

# Smart Plug

Entwicklung einer Webapplikation für das Aufladen von E-Mobilen

Vorgelegt von Ali Kemal Özkan

**Betreuer:** M. Sc. Adina Aniculaesei

Institut für Informatik – Software Systems Engineering  
Prof. Dr. Rausch

06.11.2013

# Inhaltsverzeichnis

## Inhaltsverzeichnis

<b>1. Einleitung</b> .....	<b>3</b>
<b>2. Ausgangssituation und Zielsetzung</b> .....	<b>3</b>
<b>3. Überblick auf die Architektur des Applikations</b> .....	<b>4</b>
<b>4. Schnittstellen</b> .....	<b>4</b>
4.1 Smartplug – Webserver Kommunikation.....	5
<b>5. Anforderungen eines Elektronischen Zahlungssystem</b> .....	<b>7</b>
5.1 Allgemeine Anforderungen.....	7
5.1.1 Sicherheit.....	8
5.1.2 Skalierbarkeit.....	8
5.1.3 Akzeptanz.....	8
5.1.4 Effizienz.....	8
5.2 Spezielle Anforderungen.....	9
5.2.1 Peer-to-Peer (P2P)Zahlungen.....	9
5.2.2 BEWEISBARKEIT.....	9
5.2.3 ANONYMITÄT.....	9
5.2.4 OFFLINE ZAHLUNG.....	9
5.2.5 BEDIENBARKEIT.....	9
<b>6. Schnittstelle zum Online-Zahlungssystem</b> .....	<b>9</b>
<b>7. Paypal</b> .....	<b>10</b>
7.1 Zahlungsmöglichkeiten.....	10
7.1.1 Zahlung mit der Kreditkarte.....	11
7.1.2 ZAHLUNG MIT LASTSCHRIFT.....	11
7.1.3 ONLINE ÜBERWEISUNG.....	11
7.1.4 ZAHLUNG MIT GUTHABEN AUF DEM PAYPALKONTO.....	11
7.2 API EINRICHTEN.....	12
7.2.1 API-SIGNATUR.....	12
7.2.2 API-ZERTIFIKAT.....	12
7.3 API BENUTZEN.....	12
7.3.1 NAME-VALUE-PAIR: .....	12
7.3.2 SOAP ( SIMPLE OBJECT ACCESS PROTOCOL ): .....	13
<b>8. Fazit</b> .....	<b>14</b>
<b>9. Abbildungsverzeichnis</b> .....	<b>15</b>
<b>10. Literaturverzeichnis</b> .....	<b>15</b>
<b>11. Tabellenverzeichnis</b> .....	<b>15</b>

# 1. Einleitung

Diejenigen die ein Auto besitzen wissen dass Petroleumpreise eine große Rolle in unserem Alltag spielen. Weil bisher keine andere Energiequelle Petroleum ersetzen konnte haben die Marktführer die Preise sehr hoch aufgeführt. Das hat die Menschen auf die Suche nach einigen anderen billigeren Energiequellen gebracht. Heutzutage können wir Strom auch als ein Treibstoff für unsere Fahrzeuge benutzen.

Stromtanken dauert leider wesentlich länger als Petroleum zu tanken. Strombetriebene Fahrzeuge können nicht mit eine Tankfüllung dieselbe Strecke wie ein Petroleumbetriebenes Fahrzeug machen. Wegen den hohen Ladezeiten muss man den E-Fahrzeugnutzer die Möglichkeit anbieten sein Fahrzeug überall da auf zu tanken wo er sein Fahrzeug abstellt. Damit er nicht auf die komplette Ladung warten muss.

Smartplug ermöglicht genau das, dass die E-Fahrzeugbenutzer ihre Fahrzeuge auf beliebigen Orten aufladen können wobei sie sich um andere Sachen kümmern.

## 2. Ausgangssituation und Zielsetzung

Vor einer erfolgreichen Markteinführung von E-Mobilen in Deutschland und anderen Ländern muss die zugehörige Infrastruktur in Form von mehreren Millionen Stromstellen zum Aufladen der E-Mobil-Batterien allein in Deutschland geschaffen werden.

Das hier vorgestellte SmartPlug Projekt wird in Zusammenarbeit mit der INENSUS GmbH in Goslar bearbeitet. Es hat zum Ziel, die kostenintensive Elektronik einer solchen Infrastruktur nicht in die Steckdose, sondern in den Stecker zu integrieren, so dass selbst mit wenigen E-Mobilen ein flächendeckendes Stromstellennetz schnell und kostengünstig zur Verfügung gestellt werden kann.

Die Stromstelle selbst besteht dann lediglich aus einem RFID-Tag, der im Haushalts- und Industriebereich auf einer konventionellen und im öffentlichen Bereich auf einer leicht modifizierten Steckdose mit integriertem Schalter aufgeklebt wird. Auf diese Weise wird eine finanzierbare flächendeckende Einführung der Ladeinfrastruktur für E-Mobile überhaupt erst möglich. Durch das Konzept kann jeder für ein Kleingeld zum Stromstellenbetreiber und Stromverkäufer werden und an der Elektromobilitäts-Bewegung partizipieren – selbst ohne E-Mobil.

Im Rahmen des SmartPlug Projektes wurde ein fortgeschrittenes Projekt ausgeschrieben. Dies soll eine Lösung für die sichere und unkomplizierte Abrechnung zwischen Stromstellenanbieter und Stromstellennutzer erarbeiten und realisieren. Insbesondere die sichere Kommunikation zwischen Stecker und Server-Infrastruktur stellt hierbei eine besondere Herausforderung dar

### 3. Überblick auf die Architektur des Applikations

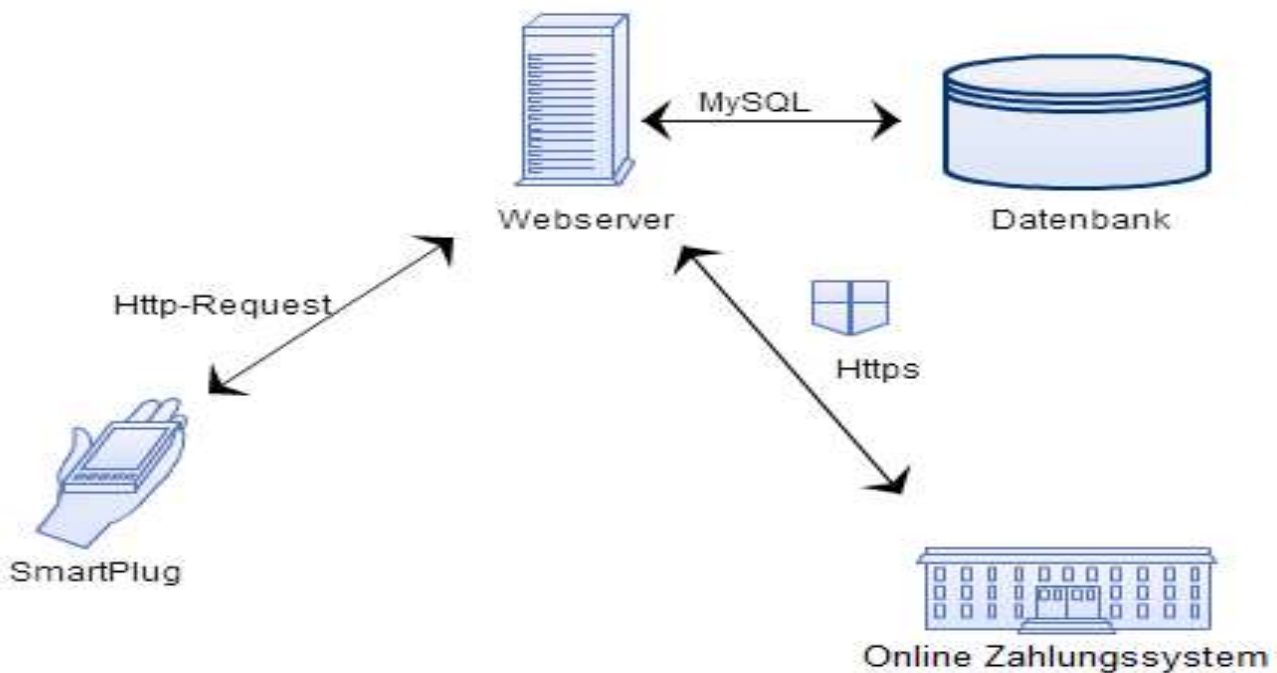


Abbildung1. Gesamtsystem überblick

Das Gesamtsystem läuft wie folgend ab.

1. Der Smartplug Kommuniziert mit dem Webserver über http-Requesten.
2. Der Webserver ruft die Datenbank mittels MySQL ab und sucht die passenden Datensätzen.
3. Nachdem der Server ein/e Rückgabewert/e von die Datenbank bekommen hat gibt er die aus und die Daten werden vom Smartplug gelesen.
4. Wenn ein Kunde seine Rechnung/en Zahlen möchte wird er mittels Https zum verwendeten E-Paymentsystem weitergeleitet. Da wird der Zahlungsablauf abgeschlossen.

## 4. Schnittstellen

### 4.1 Smartplug – Webserver Kommunikation

Die Verbindung zwischen dem Webserver und dem Smartplug erfolgt über HTTP-Requesten. Der Smartplug sendet an den Webserver Informationen wie folgend;

Der Smartplug ruft die dazugehörig Internetseite auf. Diese Adresse betätigt die zugehörig Funktion mit den übergebenden Variablen.

## 4.1.1 SmartPlug Registrieren

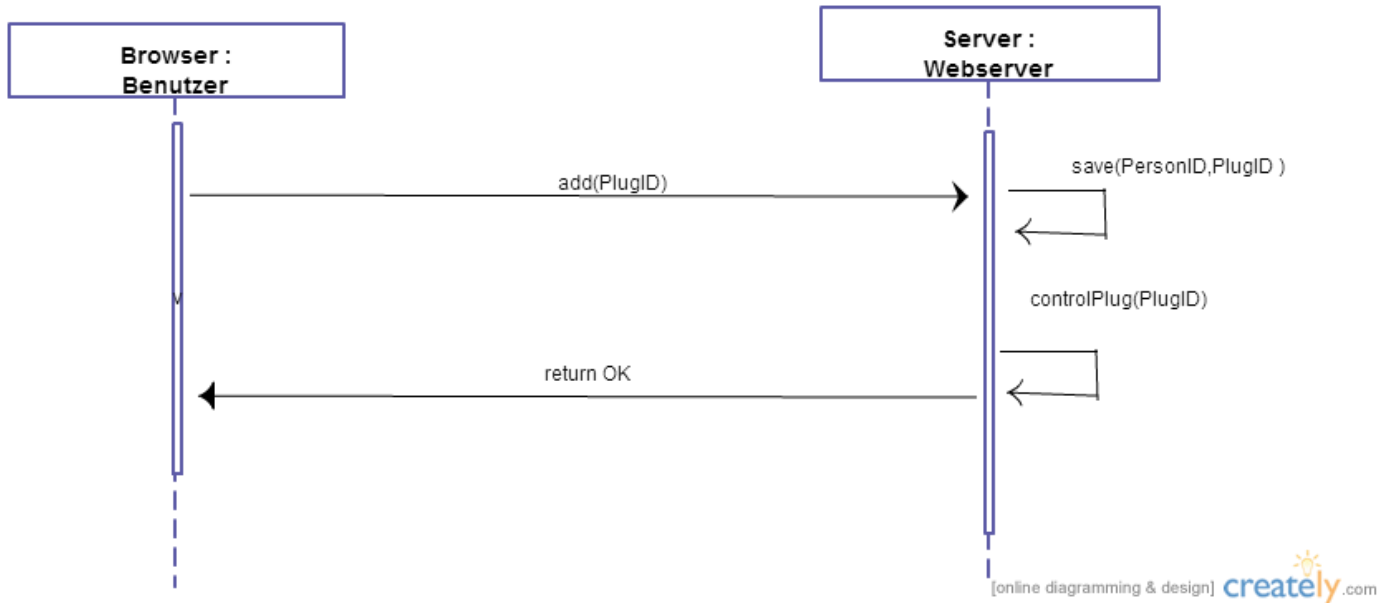


Abbildung2. Smartplug Resgistrieren

Im oben beschriebenen Sequenz-Diagramm legt sich ein bereits eingeloggter Nutzer einen Smartplug-Gerät mit dem folgende Schritten zu.

1. Der eingeloggter Nutzer betätigt „Stecker Verwaltung“ im Menüleiste.
2. Wenn der Nutzer eine Firma besitzt, wird entschieden ob das Gerät zu die Firma oder zum Privaten Eigentum hin gefügt wird.
3. Nachdem der Nutzer sich entscheiden wird „+ Neue Stecker Anlegen“ betätigt.
4. Es wird ein Modal-popup geöffnet wo der Nutzer den SteckerID eingeben muss. Also die add Funktion im plug Controller wird mit dem SteckerID aufgerufen.
5. Die dazu gehörige PersonID wird von der geöffneten Session zugefügt und wird zum Plug Model gesendet. Plug Model kontrolliert ob die eingegebene ID einen passenden Status hat.
6. Status passt ( steht zum Kauf ) dann wird „True“ zurück gegeben und der Controller gibt eine Benachrichtigung „Der Stecker wurde erfolgreich zugefügt“ wenn der Status aber nicht passt (entweder existiert die ID nicht oder Status ist „besetzt“ ) dann wird „False“ als Rückgabe wert zurück gegeben und der Nutzer bekommt eine Fehlermeldung „ Die eingegebene ist falsch bitte Kontrollieren sie die ID „.

## 4.1.2 SmartPlug Login

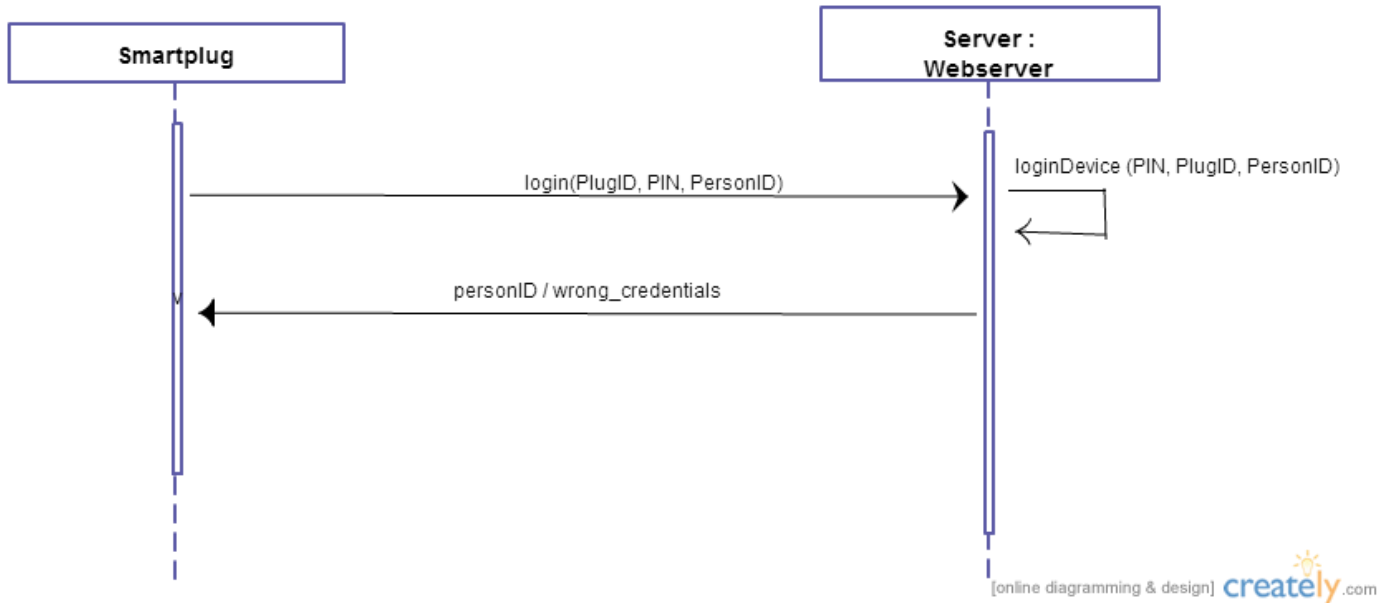


Abbildung3. Login

Im oben beschriebenen Sequenz-Diagramm loggt sich ein bereits eingetränktes Smartplug-Gerät mit folgende Schritte ein.

1. Das Smartplug-Gerät sendet seine eindeutige ID, den Benutzer angebundene PIN und die PersonID des Smartplug-Besitzer.
2. Der Webserver gibt die enthaltenen Daten unter „Plug\_model“ zu die Funktion `loginDevice` weiter.
3. „Plug\_model“ kontrolliert ob es ein Gerät mit den eingekommenen Datenpaaren existiert.
4. Wenn die Daten passen wird die dazu gehörige Datenbankzeile zurück zum Controller gegeben. Der Controller gibt dann als Rückgabewert die PersonID zurück. Wenn aber keine Datenbankzeile für die Werte gefunden werden, wird ein „False“ zum Controller zurückgegeben. Der Controller gibt dann „wrong\_credentials“ als Fehlermeldung zum Smartplug aus.

### 4.1.3 SmarPlug ChargeProcess

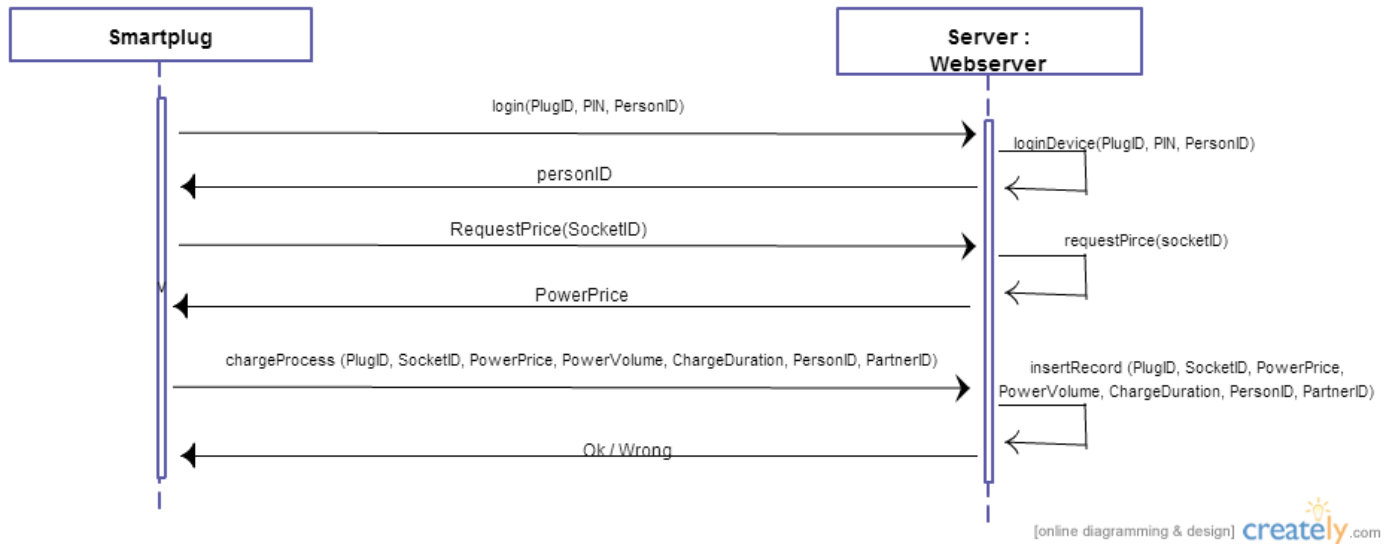


Abbildung4. Ladevorgang

Im oben beschriebenen Sequenz-Diagramm wird ein Ladungsvorgang beschrieben. Der Ladevorgang erfolgt durch folgende Schritte.

1. Nach dem sich der Nutzer mit dem Smartplug eingeloggt hat wird „Aufladen“ im Menü des SmartPlugs betätigt.
2. Danach wird eine Preisabfrage mit dem entsprechenden SteckerID (RfID) zum Webserver gesendet. Der Webserver gibt den zu hörigen Preis zu dem Stecker zurück.
3. Der Smartplug Nutzer kann danach den Ladevorgang starten.
4. Wenn der Nutzer die Ladung beendet werden die Daten zum Webserver gesendet und da gespeichert.

## 5. Anforderungen eines Elektronischen Zahlungssystem

Weil das Projekt Smartplug wie keine Tankstellenkette sein wird, wird es für die Kunden auch nicht möglich sein direkt nach der Tankung an die Kasse zu gehen und zu bezahlen. Dafür wird eine alternative Lösung benötigt. Da heutzutage fast alle Zahlungsmöglichkeiten online angeboten werden, ist die beste Lösung/Alternative ein Online-Zahlungssystem einzuführen.

Bei Elektronischen-Zahlensystemen gibt es mehrere Anforderungen die in zwei Basiskategorien aufgeteilt sind. Zum einen sind es die Allgemeinen Anforderungen die vorausgesetzt sind und zum anderen die Speziellen Anforderungen die für ein bestimmtes Projekt anwendbar sind.

### 5.1 Allgemeine Anforderungen

Die allgemeinen Anforderungen sind wie folgt.

### 5.1.1 Sicherheit

Der Zahlungsablauf von elektronischen Zahlungssystemen muss auch sicher gestellt sein. Kein Unberechtigter soll die Daten sehen / ändern können.

Die Elektronischen Zahlungssysteme müssen auch Datenverlust gesichert sein. Kein Datenverlust ( Hardwareprobleme oder Datendiebstahl ) darf nicht zum Vollständigen Verlust der Zahlungsmittel führen. Das System muss regelmäßig Sicherheitskopien erstellen und bei Bedarf muss es sich auch zurücksetzen können (Recovery-Funktion)

### 5.1.2 Skalierbarkeit

Elektronische Zahlungssysteme sollten so aufgebaut sein, dass das System für beliebige Nutzern erweiterbar ist und dass es von allen Nutzer gleichzeitig benutzt werden kann.

### 5.1.3 Akzeptanz

Die Akzeptanz des elektronischen Zahlungssystems hängt von allen beteiligten Rollen (Bank, Kunde, Händler) ab. Der Erfolg von elektronischen Zahlungssystemen ist mit dem Vertrauen der Kunden verbunden. Folgende Faktoren sind für den Kunden entscheidend ob das angebotene Zahlungssystem verwendet wird.

1. Sicherheit
2. Verbreitungsgrad
3. Technischen Kompatibilität
4. Bedienbarkeit
5. Rechtssicherheit
6. Kosten

### 5.1.4 Effizienz

Das Verhältnis zwischen den Zahlungsbeträgen und den entstehenden Kosten für den Zahlungsablauf ( Transaktion ) bestimmen die Effizienz des Zahlungssystems. Die Transaktionskosten bestehen aus den Hardwarekosten, Finanziellen Ausgaben und die Zeit bis eine Transaktion durchgeführt wird. Aus diesen Gründen werden die Zahlungssystemen auf Macro- und Micro-Zahlungssystem aufgeteilt. Wobei die Macro-Paymentsysteme für größere Beträge und Micro-Paymentsysteme für kleine Beträge geeignet sind.

	Minimal Betrag	Maximal Betrag
Micro-Zahlungssystem	0,01€	5€
Macro -Zahlungssystem	5€	500€

Tabelle1. Macro- und Microzahlungsbeträge



## **5.2 Spezielle Anforderungen**

Das elektronische Zahlungssystem sollte einige Spezielle Anforderungen erfüllen die in manchen Fällen jedoch nicht gewährleistet sind.

### **5.2.1 Peer-to-Peer (P2P) Zahlungen**

Privatpersonen sollten auch die Möglichkeit haben das elektronische Zahlungssystem untereinander anzuwenden.

### **5.2.2 BEWEISBARKEIT**

Alle durchgelaufenen Transaktionen sollten nachgewiesen werden. Diese Anforderung kann durch digital signierte Quittungen gelöst werden.

### **5.2.3 ANONYMITÄT**

Der Zahlungsvorgang sollte dieselbe Anonymität gewährleisten wie bei Barzahlung. Da es nach den meisten Zahlungsvorgängen eine Lieferadresse benötigt wird, ist eine Anonymität beschränkt darstellbar.

### **5.2.4 OFFLINE ZAHLUNG**

Eine Zahlung sollte auch ohne eine Bankschaltung möglich sein. Dieser Vorgang ist aber aus Sicherheitsgründen schwer umsetzbar. Die Smart Cards sind das einzige elektronische Zahlungssystem dass diese Anforderung erfüllt.

### **5.2.5 BEDIENBARKEIT**

Das Zahlungssystem muss folgende Voraussetzungen erfüllen

1. Plattformunabhängigkeit, muss bei jeder Plattform ausführbar sein
2. Verständlichkeit, das System soll unkompliziert sein.
3. Transparenz, der Nutzer soll den gesamten Überblick nicht verlieren.
4. Transaktionsüberblick, der Nutzer soll seine Transaktionen einfach kontrollieren können.

## **6. Schnittstelle zum Online-Zahlungssystem**

Nach dem die Ladung erfolgreich beendet ist, wird der Betrag in Perioden oder direkt nach der Tankung per Internet bezahlt. Dazu werden folgende Daten von dem Zahlenden benötigt.

1. Email-Adresse
2. Der ausstehende Betrag.

Die Kommunikation erfolgt über HTTPS (HyperText Transfer Protocol Secure). Https-Protokoll wird für die Verschlüsselung von Daten zwischen Webserver und Client benutzt. Ohne die Verschlüsselung werden die Daten nicht geschützt, also Lesbar von Unberechtigten. Diese Systeme werden in dem Webserver implementiert wo der Zahlungsvorgang stattfindet

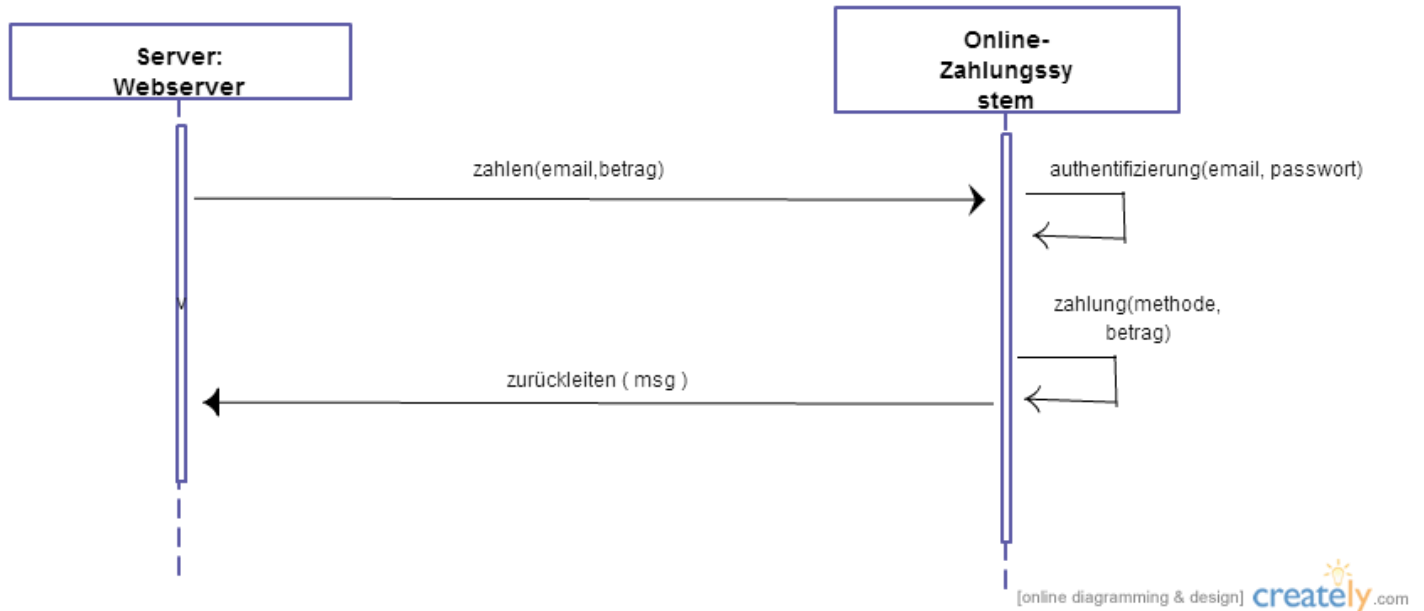


Abbildung5. Kommunikation zwischen dem Webserver und Onlinezahlungssystemen

1. Der Webserver sendet den Betrag und die zugehörige E-Mail Adresse vom Nutzer an dem Onlinezahlungssystem.
2. Da Loggt sich der Nutzer ein und wählt seine Zahlungsmethode (Kreditkarte, Lastschrift, etc.). Nachdem die Zahlungsabwicklung erfolgreich abgeschlossen wird, wird der Nutzer auf die Referenzseite mit eine Nachricht (dass die Zahlung erfolgreich abgeschlossen wurde) zurück geleitet. Falls die Zahlung nicht abgeschlossen werden kann bekommt der Nutzer eine Fehlermeldung die in deren Internetseite angezeigt wird.

## 7. Paypal

### 7.1 Zahlungsmöglichkeiten

Paypal bietet folgende Zahlungsmöglichkeiten an.

- Kreditkarte
- Lastschrift
- Onlineüberweisung ( mittels Giropay )
- Guthaben auf dem Paypalkonto

Bei jeder Zahlungsart muss der Kunde sich bei Paypal anmelden.

### **7.1.1 Zahlung mit der Kreditkarte**

Die Kreditkarte muss erst zu Paypal hinzugefügt werden. Anschließend werden die Kreditkartendaten bzw. die Kreditkarteninhaberdaten, Kreditkartennummer, Verfallsdatum und CVS-nummer (die letzten drei Stellen auf der Rückseite der Kreditkarte) eingegeben. Als nächster Schritt wird die Kreditkarte bestätigt. Darauf folgend werden zwei Beträge von der Kreditkarte abgebucht. Zum einen ist es die Kreditkartengebühr und zum anderen ist es der vier stellige Bestätigungscode den man für die Freischaltung der Kreditkarte benötigt. Der Betrag vom Paypalbestätigungscode wird dem Paypalkonto gutgeschrieben. Nach der Freischaltung kann die Kreditkarte für Zahlungen verwendet werden.

### **7.1.2 ZAHLUNG MIT LASTSCHRIFT**

Um die Zahlung mit Lastschriftverfahren abwickeln zu können müssen folgende Daten eingegeben werden:

- Land
- Vorname
- Geburtsdatum
- Bankleitzahl
- Kontonummer

Anschließend werden zwei Beträge im Cent Bereich auf das angegebene Bankkonto überwiesen. Diese Beträge werden bei Paypal eingegeben, sodass das Konto freigeschaltet wird. Nach der Freischaltung ist möglich den zu bezahlenden Betrag mit dem Lastschriftverfahren ab zu schließen.

### **7.1.3 ONLINE ÜBERWEISUNG**

Wenn die Bank Giropay unterstützt und das Konto für Online-Banking freigeschaltet ist wird die Onlineüberweisung automatisch als Zahlungsmethode bei PayPal angeboten.

Bei der Zahlungsabwicklung leitet Paypal den Zahlenden auf eine sichere, verschlüsselte Seite die in Zusammenarbeit von Giropay und der Bank angeboten wird. Der Kontoinhaber loggt sich mit seiner PIN auf dieser Seite ein. Die Zahlungsdetails werden automatisch übernommen. Um die Zahlung ab zu schließen wird die TAN eingegeben.

### **7.1.4 ZAHLUNG MIT GUTHABEN AUF DEM PAYPALKONTO**

Um dem PayPal Konto Guthaben hinzuzufügen, ist es notwendig eine Überweisung an Paypal mit einem bestimmten Verwendungszweck zu tätigen. Wenn das Guthaben ausreichend ist kann die Zahlung abgeschlossen werden

## 7.2 API EINRICHTEN

Um den API benutzen zu können benötigt DER HAENDLER ein Paypal Konto. Die Voraussetzung für den Zugriff auf den API besteht aus drei Teilen. API-Passwort, API-Benutzername und entweder das Clientseitige API-Zertifikat oder eine API-Signatur.

### 7.2.1 API-SIGNATUR

Das ist der einfachere Weg. Dabei handelt es sich um eine 128 Bit-Signatur, die bei dem Aufruf des APIs verwendet wird für die Autorisierung der Internetseite. Die Signatur ist einfach zu implementieren, weil die Signatur nicht im Server installiert wird stattdessen wird es in den Skriptcode kopiert. Eine API-Signatur wird mit folgenden Schritten erstellt.

1. Nachdem erfolgreich einloggen im Paypal, wird „Mein Profil“ betätigt.
2. In der linken Spalte wird „API-Zugriff“ erscheinen. Diese wird anschließend betätigt.
3. Im darauf folgenden Fenster wird „API-Berechtigungen“ betätigt.
4. In das Option Feld wird „API-Signatur“ ausgewählt.
5. Nachdem durchlesen der API-Nutzungsbedingungen wird diese mit „Ich stimme zu“ bestätigt.
6. Danach wird die API-Signatur, API-Benutzername und das API-Passwort generiert.

### 7.2.2 API-ZERTIFIKAT

Für ein API-Zertifikat sind je nach Serverplattform ein oder mehrere Schritte erforderlich. Bei Smartplug wurde PHP verwendet. Um eine API-Zertifikat für PHP zu erstellen sind folgende Schritte erforderlich:

1. Nachdem erfolgreich einloggen im Paypal, wird „Mein Profil“ betätigt.
2. Darauf folgend wird die „API-Berechtigungen anfordern „betätigt.
3. Die Nutzungsbedingungen werden mit „Ich Stimme zu“ bestätigt. Danach wird der „Senden“ Button betätigt.
4. Im darauf folgenden Fenster werden die API- Zertifikat, Benutzername und Password generiert.
5. Das generierte Zertifikat wird heruntergeladen. Im Bedarf wird das Zertifikat um genannt sodass der Nutzer das Zertifikat auch später erkennen kann.

## 7.3 API BENUTZEN

Nachdem die API erfolgreich eingerichtet ist, ist sie nutzungsbereit. Es stehen zwei Schnittstellen für die API zur Verfügung

### 7.3.1 NAME-VALUE-PAIR:

Bei Name-Value-Pair wird Paypal über einen einfachen HTTP-Aufruf angesprochen. Die Daten werden in einer Zeichenkette mit Variablennamen und dazu gehörigen Werten übergeben.

Die NVP besteht aus dem Benutzernamen, Passwort, den API-Methodennamen und aus einem Token-Wert. Die Zeichenkette sieht wie folgt aus,

```
USER=smart@plug.de&PWD=meinpasswort&METHOD=GetExpressCheckoutDetails&TOKEN=EC-23T233ZP3DFB..
```

Eine erfolgreiche Rückmeldung enthält die Value-Pairs ACK=Success und einen eindeutigen Token-Wert. Eine erfolgreiche Rückmeldung sieht wie folgt aus,

```
ACK=Success&TIMESTAMP=date/timeOfResponse  
&CORRELATIONID=debuggingToken&VERSION=2.300000&BUILD=buildNumber  
&TOKEN=EC-3DJ78083ES565113B&EMAIL=hans@unbekanntefirma.de  
&PAYERID=95HR9CM6D56Q2&PAYERSTATUS=verified  
&FIRSTNAME=Hans&LASTNAME=Guten...
```



Abbildung6. Zahlungsvorgang

### 7.3.2 SOAP ( SIMPLE OBJECT ACCESS PROTOCOL ):

Paypal stellt für PHP, ASP.Net und Java eine eigene SDK (Software Development Kits) zur Verfügung. Mit dem SDKs wird der Zugriff auf Paypal-APIs vereinfacht und die Komplexität von SOAP-Protokolls verringert. Auf der folgenden Adresse stehen die SDKs für verschiedene Plattformen zum Herunterladen.

Link: [http://www.paypalobjects.com/de\\_DE/html/IntegrationCenter/ic\\_sdk-resource.html](http://www.paypalobjects.com/de_DE/html/IntegrationCenter/ic_sdk-resource.html)

#### Installation (PHP):

Die herunter geladene Datei wird in einem temporären Ordner entpackt. Die folgende PHP-Kommandozeile „php install.php“ wird in diesem Verzeichnis aufgerufen. Nachdem die Kommando-Zeile bestätigt wird, wird nach einem Zielverzeichnis nachgefragt. Falls die Datei in

einem Shared-Hosting installiert wird muss das genauere Webverzeichnis angegeben werden. Wenn keine Fehlermeldungen erschienen ist, ist das SDK erfolgreich installiert worden.

## **8. Fazit**

Im Rahmen des Fortgeschrittenenprojekt wurden die Online Zahlungssystemen recherchiert und festgestellt das jedes Zahlungssystem eine andere Schnittstelle anbietet/ benötigt. Der schwierigste Teil war recherchieren das angebotene API von Paypal. Die Schnittstellen waren nicht bekannt aber erfolgreich recherchiert sodass man die einfach implementieren kann.

## 9. Abbildungsverzeichnis

<i>Abbildung1. Gesamtsystem überblick</i>	4
<i>Abbildung2. Smarttplug Resgistrieren</i>	5
<i>Abbildung3. Login</i>	6
<i>Abbildung 4. Ladevorgang</i>	7
<i>Abbildung 5 Kommunikation zwischen dem Webserver und Onlinezahlungssystemen</i>	10
<i>Abbildung 6 Zahlungsvorgang</i>	13

## 10. Literaturverzeichnis

<i>LtV 1 PayOne</i>	
<i>Per Telefon von PayOne bekommen</i>	9

### LtV2 Paypal

[http://www.paypalobjects.com/de\\_DE/html/IntegrationCenter/ic\\_api-reference.html](http://www.paypalobjects.com/de_DE/html/IntegrationCenter/ic_api-reference.html)

### LtV 3 Anforderungen des Online-Zahlungssysteme

*Elektronische Zahlungssysteme Michael Stolt [2004]*

### LtV4 Überblick auf die Architektur des Applikations

*Entwicklung einer Webapplikation für das Aufladen von E-mobilen [2013]*

## 11. Tabellenverzeichnis

### Tabelle1. Macro- und Microzahlungsbeträge

<http://de.wikipedia.org/wiki/Zahlungsverfahren> 8