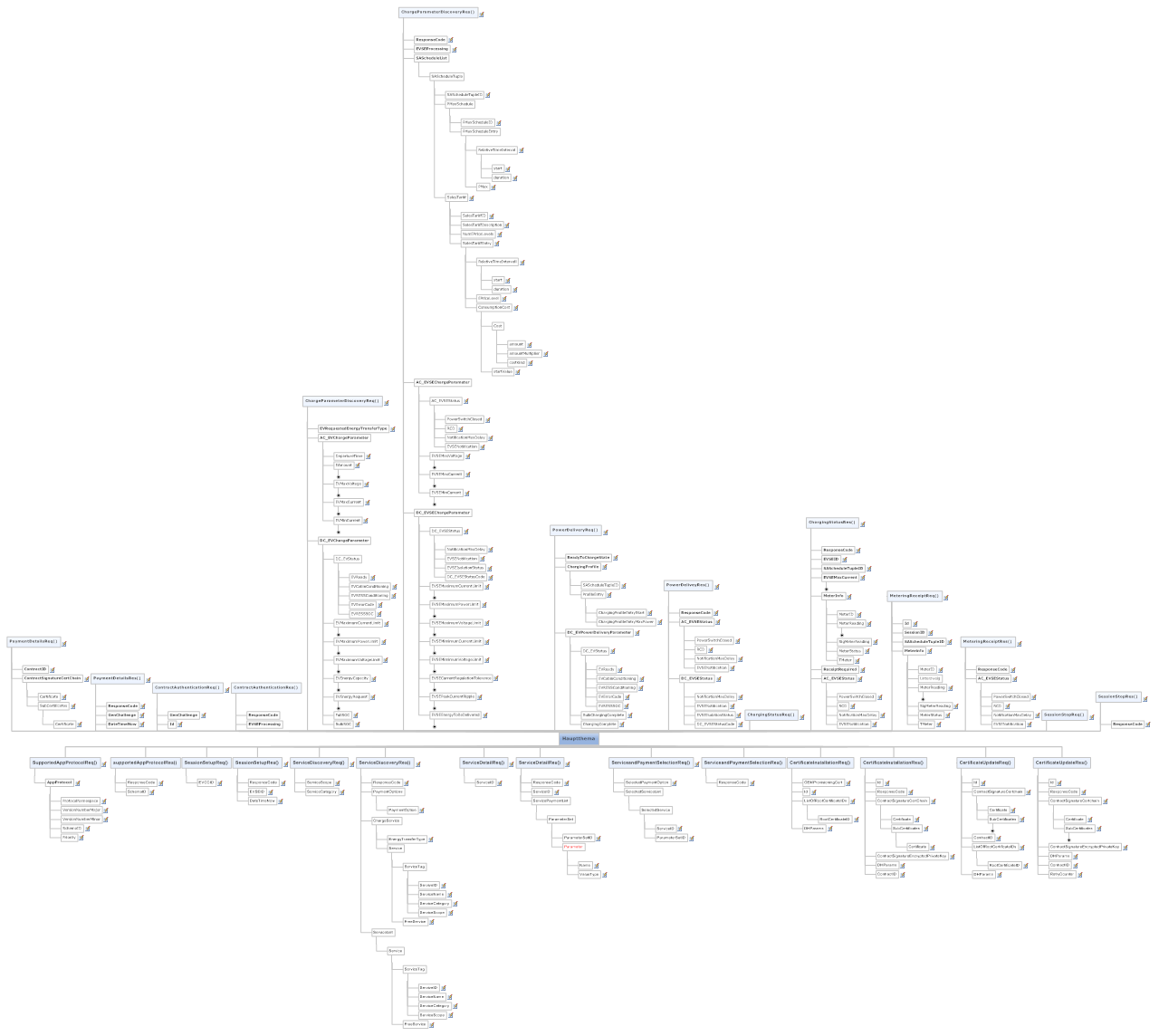
Beschreibung der Variablen



1 PaymentDetailsReq() 12

1.1 ContractID 12

1.2 ContractSignatureCertChain 12

1.2.1 Certificate 12

1.2.2 SubCertificates 12

Certificate 12

2 PaymentDetailsRes() 12

2.1 ResponseCode 12

2.2 GenChallenge 13

2.3 DateTimeNow 13

3 ContractAuthenticationReq() 13

3.1 GenChallenge 13

3.2 Id 13

4 ContractAuthenticationRes() 13

4.1 ResponseCode 13

4.2 EVSEProcessing 14

5 ChargeParameterDiscoveryReq() 14

5.1 EVRequestedEnergyTransferType 14

5.2 AC\_EVChargeParameter 14

5.2.1 DepartureTime 14

5.2.2 EAmount 14

PhysicalValueType 15

Multiplier 15

Unit 15

Value 15

5.2.3 EVMaxVoltage 15

PhysicalValueType 15

Multiplier 15

Unit 15

Value 15

5.2.4 EVMaxCurrent 15

PhysicalValueType 15

Multiplier 15

Unit 15

Value 15

5.2.5 EVMinCurrent 15

PhysicalValueType 16

Multiplier 16

Unit 16

Value 16

5.3 DC\_EVChargeParameter 16

5.3.1 DC\_EVStatus 16

EVReady 16

EVCabinConditioning 16

EVRESSConditioning 16

EVErrorCode 16

EVRESSSOC 16

5.3.2 EVMaximumCurrentLimit 16

PhysicalValueType 17

Multiplier 17

Unit 17

Value 17

5.3.3 EVMaximumPowerLimit 17

PhysicalValueType 17

Multiplier 17

Unit 17

Value 17

5.3.4 EVMaximumVoltageLimit 17

PhysicalValueType 17

Multiplier 17

Unit 17

Value 17

5.3.5 EVEnergyCapacity 17

PhysicalValueType 17

Multiplier 17

Unit 17

Value 17

5.3.6 EVEnergyRequest 17

PhysicalValueType 18

Multiplier 18

Unit 18

Value 18

5.3.7 FullSOC 18

5.3.8 BulkSOC 18

6 ChargeParameterDiscoveryRes() 18

6.1 ResponseCode 18

6.2 EVSEProcessing 18

6.3 SAScheduleList 19

6.3.1 SAScheduleTuple 19

SAScheduleTupleID 19

PMaxSchedule 19

PMaxScheduleID 19

PMaxScheduleEntry 19

RelativeTimeInterval 19

start 19

duration 19

PMax 19

SalesTariff 19

SalesTariffID 20

SalesTariffDescription 20

NumEPriceLevels 20

SalesTariffEntry 20

RelativeTimeIntervall 20

start 20

duration 20

EPriceLevel 20

ConsumptionCost 20

Cost 21

amount 21

amountMultiplier 21

costKind 21

startValue 21

6.4 AC\_EVSEChargeParameter 21

6.4.1 AC\_EVSEStatus 21

PowerSwitchClosed 21

RCD 21

NotificationMaxDelay 22

EVSENotification 22

6.4.2 EVSEMaxVoltage 22

PhysicalValueType 22

Multiplier 22

Unit 22

Value 22

6.4.3 EVSEMaxCurrent 22

PhysicalValueType 22

Multiplier 22

Unit 22

Value 22

6.4.4 EVSEMinCurrent 22

PhysicalValueType 23

Multiplier 23

Unit 23

Value 23

6.5 DC\_EVSEChargeParameter 23

6.5.1 DC\_EVSEStatus 23

NotificationMaxDelay 23

EVSENotification 23

EVSEIsolationStatus 23

DC\_EVSEStatusCode 23

6.5.2 EVSEMaximumCurrentLimit 23

PhysicalValueType 24

Multiplier 24

Unit 24

Value 24

6.5.3 EVSEMaximumPowerLimit 24

PhysicalValueType 24

Multiplier 24

Unit 24

Value 24

6.5.4 EVSEMaximumVoltageLimit 24

PhysicalValueType 24

Multiplier 24

Unit 24

Value 24

6.5.5 EVSEMimimumCurrentLimit 24

PhysicalValueType 24

Multiplier 24

Unit 24

Value 24

6.5.6 EVSEMimimumVoltageLimit 24

PhysicalValueType 25

Multiplier 25

Unit 25

Value 25

6.5.7 EVSECurrentRegulationTolerance 25

PhysicalValueType 25

Multiplier 25

Unit 25

Value 25

6.5.8 EVSEPeakCurrentRipple 25

PhysicalValueType 25

Multiplier 25

Unit 25

Value 25

6.5.9 EVSEEnergyToBeDelivered 25

PhysicalValueType 26

Multiplier 26

Unit 26

Value 26

7 PowerDeliveryReq() 26

7.1 ReadyToChargeState 26

7.2 ChargingProfile 26

7.2.1 SAScheduleTupleID 26

7.2.2 ProfileEntry 26

ChargingProfileEntryStart 26

ChargingProfileEntryMaxPower 26

7.3 DC\_EVPowerDeliveryParameter 27

7.3.1 DC\_EVStatus 27

EVReady 27

EVCabinConditioning 27

EVRESSConditioning 27

EVErrorCode 27

EVRESSSOC 27

7.3.2 BulkChargingComplete 27

7.3.3 ChargingComplete 28

8 PowerDeliveyRes() 28

8.1 ResponseCode 28

8.2 AC\_EVSEStatus 28

8.2.1 PowerSwitchClosed 28

8.2.2 RCD 28

8.2.3 NotificationMaxDelay 28

8.2.4 EVSENotification 28

8.3 DC\_EVSEStatus 29

8.3.1 NotificationMaxDelay 29

8.3.2 EVSENotification 29

8.3.3 EVSEIsolationStatus 29

8.3.4 DC\_EVSEStatusCode 29

9 ChargingStatusReq() 29

10 ChargingStatusRes() 30

10.1 ResponseCode 30

10.2 EVSEID 30

10.3 SAScheduleTupleID 30

10.4 EVSEMaxCurrent 30

10.4.1 PhysicalValueType 30

Multiplier 30

Unit 30

Value 30

10.5 MeterInfo 30

10.5.1 MeterID 31

10.5.2 MeterReading 31

PhysicalValueType 31

Multiplier 31

Unit 31

Value 31

10.5.3 SigMeterReading 31

10.5.4 MeterStatus 31

10.5.5 TMeter 31

10.6 ReceiptRequired 31

10.7 AC\_EVSEStatus 32

10.7.1 PowerSwitchClosed 32

10.7.2 RCD 32

10.7.3 NotificationMaxDelay 32

10.7.4 EVSENotification 32

11 MeteringReceiptReq() 32

11.1 Id 33

11.2 SessionID 33

11.3 SAScheduleTupleID 33

11.4 Meterinfo 33

11.4.1 MeterID 33

11.4.2 MeterReading 33

PhysicalValueType 34

Multiplier 34

Unit 34

Value 34

11.4.3 SigMeterReading 34

11.4.4 MeterStatus 34

11.4.5 TMeter 34

12 MeteringReceiptRes() 34

12.1 ResponseCode 34

12.2 AC\_EVSEStatus 34

12.2.1 PowerSwitchClosed 34

12.2.2 RCD 35

12.2.3 NotificationMaxDelay 35

12.2.4 EVSENotification 35

13 SessionStopReq() 35

14 SessionStopRes() 35

14.1 ResponseCode 35

15 SupportedAppProtocolReq() 36

15.1 AppProtocol 36

15.1.1 ProtocolNamespace 36

15.1.2 VersionNumberMajor 36

15.1.3 VersionNumberMinor 36

15.1.4 SchemaID 36

15.1.5 Priority 37

16 supportedAppProtocolRes() 37

16.1 ResponseCode 37

16.2 SchemaID 37

17 SessionSetupReq() 37

17.1 EVCCID 37

18 SessionSetupRes() 37

18.1 ResponceCode 37

18.2 EVSEID 38

18.3 DateTimeNow 38

19 ServiceDiscoveryReq() 38

19.1 ServiceScope 38

19.2 ServiceCategory 38

20 ServiceDiscoveryRes() 39

20.1 ResponseCode 39

20.2 PaymentOptions 39

20.2.1 PaymentOption 39

20.3 ChargeService 39

20.3.1 EnergyTransferType 39

20.3.2 Service 39

ServiceTag 39

ServiveID 39

ServiceName 40

ServiceCetegory 40

ServiceScope 40

FreeService 40

20.4 ServiceList 40

20.4.1 Service 40

ServiceTag 40

ServiveID 40

ServiceName 40

ServiceCetegory 40

ServiceScope 41

FreeService 41

21 ServiceDetailReq() 41

21.1 ServiceID 41

22 ServiceDetailRes() 41

22.1 ResponseCode 41

22.2 ServiceID 41

22.3 ServicePaymentList 42

22.3.1 ParameterSet 42

ParameterSetID 42

Parameter 42

Name 42

ValueType 42

23 ServiceandPaymentSelectionReq() 42

23.1 SelectedPaymentOption 42

23.2 SelectedServiceList 42

23.2.1 SelectedService 42

ServiceID 42

ParameterSetID 43

24 ServiceandPaymentSelectionRes() 43

24.1 ResponseCode 43

25 CertificateInstallationReq() 43

25.1 OEMProvisioningCert 43

25.2 Id 43

25.3 ListOfRootCertificateIDs 43

25.3.1 RootCertificateID 44

25.4 DHParams 44

26 CertificateInstallationRes() 44

26.1 Id 44

26.2 ResponseCode 44

26.3 ContractSignatureCertChain 44

26.3.1 Certificate 44

26.3.2 SubCertificates 44

Certificate 44

26.4 ContractSignatureEncryptedPrivateKey 45

26.5 DHParams 45

26.6 ContractID 45

27 CertificateUpdateReq() 45

27.1 Id 45

27.2 ContractSignatureCertchain 45

27.2.1 Certificate 45

27.2.2 SubCertificates 46

Certificate 46

27.3 ContractID 46

27.4 ListOfRootCertificateIDs 46

27.4.1 RootCertificateID 46

27.5 DHParams 46

28 CertificateUpdateRes() 46

28.1 Id 46

28.2 ResponseCode 47

28.3 ContractSignatureCertchain 47

28.3.1 Certificate 47

28.3.2 SubCertificates 47

Certificate 47

28.4 ContractSignatureEncryptedPrivateKey 47

28.5 DHParams 48

28.6 ContractID 48

28.7 RetryCounter 48

1. PaymentDetailsReq()

Mit dieser Nachricht bietet der EVCC die Zahlungsdetails an, falls die ausgewählte Zahlungsart "Contract" ist. Mit dem "signature certificate chain" und der "Contract id" fordert der EVCC den SECCD auf eine "challenge" zu senden.

* 1. ContractID

("string (max length 24)")

Dieses Element identifiziert den Lade Vertrag, Das Format ist in der DIN 91286 definiert.

* 1. ContractSignatureCertChain

Dieses Element beinhaltet das Zertifikat vom Vertrag und optional auch "SubCertificates".

* + 1. Certificate

("base64Binary (max length: 1200)")

Ein x.509v3 Zertifikat (das "client" Zertifikat)

* + 1. SubCertificates

Optional: Eine Lise an weiteren Zertifikaten zusätzlich zum Stamm-Zertifikat (enthält aber nicht das Stamm-Zertifikat).

Certificate

("base64Binary (max length: 1200)")

Ein x.509v3 Zertifikat (das "client" Zertifikat)

1. PaymentDetailsRes()

Mit der "Payment Details Response“-Nachricht informiert der SECC den EVCC ob die vorgesehene Zahlungsdetails akzeptiert wurden oder nicht. Der SECC sendet zudem eine Herausforderung in Form einer Zufallszahl, welche durch den EVCC gelöst werden muss.

* 1. ResponseCode

("enumeration")

Der Response Code gibt den Bestätigungszustand der vom SECC Empfangen Botschaft an.

* 1. GenChallenge

("string")

Die "Challenge" wird durch das SECC gesendet. Dieses Element enthält die erzeugte Zufallszahl. Jede Codierung um die Zufallszahl in einen String umzuwandeln muss mindestens 128 bit haben.

* 1. DateTimeNow

("long")

Zeitstempel der aktuellen Zeit im "Unix Time Stamp" format. Dieses Element wird vom EVCC genutzt um die Gültigkeit von Zertifikaten für „vertragsbasiertes Laden“ zu Prüfen, zudem dient es als externe Zeitreferenz. Basierend auf den Informationen könnte der EVCC eine „strategy“ implementieren, wenn das Zertifikat ein Update benötigt. Unter der Verwendung dieses Nachrichtenelementes wird verhindert, dass die Zeitbasis von dem EVCC und dem SECC synchronisiert werden muss.

1. ContractAuthenticationReq()

Nachdem der EVCC die "challange" von dem SECC empfangen hat, sendet der EVCC eine erneute Request-Nachricht welche die Signatur einer "challenge" und die "challenge" selber enthält.

* 1. GenChallenge

("string")

Optional: Die "Challenge" wird durch das SECC gesendet. Dieses Element enthält die erzeugte Zufallszahl. Jede Codierung um die Zufallszahl in einen String umzuwandeln muss mindestens 128 bit haben.

* 1. Id

("IDREF")

Dieses Element wird genutzt für die Referenzierung des Nachrichtentextes in der Nachrichtenkopfzeile wenn eine Unterschrift benötigt wird.

1. ContractAuthenticationRes()
   1. ResponseCode

("enumeration")

Der Response Code gibt den Bestätigungszustand der vom SECC Empfangen Botschaft an.

* 1. EVSEProcessing

("enumeration")

Dieser Parameter gibt an, dass das EVSE die Verarbeitung, welche durch die „ContractAuthenticationReq“-Nachricht ausgelöst wurde, beendet ist oder falls das EVSE noch in der Verarbeitung ist, die Response-Message gesendet wurde.

1. ChargeParameterDiscoveryReq()

Mit dem Absenden der „ChargeParameterDiscoveryReq()“-Nachricht übergibt der EVCC seine Ladeparameter an den SECC. Diese beinhalten Status Informationen über das EV und zusätzliche Ladeparameter wie vorraussichtliche Lademenge zum Vollladen der Batterie und den Zeitpunkt für das Ende des Ladevorganges.

* 1. EVRequestedEnergyTransferType

Ist die Ausgewählte Ladeart, welche der EVCC benötigt. Die genauen Arten können aus Tabelle 39 ausgelesen werden.

* 1. AC\_EVChargeParameter
     1. DepartureTime

(„unsignedInt“)

Dieses Element wird genutzt, um zu zeigen wann der Ladevorgang beendet sein soll. Dies passiert über einen Offset in Sekunden von der Zeit dieser Nachricht an gerechnet. Falls eine „0“ gesendet wird bedeutet dies, dass der Ladevorgang so schnell wie möglich abgeschlossen werden soll.

* + 1. EAmount

(„PhysicalValueType“)

Menge an Energie welche vom EV benötigt wird, bis die Abfahrtszeit erreicht ist oder die HV Batterie 100% erreicht hat. Es kann auch eine bestimmt Menge an Energie beinhalten, welche für andere Features benötigt wird.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVMaxVoltage

(„PhysicalValueType“)

Maximalspannung welche vom EV unterstütz wird. Diese Spannung wird zwischen einer Phase und Neutralleiter gemessen.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVMaxCurrent

(„PhysicalValueType“)

Maximaler Strom welcher pro Phase vom EV unterstützt wird.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVMinCurrent

(„PhysicalValueType“)

EVMinCurrent wird benötigt um dem SECC den minimalen Strom zu nennen, unter welchem es nicht möglich ist effizient zu laden. Dieser Wert wird vom SECC bei der Zielsetzung beachtet. Falls eine Ladestruktur nur einen geringeren Strom liefern kann, hängt es von der Implementierung ab, ob das Fahrzeug trotzdem geladen wird.

PhysicalValueType

Multiplier

Unit

Value

* 1. DC\_EVChargeParameter
     1. DC\_EVStatus

EVReady

(„boolean“)

EV ist bereit zum Laden, wenn Wert auf „true“

EVCabinConditioning

(„boolean“)

Optional: Das EV benutzt Energie von der DC Versorgung um den Innenraum zu heizen oder kühlen.

EVRESSConditioning

("boolean")

Optional: Fahrzeug RESS Anlage. Das Fahrzeug benutzt die Energie aus dem DC Ladegerät um die RESS auf Zieltemperatur zu bringen.

EVErrorCode

("enumeration")

Gibt den Status des EV an. Möglichkeiten sind aus Tabelle 94 zu entnehmen.

EVRESSSOC

("byte (range 0-100)")

Beinhaltet den Ladestatus (state of charge) von der Batterie (RESS).

* + 1. EVMaximumCurrentLimit

(„PhysicalValueType“)

Maximal unterstützter Strom des EV.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVMaximumPowerLimit

(„PhysicalValueType“)

Optional: Maximale unterstützte Leistung des EV.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVMaximumVoltageLimit

(„PhysicalValueType“)

Maximale unterstützte Spannung des EV.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVEnergyCapacity

(„PhysicalValueType“)

Optional: Maximale Leistungskapazität des EV.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVEnergyRequest

(„PhysicalValueType“)

Optional: Menge der Energie, welche das EV vom EVSE anfordert.

PhysicalValueType

Multiplier

Unit

Value

* + 1. FullSOC

(„percentValueType“)

Optional: Ladezustand (SOC), ab welchem das EV die Batterie als vollständig geladen betrachtet.

* + 1. BulkSOC

(„percentValueType“)

Optional: Ladezustand (SOC), ab welchem das EV den Schnellladeprozess als beendet betrachtet.

1. ChargeParameterDiscoveryRes()

Mit dieser Nachricht gibt das SECC an, welche Ladeparameter vom Ladenetzt aus anwendbar sind. Zudem beinhaltet diese Nachricht optional die Kosten über der Zeit, Kosten über der Nachfrage, Kosten über Verbrauch oder auch einen Kombination dieser Möglichkeiten. Aufgrund dieser Nachricht kann das EV seinen Ladeplan für die benötigte Energiemenge optimieren.

* 1. ResponseCode

("enumeration")

Der Response Code gibt den Bestätigungszustand der vom SECC Empfangen Botschaft an.

* 1. EVSEProcessing

("enumeration")

Dieser Parameter gibt an, dass das EVSE die Verarbeitung, welche durch die „ContractAuthenticationReq“-Nachricht ausgelöst wurde, beendet ist oder falls das EVSE noch in der Verarbeitung ist, die Response-Message gesendet wurde.

* 1. SAScheduleList
     1. SAScheduleTuple

SAScheduleTupleID

("short")

Eine Eindeutige Kennung innerhalb einer Ladesession für ein „SAScheduleTuple“.

PMaxSchedule

PMaxScheduleID

("short")

Eindeutige Kennung für ein Element vom Typ „PMaxSchedule“ über eine Lade Sitzung.

PMaxScheduleEntry

RelativeTimeInterval

Optional:

Definiert den Zeitabstand in dem „PMaxScheduleEntry“ gültig ist. Die Zeit ist dabei relativ angegeben.

start

("unsignedInt")

Start des Intervalls in Sekunden ab "jetzt".

duration

("unsignedInt")

Optional: Dauer des Intervalls in Sekunden

PMax

("short")

Definiert die maximale Leistungsmenge welche über das EVSE übertragen werden darf.

SalesTariff

Beschreibt alle relevanten Details für einen Tarif des „Secondary Actors“.

SalesTariffID

("short")

Die ID wird benötigt um einen Tarif genau zu identifizieren.

SalesTariffDescription

(" string (max. length: 32) ")

Optional: Kurzer normaler Text, welcher z.B dem Kunden auf dem Display angezeigt werden kann.

NumEPriceLevels

("unsignedByte")

Definiert die maximale Anzahl an Preis Levels über alle angebotenen Tarifen.

SalesTariffEntry

Beschriebt alle relevanten Details für ein Intervall der Verkaufs Tarife.

RelativeTimeIntervall

Definiert den Zeitabstand in dem „SalesTariffEntry“ gültig ist. Die Zeit ist dabei relativ angegeben.

start

("unsignedInt")

Start des Intervalls in Sekunden ab "jetzt".

duration

("unsignedInt")

Optional: Dauer des Intervalls in Sekunden

EPriceLevel

("unsignedByte")

Es definiert ein Preis Level für diesen "SalesTariffEntry". Kleine Zahlen stehen für einen günstigen Preis und große Zahlen stehen für einen teureren Tarif.

ConsumptionCost

Optional: Definiert zusätzliche Preisinformationen und/oder alternative Kosten.

Cost

amount

("unsignedInt")

Der aktuelle Preis pro Einheit (typisch kWh)

amountMultiplier

("uniMultiplierType")

Optional: Wert zwischen -3 und 3 welcher als Shift Operator für den Wert in "amount" dient.

costKind

("enumeration")

Die Art der Kosten welche sich auf "amount" beziehen. Siehe Tabelle 80.

startValue

("unsignedInt")

Der Kleinste Wert des Verbrauches, welcher den Startpunkt dieses Blockes definiert. Dieser gilt bis zum Startpunkt des nächsten Intervalls.

Das erste Element soll mit der "startValue" 0 beginnen.

* 1. AC\_EVSEChargeParameter
     1. AC\_EVSEStatus

Dieses Element wird vom SECC zum Anzeigen des SECC-Status verwendet.

PowerSwitchClosed

("boolean")

Identifiziert ob der Netzschalter geschlossen ist. Wenn der Schalter geschlossen wird, so wird der Wert auf "true" gesetzt. Dies wird nur zur Information genutzt.

RCD

("boolean")

Zeigt den aktuellen Status des RCD (Residual Current Device) an. Wenn der Wert "true" ist hat der RCD einen Fehler (error) bemerkt.

NotificationMaxDelay

("unsignedInt")

Es wird die Zeit angegeben die das EVCC benötigt, um auf die in der „EVSENotification“ erhaltene Aufgabe auszuführen. Liegt die Zeit nicht in der Zukunft, so wird das EVCC dies sofort erledigen.

EVSENotification

("enumeration")

Dieser Wert wird von dem SECC verwendet, um den EVCC zu beeinflussen. Die EVSENotification enthält eine Aktion, die das EVCC durchführen soll. Die Aktion soll bis zu der genannten Zeit in "NotificationMaxDelay" ausgeführt werden. Liegt die Zeit nicht in der Zukunft soll die Aktion sofort durchgeführt werden. Während des normalen Betriebs wird der Wert auf "none" gesetzt.

* + 1. EVSEMaxVoltage

Leiterspannung welche vom EVSE unterstützt wird. Diese Spannung wird zwischen einer Phase und dem Neutralleiter gemessen. Falls das EVSE „Netzphasen laden“ unterstützt, so kann das EV die Spannung zwischen den Phasen errechnen.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVSEMaxCurrent

Maximaler Netzstrom, welcher der EVSE bieten kann, angesichts der Leistung des EVSE einschließlich der Einschränkungen der Kabel.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVSEMinCurrent

Minimaler Netzstrom welcher das EVSE anbieten kann.

PhysicalValueType

Multiplier

Unit

Value

* 1. DC\_EVSEChargeParameter
     1. DC\_EVSEStatus

Dieses Element wird vom SECC zum Anzeigen des SECC Status verwendet.

NotificationMaxDelay

("unsignedInt")

Es wird die Zeit angegeben, die das EVCC benötigt um auf die in der „EVSENotification“ erhaltene Aufgabe auszuführen. Liegt die Zeit nicht in der Zukunft, so wird das EVCC dies sofort erledigen.

EVSENotification

("enumeration")

Dieser Wert wird von dem SECC verwendet, um den EVCC zu beeinflussen. Die EVSENotification enthält eine Aktion, die das EVCC durchführen soll. Die Aktion soll bis zu der genannten Zeit in "NotificationMaxDelay" ausgeführt werden. Liegt die Zeit nicht in der Zukunft soll die Aktion sofort durchgeführt werden. Während des normalen Betriebs wird der Wert auf "none" gesetzt.

EVSEIsolationStatus

("enumeration")

Optional: Zeigt den Zustand der Isolation an (Ergebnis der Isolationsüberwachnung)

DC\_EVSEStatusCode

("enumeration")

Zeigt den internen Status des EVSE an. Details siehe Tabelle 92.

* + 1. EVSEMaximumCurrentLimit

("PhysicalValueType")

Maximaler Strom, welcher das EVSE liefern kann.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVSEMaximumPowerLimit

("PhysicalValueType")

Optional: Maximale Leistung, welche das EVSE liefern kann.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVSEMaximumVoltageLimit

("PhysicalValueType")

Maximale Spannung welche das EVSE liefern kann.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVSEMimimumCurrentLimit

("PhysicalValueType")

Minimaler Strom welcher das EVSE mit der erwarteten Genauigkeit liefern kann.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVSEMimimumVoltageLimit

("PhysicalValueType")

Minimale Spannung welche das EVSE mit der erwarteten Genauigkeit liefern kann.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVSECurrentRegulationTolerance

("PhysicalValueType")

Optional: Absolute Größe der Toleranzen des EVSE.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVSEPeakCurrentRipple

("PhysicalValueType")

Spitze Spitze Wert der Stromwelligkeit des EVSE.

PhysicalValueType

Multiplier

Unit

Value

* + 1. EVSEEnergyToBeDelivered

("PhysicalValueType")

Optional: Energiemenge, welche vom EVSE geliefert wird.

PhysicalValueType

Multiplier

Unit

Value

1. PowerDeliveryReq()

Beim senden der "Power Delivery Request“-Nachricht möchte der EVCC das der SECC den Strom anschaltet und überträgt.

* 1. ReadyToChargeState

("boolean")

Dieses Element wird verwendet, dass das EVSE alle Bedingungen startet, dazu führen, dass die Energieübertragung so schnell wie möglich beginnt. Wenn „ReadyToCharge“ gleich „TRUE“ ist soll das EVSE den Energiefluss für einen sofortigen Start vorbereiten, wenn “ReadyToCharge“ gleich „ FALSE“ ist so soll das EVSE den Energiefluss stoppen.

* 1. ChargingProfile

Optionales Element: Erlaubt dem EV ein Ladeprofil für die aktuelle Ladung zu reservieren (z.B. maximale Menge Energie über der Zeit).

* + 1. SAScheduleTupleID

("short")

Eine Eindeutige Kennung innerhalb einer Ladesession für ein „SAScheduleTuple“.

* + 1. ProfileEntry

Element welches Gebührenschemas für die einzelnen Ladeprofile enthält.

ChargingProfileEntryStart

("unsignedInt")

Zeit ab dem "chargingProfileEntry" gültig ist. Die Zeit wird dabei als Offset in Sekunden angegeben ab "jetzt".

ChargingProfileEntryMaxPower

("short")

Maximale Leistung in Watt welche vom EV verbraucht wird innerhalb dem aktuellen Ladeprofil (beginnende ab "ChargingProfileEntryStart").

* 1. DC\_EVPowerDeliveryParameter

Optionales Element: Dieses Element wird vom EVCC zur Übertragung der Parameter für die Leistungsabgabe verwendet.

* + 1. DC\_EVStatus

Aktueller Status des EV

EVReady

(„boolean“)

EV ist bereit zum Laden wenn Wert auf „true“

EVCabinConditioning

(„boolean“)

Optional: Das EV benutzt Energie von der DC Versorgung um den Innenraum zu heizen oder kühlen.

EVRESSConditioning

("boolean")

Optional: Fahrzeug RESS Anlage. Das Fahrzeug benutzt die Energie aus dem DC Ladegerät um die RESS auf Zieltemperatur zu bringen.

EVErrorCode

("enumeration")

Gibt den Status des EV an. Möglichkeiten sind aus Tabelle 94 zu entnehmen.

EVRESSSOC

("byte (range 0-100)")

Beinhaltet den Ladestatus (state of charge) von der Batterie (RESS).

* + 1. BulkChargingComplete

("boolean")

Optional: Wenn dieser Wert auf "true" gesetzt wird, zeigt das EV an, dass SOC über 80% liegt.

* + 1. ChargingComplete

("boolean")

Wenn dieser Wert auf "true" gesetzt wird zeigt das EV an, dass die Batterie voll geladen ist (SOC 100%).

1. PowerDeliveyRes()

Nachdem die "Power Delivery Request“-Nachricht des EVCC empfangen wurde sendet das SECC die "Power Delivery Response“-Nachricht welche Informationen darüber gibt ob Energie zur Verfügung steht.

* 1. ResponseCode

("enumeration")

Der „Response Code“ gibt den Bestätigungszustand der vom SECC Empfangen Botschaft an.

* 1. AC\_EVSEStatus

Dieses Element wird vom SECC zum Anzeigen des SECC Status verwendet.

* + 1. PowerSwitchClosed

("boolean")

Identifiziert ob der Netzschalter geschlossen ist. Ist der Schalter geschlossen, so wird der Wert auf „true“ gesetzt. Dies wird nur zur Information genutzt.

* + 1. RCD

("boolean")

Zeigt den aktuellen Status des RCD (Residual Current Device) an. Wenn der Wert "true" ist hat der RCD einen Fehler (error) bemerkt.

* + 1. NotificationMaxDelay

("unsignedInt")

Es wird die Zeit angegeben, die das EVCC benötigt, um auf die in der „EVSENotification“ erhaltene Aufgabe auszuführen. Liegt die Zeit nicht in der Zukunft, so wird das EVCC dies sofort erledigen.

* + 1. EVSENotification

("enumeration")

Dieser Wert wird von dem SECC verwendet, um den EVCC zu beeinflussen. Die „EVSENotification“ enthält eine Aktion, die das EVCC durchführen soll. Die Aktion soll bis zu der genannten Zeit in "NotificationMaxDelay" ausgeführt werden. Liegt die Zeit nicht in der Zukunft soll die Aktion sofort durchgeführt werden. Während des normalen Betriebs wird der Wert auf "none" gesetzt.

* 1. DC\_EVSEStatus

Dieses Element wird vom SECC zum Anzeigen des SECC Status verwendet.

* + 1. NotificationMaxDelay

("unsignedInt")

Es wird die Zeit angegeben, die das EVCC benötigt um auf die in der „EVSENotification“ erhaltene Aufgabe auszuführen. Liegt die Zeit nicht in der Zukunft, so wird das EVCC dies sofort erledigen.

* + 1. EVSENotification

("enumeration")

Dieser Wert wird von dem SECC verwendet, um den EVCC zu beeinflussen. Die EVSENotification enthält eine Aktion, die das EVCC durchführen soll. Die Aktion soll bis zu der genannten Zeit in "NotificationMaxDelay" ausgeführt werden. Liegt die Zeit nicht in der Zukunft soll die Aktion sofort durchgeführt werden. Während des normalen Betriebs wird der Wert auf "none" gesetzt.

* + 1. EVSEIsolationStatus

("enumeration")

Optional: Zeigt den Zustand der Isolation an (Ergebnis der Isolationsüberwachnung)

* + 1. DC\_EVSEStatusCode

("enumeration")

Zeigt den internen Status des EVSE an. Details siehe Tabelle 92.

1. ChargingStatusReq()

Abhängig von den gewählten Nachrichten, welche in Abschnitt 8.6.2 definiert sind, sollen der EVCC und der SECC die obligatorischen Nachrichten implementieren, welche in der Tabelle 98 zu finden sind.

1. ChargingStatusRes()

Nachdem die "Charging Status Request“-Nachricht vom EVCC empfangen wurde, sendet der SECC die "Charging Status Response“-Nachricht. Im Falle einer erfolgreichen "Metering Status Request“-Nachricht beinhaltet die Response Nachricht die aktuellen Zählerstände aus dem "smart meter" welcher im EVSE installiert ist. Im Falle, dass die "Meter Status Request“-Nachricht gescheitert ist, wird kein Zählerstand benötigt. Der Fehler wird in der Response Nachricht angezeigt.

* 1. ResponseCode

("enumeration")

Der Response Code gibt den Bestätigungszustand der vom SECC Empfangen Botschaft an.

* 1. EVSEID

("hexBinary (max length: 32)")

Jede ID, die das EVSE eindeutig identifiziert. Das Format dieser Nachricht ist in der DIN91286 definiert. Falls der SECC keine ID vorlegen kann, wird der Wert zu Null gesetzt (00hex).

* 1. SAScheduleTupleID

("short")

Optional: Eine Eindeutige Kennung innerhalb einer Ladesession für ein SAScheduleTuple. Dieser wird vom SECC verwendet um den Tarif anzugeben für die gerade transportierte Energie.

* 1. EVSEMaxCurrent

Optional: Dieses Element wird vom SECC genutzt um den maximalen Leitungsstrom anzuzeigen welcher das EV ziehen kann. Dieses Element ist nicht Inhalt der Nachricht, falls "AC PnC" ausgewählt wurde.

* + 1. PhysicalValueType

Multiplier

Unit

Value

* 1. MeterInfo

Optional: Beinhaltet die "MeterInfo" Aufzeichnung mit den neuesten Zählerablesung und anderen "meter" relevanten Daten.

* + 1. MeterID

("string (max length: 32)")

ID des Zählers in dem EVSE

* + 1. MeterReading

Optionales Element:

Aktueller Zählerstand in Wattstunden von dem EVSE

PhysicalValueType

Multiplier

Unit

Value

* + 1. SigMeterReading

("base64Binary (max length:32)")

Optionales Element: Signatur des Zählerstandes.

Diese Signatur wird vom EVSE "meter" erzeugt. Es ist nicht in dem EVCC verifiziert. Es könnte von einem "SA" System benutzt werden für die Abrechnung falls lokale Bestimmungen es genehmigen.

* + 1. MeterStatus

("short")

Optionales Element: Aktueller Status des Zählers

* + 1. TMeter

("short")

Optionales Element: Zeitstempel der aktuellen SECC Zeit. Benutzt wird das „Unix Time Stamp Format“.

* 1. ReceiptRequired

("boolean")

Dieses Element wird vom SECC verwendet um den SECC dazu veranlassen, dass dieser eine "MeteringReceiptReq" Message sendet um den gesendeten "Meter info" Satz zu "Unterzeichnen". Wenn ReceiptRequired auf true gesetzt wird, so soll das EVCC die "MeteringReceiptReq“-Nachricht inklusive der Signatur. Ist der Wert false so muss das EVCC keine "MeteringReceiptReq“-Nachricht senden.

* 1. AC\_EVSEStatus

Aktueller Status des EVSE

* + 1. PowerSwitchClosed

("boolean")

Identifiziert ob der Netzschalter geschlossen ist. Ist der Schalter geschlossen, so wird der Wert auf "true" gesetzt. Dies wird nur zur Information genutzt.

* + 1. RCD

("boolean")

Zeigt den aktuellen Status des RCD (Residual Current Device) an. Wenn der Wert "true" ist hat der RCD einen Fehler (error) bemerkt.

* + 1. NotificationMaxDelay

("unsignedInt")

Es wird die Zeit angegeben die das EVCC benötigt um auf die in der „EVSENotification“ erhaltene Aufgabe auszuführen.

Liegt die Zeit nicht in der Zukunft, so wird das EVCC dies sofort erledigen.

* + 1. EVSENotification

("enumeration")

Dieser Wert wird von dem SECC verwendet, um den EVCC zu beeinflussen. Die EVSENotification enthält eine Aktion, die das EVCC durchführen soll. Die Aktion soll bis zu der genannten Zeit in "NotificationMaxDelay" ausgeführt werden. Liegt die Zeit nicht in der Zukunft, soll die Aktion sofort durchgeführt werden. Während des normalen Betriebs wird der Wert auf "none" gesetzt.

1. MeteringReceiptReq()

Durch das Senden der „Metering Receipt Request Message“ bestätigt der EVCC das die Elemente "MeterInfo", "SessionID" und "SAScheduleTupleID" vom SECC empfangen wurden. Diese Bestätigung wird durch eine Signatur um den Nachrichtentext bei der „MeteringReceiptReq Nachricht“ implementiert.

* 1. Id

("IDREF")

Dieses Element wird für die Referenzierung des Nachrichtentextes im Nachrichtenkopf verwendet, wenn eine Signatur angewandt werden muss.

* 1. SessionID

("hexBinary (max length: 8)")

Dieses Element wird vom EVCC und SECC genutzt, um eine V2G Kommunikationssitzung eindeutig zu identifizieren. Dieses Element ist identisch mit dem im Nachrichtenkopf. Es wird zusätzlich im "Body" platziert um eine Signatur darauf anzuwenden.

* 1. SAScheduleTupleID

("short")

Optional: Eindeutige Kennung innerhalb einer Ladesitzung für ein „SAScheduleTuple Element“. Dieses Element ist nur ein Echo des in der „ChargingStatusReq Nachricht“ empfangenen Wertes durch den SECC.

* 1. Meterinfo

Wenn der SECC in der "ChargingStatusReq“-Nachricht erkennt, dass eine "MeteringReceiptReq“-Nachricht benötigt wird, ist diese Nachricht ein Echo der "MeterInfo" Aufzeichnung. Diese wird in der "ChargingStatusReq“-Nachricht vom SECC empfangen wird.

* + 1. MeterID

("string (max length: 32)")

ID des Zählers in dem EVSE.

* + 1. MeterReading

Optionales Element:

Aktueller Zählerstand in Wattstunden von dem EVSE.

PhysicalValueType

Multiplier

Unit

Value

* + 1. SigMeterReading

("base64Binary (max length:32)")

Optionales Element: Signatur des Zählerstandes.

Diese Signatur wird vom EVSE "meter" erzeugt. Sie ist nicht in dem EVCC verifiziert. Es könnte von einem "SA" System benutzt werden, für die Abrechnung falls lokale Bestimmungen es genehmigen.

* + 1. MeterStatus

("short")

Optionales Element: Aktueller Status des Zählers.

* + 1. TMeter

("short")

Optionales Element: Zeitstempel der aktuellen SECC Zeit. Benutzt wird der Unix Time Stamp Format.

1. MeteringReceiptRes()

Nachdem die "Metering Receipt Request“-Nachricht empfangen wurde, sendet das SECC die "Metering Receipt Response“-Nachricht. In dieser Nachricht ist enthalten ob der SECC die Nachricht empfangen und akzeptiert hat.

* 1. ResponseCode

("enumeration")

Der Response Code gibt den Bestätigungszustand der vom SECC Empfangen Botschaft an.

* 1. AC\_EVSEStatus

Dieses Element wird vom SECC zum Anzeigen des SECC Status verwendet.

* + 1. PowerSwitchClosed

("boolean")

Identifiziert ob der Netzschalter geschlossen ist. Ist der Schalter geschlossen, so wird der Wert auf „true“ gesetzt. Dies wird nur zur Information genutzt.

* + 1. RCD

("boolean")

Zeigt den aktuellen Status des RCD (Residual Current Device) an. Wenn der Wert "true" ist hat der RCD einen Fehler (error) bemerkt.

* + 1. NotificationMaxDelay

("unsignedInt")

Es wird die Zeit angegeben, die das EVCC benötigt um auf die in der „EVSENotification“ erhaltene Aufgabe auszuführen. Liegt die Zeit nicht in der Zukunft, so wird das EVCC dies sofort erledigen.

* + 1. EVSENotification

("enumeration")

Dieser Wert wird von dem SECC verwendet, um den EVCC zu beeinflussen. Die „EVSENotification“ enthält eine Aktion, die das EVCC durchführen soll. Die Aktion soll bis zu der genannten Zeit in "NotificationMaxDelay" ausgeführt werden. Liegt die Zeit nicht in der Zukunft, so soll die Aktion sofort durchgeführt werden. Während des normalen Betriebs wird der Wert auf "none" gesetzt.

1. SessionStopReq()

Beim senden der "Session Stop Request“-Nachricht fordert das EVCC dazu auf den Ladevorgang zu beenden.

1. SessionStopRes()

Nachdem die "Session Stop Request"-Nachricht des EVCC empfangen wurde, sendet das SECC die "Session Stop Response"-Nachricht welche Information darüber gibt, wenn das Beenden des Ladevorganges erfolgreich war.

* 1. ResponseCode

("enumeration")

Der Response Code gibt den Bestätigungszustand der vom SECC Empfangen Botschaft an.

1. SupportedAppProtocolReq()

Der EVCC soll den "handshake" beginnen, in dem er eine "supportedAppProtocolReq"-Nachricht an den SECC sendet. Diese Nachricht enthält eine Liste mit Protokollen, welche Ladevorgänge der EVCC anbietet. Diese Liste beinhaltet sämtliche Informationen welche die Vorgänge explizit beschreiben. In der Liste die der EVCC übergibt sind die Protokolle anhand einer Prioritätsliste sortiert.

Der SECC soll nun mit "supportedAppProtocolRes" übermitteln welches Protokoll zum anschließenden Nachrichtenaustausch zwischen EVCC und SECC verwendet werden soll.

* 1. AppProtocol

("maximum number of entries: 20")

Dieses Nachrichten-Element wird vom EVCC dazu genutzt, die Liste der unterstützten Protokolle zu übertragen. Jedes Protokoll mit einer besonderen Version, die vom EVCC unterstützt wird, wird im AppProtocol mit einem Eintrag repräsentiert.

* + 1. ProtocolNamespace

("string (max length: 100)")

„ProtocolNamespace“ wird vom EVCC genutzt um den „Namespace“- URI eindeutig zu identifizieren welches vom EVCC unterstützt wird.

URI: Ein „Uniform Resource Identifier“ ist ein Identifikator und besteht aus einer Zeichenfolge, die zur Identifizierung einer abstrakten oder physischen Ressource dient. URIs werden zur Bezeichnung von Ressourcen (wie Webseiten, sonstigen Dateien, aber auch z. B. E-Mail-Empfängern) im Internet eingesetzt. Der aktuelle Stand ist als RFC 3986 publiziert.

* + 1. VersionNumberMajor

(„unsignedInt“)

„VersionNumberMajor“ wird vom EVCC genutzt um die Hauptversionsnummer des in „ProtocolNamespace“ genannten Protocol anzugeben.

* + 1. VersionNumberMinor

(„unsignedInt“)

„VersionNumerMinor“ wird vom EVCC genutzt um die Nebenversionsnummer des in „ProtocolNamespace“ genannten Protokolls anzugeben.

* + 1. SchemaID

(„unsignedByte“)

Dieses Element wird genutzt um die schemaID zu übergeben, welche zu dem vom EVCC übertragenem Protokoll passt.

* + 1. Priority

("unsignedByte: Range 1..20")

Dieses Element wird vom EVCC genutzt, um die Protokoll- Priorität eines Protokolls aufzuzeigen, sodass der SECC eine Auswahl anhand dieser Priorität ausführen kann.

1. supportedAppProtocolRes()
   1. ResponseCode

("enumeration")

Der „Response Code“ gibt den Bestätigungszustand der vom SECC Empfangenen Botschaft an.

* 1. SchemaID

(„unsignedByte“)

Mit der „SchemaID“ verweist der SECC auf ein Protokoll aus Liste von Protokollen aus der Response Nachricht.

1. SessionSetupReq()
   1. EVCCID

("hexBinary(max length:8)")

Spezifiziert die Identifikation vom EV in einem lesbaren Format. Beinhaltet die Mac-Adresse vom EVCC.

1. SessionSetupRes()

Durch das Benutzen der "SessionSetupRes" antwortet der SECC dem "SessionSetupReq".

* 1. ResponceCode

("enumeration")

Response Code gibt den Bestätigungsstatus aller V2G Nachrichten an, die vom SECC empfangen werden.

* 1. EVSEID

("hexBinary (max length:32)")

ID, welche die EVSE eindeutig identifiziert. Das Format dieser Nachricht ist in der DIN 91286. definiert. Wenn ein SECC solche ID-Daten nicht unterstützt, wird der Wert vom EVSEID auf Null (00hex) gesetzt.

* 1. DateTimeNow

("long")

Optional: Zeitstempel der aktuellen SECC Zeit, im Unix-Zeitstempel-Format. Dieses Nachricht-Element wird durch die EVCC verwendet um die Gültigkeit der Zertifikate für vorgelegten Lade-Vertrag zu prüfen. Des Weiteren dient sie als externe Zeitreferenz. Auf Basis dieser Informationen kann der EVCC eine Strategie implementieren, wenn Zertifikat-Updates erforderlich sind. Verwendung dieser Nachricht vermeidet, dass die Zeitbasis der EVCC und SECC synchronisiert werden müssen.

1. ServiceDiscoveryReq()

Während der „ServiceDiescoveryReq“-Nachricht fordert der EVCC den SECC auf, ihm alle Informationen über alle "Services" die der SECC anbietet, zu senden. Darüber hinaus kann der EVCC den Leistungsumfang für bestimmte Leistungen und Service-Typen begrenzen.

* 1. ServiceScope

("string (max length: 32)")

Optionales Element:

Definiert den Umfang der „Service Discovery“. Ein Bereich wird durch eine eindeutige URI bestimmt, die einem Dienstanbieter zugeordnet wird (z.B. Mobilitätsanbieter, Wertbezogener Service Provider etc.).Service Discovery kann mehrere Bereiche gleichzeitig in einer Anfrage verarbeiten. Durch Anlegen eines Bereichs auf die Service-Discovery kann die vom ServiceDiscoveryRes zurückgegebene Liste bis zu einem gewissen Umfang begrenzt werden, wodurch eine Vorfilterungermöglicht wird. Das SECC gibt immer alle unterstützten Dienste für alle Bereiche zurück, wenn keine spezielle ServiceScope in der Anfrage angegeben wurde.

* 1. ServiceCategory

("enumeration")

Optionales Element:

Definiert den Dienst Kategorie für die Service Discovery (z.B. EV Laden, Internetzugang etc.). Durch Anlegen einer Kategorie in der „Service Discovery“ kann die Ergebnisliste, die vom „ServiceDiscoveryRes“ zurückgegeben wird, auf eine bestimmte Kategorie von Dienstleistungen beschränkt werden. So wird eine Vorfilterung ermöglicht. Das SECC gibt immer alle Dienste für alle Kategorien zurück, wenn keine spezielle Kategorie in der Anfrage-Nachricht angegeben wurde (mit Hilfe der ServicCategory Nachrichten).

1. ServiceDiscoveryRes()

Nach dem Empfang der "ServiceDisoveryReq"- Nachricht vom EVCC sendet der SECC die "ServiceDiscoveryRes"- Nachricht. Wenn ein passender Service gefunden wurde, werden alle verfügbaren Service die der SECC für dieses Kriterium anbietet, aufgelistet. Wenn "Service Discovery" fehlschlägt, so bleibt die Liste leer und der Response Code zeigt mögliche Gründe an.

* 1. ResponseCode

("enumeration")

Der „Response Code“ gibt den Bestätigungszustand der vom SECC Empfangenen Botschaft an.

* 1. PaymentOptions
     1. PaymentOption

("enumeration")

Dieser Typ umfasst die Liste der Zahlungsoptionen. Ein SECC zeigt dem EVCC auf, welche Methoden zur Bezahlung ausgewählt werden können. Der EVCC kann nur eine der angegebenen Zahlungsmethoden auswählen. Hiermit werden alle Services bezahlt, die vom EVCC verwendet werden.

* 1. ChargeService
     1. EnergyTransferType

("enumeration")

Verfügbare Lade-Typen oder Methoden, unterstützt vom EVSE.

* + 1. Service

ServiceTag

ServiveID

("unsignedShort")

Eindeutige Bezeichnung des Service.

ServiceName

("string (max length: 32)")

Lesbarer Sercice-Name.

ServiceCetegory

("enumeration")

Kategorie eines Service, entspricht den definierten Leistungen, abgeleitet von: Basis-Service.

ServiceScope

("string (max length: 32)")

Zusätzliche Information zur Anwendung von diesem Service.

FreeService

("boolean")

Dieses Element wird vom SECC dazu genutzt, um festzustellen ob der Service vom EVCC kostenfrei genutzt werden kann. Wenn FreeService gleich „true“ ist, dann kann der EV den angebotenen Service ohne Bezahlung nutzen. Wenn „FreeService“ gleich „false“ ist, so wird dem EV eine Rechnung ausgestellt, diese wird dann mit der zuvor ausgewählten Zahlungsmethode beglichen.

* 1. ServiceList
     1. Service

ServiceTag

ServiveID

("unsignedShort")

Eindeutige Bezeichnung des Service.

ServiceName

("string (max length: 32)")

Lesbarer Sercice-Name

ServiceCetegory

("enumeration")

Kategorie eines Service, entspricht den definierten Leistungen, abgeleitet von: Basis-Service.

ServiceScope

("string (max length: 32)")

Zusätzliche Information zur Anwendung von diesem Service.

FreeService

("boolean")

Dieses Element wird vom SECC dazu genutzt, um festzustellen ob der Service vom EVCC kostenfrei genutzt werden kann. Wenn FreeService gleich „true“ ist, dann kann der EV den angebotenen Service ohne Bezahlung nutzen. Wenn „FreeService“ gleich „false“ ist, so wird dem EV eine Rechnung ausgestellt, diese wird dann mit der zuvor ausgewählten Zahlungsmethode beglichen.

1. ServiceDetailReq()

Mit dem Absenden der ServiceDetailReq Nachricht fordert der EVCC den SECC auf, bestimmte zusätzliche Informationen über die angebotenen Dienstleistungen vom EVSE zu senden.

* 1. ServiceID

("unsignedShort")

Dieses Element bezeichnet einen Service welcher vom SECC in der "ServiceDiscoveryRes"- Nachricht angeboten wird.

1. ServiceDetailRes()

Nach Erhalt der "ServiceDetailReq“-Nachricht vom EVCC sendet der SECC die "ServiceDetailRes"- Nachricht und stellt Informationen zu den Services bereit.

* 1. ResponseCode

("enumeration")

Der „Response Code“ gibt den Bestätigungszustand der vom SECC Empfangenen Botschaft an.

* 1. ServiceID

("unsignedShort")

Eindeutige Bezeichnung des Service.

* 1. ServicePaymentList
     1. ParameterSet

ParameterSetID

("short")

Dieses Element wird verwendet, um eine bestimmte Parametersatz-Einstellung für eine bestimmte Auswahl von Service-IDs auszuwählen, wenn die "ServicePaymentSelectionReq"- Nachricht verwendet wird.

Parameter

Name

("String")

Dieses Element wird verwendet um den Namen vom Parameter anzuzeigen.

ValueType

("enumeration")

Dieses Element umfasst den Wert des Parameters.

1. ServiceandPaymentSelectionReq()

Dieser Antrag Nachricht transportiert die Informationen zu den ausgewählten Services und zeigt wie die ausgewählten Dienste bezahlt werden.

* 1. SelectedPaymentOption

("enumeration")

Dieses Element zeigt an, welche Bezahlungsmethode für alle ausgewählten Services aus der "ServiceList" verwendet wird.

* 1. SelectedServiceList
     1. SelectedService

ServiceID

("unsignedShort")

Eindeutige Bezeichnung des Service.

ParameterSetID

("short")

Dieses Element wird verwendet, um eine bestimmte Parametersatz-Einstellung für eine bestimmte Auswahl von Service-IDs auszuwählen, wenn die "ServicePaymentSelectionReq"- Nachricht verwendet wird.

1. ServiceandPaymentSelectionRes()
   1. ResponseCode

("enumeration")

Der „Response Code“ gibt den Bestätigungszustand der vom SECC Empfangenen Botschaft an.

1. CertificateInstallationReq()
   1. OEMProvisioningCert

base64Binary (max length: 1200)

Ein spezielles EV-Zertifikat, das früher in der Regel auf dem EVCC von einem OEM installiert wurde. Die ID von diesem Zertifikat zusammen mit den in der SA (Vertragspartner) gespeicherten Informationen werden verwendet, um die aktuell gültigen Vertrag mit dem EV zu identifizieren.

Die Zertifikat-ID wird durch den Kunden an die SA übergeben, hierfür wird ein anderer Kommunikationskanal genutzt. Das Zertifikat selbst (d.h. seinen öffentlichen Schlüssel) wird verwendet, um Datenelemente in der Zertifikatsinstallation Antwort später zu verschlüsseln. Bitte beachten Sie auch den Anhang E und Anhang J.

* 1. Id

("IDREF")

Dieses Element wird für die Referenzierung des Nachrichtentextes im Nachrichtenkopf verwendet wird, wenn eine Unterschrift geleistet werden muss.

* 1. ListOfRootCertificateIDs

("ListOfRootCertificateIDsType")

Diese Liste beeinhaltet die Zertifikats- ID´s von allen Stamm Zertifikaten, die aktuell auf dem EVCC installiert sind.

* + 1. RootCertificateID

("rootCertificateIDType, string (max length: 40)")

Eindeutige Beschreibung vom V2G Stamm Zertifikat, welches auf dem EVCC installiert ist.

* 1. DHParams

("base64Binary (max length: 256)")

Diffie Hellman-Parameter aus der "Secundary Actor" zur Erzeugung des Sitzungsschlüssels an der EVCC, um die Vertragsunterzeichnung "Private Key" zu verschlüsseln.

1. CertificateInstallationRes()
   1. Id

("IDREF")

Dieses Element wird für die Referenzierung des Nachrichtentextes im Nachrichtenkopf verwendet wird, wenn eine Unterschrift geleistet werden muss.

* 1. ResponseCode

("enumeration")

Der „Response Code“ gibt den Bestätigungszustand der vom SECC Empfangenen Botschaft an.

* 1. ContractSignatureCertChain

Die neue Zertifikatskette für die Unterschrift, die im EVCC installiert werden muss.

* + 1. Certificate

base64Binary (max length: 1200)

Ein x.509v3 Zertifikat (Auftraggeber-Zertifikat)

* + 1. SubCertificates

Optional:

Die Kette mit Sub-Zertifikaten zum Ursprung. Zertifikat(beinhaltet nicht das Stamm- Zertifikat)

Certificate

base64Binary (max length: 1200)

Ein x.509v3 Zertifikat (Auftraggeber-Zertifikat)

* 1. ContractSignatureEncryptedPrivateKey

("base64Binary (max length: 128)")

Der private Schlüssel, welcher zum neuen Zertifikat gehört, für Unterzeichnung Zwecke. Es muss ebenfalls in der EVCC installiert werden.

Dies sind geheime Daten. Deshalb müssen alle Vorgänge mit dem alten Vertrag- Zertifikat der EVCC (Bezogen auf „ContractSignatureCertChain“ enthalten in Zerifikat Installations Anfrage) und unter Verwendung der berechneten „DH Secret“ für Verschlüsselung verschlüsselt werden, wie in Anhang I beschrieben wird.

* 1. DHParams

("base64Binary (max length: 256)")

Diffie Hellman-Parameter aus der SA zur Erzeugung des Sitzungsschlüssels an der EVCC, um die Vertragsunterzeichnung "Private Key" zu verschlüsseln.

* 1. ContractID

("string (max length: 24)")

Element indentifiziert den Lade-Auftrag

1. CertificateUpdateReq()
   1. Id

("IDREF")

Dieses Element wird für die Referenzierung des Nachrichtentextes im Nachrichtenkopf verwendet wird, wenn eine Unterschrift geleistet werden muss.

* 1. ContractSignatureCertchain

Enthält das derzeit verfügbare Signaturzertifikat einschließlich der Zertifikatskette im EVCC. Der SECC benutzt dieses Zertifikats (Kette), um die Signatur der Nachricht in der Kopfzeile der Nachricht zu überprüfen. Die komplette Kette wird übertragen um eine eigenständige Prüfung der Signatur zu erlauben.

* + 1. Certificate

base64Binary (max length: 1200)

Ein x.509v3 Zertifikat (Auftraggeber-Zertifikat)

* + 1. SubCertificates

Optional:

Complexer Typ: "SubCertificatesType"

Die Kette mit Sub-Zertifikaten zum Ursprung. Zertifikat(beinhaltet nicht das Stamm- Zertifikat)

Certificate

base64Binary (max length: 1200)

Ein x.509v3 Zertifikat (Auftraggeber-Zertifikat) .

* 1. ContractID

("string (max length: 24)")

Element indentifiziert den Lade-Auftrag.

* 1. ListOfRootCertificateIDs

Diese Liste beinhaltet alle Zertifikat- IDs von allen Stamm-Zertifikaten, die aktuell auf dem EVCC installiert sind.

* + 1. RootCertificateID

("string (max. Länge: 40) ")

Diese Nachricht identifiziert das V2G Stamm-Zertifikat, welches auf dem EVCC installiert ist.

* 1. DHParams

("base64Binary (max length: 256)")

Diffie Hellman-Parameter aus der SA zur Erzeugung des Sitzungsschlüssels an der EVCC, um die Vertragsunterzeichnung Private Key zu verschlüsseln.

1. CertificateUpdateRes()
   1. Id

("IDREF")

Dieses Element wird für die Referenzierung des Nachrichtentextes im Nachrichtenkopf verwendet wird, wenn eine Unterschrift geleistet werden muss.

* 1. ResponseCode

("enumeration")

Der „Response Code“ gibt den Bestätigungszustand der vom SECC Empfangenen Botschaft an.

* 1. ContractSignatureCertchain

Enthält das derzeit verfügbare Signaturzertifikat einschließlich der Zertifikatskette im EVCC. Der SECC benutzt dieses Zertifikats (Kette), um die Signatur der Nachricht in der Kopfzeile der Nachricht zu überprüfen.

Die komplette Kette wird übertragen um eine eigenständige Prüfung der Signatur zu erlauben.

* + 1. Certificate

Ein x.509v3 Zertifikat (Auftraggeber-Zertifikat)

simpleType: certificateType

base64Binary (max. Länge: 1200)

* + 1. SubCertificates

Optional:

Die Kette mit Sub-Zertifikaten zum Ursprung. Zertifikat(beinhaltet nicht das Stamm- Zertifikat)

Certificate

base64Binary (max length: 1200)

Ein x.509v3 Zertifikat (Auftraggeber-Zertifikat)

* 1. ContractSignatureEncryptedPrivateKey

("base64Binary (max length: 128)")

Der private Schlüssel, welcher zum neuen Zertifikat gehört, für Unterzeichnung Zwecke. Es muss ebenfalls in der EVCC installiert werden.

Dies sind geheime Daten. Deshalb müssen alle Vorgänge mit dem alten Vertrag Zertifikat der EVCC (Bezogen auf ContractSignatureCertChain enthalten in Zerifikat Installations Anfrage) und unter Verwendung der berechneten DH Secret für Verschlüsselung verschlüsselt werden wie in Anhang I beschrieben wird.

* 1. DHParams

("base64Binary (max length: 256)")

Diffie Hellman-Parameter aus der SA zur Erzeugung des Sitzungsschlüssels an der EVCC, um die Vertragsunterzeichnung Private Key zu verschlüsseln.

* 1. ContractID

("string (max length: 24)")

Element indentifiziert den Lade-Auftrag.

* 1. RetryCounter

("short")

Wenn der „Response Code“ enthält: "FAILED\_NoCertificateAvailable" oder "FAILED\_ContractCanceled", enthält dieses Feld Informationen für den Fall, dass der EVCC versuchen sollte, das neue Zertifikat wieder zu bekommen.

Folgende Einträge sind möglich:

x > 0: nach "x" Tagen

0: sofort (bei der nächsten Ladung)

-1: Nie