



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich



Institute for Pervasive Computing  
Prof. Dr. Friedemann Mattern

# Infrastruktur zum Management von e-Vouchers

30 April, 2006

**David Fuchs**  
fuchsd@student.ethz.ch

Betreuung:  
Christof Roduner



## Abstract

Throughout the world, the time people spend in elevators accumulates to hundreds of years - each day. As most people find the situation in elevators rather discomfoting, public displays enjoy high attention. Passengers prefer to look at the displays rather than their fellow passengers. Because even short elevator trips are likely to add up to an hour per month and passenger, public displays in elevators are a very promising place for advertising companies.

Lucerne-based company lift.tv specialises in context-based elevator "advertainment". Advertisements, together with other content like weather information or stock prices, are displayed during elevator rides. Which advertisements are shown is decided by the context, i.e. information like "What is the location of the building", "What floor does the passenger go to" or even "What are the preferences of the current passenger".

A special form of context-based advertising is the distribution of electronic vouchers. Imagine the following situation: When taking the elevator to your office, the display inside the elevator informs you that a new "SunDime" café has just opened in the same building. Using your mobile phone, your electronic key to the building or other mobile devices, you can receive a gift voucher for a free coffee at "SunDime".

The goal of this semester thesis is the integration of an infrastructure to distribute and validate e-Vouchers into lift.tv's context-based advertising system.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>5</b>
1.1	Werbung im Aufzug . . . . .	5
1.1.1	Über lift.tv . . . . .	5
1.1.2	e-Voucher . . . . .	7
1.2	Terminologie . . . . .	7
1.3	Danksagung . . . . .	8
<b>2</b>	<b>Evaluation verschiedener Technologien</b>	<b>9</b>
2.1	Anforderungen . . . . .	10
2.2	Interaktionsformen . . . . .	11
2.2.1	Visual Codes . . . . .	11
2.2.2	MMS / SMS . . . . .	13
2.2.3	RFID . . . . .	14
2.2.4	NFC . . . . .	16
2.2.5	Weitere Technologien . . . . .	17
2.3	Zusammenfassung . . . . .	18
2.4	Existierende Lösungen . . . . .	18
<b>3</b>	<b>Design</b>	<b>21</b>
3.1	Aufgaben der Architektur . . . . .	21
3.1.1	Anfordern des Vouchers . . . . .	22
3.1.2	Generieren des Voucher-Values . . . . .	22
3.1.3	Verwalten des Vouchers . . . . .	23
3.1.4	Generieren der Antwort . . . . .	23
3.1.5	Einlösen des Vouchers . . . . .	24
3.1.6	Validieren und Buchen des Vouchers . . . . .	24
3.2	Design der Architektur . . . . .	24
<b>4</b>	<b>Implementation</b>	<b>27</b>
4.1	Gespeicherte Daten . . . . .	27
4.2	Infrastruktur des lift.tv Systems . . . . .	29
4.3	Implementation der Services . . . . .	30
4.3.1	CreateVoucherService . . . . .	31
4.3.2	RegisterVoucherService . . . . .	32
4.3.3	ValidateVoucherService . . . . .	32
<b>5</b>	<b>Diskussion und Weiterführende Arbeiten</b>	<b>33</b>
5.1	Nachteile . . . . .	33
5.1.1	Konfiguration und Installation . . . . .	33
5.1.2	Offline-Validierung . . . . .	33
5.2	Weiterführende Arbeiten . . . . .	34

<b>A SQL Table Scripts</b>	<b>36</b>
A.1 Voucher Group Table . . . . .	36
A.2 Voucher Distributed Table . . . . .	36
A.3 Voucher Booked Table . . . . .	36
A.4 Voucher Devices Table . . . . .	36
<b>B Java APIs</b>	<b>37</b>
B.1 CreateVoucherService API . . . . .	37
B.2 RegisterVoucherService API . . . . .	38
B.3 ReplyVoucherService API . . . . .	38
B.4 ValidateVoucherService API . . . . .	38
B.5 ComputeVoucherService API . . . . .	40

# Kapitel 1

## Einführung

### 1.1 Werbung im Aufzug

*Advertising: The science of arresting the human intelligence for long enough to get money from it.* – Stephen Leacock, Canadian Humorist.

Um die “Menschliche Intelligenz zu fesseln”, werden heute alle möglichen Medien eingesetzt. Von Anzeigen in Zeitungen und Zeitschriften bis hin zu Plakaten und Werbebildschirmen ist Werbung allgegenwärtig. Das Ziel einer Werbekampagne ist dabei unabhängig vom Medium stets dasselbe: Die Werbebotschaft einem möglichst grossen Publikum nahezubringen, den grössten Werbeeffekt zum kleinstmöglichen Preis zu erzielen.

Unter allen verwendeten Medien nimmt die Werbung im Aufzug einen Sonderplatz ein. Aufzüge in grossen Gebäuden werden täglich von hunderten von Menschen benutzt, allein das schon macht Aufzüge für Werbefirmen interessant. Hinzu kommt, dass die meisten Fahrgäste die Situation während einer Aufzugsfahrt wegen des beschränkten Platzes und der ungewohnten Nähe zu anderen Passagieren als unangenehm und beengend empfinden. Dadurch kommt Werbeplakaten und Displays in Aufzugskabinen grosse Beachtung zu.

Für die meisten Benutzer eines Aufzuges ist die Fahrt schlicht verlorene, ungenutzte Zeit. Dies bedeutet, dass Werbung während dieser Zeit tendenziell weniger als störend wahrgenommen wird, etwa im Vergleich zu Kinowerbung, während der man lieber bereits den Film sehen würde. Die “richtige” Werbung, möglicherweise kombiniert mit zusätzlicher Information oder Unterhaltung, kann für den Fahrgast sogar einen Zusatznutzen bedeuten.

#### 1.1.1 Über lift.tv

Die Firma lift.tv<sup>1</sup> in Luzern ist eine junges Unternehmen, welches Aufzugsfirmen und Gebäudemanagern ein elektronisches Werbesystem für Lifte anbietet. In den Aufzugskabinen werden dabei Touchscreens installiert (“POF”), welche die Darstellung von Werbung, aber auch Informationen wie Wetterprognosen oder Börsenkursen und Unterhaltung wie Filmen oder Musik erlauben. lift.tv stellt dabei sowohl die erforderlichen, interaktiven Touch-Screen-Geräte und die Infrastruktur zur Versorgung mit Content wie auch den Content selbst zur Verfügung. Das Ziel dabei ist es, die Aufzugsfahrt für den Passagier so kurzweilig wie möglich zu gestalten. Die Attraktivität für Gebäudebesitzer und Aufzugshersteller liegt einerseits in einer finanziellen Beteiligung an den Werbeeinnahmen, andererseits im High-Tech-Image, welches eine solche Lösung ausstrahlt.

---

<sup>1</sup><http://www.lift.tv/>



Abbildung 1.1: Das POF-Gerät von lift.tv

Werbung im Aufzug ist nicht kein neues Konzept. Mehrere Werbefirmen haben sich auf Aufzugswerbung spezialisiert, sowohl in elektronischer wie in plakativer Form. Allerdings sind bisherige Werbekampagnen nicht interaktiv und nehmen auch die zur Verfügung stehenden Kontextinformationen über Fahrgäste nicht in Bezug. Beispielsweise werden Werbe-Spots nach einer fix vorgegebenen Playlist wieder und wieder abgespielt. Im Gegensatz dazu werden beim lift.tv-System Kontextinformationen in die Auswahl des Contents miteinbezogen [THM04]. Dabei ist der Kontext durch Kenntnis über das Gebäude und seine Umgebung sehr genau definiert. Durch Informationen über das Gebäude, die Tageszeit und das Zielstockwerk können viele Rückschlüsse auf die Fahrgäste gezogen werden. Konkret kann etwa unterschieden werden zwischen dem Versicherungsangestellten, der morgens zur Arbeit geht, und dem Kunden eines Ladens in einer anderen Etage. Der Versicherungsangestellte kann dann auf ein nahegelegenes Restaurant aufmerksam gemacht werden, während dem Kunden die gegenwärtigen Top-Angebote des Ladens gezeigt werden. Messwerte beliebiger Sensoren können ebenfalls zur Kontextbestimmung hinzugezogen werden. Wenn die Wettersensoren nasskaltes Wetter melden, sollte die Werbeinformation des Eisherstellers nicht gezeigt werden. Darüberhinaus bietet die Möglichkeit der Interaktion dem Aufzugspassagier die Gelegenheit, auf für ihn interessante Informationen auch zu reagieren. Somit kann das System noch besser entscheiden, welche Information in einer gegebenen Situation die "richtige" ist. Dies kann soweit gehen, dass regelmässigen Benutzern des Aufzuges ein personalisiertes Portal zur Verfügung gestellt wird. In vielen Gebäuden kann der Aufzug seinen Fahrgast nämlich aufgrund des von ihm verwendeten "elektronischen Schlüssels" (SmartCard) für den Gebäudezugang "identifizieren".

Die Berücksichtigung des Kontextes bei der Auswahl der Information und Werbung kommt sowohl den Fahrgästen wie auch denjenigen Firmen zugute, welche in Werbung auf dem System investieren. Die Fahrgäste werden nicht mit für sie uninteressanten Dokumenten genervt. Den Werbern ermöglicht das System, einen vorgegebenen Werbedruck auf ein genau definiertes Zielpublikum auszuüben und dabei den Streuverlust zu minimieren. Zusammen mit der hohen Anzahl Kontakte und der grossen Aufmerksamkeit für Werbung in Aufzügen ergibt sich eine äusserst attraktive Werbeplattform.

### 1.1.2 e-Voucher

Eine spezielle Form der Interaktion mit dem Fahrgast sind e-Voucher. Dabei handelt es sich um elektronische Voucher oder Gutscheine, die ein Benutzer des Aufzuges über das lift.tv-System beziehen kann und die zu einer bestimmten Dienstleistung oder Ware beim Aussteller des Vouchers berechtigt. e-Voucher sind also zu Vergleichen mit der Vielzahl von Rabatt- oder Geschenkgutscheinen, die man in Zeitungen oder Zeitschriften ausschneiden kann oder auf der Strasse in die Hand gedrückt bekommt. Im Gegensatz zu diesen existieren die e-Voucher allerdings nur in elektronischer Form, was die Handhabung für Aussteller wie Benutzer erleichtert.

Ein mögliches Szenario sieht folgendermassen aus. Die Kaffeehauskette “SunDime” eröffnet eine neue Filiale in einem Hochhaus. Um die im Gebäude arbeitenden Angestellten auf das neue Angebot aufmerksam zu machen, werden im Lift kurze Werbespots geschaltet. Zusätzlich besteht die Möglichkeit, während der Liftfahrt einen elektronischen Voucher für einen Kaffee zu erhalten, indem man sein Mobiltelefon an das POF-Gerät hält. Da “SunDime” vor allem die im Gebäude arbeitenden Menschen als Zielpublikum für das neue Café sieht, werden Voucher nur an diejenigen Fahrgäste verteilt, die in ein Stockwerk mit Büros fahren. Um die Kosten der Aktion im Griff zu haben, wird maximal eine zuvor festgelegte Anzahl Voucher verteilt. Um möglichst schnell neue Kunden zu gewinnen, ist die Gültigkeit der Voucher auf eine Woche nach Eröffnung beschränkt. Am Ende der Woche hat “SunDime” mit der Gutscheinaktion viele Leute von der erstklassigen Qualität seines Kaffees überzeugen können. Da die Voucher beim einlösen ebenfalls elektronisch erfasst werden, kann “SunDime” den Erfolg der Aktion sofort und auf einfache Weise überprüfen.

Das Ziel der vorliegenden Arbeit besteht aus der Konzeption und Implementation einer e-Voucher-Infrastruktur innerhalb des lift.tv-Systems. Dies umfasst das Generieren, Verteilen, Einsammeln und Validieren der Voucher. Interessierte Firmen können über die Infrastruktur Voucher an bestimmte Zielgruppen abgeben und dabei festlegen, wieviele Voucher verteilt werden und über welchen Zeitraum hinweg diese gültig sein sollen. Löst ein Kunde einen Voucher ein, überprüft die Infrastruktur dessen Gültigkeit anhand einer Datenbank mit Voucher-Informationen.

In einem ersten Schritt werden die Anforderungen an das System festgelegt und mögliche Interaktionsformen untersucht. Dabei werden verschiedene Mechanismen zum Transport und zur Speicherung der Voucher untersucht, etwa SMS, MMS, RFID, oder “Visual Codes”. Danach wird die Implementation des Systems sowie die Integration in die bestehende Umgebung von lift.tv beschrieben.

## 1.2 Terminologie

**Content.** Sämtliche Inhalte, welche über das POF verbreitet werden. Dazu gehören Bilder, Text, Filme und Musik. Der Content ist typischerweise ein Mix aus Werbung und Information (“Advertainment”), um sowohl dem Liftpassagier einen Nutzen bieten zu können als auch den Einsatz für den Betreiber finanziell interessant zu machen.

**Device ID.** Die für die Interaktion mit dem System verwendeten mobilen Devices haben typischerweise eine (dem System nicht notwendigerweise bekannte!) ID, mit der sie identifiziert werden können. Für Mobiltelefone ist dies beispielsweise die Telefonnummer, für RFID-Tags die ID-Nummer des Tags. Die Device ID kann einerseits direkt als Voucher-Value verwendet werden, liefert aber darüberhinaus zusätzliche Information. Weil die Device ID indirekt den Kunden identifiziert, kann damit beispielsweise verfolgt werden, wieviele Voucher ein Kunde durchschnittlich bezieht oder auch die maximal zulässige Anzahl Voucher pro Kunde beschränkt werden.



**Gateway.** Die “Steuereinheit” für die POFs. Typischerweise ist pro Gebäude ein Gateway installiert, von dem die POFs ihren Content beziehen.

**MMS.** Multimedia Messaging Service. Ermöglicht wie SMS das Senden von Mitteilungen auf Mobiltelefone, im Gegensatz zu SMS ist MMS aber nicht auf reine Textmeldungen beschränkt, sondern kann Bilder, Filme oder beliebige andere Dateiformate übertragen.

**Mobiles Device.** Das Gerät, mit welchem ein Liftpassagier die Voucher bezieht. Je nach Interaktionsform kann es sich dabei um Mobiltelefone, Smart Cards, RFID-Tags o.Ä. handeln.

**NFC.** Near Field Communication. Von Sony und Philips gemeinsam entwickeltes System für die Datenübertragung zwischen Geräten über kürzeste Distanzen (einige cm) und mit hohen Übertragungsraten.

**POF.** Point Of Functionality. Das Touch-Screen-Gerät, welches in der Liftkabine montiert ist und über das die Interaktion mit dem Passagier stattfindet.

**POS.** Point Of Sale. Der Verkaufsort des Ausstellers des Gutscheins. Hier wird der Gutschein über entsprechende Hard- und Software wieder eingesammelt.

**RFID.** Radio Frequency Identification. Diese Technik ermöglicht es, beliebige Gegenstände mit sogenannten “Tags” zu versehen, die eine Identifikationsnummer enthalten. Die Tags können mit Lesegeräten über kurze Distanzen (typischerweise einige cm bis einige m) gelesen und bei einigen Systemen auch beschrieben werden. RFID-Tags sind oft in Schlüsselanhängern oder Karten anzutreffen, die den Zugang zu einem Gebäude regeln.

**SMS.** Short Messaging Service. Ein Dienst, der das Senden von Textmitteilungen mit bis zu 160 Zeichen auf Mobiltelefone ermöglicht.

**Visual Codes.** Eine Art zweidimensionaler Barcodes, die an Objekten angebracht werden können. Analog zur RFID-Technik erlauben Visual Codes die Identifikation von Gegenständen, die Kommunikation findet bei dieser Variante aber optisch statt, die Codes werden mittels einer Kamera und spezieller Software erkannt.

**Voucher.** Ein elektronischer Gutschein, der vom Passagier mittels Interaktion mit dem lift.tv-System bezogen werden kann. Der Gutschein berechtigt zu einer vom Werbekunden definierten Leistung. Beim Einlösen des Gutscheins wird dieser wiederum über das lift.tv-System validiert und gebucht, d.h. als “gebraucht” markiert.

**Voucher-Value.** Ein Wert, mit dem jeder Voucher im System identifiziert wird.

### 1.3 Danksagung

Ich bedanke mich bei Christof Roduner für die Betreuung dieser Arbeit, für die zahlreichen kreativen Inputs und für die grossen Freiheiten, die ich beim Schreiben der Arbeit genoss. Weiter bedanke ich mich bei Andreas Westhoff und Thomas Meyer von der Firma lift.tv für die Möglichkeit dieser interessanten und praxisnahen Arbeit und für die gute Zusammenarbeit.

“Semesterarbeiten” heissen so, weil sie üblicherweise im Verlaufe eines Semesters bearbeitet und fertiggestellt werden. Die vorliegende Arbeit nahm ihren Anfang als Semesterarbeit; die Semester zogen ins Lande, aus der Semesterarbeit wurde eine Semesterferienarbeit und aus der Semesterferienarbeit wieder eine Semesterarbeit... ein Dankeschön an alle Beteiligten deshalb auch für ihre grosse Geduld.

## Kapitel 2

# Evaluation verschiedener Technologien

Dieses Kapitel beschäftigt sich damit, wie ein e-Voucher beziehungsweise der ihn identifizierende Voucher-Value zum Kunden gelangt und wie er später vom Kunden eingelöst werden kann. Der Voucher-Value ist dabei ein von der Infrastruktur generierter Wert, der später eindeutig wieder identifiziert werden kann. Auf das Generieren der Voucher-Values wird in Kapitel 3 näher eingegangen. Als Transportmechanismus des Vouchers zum Kunden ist der Einsatz verschiedener Technologien denkbar. Die zu erfüllende Bedingung ist: Beim Einlösen des Vouchers muss dieser eindeutig identifiziert und aufgrund der Datenbank des Gateways validiert werden können. Je nach verwendetem Transportmechanismus ergeben sich andere Anforderungen an die technische Ausrüstung der Benutzer, vor allem aber andere Interaktionsformen. Wie ein Fahrgast einen Voucher erhält und später wieder einlöst, hängt also vom verwendeten Transportmechanismus ab.

Die Aufgabenstellung der Semesterarbeit sah als Transportmechanismus lediglich die in Abschnitt 2.2.1 diskutierten Visual Codes vor. Während der Erarbeitung des Verteilungskonzeptes wurde aber schnell klar, dass eine auf mehrere Technologien und Interaktionsformen anpassbare Architektur von Vorteil sein würde. In den POF-Screens ist ein RFID-Reader (siehe Abschnitt 2.2.3) bereits integriert, und in Umgebungen, wo RFID bereits für andere Zwecke genutzt wird (beispielsweise um den Gebäudezugang zu regeln), macht es sicherlich Sinn, die bestehenden technischen Möglichkeiten auszuschöpfen. Zudem ist es auch von Vorteil, in verschiedenen Installationen die jeweils verwendete Verteilmethode den Anforderungen und Fähigkeiten der Benutzer anzupassen. Die Verbreitung von Geräten mit neuen Technologien und die Bereitschaft, sich mit diesen Technologien auch auseinandersetzen, dürfte etwa in Technoparks wie dem D4 Business Center in Luzern grösser sein als in Bürogebäuden.

In Abschnitt 2.1 werden Anforderungen definiert, welche von allen möglichen Interaktionsformen erfüllt werden sollten. Die im Rahmen der Arbeit untersuchten möglichen Interaktionsformen sind in Abschnitt 2.2 vorgestellt. Es wird jeweils kurz diskutiert, welche der Anforderungen gut und welche nur ungenügend erfüllt werden. Alle Interaktionsformen erlauben es, von einem mobilen Gerät eine eindeutige ID auszulesen oder darauf zu speichern. Allerdings ist das Speichern von Werten nicht auf jedem Gerät möglich, was bei der Ausarbeitung der Logik beachtet werden muss.

## 2.1 Anforderungen

In Zusammenarbeit mit dem Betreuer der Arbeit, Christof Roduner, und lift.tv habe ich folgende Anforderungen an eine e-Voucher-Infrastruktur identifiziert. Die Punkte sind entsprechend ihrer Wichtigkeit absteigend geordnet.

**Benutzerfreundlichkeit.** Die Lösung muss für den Benutzer einfach handhabbar sein. Das Erhalten und Einlösen von Vouchern sollte ein schneller und komfortabler Prozess sein. Dies umfasst folgende Punkte:

- **Benutzereingaben.**  
Keine oder nur wenige Eingaben des Benutzers sollten nötig sein, um einen e-Voucher zu erhalten und einzulösen. Der Prozess sollte so wenige Schritte als möglich beinhalten. Im Idealfall hält ein Benutzer ein Gerät (beispielsweise ein Mobiltelefon oder einen RFID-Tag) an ein im POF integriertes Lesegerät und erhält dadurch, ohne weitere Interaktion, einen Voucher.
- **Software-Installation.**  
Falls programmierbare Geräte für die Interaktion mit dem Voucher-System verwendet werden (etwa Mobiltelefone mit Java), ist eine Lösung vorzuziehen, bei der keine zusätzliche Software auf das Gerät geladen werden muss. Zusätzliche Software kann Probleme mit der Kompatibilität verursachen und verhindert eine spontane Nutzung des Dienstes. Darüberhinaus werden unerfahrene Nutzer davon abgeschreckt, den Dienst zu benutzen. Für Installationen in Umgebungen mit technisch versierten Benutzern, etwa dem D4 Business Center wo lift.tv zu Hause ist, sind diese Nachteile allerdings annehmbar.
- **Geschwindigkeit.**  
Das Verteilen und Einsammeln der Voucher sollte in vernünftiger Zeit machbar sein. Das Lesen (und nötigenfalls Schreiben) eines Gerätes durch das POF sollte nicht länger als einige Sekunden dauern.

**Deployment.** Nachfolgend sind die Anforderungen für die Installation von Hard- und Software an den POS aufgeführt.

- **Bescheidene Hardware-Anforderungen.**  
Eine Lösung zum Einsammeln der Gutscheine sollte auch auf Geräten mit beschränkter Prozessorleistung und Memory ausführbar sein. Dadurch kann am POS sowohl ein bestehender PC, wie auch ein weniger leistungsfähiges Gerät wie ein PDA eingesetzt werden.
- **Hardware-Kosten.**  
Damit das Verteilen von Gutscheinen wirtschaftlich attraktiv ist, sollten für die beteiligten Firmen nur moderate Hardwarekosten anfallen. Ein mögliches Szenario ist hier, dass die erforderliche Hardware mit vorinstallierter Software von lift.tv leihweise zur Verfügung gestellt wird.
- **Einfache Benutzbarkeit.**  
Das Einlösen der Gutscheine sollte für das Verkaufspersonal eine einfache und schnell zu erledigende Aufgabe sein.
- **Platzsparende Hardware.**  
Zusätzliche an den POS benötigte Geräte (z.B. Lesegeräte) sollten nicht allzu gross oder schwer ein.

- **Offline-Benutzung.**  
Vorziehen wäre eine Lösung, bei der keine persistente Netzwerkverbindung zwischen dem POS und der lift.tv-Infrastruktur erforderlich ist. Eine solche Lösung wäre weniger anfällig gegenüber Netzwerkproblemen und würde unter Umständen auch Verbindungsgebühren sparen. Allerdings kann bei einer solchen Lösung Double-Spending der Gutscheine nicht verhindert werden.

**Sicherheit und Privacy.** Da die Voucher keine sensitiven persönlichen Daten wie Kreditkartennummern beinhalten und ihr Wert beschränkt ist, ist Sicherheit nicht die erste Priorität. Trotzdem sollten folgende Punkte erfüllt sein:

- **Benutzer-Vertrauen.**  
Damit die Fahrgäste die Möglichkeit, Voucher zu erhalten, auch Nutzen, ist ein gewisses Vertrauen in das System notwendig. Konkret soll einem Benutzer zu jeder Zeit bekannt sein, welche Daten über ihn durch das System wann gelesen werden, wo und wie lange diese gespeichert werden, wem sie zugänglich sind sowie welche Daten allenfalls auf seinem Gerät gespeichert werden.
- **Fälschen oder Stehlen der Voucher.**  
Das Fälschen eines Vouchers sollte nicht auf triviale Weise möglich sein. Allerdings müssen die Voucher nicht absolut “fraud resistant” sein, da kaum jemand grossen Aufwand betreiben wird, um einen Voucher mit geringem Wert zu fälschen.  
Die Fälschungssicherheit hängt ab vom Verfahren, mit welchem der Voucher generiert wird, aber auch von seiner Speicherung. Das Stehlen von Vouchern anderer Benutzer soll nicht einfach möglich sein.
- **Double Spending.**  
Das mehrfache Ausgeben eines erhaltenen Vouchers soll nicht möglich sein.

## 2.2 Interaktionsformen

In den Folgenden Abschnitten 2.2.1 bis 2.2.5 sind die untersuchten Interaktionsformen erläutert. Jeder Abschnitt besteht aus einer Übersicht über die jeweilige Technologie für Transport und Speicherung des Voucher-Values, einer Beschreibung der damit möglichen Interaktionsform und eventuell Varianten davon, sowie einer Diskussion der sich daraus ergebenden Vor- und Nachteile.

Abschnitt 2.3 fasst tabellarisch die Beurteilung der verschiedenen Interaktionsformen im Bezug auf die in 2.1 aufgestellten Anforderungen zusammen. Der letzte Abschnitt dieses Kapitels, 2.4, gibt schliesslich einen kurzen Überblick über sich bereits auf dem Markt befindliche Lösungen zur Verteilung von Tickets, ein dem Verteilen von Vouchern verwandtes Problem.

### 2.2.1 Visual Codes

Bei den vom Institute for Pervasive Computing der ETH Zürich entwickelten Visual Codes [VCODE] handelt es sich um eine Art zweidimensionaler Barcodes. Die optischen Codes werden an physikalischen Objekten angebracht oder auf Bildschirmen dargestellt und können von handelsüblichen Digitalkameras auch mit beschränkter Auflösung gelesen werden, wie sie etwa in Mobiltelefonen verwendet werden. Mit entsprechender Software ausgestattete Geräte können dadurch mit Objekten interagieren, die über Visual Codes verfügen. Die Erkennungssoftware erkennt auch Bewegungen des Gerätes, die dann als Benutzereingaben interpretiert und weiterverarbeitet werden können. Da Mobiltelefone mit eingebauten Digitalkameras und

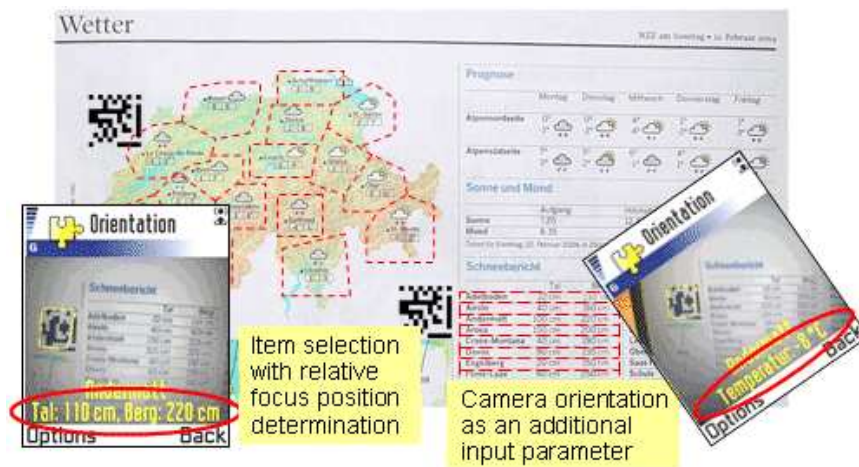


Abbildung 2.1: Beispiel einer Anwendung von Visual Codes: Wetterinformationen in Echtzeit.

leistungsstarken Mikroprozessoren immer populärer werden, ergibt sich eine grosse Zahl interessanter Anwendungsmöglichkeiten von Visual Codes und der entsprechenden Erkennungssoftware. Bestehende Anwendungen umfassen etwa den Zugriff in Echtzeit auf Wetterinformation durch Visual Codes auf der Wetterberichtseite einer Zeitung oder Fahrplaninformationen an Tram- und Bushaltestellen.

**Interaktion (mit Visual Code Software).** Um diese Lösung nutzen zu können, muss der Benutzer vorgängig ein entsprechendes Software-Applet auf sein Mobiltelefon installieren. Dieses müsste beispielsweise beim POF-Betreiber oder beim Verteiler der Voucher heruntergeladen werden. Wenn ein Kunde am POF einen Voucher anfordert, wird der generierte Voucher-Value als Visual Code auf dem POF angezeigt. Der Kunde fotografiert den Visual Code mit der Kamera seines Mobiltelefones; eine Erkennungssoftware wandelt das Bild danach in einen String um und speichert diesen. Um den Voucher einzulösen, wandelt die Software des Mobiltelefones den String wiederum in einen Visual Code um und zeigt diesen auf dem Display an. Das POS mit angeschlossener Kamera erkennt den Visual Code und validiert ihn über den Gateway mit der Datenbank.

Beim Darstellen des Vouchers durch das Mobile Device könnte auch noch zusätzliche Information zum Voucher-Value hinzugefügt werden, beispielsweise eine Kunden-ID, die in der Software enthalten ist und die Anzahl Voucher pro Kunde limitiert. Dieses Verfahren ist noch immer vollständig anonym, falls der Benutzer beim Downloaden der Software nirgends seine Personalien angeben muss.

Die Verwendung von Software auf dem Mobiltelefon erlaubt es auch, den Komfort für den Benutzer weiter zu steigern. Beispielsweise könnte die Software auch alle gesammelten Voucher verwalten. Der POS wird ebenfalls mit einem Visual Code gekennzeichnet, welcher von der Applikation erkannt wird. Die Applikation kann dann entscheiden, ob für diesen POS (genauer: für diesen Anbieter von Vouchern) Voucher auf dem Gerät gespeichert sind und gleich den richtigen präsentieren. Der Benutzer könnte dann die Auswahl etwa durch Schwenken des Gerätes bestätigen. Auch eine Erinnerungsfunktion vor dem Ablauf der Gültigkeitsdauer und das automatische Löschen gebrauchter und abgelaufener Voucher wären denkbar.

**Interaktion (ohne Software).** Eine Variante der obigen Interaktionsform ist die, dass der Benutzer den am POF angezeigten Visual Code lediglich fotografiert,

schliesslich enthält dieser ja alle benötigte Information. Das vorgängige Herunterladen und Installieren eines Software-Applets auf dem Mobiltelefon entfällt bei dieser Variante. Dem POS präsentiert der Benutzer dann die Fotografie des Visual Code.

**Beurteilung.** Bei der Interaktion ohne zusätzliche Software entfallen das Herunterladen und Installieren eines Applets, der Dienst kann spontan genutzt werden. Einzige Voraussetzung ist der Besitz eines Mobiltelefones mit eingebauter Kamera, was heute bei praktisch allen Modellen der Fall ist. Wegen der beschränkten Auflösung sowohl der beiden Kameras wie auch des Telefondisplays wird eine solche Lösung leider nur sehr unzuverlässig funktionieren. Auf die Verwendung von Software auf dem Mobiltelefon des Benutzers kann bei dieser Interaktionsform daher wohl kaum verzichtet werden.

Die Verwendung eines Software-Applets macht diese Lösung enorm flexibel. Die Software kann auf den Herausgeber des Vouchers angepasst werden und noch zusätzliche Informationen, etwa eine Web-Adresse, enthalten. Informationen wie Gültigkeitsdauer des Vouchers können dem Benutzer vom Applet angezeigt werden. Da die Übertragung des Vouchers auf optischem Weg geschieht, muss ein Benutzer seine Telefonnummer oder ähnliche persönliche Angaben nicht preisgeben; dies fördert sicherlich die Akzeptanz des Dienstes. Allerdings muss der Benutzer auch darauf vertrauen, dass das benötigte Software-Applet wirklich nur die Verwaltung der Voucher übernimmt und nicht etwa hinter seinem Rücken noch persönliche Informationen sammelt und diese für Werbezwecke übermittelt. (Beispiele von PC-Programmen, die genau das tun, sind ja sattem bekannt.)

Für die Nutzung ist ein Mobiltelefon mit Kamera und die vorherige Installation des Applets nötig. Dies dürfte nicht wenige potentielle Nutzer des Dienstes von vornherein ausschliessen. Im Weiteren ist eine spontane Nutzung des Dienstes wegen des zuvor notwendigen Software-Downloads nicht möglich, und technisch nicht versierte Benutzer werden sich die Installation der Software auch bei entsprechend ausgestattetem Mobiltelefon kaum zutrauen. Damit ist die Verteilung von Vouchern via Visual Codes nur kleineren Umgebungen mit technisch versierten Benutzern sinnvoll.

### 2.2.2 MMS / SMS

Eine weitere Möglichkeit, das Mobiltelefon eines Benutzers zur Speicherung der Voucher zu verwenden, ist das Zusenden des Voucher-Values über das GSM-Netz als MMS oder SMS.

**Interaktion.** Der Benutzer gibt am POF seine Mobiltelefonnummer ein und erhält danach den generierten Voucher-Value zugesendet. Die Darstellung des Vouchers sollte falls möglich in einer Form erfolgen, die vom POF einfach erkannt werden kann; hierfür bieten sich wiederum die oben beschriebenen Visual Codes oder ähnliche optische Darstellungen an. Das hier diskutierte Szenario unterscheidet sich von 2.2.1 also nur dadurch, dass der Code nicht fotografiert, erkannt und dann wiederum als Bild dargestellt wird, sondern direkt als Bild auf das Gerät gesendet wird.

Andere Formen der Darstellung sind ebenfalls denkbar. Es könnten auch Barcodes verwendet werden oder sogar rein textbasierte SMS. Bei einer textbasierten Übertragung müsste der Kunde wohl den übertragenen Voucher-Value beim Einlösen am POS von Hand eingeben. Das korrekte und zuverlässige Erkennen des Wertes mittels einer Kamera und OCR-Software ist aufgrund der Auflösungen von Kamera und Display wahrscheinlich nicht möglich.

Eine Variante dieser Lösung wäre der Verzicht auf eine Benutzereingabe direkt über das POF und stattdessen das Einblenden einer Nummer. Voucher können

erhalten werden durch das Senden einer SMS auf die eingeblendete Nummer. Diese Variante könnte in Umgebungen verwendet werden, wo keine direkte Interaktion mit dem Display möglich ist, beispielsweise bei den oft in öffentlichen Verkehrsmitteln anzutreffenden Werbedisplays.

**Beurteilung.** Der Vorteil gegenüber dem in 2.2.1 vorgestellten Ansatz ist die Unabhängigkeit von Software, die auf dem Mobiltelefon installiert ist. Fast jedes neuere Mobiltelefon ist in der Lage, Bildnachrichten zu empfangen und darzustellen. Visual Codes oder ähnliche “2D-Barcodes” können auch bei schlechten Lichtverhältnissen und auf verschiedenen grossen Displays sehr zuverlässig vom POS erkannt werden.

Benutzer müssen hier ihre Telefonnummer auf dem POF eintippen. Dies macht die Lösung etwas weniger benutzerfreundlich als das reine Abfotografieren des Vouchers, aber auch das Eintippen der eigenen Telefonnummer ist ein nicht allzu grosser Aufwand und dem Benutzer zumutbar. Das grössere Problem in diesem Zusammenhang ist die Preisgabe persönlicher Information (Telefonnummer) durch den Fahrgast. Haben die Benutzer kein Vertrauen in das System, etwa weil sie befürchten, dass die Nummer danach für Werbezwecke missbraucht wird, könnten sie zögern, den Dienst zu nutzen. Untersuchungen zeigen allerdings, dass Konsumenten schnell bereit sind, private Daten preiszugeben, wenn sie dafür eine Gegenleistung erhalten.<sup>1</sup>

Zu beachten ist weiter, dass bei dieser Lösung dem Betreiber zusätzliche Kosten entstehen, die der Mobilnetzbetreiber für das versenden der MMS-Nachrichten kassiert. Das Verwenden von Vouchern würde damit teurer und für Werber unter Umständen wirtschaftlich weniger attraktiv.

### 2.2.3 RFID

Der Überbegriff Radio Frequency Identification steht für eine Menge existierender Standards, die die drahtlose Identifikation von Objekten mittels RFID-Tags und Lesegeräten ermöglichen. Die Datenübertragung geschieht je nach Standard elektromagnetisch oder über induktive Felder, die Reichweite beträgt je nach Standard und Ausführung zwischen wenigen Zentimetern und einigen Metern. Es existiert eine Vielzahl verschiedenartiger Tags; meist enthält ein Tag lediglich eine fixe Identifikationsnummer, es gibt aber auch Tags mit schreibbarem Speicher, Mikroprozessoren oder integrierten Sicherheitsfunktionen (Authentifikation mit dem Lesegerät). Daneben unterscheiden sich die heute erhältlichen RFID-Systeme auch im Preis, der Datenübertragungsrate beim Lesen und Schreiben, der Grösse und dem Gewicht der Tags, der Störanfälligkeit und der Erkennungsrate (d.h. wieviele der Tags vom Leser pro Sekunde gelesen werden können). Die Tags sind fast immer passiv, benötigen also keine Batterie oder sonstige Energieversorgung.

Die am weitesten verbreiteten Systeme entsprechen den folgenden Standards:

- ISO 15693. Geräte dieses Standards arbeiten induktiv auf dem 13.56MHz-Frequenzband und haben eine Reichweite von bis zu 1.5m. Die Datenübertragungsrate beträgt typischerweise 5-25 kbit/s.  
Das POF-Gerät von lift.tv besitzt ein ISO-15693-kompatibles RFID-Lesegerät.
- ISO 14443. Ebenfalls induktiv auf 13.56MHz, allerdings mit wesentlich geringerer Reichweite als ISO 15693 (bis 20 cm). Die Datenübertragungsrate ist mit typischerweise 50-100 kbit/s höher als bei ISO 15693. NFC-Geräte entsprechen diesem Standard.

---

<sup>1</sup>So waren in einer im Zusammenhang mit der “Infosecurity Europe 2004” in London durchgeführten Befragung 71% der Computernutzer bereit, ihr Computer-Passwort zu verraten, wenn sie dafür einen Schokoladeriegel(!) erhielten.

Durch die immer kostengünstigere Fertigung der Tags erlebt die RFID-Technik seit einigen Jahren geradezu einen Boom. Die RFID-Technik wird seit vielen Jahren etwa bei Anlagen zur Verhinderung von Ladendiebstählen angewendet. Durch die stetig sinkenden Kosten pro Tag, heute typischerweise unter 1\$, werden sie vermehrt auch für Güterüberwachung, Lagerbewirtschaftung und Supply Chain Management eingesetzt. Auch für die Authentifikation von Personen (zum Beispiel Zutrittskontrolle zu Gebäuden) hat sich RFID vielerorts bewährt. So besitzen in vielen Gebäuden zutrittsberechtigte Personen Karten oder Schlüsselanhänger, die (zusammen mit einem PIN) den Zutritt gewähren.

**Interaktion (Read-Only).** Der POF-Screen enthält ein RFID-Lesegerät gemäss dem ISO-15693-Standard mit einer Reichweite von wenigen 10 cm. Um einen Voucher zu erhalten, hält ein Benutzer sein RFID-Tag an den RFID-Leser des POF, und die eindeutige ID des Gerätes wird in der Datenbank des Gateway abgelegt. Da es hierbei nicht möglich ist, Daten auf das RFID-Tag zu schreiben, kann hier kein systemgenerierter Voucher-Value verwendet werden; lediglich die Tag-ID ermöglicht die spätere Wiederidentifikation des Benutzers.

**Interaktion (Read/Write).** In diesem Fall kann das Lesegerät nicht nur die Tag-ID auslesen, sondern auch Daten auf dem Tag speichern. Analog zum obigen Fall wird zuerst die ID des Tags gelesen; danach kann aber wie bei den in den vorherigen Abschnitten besprochenen Technologien ein systemgenerierter Wert auf dem mobilen Device gespeichert werden.

**Beurteilung.** ISO-15693-Geräte arbeiten auf dem 15.56-MHz-Frequenzband und sind sehr weit verbreitet. Ein Einsatz von RFID wäre in Umgebungen sinnvoll, wo Benutzer bereits über RFID-Tags verfügen. Die Verteilung über RFID macht kaum Sinn in Umgebungen, wo Benutzer (zumindest teilweise) nicht schon über entsprechende Tags verfügen; die meisten Benutzer wären kaum willens, ein zusätzliches Device mit sich herumzutragen, dessen einzige Funktionalität das Beziehen von Vouchern ist.

Ein Vorteil der oben beschriebenen Interaktionsform ist ihre grosse Benutzerfreundlichkeit. Es ist keine einzige Benutzereingabe notwendig; lediglich der Zutrittsbadge für das Gebäude (oder ein ähnlicher Gegenstand mit RFID-Tag) muss in die Nähe des Lesegerätes gebracht werden, das spätere Einlösen des Vouchers ist genauso einfach.

Leider hat die RFID-Technik in den letzten Jahren sehr viel negative Berichterstattung in der Presse erhalten. Die Verunsicherung über die Funktion von RFID ist gross, und nicht wenige Leute halten RFID-Tags für "Spy-Chips", die eine lückenlose Überwachung des Trägers ermöglichen. Tatsächlich ist der Schutz der Privatsphäre ein Thema, das auch bei der RFID-Technologie berücksichtigt werden muss. Bei der Verwendung von RFID-Identifikationsnummern als Voucher ergeben sich aber keine Probleme, da das Beziehen eines Vouchers völlig anonym geschehen kann. Lediglich wann mit einem bestimmten Tag ein Voucher erhalten wurde und wann und wo dieser wieder eingelöst wurde, wird gespeichert. Dies sollte auch den Benutzern des Dienstes klar ersichtlich sein, damit diesem auch vertraut wird. Wenn nur bereits vorhandene RFID-Devices verwendet werden, sollte dies das Vertrauen der Benutzer ebenfalls positiv beeinflussen.

Beim Einsatz von read-only Tags kann ein Benutzer wegen der fehlenden Möglichkeit, Daten auf dem RFID-Device zu speichern, einen Voucher nicht "ansetzen" und sieht somit beispielsweise auch nicht, wann dessen Gültigkeitsdauer abläuft. Probleme macht der fehlende Speicher auch in einem anderen Fall: Wenn zwei Benutzer verschiedene Tags mit derselben Identifikationsnummer besitzen. Verwenden



alle Nutzer Tags einer bestehenden RFID-Lösung, sollte dieser Fall aber nicht auftreten. Die fehlende Möglichkeit zur Datenspeicherung auf dem mobilen Device ist ebenfalls ein Problem, falls derselbe Kunde (mit demselben Device) auch mehrere Voucher beziehen können soll. In diesem Fall können mehrere Voucher mit zwangsweise demselben Value existieren. Will der Kunde nun einen der Voucher einlösen, ist nicht offensichtlich, welcher der vorhandenen Voucher gebucht (d.h. als gebraucht markiert) werden soll. Die Lösung dieses Problems ergibt sich aus der Beobachtung, dass das Werte-Tupel (Voucher-Value=Device-ID,Erstellungszeitpunkt des Vouchers) einmalig ist, da ein Kunde niemals mehrere Voucher auf einmal generieren kann. Wird also zusammen mit dem Voucher-Value der Zeitpunkt der Erstellung gespeichert, kann folgendermassen vorgegangen werden. Es wird immer der älteste Voucher gebucht, der noch nicht gebucht wurde und noch nicht abgelaufen ist. Die Voucher werden also in der Reihenfolge abgebucht, wie sie erstellt wurden. Dies entspricht dem Verhalten eines vernünftigen Benutzers, der diejenigen Voucher mit dem kleineren Ablaufdatum zuerst einlösen würde. Die Fragen "(Wann) wurde ein Voucher eingelöst?" oder "ist ein Voucher bereits abgelaufen" lassen sich bei nicht eindeutigen Voucher-Values aber auch nicht mehr eindeutig beantworten.

Die Verwendung von read/write Tags eliminiert die Probleme im Zusammenhang mit read-only Tags unter Beibehaltung der grossen Benutzerfreundlichkeit und scheint deshalb auf den ersten Blick ideal. Leider sind aber beschreibbare RFID-Tags noch immer teuer in der Herstellung und deshalb ausserhalb von wenigen Spezialapplikationen noch wenig verbreitet. Die hohen Herstellungskosten sind kein Problem bei Applikationen, wo ein Tag beispielsweise sämtliche Stationen eines Schiffscontainers mit wertvollem oder gefährlichem Inhalt speichern soll, bei alltäglichen Anwendungen wie Gebäude-Zutrittsregelung kommen solche Tags aber kaum je zum Einsatz. Darüberhinaus unterscheiden sich beschreibbare Tags auch stark in der erzielbaren Schreibgeschwindigkeit und der vorhandenen Speichermenge, was eine wesentlich stärkere Anpassung des Voucher-Dienstes auf die verwendeten Tags bedingen würde. Aus diesem Grund kommt der Einsatz beschreibbarer RFID-Tags zur Zeit kaum in Frage.

#### 2.2.4 NFC

Der NFC (Near Field Communication) Standard wurde von Sony und Philips als Nachfolger ihrer konkurrierenden Smart-Card-Systeme (FeliCa bzw. Mifare) entwickelt und im Herbst 2002 vorgestellt (Standardisiert durch ISO 18092, ECMA 340, ETSI TS 102 190). Wie viele RFID-Geräte arbeitet NFC auf dem 13.65-MHz-Band. Anders als bei den meisten RFID-Tags können bei NFC über die reine Identifikation hinaus auch Daten übertragen und gespeichert werden. Die maximale Übertragungsrate beträgt 424 kbit/s, wesentlich mehr als bei den meisten RFID-Tags. Ein wesentliches Merkmal von NFC ist der schnelle und automatische Verbindungsaufbau; um zwei Geräte miteinander zu verbinden, muss der Benutzer sie lediglich nahe zueinander halten. Eine Konfiguration der Verbindung (wie etwa bei Bluetooth) durch den Benutzer entfällt. Ein Gerät kann dabei sowohl aktiv als auch passiv kommunizieren. Im passiven Modus können Identifikationsdaten eines Gerätes gelesen werden, selbst wenn dieses nicht eingeschaltet ist.

Eingesetzt in Mobiltelefonen, soll NFC dazu dienen, Daten wie Bilder, Rufnummern oder Web-Adressen schnell und einfach zu übertragen. Ein erstes NFC-fähiges Mobiltelefon ist seit 2005 im Handel (Abbildung 2.2).

**Interaktion.** Da NFC-Geräte wie RFID-Tags eine eindeutige Identifikationsnummer besitzen und der NFC-Standard mit ISO-14443 kompatibel ist, kann die unter 2.2.3 besprochene Lösung für read-only Tags auch für NFC-Geräte verwendet werden.



Abbildung 2.2: NFC-fähiges Mobiltelefon

Darüberhinaus sollte es mit NFC auch möglich sein, den Voucher direkt an das NFC-taugliche Gerät zu senden, analog zur Lösung für beschreibbare RFID-Tags. Ein Benutzer hält dazu beispielsweise sein NFC-fähiges Mobiltelefon an den NFC-Transponder des POF. Das Mobiltelefon speichert den zugesendeten Voucher und schickt ihn später an den NFC-Transponder des POS. Da ein NFC-Gerät über deutlich mehr Speicher als ein simpler Tag verfügt und wesentlich komplexer aufgebaut ist (ein Betriebssystem ist für die Speicherverwaltung zuständig), ist hier im Gegensatz zu der Lösung mit beschreibbaren RFID-Tags auch ein Applet zur Verwaltung der Voucher notwendig. Dieses könnte, falls noch nicht vorhanden, zusammen mit dem Voucher übertragen werden.

**Beurteilung.** Falls die NFC-Technik sich durchsetzen kann und in einigen Jahren die meisten Mobiltelefone NFC-fähig sind, ist diese Form der Voucherverteilung sehr interessant. Voucher können bequem und ohne zusätzlichen Kommunikationskanäle wie SMS oder MMS abgespeichert werden. Auch Kombinationen mit anderen Technologien sind möglich. So könnte ein Voucher via NFC ans Mobiltelefon übertragen werden und als Visual Code dargestellt vom POS dann optisch erfasst werden. Auch die Übertragung und Installation des nötigen Applets ist einfacher als bei der in 2.2.1 besprochenen Methode, da dieses Applet falls nötig einfach zusammen mit dem Voucher ans Mobiltelefon übermittelt wird.

Beim momentanen Entwicklungsstand von NFC und der fast nicht existenten Dokumentation scheint eine erfolgreiche Implementation im Moment eher schwierig. Ein weiteres Problem dürfte die Kompatibilität mit verschiedenen Betriebssystemen sein, da die Applets ja auf Geräten verschiedener Hersteller laufen sollen.

### 2.2.5 Weitere Technologien

Folgende Technologien wären im Prinzip ebenfalls geeignet für die Verteilung von Vouchern, wurden aber wegen der jeweils kurz beschriebenen Nachteile nicht näher untersucht.

- Infrarot.  
Hierbei wird der IR-Port des Mobiltelefones oder PDAs verwendet, um einen Voucher auf dem Gerät abzuspeichern. Dabei ist wiederum eine vorherige Software-Installation auf dem Gerät notwendig.  
Wegen der geringen Verbreitung von Geräten mit IR-Port sowie dem sehr langsamen und manuellen Verbindungsaufbau nicht praktikabel.
- Bluetooth.  
Der Speichervorgang ist identisch wie bei Infrarot. Die meisten neueren Ge-

räte verfügen zwar über Bluetooth; auch hier ist aber der Verbindungsaufbau vom Benutzer manuell durchzuführen. Damit eignet sich auch Bluetooth eher schlecht für sehr kurze Datentransfers.

- Fingerprint.

Analog zum Lesen einer RFID-Identifikationsnummer könnte ein Fingerabdruck oder ein anderes persönliches Merkmal eines Benutzers zur Identifikation verwendet werden. Ein kleiner und hinreichend zuverlässiger Fingerabdrucksensor, wie sie vereinzelt beispielsweise in Notebooks zur Anwendung kommen, könnte ins POF und POS eingebaut werden. Im Gegensatz zur RFID-Lösung wäre ein Fingerabdruck immer eindeutig, und die Benutzer bräuchten überhaupt keine technische Ausrüstung, um den Dienst in Anspruch zu nehmen.

Es ist allerdings sehr fragwürdig, ob die Mehrzahl der Nutzer eine so persönliche Information wie einen Fingerabdruck preisgeben würde bei dem kleinen Nutzen, den ein Voucher typischerweise bietet.

## 2.3 Zusammenfassung

Die Vor- und Nachteile der untersuchten Interaktionsformen sind untenstehend tabellarisch zusammengefasst (Tabelle 2.1). Die Erfüllung der Anforderungen bezüglich Benutzerfreundlichkeit, Deployment und Sicherheit sind von - - bis ++ bewertet.

## 2.4 Existierende Lösungen

Mehrere Firmen bieten bereits Lösungen an, die Mobiltelefone zur Bezahlung oder Verwaltung von Dienstleistungen verwenden. Dabei handelt es sich um Services zum Bezahlen, Empfangen und Speichern von Tickets; eine Verwendung eines ähnlichen Dienstes für Werbezwecke ist mir nicht bekannt. Das Speichern von Tickets ist aber ein dem Speichern von Vouchern nahe verwandtes Problem.

- Der Zürcher Verkehrsverbund (ZVV) hat im Sommer 2005 anlässlich der Street Parade einen Versuch<sup>2</sup> mit elektronischen Tickets via SMS durchgeführt. Die Ticketkosten wurden den Nutzern direkt über ein SMS belastet. Die Kontrolle der Daten erfolgte in diesem Fall manuell durch den Kontrolleur. Der Versuch war technisch ein Erfolg; allerdings war die Nachfrage nach den SMS-Tickets sehr viel kleiner als erwartet, und Störungen im Netz eines der beteiligten GSM-Provider führten bei einem Grossteil der Tickets zu einer massiv verspäteten Auslieferung.
- Die Schweizerischen Bundesbahnen (SBB) bieten auf ausgewählten Strecken seit Anfang Dezember 2005 “Mobile Tickets”<sup>3</sup> an. Das verwendete Interaktionsmodell entspricht dem in Abschnitt 2.2.1 beschriebenen Verfahren; ein zweidimensionaler “Barcode” wird per MMS auf das Mobiltelefon des Kunden gesendet. Validiert werden die elektronischen Bahnbillette mit dem “Zugpersonalgerät”, das die Kontrolleure mit sich führen. Die Billette verwenden nicht Visual Codes, sondern ein sehr ähnliches Verfahren zur optischen Codierung von Werten.

Aus völlig unerklärlichen Gründen können die “Mobile Tickets” nicht über

---

<sup>2</sup>siehe <http://www.zvv.ch/streetparade/default.asp> und [http://www.zvv.ch/ic\\_archiv\\_2005.asp](http://www.zvv.ch/ic_archiv_2005.asp).

<sup>3</sup><http://mct.sbb.ch/mct/reisemarkt/billette/mobile-ticket.htm>



Abbildung 2.3: “Mobile Ticket” der SBB

den Internet-Ticketshop der SBB bestellt werden, sondern nur über eine kostenpflichtige 0900er Nummer. Dadurch verteuern sich die Tickets nicht nur gegenüber “normalen” Bahnbilletten, es wird auch ein wesentlicher Teil des Komfortgewinnes wieder zunichte gemacht. Auch werden zur Zeit nur Kunden der Provider Orange, Sunrise und Swisscom beliefert.

- Die Firma xsmart<sup>4</sup> bietet einen elektronischen Ticket-Service, der ebenfalls ein den Visual Codes sehr ähnliches Prinzip für die Darstellung und Übermittlung der Daten benutzt. Bestellung und Bezahlung der Tickets werden hierbei konventionell wie bei gedruckten Tickets abgewickelt.
- Die Österreichische Firma paybox<sup>5</sup> bietet gar einen Bezahlservice, bei dem Kunden zwecks Autorisation der Zahlung angerufen werden. Löst ein Händler den Bezahlvorgang aus, wird der Kunde angerufen und kann die Zahlung autorisieren, indem er am Telefon seine Geheimzahl eingibt. Der Betrag wird dann vom Bankkonto des Kunden abgebucht.
- Wie NFC-Geräte zum Speichern elektronischer Fahrkarten verwendet werden können, zeigt gegenwärtig ein Versuch<sup>6</sup> des Rhein-Main-Verkehrsverbundes (RMV) in Hanau. In Zusammenarbeit mit Nokia und Philips wird 200 Testkunden die Möglichkeit geboten, sich beim Betreten und Verlassen eines öffentlichen Verkehrsmittels mit einem NFC-Telefon bei einem Lesegerät zu registrieren. Die Ein- und Aussteigeorte werden dabei auf dem Telefon gespeichert und gelten bei Kontrollen als Fahrkarten. Bezahlt wird am Monatsende über eine Rechnung, wobei für alle Fahrten automatisch der günstigste Tarif angewendet wird. Es ist also eine vorherige Registration der Kunden notwendig, und verwendet werden können nur Mobiltelefone mit zuvor aufgespielter Software, die der RMV den Kunden gratis abgibt.

<sup>4</sup><http://www.xsmart.ch/>

<sup>5</sup><http://www.paybox.at/329.php>

<sup>6</sup>[http://www.rmvplus.de/coremedia/getinfo.jsp?seite=/RMVPlus/de/Artikel-Get-in-Get-in-Info/Verweise/Handy\\_20Ticketing/NFC\\_20Handy\\_20Ticketing](http://www.rmvplus.de/coremedia/getinfo.jsp?seite=/RMVPlus/de/Artikel-Get-in-Get-in-Info/Verweise/Handy_20Ticketing/NFC_20Handy_20Ticketing)

	<i>Benutzerfreundlichkeit</i>	<i>Deployment</i>	<i>Sicherheit</i>
Visual Codes	<ul style="list-style-type: none"> <li>+ S/W kann Informationen über Voucher anzeigen</li> <li>+ Keine Benutzereingaben notwendig</li> <li>- S/W Installation benötigt</li> <li>- Kamera notwendig</li> </ul>	<ul style="list-style-type: none"> <li>+ Einfaches Einlösen (Erkennungs-S/W für Visual Codes)</li> <li>- POS benötigt Kamera</li> </ul>	<ul style="list-style-type: none"> <li>+ Anonym</li> <li>+ Diebstahl schwierig</li> </ul>
MMS / SMS	<ul style="list-style-type: none"> <li>+ Information über Voucher im SMS oder MMS</li> <li>+ Unabhängig vom verwendeten Telefon</li> <li>+ Keine S/W-Installation benötigt</li> <li>- Nummereingabe erforderlich</li> </ul>	<ul style="list-style-type: none"> <li>+ Kein Lesegerät am POF benötigt</li> <li>- Kosten für Versand von MMS oder SMS</li> <li>- Für POS Kamera notwendig bei MMS mit Visual Codes, OCR oder manuelle Eingabe bei SMS</li> </ul>	<ul style="list-style-type: none"> <li>+ Diebstahl schwierig</li> <li>- Nicht anonym</li> </ul>
RFID	<ul style="list-style-type: none"> <li>+ Keine Benutzereingaben notwendig</li> <li>+ Keine S/W-Installation benötigt (read-only) Benutzer kann Information über Voucher nicht speichern</li> </ul>	<ul style="list-style-type: none"> <li>+ Einfaches Einlösen</li> <li>- Wenig sinnvoll in Umgebungen, wo RFID nicht schon verwendet wird</li> <li>- (read/write) Einsatz von beschreibbaren Tags zur Zeit nicht realistisch</li> </ul>	<ul style="list-style-type: none"> <li>+ Anonym</li> <li>- Diebstahl einfach, da nur die ID eines Tags gelesen werden muss</li> <li>- Selbe ID bei verschiedenen Benutzern theoretisch möglich</li> </ul>
NFC	<ul style="list-style-type: none"> <li>+ S/W kann Informationen über Voucher anzeigen</li> <li>+ Keine Benutzereingaben notwendig</li> <li>+ S/W-Download automatisch</li> <li>- NFC-fähige Geräte kaum erhältlich</li> </ul>	<ul style="list-style-type: none"> <li>+ Einfaches Einlösen (Erkennungs-S/W für Visual Codes)</li> <li>- Für POS Kamera notwendig zum Lesen der Visual Codes, NFC-Reader bei Lesen über NFC</li> </ul>	<ul style="list-style-type: none"> <li>+ Anonym</li> <li>+ Diebstahl schwierig</li> </ul>

Tabelle 2.1: Zusammenfassung der Vor- und Nachteile verschiedener Technologien

# Kapitel 3

## Design

Wie die Diskussion im vorherigen Kapitel zeigt, ist die Umsetzung eines Voucher-Managements mit verschiedenen Technologien und den sich daraus ergebenden Interaktionsformen denkbar. Jede der untersuchten Interaktionsformen hat spezifische Vor- und Nachteile. Welche Methode die geeignetste ist, hängt stark von der Umgebung ab, in welcher sie eingesetzt werden soll: Sind bereits technische Geräte im Einsatz, die verwendet werden können? Ist das Zielpublikum technisch genug versiert und willens, sich mit neuen Technologien auseinanderzusetzen? Wie hoch dürfen die Kosten pro Voucher sein? Zudem können mit der fortschreitenden technischen Entwicklung neue Transportmechanismen und damit Interaktionsformen entstehen und alte abgelöst werden.

Während der Evaluation der verschiedenen Technologien wurde deshalb schnell klar, dass eine möglichst flexible Architektur, die verschiedene Interaktionsformen zulässt, von Vorteil sein würde. Die Architektur zum Verteilen der Voucher sollte sich deshalb nicht auf ein einziges Szenario beschränken. Ein Einsatz in verschiedenen Umgebungen mit unterschiedlichen Anforderungen sollte mit jeweils möglichst geringem Aufwand für die Anpassung möglich sein.

Konkret sollte die gewählte Architektur

- Sowohl Geräte mit Speicher (SMS, MMS, NFC, Visual Codes, beschreibbare RFID-Tags) wie auch Geräte ohne Speicher (RFID-Tags) bedienen können;
- Werte von verschiedenen Lesegeräten (Kamera, NFC-Reader, RFID-Reader) verarbeiten können;
- Keine Software-Installation auf dem Gerät des Kunden bedingen, aber flexibel genug sein, um diese unterstützen zu können.

Die in diesem Kapitel vorgestellte Architektur erfüllt diese Bedingungen.

### 3.1 Aufgaben der Architektur

Eine genauere Untersuchung aller beschriebenen Interaktionsformen zeigt, dass sich der gesamte Ablauf von der Generierung bis zur Validierung und Buchung eines Vouchers in einzelne Schritte aufteilen lässt. Diese Schritte bleiben stets dieselben, unabhängig von den verwendeten Technologien und Interaktionsformen. Die Aktionen innerhalb desselben Schrittes unterscheiden sich jedoch je nach Interaktionsform, allerdings nicht für alle Schritte gleich stark. Die einzelnen Schritte sind:

1. Anfordern des Vouchers.

Der Benutzer interagiert mit dem POF und fordert einen Voucher an, durch Eingabe der Benötigten Information oder Ausführen einer Aktion.

2. Generieren des Voucher-Values.  
Die Anforderung wird bearbeitet. Falls noch nicht alle verfügbaren Voucher aufgebraucht sind, wird jetzt ein neuer generiert. Die wichtigste generierte Information hierbei ist der Voucher-Value, der den Voucher identifiziert.
3. Verwalten des Vouchers.  
Der neu generierte Voucher muss verwaltet werden, d.h. in eine Datenbank mit allen verteilten Vouchern aufgenommen werden.
4. Generieren der Antwort.  
Bei erfolgreicher Generation des Vouchers wird eine Bestätigung, ansonsten eine Fehlermeldung erzeugt und an den POF geschickt, wo die Meldung dem Benutzer angezeigt wird. Falls nötig, werden in diesem Schritt auch Werte ans Mobile Device übermittelt.
5. Einlösen des Vouchers.  
Der Benutzer möchte den Voucher einlösen und lässt diesen am POS einlesen. Der POS generiert eine Anfrage zur Buchung des Vouchers.
6. Validieren und Buchen des Vouchers.  
Die Buchungsanfrage des POS wird bearbeitet. Wurde der Voucher noch nicht eingelöst und ist das Ablaufdatum noch nicht überschritten, wird der Voucher gebucht. Das Resultat der Buchung wird an den POS zurückgeschickt.

Die Details der einzelnen Schritte und ihre Abhängigkeit von der verwendeten Interaktionsform sind in den nachfolgenden Abschnitten erläutert.

### 3.1.1 Anfordern des Vouchers

Dies ist der Schritt, in welchem die eigentliche Interaktion des Benutzers mit dem System stattfindet. Der POF informiert den Benutzer über die Möglichkeit, Voucher zu beziehen. Ist der Kunde daran interessiert, folgt er den Anweisungen des POF.

Dieser Schritt variiert sehr stark je nach gewählter Interaktionsform. Werden beispielsweise SMS oder MMS verwendet, muss der Kunde eine Telefonnummer eintippen. Bei der Verwendung von RFID-Tags ist keine Benutzereingabe erforderlich, der POF muss stattdessen die mit dem eingebauten Lesegerät einen RFID-Tag lesen.

### 3.1.2 Generieren des Voucher-Values

Im Gegensatz zum vorherigen Schritt ist dieser Schritt fast völlig unabhängig von der verwendeten Interaktionsform. Es müssen lediglich zwei Szenarien unterschieden werden:

1. Gerät mit schreibbarem Speicher.  
Bei diesem Szenario ist das Mobile Device des Benutzers in der Lage, den Voucher-Value zu speichern. Wie diese Speicherung im Detail aussieht, ist in diesem Schritt noch nicht wichtig; dies kann optisch als Bild, als Text-String in einem SMS oder auf beliebige andere Art geschehen. In allen Fällen kann als Voucher-Value ein Wert verwendet werden, der von der ID des Mobilens Devices unabhängig ist. Die Länge des generierten Values muss natürlich auf die maximale Speichergröße beschränkt sein, die wiederum von der Technologie abhängig ist (z.B. 160 Zeichen für ein SMS). Dies kann aber völlig generisch gelöst werden, indem die Anfrage einen Parameter für die maximale Länge enthält.  
In diesem Szenarien muss die Device ID des Mobile Device nicht zwingend

bekannt sein. Ist sie bekannt, kann sie aber benutzt werden, um die Anzahl Voucher pro Kunde zu ermitteln (bzw. zu beschränken, falls erwünscht) und kann zusätzliche Sicherheit bieten, indem sie gespeichert wird und das Einlösen des Vouchers später nur dem Gerät mit dieser ID erlaubt wird.

2. Gerät ohne schreibbaren Speicher.

Hier muss die Device-ID als Voucher-Value verwendet werden, da das Generieren eines davon unabhängigen Wertes natürlich keinen Sinn macht, wenn dieser nicht gespeichert werden kann.

Wie der aufmerksame Leser festgestellt haben wird, impliziert obige Fallunterscheidung auch, dass die Generierung eines Vouchers nur möglich ist, falls die Device-ID bekannt ist oder das Gerät schreibbaren Speicher besitzt.

Die anderen Aufgaben, die in diesem Schritt ausgeführt werden, sind unabhängig von der Interaktionsform:

- Prüfen, ob noch Voucher verfügbar sind.  
Ein Anbieter von Vouchern kann im Vorherein eine maximale Anzahl zu verteiler Voucher festlegen, um Kontrolle über die Kosten zu behalten. Sind schon alle verfügbaren Voucher verteilt, werden keine neuen mehr generiert; stattdessen erhält der Kunde eine Fehlermeldung angezeigt.
- Erhöhen des Zählers für verteilte Voucher.  
Der Counter mit der Anzahl der bereits verteilten Voucher muss inkrementiert werden.
- Beschränkung der Anzahl Voucher pro Kunde.  
Weiter lassen sich auch die Anzahl Voucher pro Kunde beschränken. Dies bedingt aber natürlich, dass es sich um eine Interaktionsform handelt, wo die ID des Mobile Devices bekannt ist.

### 3.1.3 Verwalten des Vouchers

Der neu generierte Voucher wird in die Datenbank eingefügt, in eine Tabelle, die sämtliche generierten Voucher enthält. Dieser Schritt ist völlig unabhängig von der Interaktionsform.

### 3.1.4 Generieren der Antwort

Sind die drei vorgehenden Schritte fehlerfrei abgelaufen, wurde ein neuer Voucher generiert. In diesem Schritt wird eine entsprechende Bestätigung erzeugt und an den POF zurückgeschickt, wo sie dem Benutzer angezeigt wird. Ist etwas schiefgelaufen und konnte kein Voucher erzeugt werden (etwa weil der Benutzer die maximal mögliche Anzahl Voucher pro Kunde bereits bezogen hat), wird ebenfalls eine entsprechende Meldung erzeugt und angezeigt. Darüberhinaus können je nach Interaktionsform weitere Aktionen nötig sein, etwa den generierten Voucher-Value via MMS oder SMS zu verschicken, den Visual Code zur Darstellung des Values zu erzeugen oder Ähnliches. Je nach Interaktionsform unterscheidet sich auch die Antwortmeldung: Im Fall von beschreibbaren RFID-Tags etwa muss auch der Voucher-Value an den POF geschickt werden, damit dieser ihn auf den Tag übertragen kann. Auch die dem Benutzer angezeigte Meldung unterscheidet sich von Fall zu Fall: "Sie werden in wenigen Augenblicken eine SMS erhalten..." oder "Der Voucher wurde erfolgreich auf Ihrem Gerät gespeichert..."



### 3.1.5 Einlösen des Vouchers

Hierbei liest der POS den Voucher-Value vom Mobile Device des Benutzers und versucht, diesen zu buchen. Wie das Anfordern des Vouchers variiert auch dieser Schritt sehr stark mit den verwendeten Interaktionsformen. Je nach Fall muss der POS eine Kamera ansteuern, einen Wert von einem RFID-Lesegerät lesen, einen NFC-Reader verwenden oder auf eine manuelle Benutzereingabe warten. Lediglich das Buchen des Vouchers (eventuell mit vorheriger Prüfung, ob der Voucher noch nicht abgelaufen und noch nicht benutzt ist) ist unabhängig von der Interaktionsform.

### 3.1.6 Validieren und Buchen des Vouchers

Dieser Schritt erfüllt folgende Aufgaben:

- Prüfung des Ablaufdatums.  
Hat ein Voucher sein Ablaufdatum überschritten, kann er nicht gebucht werden. Hierbei gibt es zwei Werte, die von Bedeutung sind: Ein absolutes Enddatum der Gültigkeit, das pro Verteiler festgelegt wird, sowie eine maximale Gültigkeitsdauer für jeden einzelnen Voucher. "SunDime", der Verteiler der Voucher, kann beispielsweise festlegen, dass ein Voucher nur innerhalb von 48 Stunden nach dessen Bezug eingelöst werden kann und sämtliche Voucher ihre Gültigkeit ab einem gegebenen Datum verlieren. Beide Einschränkungen können aber auch weggelassen werden.
- Prüfen der Gültigkeit.  
Nur Voucher, die auch tatsächlich einmal erzeugt worden sind und die noch nicht benutzt worden sind, dürfen gebucht werden.
- Buchen des Vouchers.  
Ist ein Voucher gültig und nicht abgelaufen, kann er gebucht werden. Hierbei wird der Voucher als gebraucht markiert und in einer Datenbanktabelle mit gebuchten Vouchern eingetragen.
- Antwort ans POS.  
Das POS muss eine Meldung über den Erfolg oder Misserfolg des Buchungsvorganges erhalten und anzeigen.

Die Validierung und Buchung des Vouchers ist bei jeder Interaktionsform identisch.

## 3.2 Design der Architektur

Tabelle 3.1 fasst die im Abschnitt 3.1 identifizierten Aufgaben der Architektur und ihre jeweilige Abhängigkeit von der verwendeten Interaktionsform zusammen.

Wie man sieht, gibt es sowohl Schritte, die unabhängig von der Interaktionsform sind (generisch), wie auch solche, die sich je nach Interaktionsform stark unterscheiden. Die Infrastruktur zum Management der Voucher sollte diese Tatsache reflektieren. Deshalb habe ich mich für eine Architektur entschieden, die aus verschiedenen Komponenten aufgebaut ist. Jede Komponente entspricht dabei einem der oben gelisteten Schritte. Für die generischen Schritte (Schritte 2,3 und 6) braucht es jeweils lediglich eine Standard-Implementation der Komponente, für die restlichen Komponenten (Schritte 1,4 und 5) wird pro Interaktionsform eine neue Implementation nötig. Durch die Aufteilung der Funktionalität in einzelne Komponenten wird die Wiederverwendung von Code maximiert und damit die Erweiterung um neue Interaktionsformen wesentlich erleichtert und beschleunigt. Zudem wird die Wartung

<i>Schritt</i>	<i>Abhängigkeit von Interaktionsform</i>
1. Anfordern des Vouchers	variiert stark
2. Generieren des Voucher-Values	generisch, mit Fallunterscheidung zwischen Geräten mit und ohne Speicher
3. Verwalten des Vouchers	generisch
4. Generieren der Antwort	variiert stark
5. Einlösen des Vouchers	variiert stark
6. Validieren und Buchen des Vouchers	generisch

Tabelle 3.1: Aufgaben der Architektur und Abhängigkeit von Interaktionsform

des Codes erleichtert, da die Anzahl Zeilen von “Copy/Paste-Code” minimiert wird, und Änderungen oder Bugfixes lassen sich einfacher und schneller realisieren.

Damit die einzelnen Komponenten reibungslos zusammenarbeiten und deren Funktionalität genau definiert ist, ist jede Komponente durch ein API spezifiziert. Da die Schritte 2 und 3 direkt nacheinander erfolgen und beide generisch sind, wäre es auch möglich, sie zu einem einzigen Schritt zusammenzufassen. Aus Gründen der Übersichtlichkeit des Codes und der architektonischen Klarheit habe ich darauf verzichtet.

Abbildung 3.1 zeigt die Schematische Darstellung der Architektur. Das folgende Kapitel geht näher auf die Implementation des hier vorgestellten Architektur-Modells ein.

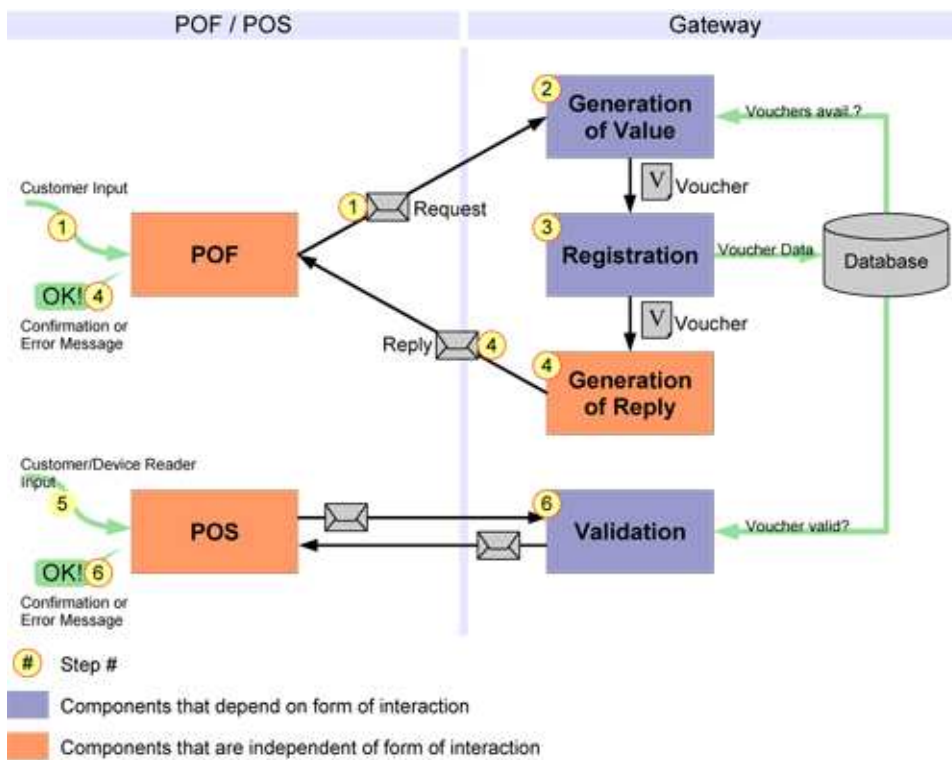


Abbildung 3.1: Schematische Darstellung der Schritte und Komponenten

# Kapitel 4

## Implementation

Die folgenden Abschnitte gehen auf die Implementation der in Kapitel 3 erarbeiteten Architektur und ihre Integration in die technische Umgebung bei lift.tv ein. Der erste Abschnitt zeigt auf, welche Daten zu welchem Zeitpunkt erfasst werden. Abschnitt 4.2 erklärt die technische Umgebung, in der der Voucher-Dienst sich einfügt, und Abschnitt 4.3 gibt einen Überblick über die implementierten Klassen. Für eine detailliertere technische Beschreibung sei hier auf die Javadoc-Dokumentation verwiesen.

### 4.1 Gespeicherte Daten

Die Implementation verwendet das unten genannte Schema, um Daten zu den erhältlichen, bereits verteilten, und gebuchten Voucher zu speichern. Der Primärschlüssel ist jeweils gekennzeichnet. (Für eine detaillierte Beschreibung der Tabellen siehe Anhang A).

#### 1. Voucher Group Table.

Eine “Voucher-Gruppe” fasst sämtliche Voucher zusammen, die ein Werbekunde zur selben Zeit und für dieselbe Aktion verteilt. Sämtliche Voucher für einen Gratiskaffee bei “SunDime”, die “SunDime” in der Eröffnungswoche verteilt, bilden also eine Voucher-Gruppe. Verteilt “SunDime” gleichzeitig auch Voucher für Gratis-Kuchen, bilden diese jedoch eine andere Gruppe. Folgende Daten charakterisieren die Gruppe.

- Group ID. Eine ID zur eindeutigen Identifikation der Gruppe.
- Current Number. Die gegenwärtige Anzahl verteilter Voucher.
- Max Number per Device. Damit lässt sich die Anzahl Voucher beschränken, die ein einzelner Kunde beziehen kann. Damit dies funktioniert, muss die Device ID des Kunden natürlich bekannt sein; vollständig anonyme Interaktionsformen (wie etwa das simple Fotografieren eines Bildes) lassen keine Beschränkung pro Kunde zu. Ist keine Beschränkung gewünscht, kann der Wert auf 0 gesetzt werden.
- Number of Vouchers. Die maximale Anzahl Voucher, die für diese Gruppe verteilt werden.
- Valid After Distribution. Die Zeitspanne, in welcher ein Voucher nach seiner Generierung gültig ist. Falls kein Wert gesetzt wird, ist die Dauer nicht beschränkt.
- Valid Until Time. Ein absolutes Ablaufdatum für alle verteilten Voucher. Falls kein Wert gesetzt wird, ist das Gültigkeitsdatum nicht beschränkt.

- **Seed.** Eine Initialisierungskonstante für die pseudo-zufällige Berechnung von Voucher-Values. Diese wird nicht benötigt bei Interaktionsformen, wo die Device-ID der Voucher-Value ist.
- **Info.** Ein Informationstext, der beispielsweise auf dem POF angezeigt wird. (“Dieser Gutschein berechtigt zu einem gratis-Kaffee beim SunDime!”.)

Zu beachten ist, dass der eigentliche “Wert” des Vouchers nicht formal abgespeichert werden kann. Dieser wird vollständig durch den Werbekunden definiert. Informal kann der Wert etwa als “1 Gratis-Kaffee” oder “Fünf Franken” abgespeichert werden, aber eine formale Quantifizierung des Wertes ist oft gar nicht möglich: Der Wert des Gratis-Kaffees kann natürlich in CHF gespeichert werden, was ist aber, wenn der Kaffeepreis sich ändert?

## 2. Voucher Distributed Table.

Hier werden alle verteilten Voucher verwaltet. Die gespeicherten Daten sind:

- **ID.** Dies eine zusammen mit der Group ID global eindeutige Identifikationsnummer des Vouchers. Der Voucher-Value kann dazu nicht verwendet werden, da er nicht notwendigerweise einmalig ist; siehe dazu die Diskussion im Abschnitt 2.2.3. Die ID entspricht dem Wert des Current Number-Zählers bei der Generierung des Vouchers.
- **Group ID.** Die Gruppe, zu welcher der Voucher gehört.
- **Voucher ID.** Hier ist der eigentliche Voucher-Value gespeichert. Das Feld hat aus historischen Gründen einen irreführenden Namen und ist nicht mit der ID zu verwechseln.
- **Time.** Der Zeitpunkt der Generierung des Vouchers. Dieser Wert wird benötigt, falls der Voucher eine beschränkte Gültigkeitsdauer hat.
- **Distribution Channel.** Das POF hat verschiedene “Channels”, über die Content geliefert wird; einen Wetter-Channel, einen Music-Channel, und so weiter. Dieser Wert bezeichnet den Channel, über den der Voucher ausgeliefert wurde.
- **Is Booked.** Ein Flag, das anzeigt, ob der Voucher bereits benutzt wurde. Schon benutzte Voucher verbleiben also in dieser Tabelle, werden aber als gebraucht markiert. Dies vereinfacht die Validation eines Vouchers.
- **Info.** Hier kann zusätzliche Information abgespeichert werden, beispielsweise an welchem POF der Voucher bezogen wurde.

## 3. Voucher Booked Table.

In dieser Relation wird ein Voucher gespeichert, sobald er gebucht wird.

- **ID.** Die Identifikationsnummer des Vouchers, analog zur Voucher Distributed Table.
- **Group ID,** analog zur Voucher Distributed Table.
- **Voucher ID,** analog zur Voucher Distributed Table.
- **Terminal ID.** Die ID des Terminals (POS), der den Voucher abgebucht hat.
- **Time.** Der Zeitpunkt der Abbuchung.
- **Info.** Ein Zusätzlicher Informations-String.

Der Voucher-Value (Feld Voucher ID) ist redundant, da er durch den Key (ID,Group ID) eindeutig bestimmt ist, und könnte in zukünftigen Versionen auch entfernt werden.

#### 4. Voucher Device Table.

Die Device Table enthält die IDs sämtlicher Devices, die Voucher bezogen haben. Sie dient dazu, die pro Device bezogene Anzahl Voucher zu überwachen und, falls gewünscht, zu limitieren.

- Device ID. Die Device ID des Mobile Device des Kunden. Ein Eintrag kann natürlich nur erfolgen, falls diese bekannt ist.
- Voucher Group ID. Die Voucher-Gruppe, deren Voucher gezählt werden.
- Count. Der Zähler für die Voucher dieser Gruppe für dieses Device.

## 4.2 Infrastruktur des lift.tv Systems

Das lift.tv System, also die Gateways und POFs, basieren auf OSGi<sup>1</sup> [OSGI-1], einer offenen Service-Plattform. Die OSGi-Spezifikationen definieren den Standard für ein Java-basiertes, Komponenten-orientiertes Framework für vernetzte Services. Entwickelt wird der Standard seit 1999 von der “OSGi-Alliance”, einem Konsortium aus Firmen aus verschiedensten Branchen. Die OSGi Alliance stellt auch eine Referenzimplementierung der OSGi-Spezifikationen zur Verfügung (zur Zeit wird diese von lift.tv verwendet), Implementierungen anderer Hersteller (etwa IBM) sind jedoch ebenfalls verfügbar. Nachfolgend sind die wichtigsten Charakteristika von OSGi aufgeführt. Für mehr Details, siehe [OSGI-2].

- Java-basiert.  
Die OSGi-Spezifikationen basieren auf Java-APIs, und die Laufzeitumgebung wird durch eine JVM zur Verfügung gestellt. OSGi kann auf PCs unter J2SE eingesetzt werden, aber auch auf mobilen Geräten mit beschränkteren Konfigurationen wie CDC und CLDC.
- Komponenten-orientiert.  
Eine Komponente ist ein Set von Services, welche auf der OSGi-Plattform ausgeführt werden. Das OSGi-Framework ist für das Lifecycle-Management der Komponenten zuständig: Komponenten können hinzugefügt, gestartet, beendet, entfernt und sogar durch neuere Versionen ersetzt werden, ohne dass die OSGi-Applikation neu gestartet werden muss.  
Komponenten werden vom OSGi-Framework in Form von jar-Files, genannt *Bundles*, verwaltet. Jeder Service ist spezifiziert durch ein API, welches vom jeweiligen Bundle beim Laden des Bundles beim Framework registriert werden kann.  
Zusätzlich zu den Services können Bundles sogar Code, d.h. Java-Klassen, anderen Bundles zur Verfügung stellen. Dies ist vor allem auf Geräten mit beschränkten Ressourcen ein Vorteil, da weniger Code gespeichert und geladen werden muss.
- Dynamisch.  
Mit der Service-Registry des Frameworks können Services von anderen Services gefunden und benutzt werden. Das Suchen von Services ist über Klassen-Namen, den Service-Namen, aber auch über beliebige, zusätzlich festzulegende

<sup>1</sup>Open Services Gateway Initiative. Siehe <http://www.osgi.org/>

Properties möglich. Abhängigkeiten unter den Services werden vom Framework verwaltet, Services können sich so über das Starten, Stoppen oder Updaten anderer Services informieren lassen. Die reibungslose Kooperation der Services wird damit auch in dynamischen Umgebungen, wo Services zeitweise nicht verfügbar sind oder durch neue ersetzt werden, gewährleistet. Durch *Start-Levels* lässt sich festlegen, welche Services miteinander gestartet werden sollen, und welche Services beim Start von bereits laufenden Services abhängen.

- Vernetzt.  
Das Lifecycle-Management der Bundles (also das Hinzufügen, Starten, Stoppen, Entfernen und Updaten) kann von entfernten Rechnern aus geschehen.
- Standard-Services.  
Die OSGi-Plattform bietet eine Anzahl oft genutzter Services bereits an. Dazu gehören etwa Services für Logging, Security-Funktionen, ein HTTP-Servlet-Container, ein XML-Parser, und weitere mehr.
- Effizient.  
Alle Services innerhalb der OSGi-Plattform werden innerhalb einer einzigen JVM ausgeführt, wodurch der Memory-Footprint und die Kosten für Kommunikation zwischen den Services minimiert werden.

Die auf OSGi aufgebaute Service-basierte Architektur des lift.tv-Systems legt nahe, die in Kapitel 3.2 identifizierten Schritte, die jeweils zur Generierung und Validierung sowie Buchung eines Vouchers notwendig sind, als jeweils einzelne Services zu implementieren. Dabei können die nicht von der Interaktionsform abhängigen Schritte beziehungsweise Services (Generieren des Voucher-Values, Verwalten des Vouchers, Validieren und Buchen des Vouchers) in ein Standard-Bundle gepackt werden; dieses ist bei jeder Interaktionsform zum betrieb des Voucher-Dienstes notwendig. Die restlichen, von der Interaktionsform abhängigen Services (Anfordern des Vouchers, Generieren der Antwort, Einlösen des Vouchers) bilden dann pro Interaktionsform je ein weiteres Bundle für POF, POS und Gateway.

### 4.3 Implementation der Services

Für die in Abschnitt 3.1 identifizierten Schritte wurde je ein API festgelegt. Die einzelnen Schritte sind dadurch unabhängig von der Implementation anderer Schritte. Folgende APIs wurden spezifiziert:

<i>Schritt</i>	<i>API</i>
1. Anfordern des Vouchers	-
2. Generieren des Voucher-Values	CreateVoucherService
3. Verwalten des Vouchers	RegisterVoucherService
4. Generieren der Antwort	ReplyVoucherService
5. Einlösen des Vouchers	-
6. Validieren und Buchen des Vouchers	ValidateVoucherService

Tabelle 4.1: Service-APIs

Für die Schritte 1 und 5 sind keine APIs festgelegt. Dies sind die Services auf dem POF (1) beziehungsweise dem POS (5); sie benutzen als Clients die anderen Services, ihr API muss diesen jedoch nicht bekannt sein.

Die Grundidee für die Kommunikation der Services ist, das der Client (also der POF oder POS) nur jeweils einen der Services benutzt. Will der POF einen

neuen Voucher erzeugen, ruft er den `CreateVoucherService` auf. Dieser kreiert ein neues Voucher-Objekt, welches er selber an den `RegisterVoucherService` weitergibt. Die `Create`-, `Register`-, und `Reply`-Services bilden also eine Kette von Services; der POF kommuniziert immer nur mit dem ersten Glied der Kette, also dem `CreateService`. Tritt innerhalb der Kette ein Fehler auf, so wird dies innerhalb der Kette mit Status-Codes des Voucher-Objektes kommuniziert. Der `ReplyService` wird also auch aufgerufen, wenn kein Voucher erzeugt werden konnte; in diesem Fall ist er für die Fehlermeldung ans POF zuständig. Lediglich wenn einer der Services nicht gefunden werden kann, wird nicht die ganze Servicekette durchlaufen und eine Exception geworfen.

Da der `ReplyService` unter Umständen vom POF Parameter benötigt, die im API des `CreateVoucherService` nicht vorgesehen sind, kann beim Aufruf des `CreateVoucherServices` ein `Properties`-Objekt als Parameter mitgegeben werden. Dieses wird ebenfalls entlang der Servicekette weitergereicht und kann beliebige zusätzliche Parameter für die Services in der Form von Name-Value-Paaren enthalten.

Eine ebenfalls denkbare Kommunikationsform wäre die Folgende: Der POF ruft den `CreateVoucherService` auf und erhält von diesem den neu kreierten Voucher. Mithilfe des `RegisterVoucherService` fügt der POF den Voucher in die Datenbank ein. Zuletzt ruft der POF mit dem Voucher-Objekt den `ReplyVoucherService` auf. Diese Kommunikationsform ist aber weniger intuitiv (es werden drei Methodenauf-rufe benötigt, um einen Voucher zu erzeugen), weniger flexibel gegenüber Änderungen der Services (der POF ist von drei APIs statt nur von einem abhängig) und erzeugt zusätzliche Kommunikationslast zwischen POF und Gateway. Die Verwendung einer "Service-Kette" ist deshalb vorzuziehen.

### 4.3.1 CreateVoucherService

Der `CreateVoucherService` wird vom POF genutzt, um einen neuen Voucher zu erzeugen. Der POF ruft dazu die Methode `public Object createVoucher(String groupId, String distrChannel, String deviceId, String info, Properties voucherProps)` auf. Die Parameter sind selbsterklärend und wurden zuvor bereits erläutert. Die zu erwartende Antwort ans POF fällt je nach Interaktionsform verschieden aus; in einigen Fällen wird als Rückgabewert ein Status-Code oder ein einfacher String genügen, aber auch komplexere Rückgabewerte mit zusätzlichen Daten (Bilder etc.) können notwendig sein. Das API lässt daher als Rückgabewert jedes beliebige Java-Objekt zu. Der Typ des Rückgabewertes hängt allein von der Implementation des `ReplyVoucherServices` ab.

Zunächst prüft der `CreateVoucherService` mithilfe der Datenbank, ob für diese Gruppe und gegebenenfalls für diesen Benutzer noch Voucher verfügbar sind. Falls ja, wird der Current Number-Zähler inkrementiert und ein Voucher-Value erzeugt. Das Prüfen und Inkrementieren des Zählers geschieht in einem einzigen atomaren Schritt, damit bei gleichzeitiger Benutzung des Dienstes keine Fehler auftreten (doppelt vergebene Voucher). Wie oben diskutiert, wird als Value je nach Fall die Device-ID oder ein systemgenerierter Wert verwendet. Die beiden Fälle werden folgendermassen unterschieden:

1. Falls keine Device-ID vorhanden ist (der Entsprechende Parameter also den Wert `null` hat), wird ein Value generiert.
2. Falls das `voucherProps`-Objekt die Property `generate_new` mit einem Value ungleich "0" besitzt, wird ein Value generiert. Dies ist sinnvoll in Fällen, wo Geräte verwendet werden, die einen Wert speichern können und wo die Device-ID bekannt ist. (zum Beispiel SMS, MMS.)
3. Die Device-ID wird als Value verwendet.



Wird ein Value generiert, so wird dazu mithilfe des folgenden Algorithmus eine pseudo-Zufallszahl generiert. Die Pseudo-Zufälligkeit ist wichtig, weil bei vorher-sagbaren Values (etwa einer einfachen Durchnummerierung) das Fälschen von Vouchern extrem einfach würde.

$$\text{Value}(n) := Q^{n+seed} \bmod p$$

Wobei  $Q$  eine primitive Wurzel von  $p$ ,  $n$  die Current-Voucher-Nummer,  $seed$  der Seed-Wert aus der Voucher Group Table, und  $p$  die grösste Primzahl kleiner als die maximale Anzahl zu verteiler Voucher für diese Gruppe sind.

Soll ein anderer Algorithmus zur Berechnung verwendet werden, kann dies durch setzen der Property `use_external` im `voucherProps`-Parameter erreicht werden. Der Wert der Property ist dabei der Name eines OSGi-Services, der API B.5 implementieren muss.

Der `CreateVoucherService` erzeugt ein neues `Voucher`-Objekt mit den als Parameter übergebenen Werten und dem berechneten Voucher-Value. Dieses Objekt übergibt er dann dem nächsten Service in der Service-Kette, dem `RegisterVoucherService`. Der `RegisterVoucherService` wird auch aufgerufen, wenn kein Voucher generiert werden konnte, beispielsweise weil für diese Gruppe oder diesen Benutzer keine Voucher mehr verfügbar sind. Lediglich falls der `RegisterVoucherService` nicht verfügbar ist, wird eine Exception geworfen.

### 4.3.2 RegisterVoucherService

Der `RegisterVoucherService` hat die Aufgabe, das vom `CreateVoucherService` erhaltene `Voucher`-Objekt in die Datenbank einzufügen und danach an den `ReplyVoucherService` weiterzureichen. Dazu muss der Typ des zu benutzenden `Reply-Service`s bekannt sein, da durchaus Fälle denkbar sind, wo verschiedene Interaktionsformen nebeneinander betrieben werden. Aus diesem Grund sollte der POF bei seinem Aufruf eine Property `type` setzen. Der `Reply-Service` sollte ebenfalls mit dieser Property beim OSGi-Framework registriert werden.

Die Verwaltung der `Reply-Service`s funktioniert im Detail so, dass der `RegisterVoucherService` das von OSGi spezifizierte `ServiceListener`-Interface implementiert. Wird ein Service der Klasse `ReplyVoucherService` dem Framework hinzugefügt oder entfernt, so wird der `RegisterVoucherService` davon in Kenntnis gesetzt. Dadurch kann intern eine Liste mit bekannten Services pro Interaktionsform aufrechterhalten werden. Sind für eine bestimmte Interaktionsform mehrere `Reply-Service`s bekannt, so wird der zuletzt beim Framework registrierte verwendet. Ist kein `Reply-Service` für den spezifizierten Typ vorhanden, so sucht der `RegisterVoucherService` zunächst nach Services, bei denen der Interaktionstyp nicht spezifiziert wurde. Ist auch kein solcher bekannt, wird eine Exception geworfen.

### 4.3.3 ValidateVoucherService

Der `ValidateVoucherService` wird vom POS benutzt, um die Gültigkeit eines Vouchers festzustellen (ein Voucher kann dann benutzt werden, wenn er noch nicht abgelaufen ist und noch nicht benutzt wurde) und den Voucher, falls gültig, zu buchen. Versucht ein POS, einen bereits benutzten oder abgelaufenen Voucher zu buchen, wird eine Exception geworfen; die POS sollten daher vor dem Buchen die Gültigkeit des Vouchers prüfen. Ist ein zu buchender Voucher-Value mehrfach vorhanden, so wird jeweils derjenige noch nicht benutzte und noch gültige Voucher mit dem kleinsten Verfallsdatum gebucht (siehe die Diskussion im Abschnitt 2.2.3).

Weiter bietet der `ValidateVoucherService` einige Methoden, um auf die Voucher-Datenbank zuzugreifen. Es kann abgefragt werden, welche Voucher-Gruppen existieren, und welche Voucher für eine Gruppe bereits verteilt wurden.

## Kapitel 5

# Diskussion und Weiterführende Arbeiten

Die vorgestellte Architektur ist sehr flexibel. Sie ist geeignet, sämtliche der untersuchten Interaktionsformen unterstützen zu können, und auch sich durch neue, noch nicht untersuchte Technologien ergebende Interaktionsformen sollten damit realisierbar sein. Durch die Aufteilung der Generierung und des Einlösen der Voucher in genau spezifizierte Schritte kann von der Interaktionsform unabhängiger Code in Standardkomponenten verfügbar gemacht werden, was das Hinzufügen neuer Interaktionsformen erleichtert und beschleunigt.

Im Folgenden soll kurz beleuchtet werden, welche Aufgaben mit der gewählten Architektur nicht erfüllt werden können, und welche Erweiterungen davon in Zukunft noch wünschenswert wären.

### 5.1 Nachteile

#### 5.1.1 Konfiguration und Installation

Durch die Aufteilung in Standard-Komponenten und Interaktionsform-abhängige Komponenten ergibt sich eine grössere Anzahl Services, die zusammen die gewünschte Funktionalität erreichen. Es müssen ein Bundle mit Standard-Services sowie pro Interaktionsform ein POS-, POF-, und Reply-Service installiert und konfiguriert werden. Dies bedeutet einen gewissen Mehraufwand gegenüber einem monolithischen Service.

#### 5.1.2 Offline-Validierung

Die Offline-Validierung von Vouchern, also die Validierung durch ein POS ohne Netzwerkverbindung zum Gateway mit der Datenbank, ist im Allgemeinen nur möglich, wenn dabei ein Double-Spending von Vouchern in Kauf genommen wird. Werden beschreibbare Mobile Devices verwendet, ist es zwar grundsätzlich möglich, dass das POS mittels kryptographischer Methoden (Digitale Signatur des Voucher-Values) die Gültigkeit eines Vouchers prüfen kann. Die Sicherheit der Digitalen Signatur hängt dabei von der maximalen Länge des speicherbaren Wertes ab. Bei Verwendung nicht beschreibbarer Mobile Devices (oder Devices mit zu kleinem Speicherplatz für sichere Digitale Signaturen) kann die Gültigkeit des Vouchers anhand einer lokalen Kopie der Voucher-Datenbank beim POS geprüft werden, die regelmässig mit dem Original synchronisiert wird.

Ein POS kann aber offline nicht prüfen, ob ein Voucher bereits einmal bei einem anderen POS verwendet wurde. Werden beispielsweise die POS stündlich mit dem Gateway synchronisiert, ist es innerhalb einer Stunde möglich, denselben Voucher bei verschiedenen POS auszugeben. Machen unehrliche Kunden von dieser Tatsache Gebrauch, sinkt dadurch sicherlich das Vertrauen der Werbekunden ins Voucher-System. Deshalb ist eine Offline-Validierung der Voucher generell nicht empfehlenswert und wurde nicht näher untersucht. Lediglich in folgenden Spezialfällen macht eine Offline-Validierung Sinn:

- Es gibt nur ein POS für die betreffende Voucher-Gruppe.  
In diesem Fall führt der POS lokal eine Liste mit bereits benutzten Vouchern. Die Gültigkeit der Voucher wird anhand einer lokalen Kopie der Distributed-Table oder mit Digitalen Signaturen durchgeführt. Eine regelmässige Synchronisation der Liste gebrauchter Voucher mit der Booked-Table des Gateways und der lokalen Kopie der Distributed-Table mit dem Original auf dem Gateway sind erforderlich.
- Löschen gebrauchter Voucher durch Mobile Device.  
Ist auf dem Mobile Device eine Applikation zur Verwaltung von Vouchern vorhanden, kann diese sicherstellen, dass Voucher nur einmal ausgegeben werden. Dazu muss das Mobile Device vom POS über das Abbuchen eines Vouchers informiert werden und diesen löschen. (Würde der Voucher ohne Kommunikation mit dem POS sofort nach dem Anzeigen oder Senden gelöscht, wäre nicht sichergestellt, dass der Benutzer ihn auch wirklich ausgegeben hat.) Die Gültigkeit der Voucher wird ebenfalls mit Digitalen Signaturen oder durch eine lokale Kopie der Distributed-Table geprüft. Das POS führt eine Liste mit eingelösten Vouchern, die regelmässig mit der Booked-Table synchronisiert wird.  
Da hier sowohl ein Software-Applet auf dem Mobile Device wie auch die Möglichkeit zur Zwei-Wege-Kommunikation zwischen POS und Mobile Device Voraussetzung sind, kommt für diesen Fall von den untersuchten Technologien lediglich NFC in Frage.

Beide beschriebenen Spezialfälle können innerhalb der gewählten Architektur implementiert werden. Zusätzlich ist aber die Implementation eines Mechanismus zur Synchronisation lokaler Datenbanken der POS mit der Gateway-Datenbank notwendig. Das Generieren und Validieren Digitaler Signaturen kann mittels entsprechender Reply- und POS-Services erfolgen. Wird zur Validierung eine lokale Kopie der Distributed-Table verwendet, resultiert als Nachteil, dass neu generierte Voucher nicht sofort gültig sind; sie können erst nach der nächsten Synchronisation von POS und Gateway-Datenbank ausgegeben werden.

## 5.2 Weiterführende Arbeiten

Mögliche Verbesserungen und Erweiterungen der Infrastruktur umfassen:

- Statistiken und Auswertungen.  
Die Voucher-Tabellen der Datenbank enthalten Informationen, die für den Werbekunden interessant sind. Eine Funktion, die die gespeicherten Daten auswertet und dem Werbekunden zur Verfügung stellt, wäre für den Kunden ein Mehrwert. Von Interesse sind dabei vor allem folgende Daten:
  - Wie viele Voucher wurden generiert?
  - Wie viele Voucher wurden eingelöst?

- Nach welcher Zeit wurde im Durchschnitt ein Voucher eingelöst?
  - Wie viele Voucher hat jeder Nutzer im Durchschnitt bezogen?
  - Wie hat sich das Interesse an den Vouchern über die Zeit entwickelt?
- Informationsfunktion für Kunden.  
Werden Mobile Devices ohne Speichermöglichkeit verwendet, kann ein Kunde im Moment nicht sehen, welche Voucher er mit seinem Device bezogen hat. Das POF könnte daher um eine Applikation erweitert werden, die einem Kunden sämtliche Voucher anzeigt, die er bezogen hat, ob sie noch gültig sind bzw. wann er sie eingelöst hat, wozu sie ihn berechtigen und wann sie ablaufen.
  - Web-Applikation zur Erfassung von Voucher-Gruppen.  
Eine entsprechende Applikation würde das Erfassen von Voucher-Gruppen erleichtern; momentan werden diese “von Hand” direkt in die Datenbank eingetragen.
  - Verkauf von Waren.  
Bisher wurde stets das Szenario diskutiert, in welchem ein Werbekunde einer bestimmten Zielgruppe Dienste oder Waren gratis anbietet. Denkbar ist aber auch, dass ein POF zu Verkaufszwecken verwendet wird. Zwar werden Liftpassagiere während einer kurzen Aufzugsfahrt wohl keine grossen Investitionen tätigen, aber Szenarien wie das folgende sind durchaus denkbar: Ein Restaurant in einem Bürogebäude schaltet Werbeanzeigen im Lift. Den Passagieren wird eine aktuelle Liste mit Tagesmenüs angezeigt. Bei Interesse können die Passagiere ein Menu auswählen und mit ihrer Gebäude-Zutrittskarte sofort bezahlen. Dabei wird auch gleich ein Platz im Restaurant für sie reserviert. Hierbei ist im Allgemeinen eine vorgängige Registration des Benutzers zur Abwicklung der Bezahlung notwendig, bei der Interaktionsform mit SMS oder MMS könnten aber auch kostenpflichtige Kurzmitteilungen verwendet werden. Da beim Verkauf von Waren über den POF dem Nutzer Waren in Rechnung gestellt werden, muss auf jeden Fall vor Einsatz einer solchen Lösung geprüft werden, ob die in Frage kommende Interaktionsform die gegebenen Sicherheitsanforderungen erfüllt. Diese können sich von Fall zu Fall unterscheiden und sind etwa in einem öffentlich zugänglichen Gebäude höher als in einem Gebäude, zu dem nur Mitglieder eines einzigen Unternehmens Zutritt haben.

# Anhang A

## SQL Table Scripts

### A.1 Voucher Group Table

```
CREATE TABLE 'voucher_group' (  
  'voucher_group_id' varchar(100) NOT NULL default '',  
  'valid_after_distribution' int(11) default NULL,  
  'valid_until_time' bigint(20) default NULL,  
  'nof_vouchers' int(11) default NULL,  
  'seed' bigint(20) default NULL,  
  'current_number' int(11) default NULL,  
  'max_number_per_device' int(11) NOT NULL default '1',  
  'info' varchar(100) default NULL,  
  PRIMARY KEY ('voucher_group_id')  
);
```

### A.2 Voucher Distributed Table

```
CREATE TABLE 'voucher_distributed' (  
  'id' mediumint(8) unsigned NOT NULL default '0',  
  'voucher_group_id' varchar(100) NOT NULL default '',  
  'voucher_id' varchar(100) NOT NULL default '',  
  'vtime' bigint(20) default NULL,  
  'distribution_channel' bigint(20) default NULL,  
  'info' varchar(100) default NULL,  
  'is_booked' char(1) NOT NULL default '0',  
  PRIMARY KEY ('id','voucher_group_id'),  
  KEY 'voucher_id' ('voucher_id')  
);
```

### A.3 Voucher Booked Table

```
CREATE TABLE 'voucher_booked' (  
  'id' mediumint(8) unsigned NOT NULL default '0',  
  'voucher_group_id' varchar(100) NOT NULL default '',  
  'voucher_id' varchar(100) NOT NULL default '',  
  'terminal_id' varchar(100) default NULL,  
  'vtime' bigint(20) default NULL,  
  'info' varchar(100) default NULL,  
  PRIMARY KEY ('id','voucher_group_id')  
);
```

### A.4 Voucher Devices Table

```
CREATE TABLE 'voucher_devices' (  
  'device_id' varchar(100) NOT NULL default '',  
  'voucher_group_id' varchar(100) NOT NULL default '',  
  'count' int(11) unsigned NOT NULL default '0',  
  PRIMARY KEY ('device_id','voucher_group_id')  
);
```

# Anhang B

## Java APIs

### B.1 CreateVoucherService API

```
public interface CreateVoucherService_api {

    /**
     * the maximal length of the generated value. if set and != "0",
     * this will be used by the CreateVoucherService service.
     * (default: no limit)
     */
    public static final String PROPERTY_VOUCHER_VALUE_MAXLEN = "maxlen";

    /**
     * if set and != "0", a new voucher value will be computed by the
     * Service even if the device ID is set.
     * (default: no)
     */
    public static final String PROPERTY_GENERATE_NEW_VALUE = "generate_new";

    /**
     * if set and != "0", an external service will be used to compute the
     * voucher value. the service name is the property value. the external
     * service must conform to the ComputeVoucherService_api API.
     * (this has no effect if a device id is present AND
     * PROPERTY_GENERATE_NEW_VALUE is not set.)
     * (default: no)
     */
    public static final String PROPERTY_USE_EXTERNAL_CREATOR = "use_external";

    /**
     * this method creates a voucher. the service can either return the Voucher
     * object to the POF, or the service itself hands the Voucher over to the
     * registration service, in which case the reply object should be returned.
     *
     * @param groupId          the Voucher's group ID
     * @param distrChannel     distribution channel
     * @param deviceId         ID of the mobile device. can be null
     * @param info             informal, arbitrary info string can be null
     * @param voucherProps    generic properties of the voucher. can be null
     *
     * @return                reply for the POF (type of return object depends on
     *                        distribution type. for exmple, a voucher can be
     *                        returned, or the CreateVoucherService can itself
     *                        call the RegisterVoucherService, in which case a
     *                        reply for the POF should be returned.)
     *
     * @throws Exception      if an unrecoverable error occurs. generally, errors
     *                        should be reported using the Voucher's status and
     *                        statusMsg fields
     */
    public Object createVoucher(String groupId, String distrChannel, String
        deviceId, String info, Properties voucherProps)
        throws VoucherException;
}
}
```

## B.2 RegisterVoucherService API

```
public interface RegisterVoucherService_api {

    /**
     * used by the register service to select the proper reply service.
     * (default: use any reply service that can be found.)
     */
    public static final String PROPERTY_VOUCHER_TYPE = "type";

    /**
     * add a voucher to the database, i.e. the voucher_distributed table.
     *
     * @param voucher      the voucher that will be added to the database
     *
     * @return             reply for the POF
     *
     * @throws VoucherException
     */
    public Object registerVoucher(Voucher voucher) throws VoucherException;

}
```

## B.3 ReplyVoucherService API

```
public interface ReplyVoucherService_api {

    /**
     * @param voucher      a voucher that has been created and registered
     *
     * @return             reply object for POF
     *
     * @throws VoucherException
     */
    public Object reply(Voucher voucher) throws VoucherException;

}
```

## B.4 ValidateVoucherService API

```
public interface ValidateVoucherService_api {

    /**
     * move voucher from 'distributed' (=issued) to 'booked' (=used) table.
     *
     * @param groupID      voucher group id
     * @param voucherID    voucher value
     * @param terminalID   the terminal where the voucher was booked
     *                    (for statistics, etc.)
     * @param info general info that will be stored in the "booked" table
     *
     * @throws VoucherException if the voucher could not be booked
     */
    public void bookVoucher(String groupID, String voucherID, String terminalID,
        String info) throws VoucherException;

    /**
     * check if a voucher is valid. note: a voucher is considered 'valid'
     * if it exists in the 'distributed' table. used vouchers are thus valid
     * too.
     * use the voucherIsUsed(String, String) method to determine whether it has
     * been used.
     *
     * @param groupID      voucher group id
     * @param voucherID    voucher value
     *
     * @return             true iff the voucher is valid
     *
     * @throws VoucherException
     */
    public boolean voucherIsValid( String groupID, String voucherID) throws
        VoucherException;

    /**
     * check if voucher has already been used.
     */
}
```

```

*
* @param groupID      voucher group id
* @param voucherID    voucher value
*
* @return             java.util.Date when the voucher was used;
*                     null if not yet used
*
* @throws VoucherException
*/
public Date voucherIsUsed( String groupID, String voucherID) throws
    VoucherException;

/**
 * check if voucher is expired.
 *
 * @param groupID      voucher group id
 * @param voucherID    voucher value
 *
 * @return             java.util.Date when the voucher expired;
 *                     null if not yet expired
 *
 * @throws VoucherException
 */
public Date voucherIsExpired(String groupID, String voucherID) throws
    VoucherException;

/*
 * Some General Functions that could be useful
 */

/**
 * returns all distributed vouchers for the given group.
 *
 * @param groupID      voucher group id
 *
 * @return             all distributed vouchers for the group
 *
 * @throws VoucherException
 */
public String[] getDistributedVouchers(String groupID) throws
    VoucherException;

/**
 * returns all voucher groups.
 *
 * @return             java.lang.String[] containing all voucher group ids
 *
 * @throws VoucherException
 */
public String[] getVoucherGroupIDs() throws VoucherException;

/**
 * test whether a voucher group exists.
 *
 * @param groupID      voucher group id
 *
 * @return             true iff the group exists in the voucher group
 *                     table
 *
 * @throws VoucherException
 */
public boolean voucherGroupExists( String voucherGroup) throws
    VoucherException;

/**
 * returns the number of vouchers for the given group.
 *
 * @param groupID      voucher group id
 *
 * @return             the number of distributed vouchers
 *
 * @throws VoucherException
 */
public long getNumberDistributedVouchers( String groupID) throws
    VoucherException;
}

```



## B.5 ComputeVoucherService API

```
public interface ComputeVoucherService_api {  
  
    /**  
     * returns a voucher value, based on arbitrary computations.  
     *  
     * @param voucher      the voucher object containing group ID, device ID,  
     *                    etc.  
     *  
     * @return             the computed voucher value.  
     *  
     * @throws Exception   if the value could not be computed.  
     */  
    public String computeVoucher(Voucher voucher) throws Exception;  
  
}
```

# Literaturverzeichnis

- [THM04] Thomas Kurt Meyer: *Kontextbasierte Werbenetze am Beispiel eines Aufzugsportals*. DISS. ETH Nr. 15601, 2004.
- [VCODE] Michael Rohs et al.: *Visual Code Recognition for Camera-Equipped Mobile Phones*. ETH Zurich, Institute for Pervasive Computing.  
<http://www.inf.ethz.ch/personal/rohs/visualcodes/>
- [OSGI-1] The OSGI Alliance. *OSGi Service Platform, Release 3*. ISBN: 1 58603 311 5, 2003  
siehe auch: [http://www.osgi.org/osgi\\_technology/download\\_specs2.asp](http://www.osgi.org/osgi_technology/download_specs2.asp), <http://bundles.osgi.org/javadoc/>.
- [OSGI-2] The OSGI Alliance. *About the OSGi Service Platform*. Technical Whitepaper, 2004.  
[http://www.osgi.org/documents/osgi\\_technology/osgi-sp-overview.pdf](http://www.osgi.org/documents/osgi_technology/osgi-sp-overview.pdf)