

Gefördert durch:


 Bundesministerium  
 für Wirtschaft  
 und Technologie

 Bundesministerium  
 für Umwelt, Naturschutz  
 und Reaktorsicherheit

aufgrund eines Beschlusses des Deutschen Bundestages

**Abrechnungs-Interface-Daten-Austausch**

# AIDA Version 1.0

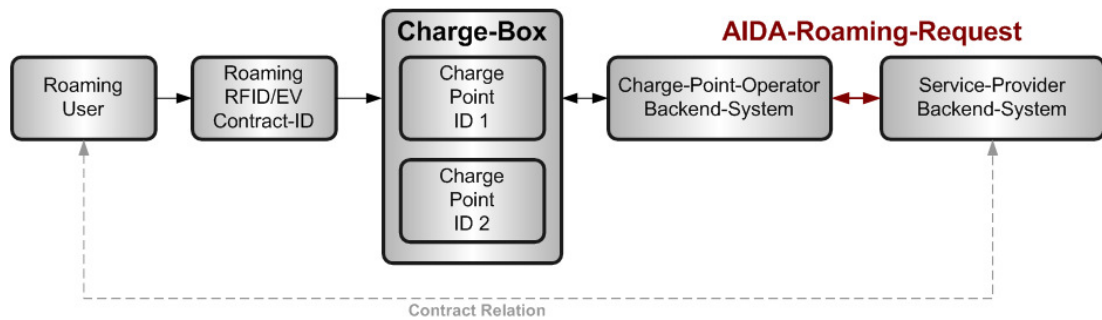
**Task Force**
**Fachgruppe Interoperabilität**

<b>0</b>	<b>ÄNDERUNGSHISTORIE</b>	<b>2</b>
<b>1</b>	<b>EINLEITUNG: ZIELSETZUNG AIDA</b>	<b>3</b>
<b>2</b>	<b>DEFINITIONEN</b>	<b>4</b>
<b>3</b>	<b>ANFORDERUNGEN / USE-CASES</b>	<b>5</b>
<b>3.1</b>	<b>Authentisierung / Autorisierung</b>	<b>5</b>
3.1.1	Annahmen	5
3.1.2	Use-Cases	5
<b>3.2</b>	<b>Abrechnung</b>	<b>6</b>
3.2.1	Annahmen	6
3.2.2	Use-Cases	6
<b>3.3</b>	<b>Sicherheit</b>	<b>6</b>
3.3.1	Transaktionssicherheit	6
3.3.2	Datenschutz	6
<b>4</b>	<b>DATENAUSTAUSCH-MODELL</b>	<b>7</b>
<b>4.1</b>	<b>RFID-Autorisierungsanfrage (AIDA)</b>	<b>7</b>
<b>4.2</b>	<b>RFID-Autorisierungsantwort (AIDA)</b>	<b>8</b>
<b>4.3</b>	<b>Phone-Access-Autorisierungsanfrage (Entwurf)</b>	<b>8</b>
<b>4.4</b>	<b>Abrechnungsdatensatz (AIDA)</b>	<b>9</b>
4.4.1	Einzel-Abrechnungsdatensatz (Push)	9
4.4.2	Bestätigung Einzel-Abrechnungsdatensatz	10
4.4.3	Anfrage erweiterter Massendatensatz	10
4.4.4	Antwort erweiterter Massendatensatz	11
<b>5</b>	<b>PROTOKOLL-KONZEPTION</b>	<b>12</b>
<b>5.1</b>	<b>WSDL-Template</b>	<b>12</b>
<b>5.2</b>	<b>RFID-Übertragungsformat</b>	<b>15</b>
<b>5.3</b>	<b>Sicherheit</b>	<b>15</b>
<b>6</b>	<b>REFERENZEN</b>	<b>16</b>

## 0 **Änderungshistorie**

Version	Datum	Änderung	durch
0.1	22.09.2010	1. Entwurfsfassung	regio iT
0.2	29.09.2010	2. Telefonkonferenz	regio iT
0.3	23.11.2010	Feedback Telefonkonferenz 10.11	regio iT
0.4	10.12.2010	Feedback Telefonkonferenz 30.11	regio iT
0.5	21.12.2010	Feedback Telefonkonferenz 15.12 Abrechnung EVSE Phone Access	regio iT
1.0	04.07.2011	<ul style="list-style-type: none"><li>• Erläuterungen Transaction-ID in Verwendung mit dem EVSE Phone Access</li><li>• Streichung gesetzliche Anforderungen</li></ul>	regio iT

# 1 Einleitung: Zielsetzung AIDA



- AIDA standardisiert ausschließlich die Kommunikation zwischen den Backend-Systemen des Ladepunktbetreibers (Charge-Point-Operator) und des Kundenlieferanten (Service Provider).
- Authentisierung / Autorisierung sowie der korrespondierende Abrechnungsdatenaustausch stehen im Fokus der Definition.
- Weiterhin ist die Zielsetzung zunächst einen sehr schlanken Implementierungsvorschlag zu machen (Ende 2010), der sich auf die elementaren Basisfunktionen beschränkt und interessierten Modellprojekten eine Umsetzung und ausreichende Erprobung noch vor Ablauf des Förderzeitraumes 30.09.2011 ermöglicht.

## 2 Definitionen

Definitionen	Beschreibung
Ladepunktbetreiber Charge-Point-Operator ≙ EVSE Operator	Betreibt eine oder mehrere Ladesäulen.
Ladepunktbetreiberkennung Charge-Point-Operator-ID ≙ EVSE ID	Bestandteil der standardisierten Ladepunktkenung
Ladesäule Electrical Vehicle Supply Equipment (EVSE)	Eine Ladesäule kann einen oder mehrere Ladepunkte logisch adressieren. Ladepunkte können ggf. physikalisch auch auf umliegende Satellitensäulen verteilt sein.
Charge-Box	Kommunikations-Gateway in einer Ladesäule. Ladesäule und Charge-Box werden häufig nicht unterschieden.
Ladepunktkenung Charge-Point-ID	Die standardisierte Ladepunktkenung: [0-9]*\[0-9\]* kennzeichnet den zugehörigen Ladepunktbetreiber sowie <u>seine</u> eindeutige Nummerierung der Ladepunkte. Jede Säule kann mehrere Ladepunkte haben. Die Ladepunktkenung kennzeichnet nicht die Ladesäule.
Kundenlieferant Service-Provider	Vertragspartner des Benutzers / Endkunden und somit Ausgeber der Karten-ID.
Karten-ID Contract-ID	Karten-ID: z.B. DE-AAA-123456-P Der vom RWE eingeführte Begriff der Contract-ID wird von den Teilnehmern als eindeutige Karten-ID verstanden und ist nicht zu verwechseln mit dem Vertragskonto der Abrechnungssysteme, welche ggf. einem Geschäftspartner die mehrere Karten-IDs zuordnen können (Flotte).
RFID-ID	Eindeutige Seriennummer, die vom Kartenhersteller unveränderbar in der Produktion aufgebracht wird.
Flotte Fleet	Einen Vertragspartner = Rechnungsempfänger (Firma, Familien etc.) sind mehrere Benutzer und Karten-IDs zugeordnet.
Benutzer / Endkunde User / Customer	Ein Benutzer ist Mitglied einer Flotte. Einem Benutzer ist eine oder optional mehrere Karten-IDs zugeordnet.
Mobilitätszentrale Backend System ≙ SCADA	Ladesäulen eines Ladepunktbetreibers sind zur Online-Autorisierung der Karten-IDs mit einer Mobilitätszentrale verbunden.
Roaming-Anfrage	Kann eine Mobilitätszentrale eine Autorisierungsanfrage einer mit ihr verbundenen Ladesäule nicht selbständig beantworten, da es sich um eine externe Karten-ID handelt, wird sie beim Karten-Ausgeber eine Roaming-Anfrage stellen.
Session-ID	Die Session-ID dient im AIDA-Konzept dem Autorisierungsprozess im Sinne einer Auftragsnummer, auf die nachfolgende Abrechnungsdaten referenziert werden können. Die Funktion der Session-ID ist abzugrenzen von der Transaction-ID (EVSE Phone Access).
Transaction-ID	In der Verwendung mit dem EVSE Phone Access wird eine Transaction-ID verwendet, die einen wiederholten Anruf eines Kunden als eine wiederholte Autorisierung erkennen lässt, die lediglich dem Öffnen einer verriegelten Schuko-Steckdose dient (Spezialfall für Ladesäulenhersteller mit verriegelten Schuko-Steckdosen).

## 3 Anforderungen / Use-Cases

### 3.1 Authentisierung / Autorisierung

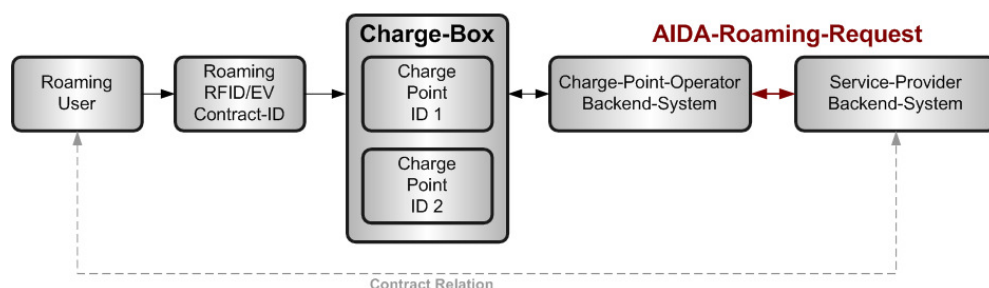
#### 3.1.1 Annahmen

- AIDA fokussiert zunächst auf Authentisierungsverfahren basierend auf RFID-Karten gemäß der technischen Spezifikation der BMWi Modellprojekte.
  - Prinzipiell sind die Roaming-Prozesse aber unabhängig vom gewählten Authentisierungsverfahren.
- Weitere Basis für AIDA ist die Abstimmung der Modellprojekte zum Contract-ID-Nummernkreis (eindeutige Karten-ID) sowie die Spezifikation EVSE Phone Access (Definition einer eindeutigen Ladepunktbezeichnung).
- Zunächst (erste AIDA Version) soll lediglich eine prinzipielle Autorisierung eines Ladevorgangs erfolgen – ohne Differenzierung nach erweiterten Services und Leistungsmerkmalen.
- Zukünftige Versionen könnten auch erweiterte Dienstleistungen oder Leistungsmerkmale berücksichtigen:
  - Berücksichtigung „dynamisches zeitversetztes Laden“
  - Rückspeisung
  - gelieferte Stromqualität
  - Batterienutzung
  - Bonuszahlungen
  - Parkgebühren zur Weiterverrechnung
  - Reservierungen

#### 3.1.2 Use-Cases

Kann eine Mobilitätszentrale eine Autorisierungsanfrage einer mit ihr verbundenen Ladesäule nicht selbständig beantworten, da es sich um eine externe Karten-ID handelt, wird sie beim Karten-Ausgeber eine Roaming-Anfrage stellen. Hierzu sind folgende Schritte erforderlich:

- Autorisierungsanfrage am Ladepunkt (Ladepunktbetreiber) entgegennehmen und zur Verifizierung über die Mobilitätszentrale des Ladepunktbetreibers an die Mobilitätszentrale des Kartenausgebers (Kundenlieferanten) senden.
- Der Ladepunktbetreiber erhält eine Autorisierungsantwort vom Kartenausgeber und gibt den Ladepunkt frei.



#### Erläuterung Grafik:

- Charge Box = Kommunikations-Gateway der Ladestation
- Charge-Point-Operator = Ladepunktbetreiber
- Service Provider = Kundenlieferant = Vertragspartner Endkunde = Kartenausgeber
- Backend Systeme = Mobilitätszentralen

## 3.2 Abrechnung

### 3.2.1 Annahmen

- Grundsätzlich erfordern mindestens zwei Geschäftsvorfälle (Abrechnung) den Datenaustausch:
  - Energieabsatz des Ladepunktbetreibers gegenüber dem Kundenlieferanten bzw. Roaming-Partner
  - Energieabsatz des Kundenlieferanten gegenüber dem Endkunden
  - Für beide Geschäftsvorfälle steht primär die Abrechnung der tatsächlich konsumierten Energie auf Basis der Beistellung im Vordergrund.
- Die Kommunikation erfolgt wiederum ausschließlich zwischen den Backend-Systemen (Mobilitätszentralen) des Ladepunktbetreibers (Charge-Point-Operator) und des Kundenlieferanten (Service Provider).
- Die eigentlichen Geschäftspartner Ladepunktbetreiber und Kundenlieferant erhalten ihre Daten aus den jeweils transferierenden Mobilitätszentralen. Die Schnittstellen der Geschäftspartner zu ihren Mobilitätszentralen sind nicht Gegenstand von AIDA.

### 3.2.2 Use-Cases

- **Einzel-Abrechnungsdatensatz (Push):** Unmittelbar nach Beendigung eines Ladevorgangs sendet die Mobilitätszentrale des Ladepunktbetreibers der korrespondierenden Mobilitätszentrale des relevanten Kundenlieferanten (Roaming-Partner) die Abrechnungsdaten = unmittelbarer Einzel-Abrechnungsdatensatz, der dessen Empfang bestätigt.
- **Erweiterter Massendatensatz (Pull):** Für eine gewünschte Periode fragt ein Kundenlieferant alle ihm zugeordneten Ladevorgänge bei einem Ladepunktbetreiber an. Es handelt sich wiederum um eine Kommunikation zwischen den jeweiligen Mobilitätszentralen.

## 3.3 Sicherheit

### 3.3.1 Transaktionssicherheit

- Wie ist sichergestellt, dass die übertragenen Abrechnungsdaten weder manipuliert noch unvollständig übertragen worden?
  - Das zu verwendende Signaturverfahren muss noch abgestimmt werden.

### 3.3.2 Datenschutz

- Die gesetzlichen Datenschutzbestimmungen zur Erbringung von IT-Dienstleistungen schreiben u.a. vor, dass bei einer Verarbeitung personenbezogener Daten im Auftrag (sog. „Auftragsdatenverarbeitung“) geregelt sein muss, wer wann berechtigt ist zur Löschung der Daten.
- Im Kontext der Elektromobilität wäre z.B. zu prüfen, ob bereits die alleinige Verwendung der Contract-ID (Karten-ID) als personenbezogene Datenverarbeitung einzustufen ist?

## 4 Datenaustausch-Modell

- IT-Service-Provider betreiben ggf. stellvertretend für mehrere Ladepunktbetreiber eine Mobilitätszentrale. Somit ist nicht die stellvertretende Mobilitätszentrale der tatsächliche Ladepunktbetreiber, der eine Roaming-Anfrage stellt. Folglich müssen die relevanten Absender-Informationen als Datenfelder innerhalb der Anfrage verarbeitet werden und können nicht aus dem Absender der Nachricht selbst abgeleitet werden.

### 4.1 RFID-Autorisierungsanfrage (AIDA)



Abbildung 1: RFID-Autorisierung (AIDA) + Abrechnungsdatenaustausch (AIDA)

#### RoamingAuthorization.req

Feldname	Typ	Kardinalität	Beschreibung
contractId	string	1..1	Contract-ID bzw. Karten-ID (DE-AAA-123456-P)
serialNumber	string	1..1	Seriennummer der RFID-Karte
signature	string	1..1	Signaturwert gemäß Modell-Projekte-Spezifikation
operatorId	string	1..1	standardisierte Charge-Point-Operator-ID (Ladepunktbetreiberkennung) = Absender der Roaming-Anfrage abgeleitet aus dem Präfix der Charge-Point-ID: [0-9]*^[0-9\]* gemäß Modell-Projekte-Spezifikation

## 4.2 RFID-Autorisierungsantwort (AIDA)

### RoamingAuthorization.resp

Feldname	Typ	Kardinalität	Beschreibung
status	integer	1..1	0 = "Denied" 1 = "Accepted"
statusMessage	string	0..1	Beschreibung/Ablehnungsgrund
sessionId	integer	1..1	Session-ID zur Referenzierung für späteres Billing
info	string	0..1	Optional: generisches Textfeld für Zusatzinfos

## 4.3 Phone-Access-Autorisierungsanfrage (Entwurf)

- In diesem Fall wird zuerst der EVSE Phone Access Server des Service Providers bzw. Kundenlieferanten adressiert.
- Entweder übermittelt dieser in seiner Antwort an den Charge Point Operator bzw. Ladestationsbetreiber neben der Transaction-ID der EVSE Phone Access Spezifikation nun auch die gemäß AIDA-Spezifikation verwendete Contract-ID und Session-ID oder man setzt die Transaction-ID und Session-ID gleich und verzichtet auch auf die Übermittlung der Contract ID.
- Es gilt zu berücksichtigen, dass einige Ladestationsbetreiber Ladesäulen verwenden, die eine Fernfreischaltung durch eine simulierte RFID-Autorisierung realisieren. Ohne Übermittlung der Contract-ID durch einen Phone Access Server, muss der Ladestationsbetreiber für eine Fernfreischaltung eine spezifische gültige Pseudo-Contract-ID für den Roaming-Partner verwenden und somit auf der Pseudo-Contract-ID alle korrespondierenden Ladevorgänge des Roaming-Partners buchen.

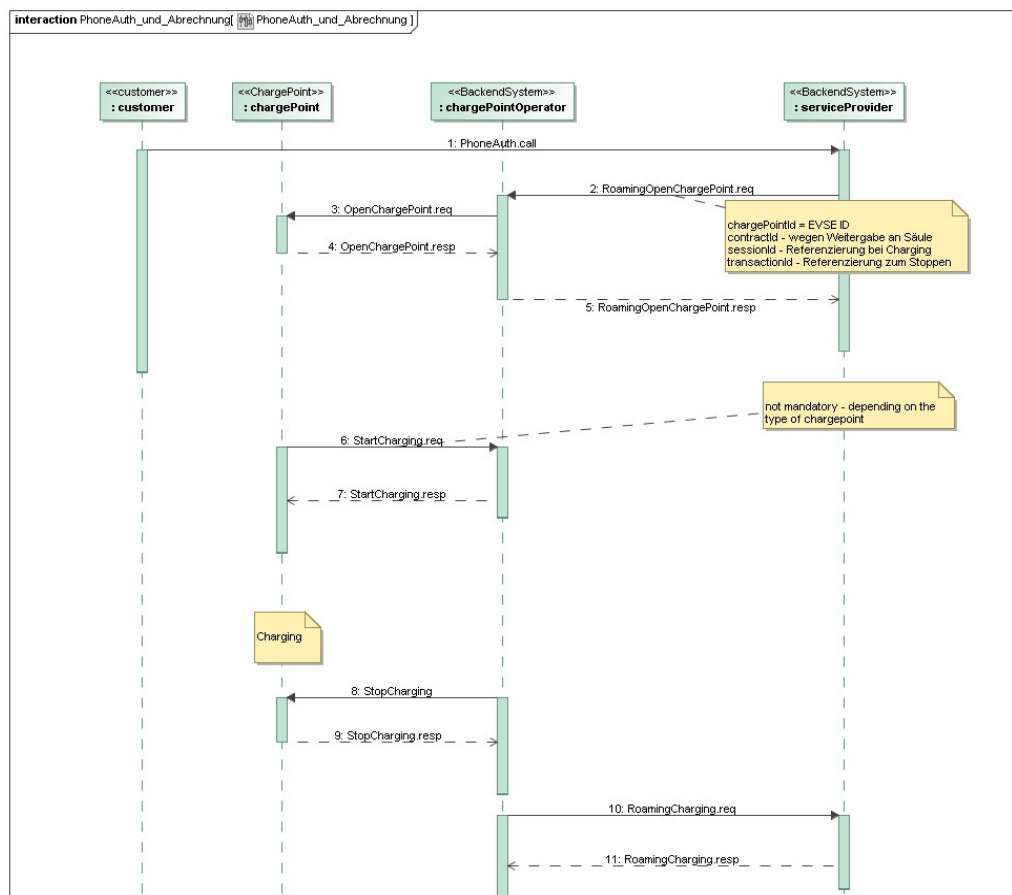
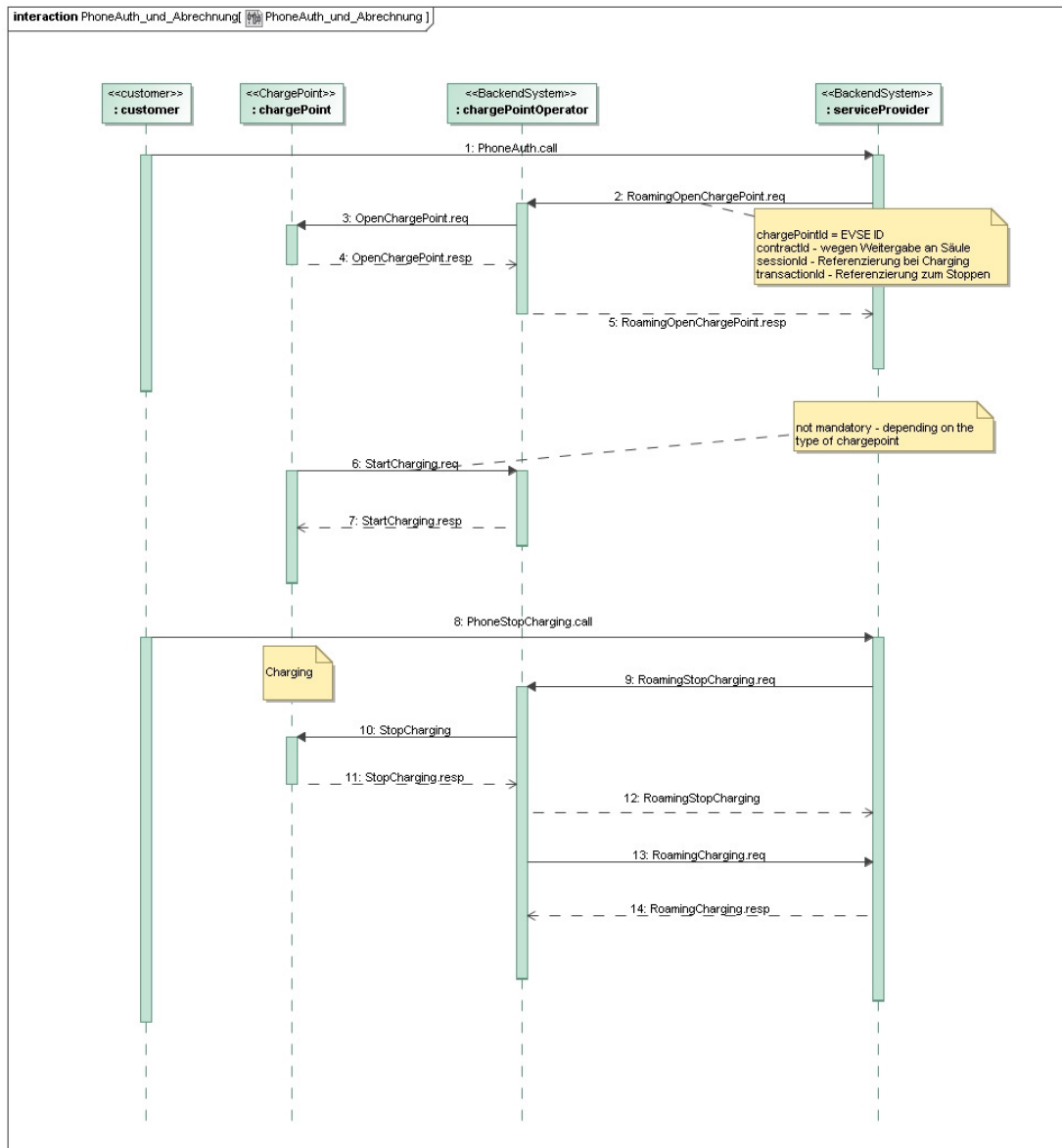


Abbildung 2: Phone-Access-Autorisierung mit erweiterter Session-ID und Contract-ID + Abrechnungsdatenaustausch (AIDA)



**Abbildung 3: Phone-Access-Autorisierung mit sicher wiederholender Autorisierung zur Beendigung des Ladevorgangs + Abrechnungsdatenaustausch (AIDA)**

## 4.4 Abrechnungsdatensatz (AIDA)

- Abstimmung 30.11.2010: Ladestationsbetreiber von Ladesäulen, die Roaming-Kunden automatisiert in ihre lokalen Whitelists (ID-Cache) aufnehmen, um so ihre Offline-Funktionsbereitschaft im Falle von Netzstörungen oder Ausfällen der Mobilitätszentralen sicherzustellen, tragen das Betreiberrisiko, dass eine nachträglich erfolgte Autorisierungsanfrage abgelehnt wird. Folglich können Abrechnungsdaten nur in Verbindung mit einer erhaltenen sessionId weitergeleitet werden. Entsprechende Vereinbarungen sind im Roaming-Agreement niederzulegen.

### 4.4.1 Einzel-Abrechnungsdatensatz (Push)

Unmittelbar nach Beendigung eines Ladevorgangs sendet die Mobilitätszentrale des Ladepunktbetreibers der korrespondierenden Mobilitätszentrale des relevanten Kundenlieferanten (Roaming-Partner) die Abrechnungsdaten = unmittelbarer Einzel-Abrechnungsdatensatz, der dessen Empfang bestätigt:

### RoamingCharging.req

Feldname	Typ	Kardinalität	Beschreibung
chargePointId	string	1..1	Charge-Point-ID: [0-9]*\[0-9\]* gemäß Modell-Projekte-Spezifikation = Absender und Ladepunkt des Abrechnungsdatensatzes
contractId	string	1..1	Contract-ID bzw. Karten-ID (DE-AAA-123456-P)
sessionId	string	1..1	Session-ID = Zuordnung zum Endkunden (sofern eine Autorisierung zuvor erfolgte – siehe hierzu vorausgehende Diskussion)
chargeAmountStart	integer	1..1	in kWh*1000
chargeAmountStartSignature	string	0..1	Optional- sofern Zählerwerte signiert werden.
chargeAmountEnd	integer	1..1	in kWh*1000
chargeAmountEndSignature	string	0..1	Optional- sofern Zählerwerte signiert werden.
chargeEndTime	timestamp	1..1	Zeitstempel des Ladeendzeitpunktes oder zumindest alternativ Empfangszeitpunkt von Ladesäule
chargeStartTime	timestamp	0..1	Zeitstempel des Ladestartzeitpunktes (nicht Bestandteil aller Ladesäulenprotokolle)
chargePointStreet	string	0..1	Adressauflösung der Ladepunktkenung gemäß Charge-Point-ID
chargePointHouseNumber	string	0..1	
chargePointPostcode	string	0..1	
chargePointCity	string	0..1	

Anmerkung: Die sessionId ist nicht zu verwechseln mit der “Transaction ID” der Spezifikation “EVSE Phone Access”, die eine sich wiederholende Autorisierungsfrage kennzeichnet.

#### 4.4.2 Bestätigung Einzel-Abrechnungsdatensatz

### RoamingCharging.resp

Feldname	Typ	Kardinalität	Beschreibung
status	integer	1..1	0 = “Denied” 1 = “Accepted”
statusMessage	string	0..1	Beschreibung Ablehnungsgrund
price	integer	0..1	optional berechneter Preis in Cent des Kundenlieferanten zur Anzeige an Säulendisplay

#### 4.4.3 Anfrage erweiterter Massendatensatz

Für eine gewünschte Periode fragt ein Kundenlieferant alle ihm zugeordneten Ladevorgänge bei einem Ladepunktbetreiber an (Pull). Es handelt sich wiederum um eine Kommunikation zwischen den jeweiligen Mobilitätszentralen:

### RoamingChargingList.req

Feldname	Typ	Kardinalität	Beschreibung
providerId	string	1..1	Kundenlieferantenkenung (gemäß Contract-ID / Karten-ID) = Absender der Anfrage
operatorId	string	1..1	standardisierte Charge-Point-Operator-ID (Ladepunktbetreiberkenung) = angefragter Ladepunktbetreiber

Feldname	Typ	Kardinalität	Beschreibung
startDate	time stamp	1..1	Beginn der Periode
endDate	time stamp	1..1	Ende der Periode

#### 4.4.4 Antwort erweiterter Massendatensatz

##### RoamingChargingList.resp

Feldname	Typ	Kardinalität	Beschreibung
chargeRecords	chargeRecord	0..n	Liste der einzelnen Abrechnungsdatensätze – auch wenn zuvor bereits einzeln versendet (Duplikate ignoriert ggf. der Empfänger)
chargeAmountSum	integer	1..1	in kWh *1000

##### chargeRecord

Feldname	Typ	Kardinalität	Beschreibung
chargePointId	string	1..1	Charge-Point-ID: [0-9]*[0-9]* gemäß Modell-Projekte-Spezifikation = Absender und Ladepunkt des Abrechnungsdatensatzes
contractId	string	1..1	Contract-ID bzw. Karten-ID (DE-AAA-123456-P)
sessionId	string	0..1	Optional: Session-ID = Zuordnung zum Endkunden (sofern eine Autorisierung zuvor erfolgte – siehe hierzu vorausgehende Diskussion)
chargeAmountStart	integer	1..1	in kWh*1000
chargeAmountStartSignature	string	0..1	Optional- sofern Zählerwerte signiert werden.
chargeAmountEnd	integer	1..1	in kWh*1000
chargeAmountEndSignature	string	0..1	Optional- sofern Zählerwerte signiert werden.
chargeEndTime	time stamp	1..1	Zeitstempel des Ladeendzeitpunktes oder zumindest alternativ Empfangszeitpunkt von Ladesäule
chargeStartTime	time stamp	0..1	Optional: Zeitstempel des Ladestartzeitpunktes (nicht Bestandteil aller Ladesäulenprotokolle)
chargePointStreet	string	0..1	Optional: Adressauflösung der Ladepunktken-nung gemäß Charge-Point-ID
chargePointHouseNumber	string	0..1	Optional: Adressauflösung der Ladepunktken-nung gemäß Charge-Point-ID
chargePointPostcode	string	0..1	Optional: Adressauflösung der Ladepunktken-nung gemäß Charge-Point-ID
chargePointCity	string	0..1	Optional: Adressauflösung der Ladepunktken-nung gemäß Charge-Point-ID

- Hierbei ist der empfangende Kundenlieferant selbst verantwortlich ggf. mehrfach erhaltene Datensätze zu ignorieren (ggf. resultierend aus überlappend angefragten Zeiträumen).

## 5 Protokoll-Konzeption

### 5.1 WSDL-Template

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn://aida/ws/2010/12/" xmlns:impl="urn://aida/ws/2010/12/"
xmlns:intf="urn://aida/ws/2010/12/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdloa="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsi="http://ws-i.org/profiles/basic/1.1/xsd"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <schema targetNamespace="urn://aida/ws/2010/12/" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <complexType name="GetChargeRecordListRequest">
        <sequence>
          <element name="providerId" nillable="true" type="xsd:string"/>
          <element name="operatorId" nillable="true" type="xsd:string"/>
          <element name="endTime" nillable="true" type="xsd:dateTime"/>
          <element name="startTime" nillable="true" type="xsd:dateTime"/>
        </sequence>
      </complexType>
      <element name="getChargeRecordListResponse">
        <complexType>
          <sequence>
            <element name="getChargeRecordListReturn" nillable="true" type="impl:GetChargeRecordListResponse"/>
          </sequence>
        </complexType>
      </element>
      <element name="authorize">
        <complexType>
          <sequence>
            <element name="parameters" nillable="true" type="impl:AuthorizeRequest"/>
          </sequence>
        </complexType>
      </element>
      <element name="authorizeResponse">
        <complexType>
          <sequence>
            <element name="authorizeReturn" nillable="true" type="impl:AuthorizeResponse"/>
          </sequence>
        </complexType>
      </element>
      <element name="charge">
        <complexType>
          <sequence>
            <element name="parameters" nillable="true" type="impl:ChargeRequest"/>
          </sequence>
        </complexType>
      </element>
      <element name="chargeResponse">
        <complexType>
          <sequence>
            <element name="chargeReturn" nillable="true" type="impl:ChargeResponse"/>
          </sequence>
        </complexType>
      </element>
      <element name="getChargeRecordList">
        <complexType>
          <sequence>
            <element name="parameters" nillable="true" type="impl:GetChargeRecordListRequest"/>
          </sequence>
        </complexType>
      </element>
      <complexType name="GetChargeRecordListResponse">
        <sequence>
          <element name="chargeRecords" nillable="true" type="impl:ArrayOfChargeRecord"/>
          <element name="chargeAmountSum" type="xsd:long"/>
        </sequence>
      </complexType>
      <complexType name="ChargeRecord">
        <sequence>
          <element name="chargePointId" nillable="true" type="xsd:string"/>
          <element name="contractId" nillable="true" type="xsd:string"/>
          <element name="sessionId" type="xsd:long"/>
          <element name="meterData" nillable="true" type="impl:MeterData"/>
          <element name="chargePointAddress" nillable="true" type="impl:Address"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>

```

```
</sequence>
</complexType>
<complexType name="MeterData">
  <sequence>
    <element name="meterReadingStart" type="xsd:long"/>
    <element name="meterReadingStartSignature" nillable="true" type="xsd:string"/>
    <element name="meterReadingEnd" type="xsd:long"/>
    <element name="meterReadingEndSignature" nillable="true" type="xsd:string"/>
    <element name="chargeEndTime" nillable="true" type="xsd:dateTime"/>
    <element name="chargeStartTime" nillable="true" type="xsd:dateTime"/>
  </sequence>
</complexType>
<complexType name="Address">
  <sequence>
    <element name="street" nillable="true" type="xsd:string"/>
    <element name="houseNumber" nillable="true" type="xsd:string"/>
    <element name="postcode" nillable="true" type="xsd:string"/>
    <element name="city" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
<complexType name="ArrayOfChargeRecord">
  <sequence>
    <element maxOccurs="unbounded" minOccurs="0" name="ChargeRecord" nillable="true" type="impl:ChargeRecord"/>
  </sequence>
</complexType>
<complexType name="AuthorizeRequest">
  <sequence>
    <element name="contractId" nillable="true" type="xsd:string"/>
    <element name="serialNumber" nillable="true" type="xsd:string"/>
    <element name="signature" nillable="true" type="xsd:string"/>
    <element name="operatorId" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
<complexType name="AuthorizeResponse">
  <sequence>
    <element name="status" type="xsd:int"/>
    <element name="statusMessage" nillable="true" type="xsd:string"/>
    <element name="sessionId" type="xsd:long"/>
    <element name="info" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
<complexType name="ChargeRequest">
  <sequence>
    <element name="chargeRecord" nillable="true" type="impl:ChargeRecord"/>
  </sequence>
</complexType>
<complexType name="ChargeResponse">
  <sequence>
    <element name="status" type="xsd:int"/>
    <element name="statusMessage" nillable="true" type="xsd:string"/>
    <element name="price" type="xsd:int"/>
  </sequence>
</complexType>
</schema>
</wsdl:types>

<wsdl:message name="authorizeResponse">
  <wsdl:part element="impl:authorizeResponse" name="parameters"/>
</wsdl:message>

<wsdl:message name="getChargeRecordListResponse">
  <wsdl:part element="impl:getChargeRecordListResponse" name="parameters"/>
</wsdl:message>

<wsdl:message name="chargeRequest">
  <wsdl:part element="impl:charge" name="parameters"/>
</wsdl:message>

<wsdl:message name="authorizeRequest">
  <wsdl:part element="impl:authorize" name="parameters"/>
</wsdl:message>

<wsdl:message name="getChargeRecordListRequest">
```

```
<wsdl:part element="impl:getChargeRecordList" name="parameters"/>
</wsdl:message>

<wsdl:message name="chargeResponse">
  <wsdl:part element="impl:chargeResponse" name="parameters"/>
</wsdl:message>

<wsdl:portType name="AidaServiceImpl">
  <wsdl:operation name="getChargeRecordList">
    <wsdl:input message="impl:getChargeRecordListRequest" name="getChargeRecordListRequest"/>

    <wsdl:output message="impl:getChargeRecordListResponse" name="getChargeRecordListResponse"/>
  </wsdl:operation>

  <wsdl:operation name="authorize">
    <wsdl:input message="impl:authorizeRequest" name="authorizeRequest"/>

    <wsdl:output message="impl:authorizeResponse" name="authorizeResponse"/>
  </wsdl:operation>

  <wsdl:operation name="charge">
    <wsdl:input message="impl:chargeRequest" name="chargeRequest"/>

    <wsdl:output message="impl:chargeResponse" name="chargeResponse"/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="AidaServiceImplSoapBinding" type="impl:AidaServiceImpl">
<wsaw:UsingAddressing wsdl:required="false" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>

  <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

  <wsdl:operation name="getChargeRecordList">
    <wsdlsoap:operation soapAction="getChargeRecordList"/>

    <wsdl:input name="getChargeRecordListRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>

    <wsdl:output name="getChargeRecordListResponse">
      <wsdlsoap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="authorize">
    <wsdlsoap:operation soapAction="authorize"/>

    <wsdl:input name="authorizeRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>

    <wsdl:output name="authorizeResponse">
      <wsdlsoap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="charge">
    <wsdlsoap:operation soapAction="charge"/>

    <wsdl:input name="chargeRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>

    <wsdl:output name="chargeResponse">
```

```
<wsdlsoap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="AidaServiceImplService">
  <wsdl:port binding="impl:AidaServiceImplSoapBinding" name="AidaServiceImpl">
    <wsdlsoap:address location="http://localhost:9080/SmartWheelsAccessServiceWs/services/AidaServiceImpl"/>
  </wsdl:port>
</wsdl:service>

</wsdl:definitions>
```

## 5.2 RFID-Übertragungsformat

- Um die RFID-ID immer auf dieselbe Art und Weise auf die Datenblöcke aufzubringen, muss definiert werden, wie dieser 7 Byte (sofern DESFire) lange Datenblock in ASCII-Form repräsentiert wird. Hierzu wird festgelegt, dass die Bytes UID0 : : : UID6 in eben dieser Reihenfolge in hexadezimaler Form verkettet werden. Hierbei wird jedes Byte der UID auf eine zwei Zeichen lange Hexadezimalzahl mit führender Null abgebildet. Für die Zahlen 10-15 (a-f) sind Kleinbuchstaben zu verwenden. Die Endrepräsentation muss also dem regulären Ausdruck [0-9a-f]{14} genügen.

## 5.3 Sicherheit

- Webservice-Security-Layer: User-ID und PW
- Serverseitig muss sichergestellt werden, dass Marktpartner nur für sie berechnigte Daten abrufen können – siehe z.B. Anfrage erweiterter Massendatensatz.

## 6 Referenzen

- Spezifikation EVSE Phone Access (Dr. J. Heuer, Siemens AG)
- Technische Spezifikation der RFID-Umgebung zur interoperablen Authentisierung (Johannes Bauer, Innovations Software Technology GmbH, Bosch Group)
- Abstimmung der Elektromobilitäts-Modellprojekte zum Contract-ID bzw. Karten-ID Nummernkreis (Stephan Voit, RWE)
- EDL21-Zähler und EDL40-Konzept (Stephan Voit, RWE)