



# OCHP Update Announcement

---

*Target Versions: OCHP 1.4 | OCHPdirect 0.2 | eCHS 1.1.4*

*Announcement Date: August 18<sup>th</sup>, 2016*

*Protocol Release Date: August 18<sup>th</sup>, 2016*

*eCHS-Q Upgrade Date (preliminary): November 2<sup>nd</sup>, 2016*

## Introduction

### Management Summary

After a full year of so far undisturbed productive operation of eCHS on OCHP Version 1.3, we are proud to announce OCHP 1.4. Starting with an integration phase from November 2016, the new interface will provide more functionality and higher convenience in the implementation.

At the same time, the OCHPdirect extension that was introduced with OCHP 1.3 will be updated to version 0.2 to enable further functionality and better usability for all parties.

### Technical Summary

In this update to OCHP 1.4 some enhancements are made to the API. Besides a few minor updates, bug fixes and spellings corrections also new features are added to the API.

- New role: parking space operator
  - Currently with the ability to supply live data regarding parking space occupation
  - Makes use of existing *UpdateStatus* method
- Tariff data exchange
  - Tariff data can now be exchanged through the platform
  - Tariffs can be defined for individual recipients to reflect the bilateral arrangements
  - New methods are introduced: *UpdateTariffs*, *GetTariffUpdates*
- Enhanced CDR life cycle
  - CDRs can now be revised by the CPO
  - The formerly manual process of handling declined CDRs is now covered in OCHP
  - New method for the CPO: *CheckCDRs* (to check on the current status of their CDRs)
  - *ConfirmCDRs* now only requires CDR-ID + EVSE-ID as input, not the full CDR
- Reworked / clarified LiveAuthorisation
  - now called *GetSingleRoamingAuthorisationInfo*
  - LiveAuth sessions are no longer tracked in eCHS and no longer get a LiveAuthID

The following changes were made to OCHPdirect 0.2:

- Removed the distinction between basic and advanced use cases
  - Made *InformProvider* a mandatory part of OCHPdirect
  - Providers are now required to provide an Endpoint for OCHPdirect 0.2
  - Made *InformProvider* a bidirectional method:
    - CPO sends *InformProvider* on charge event (as in OCHPdirect 0.1)
    - EMP requests *InformProvider* to collect / update information
- Extended *InformProvider* with further information
- Enabled reservation requests in *SelectEvse*
- Enabled basic smart charging in *ControlEvse*

## Contents

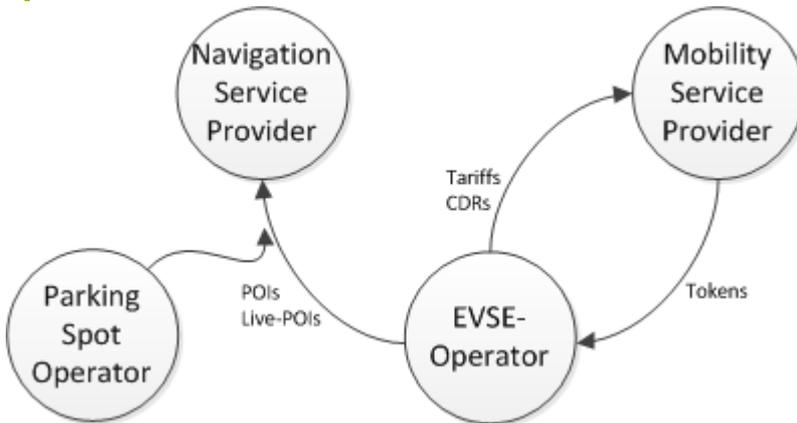
Introduction .....	2
Management Summary .....	2
Technical Summary .....	2
Contents .....	3
Changes to OCHP .....	4
1. Added PSO role .....	4
2. Added Tariff Exchange .....	8
3. Changed LiveAuth to SingleAuth .....	15
4. Extended CDR-Handling .....	17
5. Changed – to _ in <i>connectorStandard enum</i> .....	22
6. Combined address and parking related fields.....	23
7. Changed <i>ParkingRestrictionType</i> to <i>RestrictionType</i> .....	26
8. Added multiple data fields to CDR-Info .....	28
9. Changed <i>operatingTimes</i> and <i>accessTimes</i> .....	30
10. Other changes.....	32
11. Specification document .....	34
Changes to OCHPdirect .....	35
12. Extended <i>InformProvider</i> method .....	35
13. Clarified token exchange .....	41
14. Enabled reservation through <i>SelectEvse</i> .....	43
15. Specification document .....	47
Changes to the eCHS.....	48
16. Contact page in the Frontend .....	48

# Changes to OCHP

## 1. Added PSO role

Change type:	Enhancement
Impact:	<input checked="" type="checkbox"/> Specification, <input checked="" type="checkbox"/> API, <input checked="" type="checkbox"/> eCHS
Backward-Compatibility:	PSO data only available on OCHP 1.4 and onwards
Forward-Compatibility:	-
Impact for partners:	NSPs can now receive parking spot live data Partners can now send parking spot live data
Description:	<p>The role of PSO is added to the eCHS and the data exchange through <i>UpdateStatus</i> enabled. The PSO uses a <i>ParkingStatusType</i>, which is a direct copy of the EVSE major status. Roaming is enabled between PSO and CPO partners (live-POI checkbox) to enrich CPO-data and combining EVSE and parking status values into one <i>combined</i> status value.</p> <p><i>GetStatus</i> now allows the optional definition of which status to return, <i>evse</i>, <i>combined</i> or <i>parking</i>. When returning <i>combined</i> status values, eCHS will return combined status values for all EVSEs it would up to now return EVSE status values for (i.e. the same number of status is returned for combined and evse).</p> <p>Status values of EVSE and associated parking spots are combined according to the <i>parkingId</i> specified in the static POI data, which allows multiple parking spots to be mapped to one EVSE. The combined status value is generated as follows:</p> <ul style="list-style-type: none"><li>- If no <i>parkingId</i> is associated with an EVSE, then the combined status = EVSE status.</li><li>- If the EVSE status is anything other than <i>available/available</i> or <i>unknown</i>, then the combined status = EVSE status.</li><li>- If the EVSE status is <i>available/available</i> and <b>at least one</b> parking status of associated <i>parkingIds</i> is available, then the combined status = <i>available/available</i>.</li><li>- If the EVSE status is <i>available/available</i> or <i>unknown</i> and <b>all</b> parking status of associated <i>parkingIds</i> is <i>not-available</i>, then the combined status must be <i>not-available/blocked</i>.</li></ul>

## Specification



### Parking Spot Operator (PSO)

The PSO offers multiple services to the Operator as well as the EV Driver and NSP. They offer access to a parking spot associated with an EVSE to the EV driver and sometimes the location for the EVSE to the EVSE-Op. Furthermore, they may operate services that allow detailed tracking of the occupation of single parking spots, thus enhancing Operator-data sent to an NSP.

### ParkingStatusType class

Specifies the live status of a parking spot.

Field Name	Field Type	Card.	Description
parkingId	ParkingId	1	The parking spot the status is set for.
status~	MajorType	1	The status value of the parking spot.
ttl~	DateTimeType	?	The time to live is set as the deadline until which the status value is to be considered valid. Should be set to the expected status change.

### WSDL (API)

```

479 591     <s:element name="UpdateStatusRequest">
480 592         <s:complexType>
481 593             <s:sequence>
482 594                 <s:element name="evse" minOccurs="0" maxOccurs="unbounded" type="tns:EvseStatusType">
483 595                     <s:annotation>
484 596                         <s:documentation>
485 597                             This contains one EVSE id with the current status represented in a major par
486 598                         </s:documentation>
487 599                     </s:annotation>
488 600                 </s:element>
601 + <s:element name="parking" minOccurs="0" maxOccurs="unbounded" type="tns:ParkingStatusType">
602 + <s:annotation>
603 + <s:documentation>
604 +     This contains one parking-ID with the current corresponding status.
605 + </s:documentation>
606 + </s:annotation>
607 + </s:element>
489 608     <s:element name="ttl" type="tns:DateTimeTvoe" minOccurs="0" maxOccurs="1">
  
```

514	633	<s:element name="GetStatusRequest">
515	634	<s:complexType>
516	635	<s:sequence>
517	636	<s:element name="startDateTime" type="tns:DateTimeType" minOccurs="0" maxOccurs="1">
518	637	<s:annotation>
519	638	<s:documentation>
520	-	If this value is set to a point in the past, the response is limited to stat
521	-	-information that is more actual than the given value.
	639	If this value is set to a point in the past, the response is limited to stat
	640	</s:documentation>
	641	</s:annotation>
	642	</s:element>
	643	<s:element name="statusType" minOccurs="0" maxOccurs="1">
	644	<s:annotation>
	645	<s:documentation>
	646	Allows specification of either evse, parking or combined status values to be
522	647	</s:documentation>
523	648	</s:annotation>
	649	<s:simpleType>
	650	<s:restriction base="s:string">
	651	<s:enumeration value="evse"></s:enumeration>
	652	<s:enumeration value="parking"></s:enumeration>
	653	<s:enumeration value="combined"></s:enumeration>
	654	</s:restriction>
	655	</s:simpleType>
524	656	</s:element>
525	657	</s:sequence>
526	658	</s:complexType>

529	661	<s:element name="GetStatusResponse">
530	662	<s:complexType>
531	663	<s:sequence>
532	-	<s:element name="evse" type="tns:EvseStatusType" minOccurs="0" maxOccurs="unbounded"></s:element>
	664	<s:element name="combined" type="tns:EvseStatusType" minOccurs="0" maxOccurs="unbounded">
	665	<s:annotation>
	666	<s:documentation>
	667	Status values for all EVSE that have been combined with parking status value
	668	</s:documentation>
	669	</s:annotation>
	670	</s:element>
	671	<s:element name="evse" type="tns:EvseStatusType" minOccurs="0" maxOccurs="unbounded">
	672	<s:annotation>
	673	<s:documentation>
	674	Status values for all EVSE as sent to the eCHS by the EVSE operator.
	675	</s:documentation>
	676	</s:annotation>
	677	</s:element>
	678	<s:element name="parking" type="tns:ParkingStatusType" minOccurs="0" maxOccurs="unbounded">
	679	<s:annotation>
	680	<s:documentation>
	681	Status values for all parking spots as sent to the eCHS by the parking spot
	682	</s:documentation>
	683	</s:annotation>
	684	</s:element>
533	685	</s:sequence>
534	686	</s:complexType>
535	687	</s:element>

```

56 + <s:complexType name="ParkingStatusType">
57 +   <s:sequence>
58 +     <s:element name="parkingId" type="tns:ParkingId" minOccurs="1" maxOccurs="1"/></s:element>
59 +   </s:sequence>
60 +   <s:attribute name="status" use="required">
61 +     <s:annotation>
62 +       <s:documentation>
63 +         The status type reflects the overall status of the parking spot. Values:
64 +         available: the parking spot is empty;
65 +         not-available: the parking spot is occupied;
66 +         unknown: the current status of the parking spot is not known;
67 +       </s:documentation>
68 +     </s:annotation>
69 +     <s:simpleType>
70 +       <s:restriction base="s:string">
71 +         <s:enumeration value="available"/></s:enumeration>
72 +         <s:enumeration value="not-available"/></s:enumeration>
73 +         <s:enumeration value="unknown"/></s:enumeration>
74 +       </s:restriction>
75 +     </s:simpleType>
76 +   </s:attribute>
77 +   <s:attribute name="ttl" type="s:dateTime" use="optional">
78 +     <s:annotation>
79 +       <s:documentation>
80 +         The time to live is set as the deadline until which the status value is to be considered val
81 +       </s:documentation>
82 +     </s:annotation>
83 +   </s:attribute>

```

**eCHS GUI**

Create Partner

Name \*

Type \*  Operator  Provider  Nav Provider  **Parking Spot Op.**

Type Id \*

eMail

Networks  Select  
None selected

Group  Select  
None selected

- PSO-role in partner matrix:
  - roaming with CPOs
  - live-POI*
  - mandatory Type-ID

## 2. Added Tariff Exchange

Change type:	Enhancement
Impact:	<input checked="" type="checkbox"/> Specification, <input checked="" type="checkbox"/> API, <input type="checkbox"/> eCHS
Backward-Compatibility:	Tariff Exchange only with Partners using OCHP 1.4+
Forward-Compatibility:	-
Impact for partners:	-
Description:	<p>Possibility for partners to upload flexible tariff data specific to certain recipients to the eCHS and to reference those tariffs on connector level. Tariffs are exchanged according to the "RA"-roaming setting in the partner matrix.</p> <p>Methods: <i>UpdateTariffs</i> (CPO), <i>GetTariffUpdates</i> (EMP)</p> <p><i>category</i> in <i>ChargePointInfo</i> renamed <i>tariffId</i>, moved to <i>connectorType</i> and extended to string(15)</p>

### Specification

## Types for Tariff Data Exchange

These types are used to exchange tariff information between an operator and one or more providers.

### TariffInfo class

A Tariff Object consists of a list of one or more individual tariffs for individual recipients. One default individual tariff should always be defined without a specific recipient. Each individual tariff consists of tariff elements. These elements can be used to create complex Tariff structures.

Changes to a tariff can always only be made to the entire tariff object. That way it is ensured that there cannot be multiple conflicting tariffs referenced at the same connector.

Field Name	Field Type	Card.	Description
tariffId	TariffId	1	Uniquely identifies the tariff.
individualTariff	IndividualTariffType	+	Contains multiple individual tariffs dependant on intended recipient.

## IndividualTariffType *class*

Field Name	Field Type	Card.	Description
currency	string (3)	1	Currency of this tariff, ISO 4217 Code
tariffElement	TariffElementType	+	List of tariff elements.
recipient	string (5)	*	Provider-IDs of the intended recipients for this tariff. If no recipient is provided, this individual tariff is considered the default tariff.

**Note:** Tariffs are referenced on connector level by their tariff-ID only. Every EVSE Operator is advised to define one default individual tariff for each tariffId as a fallback in addition to the individual tariffs defined for certain recipients. A provider will only receive tariffs that they are recipient for.

## TariffElementType class

Field Name	Field Type	Card.	Description
priceComponent	PriceComponentType	+	List of price components that make up the pricing of this tariff
tariffRestriction	TariffRestrictionType	?	List of tariff restrictions

## PriceComponentType class

Field Name	Field Type	Card.	Description
billingItem	BillingItemType	1	Type of tariff dimension
itemPrice	float	1	price per unit for this tariff dimension (unit according to dimension, see BillingItemType description)
stepSize	int	1	Minimum amount to be billed. This unit will be billed in this stepSize blocks. For example: if type is time and stepSize is 300, then time will be billed in blocks of 5 minutes, so if 6 minutes is used, 10 minutes (2 blocks of stepSize) will be billed. In case of one-time payments, this is to be set to 1.0.

## TariffRestrictionType class

Field Name	Field Type	Card.	Description
regularHours	RegularHoursType	*	Regular hours that this tariff element should be valid for (maximum of 14 entries). If always valid (24/7), don't set (as this is a tariff restriction).
startDate	DateType	?	Start date, for example: 2015-12-24, valid from this day (midnight, i.e. including this day)
endDate	DateType	?	End date, for example: 2015-12-27, valid until this day (midnight, i.e. excluding this day)
minEnergy	float	?	Minimum used energy in kWh, for example 20.0, valid from this amount of energy is used
maxEnergy	float	?	Maximum used energy in kWh, for example 50.0, valid until this amount of energy is used
minPower	float	?	Minimum power in kW, for example 0.0, valid from this charging speed
maxPower	float	?	Maximum power in kW, for example 20.0, valid up to this charging speed
minDuration	int	?	Minimum duration in seconds, valid for a duration from x seconds
maxDuration	int	?	Maximum duration in seconds, valid for a duration up to x seconds

## TariffId

The tariff-ID follows a similar syntax to that of contract- and EVSE-IDs. The Operator-ID is followed by a 'T' that signifies a tariff and a unique instance of up to 9 characters.

## WSDL (API)

```

171 - <wsdl:message name="RequestLiveRoamingAuthorisationRequest">
188 + <wsdl:message name="UpdateTariffsRequest">
172 189 <wsdl:part name="parameters"
173 - element="tns:RequestLiveRoamingAuthorisationRequest" />
190 + element="tns:UpdateTariffsRequest" />
174 191 </wsdl:message>
175 - <wsdl:message name="RequestLiveRoamingAuthorisationResponse">
192 + <wsdl:message name="UpdateTariffsResponse">
176 193 <wsdl:part name="parameters"
177 - element="tns:RequestLiveRoamingAuthorisationResponse" />
194 + element="tns:UpdateTariffsResponse" />
178 195 </wsdl:message>
179 -
196 +
197 + <wsdl:message name="GetTariffUpdatesRequest">
198 + <wsdl:part name="parameters"
199 + element="tns:GetTariffUpdatesRequest" />
200 + </wsdl:message>
201 + <wsdl:message name="GetTariffUpdatesResponse">
202 + <wsdl:part name="parameters"
203 + element="tns:GetTariffUpdatesResponse" />
204 + </wsdl:message>

```

```

320 + <wsdl:operation name="UpdateTariffs">
321 + <wsdl:documentation>
322 + For all updates to tariff objects from the CIMS to the Clearing House and the EVSPs. Can contain sir
323 + </wsdl:documentation>
324 + <wsdl:input message="tns:UpdateTariffsRequest"></wsdl:input>
325 + <wsdl:output message="tns:UpdateTariffsResponse"></wsdl:output>
326 + </wsdl:operation>
327 + <wsdl:operation name="GetTariffUpdates">
328 + <wsdl:documentation>
329 + Downloads all tariff objects from the Clearing House that have been updated since a given date and t
330 + </wsdl:documentation>
331 + <wsdl:input message="tns:GetTariffUpdatesRequest"></wsdl:input>
332 + <wsdl:output message="tns:GetTariffUpdatesResponse"></wsdl:output>
333 + </wsdl:operation>

```

```

497 + <wsdl:operation name="UpdateTariffs">
498 + <soap:operation soapAction="http://ochp.eu/1.4/UpdateTariffs" />
499 + <wsdl:input>
500 + <soap:body use="literal" />
501 + </wsdl:input>
502 + <wsdl:output>
503 + <soap:body use="literal" />
504 + </wsdl:output>
505 + </wsdl:operation>
506 +
507 + <wsdl:operation name="GetTariffUpdates">
508 + <soap:operation soapAction="http://ochp.eu/1.4/GetTariffUpdates" />
432 509 <wsdl:input>
433 510 <soap:body use="literal" />

```

```

398 + <s:complexType name="TariffInfo">
399 + <s:annotation>
400 + <s:documentation>
401 + Information about a tariff referenced at connector level.
402 + </s:documentation>
403 + </s:annotation>
404 + <s:sequence>
405 + <s:element name="tariffId" minOccurs="1" maxOccurs="1" type="tns:TariffId">
406 + <s:annotation>
407 + <s:documentation>
408 + Alphanumeric. Identifies a tariff. Unique within one EVSE Operator. Must begin with
409 + </s:documentation>
410 + </s:annotation>
411 + </s:element>
412 + <s:element name="individualTariff" type="tns:IndividualTariffType" minOccurs="1" maxOccurs="unbounded">
413 + <s:annotation>
414 + <s:documentation>
415 + Element describing an individual tariff for a specific recipient. One default tariff
416 + </s:documentation>
417 + </s:annotation>
418 + </s:element>
419 + </s:sequence>
420 + </s:complexType>

```

```

1001 + <s:complexType name="IndividualTariffType">
1002 +   <s:sequence>
1003 +     <s:element name="tariffElement" type="tns:TariffElementType" minOccurs="1" maxOccurs="unbounded">
1004 +       <s:annotation>
1005 +         <s:documentation>
1006 +           Contains information about the pricing structure of the tariff element.
1007 +         </s:documentation>
1008 +       </s:annotation>
1009 +     </s:element>
1010 +     <s:element name="recipient" minOccurs="0" maxOccurs="unbounded">
1011 +       <s:annotation>
1012 +         <s:documentation>
1013 +           Alphanumeric. Identifies a recipient EMSP according to EMSP-ID without separators. I
1014 +         </s:documentation>
1015 +       </s:annotation>
1016 +       <s:simpleType>
1017 +         <s:restriction base="s:string">
1018 +           <s:minLength value="1"></s:minLength>
1019 +           <s:maxLength value="5"></s:maxLength>
1020 +           <s:pattern value="[A-Z0-9]{1,5}"></s:pattern>
1021 +         </s:restriction>
1022 +       </s:simpleType>
1023 +     </s:element>
1024 +     <s:element name="currency" minOccurs="1" maxOccurs="1">
1025 +       <s:annotation>
1026 +         <s:documentation>
1027 +           Alphabetic. The displayed and charged currency. Defined in ISO 4217 - Table A.1, alp
1028 +         </s:documentation>
1029 +       </s:annotation>
1030 +       <s:simpleType>
1031 +         <s:restriction base="s:string">
1032 +           <s:whiteSpace value="collapse"></s:whiteSpace>
1033 +         </s:restriction>
1034 +       </s:simpleType>
1035 +     </s:element>
1036 +   </s:sequence>
1037 + </s:complexType>

```

```

1039 + <s:complexType name="TariffElementType">
1040 +   <s:sequence>
1041 +     <s:element name="priceComponent" type="tns:PriceComponentType" minOccurs="1" maxOccurs="1">
1042 +       <s:annotation>
1043 +         <s:documentation>
1044 +           Contains information about the pricing structure of the tariff element.
1045 +         </s:documentation>
1046 +       </s:annotation>
1047 +     </s:element>
1048 +     <s:element name="tariffRestriction" type="TariffRestrictionType" minOccurs="1" maxOccurs="1">
1049 +       <s:annotation>
1050 +         <s:documentation>
1051 +           Contains information about when to apply the defined priceComponent / tariffElement.
1052 +         </s:documentation>
1053 +       </s:annotation>
1054 +     </s:element>
1055 +   </s:sequence>
1056 + </s:complexType>
1057 +
1058 + <s:complexType name="PriceComponentType">
1059 +   <s:sequence>
1060 +     <s:element name="billingItem" type="tns:BillingItemType" minOccurs="1" maxOccurs="1">
1061 +       <s:annotation>
1062 +         <s:documentation>
1063 +           What dimension is part of this tariff element.
1064 +         </s:documentation>
1065 +       </s:annotation>
1066 +     </s:element>
1067 +     <s:element name="itemPrice" type="s:float" minOccurs="1" maxOccurs="1">
1068 +       <s:annotation>
1069 +         <s:documentation>
1070 +           Price per unit of the billingItem in the given currency.
1071 +         </s:documentation>
1072 +       </s:annotation>
1073 +     </s:element>
1074 +     <s:element name="stepSize" type="s:int" minOccurs="1" maxOccurs="1">
1075 +       <s:annotation>
1076 +         <s:documentation>
1077 +           Minimum amount to be billed (billing will happen in stepSize increments). In case th
1078 +         </s:documentation>
1079 +       </s:annotation>
1080 +     </s:element>
1081 +   </s:sequence>
1082 + </s:complexType>

```

```

1084 + <s:complexType name="TariffRestrictionType">
1085 +   <s:sequence>
1086 +     <s:element name="regularHours" type="tns:RegularHoursType" minOccurs="0" maxOccurs="14">
1087 +       <s:annotation>
1088 +         <s:documentation>
1089 +           Regular hours, weekday based. No more than two periods per weekday.
1090 +         </s:documentation>
1091 +       </s:annotation>
1092 +     </s:element>
1093 +     <s:element name="startDateTime" type="tns:DateTimeType" minOccurs="0" maxOccurs="1">
1094 +       <s:annotation>
1095 +         <s:documentation>
1096 +           Valid from this date/time.
1097 +         </s:documentation>
1098 +       </s:annotation>
1099 +     </s:element>
1100 +     <s:element name="endDateTime" type="tns:DateTimeType" minOccurs="0" maxOccurs="1">
1101 +       <s:annotation>
1102 +         <s:documentation>
1103 +           Valid until this date/time.
1104 +         </s:documentation>
1105 +       </s:annotation>
1106 +     </s:element>
1107 +     <s:element name="minEnergy" type="s:float" minOccurs="1" maxOccurs="1">
1108 +       <s:annotation>
1109 +         <s:documentation>
1110 +           Minimum energy used in kWh. Valid from this amount used.
1111 +         </s:documentation>
1112 +       </s:annotation>
1113 +     </s:element>
1114 +     <s:element name="maxEnergy" type="s:float" minOccurs="1" maxOccurs="1">
1115 +       <s:annotation>
1116 +         <s:documentation>
1117 +           Maximum energy used in kWh. Valid up to this amount used.
1118 +         </s:documentation>
1119 +       </s:annotation>
1120 +     </s:element>
1121 +     <s:element name="minPower" type="s:float" minOccurs="1" maxOccurs="1">
1122 +       <s:annotation>
1123 +         <s:documentation>
1124 +           Minimum power in kw. Valid from this power.
1125 +         </s:documentation>
1126 +       </s:annotation>
1127 +     </s:element>
1128 +     <s:element name="maxPower" type="s:float" minOccurs="1" maxOccurs="1">
1129 +       <s:annotation>
1130 +         <s:documentation>
1131 +           Maximum power in kw. Valid up to this power.
1132 +         </s:documentation>
1133 +       </s:annotation>
1134 +     </s:element>

```

```

1135 +     <s:element name="minDuration" type="s:int" minOccurs="1" maxOccurs="1">
1136 +       <s:annotation>
1137 +         <s:documentation>
1138 +           Minimum duration in seconds. Valid from this duration.
1139 +         </s:documentation>
1140 +       </s:annotation>
1141 +     </s:element>
1142 +     <s:element name="maxDuration" type="s:int" minOccurs="1" maxOccurs="1">
1143 +       <s:annotation>
1144 +         <s:documentation>
1145 +           Maximum duration in seconds. Valid up to this duration.
1146 +         </s:documentation>
1147 +       </s:annotation>
1148 +     </s:element>
1149 +   </s:sequence>
1150 + </s:complexType>

```

```

1190 + <s:simpleType name="TariffId">
1191 +   <s:annotation>
1192 +     <s:documentation>
1193 +       The Tariff-ID uniquely references one tariff.
1194 +     </s:documentation>
1195 +   </s:annotation>
1196 +   <s:restriction base="s:string">
1197 +     <s:pattern value="[A-Za-z]{2}\*[A-Za-z0-9]{3}\*[Tt][A-Za-z0-9][A-Za-z0-9\*]{0,9}"></s:pattern>
1198 +     <s:pattern value="[A-Za-z]{2}[A-Za-z0-9]{3}[Tt][A-Za-z0-9][A-Za-z0-9\*]{0,9}"></s:pattern>
1199 +   </s:restriction>
1200 + </s:simpleType>

```

***eCHS GUI***

(none)

### 3. Changed LiveAuth to SingleAuth

Change type:	Enhancement
Impact:	<input checked="" type="checkbox"/> Specification, <input checked="" type="checkbox"/> API, <input type="checkbox"/> eCHS
Backward-Compatibility:	Functionality is the same. SingleAuth Requests are no longer tracked in eCHS and don't get a LiveAuth-ID.
Forward-Compatibility:	-
Impact for partners:	Partners using SingleAuth no longer have to provide an EVSE-ID with their request. Otherwise usage is the same.
Description:	<p>Since LiveAuth was being misunderstood by many partners, it has been renamed SingleAuth to clarify the fact that only the eCHS-internal whitelist is checked, not the EMP-system.</p> <p>The need for an EVSE-ID in the request was removed (which was not compatible with OCPP).</p>

#### Specification

### Messages for single authorisation

#### RequestSingleRoamingAuthorisation.req

This contains the field definition of the RequestSingleRoamingAuthorisation.req sent by CMS to the CHS.

Field Name	Field Type	Card.	Description
emtlid	EmtId	1	This contains the ID of the token which is to be validated.

#### RequestSingleRoamingAuthorisation.conf

This contains the field definition of the RequestSingleRoamingAuthorisation.conf sent by the CHS as response to the RequestSingleRoamingAuthorisation.req.

Field Name	Field Type	Card.	Description
result	Result	1	This contains the result of GetRoamingAuthorisationList.req.
roamingAuthorisationInfo	RoamingAuthorisationInfo	?	This contains the roaming authorisation record for the requested token, if the request was valid.

#### WSDL (API)

```

102 + <wsdl:message name="GetSingleRoamingAuthorisationRequest">
103 +   <wsdl:part name="parameters"
104 +     element="tns:GetSingleRoamingAuthorisationRequest" />
105 + </wsdl:message>
106 + <wsdl:message name="GetSingleRoamingAuthorisationResponse">
107 +   <wsdl:part name="parameters"
108 +     element="tns:GetSingleRoamingAuthorisationResponse" />
109 + </wsdl:message>

```

	334	+	<wsdl:operation name="GetSingleRoamingAuthorisation">
288	335		<wsdl:documentation>
289	336		A CMS may request the Clearing House to authorize one single token for a charging session
290	337		</wsdl:documentation>
291		-	<wsdl:input message="tns:RequestLiveRoamingAuthorisationRequest"></wsdl:input>
292		-	<wsdl:output message="tns:RequestLiveRoamingAuthorisationResponse"></wsdl:output>
	338	+	<wsdl:input message="tns:GetSingleRoamingAuthorisationRequest"></wsdl:input>
	339	+	<wsdl:output message="tns:GetSingleRoamingAuthorisationResponse"></wsdl:output>
293	340		</wsdl:operation>

	556	+	<s:element name="GetSingleRoamingAuthorisationRequest">
	557	+	<s:complexType>
	558	+	<s:sequence>
	559	+	<s:element name="emtId" type="tns:EmtId" minOccurs="1" maxOccurs="1">
	560	+	<s:annotation>
	561	+	<s:documentation>
	562	+	This contains the ID of the token which is to be retrieved from the CHS.
	563	+	</s:documentation>
	564	+	</s:annotation>
	565	+	</s:element>
	566	+	</s:sequence>
	567	+	</s:complexType>
	568	+	</s:element>
	569	+	
	570	+	<s:element name="GetSingleRoamingAuthorisationResponse">
	571	+	<s:complexType>
	572	+	<s:sequence>
	573	+	<s:element name="result" type="tns:Result" minOccurs="1" maxOccurs="1">
	574	+	<s:annotation>
	575	+	<s:documentation>
	576	+	This contains the result of GetSingleRoamingAuthorisation.req
465	577		</s:documentation>
466	578		</s:annotation>
467	579		</s:element>
468		-	<s:element name="liveAuthId" type="tns:LiveAuthId" minOccurs="0" maxOccurs="1">
	580	+	<s:element name="RoamingAuthorisationInfo" type="tns:RoamingAuthorisationInfo" minOccurs="0" maxOccurs="1">
469	581		<s:annotation>
470	582		<s:documentation>
471		-	Unique ID of the live authorisation request to the clearing house. Must be unique for each token.
	583	+	This contains the roaming authorisation record for the requested token, if it exists.
472	584		</s:documentation>
473	585		</s:annotation>
474	586		</s:element>

### eCHS GUI

LiveAuth Frontend page can be removed from eCHS once OCHP 1.3 is deprecated. OCHP 1.4 SingleAuth requests are no longer tracked here.



## Process Charge Data Records

Incoming roaming charge data records, held at the CHS are sent from the CHS to the MDM. Note that only CDRs that concern the particular EVSP are sent. The download has to be done in the following way:

- MDM sends GetCDRs.req PDU.
- CHS responds with a GetCDRs.conf PDU.
- MDM confirms or declines individual CDRs by sending ConfirmCDRs.req PDU.
- CHS responds with a ConfirmCDRs.conf PDU.

Furthermore, the CMS may download declined CDRs and attempt to fix any issues there were by re-uploading the CDRs as "revised" or to finally reject them (forego payment) by setting their status to "rejected".

- CMS may send the CheckCDRs.req PDU.
- CHS responds with CheckCDRs.conf PDU according to the status provided in the request.
- CMS may revise CDRs by sending AddCDRs.req PDU.
- CHS responds with an AddCDRs.conf PDU.
- CMS may reject individual CDRs by sending AddCDRs.req PDU.
- CHS responds with a AddCDRs.conf PDU.

### Implementation

All CDRs stay in the download queue (GetCDRs without explicitly stating the status) until their successful download was confirmed by a call to ConfirmCDRs.req. Declined CDRs may be handled in a separate process as described above.

## GetCDRs.req

This contains the field definition of the GetCDRs.req sent by a provider's system to the CHS.

Field Name	Field Type	Card.	Description
cdrStatus	CdrStatusType	?	Defines which status of CDRs to return: accepted, revised, rejected, approved. If not set, will return accepted and revised CDRs.

## CheckCDRs.req

This contains the field definition of the CheckCDRs.req sent by an EVSE operator's system to the CHS.

Field Name	Field Type	Card.	Description
cdrStatus	CdrStatusType	?	Defines which status of CDRs to return: declined, rejected, approved. If not set, will return declined CDRs.

## CheckCDRs.conf

This contains the field definition of the CheckCDRs.conf sent by the CHS as response to the CheckCDRs.req.

Field Name	Field Type	Card.	Description
result	Result	1	This contains the result of GetCDRs.req.
cdrInfoArray	Array(CDRInfo)	*	This contains the CDRs according to the status specified in the request.

## ConfirmCDRs.req

This contains the field definition of the ConfirmCDRs.req sent by a partner's system to the CHS.

Field Name	Field Type	Card.	Description
approved	CdrId, EvseId	*	This contains the CDR-ID and EVSE-ID for CDRs that have been approved by the EVSP.
declined	CdrId, EvseId	*	This contains the CDR-ID and EVSE-ID for CDRs that have been declined by the EVSP.

## CdrStatusType enum

Reflects the current status of the CDR. This is reflecting the status of internal processing in the clearing house. The value cannot be changed by the partner's systems directly. Implicit changes are made while uploading (including revised, rejected CDRs), approving or declining CDRs.

Value	Description
new	A new CDR before upload to the CHS.
accepted	An uploaded CDR was accepted by the CHS as plausible.
rejected	The checked CDR again rejected by the CHS and is to be archived.
declined	The CDR was declined by the owner (EVSP).
approved	The CDR was approved by the owner (EVSP).
revised	The CDR was revised by the CPO and uploaded again. Only previously accepted or declined CDRs can be revised.

## WSDL (API)

```

130 + <wsdl:message name="CheckCDRsInput">
131 +   <wsdl:part name="parameters"
132 +     element="tns:CheckCDRsRequest" />
133 + </wsdl:message>
134 + <wsdl:message name="CheckCDRsOutput">
135 +   <wsdl:part name="parameters"
136 +     element="tns:CheckCDRsResponse" />
137 + </wsdl:message>

250 + <wsdl:operation name="CheckCDRs">
251 +   <wsdl:documentation>
252 +     Cleared roaming charge data records, held at the CHS are sent from the CHS to the CMS. Note that onl
253 +   </wsdl:documentation>
254 +   <wsdl:input message="tns:CheckCDRsInput" />
255 +   <wsdl:output message="tns:CheckCDRsOutput" />
256 + </wsdl:operation>

397 + <wsdl:operation name="CheckCDRs">
398 +   <soap:operation soapAction="http://ochp.eu/1.4/CheckCDRs" />
399 +   <wsdl:input>
400 +     <soap:body use="literal" />
401 +   </wsdl:input>
402 +   <wsdl:output>
403 +     <soap:body use="literal" />
404 +   </wsdl:output>
405 + </wsdl:operation>
406 +

```

```

105 105 <s:complexType name="CdrStatusType">
106 106 <s:sequence>
107 107 <s:element name="CdrStatusType">
@@ -112,15 +112,17 @@
112 112 rejected; The checked CDR again rejected by the CHS and is to be archived.
113 113 owner declined; The CDR was declined by the owner (EVSP)
114 114 approved; The CDR was approved by the owner(EVSP)
115 115 + revised; The CDR was revised by the CPO after a rejection by the owner.
116 116 </s:documentation>
117 117 </s:annotation>
118 118 <s:simpleType>
119 119 <s:restriction base="s:string">
120 120 <s:enumeration value="new"></s:enumeration>
121 121 <s:enumeration value="accepted"></s:enumeration>
122 122 <s:enumeration value="rejected"></s:enumeration>
123 123 <s:enumeration value="owner declined"></s:enumeration>
124 124 <s:enumeration value="declined"></s:enumeration>
125 125 + <s:enumeration value="approved"></s:enumeration>
126 126 <s:enumeration value="revised"></s:enumeration>
127 127 </s:restriction>
128 128 </s:simpleType>
129 129 </s:element>

```

```

77 76 Defines the GetCDRs.req.
78 77 </s:documentation>
79 78 </s:annotation>
80 79 + <s:sequence>
81 80 + <s:element minOccurs="0" maxOccurs="1" name="cdrStatus" type="tns:CdrStatusType">
82 81 + <s:annotation>
83 82 + <s:documentation>
84 83 + Defines which status of CDRs to return to the EMP.
85 84 + Valid options: accepted, revised, approved, rejected.
86 85 + If not set, returns accepted and revised CDRs.
87 86 + </s:documentation>
88 87 + </s:annotation>
89 88 + </s:element>
90 89 + </s:sequence>

```

```

119 + <s:element name="CheckCDRsRequest">
120 + <s:complexType>
121 + <s:annotation>
122 + <s:documentation>
123 + Defines the CheckCDRs.req.
124 + </s:documentation>
125 + </s:annotation>
126 + <s:sequence>
127 + <s:element minOccurs="0" maxOccurs="1" name="cdrStatus" type="tns:CdrStatusType">
128 + <s:annotation>
129 + <s:documentation>
130 + Defines which status of CDRs to return to the CPO making the call.
131 + Valid options: declined (default, if not set), approved, rejected.
132 + </s:documentation>
133 + </s:annotation>
134 + </s:element>
135 + </s:sequence>
136 + </s:complexType>
137 + </s:element>
138 +
139 + <s:element name="CheckCDRsResponse">
140 + <s:complexType>
141 + <s:annotation>
142 + <s:documentation>
143 + This result will be returned on a CheckCDRs request.
144 + </s:documentation>
145 + </s:annotation>
146 + <s:sequence>
147 + <s:element minOccurs="1" maxOccurs="1" name="result" type="tns:Result">
148 + <s:annotation>
149 + <s:documentation>
150 + This contains the result of CheckCDRs.req
151 + </s:documentation>
152 + </s:annotation>
153 + </s:element>
154 + <s:element minOccurs="0" maxOccurs="unbounded" name="cdrInfoArray" type="tns:CDRInfo">
155 + <s:annotation>
156 + <s:documentation>
157 + This contains the CDRs with the defined status that have been sent by the ce

```

321	381	<s:element name="ConfirmCDRsRequest">
322	382	<s:complexType>
323	383	<s:sequence>
324	-	<s:element name="approved" type="tns:CDRInfo" minOccurs="0" maxOccurs="unbounded">
	384	<s:element name="approved" minOccurs="0" maxOccurs="unbounded">
325	385	<s:annotation>
326	386	<s:documentation>
327	-	This contains the CDRs that have been approved by the EVSP
	387	This contains the CDRs that have been approved by the EVSP.
328	388	</s:documentation>
329	389	</s:annotation>
	390	<s:complexType>
	391	<s:sequence>
	392	<s:element name="cdrId" minOccurs="1" maxOccurs="1" type="tns:CdrId">
	393	<s:annotation>
	394	<s:documentation>
	395	This contains the CDR-ID of the CDR to be ap
	396	</s:documentation>
	397	</s:annotation>
	398	</s:element>
	399	<s:element name="evseId" minOccurs="1" maxOccurs="1" type="tns:EvseId">
	400	<s:annotation>
	401	<s:documentation>
	402	This contains the EVSE-ID belonging to the C
	403	</s:documentation>
	404	</s:annotation>
	405	</s:element>
	406	</s:sequence>
	407	</s:complexType>
330	408	</s:element>

331	-	<s:element name="declined" type="tns:CDRInfo" minOccurs="0" maxOccurs="unbounded">
	409	<s:element name="declined" minOccurs="0" maxOccurs="unbounded">
332	410	<s:annotation>
333	411	<s:documentation>
334	412	This contains the CDRs that have been declined by the EVSP.
335	413	</s:documentation>
336	414	</s:annotation>
	415	<s:complexType>
	416	<s:sequence>
	417	<s:element name="cdrId" minOccurs="1" maxOccurs="1" type="tns:CdrId">
	418	<s:annotation>
	419	<s:documentation>
	420	This contains the CDR-ID of the CDR to be de
	421	</s:documentation>
	422	</s:annotation>
	423	</s:element>
	424	<s:element name="evseId" minOccurs="1" maxOccurs="1" type="tns:EvseId">
	425	<s:annotation>
	426	<s:documentation>
	427	This contains the EVSE-ID belonging to the C
	428	</s:documentation>
	429	</s:annotation>
	430	</s:element>
	431	</s:sequence>
	432	</s:complexType>
337	433	</s:element>

**eCHS GUI**

(none)

## 5. Changed – to \_ in connectorStandard enum

Change type: Bug Fix

Impact:  Specification,  API,  eCHS

Backward-Compatibility: eCHS translates between the two versions.

Forward-Compatibility: eCHS translates between the two versions.

Impact for partners: none

Description: Certain programming languages show problems when having to handle hyphens.

### Specification

See WSDL.

### WSDL (API)

398	-	IEC-62196-T1	The connector type is IEC 62196 Type 1 "SAE J1772"
399	-	IEC-62196-T1-COMBO	The connector type is Combo Type 1 based, DC
400	-	IEC-62196-T2	The connector type is IEC 62196 Type 2 "Mennekes"
401	-	IEC-62196-T2-COMBO	The connector type is Combo Type 2 based, DC
402	-	IEC-62196-T3A	The connector type is IEC 62196 Type 3A
403	-	IEC-62196-T3C	The connector type is IEC 62196 Type 3C "Scame"
404	-	DOMESTIC-A	The connector type is Standard/Domestic household, ty
405	-	DOMESTIC-B	The connector type is Standard/Domestic household, ty
406	-	DOMESTIC-C	The connector type is Standard/Domestic household, ty
407	-	DOMESTIC-D	The connector type is Standard/Domestic household, ty
408	-	DOMESTIC-E	The connector type is Standard/Domestic household, ty
409	-	DOMESTIC-F	The connector type is Standard/Domestic household, ty
410	-	DOMESTIC-G	The connector type is Standard/Domestic household, ty
411	-	DOMESTIC-H	The connector type is Standard/Domestic household, ty
412	-	DOMESTIC-I	The connector type is Standard/Domestic household, ty
413	-	DOMESTIC-J	The connector type is Standard/Domestic household, ty
414	-	DOMESTIC-K	The connector type is Standard/Domestic household, ty
415	-	DOMESTIC-L	The connector type is Standard/Domestic household, ty
416	-	TESLA-R	The connector type is Tesla Connector "Roadster"-type
417	-	TESLA-S	The connector type is Tesla Connector "Model-S"-type
418	-	IEC-60309-2-single-16	The connector type is IEC 60309-2 Industrial Connecto
419	-	IEC-60309-2-three-16	The connector type is IEC 60309-2 Industrial Connecto
420	-	IEC-60309-2-three-32	The connector type is IEC 60309-2 Industrial Connecto
421	-	IEC-60309-2-three-64	The connector type is IEC 60309-2 Industrial Connecto
424	+	IEC_62196_T1	The connector type is IEC 62196 Type 1 "SAE J1772"
425	+	IEC_62196_T1_COMBO	The connector type is Combo Type 1 based, DC
426	+	IEC_62196_T2	The connector type is IEC 62196 Type 2 "Mennekes"
427	+	IEC_62196_T2_COMBO	The connector type is Combo Type 2 based, DC
428	+	IEC_62196_T3A	The connector type is IEC 62196 Type 3A
429	+	IEC_62196_T3C	The connector type is IEC 62196 Type 3C "Scame"
430	+	DOMESTIC_A	The connector type is Standard/Domestic household, ty
431	+	DOMESTIC_B	The connector type is Standard/Domestic household, ty
432	+	DOMESTIC_C	The connector type is Standard/Domestic household, ty
433	+	DOMESTIC_D	The connector type is Standard/Domestic household, ty
434	+	DOMESTIC_E	The connector type is Standard/Domestic household, ty
435	+	DOMESTIC_F	The connector type is Standard/Domestic household, ty
436	+	DOMESTIC_G	The connector type is Standard/Domestic household, ty
437	+	DOMESTIC_H	The connector type is Standard/Domestic household, ty
438	+	DOMESTIC_I	The connector type is standard/Domestic household, ty
439	+	DOMESTIC_J	The connector type is Standard/Domestic household, ty
440	+	DOMESTIC_K	The connector type is Standard/Domestic household, ty
441	+	DOMESTIC_L	The connector type is Standard/Domestic household, ty
442	+	TESLA_R	The connector type is Tesla Connector "Roadster"-type
443	+	TESLA_S	The connector type is Tesla Connector "Model-S"-type
444	+	IEC_60309_2_single_16	The connector type is IEC 60309-2 Industrial Connecto
445	+	IEC_60309_2_three_16	The connector type is IEC 60309-2 Industrial Connecto
446	+	IEC_60309_2_three_32	The connector type is IEC 60309-2 Industrial Connecto
447	+	IEC_60309_2_three_64	The connector type is IEC 60309-2 Industrial Connecto

### eCHS GUI

(none)

## 6. Combined address and parking related fields

Change type:	Enhancement
Impact:	<input checked="" type="checkbox"/> Specification, <input checked="" type="checkbox"/> API, <input type="checkbox"/> eCHS
Backward-Compatibility:	eCHS "unpacks" the new address-element
Forward-Compatibility:	eCHS combines existing elements, CDRs missing address-information are completed from EVSE-database
Impact for partners:	
Description:	All address related fields (in <i>ChargePointInfo</i> , <i>CdrInfo</i> ) got combined into a new <i>AddressType</i> element, all parking related fields got combined into a <i>ParkingSpotType</i> element.  <i>ParkingSpotType</i> allows referencing of a unique ID by a PSO.

### Specification

#### AddressType class

This class contains all address related information in regards to a charge point.

Field Name	Field Type	Card.	Description
houseNumber	string(6)	?	Alphanumeric, for example "10", "255B". Characters: [A-Z], [0-9], -,
address	string(45)	1	Alphanumeric, for example "Av. Saint-Jean". Optionally also containing the house number if not in field houseNumber.
city	string(45)	1	Alphabetic, in the language defined in locationNameLang
zipCode	string(10)	1	Alphanumeric, Examples: "60439", "8011 PK". Without leading country code. Characters: [A-Z], [0-9], -,
country	string(3)	1	Alpha, three characters. ISO 3166 country code

#### ParkingSpotType class

This class contains all parking related information. If a parkingId is given, this ID can be used to associate parking spot live information from a PSO with this EVSE.

Field Name	Field Type	Card.	Description
parkingId	ParkingId	?	Globally unique identifier for this parking spot.
restriction	RestrictionType	*	Restrictions applying to the usage of the parking spot. If set, should include the restrictions to EVSE-usage as well.
floorlevel	string(4)	?	Alphanumeric. Level on which the charge station is located (in garage buildings) in the locally displayed numbering scheme. Examples: "-2", "P-5", "2", "+5"
parkingSpotNumber	string(5)	?	Alphanumeric. Locally displayed parking slot number. Examples: "10", "251", "B25", "P-234"

## WSDL (API)

```

792 + <s:complexType name="AddressType">
793 +   <s:sequence>
794 +     <s:element name="houseNumber" minOccurs="0" maxOccurs="1">
795 +       <s:annotation>
796 +         <s:documentation>
797 +           Alphanumeric, for example "10", "2550".
798 +         </s:documentation>
799 +       </s:annotation>
800 +       <s:simpleType>
801 +         <s:restriction base="s:string">
802 +           <s:minLength value="1"></s:minLength>
803 +           <s:maxLength value="6"></s:maxLength>
804 +           <s:pattern value="[A-Z0-9 \-]{1,6}"></s:pattern>
805 +         </s:restriction>
806 +       </s:simpleType>
807 +     </s:element>
808 +     <s:element name="address" minOccurs="1" maxOccurs="1">
809 +       <s:annotation>
810 +         <s:documentation>
811 +           Alphanumeric; for example "Av. Saint-Jean". Optionally also containing the house num
812 +         </s:documentation>
813 +       </s:annotation>
814 +       <s:simpleType>
815 +         <s:restriction base="s:string">
816 +           <s:minLength value="2"></s:minLength>
817 +           <s:maxLength value="45"></s:maxLength>
818 +         </s:restriction>
819 +       </s:simpleType>
820 +     </s:element>
821 +     <s:element name="city" minOccurs="1" maxOccurs="1">
822 +       <s:annotation>
823 +         <s:documentation>
824 +           Alphabetic, in the language defined in locationNameLang
825 +         </s:documentation>
826 +       </s:annotation>
827 +       <s:simpleType>
828 +         <s:restriction base="s:string">
829 +           <s:minLength value="1"></s:minLength>
830 +           <s:maxLength value="45"></s:maxLength>
831 +         </s:restriction>
832 +       </s:simpleType>
833 +     </s:element>
834 +     <s:element name="zipCode" minOccurs="1" maxOccurs="1">
835 +       <s:annotation>
836 +         <s:documentation>
837 +           Alphanumeric, examples: "60439", "8011 PK". Without leading country code.
838 +         </s:documentation>
839 +       </s:annotation>
840 +       <s:simpleType>
841 +         <s:restriction base="s:string">
842 +           <s:pattern value="[A-Z0-9 \-]{1,10}"></s:pattern>
843 +         </s:restriction>
844 +       </s:simpleType>
845 +     </s:element>

```

```

846 +     <s:element name="country" minOccurs="1" maxOccurs="1">
847 +       <s:annotation>
848 +         <s:documentation>
849 +           Alpha, three characters. Iso 3166 country code
850 +         </s:documentation>
851 +       </s:annotation>
852 +       <s:simpleType>
853 +         <s:restriction base="s:string">
854 +           <s:pattern value="[A-Z]{3}"></s:pattern>
855 +         </s:restriction>
856 +       </s:simpleType>
857 +     </s:element>
858 +   </s:sequence>
859 + </s:complexType>

```

```

215 + <:complexType name="ParkingSpotInfo">
216 +   <:annotation>
217 +     <:documentation>
218 +       Static POI data regarding a parking spot.
219 +     </:documentation>
220 +   </:annotation>
221 +   <:sequence>
222 +     <:element name="parkingId" type="tns:ParkingId" minOccurs="1" maxOccurs="1">
223 +       <:annotation>
224 +         <:documentation>
225 +           Globally unique identifier for this parking spot.
226 +         </:documentation>
227 +       </:annotation>
228 +     </:element>
229 +     <:element name="restriction" type="tns:RestrictionType" minOccurs="0" maxOccurs="unbounded">
230 +       <:annotation>
231 +         <:documentation>
232 +           Restrictions to the usage of this parking spot.
233 +         </:documentation>
234 +       </:annotation>
235 +     </:element>
236 +     <:element name="floorLevel" minOccurs="0" maxOccurs="1">
237 +       <:annotation>
238 +         <:documentation>
239 +           Alphanumeric. Level on which the charge station is located (in garage buildings) in
240 +         </:documentation>
241 +       </:annotation>
242 +       <:simpleType>
243 +         <:restriction base="s:string">
244 +           <:pattern value="[A-Z0-9\-\+/\]{1,4}"></:pattern>
245 +         </:restriction>
246 +       </:simpleType>
247 +     </:element>
248 +     <:element name="parkingSpotNumber" minOccurs="0" maxOccurs="1">
249 +       <:annotation>
250 +         <:documentation>
251 +           Alphanumeric. Locally displayed parking slot number. Examples: "10", "251", "B25", "F
252 +         </:documentation>
253 +       </:annotation>
254 +       <:simpleType>
255 +         <:restriction base="s:string">
256 +           <:pattern value="[A-Z0-9\-\+/\]{1,5}"></:pattern>
257 +         </:restriction>
258 +       </:simpleType>
259 +     </:element>
260 +   </:sequence>
261 + </:complexType>

```

**eCHS GUI**

(none)

## 7. Changed *ParkingRestrictionType* to *RestrictionType*

Change type:	Enhancement
Impact:	<input checked="" type="checkbox"/> Specification, <input checked="" type="checkbox"/> API, <input type="checkbox"/> eCHS
Backward-Compatibility:	eCHS combines all restrictions on the EVSE level. <i>carsharing</i> restriction is set to <i>customers</i>
Forward-Compatibility:	eCHS uses restrictions for both EVSE and parkingspot (only if there is further parking spot information defined, otherwise not)
Impact for partners:	Possibility to have distinct restrictions apply to the EVSE and the parking spot.
Description:	The former <i>ParkingRestrictionType</i> may now be separately used for EVSE and parking spot and may now contain <i>carsharing</i> .

### Specification

#### RestrictionType enum

This value, if provided, represents the restrictions to the usage of the charging station or parking spot for different purposes.

Value	Description
evonly	reserved parking spot for electric vehicles
plugged	parking allowed only while plugged in (charging)
disabled	reserved parking spot for disabled persons with valid ID
customers	charging / parking for customers / guests only, for example in case of a hotel or shop
motorcycles	parking spot only suitable for (electric) motorcycles, scooters or bicycles
carsharing	charging / parking only for carsharing vehicles

### WSDL (API)

```

571 | - | <s:complexType name="ParkingRestrictionType">
572 | 606 | + | <s:complexType name="RestrictionType">
573 | 607 | - | <s:sequence>
574 | 608 | + | <s:element name="ParkingRestrictionType">
575 | 609 | - | <s:element name="RestrictionType">
576 | 610 | - | <s:documentation>
577 | 611 | + | * evonly reserved parking spot for electric vehicles
578 | 612 | + | * evonly reserved parking spot for electric vehicles
579 | 613 | + | * plugged parking allowed only while plugged in (and charging)
580 | 614 | - | * disabled reserved parking spot for disabled people with valid ID
581 | 615 | - | * customers parking spot for customer/guests only, for example in case of a hotel or
582 | 616 | - | * motorcycles parking spot only suitable for (electric) motorcycles or scooters
583 | 617 | + | * customers parking or charging for customer/guests only, for example in case of a h
584 | 618 | + | * motorcycles parking spot only suitable for (electric) motorcycles, scooters or bic
585 | 619 | + | * carsharing charging / parking only for carsharing vehicles
586 | 620 | - | </s:documentation>
587 | 621 | - | </s:annotation>
588 | 622 | - | <s:simpleType>
589 | 623 | - | @@ -588,6 +625,7 @@
590 | 624 | - |
591 | 625 | + | <s:enumeration value="disabled"></s:enumeration>
592 | 626 | + | <s:enumeration value="customers"></s:enumeration>
593 | 627 | + | <s:enumeration value="motorcycles"></s:enumeration>
594 | 628 | + | <s:enumeration value="carsharing"></s:enumeration>
595 | 629 | - | </s:restriction>
596 | 630 | - | </s:simpleType>
597 | 631 | - | </s:element>

```

***eCHS GUI***

(none)

## 8. Added multiple data fields to CDR-Info

Change type:	Enhancement
Impact:	<input checked="" type="checkbox"/> Specification, <input checked="" type="checkbox"/> API, <input type="checkbox"/> eCHS
Backward-Compatibility:	optional data not available in OCHP 1.3, <i>BillingItem reservation</i> → <i>serviceFee</i> , <i>reservationtime</i> → <i>usagetime</i>
Forward-Compatibility:	-
Impact for partners:	-
Description:	Introduction of reservation in OCHPdirect makes it necessary for partners to be able to bill their customers for it. Thus, <i>reservation</i> and <i>reservationtime</i> are added to <i>BillingItemType</i> . Further information for European roaming necessitates <i>taxrate</i> , <i>currency</i> . Easier handling: <i>totalCost</i>

### Specification

#### BillingItemType enum

The billing items for charging periods and tariffs.

Value	Description
parkingtime	Price for the time of parking. The billingValue represents the time in hours.
usagetime	Price for the time of EVSE usage. The billingValue represents the time in hours.
energy	Price for the consumed energy. The billingValue represents the energy in kilowatt-hours.
power	Price for the used power level. The billingValue represents the maximum power in kilowatts.
serviceFee	General service fee per charging process. The billingValue represents a multiplier and thus has to be set to "1.0".
reservation	One time fee for a reservation of the EVSE. The billingValue represents a multiplier and thus has to be set to "1.0".
reservationtime	Price for the duration of a reservation. The billingValue represents the time in hours.

#### CDRInfo class

Contains all information concerning a Charge Data Record

[...]

totalCost	float	?	Total cost for the entire charging process. Should always equal the sum of the individual periodCosts.
currency	string(3)	1	Alphabetic. The displayed and charged currency. Defined in ISO 4217 - Table A.1, alphabetic list.

## CdrPeriodType class

[...]

taxrate	int	?	Tax rate in percent to be paid for the charging process in the EVSE operator's country.
---------	-----	---	---

### WSDL (API)

265	317	<s:complexType name="BillingItemType">
266	318	<s:sequence>
267	319	<s:element name="BillingItemType">
268	320	<s:simpleType>
269	321	<s:annotation>
270	322	<s:documentation>
271	323	parkingtime; Price for parking, The billingvalue represents the time in hour
272	324	usagetime; Price for the time of EVSE usage. The billingvalue represents the
273	325	energy; Price for the consumed energy. The billingvalue represents the energ
274	326	power; Price for the consumed energy. The billingvalue represents the maximu
275	-	serviceFee; General service fee per charging process. The billingvalue repre
	327	serviceFee; General service fee per charging process. The billingvalue repre
	328	reservation; One time fee for a reservation of the EVSE. The billingvalue re
	329	reservationtime; Price for the duration of a reservation. The billingvalue r
276	330	</s:documentation>
277	331	</s:annotation>
278	332	<s:restriction base="s:string">
279	333	<s:enumeration value="parkingtime"></s:enumeration>
280	334	<s:enumeration value="usagetime"></s:enumeration>
281	335	<s:enumeration value="energy"></s:enumeration>
282	336	<s:enumeration value="power"></s:enumeration>
283	337	<s:enumeration value="serviceFee"></s:enumeration>
	338	<s:enumeration value="reservation"></s:enumeration>
	339	<s:enumeration value="reservationtime"></s:enumeration>
284	340	</s:restriction>
285	341	</s:simpleType>
286	342	</s:element>

### eCHS GUI

(none)

## 9. Changed *operatingTimes* and *accessTimes*

Change type:	Enhancement
Impact:	<input checked="" type="checkbox"/> Specification, <input checked="" type="checkbox"/> API, <input type="checkbox"/> eCHS
Backward-Compatibility:	Copy <i>openingTimes</i> to <i>operatingTimes</i>
Forward-Compatibility:	Copy <i>operatingTimes</i> to <i>openingTimes</i> , <i>closedCharging</i> = false
Impact for partners:	Less confusion as to which type to use.
Description:	<i>operatingTimes</i> and <i>accessTimes</i> have been simplified to only contain <i>openingTimes</i> which in turn may contain a Boolean <i>closedCharging</i> (default = false)

### Specification

#### HoursType class

Opening hours for the charge point.

Field Name	Field Type	Card.	Description
<i>Choice: one of two</i>			
> regularHours	regularHoursType	*	Regular hours, weekday based. Should not be set for representing 24/7 as this is the most common case.
> twentyfourseven	boolean	1	True to represent 24 hours per day and 7 days per week, except the given exceptions. May be set to false if opening hours are defined only by exceptionalOpenings.
closedCharging	boolean	1	Should be set to true in case an EV can be charged when plugged in during off-times (i.e. when the location is closed according to the specified hours).
exceptionalOpenings	exceptionalPeriodType	*	Exceptions for specified calendar dates, time-range based. Periods the station is operating/accessible. For irregular hours or as addition to regular hours. May overlap regular rules.
exceptionalClosings	exceptionalPeriodType	*	Exceptions for specified calendar dates, time-range based. Periods the station is not operating/accessible. Overwriting regularHours and twentyfourseven. Should not overlap exceptionalOpenings.

### WSDL (API)

422	-	</s:element>
423	-	<s:element name="operatingTimes" type="tns:HoursType" minOccurs="0" maxOccurs="1">
298	+	<s:element name="openingTimes" type="tns:HoursType" minOccurs="0" maxOccurs="1">
424	299	<s:annotation>
425	300	<s:documentation>
426	301	The times the EVSE is operating and can be used for charging. Must not be provided i
427	302	</s:documentation>
428	303	</s:annotation>
429	304	</s:element>
430	-	<s:element name="accessTimes" type="tns:HoursType" minOccurs="0" maxOccurs="1">
431	-	<s:annotation>

	514	+	<s:element name="closedCharging" type="tns:boolean" minOccurs="1" maxOccurs="1">
	515	+	<s:annotation>
	516	+	<s:documentation>
	517	+	Should be set to true in case an EV can be charged when plugged in during off-times
485	518		</s:documentation>
486	519		</s:annotation>
487	520		</s:element>

**eCHS GUI**

(none)

## 10. Other changes

Change type:	Enhancement
Impact:	<input checked="" type="checkbox"/> Specification, <input checked="" type="checkbox"/> API, <input type="checkbox"/> eCHS
Backward-Compatibility:	-
Forward-Compatibility:	-
Impact for partners:	-
Description:	<p>Changed <i>EmtId</i> default representation to "plain" (was "sha-160")</p> <p>Added possibility to add charging stations / points with limited accessibility: <i>GeneralLocationType</i> may contain "private".</p> <p>Added <i>maxReservation</i> to <i>ChargePointInfo</i> to enable CPOs to specify a maximum allowable duration for a reservation.</p>

### Specification

maxReservation	float	?	If a reservation of this charge point is possible, this is the maximum duration the CPO will allow a reservation for (in minutes). Recommendation: 30 minutes.
----------------	-------	---	--

### WSDL (API)

23	42	<code>&lt;s:complexType name="EmtId"&gt;</code>
24	43	<code>&lt;s:sequence&gt;</code>
25	44	<code>&lt;s:element name="instance" minOccurs="1" maxOccurs="1"&gt;</code>
		<code>@@ -64,7 +83,7 @@</code>
64	83	<code>&lt;/s:simpleType&gt;</code>
65	84	<code>&lt;/s:element&gt;</code>
66	85	<code>&lt;/s:sequence&gt;</code>
67	-	<code>&lt;s:attribute name="representation" default="sha-160"&gt;</code>
	86	<code>&lt;s:attribute name="representation" default="plain"&gt;</code>
68	87	<code>.....</code>

542	575	<s:complexType name="GeneralLocationType">
543	576	<s:sequence>
544	577	<s:element name="GeneralLocationType">
545	578	<s:annotation>
546	579	<s:documentation>
547	580	* on-street parking in public space
548	581	* parking-garage multistorey car park
549	582	* underground-garage multistorey car park, mainly underground
550	583	* parking-lot a cleared area that is intended for parking vehicles, i.e. at super ma
551	584	* other none of the given possibilities
552	585	* unknown parking location type is not known by the operator
553	586	* private located in private or corporate grounds, may not be accessible to the publ
553	587	</s:documentation>
554	588	</s:annotation>
555	589	<s:simpleType>
556	590	<s:annotation>
557	591	</s:annotation>
558	592	<s:restriction base="s:string">
559	593	<s:enumeration value="on-street"></s:enumeration>
560	594	<s:enumeration value="parking-garage"></s:enumeration>
561	595	<s:enumeration value="underground-garage"></s:enumeration>
562	596	<s:enumeration value="parking-lot"></s:enumeration>
563	597	<s:enumeration value="other"></s:enumeration>
564	598	<s:enumeration value="unknown"></s:enumeration>
565	599	<s:enumeration value="private"></s:enumeration>
565	600	</s:restriction>
566	601	</s:simpleType>
567	602	</s:element>
568	603	</s:sequence>
569	604	</s:complexType>
---	---	

### eCHS GUI

(none)

## 11. Specification document

Change type: Bug Fix, Enhancement

Impact: Specification, API, eCHS

Backward-Compatibility: n/a

Forward-Compatibility: n/a

Impact for partners: none

Description: Various spelling errors in the specification document get corrected.  
Added more information and clarifications to different definitions.

## Changes to OCHPdirect

Note: there is no direct interoperability between distinct versions of OCHPdirect. A partner is advised to run multiple version endpoints to cover as big a portion of roaming partners as possible. Running a 0.1-Endpoint in addition to a 0.2-Endpoint should only require minimal effort to remove the data not compatible with the older version.

Version information is exchanged through the eCHS in the now mandatory *namespaceURL* element of the endpoint exchange. All endpoints uploaded via OCHP 1.3 without this information will be treated as though OCHP 1.3 was defined as namespace in this position.

### 12. Extended *InformProvider* method

Change type: Enhancement

Impact:  Specification,  API,  eCHS

Impact for partners: Enhanced possibility for information exchange through *InformProvider*.

The CPO Endpoint has to handle an *InformProvider* request from the EMP as a new method.

Description: *InformProvider* is now a mandatory part of OCHPdirect for all partners. Additionally, it can now be requested from the CPO-Endpoint by the EMP if they wish to collect additional information.

Further information not before part of *InformProvider* has been added to enable the transfer of meter readings as well as parts of the CDR information.

### Specification

#### Inform a provider about a charging process (both interfaces)

The provider must get informed by the operator about any status updates to an OCHP-direct charging process. This should cover at least the start, change and end functionalities. The operator's backend must make use of a threshold in order to avoid too many messages (it is recommended not to send *InformProvider* more than once every 15 minutes, unless the parameters of the charging process were changed or an *InformProvider* was requested by the Provider).

The provider may request additional *InformProvider* messages from the CPO to follow the progress of the charging process. In order not to overwhelm any system involved, it is strongly recommended not to exceed a frequency of 15 minutes between each such request.

#### Send charging process information of a provider's customer

When a status update to a charging process gets available, the operator informs the concerned provider.

- CMS sends the *InformProviderMessage* PDU.
- MDM responds with a *InformProvider.conf* PDU.

## Request charging process information for a provider's customer

When a customer requests updated charging information to their charging process or when the provider deems it necessary to collect updated information on their behalf. May be called directly after SelectEvse to collect current meter values for the selected EVSE.

- MDM sends the InformProvider.req PDU.
- CMS responds with a InformProviderMessage PDU.

### InformProviderMessage

This contains the field definition of the InformProviderMessage sent by the CMS towards the MDM.

Field Name	Field Type	Card.	Description
result	DirectResult	1	This contains the result of InformProvider.req. Also to be sent for InformProviderMessages initiated by the CPO.
message	DirectMessage	1	The operation that triggered the operator to send this message.
evseId	EvseId	1	The charge point which is used for this charging process.
contractId	ContractId	1	Contract-ID to which the charge point is assigned.
directId	DirectId	1	The session id for this direct charging process.
ttl	DateTimeType	?	On success the timeout for this session.
stateOfCharge	float	?	Current state of charge of the connected EV in percent.
maxPower	float	?	Maximum authorised power in kilowatts. Example: "3.7", "8", "15"
maxEnergy	float	?	Maximum authorised energy in kilowatthours. Example: "5.5", "20", "85"
minEnergy	float	?	Minimum required energy in kilowatthours. Example: "5.5", "20", "85"
departure	DateTimeType	?	Scheduled time of departure. Only to be used for smart charging purposes.
currentPower	float	?	The currently supplied power limit in kilowatts in case of load management. Example: "3.7", "8", "15"
chargedEnergy	float	?	The amount of energy in kilowatthours transferred during this charging process. Example: "5.5", "20", "85"
meterReading	MeterReading	?	The current meter value as displayed on the meter with corresponding timestamp to enable displaying it to the user. Example: "12345.67"
chargingPeriods	CdrPeriodType	*	Can be used to transfer billing information to the provider in near real time.
currentCost	float	?	The total cost of the charging process that will be billed by the operator up to this point.
currency	string(3)	?	The displayed and charged currency. Defined in ISO 4217 - Table A.1, alphabetic list.

## InformProvider.req

This contains the field definition of the InformProvider.req sent by the MDM towards the CMS.

Field Name	Field Type	Card.	Description
directId	DirectId	1	The session id for this direct charging process.

## InformProvider.conf

This contains the field definition of the InformProvider.conf sent by the MDM as a response to InformProviderMessage.

Field Name	Field Type	Card.	Description
result	DirectResult	1	This contains the result of InformProviderMessage.

## MeterReading class

Contains one meter reading from a charge point.

Field Name	Field Type	Card.	Description
meterValue	float	1	The meter reading as transferred by the charging station (OCPP default unit: Wh).
meterTime	LocalDateTimeType	1	The exact local time stamp of the time the meter reading was taken (belonging to the meter reading, e.g. as transferred by OCPP).

## WSDL (API)

```

709 - <wsdl:message name="InformProviderInput">
802 + <wsdl:message name="InformProviderEMPInput">
710 803 <wsdl:part name="parameters"
711 - element="tns:InformProviderRequest" />
804 + element="tns:InformProviderMessage" />
712 805 </wsdl:message>
713 - <wsdl:message name="InformProviderOutput">
806 + <wsdl:message name="InformProviderEMPOutput">
714 807 <wsdl:part name="parameters"
715 808 element="tns:InformProviderResponse" />
716 809 </wsdl:message>
810 +
811 + <wsdl:message name="InformProviderCPOInput">
812 + <wsdl:part name="parameters"
813 + element="tns:InformProviderRequest" />
814 + </wsdl:message>
815 + <wsdl:message name="InformProviderCPOOutput">
816 + <wsdl:part name="parameters"
817 + element="tns:InformProviderMessage" />
818 + </wsdl:message>
717 819

```

491	-	<s:element name="InformProviderRequest">
	534	+ <s:element name="InformProviderMessage">
492	535	<s:complexType>
493	536	<s:sequence>
	537	+ <s:element minOccurs="1" maxOccurs="1" name="result" type="tns:DirectResult">
	538	+ <s:annotation>
	539	+ <s:documentation>
	540	+ This contains the result if the EMP requested InformProvider
	541	+ </s:documentation>
	542	+ </s:annotation>
	543	+ </s:element>
494	544	<s:element minOccurs="1" maxOccurs="1" name="message" type="tns:DirectMessage">
495	545	<s:annotation>
496	546	<s:documentation>
 @@ -533,13 +583,41 @@ <a href="http://creativecommons.org/licenses/by-nd/4.0/">http://creativecommons.org/licenses/by-nd/4.0/</a> .		
533	583	</s:documentation>
534	584	</s:annotation>
535	585	</s:element>
	586	+ <s:element name="maxCurrent" type="s:float" minOccurs="0" maxOccurs="1">
	587	+ <s:annotation>
	588	+ <s:documentation>
	589	+ Maximum authorised current in ampere. Example: "16", "25", "
	590	+ </s:documentation>
	591	+ </s:annotation>
	592	+ </s:element>
	593	+ <s:element name="onePhase" type="s:boolean" minOccurs="0" maxOccurs="1">
	594	+ <s:annotation>
	595	+ <s:documentation>
	596	+ Marks an AC-charging session to be limited to one-phase char
	597	+ </s:documentation>
	598	+ </s:annotation>
	599	+ </s:element>
	607	+ <s:element name="minEnergy" type="s:float" minOccurs="0" maxOccurs="1">
	608	+ <s:annotation>
	609	+ <s:documentation>
	610	+ Minimum required energy in kilowatthours. Example: "5.5", "2
	611	+ </s:documentation>
	612	+ </s:annotation>
	613	+ </s:element>
	614	+ <s:element name="departure" type="tns:DateTimeType" minOccurs="0" maxOccurs="1">
	615	+ <s:annotation>
	616	+ <s:documentation>
	617	+ Scheduled time of departure. Only to be used for smart charg
	618	+ </s:documentation>
	619	+ </s:annotation>
	620	+ </s:element>

```

635 +         <s:element name="meterReading" type="tns:MeterReading" minOccurs="0" maxOccurs="1">
636 +             <s:annotation>
637 +                 <s:documentation>
638 +                     The current meter value (in kWh) as displayed on the meter t
639 +                 </s:documentation>
640 +             </s:annotation>
641 +         </s:element>
642 +         <s:element name="chargingPeriods" type="tns:CdrPeriodType" minOccurs="0" maxOccurs="
643 +             <s:annotation>
644 +                 <s:documentation>
645 +                     One period per item on the bill.
646 +                 </s:documentation>
647 +             </s:annotation>
648 +         </s:element>
649 +         <s:element name="currentCost" type="s:float" minOccurs="0" maxOccurs="1">
650 +             <s:annotation>
651 +                 <s:documentation>
652 +                     The total cost of the charging process up to this point.
653 +                 </s:documentation>
654 +             </s:annotation>
655 +         </s:element>
656 +         <s:element name="currency" type="s:string" minOccurs="0" maxOccurs="1" nillable="tru
657 +             <s:annotation>
658 +                 <s:documentation>
659 +                     Alphabetic. The displayed and charged currency. Defined in I
660 +                 </s:documentation>
661 +             </s:annotation>
662 +             <s:restriction base="s:string">
663 +                 <s:pattern value="[A-Z]*"></s:pattern>
664 +                 <s:minLength value="3"></s:minLength>
665 +                 <s:maxLength value="3"></s:maxLength>
666 +             </s:restriction>
667 +         </s:element>

```

```

630 +
631 +         <s:element name="InformProviderRequest">
632 +             <s:complexType>
633 +                 <s:sequence>
634 +                     <s:element minOccurs="1" maxOccurs="1" name="directId" type="tns:DirectId">
635 +                         <s:annotation>
636 +                             <s:documentation>
637 +                                 The session id for this direct charging process.
638 +                             </s:documentation>
639 +                         </s:annotation>
640 +                     </s:element>
641 +                 </s:sequence>
642 +             </s:complexType>
643 +         </s:element>

```

```

704 +         <s:complexType name="MeterReading">
705 +             <s:annotation>
706 +                 <s:documentation>
707 +                     Most current meter readings transferred from the charge point including time stamp.
708 +                 </s:documentation>
709 +             </s:annotation>
710 +             <s:sequence>
711 +                 <s:element minOccurs="1" maxOccurs="1" name="meterValue" type="s:float">
712 +                     <s:annotation>
713 +                         <s:documentation>
714 +                             The meter reading as transferred by the charging station (OCPP defau
715 +                         </s:documentation>
716 +                     </s:annotation>
717 +                 </s:element>
718 +                 <s:element minOccurs="1" maxOccurs="1" name="meterTime" type="s:LocalDateTimeType">
719 +                     <s:annotation>
720 +                         <s:documentation>
721 +                             The exact local time stamp of the time the meter reading was taken (
722 +                         </s:documentation>
723 +                     </s:annotation>
724 +                 </s:element>
725 +             </s:sequence>
726 +         </s:complexType>

```

***eCHS GUI***

(none)

### 13. Clarified token exchange

Change type: Enhancement

Impact: Specification, API, eCHS

Impact for partners: Easier to understand the implementation, more strict validity of tokens.

Description: Changed *validUntil* element in direct endpoints to *validDate* to specify a single day that the token is valid for (plus overlap).

For OCHP 1.3 partners the eCHS has to translate the *validDate* into a *validUntil* of 0:30 UTC of the following day.

For every partner and *namespaceURL*, there may only be one token/endpoint for each day (*validDate*). This means, up to a time cutoff (12:00 UTC of day "*validDate* - 1"), security tokens can be overwritten.

Additionally, the token used must be unique and shall be rejected by eCHS if the same token is being used by another endpoint for this *validDate*.

#### Specification

##### Identification token distribution

The partner's tokens for identification and authorisation are exchanged and distributed through the clearing house. Based on the set roaming connections the tokens are made available. Each token is valid for a period of one full calendar day, synchronus to the UTC time, with an additional short overlap for token exchange.

This mechanism is used to guarantee uninterrupted service in combination with a high security level and compatibility with the majority of systems. The synchronisation and token-exchange-cycle is as follows. On day *N* do:

1. At 00:30 UTC: Invalidate/delete all token combinations of day *N-1*.
2. Generate new own token for day *N+1*.
3. Before 11:55 UTC: Send/upload own token for day *N+1* to the clearing house.
4. After 12:05 UTC: Fetch/download partner's tokens for day *N+1* from the clearing house.
5. Generate token combinations for day *N+1* from own and partner's tokens. Here *A<sub>B2</sub>*.
6. At 23:50 UTC: Make token combinations for day *N+1* valid.

This means that each night from 23:50 UTC to 0:30 UTC, a set of two token combinations is valid. Each token that is valid for a given day *N* shall be considered valid from 23:50 UTC on day *N-1* until 00:30 UTC on day *N+1*, giving it a total validity of 24:40 hours. When communicating with another partners OCHPdirect system, the token combination for day *N* shall be used from 00:00 UTC (day *N*) until 00:00 UTC of day *N+1*.

## DirectEndpoint class

Contains a generic endpoint definition.

Field Name	Field Type	Card.	Description
url	string(255)	1	The endpoint address.
namespaceUrl	string(255)	1	The WSDL namespace definition (i.e. version definition).
accessToken	string(255)	1	The secret token to access this endpoint.
validDate	string(10)	1	The day on which this endpoint/token combination is valid.

**Note:** Any token for day N has to be treated as valid from day N-1 23:50 UTC to day N+1 0:30 UTC. **Note:** Any partner may operate more than one OCHPdirect endpoint to enable compatibility with a larger number of partners.

### WSDL (API)

156	-	<s:element name="validUntil" type="tns:DateTimeType" minOccurs="1" maxOccurs="1">
157	+	<s:element name="validDate" minOccurs="1" maxOccurs="1">
158		<s:annotation>
159	-	<s:documentation>
160	+	The date <b>till</b> when this endpoint/token combination is valid.
161	+	The date <b>on which</b> this endpoint/token combination is valid (plus over
162		</s:documentation>
163	+	</s:annotation>
164	+	<s:simpleType>
165	+	<s:restriction base="s:string">
166	+	<s:length value="10"/></s:length>
167	+	<s:whiteSpace value="collapse"/></s:whiteSpace>
168	+	<s:pattern value="\d\d\d\d-(\d\d)-(\d\d)"/></s:pattern>
169	+	</s:restriction>
170	+	</s:simpleType>
171		</s:element>

### eCHS GUI

(none)

## 14. Enabled reservation through *SelectEvse*

Change type: Enhancement

Impact: Specification, API, eCHS

Impact for partners: - (Reservation upon *SelectEvse* is already in place)

Description: *SelectEvse* is extended with a *reserveUntil* field, in which the EMP may request a time until which a reservation should be placed upon the selected EVSE.

Usage: it is recommended not to exceed a reservation of 30 minutes or to consider the *maxReservation* field from the *ChargePointInfo*. If a CPO does not wish to grant the reservation for the requested duration, they can respond with a different TTL for the session and it is up to the EMP to either accept this TTL (keep the session open) or to close the session (*ReleaseEvse*).

In case the reserving Contract-ID is associated with multiple physical RFID tokens *and* the charging process shall be started with one of those physical tokens, an EmtId may be part of the reservation request as well. *This is to be used only in the case specified here, not at any other time and is a workaround for an OCPP limitation (ReserveNow has to contain a physical token).*

### Specification

#### Select a charge point of an operator

Before a charging process can be started, the provider needs to select an EVSE in an operator's backend. This establishes the session and generates the OCHP-direct session ID. The operator must reserve the selected charging station for the communicated Contract-ID.

The provider may request reservation until a certain time and for a specific RFID token. It is up to the operator to decide whether or not to accept the provider's request for a reservation (duration). If the reservation request cannot be fulfilled the operator should return the maximum TTL for the reservation they would accept, but establish the session nonetheless. It is then up to the provider to decide whether to accept this offer from the operator (and otherwise release the EVSE, cancelling the reservation).

- MDM sends the *SelectEvse.req* PDU.
- CMS responds with a *SelectEvse.conf* PDU.

## SelectEvse.req

This contains the field definition of the SelectEvse.req sent by the MDM towards the CMS.

Field Name	Field Type	Card.	Description
evseld	Evseld	1	The charge point which is selected by the provider.
contractId	ContractId	1	Contract-ID for which the charge point is selected.
reserveUntil	DateTimeType	?	The desired TTL for the reservation created for the selected EVSE (in UTC).
reserveEmitd	Emitd	?	If defined, a reservation can be made for a specific physical token that is to be used to start the charging process.

**Note:** If no reserveUntil is defined in the request, it is up to the CPO to set a pre-defined TTL for the reservation and the session established (recommendation: 5 minutes). Once that TTL expires, the session and reservation should be invalidated.

**Note:** reserveUntil is intended to reserve an EVSE from now on and thus should not be requested too far into the future. It is recommended not to exceed a reservation request of 30 minutes. **Note:** A specific token may be defined for a reservation, which would enable access to the charging station with the customers RFID card as well. This field shall cover the use case when one Contract-ID is associated with more than one physical token and thus cannot be reliably reserved for the correct token (which has to be supplied in an OCPP ReserveNow request). In case a Contract-ID is associated with one physical token only, this token does not have to be further specified here and must be used for the reservation of the charge point by the CPO. If this is the case, the operator must still accept all ControlEvse requests coming in for that session ID.

## SelectEvse.conf

This contains the field definition of the SelectEvse.conf sent by the CMS as a response to SelectEvse.req.

Field Name	Field Type	Card.	Description
result	DirectResult	1	This contains the result of SelectEvse.req.
directId	DirectId	?	The session id for this direct charging process on success.
ttd	DateTimeType	?	On success the time until this selection is valid.

**Note:** Should the CPO not accept the extended duration of reservation for the selected EVSE, they should still create the session and return the maximum TTL they are willing to grant to the provider. It is then up to provider or their customer to accept this proposed TTL / reservation or to reject it (in which case the session should be closed by calling *ReleaseEvse.req*). If no reserveUntil is defined in the request, it is up to the CPO to set a session timeout (recommended: 5 minutes).

## WSDL (API)

337	345	<s:element name="SelectEvseRequest">
338	346	<s:complexType>
339	347	<s:sequence>
340	348	<s:element name="evseId" type="tns:EvseId" minOccurs="1" maxOccurs="1">
341	349	<s:annotation>
342	350	<s:documentation>
343	351	The charge point which is selected by the provider.
344	352	</s:documentation>
345	353	</s:annotation>
346	354	</s:element>
347	355	<s:element name="contractId" type="tns:ContractId" minOccurs="1" maxOccurs="1">
348	356	<s:annotation>
349	357	<s:documentation>
350	358	Contract-ID for which the charge point is selected.
351	359	</s:documentation>
352	360	</s:annotation>
353	361	</s:element>
362	+	<s:element name="reserveUntil" type="tns:DateTimeType" minOccurs="0" maxOccurs="1">
363	+	<s:annotation>
364	+	<s:documentation>
365	+	The time until which a reservation of the charge point is re
366	+	</s:documentation>
367	+	</s:annotation>
368	+	</s:element>
369	+	<s:element name="reserveEmitId" type="tns:EmitId" minOccurs="0" maxOccurs="1">
370	+	<s:annotation>
371	+	<s:documentation>
372	+	Physical token for which the EVSE shall be reserved. Only to
373	+	Only to be used in case the Contract-ID specified has multip
374	+	</s:documentation>
375	+	</s:annotation>
376	+	</s:element>
377		</s:sequence>
378		</s:complexType>
379		</s:element>

## eCHS GUI

(none)

## 15. Enhanced OCHPdirect with smart charging capabilities

Change type: Enhancement

Impact: Specification, API, eCHS

Impact for partners: -

Description: Added the following elements to the *ControlEvse* method:  
*maxCurrent, onePhase, minEnergy, departure*

Added *stateOfCharge* and the abovementioned elements to the *InformProvider* method.

### Specification

#### ControlEvse.req

This contains the field definition of the ControlEvse.req sent by the MDM towards the CMS.

Field Name	Field Type	Card.	Description
directId	DirectId	1	The session id referencing the direct charging process to be controlled.
operation	DirectOperation	1	The operation to be performed for the selected charge point.
maxPower	float	?	Maximum authorised power in kilowatts. Example: "3.7", "8", "15"
maxCurrent	float	?	Maximum authorised current in ampere. Example: "16", "25", "32"
onePhase	boolean	?	Marks an AC-charging session to be limited to one-phase charging.
maxEnergy	float	?	Maximum authorised energy in kilowatthours. Example: "5.5", "20", "85"
minEnergy	float	?	Minimum required energy in kilowatthours. Example: "5.5", "20", "85"
departure	DateTimeType	?	Scheduled time of departure. Only to be used for smart charging purposes.

**Note:** maxPower and maxCurrent should be used exclusively, otherwise it may be up to the CPO to choose the lower limit. The same goes for the one-phase limit, which may override maxPower limits only possible through three-phase charging.

### WSDL (API)

#### eCHS GUI

(none)

## 16. Specification document

Change type:	Bug Fix, Enhancement
Impact:	<input checked="" type="checkbox"/> Specification, <input type="checkbox"/> API, <input type="checkbox"/> eCHS
Backward-Compatibility:	n/a
Forward-Compatibility:	n/a
Impact for partners:	none
Description:	Various spelling errors in the specification document get corrected. Added more information to multiple definitions.

## Changes to the eCHS

### 17. Contact page in the Frontend

Instead of having just a contact e-mail address for each partner on the eCHS, a full contact page will be added, containing information for both a business contact and a technical contact (publicly accessible) as well as a list of possibly multiple e-mail addresses that should be used by the eCHS for notifications (only accessible by the partner themselves).

#### Business Contact

Name: A. B.

Company: B

Phone: +123546879

E-Mail: a@b.com

#### Technical Contact

Name: B. A.

Company: B

Phone: +987654321

E-mail: b@a.com

#### Notification E-Mails:

- 1@a.com

- 2@a.com

- ...

This info can be assigned to multiple Partners, just as we have network and group selection, we can select contacts for partners (i.e. one contact page can be assigned to multiple partners). While introducing this, the current contact e-mail address will be used as a notification e-mail as well as the initial e-mail for the business contact.