



Die Mobile Payment Libraries von Datatrans

Payment-Lösungen

Mobile Commerce gewinnt stetig an Bedeutung. Hier erfahren Sie, welche Zahlungsarten und technischen Optionen sich für mobile Shops anbieten. **Urs Kipfer und Basil Achermann, Vorwort von Ralf Wölfle**

Auf einen Blick

Der Artikel erläutert verschiedene Konzepte zur Abwicklung von Mobile-Commerce-Zahlungen und stellt Zahlungsarten und Client-Varianten für native Apps sowie Web-Apps vor. Im technischen Teil wird beschrieben, wie sich die „Mobile Payment Libraries“ von Datatrans in iPhone- und Android-Apps integrieren lassen.

Plattform
iOS, Android

So gering der Anteil an Anwendungen für mobile Computer noch sein mag, so häufig die schicken Spielzeuge als Liebhaberei abgetan werden: Einmal mehr werden wir Zeuge, wie sich die IT-Welt grundlegend verändert.

Im zweiten Anlauf – nachdem das Wireless Application Protocol WAP einige Jahre zuvor noch nichts bewirken konnte –, im zweiten Anlauf also gelang der Durchbruch. Zunächst wurde Anwendern vermittelt, wie man einen Computer auch bedienen kann: nämlich in einer Hand und nur mit dem Daumen. Um die Lernenden nicht zu überfordern, wurde mit Geräten geübt, die nur einen einzigen Zweck hatten, nämlich Musik abzuspielen. Das gesamte Handling musste dabei äußerst einfach sein, auch für Menschen ohne tief verinnerlichtes MS-DOS-Basisverständnis. Und die Musik sollte verkauft werden – was damals ein äußerst anspruchsvolles Ziel war. iPod, iTunes und der iTunes Store nahmen all diese Hürden. Und sie schufen sich eine Fangemeinde, die alle Neuerungen aus dem Hause Apple mit Begeisterung aufnahm. Nachdem die Menschen also gelernt hatten,

Computer ohne Tastatur mit einfachen Fingerbewegungen zu bedienen, nachdem sie erfahren hatten, dass Apple ihre digitalen Assets besser organisieren kann als sie selber, und nachdem der Kauf derselben so viel einfacher geworden war, da war die Zeit reif für das iPhone.

Es war so begehrt, dass sich die Vertriebspartner darum rissen – und mit Flatrates für den funkgestützten Datenverkehr einen zweiten wichtigen Erfolgsfaktor schufen. Dass man damit auch telefonieren kann, legitimierte die Anschaffung. Das eigentliche Interesse galt dieser neuen Form von mobilen Computern, ihrer radikalen Einfachheit nicht nur in der Bedienung, sondern auch in der funktionalen Erweiterung. Das iPhone als Inbegriff und Maßstab jeglicher Smartphones schafft echte Ubiquität und erweitert die PC-Funktionalität durch Kamera, Bewegungssensor, Lokalisierung via GPS und mehr. Die durch die Apps erweiterte iPhone-Welt vermag alle zu beeindrucken: Arbeitskollegen mit der Fahrplan- und Ticket-App, Kinder mit der Schlaf-gut!-App und die Stammtischkollegen mit der Puppskissen-App. Wie dröge ist dagegen ein PC!

Einerseits schon wieder Commodity ...

Der Ruhm, den Durchbruch für „Mobile“ geschaffen zu haben, steht unbestreitbar Apple zu. Und Apple hat sich einen großen Vorsprung herausgearbeitet. Aber Smartphones mit ähnlicher Ausstattung, Touchscreens – auch auf anderen Geräten –, das neue Verständnis von Benutzerfreundlichkeit – vor allem bei funktionalen Erweiterungen –, der relativ günstige mobile Datenverkehr, all das ist bereits wieder Commodity geworden. Alle Computer- und Handy-Plattformanbieter arbeiten mit Hochdruck an ihren eigenen Interpretationen und Beiträgen zu den jüngsten Innovationen. Anbieter von Online-Services ziehen mit und machen ihre Angebote für diese neuen Plattformen fit. Das zeigte sich eindrücklich in den Erhebungen zum E-Commerce-Report 2011: 21 von 23 führenden Schweizer E-Commerce-Anbietern erklärten, 2011 in mobile Anwendungen investieren zu wollen [1].

Vom „Mobile“-Trend profitieren nicht nur Computeranwendungen an sich – er ermöglicht auch ein „Permanent Shopping“, so Samy Liechti von Blacksocks.com. Dabei bezieht sich Shopping nicht nur auf physische Produkte, wie das Beispiel Musik gezeigt hat. Anbieter digitaler Produkte – News, Spiele und viele mehr – verbinden mit den neuen Technologien eine neue Chance, das „all for free“ zu durchbrechen. So wird zum Beispiel nach dem Konzept des Freemium [2] versucht, frei nutzbare Basisleistungen in erweiterten Leistungsstufen nur noch kostenpflichtig anzubieten. Das Distributionssystem der App-Stores stellt dazu die gesamte benötigte Infrastruktur bereit, vom Vertrieb über die Abrechnung bis hin zur Ablage oder Installation auf der Plattform des Anwenders.

... andererseits eine Herausforderung

Doch so einfach ist es dann doch nicht. Das rasante Tempo der Innovation mit vielen konkurrierenden Akteuren brachte eine neue Vielfalt in die Computerwelt. Immer mehr Gerätetypen mit unterschiedlicher Ausstattung und Bildschirmauflösung sowie eine zunehmende Zahl von Betriebssystemen wollen berücksichtigt werden. Binnen eines Jahres konnte Googles Android den Vorsprung von Apples iOS dramatisch verringern. „Mobile“ besteht zudem nicht nur aus Apps, auch wenn diese im Moment in der Beliebtheit ganz oben stehen. Es ist leicht nachvollziehbar, dass die Vielfalt der Online-Anwendungen eines Durchschnittsanwenders zu groß ist, um über lokal installierte Apps abgedeckt zu werden. Viel mehr als 10 bis 15 Apps werden wohl die wenigsten regelmäßig nutzen – alles andere muss auch auf mobilen Endgeräten über den Browser erschlossen werden. Mo-

bile Sites sind deshalb der zweite Investitionsschwerpunkt der Studienteilnehmer zum E-Commerce-Report 2011. Allgemein wird erwartet, dass die Betriebssysteme für Mobilplattformen bald auch spezifische Hardwaremerkmale wie die eingebaute Kamera unterstützen können. Mittels Web-Apps sollte sich dann in JavaScript-fähigen Mobile-Browsern mit Offline-Funktionalität ein Funktionsumfang realisieren lassen, der nativen Apps nahekommt.

Sichere Zahlungsabwicklung trotz Plattformvielfalt

Die in naher Zukunft wohl unumgängliche App-Vielfalt ist eine Belastung für die Entwicklung mobiler Anwendungen. Eine Erleichterung ist dagegen die Möglichkeit, generische Teilfunktionalität auszulagern und als externen Service einzubinden. Aufgrund der hohen Anforderungen bietet sich das in besonderem Maße für die Zahlungsfunktionalität für Leistungen an, die nicht via App-Store abgerechnet werden. (Ende Vorwort von Prof. Ralf Wölfle)

Mobile Zahlungslösungen im Überblick

Beim größten Teil der Mobile-Payment-Lösungen auf dem Markt dient das Handy in erster Linie als Instrument zur Zahlungsabwicklung, wobei der Einkauf in einem Ladengeschäft oder im Internet erfolgen kann. Beispiele hierfür sind Paybox, mpass oder Handypayment der Schweizer PostFinance. Für Mobile Commerce (MCommerce) gibt es dagegen bislang nur wenige spezialisierte Anwendungen. Diese lassen sich in die folgenden Kategorien aufteilen:

- **Mobile Payment Pages (MPP)** unterscheiden sich technisch kaum von Payment Pages für herkömmliche E-Shops. Sie laufen in allen mobilen Browsern, die HTTPS unterstützen, und funktionieren mit zahlreichen Zahlungsarten.
- **Mobile Payment Libraries** sollten alle gängigen Kreditkartentypen inklusive 3-D Secure [3] sowie PayPal und die Schweizer PostFinance unterstützen. Da es Softwarekomponenten sind, die in die nativen Apps eingebaut werden, erfordert das Hinzufügen neuer Zahlungsarten in der Regel ein Update der jeweiligen Apps.
- Bei **M-Commerce-Anwendungen mit Server-zu-Server-Autorisierung** werden Kreditkartennummern zwar auf dem Smartphone erfasst, danach aber auf den Applikationsserver des Händlers übermittelt und von dort über eine XML- oder SOAP-Schnittstelle an den PSP-Host übertragen. Dieser Prozess erfordert, dass der Händler PCI-zertifiziert ist. Außerdem können keine externen Prozesse (etwa 3-D Secure) oder externe Zahlungsarten (etwa PayPal oder Schweizer PostFinance) abgewickelt werden. ▶

Autoren



Prof. Ralf Wölfle ist Leiter des Competence Center E-Business Basel (CCEB) der Fachhochschule Nordwestschweiz FHNW. Das CCEB (www.fhnw.ch/wirtschaft/iwi/cceb) ist Teil des Instituts für Wirtschaftsinformatik an der Hochschule für Wirtschaft der FHNW.



Urs Kipfer ist bei Datatrans als Key Account Manager tätig. Er ist für die technische Betreuung von wichtigen Kunden und Partnern zuständig. Datatrans (www.datatrans.ch) ist ein international tätiger Spezialist für Internet-Zahlungen mit Sitz in der Schweiz.



Basil Achermann ist Chefarchitekt bei der iEffects GmbH (www.ieffects.com). Er ist verantwortlich für die nachhaltige Entwicklung der MCommerce-Lösung auf den verschiedenen mobilen Plattformen. iEffects hat die Payment-Library von Datatrans entwickelt.

M-Commerce-Payment-Libraries

Die Integration von Zahlungsfunktionen in native Apps für iPhone und Android bietet den Vorteil, dass Zahlungen stabil und zuverlässig abgewickelt werden können, ohne dass der Anwender die App verlässt (Bild 1). Außerdem sind die Transaktionskosten bei der Verwendung von spezialisierten Payment-Services weitaus günstiger als bei iTunes. Man muss als Anbieter jedoch beachten, dass Apple die Verwendung von Zahlungsservices außerhalb von iTunes nur dann zulässt, wenn entweder physische Güter versandt oder externe Dienstleistungen (zum Beispiel Bahntickets) in Rechnung gestellt werden. Downloads von Multimedia-Content (Musik, Filme, E-Books, E-Papers et cetera) müssen dagegen über iTunes verkauft und abgerechnet werden.

Merkmale einer Payment-Integration

Zu Beginn eines MCommerce-Projekts sollten im Zusammenhang mit dem Bezahlprozess die folgenden Themen geklärt werden:



Der M-Commerce-Zahlungsvorgang (Bild 1)

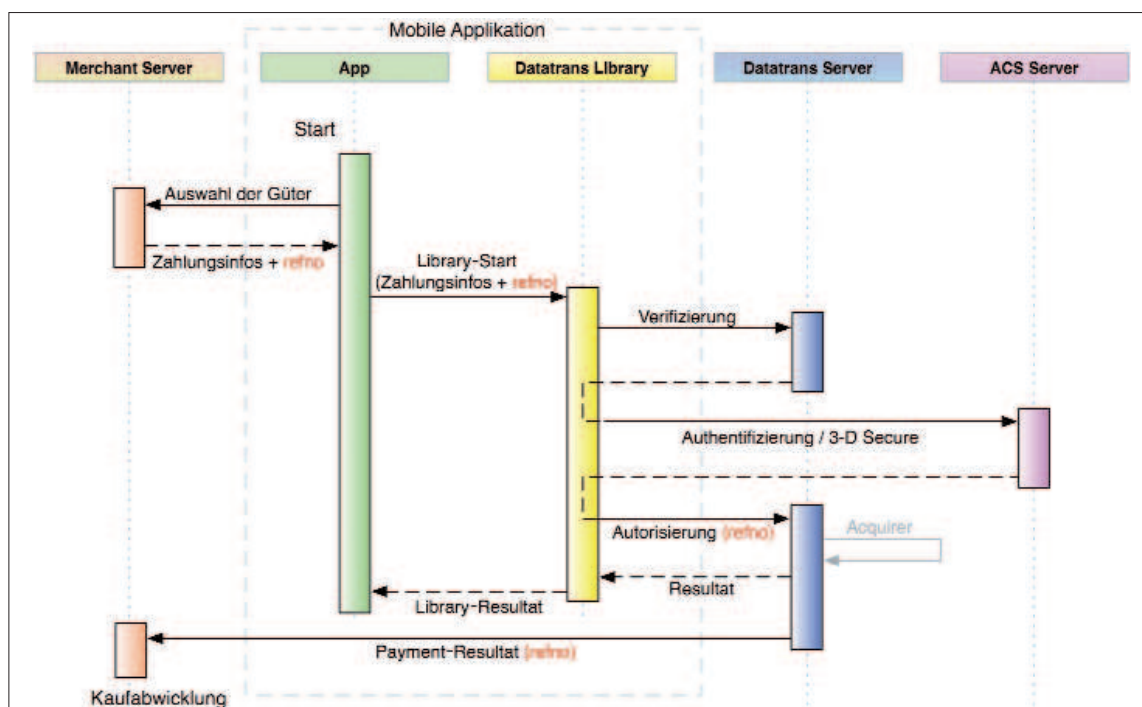
- Die Bezahlprozesse für herkömmlichen E-Commerce und M-Commerce sollten in einem schlüssigen Gesamtkonzept vereint sein. So kann ein Alias (siehe unten) über eine E-Commerce-Seite generiert und über eine M-Commerce-Applikation für wiederkehrende Zahlungen von Stammkunden eingesetzt werden.
- Da die PCI DSS (Payment Card Industry Data Security Standards) dem Händler das Speichern und Weiterleiten von Kreditkartendaten – und sei es auch nur als flüchtige Variable ohne Datenbankspeicherung – strikt untersagen, müssen die Payment-Clients direkt mit dem Applikationsserver des Payment-Service-Providers (PSP) kommunizieren. Der Händler darf zu keinem Zeitpunkt mit der echten Kreditkartennummer in Berührung kommen.
- Gleich wie im herkömmlichen E-Commerce sollte ein PSP auch für den M-Commerce verschiedene grafische Gestaltungsmöglichkeiten

anbieten. Es sollte möglich sein, die Auswahl des Zahlungsmittels, die Eingabe der Kreditkartennummer und die Abwicklung der Zahlung über die Standard-Payment-Page oder mittels der Payment-Library des PSP erfassen zu lassen. Händler mit höheren Ansprüchen an das Design müssten dagegen alle Auswahl- und Eingabeformulare selbst gestalten können. Die PSP-Komponenten regeln in diesem Fall nur die Kommunikation zwischen Smartphone und PSP-Host.

- Aus Sicherheitsgründen wird bei mobilen Zahlungen zuerst jeweils ein Alias generiert. Das heißt, die echte Kreditkartennummer wird durch einen numerischen Wert ersetzt, der an deren Stelle für die Abwicklung der aktuellen Zahlung verwendet wird. Dieser Alias unterliegt nicht den PCI-DSS-Richtlinien und kann deshalb vom Händler problemlos gespeichert werden. Er steht dann auch für künftige Zahlungen zur Verfügung und kann somit zur Vereinfachung des Bezahlprozesses (Stichwort „One-Click-Check-out“) verwendet werden.
- Der PSP sollte für E-Commerce und M-Commerce einheitliche Backoffice-Schnittstellen wie beispielsweise ein Web-Administrations-tool sowie eine umfassende Auswahl an XML-Interfaces oder Webservices (Capture, Status Request, Credit, Cancel) anbieten.
- Ein wesentliches Merkmal einer qualitativ guten Payment-Library muss sein, dass der Kunde zur Bezahlung nicht aus der App des Händlers herausgeführt wird, und zwar auch dann nicht, wenn er zur 3-D-Secure-Passwortabfrage oder zur Freigabe der Zahlung bei PayPal oder PostFinance auf die entsprechenden Webseiten weitergeleitet wird. Der Kaufprozess inklusive Bezahlung endet also immer in der App des Händlers. Um dies sicherzustellen, werden externe Prozesse in einer in die Library eingebetteten Web-View abgebildet.
- Gute Payment-Libraries sind so ausgelegt, dass sie mit einem Minimum an Kommunikation zwischen Smartphone und PSP-Host auskommen. Web-View-Aufrufe erfolgen nur dann, wenn sie auch wirklich erforderlich sind. Hinzu kommt, dass der Status der Zahlung nicht nur an die App des Händlers, sondern auch per Server-Post-Request direkt an den Applikationsserver des Händlers zurückgegeben wird. So ist sichergestellt, dass die Zahlung dem Händler selbst dann quittiert wird, wenn das Smartphone die Netzverbindung während des Bezahlprozesses verliert.

Die Payment-Lösung von Datatrans

Die Payment-Lösung von Datatrans für Shops auf dem iPhone und auf Android-Geräten erfüllt all diese Merkmale. Als Zahlungsarten



Bezahlprozess in der mobilen Applikation (Bild 2)

stehen momentan alle gängigen Kreditkarten, PayPal und die PostFinance-Zahlungsarten (nur iPhone) zur Verfügung.

In ihre Entwicklung sind bereits langjährige Erfahrungen eingeflossen. Bereits 2005 entwickelte Datatrans eine Payment-Page, die auch produktiv implementiert wurde. Sie sollte auf den damals noch kleinen Displays von Mobiltelefonen funktionieren. Das wirklich Spezielle an diesem Projekt war allerdings, dass auch 3-D Secure unterstützt werden sollte. Da die Standardgröße von 3-D-Secure-Passwortabfrageseiten jedoch 390 x 400 Pixel beträgt, musste die Kreditkartenakzeptanz auf einen Kartenherausgeber limitiert werden, der eigens für dieses Projekt eine Art Mini-Page entwickelte. Aufgrund der großen Zahl von Geräteherstellern und -typen, die mit unterschiedlichsten Browsertypen und Versionen ausgestattet waren, erwies sich dieses Vorhaben jedoch bald als kaum realisierbar. Das Projekt wurde deshalb noch in der Pilotphase wieder eingestellt.

Auch wenn man bei Datatrans der Überzeugung blieb, dass es für mobile Applikationen viele nutzbringende Anwendungsbereiche geben würde, trat dieses Thema nach dem ersten Anlauf etwas in den Hintergrund. Im Frühling 2009 führte mit den Schweizerischen Bundesbahnen (SBB) einer der größten Kunden von Datatrans eine mobile Version des Fahrplans mit integriertem Ticketshop ein. Dabei wurde bewusst auf native Applikationen gesetzt, allerdings zunächst noch ohne eine Version für das iPhone. Zwar wurde mit großem Aufwand für alle gängigen Smartphones eine eigene Client-Variante auf Java-Basis entwickelt und gratis zum Download bereitgestellt. Dem mobilen Ti-

cketshop war anfänglich jedoch kein großer Erfolg beschieden. Im August 2009 kam dann die Version für das iPhone, worauf sich die Zahl der mobilen Ticketverkäufe sofort vervielfachte. Der Erfolg des mobilen SBB-Ticketshops bewog Datatrans, dem Bereich M-Commerce strategische Bedeutung beizumessen. Im Sommer 2010 wurde deshalb mit der Entwicklung einer Payment-Library für native iPhone-Apps begonnen. Im September 2010 konnte das erste Release der Datatrans iPhone Payment Library (DTiPL) freigegeben werden.

Architektur der Payment-Libraries

Die Datatrans-Zahlungsbibliothek ist sowohl für das iPhone als auch für Android als native Library verfügbar. Sie lässt sich mit minimalem Integrationsaufwand in Apps integrieren, die ihren Anwendern auch während des Bestellvorgangs bestmöglichen Komfort bieten wollen. Aufgrund der fragilen Netzinfrastruktur, mit der sich mobile Anwender ständig konfrontiert sehen, ist die Bibliothek auf hohe Ausfallsicherheit ausgelegt. Ein guter Teil der Kommunikation, die normalerweise zwischen Servern des Händlers (Merchant) und dem Zahlungsanbieter stattfindet, wurde direkt auf das Gerät verlegt. Dadurch kann auf lange geöffnete und fehleranfällige Kommunikationskanäle verzichtet werden. Des Weiteren werden externe Webseiten wo möglich vermieden. Muss trotzdem eine Webseite dargestellt werden, was für gewisse Zahlungsarten leider unumgänglich ist, wird diese in einer integrierten Web-View angezeigt, ohne in einen Browser zu wechseln. Zoom- und Viewport-Einstellungen der ▶



Zahlungsabwicklung per Kreditkarte (Bild 3)

Web-View sind so parametrisiert, dass Zahlungsseiten möglichst anwenderfreundlich dargestellt werden. Insgesamt finden während eines Bezahlsprozesses lediglich einige wenige Netzaufrufe statt. Sollte es zu Fehlern kommen, lassen sich die Teilschritte wiederholen, ohne die Transaktion zu gefährden.

Bezahlprozess

Der Prozess, der bei einer Bezahlung mit der Bibliothek abläuft, ist in Bild 2 dargestellt. Zunächst wählt der Anwender die zu kaufenden Güter oder Dienstleistungen in der App aus. Zusammen mit anderen Bestellinformationen, wie etwa Lieferdatum, Adresse et cetera, wird der Warenkorb mit dem Server des Merchants abgeglichen. Beim Checkout reserviert der Server die Waren und retourniert der App im Gegenzug eine für diese Bestellung gültige Referenznummer *refno* und den endgültigen Bestellbetrag. Dabei



Auswahl der Zahlungsart (Bild 4)

macht die Bibliothek keine Vorgaben, wie eine solche provisorische Bestellung zu erfolgen hat, dies ist allein Sache der App.

Mit Referenznummer und Preis ruft die App die Datatrans-Bibliothek auf. Die weiteren Schritte werden nun von der Bibliothek abgewickelt. Hierbei kommuniziert sie mit den Servern von Datatrans. Falls eine Kreditkarte 3-D Secure unterstützt, wird dem Benutzer in einer Web-View die jeweilige Webseite der Access Control Server (ACS) der Herausgeberbank angezeigt. Sind Authentifizierung (3-D Secure) und Autorisierung der Transaktion erfolgreich (Betrag reserviert), wird dies vom Datatrans-Backend einerseits dem Webserver des Merchants und andererseits der Bibliothek mitgeteilt. Die Bibliothek informiert die App, dass die Transaktion erfolgreich verarbeitet werden konnte. Diese wiederum leitet die Meldung an den Anwender weiter (Bild 3). Über die zuvor verwendete Referenznummer kann der Server des Merchants die erfolgreiche Transaktion einer Bestellung zuordnen und mit deren Abwicklung beginnen.

Die iPhone-Bibliothek

Die iPhone-Bibliothek ist eine statische Library, die in Xcode zum App-Projekt gelinkt wird. Unterstützt werden alle iPhones mit iOS 3.1.3 und höher. Die Bibliothek kann in zwei Modi betrieben werden. Im Standardmodus kann der Benutzer die Zahlungsart über eine grafische Bedienoberfläche auswählen. Im Hidden-Mode übergibt die App der Bibliothek alle benötigten Zahlungsinformationen. Anschließend findet der Prozess bis auf eine allfällige Anzeige der 3-D-Secure-Webseite in einer *UIWebView* im Hintergrund statt.

Die Views der Bibliothek werden in einem bestehenden *UINavigationController* angezeigt, der von der App bereitgestellt wird. Anders als bei anderen Bibliotheken erlaubt dies das nahtlose Einbetten in bestehende Navigationsstrukturen. Selbstverständlich kann für die Bezahlung auch ein neuer *UINavigationController* instanziiert und

Listing 1: Bezahlung im Standardmodus

```
DTPaymentRequest* pr = [[DTPaymentRequest alloc] autorelease];
pr.amountInSmallestCurrencyUnit = 1000; // 1000 Cent
pr.currencyCode = @"EUR";
pr.localizedPriceDescription = @"Euro 10.-";
pr.merchantId = @"12345";
pr.refno = refno; // Vorgängig vom Merchant Server geliefert

NSArray* creditCards = [NSArray arrayWithObjects:DTPaymentMethodVisa,
DTPaymentMethodMasterCard, DTPaymentMethodAmericanExpress,
DTPaymentMethodDinersClub, nil];
NSArray* paypal = [NSArray arrayWithObjects:DTPaymentMethodPayPal, nil];
NSArray* methods = [NSArray arrayWithObjects:creditCards, paypal, nil];

DTPaymentController* pc =
[DTPaymentController paymentControllerWithDelegate:self
paymentRequest:pr paymentMethods:methods];
[pc presentInNavigationController:self.navigationController
animated:YES];
```

Listing 2: Benachrichtigen des Delegaten im Erfolgsfall

```
- (void)paymentController:(DTPaymentController *)controller
paymentRequestDidSucceed:(DTPaymentRequest *)request {
if (controller.creditCard != nil) {
DTCreditCard* cc = controller.creditCard;
NSString* holder = cc.cardHolder;
NSString* maskedCC = cc.maskedCC;
NSString* alias = cc.alias;
// Details sicher zum Merchant Server übermitteln, oder...

NSData* ccData = [cc data];
// Daten allenfalls verschlüsseln und lokal speichern
}

// Zahlungsbibliothek abräumen
[controller dismissAnimated:YES];
}
```

modal präsentiert werden. Dadurch erscheint der Vorgang mehr als eigenständiger denn als integraler Bestandteil des Checkouts. Der Entwickler hat die volle Flexibilität.

iPhone-Standard-Mode

Die Verwendung der Bibliothek im Standardmodus ist in [Listing 1](#) zu sehen. Zunächst werden die Parameter für die Bezahlung zusammengestellt. Hierbei bezeichnet das Feld *merchantId* eine Kontonummer des Merchants bei Datatrans. Im Beispiel werden 1000 Eurocents bezahlt. Die zur Auswahl stehenden Zahlungsarten sind in zwei *NSArray*s eingeteilt. Dadurch wird eine Gruppierung der Zahlungsarten erreicht ([Bild 4, links](#)). Am Schluss wird ein *DTPaymentController*-Objekt instanziiert und im Navigation-Controller dargestellt. Damit wird die Kontrolle an die Zahlungsbibliothek übergeben.

Nach abgeschlossenem Zahlungsvorgang informiert die Bibliothek die App über den Status mittels eines Callback-Delegat-Objekts, wobei typischerweise der aufrufende View-Controller gleich selbst als Delegat agiert. In den Delegatmethoden muss einerseits die Zahlungsbibliothek abgeräumt und die Benachrichtigung des Anwenders veranlasst werden. Andererseits ergibt sich hier die Möglichkeit, die vom Benutzer eingegebenen Daten auszulesen und für späteren Gebrauch zu speichern, sofern der Anwender damit einverstanden ist ([Listing 2](#)).

Es handelt sich dabei selbstverständlich nicht um sensible Daten wie Kreditkartennummer und Verifikationscode, sondern lediglich um einen Alias, der bei zukünftigen Zahlungen mit Datatrans anstelle der Kreditkartennummer verwendet werden kann, andernorts aber nicht zur Zahlung zugelassen ist. Es ist trotzdem zu empfehlen, diese Daten sicher zum Server des Merchants zu übertragen und im Benutzerprofil zu hinterlegen. Werden die Daten lokal gespeichert, sollten sie gut verschlüsselt werden, beispielsweise mit AES-128 und einem Passwort, das aus Benutzerpasswort und App-Passwort zusammengesetzt wird.

Im Beispielcode wird der Payment-Controller mittels *dismissAnimated:* abgeräumt. Dies führt dazu, dass die Views der Bibliothek ausgeblendet werden und wieder die letzte View vor deren Aufruf angezeigt wird. In einer echten Anwendung würde man noch eine View mit einer Erfolgsmeldung darstellen und für die Bestellung danken.

iPhone-Hidden-Mode

Im Hidden-Mode werden die Bildschirme für die Auswahl der Zahlungsart übersprungen. Dazu muss die App vorgängig über alle relevan-

Listing 3: Alias erstellen

```
(void)createAlias {
    DTAliasRequest* request =
        [[DTAliasRequest alloc] initWithMerchantId:@"12345"
        paymentMethod:DTPaymentMethodVISA number:@"49000000000000003"
        expMonth:12 expYear:2015 cvv:@"123" holder:@"Max Muster"];

    [DTAliasController createAliasWithAliasRequest:request delegate:self];
    [request release];
}
// Benachrichtigung
-(void)aliasRequest:(DTAliasRequest *)aliasRequest
didFinishWithAlias:(DTCreditCard *)cc {
    // cc-Objekt kann gespeichert oder direkt zur Zahlung verwendet werden
}
```

Listing 4: Bezahlen im Hidden-Mode

```
DTPaymentRequest* pr =
    [[DTPaymentRequest alloc] init] autorelease];
...
NSData* ccData = user.ccData; // Daten aus Listing 2 oder 3
DTCreditCard* cc = [DTCreditCard creditCardWithData:ccData];

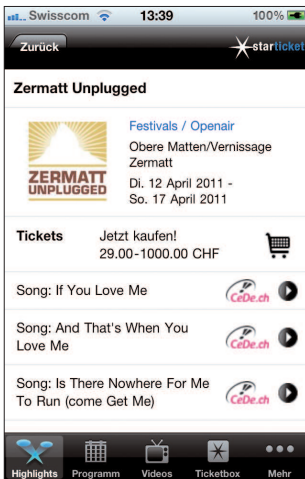
DTPaymentController* pc =
    [DTPaymentController paymentControllerWithDelegate:self
    paymentRequest:paymentRequest creditCard:cc];

[pc presentInNavigationController:self.navigationController
    animated:YES];
```

ten Daten verfügen. Es ist der App überlassen, wie sie an diese Daten gelangt. Im einfachsten Fall geschieht dies über eine vorher erfolgte Transaktion im Standardmodus und dem daraus hervorgegangenen *DTCreditCard*-Objekt mit Alias. Unter Umständen verwendet eine App aber auch ein eigenes User Interface zur Auswahl der Zahlungsart. In diesem Fall muss die eingegebene Kreditkartennummer zunächst in einen Alias umgewandelt werden. Dabei werden alle Daten inklusive Verifikationscode und Ablaufdatum überprüft, jedoch ohne eine Zahlung auszuführen. In [Listing 3](#) ist dies exemplarisch dargestellt. Der Delegat wird asynchron mit einem *DTCreditCard*-Objekt aufgerufen. Dieses kann für den späteren Gebrauch hinterlegt oder unmittelbar für einen Zahlungsvorgang im Hidden-Mode verwendet werden, welcher analog zum Standardmodus aufgerufen wird ([Listing 4](#)). Anstelle der Liste mit den möglichen Zahlungsarten wird einfach das festgelegte Zahlungsmittel verwendet.

Styling und Optionen

Die Bibliothek unterstützt umfangreiche Darstellungseinstellungen, damit sie sich mög- ▶



Beispiel einer Starticket-Anwendung (Bild 5)



Ticketshop der SBB auf dem iPhone (Bild 6)

lichst gut in das Design einer App einfügen lässt. Mittels

```
style = [DTVisualStyle defaultStyle]
```

kann eine gute Grundlage erzeugt werden. Hintergrundfarben, Zellfarben, Schriftfarben und Schriftarten können je nach Bedarf ausgetauscht werden. In Bild 4 wurde für das linke Bild der Hintergrund auf Rot gesetzt, die Zellfarbe auf Weiß, die restlichen Einstellungen entsprechen den Standardwerten. Im rechten Bild wurde zusätzlich die Schriftart ausgetauscht.

Die Bibliothek weist einige Zusatzoptionen auf. Beispielsweise kann beim Bezahlvorgang zusätzlich eine HMAC-Signatur basierend auf einem Preshared-Key mitgegeben werden. Die Signatur wird auf dem Server des Merchants für die Bestellung generiert und während des Autorisierungsvorgangs von Datatrans überprüft. Dadurch kann sichergestellt werden, dass keine Manipulationen am Preis vorgenommen werden. Der Merchant muss somit beim Eingang einer erfolgreichen Zahlungsmeldung Preis und Währung nicht noch einmal überprüfen.

Die Android-Bibliothek

Die Android-Bibliothek ist ein Java-Archiv (jar), welches zum App-Projekt hinzugefügt wird. Aufgrund ungenügender SSL-Unterstützung der *WebView* älterer OS-Versionen kann die Bibliothek nur auf Geräten mit Android-OS-2.2 (API-Level 8) und höher verwendet werden. Die Bibliothek lässt sich wie die iPhone-Bibliothek sowohl im Standardmodus als auch im Hidden-Mode einsetzen.

Android-Standard- & -Hidden-Mode

Die Auswahl der Zahlungsart erfolgt im Standardmodus über mobile Webseiten. Sie werden in einer Android-Web-View in einer eigenen Activity bildschirmfüllend dargestellt. Dieser Modus kann sehr schnell in eine App integriert werden, hat aber den Nachteil, dass am Ende der Transaktion keine Zahlungsdaten ausgelesen und für weitere Zahlungsvorgänge desselben Benutzers wiederverwendet werden können. Für Apps mit hohen Ansprüchen ist daher eine eigene Eingabemaske verbunden mit dem Hidden-Mode zu empfehlen.

Der Aufruf der Bibliothek im Hidden-Mode ist in Listing 5 dargestellt. Nach Erzeugung des Zahlungsmittels wird ein *PaymentProcessAndroid*-Objekt instanziiert und gestartet. Dabei bleibt es der App überlassen, wo auf dem Bildschirm allfällige Webseiten dargestellt werden. Die App übergibt hierfür eine *Android-View-Group*, worin die Bibliothek bei Bedarf eine *WebView* installiert.

Die Benachrichtigung der App erfolgt über einen Listener (Listing 6). Auf keinen Fall dürfen die original Kreditkartennummer oder der Verifikationscode gespeichert werden. Die Bibliothek stellt deshalb im erfolgreichen Fall mit *getAliasPaymentMethod* eine Zahlungsart zur Verfügung, die in zukünftigen Zahlungen mit der Bibliothek anstelle echter Kreditkartendaten verwendet werden kann. Auch hier gilt, dass diese Alias-Zahlungsart in anderen Zusammenhängen nicht eingesetzt werden kann. Trotzdem sollte sie sicher hinterlegt werden.

Zusammenfassung

Die Datatrans-Bibliothek ist eine flexible und einfach zu integrierende Komponente für M-Commerce-Applikationen. Sie stellt momentan die einzige Möglichkeit dar, Anwendern auf mobilen Geräten einen komfortablen und sicheren Zahlungsprozess anzubieten, ohne dabei funktionale oder kommerzielle Abstriche machen zu müssen.

Starticket

Die mobilen Zahlungslösungen von Datatrans kommen bereits in vielen für den Schweizer Markt bedeutenden Projekten zum Einsatz. Stellvertretend für viele andere werden im Folgenden drei typische M-Commerce-Shops vorgestellt.

Starticket [4] ist einer der größten Event-Ticketing-Anbieter in der Schweiz (Bild 5). Das Unternehmen machte schon früh mit Innovationen wie beispielsweise einem „Print at Home“-System auf sich aufmerksam. Grundlage für dieses

Listing 5: Bezahlen im Hidden-Mode (Android)

```
int amount = 1000; // 1000 Cents
String currencyCode = "EUR";
String merchantId = "12345";
String refno = ...; // vorgängig vom
// Merchant Server geliefert
String signature = null; // allfälliger HMAC,
// vom Server geliefert

Payment payment =
    new Payment(merchantId, refno, currencyCode,
        amount, signature, null);
PaymentMethodCreditCard pm =
    new PaymentMethodCreditCard(PaymentMethodType.VISA,
        "4900000000000003", 2015, 12, 123, "Max Muster");
ViewGroup viewGroup =
    (ViewGroup)findViewById(R.id.paymentContainer);
PaymentProcessAndroid ppa =
    new PaymentProcessAndroid(new ResourceProvider(),
        viewGroup, payment, pm);
ppa.addStateListener(myListener);
ppa.start();
```

System ist die Eingangskontrolle mittels Barcode-Leser. Dasselbe Prinzip kommt nun auch für die iPhone-App zur Anwendung. Anstelle eines Ausdrucks auf Papier wird das Ticket als 3D-Barcode auf dem Display des Smartphones angezeigt. Das Smartphone dient so sowohl zur Bestellung als auch als Ticketträger. Stammkunden können sich für die Ticketbox, das Benutzerkonto bei Starticket, registrieren und somit mit der hinterlegten Kreditkartennummer (Datatrans Alias) bezahlen. Die Highlights sind:

- Die Liste aller Vorverkäufe wird in Echtzeit aktualisiert.
 - Mobiler Zugriff auf den Ticketbox-Account.
 - Hinterlegen der Kartennummer als Alias in der Ticketbox.
 - Ticket-Anzeige und Einlasskontrolle über 3D-Barcode.
 - Postversand der Papiertickets möglich.
- MobileTicket ist derzeit als iPhone-App verfügbar. Als Zahlungsarten werden MasterCard und Visa (beide mit 3-D Secure) akzeptiert.

Schweizerische Bundesbahnen (SBB)

Unter dem Titel „Mobileworld“ bieten die SBB seit 2009 ein mobiles Dienstleistungszentrum mit Fahrplan und Ticketshop als native Applikation für alle gängigen Smartphones inklusive iPhone und Android. Die Highlights sind:

- Tür-zu-Tür Reiseinformationen.
- Ereignisinformationen in Echtzeit.
- „Take me home“-Funktionalität mit GPS-Unterstützung.
- Fahrschein des öffentlichen Verkehrs kaufen und anzeigen.
- Hinterlegen der Kartennummer als Alias im SBB-Kundenkonto.
- Bezahlen der Fahrscheine per Kreditkarte.

Der Ticketverkauf ist ausschließlich registrierten Kunden vorbehalten. Die Registrierung erfolgt auf der SBB-Homepage [5]. Die SBB-iPhone-App ist derzeit diejenige mit den meisten Downloads in der Schweiz (Bild 6). SBB Mobile ist für alle gängigen Smartphones verfügbar.

LeShop

Als größter Online-Supermarkt der Schweiz bietet LeShop [6] auch eine iPhone- und Android-App. Das Sortiment ist auf dem Gerät verfügbar und wird durch Synchronisation aktuell gehalten. Damit kann der Einkaufskorb offline zusammengestellt werden. Nur für den Abschluss der Bestellung und die Bezahlung ist ein Internet-Link erforderlich. Die Highlights:

- Vollsortiment mit über 12 000 Produkten mobil verfügbar.
- Überblick über Aktionen und Neuheiten.
- Offline-Modus.

Listing 6: Benachrichtigung (Android)

```
@Override
public void paymentProcessStateChanged(
    PaymentProcessAndroid process) {
    switch (process.getState()) {
        case COMPLETED:
            AliasPaymentMethod pm =
                process.getAliasPaymentMethod();
            if (pm != null) {
                // pm serialisieren und sicher speichern
            }
            break;
        case CANCELED:
            // Bezahlung abgebrochen
            break;
        case ERROR:
            // Fehler behandeln (process.getException())
            break;
    }
}
```

- Speichern des Warenkorbs als elektronischer Einkaufszettel.
 - Verwaltung des LeShop-Kundenkontos inklusive Kreditkartennummer als Alias.
- Als Zahlungsarten werden MasterCard und Visa (beide mit 3-D Secure) sowie American Express und Diners Club akzeptiert (Bild 7).

Geplante Erweiterungen

Mit der zunehmenden Verbreitung der Mobile-Payment-Libraries werden auch laufend neue Wünsche nach neuen Zahlungsmitteln an Datatrans herangetragen. Es sind jedoch nicht alle Zahlungsmittel gleichermaßen für M-Commerce geeignet. Vielfach sind die externen Payment Pages noch für herkömmliche E-Commerce-Anwendungen ausgelegt.

Die nächsten Versionen der Mobile-Payment-Libraries von Datatrans werden um die folgenden Zahlungsmittel erweitert:

- deutsches ELV inklusive erweiterter Bonitäts- und Adressprüfung via CreditPass,
- iDeal (NL),
- PayPal Reference Transactions,
- Alias der Schweizer PostFinance.

Sowohl PayPal Reference Transactions als auch die PostFinance-Alias-Lösung erlauben die Abwicklung von Zahlungen ohne Passwortprüfung, basierend auf einer Ermächtigung durch den Kunden. Das bedeutet eine starke Vereinfachung des Bezahlprozesses für die betreffenden Kunden.

[bl]

[1] www.e-commerce-report.ch

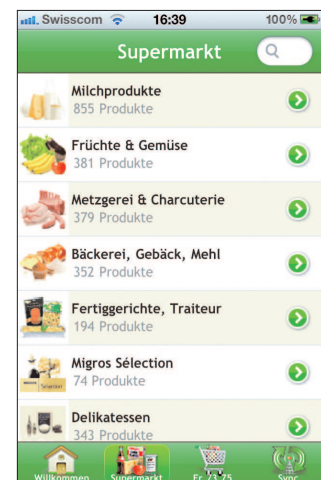
[2] <http://de.wikipedia.org/wiki/Freemium>

[3] http://de.wikipedia.org/wiki/3-D_Secure

[4] www.starticket.ch

[5] www.sbb.ch

[6] www.leshop.ch



LeShop: Lebensmittel per iPhone oder Android-Handy einkaufen (Bild 7)