag-Gruppe beinhaltet. Im Falle eines Raumes wird dessen eindeutige ID, welche im CAFM-System angelegt ist, gespeichert. Diese ID wird in weiterer Folge genutzt, um raumspezifische Anfragen an den Webservice zu übermitteln. Letzterer weiß dadurch die notwendigen Daten aus dem CAFM-System abzurufen und entsprechende Antworten zu erzeugen. Diese Raum-IDs werden auch als lokale Zwischenspeicher im Smartphone-Client verwendet, um raumabhängige Aktionen durchzuführen, die nach dem Scannen eines Raum-Tags möglich sind, etwa das Erzeugen oder das Verwalten von Defekten, das Einsehen von Reservierungen und im Kontext der jeweiligen Rolle das Übermitteln von Protokollen oder Bewertungen.

Analog zur *Payload* bei Räumen verhält sich die *Payload* bei Personal-Tags. Der Client und in weiterer Folge das CAFM-System kann durch die Identifikationsnummer des Personal-Tags Benutzerdaten abrufen. Ein wichtiges Entscheidungsmerkmal bei diesen Daten ist der Rollenstatus, der am Server hinterlegt ist. Mit diesem Status kann ermittelt werden, welche Funktionalitäten die Applikation bereitstellen soll. Auch dieser Status befindet sich, solange der Benutzer am Client angemeldet ist, im RAM und wird beim Durchführen von rollenspezifischen Aktionen sowie zur Darstellung der Benutzerschnittstelle verwendet.

Ähnlich wie Raum-Tags sind Aktions-Tags an eine Liegenschaft (Objekt) gebunden. Bei Raum-Tags ist dem CAFM-System bereits durch die Raum-ID klar, in welcher Liegenschaft sich der Raum befindet. Deshalb war es notwendig, Aktions-Tags in der *Payload* mit einer Objekt-ID zu bereichern. Beim Arbeitsbeginn bzw. beim Erzeugen einer Stichprobe wird diese Objekt-ID

verwendet, um nur Räumlichkeiten der betreffenden Liegenschaft auszuwählen. Die Vorgehensweise dafür verläuft analog zu den bereits beschriebenen Tag-Gruppen.

Die oben beschriebenen Datenwerte zur Speicherung auf den NFC-Tags sollen durch den Smartphone-Client verarbeitbar und nutzbar sein. Zur praktischen Umsetzung muss das von *Android* bereitgestellte *NFC Tag Dispatch System* (4.4) in Verbindung mit dem NDEF-Format, das im Abschnitt 2.1 beschrieben wurde, genutzt werden. Tabelle 4.2 fasst die notwendigen Daten zusammen, welche auf den jeweiligen Tag-Gruppen hinterlegt werden müssen, um den Smartphone-Client zu implementieren.

Tag ▷ Attribut▽	Raum	Personal	Aktion
Gruppen-ID	10	20	Beginn/Ende (30/31)
Payload	Raum-ID	Personal-ID	Objekt-ID
Zusatz	Raum-URI	-	-

Tabelle 4.2: NFC-Tag Datenspeicherung

Die zu speichernden Daten mussten entsprechend dem NDEF-Format in eine *Message* gekapselt und auf den jeweiligen Tags hinterlegt werden. Für Raum-Tags mussten NDEF-Records mit der *well-known* TNF, `0x01` und dem Payload-Typ „U“ erzeugt werden, um die global zugängliche Raum-URL für Endnutzer zu speichern. Zur internen Verarbeitung mit dem mobilen Smartphone-Client musste zusätzlich die zu verarbeitende Gruppen-ID mitsamt *Payload* in einem weiteren NDEF-Record gespeichert werden. Hierfür wurde ein eigener MIME-Media Typ `x/fmc` definiert, welcher TNF `0x02` besitzt und die *Payload* speichern kann.

Für Personal- und Aktions-Tags reichte jeweils ein NDEF-Record zur einfachen Speicherung einer Payload im besagten MIME-Media Typ (ebenfalls TNF `0x02`). Abschnitt 4.4 beschreibt im Zusammenhang mit dem Webservice und dem Bezug der zu programmierenden Tag-Inhalte, wie diese Inhalte mit dem Smartphone-Client im Rollenmodus für Objektleiter auf die jeweiligen Tags geschrieben werden.

Speichermanagament im gewählten NFC-Tag

Die Typ-2 Tag Plattform basiert auf einem speziellen Speicherchip mit einer bestimmten Speichergröße und einem eigenen Bereich für Benutzerdaten. Die Speicherstruktur (Layout) ist von der Speichergröße abhängig [O13]. Es wird zwischen zwei Layouts unterschieden:

- Static Memory Structure (Speichergröße mit exakt 64 Bytes)
- Dynamic Memory Structure (Speichergröße mehr als 64 Bytes)

Dabei wird die Speicherstruktur in mehrere aufeinanderfolgende von 0 an beginnende nummerierte Blöcke zu je 4 Byte eingeteilt. Diese Blöcke werden wiederum in Sektoren unterteilt, wobei ein Sektor 256 aufeinanderfolgende Blöcke hat (1 KB). Die für den Prototyp verwendeten *Mifare Ultralight C* Tags haben eine dynamische Speicherstruktur (192 Bytes). Sie sind kompatibel mit

der vom NFC-Forum definierten dynamischen Layout-Struktur und verfügen insgesamt über 144 Bytes an Daten zur beliebigen Speicherung. In den Dokumentationen zu *Mifare Ultralight (C)* werden Blöcke im Gegensatz zur NFC-Forum Dokumentation auch als *Pages* bezeichnet [O28]. Auch bei der von *Android* zur Verfügung gestellten `MifareUltralight` Klasse werden Methoden, welche das Arbeiten mit Blöcken bewerkstelligen, als *Page* annotiert. Das Schreiben von *NDEF-Messages* beginnt dabei immer bei der ersten Page, welche den Beginn des Datenblocks kennzeichnet (*Page* 4).

Byte Number	0	1	2	3	Block
UID / Internal	Internal0	Internal1	Internal2	Internal3	0
Serial Number	Internal4	Internal5	Internal6	Internal7	1
Interal / Lock	Internal8	Internal9	Lock0	Lock1	2
CC	CC0	CC1	CC2	CC3	3
Data	Data0	Data1	Data2	Data3	4
Data	Data4	Data5	Data6	Data7	5
Data	Data8	Data9	Data10	Data11	6
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	.
Data	..	..	..	..	n
Lock / Reserved	..	..	..	..	.
Lock / Reserved	..	..	..	..	.
Lock / Reserved	..	..	..	..	k

Tabelle 4.3: Dynamische Speicherstruktur bei Typ-2 Tags [O13]

Tabelle 4.3 visualisiert zusätzlich zur dynamischen Struktur auch den Aufbau der Felder eines Typ-2 Tags. Im Falle einer statischen Layoutstruktur reicht die Blocknummer exakt bis 15 und Lock-Bytes werden nicht verwendet. In Anhang B wird der Speicherinhalt und die Struktur eines fabrikneuen *Mifare Ultralight C* Tags, wie er bei der Umsetzung dieses Prototyps verwendet wird, angeführt.

Datensicherheit im gewählten NFC-Tag

Für den Prototyp war es laut Konzept notwendig, Tag-Inhalte auf eine geschützte Art und Weise zugänglich zu machen. Im Abschnitt 3.2 wurde im Zusammenhang mit der Wahl des Tags für den Prototyp in Tabelle 3.1 zusammengefasst, dass für alle Tag-Gruppen der Inhalt nur mit einem Zugangsschlüssel änderbar sein soll. Für die beiden Tag-Gruppen *Personal* und *Aktion* ist es sogar Anforderung, den Inhalt nur in einem authentifizierten Status bereitzustellen, was

beim öffentlichen Raum-Tag nicht als notwendig erachtet wurde. Hier gilt nur, dass unbefugte Manipulation verhindert werden muss.

> ☞ Alle im weiteren Dokument folgenden Informationen welche im Zusammenhang mit der Tag-Authentifizierung stehen, entstammen nicht aus offiziellen Dokumentationen der Firma *NXP Semicoductors*. Der Bezug dieser Informationen erfolgte ausschließlich in öffentlich zugänglichen und nicht passwortgeschützten Bereichen wie dem des Internetauftritts der *libnfc*-Community [a] oder durch *Reverse Engineering* des Quellcodes der *libfreefare* [b] sowie der in [L9, O9] beschriebenen Informationen. Die besagten Quellen wurden entsprechend als Entlehnung angeführt.
>
> [a] `http://www.libnfc.org/community`
> [b] `https://code.google.com/p/libfreefare/source/browse/libfreefare/mifare_ultralight.c`

Mifare Ultralight C beschreibt in der Dokumentation [7] das *C* in dessen Namen, welches für *cryptography* steht und ein *Admin Security Feature* bereitstellt mit welchem eine Triple-DES (3DES) Authentifizierung am Tag durchgeführt werden kann, um nur im authentifizierten Status Daten zu lesen oder zu schreiben [O28]. Das Security Design der *Mifare Ultralight C* Tags entstand aus dem Bestreben, die Sicherheit zu erhöhen und dadurch auch Limitierungen anderer PICCs zu überwinden [L9].

Der implementierte Algorithmus *ek()* ist eine klassische Zwei-Schlüssel 3DES-Verschlüsselung im *Encryption-Decryption-Encryption* Modus (DESEde). Bei einer 3DES-Authentifizierung ist es notwendig, dass beide Seiten (PICC und PCD) Wissen vom geheimen Schlüssel haben. Das (vereinfachte) Protokoll ist wie folgt illustriert:

```
1. PCD -> PICC: 1A:00
2. PICC -> PCD: AF:ek(RndB)
3. PCD -> PICC: AF:ek(RndA||RndB')
4. PICC -> PCD: 00:ek(RndA')
```

Listing 4.4: Authentifizierungsprotokoll bei *Mifare Ultralight C* Tags nach [L9]

Dabei definiert *ek()* die 3DES-Verschlüsselungsprozedur mit dem Schlüssel *k* und *RndA/RndB* zufällig erzeugte 8 Byte lange Zeichenfolgen.

Im ersten Schritt wird ein Authentifizierungskommando vom PCD an die PICC gesendet. Da auf der PICC der Authentifizierungsschlüssel hinterlegt ist, generiert diese eine zufällige Zeichenfolge *RndB*, welche sie zu *ek(RndB)* verschlüsselt und wieder an den PCD sendet. Dieser generiert ebenfalls eine zufällige Zeichenfolge *RndA* und sendet diese konkateniert mit dem zuvor entschlüsselten rotierten *RndB* zurück an die PICC. Letztere entschlüsselt *ek(RndA||RndB')*

[7] [O29] enthält vertraulich zu behandelnde Informationen und wird von *NXP* nur auf Anfrage mit Unterzeichnung eines NDA (non-disclosure agreement) freigegeben.

zu *RndA||RndB'* und erlangt dadurch das rotierte *RndB'*. Durch Rotieren des von der PICC generierten *RndB* zu *RndB'* können diese auf Gleichheit und somit die Anwendung des selben geteilten Schlüssels verifiziert werden. Die PICC weiß nun außerdem die vom PCD generierten *RndA* und sendet diese rotiert und verschlüsselt an den PCD. Letzterer kann das empfangene *ek(RndA')* entschlüsseln und mit dem selbst generierten rotierten *RndA'* auf Gleichheit überprüfen. Im Falle einer erfolgreichen Authentifizierung können in weiterer Folge Lese- und Schreibkommandos an die PICC übermittelt werden. Die Authentifizierung hält so lange, bis die PICC vom PCD entfernt wurde [L9, O9].

Beim Lesen und Beschreiben von Personal- und Aktions-Tags wird die beschriebene Authentifizierung bei jedem *Kontakt* der PICC mit dem Wechselfeld des PCD probiert. Für Raum-Tags ist keine Authentifizierung notwendig, da Endnutzer die Daten lesen können sollen. Hier war nur mehr die Authentifizierung beim Ändern der Tag-Daten durchzuführen.

4.4 Smartphone-Client

Die Implementierung des Smartphone-Clients folgte nach dem im Abschnitt 3.2 vorgestellten Konzept. Dabei wurde beschrieben, dass dieser Client die Hauptkomponente zur Unterstützung von Reinigungsprozessen und dessen Qualitätssicherung im Gesamtsystems darstellt und den zuvor implementierten Webservice in Verbindung mit der NFC-Infrastruktur als Schnittstelle für Interaktionen mit dem CAFM-System und eigenen Logiken verwendet.

Anders als beim Webservice, welcher auch Funktionalitäten für den Prototyp simulieren soll, handelt es sich hierbei um jene Komponente, welche sich diese Forschungsarbeit im Zusammenspiel mit NFC und dem zugrundeliegenden Gesamtkonzept aus Kapitel 3 als wesentlichen Hauptbestandteil widmet und auf welchen besonderes Hauptaugenmerk gesetzt wird. Der Smartphone-Client ist im Gegensatz zur NFC Infrastruktur und dem Webservice schwerer zu ersetzen, da er keine Middleware darstellt, welche durch eine Abstraktionsschicht ansprechbar ist.

Bei der Gestaltung und für die Umsetzung der *Usability* wurde den Konzepten und Entwurfsprinzipien entsprechend den offiziellen Richtlinien zur Gestaltung von *Android*-Applikationen [O20] Folge geleistet. Dabei wurde so vorgegangen, dass letztendlich ein Smartphone-Client entstand, die Anwendung dennoch jederzeit mit wenig Mehraufwand auf Tablets portiert werden kann.

Als Name der implementierten Applikation wurde *CleanControl* gewählt.

Verwendete Technologien

Der Smartphone-Client wurde für das *Android*-Betriebssystem entwickelt, welches auf einem Linux-Kernel aufbaut. *Android* erweitert mit dessen Klassenbibliotheken die Programmiersprache *JAVA*. In der *Dalvik Virtual Machine (DVM)* wird Software ausgeführt, welche zuvor in JAVA programmiert und für eine *Java Virtual Machine (JVM)* übersetzt wurde. Bevor das gemacht werden kann, übersetzt *Dalvik* mit einem eigens entwickelten Cross-Assembler den vorliegenden Bytecode (Kellerautomatencode) in ein eigenes Bytecode-Format. Dabei wurde die DVM so entworfen, dass sie Registermaschinencode verarbeiten kann und somit ressourcenschonend

und schnell ist. Das ist Voraussetzung, da jede Anwendung und jeder Prozess in einer eigenen DVM läuft. Diese befindet sich jedoch kurz vor ihrer Ablöse durch die *Android Runtime (ART)*, welche mit der aktuellen *Android* Version 4.4 auch testweise verwendet werden kann [O10].

Für die Entwicklung des nativen Prototypen wurde das *Android*-SDK in der API-Version 20 verwendet, welche das *Java*-SDK benötigt. Der *Java Compiler* zum Übersetzen des Quellcodes in JVM-Bytecode wurde in der Version 1.7 verwendet, da Version 1.8 noch nicht von Dalvik unterstützt wird. Bei der Entwicklung der App wurde keine Rücksicht auf Abwärtskompatibilität zu *Android*-Versionen < 4.0 genommen.

Der Prototyp wurde mit einer freien Integrierten Entwicklungsumgebung (IDE) speziell zur Programmierung von *Android*-Applikationen, dem *Android Studio* [8] realisiert. Da diese IDE ein auf *Gradle* [9] basierendes Build-Management-Automatisierungs-Tool verwendet, konnten von der App verwendete Bibliotheken einfach aus einem globalen *Repository* automatisiert und versionskontrolliert bezogen werden. Da *Android Studio* selbst noch in einer Vorabversion (0.8.9) vorliegt, handelt es sich hierbei offiziell um noch keine stabile Version. Dennoch bietet diese IDE seinen Entwicklern Funktionalitäten, welche in den *Android Developer Tools (ADT)* Plugin für Eclipse nicht bzw. nur rudimentär umgesetzt sind. Das auf *IntelliJ IDEA* [10] basierte *Android Studio* ist somit eigens und speziell für die Entwicklung von *Android*-Applikationen entwickelt worden und wird in Zukunft der von *Google* offiziell vorangetriebene IDE zur App-Entwicklung für dessen hauseigenes mobiles Betriebssystem. Die schrittweise Abarbeitung der *User Stories*

Abbildung 4.3: *Android* Emulator

aus Abschnitt 4.1 und die damit verbundene Notwendigkeit, die Applikation zu testen, kann grundsätzlich auf zwei verschiedene Arten bewerkstelligt werden. Einerseits bietet das *Android*-SDK die Möglichkeit auf einem Software-Emulator (Abb. 4.3) zu testen, welcher direkt auf derselben Instanz läuft wie die besagte Entwicklungsumgebung. Die zweite, und für die Ent-

[8] `http://developer.android.com/sdk/installing/studio.html`
[9] `http://www.gradle.org`
[10] `http://www.jetbrains.com/idea`

wicklung des Prototypen notwendige Variante ist jene, die Anwendung auf ein reales Smartphone zu übertragen. Die Notwendigkeit letztere Variante einzusetzen ergab sich zwangsläufig durch die Tatsache, dass das *Android*-SDK noch keine Möglichkeit bietet NFC-Tags softwareseitig zu simulieren. Aufgrund dieser Umstände muss der komplette Testprozess auf einem realen Endgerät stattfinden, was durch das *Android*-SDK problemlos unterstützt wird.

Zur Kommunikation mit der Komponente *Webservice* aus Abschnitt 4.2 und der Abbildung von JSON-Objekten auf POJOs wurden zwei spezielle Bibliotheken eingesetzt. Diese werden im Abschnitt 4.4 genauer beschrieben.

Architektur

Die *Android* GUI-Architektur orientiert sich an einem bekannten Entwurfsmuster, dem *Model View Controller (MVC)*. Dabei ist es das Ziel dieses Musters, einen flexiblen Programmentwurf zu realisieren. Spätere Anpassungen und Erweiterungen oder ein kompletter Austausch einer Komponente sollen dadurch vereinfacht werden, und Wiederverwendbarkeit spielt dabei eine wesentliche Rolle.

Dabei repräsentiert das *Model* die eigentlichen Daten, um welche es sich in der Applikation handelt. Entsprechend den Daten in *CleanControl* sind das jene, die vom Webservice zur Verfügung gestellt werden. Die *View* hingegen ist dafür verantwortlich, um dieses *Model* zu visualisieren, sprich den Nutzern die Daten auf dem *Display* in geeigneter Form zu präsentieren. Die letzte Komponente des MVC-Musters ist der *Controller*, welcher dafür verantwortlich ist, auf externe Aktionen wie Tasteneingaben oder *Touch-Events* zu reagieren [O26].

In den folgenden Abschnitten werden wichtige *Android*-Komponenten beschrieben, und das Schema für den konsequent durchgezogenen technischen Aufbau von *CleanControl* vorgestellt.

App-Komponenten

Eine *View* in *Android* repräsentiert einen darstellbaren Block eines User Interfaces. Sie ist von rechteckiger Form und für dessen visuelle Erscheinung und Ereignis-Handhabung verantwortlich. Es ist die Basisklasse für sogenannte *Widgets*, welche für interaktive UI-Komponenten verwendet werden. *ViewGroups* hingegen sind die Basisklasse für alle Layout-Container, welche *Views* beinhalten können.

Activities sind in *Android* jene Applikationskomponenten, welche einen *Screen* darstellen können, mit welchem der Benutzer interagieren kann. Dabei besitzt jede *Activity* ein *Window*, in welchem das UI bestehend aus *Views* und *ViewGroups* gerendert wird. Typischerweise besitzen *Android*-Applikationen mehrere *Activities*, welche lose zueinander gekoppelt sind. Gleichzeitig ist es natürlich möglich zwischen *Activities* zu wechseln, selbige zu pausieren oder sie zu schließen. Hierfür implementiert das Framework einen sogenannten *Activity-Lifecycle* mit *Callbacks* um auf diese und mehrere Ereignisse reagieren zu können. Im Kontext des MVC-Musters stellen *Activities* die Komponente des *Controllers* dar und sind verantwortlich die *View* zu präsentieren und beinhalten Geschäftslogik.

Ein *Fragment* repräsentiert ein Verhalten oder eine zusammengefasste Menge an UI-Komponenten in einer *Activity*. Mehrere solche Fragmente können mit ihrem eigenen UI einer *Activity* hinzugefügt werden. Das Ziel war es, Inhalte eines Fragments einfach wiederzuverwenden und

das Konzept eines *multi-pane* UI (Abb. 4.4) zu realisieren. Dabei kann ein Fragment als ein Modul einer *Activity* gesehen werden, welches einen eigenen *Lifecycle* besitzt, der wiederum von jenem der *Activity* beeinflusst werden kann. Während die *Activity* läuft, ist es mit Fragmenten möglich, sogenannte *Transactions* durchzuführen, um diese zur Laufzeit zu verändern.

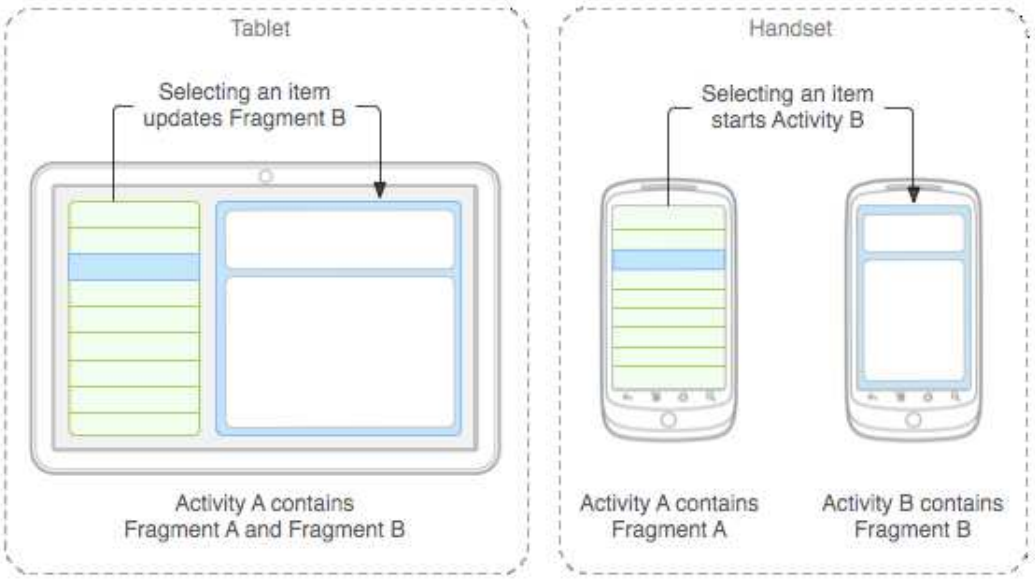


Abbildung 4.4: Zusammenspiel zwischen *Activities* und *Fragments* in *CleanControl*
Quelle: http://developer.android.com

Intents sind Objekte, um andere Applikationskomponenten über gewisse Ereignisse wie etwa *Notifications* zu benachrichtigen. Daten können dadurch abgerufen oder das Ausführen applikationsspezifischer Aktionen kann angestoßen werden. *Intents* können auch Parameter entgegennehmen, um diese Nachrichten für dessen Empfänger genauer zu spezifizieren. Sie sind in *Android* der wesentliche Bestandteil, um Interprozesskommunikation (IPC) zu realisieren.

Bei den angeführten Komponenten handelt es sich nicht um eine vollständige Liste im Framework. Für die Entwicklung von *CleanControl* wurden diese jedoch als die Wichtigsten und als die am häufigsten verwendeten angesehen. Für eine komplette Liste und eine vollständige Beschreibung der im Framework beinhalteten Komponenten soll auf den offiziellen *Guide* [O21] verwiesen werden. Im umgesetzten Smartphone-Prototyp wurde das Zusammenspiel zwischen *Activities* und *Fragments* für jeden dargestellten Screen implementiert. Abb. 4.4 veranschaulicht dieses Zusammenspiel am Beispiel eines Tablets in Gegenüberstellung zur Smartphone-Version.

Webservice Kommunikation

Die Kommunikation mit dem Webservice aus Abschnitt 4.2 wurde mit einer Netzwerk-Bibliothek namens *Volley* realisiert. Diese wird von Google implementiert und gewartet, muss aber als externe Abhängigkeit in das Projekt eingefügt werden, da sie noch nicht Bestandteil des *Android*-Betriebssystems ist. Sie wurde ursprünglich dafür entwickelt, um asynchrone Netzwerkaufrufe

einfacher zu gestalten. Eine der wichtigsten Funktionalitäten ist die der *Request Queues*. So können auf einfache Weise REST-Anfragen in die *Queue* hinzugefügt werden, und die Applikation kann diese nacheinander abarbeiten. Durch *Callbacks* ist es damit möglich, die Antworten des Webservers entsprechend der zu tätigenden Aktion zu behandeln.

Da Antworten des Webservices im JSON-Format generiert werden, war die Verwendung einer weiteren Bibliothek, *Gson*, von Vorteil. Auch dieses Archiv wird von Google als externe Bibliothek zur Verfügung gestellt. Es handelt sich dabei um eine *Library*, welche JSON-Strings auf *Java*-Objekte und umgekehrt abbilden kann. Als Beispiel sei im Folgenden das Abbilden einer Raumliste als JSON-Array in dessen Listenrepräsentation als *Java*-Objekt gegeben.

```
1 public RoomListResponse parseJsonString(String json)
2                                         throws JSONException {
3     Gson gson = new GsonBuilder().create();
4     Type listType = new TypeToken<List<Room>>(){}.getType();
5     List<Room> rooms = (List<Room>) gson.fromJson(json, listType);
6     return new RoomListResponse(rooms);
7 }
```

Listing 4.5: Das Mapping einer JSON-Raumliste auf ein *Java*-Listenobjekt mit *Gson*

Für alle REST Anfragen im implementierten Smartphone-Client wurde das Konzept der *Request Queues*, sowie das Deserialiseren von JSON-Strings in dessen POJO Repräsentation angewandt.

NFC Implementierung

Im entwickelten Smartphone-Prototyp war es notwendig, das im Abschnitt 3.1 vorgestellte Konzept der NFC-Tag-Infrastruktur zu implementieren. Zuvor wurden bereits drei Anwendermodi (2.1) vorgestellt, in welchen NFC verwendet werden kann. Auf Basis des vorliegenden Konzeptes und den darin enthaltenen Funktionalitäten ergab es sich, dass der *Reader/Writer* Modus vorläufig als ausreichend hinsichtlich der Implementierung angesehen werden kann.

Um in *Android* mit NFC arbeiten zu können, müssen zwei Voraussetzungen erfüllt sein. Zum einen ist es notwendig, dass der NFC-Adapter im Smartphone integriert ist und dieser in den systemweiten *Android* Einstellungen aktiviert sein muss, sodass das Betriebssystem NFC-Events verarbeiten kann. Für die Überprüfung des korrekten NFC Status wurde eine Prozedur implementiert, welche im Abschnitt 4.4 genauer beschrieben wird. Für die Verarbeitung von NFC-Ereignissen ist es außerdem notwendig, dass sich das Smartphone im einsperrten Zustand befindet, und die Applikation die Berechtigung zum Verwenden des NFC-Adapters besitzt.

Grundsätzlich ist es nicht nur möglich Daten, als NDEF-Format auf Tags zu schreiben und diese zu lesen, sondern es besteht auch die Möglichkeit direkte, Byte-basierte Kommunikation mit Tags aufzubauen um Daten in einem eigenen Format zu speichern und abzurufen. Für den Prototyp wurde jedoch das NDEF-Format eingesetzt, da dies, wie im Abschnitt 2.1 begründet, wesentliche Vorteile gegenüber RFID hat. Die erwähnte Byte-basierte Kommunikation mit den

NFC-Tags musste dennoch eingesetzt werden. Dieser Sachverhalt wird im Zusammenhang mit der notwendigen Tag-Authentifizierung genauer erörtert.

NFC Tag Dispatch System

Sobald ein NFC-Tag durch das *Android*-Betriebssystem erkannt wurde, werden für den Benutzer folgende Arbeiten erledigt, bevor dieser mit der tatsächlich zu nutzenden Funktionalität in der Applikation fortfahren kann:

1. Tag-Parsing mit Erkennung des MIME-Typ oder URI, welcher die Payload Daten am Tag identifiziert
2. MIME-Typ oder URI wird in einem *Intent* gekapselt
3. *Activity* basierend auf den zuvor generierten *Intent* wird gestartet

Um sich mit dem Smartphone-Client für die gewünschten *Intents* zu registrieren, mussten sogenannte *Intent Filter* verwendet werden. Sollten mehrere Applikationen diesen Filter nutzen, erkennt dies das *Android*-Betriebssystem und erzeugt einen Auswahldialog, in welchem der Benutzer entscheiden kann, welche App den *Intent* verarbeiten soll. Um diesen störenden Dialog weitgehend zu vermeiden, ist es sinnvoll, möglichst spezifische *Intent Filter* zu registrieren, welche die Applikation verarbeiten kann.

Intent Filter

Android bietet drei *Intent Filter*, welche benutzt werden können, um sich für interessierte Tag-Daten beim Betriebssystem zu registrieren. Die Liste ist dabei so sortiert, dass der Filter mit der höchsten Priorität zuerst gelistet wird.

`ACTION_NDEF_DISCOVERED` Wenn ein Tag mit einer gültigen *NDEF-Message* beschrieben wurde, wird beim Scannen ein *Intent* mit NDEF formatierten Daten darin erzeugt und an die *Activity* zur weiteren Verarbeitung geschickt.

`ACTION_TECH_DISCOVERED` Wenn keine *Activity* für den zuvor beschriebenen NDEF Filter registriert ist, oder wenn keine NDEF oder NDEF Daten vorhanden sind, die nicht einem MIME-Typ oder einer URI zugeordnet werden können, wird ein *Intent* erzeugt, der Daten bezüglich der Tag-Technologie beinhaltet. Dieser *Intent* wird anschließend an die *Activity* weitergeleitet.

`ACTION_TAG_DISCOVERED` Sollte kein zuvor beschriebener Filter registriert worden sein, kann mit diesem Filter jegliche Art von Tag zur Verarbeitung an die *Activity* durchgeschleift werden.

Abb. 4.5 visualisiert die Vorgehensweise des *NFC Tag Dispatch Systems* in *Android*.

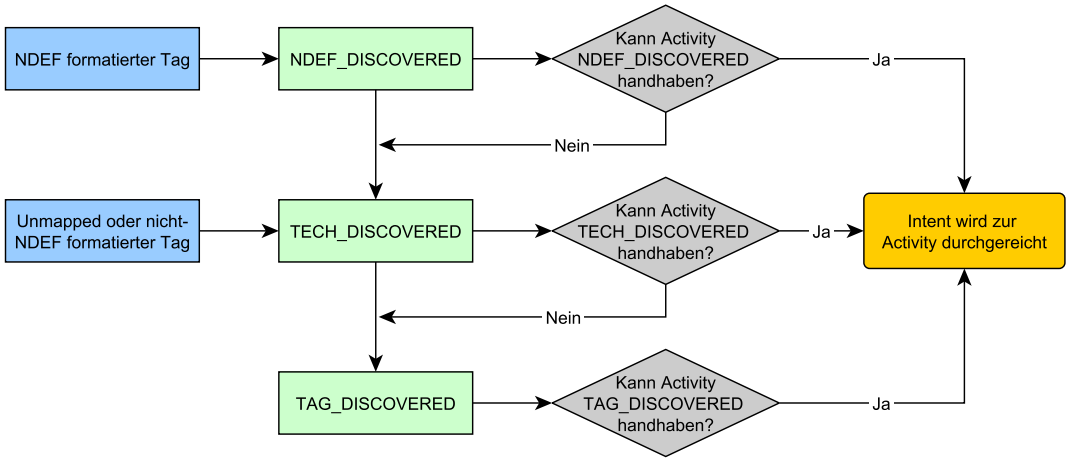


Abbildung 4.5: Tag Dispatch System
Quelle: developer.android.com

Intent Filter und die Problematik der Tag-Authentifizierung

Mit der notwendigen 3DES-Authentifizierung (4.3) beim Arbeiten mit den NFC-Tags kommt es zu einer speziellen Behandlung von *NDEF-Messages* für die Implementierung der Client-App. Da der Dateninhalt von Personal- und Aktions-Tags nur mit einer zuvor gültigen Authentifizierung zugänglich ist, kann das Betriebssystem selbst nicht entscheiden, dass ein *Intent* des Typs `ACTION_NDEF_DISCOVERED` zur *Activity* geschickt werden soll. Das *NFC Tag Dispatch System* weiß keine 3DES-Authentifizierung durchzuführen und kann damit nicht das Format des Dateninhaltes bestimmen. Das Durchschleifen der lesegeschützten *NDEF-Messages* konnte für den Prototyp somit nicht auf Basis eines NDEF-Filters durchgeführt werden. Vielmehr war es notwendig, sich auf die spezielle Technologie der *Mifare Ultralight (C)* Tags zu registrieren und den NDEF-*Container* auf Byte-basierten Kommunikationswegen auszulesen, um in weiterer Folge die Daten auf interne *Android*-Objekte abzubilden. Eine ähnliche Problematik stellte sich beim Wiederbeschreiben der Tags heraus.

Lesen von Raum-Tags

Raum-Tags sind zwar schreibgeschützt, dürfen dennoch mit jedem PCD ausgelesen werden, denn sie beinhalten auch öffentlich lesbare Daten. Bei der Entwicklung des Prototyps musste daher zunächst nur ein NDEF-Filter registriert werden, welcher beim Anzeigen einer NFC-*Activity* im *Foreground Dispatch System* (`NfcAdapter`) aktiviert wurde.

```
1 mAdapter = NfcAdapter.getDefaultAdapter(context);
2 mPendingIntent = PendingIntent.getActivity(
3     context, 0,
4     new Intent(context, context.getClass())
```

```
        .addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP),
    0
);
IntentFilter ndef = new IntentFilter(
    NfcAdapter.ACTION_NDEF_DISCOVERED
);
ndef.addDataScheme("x/fmc");
mFilters = new IntentFilter[]{ndef};
```

Listing 4.6: Intent Filter zum Registrieren auf NDEF-formatierten Tags im *Foreground Dispatch System*

Das Durchschleifen der schreib-, aber nicht lesegeschützten Raum-Tags konnte mit einer gewöhnlichen Registrierung auf `ACTION_NDEF_DISCOVERED` bewerkstelligt werden. Durch die Speicherung einer MIME-Media *NDEF-Message* mit einem eigens definierten MIME-Typ `x/fmc` kann das *NFC Tag Dispatch System* den erzeugten *Intent* sofort an die *Activity* weiterleiten. Die darin enthaltenen Daten konnten ebenfalls auf herkömmliche Art und Weise mit der `Ndef`-Klasse extrahiert werden:

```
public void resolveNdef(Ndef tech) throws Exception {
    tech.connect();           //Verbindung zum Ndef Tag aufbauen
    DataRecord record = null;  //internes Datenhalter-Objekt
    //NDEF-Messages holen
    NdefMessage ndefMsg = tech.getNdefMessage();

    //NdefRecords iterieren, MIME-Record suchen
    for(NdefRecord r : ndefMsg.getRecords()) {
        if(r.getTnf() == NdefRecord.TNF_MIME_MEDIA) {
            //Daten abrufen und aufbereiten
            final String[] mimeData = new String(
                r.getPayload(), "UTF-8").split("/");
            //Daten auf internes Objekt abbilden
            record = new DataRecord(
                Integer.parseInt(mimeData[0]), mimeData[1]);
            break;
        }
    }
    tech.close();
    //Inhaltsspezifische Aktion ausfuehren
    handleNDEFEvent(record);
}
```

Listing 4.7: Lesen von NDEF-formatierten NFC-Tags für Räumlichkeiten

Die eingangs erwähnte Problematik, dass geschützte *Mifare Ultralight C* Tags ohne Authentifizierung dem *Tag Dispatch System* ihr Format nicht preisgeben, musste nun anderweitig gelöst

werden.

Lesen von Personal- und Aktions-Tags

Der beschriebene Ansatz zur Byte-basierten Kommunikation mit ganzheitlich geschützten Personal- und Aktions-Tags verlangte es einen zusätzlichen *Intent Filter* zu registrieren, welcher auf die spezielle Tag Technologie ausgelegt ist, um über die NFC-A (ISO 14443-3A) Implementierung mit dem Tag zu kommunizieren.

```
// ...
IntentFilter tech = new IntentFilter(
    NfcAdapter.ACTION_TECH_DISCOVERED
);
mFilters = new IntentFilter[]{ndef, tech};
mTechLists = new String[1][1];
mTechLists[0] = new String[] {
    MifareUltralight.class.getName(),
    NfcA.class.getName()
};
```

Listing 4.8: Registrierung eines *Intent Filters* im *Foregrund Dispatch System* für *Mifare Ultralight* Tags

Das Aktivieren der *Intent Filter* im *Foreground Dispatch System* wurde somit gesamt wie folgt bewerkstelligt:

```
mAdapter.enableForegroundDispatch(
    context, mPendingIntent, mFilters, mTechLists
);
```

Listing 4.9: Aktivieren des *Forground Dispatch Systems*

Für diese Vorgehensweise musste eine Tech-Liste definiert werden. Das *Android-OS* prüft hierbei, ob der Tag den in der Liste angeführten Technologien entspricht und leitet den erzeugten Tech-*Intent* falls notwendig an die *Activity* weiter. Für die Entwicklung des Smartphone-Clients reichte eine Liste mit `MifareUltralight` und `NfcA` Technologien. Erstere Klasse erweitert dabei die NFC-A Implementierung, um über die besagten Byte-basierten Kommunikationswege Daten zugänglich zu machen.

Bevor nun die hinterlegten NDEF-Daten abrufbar waren, musste eine Tag-Authentifizierung folgen. Die Implementierung dieser folgte dabei dem im Abschnitt 4.3 beschriebenen Authentifizierungsprotokoll. Dabei wurden die Klassen des `javax.crypto` Package zur Ver- und Entschlüsselung mit Konfiguration für 3DES verwendet. Dieses Package ist in *Android* bereits integriert und muss nicht zusätzlich als *Library* geladen werden. Es folgt die Implementierung des Protokolls in *Java* wobei `mfulc` eine Instanz der Klasse `MifareUltralight` darstellt,

welche eine aufrechte Verbindung zum Tag beinhaltet.

```
//Auth Kommando an PICC senden, ek(rndB) aus Antwort extrahieren
byte[] authCmd = new byte[]{(byte) 0x1A, 0x00};
byte[] AF_ekRndB = mfulc.transceive(authCmd);
byte[] ekRndB = Arrays.copyOfRange(AF_ekRndB, 1, AF_ekRndB.length);
//rndB von PICC entschluesseln
byte[] rndB = decrypt(
    ekRndB,
    //Initialisierungsvektoren
    new byte[] {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
    KEY
);
//rndA am lokalen PCD generieren
byte[] rndA = new byte[8];
Random random = new Random();
random.nextBytes(rndA);
//rndB rotieren
byte[] rotRndB = rotateLeftBy8Bits(rndB);
//Verschluesseln von rndA||rotRndB und Aufbereitung fuer PICC
byte[] ek_RndA_rotRndB = encrypt(rndA, rotRndB, ekRndB, KEY);
byte[] AF_ek_RndA_rotRndB = new byte[ek_RndA_rotRndB.length + 1];
AF_ek_RndA_rotRndB[0] = (byte) 0xAF;
System.arraycopy(ek_RndA_rotRndB, 0,
    AF_ek_RndA_rotRndB, 1, ek_RndA_rotRndB.length
);
//an PICC senden und ek(rotRndA) extrahieren
byte[] ZZ_ek_rotRndA = mfulc.transceive(AF_ek_RndA_rotRndB);
byte[] ek_rotRndA = Arrays.copyOfRange(
    ZZ_ek_rotRndA, 1, ZZ_ek_rotRndA.length
);
//Initialisierungsvektoren fuer Entschluesselung
byte[] iv = Arrays.copyOfRange(
    ek_RndA_rotRndB, 8, ek_RndA_rotRndB.length
);
//Entschluesselung und rotiertes rndA beziehen
byte[] rotRndA = decrypt(ek_rotRndA, iv, KEY);
//Erfolg: rotiertes rndA ist intern generiertes rotiertes rndA
boolean authed = Arrays.equals(rotRndA, rotateLeftBy8Bits(rndA));
```

Listing 4.10: *Mifare Ultralight C* 3DES-Authentifizierung

Die beiden Methoden `encrypt` und `decrypt` sind jeweils für die Ver- und Entschlüsselung mittels 3DES implementiert und werden mit den Standard Cipher Transformationen mittels `DESede/CBC/NoPadding` und den aktuellen Initialisierungsvektoren verwendet.

Nach erfolgreicher Tag-Authentifizierung konnte nun die eigentliche Aufgabe, das Abrufen der `NdefMessage`, beginnen. Im Abschnitt 4.4 wurde hierfür die Prozedur `resolveNdef`

beschrieben, welche die `Ndef`-Klasse verwendet, um dessen *NDEF-Messages* zu beziehen. Unglücklicherweise handelt es sich dabei um eine Klasse, welche ein gemeinsames Interface implementiert, jedoch einen Verbindungsaufbau ohne Authentifizierung verwendet und somit die bereits vorhandene Tag-Authentifizierung verloren ginge. Die Prozedur zum manuellen Abrufen der `NdefMessage` wurde wie folgt implementiert:

```
public void resolveMFULC(MifareUltralight tech) throws IOException {
    //Tag Verbindung zu MifareUltralight
    tech.connect();
    //Lese alle Datenpages vom Tag >=4 <=24
    ByteArrayBuffer mReadBuffer = new ByteArrayBuffer(0);
    for(int i=4; i<=24; i+=4) {
        byte[] pageContent = mful.readPages(i);
        mReadBuffer.append(pageContent, 0, pageContent.length);
    }
    //Zwischenspeicher der Daten
    //nach Struktur: 03 MsgLen 1Record 2Record ... nRecord FE
    byte[] rawData = mReadBuffer.buffer();
    //Lese Laenge der NDEF-Message
    int msgLen = rawData[1];
    //Extrahiere NDEF-Message Bytes
    byte[] ndef = Arrays.copyOfRange(rawData, 2, 2+msgLen);
    //Raw NDEF-Message Objekt erzeugen, NdefRecords abrufen
    NdefRecord[] records;
    try {
        NdefMessage m = new NdefMessage(ndef);
        records = m.getRecords();
    } catch (FormatException e) {
        throw new IOException(e.getMessage());
    }
    tech.close();
    //Behandlung der NdefRecords wie in resolveNdef(...);
    //DataRecord abbilden, handleNDEFEvent(record)
}
```

Listing 4.11: Manuelles Abrufen von *NDEF-Messages* bei geschützten *Mifare Ultralight* Tags

Schreiben von Tags

Das Konzept aus Abschnitt 3.1 beinhaltet auf Seite des Auftragnehmers auch die Wartung der NFC-Infrastruktur und somit das Manipulieren von Tag-Inhalten. Da jede Tag-Gruppe als schreibgeschützt konfiguriert sein muss, war es auch hier notwendig, die bereits beschriebene Tag-Authentifizierung zuvor durchzuführen und die Implementierung der *Intent Filter* zu nutzen.

Auf Basis der zu schreibenden Tag-Daten aus Abschnitt 4.3 konnten mit Hilfe der nativen

NDEF-Klassen auf einfache Weise die *NDEF-Messages* erzeugt werden.

```
NdefRecord[] records;
String mimePayload = record.getId()+"/"+record.getPayload();

//MIME-Record fuer alle Tag-Gruppen
NdefRecord mimeRecord = NdefRecord.createMime(
    "x/fmc", mimePayload.getBytes()
);

//URI fuer Raum-Tags
if(record.getId() == Type.ROOM) {
    String url = Config.API_BASE_URL + "r/" + record.getPayload();
    records = new NdefRecord[2];
    records[0] = NdefRecord.createUri(url);
    records[1] = mimeRecord;
} else {
    records = new NdefRecord[1];
    records[0] = mimeRecord;
}
NdefMessage msg = new NdefMessage(records);
```

Listing 4.12: Erzeugen einer *NDEF-Message*, die auf den Tag geschrieben wird

Für das Schreiben der *NDEF-Message* musste aufgrund der aufrechten Tag-Authentifizierung eine eigene Prozedur entwickelt werden, welche die *Message* auf Byte-basiertem Kommunikationsweg auf den Tag schreibt:

```
byte[] rawNdefMessage = new byte[msg.getByteArrayLength()+3];
rawNdefMessage[0] = 0x03;   //NDEF TLV
rawNdefMessage[1] = (byte) msg.getByteArrayLength();
System.arraycopy(
    msg.toByteArray(), 0,
    rawNdefMessage, 2, msg.getByteArrayLength()
);

//Terminator TLV
rawNdefMessage[rawNdefMessage.length-1] = (byte) 0xFE;
int pagesNeeded = rawNdefMessage.length/MifareUltralight.PAGE_SIZE;

for(int pn=0, i=0, pageOffset=4;
            pn<=pagesNeeded;
            i+=MifareUltralight.PAGE_SIZE, pn++, pageOffset++) {
    int from = i;
    int to = i+MifareUltralight.PAGE_SIZE;
    if(to > rawNdefMessage.length) {
        to = rawNdefMessage.length;
```

```
20      }
21      byte[] part = Arrays.copyOfRange(rawNdefMessage, from, to);
22      tech.writePage(
23          pageOffset, Utils.prepareContent(
24              part, MifareUltralight.PAGE_SIZE
25          )
26      );
27  }
```

Listing 4.13: Byte-basiertes Schreiben der *NDEF-Message* auf den Tag

Nach dem erfolgreichen Schreiben konnte der Tag entsprechend der Gruppe gesichert werden. Hierzu war es notwendig die beiden 8 Byte-Keys auf den *Pages* 44 - 47 sowie die entsprechenden Bytes zum Sichern des Tags auf *Page* 43 zu schreiben. Diese Operationen konnten mit der `MifareUltralight`-Instanz und der Methode `writePage()` auf einfache Weise durchgeführt werden.

Besonderheiten im NFC App-Ablauf

Zur besseren Veranschaulichung der Funktionalität und Abläufe im Smartphone-Client werden im Abschnitt 4.4 einige Screenshots präsentiert. Diese sollen hauptsächlich die Prozesse aus Sicht der verschiedenen Rollen visualisieren, um hier genauere Beschreibungen nicht anführen zu müssen. Dennoch werden nun folgende Besonderheiten des Ablaufs im Client betreffend NFC-spezifischer Merkmale und der Verwaltung von Tags in der Komponente *Webservice* beschrieben.

App-Start und NFC-Status

Da nicht jedes *Android* Smartphone NFC unterstützt und auch nicht immer aktiviert ist, war es beim App-Start notwendig zu überprüfen, ob NFC-Hardware vorhanden und diese auch softwareseitig aktiviert ist. *Android* bietet hierfür den `PackageManager`, um die hardwareseitige Präsenz der NFC Funktionalität zu überprüfen. Mit Hilfe der `NfcAdapter`-Klasse konnte anschließend der Status des Adapters abgerufen werden. Hierfür wurde folgende Prozedur beim unmittelbaren App-Start ausgeführt, um den Benutzer über den korrekten Status des Smartphones zu benachrichtigen:

```
1  PackageManager pm = context.getPackageManager();
2  if(!pm.hasSystemFeature(PackageManager.FEATURE_NFC)) {
3      //nfc ist hardwareseitig nicht vorhanden
4  } else {
5      if (NfcAdapter.getDefaultAdapter(context).isEnabled())
6          //korrekter NFC Status
7      else
8          //NFC Hardware vorhanden jedoch nicht aktiviert
9  }
```

Listing 4.14: Ermitteln des NFC-Status des Smartphones beim App-Start

Letztere Abfrage zur Überprüfung des Adapter-Status ist von Bedeutung, da unbeabsichtigte Deaktivierung zu einem kompletten Ausfall der NFC-Funktionalität in der Applikation führt, und schon die Anmeldung mit dem Personal-Tag nicht mehr möglich wäre. Da der Smartphone-Client nicht an die Öffentlichkeit ausgegeben wird und nur zur internen Unterstützung der Reinigungsprozesse eingesetzt werden soll, sollte die Anwendung prinzipiell vom technischen Personal nur auf geeigneten Smartphones installiert werden. Erstere Abfrage zur Überprüfung der hardwareseitigen Präsenz des Adapters sollte ebenfalls nicht vernachlässigt werden, da sich das technische Personal bereits beim App-Start sicher sein kann, dass NFC im korrekten Modus betrieben wird. Mögliche auftretende Fehler können hierbei durch die besagte Validierung der NFC-Funktionalität eingegrenzt werden.

Die Rolle der UID

Im Abschnitt 2.1 wurde der Nutzen und die verschiedenen Arten von *Unique Identification Numbers (UIDs)* beschrieben. Typ-2 Tags implementieren eine *Double Size UID* mit einer Größe von insgesamt 7 Byte. Abschnitt 4.2 beschreibt den Aufbau der Webservice- und CAFM-Datenbank, in welcher u.a. drei Tabellen zur Verwaltung von Tags erzeugt wurden. Die drei Tabellen `Usertag`, `Roomtag`, `Tag` sollen das im Abschnitt 3.1 beschriebene Alter Ego Konzept des zu repräsentierenden *Dings* im IoT-Kontext implementieren.

Der Smartphone-Client ermöglicht es Objektleitern die NFC-Infrastruktur zu erweitern. Für dies war es notwendig, die Tags entsprechend ihrer Gruppe (Personal, Raum, Aktion) zu beschreiben, was im Abschnitt 4.4 erläutert wurde. Bis jetzt gab es noch keine Möglichkeit eine Übersicht der gesamten Tag-Infrastruktur zu erlangen. Ein naheliegender Ansatz war es, die Tags in einem Online-System, dem Webservice, zu verwalten. Da es sich bei der *Double Size UID* um eine vom Hersteller tatsächlich eindeutige Tag-ID handelt kann diese für alle Produktionslinien als eindeutige, nicht änderbare Identifikationsnummer angesehen werden.

Räumlichkeiten und einem Mitarbeiter (Personal) kann jeweils nur ein Tag zugeordnet werden. Mit der Verwaltung der UIDs in der Komponente *Webservice* kann erreicht werden, dass bei Verlust eines Tags, dieser aus der Datenbank entfernt oder als ungültig markiert werden kann. Auch wird durch diese Vorgehensweise die Sicherheit erhöht, denn durch die Bekanntmachung der UIDs im Hintergrundsystem ist auch nach einer möglichen Fälschung eines Tags dieser sofort erkennbar und aus der NFC-Infrastruktur entfernbar und somit jederzeit ersetzbar.

Die Validierung der UID eines Tags im Webservice gestaltet sich als einfach. So wird für die HTTP-Kommunikation mit dem Webservice im Falle einer Anfrage welche, durch Scannen eines Tags generiert wurde, die UID vom Tag ausgelesen und mit der zu behandelnden Anfrage im HTTP-Header mitgesendet. *Android* bietet zum Extrahieren der UID nach Scannen des Tags und Abrufen der Tag-Informationen aus dem übermittelten *Intent* die Methode `getId()`, welche die 7-Byte UID als Array zurückliefert. Diese wurde in dessen lesbare Hex-Repräsentation umgewandelt und als HTTP-Header-Feld an das Backend zur Assoziation mit der jeweiligen Entität (Raum, Personal, Aktion) übermittelt. Bevor nun am Server die eigentliche Anfrage zum

CAFM-System oder zur Webservice-Datenbank gesendet werden kann, wird überprüft, ob ein Tag mit der übermittelten UID im System hinterlegt wurde. Sollte das nicht der Fall sein, so wird ein HTTP-Statuscode `400` (Bad Request) an den Client zurückgesendet und die Anfrage ignoriert.

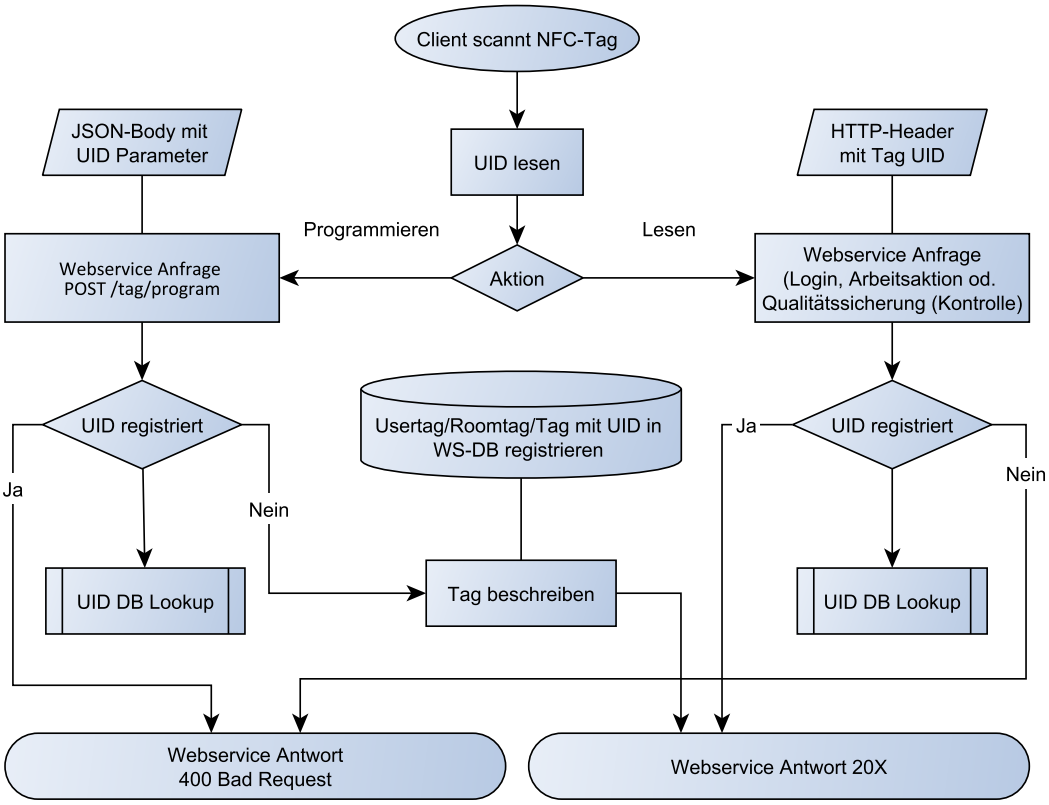


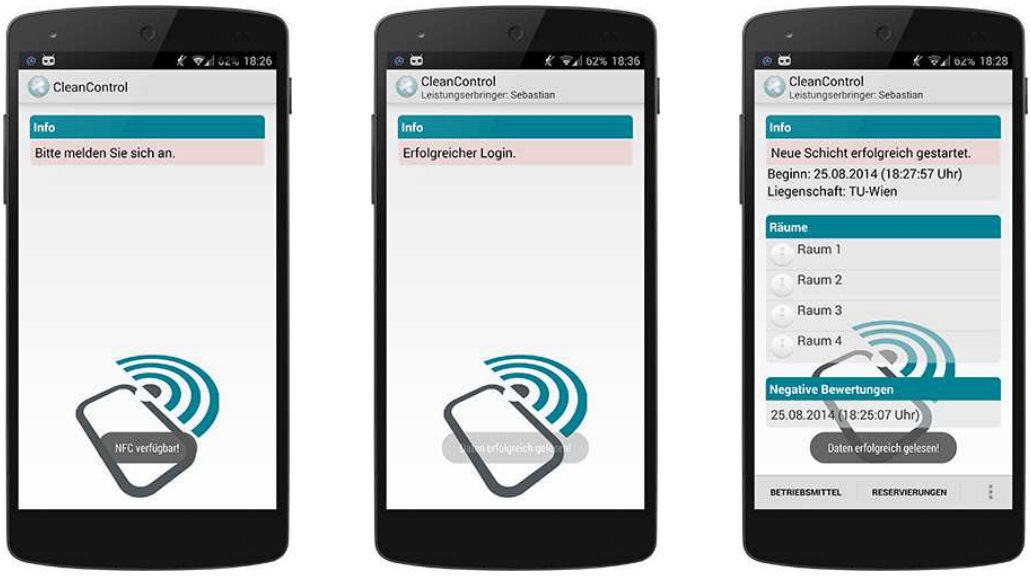
Abbildung 4.6: Die Rolle der UID beim Programmieren und Lesen von NFC-Tags

Für die Registrierung dieser UIDs im Webservice (Tabellen `Usertag`, `Roomtag`, `Tag`) wurde im Zusammenhang mit der Prozedur zum Beschreiben der Tags (4.4) der Webservice Endpunkt `/tag/program` mit dessen `POST`-*Body* eingesetzt, um die UID zu registrieren. Das *Programmieren von Tags* wurde für den Prototyp als das Beschreiben der Tags mit Daten sowie das gleichzeitige Registrieren der UID im Webservice bezeichnet. Dieser Ablauf, sowie das Lesen und Validieren einer UID mit der Datenbank, wird in Abb. 4.6 genauer illustriert.

Benutzerschnittstelle

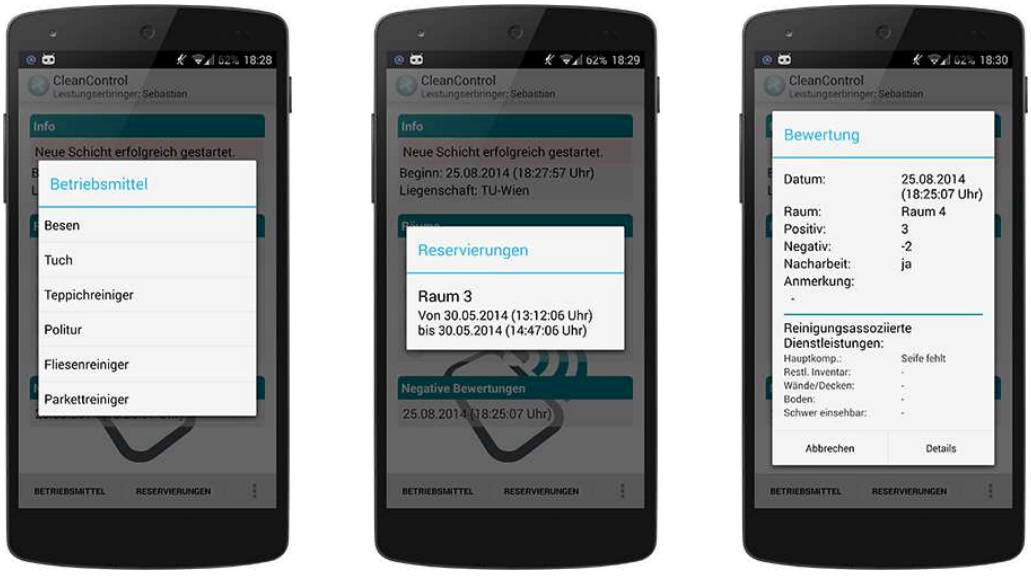
Zur besseren Veranschaulichung der Abläufe werden nachfolgend einige Screenshots der Smartphone Applikation entsprechend den Benutzerrollen kategorisiert präsentiert. Da sich Funktionalitäten überschneiden, wurde darauf verzichtet Abbildungen mehrfach darzustellen.

Leistungserbringer



(a) Startseite (b) erfolgreicher Login (c) Übersichtsseite

Abbildung 4.7: App Screenshots: Anmeldung und Übersichtsseite

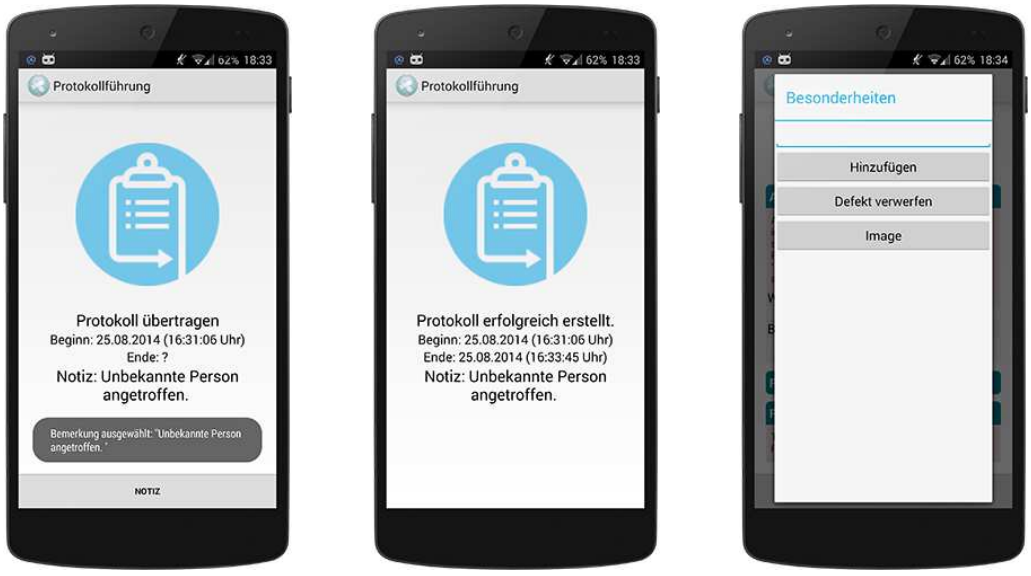


(d) Betriebsmittel (e) Raumreservierungen (f) negative Bewertung

Abbildung 4.8: App Screenshots: Betriebsmittel, Reservierungen und negative Bewertungen

(g) Raumgruppenansicht (h) Selbstkontrolle (i) Protokollnotiz

Abbildung 4.9: App Screenshots: Raumgruppeninformationen, Selbstkontrolle und Protokollnotizen



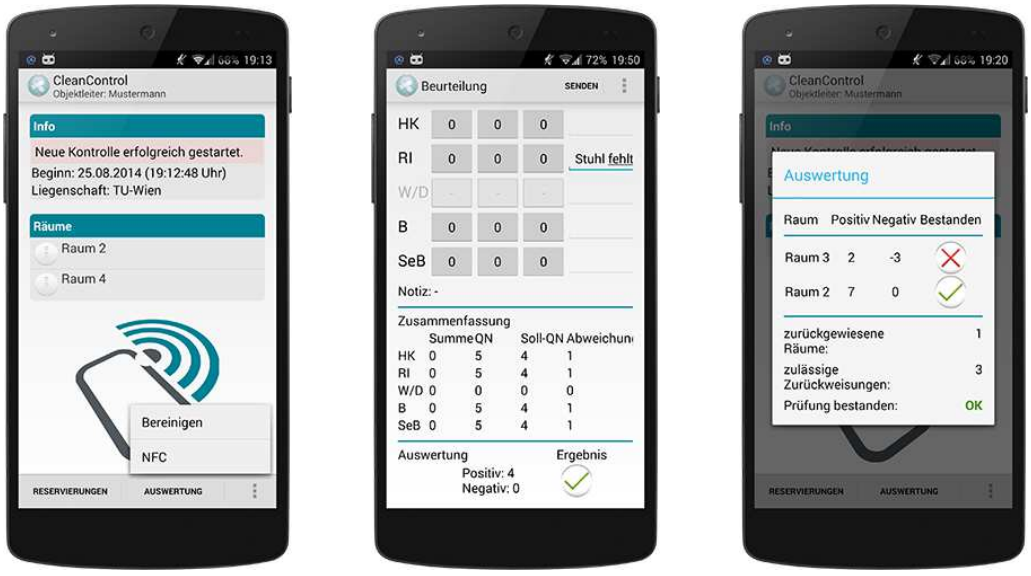
(j) vor Protokollabgabe (k) nach Protokollabgabe (l) Mangelansicht

Abbildung 4.10: App Screenshots: Protokollübermittlung und Mangelmanagement

(m) Schichtende

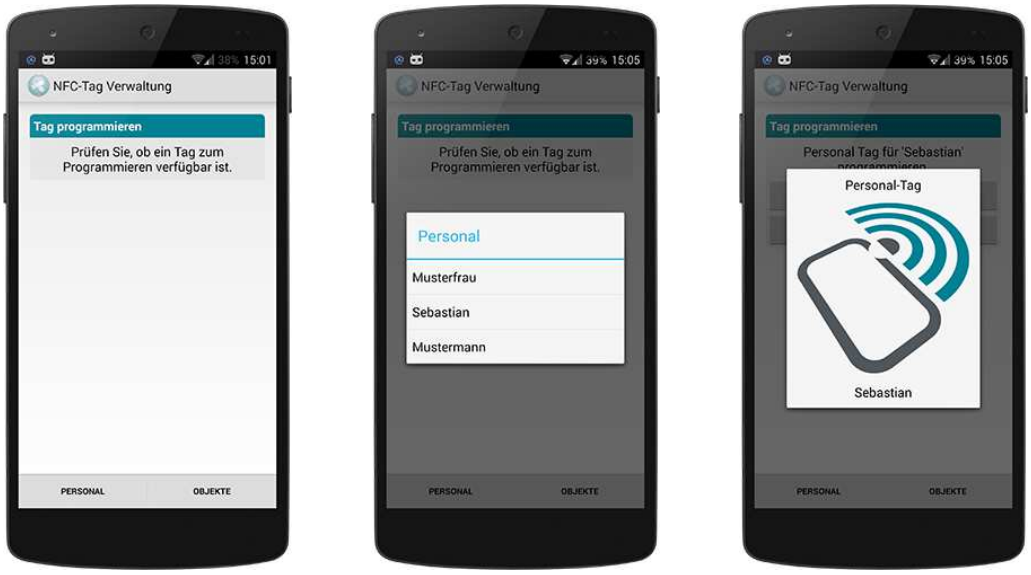
Abbildung 4.11: App Screenshots: Ende der Arbeit

Auftragnehmer



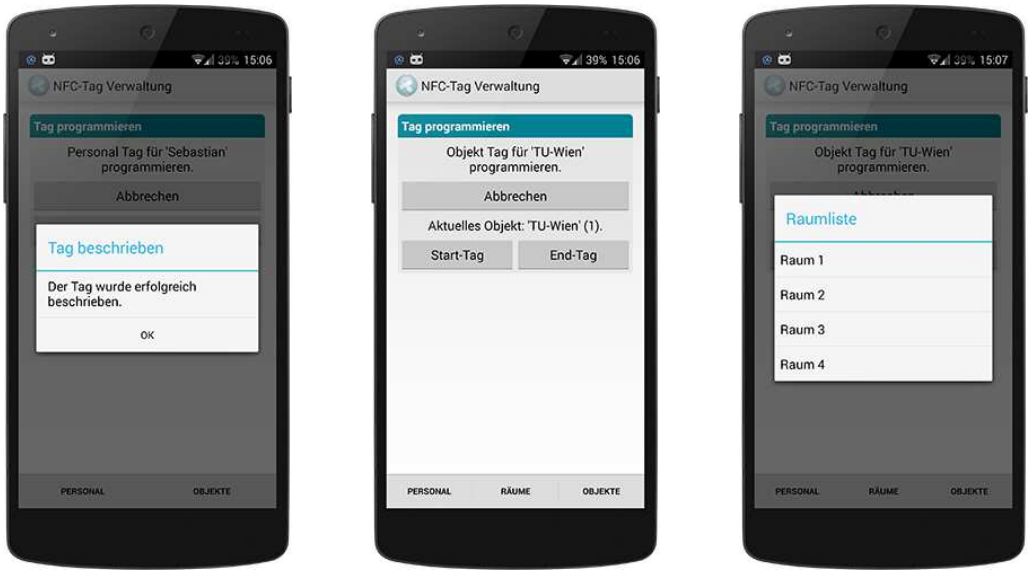
(a) Übersichtsseite (b) Raumbewertung (c) Auswertung

Abbildung 4.12: App Screenshots: Übersichtsseite, Raumbewertung und Auswertung der Qualitätssicherung



(d) NFC-Tag Verwaltung (e) Personal-Tag Liste (f) Programmierbereitschaft

Abbildung 4.13: App Screenshots: NFC-Tag Verwaltung, Personal-Tag Liste, Programmierbereitschaft



(g) nach dem Schreiben (h) Aktions-Tag Verwaltung (i) Raum-Tag Liste

Abbildung 4.14: App Screenshots: Erfolgsmeldung nach d. Schreiben, Aktions-Tags programmieren, Raum-Tag Liste

(j) Objektliste

Abbildung 4.15: App Screenshots: Objektauswahl-Liste

4.5 Website

Die Implementierung der Endnutzer Website folgte dem im Abschnitt 3.2 vorgestellten Konzept. Dabei wurde beschrieben, dass Endnutzer nicht für den direkten Bezug der *CleanControl* vorgesehen sind, sondern zwei mögliche Einstiegspunkte zur Benutzung einer mobilen Website zur Bestellung oder Bewertung von Reinigung in einer Räumlichkeit haben sollen.

Durch den Smartphone-Client und die damit mögliche Programmierung der Raum-Tags, wurde die notwendige Bedingung für den Miteinbezug des Endnutzers bereits realisiert, denn diese Tags beinhalten eine öffentlich zugängliche Schnittstelle in Form einer mit dem NDEF-Format gespeicherten URI (4.3), welche nach dem Scannen eines Raum-Tags die automatische Weiterleitung zum mobilen Webbrowser durch das *NFC Tag Dispatch System* verarbeiten kann. Somit ist es keine Voraussetzung, dass Endnutzer eine spezielle Applikation besitzen müssen, um mit NFC diese am Raum-Tag hinterlegte URL abrufen zu können.

Auch wurde im Konzept der Endnutzer Website beschrieben, dass nicht nur Nutzer mit NFC-fähigen Mobiltelefonen diese Funktionalität nutzen können sollten. Deshalb musste auch ein zweiter, allgemeiner Einstiegspunkt zur Website realisiert werden, welcher nicht von Beginn an das Wissen der betreffenden Räumlichkeit voraussetzt, sondern dem Benutzer eine Auswahl hinsichtlich dem zu kommentierenden Raum in der Liegenschaft bietet.

In den folgenden Abschnitten wird die Implementierung und das Design dieser Website genauer beschrieben.

Verwendete Technologien

Die technische Umsetzung der Website wurde in dieselbe Umgebung wie die des Webservices aus Abschnitt 4.2 integriert. In der dort verwendeten *Flask-API* mussten zusätzliche Endpunkte registriert werden, welche jedoch keine Schnittstellen im Sinne von, durch den Smartphone-Client zu verarbeitenden Anfragen darstellt, sondern dem Benutzer direkt im Browser durch HTML präsentiert werden kann.

Auf Basis einer von *Flask* zur Verfügung gestellten *Template Engine* konnten die zu visualisierenden Inhalte aus der Datenbank des Webservices abgerufen und dem Benutzer zur Interaktion oder zum einfachen Betrachten bereitgestellt werden.

Da bei der Implementierung nach einem gestalterischen und technischen Ansatz zur Erstellung von Websites für Smartphones und Tablets, dem *Responsive Webdesign*, vorgegangen wurde, wurde für die Umsetzung das *Bootstrap Framework* [11], welches *JavaScript* verwendet und auf *CSS* [12] basiert, eingesetzt.

Architektur

Da das Web prinzipiell auf der im Abschnitt 4.2 beschriebenen REST-Architektur aufbaut, war es naheliegend, dass die implementierte Endnutzer Website in den Webservice integriert werden konnte.

Ethan Marcotte etablierte 2010 den Begriff *Responsive Webdesign (RWD)* [13], welcher auf flexiblen, sich an den Bildschirm anpassenden Layouts basiert. Die Idee dahinter war es, eine Website auf großen Bildschirmen und auch auf kleinen *Displays*, wie etwa jenen von Smartphones, in einer einfach zu benutzenden Art und Weise anzuzeigen.

Für den Prototyp wurden einige dieser Konzepte aufgegriffen, jedoch wurde die Website bezüglich ihrem Design und dem Inhalt auch bewusst einfach gehalten, sodass diese nicht in ihrem vollen Maß zur Geltung kommen konnten. Für zukünftige komplexere inhaltliche Erweiterungen ist jedoch mit dieser Umsetzung schon ein Schritt in die von vielen als notwendig erachtete mobile Zukunft gegeben [O32].

Erweiterung der Webservice API

Wie bereits beschrieben, musste der Webservice durch Endpunkte zum Abrufen der Website erweitert werden. Entsprechend den erwähnten Eintrittspunkten mussten die nachfolgenden Endpunkte in *Flask* registriert werden:

Eintrittspunkt manuelle Variante Eine Seite, welche eine Auswahl der Räumlichkeiten bereitstellt, die sich im Objekt befinden. Da diese die Hauptseite darstellt, wurde sie, wie auf Webservern üblich, als `index`-Seite bezeichnet. Diese Angabe ist optional, und somit ergeben sich zwei zusätzliche API-Endpunkte für den manuellen Abruf und den damit notwendigen Ausgangspunkt zur Auswahl einer zu kommentierenden Räumlichkeit.

[11] `http://www.getbootstrap.com`
[12] `http://www.w3.org/Style/CSS`
[13] `http://alistapart.com/article/responsive-web-design`

Eintrittspunkt NFC-Variante Eine Seite mit detaillierten Informationen zu einer Räumlichkeit und deren hinterlegte Daten aus dem Webservice, sowie dem CAFM-System und der Möglichkeit zum Ausfüllen und Absenden eines Formulars.

Für die Registrierung der Endpunkte im *Python*-Framework wurden die entsprechenden Methoden für das korrekte HTTP-*Routing decoriert*:

```
@app.route('/', methods = ['GET', 'POST'])
@app.route('/index', methods = ['GET', 'POST'])
def index():
        ...

@app.route('/r/<int:room_id>', methods = ['GET', 'POST'])
def room(room_id):
        ...
```

Listing 4.15: Registrierung und *Routing* der Website-Endpunkte am Webservice

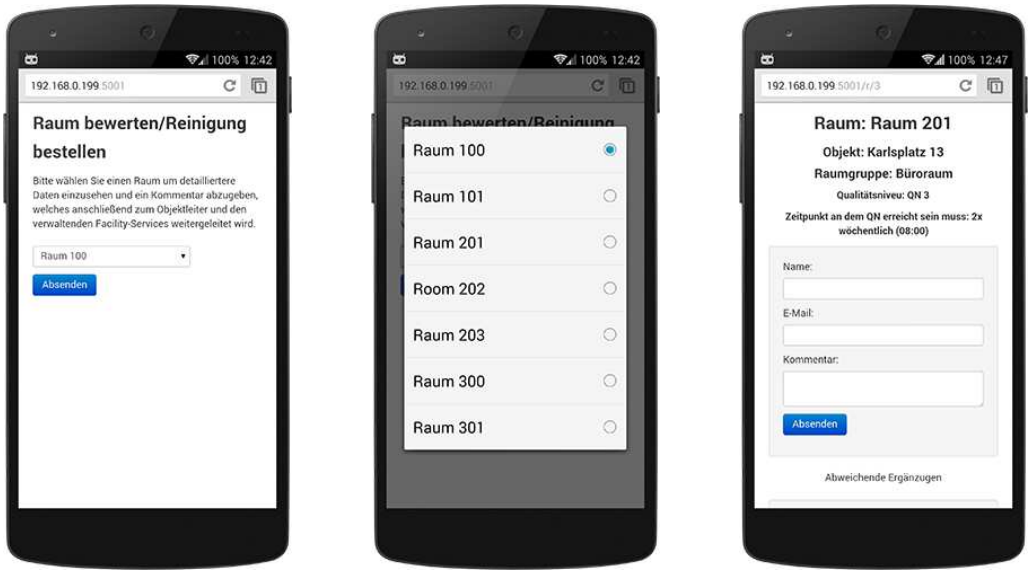
Alle für die Website registrierten Endpunkte mussten `GET`- und `POST`-Anfragen verarbeiten können, da sowohl Daten abgerufen und auch durch Formulare abgesendet werden können müssen. Im Falle der Übermittlung des Kommentars wurde automatisiert eine E-Mail an den im System eingetragenen Administrator (Objektleiter) gesendet.

Funktionsweise der NFC-Variante

Da jeder Raum-Tag öffentlich lesbar ist und eine *NDEF-Message* mit einer URI beinhalten, kann das *NFC Tag Dispatch System* (4.4) diese *Message* ohne Authentifizierung verarbeiten und als einen `ACTION_NDEF_DISCOVERED`-*Intent* (4.4) an das Betriebssystem übermitteln. Dieses kennt alle Applikationen, welche auf URI-*Intents* registriert sind und präsentiert einen Auswahldialog mit Apps, welche die URI verarbeiten können. Sollte der Benutzer eine *default*-App gewählt haben oder das Betriebssystem nur eine Applikation zur Verarbeitung der URI kennen, so wird automatisch jene Applikation aufgerufen, welche die hinterlegte Internetadresse darstellen kann. Für gewöhnlich handelt es sich bei dieser Art von App um einen mobilen Browser wie *Google Chrome* oder *Opera*.

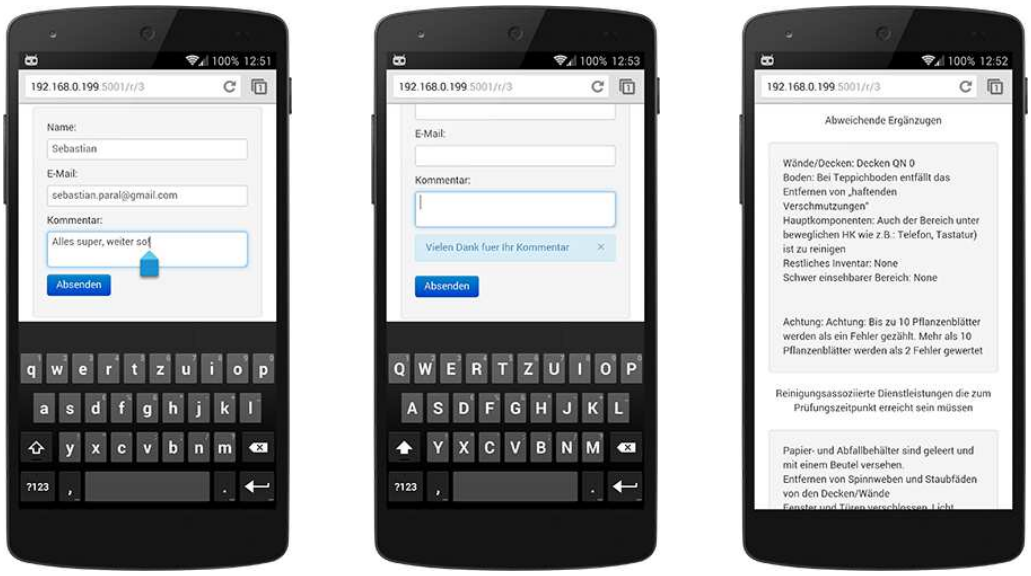
Benutzerschnittstelle

Zur besseren Veranschaulichung der Abläufe werden nachfolgend einige Screenshots der Endnutzer Website präsentiert. Dabei stellt 4.16a den Eintrittspunkt der manuellen Variante und 4.16c den Eintrittspunkt für Endnutzer, welche NFC zum Abrufen der Website verwenden, dar.



(a) Startseite (b) Raumauswahl (c) Eingabemaske

Abbildung 4.16: Screenshots der Endnutzer Website im mobilen Internetbrowser



(d) ausgefülltes Formular (e) abgesendetes Formular (f) Rauminformationen

Abbildung 4.17: Weitere Screenshots der Endnutzer Website

KAPITEL 5

Prototyp Evaluierung

Im Kapitel 4 wurde die Implementierung des Prototyps zum erarbeiteten Konzept (Kapitel 3) praktisch umgesetzt und beschrieben. Die daraus resultierenden Komponenten ergeben das Gesamtsystem, welches in weiterer Folge auf Mängel betreffend der Funktionalitäten und der Benutzerfreundlichkeit (*Usability*) überprüft werden soll. Hierfür soll ein ausgewählter Personenkreis mit dem Prototyp konfrontiert werden um Lösungen von Aufgabenstellungen an, bei der Entwicklung unbeteiligten Teilhabern, zu evaluieren um ggf. in weiteren Schritten Anpassungen an der Software durchführen zu können.

Vor der Durchführung des Usability-Tests muss zunächst die Vorgehensweise definiert werden, um in weiterer Folge die Ergebnisse zu präsentieren. Auf deren Basis können anschließend gewünschte Änderungen oder Erweiterungen der Komponenten beschrieben und implementiert werden.

5.1 Vorgehensweise und Vorbereitung

Der im September 2013 international normierte Standard ISO/IEC/IEEE 29119 für *Software Testen* löste bis dato bekannte Normen aus demselben Bereich ab, welche als Basis für den neuen Standard galten. Das Ziel war es, einen endgültigen international vereinbarten Standard zur Prüfung von Software zur Verfügung zu stellen, welcher innerhalb eines jeden möglichen Softwareentwicklungs-Lebenszyklus und jeder Organisation verwendet werden kann [O30].

Die neue Norm gliedert sich in fünf Teile, wobei zum aktuellen Zeitpunkt (Mitte 2014) erst die ersten drei in ihrer Endfassung veröffentlicht wurden. Teil 4, welcher aktuell als *Draft* zugänglich ist, beschreibt u.a. Testtechniken, worunter auch der Nutzen eines *Usability-Tests* angeführt ist [NS3]:

> „*The purpose of usability testing is to evaluate whether specified users can use the test item to achieve assigned goals with effectiveness, efficiency and satisfaction in specified contexts of use.*“

Als „*specified contexts of use*“ können nicht nur der Kontext der Funktionalität, sondern auch die verschiedenen Rollen (2.2) interpretiert werden, welche den Smartphone-Client und in weiterer Folge das Gesamtsystem nutzen sollen. Der Usability-Test setzt seinen Fokus auf das Zusammenspiel der Software mit dem Benutzer und klares Hauptaugenmerk auf dessen Zufriedenstellung im Belangen der Interaktion mit der Benutzerschnittstelle.

Der bekannte Schriftsteller und Usability-Experte *Jakob Nielsen* publizierte im Jahr 2000 einen Artikel, in welchem er die Resultate seiner Studie zur Bestimmung der Anzahl von Testprobanden begründete. Dabei schrieb er, dass aufwändige Usability-Tests eine „*Verschwendung von Ressourcen*“ sei und „*die besten Ergebnisse mit einer Prüfung von nicht mehr als fünf Benutzer erzielt werden können*“. Mit diesen Probanden sollen aber „*so viele kleine Tests wie möglich*“, durchgeführt werden [O24].

Usability-Test Methoden

Für die praktische Umsetzung des Tests wurden zwei Methoden zur Messung der *Usability* angewandt, welche nachfolgend erläutert werden.

Usability-Test durch Aufgabenanalyse

Bevor die Implementierung des Systems begonnen wurde, wurde im Abschnitt 4.1 das Vorgehensmodell beschrieben. Dieses beinhaltet letztendlich *User Stories* als Resultat zur Beschreibung der Sinnhaftigkeit einer Funktionalität. Auf Basis dieser User Stories und der zuvor im Konzept erarbeiteten Funktionalitätenliste aus Abschnitt 2.3 konnten in weiterer Folge exakte Aufgabenstellungen definiert werden. Dabei war es das Ziel, mögliche Modelle als Grundlage zur Darstellung eines nutzerzentrierten Entwicklungsprozesses zu ermitteln. Hierzu wurden die Aufgabenbereiche erfasst und daraus theoretische Modelle über die angenommenen Qualifikationen und Verhaltensmuster der Zielgruppen abgeleitet und sogenannte *Use Cases* erstellt [O5]. Diese dienten letztendlich als Vorlage zum Usability-Test mit den Probanden.

Usability-Test im Labor

Zur Durchführung eines szenariobasierten Tests mit einer möglichst zielgruppenspezifischen Auswahl von Nutzern wurde in einer Laborumgebung möglichst realitätsnah getestet. So wurde nicht starr in einem Raum verweilt solange der Test aktiv war, sondern dieser auf mehrere Räumlichkeiten und Gänge verteilt. Sowohl die Feststellung von Schwachstellen und Verbesserungsmöglichkeiten, als auch das Testen der Anwendung auf die Erreichung bestimmter Ziele durch die Nutzer, konnten erhoben werden [O5].

Die Probanden wurden beim aktiven Testen und Lösen der Aufgabenstellungen beobachtet und konnten ggf. zeitgleich auch befragt werden. Außerdem wurden die Testpersonen dazu angehalten, ihre Empfindungen und Gedanken zur Anwendung auszudrücken, was sich als *Thinking-aloud-Methode* etabliert hat. Nachdem der praktische Test beendet wurde, wurden die Testpersonen gebeten ihren Gesamteindruck widerzuspiegeln und einen Fragebogen (Anhang A.2) zu beantworten.

Identifikation der Aufgaben

Die Erstellung der Aufgabenliste (Testskript), welche durch die Probanden abgearbeitet und gelöst werden sollte, erfolgte für jede Aufgabe in vier Schritten:

1. Aufgabenbezeichnung
2. Vorbedingung
3. Erfolgsbeschreibung
4. Textuelle Beschreibung

Dabei ergaben sich je Rolle insgesamt drei voneinander getrennte Listen an Aufgabenstellungen, welche in Anhang A komplett angeführt sind. Die zu lösenden Aufgaben sollten die komplette Funktionalität des zu bedienenden Systems abdecken.

5.2 Testausführung

Da nicht jeder Proband mit der NFC-Funktionalität vertraut war, wurde vor Testbeginn eine Einführung in die Bedienung gegeben. Im Zuge dessen wurde auch der Nutzen des Prototyps sowie die verschiedenen Rollen, welche das System anwenden sollen, klargestellt. Die Testpersonen wurden anschließend in ihre Funktion eingeteilt und wussten nachfolgend ihre Aufgabenbereiche. Auch wurden die benötigten Materialien wie Testsmartphones, Notebooks und NFC-Tags vorgestellt und klargestellt, dass der Test pro Person nicht länger als eine halbe Stunde dauern wird. Die Testumgebung wurde noch vor Beginn vom Testleiter in einen initialen Anfangsstatus versetzt, sodass notwendige NFC-Tags im System registriert waren, um diese auch nutzen zu können. Personal-Tags wurden den Probanden ausgehändigt, fix montierte NFC-Tags könnten durch ihre Klebeeigenschaften an den vorgesehen Stellen testweise angebracht werden. Entsprechend der Rolle der Testperson wurde das Testskript mit den zu lösenden Aufgabenstellungen ausgehändigt, und der Test konnte beginnen.

5.3 Testergebnisse

Im Mittelpunkt dieser Forschungsarbeit stand ein neuartiges Konzept zur Unterstützung der Reinigungs-, Qualitätssicherungs- und Bewertungsprozesse aus Sicht dreier Rollen. Dieses und der dadurch entstandene Prototyp mit Verwendung der NFC-Technologie sollten durch die zuvor beschriebene Vorgehensweise des Usability-Tests evaluiert werden. Die Ergebnisse gruppieren sich daher einerseits in das beobachtete Verhalten der Bedienung des Smartphone-Systems und andererseits ist die Gesamtsicht der Probanden zum erarbeiteten Konzept mit Unterstützung der NFC-Technologie ausschlaggebend um, in weiterer Folge Rückschlüsse auf eine optimale Bedienbarkeit zu ziehen. Folglich wurden die Meinungen und Vorschläge der Probanden einer Diskussion unterzogen und mögliche Lösungsansätze zur Verbesserung des Konzeptes und des Prototyps beschrieben.

Ergebnisse zum Prototyp

Grundsätzlich konnten die Aufgabenstellungen, mit denen die Probanden konfrontiert wurden, allesamt in einen annehmbaren Zeitrahmen gelöst werden. Die Menüführung der Applikation wurde von allen Personen als „intuitiv“ bezeichnet, da es sich dabei um jene handelt, die in den *Android-Guidelines* empfohlen wird und dadurch schon aus anderen Applikationen bekannt war.

Beim erstmaligen Nutzen des Personal-Tags zum Anmelden am System mussten jene Probanden, welche noch nicht mit der Technologie konfrontiert waren, zuerst die richtige Position zum Lesen der Tags herausfinden. Dieses etwas unvertraute Verhalten mit der Technologie konnte jedoch rasch und nach wenigen weiteren Tag-Scans nicht mehr beobachtet werden.

Es werden nun folgend einige Problematiken beschrieben, die bei der Bedienung und der Auswertung des Fragebogens betreffend des Prototyps aus Anhang A.2 aufgefallen sind.

paused-State Problematik

Bei allen Testpersonen wurde beobachtet, dass oftmals durch unbeabsichtigtes Drücken des *Home-Buttons* am Smartphone die Applikation in den Hintergrund und in den sogenannten *paused-State* versetzt wurde. Diese Tatsache fanden die Probanden allesamt als störend, da die Applikation manuell ausgewählt werden musste, um sie weiter bedienen zu können. Diese Problematik konnte jedoch nur beim Scannen von Raum-Tags beobachtet werden, da die Benutzer das Smartphone währenddessen sie die tatsächlich zu erledigende Arbeit erbrachten in eine Gürteltasche oder die Hosentasche steckten. Beim Verstauen des Gerätes in der Tasche wurde der *Home-Button* unbeabsichtigt gedrückt und die Applikation in den Hintergrund geschoben und anschließend automatisch gesperrt. Dieses Verhalten führte einerseits dazu, dass durch das Wiederaufrufen der Applikation aus dem Hintergrund mehr Zeit benötigt und in weiterer Folge das Smartphone als solches etwas unsicherer bedient wurde. Das Hantieren mit Personal- und Aktions-Tags brachte diese Problematik nicht zum Vorschein.

Die Lösung dieser Problematik könnte auf zwei verschiedene Arten stattfinden:

NFC Nachdem die beschriebene Problematik beim Scannen von Raum-Tags zu beobachten war, ergab sich ein Lösungsansatz, welcher mit NFC implementiert werden konnte. Sollte das Smartphone die Applikation in den besagten *paused-State* versetzen, könnte durch Scannen eines Raum-Tags außerhalb der Applikation diese automatisch weitergeführt werden und zeitgleich auch den Raum-Tag entsprechend behandeln. Die eigentliche Problematik kann hierdurch nicht verhindert werden, es ergibt sich jedoch ein Lösungsansatz, um den Benutzer zusätzliche Arbeit zu ersparen sofern dieser Status eintritt. Dieser Lösungsansatz müsste den Probanden jedoch zuvor beschrieben und erklärt werden, da dieser nicht als intuitiv angesehen werden kann.

Home-App *Android* bietet die Möglichkeit eine Applikation als sogenannte *Home-App* zu deklarieren. Dies ermöglicht es, nach dem Booten automatisch diese Applikation zu starten und auch in dieser zu bleiben. Wird in weiterer Folge der *Home-Button* gedrückt, so erkennt dies das Betriebssystem und die aktive Anwendung (*Home-App*) bleibt im Vordergrund. Diese Vorgehensweise wird auch oft dazu verwendet, um Benutzer davon abzuhalten andere Applikationen als die gewollte zu verwenden. Jedoch ist dies keine Möglichkeit

die Problematik komplett auszuschließen, denn Benutzer, welche ausreichend Kenntnis haben, können diese Limitierung einfach umgehen. Diese Vorgehensweise stellt jedoch einen angenehmen Ansatz dar, um unerfahrenere Benutzer von unerwarteten Bedienungsfehlern abzuhalten. In diesem Fall verliert das Smartphone jedoch schnell die Möglichkeit des einfachen Telefonierens, welches ev. im Gesamtprozess als Notfallfunktionalität durchaus Anwendung finden kann. Aber auch hier könnte durch einfache Integration in die *Home-App* diese Problematik durch *Intents* und die Übergabe einer Rufnummer (etwa jene zum Objektleiter) übergeben werden, um die Applikation zum einfachen Telefonieren aufzurufen.

Da die vorliegende Arbeit NFC als Hauptbestandteil behandelt, wurde ersterer Lösungsansatz in den Prototyp implementiert. Nachfolgend werden jedoch beide Varianten hinsichtlich der Implementierung näher beschrieben.

Für beide Lösungsansätze ist das *Android Manifest* mit der Verwendung von *Intent Filtern* (4.4) von entscheidender Bedeutung. Dieses stellt essentielle Informationen über die Applikation dem Betriebssystem zur Verfügung und beinhaltet u.a. die Registrierung der Komponenten wie *Activities*. Für jene *Activities*, welche das NFC-Protokoll einsetzen sollen, mussten zur Lösung der Problematik *Intent Filter* verwendet werden, um nach dem Scannen von Raum-Tags die Applikation wieder zu starten und diese Tags entsprechend zu behandeln.

```
<activity
    android:name="org.paral.fmcleaning.MainActivity" ...>

    <!-- Intent Filter zur Handhabung von
                          schreibgeschuetzten Raum-Tags -->
    <intent-filter>
        <action android:name="android.nfc.action.NDEF_DISCOVERED" />
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:scheme="http" />
    </intent-filter>
</activity>
```

Listing 5.1: *Activity* mit Registrierung eines *Intent Filters* zur Handhabung von NDEF-formatierten Tags im *Android Manifest*

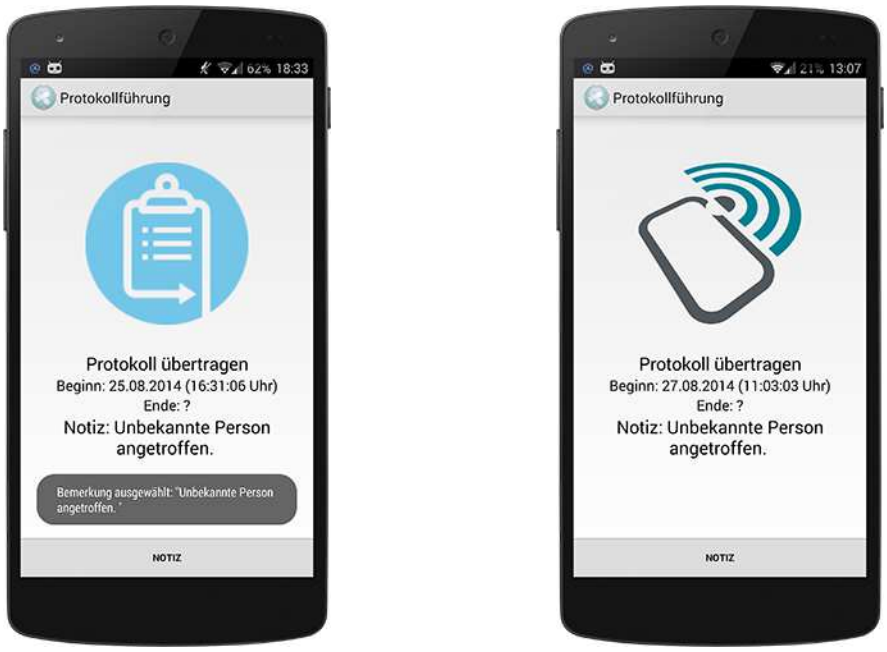
Da es sich bei Raum-Tags nur um schreibgeschützte Tags handelt, kann das *NFC Tag Dispatch System* (4.4) erkennen, dass es sich bei den Tag-Daten um das NDEF-Format handelt. Die ausgelesenen Daten werden vom Betriebssystem in einen *Intent* gekapselt, und durch den erkannten Schema-Typ (http) können diese automatisch zum Smartphone-Client durchgeschleift werden. Letzterer wird zuerst in den Vordergrund gebracht und anschließend mit den *Intent* versorgt. Die Prozedur zum Handhaben der Daten erfolgt anschließend wie im Abschnitt 4.4 beschrieben.

Der Lösungsansatz um den Smartphone-Client als *Home-App* zu deklarieren, verwendet ebenfalls den Ansatz der *Intent Filter* im *Android Manifest*. Sobald das Betriebssystem gebootet, ist

bekommt dieses eine *Notification*, um alle Applikationen, welche einen *Intent Filter* der Kategorie `android.intent.category.HOME` besitzen, zur Auswahl zu stellen. Das *Android* Betriebssystem besitzt mindestens eine Applikation dieser Kategorie. Nachdem im *Android Manifest* die besagte Kategorie zur *Launcher-Activity* hinzugefügt wurde, könnte nun die *Clean-Control* als Standard *Home-App* definiert und die besagte Problematik gelöst werden.

Protokollübermittlung bei Leistungserbringer

Die Übermittlung des Protokolls ist laut Konzept nach Scannen eines Raum Tags und somit erst nach Einsicht der raumgruppenspezifischen Daten möglich und erfolgt durch Klick auf ein Protokollsymbol auf einer separaten Ansicht. Ein Proband bemängelte, dass beim Hinzufügen einer Protokollnotiz eine Übermittlung zum Server stattgefunden hat, da er das Protokollsymbol unbeabsichtigt berührt hat. Als Verbesserung wurde anfangs eine Lösung in Form einer Bestätigung durch einen Dialog vorgeschlagen. Ein Proband brachte jedoch die Idee auch Raum-Tags zur Übermittlung dieses Protokolls nutzen zu können. Diese Anpassung beugt einerseits dem Problem der unbeabsichtigten Protokollübermittlung vor, und andererseits kann dadurch auch auf die weitere Anwesenheit des Leistungserbringers bei der betreffenden Räumlichkeit geschlossen werden. Die nachfolgenden Abbildungen zeigen die Funktionalität in einer vorher-nachher Gegenüberstellung.



(a) Protokollabgabe (vorher) (b) Protokollabagbe (nachher)

Abbildung 5.1: verbesserte Protokollführung mit NFC Funktionalität

Abb. 5.2 illustriert die Anpassung des in Abb. 3.3 ursprünglich erarbeiteten Ablaufkonzeptes zur NFC-Funktionalität beim Übermitteln eines Protokolls bei Leistungserbringern im

Smartphone-Client.

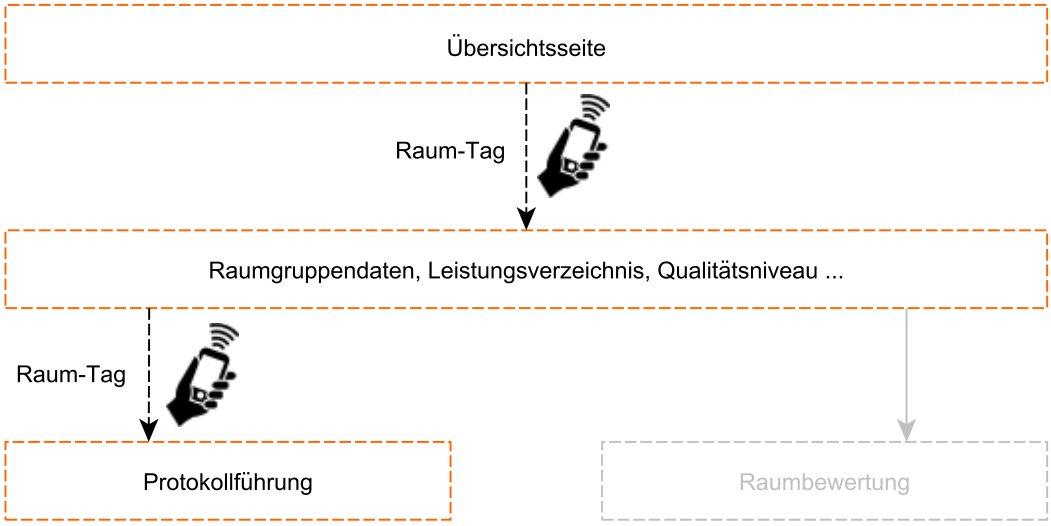


Abbildung 5.2: Überarbeiteter Ablauf der Protokollierungsfunktionalität mit NFC bei Leistungserbringern

Betriebsmittelliste

Zwei der Probanden in der Rolle des Leistungserbringers bemängelten, dass nach Arbeitsbeginn die Betriebsmittelliste nur einsehbar ist. Da diese Liste abhängig von den Gruppen der Räumlichkeiten jedoch sehr lange und damit äußerst unübersichtlich werden kann, wurde zur Benutzerfreundlichkeit eine Funktionalität zum Markieren der Listeneinträgen gewünscht. Durch diese Vorgehensweise zur Verbesserung der *Usability* kann die Liste beim Bezug der Betriebsmittel auf eine geordnete Weise abgearbeitet werden.

Ergebnisse zum Konzept

Mit dem abschließenden Fragebogen aus Anhang A.2 wurden die Probanden nochmals gebeten ihre Meinung zum erarbeiteten Konzept aus Kapitel 3 preiszugeben. Der nachfolgende Abschnitt soll diese Ansichten einer Diskussion unterziehen und Gegenargumente für eventuelle Kritikpunkte anführen.

Die einleitende Frage befasste sich damit, herauszufinden, ob die Testpersonen die Identifikation von Objekten mit der NFC-Technologie gegenüber manuellen Eingaben bevorzugen würden. Die teilnehmenden Personen waren sich allesamt einig und führten als Grund zur Bevorzugung von automatischer Identifikation als wesentliches Merkmal die Schnelligkeit und die geringere Fehleranfälligkeit gegenüber dem manuellen Suchen und Eingeben von Datensätzen an. Lediglich eine Testperson brachte den Vorschlag zusätzlich eine Identifikation mittels QR- oder Barcodes zur Verfügung zu stellen, da diese Technologie unabhängig davon genutzt werden

kann, ob NFC von der lesenden Einheit unterstützt wird oder nicht. Diese Tatsache wurde von jener Testperson hervorgebracht, welche zu privaten Zwecken ein *Apple iPhone* nutzte, welches nicht kompatibel wäre. Auf die Unterstützung dieser Smartphone-Typen soll daher nochmals kurz im Kapitel 6 im Zusammenhang mit den notwendigen Schritten zur Marktreife eingegangen werden.

Die Rückmeldungen auf die Frage ob die Testperson ein anderes Gerät als ein Smartphone bevorzugen würde waren gemischt. Einerseits lobten die Probanden die Umsetzung des Konzeptes auf das Smartphone, jedoch schlug sich bei zwei Personen auch Skepsis in Form einer Ungewissheit bezüglich der Lebensdauer dieser Hardware im täglichen Reinigungsalltag auf Seiten der Leistungserbringer wider. Smartphones sind durch die Tatsache, dass sie keine (oder nur wenige) Tasten besitzen insofern gut gegen das Eindringen von Flüssigkeit geeignet, stellen aber durch den Touchscreen ein erhöhtes Risiko für Bruchschäden am Display dar. Die Herstellerfirmen sind jedoch mittlerweile so weit, dass es für verschiedene Modelle bereits spritz- und wasserfeste Editionen am Markt gibt. Auch wirken dieser Problematik die immer weiterentwickelten Accessoires in Form von stoß- und feuchtigkeitsresistenten Schutzgehäusen entgegen, welche auch den Einzug dieser modernen Geräte auf Baustellen gewähren.

Betreffend der Frage, ob die Bedienung des Smartphones von der eigentlichen Arbeitserledigung abgelenkt hat, waren drei der Testpersonen der Meinung, dass jedenfalls eine gewisse Einarbeitungszeit notwendig wäre, um die Reinigungstätigkeiten der Leistungserbringer mit dem Bedienungskonzept des Smartphones in Einklang zu bringen. Ist der Gebrauch des Gerätes jedoch einmal fest im Arbeitsalltag verankert, so kann davon ausgegangen werden, dass das Smartphone jedenfalls nicht als eine Belastung angesehen wird. Diese Einarbeitungszeit, so stellten die Probanden fest, sollte in Form von Schulungen vor der Eingliederung des Smartphones in den tatsächlichen Arbeitsablauf gefestigt werden. Auf Seite der Arbeitnehmer – sind sich die Probanden einig – ist das Smartphone mit der Unterstützung der Qualitätssicherungsprozesse jedenfalls ein gelungenes Werkzeug.

Drei Testpersonen verneinten die Frage, ob Probleme beim Scannen von NFC-Tags aufgetreten sind. Die übrigen Probanden (sie waren mit NFC noch nicht vertraut, Anm.) erwähnten den anfangs ungewohnten Umgang mit den NFC-Tags. Nach wenigen weiteren Lese- und Schreibaktionen konnten sie sich jedoch mit dem Umgang schnell anfreunden und hatten auch sichtlich Spaß an der Bedienung des Systems. Letztendlich waren sich alle Personen einig, dass die NFC-Tags nach ein wenig Training schnell und einfach mit dem Smartphone ausgelesen oder beschrieben werden konnten. Der NFC-Leseadapter wird von verschiedenen Herstellern bei deren Modellen an unterschiedlichen Stellen verbaut. Die ersten Lesetests mit NFC-Smartphones gestalten sich daher zunächst nach dem Motto „Wer suchet der findet“. Auch sind NFC-Tags in unterschiedlichen Größen erhältlich. Für den Prototyp wurden Tags im Durchmesser von 4 cm verwendet. Je größer der Tag, desto einfacher und schneller gestalten sich die Bedienprozesse.

Auch bei der Frage, ob der Proband in der Rolle eines möglichen Endnutzers Gebrauch von Bewertung oder Bestellung der Reinigung machen würde, wurde durchaus positiv aufgefasst. Vier Personen meinten, dass sie Interesse an solch einer Funktionalität hätten und sich auch im Alltag bei Problemen oder auffälligen Besonderheiten (sei es im positiven und auch im negativen Sinne) schnell und einfach an den Objektleiter oder den verwaltenden *Facility-Service* wenden könnten. Auch erwähnte eine dieser Personen, dass die Bewahrung der Anonymität dadurch

eine wichtige Rolle spielen kann. Sie lobten allesamt die Integration der NFC-Tags in die Öffentlichkeit und würden auch zukünftigen Funktionalitätserweiterungen wie *Indoor*-Navigation oder Zutrittssystemen mit NFC positiv gegenüberstehen.

Gegen Ende wurden die Probanden noch mit der Fragestellung konfrontiert, ob die zentralen Handlungsprozesse im Konzept der Smartphone-Applikation bei Leistungserbringern und Auftragnehmern als „logisch“ anzusehen sind. Hierbei wurde nochmals die Umsetzung der *Usability* positiv in den Vordergrund gebracht und gelobt. Die Probanden waren überrascht, dass das NFC-Konzept so reibungslos in die Gesamtprozesse integriert werden konnte und meinten, dass bei Verwendung der Applikation immer gut ersichtlich war, was gerade passiert und was bei bestimmten Aktionen passieren wird.

Abschließend wurde die Frage gestellt, ob es vorstellbar ist all diese Prozesse des Konzeptes in Papierform erledigen zu müssen. Die Probanden meinten auf diese Frage, dass das durchwegs vorstellbar ist, würden aber durch das Wissen der Existenz einer solchen Infrastruktur lieber drauf verzichten.

KAPITEL 6

Zusammenfassung und Ausblick

Die zuvor beschriebenen Kapitel – und somit das zentrale Thema dieser Arbeit – befassten sich mit der Konzeption, Implementierung und Evaluierung eines Gesamtsystems, das die Rollen *Leistungserbringer*, *Auftragnehmer* und *Endnutzer* der Geschäftsprozesse eines Gebäudereinigungsdienstes im Facility-Management mit der NFC-Technologie für Smartphones vereinen soll. Nachfolgend sollen wichtige Schritte zur möglichen Marktreife des Gesamtsystems erläutert und klargestellt werden, ob die Forschungsfrage dieser Arbeit aus Abschnitt 1.1 positiv beantwortet werden kann.

6.1 Handlungsbedarf zur Marktreife

Bevor das Gesamtsystem auch tatsächlich einen Produktionsstatus erlangen kann, muss noch einigen Gesichtspunkten Aufmerksamkeit geschenkt werden.

Betreffend der im Webservice-Prototyp verwendeten Technologien wurde bereits im Kapitel 4 angemerkt, dass Sicherheit, Skalierbarkeit und Performance aus Gründen des geringen Neuheitswertes und der Tatsache, dass diese Eigenschaften nicht im Mittelpunkt der vorliegenden Arbeit standen, außer Acht gelassen wurden. So wurde für den Prototyp die Kommunikation mit den API-Endpunkten keinem *Session Handling* unterzogen, sondern die Schnittstellen ungeschützt implementiert. Außerdem gäbe es vor einem möglichen Feldversuch die Frage der serverseitigen Lastverteilung zu klären. Als Datenbankmanagementsystem wurde eine Technologie verwendet, welche bei höheren Zugriffszahlen und wachsenden Datenmengen an ihre Grenzen stoßen wird. Hierbei gilt es abzuklären auf welche Systeme migriert werden sollte. Mit dem Einsatz des vorliegenden **ORM-Frameworks!** wurde diesbezüglich jedoch schon vorab der Grundstein für eine einfache Migration gelegt. Des weiteren gilt es noch mit Spezialisten zu klären, ob die Serverumgebung nicht auf einen herkömmlichen Webserver wie dem der *Apache Foundation* integriert werden sollte.

Aus Gründen der Kompatibilität zu Smartphones welche NFC nicht unterstützen, könnten NFC-Tags auf ihrer Oberfläche zusätzlich mit QR- oder Barcodes bedruckt werden. Die Problematiken dieser Technologie wurden bereits im Abschnitt 2.3 diskutiert, für eine Unterstützung

einer *iOS* Applikationen für das *Apple iPhone* müssten diese Einschränkungen jedoch aktuell in Kauf genommen werden.

Auch ist die zugrundeliegende NFC-Infrastruktur durch den Einsatz des NDEF-Formats (2.1) auf den NFC-Tags nicht an die im Prototyp verwendeten *Mifare Ultralight C* Tag-Typen gebunden. Für zukünftige Tag-Migrationen wurde hierfür somit schon vorab Sorge getragen. Jedoch stellt sich eine Problematik in den Raum, denn die verwendeten NFC-Tags und die Smartphone Applikation implementieren den 3DES-Algorithmus zur Tag-Authentifizierung. Dieser *Security-Layer* müsste in solch einem Fall abhängig vom NFC-Tag erweitert, oder im Falle von permanent geschützten Tags, deaktiviert werden. Mittels dem *Android*-Betriebssystem und dem *NFC Tag Dispatch System* (4.4) ist es möglich, auf Tag-Typen und deren Eigenschaften dynamisch zu reagieren. Einer gemischten Tag-Typ-Infrastruktur würde somit nichts im Wege stehen.

Da für den Prototyp das CAFM-System auf der Webservice-Komponente simuliert wurde, wären hierbei ebenfalls noch Anpassungen notwendig. So müsste das zugrundeliegende CAFM-System eine Schnittstelle für den Webservice zur Verfügung stellen, um auf dessen Daten zugreifen zu können. Im Zuge dessen wird es ebenfalls notwendig sein ein Tool zur Verwaltung der Webservice Datenbank zu erstellen. Darunter würde u. a. das gesamte Benutzermanagement, die Verwaltung der Qualitätssicherungs- und Protokolldaten, das Management der Reinigungspläne, Betriebsmittel, Mängel und natürlich auch die in diesem Konzept im Mittelpunkt stehende NFC-Infrastruktur fallen.

Zu guter Letzt ist zu erwähnen, dass keine Software nur durch einen Usability-Test das Potential zur Marktreife erlangen kann. Hierfür müsste vor einem Produktionsstatus ein Pilotprojekt gestartet werden, um eine verlässliche Aussage über die Reife und das Marktpotential einer Software im Alltag treffen zu können. Mit dieser Version des Prototyps wurde ein bestmöglicher Grundstein dafür gelegt.

6.2 Forschungsergebnisse

Die mit dieser Arbeit angestrebte Beantwortung der Fragestellung inwieweit es möglich ist, mit Hilfe eines Smartphones unter Einsatz der NFC-Technologie die FM-Prozesse am Beispiel eines Gebäudereinigungsdienstes geeignet zu unterstützen und zu optimieren, konnte durch die Erarbeitung eines Konzeptes und der Umsetzung eines Prototyps mit anschließender Evaluierung mittels eines Usability-Tests gezeigt werden.

Die angestrebte Ersetzung aller bisher aus dem CAFM-System erzeugten papierbasierten Dokumente zur Dokumentation und Unterstützung bei der Abarbeitung von Prozessen, sowie die anfangs fragliche Bewahrung der Benutzerfreundlichkeit aus Sicht dreier am Geschäftsprozess beteiligten Rollen konnte mit der Umsetzung vier wesentlicher Komponenten gezeigt werden.

Hierbei spielte der Begriff *Internet of Things* eine entscheidende Rolle. Ein Webservice stellt die internet-basierte Komponente dar, um Daten zur Verfügung zu stellen und zu speichern. Der relevante Status eines realen Objektes wird auf NFC-Tags gespeichert. Diese Eigenschaften wurden benutzt, um das beschriebene Alter Ego Konzept aus Abschnitt 3.1 zu realisieren. Die entwickelte Smartphone Applikation agiert dabei im R*eader/Writer* Modus und liegt auf einer Ebene zwischen dem *Ding*, das durch NFC-Tags repräsentiert wird, und dem Webservice, welcher verwendet wird, um diesen Objektstatus abzurufen und zu manipulieren.

Es konnte durch einen Usability-Test und den daraus gemessenen Ergebnissen abgelesen werden, dass die Probanden der Meinung waren, eine Integration und eine komfortable Handhabung eines solchen Systems durchwegs in den Arbeitsalltag des gewählten Geschäftsprozesses einfließen zu lassen. Dabei spielten die drei Hauptanwendungsfälle *Dokumentation, Qualitätssicherung* und *Bewertung/Bestellung von Reinigung* eine wesentliche Rolle.

Letztendlich bleibt nur mehr die Sichtweise des Auftraggebers, welcher die Reinigung erwirbt. Das Institut für Reinigungsanalytik [O25] schreibt hier treffend:

> „*Wer Reinigung erwirbt, hat weniger die Absicht den Reinigungsprozess zu kaufen, sondern vielmehr das Resultat. Vielleicht ist das Resultat nicht das einzige was zählt, aber es ist unbedingt das Wichtigste.*“

Mittels des erarbeiteten Konzeptes und dem entwickelten Prototyp wurde gezeigt, dass nicht nur die Hauptaufgaben aus Sicht der aktiv teilnehmenden Rollen mittels mobilen Smartphonesystemen und der NFC-Technologie erfolgreich unterstützt werden konnten, sondern auch das Gesamtergebnis – die geforderte Sauberkeit aus Sicht des Auftraggebers – einfacher erreicht, und auch langfristig gesichert werden kann.

ANHANG A

Usability-Test

A.1 Aufgabenlisten der Rollen (Testskript)

Testskript für Auftragsnehmer

Aufgabe 1	Beschreibung
Bezeichnung	**Login**
Vorbedingung	Smartphone-Client muss gestartet sein und die Übersichtsseite anzeigen.
Erfolgsbeschreibung	Ein Benutzer der Rolle *Auftragnehmer* (Sie) ist eingeloggt. Ihr Name sowie der Ihrer Rolle muss nach dem Login ersichtlich sein.
Textuelle Beschreibung	Verwenden Sie Ihren Personal-Tag, um sich im Smartphone-Client anzumelden und in weiterer Folge Funktionalitäten für Auftragnehmer nutzen zu können.

Aufgabe 2	Beschreibung
Bezeichnung	**Qualitätssicherung beginnen**
Vorbedingung	Aktiver Login, Ansicht: Übersichtsseite
Erfolgsbeschreibung	Eine Stichprobe mit Räumlichkeiten wird erzeugt und angezeigt.
Textuelle Beschreibung	Verwenden Sie den fix montierten Aktions-Tag zum Erzeugen einer Stichprobe im Objekt *Karlsplatz 13*. Benutzen Sie die Funktionalität zum Betrachten der Reservierungen für die anstehenden Positionen in der Stichprobe.

Aufgabe 3	Beschreibung
Bezeichnung	**Daten einer Stichprobe anzeigen**
Vorbedingung	Aktiver Qualitätssicherungsprozess, Ansicht: Übersichtsseite
Erfolgsbeschreibung	Raumgruppendaten des Raumes 205 werden angezeigt.
Textuelle Beschreibung	Verwenden Sie den NFC-Tag des Raumes 205, um dessen Raumgruppendaten anzuzeigen. Diese sind der Zeitpunkt des vereinbarten Qualitätsniveaus, abweichende Ergänzungen, reinigungsassoziierte Dienstleistungen und Daten zu Raumkomponenten sowie Ausstattung. Fügen Sie auch einen Defekt mit einer Beschreibung Ihrer Wahl hinzu.

Aufgabe 4	Beschreibung
Bezeichnung	**Auswertung und Abschluss eines Qualitätssicherungsprozesses**
Vorbedingung	Aktiver Qualitätssicherungsprozess
Erfolgsbeschreibung	Alle Stichproben wurden abgearbeitet, die Qualitätssicherung wurde beendet.
Textuelle Beschreibung	Bewerten Sie den Raum 205 der aktiven Stichprobe als *durchgefallen* und fügen Sie einen Kommentar hinzu. Betrachten Sie anschließend die zwischenzeitliche Gesamtauswertung Ihres Qualitätssicherungsprozesses. Sind noch Stichproben zur Bewertung ausständig, wiederholen Sie den Vorgang bis alle Räumlichkeiten bewertet wurden. Gehen Sie bei den nachfolgenden Bewertungen so vor, dass Sie im Bewertungsformular auch die Eingabefelder der Raumkomponenten nutzen. Nachdem alle Stichproben bewertet wurden, nutzen Sie den Aktions-Tag zum Beenden der Qualitätssicherung im Objekt.

Aufgabe 5	Beschreibung
Bezeichnung	**NFC-Infrastruktur erweitern**
Vorbedingung	Aktiver Login als Objektleiter
Erfolgsbeschreibung	Neue NFC-Tags für Personal, Räume und Aktionen wurden beschrieben und nutzbar gemacht.
Textuelle Beschreibung	Nutzen Sie die Funktionalität zum Warten der NFC-Infrastruktur, um einen neuen Personal-Tag für die Person *Max Mustermann* zu erstellen. Diese Person ist laut Datenbank ebenfalls ein Objektleiter. Führen Sie daher unmittelbar nach Erzeugung dieses Tags einen neuen Login mit diesem NFC-Tag durch, und kehren Sie wieder auf die aktuelle Seite zurück. Wählen Sie nun das Objekt *Informatiklabor* und erzeugen Sie beide NFC-Tags für Aktions-Beginn und Aktions-Ende. Prüfen Sie, ob in diesem Objekt Räumlichkeiten zum Programmieren eingetragen sind. Wenn ja, arbeiten Sie die Liste ab, und testen Sie anschließend wie in Aufgabe 2 und 3 beschrieben Ihre erzeugten NFC-Tags.

Testskript für Leistungserbringer

Aufgabe 1	Beschreibung
Bezeichnung	**Login**
Vorbedingung	Smartphone-Client muss gestartet sein und die Übersichtsseite anzeigen.
Erfolgsbeschreibung	Ein Benutzer der Rolle *Leistungserbringer* (Sie) ist eingeloggt. Ihr Name sowie der Ihrer Rolle muss nach dem Login ersichtlich sein.
Textuelle Beschreibung	Verwenden Sie Ihren Personal-Tag, um sich im Smartphone-Client anzumelden und in weiterer Folge Funktionalitäten für Leistungserbringer nutzen zu können.

Aufgabe 2	Beschreibung
Bezeichnung	**Arbeit beginnen, Reinigungsplan mit Nacharbeiten abrufen**
Vorbedingung	Aktiver Login, Ansicht: Übersichtsseite
Erfolgsbeschreibung	Die Arbeit wurde begonnen, der Reinigungsplan und mögliche Nacharbeiten mit Räumlichkeiten wurde abgerufen und angezeigt.
Textuelle Beschreibung	Verwenden Sie den fix montierten Aktions-Tag zum Abrufen des aktuellen Reinigungsplans im Objekt *Informatiklabor*. Benutzen Sie die Funktionalität zum Betrachten der Betriebsmittel und der Reservierungen für die anstehenden Positionen im Reinigungsplan. Sind Nacharbeiten von früheren Protokollübermittlungen notwendig? Wenn ja, finden Sie heraus was der Grund war, und ob Nacharbeit notwendig ist. Betrachten Sie in solch einem Fall auch die Bewertungsmaske der Qualitätssicherung, die übermittelt wurde.

Aufgabe 3	Beschreibung
Bezeichnung	**Daten einer Räumlichkeit im Reinigungsplan anzeigen**
Vorbedingung	Aktive Arbeit mit Reinigungsplan, Ansicht: Übersichtsseite
Erfolgsbeschreibung	Raumgruppendaten des Raumes 208 werden angezeigt.
Textuelle Beschreibung	Verwenden Sie den NFC-Tag des Raums 208, um dessen Raumgruppendaten anzuzeigen. Diese sind der Zeitpunkt des vereinbarten Qualitätsniveaus, abweichende Ergänzungen, reinigungsassoziierte Dienstleistungen und Daten zu Raumkomponenten sowie Ausstattung. Fügen Sie auch einen Defekt mit einer Beschreibung Ihrer Wahl hinzu. Finden Sie heraus, ob Reservierungen für diesen Raum eingetragen sind. Nutzen Sie die Möglichkeit der Selbstkontrolle, und probieren Sie verschiedene Bewertungskombinationen aus.

Aufgabe 4	Beschreibung
Bezeichnung	**Übermittlung eines Protokolls und Arbeitsabschluss**
Vorbedingung	Aktive Arbeit mit Reinigungsplan, Ansicht: Raumgruppendaten
Erfolgsbeschreibung	Alle Räumlichkeiten im Reinigungsplan wurden abgearbeitet, die Arbeit wurde beendet.
Textuelle Beschreibung	Protokollieren Sie den Raum 208 im aktuellen Reinigungsplan als abgearbeitet, und fügen Sie einen Kommentar hinzu. Betrachten Sie anschließend die noch abzuarbeitenden Räumlichkeiten im Plan. Sind noch Protokolle ausständig, wiederholen Sie den Vorgang bis alle Räumlichkeiten bewertet wurden. Achten Sie auch darauf, dass Räumlichkeiten mit notwendiger Nacharbeit ebenfalls abgearbeitet werden. Nachdem alle Protokolle übermittelt wurden, nutzen Sie den Aktions-Tag zum Beenden der Arbeit im Objekt.

Testskript für Endnutzer

Aufgabe 1	Beschreibung
Bezeichnung	**Raumdaten anzeigen (NFC Variante)**
Vorbedingung	Entsperrtes Smartphone mit aktiver NFC-Funktionalität
Erfolgsbeschreibung	Öffentlich einsehbare Daten des Raumes 203 werden gemeinsam mit einem Eingabeformular im Smartphone-Browser angezeigt.
Textuelle Beschreibung	Verwenden Sie auf einem NFC-Smartphone Ihrer Wahl den NFC-Tag des Raumes 203 um, dessen öffentlich einsehbare Raumgruppendaten im Smartphone-Browser anzuzeigen.

Aufgabe 2	Beschreibung
Bezeichnung	**Raum bewerten/Reinigung bestellen**
Vorbedingung	Daten des Raumes 203 sowie ein Eingabeformular werden im Smartphone-Browser angezeigt.
Erfolgsbeschreibung	Die Bewertung/Bestellung der Reinigung für Raum 203 wurde übermittelt, und dies wird per Ausgabe bestätigt. Der Testleiter bestätigt Ihnen nachfolgend die übermittelte Meldung zur Kontrolle.
Textuelle Beschreibung	Übermitteln Sie mit Hilfe der Eingabemöglichkeiten Ihres Smartphones einen Kommentar Ihrer Wahl zur Bewertung oder Bestellung einer speziellen Reinigung für Raum 203.

Aufgabe 3	Beschreibung
Bezeichnung	**Raumauswahl im Webbrowser**
Vorbedingung	Smartphone- oder Notebook-Browser ist geöffnet und zur Eingabe einer URL bereit.
Erfolgsbeschreibung	Eine Raumauswahl im betreffenden Objekt wird angezeigt.
Textuelle Beschreibung	Verwenden Sie auf einem Notebook oder einem Smartphone, das kein NFC hat, den Webbrowser, um die Auswahlseite der Räumlichkeiten im betreffenden Objekt anzuzeigen. Die Webbrowser-URL wurde im Test-Labor auf einem Flipchart notiert.

Aufgabe 4	Beschreibung
Bezeichnung	**Raumdaten anzeigen (manuelle Variante)**
Vorbedingung	Raumauswahl wird im Webbrowser (Smartphone oder Notebook) angezeigt.
Erfolgsbeschreibung	Die öffentlich verfügbaren Daten des Raumes 102 sowie ein Eingabeformular zur Abgabe eines Kommentars werden im Webbrowser angezeigt.
Textuelle Beschreibung	Verwenden Sie die Ihnen angezeigten Formularkomponenten, um Raum 102 auszuwählen, damit dessen öffentlich zugängliche Raumgruppendaten im Webbrowser am Gerät Ihrer Wahl angezeigt werden.

A.2 Abschlussfragebogen

Fragen zum Prototyp

Bitte beantworten Sie folgende Fragen betreffend dem Prototyp:

1. Was ist Ihr allgemeiner Eindruck zum Systems?
2. Finden Sie die App übersichtlich oder überladen?
3. Wie haben Sie sich insgesamt zurechtgefunden?
4. Empfanden Sie die Benutzung der Anwendungen als einfach oder kompliziert?
5. Welche besonderen Stärken sind Ihnen aufgefallen?
6. Welche besonderen Schwächen sind Ihnen aufgefallen?
7. Was sind die wichtigsten Punkte, die Sie verbessern/verändern würden?
8. Hatten Sie Spaß an der Benutzung?
9. Nennen Sie bitte die 3 Aspekte, die Ihnen besonders gut gefallen, und die Sie unbedingt beibehalten würden?
10. Weitere Anmerkungen: .....

Fragen zum Konzept

Bitte beantworten Sie folgende Fragen betreffend das Konzept, und begründen Sie kurz Ihre Antwort:

1. Würden Sie die Identifikation von *Objekten* mit der NFC-Technologie gegenüber manuellen Eingaben bevorzugen?
2. Würden Sie lieber ein anderes Gerät als ein Smartphone einsetzen?
3. Sind Sie der Meinung, dass Sie die Benutzung des Systems von der tatsächlich zu erledigenden Arbeit abgelenkt hat?
4. Hatten Sie Probleme beim Scannen der NFC-Tags?
5. Würden Sie als *Endnutzer* Gebrauch von der Bewertung von Reinigungsdienstleistungen oder die Bestellung selbiger mit diesem System machen?
6. Empfinden Sie als *Auftragnehmer* oder *Leistungserbringer*, dass die Interaktionssequenzen bei zentralen Handlungsprozessen wie Protokollvergabe oder Bewertungen als “logisch” anzusehen sind?
7. Können Sie sich vorstellen all diese Prozesse in Papierform erledigen zu müssen?
8. Weitere Anmerkungen: .....

ANHANG B

Mifare Ultralight C Taginfo

Die Implementierung des Prototyps wurde mit *Mifare Ultralight C* Tags realisiert. Es folgt die Ausgabe des *Logs* eines Tags wie er vom Werk aus geliefert wurde. Die Ausgabe wurde mit der *Android* App *NFC TagInfo by NXP* [1] der Firma *NXP Semiconductors* erzeugt.

```
** TagInfo scan (version 2.00) 2014-06-11 19:59:49 **

-- INFO -----------------------------
# IC manufacturer: NXP Semiconductors
# IC type: MIFARE Ultralight C (MF0ICU2)
-- NDEF -----------------------------
# No NFC data set available:

# NDEF Capability Container (CC):
Mapping version: 1.1
Maximum NDEF data size: 144 bytes
NDEF access: Read & Write
E1 11 12 00                      |....            |

# Control TLVs:
Lock Control TLV at address 0x04, offset 0
* Dynamic lock bytes at address 0x28, offset 0
    - 16 lock bits
    - 16 bytes locked per lock bit
01 03 A0 10 44                   |....D           |
-- EXTRA ----------------------------
# Memory size:
192 bytes total memory
* 48 pages, with 4 bytes per page
* 144 bytes user memory (36 pages)

# Authentication information: Default sample key
-- TECH -----------------------------
```

[1] https://play.google.com/store/apps/details?id=com.nxp.taginfolite

```
# Technologies supported:
ISO/IEC 14443-3 (Type A) compatible
ISO/IEC 14443-2 (Type A) compatible

# Android technology information:
Tag description:
* TAG: Tech [android.nfc.tech.NfcA, android.nfc.tech.MifareUltralight ,
    android.nfc.tech.Ndef, android.nfc.tech.NdefFormatable]
android.nfc.tech.Ndef
android.nfc.tech.NdefFormatable
android.nfc.tech.MifareUltralight
android.nfc.tech.NfcA
* Maximum transceive length: 253 bytes
* Default maximum transceive time-out: 618 ms
MIFARE Classic support present in Android

# Detailed protocol information:
ID: 04:51:22:A2:8D:2A:80
ATQA: 0x4400
SAK: 0x00

# Memory content:
[00] *  04:51:22 FF (UID0-UID2, BCC0)
[01] *  A2:8D:2A:80 (UID3-UID6)
[02] .  85 48 00 00 (BCC1, INT, LOCK0-LOCK1)
[03] .  E1:11:12:00 (OTP0-OTP3)
[04] .  01 03 A0 10 |....|
[05] .  44 03 00 FE |D...|
[06] .  00 00 00 00 |....|
[07] .  00 00 00 00 |....|
[08] .  00 00 00 00 |....|
[09] .  00 00 00 00 |....|
[0A] .  00 00 00 00 |....|
[0B] .  00 00 00 00 |....|
[0C] .  00 00 00 00 |....|
[0D] .  00 00 00 00 |....|
[0E] .  00 00 00 00 |....|
[0F] .  00 00 00 00 |....|
[10] .  00 00 00 00 |....|
[11] .  00 00 00 00 |....|
[12] .  00 00 00 00 |....|
[13] .  00 00 00 00 |....|
[14] .  00 00 00 00 |....|
[15] .  00 00 00 00 |....|
[16] .  00 00 00 00 |....|
[17] .  00 00 00 00 |....|
[18] .  00 00 00 00 |....|
[19] .  00 00 00 00 |....|
[1A] .  00 00 00 00 |....|
[1B] .  00 00 00 00 |....|
[1C] .  00 00 00 00 |....|
[1D] .  00 00 00 00 |....|
```

```
[1E] .  00 00 00 00 |....|
[1F] .  00 00 00 00 |....|
[20] .  00 00 00 00 |....|
[21] .  00 00 00 00 |....|
[22] .  00 00 00 00 |....|
[23] .  00 00 00 00 |....|
[24] .  00 00 00 00 |....|
[25] .  00 00 00 00 |....|
[26] .  00 00 00 00 |....|
[27] .  00 00 00 00 |....|
[28] .  00 00 -- -- (LOCK2-LOCK3)
[29] .  00 00 -- -- (CNT0-CNT1, value: 0)
[2A] .  30 -- -- -- (AUTH0)
[2B] .  01 -- -- -- (AUTH1)
[2C] .- 42 52 45 41 |BREA|
[2D] .- 4B 4D 45 49 |KMEI|
[2E] .- 46 59 4F 55 |FYOU|
[2F] .- 43 41 4E 21 |CAN!|

 *:locked & blocked, x:locked,
 +:blocked, .:un(b)locked, ?:unknown
 r:readable (write-protected),
 p:password protected, -:write-only
 P:password protected write-only
-------------------------------------
```

Literatur

Literatur

[L1] M. Aazam u. a. „Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved“. In: *2014 11th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. 2014, S. 414–419.

[L2] Olaf Th. Buck, Robert Gajcy und Volker Linnemann. „Mobiles Computer Aided Facility Management.“ In: *BTW*. Hrsg. von Alfons Kemper u. a. Bd. 103. LNI. GI, 2007, S. 522–531. ISBN: 978-3-88579-197-3.

[L3] Dominique Guinard, Iulia Ion und Simon Mayer. „In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers' Perspective“. In: *Proceedings of Mobiquitous 2011 (8th International ICST Conference on Mobile and Ubiquitous Systems)*. Copenhagen, Denmark, 2011, S. 326–337.

[L4] O. Kleine. „Integrating the Physical World with the Internet – A Concept Evaluation“. In: *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications (SOCA)*. 2013, S. 323–327.

[L5] M. Landhausser und A. Genaid. „Connecting User Stories and code for test development“. In: *Recommendation Systems for Software Engineering (RSSE), 2012 Third International Workshop on*. 2012, S. 33–37.

[L6] D. Lau u. a. „A cloud-based approach for smart facilities management“. In: *2013 IEEE Conference on Prognostics and Health Management (PHM)*. 2013, S. 1–8.

[L7] Yu-Cheng Lin. „Enhancing Facility Management Using RFID and Web Technology in Construction“. In: *Robotics and Automation in Construction*. Hrsg. von Carlos Balaguer und Mohamed Abderrahim. National Taipei University of Technology/ Civil Engineering, Taiwan: InTech, 2008, S. 199–210.

[L8] A. Malatras, A. Asgari und T. Bauge. „Web Enabled Wireless Sensor Networks for Facilities Management“. In: *Systems Journal, IEEE* 2.4 (2008), S. 500–512.

[L9] M. Merhi, J.C. Hernandez-Castro und P. Peris-Lopez. „Studying the pseudo random number generator of a low-cost RFID tag“. In: *2011 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*. 2011, S. 381–385.

[L10] K. Ok u. a. „Current benefits and future directions of NFC services“. In: *2010 International Conference on Education and Management Technology (ICEMT)*. 2010, S. 334–338.

[L11] S. Ortiz. „Is near-field communication close to success?“ In: *IEEE Computer* 39.3 (2006), S. 18–20.

[L12] David Oswald und Christof Paar. „Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World“. In: *Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems*. CHES'11. Nara, Japan: Springer-Verlag, 2011, S. 207–222. ISBN: 978-3-642-23950-2.

[L13] P. Stack. „Development of a Mobile Platform to Support Building Maintenance Engineering“. In: *Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual*. 2012, S. 482–487.

[L14] Sandeep Tamrakar, Jan-Erik Ekberg und N. Asokan. „Identity Verification Schemes for Public Transport Ticketing with NFC Phones“. In: *Proceedings of the Sixth ACM Workshop on Scalable Trusted Computing*. STC '11. Chicago, Illinois, USA: ACM, 2011, S. 37–48. ISBN: 978-1-4503-1001-7.

[L15] Lu Tan und Neng Wang. „Future internet: The Internet of Things“. In: *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. Bd. 5. 2010, S. 376–380.

[L16] Meng Wang u. a. „Implementation of Internet of Things Oriented Data Sharing Platform Based on RESTful Web Service“. In: *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*. 2011, S. 1–4.

[L17] Szu-Hui Wu und Chyan Yang. „Promoting Collaborative Mobile Payment by Using NFC-Micro SD Technology“. In: *2013 IEEE International Conference on Services Computing (SCC)*. 2013, S. 454–461.

[L18] A. Zeaaraoui u. a. „User stories template for object-oriented applications“. In: *2013 Third International Conference on Innovative Computing Technology (INTECH)*. 2013, S. 407–410.

Normen & Standards

[NS1] *Begriffe zum Qualitätsmanagement - Teil 11: Ergänzung zu DIN EN ISO 9000:2005*. Norm. 2008.

[NS2] *Facility Management - Teil 1: Begriffe; Deutsche Fassung EN 15221-1:2006*. Norm. 2006.

[NS3] *IEEE Draft International Standard for Software and Systems Engineering–Software Testing–Part 4: Test Techniques*. 2014.

[NS4] *ÖNORM A 7000 - Facility Management - Grundkonzepte*. Norm. 2000.

[NS5] *Reinigungsdienstleistungen - Grundanforderungen und Empfehlungen für Qualitätsmesssysteme; Deutsche Fassung*. Norm. 2001.

Oline-Referenzen

[O1] W3C Working Group Note 11 February 2004. *Web Services Architecture*. 2. Mai 2014. URL: http://www.w3.org/TR/ws-arch/.

[O2] ServiceControl TransparentManagement AG. *easyBird Cleaning CLOUD 1.1 - Vorsprung im Wettbewerb durch die Kombination aus mobilem Lesegerät und webbasierender Software*. 9. Apr. 2014. URL: https://www.sc-protec.de/samweb/homepage/products/CleaningTrackware.php.

[O3] Der Gebäude Manager Die österreichische Monatszeitung für die nachhaltige Bewirtschaftung von Gebäuden und Anlagen. *Facility-Management-Aufgaben mobil erledigen*. 9. Apr. 2014. URL: http://www.gebÃd'udemanager.at/betrieb/facility-management-aufgaben-mobil-erledigen/.

[O4] Kevin Ashton. *That 'Internet of Things' Thing*. 30. Apr. 2014. URL: http://www.rfidjournal.com/articles/view?4986.

[O5] Martin Beschnitt. *Usability-Test - 16 Methoden zur Messung der Usability*. 1. Juli 2014. URL: http://www.onlinemarketing-praxis.de/web-usability/usability-test-16-methoden-zur-messung-der-usability.

[O6] Nokia Developer BETA. *Differences among different NFC tags*. 16. Apr. 2014. URL: http://developer.nokia.com/community/wiki/Differences_among_different_NFC_tags.

[O7] Sebastan Cech. *Ad-hoc Vernetzung im Bauwesen und Facility Management unter Nutzung von aktiven RFID*. 9. Apr. 2014. URL: http://www.schnitzlers.de/wp-content/uploads/2007/10/rfid-im-bauwesen-und-facility-management.pdf.

[O8] nfc-tag.de CleanCut Werbeagentur. *NFC Chiptypen*. 16. Apr. 2014. URL: http://www.nfc-tag.de/nfc-chiptypen.

[O9] libnfc developers community. *MIFARE UltraLight C 3DES authentication APDUs*. 11. Juni 2014. URL: http://www.libnfc.org/community/topic/665/mifare-ultralight-c-3des-authentication-apdus.

[O10] *Dalvik Technical Information*. 8. Mai 2014. URL: http://source.android.com/devices/tech/dalvik/index.html.

[O11] speedikon FM Aktiengesellschaft. *Reinigungskontrolle*. 14. März 2014. URL: https://www.speedikonfm.com/download/produkte/reinigungskontrolle.pdf.

[O12] NFC Forum. *NFC Forum Technical Specifications*. 24. Apr. 2014. URL: http://members.nfc-forum.org/specs/spec_list/.

[O13] NFC Forum. *Typ 2 Tag Operation Technical Specification Version 1.2*. 5. Juni 2014. URL: http://www.nfc-forum.org.

[O14] Bundesinnungsverband des Gebäudereiniger-Handwerks. *Qualitätsmesssystem für ergebnisorientierte Reinigungsleistungen - Systembeschreibung*. 15. Jan. 2014. URL: http://www.kruse-w.de/cms_sources/dateien/Systembeschreibung_02.pdf.

[O15] pit cup GmbH. *pit-Mobile - Die perfekte Lösung für Ihren mobilen Einsatz*. 9. Apr. 2014. URL: http://www.pit.de/file/Prospekte/W-pit-Mobile-2014-03.pdf.

[O16] Prevera Service GmbH. *Capitano - Die mobile Qualitätskontrolle für Facility Manager, Gebäude- und Hausverwalte*. 9. Apr. 2014. URL: http://prevera.at/capitano/PREVERA_Capitano_Folder_FM.pdf.

[O17] Peter Harrop, Raghu Das und Glyn Holland. *Near Field Communication (NFC) 2014-2024: Mobile phone and other NFC: market forecasts, technology, players*. 15. Jan. 2014. URL: http://www.researchmoz.us/near-field-communication-nfc-2014-2024-report.html.

[O18] *History of Near Field Communication*. 3. Jan. 2014. URL: http://www.nearfieldcommunication.org/history-nfc.html.

[O19] Identcode-Systeme ICS International AG. *MDE-Geräte zur mobilen Datenerfassung*. 15. März 2014. URL: http://www.ics-ident.de/ICS-CMS1/IT-Logistik-Systeme/MDE-Geraete,49.html.

[O20] Google Inc. *Android Design*. 9. Mai 2014. URL: http://developer.android.com/design.

[O21] Google Inc. *App Components*. 10. Mai 2014. URL: http://developer.android.com/guide/components.

[O22] *Near Field Communication – Technology Standards*. 27. Dez. 2013. URL: http://www.nearfieldcommunication.org/technology.html.

[O23] *NFC-Forum*. 27. Dez. 2013. URL: http://nfc-forum.org/.

[O24] Jakob Nielsen. *Why You Only Need to Test with 5 Users*. 28. Juni 2014. URL: http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users.

[O25] Institut für Reinigungsanalytik. *Qualitätssicherung*. 14. Feb. 2014. URL: http://www.ira-institut.de/down_qualitaetssicherung.php.

[O26] Portland Pattern Repository. *Model View Controller*. 10. Mai 2014. URL: http://c2.com/cgi-bin/wiki?ModelViewController.

[O27] NXP Semicoductors. *AN10927 - MIFARE and handling of UIDs*. 5. Juni 2014. URL: http://www.nxp.com/documents/application_note/AN10927.pdf.

[O28] NXP Semiconductors. *AN1303 (Application note) - MIFARE Ultralight as Type 2 Tag*. 5. Juni 2014. URL: http://www.nxp.com/documents/application_note/AN1303.pdf.

[O29] NXP Semiconductors. *MIFARE Ultralight C security design and its impact on chip performance*. 11. Juni 2014. URL: http://www.nxp.com/restricted_documents/53420/AN183610_MIFARE_Ultralight_C_23092009.pdf.

[O30] SoftwareTestingStandard.org. *ISO/IEC/IEEE 29119 Software Testing*. 28. Juni 2014. URL: http://www.softwaretestingstandard.org.

[O31] Sony. *NFC Forum Specifications*. 23. Apr. 2014. URL: http://www.sony.net/Products/felica/NFC/forum.html.

[O32] Joshua Steimle. *Why Your Business Needs A Responsive Website Before 2014*. 12. Juni 2014. URL: http://www.forbes.com/sites/joshsteimle/2013/11/08/why-your-business-needs-a-responsive-website-before-2014.

[O33] Frank Tenzer. *Statistiken und Studien zum Thema Smartphones*. 15. Nov. 2013. URL: http://de.statista.com/themen/581/smartphones/.

[O34] James Thrasher. *A Primer On The Internet of Things & RFID*. 30. Apr. 2014. URL: http://blog.atlasrfidstore.com/internet-of-things-and-rfid.

[O35] Ashwin Viswanath. *How JSON and Big Data Will Shape the Internet of Things*. 3. Juni 2014. URL: http://xml.sys-con.com/node/2881856.

[O36] Don Wells. *User Stories*. 5. Mai 2014. URL: http://www.extremeprogramming.org/rules/userstories.html.

[O37] Florian Kalenda Leitender Redakteur ZDNet.de. *Studie: 93 Prozent aller Smartphones mit NFC laufen unter Android*. 13. Apr. 2014. URL: http://www.zdnet.de/88184166/studie-93-prozent-aller-smartphones-mit-nfc-laufen-unter-android/.