eos.uptrade
DRIVING INNOVATION

Dokumentation der produktiven TICKeos Semesterticket-Lösungen

beim Verkehrsverbund Rhein Ruhr

VN20171148

Stand: 20.05.2019

Ihr Ansprechpartner:
Adam Laskowski
eos.uptrade GmbH · Schanzenstraße 70 · 20357 Hamburg
Tel: +49 40 - 808070-119 · Fax: 040-808070-100
alaskowski@eos-uptrade.de · www.eos-uptrade.de

Änderungsverzeichnis

Version	Datum	Autor	Beschreibung der Änderung
1.0	30.10.2018	Adam Laskowski eos.uptrade GmbH	Ersterstellung
1.1	15.03.2019	Adam Laskowski eos.uptrade GmbH	Ergänzungen gem. Feedback des Auftraggebers vom 24.01.2019
1.2	19.03.2019.	Adam Laskowski eos.uptrade GmbH	Ergänzung der Doku um eine tabellarische Gegenüberstellung
1.21	14.05.2019	Adam Laskowski/ Vera Mathiszik eos.uptrade GmbH	Ergänzung gem. Auftraggeber-Feedback. Korrektur
1.22	20.05.2019	Adam Laskowski eos.uptrade GmbH	Feedback des Auftraggebers eingearbeitet. Kapitel 11 ergänzt

Inhaltsverzeichnis

Änderungsverzeichnis **2**
1. Abbildungsverzeichnis **5**
2. Glossar **6**
3. Einführung **8**
4. Systemumgebung **8**
5. Funktionale Beschreibung **9**
5.1. Variante 1: KombiTicket-Schnittstelle 10
5.1.1. Rollenverständnis 11
5.1.2. Backend 12
5.1.3. Optionale Ticketdarstellung in Dritt-App via eos.ticketManager 13
5.2. Variante 2: Library / Standalone-App 16
5.2.1. Optionale Einrichtung: Immatrikulationsprüfung 17
5.2.2. Optionale Einrichtung: TICKeos als SSO-Client 19
6. User-Stories **20**
6.1. Variante 1: KombiTicket 20
6.1.1. KombiTicket – Printticket (statisch – PDF) 21
6.1.2. KombiTicket – Mobiles Ticket (statisch - PNG) 21
6.1.3. KombiTicket – Mobiles Ticket (dynamisch) 21
6.2. Variante 2: Mobile Library / Online-Shop 22
6.2.1. Online-Shop – Print Ticket (statisch) 22
6.2.2. Mobile-Shop – Mobiles Ticket (dynamisch) 22
7. Validierung **23**
8. Tabellarische Gegenüberstellung **24**
9. Schnittstelleninformationen **24**
9.1. Variante 1: Kombi-Ticket-Schnittstelle 24
9.1.1. Request (Kombi-Ticket) 24
9.1.2. Response (Kombi-Ticket) 25
9.2. Variante 2: Mobile-Shop API 25
10. Ticket-Templates **26**

10.1. Kombi-Ticket 26
10.2. Mobile-Shop (Lib.) 27
11. Benutzerdaten und Datenschutz 28

1. Abbildungsverzeichnis

Abbildung 1: Systemdiagram der Kombiticket-Schnittstelle im Studententicket-Kontext 10
Abbildung 2: Filtermöglichkeit durchgeführter Ticketanfragen im TICKeos-Backend 12
Abbildung 3: Ticket-Request und ReRequest einer Fahrtberechtigung im TICKeos-Backend .. 12
Abbildung 4: Kombi-Ticket Anfragen im Statistik-Tool des TICKeos-Backend 13
Abbildung 5: Systemdiagram des implementierten eos.ticketManager in einer Fremd-App 13
Abbildung 6: Swimlane-Diagramm Darstellung bei Verwendung des eos.ticketManagers in Fremd-App 14
Abbildung 7: Systemdiagram bei Verwendung einer eos Mobile-Shop Lib. 16
Abbildung 8: Print-Ticket 27
Abbildung 9: Mobiles Ticket zur Darstellung im Smartphone 28

2. Glossar

Begriff	Beschreibung
Base-64	Verfahren zur Kodierung von 8-Bit-Binärdaten (z. B. Grafiken, PDFs) in eine Zeichenfolge, die nur aus lesbaren, Codepage-unabhängigen ASCII-Zeichen besteht
eos.connect	TICKeos Modul zur Authentifizierung bei Dritt-Systemen (analog zu SSO)
Frontend	Benutzeroberfläche
IPSI	Interoperables Produkt Service Interface (Link)
JSON-Format	JavaScript Object Notation: Kompaktes Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen.
Library (Lib.)	Auch: Bibliothek. Wird als Applikation in eine App eingebunden und kann aus der App heraus aufgerufen werden.
Matrikelnummer	Kennung zur Identifizierung einer Person in einem Personenverzeichnis (Matrikel)
Merging	Auch: Verschmelzen. Steht für das Zusammenführen von Entwicklungszweigen (sog. Branches)
PDF-Format	Portable Document Format: Dateiformat für elektronische Schriftstücke, zur plattformunabhängigen und originalgetreuen Darstellung
PNG-Format	Portable Network Graphics: Verlustfreies Grafikformat
Sicherheitsmerkmale	Charakteristische Eigenschaften, die die Echtheit eines Gegenstandes beweisen und eine Fälschung unmöglich machen oder zumindest erheblich erschweren sollen
SMS	Short Message Service: Telekommunikationsdienst zur Übertragung von kurzen Textnachrichten
SSO	Single-Sign-On: Bedeutet für Anwender, dass bei einer einmaligen Anmeldung (Authentifikation), die anschließende Nutzung verschiedener Programme, Dienste und Services ermöglicht wird, ohne sich noch erneut registrieren zu müssen.
TICKeos	Vertriebshintergrundsystem (siehe „Kap. 4 - Systemumgebung“)
UIC918.3	Standard Barcode des internationalen Eisenbahnverbands (siehe „https://uic.org“)

UIC918.3*	Auf dem UIC918.3 aufbauender Barcode der Deutschen Bahn (weiterführende Informationen “Interoperabilität Barcode DB Online-Ticket VDV-KA“)
STB	Statische Berechtigung (Datenbasis für 2D Barcode), auch als VDV-Barcode bezeichnet
VDV-KA	VDV-Kernapplikation

3. Einführung

Diese Dokumentation beschreibt die beim Verkehrsverbund Rhein Ruhr (VRR) produktiven Varianten zur Erzeugung und Ausgabe gültiger Semestertickets. Dabei wird je Variante auch die optionale Implementierung einer Immatrikulationsprüfung erläutert.

Aktuell sind hierfür über das TICKeos Vertriebssystem folgende zwei Ausprägungen im Einsatz:

Variante 1: **KombiTicket-Schnittstelle**

- Hochschule Niederrhein (produktiv)

Variante 2: Mobile-Shop als Library (Lib.) zur Implementierung in die App (iOS/ Android)

- Universität Duisburg-Essen (produktiv)

Beide Varianten greifen auf das TICKeos Vertriebssystem des VRR zu und können unterschiedliche Anwendungsfälle abdecken.

Im nachfolgenden werden die Eigenschaften der Varianten beschrieben, deren primäres Ziel die korrekte Erzeugung einer gültigen Fahrtberechtigung ist. Dabei kann die Fahrtberechtigung auf Wunsch des Auftraggebers einen nach VDV-KA-Standard generierten Barcode enthalten und branchentypische Merkmale zur Sichtkontrolle aufweisen.

Anforderungen, die über die in der Dokumentation beschriebenen hinausgehen, müssen gesondert angefragt und bewertet werden.

4. Systemumgebung

Als Basis für die Generierung der Semestertickets dient die Vertriebsplattform „TICKeos“, die für den VRR eingerichtet wurde. TICKeos ist eine modular aufgebaute Anwendung, die für die speziellen Anforderungen von Verkehrsunternehmen entwickelt wurde und durch neue Features und Anpassungen weiterentwickelt wird. Das System ist flexibel aufgebaut und darauf ausgelegt, einen einheitlichen und integrativen Gesamtansatz auf Basis von Webtechnologie für Applikationen für mobile Endgeräte und andere Vertriebsaktivitäten zu ermöglichen. Es enthält u.a. die Logik zur Erzeugung von Fahrtberechtigungen unter Berücksichtigung aller implementierten Sicherheitsmerkmale, Tarife und der vom Endkunden übergebenen Parameter und kann gleichzeitig auch als zentrale Kundendatenbank für Mobile & Online-Shops genutzt

werden. In TICKeos können Versand-, Print- und Mobile-Tickets generiert werden. Der Barcode kann dabei sowohl nach UIC918.3/ UIC918.3* oder nach VDV-KA Spezifikation signiert werden.

Es existieren für den VRR insgesamt drei TICKeos-Instanzen, die jeweils unabhängig voneinander im eingesetzt werden:

TICKeos – Test-Instanz:
Diese wird bei der Neu- & Weiterentwicklung verwendet und hält immer die neueste Release Version, da Neuentwicklungen in der Regel nicht in alte Versionsstände gemerged werden und eine Umsetzung auf dem Produktivsystem stets ein entsprechendes Release-Update voraussetzt.

TICKeos QS-Instanz:
Bevor Änderungen und Anpassungen auf dem Live-System produktiv gehen, werden sie auf dem QS-System implementiert, wo sie ausführlich durch eine Fachabteilung getestet und anschließend durch den Auftraggeber freigegeben werden. Die QS-Instanz stellt idealerweise ein konfigurationsseitiges Abbild des Live-Systems dar, damit es sich identisch verhält und sinnvolles Testen möglich ist. Aus datenschutzrechtlichen Gründen werden auf dieser Instanz keine echten Kundendaten verwendet.

TICKeos Live-Instanz:
Die Live-Instanz ist das Produktivsystem, welches mit realen Daten arbeitet. Umsetzungen auf dem Live-System erfolgen immer nur nach Freigabe des QS-Systems durch den Auftraggeber. Dadurch vermeidet man ungewolltes/ unvorhersehbares Verhalten der Live-Instanz.

Neben dem TICKeos-Vertriebshintergrundsystem kann es zusätzlich weitere Services geben, mit denen eine Kommunikation stattfinden kann:

- Hochschul-Webservice für die Studenten-Immatrikulation
- Zahlungsmittelanbieter
- VDV-Infrastruktur zur Generierung von VDV-Barcodes
- etc.

5. Funktionale Beschreibung

Die Funktionalitäten und Optionen, sowie die bei der Implementierung entstehenden Aufwände, unterscheiden sich im Wesentlichen je umgesetzter Variante wie folgt.

5.1. Variante 1: KombiTicket-Schnittstelle

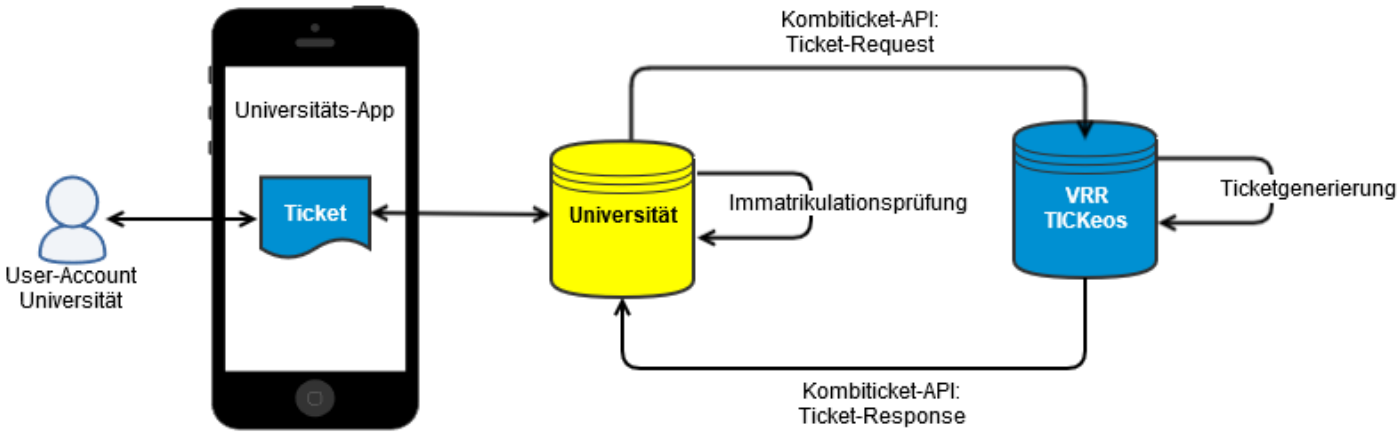


Abbildung 1: Systemdiagram der Kombiticket-Schnittstelle im Studententicket-Kontext

Bei der KombiTicket-Schnittstelle erfolgt die Generierung und Ausgabe von Fahrtberechtigungen auf Basis von XML-Anfragen, die an die TICKeos-Schnittstelle geschickt werden. Hierbei können verschiedenste Parameter zur Erzeugung des Tickets übergeben und verarbeitet werden.

Die Prüfung einer Matrikelnummer sollte bei dieser Variante bereits vor dem Request durch den Systemanbieter des anfragenden Systems erfolgen.

Wurden die im Request übergebenden Parameter erfolgreich verarbeitet, wird in TICKeos eine auf Kundenwunsch konfigurierte Fahrtberechtigung mit allen relevanten Daten generiert. Diese wird zusammen mit einem Statuscode im Base64-Format als Response zurückgesendet und kann vom Systemanbieter anschließend weiterverarbeitet werden.

Für das Ticketlayout werden sogenannte Ticket-Templates im JSON-Format konfiguriert, die auf Kundenwunsch und unter Berücksichtigung aller Sicherheitsaspekte mit Daten befüllt werden (siehe auch Kap. 10 – Ticket-Templates). Die erzeugte Berechtigung kann dabei nur im PDF- (Printticket) oder PNG-Format (z.B. zur Darstellung im mobilen Endgerät) ausgeliefert werden. Optional kann das generierte Ticket auf einem Server gespeichert und über einen in der Response zurückgelieferten Link aufgerufen werden (z.B. für den Abruf via SMS).

Diese Variante erzeugt den geringsten Aufwand bei der Einrichtung. Das Frontend sowie die Logik zur Kundenverwaltung und Immatrikulationsprüfung wird durch die Hochschule bzw. das anfragende System durchgeführt. Die Schnittstelle gibt lediglich eine gültige Fahrtberechtigung auf Basis der erhaltenen Daten zurück. Ist eine Erzeugung nicht möglich, wird ein

entsprechender Status-Code zurückgespielt, der vom Systemanbieter weiterverarbeitet und an den Endbenutzer ausgegeben werden kann.

Geeignet ist diese Art der Implementierung z.B. bei Hochschulen, die bereits über ein eigenes Kunden- bzw. Studentenportal (Online oder App) verfügen und um eine Ausgabe des Print- bzw. mobilen Tickets, erweitern möchten.

5.1.1. Rollenverständnis

Da die KombiTicket-Schnittstelle ursprünglich aus dem Veranstaltungsbereich stammt und für die Semesterticket-Lösung weiterentwickelt wurde, sollen nachfolgend die entsprechenden Rollen beschrieben werden, zwischen denen bei der Konfiguration des TICKeos-Systems unterschieden wird:

Systemanbieter
Im Kontext der KombiTicket-Schnittstelle für Semestertickets ist der VRR als Systemanbieter eingerichtet. Dem Systemanbieter sind die konfigurierten Produkte und Ticket-Templates zugeordnet und über ihn werden die entsprechenden Fahrtberechtigungen ausgestellt. Im Kontext von Veranstaltungen kann dies z.B. der Betreiber der Veranstaltungshomepage sein.

Veranstalter
Als Veranstalter ist im klassischen Sinne der Anbieter eines Events gemeint, der beim Verkauf eines Veranstaltungstickets eine entsprechende Berechtigung anfragt und diese z.B. auf dem Veranstaltungsticket zusätzlich abbildet. Er kann u.U. auch gleichzeitig der Systemanbieter sein.

Im Kontext der Semestertickets ist der Veranstalter die jeweilige Hochschule, welche die Tickets über die Schnittstelle anfragt. Dem Veranstalter können verschiedene Ticket-Templates zugeordnet sein.

Veranstaltung
Veranstalter legen für ihre geplanten Events die entsprechenden Veranstaltungen an. Im Ticket-Template kann konfiguriert werden, dass der Name der Veranstaltung z.B. auf der Berechtigung abgebildet wird. Zudem kann im Nachhinein nachvollzogen werden, wie viele KombiTickets für eine Veranstaltung generiert und/ oder storniert wurden.

Im Kontext der Semester-Tickets gilt ein Semester als Veranstaltung. Damit eine entsprechende Darstellung und Zuordnung möglich ist, muss für jedes Semester eine Veranstaltung angelegt werden.

5.1.2. Backend

Im TICKeos-Backend werden die verarbeiteten Ticket-Anfragen mit den jeweils verwendeten Parametern und den daraus resultierenden Fahrtberechtigungen gespeichert. Dadurch können diese jederzeit abgerufen und nachvollzogen werden.

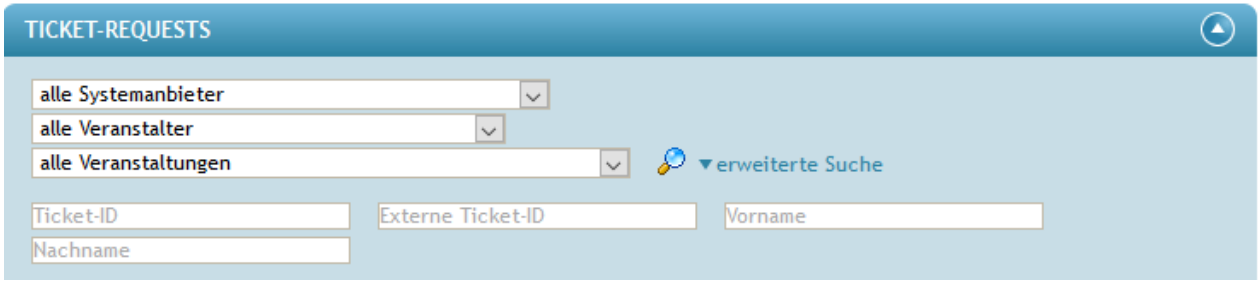


Abbildung 2: Filtermöglichkeit durchgeführter Ticketanfragen im TICKeos-Backend

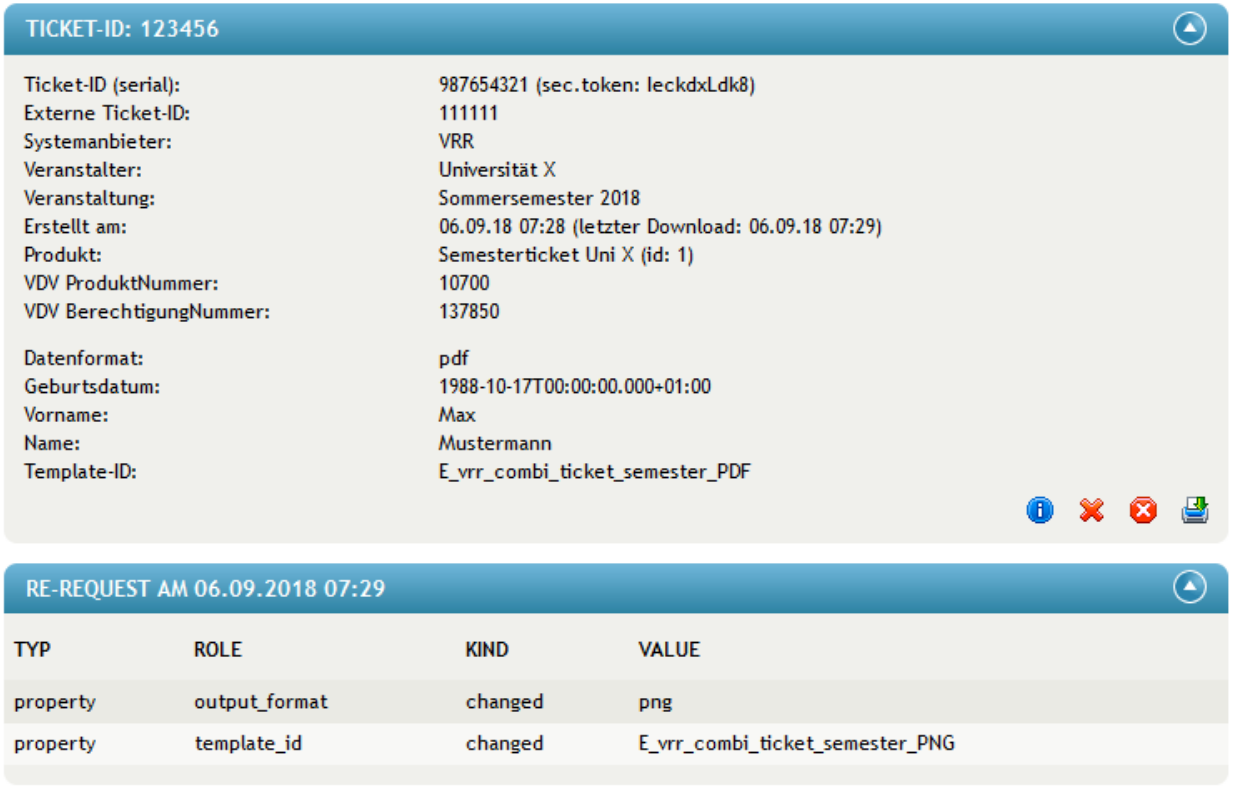
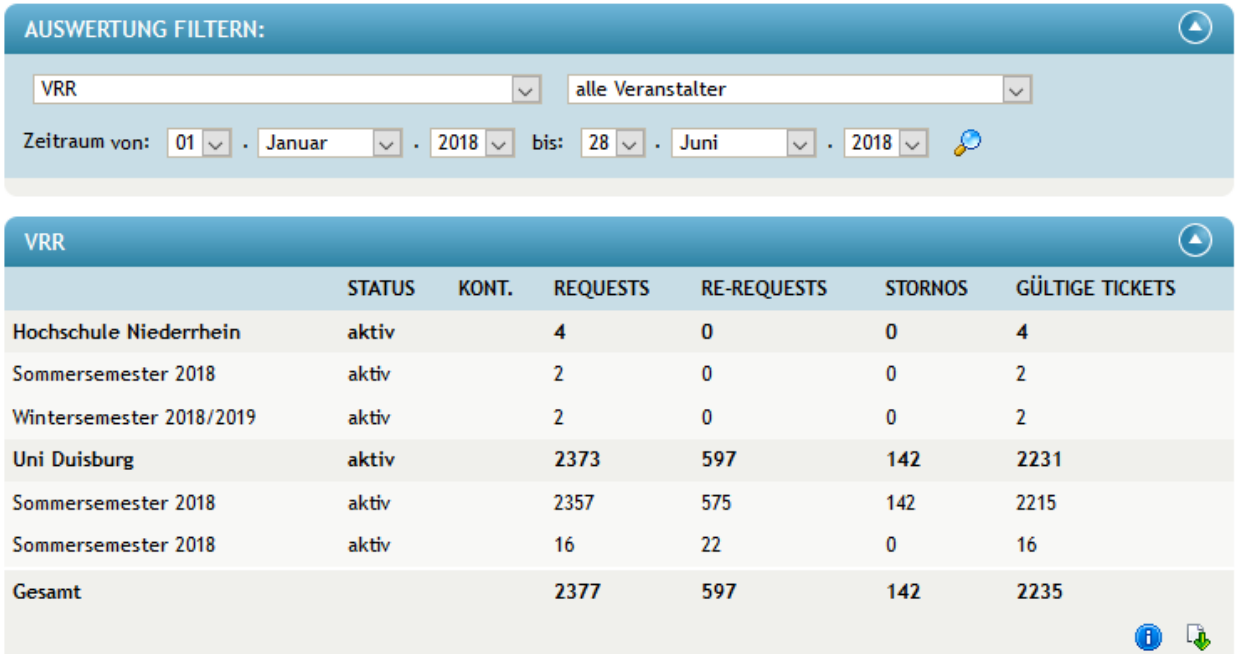


Abbildung 3: Ticket-Request und ReRequest einer Fahrtberechtigung im TICKeos-Backend

Im Statistikmodul des TICKeos-Backends ist eine statistische Übersicht über die Anfragen möglich. Hier kann nach der entsprechenden Hochschule sowie einem Zeitraum gefiltert und die Daten angezeigt und als Datei im CSV-Format heruntergeladen werden.



	STATUS	KONT.	REQUESTS	RE-REQUESTS	STORNOS	GÜLTIGE TICKETS
Hochschule Niederrhein	**aktiv**		**4**	**0**	**0**	**4**
Sommersemester 2018	aktiv		2	0	0	2
Wintersemester 2018/2019	aktiv		2	0	0	2
Uni Duisburg	**aktiv**		**2373**	**597**	**142**	**2231**
Sommersemester 2018	aktiv		2357	575	142	2215
Sommersemester 2018	aktiv		16	22	0	16
Gesamt			**2377**	**597**	**142**	**2235**

Abbildung 4: Kombi-Ticket Anfragen im Statistik-Tool des TICKeos-Backend

Ticket-Anfragen können aber auch darüber hinaus über die Kombi-Ticket-Schnittstelle angefragt werden in dem eine bestimmte Methode aufgerufen wird. Ein TICKeos-Backendzugang wird somit für den Aufruf generierter Tickets etc. nicht zwingend für einen Veranstalter bzw. die Hochschule benötigt.

5.1.3. Optionale Ticketdarstellung in Dritt-App via eos.ticketManager

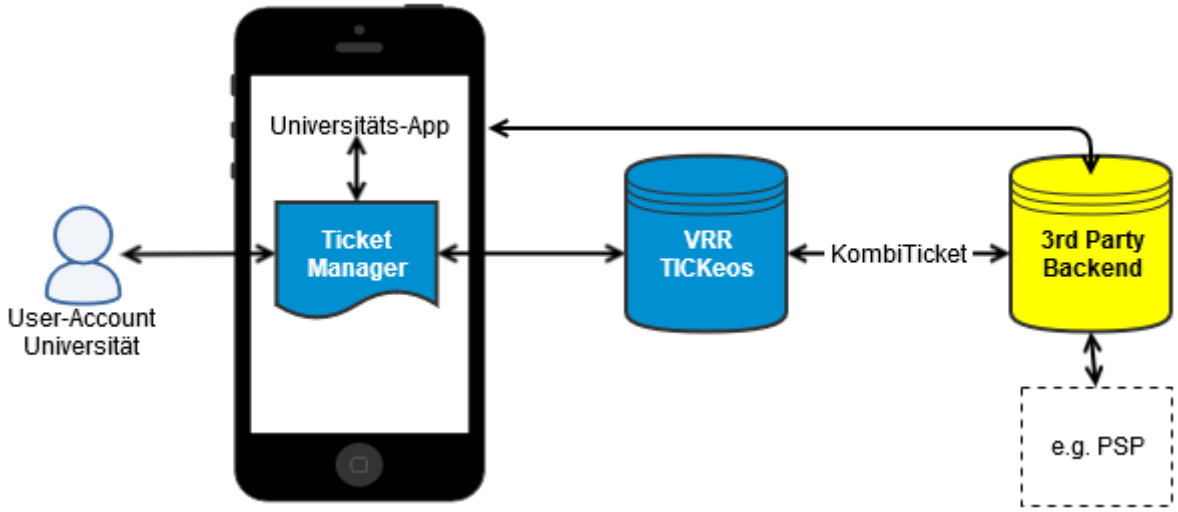


Abbildung 5: Systemdiagram des implementierten eos.ticketManager in einer Fremd-App

Der eos.ticketManager bietet die Möglichkeit, mobile Tickets, die über die TICKeos B2B oder KombiTicket-Schnittstelle generiert wurden, in eine mobilen App herunterzuladen, zu

speichern und anzuzeigen, ohne dass ein vollumfänglicher Mobile-Shop mit Kauffunktion implementiert sein muss.

Diese Komponente lässt sich in jede beliebige App einbetten, ohne dass die Benutzeroberfläche optisch verändert werden muss. Dadurch ist die Möglichkeit gegeben, mobile Tickets mit und ohne dynamische Sicherheitsmerkmale im JSON-Format (z.B. nach IPSI-Vorgaben) darzustellen, wodurch sie sich optisch nicht von mobilen Ticketlayouts der Verkehrsunternehmens-Applikationen unterscheiden müssen.

Der eos.ticketManager ist in der Lage, sich mit mehreren TICKeos Hintergrundsystemen zu vernetzen. Dies hat u.a. den Vorteil, dass der eos.TicketManager bei der Initialisierung auf das QS-System zugreift und nach Freigabe per Konfiguration ein Zugriff auf das Live-System möglich ist ohne eine neue Lib. für die Fremd-App bauen zu müssen.

Das Backend der Applikation erzeugt über die KombiTicket-API in TICKeos ein Ticket und gibt die in der Antwort enthaltene Referenz an die Fremd-App, welche diese Information wiederum an den eos.ticketManager weiterleitet. Dieser ruft das Ticket ab und kann es anschließend darstellen. TICKeos-Funktionen wie die Nutzerverwaltung oder das Payment können in diesem Fall nicht genutzt werden.

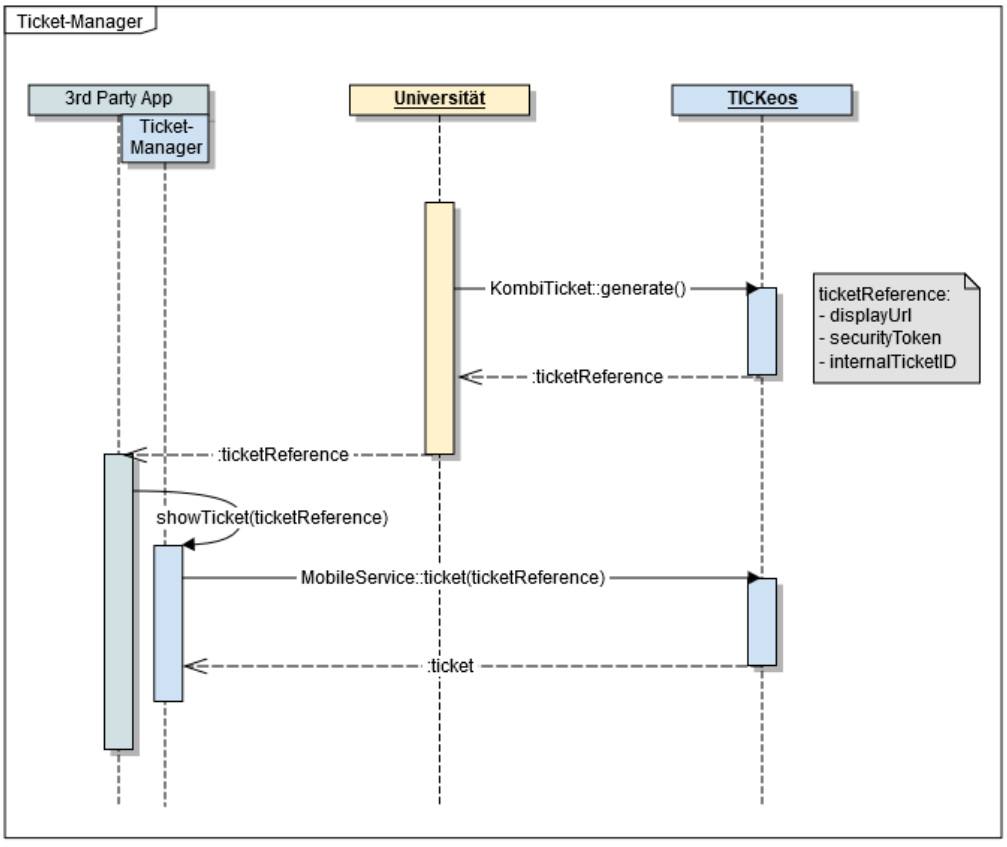


Abbildung 6: Swimlane-Diagramm Darstellung bei Verwendung des eos.ticketManagers in Fremd-App

Das Ticket wird demnach komplett vom eos.ticketManager verwaltet und der Betreiber der Fremd-App kommt mit dem eigentlichen Ticket nicht in Kontakt.

5.1.3.1 Einbindung des eos.ticketManagers

5.1.3.1.1 Android

Es ist wichtig, dass die TickeosTicketActivity des Ticket-Managers in das Android-Manifest der Applikation eingebunden wird. Dies kann z.B. folgendermaßen geschehen:

```
<activity
android:name="de.eosuptrade.mobileshop.ticketmanager.ticket.TickeosTicketAct
ivity"/>
```

Einige Funktionen des eos.ticketManagers wurden in separate Services ausgelagert. Diese Services müssen genauso wie die Activities des eos.ticketManager, in das Android-Manifest der Applikation aufgenommen werden. Die Einbindung kann dabei wie folgt aussehen:

```
<service
android:name="de.eosuptrade.mobileshop.ticketkauf.mticket.services.sync.tick
ets.TicketSyncService" />
```

```
<service
android:name="de.eosuptrade.mobileshop.ticketkauf.mticket.services.sync.tick
ets.TicketDownloadService" />
```

```
<service
android:name="de.eosuptrade.mobileshop.ticketkauf.mticket.services.log.LogSe
rvice" android:permission="android.permission.BIND_JOB_SERVICE"/>
```

Weiterhin ist es notwendig, dass alle JAR-Libraries aus dem Lib.-Ordner des Ticket-Managers in die Applikation eingebunden werden, sofern noch nicht geschehen. Üblicherweise erfolgt dies durch die Angabe der Pfade zu den JARS im Build-Pfad.

Für die Enbindung des eos.ticketManagers müssen folgende Voraussetzungen erfüllt sein:

- minSdkVersion: API Level 14 (Android 4.0) oder höher
- targetSdkVersion: API Level 27 (Android 8.1) oder höher
- Android Support Library Appcompat-v7 Version 27 oder höher

Eine aktuelle Dokumentation wird bei der Auslieferung der angeforderten Bibliothek beigelegt.

5.1.3.1.2 iOS

Der eos.ticketManager erwartet als Deployment Target mindestens iOS 9.0 und als Base SDK iOS 9.x. Es werden die folgenden Architekturen unterstützt: „armv7“, „armv7s“, arm64“ und für den Simulator die Architekturen „i386“ und „x86_64“.

Es werden folgende, unter FOSS-Lizenzen veröffentlichte Pakete und Frameworks benötigt:

- AFNetworking → 3.0 https://github.com/AFNetworking/AFNetworking

- MBProgressHUD → 0.9 https://github.com/jdg/MBProgressHUD

Aufgrund der vereinfachten Einbindung und der klaren Abbildung der benötigten Pakete und Frameworks empfehlen wir die Einbindung als "CocoaPod".

Wird der eos.ticketManager nicht als "CocoaPod" eingebunden, sind die benötigten Pakete und Frameworks manuell zu implementieren.

Es werden folgende System-Frameworks vorausgesetzt:

- CoreData.framework
- CoreGraphics.framework
- Foundation.framework
- QuartzCore.framework
- Security.framework
- CoreBluetooth.framework
- CoreLocation.framework
- CoreMotion.framework
- libz.tdb

Eine aktuelle Dokumentation wird bei der Auslieferung der angeforderten Bibliothek beigelegt.

5.2. Variante 2: Library / Standalone-App

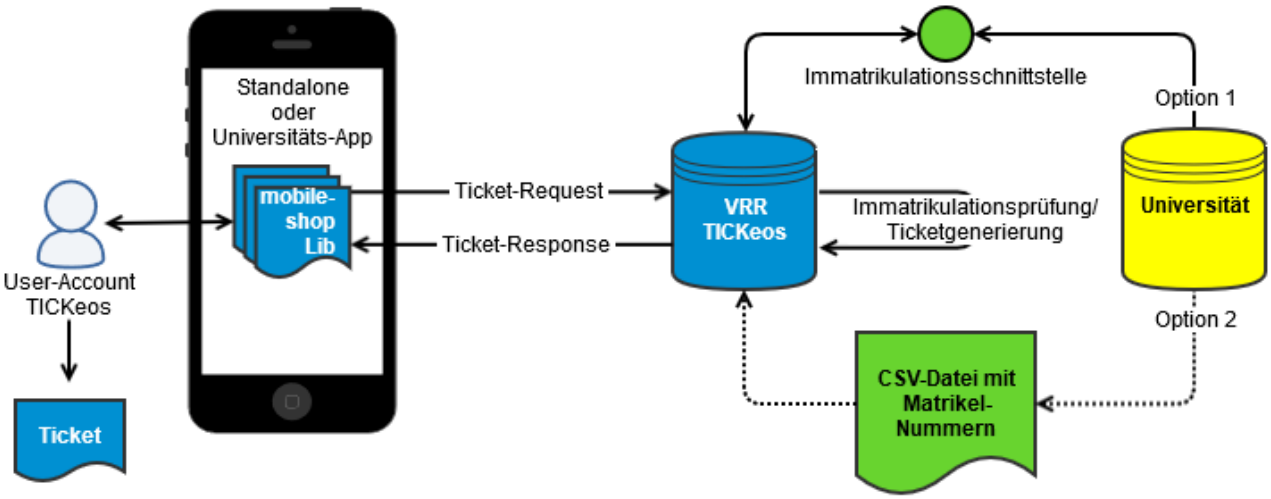


Abbildung 7: Systemdiagram bei Verwendung einer eos Mobile-Shop Lib.

Bei der Mobile-Shop-Variante wird ein Vertriebssystem eingerichtet, das als Library ausgeliefert und in eine Fremd-App eingebunden werden kann. Dadurch ermöglicht man z.B. über einen bestimmten Einstiegspunkt innerhalb der Fremd-App (z.B: über einen Menüpunkt „Semesterticket“) den Zugriff auf den Mobile-Shop.

Bei der Nutzung des Mobile-Shops ist der Benutzer dazu angehalten, sich einen persönlichen Kundenzugang zu erstellen. Dadurch ist eine Selbstverwaltung der persönlichen Daten durch den Benutzer jederzeit möglich. Zudem ist das elektronische Ticket nicht an ein bestimmtes Gerät gebunden und kann bei einem erneuten Download der App, sowie einem erfolgreichen Login mit den persönlichen TICKeos-Zugangsdaten, jederzeit abgerufen werden. Alle Tickets sind stets einem konkreten Benutzeraccount zugeordnet. Ob und wie oft ein Ticket heruntergeladen wurde, kann im Backend eingesehen werden.

Die für die Registrierung erforderlichen Kundendaten können durch den Auftraggeber individuell im TICKeos-Backend festgelegt werden. Verpflichtend ist standardmäßig die Emailadresse, der Vor- & der Nachname, sowie das Geburtsdatum. Die Emailadresse dient dabei immer als Benutzername und kann sowohl für den Login in der mobilen App, als auch im Online-Shop (sofern vorhanden) verwendet werden.

Darüber hinaus können im Zuge der Registrierung z.B. Marketing-Permissions eingeholt werden, die der Benutzer jederzeit in den Einstellungen widerrufen kann.

Neben der Ausgabe des Tickets innerhalb der Smartphone-App kann auch eine Ausgabe als Printticket optional über den Online-Shop realisiert werden. Dabei können der Mobile- als auch der Online-Shop an das bestehende Layout angepasst werden, was eine nahtlose Implementierung in die bestehende Onlinepräsenz gewährleistet.

Für das Semesterticket wird im TICKeos-Hintergrundsystem ein Produkt konfiguriert. Hierbei können Änderungen an den Tarifdaten sowie den Parametern wie Gültigkeitszeitraum, Verfügbarkeit, Bezeichnung, VDV-KA relevante Angaben, personalisierungsrelevante Einstellungen etc. vorgenommen werden. So kann ein Benutzerfoto für die Generierung eines Semestertickets (als zusätzliches Sicherheitsmerkmal) verpflichtend sein, das durch den Benutzer hochgeladen und verwaltet werden muss.

Bei der Nutzung des Mobile-Shops erfolgt der vollständige Prozess der Semesterticket-Ausgabe im TICKeos System. Es muss lediglich die Library in die Dritt-App eingebunden werden. Optional kann auch eine Standalone-App mit dem Funktionsumfang des Mobile-Shops bereitgestellt werden.

5.2.1. Optionale Einrichtung: Immatrikulationsprüfung

Im Falle des Mobile-Shops wird die Immatrikulationsprüfung in das TICKeos-System ausgelagert. Dabei wird im Wesentlichen zwischen zwei Methoden unterschieden:

5.2.1.1 Immatrikulationsprüfung über einen Webservice (Option 1)

Es ist möglich, eine Schnittstelle zu einem Hochschul-Webservice einzurichten, der die aktualisierten Immatrikulationslisten vorhält, gegen die geprüft wird. Vorteil bei dieser Variante ist, dass die Daten stets aktuell sind und jederzeit direkt von der Hochschule verwaltet werden können.

Hierfür bietet TICKeos eine allgemeine Schnittstelle für Hochschulen, die folgendes voraussetzt:

- Einen End-Point der WSDL (https empfohlen)
- URL der WSDL
- Optional:
  - htaccess-Zugangsdaten (falls eingerichtet)
  - Authentifizierungstoken (falls vorgesehen)

Es gibt eine Empfehlung zur Schnittstellen-Struktur. Wenn diese verwendet wird, können Kosten für Anpassungen vom VU gespart werden.

Ein Request würde damit beispielsweise so aussehen:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <verifyDataRequest>
      <token>3423k24</token>
      <Matrikelnummer>1234</Matrikelnummer>
      <Name>Mustermann</Name>
      <Vorname>Max</Vorname>
      <Geburtsdatum>1990-01-01</Geburtsdatum>
      <Postleitzahl>1234</Postleitzahl>
      <Semesterkuerzel>10W</Semesterkuerzel>
    </verifyDataRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Der Response wäre bei erfolgreicher Validierung wie folgt:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <verifyDataResponse>
      <result>true</result>
      <fehlerCode>0</fehlerCode>
    </verifyDataResponse>
  </s:Body>
```

```
</s:Envelope>
```

5.2.1.2 *Immatrikulationsprüfung über bereitgestellte CSV-Dateien (Option 2)*

Verfügt eine Hochschule über keine entsprechende Schnittstelle, ist auch ein Import der Immatrikulationslisten im CSV-Format möglich. Eine automatische Prüfung findet dabei gegen die hinterlegte Liste statt. Sollte sich die Liste ändern, liegt es in der Verantwortung der Hochschule, eine neue Version rechtzeitig bereitzustellen. Die parallele Nutzung mehrerer Listen ist nicht vorgesehen.

5.2.2. Optionale Einrichtung: TICKeos als SSO-Client

5.2.2.1 *Shibboleth-Schnittstelle*

Es gibt in TICKeos die Möglichkeit einer Shibboleth-Schnittstelle. Hierbei handelt es sich um eine Validierungs-Alternative zu dem im vorherigen Abschnitt beschriebenen Vorgehen mit WSDLs (Webservice). Dieses Vorgehen läuft aus Benutzersicht wie folgt ab:

Der Benutzer wählt im Produktformular aus einer Liste seine Bildungseinrichtung aus. Diese Bildungseinrichtung ist der „IdentityProvider“. Ein paar Schritte später, wenn der Benutzer auf "Jetzt kostenpflichtig bestellen" klickt, wird der Browser des Kunden auf eine Seite des „IdentityProviders“ weitergeleitet, wo er sich authentifizieren muss, z.B. per Username und Passwort. Danach wird der Benutzer wieder zurück in den TICKeos-Shop geleitet. TICKeos wertet die vom „IdentityProvider“ gelieferten Daten (SAML-Attribute) aus und entscheidet anhand derer, ob der Kunde zum Kauf des Produkts berechtigt ist oder nicht.
Von der Universität wird folgendes benötigt:

- Location/URL der Idps (dev/live)
- Test-Account mit eingeschriebenem Semester auf dev und live

Folgende SAML-Attribute müssen nach Rücksprung von der Universität an TICKeos übergeben werden:

- givenName (Vorname)
- sn (Nachname)
- eduPersonScopedAffiliation
- studySemester (Semesterkürzel)

Beim Auswerten der vom IdentityProvider gelieferten Daten wird Folgendes gemacht:

- Prüfen ob der vom IdentityProvider übermittelte Semesterschlüssel mit dem erwarteten Schlüssel übereinstimmt. Stimmen die Schlüssel nicht überein, ist der Kunde nicht berechtigt das Produkt zu kaufen.
- Validieren des vom IdentityProvider übertragenen Attributs „eduPersonScopedAffiliation". Ist darin die Zeichenfolge vor dem "@"-Zeichen nicht "student" ist der Kunde nicht berechtigt, das Produkt zu kaufen. Nähere Informationen

zu dem Attribut "eduPersonScopedAffiliation" findet man unter http://wiki.aaf.edu.au/tech-info/attributes/edupersonscopedaffiliation.

- Bestimmte Personalisierungsfelder werden mit den vom IdentityProvider übertragenen Daten überschrieben. Dies ist notwendig, um Missbrauch zu verhindern. Ansonsten wäre beispielsweise folgendes Szenario möglich:
    - Person A ist Student, Person B ist kein Student. Person A versucht ein auf Person B personalisiertes Ticket zu kaufen. Dazu authentifiziert sich Person A an seiner Bildungseinrichtung. Ohne das Überschreiben würde ein ermäßigtes und auf Person B ausgestelltes Ticket verkauft werden, obwohl Person B keinen Anspruch auf die Ermäßigung hat. Dadurch dass Name und Vorname überschrieben werden, wird dies verhindert.

TICKeos überträgt also keine Informationen über den Kunden an den IdentityProvider. Der Kunde authentifiziert sich am IdentityProvider und dieser gibt nur die für TICKeos wirklich notwendigen Daten über den Kunden zurück.

5.2.2.2 OAuth2 & OpenID

Mit der Weiterentwicklung von eos.connect, soll durch Nutzung der SSO-Standards „OAuth2“ & OpenID-Connect das TICKeos System als SSO-Client eingerichtet werden können. Vorausgesetzt wird hierfür, dass das datenführende System der Hochschule diesen Standard unterstützt.

Dabei soll sich der Benutzer mittels seiner Zugangsdaten des datenführenden Hochschul-Portals im Online- / Mobile-Shop registrieren bzw. einloggen können. Bei initialer Eingabe der genannten Zugangsdaten wird durch Austausch von Tokens ein neuer TICKeos Account erstellt, der an den Benutzer-Account der Hochschule geknüpft ist. Bei dem Prozess erteilt der Benutzer dem TICKeos-System die Erlaubnis der Nutzung bestimmter Benutzerdaten. Diese können definiert werden und werden zur Erzeugung der Fahrtberechtigung verwendet. Benutzerdaten, die nicht bereitstehen, können nachgelagert durch den Benutzer ergänzt werden (z.B. ein Benutzerfoto).

Die Implementierung des SSO muss immer als individuelle Implementierung betrachtet werden.

6. User-Stories

Für eine anschaulichere Beschreibung, werden die beiden Varianten unter Berücksichtigung der jeweiligen Optionen als User-Stories beschrieben:

6.1. Variante 1: KombiTicket

Student A ist an der Universität X immatrikuliert, welche ihren Studenten bereits ein Webportal mit allerlei Informationen und Diensten bereitstellt. Da die Studenten Anspruch auf ein

Semesterticket haben, wurde das Webportal um die Ausgabe der Studententickets erweitert. Da die Universität ihr Online-Portal überwiegend selbstständig pflegt und verwaltet, soll dies lediglich um eine Ausgabe gültiger Semestertickets erweitert werden. Die Immatrikulationsprüfung findet innerhalb des Portals statt. Studenten, die nicht berechtigt sind, haben keinen Zugriff auf den Service.

6.1.1. KombiTicket – Printticket (statisch – PDF)

Da die Universität weiß, welche Studenten für welches Semester immatrikuliert sind, stellt sie sicher, dass der Service nur die berechtigten Studenten berücksichtigt. So erhält Student A zum Beginn des Semesters (bzw. ab einem Stichtag) die Möglichkeit, sich in seinem persönlichen Portal ein Semesterticket generieren zu lassen, das ausgedruckt werden kann. Dabei wird ein definierter SOAP Request mit den relevanten Parametern an eine eingerichtete KombiTicket-Schnittstelle gesendet. Als Antwort erhält das aufrufende System im Response ein Base64 codiertes Ticket als PDF oder PNG Datei.

6.1.2. KombiTicket – Mobiles Ticket (statisch - PNG)

Die Universität X betreibt eine eigene App in der sich die Studenten anmelden können um auch von unterwegs auf die entsprechenden Services zugreifen zu können. So soll den Studenten eine Ausgabe des Semestertickets innerhalb der App als PNG-Datei geboten werden. Hierfür wird ein neues Ticket-Template für PNG-Formate im TICKeos-Backend eingerichtet. Nach erfolgreichem Response der ersten Ticketanfrage im PDF-Format wird ein ReRequest an TICKeos geschickt wird, in dem als outputFormat eine PNG für die mobile Darstellung angefordert wird.

Student A hat sein Printticket zu Hause vergessen und loggt sich daher in der Universitäts-App ein, um sich sein Semesterticket anzeigen lassen zu können. Weil Student A zuvor schon ein Semesterticket generiert hat, wurde neben der PDF im Webview, auch eine PNG für die mobile Anzeige generiert, welches durch die Universität an das korrekte Profil verknüpft wurde und durch den Studenten abgerufen werden kann.

6.1.3. KombiTicket – Mobiles Ticket (dynamisch)

Um die Studententickets mit dynamischen Sicherheitsmerkmalen abbilden zu können, wird der eos.ticketManager in die Universitäts-App eingebunden und ein neues mobiles Template im TICKeos-Backend eingerichtet.

Nachdem sich Student A zum Semesterwechsel ein App-Update durchführt, lässt er sich ein Semesterticket über die KombiTicket-Schnittstelle generieren. Im entsprechenden Menüpunkt

der App wird auf die Library des eos.ticketManagers zugegriffen, wo das korrekte Mobile-Ticket heruntergeladen und aufgerufen werden kann. Dabei werden alle entsprechenden Sicherheitsmerkmale (siehe 4.3.2) korrekt abgebildet.

6.2. Variante 2: Mobile Library / Online-Shop

Student B ist an der Hochschule Y immatrikuliert, die ihren Studenten eine Webseite mit allgemeinen Informationen und Services bereitstellt. Das persönliche Portal ist geplant, wird jedoch aus strukturellen Gründen, vorerst nicht umgesetzt. Analog zur Webseite verfügt die Hochschule über eine eigene App mit allgemeinen Informationen.

Man will die Verwaltung der Studententickets digitalisieren aber nicht zu viele Ressourcen für eine Implementierung aufwenden. Es soll daher ein Web- & Mobile-Shop durch eos.uptrade eingesetzt werden. Da die Hochschule keinen eigenen Webservice für die Immatrikulationsprüfung bereitstellen kann, werden aktuelle Listen in Form einer CSV-Datei zu jedem neuen Semester bereitgestellt und in das TICKeos Backend durch den Support bei eos.uptrade importiert.

6.2.1. Online-Shop – Print Ticket (statisch)

Durch einen Hinweis auf der Webseite der Hochschule erfährt Student B dass er sich sein Semesterticket nun auch ausdrucken lassen kann. Hierfür klickt er auf einen entsprechenden Link auf der Webseite und gelangt auf die von eos.uptrade gehostete VRR-Plattform, wo die entsprechende Hochschule ausgewählt werden kann.

Nach Auswahl der korrekten Hochschule wird das entsprechende Semester ausgewählt und die personalisierte Immatrikulationsnummer des Studenten eingegeben. Nach Prüfung des Warenkorbs wird der Benutzer nun aufgefordert, sich neu zu registrieren oder anzumelden, sofern eine Registrierung in der Vergangenheit bereits stattgefunden hat. Dabei wird die eingegebene Immatrikulationsnummer mit der zuvor importierten CSV-Datei der Hochschule abgeglichen. Verläuft das Matching erfolgreich, erhält der Benutzer eine Bestätigung mit der Option, sich das personalisierte Ticket direkt als PDF zu generieren und herunterladen zu können.

6.2.2. Mobile-Shop – Mobiles Ticket (dynamisch)

Da der Student B sein Printticket vergessen hat, öffnet er die App seiner Hochschule, wo er die Option „Semesterticket“-Auswahl und in die Mobile-Shop Library gelangt. Da der Student sich zuvor schon einmal registriert hat und sich sein Semesterticket bereits generiert hat, kann er sich mit seinen Zugangsdaten einloggen. Nach erfolgreichem Login landet Student B auf der

Startseite des Shops, wo er sein gültiges Ticket entdeckt. Nach Öffnen des Tickets stellt er fest, dass das Ticket über ein anderes Layout verfügt und neben den typischen Merkmalen wie persönliche Daten und dem Barcode auch dynamische Sicherheitsmerkmale aufweist.

7. Validierung

Um die Gültigkeit einer Fahrtberechtigung validieren zu können, kann mit der entsprechenden Hard- und/oder Software der Barcode ausgelesen werden, der neben den abgebildeten Inhalten im Ticket auch weitere Informationen über die Berechtigung enthalten kann.

Darüber hinaus gibt es im Ticket mehrere Sicherheitsmerkmale, die bei der Sichtkontrolle geprüft werden können. Diese können aus den folgenden Elementen bestehen:

- Hintergrundgrafik aus personenbezogenen Daten und Ticket-Parametern

- Gespiegelte Ticketnummer

- Fingerprint, bestehend aus definierten Inhalten des Tickets (z.B. zweiter Buchstabe vom Namen, zufälliges Zeichen, 4. Stelle der Ticket-ID)

- Dynamische Elemente
    - Gyroskopisches Element (Grafik die auf die Neigung des Geräts reagiert)
    - Animationen, wie Laufbalken etc.
    - Animierte Uhr
- Kundenfoto

8. Tabellarische Gegenüberstellung

Eigenschaften	KombiTicket		Mobile-Shop (Lib.)
	Schnittstelle	eos.ticketManager	
Kundenaccount/ Frontend	✗	✗	✓
Immatrikulationsprüfung	✗	✗	✓
Statistiken	✓	✓	✓
Dynamisches Ticket-Layout	✗	✓	✓
Ticket-Format	PNG/ PDF	JSON	JSON/ PDF[1]
Aufwand	$	$$	$$$

9. Schnittstelleninformationen

9.1. Variante 1: Kombi-Ticket-Schnittstelle

Für die Kombi-Ticket-Variante erhält die Hochschule Zugangsdaten für die Kommunikation mit der TICKeos Kombi-Ticket-Schnittstelle des VRR. Die Kommunikation kann mit externen Applikationen via XML erfolgen.

Je nachdem ob der Barcode VDV Fähig sein soll, müssen die entsprechenden Parameter mit dem Request übergeben werden. Im System muss gemäß den Vorgaben ein entsprechendes Produkt konfiguriert werden, damit die übermittelten Parameter korrekt verarbeitet werden können.

Nachfolgend wird ein Beispiel-Request, sowie der dazugehörige Response abgebildet.

9.1.1. Request (Kombi-Ticket)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:kom="https://tickeos.de/service.php/kombiticket/1_5">
  <soapenv:Header/>
  <soapenv:Body>
     <kom:generate>
        <authToken>XXXXX</authToken>
        <systemID>Testsystem</systemID>
        <organizerID>Universitaet_A</organizerID>
```

[1] Ticketausgabe im PDF-Format nur über Online-Shop (nicht über App)

```
        <eventID>Sommersemester2018</eventID>
        <ticketID>123456</ticketID>
        <productID>produktbezeichnung_semesterticket</productID>
        <passengerName>Max</passengerName>
        <passengerSurname>MusterStudent</passengerSurname>
        <passengerBirthdate>21.05.1998</passengerBirthdate>
        <outputFormat>png | pdf</outputFormat>
        <templateID>templatename</templateID>
        <assign>
            <name>salutation</name>
               <value>Herr | Frau</value>
               <type>string</type>
   </assign>
     </kom:generate>
  </soapenv:Body>
</soapenv:Envelope>
```

9.1.2. Response (Kombi-Ticket)

Der Response enthält neben dem Status in Form einer Beschreibung und einem Code auch die relevante Fahrtberechtigung im Base64-Format. Zusätzlich werden sowohl eine externe, als auch eine vom System generierte Ticket-ID zurückgesendet, die auch auf der Fahrtberechtigung abgebildet werden kann.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="https://tickeos.de/service.php/kombiticket/">
   <SOAP-ENV:Body>
      <ns1:generateResponse>
         <returnCode>0</returnCode>
         <returnValue>OK: Ticket erfolgreich generiert. (4.407
s)</returnValue>
         <ticketID>123456</ticketID>
         <internalEventID>24</internalEventID>
         <internalOrganizerID>16</internalOrganizerID>
         <ticketData>BASE64-Code</ticketData>
         <internalTicketID>91806249</internalTicketID>
         <sessionID/>
      </ns1:generateResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

9.2. Variante 2: Mobile-Shop API

Bei dieser Variante wird der Mobile-Shop als Bibliothek in eine Fremd-App integriert oder als Standalone-App ausgeliefert. Dabei gelangt man z.B. über einen bestimmten Menüpunkt auf

die Startseite des Mobile-Shops, wo man sich registrieren/einloggen und die gültigen Fahrtberechtigungen einsehen kann.

Die Schritte zur Implementierung der TICKeos Mobile-Shop Bibliothek sind in der TICKeos Mobile-Shop Dokumentation beschrieben. Diese Dokumentation wird bei der Auslieferung der angeforderten Bibliothek beigelegt.

10. Ticket-Templates

Es können neben dem VDV-Einheitslayout verschiedene weitere Ticket-Templates verwendet werden, wobei der Auftraggeber frei entscheiden kann, welche Elemente an welcher Stelle abgebildet werden sollen. Wesentlicher Unterschied ist dabei das Ausgabeformat, das unterschiedliche Merkmale/Elemente zulässt (statisch/dynamisch) und primär der Sichtprüfung dient.

Ticket-Templates werden im TICKeos Hintergrundsystem als JSON-Datei hinterlegt. Die darin abgebildeten Parameter und Grafiken können fest am Template hinterlegt aber auch dynamisch befüllt werden: Z.B. im Falle einer Ticket-Personalisierung oder wenn ein Ticket-Template bei verschiedenen Veranstaltern angewandt wird und je Veranstalter ein abweichendes Logo auf dem Ticket abgebildet werden soll. Die Generierung der Fahrtberechtigung erfolgt im TICKeos-Hintergrundsystem. Von dort aus erfolgt je Kanal die entsprechende Bereitstellung.

10.1. Kombi-Ticket

Bei der Kombi-Ticket-Variante werden nur statische Fahrtberechtigungen ausgegeben, die als Printticket oder als Grafik ausgegeben werden.

Im nachfolgenden werden zwei beispielhafte Templates abgebildet, wie sie beim VRR eingesetzt werden. Wird im Request in der Methode <Format> der Wert „PDF“ verwendet, so wird ein entsprechendes PDF Printticket im Response zurückgegeben. In dem nachfolgenden Beispiel wird der Benutzer aufgefordert, das Ticket entlang der gestrichelten Linie auszuschneiden und in der Mitte zu falten. Das Ticket hat dann eine praktische Checkkartengröße und passt in jede Tasche:

Abbildung 8: Print-Ticket

Mit dem Wert „PNG“ erhält man eine Fahrtberechtigung im PNG-Format, welches in einer externen App eingebunden und entsprechend abgebildet werden kann.

10.2. Mobile-Shop (Lib.)

Im Mobile-Shop werden nur mobile Tickets mit speziellem Layout angeboten. Zur Layout-Definition werden ebenfalls Templates im JSON-Format genutzt. Die Sichtprüfungsmerkmale können dabei individuell konfiguriert werden. Dabei kann das Template neben den einheitlichen, statischen Sicherheitsmerkmalen auch dynamische Elemente enthalten, wie z.B. ein gyroskopisches Element etc.

Wird im TICKeos-Hintergrundsystem das Template oder ein Parameter nachträglich geändert (z.B. der Name), erfolgt bei einer vorhandenen Internetverbindung ein erneuter Download des aktualisierten Tickets.

Abbildung 9: Mobiles Ticket zur Darstellung im Smartphone

11. Benutzerdaten und Datenschutz

Im TICKeos-Vertriebshintergrundsystem kann grundsätzlich zwischen zwei Bereichen unterschieden werden, in denen Benutzerdaten verarbeitet werden:

- **Benutzerdaten**
  Daten die der Endnutzer bei der Registrierung im Online- oder Mobile-Shop angibt und stets verwalten kann. Beim Kauf eines Tickets wird bei der Ticketpersonalisierung, standardmäßig auf die hinterlegten Daten zugegriffen
- **Bestelldaten**
  Persönliche Daten die bei der Personalisierung der erzeugten Fahrtberechtigung abgefragt und ggf. zur Sichtkontrolle auf das Ticket-Template abgebildet werden. Diese müssen nicht immer den Daten des Benutzerkontos entsprechen.

Im Bestellprozess werden Bestelldaten, wie Produkt- und die persönlichen Daten (Anrede, Vor- & Nachname und Geburtsdatum) dupliziert und als „Buchungsbeleg“ abgelegt. Der Bestellverlauf bleibt stets unangetastet, d.h. hier greifen keine Anonymisierungsprozesse. Dies stellt die Nachvollziehbarkeit der Bestellungen sicher. Die entsprechenden Daten können an das angeschlossene Hintergrund-/Buchhaltungssystem exportiert werden.

Auf Wunsch des Endkunden kann das jeweilige Benutzerkonto, anonymisiert werden. Dabei werden alle hinterlegten Kundenparameter, durch eine zufällig generierte Zeichenfolge ersetzt, sodass ein Rückschluss auf die Person im TICKeos-Hintergrundsystem nicht mehr möglich ist.

Anmerkung: Es werden die Daten stets anonymisiert (überschrieben) aber niemals gelöscht. Eine Löschung würde die referentielle Integrität d.h. die Verweise innerhalb der Datenbank auflösen und somit das Datenmodell zerstören. Eine Weiterleitung der Anonymisierungsanfrage an Drittsysteme findet nicht statt.

Für den Zugriff auf das TICKeos-Hintergrundsystem wird immer ein entsprechender Zugang benötigt. Dieser besteht aus einem Benutzernamen und einem Passwort. TICKeos bietet dabei die Möglichkeit, verschiedene Rollen mit dedizierten Berechtigungen je Modul festzulegen (editieren, lesen oder keinen Zugriff). Die Einrichtung der Berechtigungen für Mitarbeiter des VU erfolgt nach Anforderungen des VU bei Initialisierung des Systems und kann auch zu einem späteren Zeitpunkt vorgenommen bzw. geändert werden. Zudem kann jeder Backend-Zugang zeitlich befristet werden, woraufhin der Account nach Erreichen der Frist automatisch deaktiviert wird. Werden im Hintergrundsystem Änderungen durchgeführt, so werden diese geloggt und können jederzeit nachvollzogen werden.

Auf Wunsch kann das Hintergrundsystem mit einem zusätzlichen htaccess-Schutz oder eine IP-Beschränkung ausgestattet werden.