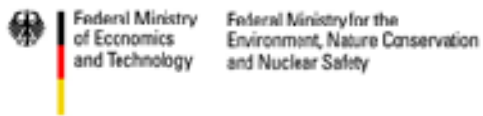


Specification of EVSE Phone Access v1.6

Task Force “Authentication and Identification EV – EVSE”

Supported by

ICT FOR 
ELECTRIC MOBILITY



This document is the result of contributions from the research projects in the German research program “ICT for Electric Mobility” which has been funded by the Federal Ministry of Economics and Technology and the Federal Ministry for the Environment, Nature Conservation and Nuclear Safety. Interoperability according to this specification has been tested between research projects in May 2011.

Edited by

Sebastian Käbisch
Dr. Jörg Heuer
Corporate Technology
Siemens AG

1.Document History

Version	Date	Editor	Changes
0.1	30.6.2010	Dr. J. Heuer, Siemens AG	Initial Document
0.2	17.8.2010	Dr. J. Heuer, Siemens AG	Comments of TelConf 26.7.10 integrated
0.3	23.9.2010	Sebastian Käbisch, Siemens AG	XML Security description
1.0	6.11.2010	Dr. J. Heuer, Siemens AG	Final Editing for Version 1.0
1.1	20.1.2011	Dr. J. Heuer, Siemens AG	Added description of flap unlock by IEC61851 signalling
1.2	14.3.2011	Dr. J. Heuer, Siemens AG	Extensions for Accounting
1.3	14.3.2011	Sebastian Käbisch, Siemens AG	Bug fixes in WSDL and update of XML Signature references in WSDL/ XSD
1.4	15.4.2011	Sebastian Käbisch, Siemens AG	Update XSD and type descriptions
1.5	13.5.2011	Dr. J. Heuer, Siemens AG	Clarified transaction ID Semantics to support AIDA Specification
1.6	16.8.2011	Sebastian Käbisch, Dr. J. Heuer, Siemens AG	Editing for publication

2. Terms and Definitions

- **EVSE:** Electrical Vehicle Supply Equipment
- **EVSE ID:** The unique identifier of the EVSE (see section 6 for the format definition)
- **EVSE Operator:** describes the operator of the EVSE
- **Phone access server:** describes a server which can be accessed by a user to request access to an EVSE
- **SCADA:** in the context of this document describes a remote operation centre for the EVSE managed by the EVSE operator
- **Transaction ID:** describes an ID which enables the EVSE operator to resolve on request the user who has initiated the transaction
- **User DB:** describes a data base holding all credentials of subscribed users

3. Motivation and Scope

This specification targets the migration of legacy EVSEs such as those equipped with Schuko sockets or cases where sockets in the EVSE according to [IEC 62196-2] are covered by a locked flap in general. For instance in the case of a Schuko socket EVSEs can provide a lockable flap to secure the plug in the socket. To enable the access to these EVSEs i.e. to unlock the flap a phone access solution is desirable to provide charging services to a large number of even spontaneous users at a low cost.

Note: this specification is a migration solution for existing EVSEs. It is expected that with the finalisation of [ISO/IEC 15118-2] EVSEs will support authentication in a convenient way by the EV via communication over the charging cable. In this case the charging cable has plugs according to [IEC 62196-2] which can be locked in the socket and which does not require lockable flaps in the EVSE.

While the phone access solution itself can be implemented according to the requirements of a specific operator this document focuses on the interface between operators so a customer of operator B (home operator, HO) can also access EVSEs of operator A (visited operator, VO) and vice versa.

In the present version of the document the functionality is limited to an access solution including accounting information. However it is envisaged that in a later version the description will be extended in the future.

Note: This document has been motivated by the German “ICT for Electric Mobility” research projects which are funded by the Federal Ministry of Economics and Technology and the Federal Ministry for the Environment, Nature Conservation and Nuclear Safety.. Interoperability according to this specification has been tested between research projects in May 2011.

4. Architecture and Interfaces for Interoperability

In the task force *Authentication and Identification EV – EVSE* several architectures have been discussed. The selected architecture described in this section has been selected due to the following reasons:

- **User friendly:** the user can always use the same user interface of the EVSE operator he has subscribed to and the user interface can be optimized for the particular user

- Limited interface complexity as no customer or account related data has to be exchanged
- The principle of data frugality is followed as for the visiting EVSE operator the user is not known

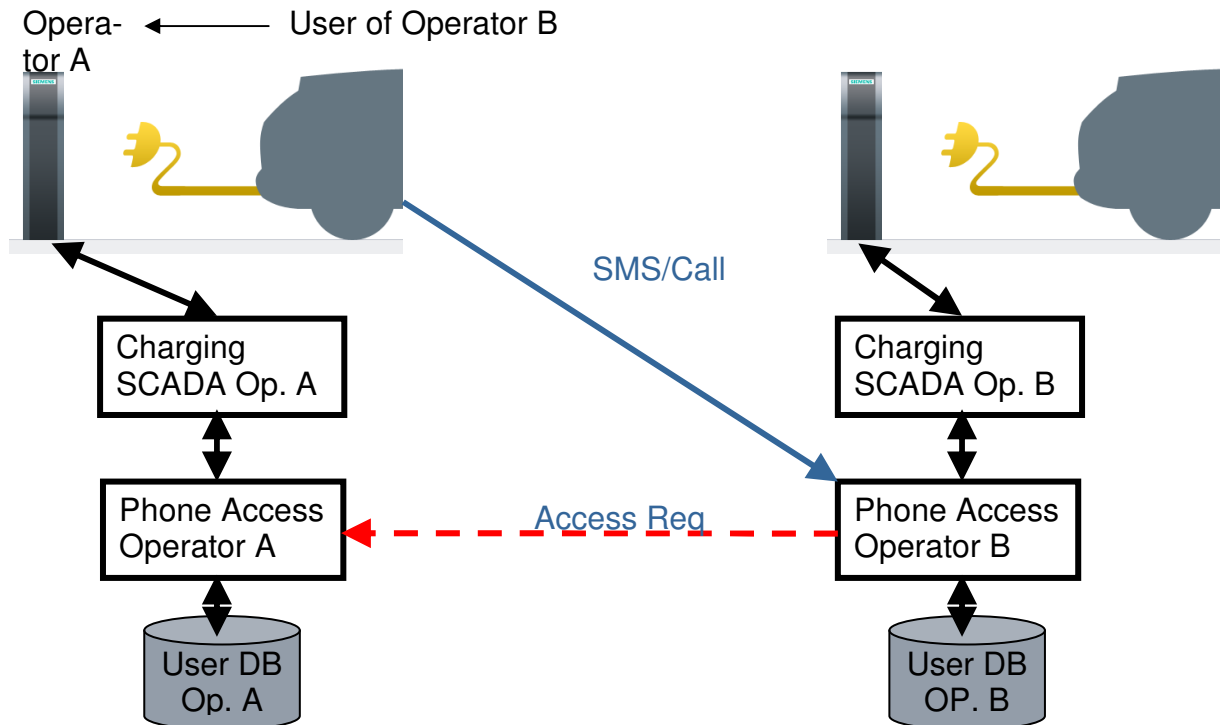


Figure 1: EVSE Phone Access architecture described in this document, including the interface (red dashed line) specified for interoperability between different EVSE operators

In fig. 1 an EVSE phone access architecture is described especially for the case of a user subscribed to an EVSE operator B (HO) who is requesting access to an EVSE of operator A (VO). It is assumed that each operator operates EVSE, SCADA systems, a phone access server and a User DB. Note that these are logical blocks which might be implemented differently, e.g. the SCADA system might contain a phone access server.

To gain access to an EVSE of operator A (VO) the user subscribed to an EVSE operator B (HO) can signal the EVSE ID of an EVSE of operator A (VO) via SMS or phone call to the phone access server of operator B (HO)¹. After verifying the identity of the user e.g. by the caller ID the Phone Access Server of operator B (HO) sends an access request to the Phone Access Server of operator A (VO) signalling the EVSE ID, a Transaction ID and a signature. After verifying the signature operator A (VO) enables the access to the identified EVSE and responds to the request.

As it might be the case that the access has to be granted when inserting and removing the plug in the EVSE the phone access server of operator B (HO) will keep the TransactionID unchanged as long the operator A (VO) has not signalled a release status either in an AccessResponse or StatusResponse message. This enables the phone access server of operator A (VO) to verify if the user is authorised to access the EVSE again for instance to release the plug. To avoid dangling TransactionIDs at the operator B (HO) the current status of the TransactionID at the operator A (VO) can be checked by a StatusRequest.

¹ In addition the user might interact with the EVSE itself e.g. to initiate a connection between the EVSE and the SCADA of operator A if this is not possible remotely from the SCADA of operator A.

This might be necessary for instance if the charging process is terminated by signalling state B according to [IEC61851-1]. In this case the EVSE shall also grant access according to this specification.

The sequence diagram following this interaction flow is shown in fig. 2.

In this document the interface between the phone access server of operator B and A (see dashed line in fig. 1) is specified for interoperability.

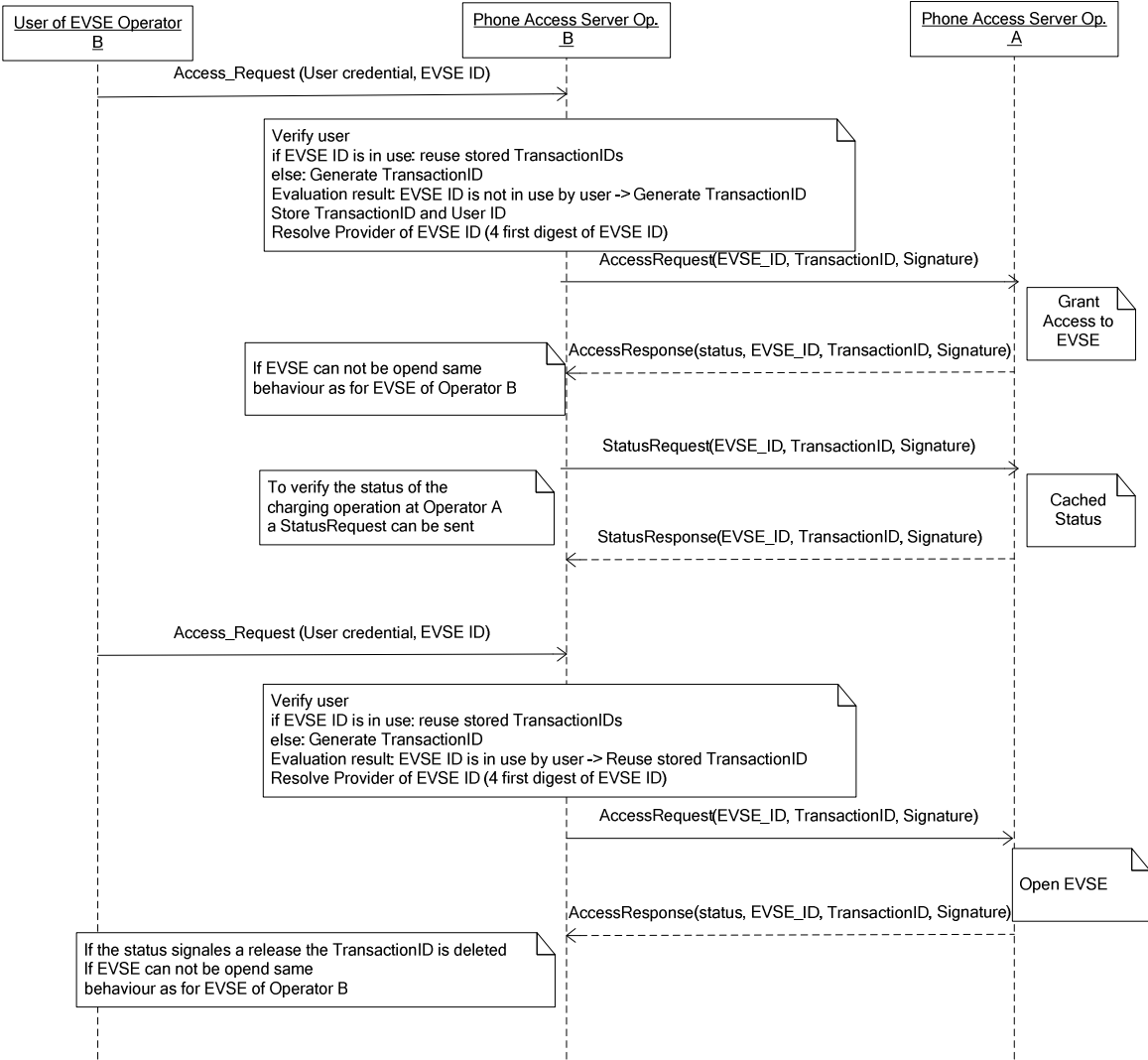


Figure 2: Sequence diagram of a User of an EVSE Operator B requesting access for an EVSE of operator A.

Note that the interface between the phone and the phone access server of operator B (HO) is not in the scope of this document as it can be specific to operator B (HO) as long as the EVSE ID of the EVSE operator A (VO) can be signalled. Therefore in this document also the EVSE ID is specified for interoperability purposes.

5. Protocol Specification

The interface between the phone access servers is based on Web service protocols. The SOAP XML message framework [SOAP] shall be used to exchange data between both parties. The corresponding Web Service Description Language (WSDL) [WSDL] which describes the Phone Access Web service interface can be found in Annex A of this document.

W3C XML Signatures [XML Signature] is used to sign some particular data elements of the phone access messages. The signature will be signalled as part of the header in the SOAP envelope. The abstract view of an XML signature is shown in fig. 3 below.

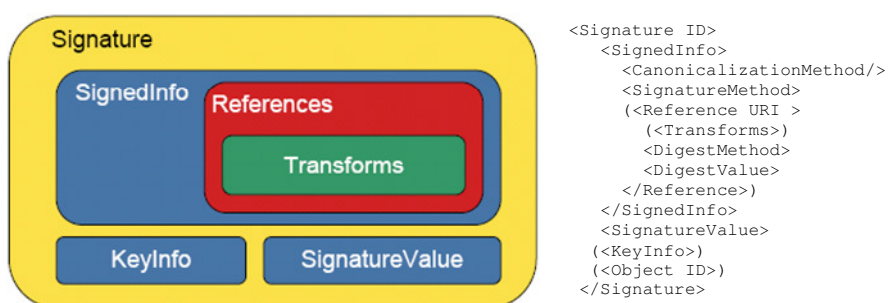


Figure 3 — XML Signature Overview

Table 1 provides the selected XML Signature form and algorithms for the phone access messages.

XML Signature Form	Enveloping XML Signature
XML Canonicalization	Canonical XML Version 1.0 (http://www.w3.org/TR/2001/REC-xml-c14n-20010315)
Signature Method	RSA-SHA1 (http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd#rsa-sha1)
Digest Method	SHA1 (http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd#sha1)

Table 1: W3C XML Signature form and algorithms used for the phone access messages

Certificates may be exchanged by the provided `<KeyInfo>` element of the XML Signature framework. In this case the X.509v3 format definition shall be used and the corresponding information shall be carried within the `<X509Data >` element. An XML skeleton of `<KeyInfo>` is shown in the following listing:

```

<KeyInfo>
  <X509Data >
    <X509SubjectName/>
    <X509IssuerSerial/>
    <X509Certificate/>
  </X509Data>
  <KeyValue />
</KeyInfo>
  
```

Annex C provides SOAP instances that include the usage of the XML Signature within the SOAP header.

The URL of the phone server web service interface of a particular EVSE operator is resolved by the EVSE ID. Accordingly in this section only the datatypes and elements used in the SOAP messages are specified.

5.1 Basic Protocol Specification

```
<xs:schema targetNamespace="urn:bmwi:EVSE_Phone_Access:2010:types"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="
  urn:bmwi:EVSE_Phone_Access:2010:types" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
```

Syntax

```
<!-- access request definition -->
<xs:element name="AccessReq" type="AccessCredentialsType"/>

<!-- access response definition -->
<xs:element name="AccessRes" type="AccessResType"/>

<!-- status request definition -->
<xs:element name="StatusReq" type="AccessCredentialsType"/> <!-- same content as AccessReq
-->

<!-- status response definition -->
<xs:element name="StatusRes" type="StatusResType"/>

<!-- Complex Types -->
<xs:complexType name="AccessResType">
  <xs:sequence>
    <xs:element name="Status" type="statusAccType"/>
    <xs:element name="AccessCredentials" type="AccessCredentialsType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="StatusResType">
  <xs:sequence>
    <xs:element name="Status" type="statusStaType"/>
    <xs:element name="AccessCredentials" type="AccessCredentialsType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AccessCredentialsType">
  <xs:sequence>
    <xs:element name="EVSEID" type="EVSEIDType"/>
    <xs:element name="TransactionID" type="transactionIDType"/>
  </xs:sequence>
  <xs:attribute name="Id" type="xs:IDREF" use="required" /> <!-- ID ref -->
</xs:complexType>

<!-- Simple Types -->
<xs:simpleType name="EVSEIDType" >
  <xs:restriction base="xs:string">
```

```

    <xs:pattern value="EVSE_ID:[0-9]{4}\*[0-9]*"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="transactionIDType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="32"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="statusAccType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Access granted"/>
    <xs:enumeration value="Access denied"/>
    <xs:enumeration value="Access granted, EV hooked up"/>
    <xs:enumeration value="Access granted, EV removed"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="statusStaType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Idle"/>
    <xs:enumeration value="Waiting to connect"/>
    <xs:enumeration value="Charging"/>
    <xs:enumeration value="Waiting to disconnect"/>
    <xs:enumeration value="Interrupted/Error"/>
    <xs:enumeration value="Completed"/>
  </xs:restriction>
</xs:simpleType>

```

```
</xs:schema>
```

Semantics

Type Name	Semantics
EVSEIDType	Signals the EVSE ID for which access is requested. The ID format is described in section 6.
transactionIDType	Signals a transaction ID. The transaction ID is unique in the scope of home operator and can be used as a reference for identifying the transaction of that home operator. The ID is in maximum 32 char long. The format is defined by the home operator.
statusAccType	<p>Signals the status of the Access request. The following values are valid:</p> <ul style="list-style-type: none"> • Access granted • Access denied • Access granted, EV hooked up • Access granted, EV removed <p>Depending on the type of EVSE beside the granted access also the presence of an EV can be indicated.</p>
statusStaType	<p>Signals the status of the Status request. The following values are valid:</p> <ul style="list-style-type: none"> • Idle • Waiting to connect • Charging • Waiting to disconnect

	<ul style="list-style-type: none"> • Interrupted/Error • Completed
--	--

5.1 Extension to support accounting information

```
<xs:schema targetNamespace="urn:bmwi:EVSE_Phone_Access:2011:types"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="
  urn:bmwi:EVSE_Phone_Access:2010:types"
  xmlns:bT="urn:bmwi:EVSE_Phone_Access:2010:types"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
```

Syntax

```
<xs:complexType name="AccessAccountingResType">
  <xs:complexContent>
    <xs:extension base="bT:AccessResType">
      <xs:sequence>
        <xs:element name="AccountingStart" type="AccountingDataType"/>
        <xs:choice minOccurs="0">
          <xs:element name="AccountingStatus" type="AccountingDataType"/>
          <xs:element name="AccountingEnd" type="AccountingDataType"/>
        </xs:choice>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="AccountingDataType">
  <xs:sequence>
    <xs:element name="TimeOfCapture" type="xs:dateTime"/>
    <xs:element name="ChargeAmount" type="ChargeAmountType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ChargeAmountType">
  <xs:attribute name="amount" type="xs:float"/>
  <xs:attribute name="unit" type="xs:string"/>
</xs:complexType>
```

```
</xs:schema>
```

Semantics

Type Name	Semantics
AccountingStart	Signals accounting information present at the EVSE when access is granted.
AccountingStatus	Signals accounting information currently present at the EVSE. This element is optional but should be signalled in a StatusRes if the charging process is ongoing.
AccountingStop	Signals accounting information present at the EVSE at the end of the charging process. This element is optional but should be signalled in a StatusRes or AccessRes if the charging process is ended.
TimeOfCapture	Signals the time point when the ChargeAmount information is captured.

ChargeAmount	Signals the amount of charging and unit used for accounting. The unit can be one of the following values: KWh seconds
---------------------	---

6. EVSE ID Format

To enable interoperability with any operator specific phone access solution described in section 4 the format of the EVSE ID is defined in this document by the following regular expression:

EVSE_ID:[0-9]*\[0-9*\]

The resulting EVSE ID has a format like:

EVSE_ID:AAAA*BBB*BB

Where A and B are single digits. The digits AAAA before the first ‘*’ identify the EVSE operator. The instantiations starting with ‘00’ is reserved for future international use.

The variable number of digits after the separator ‘*’ identify a particular EVSE of the EVSE operator identified by AAAA. How many digits are used for the identification is up to the EVSE Operator.

Note: a server including specified interfaces for registration and resolution of EVSE Operators is currently under discussion

7. References

- [IEC/ISO 15118-2] Road vehicles — Vehicle to grid communication interface — Part 2: Technical protocol description and Open Systems Interconnections (OSI) layer requirements
- [IEC61851-1] Electric vehicle conductive charging system - Part 1: General requirements
- [IEC 62196-2] Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles - Part 2: Dimensional compatibility and interchangeability requirements for a.c. pin and contact-tube accessories
- [WS Security] Web Services Security: SOAP Message Security 1.1 (WS-Security 2004), <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>
- [XML Signature] XML Signature Syntax and Processing (Second Edition), W3C Recommendation 10 June 2008, <http://www.w3.org/TR/2008/REC-xmlsig-core-20080610/>

Annex A: WSDL of the phone access server interface

In this section the WSDL for the phone access server interface specified in this document is provided.

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:pasi="urn:bmwi:EVSE_Phone_Access:2010"
  xmlns:pt="urn:bmwi:EVSE_Phone_Access:2010:types"
  xmlns:xmlsig="http://www.w3.org/2000/09/xmlsig#"
  targetNamespace="urn:bmwi:EVSE_Phone_Access:2010">

  <types>
    <xs:schema targetNamespace="urn:bmwi:EVSE_Phone_Access:2010:types"
      xmlns="urn:bmwi:EVSE_Phone_Access:2010:types"

      elementFormDefault="unqualified" attributeFormDefault="unqualified">
      <xs:import namespace="http://www.w3.org/2000/09/xmlsig#" schemaLocation="xmlsig-core-
        schema.xsd"/>

      <!-- access request definition -->
      <xs:element name="AccessReq" type="AccessCredentialsType"/>

      <!-- access response definition -->
      <xs:element name="AccessRes" type="AccessResType"/>

      <!-- status request definition -->
      <xs:element name="StatusReq" type="AccessCredentialsType"/> <!-- same content as
        AccessReq -->

      <!-- status response definition -->
      <xs:element name="StatusRes" type="StatusResType"/>

      <!-- Complex Types -->

      <xs:complexType name="AccessResType">
        <xs:sequence>
          <xs:element name="Status" type="statusAccType"/>
          <xs:element name="AccessCredentials" type="AccessCredentialsType"/>
        </xs:sequence>
      </xs:complexType>

      <xs:complexType name="StatusResType">
        <xs:sequence>
          <xs:element name="Status" type="statusStaType"/>
          <xs:element name="AccessCredentials" type="AccessCredentialsType"/>
        </xs:sequence>
      </xs:complexType>

      <xs:complexType name="AccessCredentialsType">
        <xs:sequence>
          <xs:element name="EVSEID" type="EVSEIDType"/>
          <xs:element name="TransactionID" type="transactionIDType"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </types>
</definitions>
```

```

    <xs:attribute name="Id" type="xs:IDREF" use="required" /> <!-- ID ref -->
  </xs:complexType>

  <!-- Simple Types -->
  <xs:simpleType name="EVSEIDType" >
    <xs:restriction base="xs:string">
      <xs:pattern value="EVSE_ID:[0-9]{4}\*[0-9]*"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="transactionIDType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="32"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="statusAccType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Access granted"/>
      <xs:enumeration value="Access denied"/>
      <xs:enumeration value="Access granted, EV hooked up"/>
      <xs:enumeration value="Access granted, EV removed"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="statusStaType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Idle"/>
      <xs:enumeration value="Waiting to connect"/>
      <xs:enumeration value="Charging"/>
      <xs:enumeration value="Waiting to disconnect"/>
      <xs:enumeration value="Interrupted/Error"/>
      <xs:enumeration value="Completed"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
</types>

<!-- SOAP Header -->
<message name="SOAPHeader">
  <part name="header" element="xmsig:Signature"/>
</message>

<!-- SOAP body definitions -->
<!-- AccessReq message -->
<message name="AccessReqMsg">
  <part name="parameters" element="pt:AccessReq"/>
</message>

<!-- AccessRes message -->
<message name="AccessResMsg">
  <part name="parameters" element="pt:AccessRes"/>
</message>

<!-- StatusReq message -->
<message name="StatusReqMsg">
  <part name="parameters" element="pt:StatusReq"/>
</message>

```

```

<!-- StatusRes message -->
<message name="StatusResMsg">
  <part name="parameters" element="pt:StatusRes"/>
</message>

<!-- Port -->
<portType name="EVSEAccessInterface">

  <!-- Access Operation -->
  <operation name="Access">
    <input message="pasi:AccessReqMsg"/>
    <output message="pasi:AccessResMsg"/>
  </operation>

  <!-- Status Operation -->
  <operation name="Status">
    <input message="pasi:StatusReqMsg"/>
    <output message="pasi:StatusResMsg"/>
  </operation>
</portType>

<binding name="EVSEAccessSoapHttpBinding" type="pasi:EVSEAccessInterface">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

  <!-- access operation -->
  <operation name="Access">
    <soap:operation soapAction="http://example.org/EVSEAccessInterface/#Access"/>
    <input>
      <soap:header message="pasi:SOAPHeader" part="header" use="literal"/>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:header message="pasi:SOAPHeader" part="header" use="literal"/>
      <soap:body use="literal"/>
    </output>
  </operation>

  <!-- status operation -->
  <operation name="Status">
    <soap:operation soapAction="http://example.org/EVSEAccessInterface/#Status"/>
    <input>
      <soap:header message="pasi:SOAPHeader" part="header" use="literal"/>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:header message="pasi:SOAPHeader" part="header" use="literal"/>
      <soap:body use="literal"/>
    </output>
  </operation>

</binding>

<!-- Service location -->
<service name="EVSEAccessService">
  <port name="EVSEAccess" binding="pasi:EVSEAccessSoapHttpBinding">
    <soap:address location="http://localhost"/>
  </port>
</service>

```

</definitions>

Annex B: XML Schema definition of the phone access SOAP messages (informative)

In this section the XML Schema definition for the SOAP messages exchanged via the phone access server interface are specified. This definition can be used for validation of the exchanged messages and the code generation for the xml messages itself.

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
    xmlns:xmlsig="http://www.w3.org/2000/09/xmlsig#"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:pa="urn:bmwi:EVSE_Phone_Access:2010:types"
    targetNamespace="http://schemas.xmlsoap.org/wsdl/soap12/"
    <xs:import namespace="urn:bmwi:EVSE_Phone_Access:2010:types"
      schemaLocation="schema.xsd"/>
    <xs:element name="Envelope">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="SOAP-ENV:Header"/>
          <xs:element ref="SOAP-ENV:Body"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Header">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="xmlsig:Signature"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Body">
      <xs:complexType>
        <xs:choice>
          <xs:element ref="pa:AccessReq"/>
          <xs:element ref="pa:AccessRes"/>
          <xs:element ref="pa:StatusReq"/>
          <xs:element ref="pa:StatusRes"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

Annex C: XML message examples of the phone access interface (informative)

In this section some XML example instances of SOAP messages exchanged via the phone access server interface are provided. These examples can be used for functionality verification.

Example of signing EVSEID and TransactionID Element by using the defined XML Security modes in table xxx.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:pa="urn:bmwi:EVSE_Phone_Access:2010:types"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
  <SOAP-ENV:Header>
    <ds:Signature>
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd#rsa-sha1"/>
        <ds:Reference URI="#id-1">
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>wLumPkKZ+X48rjao/XUUQDp0xk0=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>ZGVmYXVsdA==</ds:SignatureValue>
    </ds:Signature>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <pa:StatusReq Id="id-1">
      <pa:EVSEID>EVSE_ID:0000*</pa:EVSEID>
      <pa:TransactionID>TransactionID0</pa:TransactionID>
    </pa:StatusReq>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```