Inhaltsverzeichnis

Inhaltsverzeichnis i
1 Einleitung 1
1.1 Motivation / Bedeutung dieser Arbeit 1
1.2 Ziel der Arbeit 1
1.3 Gliederung 1
2 Grundlagen 2
2.1 Einleitung 2
2.2 Mobile Geräte 2
2.2.1 Funktionalität 2
2.2.2 Einschränkungen 3
2.2.3 Beispiel für ein Endgerät: Das G1 von HTC 6
2.3 Sicherheit 6
2.3.1 Einleitung 6
2.3.2 Sicherheitsziele 7
2.3.3 Bedrohungen, Schwachstellen und Gefährdungen 10
2.3.4 Spezielle Bedrohungen 11
2.3.5 Techniken zur Bedrohungsabwehr 13
2.3.6 Technische Maßnahmen 14
2.3.7 Schutzbedarfsermittlung 16
2.4 eCommerce 17
2.4.1 Mobile Commerce 18
2.4.2 Mobile Ticketing 18
2.4.3 Mobile Payment 19
2.4.4 Sicherheitsanforderungen 20
2.5 Android 21
2.5.1 Einleitung 21
2.5.2 Plattform-Architektur 21
2.5.3 Anwendungsentwicklung 23
2.5.4 Sicherheitsmodell 23
2.6 Vergleichbare Arbeiten zum Thema 24
2.7 Zusammenfassung 25
3 Beispielszenario 26

3.1 Einleitung ... 26
3.1.1 Thin-Client-Architektur ... 28
3.1.2 Mobile Kommunikationsinfrastruktur ... 29
3.1.3 Mobiler Informations- und Ticketingdienst ... 30
3.2 Anwendungsfälle und Funktionsweise ... 32
3.2.1 Softwareinstallation auf dem Gerät ... 32
3.2.2 Allgemeine Funktionsweise und Programmablauf ... 33
3.2.3 Fahrplanauskunft ... 34
3.2.4 Ticket kaufen ... 35
3.2.5 Fahrkartenkontrolle ... 36
3.2.6 Kundeneinstellungen ... 38
3.3 Architektur ... 38
3.3.1 Übersicht über Komponenten ... 39
3.3.2 Activities ... 39
3.3.3 Services ... 40
3.3.4 Abläufe ... 41
3.3.5 Daten ... 42
3.4 Zusammenfassung ... 46
4 Security Engineering ... 47
4.1 Sicherheitsanforderungen ... 47
4.1.1 IT-Strukturanalyse ... 47
1.1.1 Identifikation schützenswerter Informationen ... 48
4.1.2 Schutzbedarfsfeststellung ... 52
4.1.2.1 Verfügbarkeit ... 52
4.1.2.2 Vertraulichkeit ... 55
4.1.2.3 Integrität ... 57
4.1.2.4 Zusammenfassung ... 59
4.1.2.5 Verbindlichkeit ... 59
4.1.2.6 Authentizität ... 60
4.1.3 Bedrohungsanalyse ... 60
4.1.3.1 Direkter Zugriff auf das Gerät ... 61
4.1.3.2 Malware ... 62
4.1.3.3 Bedrohungen der Kommunikation ... 65
4.1.3.4 Benutzerverhalten ... 65
4.1.3.5 Anforderungen des eCommerce ... 66
4.1.4 Zusammenfassung der Sicherheitsanforderungen ... 66

4.2 Entwicklung und Realisierung ... 67
4.2.1 Sicherheitsstrategie ... 67
4.2.1.1 Zugriffskontrolle ... 67
4.2.2 Technische Maßnahmen auf Anwendungsebene ... 70
4.2.2.1 Absicherung des Kommunikationskanals ... 70
4.2.2.2 Verschlüsselte Datenspeicherung ... 72
4.2.2.3 Digitale Signierung ... 73
4.2.2.4 Verfügbarkeit der Daten ... 75
4.2.2.5 Zusammenfassung ... 75
4.2.3 Sicherheitsmaßnahmen von Android ... 76
4.2.3.1 Verwendung von Java ... 77
4.2.3.2 Authentifikation am Gerät ... 78
4.2.3.3 Zugriffskontrolle ... 78
4.2.3.4 Anwendungsverteilung ... 79
4.3 Bewertung ... 79
4.3.1 Testen auf Schwachstellen anhand der Bedrohungen ... 80
4.3.1.1 Kommunikationsschwachstellen ... 80
4.3.1.2 Direkter Zugriff auf das Gerät ... 81
4.3.1.3 Malware ... 82
4.3.1.4 Weitere Gefahren ... 84
4.3.2 Kritikpunkte und mögliche Lösungen ... 84
5 Fazit und Ausblick ... 87
5.1 Zusammenfassung ... 87
5.2 Fazit ... 88
5.3 Ausblick ... 89
Abbildungsverzeichnis ... 90
Tabellenverzeichnis ... 91
Abkürzungsverzeichnis ... 92
Literaturverzeichnis ... 93
Anhang ... 95

1 Einleitung

1.1 Motivation / Bedeutung dieser Arbeit

Durch den technischen Fortschritt wächst die Leistungsfähigkeit mobiler tragbarer Geräte ständig an, sodass es immer neue Einsatzgebiete für die Verwendung solcher Geräte gibt. Zu dieser Geräteklasse gehören unter anderem moderne Mobiltelefone und Smartphones. Insbesondere im e-Commerce-Bereich, zu dem u.a. auch der elektronische Zahlungsverkehr gehört, werden dabei hohe Sicherheitsanforderungen an die zugrundeliegende Soft- und Hardware gestellt. Die *Open Handset Alliance* unter Führung von *Google* bietet mit *Android* nun speziell für solche mobilen Geräte eine neue Software-Plattform an. Diese enthält etliche neue Features, die viel Potential für zukünftige neuartige Entwicklungen bieten.

1.2 Ziel der Arbeit

Diese Arbeit prüft die Tauglichkeit von *Android* als eine sichere Umgebung für mobile Anwendungen aus dem eCommerce-Bereich, genauer gesagt, aus dem mCommerce. Dies geschieht anhand einer Beispielanwendung für ein fiktives Anwendungsszenario, für das eine Sicherheitsanalyse durchgeführt wird. Eine abschließende Bewertung der Umsetzbarkeit zuvor aufgestellter Sicherheitsanforderungen soll das Ergebnis dieser Arbeit liefern, welches auch einen Transfer auf andere Anwendungsszenarien ermöglicht.

1.3 Gliederung

Das folgende Kapitel (Grundlagen) geht auf die verschiedenen Teilgebiete der Sicherheitsanalyse für eCommerce Anwendungen in *Android* ein. Dazu gehört *Android* als die Software-Plattform, mobile Geräte als die Hardware-Plattform, eCommerce als das Anwendungsgebiet, und Sicherheit als die Eigenschaft von IT-Systemen, welche im Verlauf dieser Arbeit näher analysiert werden soll.

Im dritten Kapitel (Beispielszenario) wird ein fiktives Anwendungsszenario aus dem Bereich des *Mobile Ticketing* vorgestellt. Dabei handelt es sich um ein IT-System für die mobile Nutzung von Dienstleistungen im öffentlichen Personennahverkehr (ÖPV), zu dem die Fahrplanauskunft als auch der Erwerb von Tickets für die Fahrgäste gehört. Anhand dessen wird im weiteren Verlauf dieser Arbeit eine umfassende Sicherheitsanalyse durchgeführt.

Das vierte Kapitel (Security Engineering) konkretisiert den technischen Entwurf für das zuvor vorgestellte Szenario. Es werden Sicherheitsanforderungen aufgestellt, welche im Folgenden innerhalb der Realisierung umgesetzt werden sollen. In einer abschließenden Analyse findet eine Bewertung statt, ob die zuvor aufgestellten Sicherheitsanforderungen eingehalten werden konnten und mit welchem Aufwand dies möglich ist.

Das letzte Kapitel fasst die Ergebnisse dieser Arbeit zusammen und bietet einen Ausblick auf zukünftige Entwicklungen.

2 Grundlagen

2.1 Einleitung

Dieses Kapitel stellt einige Grundlagen dieser Arbeit vor und dient als Einstieg in die Thematik. Es werden zunächst mobile Endgeräte vorgestellt, die als zugrundeliegende Hardware Plattform dienen und auf denen Betriebssysteme wie *Android* typischerweise eingesetzt werden. Anschließend wird auf das Thema Sicherheit eingegangen. Es ist die zu analysierende Eigenschaft des Systems und soll den Schwerpunkt in dieser Arbeit darstellen. Danach wird auf eCommerce als den in dieser Arbeit verwendeten Anwendungsbereich näher eingegangen, sowie auch auf dessen Besonderheiten bezüglich der Sicherheitsanforderungen. Es folgt eine Einführung in die grundlegenden Konzepte und Eigenheiten von *Android*, die essentiell für das weitere technische Verständnis der Plattform und der späteren Anwendungsentwicklung für diese sind.

2.2 Mobile Geräte

Android ist als eine Software-Plattform für mobile Endgeräte entwickelt worden. Dazu gehören hauptsächlich Handhelds, wie Smartphones, Mobiltelefone, Communicator, PDAs und Pocket PCs. Eine geeignete Definition zu dieser Geräteklasse liefert Jörg Roth [Roth05]:

„Handhelds sind kleine mobile Computer, die gegenüber Standardcomputern über eine sehr reduzierte Leistungsfähigkeit verfügen. Sie sind so klein, dass sie in einer Hand gehalten und im gehaltenen Zustand bedient werden können. Darin unterscheiden sie sich wesentlich von mobilen Standardcomputern, die in der Regel auf einer ebenen Unterlage aufgestellt werden müssen.“

Damit ist die in dieser Arbeit zu untersuchende Geräteklasse größtenteils eingegrenzt. Größere Geräte wie Netbooks und PCs, für die es auch *Android*-Portierungen[1] gibt, sind damit explizit ausgeschlossen.

Bei mobilen Geräten handelt es sich in der Regel um Smartphones, die laut Jörg Roth [Roth05] die Funktionalität von Mobiltelefonen und PDAs in einem Gerät vereinen. Im weiteren Verlauf meiner Arbeit wird auf diese Geräteklasse zwecks besserer Lesbarkeit nur noch als **mobile Geräte** oder **Endgeräte** bezeichnen und keine weiteren Differenzierungen zwischen diesen vornehmen.

Mobile Geräte werden im Gegensatz zu Desktop-Rechner Systeme gewöhnlicherweise nur von einer Person verwendet. Eine gemeinsame Nutzung des Gerätes durch mehrere Anwender findet in der Regel nicht statt.

2.2.1 Funktionalität

Mobile Geräte besitzen typischerweise eine Menge an Funktionen. Dazu gehören unter anderem:

- Kommunikationsfunktionen (Telefonie, SMS, MMS, IM, E-Mail, WWW, Fax)
- Personal Information Manager (kurz: PIM; Kontakte, Termine, Aufgaben, Notizen)
- Multimedia-Funktionen (Audioplayer, Radio, Videoplayer, Bildbetrachtung, Diktiergerät)
- Kamera (Fotos und Videos)
- Lokalisation und Navigation (z.B. mittels GPS oder W-LAN)
- Mobiler Internetzugang (per GSM/GPRS/UMTS etc.)
- Text-, Bild, u. Tabellenverarbeitung, E-Book-Reader

[1] Live*Android*: http://code.*Google*.com/p/live-*Android*

- Datenspeicherung (über Schnittstellen wie USB, Bluetooth, etc.)
- Weitere Anwendungen (Taschenrechner, Wecker, Spiele)

Dies zeigt, wie vielfältig mobile Geräte eingesetzt werden können. Es gilt also für die zugrundeliegende Plattform, die Nutzung dieser vielen verschiedenen Funktionalitäten durch den Anwender zu ermöglichen, sämtliche Ressourcen zu verwalten, sowie die Ausführung der verschiedenen Anwendungen zu steuern.

Durch diese universelle Nutzung der verschiedenen Funktionalitäten eines solchen mobilen Geräts fallen eine Menge an persönlichen Daten an, die es zu schützen gilt. Das Betriebssystem sollte die Anwender hierbei in einem gewissen Rahmen unterstützen, damit insbesondere auch technisch weniger versierte Anwender mobile Geräte sicher verwenden können.

2.2.2 Einschränkungen

Allerdings gibt es speziell bei mobilen Geräten einige Einschränkungen zu beachten, die besondere Anforderungen an das Betriebssystem stellen, welche sich in der Leistungsfähigkeit, einem erhöhten Diebstahl-Risiko oder vergleichsweise hohen Nutzungskosten bemerkbar machen.

Ressourcen

Ressourcen wie Rechenkapazität, Speicherkapazität, und Energie sind auf mobilen Geräten um ein vielfaches begrenzter verfügbar als auf aktuellen Desktop-Rechnern.

Jörg Roth [Roth05] beschreibt die Leistungsfähigkeit mobiler Endgeräte relativ zu denen von Desktop-Computern: Eine CPU hat dabei eine um den Faktor 5 verminderte Taktrate, teilweise sogar noch größer. Anstelle einer Festplatte werden Flash-Speicher eingesetzt, deren Kapazität typischerweise um den Faktor 50-100 verringert ist.

Begründet durch den mobilen Einsatz, der nur drahtlos erfolgen kann, verfügen mobile Geräte über eine integrierte Energieversorgung, die nicht unbegrenzt zur Verfügung steht, sodass im Gegensatz zu verkabelten Geräten auf eine sparsame Nutzung dieser Ressource zu achten ist. Jörg Roth [Roth05] beschriebt dies wie folgt:

„Energie ist für mobile Endgeräte nur sehr beschränkt verfügbar. Diese Beschränkung beeinflusst auch die Interaktion mit mobilen Endgeräten. Geräte müssen sich so häufig wie möglich im energiesparenden Modus befinden oder ausgeschaltet werden. Anwendungen sollten auf umfangreiche Berechnungen verzichten.“

Drahtlose Kommunikation

Für mobile Geräte stehen verschiedene Möglichkeiten zur Informations- und Datenübertragung mit anderen Systemen zur Verfügung. Um die Mobilität der Geräte nicht einzuschränken, muss es sich dabei zwangläufig um drahtlose Technologien handeln, die jedoch einige Einschränkungen im Vergleich zu kabelgebundenen Lösungen mit sich bringen. Dazu gehören eine verminderte Verfügbarkeit, geringere Geschwindigkeit und evtl. zusätzlich anfallende Kosten bei der Verwendung moderner Kommunikationstechnologien.

Für mobile Geräte gibt es verschiedene Möglichkeiten zur Informations- und Datenübertragung mit anderen Systemen: Verbindungen über das Mobilfunknetz, drahtlose lokale Netzwerke, sowie im Nahbereich die *Wireless Personal Area Networks (WPAN)* Einen Überblick über einige der Technologien bietet Tabelle 2.1.

Netzwerk	Technologie / Standard
Mobilfunknetz	UMTS, HSDPA, EDGE, GPRS, GSM
Drahtlose lokale Netzwerke	Wireless LAN: Wi-Fi / IEEE-802.11
Wireless Personal Area Network (WPAN)	Bluetooth Infrarot
Sonstige	Near Field Communication (NFC) RFID Barcodes

Tabelle 2.1: Drahtlose Kommunikationstechnologien

Diese unterscheiden sich in der Verfügbarkeit, Reichweite und Bandbreite. Zudem sind nicht alle mobilen Endgeräte mit allen mobilen Technologien ausgestattet.

Eine Möglichkeit zur mobilen Nutzung von Diensten über das Internet stellt die Verbindung über ein drahtloses lokales Netzwerk dar, bei dem die IEEE 802.11 Standards die wohl bekanntesten und verbreitetesten Vertreter dieser Technologien darstellen. Ein solches Netzwerk ist jedoch nur im unmittelbaren Umkreis des drahtlosen Zugangspunktes verfügbar, der je nach örtlichen Begebenheiten und verwendeter Technik ca. 10-100 Meter beträgt.

Beim Einsatz mobiler Technologien in eCommerce Szenarien gilt es auch finanzielle Aspekte zu berücksichtigen[1]: Zu unterscheiden sind hier öffentliche und nicht-öffentliche Netzwerke. Öffentliche Netzwerke werden dabei von einem Hotspot-Betreiber meist nur kostenpflichtig zum Zugang ins Internet bereitgestellt. Nicht-Öffentliche Netzwerke oder private Netzwerke, wie sie beispielsweise Zuhause oder im Büro verwendet werden, können in der Regel kostenfrei genutzt werden und befinden sich in einem meist gesicherterem Umfeld, sodass diese gewöhnlicher weise bevorzugt verwendet werden.

Lokale Netzwerke bieten in der Regel Geschwindigkeitsvorteile im Gegensatz zum zuvor vorgestellten Mobilfunknetz. Allerdings sind drahtlose lokale Netzwerke nicht überall verfügbar. Ein ubiquitärer[2] Zugang zum Internet ist mit ihnen nicht möglich. Es bedarf daher alternativer Techniken, um die gewünschte Ubiquität von Diensten zu ermöglichen, die über das Internet angeboten werden.

Datenverbindungen über das Mobilfunknetz bieten eine erhöhte Verfügbarkeit, da sie als einzige drahtlose Technologie einen flächendeckenden Zugang zum Internet bieten können. Nachteilig daran zu bewerten sind die vergleichsweise geringe Bandbreite zu aktuellen DSL Anschlüssen und die relativ hohen anfallenden Kosten, die bei Nutzung dieser Verbindungen anfallen und für gewöhnlich an den Mobilfunkanbieter (engl. *Mobile Network Operator*, kurz: MNO) zu zahlen sind. Bei der Verwendung mobiler Datenverbindungen ins Internet ist daher auf einen sparsamen Gebrauch dieser zu achten, wobei auch die Abrechnungsart - nach Verbindungszeit oder nach übertragener Datenmenge - berücksichtigt werden muss.

[1] Der Preis beeinflusst die Kunden in ihrer Entscheidung für ein Produkt oder eine Dienstleistung
[2] ubiquitär bedeutet allgegenwärtig

Im Nahbereich stehen zusätzlich noch die *Wireless Personal Area Networks* (WPAN) zur Verfügung. Die Bluetooth-Technologie, mit der viele aktuelle Endgeräte ausgestattet sind, ist diesen zuzuordnen. Aufgrund der geringen Reichweite von nur wenigen Metern sind sie im praktischen mobilen Einsatz eher ungeeignet. Sie können jedoch als eine drahtlose Alternative zu kurzfristig genutzten provisorischen, kabelgebundenen Verbindung verwendet werden. Es entstehen jedoch zusätzliche Sicherheitsbedrohungen auf Grund der Drahtloseigenschaft.

Laut [Roth05] lässt die Funkkommunikation nur wesentlich niedrigere Datenraten zu, als dies bei einer drahtgebundenen Kommunikation möglich ist. Dies liege einerseits an den eingesetzten Frequenzbändern, bei denen geringere Bandbreiten zur Datenübertragung zur Verfügung stehen, und andererseits an der Teilung des Medium unter viel mehr Benutzern. Höhere Übertragungsraten durch höhere Frequenzen benötigen mehr Energie. Dies widerspreche *„aber dem mobilen Einsatz, da die Energie aus Batterien oder Akkus mit beschränkter Kapazität gewonnen wird“*.

Interessant in diesem Zusammenhang ist die Frage, inwieweit die auf Desktop-Rechnern etablierten Verfahren zum Erreichen der Sicherheitsziele auf mobilen Geräten umgesetzt werden können. Berechnungen aus der Kryptographie sind verhältnismäßig rechenintensiv und sollten auf solch leistungsschwacher Hardware dennoch eingesetzt werden können. Zu beachten sind die verfügbaren Ressourcen (CPU, Speicher oder Energie), von denen keine besonders beansprucht werden sollte.

Benutzerschnittstelle

Mobile Geräte verfügen im Vergleich zu stationären Computer nur über eingeschränkte Interaktionsmöglichkeiten mit dem Anwender. Die Ein- und Ausgabefähigkeiten solcher Geräte sind limitiert.

Jörg Roth [Roth05] vergleicht die Eigenschaften mobiler Geräte hinsichtlich der Bildschirmgröße, der Interaktionsrate und der Eingabemethoden: Dabei sei der Bildschirm bezüglich Größe und Auflösung limitiert und sein maximales Ausmaß durch die kleine Gehäusegröße des Endgerätes beschränkt. Hinsichtlich der Interaktionsrate würden mobile Anwendungen im Vergleich zu Desktop Anwendungen mehrmals täglich verwendet werden, jedoch meistens nur für eine kurze Dauer, weshalb die Anwendungen kurze Ladezeiten aufweisen sollten. Die von Desktop Systemen gewohnten Eingabemöglichkeiten über Tastatur und Maus seien nicht oder nur sehr stark begrenzt verwendbar, sodass nach Möglichkeit auf die Eingabe von Texteingaben verzichtet werden sollte.

Diebstahl und Verlust

Laut Dr. Thomas Jurisch [Jurisc09] ist *die „wohl bekannteste Gefahr [] der physische Verlust von mobilen Endgeräten durch Diebstahl oder Nachlässigkeit womit natürlich auch der Datenträger verloren ist.“*

Mobile Geräte sind klein, handlich und durch das ständige Mitführen - auch in fremden Umgebungen - einem erhöhtem Risiko durch Diebstahl oder sonstigem Verlust ausgesetzt.

Im Vergleich zu Desktop-Rechnern befinden sie sich nicht ständig in sicheren Räumlichkeiten, die über ihre eigenen Sicherheitsmaßnahmen verfügen, wie etwa durch Türschlösser, und bieten daher weniger Schutz vor unberechtigtem Zugriff durch Dritte.

Mobile Geräte sind durch ihre Größe und ihr Gewicht mobil einsetzbar und wechseln zusammen mit ihrem Besitzer ständig das Umfeld, was oftmals nicht besonders sicher ist, wie beispielsweise bei einem eingeschlafenen Fahrgast auf seinem nächtlichen Weg nach Hause.

Einer britischen Umfrage[1] [Flatle09] zufolge sind 4% aller Haushalte und 2% aller Mobiltelefonbesitzer innerhalb eines Jahres Opfer eines Mobiltelefon-Diebstahls geworden. Hochgerechnet entspricht dies rund 800.000 Diebstählen jährlich.

Ähnliche verlässliche Zahlen dazu in Deutschland oder der Welt sind dem Autor nicht bekannt, dürften sich jedoch in einem ähnlichen Rahmen bewegen.

Das Ausmaß dieser Gefahr geht dabei über den Verlust des Endgerätes hinaus, da auch sämtliche auf dem Gerät gespeicherten Daten bedroht sind. Eine weitere Betrachtung dieser Bedrohung folgt später in Abschnitt 2.3.3.

2.2.3 Beispiel für ein Endgerät: Das G1 von HTC

Ein Beispiel für ein solches mobiles Gerät ist das G1 von HTC, welches zudem das erste „*Android-powered mobile device*“ ist, also das erste Gerät, das werksseitig mit Android ausgeliefert wurde. Dieses wird in dieser Bachelorarbeit verwendet und ist das Zielsystem für die Entwicklung des späteren Prototyps, der zusammen mit dem Beispielszenario in Kapitel 3 vorgestellt wird.

Details über die genaue Produktspezifikation des G1 enthält Tabelle 0.1 im Anhang.

2.3 Sicherheit

2.3.1 Einleitung

Laut Claudia Eckert [Eckert08] umfasst Sicherheit die Bereiche Funktionssicherheit, Informationssicherheit, Datensicherheit und Datenschutz und beschreibt diese wie folgt:

- Funktionssicherheit (engl. s*afety*) ist die Eigenschaft eines Systems, bei dem die Ist-Funktionalität der Komponenten mit der spezifizierten Soll-Funktionalität übereinstimmt.
- Informationssicherheit (engl. *security*) ist die Eigenschaft eines funktionssicheren Systems, nur solche Zustände anzunehmen, die zu keiner unautorisierten Informationsveränderung oder -gewinnung führen.
- Datensicherheit (engl. *protection*) ist die Eigenschaft eines funktionssicheren Systems, nur solche Systemzustände anzunehmen, die zu keinem unautorisierten Zugriff auf Systemressourcen und insbesondere auf Daten führen können.
- Datenschutz (engl. *privacy)* ist gemäß des Bundesdatenschutzgesetzes (BDSG) die Fähigkeit einer natürlichen Person zur Kontrolle der Weitergabe von Informationen, die sie persönlich betreffen.

Die Funktionssicherheit eines IT-Systems ist dabei Grundlage für die Informations- und Datensicherheit. Eine Übersicht über das Thema Sicherheit verdeutlicht der McCumber Cube [Schou07] in Abbildung 2.1.

[1] zach Zahlen des *British Crime Survey* aus dem Jahr *2005/06*

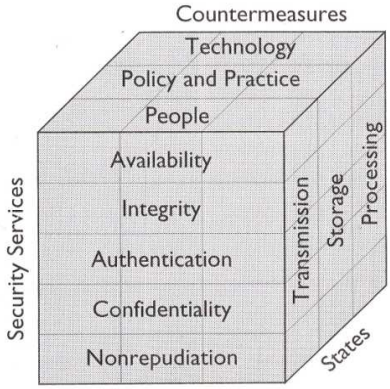

Abbildung 2.1: McCumber Cube - Extended MSR Model [Schou07]

„Ein IT-System ist ein geschlossenes oder offenes, dynamisches technisches System mit der Fähigkeit zur Speicherung und Verarbeitung von Informationen“ sagt Claudia Eckert [Eckert08]. Die darin verarbeiteten Informationen haben einen zu schützenden Wert und dieser sollte durch die Informationssicherheit gewahrt bleiben. Innerhalb eines IT-Systems werden diese in Datenobjekten repräsentiert. Diese können sich in verschiedenen Zuständen befinden:

- Im Speicher des mobilen Geräts, sowohl persistent als auch flüchtig
- In Übertragung über einen Kommunikationskanal
- In Laufzeit, während der Verarbeitung der Daten durch den Prozessor

Die Daten, die schützenswerte Informationen enthalten, müssen identifiziert und durch geeignete Maßnahmen (engl. *countermeasures*) geschützt werden.

2.3.2 Sicherheitsziele

Die zentralen Sicherheitsziele eines IT-Systems sind laut Claudia Eckert [Eckert08] Authentizität, Datenintegrität, Informationsvertraulichkeit, Verfügbarkeit, Verbindlichkeit und Privatheit. Corey Schou [Schou07] verwendet in diesem Zusammenhang die fünf *Security Services* (vgl. Abbildung 2.1): *Availability, Integrity, Authentication, Confidentiality, Nonrepudiation.*

Angelehnt an beide Autoren verwendet der Autor im Folgenden die fünf Sicherheitsziele, wie sie in Tabelle 2.2 dargestellt sind und werde diese im Einzelnen vorstellen.

Sicherheitsziel	Verwandte Begriffe
Verfügbarkeit	Erreichbarkeit
Vertraulichkeit	Privatheit, Abhörsicherheit, Anonymität, Verdecktheit, Unbeobachtbarkeit
Integrität	Datenintegrität
Authentizität	Echtheit, Glaubwürdigkeit
Verbindlichkeit	Nichtabstreitbarkeit, Zurechenbarkeit, Rechtsverbindlichkeit

Tabelle 2.2: Allgemeine Sicherheitsziele

Verfügbarkeit

Nach Corey Schou [Schou07] beschreibt Verfügbarkeit (engl. availability) die Qualität, die sicherstellt, dass eine Nutzung des Dienstes für autorisierte Benutzer möglich ist, wenn diese ihn benötigten. Dabei darf die Wahrnehmung der Berechtigungen von authentifizierten und autorisierten Subjekten nicht unautorisiert beeinträchtigt werden.

Dazu nennt Schou [Schou07] als Beispiel, dass durch den Wegfall sämtlicher Gehaltsabrechnungen in einem großen Unternehmen ein ziemliches Chaos zu erwarten sei, welches es zu verhindern gelte.

Erreichbarkeit beschreibt die Eigenschaft, dass zu einer Ressource der Kontakt aufgenommen werden kann, wenn dieser gewünscht ist.

Vertraulichkeit

Nach Claudia Eckert [Eckert08] ist **Informationsvertraulichkeit** (engl. *confidentiality*) die Eigenschaft eines Systems, die keine unautorisierte Informationsgewinnung ermöglicht. Um dies zu gewährleisten, wird die Festlegung von Berechtigungen und Kontrollen beim Zugriff auf Objekte benötigt, sodass sichergestellt ist, dass Subjekte nicht unautorisiert in Kenntnis von Informationen gelangen können.

„In informationssicheren Systemen sind Maßnahmen zu Festlegung sowie zur Kontrolle zulässiger Informationsflüsse zwischen den Subjekten des Systems erforderlich, so dass ausgeschlossen werden kann, dass Information zu unautorisierten Subjekten „durchsickert“ Das entsprechende Problem wird Confinement Problem genannt.

Zur Umsetzung der Informationsvertraulichkeit stehen dazu Zugriffskontroll- und Informationsfluss-Strategien zu Verfügung.

Es gibt verschiedene Sicherheitsziele, welche ebenfalls in Zusammenhang mit der Vertraulichkeit stehen: Privatheit, Abhörsicherheit, Anonymität, Verdecktheit und Unbeobachtbarkeit.

Privatheit (engl. *privacy*) beschäftigt sich mit dem Recht zu informationeller Selbstbestimmung, welches ein Grundbedürfnis des Menschen darstellt. Es ist in den Grundrechten verankert und wird durch Datenschutzgesetze geschützt. In der Allgemeinen Erklärung der Menschenrechte der Vereinten Nationen [UN09] wird dies wie folgt festgelegt:

“No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.”

Bei der Verarbeitung von personenbezogenen Daten kann nach Volkmar Scharf [Scharf09] das informationelle Selbstbestimmungsrecht beeinträchtigt werden, ist daher grundsätzlich genehmigungspflichtig und darf in einem ungeschützten Netz nicht durchgeführt werden.

Dazu findet unter anderem das Bundesdatenschutzgesetz [BDSG09] Anwendung:

(1) Zweck dieses Gesetzes ist es, den Einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird.

(2) Dieses Gesetz gilt für die Erhebung, Verarbeitung und Nutzung personenbezogener Daten durch 1. öffentliche Stellen des Bundes, 2. öffentliche Stellen der Länder [] 3. nicht-öffentliche Stellen, soweit sie die Daten unter Einsatz von Datenverarbeitungsanlagen verarbeiten, nutzen oder dafür erheben oder die Daten in oder aus nicht automatisierten Dateien verarbeiten, nutzen oder dafür erheben[].

Anonymität beschäftigt sich damit, dass Dienste und Ressourcen ohne Preisgabe der eigenen Identität genutzt werden können und stellt ein mehrseitiges Sicherheitsziel dar, bei dem die beteiligten Parteien individuelle Schutzziele verfolgen.

Abhörsicherheit sorgt generell für die Sicherheit gegen ein unberechtigtes Abhören von Daten während der Übertragung durch unberechtigte Dritte. Kommunikationskanäle, die durch öffentliche Netze gehen, unterliegen nicht der eigenen Kontrolle, und bieten keine Abhörsicherheit. Durch das unkenntlich machen der enthaltenen Informationen können diese dennoch geschützt werden, wie dies beispielsweise bei der Verschlüsselung der Fall ist.

Verdecktheit verbirgt die Existenz der Übertragung.

Die **Unbeobachtbarkeit** sorgt dafür, dass Dienste und Ressourcen ohne fremde Beobachtung genutzt werden können.

Integrität

Nach Claudia Eckert [Eckert08] ist **Datenintegrität** die Eigenschaft, bei der Subjekte die zu schützenden Daten nicht unautorisiert oder unbemerkt manipulieren können. Die Datenintegrität stellt sicher, dass derartige Manipulationen erkannt werden.

Weiter schreibt Eckert [Eckert08], dass „*in Umgebungen, in denen eine Manipulation a priori nicht verhindert werden kann, Techniken erforderlich sind, mit deren Hilfe unautorisierte Manipulationen a posteriori erkennbar sind*“, damit die Integritätseigenschaft eingehalten werden kann.

In einem eCommerce Szenario kommunizieren typischerweise zwei Geschäftspartner miteinander über ein unsicheres Kommunikationsmedium, auf dem die Daten von Dritten verfälscht oder manipuliert werden könnten.

Nach [Winkel04] „*ist es essentiell zu wissen, ob die Person, mit der man kommuniziert, tatsächlich diejenige ist, für die sie sich ausgibt, und dass die Daten beim Empfänger so ankommen, wie sie abgeschickt wurden: unverfälscht, nicht manipuliert, weder verkürzt noch mit Zusätzen versehen.*

Authentizität

Unter Authentizität (engl. *authenticity*) versteht man die Echtheit und Glaubwürdigkeit eines Objekts bzw. Subjekts. Sie lässt sich anhand einer eindeutigen Identität und charakteristischen Eigenschaften überprüfen, wie geheimen Informationen (z.B. Passwörter), persönlichen Besitz oder biometrischen Merkmalen.

Subjekte sind zum einen die Anwender selbst, als auch sämtliche Systemkomponenten, die in dessen Auftrag handeln können, wie z.B. Prozesse.

Laut Claudia Eckert [Eckert08] beruhen viele Anwendungen des elektronischen Handels darauf, dass die Benutzer darauf vertrauen können, mit dem authentischen Web-Server Daten auszutauschen.

Das Gerät bzw. die sicherheitskritischen Bereiche des Systems sollten vor einer Verwendung eine Authentifikation[1] des Benutzers vornehmen, damit ein unautorisierter Zugriff verhindert werden kann.

Verbindlichkeit

Nach [Eckert08] gewährleistet Verbindlichkeit bzw. Nichtabstreitbarkeit (engl. *non-repudiation*), dass es den an einer Aktion beteiligten Subjekten nicht möglich ist, im Nachhinein die Durchführung dieser Aktion abzustreiten.

[1] Eine Unterscheidung zwischen der beiden Begrifflichkeiten Authentifizierung und Authentisierung erfolgt nicht und beide Begriffe werden in dieser Arbeit synonym verwendet.

Weiter sind nach Eckert [Eckert08] derartige Verbindlichkeitseigenschaften besonders im rasant wachsendem Bereich des eCommerce von großer Bedeutung, damit die Rechtsverbindlichkeit geschäftlicher Verträge und Transaktionen garantiert werden kann.

Die **Rechtsverbindlichkeit** beschäftigt sich dabei mit der Belangbarkeit des Nutzers bezüglich seiner Verantwortlichkeiten.

Die **Zurechenbarkeit** beweist gegenüber Sendern oder Empfängern, dass dieser Daten gesendet bzw. empfangen hat.

Nach [Winkel04] *„sollte der Kontakt zwischen Menschen, die sich nicht kennen, in einem rechtssicheren Raum stattfinden und die Verbindlichkeit von Willenserklärungen soll gewährleistet werden.“*

Nach [Eckert08] können Verbindlichkeitsanforderungen durch den Einsatz digitaler Signaturen erfüllt werden.

2.3.3 Bedrohungen, Schwachstellen und Gefährdungen

Nach Claudia Eckert [Eckert08] zielt eine Bedrohung darauf ab, Schwachstellen auszunutzen, um einen Verlust der Datenintegrität, der Informationsvertraulichkeit, der Verfügbarkeit oder der Authentizität von Subjekten zu gefährden.

Das BSI [BSI08a] definiert eine Bedrohung wie folgt:

„Eine Bedrohung ist ganz allgemein ein Umstand oder Ereignis, durch den oder das ein Schaden entstehen kann. Der Schaden bezieht sich dabei auf einen konkreten Wert wie Vermögen, Wissen, Gegenstände oder Gesundheit. Übertragen in die Welt der Informationstechnik ist eine Bedrohung ein Umstand oder Ereignis, der oder das die Verfügbarkeit, Integrität oder Vertraulichkeit von Informationen beeinträchtigen kann, wodurch dem Besitzer bzw. Benutzer der Informationen ein Schaden entstehen kann. Beispiele für Bedrohungen sind höhere Gewalt, menschliche Fehlhandlungen, technisches Versagen oder vorsätzliche Handlungen. Trifft eine Bedrohung auf eine Schwachstelle (insbesondere technische oder organisatorische Mängel), so entsteht eine Gefährdung.“

Weiter definiert das BSI eine Gefährdung:

„Eine Gefährdung ist eine Bedrohung, die konkret über eine Schwachstelle auf ein Objekt einwirkt. Eine Bedrohung wird somit erst durch eine vorhandene Schwachstelle zur Gefährdung für ein Objekt.“

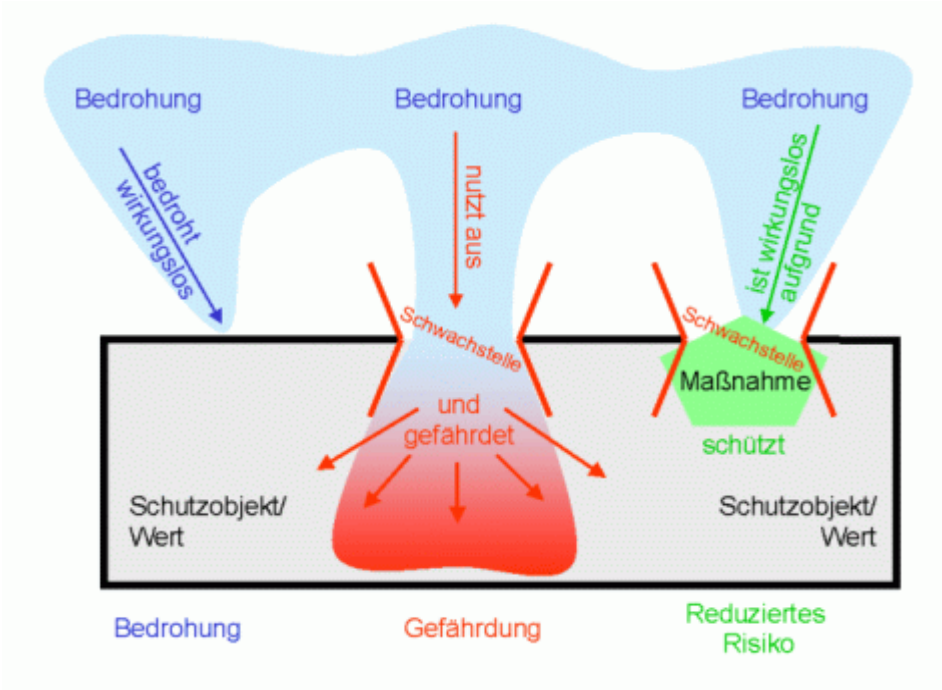

Abbildung 2.2: Bedrohungen + Schwachstellen = Gefährdungen [BSI09]

Das Security Engineering beschreibt eine Disziplin innerhalb der Informatik, bei der methodisch Sicherheit in den Entwicklungsprozess hineinfließt, um sichere Anwendungen zu konstruieren. Es umfasst Maßnahmen und Methoden, durch die die angestrebten Sicherheitsziele umgesetzt werden können.

Sicherheitskette

Wie bei einer Kette bestimmt das schwächste Glied eines IT-Systems über deren Gesamtsicherheit. Alle beteiligten Komponenten eines Informationssystems müssen bei einer Sicherheitsanalyse betrachtet werden, um Aussagen über die Sicherheit des gesamten Systems machen zu können. Befindet sich nur eine Komponente im System, die als unsicher gilt, so ist das Gesamtsystem ebenfalls als unsicher einzustufen. Daher ist bei der Sicherheitsanalyse sorgfältiges Vorgehen wichtig, damit keine möglichen Angriffsflächen auf das System übersehen werden.

2.3.4 Spezielle Bedrohungen

Physischer Verlust des Endgerätes

Zum einen gilt es den Verlust des Endgerätes generell zu verhindern und zum anderen sollten Auswirkungen im Schadenfall so gering wie möglich gehalten werden. Neben der Hardware selbst würden auch sämtliche darauf befindliche Daten verloren gehen, sodass bei vorhandener krimineller Energie schnell weitere Schäden entstehen können.

Die Verfügbarkeit, Vertraulichkeit und Integrität der auf dem Gerät gespeicherten Daten und Informationen sollte auch im Schadenfall erhalten bleiben.

Die Daten sollten daher entsprechend gesichert sein, damit sie im Falle des Verlustes nicht verloren sind. Durch Backups oder andere Replikate können die Daten im Schadensfall rekonstruiert werden. Die Informationsvertraulichkeit kann durch Techniken wie Verschlüsselung und *Remote Wiping*[1] gewahrt bleiben, damit solche Informationen für unbefugte Dritte nicht einsehbar und missbraucht werden können. Auch sollten unautorisierte durchgeführte Manipulationen entdeckt oder verhindert werden können.

Weiter ist es sinnvoll, wenn das System den rechtmäßigen Besitzer durch entsprechende Maßnahmen unterstützt, um das Gerät wiederzufinden und zum anderen die darauf befindlichen Daten zurück zu erhalten. Einen guten Überblick über diese Thematik bietet Takashi Matsunaka in *A Lightweight Approach to Protect Mobile Data* [Matsun07].

Benutzerverhalten

Es gibt Defizite im Benutzerverhalten bezogen auf die Sicherheit beim Umgang mit technischen Systemen. Dies belegen einige Studien, unter anderem von Stewart Kowalski [Kowals06] oder Jeffrey M. Stanton [Stanton04].

So fehlt den Anwendern oftmals das nötige Fachwissen, um sich richtig verhalten zu können. Es gibt in der Regel ausreichende/umfassende technische Maßnahmen, die ein Erreichen der Sicherheitsziele gewährleisten können, jedoch werden diese oftmals vom Anwender gar nicht oder nicht richtig verstanden. Die Nicht-Nutzung oder Fehlkonfigurationen sind die Folge.

So schenken einige Verbraucher zu wenig oder kein Vertrauen in die Authentifikationsmechanismen mobiler Endgeräte oder empfinden diese gar als störend für die Benutzbarkeit des Systems, sodass sie diese Instrumente nicht verwenden.

Dadurch entsteht eine zusätzliche Bedrohung für das Gesamtsystem, sodass allein durch technische Maßnahmen die Sicherheitsziele nicht mehr erreicht werden können.

Bezogen auf Schadsoftware beispielsweise kann ähnlich zur Verbreitung des HI-Virus in der Medizin auch in der Informatik mangelnde Aufklärung seitens der beteiligten Subjekte schnell zu einer Infektion des Gesamtsystems führen. Xiaoni Zhang [Zhang05] konnte dazu folgenden Sachverhalt zum Thema *Spyware* feststellen, einer speziellen Kategorie von Schadsoftware:

„The results show that privacy invasions created by spyware are not noticed adequately by respondents. Despite experience with computers and the Internet, many consumers are neither aware of spyware intrusion nor knowledgeable of spyware removal tools available to them. Due to this lack of knowledge, they have not taken adequate steps to protect themselves. This calls for an urgent need for consumer education. “

Daher sollte bei Benutzern generell mehr Sensibilität in sicherheitskritischen Fragestellungen geweckt werden. Dies ist durch entsprechendes Software-Design möglich, sodass Anwender bei der Verwendung des Systems auf Fehler explizit hingewiesen werden können. Insbesondere technisch unversierte Anwender sind sich den möglichen Gefahren oftmals nicht bewusst, und sollten daher bei sicherheitskritischen Entscheidungen durch die Plattform weitestgehend unterstützt werden.

[1] *Remote Wiping* ermöglich das Löschen der Daten durch einen entfernten Zugriff

Schadsoftware

Schadsoftware (engl. *Malware)* enthält beabsichtigte (und meist auch böswillige) Funktionen, die vom Anwender unerwünscht und als schädlich eingestuft werden. Die bekanntesten Vertreter von Schadsoftware sind Viren, Würmer, sowie Trojanische Pferde, deren schädliche Funktionalität dem Anwender hinter einer nützlichen Funktionalität zunächst verborgen bleibt.

Programmschwachstellen

Claudia Eckert [Eckert08] zufolge können Programmschwachstellen als Folge einer nachlässigen Implementierung entstehen. Die daraus resultierenden Schwachstellen können beispielsweise von Buffer-Overflow-Angriffen ausgenutzt werden. Bei einem solchen Angriff werden absichtlich zu große Daten in einen dafür zu kleinen Speicherbereich geschriebenen:

„Der Ansatzpunkt für entsprechende Angriffe sind Programme, in denen Daten in einen Bereich einer Variablen fester Länge, z.B. in einen String oder ein Feld fester Länge, eingelesen oder kopiert werden, ohne dass das Programm prüft, ob die kopierte Eingabe überhaupt in den bereitgestellten Bereich passt. [] Durch das Schreiben einer zu großen Datenmenge in den Bereich des Puffers wird dieser zum Überlauf gebracht.“

Auf diese Weise kann Schadcode zunächst eingeschleust werden, um danach ihn anschließend zur Ausführung zu bringen.

Man in the Middle

Ein *Man-in-the-Middle* (MITM) steht zwischen zwei Kommunikationspartnern und kann den Nachrichtenverkehr abhören und verändern. Einem Angreifer bietet dies eine gute Ausgangssituation und stellt eine Bedrohung für das System und die verarbeiteten Informationen dar. Jean-Claude Rosenthal [Rosent04] beschreibt diese Angriffsmethode wie folgt:

„Diese Angriffsmethode beruht auf der Möglichkeit während des Verbindungsprozesses Nachrichten von der Serverseite abzufangen und diese mit seinem eigenen Zertifikat an den Client weiterzuleiten. Dieses gefälschte Zertifikat sieht für den Client legitim aus. So kann der Angreifer die komplette Datenübertragung zwischen Client und Server verfolgen. Eine Möglichkeit dies zu verhindern ist das Zwischenschalten einer vertrauenswürdigen dritten Partei und deren Zertifikat.“

2.3.5 Techniken zur Bedrohungsabwehr

Laut Eckert [Eckert08] können zur Abwehr von Bedrohungen drei verschiedene Ansätze verfolgt werden:

- Verhindern
- Erkennen und Abwehren
- Begrenzen des Schadens

Diese drei Ansätze werden im Folgenden detaillierter vorgestellt:

Verhindern von Bedrohungen

Zum Ersten soll das Entstehen von Bedrohungen gar nicht erst zugelassen werden, indem es durch bestimmte Techniken verhindert wird, wie z.B. durch das Verwenden von stark typisierten Programmiersprachen, Code-Verifikation, Zugriffsbeschränkungen durch systembedingte Regeln und Maßnahmen zur Sensibilisierung des Sicherheitsbewusstseins, sowohl beim Anwender, als auch beim Entwickler.

Dazu schreibt Eckert [Eckert08]: *„Klar ist, dass Vermeidungstechniken nach dem Motto „Sicherheit per Design“ die konzeptuell befriedigendste Lösung darstellen, wir aber heute in der Praxis der System-Konstruktion hiervon noch sehr weit entfernt sind.“*

Bedrohungen Erkennen und Abwehren

Zum zweiten sollen mögliche Bedrohungen rechtzeitig erkannt und abgewehrt werden können. Beispielsweise können durch statische Analysen des Programmcodes mögliche Buffer-Overflows erkannt und korrigiert werden. Aber auch bekannte Techniken wie der Einsatz von Viren-Scannern, Firewalls und *Intrusion Detection Systeme* (IDS) gehören dazu.

Nachdem eine Identifizierung[1] der Schadsoftware stattgefunden hat, wird veranlasst, dass diese umgehend entfernt wird, um die Sicherheit des Gesamtsystems nicht zu gefährden.

Zunächst einmal muss jedoch Kenntnis der ungewünschten Programmaktivitäten erlangt werden. Handelt es sich bei der Schadsoftware beispielsweise um ein Trojanisches Pferd, so kann eine Erkennung durch den Benutzer allein sehr schwierig werden, da das Programm für den Benutzer die gewünschte Funktion augenscheinlich erfüllt. Entsprechende technische Systeme wie Virenscanner könnten auch solche Schadsoftware erkennen.

Virenscanner können bereits vor Ausführung der Anwendung aktiv werden, um Bedrohungen rechtzeitig, d.h. bevor ein Schaden eintritt, mittels geeigneter Maßnahmen abzuwehren.

Schaden begrenzen

Zum dritten gibt es Techniken, die die Auswirkungen von eingetretenen Bedrohungen begrenzen sollen, zu der auch das *Sandboxing* gehört. Dadurch werden verschiedene Anwendungen gegeneinander abschottet und so mögliche Bedrohungen innerhalb der Anwendung isoliert.

Claudia Eckert [Eckert08] definiert eine *Sandbox* als eine Ausführungsumgebung für Code in einen isolierten und isolierenden Speicherbereich, innerhalb dessen der Code freien Zugriff besitzt, aber Zugriff auf Objekte außerhalb des Bereichs kaum oder nur unter sehr strengen Kontrollen möglich sind.

2.3.6 Technische Maßnahmen

Zugriffskontrolle

Um die Schutzziele Verfügbarkeit, Vertraulichkeit und Integrität zu gewährleisten, ist zunächst die Festlegung von Zugriffsrechten erforderlich. Zugriffsrechte autorisieren Subjekte zur Verwendung von Objekten, wie z.B. Daten, und beschreiben, wie diese genutzt werden dürfen. Beim Zugriff auf Objekte durch Operationen müssen diese Berechtigungen überprüft werden, sodass nur autorisierte Zugriffe möglich sind.

In Systemen, die nicht vollständig unter eigener Kontrolle stehen, können keine Zugriffskontrollmechanismen verwendet werden. Anstelle dieser treten dort in der Regel Verfahren aus der Kryptographie, wie Verschlüsselung, digitale Signaturen oder kryptographisch sichere Hashing Algorithmen.

[1] Durch den Benutzer manuell oder durch technische Sicherheitsmaßnahmen

Kryptographische Verfahren

Laut Claudia Eckert [Eckert08] werden kryptographische Verfahren in unterschiedlichen sicherheitsrelevanten Bereichen benötigt und dienen zur bloßen Geheimhaltung durch Unkenntlichmachung der Daten, zur Identifikation und Authentifikation, sowie zur Gewährleistung der Datenintegrität und der Verbindlichkeit.

Laut Jörg Roth [Roth05] müssen in drahtlosen Netzen Sicherheitsmechanismen eingesetzt werden, um einen Schutz gegen Mithören bieten zu können, wie auch in vergleichbaren drahtgebundenen Netzen. Dies könne grundsätzlich zwar nicht verhindert werden, jedoch kann durch die Verschlüsselung das Auswerten von Nutzdaten verhindert werden.

Verschlüsselung dient dem Zweck der Geheimhaltung von Informationen. Mittels kryptographischer Verfahren werden dabei Klartexte in Kryptotexte (Verschlüsselung) oder vice versa (Entschlüsselung) überführt. Die Geheimhaltung der Informationen kann dadurch gewährleistet werden, dass nur die zugriffsberechtigen Subjekte Kenntnis des Schlüssels haben.

Dabei wird zwischen symmetrischen Verfahren, die einen geheimen Schlüssel, und asymmetrischen Verfahren, die separate Schlüssel für beide Richtungen verwenden, unterschieden.

Als ein synchrones Verschlüsselungsverfahren ist AES nennenswert. Es wird vielfach eingesetzt und bietet eine Sicherheit nach dem heutigen Stand der Technik. Zudem ist es frei verfügbar. Laut Claudia Eckert [Eckert08] *„gibt es zurzeit keinen praktikablen Angriff auf AES“* und *„wirklich erfolgreiche Angriffe auf die algebraische Struktur, die nicht nur theoretische Bedeutung haben, sind bisher nicht bekannt.“*

Das RSA-Verfahren ist laut Eckert [Eckert08] das in der Praxis am häufigsten eingesetzte asymmetrische Kryptosystem. *„Es basiert auf dem Faktorisierungsproblem großer, natürlicher Zahlen“* und kommt in der Praxis *„sowohl zum Verschlüsseln als auch zum sicheren Austausch von Kommunikationsschlüsseln (z.B. von AES-Schlüsseln) sowie zur Erstellung und Überprüfung digitaler Unterschriften zum Einsatz.“*

Das *One-Time-Pad* Verfahren gilt als absolut sicher. Bei diesem Chiffrierverfahren wird ein Klartext mit einer gleichlangen zufällig erzeugten Schlüsselfolge verknüpft. Problematisch hierbei sind jedoch der Austausch des Schlüssels und die Erzeugung einer absolut zufälligen Schlüsselfolge.

Netzwerkprotokoll

Zur Übertragung von Daten über das Internet steht das *Transmission Control Protokoll* (TCP) zur Verfügung. Es bietet bereits einige grundlegende Funktionalitäten, die auch für die Sicherheit von Bedeutung sind:

Laut Jörg Roth [Roth05] stellt es eine zuverlässige Ende-zu-Ende Verbindung bereit, ohne sich mit Fehlern und Engpässen auf dem Übertragungskanal befassen zu müssen und stellt sicher, dass jede Nachricht korrekt beim Empfänger ankommt, sowie auch in der korrekten Reihenfolge.

Aufsetzend auf TCP kann durch das Sicherungsprotokoll *Secure Sockets Layer* (SSL) die Authentizität der Kommunikationspartner, sowie die Vertraulichkeit und Integrität von Daten auf dem Kommunikationswege, erfüllt werden. Zunächst authentifizieren sich beide unter Verwendung von zertifikatsbasierten asymmetrischen Verfahren. Danach können Daten über die performanteren symmetrischen Verfahren verschlüsselt transportiert werden, wodurch die Vertraulichkeit gewahrt bleibt. Die Integrität wird mit *Message Authentication Codes* (MAC) gewährleistet. (vgl. [Eckert08])

Digitale Signaturen

Digitale Signaturen werden als Äquivalent zu einer handschriftlichen Unterschrift eingesetzt und basieren auf asymmetrischen kryptographischen Verfahren.

Laut Claudia Eckert [Eckert08] gewährleisten sie die zweifelsfreie Identifikation einer Person, die Nicht-Wiederverwendbarkeit der Signatur, die Unveränderbarkeit des signierten Dokumentes und die Verbindlichkeit der Signierung, sofern die folgenden Voraussetzungen eingehalten werden:

- Der private Schlüssel nicht kompromittiert wurde und sich nur im Besitz der signierenden Person befindet
- Eine Public-Key-Infrastruktur als Vertrauensmodell verwendet wird, um die Herkunft des zugehörigen öffentlichen Schlüssel zu bestätigen
- Die verwendeten Schlüssel nur bei der digitalen Signatur eingesetzt werden, um die Angriffsfläche nicht zu vergrößern

Authentifikation

Durch Authentifikationsverfahren, bei dem sich die zugreifenden Subjekte eindeutig identifizieren, kann die Authentizität sichergestellt werden. Insbesondere bei einer Kommunikation mit entfernten Geschäftspartnern über eine Datenverbindung ist dies keinesfalls trivial, da sich beide Parteien typischerweise nicht sehen können, geschweige denn überhaupt kennen.

Durch den Einsatz von kryptographischen Verfahren, wie digitalen Signaturen und *Message Authentication Codes* (MAC), die als Nachweis über die Identität dienen, kann die Authentizität gewährleistet werden. Diese Verfahren basieren auf öffentlich herausgegebenen Schlüsseln, deren Echtheit über digitale Zertifikate bestätigt werden kann.

Nach Eckert [Eckert08] wird mit einem solchen Zertifikat *„eine digitale Bescheinigung über die Zuordnung eines öffentlichen Signierschlüssels zu einer natürlichen oder juristischen Person ausgestellt.“*.

Als Herausgeber der Zertifikate wird dazu eine unabhängige und vertrauenswürdige dritte Instanz verwendet: Die *Certification Authority* (CA, engl. Zertifizierungsstelle). Diese sind Teil einer Public-Key-Infrastruktur (PKI), wodurch deren eigene Authentizität gewährleistet ist, und sind für die Ausstellung, Verteilung und Prüfung der Zertifikate zuständig.

2.3.7 Schutzbedarfsermittlung

Ziel einer Schutzbedarfsermittlung ist es festzustellen, wie groß der Schutzbedarf bezüglich der Sicherheitsziele für eine Anwendung oder eine ganze IT-Landschaft ist, sowie von sämtlichen Informationen, die von ihr verarbeitet werden. Die Sicherheitsziele umfassen dabei Vertraulichkeit, Integrität, Verfügbarkeit, Authentizität, Verbindlichkeit.

Nach Volkmar Scharf [Scharf09] müssen sich *„Art und Umfang der Schutzmaßnahmen für ein IT-System [] an dem Schutzbedarf der IT-Anwendungen bzw. Daten orientieren, die auf dem betreffenden IT-System verarbeitet werden. Als Richtmaß gilt hier das zu erwartende Schadensszenario, das eintreten kann, sollte auf dem IT-System eingebrochen und dieses missbraucht werden. Die Schäden könnten beispielsweise Verstöße gegen Verträge, Beeinträchtigung der Aufgabenerfüllung, negative Außenwirkung oder finanzielle Einbußen sein.*

Im Grundschutzhandbuch des BSI [BSI08a] werden drei Schutzbedarfskategorien für die Kategorisierung verwendet:

- **Normal:** Die Schadensauswirkungen sind begrenzt und überschaubar
- **Hoch:** Die Schadensauswirkungen können beträchtlich sein.
- **Sehr hoch:** Die Schadensauswirkungen können ein existentiell bedrohliches, katastrophales Ausmaß erreichen.

Und klassifiziert verschiedene Schadensszenarien:

- Verstöße gegen Gesetze, Vorschriften oder Verträge
- Beeinträchtigungen des informationellen Selbstbestimmungsrechts,
- Beeinträchtigungen der persönlichen Unversehrtheit
- Beeinträchtigungen der Aufgabenerfüllung
- negative Außenwirkung oder
- finanzielle Auswirkungen.

Weiter werden Empfehlungen bei der Wahl der Schutzbedarfskategorie für die betreffenden Schadensszenarien gegeben (vgl. [BSI08a] oder auch [Eckert08]).

2.4 eCommerce

eCommerce steht für „electronic Commerce“ und umfasst jeglichen Handel, der elektronisch stattfindet. Michael Merz [Merz02] liefert dazu eine eher allgemeine Definition:

„Electronic Commerce ist die Unterstützung von Handelsaktivitäten über Kommunikationsnetze“

Merz liefert des Weiteren auch eine speziellere Definition, die auf konkrete Anwendungsbereiche näher eingeht:

„Electronic Commerce ist der Einsatz von Kommunikationsprotokollen, Sicherheitsinfrastrukturen, digitalem Geld, Electronic Shopping-Malls, elektronischen Datenaustausch, Smart Cards, mobilen und/oder intelligenten Agenten, Verhandlungsprotokollen und -strategien, elektronischen Notaren, Zertifizierungsautoritäten, interorganisatoralem Workflow Management, elektronischen Verträgen und vielen weiteren Technologien zur Anbahnung und Durchführung von Handelstransaktionen im Internet.“

Unter dem Schlagwort eCommerce versteht [Lepsch00] unter anderem folgende Punkte:

- Finanzdienstleistungen wie Internet-Banking und die Verwendung von elektronischem Geld
- Elektronischer Handel mit Informationen, wie z.B. Online-Verlage und Bibliotheken, Börseninformationen, Geldspiele, Werbung, Hotel und Flugbuchungen
- Vertrieb von Software über das Internet
- Versandhandel im Internet
- Handel zwischen Firmen (*Business to Business*)
- Elektronische „Behördengeschäfte“, wie z.B. Steuererklärungen
- Handel von Dienstleistungen im Internet
- Internetauktionen

Weiter lässt sich EC anhand der beteiligten Transaktionspartner in verschiedene Bereiche kategorisieren: *Business to Business* (B2B), *Business to Consumer* (B2C), oder auch *Business to Administration* (B2A), womit staatliche Stellen gemeint sind.

2.4.1 Mobile Commerce

Mobiltelefone und andere mobile Endgeräte entwickeln sich zunehmend zu Plattformen, die für viele kommerzielle Anwendungen verwendet werden können. Solche mobilen Geräte eigenen sich gut als ein neuartiger Vertriebsweg für von Unternehmen an Konsumenten gerichtete Angebote (Business to Consumer: B2C) als Ergänzung zu den herkömmlichen eCommerce Anwendungen, die meist über einen Webbrowser von einem kabelgebundenen Computer aus aufgerufen und bedient werden. So gibt es zunehmend mehr Unternehmen, die Handys und andere mobile Geräte als Vertriebsplattform für ihre Produkte entdeckt haben.

Nach Klaus Turowski *[Turows04]* bezeichnet Mobile Commerce (MC) *„jede Art von geschäftlicher Transaktion, bei der die Transaktionspartner im Rahmen von Leistungsanbahnung, Leistungsvereinbarung oder Leistungserbringung mobile elektronische Kommunikationstechniken (in Verbindung mit mobilen Endgeräten) einsetzten.“*

2.4.2 Mobile Ticketing

Mobile Ticketing beschäftigt sich damit, dass übliche Belege, wie beispielsweise Eintrittskarten oder Fahrscheine, durch eine auf einem mobilen Gerät gespeicherte digitale Information ersetzt werden. Dabei wird dem Kunden nach einer Bestellung ein digitales Ticket auf sein mobiles Gerät übertragen.

Für die Ticketbestellung und -auslieferung stehen dabei verschiedene Verfahren zur Verfügung, wie per Anruf, per SMS, MMS oder über eine mobile Datenverbindung. Die Abrechnung der Dienstleistung erfolgt dabei über mobile Zahlungsmethoden.

Fahrkartenkontrolle

Beim Mobile Ticketing muss gewährleistet werden, dass eine Prüfung auf Gültigkeit der Tickets durch das ÖPV-Unternehmen stattfinden kann, wie man es von einer herkömmlichen Fahrkartenkontrolle kennt. Die Fahrscheine sind dabei zwingend zu prüfen, um Schwarzfahren und weiterem Betrug durch Kunden entgegenzuwirken. Dies geschieht entweder beim Betreten oder Verlassen des fahrkartenpflichtigen Bereiches, beim Einsteigen in das Fahrzeug, oder durch Fahrkarten-Kontrolleure, die auf der Strecke prüfen. Ziel der Kontrolle aus Sicht des Betreibers ist nach Thomas Mügge [Mügge07] das Prüfen des Tickets auf Echtheit, damit Fälschungen durch Duplizieren, Nachbilden oder Verändern erkannt werden können.

Kontrollverfahren

Dazu stehen verschiedene Verfahren zur Kontrolle zur Verfügung. Dr. Walter Leitner [Leitne07] unterscheidet hier zwischen Sichtkontrolle, Onlinekontrolle und Offlinekontrolle:

Eine **Sichtkontrolle** durch den Kontrolleur sollte wie bei einem Papierticket durchgeführt werden können. Laut Thomas Mügge [Mügge07] kann dies durch die Verwendung von sogenannten Kontrolleurskurzcodes in Verbindung mit Kontrollmedien und zeitabhängigen Codewörtern gewährleistet werden.

Bei der **Onlinekontrolle** wird eine Verbindung zum Rechenzentrum des Betreibers hergestellt, sodass dort anhand einer eindeutigen Ticket-ID abgefragt werden kann, ob dieses Ticket gültig ist. Dazu muss sichergestellt sein, dass die hierfür erforderliche Onlineverbindung jederzeit zur Verfügung steht. Problematisch ist dies beispielsweise in Tunneln, in denen keine mobilen Datenverbindungen über das Mobilfunknetz zur Verfügung stehen.

Des Weiteren gibt es die Möglichkeit der **Offlinekontrolle**, bei der das Ticket in maschinenlesbarer Form (wie z.B. 2D-Barcodes oder Signaturen) auslesbar ist. Diese ist personengebunden und das Ticket ist nur in Verbindung mit einem entsprechenden Kontrollmedium (wie z.B. Personalausweis) gültig. Vorteil dieses Verfahrens ist, dass keine weitere Kommunikationsinfrastruktur benötigt wird, bei der eine lückenlose Abdeckung mit drahtlosen Internetverbindungen im Gesamtnetz des Verkehrsunternehmens garantiert sein müsste.

Bei allen Verfahren gilt es zu beachten, dass die benötigte Zeit für eine Fahrscheinkontrolle beschränkt ist, damit einer Kontrolle aller Fahrgäste in einem vollen Fahrzeug durchgeführt werden kann.

Pilotprojekte

Es gibt derzeit (Stand: September 2009) einige laufende Pilotprojekte für den Einsatz von *Mobile Ticketing* im ÖPV. Das „HandyTicket“ ist ein Projekt der Mitgliedsunternehmen vom Verband Deutscher Verkehrsunternehmen (VDV)[1]. Weiter existiert mit dem sehr ähnlichen Namen das „Handy-Ticket“, sowie „touchandtravel“ von der Deutschen Bahn[2].

2.4.3 Mobile Payment

Key Pousttchi [Poustt03] definiert *Mobile Payment* (engl. für „mobiles Bezahlen“) wie folgt:

„Mobile Payment bezeichnet diejenige Art der Abwicklung von Bezahlvorgängen, bei der im Rahmen eines elektronischen Verfahrens mindestens der Zahlungspflichtige mobile Kommunikationstechniken (in Verbindung mit mobilen Endgeräten) für Initiierung, Autorisierung oder Realisierung der Zahlung einsetzt.“

Für die Nutzung einer kostenpflichtigen Dienstleistung durch den Kunden gibt es verschiedene Abrechnungsmodelle. Dazu werden in der Regel dritte Zahlungsdienstleister hinzugezogen, die als Anbieter mobiler Bezahlverfahren fungieren. Eine gute Übersicht hierzu liefert Dietmar Wiedemann [Wiedem08]. Als die wichtigsten Anbieter für solche *Mobile Payment Service Provider* (MPSP) nennt er Mobilfunkanbieter und Finanzdienstleister, zu denen Banken und Kreditkartengesellschaften gehören. Als weitere Gruppe nennt er spezialisierte Intermediäre, deren Kerngeschäft der Betrieb eines mobilen Bezahlverfahrens darstellt.

Der einfachste und bequemste Weg der Zahlung stellt für den Kunden meist die Abrechnung als Mehrwertdienst über die Rechnung des Mobilfunkanbieters dar. Es erfordert lediglich Identifikation des Kunden über seine Telefonnummer. Ein Geschäftsverhältnis zwischen dem Kunden und dem Netzbetreiber besteht bereits, sodass der restliche Zahlungsausgleich auf B2B-Ebene mit dem Anbieter der genutzten Dienstleistung stattfinden kann.

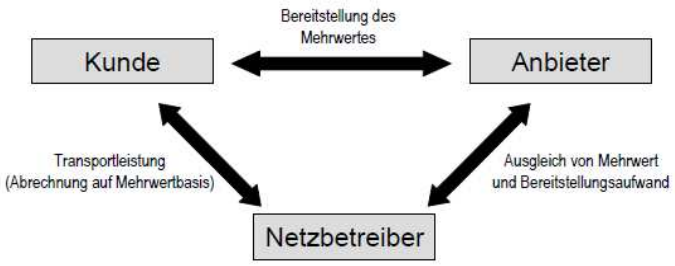

Abbildung 2.3: Abrechnung als Mehrwertdienst[Poustt03]

[1] http://www.vdv.de/ oder http://www.dashandyticket.de/
[2] http://www.bahn.de oder http://www.touchandtravel.de

Laut der Studie zur Ausgestaltung mobiler Bezahlverfahren von Dietmar Wiedemann [Wiedem08] vertrauen die meisten Kunden (74,7%) einer Bank oder Kreditkartengesellschaften als MPSP. Es gibt zwar viele weitere Zahlungsdienstleister, über die die Abrechnung einer Dienstleistung stattfinden kann. Diese genießen jedoch weniger Vertrauen bei den Kunden.

Im weiteren Verlauf der Arbeit wird nicht weiter auf die konkrete Ausprägung des Zahlungsdienstleisters eingegangen, sondern dieser wird nur abstrakt als „Mobile Payment Service Provider" (MPSP) behandelt. Es handelt sich dabei um eine beliebige dritte Partei, die von dem Kunden stets als ausreichend vertrauenswürdig eingeschätzt wird, um den Zahlungsverkehr für ihn zu übernehmen.

2.4.4 Sicherheitsanforderungen

Der elektronische Handel sowie auch der Mobile Commerce (MC) erfordern ein hohes Niveau an Informationssicherheit.

Nach Klaus Turowski [Turows04] liegt aus der Sicht des Anbieters eine Schwierigkeit der Sicherheitsproblematik darin, dass Sicherheit zwar Kosten verursacht, aber keinen unmittelbaren Beitrag zur Aufgabenerfüllung liefert. Nach Prof. Dr. W. Riggert [Rigger06] sei zunehmende Sicherheit mit überproportional wachsenden Kosten verbunden.

Der Dienstanbieter hat darauf zu achten, dass die auf Gewinn ausgerichtete Geschäftstätigkeit durch den Betrieb des eCommerce Angebots unterstützt und nicht gefährdet wird.

Weiter nach Riggert [Rigger06] wollen die Nutzer von eCommerce-Angeboten einen Mehrwert erzielen, wie durch die Aufhebung von räumlichen oder temporären Beschränkungen. *„Sicherheit realisiert keinen solchen Mehrwert: kein Kunde nutzt ein Angebot nur auf Grund der Tatsache, dass es sicher ist, lehnt aber ein Angebot ab, wenn es unsicher ist.“* Daher sei nicht ein Höchstmaß an Sicherheit entscheidend, sondern lediglich in einem der Anwendung und dem Risiko angemessenen Maße.

Weiter unterscheidet [Turows04] zwischen objektiver und subjektiver Sicherheit. Die objektive Sicherheit gewährleistet dabei den sicheren Betrieb der Anwendung. Dabei sollen Schäden durch Angriffe Dritter oder durch Betrug des Kunden verhindert werden. Im Gegensatz dazu entscheidet die subjektive Sicherheit des Kunden darüber, ob das Angebot von diesem als sicher angesehen wird und ob er dieses wahrnimmt. So sei beim Design von eCommerce-Anwendungen neben der tatsächlichen auch die gefühlte Sicherheit zu beachten, damit die Kunden die Nutzung des Angebots nicht aufgrund von Sicherheitsbedenken verweigern.

Die Sicherheitsanforderungen an MC Anwendungen sind ähnlich zu denen von eCommerce Anwendungen einzustufen. Nach Federrath [Federr09] kann ein mobiles Umfeld nicht als vertrauenswürdig vorausgesetzt werden. Im Gegensatz dazu stehen kontrollierte Netzwerke, in denen der Zugriff auf die Daten kontrolliert werden kann und nicht offen ist.

Mithilfe technischer Schutzmaßnahmen, die insbesondere bei Einsatz kryptographischer Berechnungen eine entsprechende Leistungsfähigkeit der Hardware voraussetzen, kann die Anwendung auf das gewünschte Maß an Sicherheit gebracht werden.

2.5 Android

2.5.1 Einleitung

Android ist eine Softwareplattform für mobile Geräte. Die *Android* Developers Webseite [AndDev09] definiert *Android* als einen *Software Stack* für mobile Geräte, bestehend aus einem Betriebssystem, Middleware und einigen Anwendungen.

Android wird von der *Open Handset Alliance* (OHA) unter der Führung von Google entwickelt. Die *OHA* ist eine Allianz aus 47 Mitgliedern, welche unter anderem aus namhaften Unternehmen der Mobilfunkbranche, der Softwareentwicklung, des Marketings, der Halbleiterindustrie, sowie aus einigen Endgeräteherstellern besteht.

Es gibt einige vergleichbare Plattformen zu *Android* speziell für mobile Geräte. Erwähnenswert sind hier RIMs[1] BlackBerry, Apples Mac OS X auf dem iPhone, das Symbian OS[2] der gleichnamigen Firma oder Microsofts Windows Mobile[3].

2.5.2 Plattform-Architektur

[AndDev09] beschreibt weiter die Architektur der *Android* Plattform, welche aus fünf Teilkomponenten besteht.

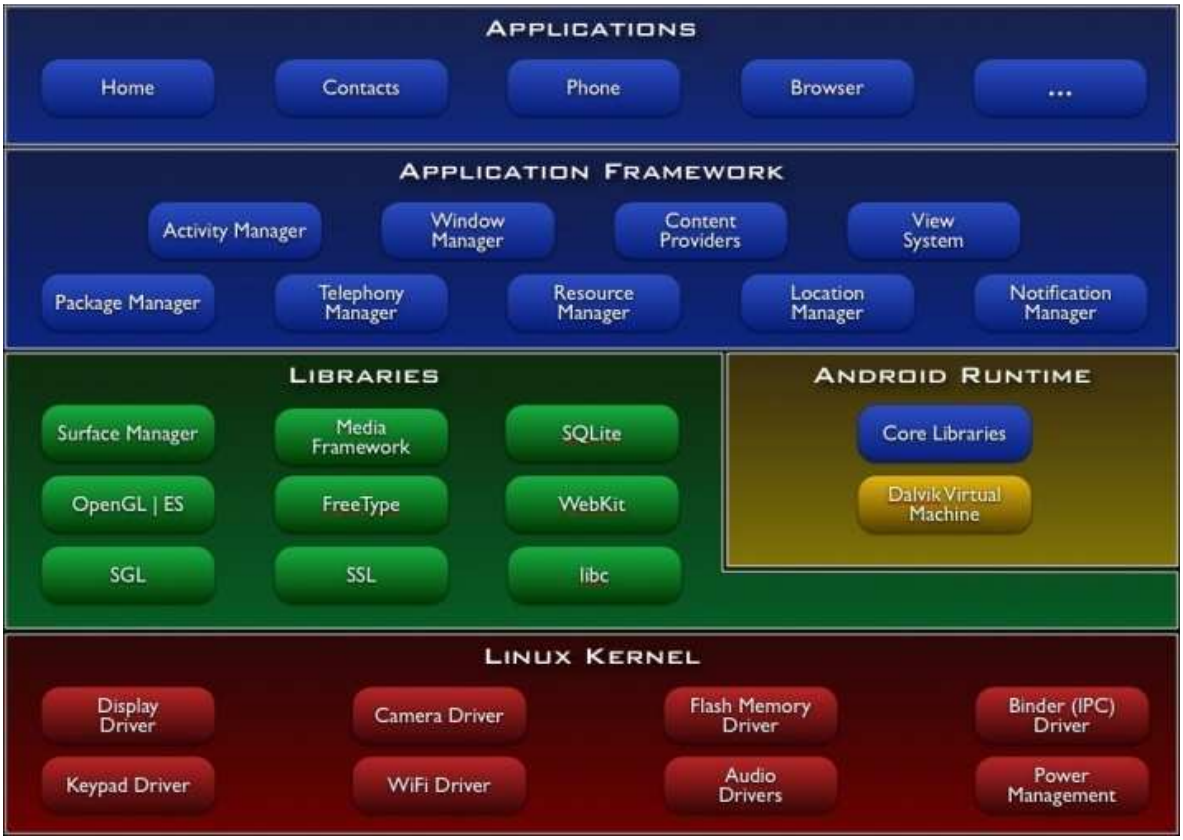

Abbildung 2.4: Android Architektur [AndDev09]

[1] http://www.rim.com/
[2] http://www.symbian.org/
[3] http://www.microsoft.com/windowsmobile/

Android basiert im Kern auf dem **Linux Kernel** 2.6 (Abbildung 2.4, Linux Kernel), um Systemdienste wie Sicherheit, Speicherverwaltung, Prozessverwaltung, Netzwerkfunktionalität und den Hardwarezugriff über Treiber zu realisieren. Zudem agiert er als eine Abstraktionsschicht zwischen der Hardware und den darauf aufsetzenden Systemkomponenten.

Android enthält zudem einige **Bibliotheken** ((Abbildung 2.4, Libraries), die in C/C++ geschrieben sind und von verschiedenen Komponenten des *Android* Systems genutzt werden. Entwicklern wird deren Funktionalität mittels der darüber liegenden Schicht „*Android Application Framework*“ angeboten.

Die **Laufzeitumgebung** ((Abbildung 2.4, *Android Runtime*) enthält einige Standard Bibliotheken, die für die Ausführung von Java-Anwendungen auf der „Dalvik Virtual Machine“ benötigt werden.

Die **Dalvik Virtual Machine**[1], (DVM) ist die Laufzeitumgebung für Java-Anwendungen innerhalb der *Android* Plattform. Es wird keine gewöhnliche Java Virtual Machine (JVM) verwendet, da die DVM im Gegensatz zur JVM einige Vorteile besitzt auf die im Folgenden eingegangen wird. Laut Dan Bornstein [Bornst09] wurde die Dalvik Virtual Machine (DVM) speziell für die Ausführung auf Geräten mit begrenzter Leistungsfähigkeit konzipiert. Die DVM basiert im Gegensatz zur JVM auf der Registermaschine anstelle eines Kellerautomaten und wurde speziell für die speicherschonende Ausführung von Java-Anwendungen entworfen, die zuvor mittels eines speziellen Cross-Compilers in das „Dalvik Executable Format“ konvertiert wurden.

Jede *Android*-Anwendung läuft als eigenständiger Prozess im System innerhalb seiner eigenen Instanz der DVM. Die DVM setz auf dem zugrunde liegendem Linux Kernel auf, um unter anderen Funktionalitäten wie Threading und Memory-Management zu realisieren, sowie auch den für die Sicherheit relevanten Komponenten zur Steuerung der Autorisierung und Zugriffskontrolle.

Die Schicht **„Applications Framework**“ (Abbildung 2.4, Application Framework) besteht aus Diensten und Systemen, wie Views, Content Provider, Ressource Manager, Notification Manager und Activity Manager und können von den darüber liegenden Anwendungen genutzt werden.

Die **Anwendungsschicht** (Abbildung 2.4, Applications) umfasst einige Kernanwendungen wie sie auf einem mobilen Gerät typischerweise gebraucht werden (u.a. einen Email-Client, ein SMS Programm, einen Kalender, Karten-Software, einen Browser, eine Kontaktverwaltung sowie eine Telefonfunktion). Sämtliche Anwendungen der Anwendungsschicht sind in der Programmiersprache Java geschrieben.

Lizenz

Ein großer Teil des Quellcodes ist Open Source. *Android* ist frei und steht unter der *Apache 2.0 Lizenz.* Dies bietet dabei für die Nutzer eine Reihe von Vorteilen und Nachteilen, von denen im Folgenden einige aufgeführt werden:

Michael Merz [Merz02] nennt dazu unter anderem, dass für die Nutzung solcher Software keine Lizenzkosten gezahlt werden müssten, und dass durch die mögliche öffentliche Code-Evaluation eine erhöhte Zuverlässigkeit zu erwarten sei. Als nachteilig könnte sich eine divergente Entwicklung der Plattform durch verschiedene Hersteller erweisen, die *Android* als Betriebssystem an ihre Endgeräte anpassen. Es könnte so zu Entwicklung verschiedener Derivate kommen, in denen ständig neue Sicherheitslöcher entstehen könnten.

[1] http://www.dalvikvm.com

Zudem konnten durch die Entwicklung und Verwendung der DVM - einer eigenen virtuellen Maschine als Teil von *Android* - Lizenzierungsgebühren an Sun umgangen werden. Proprietäre Systemteile der Klassenbibliothek von Java durch Sun wurden entfernt und durch freie Alternativen ersetzt.

2.5.3 Anwendungsentwicklung

Zur Entwicklung von Anwendung für *Android* steht ein Software Development Kit (SDK) zur Verfügung[1]. Des Weiteren werden mit dem SDK dem Anwendungsentwickler ein Hardware-Emulator, ein Debugger, sowie einige weitere Tools zur Verfügung gestellt. Als zentrale Programmiersprache für Anwendungen, die auf *Android* laufen sollen, wird Java verwendet. Bei der Ausführung kommt - wie bereits erwähnt - die Dalvik Virtual Maschine zum Einsatz.

Ab Version 1.5 wird zusätzlich zum SDK das Native Development Kit (NDK) zur Verfügung gestellt. Damit ist es möglich einige Anwendungsteile in nativen Programmiersprachen, wie C oder C++ zu schreiben. Diese Programmteile laufen außerhalb der Dalvik Virtual Machine, werden jedoch durch den zugrundeliegenden Linux Kernel mit den Berechtigungen des Unix-Benutzers ausgeführt.

Der *Android Market*[2] ist eine zentrale Verteilungsplattform für *Android*-Anwendungen über die Entwickler ihre Software zum Download für Endanwender anbieten können und wird von *Google* Inc. betrieben. Die Verwendung des *Android Market* ist nicht zwingend. Innerhalb der Plattform gibt es für den Benutzer eine Option zum Wählen, ob er auf dem System die Installation von Anwendungen zulassen möchte, die nicht über den *Android* Market vertrieben werden.

2.5.4 Sicherheitsmodell

Wie jedes moderne Betriebssystem enthält auch *Android* ein umfassendes Sicherheitskonzept, welches in diesem Abschnitt vorgestellt wird. Es werden einige Techniken zur Bedrohungsabwehr (vgl. Kapitel 2.3.5 in *Android* verwendet. Dazu gehören die Verwendung von Java als eine höhere Programmiersprache zum Verhindern von Bedrohungen, sowie das Begrenzen von Schäden durch die Isolation von Anwendungen untereinander.

Isolation

In *Android* werden die Anwendungen isoliert voneinander ausgeführt. Dabei handelt es sich um eine Technik zur Begrenzung von Schäden, wie sie im Abschnitt 2.3.5vorgestellt wurde. [AndDev09] beschreibt dies wie folgt:

"A central design point of the Android security architecture is that no application, by default, has permission to perform any operations that would adversely impact other applications, the operating system, or the user. This includes reading or writing the user's private data (such as contacts or e-mails), reading or writing another application's files, performing network access, keeping the device awake, etc."

Laut Avik Chaudhuri [Chaudh09] werden dabei einige Sicherheitsfeatures des darunterliegenden Linux-Kernels genutzt: Zunächst einmal läuft jede Anwendung innerhalb eines eigenen Linux Prozesses. *Android* startet und stoppt dabei die laufenden Prozesse, um die Anwendungen auszuführen. Weiter läuft in jedem Prozess eine eigene Instanz der DVM, die Dalvik-Bytecode ausführen kann.

[1] wahlweise mit oder ohne Verwendung der Software-Entwicklungsumgebung Eclipse (www.eclipse.org)
[2] http://market.android.com

Nach Dhruv Mohindra [Mohind08] wird dies in *Android* in der Weise umgesetzt, dass jede *Android*-Anwendung[1] eine eigene Unix User ID (UID) erhält. Damit ist gewährleistet, dass keine Anwendung auf eine andere störend einwirken kann. Die Entscheidung über die verwendete UID erfolgt dabei einmalig zum Installationszeitpunkt. Zudem gibt es die Möglichkeit, dass zwei Anwendungen eine *Shared User ID* verwenden, wenn sie vor Auslieferung vom Entwickler mit der gleichen digitalen Signatur versehen wurden.

Die Verwendung verschiedener UIDs für jede Anwendung in *Android* wird dadurch ermöglicht, dass derartige mobile Geräten typische Einzelbenutzersysteme sind. Eine Verwendung der UID für verschiedene Benutzer scheint daher unnötig, sodass die Vorteile einer Aufteilung der Anwendungen auf verschiedene UIDs frei zur Verfügung stehen, einhergehend mit vielen Sicherheitsvorteilen, die dadurch genutzt werden können.

Jede *Android*-Anwendung läuft damit zunächst in einer sicheren Umgebung, einer Sandbox, und es ist für Anwendungen nicht möglich, sich gegenseitig zu beeinflussen, also auch nicht auf negative Weise.

Permissions

Nach Jesse Burns [Burns08] können durch das vorgestellte Sandboxing weder Lese- noch Schreibzugriffe auf die Daten oder den Programmcode anderer Anwendungen stattfinden. Die gemeinsame Verwendung von Daten muss explizit stattfinden.

Ausnahmen der vollständigen Isolation der Anwendungen bilden dabei die sogenannten explizit deklarierten *Permissions*. Sie ermöglichen ein Umgehen des Sandboxing-Prinzips und machen damit Interprozesskommunikation möglich. [AndDev09] beschreibt sie wie folgt:

"An application's process is a secure sandbox. It can't disrupt other applications, except by explicitly declaring the permissions it needs for additional capabilities not provided by the basic sandbox. These permissions it requests can be handled by the operating in various ways, typically by automatically allowing or disallowing based on certificates or by prompting the user. The permissions required by an application are declared statically in that application, so they can be known up-front at install time and will not change after that."

Die Einräumung dieser Rechte durch den Benutzer findet dabei nur einmalig zum Installationszeitpunkt statt und bleibt bis zur Deinstallation der Anwendung vorhanden. Abbildung 3.3 zeigt exemplarisch die dem Anwender bei diesem Vorgang präsentierte Bildschirmmaske.

2.6 Vergleichbare Arbeiten zum Thema

Es läuft zurzeit eine ähnliche Arbeit zu dem vom Autor dieser Arbeit behandelten Thema von Deniz Binay [Binay09] mit dem Titel *Android Security Improvement.* Der Fokus dieser Arbeit liegt dabei nicht im eCommerce, sondern eher generell in Sicherheitsproblemen, denen mit *Android* basierenden Technologien begegnet werden soll, wie aus der folgenden Beschreibung deutlich wird:

"The aim of this project is to analyze the different security issues of the smartphones and implementing an innovative solution on the new Google Android platform.

[1] Als eine *Android*-Anwendung in diesem Zusammenhang ist all das zu verstehen, was sich innerhalb eines *Android Package (.apk)* befindet, und als ein solches ausgeliefert und installiert wird. Andere *Executables*, die auf dem System als *root* ausgeführt werden, sind daher von dieser Definition ausgeschlossen.

The smartphone is now more commonly used for critical operations. As a convergent device of many technologies, it opens the way for new types of attacks. With the new platforms such as the iPhone or Android arriving on the market, improving the security of such devices is critical. A smartphone can have vulnaribilites caused by its communication layers (wi-fi, bluetooth, ...), in its kernel, in its browser, some installed applications, ...

My project will focus on the open source platform Android and more precisely it will cover some security aspects like SMSishing. It will lead in the development of an application or an improvement in the source code of Android. An example of such an application would be a software that would verify the authenticity of the web links received in a SMS by checking it with an online database."

Der genaue Inhalt dieser Arbeit, sowie die Ergebnisse dieser, sind dem Autor dieser Arbeit jedoch zum Zeitpunkt des Schreibens dieser Arbeit nicht bekannt. Es bleibt abzuwarten, wie viele gemeinsame Schnittmengen es in beiden Arbeiten geben wird, und ob sich beide Arbeiten inhaltlich ergänzen können.

2.7 Zusammenfassung

Die verschiedenen Themengebiete dieser Arbeit, Sicherheit als die zu untersuchende Eigenschaft, mobile Geräte, sowie *Android* als Hard- und Softwareplattformen und eCommerce als das Einsatzgebiet, wurden nun in einem Rahmen vorgestellt, der als Grundlage für weitere Analysen dienen soll.

Wie in Kapitel 2.4.4 festgestellt benötigt der EC, sowie auch der MC, bestimmte Sicherheitsanforderugen. Ziel dabei ist nicht, eine absolute Sicherheit des Systems zu erreichen, sondern nur das Erreichen von Sicherheit auf einem ökonomisch sinnvollen Niveau.

Ziel des weiteren Vorgehens ist es, die Umsetzbarkeit dieser, auf der relativ neuen Plattform *Android* zu untersuchen. Dazu wird im Folgenden eine Beispielanwendung für ein konkretes Szenario entworfen.

Am Ende dieser Arbeit sollen nicht nur Aussagen über die Sicherheit in genau diesem einem Szenario, sondern generellere Aussagen über die Sicherheit im gesamten MC Umfeld getroffen werden können.

3 Beispielszenario

3.1 Einleitung

Für die weitere Analyse der Sicherheit von *Android* und deren Eignung für eCommerce-Anwendungen stellt der Autor in diesem Kapitel ein Anwendungsszenario aus dem Bereich des Mobile Ticketing vor. Dabei handelt es sich um ein IT-System für die mobile Nutzung von ÖPV-Dienstleistungen, zu dem zum einen die Fahrplanauskunft, als auch der Erwerb von Tickets für die Fahrgäste gehört.

Dieses Kapitel beschreibt zunächst das Beispielszenario und die dazugehörige Anwendung. Dabei ist das Thema Sicherheit noch nicht Kernthema und der Autor wird zunächst weitestgehend auf die Beachtung von Sicherheitsaspekten verzichten, da diese umfassend im folgenden Kapitel (Security Engineering) behandelt werden. Zunächst einmal wird ein Überblick über die zur Verfügung stehende technische Infrastruktur gegeben, gefolgt von der Vorstellung der verschiedenen Funktionen anhand von Anwendungsfällen unter Zuhilfenahme des entworfenen Prototyps dazu. Die genaue technische Funktionsweise, sowie die Architektur folgen dann später.

Wahl des Beispielszenarios

Es gibt unzählige Anwendungsbereiche im eCommerce, in denen eine derartige Sicherheitsanalyse für *Android* stattfinden könnte. In diesem Abschnitt wird daher zunächst die Wahl des gewählten Szenarios begründet. David Kleidermacher [Kleider09] verdeutlicht diese Vielfalt der möglichen Anwendungsbereiche wie folgt:

"We bring our cell phones wherever we go. How awesome would it be if our cell phone also acted as the key to our automobile, a smart card for safe Internet banking, our virtual credit card for retail payments, our ticket for public transportation, and our driver's license and/or passport? There is a compelling world of personal digital convenience awaiting us."

Es mangelt ihm jedoch noch an sicheren Umgebungen für derartige Anwendungen, die ausreichend viel Vertrauen beim Konsumenten erzeugen könnten. Derartige sichere Umgebungen könnten eine Vielzahl an verschiedenen (auch sicherheitskritischen) Anwendungen beherbergen, die dann alle durch diese geschützt wären. Durch die gemeinsame Nutzung dieser mit beispielsweise Multimedia-Anwendungen könnten zudem Ressourcen gespart, und der Energieverbrauch, sowie die Produktionskosten gesenkt werden.

Als handelbare Produkte und Dienstleistungen im mobilen Umfeld eignen sich insbesondere diejenigen gut, von denen der Kunde unmittelbar profitieren kann, d.h. unabhängig von seinem derzeitigen Aufenthaltsort sollte er einen spürbaren Mehrwert gegenüber einer vergleichbaren ortsgebundenen Anwendung erfahren.

Der Autor hat sich für eine Anwendung aus dem Bereich des Mobile Ticketing entschieden. Es handelt sich dabei um ein fiktives Szenario. Pilotprojekte im Bereich des ÖPV (vgl. Abschnitt 2.4.2) zeigen, dass durchaus Interesse für derartige Anwendungen besteht, die Realisierung jedoch noch nicht weit genug fortgeschritten ist, um von der breiten Masse der Bevölkerung akzeptiert zu werden.

Mehrwert durch mobile Nutzung

Laut Klaus Turowski [Turows04] genügt es beim Entwurf einer MC-Anwendung nicht, *„ein existierendes Angebot auf einem mobilen Endgerät verfügbar zu machen, etwa eine bestehende Webseite mobil zugänglich zu machen“*, da die Verwendung mobiler Endgeräte und Kommunikationstechniken spezifische Vor- und Nachteile habe. Daraus leitet er für den Anwendungsentwurf den Grundsatz *„Design to Mobile“* ab, bei dem *„eine mobile Anwendung speziell auf die Potenziale und Probleme der Mobilität maßgeschneidert sein muss“*.

Er empfiehlt dazu als einen ersten Analyseschritt, die Potentiale der Anwendung durch die Beantwortung der Frage, welchen entscheidenden Mehrwert der Nutzer durch die Mobilität der Anwendung erhalten würde, zu fokussieren:

Als vorteilhaft für den Kunden ist nach Thomas Mügge [Mügge07] eine erhöhte Verfügbarkeit des Systems, da ein Fahrkartenerwerb unabhängig von Fahrkartenautomaten und den Öffnungszeiten von Verkaufsstellen stattfinden könnte. Das benötigte Gerät hätte der Kunde dazu bereits in der Tasche. Dies kann er ortsunabhängig nutzten, wie beispielsweise auf dem Weg zur Abfahrtshaltestelle, sodass lästiges Warten am Fahrkartenautomaten entfällt.

Nach Dr. Walter Leitner [Leitne07] könnten die Fahrgäste bei der Ticketauswahl vom System unterstützt werden: Durch eine Kombination mit der mobilen Fahrplanauskunft könne dem Kunden direkt nach der Fahrplanauskunft ein passendes Ticket angeboten werden. Weiter würde den Kunden der Fahrausweiserwerb auch ohne oder mit nur geringen Tarifkenntnissen ermöglicht werden, wenn die Nutzerführung auch über die Tarifgrenzen hinweg strukturell stets einheitlich gehaltenen werden könnte.

Durch die bargeldlose Zahlung können schnelle und bequeme Zahlungen durchgeführt werden und Kunden müssen zum Ticketkauf kein Bargeld mitführen. Aufgrund der geringeren Bargeldhaltung und des nicht mehr benötigten Mitführens eines Papiertickets, welches dem Risiko des Verlustes durch Diebstahl oder sonstiger Unachtsamkeit ausgesetzt wäre, entsteht ein Mehr an Sicherheit für den Kunden. Des Weiteren können Kunden von erhöhten Zinserträgen profitieren, da nicht benötigtes Bargeld beim Kreditinstitut verweilen kann.

Bei Verwendung von Mobile Ticketing kann eine individuellere Fahrplanauskunft stattfinden, die ganz auf die Bedürfnisse der Kunden zugeschnitten ist, da persönliche Präferenzen bei der Routenwahl möglich sind. Insbesondere könnten Fahrgäste, die in ihrer Mobilität eingeschränkt sind, eine speziell an ihre Bedürfnisse angepasste Fahrplanauskunft erhalten, bei der beispielsweise die Fußwege oder die Barrierefreiheit bei der Wahl der Bahnhöfe und Haltestellen Berücksichtigung findet.

Analyse der technischen Infrastruktur im mobiles Umfeld

Als einen zweiten Analyseschritt empfiehlt Klaus Turowski [Turows04] die Bedeutung der Einschränkungen und Heterogenitäten mobiler Kommunikationstechniken für die zu entwerfende Anwendung zu untersuchen. Viele der Einschränkungen mobiler Geräte wurden bereits in Abschnitt 2.2.2 betrachtet, weshalb im Folgenden speziell auf Nachteile eingegangen wird, die sich durch die Verwendung einer Webanwendung[1] für das Beispielszenario ergeben würden.

[1] Eine Webanwendung verwendet einen *Very-Thin-Client*, der nur die Präsentationsschicht umfasst.

Auf den Webbrowsern vieler moderner Smartphones lassen sich gewöhnliche Webseiten mitsamt darin enthaltenen Webanwendungen darstellen. Diese sind allerdings, bedingt durch das kleinere Display und eine andere Handhabung, oft nur schwer zu bedienen. Für viele Webseiten steht daher eine Mobil-Variante zur Verfügung, deren Design und Bedienung speziell für mobile Endgeräte optimiert wurde. So beispielsweise auch die Webseite des Hamburger Verkehrsverbundes[1]. Diese erfüllt zwar ihre Funktion als ein Online-Fahrplanauskunftsdienst, und lässt sich auch auf einem kleinen mobilen Endgerät noch bedienen, jedoch gibt es weitere Nachteile gegenüber einer echten Thin-Client-Anwendung, die clientseitig Code ausführen können.

Während des gesamten Vorganges einer Fahrplanabfrage durch Kunden muss eine mobile Datenverbindung aufrecht erhalten bleiben, obwohl diese nur für die eigentliche Abfrage der Daten vom Server benötigt wird. Insbesondere bei verbindungsorientierten Datenverbindungen, bei denen nach Zeit abgerechnet wird, können so für den Kunden schnell hohe Kosten entstehen.

Eine mobile Datenverbindung über das Telekommunikationsnetz ist zwar flächendeckend verfügbar, jedoch auch nicht überall[2], was zur Folge hat, dass das System nicht lückenlos verwendet werden kann. Insbesondere für eine schlichte Fahrplanauskunft könnten auf dem Gerät vorgehaltene Fahrplaninformationen als Basis für weitere Abfragen durch den Benutzer dienen, ohne dass je eine mobile Datenverbindung hergestellt werden müsse.

3.1.1 Thin-Client-Architektur

Gunter Lepschies [Lepsch00] nennt einige allgemeine Vorteile von Thick- und Thin-Client-Anwendungen im Gegensatz zu rein web-basierten Lösungen. Dies sind im Einzelnen die Bandbreite, Caching-Funktionalitäten, sowie die Nutzung von Features des Endgeräts.

In browserbasierten Anwendungen werde eine Menge an Bandbreite verschwendet. Speziell auf mobilen Geräten, die langsame und teure Datenverbindungen verwenden, könne mit „richtigen“ Anwendungen die Bandbreitenausnutzung verbessert werden, da die Geschwindigkeit so erhöht und das Datenvolumen vermindert werden könne.

Da mobile Internetzugänge noch nicht in der gewohnten Qualität und Stabilität wie vergleichbare kabelgebundene Lösungen verfügbar sind, könne eine Anwendung durch den Einsatz von Caching weiterhin lauffähig bleiben, auch wenn keine oder eine zu langsame Internetverbindung bestünde.

Weiterhin könne durch Nutzung von Features der Endgeräte speziell im mobilen Umfeld die User Experience mit der Anwendung verbessert werden, wie beispielsweise durch GPS, Kamera-Hardware oder Beschleunigungs-Sensoren.

Ein weiterer Vorteil eines eigenständigen Thin-Clints, der für genau einen Anwendungszweck entworfen wurde, besteht in dem relativ schlanken Programmcode. Dieser fällt bei einer eigenständigen Client-Anwendung vergleichsweise kurz aus im Gegensatz zur Verwendung eines Webbrowsers, der viele unterschiedliche Inhalte und Webanwendungen darstellen kann. Weniger Code ist generell überschaubarer und wartbarer, und es werden weniger Komponenten genutzt, sodass eine sicherere Programmausführung einfacher gewährleistet werden kann, als bei der Verwendung eines „riesigen Multi-Funktions-Webbrowsers“.

[1] www.hvv.de bzw. die mobile Variante mobil.hvv.de
[2] Speziell im ÖPV Bereich sind hier unterirdische Stationen oder Tunnel zu nennen, in denen die Erreichbarkeit des Mobilfunknetzes eingeschränkt sein könnte.

Aus diesem Grunde erachtet es der Autor dieser Arbeit in dem gewählten Beispielszenario aus dem Bereich des Mobile Ticketing für sinnvoll, die Verwendung einer eigenständigen clientseitig ausgeführten Anwendung der Verwendung einer rein webbasierten Lösung vorzuziehen, die über einen mobilen Browser aufgerufen werden würde. Eine Übersicht über die daraus resultierende Architektur der Anwendung veranschaulicht Abbildung 3.1.

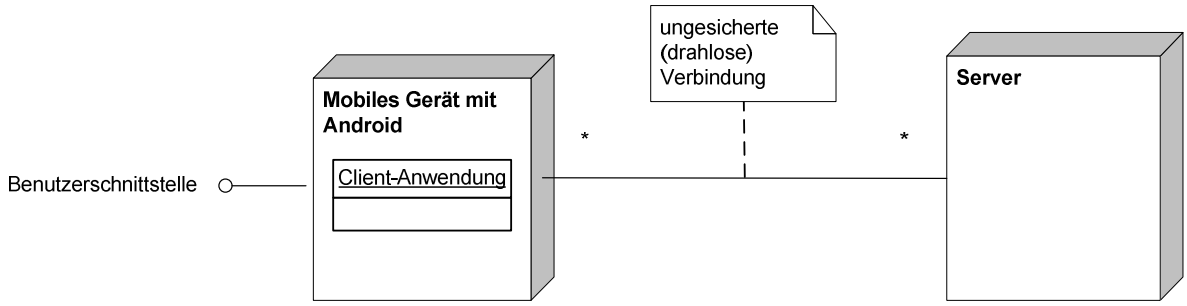

Abbildung 3.1: Thin-Client Architektur

3.1.2 Mobile Kommunikationsinfrastruktur

Die Nutzung des Fahrplanauskunfts- und Ticketingdienstes soll ubiquitär möglich sein. Die Bereitstellung des Dienstes über das Internet ermöglicht dies, da dazu lediglich noch ein Zugang vom mobilen Gerät ins Internet benötigt wird. Die Diensterfüllung kann so letztlich beim Zusammenspiel zwischen dem mobilen Client und dem Server im Rechenzentrum des Betreibers erbracht werden.

Im mobilen Umfeld stehen verschiedene drahtlose Kommunikationstechnologien zur Verfügung, die samt ihrer Vor- und Nachteile bereits in Kapitel 2.2.2 vorgestellt wurden, und umfassen:

- Private oder andere nicht öffentliche drahtlose lokale Netzwerke
- Öffentliche, drahtlose lokale Netzwerke (kostenlos oder kostenfrei)
- Das Mobilfunknetz

Für die Auswahl der genutzten Technologie und des Netzwerkes gelten dabei verschiedene Kriterien: Verfügbarkeit, Geschwindigkeit, Sicherheit, Kosten, Komfort. Der Fahrgast entscheidet dabei nach persönlichen Vorlieben über das zu verwendende Netzwerk, um den Dienst nutzen zu können.

Die Verfügbarkeit des Netzwerkes stellt dabei ein „KO-Kriterium“ dar. Während das Mobilfunknetz flächendeckend zu Verfügung steht, sind lokale Netzwerke nur in Abhängigkeit des Aufenthaltsortes verfügbar. Ist die Verfügbarkeit nicht gegeben, so kann das Netzwerk nicht verwendet werden. Steht dem Fahrgast an seinem Aufenthaltsort kein geeignetes lokales drahtloses Netzwerk zum Zugang ins Internet zur Verfügung, kann auf die Nutzung mobiler Datenverbindungen des Mobilfunknetzes ausgewichen werden, über die ein flächendeckender Zugang gegeben ist.

Durch die in diesem Anwendungsszenario nur geringen Anforderungen an Geschwindigkeit und Datenmenge sollten die Kosten für die Nutzung der mobilen Datenverbindung, die an den Mobilfunkanbieter zu zahlen sind, im Vergleich zum Ticketpreis eher gering bleiben, sowohl bei verbindungsorientieren als auch bei datenorientierten Abrechungsmodellen.

Der Fahrplanauskunfts- und Ticketingdienst kann damit von (fast) überall aus mobil genutzt werden, sodass relativ komfortabel Fahrplaninformationen abgefragt und Tickets gekauft werden können, wie beispielsweise auf dem Weg zur Abfahrtshaltestelle. Weiter kann durch eine automatische Auswahl des Netzwerks die Ubiquität des Fahrplanauskunfts- und Ticketingdienstes verbessert werden.

Um den Fahrgästen eine kostenfreie Nutzung des Mobile Ticketing Dienstes zu ermöglichen, könnten vom Betreiber aus Access Points an Bahnhöfen und Haltestellen zur Verfügung gestellt werden. Nach Dr. Walter Leitner [Leitne07] könnten durch den Betrieb einer solchen Kommunikationsinfrastruktur in Ballungszentren viele Kunden auf einmal erreicht werden. Diese kann dabei parallel zu den herkömmlichen Fahrkartenautomaten betrieben werden und bietet weitere ökonomische Vorteile für das Unternehmen. [1]

Des Weiteren steht auf diesem Wege eine Möglichkeit der Lokalisierung für den Kunden zur Verfügung, sodass beispielsweise eine Eingabe des Abfahrtsortes bei einer Fahrplanauskunft für den Fahrgast überflüssig werden könnte.

Zu beachten ist, dass bei den hier vorgestellten drahtlosen Kommunikationstechniken die gesamte Datenübertragung über zunächst ungesicherte Netzwerke stattfindet und diese keinerlei Schutz der übertragenen Daten gegen verschiedene Bedrohungen bieten können. Diese Kanäle müssen daher entsprechend geschützt werden.

3.1.3 Mobiler Informations- und Ticketingdienst

Mobiler Fahrplanauskunftsdienst

Der mobile Fahrplanauskunftsdienst ermöglicht es Fahrgästen über ihr Endgerät Fahrpläne und Routeninformationen des ÖPV-Unternehmens abzurufen. Im Wesentlichen erhält ein Nutzer nach Eingabe eines Start- und Zielortes, sowie der gewünschten Reisezeit, einen persönlichen Fahrplan generiert und zugestellt. Dieser Dienst wird den Fahrgästen im Rahmen der Angebotspräsentation des ÖPV-Unternehmens kostenfrei zur Verfügung gestellt.

Mobiler Ticketingdienst

Der mobile Ticketingdienst bietet den Fahrgästen den Erwerb von Fahrscheinen an, und damit die Berechtigung zur Nutzung der Personentransportdienstleistung. Die Abrechnung dieser erfolgt durch Verwendung eines der zur Verfügung stehenden mobilen Zahlungsverfahren. Wie auch an einem Fahrkartenautomaten oder einer anderen Verkaufsstelle trifft der Fahrgast hierbei zunächst die Auswahl über ein Ticket.

Tickets gibt es in verschiedenen Kategorien. Sie sind in unterschiedlichen Entfernungsstufen, Gültigkeits-Zeiträumen und Anzahl an fahrberechtigten Personen erhältlich. Zu Verfügung stehen in diesem Szenario beispielsweise „Einzelkarte“, „Ganztageskarte“, „9-Uhr-Tageskarte“, „Gruppenkarte“ und Ergänzungskarte, auf die der Autor jedoch nicht weiter eingehen wird.

Digitales Ticket

Das herkömmliche Papierticket soll in dieser Anwendung durch ein digitales Ticket ersetzt werden, welches die gleiche Funktionalität bietet. Dazu gehört das Erfüllen mindestens der gleichen Sicherheitsanforderungen, zu denen zum einen der Schutz vor Fälschungen gehört und zum anderen die Wahrung der Anonymität des Kunden. Weiter ist die Ausstellung eines Tickets stark mit dem Zahlungseingang durch den Fahrgast gekoppelt, wie man es von Fahrkartenautomaten kennt, und sollte nur paarweise mit diesem stattfinden.

[1] Als eine zusätzliche Nutzungsmöglichkeit der Access Points bietet es sich für den Betreiber der Hotspots an, den Fahrgästen kostenpflichtige schnelle und beliebig nutzbare Internetzugänge bereitzustellen, die für den Betreiber eine zusätzliche Einnahmemöglichkeit darstellen könnten.

Papiertickets bieten einen gewissen Schutz vor derartigen Betrugsversuchen. Die Fahrscheine sind dazu mit einigen Sicherheitsmerkmalen ausgestattet, sodass Fälschungen durch Kopieren oder Modifizieren nicht ohne weiteres möglich sind. Dadurch kann zwar kein 100%iger Schutz vor Fälschungen gewährleistet werden, jedoch werden dadurch der Aufwand und das Risiko für Betrüger um ein Vielfaches erhöht, sodass sich das Fälschen praktisch als nicht lohnenswert darstellt.

Beim Design eines digitalen Tickets ist es daher ebenso das Ziel, das Fälschen des Tickets derart unattraktiv zu gestalten, damit Betrüger von dieser Möglichkeit absehen werden. Die Ausstellung des Tickets erfolgt dabei ausschließlich durch das ÖPV-Unternehmen oder in dessen Auftrage.

Kontrollmedium

Das Kontrollmedium stellt in diesem Szenario ein beliebiges personenbezogenes Medium dar und kann als fälschungssicher angesehen werden. Das digitale Ticket ist nur in Verbindung mit dem Kontrollmedium gültig, und muss bei einer Fahrscheinkontrolle zusammen mit diesem dem Kontrolleur vorgelegt werden. Dies ist Teil einer Sicherheitsmaßnahme, die im späteren Verlauf dieser Arbeit weiter begründet wird.

Dabei kann es sich beispielsweise um einen Personalausweis oder um einen Führerschein handeln. Eine Verwendung der Seriennummer des Endgerätes, der MAC-Adresse der Netzwerkkarte, einer Telefonnummer oder eines Foto von der fahrberechtigten Person könnten ebenfalls in Frage kommen. Diese Möglichkeiten wurden vom Autor jedoch nicht weiter auf ihre Eignung hin überprüft, und bieten damit weiteren Analysebedarf.

Durch die Einführung eines Kontrollmediums wird die Anonymität für den Fahrgast bei Nutzung der Personentransportdienstleistung nun erheblich bedroht. Damit entsteht ein Zielkonflikt zwischen der Fälschungssicherheit als Ziel des Unternehmens und der Wahrung der Anonymität für die Fahrgäste.

Zahlungsmöglichkeiten

Bei Erwerb von Tickets hat der Fahrgast einen je nach Tarif variierenden Geldbetrag an den Betreiber zu zahlen. Bisweilen wird der Zahlungsverkehr im ÖPV-Bereich hauptsächlich über die an den Haltestellen installierten Fahrscheinautomaten oder über andere stationären Verkaufsstellen abgewickelt, und zwar unmittelbar vor oder nach dem Kauf eines Fahrscheines.

Dabei kann der Kunde mit Bargeld bezahlen, was der Verkaufsstelle beim Ticketkauf die Identität des Kunden nicht preisgibt. Nutzt der Kunde jedoch ein bargeldloses Bezahlverfahren, so hinterlässt der Kunde dem Verkehrsunternehmen in der Regel eine Identität, über welche die Abrechnung stattfinden könnte, z.B. in Form einer Konto- oder Kreditkartennummer. Diese könnte in Assoziation mit seiner eigenen persönlichen Identität stehen oder gebracht werden, wenn der Zahlungsdienstleister die dazu nötigen Informationen herausgibt.

So bieten etwa einige bargeldlose Bezahlverfahren, wie beispielweise die Geldkarte oder die *Paysafecard*[1], die nach dem Prepaid-Verfahren arbeiten, eine weitestgehende Anonymität für den Kunden beim Zahlungsvorgang. Rückschlüsse auf seine persönliche Identität sind so nur sehr schwer und unter Zuhilfenahme von zusätzlichen Informationen möglich, welche nicht in Verbindung mit dem Kauf des Tickets stehen.

Je nach Wahl des Zahlungsverfahrens bleibt es also dem Kunden überlassen, ob er anonym bleiben möchte, oder eben nicht. Dies gilt auch als Anforderung an die mobilen Zahlungsverfahren, die im Rahmen der Beispielanwendung verwendet werden können.

[1] http://www.paysafecard.com/de/

In dem Beispielszenario soll die Zahlung per *Mobile Payment* stattfinden, d.h. auf rein elektronischem Wege. Diese Zahlungsmethode kann direkt vom mobilen Endgerät des Fahrgastes bei Erwerb eines Tickets genutzt werden. Wie bereits in Kapitel 2.4.3 (Mobile) erwähnt wurde, werden für die Abwicklung des Zahlungsverkehrs in diesem Szenario weder eine konkrete Technologie noch bestimmte dritte Zahlungsdienstleister genannt und stattdessen ein nicht näher konkretisierter Zahlungsdienstleister (MPSP) verwendet, der die gesamte Abwicklung des Zahlungsverkehrs übernehmen soll.

Dieser kann zudem für die gewünschte Anonymität bei der Zahlung sorgen, indem dem ÖPV-Unternehmen vom Zahlungsdienstleister keine Informationen über die Herkunft der Gelder zugeführt wird, sodass diesem die genauen Identitäten der Fahrgäste verborgen bleiben. Damit kann die Anonymität bei der Zahlung garantiert werden, ganz im Gegensatz zu anderen Zahlungsarten, bei denen dies durch die Verarbeitung personenbezogener Daten, wie beispielsweise dem Namen des Kontoinhabers bei einer Banküberweisung, nicht möglich ist. Bei elektronischen Zahlungen müssen die rechtlichen Rahmenbedingungen beachtet werden, welche jedoch nicht weiter untersucht wurden.

3.2 Anwendungsfälle und Funktionsweise

Die Funktionsweise der Anwendung wird im Folgenden anhand der verschiedenen Anwendungsfälle beschreiben, die in Abbildung 3.2 dargestellt sind.

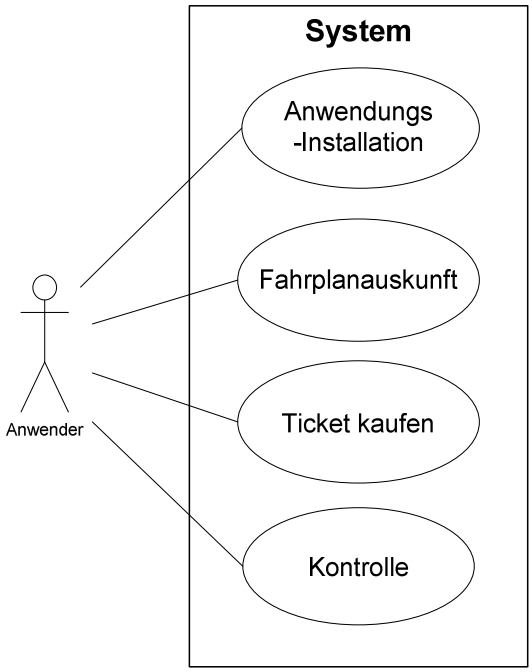

Abbildung 3.2: Anwendungsfälle

Auf die technischen Details des hierbei verwendeten Prototyps wird im folgenden Abschnitt 0 näher eingegangen.

3.2.1 Softwareinstallation auf dem Gerät

Die Softwareinstallation auf dem Gerät ist zwar kein eigentlicher Anwendungsfall in dem Sinne, dass der Anwender von ihm alleine profitieren könnte, jedoch stellt er einen sicherheitskritischen Vorgang dar, der im Weiteren auf seine Sicherheitsprobleme hin untersucht werden sollte.

Bei der Softwareinstallation erhält der Fahrgast zunächst eine URL zu der Webseite, von der er sich die Client-Software herunterladen kann. Der Download-Link kann dabei an einer beliebigen Stelle publiziert worden sein, so zum Beispiel an Bahnhöfen, Haltestellen, als auch im Web. Durch ein Zusenden des Links per SMS auf das mobile Gerät kann dem Fahrgast hierbei die manuelle Eingabe der Betreiber-URL erspart bleiben. Steht am aktuellen Aufenthaltsort des Fahrgastes ein Access Point des ÖPV zur Verfügung, so bekommt er über den Webbrowser des Endgerätes automatisch die Betreiber-Webseite angezeigt, von der er ebenfalls die Client-Software herunterladen kann.

Der Download des *Android*-Installationspakets (.apk) erfolgt dann direkt auf das mobile Gerät durch die auf dem Gerät vorhandene Browser-Software.

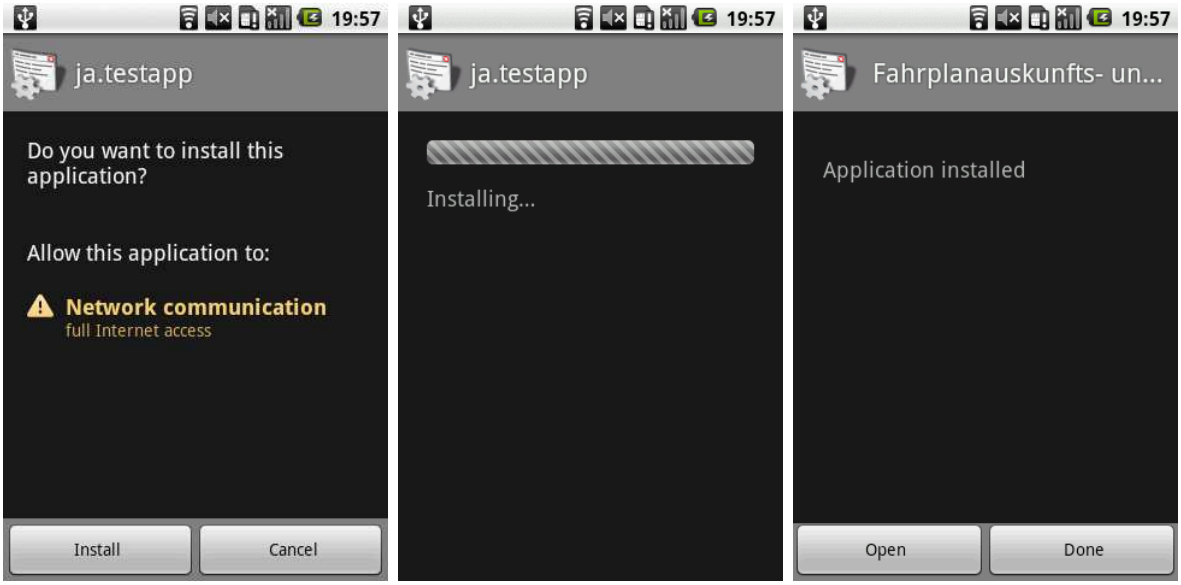

Abbildung 3.3: Installation der Client-Anwendung

Im Anschluss an den Download kann mit der Installation der Software begonnen werden. Die Anwendung benötigt dabei Rechte, die Internetverbindung des Geräts nutzen zu dürfen. Vor Beginn der Installation wird der Benutzer gefragt, ob er diese der Anwendung einräumt (siehe Abbildung 3.3). Lehnt er ab, wird die Anwendung nicht installiert, und die angebotenen Dienste können folglich nicht genutzt werden.

3.2.2 Allgemeine Funktionsweise und Programmablauf

Nach dem Starten der Anwendung befindet sich der Benutzer im Hauptmenü, wie in Abbildung 3.4 dargestellt ist. Von dort aus erhält der Fahrgast Zugriff auf die weiteren Programmfunktionen. Zur Verfügung stehen: „Fahrplanauskunft“ zum Abfragen von Routeninformationen (siehe 3.2.3), „Ticket kaufen“ zum Erwerben von Fahrscheinen (siehe 3.2.4), „Tickets anzeigen“ welcher zum einen der reinen Information der Fahrgäste dient, sowie zu den obligatorischen Fahrkartenkontrollen durch die Kontrolleure (siehe 2.4.2: Mobile Ticketing). Der Menüpunkt „Kundeneinstellungen“ dient dem Festlegen der Zahlungsinformationen und des Kontrollmediums.

Abbildung 3.4: Hauptmenü im Emulator[1]

3.2.3 Fahrplanauskunft

Eine Person P möchte mit den öffentlichen Verkehrsmitteln von Standort A nach Standort B gelangen. Durch Verwendung der Client-Anwendung auf dem Endgerät kann sich P nun mit dem Fahrplaninformationsdienst des ÖPV-Unternehmens verbinden, welches ihm nach Eingabe einiger Informationen Auskunft über die beste zur Verfügung stehende Route liefern kann.

Durch eine möglichst einfache und schnell zu bedienende Bildschirmmaske gibt P hierbei zunächst mindestens seinen gewünschten Abfahrtsort A und Zielort B, sowie die gewünschte Reisezeit an (vgl. Abbildung 3.5).

Um das Suchergebnis zu individualisieren und zu präzisieren, könnten zusätzlich weitere Eingaben getätigt werden. Dazu gehören beispielsweise gewünschte Zwischenhaltestellen, Angaben über die persönliche Mobilität (zu Fuß, Barrierefreiheit), sowie Präferenzen bei der Wahl der Verkehrsmittel. Diese haben in dem Prototyp keine Beachtung gefunden, wären für eine Verwendung einer solchen Anwendung in realen Szenarien jedoch sinnvoll.

[1] Emulator des *Android* SDK 1.6

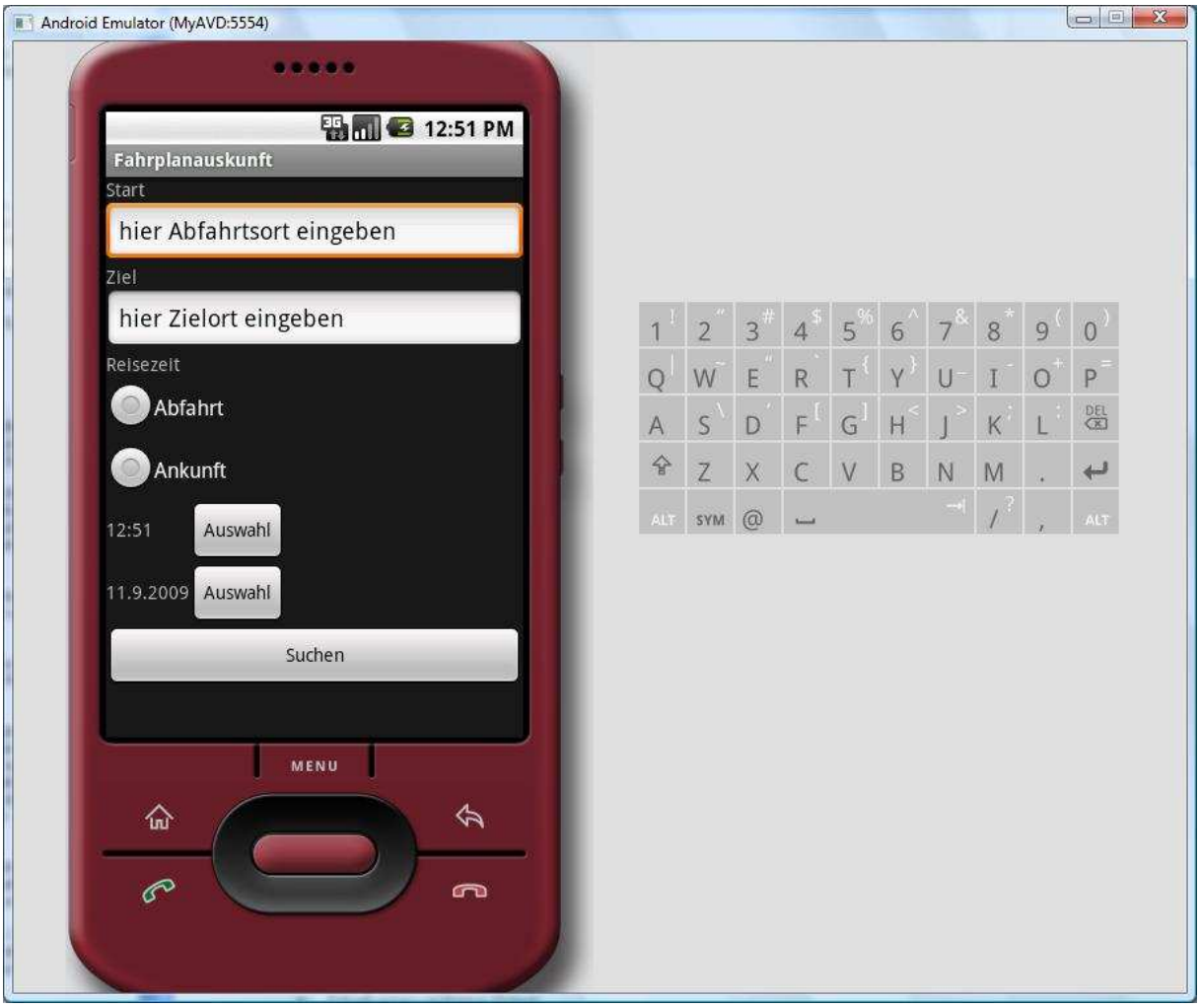

Abbildung 3.5 : Fahrplanauskuft im Emulator[1]

Als zusätzliche Ergänzung für eine bessere Benutzbarkeit könnte P an dieser Stelle das Verwenden seines aktuellen Aufenthaltsortes als Abfahrtsort A vorgeschlagen werden, wenn dieser durch Lokalisierung feststellbar ist (z.B. per GPS oder WLAN).

Nach der Anfrage erhält P als Suchergebnis (nach einer kurzen Antwortzeit) einen persönlichen Fahrplan auf dem Display des mobilen Gerätes präsentiert. Dieser enthält alle auf dem Weg befindlichen Haltestellen mit deren An- und Abfahrtszeiten, sowie der Fahrtrichtung.

Sinnvoll, jedoch in diesem Prototyp nicht beachtet, könnte dem Fahrgast an dieser Stelle vorgeschlagen werden, ein passendes Ticket für den zuvor erstellten Fahrplan zu kaufen.

Bei der Nutzung dieses Dienstes können für P zusätzliche Verbindungskosten anfallen, wenn eine mobile Datenverbindung über das Mobilfunknetz oder ein kostenpflichtiger öffentlicher W-LAN Hotspot für den Internetzugang verwendet wird.

3.2.4 Ticket kaufen

Ein Kunde K möchte ein Ticket beim Verkäufer V erwerben, und damit die Berechtigung, die Dienstleistung, also die Fahrt von A nach B, nutzen zu dürfen.

[1] Emulator in der vorherigen Version (*Android* SDK 1.5)

Voraussetzung hierfür ist neben der vorherigen Installation der Software auf dem mobilen Gerät von K (siehe 0) auch die Eingabe der Zahlungsinformationen, sowie der Angabe eines Kontrollmediums über den Anwendungsfall „Kundeneinstellungen“ (siehe 3.2.6).

Abbildung 3.6: Ticketauswahl

Der Ablauf dieses Vorgangs ist generell wie folgt:

- K wählt mit der Software das zu kaufendes Ticket aus (Abbildung 3.6)
- bestätigt die Zahlung durch die Eingabe einer PIN,
- K wartet auf den Abschluss des Vorgangs,
- erhält das Ticket von V ausgestellt und als Bestätigung angezeigt (Abbildung 3.7)
- K ist nun berechtigt die Transportdienstleistung von V zu nutzen

Die hier eingeführte PIN-Eingabe dient in erster Linie der subjektiven Sicherheit. Dabei fühlt sich der Fahrgast bei der Verwendung der Anwendung sicher, da es augenscheinlich nur ihm möglich ist, Tickets unter Verwendung seiner Zahlungsinformationen zu kaufen. Die aus informationstechnischer Sicht viel bedeutendere objektive Sicherheit dazu wird später in Kapitel 4 behandelt,

3.2.5 Fahrkartenkontrolle

Die Fahrscheinkontrolle dient der Überprüfung der Existenz eines gültigen Fahrscheins beim Kunden.

In dieser Beispielanwendung wird ein Verfahren durch stichprobenartige Kontrollen verwendet, einem in Deutschland gebräuchlichen Verfahren bei der Nutzung von Öffentlichen Verkehrsmitteln. Dazu werden entsprechende Prüfer- bzw. Prüferkolonnen eingesetzt. Diese können die Tickets jederzeit überprüfen, wenn sich der Kunde in einem fahrkartenpflichtigen Bereich aufhält.

Abbildung 3.7: Ticket anzeigen

Im Falle einer Kontrolle steht dem Fahrgast F dazu der Menüpunkt „Gekaufte Tickets anzeigen“ zur Verfügung, mit dem er das entsprechende gekaufte Ticket T aufrufen und dem Prüfer P vorzeigen kann. Diesem stehen mehrere Verfahren zu Verfügung: Sicht-, Online- und Offlinekontrolle (vgl. dazu Kapitel 2.4.2 - Mobile Ticketing).

Eine Sichtkontrolle findet dabei in jedem Fall statt. Auf diese Weise kann von P relativ schnell die Ungültigkeit oder Nicht-Existenz eines Tickets bei F festgestellt werden. Für eine weitere Gültigkeitsprüfung stehen den Fahrkartenkontrolleuren zusätzlich Online- und Offline-Prüfungsverfahren zur Verfügung.

In der Beispielanwendung wird ein Offlineprüfungsverfahren angewendet. Das Ticket verfügt dazu über einen integrierten Barcode, den F dem Kontrolleur P vorzeigt. Zusätzlich wird das Kontrollmedium vorgezeigt. Durch beide Medien und unter Zuhilfenahme eines Kontrollgerätes kann P zweifelsfrei die Gültigkeit des Tickets T feststellen.

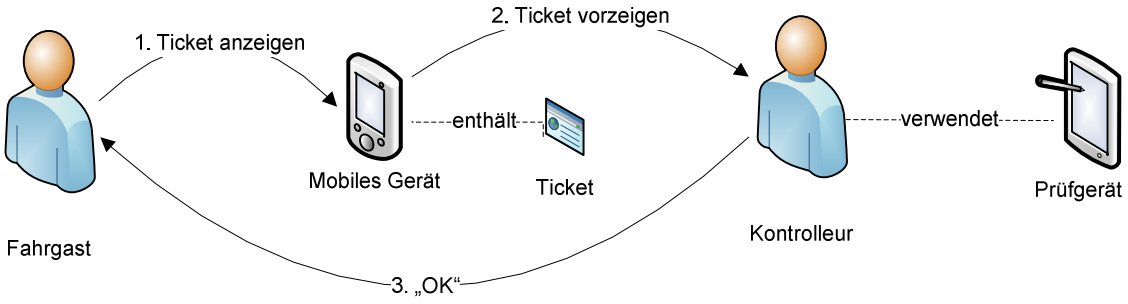

Abbildung 3.8: Kontrolle des Tickets durch den Fahrkartenprüfer

Das Kontrollgerät enthält einen Barcode-Scanner zum Auslesen der in dem Ticket enthaltenen Informationen, welche sich direkt vom Display der Endgeräte der Fahrgäste lesen lassen. Damit kann die Gültigkeit der in dem Ticket enthaltenen Sicherheitsmerkmale überprüft werden.

Mit der im Kontrolleurs-Endgerät enthaltenen kryptographischen Einheit, sowie einer zuverlässigen Barcode-Erkennung, können die Sicherheitsmerkmale schnell überprüft werden, sodass eine Kontrolle innerhalb eines kurzen Augenblicks möglich ist.

3.2.6 Kundeneinstellungen

Damit der Kauf von Tickets, wie im Anwendungsfall „Ticket kaufen“ beschrieben, stattfinden kann, ist es erforderlich, dass der Fahrgast dem System die Zahlungsinformationen, sowie auch ein Kontrollmedium, mitgeteilt hat, mit dem er seine Identität im Falle einer Fahrkartenkontrolle nachweisen kann. Dies geschieht über eine entsprechende Bildschirmmaske (siehe Abbildung 3.9).

Beide Informationen werden dabei durch den Fahrgast in das Gerät eingegeben und bleiben dort zum Zwecke einer besseren Benutzbarkeit der Anwendung gespeichert. Damit stehen sie bei zukünftigen Ticketkäufen zur Verfügung und müssen nicht erneut eingegeben werden. Zusätzliche Gefährdungen sollen jedoch durch die Speicherung keinesfalls ermöglicht werden.

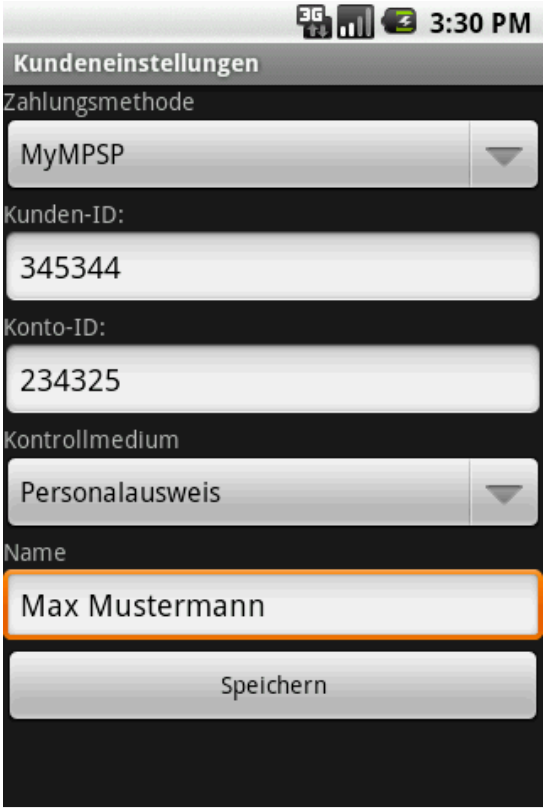

Abbildung 3.9: Kundeneinstellungen

Nach Eingabe der Informationen legt der Fahrgast hier eine PIN fest, die er später im Rahmen eines Ticketkaufs für die Bestätigung des Zahlungsvorganges benötigt. Dies verbessert zudem die subjektiv wahrgenommene Sicherheit und vermittelt dem Fahrgast ein gutes Gefühl bei Verwendung der Anwendung. Mit der aus informationstechnischer Sicht viel relevanteren objektiven Sicherheit hat dies wenig zu tun, bietet jedoch weitere Möglichkeiten für deren Umsetzung, die in Kapitel 4 behandelt wird.

3.3 Architektur

Von nun an steht die technische Seite der Beispielanwendung im Vordergrund. Im Folgenden wird die Architektur für die Beispielanwendung vorgestellt.

Hierzu wurde ein Prototyp entwickelt, der jedoch nicht funktionsfähig ist. Lediglich einige wenige relevante Systemteile wurden unter Verwendung des SDK implementiert. Für Testzwecke stand zum einen der im SDK enthaltene Emulator für *Android-Powered-Devices* und zum anderen das in Kapitel 2.2.3 vorgestellte G1 von HTC zur Verfügung.

Die Architektur der Client-Anwendung lässt sich grob in Front-End und Back-End unterteilen:

Das Front-End enthält dabei die gesamte Benutzerschnittstelle. Dazu gehören alle Eingabemasken und jegliche Komponenten, die direkt mit dem Benutzer interagieren. Deren mögliche optische Präsentation wurde bereits im vorigen Kapitel durch zahlreiche Screenshots des Prototyps veranschaulicht.

Zum Back-End gehört sämtlicher Programmablauf, der im Hintergrund abläuft, und für den Benutzer transparent stattfindet. Dazu gehören *Services*, die einen Großteil der aufwändigeren Berechnungen durchführen, wie sie für kryptographische Verfahren als Teil der Realisierung der Sicherheitsmaßnahmen im weiteren Verlauf dieser Arbeit umgesetzt werden.

3.3.1 Übersicht über Komponenten

Das System lässt sich in zwei Teilsysteme untergliedern:

- Fahrplanauskunftsdienst
- Ticketing Dienst

3.3.2 Activities

In *Android* steht als generelle Oberklasse für die Benutzeroberfläche die Klasse *Activity* zur Verfügung. Von ihr erben alle in der Beispielanwendung verwendeten Bildschirmmasken und wurden um ihre eigene Funktionalität ergänzt.

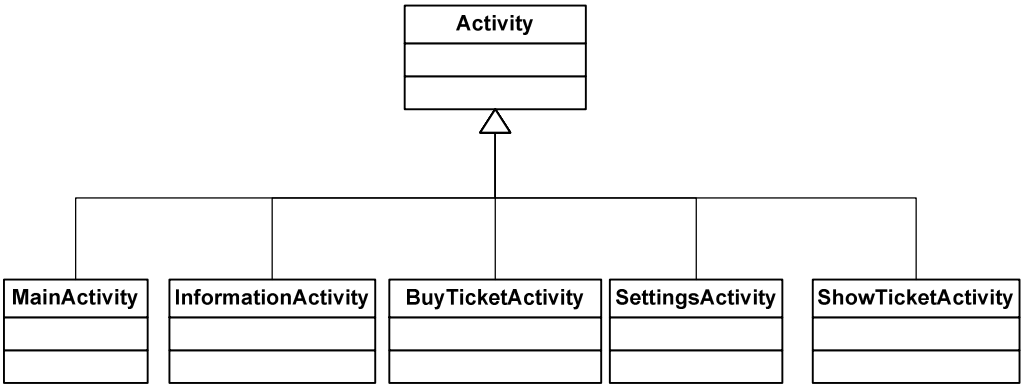

Abbildung 3.10: *Activities* als Bildschirmmasken

Die Namen der *Activities* sprechen größtenteils für sich:

- Die *MainActivity* ist das Hauptmenü, über das alle weiteren *Activities* erreichbar sind (siehe Abbildung 3.4)
- Die *InformationActivity* ist die Benutzeroberfläche zur Abfrage von Fahrplaninformationen (siehe Abbildung 3.5)
- Die *BuyTicketActiviy (*Abbildung 3.6) die *SettingsActiviy* (Abbildung 3.9) und die *ShowTicketActivity* (Abbildung 3.7) dienen der Nutzung des Ticketing Dienstes

3.3.3 Services

Im Backend befinden sich die verschiedene Dienste. Für diese bietet *Android* die Oberklasse *Service* an. Diese Programmteile werden vollständig transparent für diesen ausgeführt und besitzen keine für den Benutzer sichtbaren Komponenten, verrichten jedoch den Großteil der Arbeit.

In dem Prototyp gibt es drei verschiedene Services. Für die Fahrplanauskunft wird der *InformationService* von der *InformationActivity* verwendet.

Beim Kauf von Tickets greift dazu die *BuyTicketActicity* auf Dienste des *TicketingService* zu, welcher selber wiederum den *PaymentService* zum Abwickeln des Zahlungsverkehrs nutzt.

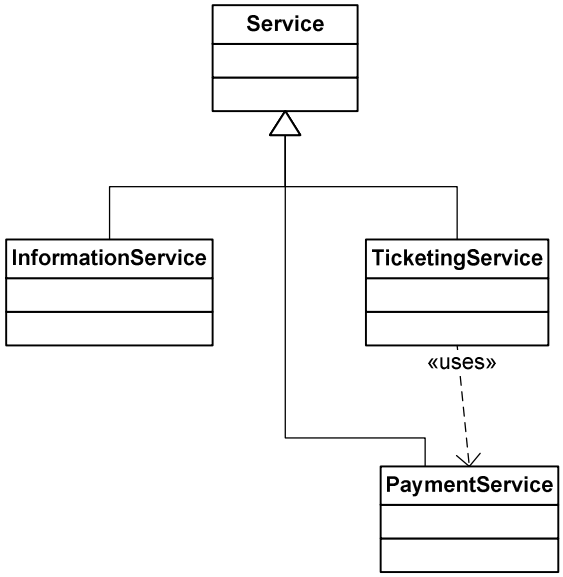

Abbildung 3.11: Services als Komponenten des Backend

3.3.4 Abläufe

Der grundlegende Ablauf der Anwendung wird durch Abbildung 3.12 verdeutlicht:

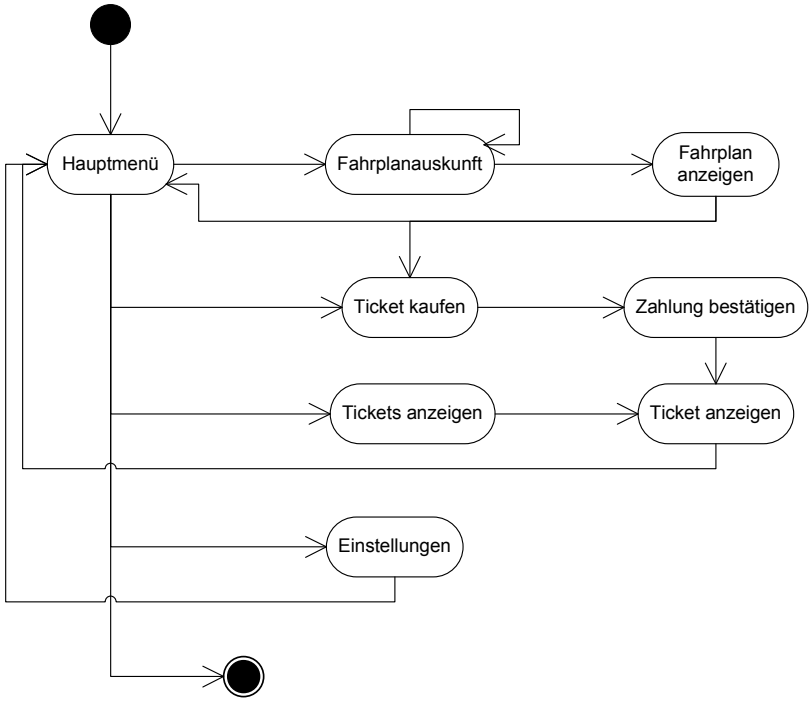

Abbildung 3.12 Programmablauf

Ticket Kaufen

Dieser Vorgang ist relativ komplex, weshalb auf ihn im Folgenden näher eingegangen wird.

Abbildung 3.13 verdeutlicht zunächst den prinzipiellen Ablauf eines Ticket-Kaufs

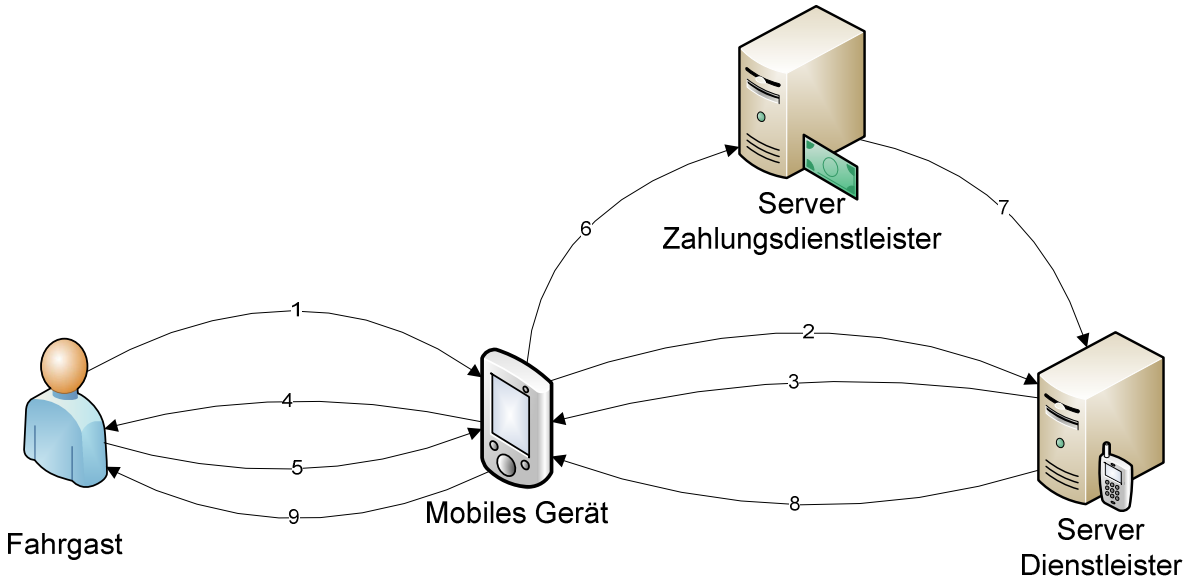

Abbildung 3.13: Ablauf eines Ticketkaufs

1. Der Fahrgast wählt ein Ticket über die Anwendung aus (*BuyTicketActivity)*
2. Die Ticketauswahl wird an den Server des ÖPV-Unternehmens übermittelt (*TicketingService)*

3. Preisauskunft und Aufforderung zur Zahlung (enthält Zahlungsreferenz)
4. Aufforderung zur Bestätigung der Zahlung durch PIN-Eingabe
5. PIN-Eingabe
6. Aufforderung des Kunden an den Zahlungsdienstleister zur Zahlung des Betrages an den Dienstleister durch Übermittlung der Zahlungsinformationen, einer Zahlungsreferenz und eines Betrages
7. Abwicklung des Zahlungsverkehrs (auf B2B-Ebene)
8. Übermittlung des Tickets an das mobile Gerät
9. Der Fahrgast erhält eine Bestätigung über den Erfolg des Ticketkaufes durch Anzeige des ausgestellten Tickets.

Die Punkte 6-8 sind dabei Teil einer Transaktion, d.h. sie finden entweder alle ordnungsgemäß statt, oder gar nicht. Damit ist sichergestellt, dass das System keine inkonsistenten Zustände erreichen kann oder unvollständige Geschäftsvorfälle ermöglicht werden. Beispielsweise könnte der Kunde nach der Zahlung keinen Fahrschein erhalten oder einen erhalten, ohne gezahlt zu haben.

Da der MPSP bei diesem Vorgang alle drei beteiligten Parteien kennt und er als vertrauenswürdige Instanz für den Kunden auch schon die Anonymisierung übernimmt, könnte er auch für die Bereitstellung eines verteilten Transaktionsdienstes sorgen. Auf diese Problematik wird nicht weiter eingegangen und sollte auf Anwendungsebene durch die Verwendung geeigneter Protokolle realisierbar sein.

Zuvor hat der Kunde im Frontend das gewünschte Ticket ausgewählt. Die weiter für diesen Vorgang benötigten Zahlungsinformationen zum Erreichen des MPSP befinden sich bereits im Speicher des Gerätes und müssen daher nicht erneut eingegeben werden.

3.3.5 Daten

Es gibt einige Datenobjekte, die die im Rahmen dieser Anwendung verarbeiteten Informationen repräsentieren. Dies sind im Einzelnen:

- Fahrplandaten
- Zahlungsinformationen
- Kontrollmedium
- Ticket
- PIN (wird sofort nach Verwendung überschrieben)

Fahrplandaten

Bei Nutzung des Fahrplanauskunftsdienstes wird ein Objekt *Fahrplananfrage* an den Server übermittelt. Dieser generiert einen *Fahrplan*, welcher zurück an die Client-Anwendung gesendet wird. Der *Fahrplan* enthält eine Reihe von Objekten des Typs *Teilstrecke.* Abbildung 3.14 zeigte die dabei verwendeten Datenobjekte:

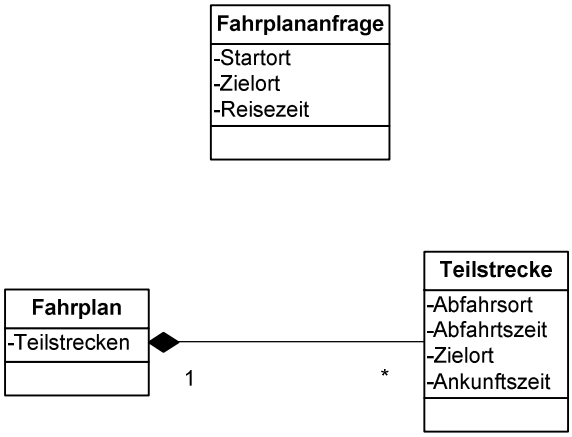

Abbildung 3.14: Fahrplandaten

Zahlungsdaten

In den Zahlungsdaten sind sämtliche Informationen zur Abwicklung des Zahlungstransfers enthalten. Dazu gehört mindestens eine Referenz oder Adresse auf einen MPSP, über welche der Zahlungsdienst mobil erreichbar ist und seine Dienste anbietet, sowie eine zusätzliche Information, über den dieser auf das bestehende Geschäftsverhältnis schließen kann, wie in Form einer Kunden- oder Kontonummer. Abbildung 2.1 verdeutlicht dies. Mit den Zahlungsdaten kann der MPSP kontaktiert werden, um die Zahlung an das ÖPV-Unternehmen im Auftrag des Fahrgastes zu tätigen.

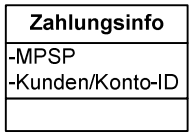

Abbildung 3.15: Zahlungsdaten

Die Zahlungsinformationen werden durch den Kunden über den Anwendungsfall „Kundeneinstellungen“ eingegeben und fortan innerhalb der Anwendung als Zahlungsdaten repräsentiert.

Ticket

Das digitale Ticket enthält ähnliche Informationen wie ein vergleichbares Papierticket. So sind darin neben einer Ticket-Nr. *TID* allgemeine Informationen über den Gültigkeitsbereich *G* gespeichert. Dieser kann räumlich, zeitlich und auf eine bestimmte Personenanzahl beschränkt sein. Weiter verfügt es über Sicherheitsmerkmale, die zur Umsetzung der Sicherheitsanforderungen an ein digitales Ticket verwendet werden:

- Digitale Signatur
- Bindung an ein Kontrollmedium

Wie in Abschnitt 3.1.3 bereits festgestellt wurde, existiert ein Zielkonflikt zwischen der vom Kunden geforderten Anonymität und der Fälschungssicherheit seitens des Dienstleisters, weshalb speziell auf den Aufbau des Tickets detaillierter eingegangen wird. Zudem gehen die Sicherheitsmerkmale des Tickets daraus näher hervor.

Das erste Sicherheitsmerkmal ist die Digitale Signatur mit einem privaten Schlüssel des ÖPV-Unternehmen. Dazu werden die *TID* und *G* verwendet. Damit ist sichergestellt, dass keine gültigen Tickets von Betrügern erzeugt oder verändert werden können, da diese nicht im Besitz des privaten Schlüssels sind. Das Ticket würde bei dem Versuch sofort seine Gültigkeit verlieren.

Allerdings kann durch die Verwendung einer digitalen Signatur *S* eine Duplizierung von bereits gültigen Tickets mit der anschließenden Weitergabe an nicht-nutzungsberechtigte Fahrgäste nicht ausgeschlossen werden, dass durch direkte Speicherzugriffe auf die entsprechenden Speicherbereiche die Sicherheitsmerkmale einfach mit kopiert werden.

Um dies zu verhindern, werden in diesem Szenario personengebundene Fahrscheine verwendet. Dazu findet eine Bindung des Tickets an ein Kontrollmedium statt. Nur mit diesem zusammen erhält es seine Gültigkeit. Dieses Medium ist bereits personenbezogen und fälschungssicher, sodass gewährleistet ist, dass die Tickets dies nun ebenfalls sind. Im Falle einer Duplizierung, welche anderen Fahrgästen eine unberechtigte Nutzungsberechtigung bieten würde, würde das Ticket somit sofort seine Gültigkeit verlieren.

Zur Umsetzung könnte das Ticket direkt eine Referenz auf das Kontrollmedium *KMR* erhalten, wie Abbildung 3.16 zeigt, da es nur mit diesem zusammen gültig sein soll. Die Fälschungssicherheit wäre damit zwar gewährleistet, jedoch kann die Anonymitätsforderung der Kunden nicht mehr eingehalten werden. *KMR* könnte personenbezogene Daten, wie beispielsweise den Namen des Fahrgastes enthalten, der durch Mitführen des Personalausweises kontrolliert werden kann.

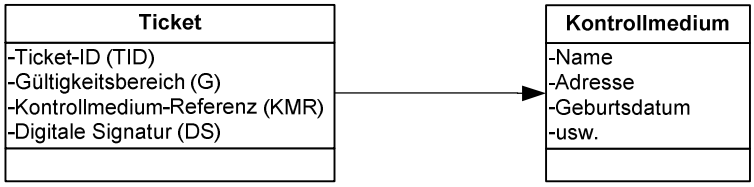

Abbildung 3.16: Nicht-anonymes personenbezogenes Ticket

Stattdessen könnte das Ticket lediglich einen Hashwert von *KMR* enthalten, wodurch jedoch lediglich eine Pseudonymisierung anstelle der gewünschten Anonymisierung stattfindet. Wird beispielsweise als *KMR* der Name einer Person verwendet, wie „Max Mustermann“, würde daraus immer der gleiche Hashwert erzeugt werden.

Zudem müsste bei einer Kontrolle die Referenz auf das Kontrollmedium durch einen Kontrolleur manuell eingegeben werden, damit der Hashwert berechnet und ein Vergleich stattfinden kann. Dies ist von einem Personalausweis[1] beispielsweise zurzeit nur manuell möglich und eine zügige Fahrscheinkontrolle wäre mit diesem Verfahren nicht realisierbar.

Daher findet statt dem Hashing eine XOR-Verknüpfung zwischen der Kontrollmedienreferenz *KMR* und einer nicht öffentlichen Zufallszahl Z statt, welche bei jedem Ticket neu erzeugt wird. Das Ergebnis dieser wird im Folgenden als Kontrollwert *KW* bezeichnet, welches Bestandteil des *Tickets* ist.

[1] Wie er in Deutschland zum Einsatz kommt

Ein Austauschen von *KMR* durch direkten Speicherzugriff könnte dennoch die unberechtigte Weitergabe von gültigen Fahrscheinen ermöglichen, indem *Z* so angepasst wird, dass es den gleichen Wert *KW* liefert. Um dies auszuschließen, wird neben *KW* auch noch der Hashwert der Zufallszahl *Hash(Z)* als Teil des Tickets gespeichert.

Die Werte *KMR* und *Z* werden bei einem Ticketkauf generiert und verbleiben als *privates Ticket* beim Fahrgast. Der daraus resultierende Aufbau ist Abbildung 3.17 zu entnehmen. Dadurch ist sichergestellt, dass von dem Ticket nicht mehr ohne das private Wissen des Fahrgastes (*Privates Ticket*) auf dessen Identität geschlossen werden kann. Das *Ticket* ist somit anonym, jedoch über das *Private Ticket* personenbezogen.

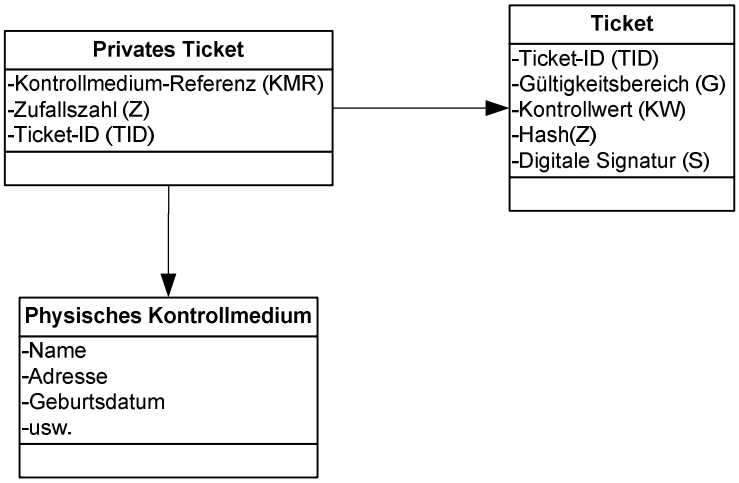

Abbildung 3.17: Anonymes personenbezogenes Ticket

Erst bei einer Fahrkartenkontrolle wird durch Vorzeigen des gesamten Tickets (*Ticket* und *Privates Ticket)* dem Unternehmen die Identität des Fahrgastes durch *KMR* bekannt gegeben. *S* ist die digitale Signatur des Betreiber-Unternehmens, wird im Rahmen eines Ticketkaufes erzeugt und signiert die Werte *TID*, *G*, *KW* und *Hash(Z)* mit dem privaten Signaturschlüssel[1]. Der zugehörige öffentliche Schlüssel muss nur den Fahrkartenkontrolleuren bekannt sein.

Bezüglich der Sicherheit ist anzumerken, dass durch die Kenntnis zweier der drei Informationen *KMR*, *Z*, oder *KW* der fehlende dritte Wert über die XOR-Verknüpfung berechnet werden kann. Dem ÖPV-Unternehmen wird über das *Ticket* jedoch nur der Kontrollwert *KW* bekannt, sodass die anderen beiden Informationen, insbesondere der *KMR*, nicht berechenbar sind.

Personenbezogenen Daten, die die Referenz *KMR* enthalten könnte, werden somit nicht weitergegeben und die Anonymität des Fahrgastes beim Ticketkauf kann erhalten bleiben, wie dies auch beim Erwerb von herkömmlichen Papiertickets am Fahrkartenautomaten der Fall ist.[2]

Dabei handelt es sich im Übrigen um eine absolut sichere Verschlüsselung nach dem *One-Time-Pad* Verfahren. Die Zufallszahl *Z* stellt dabei den geheimen Schlüssel dar und verschlüsselt *KMR*.

Vorteilhaft ist, dass bis zur Fahrkartenkontrolle kein Schlüsselaustausch stattfindet. Der Schlüssel *Z* bleibt bis zu diesem Zeitpunkt nur auf dem mobilen Endgerät gespeichert. Erst bei der Fahrkartenkontrolle wird dieser über den Barcode an den Kontrolleur übermittelt, wodurch *KMR* berechnet kann.

[1] Dieser private Schlüssel darf ausschließlich zur digitalen Signierung verwendet werden.
[2] Vorausgesetzt, es wird auch ein anonymes Zahlungsverfahren verwendet.

Für den benötigten kryptographisch sicheren Zufallszahlengenerator könnten dazu physikalische Einflussgrößen, die speziell im mobilen Umfeld zur Verfügung stehen,[1] verwendet werden.

3.4 Zusammenfassung

In diesem Kapitel wurde ein Beispielszenario dargestellt, welches für die weitere Sicherheitsanalyse verwendet werden soll.

Zunächst wurde das Umfeld des Beispielszenarios veranschaulicht und die Wahl für eine Anwendung aus dem *Mobile Ticketing* begründet. Der mobile Anwendungsbereich benötigt dabei aufgrund der dortigen Einschränkungen spezielle Anforderungen. Gleichzeitig bietet er aber auch einen Mehrwert durch das Aufheben von Beschränkungen. Weitere Entwicklungen in dieser Richtung sind zukünftig zu erwarten.

Das Beispielszenario wurde anschließend anhand von Anwendungsfällen präzisiert, aus denen die grundlegende Funktionsweise und Bedienung für Nutzer ersichtlich wurde. Anschließend folgte ein Überblick über die Architektur des dabei verwendeten Prototyps.

[1] Z.B. durch Bewegungssensoren an den Endgeräten.

4 Security Engineering

Dieses Kapitel beschäftigt sich mit dem Entwurf, der die Umsetzung der in Kapitel 3 vorgestellten Beispielanwendung hinsichtlich sicherheitsrelevanter Aspekte mit Android zum Gegenstand hat. Das Thema Sicherheit ist hier Kernthema und der bisherige Entwurf der Anwendung wird um diese erweitert.

In Kapitel 4.1 werden zunächst die Sicherheitsanforderungen für die Anwendung aufgestellt.

In Kapitel 0 (**Entwicklung und Realisierung**) werden Maßnahmen vorgestellt, um den bisherigen Entwurf der Anwendung um Sicherheitsaspekte zu erweitern.

Im Anschluss daran findet in Kapitel 0 eine Bewertung statt, in der überprüft wird, ob und wie gut sich die Sicherheitsmaßnahmen für das gewählte Szenario innerhalb der Plattform *Android* umsetzen lassen.

4.1 Sicherheitsanforderungen

Ziel dieses Abschnitts ist es, die genauen Sicherheitsanforderungen für die Anwendung aufzustellen, um daraus im weiteren Verlauf ein genaues Sicherheitskonzept zu entwickeln, welches unter Verwendung von *Android* umgesetzt werden soll.

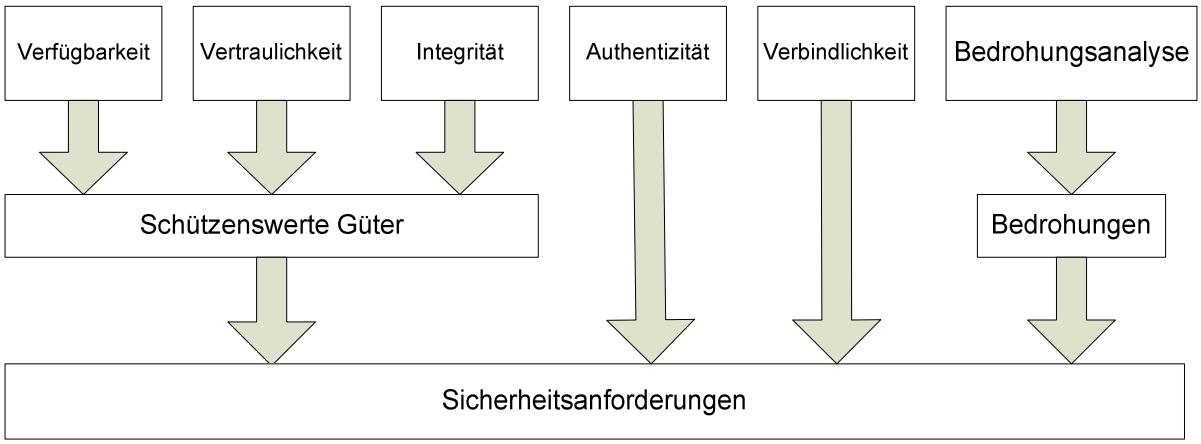

Abbildung 4.1: Aufstellen der Sicherheitsanforderungen

Zuerst wird im Rahmen einer IT-Strukturanalyse (Abschnitt 4.1.1) die verwendete IT-Landschaft analysiert, die in direktem Zusammenhang mit der Anwendung steht. Danach folgt die Identifikation der zu schützenden Werte (1.1.1), deren Schutzbedarf anhand der möglichen Schadensszenarien bestimmt wird (4.1.2).

4.1.1 IT-Strukturanalyse

Abbildung 4.2 zeigt die an dem System beteiligten Komponenten:

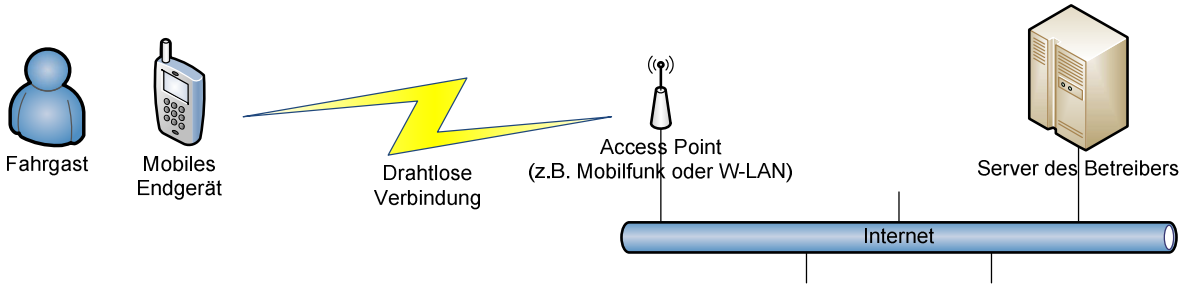

Abbildung 4.2: Netzplan der eingesetzten Komponenten

Der Fahrgast ist zwar kein eigentlicher Teil des Systems, jedoch stellt auch er einen Angriffspunkt für das Gesamtsystem dar, sodass er keinesfalls unbeachtet bleiben darf.

Bezüglich der Multiplizitäten zwischen den Komponenten können dabei alle mehrfach vorkommen, sich ein mobiles Gerät gewöhnlicherweise nur in dem Besitz von einer Person befindet.

Schwerpunkt der Analyse dabei wird die Sicherheit des mobilen Endgerätes im Rahmen des vorgestellten Anwendungsszenarios sein, mit dem darauf laufenden Client, sowie *Android* als die umgebende Plattform. Weiter werden noch die unmittelbaren „Nachbarn“, d.h. der Fahrgast sowie der zwischen Endgerät und Server liegende Kommunikationskanal, in die Analysen mit einbezogen werden. Andere Systemkomponenten, wie z.B. Server des Betreibers, sind nicht Schwerpunkt dieser Arbeit und werden nicht weiter behandelt. In einem Produktivsystem sollte dies jedoch in jedem Fall durchgeführt werden, um das Maß der geforderten Sicherheit bieten zu können.

Die Erhebung anderer IT-Komponenten würde nur im Rahmen eines umfangreichen Security Engineerings benötigt werden, wie es beispielsweise in einem Unternehmen stattfinden würde, bei dem die gesamte IT-Landschaft abgesichert werden soll. Ziel dieser Arbeit ist jedoch nur der sichere Betrieb der im vorherigen Kapitel vorgestellten Anwendung, weshalb sich die IT-Strukturanalyse lediglich auf die im Rahmen dieser Anwendung verwendeten Systeme beschränkt.

1.1.1 Identifikation schützenswerter Informationen

Im nächsten Schritt der Sicherheitsanalyse sollen die zu schützenden Informationen identifiziert werden. Alle Informationen, die im Zusammenhang mit der Anwendung verarbeitet werden, sind potenziell gefährdet, und sollten daher auf ihren Schutzbedarf hin untersucht werden. Eine Übersicht über die verarbeiteten Informationen in den verschiedenen Anwendungsfällen liefert dazu Tabelle 4.1. Die genaue Beschreibung dieser erfolgt im Anschluss.

Anwendungsfall	**Verarbeitete Informationen**
Softwareinstallation	Anwendungsdaten
Fahrplanauskunft	Fahrplaninformationen
Ticket kaufen	Ticketinformationen Kundeninformationen Zahlungsinformationen
Tickets anzeigen	Ticketinformationen Kundeninformationen
Kundeneinstellungen	Zahlungsinformationen Kundeninformationen
Fahrkartenkontrolle	Ticketinformationen Kundeninformationen

Tabelle 4.1: In den Anwendungsfällen verarbeitete Informationen

Fahrplaninformationen

Die Fahrplaninformationen umfassen zum einen alle Eingaben, die der Nutzer zur Abfrage einer Route an den Dienst stellt, wie im Anwendungsfall „Fahrplanauskunft“ beschrieben, und zum anderen den daraufhin vom Dienstleistungsserver generierten Fahrplan.

Dazu gehören als Benutzereingaben mindestens der Start- und der Zielort, sowie der Zeitpunkt der Abreise bzw. der Ankunft. Ebenfalls dazu gehören würden alle weiteren Informationen, die zu einer individuelleren Fahrplanauskunft führen könnten, jedoch im Rahmen des Prototyps nicht verwendet werden.

Der als Suchresultat vom Server erstellte persönliche Fahrplan ist ebenfalls Teil der Fahrplaninformationen und enthält Details über die zu nehmende Route samt Teilstecken und Zeitangaben. Er wird dem Benutzer über das Display präsentiert.

Zahlungsinformationen

Die Zahlungsinformationen enthalten alle zur Bezahlung des Tickets notwendigen Informationen.

Der Kunde ist dabei bereits vor Verwendung des Systems im Besitz der Zahlungsinformationen, die er von dem Zahlungsdienstleister (MPSP) seines Vertrauens erhalten hat, wie beispielsweise seiner Hausbank, die mobile Bezahldienste anbietet.

Dazu gehören mindestens Informationen zur Erreichbarkeit des MPSP, sowie eine Referenz auf das dortige Konto des Kunden. Dies können beispielsweise die URL der Bank und eine Kontonummer sein, aber auch Kreditkartennummern wären denkbar. Damit können Abbuchungen dem Kunden zugeordnet werden, sodass er diese im Rahmen seines Geschäftsverhältnisses mit dem Zahlungsdienstleister begleichen kann.

Die Zahlungsinformationen stellen damit eine gemeinsame schützenswerte Information zwischen dem Kunden und dem MPSP dar und werden nicht weitergeben. Bei der Zahlung erfüllt der MPSP zudem eine anonymisierende Funktion, indem er keine personenbezogenen Daten des Kunden an das ÖPV-Unternehmen preisgibt.

Ticketinformationen

Die Ticketinformationen umfassen alle Informationen des Datenobjektes *Ticket*, wie es in Abschnitt 3.3.3 beschrieben wurde. Die Bedeutung des daraus hervorgehenden Wissens steht hierbei im Vordergrund und nicht deren Repräsentation innerhalb des Systems.

Ticket
-Ticket-ID (TID) -Gültigkeitsbereich (G) -Kontrollwert (KW) -Hash(Z) -Digitale Signatur (S)

Tabelle 4.2: Digitales Ticket

Es handelt sich dabei um gemeinsame Informationen zwischen Fahrgast und ÖPV-Unternehmen. Das Ticket wird beim Erwerb durch den Fahrgast vom Server im Rechenzentrum des ÖPV-Unternehmens erzeugt und über einen Kommunikationskanal auf das mobile Endgerät des Fahrgastes übermittelt. Dort wird es abgelegt und verbleibt im Speicher, bis es gelöscht wird.

Kundeninformationen

Zu den Kundeninformationen gehörten die Referenz auf das physische Kontrollmedium, sowie die privaten Teile der Tickets (*Privates Ticket*), wie in Abschnitt 3.3.3 beschrieben, die den Bezug zwischen den *Tickets* und einer natürlichen Person herstellen können.

Die Referenz auf das Kontrollmedium kann dabei bereits persönliche Informationen enthalten, wie beispielsweise der Name der Person, und verbleibt ausschließlich auf dem mobilen Gerät des Kunden. Sie wird lediglich für die Erstellung von *Privaten Tickets* genutzt, verbleibt als Teil des Gesamttickets auf dem Endgerät des Kunden und verlässt das Gerät nur im Rahmen einer Fahrkartenkontrolle durch autorisierte Kontrolleure.

Anwendungsdaten

Die Anwendungsdaten umfassen alle technischen Daten, die unmittelbar mit der technischen Umsetzung der Client-Anwendung in Zusammenhang stehen. Dazu gehören der Programmcode, Informationen über die Funktionsweise der Anwendung und die darin verwendeten Algorithmen. Weiter umfassen sie den Inhalt des Speichers zur Laufzeit der Anwendung.

Personenbezogene Daten des Kunden

Die persönlichen Daten des Kunden stellen für diesen eine schützenswerte Information dar. Sie umfassen den Namen und die Adresse des Kunden, aber auch andere Daten wie Telefonnummer, Personalausweisnummer oder gar biometrische Merkmale. Zudem gibt es für die Verarbeitung personenbezogener Daten gesetzliche Regelungen, die zwingend eingehalten werden müssen und daher bei der Realisierung eines Informationssystems vor unautorisiertem Zugriff geschützt werden sollten.

In der Beispielanwendung wird weitestgehend auf die Verarbeitung von personenbezogenen Daten verzichtet, wie dies auch bei dem Kauf eines herkömmlichen Fahrscheins am Automaten der Fall ist. Nur durch die Angabe eines Kontrollmediums durch den Fahrgast, mit dem die Fälschungssicherheit des Tickets gewährleistet werden kann, werden persönliche Daten verwendet.

Auch die Zahlungsinformationen können Rückschlüsse auf die Identität des Fahrgastes liefern, werden dem ÖPV-Unternehmen jedoch nicht preisgegeben, wenn der MPSP dessen Anonymität gewahrt.

Diese persönlichen Informationen werden bei der Bewertung nicht gesondert betrachtet, sondern im Rahmen der *Privaten Daten des Kunden* und der *Zahlungsinformationen* auf ihren Schutzbedarf hin untersucht.

Weitere ableitbare Informationen

Aus der Kenntnis von Einzelinformationen bei einer Fahrplanauskunft können weitere Informationen abgeleitet werden. So werden Anwender den Fahrplanauskunftsdienst normalerweise mit der Intention nutzen, sich über eine bestimmte Verkehrsverbindung zu informieren und nicht rein willkürlich. Sind sie mit dem Ergebnis zufrieden, besteht die Möglichkeit, dass sie selbst oder andere Personen die Reise über die vom System vorgeschlagene Route antreten werden.

So können aus einer solch schlichten Fahrplanauskunft Annahmen über den Aufenthaltsort von Personen getroffen werden. Weiter werden präzisere Aussagen über zukünftige Bewegungen von Personen innerhalb des Verkehrssystems vorrausagbar, zu der das Ziel, sowie alle Orte auf dem Weg mit einer relativ genauen Zeitangabe gehören.

Auch aus den Ticketinformationen lassen sich weitere Informationen ableiten. So kann man in der Regel auch davon ausgehen, dass sich diese Person in der gewählten Region und Zeitspanne, innerhalb der das Ticket gültig ist, aufhalten wird, um die Dienstleistung zu nutzen, da sie zuvor mit dem Kauf des Tickets die Berechtigung dazu erworben hat.

Folgt unmittelbar nach Nutzung des Fahrplanauskunftsdienstes ein Ticketkauf, so kann dieser Effekt weiter verstärkt auftreten, sodass noch präzisiere Annahmen über die Aufenthaltsorte der Fahrgäste möglich werden. In diesem Fall könnte man davon ausgehen, dass die zuvor vorgeschlagene Route für den jeweiligen Fahrgast ein zufriedenstellendes Ergebnis dargestellt hat und dieser sich daraufhin ein Ticket gekauft hat, um die Reise über genau diese Route anzutreten.

Nicht nur aus den Fahrplanauskunftsdaten und den Ticketinformationen selbst können weitere Informationen abgeleitet werden, sondern diese können auch bei der reinen Nutzung der Dienste entstehen, unabhängig davon, welche Daten dabei übertragen werden. So beispielsweise Informationen über Zeit und der Ort der Nutzung, wenn eine der Dienstleistungen über den lokalen Hotspot der Haltestelle abgerufen wird.

Durch die technische Infrastruktur allein wird die Anonymität gewährleistet, da der Fahrgast dem System bei Nutzung keine Identität hinterlassen muss. Die Anonymität kann jedoch in der Realität schnell verloren gehen. Befindet sich der Fahrgast alleine an besagter Position, so kann darauf geschlossen werden, dass genau er für den stattfindenden Datenverkehr verantwortlich ist. Folglich kann ein Bezug zwischen den übertragenen Daten und einer bestimmten natürlichen Person hergestellt werden.

Es handelt sich dabei um eine unidirektionale Abbildung der Form *Ort* → *Person*. Durch mangelnde Interferenzkontrolle kann auch auf die umgekehrte Form *Person* → *Ort* geschlossen werden, wodurch der Aufenthaltsort von bestimmten Personen gezielt ausfindig gemacht werden könnte. Im Folgenden wird der Gebrauch dieser Möglichkeit nicht weiter berücksichtigt, da die Erhebung der zusätzlich benötigten Daten außerhalb des analysierten technischen Systems erfolgen müsste.

Folglich müssen alle hier vorgestellten ableitbaren Informationen aus den vorhandenen Daten bei der Einordnung in eine Schutzbedarfskategorie (Kapitel 4.1.2) Berücksichtigung finden, werden jedoch nicht gesondert betrachtet.

Physisches Kontrollmedium

Das physische Kontrollmedium befindet sich außerhalb des technischen Systems dieser Anwendung und wird daher in der folgenden Sicherheitsanalyse nicht weiter berücksichtigt. Es wird vom Kunden mitgeführt, um im Falle einer Fahrkartenkontrolle verfügbar zu sein. Es enthält Informationen, die bei Nutzung des Dienstes nur dem Kunden bekannt bleiben, mit Ausnahme einer Fahrscheinkontrolle, bei der sie zum Überprüfen der Gültigkeit verwendet werden.

Analog zu den Zahlungsinformationen ist der Kunde bereits vor Verwendung der Anwendung in Besitz dieser Information, die je nach Art des Mediums sehr eng in Verbindung mit seiner eigenen Identität steht, da es sich im Gegensatz zur Kontonummer hier um den Personalausweis handeln kann.

Im Rahmen der Anwendung kann natürlich nicht das physische Medium selbst gespeichert werden, weshalb nur eine Referenz auf dieses verwendet wird. Sie gibt zu erkennen, dass es sich um das geforderte Kontrollmedium handelt. Dabei kann es sich beispielsweise um den Namen der Person handeln, oder auch die Personalausweisnummer, die leicht mit dem physischen Kontrollmedium abgeglichen werden kann.

Die vollständige Anonymität geht für den Fahrgast für einen kurzen Augenblick verloren, so wie es auch bei einer personenbezogenen Fahrscheinkontrolle eines Fahrgastes mit Papierticket der Fall wäre.

Zusammenfassung

Die im Rahmen dieses Anwendungsszenarios verwendeten Informationen wurden nun identifiziert. Dies sind im Einzelnen:

- Fahrplanauskunftsinformationen
- Zahlungsinformationen
- Ticketinformationen
- Kundeninformationen
- Anwendungsdaten

Die personenbezogenen Daten, die ableitbaren Informationen, sowie die außerhalb des technischen Systems vorhandenen (physischen) Kontrollmedien werden im Folgenden nicht gesondert betrachtet und finden im Rahmen der anderen Informationen Berücksichtigung.

4.1.2 Schutzbedarfsfeststellung

Nach Hannes Federrath [Federr09] bestehen die allgemeinen Sicherheitsanforderungen bei mobilen Systemen aus dem Schutz der Vertraulichkeit, dem Schutz der Integrität, sowie dem Schutz der Verfügbarkeit. So auch in diesem Anwendungsszenario des Mobile Ticketing.

Nachfolgend soll für die im vorherigen Abschnitt identifizierten schützenswerten Güter der benötigte Schutzbedarf für die Schutzziele ermittelt werden. Dazu zeigt Tabelle 4.3 eine Matrix, die durch die Schutzbedarfsbestimmung mit Schutzbedarfskategorien gefüllt werden soll.

Schutzbedarf	Verfügbarkeit	Vertraulichkeit	Integrität
Fahrplaninformationen	?	?	?
Zahlungsinformationen	?	?	?
Ticketinformationen	?	?	?
Kundeninformationen	?	?	?
Anwendungsdaten	?	?	?

Tabelle 4.3: Bedrohungen für schützenswerte Güter und Schutzziele

Angelehnt an die Vorgehensweise des BSI [BSI08a] verwende werden für die Kategorisierung die drei Schutzbedarfskategorien „Normal“, „Hoch“ und „Sehr hoch“ verwendet.

Wichtig ist jeweils die „gefährlichste Bedrohung“, da sie allein über die für die Einstufung in eine Schutzbedarfskategorie entscheidet.

4.1.2.1 Verfügbarkeit

Fahrplaninformationen

Bei dem Verlust der Verfügbarkeit der Fahrplaninformationen stehen alle Informationen, die über den Fahrplanauskunftsdienst abgerufen werden können, nicht mehr zur Verfügung. Fahrgästen kann in diesem Fall keine mobile Fahrplanauskunft mehr erteilt werden.

Mangels Information über die angebotenen Verbindungen könnten schlimmstenfalls Fahrgäste auf andere Verkehrsmittel umsteigen, sollte es den Fahrgästen nicht gelingen, die Routeninformationen auf anderem Wege in Erfahrung zum bringen. Unmittelbare Einnahmeausfälle wären in diesem Schadensszenario die Folge. Weiter könnte durch eine gestiegene Unzufriedenheit seitens der Kunden eine negative Außenwirkung des Unternehmens die Folge sein, sodass das Unternehmensimage leidet.

Die konkrete Schadenshöhe wird stark davon beeinflusst, wie viele Fahrgäste die mobile Fahrplanauskunft tatsächlich in der Zeit des Ausfalls nutzen würden. Bei nur sehr geringer Nutzung des Systems, wie es in Markteinführungssituationen der Fall wäre, könnte auf eine zeitweilige Verfügbarkeit des Dienstes verzichtet werden.

Der Schaden bleibt somit zunächst überschaubar und ein Schutzbedarf der Kategorie **normal** ist für die Fahrplaninformationen angemessen.

Zahlungsinformationen

Stehen die Zahlungsinformationen dem System nicht zur Verfügung, so können keine mobilen Bezahldienste über den externen Zahlungsdienstleister MPSP abgewickelt werden.

Als eine Konsequenz daraus können keine Fahrscheine mehr verkauft werden, da der Zahlungsverkehr über den MPSP im Rahmen dieses Geschäftsvorganges stattfindet. Die Bezahlung der Tickets ist obligatorisch und ohne ihn kann der Ticketerwerb nicht stattfinden. Unmittelbare Einnahmeausfälle wären die Folge für das ÖPV-Unternehmen, wenn der Kunde als eine Konsequenz daraus von dem Kauf eines Fahrscheins gänzlich absehen würde.

Zum Anderen ist auch der Fahrgast benachteiligt, da er nun dazu gezwungen wird, die Eingabe der Zahlungsinformationen über den Anwendungsfall „Kundeneinstellungen“ (vgl. Abschnitt 3.2.6) erneut vorzunehmen oder alternative Möglichkeiten zum Erwerb eines Tickets zu nutzen, die ein anderes Bezahlverfahren verwenden.

Findet die Nichtverfügbarkeit der Zahlungsinformationen in einem größeren Rahmen statt, sodass diese bei vielen Kunden über einen längeren Zeitraum hinweg nicht verfügbar sind, so sind die Folgen weit bedrohlicher.

In Markteinführungssituationen wird der mobile Ticketingdienst sicherlich mit einer eher niedrigen Frequenz betrieben, sodass eine Verfügbarkeit als nicht so wichtig angesehen werden muss, da parallel betriebene Ausweichmöglichkeiten zum Kauf eines Fahrscheines zur Verfügung stehen und die Folgen überschaubar bleiben.

Einen Schutzbedarf von **normal** für die Verfügbarkeit der Zahlungsinformationen ist daher für angemessen.

Ticketinformationen

Wie auch bei einem Papierticket sollte dieses digitale Ticket solange verfügbar sein, wie eine Fahrscheinkontrolle zu erwarten ist, und dürfte aus softwaretechnischer Sicht nicht einfach verloren gehen können.

Ein Verlust der Verfügbarkeit der Ticketinformationen ist mit dem Verlust des digitalen Tickets gleichzusetzen und deren Folgen sind mit dem Verlust eines Papiertickets vergleichbar. Im Falle einer Fahrscheinkontrolle könnte der Fahrgast zunächst kein Ticket vorweisen, sodass durch den Verlust des Tickets finanzielle und rechtliche Konsequenzen zu erwarten sind.

Da das Ticket auf dem Endgerät gespeichert wird, ist die Verfügbarkeit der Ticketinformationen streng an die Verfügbarkeit des mobilen Gerätes gekoppelt und kann durch diese eingeschränkt werden, zum Beispiel durch einen leeren Akku. Weiterhin zieht ein Diebstahl oder ein sonstiger Verlust des Gerätes auch selbiges des Tickets nach sich.

Weiter könnten die Ticketinformationen gar nicht erst erzeugt werden, wenn der zugehörige mobile Ticketingdienst ausgefallen ist und nicht erreichbar ist. Fahrgästen wäre damit der Kauf von Tickets unmöglich.

Die Nicht-Verfügbarkeit der Ticketinformationen gilt es durch entsprechende Maßnahmen zu verhindern und stellt einen Schutzbedarf der Kategorie **hoch** dar.

Kundeninformationen

Einschränkungen in der Verfügbarkeit der Kundeninformationen sowie der Zahlungs- und Ticketinformationen führen zum Verlust der Nutzbarkeit des mobilen Ticketingdienstes und sind daher ähnlich zu bewerten.

Zunächst ist es nicht möglich, Tickets auf dem mobilen Gerät zu erstellen, da bei diesem Vorgang das Ticket fest an eine bestimmte Person gebunden werden soll mit der es gültig ist. Diese Information kann jedoch ohne Weiteres durch den Fahrgast erneut eingegeben werden, sodass lediglich leichte Einschränkungen in der Benutzbarkeit der Anwendung hinzunehmen sind.

Sind die Kundeninformationen hingegen bei einer Kontrolle nicht verfügbar, so sind die Folgen gravierender. Der Kontrolleur kann die Gültigkeit des Tickets nicht mehr feststellen. Lediglich das erste Sicherheitsmerkmal, die digitale Signatur, kann geprüft werden. Das Kontrollmedium hingegen kann nicht überprüft werden, da die in den *privaten Tickets* enthaltene Referenz auf dieses nicht verfügbar ist. Kontrollen wären damit zwar noch durchführbar, jedoch müssten Abstriche in der Fälschungssicherheit der Tickets gemacht werden.

Digitale Tickets könnten mitsamt der digitalen Signatur dupliziert, und an beliebig viele andere Fahrgäste weitergegeben werden, ohne dabei ihre Gültigkeit zu verlieren. Findet dies in einem größeren Umfang statt und können Angreifer ihr Wissen von der Anfälligkeit des Systems auf Weise nutzen, so können in einem solchen Szenario enorme wirtschaftliche Schäden für das Unternehmen entstehen.

Ein kurzzeitiger Verzicht auf die Kundeninformationen wäre also „verschmerzbar“, eine längere Ausfallzeit hingegen ist inakzeptabel, falls Wissen über die Angreifbarkeit des Systems bekannt würde und Betrüger davon Gebrauch machen. Der Schutzbedarf der Kundeninformationen ist daher mit **hoch** einzustufen.

Anwendungsdaten

Sind die Anwendungsdaten nicht verfügbar, so kann die Client-Anwendung nicht ausgeführt werden. Folglich kann auch keiner der Dienste genutzt werden. Der Verlust der Verfügbarkeit der Anwendungsdaten ist daher gleichzusetzen mit dem Verlust jeglicher Funktionalität, die die Anwendung bietet. Je nach Zeitpunkt des Ausfalls können verschiedene Schäden entstehen.

Sind die Anwendungsdaten im Falle einer Fahrscheinkontrolle nicht verfügbar, so können keine Tickets mehr vorgezeigt werden. Dieser Fall sollte daher keinesfalls eintreten und ist daher mit der Schutzbedarfskategorie **hoch** zu bewerten.

Über die Schuldfrage lässt sich streiten, wenn beispielsweise aufgrund eines leeren Akkus am mobilen Gerät die Anwendungsdaten nicht verfügbar sind, und eine Fahrkartenkontrolle nicht durchgeführt werden kann. Bei einem vergleichbaren Papierticket kann dies nicht passieren. Das digitale Ticket könnte jedoch durch Aufnehmen der Personalien des Fahrgastes auch „nachträglich vorgezeigt“ werden, wenn die Energieversorgung kurzfristig nicht wiederhergestellt werden kann.

4.1.2.2 Vertraulichkeit

Fahrplaninformationen

Eine Verletzung der Vertraulichkeit der Fahrplaninformationen führt dazu, dass Dritte Kenntnis davon erlangen, dass irgendeine Person bestimmte Routeninformation abgefragt hat. Ein Bezug zu einer konkreten natürlichen Person kann durch diese Informationen nicht hergestellt werden, sodass die Fahrgäste bei der Nutzung des Dienstes anonym bleiben. Diese Tatsache an sich ist als unkritisch einzustufen und die Vertraulichkeit dieser Informationen bedarf keines besonderen Schutzes.

Der Schutzbedarf ist daher als **normal** einzuordnen.

Die Privatsphäre des Kunden wird durch den Inhalt der Daten selbst zwar nicht gefährdet, kann jedoch durch zusätzliche Informationen, wie Aufenthaltsort und Zeitraum der Nutzung, sowie aus den zuvor vorgestellten ableitbaren Informationen, gefährdet werden, sodass die Anonymität[1] gefährdet wird.

Sollte ein Bezug zwischen den Fahrplaninformationen und einer bestimmten natürlichen Personen hergestellt werden können, so besteht die Gefahr, dass diese zunächst lokalisiert werden und mittels der angeforderten Informationen auch auf dessen geplante Reiseroute geschlossen werden kann. Dadurch könnten Bewegungsprofile erstellt werden, was einen erheblichen Einschnitt in die Privatsphären der einzelnen Fahrgäste zur Folge hätte.

Würde es durch die Hinzunahme von zusätzlichen Informationen, die erst außerhalb des Systems gewonnen werden müssten, gelingen, die Nutzer des Fahrplanauskunftsdienstes zu identifizieren, wäre hier ein Schutzbedarf der Stufe **hoch** eher angemessen. Zudem könnte das Unternehmen bei nicht vertrauenswürdigem Umgang mit diesen Informationen einen Imageschaden erleiden, sodass das Gesamtsystem vom Kunden nicht akzeptiert wird.

Zahlungsinformationen

Die Vertraulichkeit der Zahlungsinformationen stellt ein besonders schützenswertes Gut für den Kunden dar. Es liegt auf der Hand, dass diese leicht missbraucht werden könnten und für den Kunden dadurch finanzielle Schäden entstehen. Durch die unberechtigte Nutzung der Zahlungsinformationen kann sich ein Krimineller einen unmittelbaren finanziellen Vorteil verschaffen, der zu Lasten des Kunden erfolgt.

Das Schadenspotential kann dabei von dem Verlust eines unbedeutenden Betrages für den Kunden bis hin zur Überziehung des Kontos reichen. Sinnvolle Gegenmaßnahmen wären hier Volumenbeschränkungen auf den Micropayment-Bereich einzuführen, die zwischen Zahlungsdienstleister und Kunde vereinbart werden.

Ein konkretes Motiv für Täter besteht in der finanziellen Bereicherung, sodass gezielt versucht werden könnte, in den Besitz von Zahlungsinformationen zu geraten. Durch hohe kriminelle Energie seitens des Täters könnte selbst ein großer Aufwand in Kauf genommen werden, um vorhandene Sicherheitsvorkehrungen zu durchbrechen. Ein solcher Angriff kann sich prinzipiell gegen jeden Kunden richten, sodass ein Angreifer bei der Wahl seiner Opfer wohl kaum zwischen verschiedenen Personen unterscheiden wird. Jeder Kunde ist somit dieser Bedrohung ausgesetzt. Die Wahrscheinlichkeit eines solchen Angriffs ist also hoch.

[1] In der Realität kann die Anonymität jedoch in Einzelfällen verloren gehen. Steht beispielweise nur eine Person an einer Haltestelle und fragt über den dortigen W-LAN -Hotspot Routeninformationen ab, so könnte ein Angreifer, der diese Informationen über das drahtlose Netzwerk mitliest, auf den zukünftigen Weg von genau dieser einen Person schließen.

Neben dem finanziellen Schaden des Kunden könnten über die Zahlungsinformationen auch Rückschlüsse auf seine Identität gezogen werden, wenn der MPSP diese herausgibt, sodass indirekt auch die Anonymität der Fahrgäste gefährdet ist. Weiter darf bei der Verarbeitung von den in den Zahlungsinformationen personenbezogenen Daten nicht gegen geltende Gesetze verstoßen werden.

Der Schutzbedarf der Zahlungsinformationen hinsichtlich der Vertraulichkeit ist sowohl durch die möglichen eintretenden finanziellen Schäden, als auch des Verlustes persönlicher Informationen mit einem Schutzbedarf der Kategorie **hoch** zu bewerten.

Ticketinformationen

Wie bereits die Vertraulichkeit der Fahrplanauskunftsdaten ist der Schutzbedarf der Ticketinformationen bezüglich der Anonymität des Fahrgastes und dessen Lokalisierbarkeit aufgrund der daraus ableitbaren Informationen ähnlich zu bewerten.

Begründeten Schutzbedarf der Ticketinformationen an Verdecktheit oder Unbeobachtbarkeit konnte nicht festgestellt werden. Zumal ist dies bei den herkömmlichen Papiertickets ebenfalls nicht gegeben, da jeder Fahrgast beim Kauf eines Tickets beobachtet werden könnte.

Die Ticketinformationen sind zwar mit der Identität eines Kunden verknüpft, enthalten diese jedoch nicht, sodass durch den Verlust der Vertraulichkeit keine persönlichen Informationen über den Fahrgast bekannt werden. Dies ist vergleichbar mit dem Verlust eines Papiertickets. Bei diesem ist auch nicht feststellbar, wem es gehörte.

Der Schutzbedarf der Ticketinformationen ist in punkto Vertraulichkeit daher mit **normal** einzustufen.

Kundeninformationen

Durch den Verlust der Vertraulichkeit der Kundeninformationen gibt der Fahrgast persönliche Informationen über seine Identität preis. Diese gilt es aus Gründen der Privatheit und des Rechts an informationeller Selbstbestimmung zu schützen. Ferner muss das Datenschutzgesetz eingehalten werden, sodass der Schutzbedarf hier mit **hoch** einzustufen ist.

Anwendungsdaten

Durch den Verlust der Vertraulichkeit der Anwendungsdaten ist es einem Angreifer möglich sämtliche Aktivitäten der Anwendung mitzulesen. Dies umfasst Speicherabbilder zur Laufzeit, in denen auch die zuvor untersuchten Informationen durch deren Repräsentation als Datenobjekte enthalten sind.

Nach der Theorie, dass das schwächste Glied einer Kette für die Gesamtsicherheit dieser verantwortlich ist, sollte der Schutzbedarf dieser Daten mindestens mit dem höchsten der vier zuvor untersuchten Informationen (Fahrplanauskunfts-, Ticket- und Zahlungs- und Kundeninformationen) bewertet werden.

Neben dem Speicher zur Laufzeit könnte auch der Code der Anwendung Ziel eines Angriffs werden, wodurch die verwendeten Techniken und Algorithmen offengelegt würden. Durch diese zusätzlichen Kenntnisse können weitere gezieltere Angriffe auf die Anwendung vorgenommen werden, sodass ein Angreifer beispielsweise manipulierte Plagiate der Anwendung zu Phishing-Zwecken in Umlauf bringen könnte.

Daher ist Schutzbedarf der Vertraulichkeit der Anwendungsdaten mit **hoch** einzustufen.

4.1.2.3 Integrität

Fahrplaninformationen

Die Integrität der Fahrplaninformationen ist für die zuverlässige Verwendung des Fahrplanauskunftsdienstes erforderlich. Wird diese verletzt, so erhalten die Fahrgäste keine oder falsche Fahrplaninformationen. Durch Manipulation der Suchanfrage des Nutzers oder dem daraus generierten persönlichen Fahrplan ist mit keiner richtigen Antwort des Dienstes mehr zu rechnen. Im schlimmsten Fall würde der Fahrgast einer falschen Route folgen und sein Reiseziel womöglich nicht rechtzeitig erreichen können, sodass weitere Schäden entstehen können.

Durch gezielte Manipulationen der Fahrplaninformationen könnten einzelne Personen umgeleitet werden, damit sie ein bestimmtes Ziel gar nicht, oder anders erreichen. Ein Angreifer wird sich dabei persönlich jedoch nicht bereichern können, sondern auf diese Art eher nur anderen Personen einen Schaden zufügen können. Deswegen scheinen Angriffe gegen Integrität der Fahrplaninformationen eher unrealistisch.

Können diese jedoch in einem größeren Stil erfolgen und dabei beliebig viele Fahrgäste geschädigt werden, so könnten für das Unternehmen ökonomische Probleme entstehen, da zum einen durch publizierte Schadensmeldungen ein negatives Bild in der Öffentlichkeit erzeugt wird und zum anderen die Echtheit und Glaubwürdigkeit zukünftiger Fahrplaninformationen durch die Fahrgäste sinkt.

Die Forderung nach einzuhaltender Integrität der Fahrplaninformationen ist daher **hoch**, zumal das Unternehmen für weitere Schäden ursächlich falscher Falschinformationen haftbar gemacht werden könnte.

Zahlungsinformationen

Kann die Integrität der Zahlungsinformationen nicht eingehalten werden, so drohen sowohl dem Kunden als auch dem ÖPV-Unternehmen unvollständige, falsche oder im besten Fall „nur“ fehlgeschlagene Geschäftsfälle.

Ähnlich zur Verletzung der Vertraulichkeit der Zahlungsinformationen können auch hier Fahrgäste finanzielle Schäden durch gezielte Manipulationen der Zahlungsinformationen erleiden, da Beträge oder Ziel der Zahlungen verändert werden können.

Die Integrität der Zahlungsinformationen besitzt daher einen **hohen Schutzbedarf**.

Ticketinformationen

Das Ticket bzw. der Fahrschein stellt in diesem Szenario die Berechtigung zur Nutzung der Transportdienstleistung für den Fahrgast dar. Um Betrug zu verhindern, sollte das Ticket fälschungssicher sein, d.h. es sollte nicht, oder nur mit sehr großem Aufwand möglich sein, ein solches Ticket zu erstellen, zu modifizieren, zu kopieren oder auf jegliche andere Weise zu verfälschen.

Mit der Erzeugung, Modifizierung und Duplizierung von Fahrscheinen könnte die Dienstleistung von „Schwarzfahrern“ genutzt werden. Dies hat pro Einzelfall unmittelbar negative finanzielle Auswirkungen für den Betreiber. Bei einer Häufung der Betrugsfälle kann dies größere ökonomische Folgen nach sich ziehen, die aufgrund ihres bedrohlichen Niveaus die Existenz des Unternehmens gefährden könnten.

Auch könnte die Integrität durch technische Mängel unabsichtlich verloren gehen, wodurch die digitalen Tickets ihre Gültigkeit verlieren würden. Folglich sind auch keine Kontrollen mehr möglich. Gerät die Kenntnis über das „Nichtfunktionieren“ der Tickets an die Öffentlichkeit, so ist fraglich, ob Fahrgäste überhaupt noch Tickets erwerben oder direkt schwarzfahren.

Durch dieses große Schadenspotential ist der Schutzbedarf mit **hoch** einzustufen und sollte auf jeden Fall vermieden werden. Insbesondere wenn ausschließlich digitale Tickets in Umlauf sind, kann nicht mehr „mal eben“ auf die herkömmliche Technik mit Papiertickets am Fahrkartenautomaten zurückgewechselt werden, da hierzu die nötige Infrastruktur nicht mehr existieren könnte.

Kundeninformationen

Die Integrität der Kundeninformationen kann durch Manipulation zerstört werden.

Den ersten Teil der Kundeninformationen, die Referenz auf das Kontrollmedium, die im Anwendungsfall „Kundeneinstellungen“ (siehe Abschnitt 3.2.6) eingegeben wird, kann ohne weiteres manipuliert werden, und stellt keine Bedrohung für das System dar. Neu erworbene Tickets würden dann mit einer anderen Person verknüpft werden.

Die in den *privaten Tickets* enthaltene persönliche Referenz ist nach Ausstellung des Tickets nicht mehr veränderbar. Durch einen Manipulationsversuch der zuvor festgelegten Personendaten würde die Gültigkeit des Tickets verloren gehen.

Es sei denn, es gibt eine Zahl Z, die den gleichen Hashwert wie der in dem Ticket enthaltene Wert *Hash(Z)* liefert, und gleichzeitig eine andere KMR, die zusammen mit der XOR Verknüpfung zum gleichen KW führt. In diesem Fall kann die Gültigkeit unter Veränderung der *KMR* beibehalten werden. Aus Komplexitätsgründen und der mangelnden Fähigkeit zur künstlichen Erzeugung von Kollisionen in der Hashfunktion ist eine sinnvolle Manipulation mit einem zu großen Aufwand verbunden.

Im Normalfall führt eine Integritätsverletzung der Kundeninformationen jedoch dazu, dass das Ticket die Bindung zu dem Kontrollmedium verliert, mit dem es nur kombiniert gültig ist und verliert damit selbst seine Gültigkeit.

Die Integrität der Kundeninformationen stellt somit ein besonders schützenswertes Gut dar. Sie ist daher in die Schutzbedarfskategorie **hoch** einzuordnen.

Anwendungsdaten

Durch eine Manipulation des Anwendungs-Packages und der im Anwendungscode enthaltenen Operationen kann die Funktionsfähigkeit der Anwendung negativ beeinträchtigt werden und bis hin zur vollständigen Zerstörung dieser führen. Das System könnte in unvorhersehbare Zustände gelangen und die Funktionssicherheit der Anwendung kann nicht mehr gewährleistet werden. Als Folge dessen kann für keine weiteren Sicherheitsziele mehr garantiert werden, da von der funktionssicheren Ausführung (engl. *safety*) der Anwendung alle weiteren Sicherheitsziele abhängen. Die Einhaltung dieser kann nicht mehr garantiert werden, sodass das Gesamtsystem als unsicher einzustufen ist.

Kennt ein Angreifer die Bedeutung der im Anwendungscode enthaltenen Operationen, was meistens gleichbedeutend mit dem Verlust der Vertraulichkeit der Informationen ist, und kann er diese gezielt einsetzen, so kann eine absichtliche Manipulation zur Gefährdung der anderen Schutzziele führen. Beispielsweise könnte ein Angreifer vertrauliche Informationen, wie sie im Rahmen dieser Anwendung verwendet werden, durch entsprechend eingefügte Anweisungen an sich selbst schicken lassen, oder durch direkten Zugriff auf die Datenobjekte die Integrität dieser zerstören.

Der Schutzbedarf für die Anwendungsdaten bezüglich der Integrität ist daher tendenziell als hoch bis sehr hoch einzustufen. Unter Berücksichtigung der anderen schützenswerten Güter, die in diesem Szenario keinen Schutzbedarf der Kategorie sehr hoch bedürfen, ist der Schutzbedarf für die Integrität der Anwendungsdaten **hoch**.

4.1.2.4 Zusammenfassung

Schutzbedarf	Verfügbarkeit	Vertraulichkeit	Integrität
Fahrplaninformationen	Normal	Normal[1]	Hoch
Zahlungsinformationen	Normal	Hoch	Hoch
Ticketinformationen	Hoch	Normal	Hoch
Kundeninformationen	Hoch	Hoch	Hoch
Anwendungsdaten	Hoch	Hoch	Hoch

Tabelle 4.4: Schutzbedarf der schützenswerten Informationen

Alle identifizierten Informationen besitzen einen Schutzbedarf von „Normal“ oder „Hoch“.

Ein Schutzbedarf von „Sehr hoch“ konnte für keine der analysierten Informationen festgestellt werden, und tritt nur bei Schäden ein, die ein katastrophales Ausmaß einnehmen können, bei der Leib und Leben von Personen gefährdet werden können.

Durch eingeschleusten Schadcode in Anwendungsdaten könnte beispielsweise erreicht werden, dass die Funktionalität des Gesamtsystems eingeschränkt und über die Telefon-Anwendung des mobilen Gerätes keine Notrufe mehr getätigt werden können. Die Verantwortlichkeit für die Sicherheit und die Funktionalität ist in diesem Fall primär bei der Telefon-Anwendung selbst, sowie der zugrundeliegenden Plattform *Android*, zu suchen. Zudem liegt die Absicherung fremder Komponenten außerhalb des eigenen Handlungsspielraums. Bei der Entwicklung sollte jedoch beachtet werden, dass der eigene Code durch sichere Programmierung selbst keine Bedrohung für die Sicherheit anderer Komponenten darstellt.

4.1.2.5 Verbindlichkeit

Im eCommerce muss die Verbindlichkeit der Geschäftsprozesse gewährleistet werden können. Der in diesem Szenario stattfindende Kaufvertrag sollte rechtsverbindlich sein, damit die aus dem Vertrag entstehenden Rechte und Pflichten für beide Geschäftspartner, hier Fahrgast und Unternehmen, bindend sind. Im Falle eines Vertragsbruchs sollte der Geschäftspartner für seine daraus entstandenen Verantwortlichkeiten belangbar sein.

In diesem Szenario ist davon der Ticketerwerb betroffen, bei dem Leistung und Gegenleistung wie folgt zueinander stehen: Der Fahrgast erhält als Vertragsgegenstand die Berechtigung zur Verkehrsmittelnutzung, für welche er als Gegenleistung ein Entgelt an den Dienstleister zu entrichten hat. Der Betreiber sorgt für die Bereitstellung der Verkehrsmittel und erhält im Gegenzug eine finanzielle Vergütung. Die Abrechnung erfolgt durch den Verkauf von Fahrscheinen, welche als Nutzungsberechtigung für die Inanspruchnahme der Dienstleistung eingesetzt werden.

Weiter ist auch der Zahlungsdienstleister im Auftrag des Kunden an der Einhaltung des Vertrages beteiligt, da er die Abwicklung des Zahlungsverkehres übernimmt, wodurch die vom Betreiber an den Fahrgast gerichtete Zahlungsforderung letztlich ausgeglichen wird.

[1] Bzw. durch die Hinzunahme zusätzlicher Informationen auch Hoch

Im Mobile Commerce befinden sich die Vertragspartner zum Vertragsabschluss an verschiedenen Orten und kommunizieren drahtlos auf elektronischem Wege miteinander. Ein direkter Kontakt zwischen den Teilnehmern findet nicht statt. Daraus ergeben sich folgende Anforderungen, die nicht trivial sind:

- Authentizität: Mit wem wurde kommuniziert?
- Verbindlichkeit: Was wurde vereinbart?

Durch den Einsatz von digitalen Signaturen kann die Verbindlichkeit der Vertragsinhalte von den verschiedenen Parteien gewährleistet werden. Dies erfolgt analog zum Einsatz handschriftlicher Unterschriften und erfüllt damit den gleichen Zweck.

Das digitale Ticket enthält als Sicherheitsmerkmal eine digitale Signatur und erfüllt damit die Verbindlichkeit: Das Unternehmen berechtigt den Fahrgast zur Nutzung der Verkehrsmittel.

Voraussetzung dafür ist jedoch die Authentizität beider Vertragspartner, damit der Handel in einem rechtssicheren Raum stattfinden kann. Vergleichbar mit der handschriftlichen Unterschrift bedeutet dies: Wenn Person A etwas unterschreibt, was Person B unterschreiben sollte, so wurde das Dokument zwar unterschrieben, stellt jedoch keine Willenserklärung von B dar, da die Unterschrift von A stammt. Daher gilt es zu prüfen, von wem das Ticket „unterschieben“ wurde. Damit beschäftigt sich die Authentizität.

4.1.2.6 Authentizität

Authentizität steht eng in Zusammenhang mit den anderen Schutzzielen und ist Voraussetzung für die Vertraulichkeit, die Integrität und die Verbindlichkeit. Nur durch die Echtheit der Identitäten kann dies gewährleistet werden. Ansonsten wäre das System durch Maskierung sehr leicht angreifbar, d.h. durch Fälschen der Identität.

In Anbetracht an die Sicherheitskette[1] und bezogen auf das Beispielszenario bedeutet dies, dass die Authentizität der beteiligten Subjekte gewährleistet sein muss, zu denen der Fahrgast, das ÖPV-Unternehmen, sowie auch der Zahlungsdienstleister gehören, damit eine Umsetzung der weiteren Schutzziele überhaupt sinnvoll ist.

Insbesondere in Zusammenhang mit der Verbindlichkeit ist - wie gerade festgestellt - die Authentizität der beteiligten Subjekte von essentieller Bedeutung für die Rechtsverbindlichkeit von Verträgen.

4.1.3 Bedrohungsanalyse

Die Bedrohungsanalyse beginnt mit der Aufzählung aller möglichen Bedrohungen. Die Schwierigkeit besteht darin, wirklich alle möglichen Bedrohungsszenarien zu beachten. Nach Stefan Lucks [Lucks02] sollte ein „Security Engineer“ dabei noch erfinderischer sein, als alle seine Gegner zusammen. Abbildung 4.3 liefert dazu einen Überblick.

[1] Sicherheitskette: siehe Seite 11

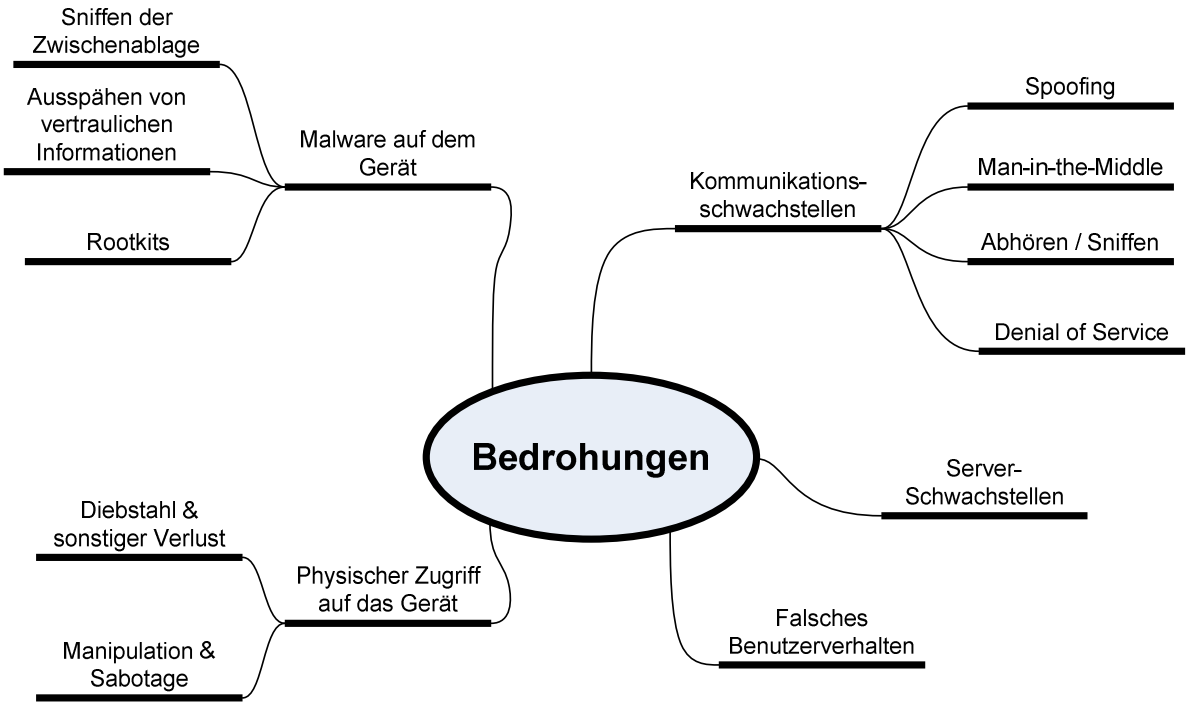

Abbildung 4.3: Übersicht über verschiedene Bedrohungen

Es wird zwischen den Bedrohungen des IT-Systems angelehnt an die im Rahmen der Strukturanalyse identifizierten Komponenten unterschieden:

- Malware / Schadsoftware
- Physischer Zugriff
- Kommunikationskanal
- Benutzer

4.1.3.1 Direkter Zugriff auf das Gerät

Eine ernst zu nehmende Bedrohung stellt der direkte Zugriff auf das Endgerät durch unautorisierte Personen dar. Ein kurzer unbeobachteter Moment reicht in der Regel für einen Zugriff aus. Im Falle eines Diebstahls könnte der Angreifer das Gerät nun dauerhaft entwenden, um in Besitz des Gerätes selbst zuzüglich der darauf enthaltenen Daten zu kommen. Auch könnte er es „nur“ manipulieren und es somit für weitere Angriffe vorbereiten.

Neben dem Verlust der Hardware sind in beiden Szenarien sämtliche momentan auf dem Endgerät gespeicherten Daten bedroht. Bei einer Manipulation werden zusätzlich auch noch die zukünftigen Daten auf dem System bedroht.

Bei einem Diebsstahlszenario ist unautorisierter Zugriff auf alle schützenswerten Daten möglich und die Vertraulichkeit dieser kann nicht mehr gewährleistet werden. Im Falle einer Manipulation (oder Sabotage) ist zusätzlich auch die Integrität und die Verfügbarkeit der Daten bedroht, da die Daten verändert oder gelöscht werden könnten.

Eine Offenlegung der schützenswerten Informationen kann dabei weitaus bedrohlichere Szenarien für das Opfer zur Folge haben, als nur den materiellen Schaden durch den Verlust des Gerätes selbst. Insbesondere der Missbrauch der Zahlungsinformationen ist am wahrscheinlichsten, da ein Täter sich damit finanziell bereichern kann. Er besitzt dabei ein Motiv für sein kriminelles Verhalten.

4.1.3.2 Malware

Zunächst muss die Malware auf das System gelangen und dort zur Ausführung gebracht werden. Ist dies erfolgt, so kann der darin enthaltene Schadcode die Schutzziele der Anwendung gefährden.

Es gibt mehrere Möglichkeiten, wie eine Schadsoftware auf das System gelangen kann:

- Installation durch Anwender
- Installation durch unberechtigte Dritte
- Schwachstellen bzw. „Sicherheitslücken“

Installation durch Anwender

Der Anwender kann beliebige Anwendungen für *Android* auf dem Gerät installieren, zu der auch Schadsoftware gehört. Dies kann er aus Unachtsamkeit, aus mangelndem Sicherheitsbewusstsein oder schlicht aus Unwissenheit über mögliche Gefahren veranlasst haben, sodass Schadsoftware auf das Gerät gelangt.

Auch könnte er durch einen Spoofing-Angriff getäuscht worden sein, bei der der Anwender fälschlicherweise eine Schadsoftware anstelle der gewünschten Anwendung installiert. Er vertraut dabei einer vorgetäuschten Identität.

Jesse Burns [Burns08] sagt: *“Malicious software is an unfortunate reality on popular platforms, and through its features Android tries to minimize the impact of malware. However, even unprivileged malware that gets installed on an Android device (perhaps by pretending to be a useful application) can still temporarily wreck the user’s experience. Android [] minimizes the extent of abuse possible, by requiring user permission for programs that do dangerous things like: directly dialing calls (which may incur tolls), disclosing the user’s private data, or destroying address books, email, etc. ”*

Dies bietet insbesondere bei technisch nicht versierten Anwendern mit wenig Sicherheitsbewusstsein viel Angriffsfläche und wird später im Abschnitt Benutzerverhalten (4.1.3.4) behandelt.

Zugriff durch unberechtigte Dritte

Weiter könnte die Installation von Schadsoftware auch absichtlich durch einen direkten physischen Zugriff unberechtigter Dritter auf das Gerät durchgeführt werden, wie zuvor beschrieben (vgl. Abschnitt 4.1.3.1).

Weiter könnte die Installation von Schadsoftware auch absichtlich durch einen direkten physischen Zugriff durch unberechtigte Dritte auf das Gerät durchgeführt werden, wie zuvor in Abschnitt beschrieben.

Sicherheitslücken

Mit Sicherheitslücken in diesem Zusammenhang werden Schwachstellen aufgrund fehlerhafter Implementierungen verstanden, die durch nachlässige Programmierung entstehen. Befindet sich derartige Software auf dem mobilen Gerät, so ist es möglich, dass diese Fehler von Angreifern ausgenutzt werden können, um Schadcode auf das mobile Gerät einzuschleusen und zur Ausführung zu bringen.

Malware-Arten

Der mobile Fahrplanauskunfts- und Ticketingdienst läuft nicht alleine auf dem System. Andere Anwendungen teilen sich die gleiche Umgebung mit ihm und werden genauso durch die Plattform verwaltet. Dabei handelt es sich nicht immer um friedfertige Software, sondern sie kann auch ungewünschten Schadcode enthalten, durch den die entworfene Client-Anwendung angegriffen werden könnte.

Der Autor unterscheidet hier zwischen drei verschiedenen Arten von Schadsoftware (engl. *Malware*):

- „Normale“ Malware, d.h. Android-Anwendungen ohne Root-Rechte wahlweise mit oder ohne eingeräumten *Permissions*
- „Bösartige“ Malware, d.h. Programme, die mit Root-Rechten ausgeführt werden und sich damit den gewöhnlichen Zugriffskontrollverfahren von *Android* entziehen können.
- Plagiate der Client-Anwendung (z.B. zu Phishing-Zwecken)

„Normale“ Malware

Unter „Normaler“ Malware versteht der Autor Malware, die sich wie eine normale *Android*-Anwendung verhält und mit deren Rechten ausgeführt wird.

Die Auswirkungen derartiger Schadanwendungen werden durch das Isolationskonzept von *Android* zunächst komplett verhindert. Somit wird auch die Beispielanwendung vor derartiger Malware geschützt. Im Rahmen der von *Android* durchgeführten Paket-Installation erteilt der Anwender hierbei der Schadsoftware *Permissions*[1], wodurch die Isolation teilweise wieder aufgehoben wird, sodass ein Zugriff oder Eindringen in fremde Anwendungen durch diese ermöglicht wird.

Je nachdem, wie viele Rechte der Malware zum Installationszeitpunkt eingeräumt wurden, desto größer können die Auswirkungen der Schadsoftware werden. Wenn der Benutzer das Sicherheitskonzept verstanden hat, kann das Entstehen von Schäden auf diese Weise effektiv verhindert werden.

Daher gilt es, das Benutzerverhalten näher zu analysieren, um festzustellen, ob sich Benutzer bei der Anwendungsinstallation richtig verhalten. Dies geschieht später in Abschnitt 4.1.3.4.

„Bösartige“ Malware

Diese Art von Schadsoftware könnte erweiterte Systemrechte erlangen, durch die sämtliche Schutzziele der Anwendung bedroht werden. In *Android* erfolgt dies über den Benutzer „root“, wie er von Linux Systemen bekannt ist. Eine Erlangung dieser weitreichenden Systemrechte soll in *Android* prinzipiell nicht möglich sein und ist Teil dessen Sicherheitskonzepts.

Durch das Ausnutzten entsprechender Sicherheitslücken kann von dieser Möglichkeit dennoch Gebrauch gemacht werden. Das Betriebssystem enthält viele Millionen Zeilen an Quellcode und durch die ständige Weiterentwicklung und das Aufkommen neuer Derivate durch Endgerätehersteller, die *Android* an ihre Hardware anpassen müssen, kann keinesfalls die Existenz von Sicherheitslücken ausgeschlossen werden. Die mögliche Angriffsfläche ist somit groß.

Schadcode, der als Root-Benutzer ausgeführt wird, umgeht damit die Zugriffskontrollverfahren, wie sie für normale *Android*-Anwendungen gelten. Diese finden erst auf einer höheren Ebene der Plattform statt. „Bösartige“ Malware arbeitet jedoch auf niedrigeren Ebenen im System.

[1] *Permissions*: Siehe Abschnitt 2.5.4

Plagiate

Bei der Schadsoftware kann es sich auch um eine gefälschte Kopie der Client-Anwendung des Beispielszenarios handeln, die zu Phishing-Zwecken dienen soll. Die zuvor aufgestellten Schutzziele können damit direkt gefährdet werden.

Exemplarisch könnte ein „Fahrgast“ seine Zahlungsinformationen in eine derartige Phishinganwendung eingeben, welche diese direkt an einen Angreifer übermittelt. Auch könnte das Plagiat als Trojaner fungieren und neben der gewünschten Funktionalität auch zusätzlichen Schadcode ausführen, der zu weiteren Schäden führen kann.

Daher ist die Echtheit der Anwendung vor Verwendung sicherzustellen. Eine sichere Installation der Client-Software ist Voraussetzung für einen weiteren sicheren Betrieb des Gesamtsystems.

Beispiele

Im Folgenden werden zwei Beispiele für Malware vorgestellt:

Monopolisieren der CPU

Durch Monopolisieren der CPU mit einem hoch priorisierten Prozess könnten die verfügbaren Ressourcen des mobilen Gerätes reserviert werden, sodass diese nicht mehr in vollem Umfang für die anderen Anwendungen zu Verfügung stehen. Als Folge wird diese langsamer als gewohnt oder gar nicht mehr ausgeführt werden können. Der daraus entstehende erhöhte Energieverbrauch schränkt die Verfügbarkeit des gesamten Systems weiter ein.

Zwischenablagen-Sniffer

Die in *Android* verwendeten *Permissions* dienen dem Zweck, die Isolation gezielt zu lockern, damit die gewünschte Interprozesskommunikation stattfinden kann. Über Prozessgrenzen hinweg könnte es jedoch trotzdem zu gefährlichen Informationsflüssen kommen, da die Informationen nicht zwangsläufig an die enthaltenen Datenobjekte gebunden sind.

Dies geschieht beispielsweise über das Kopieren von Informationen in die Zwischenablage, oder über die Anzeige der Informationen über das Display.

Das Auslesen bzw. Sniffen des Inhalts der Zwischenablage in *Android* durch andere Programme ist beispielsweise mit dem folgenden Programmcode möglich:

```
//import Android.text.ClipboardManager;
ClipboardManager cm = (ClipboardManager)
      getSystemService(CLIPBOARD_SERVICE);
String text = clipboardManager.getText().toString();
System.out.println(text);
```

Informationen, die der Benutzer während der Programmeingabe in die Zwischenablage kopiert, sind somit nicht geschützt. Ein Benutzer, der sich dieser Gefahr nicht bewusst ist, sollte am Kopieren von besonders schützenswerten Informationen in die Zwischenablage gehindert werden.

Dazu sind im Übrigen keine besonderen *Permission*s erforderlich. Somit könnte jede andere Anwendung die Zwischenablage auslesen, ohne bei der Installation der Anwendung vom Benutzer die Freigabe dazu erhalten zu haben.

4.1.3.3 Bedrohungen der Kommunikation

In der Beispielanwendung werden aufgrund des mobilen Einsatzbereiches drahtlose Verbindungen genutzt. Für Angreifer bietet dies zunächst einmal gute Voraussetzungen, den genutzten Kommunikationskanal anzugreifen. Ein Zugriff über die Luftschnittstelle ist leicht möglich, da kein direkter physischer Zugriff mit einem Kabel benötigt wird.

Abhören

Das Abhören und Mithören der Kommunikation ist aufgrund der Drahtlos-Eigenschaft der verwendeten Netzwerke durch andere Teilnehmer ohne weiteres möglich. Im Gegensatz zu kabelgebundenen Netzwerken gibt es keine physikalischen Hürden, die den Zugriff beschränken könnten, sodass sämtliche Daten mitgelesen werden können. Geschieht dies im Klartext, so geht damit der Schutz der Vertraulichkeit verloren. Im Gegensatz zu den folgenden Angriffen ist dieser passiv, kann lediglich die Vertraulichkeit der übertragenen Daten stören.

Maskierung

Es sind einige Spoofing- bzw. Man-In-The-Middle Szenarien vorstellbar, bei denen eine böswillige dritte Partei ihre Identität maskiert. Dieser Angreifer gibt sich dabei den beiden Kommunikationspartnern (Fahrgast und ÖPV-Unternehmen) als der jeweils andere aus, kann dabei die gesamte Nachrichtenübertragung im Klartext mitlesen und beliebige Veränderungen vornehmen. Durch Einflussnahme auf den Zahlungsverkehr könnten finanzielle Schäden verursachen werden.

Störung der Kommunikation

Laut Jörg Roth [Roth05] ist die Funkkommunikation wesentlich störanfälliger als die drahtgebundene Kommunikation. Störsender könnten absichtlich die Verfügbarkeit von drahtlosen Netzwerken einschränken. Solche Angriffe auf die Verfügbarkeit sind jedoch eher unwahrscheinlich, da Angreifer in der Regel kein Motiv dafür haben.

Zur Teilnahme am Netzwerk wird lediglich die Verwendung des gleichen Frequenzbandes an den Empfangs- oder Sendeeinheiten benötigt, sodass beispielsweise auf Transportebene synthetisch erzeugte Nachrichtenpakete Stausituationen verursachen können.

4.1.3.4 Benutzerverhalten

Das Benutzerverhalten erfolgt nicht zwangsläufig durch logische Schlüsse und ist durch technische Maßnahmen nicht kontrollierbar, jedoch ist es durch ein entsprechendes Anwendungsdesign beeinflussbar.

Wie zuvor beschrieben (vgl. Abschnitt 4.1.3.2), kann ein falsches Benutzerverhalten dazu führen, dass Malware auf dem System installiert wird. Bei der Installation der Software bestätigt der Anwender dabei die *Permissions*. Zu diesem Zeitpunkt ahnt er noch nichts von dem bevorstehenden Schaden.

Damit übergibt *Android* die Kontrolle über die Sicherheit an den Benutzer.

Dabei stellt der Benutzer die einzige „Komponente“ dar, die die Installation von Schadsoftware effektiv verhindern könnte. Virenscanner oder andere Software zur Erkennung von Malware, oder sonstige Sicherheitsmaßnahmen, die nur die Installation und Ausführung von geprüftem oder signiertem Code zulassen, gibt es in *Android* bisher nicht.

4.1.3.5 Anforderungen des eCommerce

eCommerce-Anwendungen benötigten ein Sicherheitsniveau, mit dem sie ökonomisch sinnvoll betrieben werden können. Wie in Kapitel 2.4.4 beschrieben, kann die absolute Sicherheit des Systems nicht das Ziel sein, da auch die Kosten für die Umsetzung der Sicherheitsmaßnahmen beachtet werden müssen. Daher reicht oftmals die Inkaufnahme eines als handhabbar eingeschätzten Risikos aus.

Zudem muss durch ein entsprechendes Systemdesign Sorge dafür getragen werden, dass die subjektive Sicherheit der Anwendung aus Kundensicht hoch genug ist, damit Kunden nicht aufgrund von Sicherheitsbedenken von der Verwendung der Anwendung absehen werden. Es muss daher ausreichend Vertrauen erzeugt werden. Eine unmittelbar vor der Zahlung stattfindende Authentifizierung des Benutzers am System könnte beispielsweise den gewünschten Effekt erzielen

4.1.4 Zusammenfassung der Sicherheitsanforderungen

In diesem Abschnitt wurden zunächst die Informationen als die zu schützenden Werte der Anwendung identifiziert und für diese ein Schutzbedarf anhand von Kategorien aufgestellt, die sich an den möglichen Schäden orientieren. Weiter fand eine Bedrohungsanalyse statt, in der konkrete Bedrohungen des Systems präsentiert wurden.

Daraus ergibt sich als Anforderung an das System, dass die Sicherheit der in der Anwendung verarbeiteten Informationen hinsichtlich verschiedener Schutzziele, sowie auch vor konkreten Bedrohungen, gewährleistet werden sollte. Tabelle 4.5 verdeutlicht dies:

Informationen	Schutzziele	Bedrohungen
Fahrplaninformationen Zahlungsinformationen Ticketinformationen Kundeninformationen Anwendungsdaten	Vertraulichkeit Integrität Verfügbarkeit Verbindlichkeit Authentizität	Direkter physischer Zugriff Malware auf dem System Falsches Benutzerverhalten Kommunikationsschwachstellen

Tabelle 4.5: Schutzbedarf

4.2 Entwicklung und Realisierung

Dieses Kapitel stellt die getroffenen Sicherheitsmaßnahmen und -mechanismen vor, durch die die im vorherigen Kapitel aufgestellten Sicherheitsanforderungen für die Beispielanwendung eingehalten werden sollen.

Anhand der Anforderungen wird zunächst eine Strategie erarbeitet, mit der das gewünschte Niveau an Sicherheit erreicht werden kann. Diese soll unter Verwendung geeigneter technischer Maßnahmen umgesetzt werden. Dies können zum einen eigene Implementierungen auf Anwendungsebene oder zum anderen Mechanismen der Plattform *Android* sein. Die Feststellung der Machbarkeit steht dabei im Vordergrund.

Die zuvor analysierten Bedrohungen werden dabei nur am Rande beachtet und können später bei der Bewertung im nächsten Kapitel 4.3 zum Testen auf potenziell noch bestehende Sicherheitsschwachstellen eingesetzt werden.

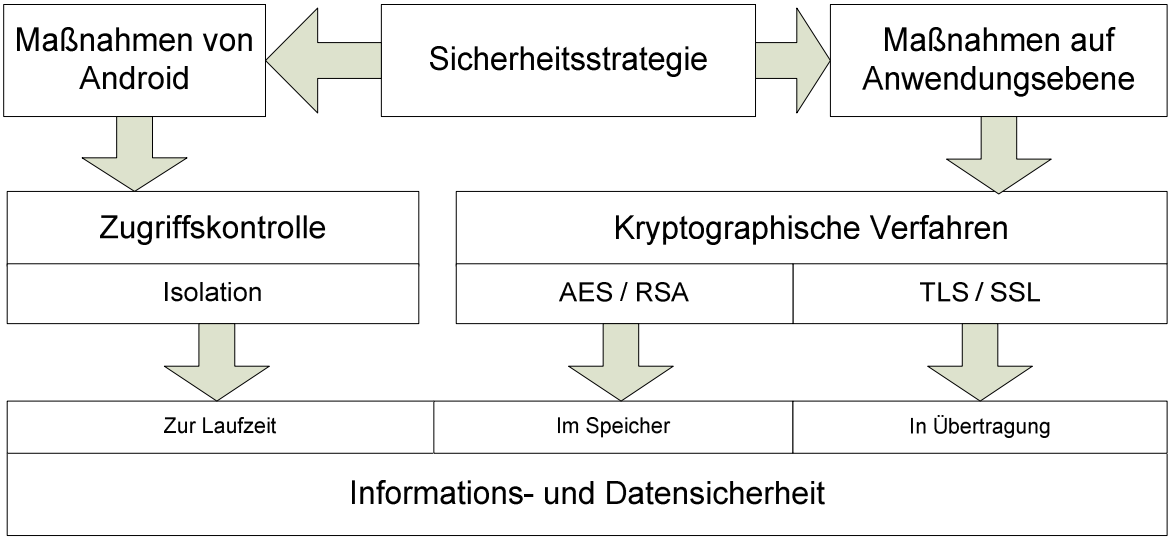

Abbildung 4.4: Entwicklung und Realisierung

4.2.1 Sicherheitsstrategie

Im Folgenden wird eine Sicherheitsstrategie aufgestellt, mit der für die identifizierten Informationen die Sicherheitsziele erreicht werden können. Sie umfasst die Festlegung von Zugriffsrechten, welche im Rahmen von Zugriffskontrollverfahren, bestehend aus Authentifizierung der Subjekte und Autorisierung des Zugriffes auf die Subjekte, verwendet werden.

4.2.1.1 Zugriffskontrolle

Durch Zugriffskontrollverfahren kann der Zugriff von Subjekten auf die als schützenswert identifizierten Objekte kontrolliert werden, sodass die Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit eingehalten werden können. Es besteht aus den Entitäten:

- Subjekte, Benutzer, Rollen
- Objekte, Daten
- Zugriffe, Operationen
- Rechte

Vorausgesetzt wird hierbei die Authentizität der Subjekte, sowie die Verbindlichkeit derer Daten.

Subjekten

Die Echtheit der Subjekte kann durch Authentifikation sichergestellt werden und zur Gewährleistung der Verbindlichkeit der von diesen Subjekten stammenden Daten können digitale Signaturen eingesetzt werden.

Subjekte & Rollen

In der Beispielanwendung sind die beteiligten Subjekte primär der Fahrgast, der Dienstleister, sowie der MPSP. Der Benutzer verwendet dabei zur Nutzung der angebotenen Dienste die Client-Anwendung, welche folglich in dessen Namen handelt, Aktionen durchführen kann und daher auch als ein zugreifendes Subjekt einzuordnen ist.

Weiter könnte ebenfalls der verwendete Kommunikationskanal, zu dem alle daran teilnehmenden Komponenten gehören, als ein Subjekt angesehen werden. Dieses ist nur zum Weitertransport der Daten berechtigt ist, jedoch nicht zum abhören, mitlesen, aufzeichnen, speichern, verändern, umleiten oder löschen dieser.

Die Subjekte können dabei verschiedene Rollen einnehmen. Der Nutzer der Anwendung kann als Besitzer des mobilen Gerätes, als Interessent am Fahrplan, als Kunde beim Kauf eines Tickets, als Kontoinhaber bei der Zahlung, sowie als Fahrgast bei einer Kontrolle auftreten. Auch das ÖPV-Unternehmen kann verschiedene Rollen einnehmen, deren Unterscheidung jedoch im Rahmen der Umsetzung irrelevant ist: Auskunftsgeber, Verkäufer, Dienstleister, Kontrolleur, Entwickler.

Authentifikation und Verbindlichkeit

Werden für die Aktionen Zugriffsrechte benötigt, müssen sich die beteiligten Subjekte zunächst authentifizieren. Dabei wird überprüft, ob es sich tatsächlich um die behauptete Identität handelt. Die Authentifizierung der Subjekte ist dabei Pflicht. Ohne sie wären die darauf aufsetzenden Zugriffskontrollverfahren hinfällig, da sich unberechtigte Subjekte sonst einfach mit einer der zugriffsberechtigten Identitäten ausgeben könnten (Maskierungsangriff).

Digitale Signaturen & Zertifikate & Trust

Mittels digitaler Signaturen, Zertifikaten, und einer Trust Infrastruktur kann die Authentizität auch über Rechnergrenzen hinweg erfolgen, wodurch auch die Verbindlichkeit gewährleistet werden kann.

Für die im Rahmen der Beispielanwendungen verwendeten Daten können digitale Signaturen eingesetzt werden, damit die Authentizität von Daten bezüglich ihrer Herkunft garantiert werden kann. Die Authentizität stellt damit die richtige Zuordnung zwischen der digitalen Signatur (der Handschrift) und der signierenden Entität (dem Unterzeichner) dar.

Die im Rahmen der asynchronen kryptographischen Verfahren verwendeten privaten Schlüssel sind im Besitz von genau einer Instanz und werden zur digitalen Signierung der Daten verwendet. Die Echtheit der digitalen Signatur lässt sich über die zugehörigen öffentlichen Schlüssel prüfen. Voraussetzung dafür ist jedoch, dass bei dieser Prüfung der echte öffentliche Schlüssel verwendet wird.

Dies geschieht in der Beispielanwendung mittels Zertifikaten, die von einer vertrauenswürdigen dritten Partei, der *Certification Authoriy* (CA), herausgegeben wurden. Diese beglaubigt mit ihrer eigenen digitalen Signatur die Echtheit des Zertifikats und damit mit auch die Identität des Herausgebers. Die CA ist dabei Teil einer Public-Key-Infrastruktur (PKI), die „Hierrarchial Trust“ als Vertrauensmodell verwendet, sodass über diese die CA als glaubwürdig und authentisch angesehen werden kann.[1]

Somit können sich die verschiedenen Parteien - Fahrgast, ÖPV-Unternehmen und auch der MPSP - unter Verwendung des Netzwerkes die Echtheit der Identitäten belegen lassen. Dadurch kann die Authentizität der Subjekte garantiert werden, auch wenn diese nie zuvor miteinander kommuniziert haben und Authentifikationsmerkmale austauschen konnten.

Erst nach der Authentifikation der Subjekte kann die Autorisierung, die Überprüfung der Rechte beim Zugriff auf ein Objekt, stattfinden. Dabei wird überprüft, ob das Subjekt die nötigen Zugriffsrechte für die gewünschte Operation auf dem Objekt besitzt.

Datenobjekte

Die bereits in Kapitel 3.3.5 eingeführten Datenobjekte repräsentieren die als schützenswert identifizierten Informationen:

- Fahrplanauskunftsdaten
  - *Fahrplananfrage*
  - *Fahrplan*
  - *Teilstrecke*
- Zahlungsdaten
  - *Zahlungsinfo*
- Ticketdaten
  - *Ticket*
- Kundendaten
  - *Privates Ticket*
  - *Kontrollmedium*
- Anwendungsdaten
  - Das „*Android* Applikation Package“
  - Dex-Files

Eine Überführung der Informationen in deren Datenobjekte oder umgekehrt ist aufgrund der Ähnlichkeit der Bezeichnungen trivial.

Damit die in den Datenobjekten enthaltenen Informationen durch das verwendete Zugriffskontrollverfahren geschützt werden können, muss sichergestellt sein, dass die Informationen an die Datenobjekte fest gebunden sind, und diese nicht verlassen können. Dies schließt einen direkten Speicherzugriff explizit aus.

[1] Die Kenntnis über ein Wurzelzertifikat am Anfang der Vertrauenskette ist zur Nutzung der PKI erforderlich.

Rechtefestlegung

Dazu ist zunächst die Festlegung von Rechten notwendig, in denen präzisiert ist, welche Subjekte welche Zugriffsrechte besitzen. Dies berechtigt sie zu einer bestimmten Operation auf einem Objekt. Abbildung 4.5 zeigt die zur Funktionserfüllung nötigten Rechte in der Beispielanwendung.

Objekt	**Zugriffsberechtigte Subjekte**	
Fahrplandaten	Person: Interessent ÖPV-Unternehmen: Auskunft	
Ticketdaten	Bei Ticketkauf: Person: Kunde ÖPV-Unternehmen: Verkäufer	<u>Bei Kontrolle:</u> Person: Fahrgast ÖPV-Unternehmen: Kontrolleur
Zahlungsdaten	Person: Kunde MPSP	
Kundendaten	Person: Kunde / Fahrgast ÖPV-Unternehmen: Kontrolleur	
Anwendungsdaten	Person: Interessent / Kunde / Fahrgast ÖPV-Unternehmen: Entwickler	

Abbildung 4.5: Zugriffsrechte auf die Informationen

Nicht aufgeführte Subjekte haben keine Zugriffsrechte, sodass ihnen der Zugriff verwehrt bleiben muss.

4.2.2 Technische Maßnahmen auf Anwendungsebene

Es stellt sich die Frage, wie und inwieweit die im Rahmen der Sicherheitsstrategie festgelegten Zugriffsrechte nun unter Verwendung von *Android* umgesetzt werden können, um die Sicherheitsanforderungen zu erfüllen.

Einige Sicherheitstechniken dazu bringt *Android* als Plattform direkt mit, und andere müssen anwendungsintern umgesetzt werden. Dazu stellt *Android* die Nutzung der Java-Klassenbibliotheken zur Verfügung.

Sämtliche Daten, mit Ausnahme der Anwendungsdaten, können während der Speicherung auf dem Gerät, sowie während der Übertragung über das Netzwerk, durch kryptographische Verfahren geschützt werden, die sich auf Anwendungsebene umsetzten lassen:

- Absicherung des Kommunikationskanals mit SSL (Integrität, Vertraulichkeit, Authentizität)
- Verschlüsselte Datenspeicherung auf dem Gerät (Vertraulichkeit, Authentizität)
- Digitale Signaturen der Daten (Integrität, Authentizität)
- Replikation und Backups (Verfügbarkeit)

Zur Realisierung der technischen Maßnahmen können Klassen aus den Java-Bibliotheken genutzt werden.

Der Schutz der Anwendungsdaten, sowie der Schutz aller Daten zur Laufzeit hingegen kann nicht auf Anwendungsebene umgesetzt werden, sondern nur durch die Plattform. Die Anwendung ist dabei zwingend an die Vorgaben von *Android* gebunden, sodass in diese Richtung keine eigenen technischen Maßnahmen verwendet werden können.

4.2.2.1 Absicherung des Kommunikationskanals

Auf Anwendungsebene findet eine verschlüsselte Datenübertragung statt, mit der die Vertraulichkeit und die Integrität sämtlicher übertragener Daten, mit Ausnahme der Anwendungsdaten auf der Kommunikationskanal garantiert werden kann.

In dem Prototyp wird zur dazu das Protokoll *Secure Sockets Layer* (SSL) verwendet. Aufsetzend auf dem TCP-Protokoll bietet es eine zuverlässige Ende-zu-Ende-Sicherheit auf Anwendungsebene. Wie bereits in Abschnitt 2.3.6 vorgestellt, können durch die Verwendung dieses Protokolls bereits einige Reihe von Sicherheitszielen für die Daten während der Übertragung erfüllt werden:

- Authentizität der Kommunikationspartner vor Verbindungsaufbau
- Vertraulichkeit durch Verschlüsselung (Unkenntlichmachung)
- Integrität durch *Message Authentication Codes* (MAC)

Der Schlüsselaustausch zur Verwendung des symmetrischen Algorithmus geschieht bei SSL durch digitale Zertifikate, die öffentlich bei einer CA erhältlich sind und von dieser „beglaubigt“ wurden. Damit können die Kommunikationspartner zugleich authentifiziert werden. In der Beispielanwendung wird zum Zertifikatsaustausch zusätzlich eine Public-Key-Infrastruktur verwendet, über die die Echtheit des Kommunikationspartners garantiert werden kann, sofern das auf dem System vorhandene Wurzelzertifikat gültig ist.

Alle Daten werden nun mit dem symmetrischen Verfahren AES verschlüsselt übertragen, sodass es einem Angreifer nur mit sehr großem Aufwand möglich sein wird, die Bedeutung der übertragenen Daten zu verstehen, sollte er die Verbindung abhören. Die Vertraulichkeit kann somit gewahrt bleiben.

Durch die Verwendung von *Message Authentication Codes* (MAC) in SSL kann die Integrität gewahrt bleiben. Manipulationen der verschlüsselten Daten können so durch das Protokoll zuverlässig erkannt werden.

Die Umsetzung des Protokolls in der Beispielanwendung geschieht auf Anwendungsebene. Für die Implementierung der gesicherten Verbindung über SSL werden Klassen der Standard-Java-Bibliotheken (*javax.net.ssl*) genutzt. Diese sind Teil der Dalvik VM, der Laufzeitumgebung von *Android*. Mögliche Alternativen hierzu wären nativ ausgeführte Bibliotheksfunktionen von *Android*, bei denen ein Performancegewinn zu erwarten wäre.

Dazu hat der Autor dieser Arbeit eine Implementierung erstellt, wobei anzumerken ist, dass bei dieser auf die Verwendung einer PKI verzichtet wurde und stattdessen lediglich das unsichere Vertrauensmodell „Direct Trust“ verwendet wird. Das hierfür nötige Zertifikat des Servers sich befindet in der Implementierung des Prototyps auf der externen Speicherkarte. In einer Produktivumgebung liegt dies an einem sichereren Ort verwahrt, damit kein unbemerkter Austausch des Zertifikats Spoofing-Angriffe ermöglicht. Das Zertifikat liegt dabei im Format *X.509* vor und es wird der Standard *PKCS#12*[1] verwendet.

Genügt das Zertifikat nicht den Anforderungen, und besteht der Verdacht einer Fälschung, so wird es auf Anwendungsebene automatisch abgelehnt.

Für die implementierte Java-Variante konnten bei einem subjektiv wahrgenommenen Performancetest keine problematischen Leistungsprobleme durch die schwache Hardware des G1 festgestellt werden. Die Initialisierung der Verbindung, zu der auch das Einlesen der Zertifikate gehört, nahm dabei bis zu mehreren Sekunden[2] in Anspruch. Dies ist jedoch hinnehmbar, da der Verbindungsaufbau bereits im Hintergrund soweit vorbereitet werden kann, dass dieser dem Benutzer zur Laufzeit als sehr performant und flüssig erscheinen kann.

[1] PKCS: *Public Key Cryptography Standards*
[2] In der Regel dauerte dies ca. zwei Sekunden

Da es sich bei der Anwendung um einen echten Thin-Client handelt, kann die Menge an Datenumsatz auf dem Kommunikationskanal auf ein Minimum reduziert werden, sodass nur sehr wenige Nutzdaten übertragen werden müssen. Die Verschlüsselung einer solch geringen Datenmenge sollte so selbst auf mobilen Geräten ohne weitere Leistungsprobleme möglich sein.

4.2.2.2 Verschlüsselte Datenspeicherung

Die Zugriffskontrollverfahren von *Android* können umgangen werden, z.B. im Falle eines direkten Speicherzugriffes bei einem direkten physischen Zugriff auf das Gerät oder durch Malware mit Root-Rechten.

Um dennoch die Vertraulichkeit der verarbeiteten Informationen gewährleisten zu können, sollte die Speicherung der schützenswerten Datenobjekte verschlüsselt auf dem Gerät stattfinden. Zahlungs-, Kunden- und Anwendungsdaten besitzen einen hohen Schutzbedarf in der Vertraulichkeit. Es stehen dazu sowohl symmetrische als auch asymmetrische Verfahren zur Verfügung. Weiter ist die Wahl der Schlüssel wichtig. Dies ist je nach Datum anders:

- Zahlungsdaten: symmetrisch über PIN zuzüglich weiterer Authentifikationsmaßnahmen
- Kundendaten: asymmetrisch
- Anwendungsdaten: nur durch die Ausführungsumgebung realisierbar

Zahlungsdaten

Die Zahlungsinformationen sind eine gemeinsame geheime Information zwischen dem Kunden und dem MPSP, werden für den Zahlungsverkehr verwendet und im System durch das *Zahlungsinfo* Objekt gespeichert (vgl. Abbildung 3.15).

Eine Nutzung der Daten sollte nur mit Zustimmung des Kontoinhabers stattfinden, sodass dieser sich vor Verwendung der Zahlungsdaten zwingend authentifizieren muss. *Android* bietet dazu keine geeigneten Authentifizierungsmöglichkeiten an, sodass dies durch eine eigene Implementierung auf Anwendungsebene erfolgen kann.

Durch die PIN-Eingabe in Anwendungsfall Kundeneinstellungen (vgl. Abschnitt 3.2.6) stellt dazu geheimes Wissen zur Verfügung, welches zur Authentifizierung des Benutzers am Gerät verwendet werden könnte. Für die Sicherheit des Authentifizierungsverfahrens ist hierbei die Komplexität der PIN entscheidend, sowie auch der sichere Umgang des Benutzers mit dieser, wobei Zweites sich vom technischen System nicht effizient durchsetzen lässt.

Es ist daher anzuraten, weitere sicherere Authentifizierungsverfahren zwischen Kontoinhaber und Zahlungsdienstleister einzusetzen, um die Authentizität beider Parteien garantieren zu können.

Kundendaten

Die Kundeninformationen sind in den zugehörigen Datenobjekten *privates Ticket* und *Kontrollmedienreferenz* enthalten und werden im Anwendungsspeicher gehalten, nachdem diese durch den Fahrgast einmalig eingegeben wurden und werden bei einer Fahrscheinkontrolle durch Kontrolleure überprüft.

Unter Beachtung des Schutzbedarfes der Vertraulichkeit, insbesondere der Privatheit, kann und soll eine Offenlegung der Identität des Fahrgastes nur mit dessen Zustimmung erfolgen.

Die verwendete Technik mit den Barcodes bietet ein für den Kunden transparentes und einfaches Verfahren. Eine Datenübertragung seiner persönlichen Informationen ist für ihn augenscheinlich feststellbar und erfolgt nur mit seiner Zustimmung, indem er dem Kontrolleur den Barcode auf dem Display des mobilen Gerätes scannen lässt. Die Übertragung von „unsichtbaren Daten“ geschieht nicht.

Da ein Auslesen durch den Fahrgast selbst nicht notwendig ist, können die Kundendaten unmittelbar nach der Eingabe mit einem öffentlichen Schlüssel des ÖPV-Unternehmen verschlüsselt werden und sind damit nur noch für die berechtigen Kontrolleure auslesbar. Nur diese besitzen die zugehörien privaten Schlüssel und können damit die Kundendaten entschlüsseln. Dadurch werden Kontrollen ermöglicht, bei denen die Fahrgäste, im Gegensatz zur Entschlüsselung der Zahlungsinformationen, ihre PIN nicht eingeben müssen, was zu einer besseren Benutzbarkeit des IT-Systems führt.

Anwendungsdaten

Diese liegen nach der Installation im Klartext im Speicher des Gerätes vor, jedoch in kompilierter Form.

Ein Verlust der Vertraulichkeit von Informationen über verwendete Algorithmen, die in der Client-Anwendung Anwendung finden, ist durch *Reverse Engineering* zu erwarten, erfordert jedoch einen vergleichsweise hohen Aufwand des Angreifers. Da die Sicherheit der Fahrplanauskunfts- und Ticketinganwendung nicht in der Geheimhaltung der verwendeten Algorithmen liegen sollte, kann ein Verlust der Vertraulichkeit diesbezüglich hinnehmbar sein.

Eine Verschlüsselung dieser kann nicht auf Anwendungsebene umgesetzt werden, sondern liegt im Zuständigkeitsbereich der umgebenen Plattform. Android bietet dazu keine Maßnahmen an.

4.2.2.3 Digitale Signierung

Für Fahrplan-, Zahlungs-, Ticket- und Kundeninformationen wurde ein hoher Schutzbedarf bezüglich der Integrität festgestellt. Durch die Verwendung von SSL kann Datenintegrität auf dem Übertragungsweg bereits garantiert werden. Durch den Einsatz digitaler Signaturen kann die Datenintegrität auch im Speicher, sowie zur Laufzeit gewährleistet werden, da die Signatur direkt und nur zusammen mit den Daten gültig ist.

Fahrplandaten

Die Fahrplandaten enthalten die Informationen, die bei einer Fahrplanauskunft vom Anwender eingegebenen wurden, sowie den angeforderten und vom Betreiber herausgegebenen Fahrplan.

Durch digitale Signierung der zugehörigen Datenobjekte (siehe Abbildung 3.14) können diese geschützt werden, sodass diese nicht durch unberechtigte Manipulationen verloren oder verfälscht werden können.

Fahrplandaten werden zunächst einmal nur sehr kurzzeitig verwendet und eine permanente Speicherung dieser Daten auf dem Gerät ist in dem Prototyp nicht vorgesehen. Sie befinden sich lediglich für einen sehr kurzen Moment im Speicher des Gerätes. Konkret sind dies die Zeiträume zwischen Benutzereingabe und Übermittlung an den Server, sowie nach Empfang der Antwort auf dem Gerät bis zur Präsentation der Informationen über das Display. Eine unbemerkte Manipulation in genau diesen kurzen Zeitfenstern könnte dennoch die Integrität der Fahrplandaten gefährden.

Durch die Verwendung von digitalen Signaturen kann dies verhindert werden. Die Echtheit und Glaubwürdigkeit der Fahrplandaten wäre damit garantiert, da Manipulationen zuverlässig erkannt werden können.

Zahlungsdaten

Der Kunde möchte in jedem Fall nach einer Zahlung ein gültiges Ticket erhalten. Somit darf der Zahlungseingang beim Händler von diesem nicht abgestritten werden können. Ebenso darf der Erhalt eines gültigen Tickets vom Kunden nicht abgestritten werden können.

Die daraus entstehenden Verbindlichkeitsanforderungen können ebenfalls durch den Einsatz digitaler Signaturen erfüllt werden. Die Verantwortung hierrüber sollte auf den MPSP ausgeweitet werden, da dieser an dem Zahlungsvorgang beteiligt ist. Weitere Details über den Ablauf des digitalen Signaturverfahrens bietet der Autor dieser Arbeit diesbezüglich nicht an. Es sollte sich jedoch auf Anwendungsebene umsetzten lassen.

Ticketdaten

Die Ticketdaten enthalten bereits eine digitale Signatur des Betreibers (vgl. Abschnitt 3.3.5), sodass die Integrität dieser während der Speicherung sichergestellt ist. Manipulationen werden erkannt, und führen unweigerlich zur Ungültigkeit des Tickets.

Kundendaten

Durch das Kontrollmedium, sowie den privaten Schlüssel, findet dabei eine gegenseitige Authentifizierung statt. Das Kontrollmedium wird lediglich dazu verwendet wird, um zu beweisen, dass es sich um den zur Nutzung des Tickets berechtigten Fahrgast handelt. Die genaue Identität hingegen ist irrelevant.

Anwendungsdaten

Die Integrität der Anwendungsdaten lässt sich ebenfalls durch digitale Signaturen umsetzen. Android-Installationspakete enthalten bereits eine digitale Signatur des Entwicklers.

Diese dienen jedoch lediglich dem Zweck, dass ein Entwickler IPC zwischen seinen Anwendungen betreiben kann[1] und um Updates zu ermöglichen. Bei der Installation findet von *Android* lediglich eine Überprüfung auf die Existenz einer solchen Signatur statt, jedoch wird die Echtheit dieser nicht überprüft.

Im Rahmen der Softwareinstallation der Client-Anwendung (siehe 3.2.1) sollte daher unmittelbar nach dem Download eine Verifikation der Integrität der Anwendungsdaten durchgeführt werden. Diese Prozedur müsste dabei manuell durch den Anwender mittels weiterer softwaretechnischer Werkzeuge durchgeführt werden, da die Installationsroutine von *Android* dies nicht unterstützt und generell jede Software ungeprüft installiert. Diese Verifikation ist dabei optional, da sie durch den Anwender übersprungen werden könnte, sodass man nicht davon ausgehen kann, dass diese auch tatsächlich durchgeführt wurde.

Bei Verwendung des *Android Market* ist die Anwendungsinstallation automatisiert und beginnt unmittelbar nach dem Download. Diese Prozedur ist auf Benutzbarkeit optimiert und der Download erfolgt transparent für den Benutzer, und lässt keine derartige manuelle Verifikation zu. Der *Android Market* kann für die Verteilung der Client-Anwendung daher nicht verwendet werden, da sicherheitskritische Probleme ungeachtet bleiben.

[1] Dies wird durch Verwendung einer Shared-User-ID im Linux-Kernel realisiert

Die Anwendung sollte daher viel mehr über eine als vertrauenswürdige bekannte Identität heruntergeladen werden, die sich vorher beim Fahrgast authentifiziert hat. Dies kann beim Download über den Webbrowser unter Verwendung von Zertifikaten geschehen und liegt außerhalb des Einflussbereiches des Entwicklers.

Für Integrität der Anwendungsdaten kann also nicht garantiert werden. Durch entsprechende Hinweise an die Fahrgäste müssten diese für das Thema Sicherheit sensibilisiert werden, um sich fortan diesem gegenüber verantwortungsvoll zu verhalten. Dies bietet zwar keine absolute Sicherheit, kann im eCommerce dennoch eingesetzt werden, da das Risiko dadurch handhabbarer wird.

Durch technische Maßnahmen der Plattform ließen sich auch unter *Android* sowohl Sicherheit als auch Benutzbarkeit bei der Anwendungsinstallation miteinander vereinbaren. Einige Verfahren zur Umsetzung sicherer Softwareverteilung bietet hierzu Pern Hui Chia in seiner Masterarbeit *Secure Software Installation via Social Rating* [Chia08].

4.2.2.4 Verfügbarkeit der Daten

Fahrplan- und Zahlungsdaten besitzen keinen besonderen Schutzbedarf hinsichtlich der Verfügbarkeit. Die Fahrplandaten benötigten keine permanente Speicherung und können bei Bedarf, wie auch die Zahlungsdaten bei einem Verlust der Verfügbarkeit, durch den Anwender ohne Weiteres erneut eingegeben werden.

Ticket-, Kunden- und Anwendungsdaten hingegen besitzen einen hohen Schutzbedarf bezüglich der Verfügbarkeit.

Die Verfügbarkeit der Daten lässt sich über Backup und Replikationsverfahren steigern. Fällt eine Ressource aus, die für die Speicherung verantwortlich ist, so können die Daten auf diese Weise rekonstruiert werden. Als geeignete Medien für die Replikate könnten die Speicherkarte des mobilen Gerätes, sowie auch entfernte Backup-Server verwendet werden.

Die Kompetenz für die Verfügbarkeit sämtlicher anwendungsspezifischer Daten könnte durch einen Dienst der Plattform - hier *Android* - übernommen werden. Damit bestünde eine zentrale Backup-Lösung für alle Android-Anwendungen, und eigene Verfahren jeder Anwendung würden hinfällig werden.

Android bietet hierzu keine jedoch keine Möglichkeit an. Für die Beispielanwendung lassen sich dazu ohne Weiteres eigene Replikationsmechanismen auf Anwendungsebene implementieren.

Bei mobilen Geräten bieten sich besonders entfernte Backup-Server als ein geeignetes Medium an, die die Datenverfügbarkeit auch im Falle eines physischen Verlustes des Gerätes gewährleisten können und dem hohen Schutzbedarf der Ticket- und Kundendaten in puncto Verfügbarkeit kann nachgekommen werden.

4.2.2.5 Zusammenfassung

Auf Anwendungsebene ließen sich bereits eine Reihe von Sicherheitsmaßnahmen umsetzten, mit der die Sicherheitsstrategie verfolgt werden konnte. Dabei wurden eine Reihe kryptographischer Verfahren eingesetzt, durch die der Schutz der Datenobjekte garantiert werden kann, welche die zu schützenswerten Informationen enthalten.

Durch die Verwendung von SSL auf Anwendungsebene konnte der Kommunikationskanal vor unautorisiertem Zugriff geschützt werden, sodass die folgenden Schutzziele erfüllt werden konnten:

- Authentizität der Kommunikationspartner durch Zertifikate unter Verwendung einer PKI

- Vertraulichkeit durch Datenverschlüsselung
- Integrität durch *Message Authentication Codes*

Die Vertraulichkeit der Daten bei der Speicherung konnte durch Verschlüsselung erreicht werden, wodurch Geheimhaltung vor unautorisierten Dritten gegeben ist. Ergänzend dazu konnten die Daten mit digitalen Signaturen versehen werden, sodass auch die Integrität dieser garantiert ist.

Zur Verbesserung der Verfügbarkeit können Replikationsstrategien auf Anwendungsebene verfolgt werden, bei denen Backup-Server eingesetzt werden.

Mit Ausnahme der Anwendungsdaten konnten für die anderen Informationen (Fahrplan-, Zahlungs-, Ticket-, und Kundeninformationen) bisher alle Sicherheitsziele auf Anwendungsebene erreicht werden. Tabelle 4.6 zeigt dies schematisch:

<table>
<tr><th>Daten</th><th>Verfügbarkeit</th><th>Vertraulichkeit</th><th>Integrität</th></tr>
<tr><td>Im Speicher</td><td>Replikation</td><td>Verschlüsselung</td><td rowspan="2">Digitale Signaturen</td></tr>
<tr><td>Zur Laufzeit</td><td colspan="2">Isolation durch Android</td></tr>
<tr><td>In Übertragung</td><td>Nicht Teil dieser Arbeit[1]</td><td colspan="2">SSL</td></tr>
</table>

Tabelle 4.6: Umgesetzte Sicherheitsmaßnahmen auf Anwendungsebene

Dies setzt jedoch voraus, dass Anwendung funktionssicher arbeitet, d.h. dass ihre Ist-Funktionalität mit der Soll-Funktionalität übereinstimmt. Dem kann durch Schutz der Integrität der Anwendungsdaten gerecht werden, welcher im Folgenden durch die Sicherungsmaßnahmen von Android umgesetzt werden soll.

Der Autor sieht die Kompetenz hierfür bei der zugrundeliegenenden Plattform, weil dies auf Anwendungsebene nicht umsetzbar sei, begründend dies mit der Fragestellung „W*ie kann eine unautorisierte manipulierte Anwendung, die damit die Integrität verletzt, ihre eigene Integrität sicherstellen?*“

4.2.3 Sicherheitsmaßnahmen von Android

Die Anwendungsdaten konnten bisher noch nicht geschützt werden. Dies ist Aufgabe der umgebenen Plattform und soll im Folgenden durch die Betrachtung der Sicherheitsmaßnahmen von Android erfolgen.

[1] Die Gewährleistung der Datenverfügbarkeit während der Übertragung setzt die Kontrolle über die verwendete Kommunikationsinfrastruktur und allen darauf befindlichen Knoten voraus.

4.2.3.1 Verwendung von Java

Bei Entwicklung des Prototyps mit dem SDK wurde Java verwendet. Durch die von Java angebotenen Bibliotheken konnten zuvor bereits einige Sicherheitsfunktionalitäten auf Anwendungsebene implementiert werden. Darüber hinaus bietet Java eine sichere Anwendungsentwicklung, da Java als höhere Programmiersprache eine typsichere Programmierung ermöglicht. Damit können Bedrohungen bereits verhindert werden, bevor sie entstehen[1].

Buffer Overflow Angriffe werden verhindert

So beispielsweise Buffer-Overflow-Angriffe: Anwendungen, die mit dem *Android* SDK entwickelt wurden, werden innerhalb der DVM ausgeführt, welche die Grenzen der Speicherbereiche zur Laufzeit überwacht. Buffer-Overflow-Fehler in Programmen können also nicht durch die Anwendung selbst entstehen, so auch nicht in der Beispielanwendung.

Weiter besteht seit *Android* 1.5 mit der Einführung des *NDK*s auch die Möglichkeit, nativen Programmcode in C oder C++ auszuführen, sodass der Ausführungspfad eines Programmes die virtuelle Maschine verlassen könnte. Nativer Code bietet in der Regel eine größere Angriffsfläche zur gezielten Ausnutzung von Schwachstellen, durch die Schadcode in das System eingeschleust und zur Ausführung gebracht werden kann.

Die Beispielanwendung verwendet ausschließlich Code, der mit dem SDK in Java geschrieben ist. Angriffe auf die Anwendung selbst, bei denen beispielsweise vorsätzliche Speicherüberläufe hervorgerufen werden, sind daher unmöglich.

Java-Klassenbibliothek

Weiter wurden in der Beispielanwendung Sicherheitsfunktionalitäten genutzt, die durch die Java-Klassenbibliothek zur Verfügung gestellt wurden. Durch die viel frühere Markteinführung stuft der Autor diese Technologie tendenziell als sicher ein, relativ zu neuartigen nativen Bibliotheksfunktionen von *Android*.

Keine Sicherheit durch Virtualisieung

Der kompilierte Anwendungscode bzw. Dalvik-Bytecode wird im Rahmen der stattfindenden Virtualisieung zur Laufzeit innerhalb der DVM ausgeführt. Die dabei stattfindende Virtualisieung kann generell als ein Sicherheitskonzept verwendet werden, wird in *Android* als ein solches aber nicht eingesetzt.

Avik Chaudhuri [Chaudh09] berichtet zwar, dass durch die Virtualisierung die Anwendungen isoliert voneinander ausgeführt werden können: *"Next, each process runs on its own Java VM, so the application's code runs in isolation from the code of all other applications."*

Laut einiger Sicherheitsverantwortlicher von *Android* würde die Sicherheit jedoch nur über die Trennung der Prozesse auf verschiedene UIDs zustande kommen. Die „Java VM", wie Chaudhuri die DVM nennt, ist in *Android* nicht die Komponente, die für gewünschte Isolation zuständig ist. Das *Android* Entwicklerteam setzt hier ganz auf die jahrelang etablierten UNIX-Sicherheitsmechanismen. Die DVM hingegen würde lediglich für die gewünschte Ausführungsgeschwindigkeit der mobilen Anwendungen sorgen.

[1] Wie von Claudia Eckert als Technik zur Bedrohungsabwehr eingeordnet (vgl. Abschnitt 2.3.4).

4.2.3.2 Authentifikation am Gerät

Durch den mobilen Anwendungsbereich gewinnt die Authentifikation des Benutzers am Gerät an Bedeutung, da mobile Geräte dort einem erhöhten Risiko durch direktem physischen Zugriff unberechtigter Dritter ausgesetzt sind, wie es bei einem Diebstahl der Fall sein könnte.

Daher sollte vor der Verwendung des Gerätes die Authentizität des Benutzers durch entsprechende Maßnahmen gewährleistet werden. Die Kompetenz dafür sieht der Autor primär in der Plattform *Android*, da dies kein anwendungsspezifisches Problem darstellt.

Für die Authentifizierung des Benutzers an der Plattform *Android* gibt es dazu als Verfahren die Nutzung eines „Patterns“. Dies basiert auf geheimem Wissen, ist vergleichbar mit einer Passwortabfrage und geschieht eher über visuelle Verfahren, die für den Benutzer einprägsamer sein könnten. Den Mechanismus hat der Autor nicht tiefergehend untersucht und kann daher keine präzisen Aussagen über dessen Sicherheit treffen.[1]

Weiter findet diese Authentifikation vor Verwendung des Gerätes nur optional statt, da sie vom Benutzer deaktiviert bzw. nicht aktiviert wird, d.h. man kann nicht davon ausgehen, dass es sich bei dem Anwender tatsächlich um die behauptete Identität handelt, zumal eine ständige Authentifikation am Gerät einen Einschnitt in der Benutzbarkeit bedeutet, speziell auf mobilen Geräten.[2]

Die aufgestellte Sicherheitsstrategie bezüglich der Authentifikation des Benutzers am Gerät konnte durch die verwendete Plattform *Android* nicht umgesetzt werden. Es werden daher weitere Authentifizierungsmaßnahmen auf Anwendungsebene benötigt, um der Forderung nach Authentizität nachzukommen, die für ein Erreichen der Schutzziele Vertraulichkeit und Integrität erforderlich sind. Die Anwendungsdaten verbleiben also ungeschützt.

4.2.3.3 Zugriffskontrolle

Die aus Linux bekannten Verfahren zur Zuweisung unterschiedlicher Zugriffsrechte an verschiedene Benutzer werden in *Android* stattdessen bereits an Anwendungen vergeben, um den gewünschten Isolationseffekt dieser untereinander herbeizuführen.

Android bietet somit zum Schutze der Anwendungsdaten die Isolation vor anderen Android-Anwendungen an. Damit können diese schon relativ gut gegen fremde Anwendungen auf dem System geschützt werden.

Subjekte bezüglich des eingesetzten Zugriffskontrollverfahrens sind die Anwendungen, bzw. die Prozesse, die sie beherbergen. Benutzer oder Rollen als Subjekte sind in Android nicht vorgesehen und können daher von der Plattform nicht zur Differenzierung und Festlegung von umfangreichen Zugriffskontrollen verwendet werden.

Bekannte Betriebssysteme von Desktop-Systemen unterscheiden hier gewöhnlicher weise zwischen einem Anwender mit Administratorrechten und einem normalen Benutzer. Nur der Administrator ist hierbei zum Installieren von Software berechtigt. *Android* kennt nur einen Anwender, welcher über sämtliche Zugriffsrechte verfügt, wie auch zur Veränderung des Systems durch die Installation neuer Software.

---

[1] Schätzungsweise könnte das Verfahren eine leicht verbesserte Benutzbarkeit bieten, der gegenüber jedoch Abstriche in der objektiven Sicherheit hingenommen werden müssten, begründend durch die geringere Komplexität des geheimen Wissens.

[2] Hinsichtlich der Interaktionsrate (siehe Abschnitt 2.2.2)

Android unterscheidet nicht zwischen verschiedenen Benutzern oder Rollen und kennt nur „den einen Benutzer“, der über ein festgelegtes Maß an Berechtigungen auf dem System verfügt. Differenzierte Zugriffsrechte für verschiedene Personengruppen oder Rollen sind in *Android* nicht möglich und können nur auf Anwendungsebene umgesetzt werden. So auch in dem Beispielszenario.

4.2.3.4 Anwendungsverteilung

Die Anwendungsdaten sollten durch eine sichere Softwareinstallation mit der Zustimmung des Besitzers des Endgerätes durchgeführt werden. Danach ist lediglich der Anwender zum Zugriff auf die Anwendung berechtigt, und benötigt demnach Ausführungsrechte, um die Anwendung starten zu können. Für Softwareupdates erhält der Hersteller zusätzliche Schreibrechte auf die Anwendungsdaten.

Dies setzt eine Authentifizierung des Benutzers am Gerät voraus und sollte durch die Plattform *Android* geschehen, da diese über sämtliche Kompetenzen bezüglich der Installation von Anwendungen verfügt.

Der Anwendungscode lässt sich bei Verwendung des SDK in ein Paket exportieren, welches zur Installation beim Endanwender verwendet werden kann. Die anschließende Verteilung erfolgt dabei über eine beliebige Download-Plattform, wie beispielsweise auch über den von *Google Inc.* betriebenen *Android* Market.

Die Verantwortung für die Deinstallation liegt wie auch die Installation bei der Plattform *Android* und bedarf keiner weiteren Maßnahmen.

4.3 Bewertung

Maßnahmen zur Erfüllung der Schutzziele bezüglich der schützenswerten Informationen				
		Verfügbarkeit	Vertraulichkeit	Integrität
Im Speicher	Fahrplaninformationen	Normal	Normal	Digitale Signaturen
	Zahlungsinformationen	Normal	Verschlüsselung	
	Ticketinformationen	Backups	Normal	
	Kundeninformationen		Verschlüsselung	
	Anwendungsdaten	Neuinstallation	Hoch	Hoch
Zur Laufzeit	Fahrplaninformationen	Normal	Normal	Digitale Signaturen
	Zahlungsinformationen	Normal	Hoch	
	Ticketinformationen	Hoch	Normal	
	Kundeninformationen	Hoch	Hoch	
	Anwendungsdaten	Hoch	Hoch	Hoch
In Übertragung	Fahrplaninformationen	Nicht Teil dieser Arbeit	SSL	
	Zahlungsinformationen			
	Ticketinformationen			
	Kundeninformationen			
	Anwendungsdaten		Hoch	Hoch

Tabelle 4.7: Bewertung der Sicherheitsmaßnahmen

Der geforderte Schutzbedarf für die Anwendungsdaten ist hoch. Dieser Forderung kann aufgrund der mangelnden Authentifikation beim direkten Zugriff auf das Gerät nicht nachgekommen werden.

Durch eine unberechtigte Deinstallation könnte beispielsweise die Verfügbarkeit verletzt werden, und das digitale Ticket wäre verloren, und würde bei einer Fahrkartenkontrolle fehlen. Viel bedrohlicher ist jedoch das Ersetzen der Anwendung durch ein Plagiat der Anwendung, welches für Phishing-Angriffe verwendet werden könnte.

4.3.1 Testen auf Schwachstellen anhand der Bedrohungen

Aufbauend auf der Bedrohungsanalyse aus Abschnitt 4.1.3 soll die Client-Anwendung nun - nach der Erweiterung um sicherheitstechnische Maßnahmen - noch einmal abschließend untersucht werden.

4.3.1.1 Kommunikationsschwachstellen

Abhören

Ein bloßes Abhören der Verbindung reicht für einen Angreifer zum Gefährden der Schutzziele nicht aus. Die Bedeutung bleibt ihm Verborgen, da die Daten mit dem SSL-Protokoll verschlüsselt übertragen werden.

Stören

Durch das Einspielen von Daten in das Netzwerk kann die Kommunikation gestört werden. Die drahtlose Verbindung bietet für einen solchen Angriff gute Bedingungen, allerdings gibt es bei dieser Art von Angriff in der Regel kein Motiv für Täter, sodass ein solcher Angriff relativ unwahrscheinlich ist.

Knacken der Verschlüsselung

Das Entschlüsseln der Verschlüsselten Daten ist nur mit dem geheimen (synchronen) Sitzungsschlüssel möglich. Ein Knacken über reine Brute-Force-Verfahren ist sehr aufwändig und kann mit heutigen Rechnersystemen nicht effizient und schnell genug durchgeführt werden, um die Informationen noch während der Sitzung gefährden zu können. Daher wird ein Angreifer effizientere Verfahren verwenden, zumal er sonst nur eine Sitzung entschlüsseln würde, die wahrscheinlich zum Zeitpunkt der Entschlüsselung schon lange beendet sein wird. Er wird daher versuchen, den Sitzungsschlüssel auf eine andere Art und Weise zu kompromittieren.

Ein Angreifer könnte daher versuchen, den Austausch des Sitzungsschlüssels zwischen Client und Server abzuhören. Jedoch findet der Schlüsselaustausch bei TLS über moderne asynchrone kryptographische Verfahren statt, sodass er zunächst in Besitz eines privaten Schlüssels gelangen müsste. Die Besitzergreifung bzw. das Knacken des privaten Schlüssels scheint wesentlich attraktiver, da damit nicht nur die Daten einer Sitzung, sondern die Daten aller folgenden Sitzungen offengelegt werden würden.

Ein Angriff durch bloßes Knacken einer der Schlüssel ist daher sehr unwahrscheinlich.

Maskierungsangriffe

Wesentlich erfolgsversprechender wären Spoofing- und/oder *Man-in-the-Middle*-Angriffe, bei dem ein Angreifer zunächst eine Position *in the Middle* einnehmen muss. Dies wird in diesem Szenario durch Spoofing eines eigenen Access-Points realisierbar, der beispielweise an Haltestellen eingesetzt werden könnte, und unter der vollen Kontrolle eines Angreifers steht.

Durch Maskieren mit einer falschen Identität kann ein Angreifer beide Geschäftspartner täuschen, indem er sich als der jeweils andere ausgibt und sie glauben lässt, sie würden direkt miteinander kommunizieren. Die Tatsache, dass es sich um eine verschlüsselte Verbindung handelt, kann alleine keine Sicherheit bieten. Daher werden in diesem Szenario Zertfikate verwendet werden, die die Echtheit der Kommunikationspartner bestätigen können. Diese wurden von einer der *Certification Authority* beglaubigt.

Fälschen der Zertifikate

Ein Angreifer könnte nun versuchen, das Server-Zertifikat zu fälschen und zu hoffen, dass dieses auf Client-Seite als vertrauenswürdig eingestuft und akzeptiert wird.

Bei einer vergleichbaren Webanwendung würde der Browser an dieser Stelle den Benutzer fragen, ob er dem gefälschten Zertifikat vertraut. Dadurch würde der Anwender Teil der Sicherheitskette werden und eine kritische Rolle beim Erfolg der Sicherungsmaßnahmen spielen. In der Beispielanwendung geschieht dies automatisch, und eine falsche Entscheidung des Benutzers kann somit die Gesamtsicherheit des Systems nicht gefährden.

Durch den Einsatz einer Public-Key Infrastruktur (PKI) in der Beispielanwendung kann die Echtheit des Zertifikats überprüft werden.

Um einen Angriff dieser Art dennoch durchführen zu können, könnte ein Angreifer das System manipulieren, sodass auch gefälschte Zertifikate akzeptiert werden. Die Integrität der Anwendungsdaten könnte diese Bedrohung verhindern, was im Rahmen der Beispielanwendung jedoch nicht stattfinden konnte.

Bewertung der Kommunikationssicherheit

Das in der Beispielanwendung verwendete Verschlüsselungsprotokoll SSL kann somit eine zuverlässige Ende-zu-Ende-Sicherheit auf Anwendungsebene gewährleisten, jedoch erst, nachdem ein geheimer Schlüssel erfolgreich ausgetauscht werden konnte. Ist der *Man-In-The-Middle* jedoch bereits vor der Übergabe des Schlüssels aktiv gewesen, so kann dieses Verfahren keinen Schutz mehr bieten. Durch die Verwendung von Zertifikaten im Rahmen einer PKI konnte dieser Forderung dennoch nachgekommen werden.

Der nötige Aufwand für einen Angreifer konnte durch diese Sicherheitsmaßnahmen sehr stark vergrößert werden und bietet eine zu kleine Angriffsfläche. Andere Schwachstellen versprechen mehr Aussicht auf Erfolg.

4.3.1.2 Direkter Zugriff auf das Gerät

Direkter Speicherzugriff

Angreifer können die Kontrollen von Android beim Zugriff auf den Speicher umgehen.

Dies ist durch direkte Speicherzugriffe auf die Hardware, oder durch das Erlangen von Root-Rechten auf dem System möglich.

Dies ermöglicht einen Zugriff auf die Daten des Client. Durch die eingesetzten Schutzmaßnahmen konnten diese jedoch alle ausreichend geschützt werden. Sie liegen nicht im Klartext im Speicher und unautorisierte Manipulationen bleiben durch die eingesetzten digitalen Signaturen nicht unerkannt.

Lediglich die Anwendungsdaten wären angreifbar. Diese liegen in kompilierter Form vor und durch *Reverse Engineering* könnten die Anwendung derart manipuliert werden, dass die schützenswerten Daten an den Angreifer übermittelt werden. Dies bedeutet eine Verletzung der Datenintegrität. Sämtliche schützenswerten Informationen würden offengelegt werden und den aufgestellten Sicherheitsanforderungen kann nicht nachgekommen werden.

Die Anwendung ist somit durch direkten physischen Zugriff angreifbar.

Weiter gilt es, zwischen dem Anwendungsspeicher auf dem Gerät, für welche Root-Rechte benötigt werden, und den Speicherkarten, bei denen keine Zugriffskontrollen stattfinden, zu unterscheiden. Die Fahrplanauskunfts- und Ticketinganwendung verwendet die externen Speicherkarten nicht, sodass über diese keine Angriffe auf die schützenswerten Datenobjekte möglich werden.

Hat ein Angreifer bei einem physischen Zugriff auf das Gerät ausreichend Zeit, um die Zugriffskontrollverfahren von *Android* zu umgehen, so stellt dies eine Gefahr für die Anwendung dar.

4.3.1.3 Malware

Eine technische Routine zum Erkennen und Abwehren derartiger Bedrohungen existiert in *Android* im Auslieferungszustand nicht, sodass nach der Installation nur das Isolieren der Anwendung durch das Sandboxing-Prinzip verbleibt. Schädlicher Programmcode wird von *Android* auf dem System „geduldet“ und bleibt bis zur Deinstallation durch den Anwender auf dem Gerät.

Weiter gibt es in *Android* keine Differenzierung zwischen Malware und normalen Android-Anwendungen. Beide werden von der Plattform als gleichwertig betrachtet.

Buffer Overflow

Das Risiko eines Pufferüberlaufs in der Beispielanwendung selber ist gering, da der Anwendungscode in Java geschrieben und innerhalb der DVM ausgeführt wird.

Andere auf dem System laufende Software könnte jedoch Ziel derartiger Angriffe werden, wenn sie eine zuvor beschriebene Schwachstelle besitzen, wodurch auch die Sicherheit des Fahrkartenauskunfts- und Ticketingdienstes bedroht wird.

Im diesem Falle ist zu erwarten, dass der eingeschleuste Code zunächst im Rahmen der Zugriffsrechte des umgebenen Prozesses ausgeführt wird und nur diese eine Anwendung davon betroffen ist, die sich über die Isolationsverfahren von *Android* in einer sicheren Sandbox befindet und keinen weiteren Schaden anrichten kann. Verfügt die fehlerhafte Anwendung jedoch über erweiterte Systemrechte[1], so kann durch eine fehlerhafte andere Anwendung auf der Plattform auch die Sicherheit der Beispielanwendung bedroht werden, da die Isolation der Anwendungen in *Android* umgangen wird.

Phishing

Bezüglich der mangelnden Integrität für die Anwendungsdaten können manipulierte Plagiate der Anwendung auf dem Gerät installiert werden, die zu Phishing-Zwecken genutzt werden können.

[1] Root-Rechte unter Linux

Keine Code Überprüfung zur Laufzeit

Bei derartigen Systemen, die ständig im Hintergrund wachen und das System auf Bedrohungen überprüfen, kann es im speziell im mobilen Umfeld zu Leistungsproblemen kommen. Daher sind die von herkömmlichen Rechnersystemen bekannten Techniken nicht auf die gleiche Weise umsetzbar, sodass angepasste Verfahren verwendet werden sollten, die auch auf solch leistungsbeschränkter Hardware zum Einsatz kommen können.

Der auszuführende Code in *Android*-Anwendungen zeigt typischerweise das Verhalten, nur sehr selten verändert zu werden, sodass sich eine Code-Überprüfung anbieten könnte, die nur einmalig zum Zeitpunkt der Installation oder Update stattfindet.

Keine Code-Signatur durch vertrauenswürdige Dritte

Auch gibt es die Möglichkeit durch eine als vertrauenswürdig angesehene Dritte Instanz die Anwendung als sicher zu zertifizieren. Laut Aubrey-Derrick Schmidt [Schmid08] konnte durch die Einführung einer obligatorischen Code-Signatur für die vergleichbare Symbian-Plattform die Verbreitung schädlicher Software verhindert werden:

„With the introduction of application signing in Symbian [] things changed in the field of Symbian OS malware. In the signing process, a trusted Symbian partner checks the complete source code and binaries for meeting certain criteria, like being free from memory leaks and abusive methods. If the check is successful, the application gets signed with an certificate and stays clearly identifiable through a given unique ID. [] By now, no malware for this Symbian version appeared to the knowledge of the author.”

In *Android* gibt es diesen Ansatz nicht, und es lässt sich jede Software installieren. Es gibt zwar eine digitale Signatur für Anwendungen in *Android*, diese verfolgen jedoch eine ganz andere Intention. Sie dienen lediglich zur Ausführung der Software unter Verwendung der gleichen UID, wie es in Abschnitt 2.5.4 beschrieben wurde. Weitere Signaturen von Drittparteien, die für den Benutzer als vertrauenswürdig eingestuft werden, sind nicht Bestandteil der Installationspakete.

Schadsoftware über Android Market

Der *Android Market bietet* keine besonderen Sicherheitsfeatures und stellt nur geringe Anforderungen an die Software, um dort aufgenommen zu werden. Es ist keinesfalls gewährleistet, dass die dort angebotenen Inhalte für Endkunden sicher verwendbar sind. Ein Code-Review findet nicht statt. Auf diesem Wege könnte der Anwender bereits andere Anwendungen auf dem mobilen Gerät installiert haben, die auch die Beispielanwendung gefährden können.

Einfache Entfernung von „normaler“ Malware

„Normale“ Malware, kann neben der Anwendung auf der gleichen Plattform installiert sein. Diese kann laut Jesse Burns [Burns08] deinstalliert werden:

„Users in this unfortunate state will have to identify and remove the hostile application. Android helps users do this, and [] generally a user’s response to annoying, buggy or malicious software is simply to uninstall it. If the software is disrupting the phone enough that the user can't uninstall it, they can reboot the phone (optionally in safe mode, which stops non-system code from running) and then remove the software before it has a chance to run again.”

Voraussetzung für das Entfernen Solch unerwünschter Malware ist die Erkennung derselben durch den Benutzer. Im Falle eines Trojanischen Pferdes ist dies in der Regel nicht gegeben.

4.3.1.4 Weitere Gefahren

Darüber hinaus könnte die Anwendung weiteren Bedrohungen ausgesetzt sein, die im bisherigen Verlauf nicht thematisiert worden sind, aber nicht gänzlich unerwähnt bleiben sollen.

Signaturen von Android-Anwendungen

Weiter könnten die privaten Schlüssel, mit denen Entwickler Android-Anwendung vor Auslieferung signieren, kompromittiert werden, sodass der Zugriff auf sämtliche von diesem Hersteller installierten Anwendungen stattfinden kann. Die verwendeten Zertifikate sind in der Regel 25 Jahre gültig. Eine Möglichkeit zum Sperren dieser gibt es nicht, da keine CA verwendet wird. Dies stellt eine Schwachstelle in Android dar und bedarf weiterer Nachforschungen.

Absichtliches „Rooten" des Gerätes

Der Autor ist zudem in Kenntnis, dass es Besitzer von Android-Geräten gibt, die gezielt versuchen durch Sicherheitslücken Root-Rechte auf ihrem System zu erlangen. Die Gründe sind dabei verschieden.[1]

4.3.2 Kritikpunkte und mögliche Lösungen

Keine Erkennung von Schadsoftware

In *Android* gibt es keine von Personal Computer bekannten Techniken zum Erkennen und Abwehren von Bedrohungen. Virenscanner und *Intrusion Detection Systeme* (IDS) fehlen im Auslieferungszustand von *Android.* Dies ermöglicht es Schadcode unbemerkt zu bleiben. Zudem werden die Benutzer werden nicht dazu angehalten, derartige Software zu verwenden.

Wie in einem „Horrorszenario" könnte sich so eine Schadsoftware, die über erweiterte Systemrechte[2] verfügt, über viele Systeme ausbreiten und dabei unentdeckt bleiben. Die Vergangenheit hat gezeigt, welches Potential Schadsoftware besitzt. Voraussetzung für die Existenz einer solchen Malware ist eine geeignete Sicherheitslücke in Android.

Einzig die offiziellen Sicherheitspatches für *Android* könnten in einem solchen Szenario für Abhilfe sorgen. Ein reaktionsschnelles Handeln sowohl vom Hersteller der Plattform als auch vonseiten der Endkunden wird dabei vorausgesetzt.

Wie schnell und ob die Sicherheitspatches in einem Ernstfall überhaupt in die betroffenen Systeme eingespielt werden können und ob durch die Hinzuziehung Dritter, wie etwa Virenscanner-Herstellern, Bedrohungen in *Android* besser abgewehrt werden können, vermag der Autor aus heutiger Sicht nicht zu beurteilen und erfordert weiteren Analysebedarf zu dieser Thematik.

Auch sind durch entstehende *Android*-Derivate durch verschiedene Endgeräte-Hersteller keine präzisen Aussagen zur Codesicherheit möglich. Es könnten neue Sicherheitslücken entstehen, von denen nur bestimmte Geräte oder gar ganze Produktgenerationen betroffen sind.

---

[1] Ein Grund ist beispielsweise der Betrieb von Server Funktionalitäten auf dem mobilen Gerät, wie z.B. die Bereitstellung eines W-LAN-Access-Point, über den die UMTS-Verbindung zum Zugang ins Internet genutzt werden kann. Für eine solche Anwendung sind Root-Rechte auf dem *Android*-System erforderlich.

[2] Root-Rechte beim Linux-Kernel

Daher sollten auch in *Android* Sicherheitstechniken zum Abwehren und Verhindern von Bedrohungen eingesetzt werden, wie dies etwa Virenscanner bieten. Alternativ könnten Anwender über diese Gefahr, die von der Schwachstelle der Plattform ausgeht, informiert und gewarnt werden. Durch entsprechende Hinweise[1] könnte technisch weniger versierten Anwender bei der Absicherung ihres mobilen Gerätes geholfen werden, indem sie durch neu erlangtes Sicherheitsbewusstsein diesbezüglich nachbessern können.

Authentifikation

Vor besonders sicherheitskritischen Aktionen wie beispielsweise der Softwareinstallation findet in *Android* keine obligatorische Authentifikation des Benutzers am System statt. Dies erleichtert die Einschleusung von Schadsoftware durch unautorisierte Dritte bei einem direkten physischen Zugriff.

Verfahren, die auf persönlichen Besitz oder biometrischen Merkmalen basieren, wie beispielsweise Smartcards oder Fingerabdrücke, könnten die Authentifizierung des rechtmäßigen Benutzers hier ergänzen oder ersetzten.

Smartcards bieten hierzu eine geeignete Möglichkeit, die als sicher bekannt sind, sofern diese nicht selber kompromittiert werden. Die SIM-Karte des Mobilfunkanbieters, durch die eine mobile Rufnummer mit einem Endanwender gekoppelt ist, könnte in Zukunft zu derartigen Zwecken genutzt werden. Sie wird ohnehin schon mitgeführt. Das Risiko ist dabei bedingt durch das mobile Umfeld entsprechend hoch und ein Schadensfall hätte den Verlust der Smartcard zur Folge. Dieses Verfahren ist daher ungeeignet für die Authentifizierung des rechtmäßigen Besitzers beim seinem mobilen Gerät, wohl aber zur Authentifizierung bei entfernten Servern. Speziell im Mobile Commerce bietet dies viele Chancen auf neue Nutzungsmöglichkeiten. So auch in dem Beispielszenario dieser Arbeit.

Fingerabdrücke, oder andere biometrische Merkmale, sind hingegen einem weit geringeren Risiko durch Diebstahl ausgesetzt und eignen sich besser zur Authentifikation am Gerät. Durch technischen Fortschritt könnten solche Authentifikationsverfahren in Zukunft vollständig transparent in Erscheinung treten, wodurch eine gute Benutzbarkeit von mobilen Geräten gewährleistet werden kann, ohne jedoch Abstriche bei der Authentizität machen zu müssen.

Beide Authentifikationsverfahren könnten den mobilen Anwendungsbereich in Zukunft revolutionieren, da durch sie eine erhöhte Sicherheit in Verbindung mit einer guten Benutzbarkeit realisiert werden könnte.

Benutzerverhalten

Das Benutzerverhalten ist wie in vorigen Kapiteln festgestellt nicht durch die technischen Maßnahmen kontrollierbar, sondern lediglich beeinflussbar. *Android* bezieht den Benutzer bei einigen sicherheitskritischen Entscheidungen mit ein, sodass durch ein falsches Benutzerverhaltens neue Möglichkeiten zum Angriff IT-Systeme ermöglicht werden. So auch in der vorgeführten Anwendung.

Dazu gehört die verwendete Technik zur Installation von Android-Anwendungen.

[1] In einer aktuellen Version von Microsoft Windows beispielsweise befindet sich eine integrierte Firewall, sowie ein „Sicherheitscenter“, der Anwender warnt, wenn keine Antiviren-Software eingesetzt wird.

Bei dieser soll der Benutzer von der Anwendung eingeforderte Rechte[1] auf dem System bestätigen, jedoch kennt er zu diesem Zeitpunkt das zukünftige Verhalten der Anwendung noch nicht oder hat verlässliche Anhaltspunkte zu deren Einschätzung. Dabei könnte es sich um eine Schadsoftware handeln. Er soll also ohne eine vernünftige Grundlage entscheiden, welche Rechte er dieser auf dem System einräumt.

Provokant formuliert könnte man nun sagen, dass mögliche Schäden, die eine solche Software erzeugen kann, vorher vom Benutzer ausdrücklich genehmigt wurden, da sie ohne die Zustimmung des Anwenders gar nicht hätten eintreten können. Im Nachhinein ist dieser natürlich nicht mehr damit einverstanden. Dies stellt ein Problem in *Android* dar.

Als eine Hilfestellung bei Verwendung des *Android Market* gibt es die Möglichkeit, die Software zu bewerten und bei Nichtgefallen entsprechende Hinweise zu hinterlassen, um andere Anwender über ungewünschte Programmaktivitäten zu informieren., damit diese nicht den gleichen Fehler begehen und dabei negativ überrascht werden.

[1] Dies geschieht über die explizit erlaubten Permissions der Android-Anwendung.

5 Fazit und Ausblick

5.1 Zusammenfassung

Diese Arbeit beschäftigte sich mit der Sicherheit von *Android* in mobilen Anwendungsbereichen des eCommerce. Zunächst wurden einige Grundlagen rund um diese zentralen Begriffe dieser Arbeit gelegt: mobile Geräte, Sicherheit, eCommerce und *Android*, welches ein neuartiges Sicherheitskonzept verwendet, dessen Erfolg zum Zeitpunkt des Schreiben dieser Arbeit noch abzuwarten bleibt.

Ziel der Arbeit war es festzustellen, ob *Android* für derartige Anwendungen eine geeignete Plattform bietet, und ob diese den Anforderungen an die Sicherheit genügen kann. Stellvertretend für alle machbaren Anwendungen wurde dazu ein konkretes Beispielszenario aus dem Mobile Ticketing entworfen.

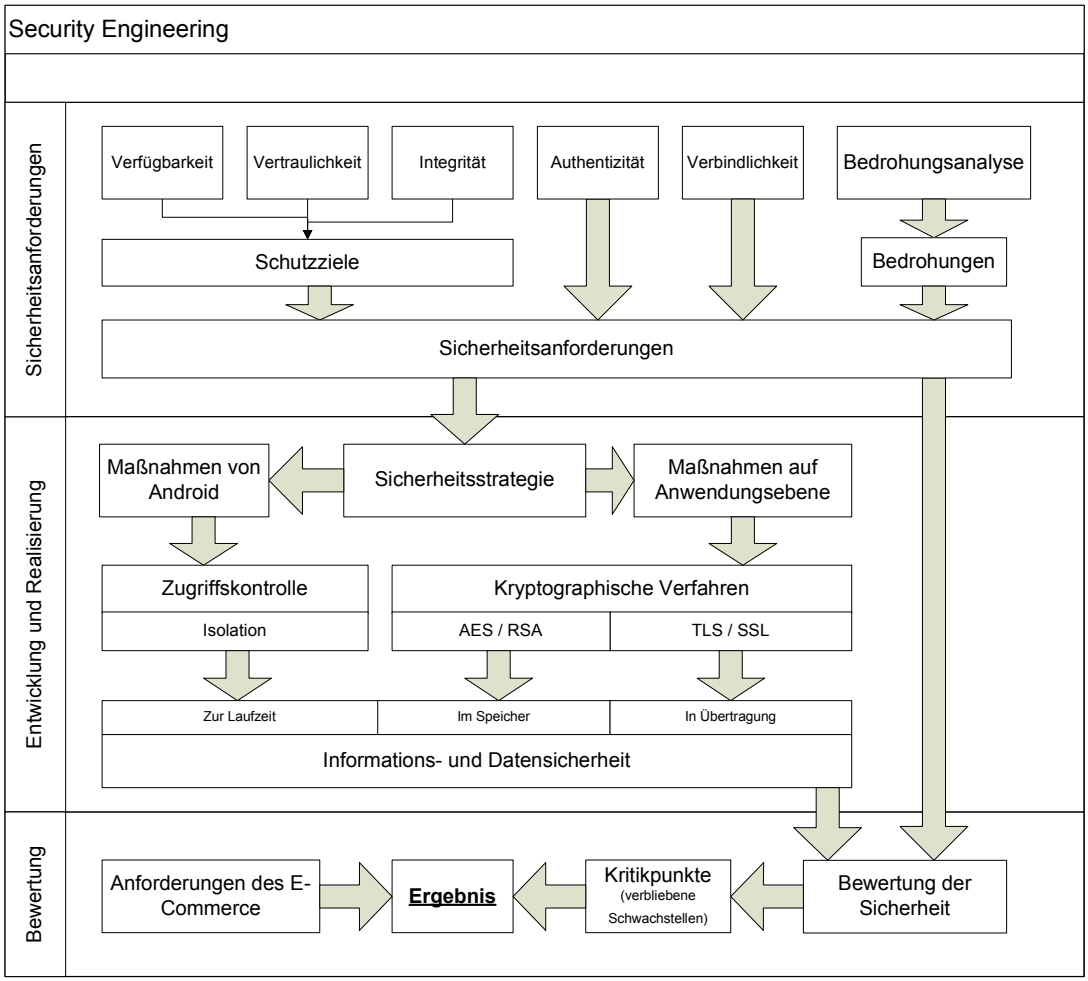

Abbildung 5.1: Security Engineering

Es umfasste die Aufstellung und Realisierung sowie eine abschließende Analyse und Bewertung der Sicherheitsanforderungen, aus der erreichte Maß an Sicherheit veranschaulicht werden konnte.

Abbildung 5.1 verdeutlicht das Vorgehen.

5.2 Fazit

Bezüglich der Plattform konnten mittels der Beispielanwendung sowohl einige positive als auch einige negative Aspekte festgestellt werden.

Es konnte festgestellt werden, dass die Plattform wie die meisten anderen Plattformen keine besonders hohe Sicherheit bieten kann. Für viele eCommerce Szenarien mag dies dennoch ausreichen, wenn die verbleibenden Risiken ökonomisch sinnvoll getragen werden können. So auch in dem Anwendungsszenario dieser Arbeit. Für sicherheitskritischere Anwendungsbereiche reicht in *Android* die Sicherheit nicht aus, wie die vorhandenen Schwachstellen zeigen

Ein Kritikpunkt an *Android* ist dabei die mangelnde Unterstützung der Plattform beim Abwehren von Schadsoftware. Durch das unsichere mobile Umfeld oder Malware direkt auf der Plattform kann die Umgebung nicht als sicher eingeschätzt werden.

Weiter ist dabei eine mangelnde Authentifikation des Benutzers am Gerät entscheidend. Durch sie kann durch unberechtigte Dritte das System manipuliert und Schadsoftware zur Ausführung gebracht werden.

In den Mittelpunkt der Sicherheit wird oftmals der Benutzer gerückt, dessen Verhalten von der technischen Seite her zwar beeinflussbar, aber nicht kontrollierbar ist. Insbesondere im B2C-Bereich können keine zwingend einzuhaltenden organisatorischen Richtlinien für die ordnungsgemäße Verwendung der Systeme umgesetzt werden. Es fehlt an den nötigen technischen Maßnahmen, um Benutzer bei sicherheitskritischen Entscheidungen zu unterstützen. Auch wenn sich die Benutzer im direkten Umgang mit der Anwendung korrekt verhalten, kann ein falsches Verhalten beim Umgang mit dem mobilen Gerät fatale Folgen haben.

Der Benutzer hat selber dafür Sorge zu tragen, wer Zugriff auf das Gerät erhält. Sollte das Gerät dennoch in die falschen Hände geraten, so kann *Android* keine ausreichende Sicherheit auf Plattformebene bieten. Die Verantwortung bezüglich der Sicherheit überträgt *Android* damit an den Besitzer des mobilen Gerätes, sowie an die Anwendungsentwickler.

Die Resultate bezüglich *Android‘s* Sicherheit lassen sich anschließend auf andere Bereiche des eCommerce transferiert werden, um auch allgemeinere Aussagen für andere Szenarien treffen zu können. So sind Anwendungen denkbar, die in ähnlichen Szenarien wie dem Präsentierten stattfinden, auch über den Bereich des Mobile Ticketing hinaus.

5.3 Ausblick

Durch eine Authentifizierung des Benutzers bei jedem Zugriff an dem *Android*-Endgerät könnte die Systemsicherheit gesteigert werden, fordert jedoch die Akzeptanz der Bediener der Geräte. Diese werden damit erheblich in der Benutzbarkeit des Systems eingeschränkt. Die Forderung nach einer für den Benutzer transparenten Authentifikationsverfahren an dem Gerät ist daher groß. Die Verwendung von biometrischen Merkmalen zur Erkennung des rechtmäßigen Besitzers könnte durch weiteren technischen Fortschritt in nicht allzu ferner Zukunft Einzug in die Authentifikationsmechanismen mobiler Geräte erhalten. Die Verwendung von Smartcards könnte Teil der Authentifikation bei verschiedenen mobil nutzbaren Diensten des eCommerce werden. Diese Verfahren sind sicher und ausgereift.

Durch eine sichere Plattform für mobile Geräte wären neuartige Anwendungsbereiche möglich, bei denen die Sicherheit bedeutsam ist. Wenn das System tatsächlich sicher ist und von vielen Leuten auch als ein solches angesehen wird, sind Anwendungen vorstellbar, die als Schlüsselbund für Haus oder Auto, als Personalausweis, als Kreditkarte, oder - wie in dem hier gezeigten Szenario - als eine „Fahrkarte samt Fahrkartenautomat“ für den Öffentlichen Personennahverkehr genutzt werden.

Abbildungsverzeichnis

Abbildung 2.1: McCumber Cube - Extended MSR Model [Schou07] 7
Abbildung 2.2: Bedrohungen + Schwachstellen = Gefährdungen [BSI09] 11
Abbildung 2.3: Abrechnung als Mehrwertdienst[Poustt03] 19
Abbildung 2.4: Android Architektur [AndDev09] 21
Abbildung 3.1: Thin-Client Architektur 29
Abbildung 3.2: Anwendungsfälle 32
Abbildung 3.3: Installation der Client-Anwendung 33
Abbildung 3.4: Hauptmenü im Emulator 34
Abbildung 3.5 : Fahrplanauskuft im Emulator 35
Abbildung 3.6: Ticketauswahl 36
Abbildung 3.7: Ticket anzeigen 37
Abbildung 3.8: Kontrolle des Tickets durch den Fahrkartenprüfer 37
Abbildung 3.9: Kundeneinstellungen 38
Abbildung 3.10: *Activities* als Bildschirmmasken 39
Abbildung 3.11: Services als Komponenten des Backend 40
Abbildung 3.12 Programmablauf 41
Abbildung 3.13: Ablauf eines Ticketkaufs 41
Abbildung 3.14: Fahrplandaten 43
Abbildung 3.15: Zahlungsdaten 43
Abbildung 3.16: Nicht-anonymes personenbezogenes Ticket 44
Abbildung 3.17: Anonymes personenbezogenes Ticket 45
Abbildung 4.1: Aufstellen der Sicherheitsanforderungen 47
Abbildung 4.2: Netzplan der eingesetzten Komponenten 48
Abbildung 4.3: Übersicht über verschiedene Bedrohungen 61
Abbildung 4.4: Entwicklung und Realisierung 67
Abbildung 4.5: Zugriffsrechte auf die Informationen 70
Abbildung 5.1: Security Engineering 87

Tabellenverzeichnis

Tabelle 2.1: Drahtlose Kommunikationstechnologien 4
Tabelle 2.2: Allgemeine Sicherheitsziele 7
Tabelle 4.1: In den Anwendungsfällen verarbeitete Informationen 48
Tabelle 4.2: Digitales Ticket 49
Tabelle 4.3: Bedrohungen für Schützenswerte Güter und Schutzziele 52
Tabelle 4.4: Schutzbedarf der schützenswerten Informationen 59
Tabelle 4.5: Schutzbedarf 66
Tabelle 4.6: Umgesetzte Sicherheitsmaßnahmen auf Anwendungsebene 76
Tabelle 4.7: Bewertung der Sicherheitsmaßnahmen 79
Tabelle 10.1: HTC G1 Produktspezifikation [HTC09] 95

Abkürzungsverzeichnis

BSI	Bundesamt für Sicherheit in der Informationstechnik
DVM	Dalvik Virtual Machine
EC	Electronic Commerce
HSPA	High Speed Packet Access
IDS	Intrusion Detection System
IPC	Inter-Process Communication
OHA	*Open Handset Alliance*
ÖPV	Öffentlicher Personenverkehr
MAC	Message Authentication Code
MNO	Mobile Network Operator
MC	Mobile Commerce
MPSP	Mobile Payment Service Provider
PKI	Public-Key Infrastruktur
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UMTS	Universal Mobile Telecommunication System
VDV	Verband Deutscher Verkehrsunternehmen
WPAN	Wireless Personal Area Network

Literaturverzeichnis

[AndDev09] (2009): Android Developers
http://developer.android.com, letzter Zugriff: 21.05.2009

[BDSG09] Justiz, B. d. (2009): Bundesdatenschutzgesetz

[Binay09] Binay, D. (2009): Android Security Improvement

[Bornst09] Bornstein, D. (2008): Google I/O Session Videos and Slides: Dalvik VM Internals
http://sites.google.com/site/io/dalvik-vm-internals, letzter Zugriff: 17.10.2009

[BSI08a] BSI (2008): IT-Grundschutz-Kataloge

[Burns08] Burns, J. (2008): Developing Secure Mobile Applications for Android
An introduction to making secure Android applications

[Chaudh09] Chaudhuri, A. (2009): Language-Based Security on Android

[Chia08] Chia, P. H. (2008): Secure Software Installation via Social Rating

[Eckert08] Eckert, C. (2008): IT-Sicherheit

[Federr09] Federrath, P. D.-I. H. (2009): Sicherheit mobiler Systeme

[Flatle09] Flatley, J. (2007): British Crime Survey

[HTC09] HTC (2009): T-Mobile G1 - Specification
http://www.htc.com/www/product/g1/specification.html, letzter Zugriff: 09.10.2009

[Jurisc09] Dr. Thomas Jurisch, S. W. (2009): Whitepaper: Mobile IT-Sicherheit

[Kleider09] Kleidermacher, D. (2009): The future of mobile devices
http://www2.electronicproducts.com/The_future_of_mobile_devices-article-facn_GREENHILLS_apr2009-html.aspx, letzter Zugriff: 07.09.2009

[Kowals06] Stewart Kowalski, M. G. (2006): Consumers' Awareness of, Attitudes Towards and Adoption of Mobile Phone Security

[Leitne07] Leitner, D. W. (2007): Mobile Ticketing

[Lepsch00] Lepschies, G. (2000): E-Commerce und Hackerschutz

[Lucks02] Lucks, S. (2002): Security Engineering

[Matsun07] Takashi Matsunaka, T. W. a. K. S. (2007): A Lightweight Approach to Protect Mobile Data

[Merz02] Merz, M. (2002): E-Commerce und E-Business

[Mohind08] Mohindra, D. (2008): Android, Incident Response and Forensics

[Mügge07] Mügge, T. (2007): HandyTicket Pilot von VDV-Mitgliedsunternehmen

[Poustt03] Pousttchi, K. (2003): Abrechnung mobiler Mehrwertdienste

[Rigger06] Riggert, P. D. W. (2006): Technologie des Web-Business - Sicherheit und Bezahlen im Internet

[Rosent04] Rosenthal, J.-C. (2004): Sicherheitsprotokolle auf der Transportschicht

[Roth05] Roth, J. (2005): Mobile Computing

[Scharf09] Scharf, D.-B. V. (2009): IT-Sicherheitshandbuch (der Universität Erlangen-Nürnberg) http://www.rrze.uni-erlangen.de/hilfe/security/it-sicherheitshandbuch/, letzter Zugriff: 07.10.2009

[Schmid08] Albayrak, A.-D. S. a. S. (2008): Malicious Software for Smartphones

[Schou07] Corey Schou, D. S. (2007): Information Assurance for the Enterprise

[Stanton04] Jeffrey M. Stanton, K. R. S., Paul Mastrangelo, Jeffrey Jolton (2004): Analysis of end user security behaviors

[Turows04] Klaus Turowski, K. P. (2004): Mobile Commerce - Grundlagen und Techniken

[UN09] United-Nations (1948): Universal Declaration of Human Rights

[Wiedem08] Dietmar Wiedemann, L. G., Key Pousttchi (2008): Ausgestaltung mobiler Bezahlverfahren – Ergebnisse der Studie MP3

[Winkel04] Winkels, P. D. H.-M. (2004): IT-Sicherheit

[Zhang05] Zhang, X. (2005): What Do Consumers Really Know

Anhang

Rechen- und Speicherkapazität	
CPU	Qualcomm® MSM7201A, 528 MHz
Speicher	256 MB ROM 192 MB RAM Speicherslot für microSD Karten
Energie	Lithium-Ionen Akku Kapazität: 1150 mAh Bis zu 350/406 Minuten Gesprächszeit Bis zu 402/319 Stunden im Standby
Mobilität	
Ausmaße	117,7 mm lang, 55,7 mm breit, x 17,1 mm hoch
Gewicht	158 g
Drahtlose Kommunikation	
	Bluetooth 2.0 mit EDR Wi-Fi (IEEE 802.11b/g) HSPA/WCDMA Quad-band GSM/GPRS/EDGE
Benutzerschnittstelle	
Ausmaße	3,2 Zoll TFT-LCD Touchscreen
Auflösung	320 x 480 Pixel (HVGA)
Eingabemöglichkeiten	5 Tasten mit Trackball aufschiebbare 5-Zeilen QWERTZ Tastatur mit 48 Tasten

Tabelle 0.1: HTC G1 Produktspezifikation [HTC09]