SAE International	SURFACE VEHICLE RECOMMENDED PRACTICE	SAE J2847-2™ AUG2012
		Issued 2011-10 Revised 2012-08 Superseding J2847-2™ OCT2011

(R) Communication Between Plug-In Vehicles and Off-Board DC Chargers

RATIONALE

SAE J2847-2 provides messages for DC energy transfer. This is based on the use cases in SAE J2836/2™ that established the DC charging requirements. The initial published version, in October, 2011 includes the signals, sequence diagrams and other requirements. Lab and field testing is being conducted on PowerLine Carrier (PLC) products to insure a robust and viable solution for both Utility and DC charging messages. This testing is underway and this document is being updated with these results to provide a more mature production solution. The initial publication was also completed as a 1st step of completion, to release this from the Draft status within the task force and expose it to other standard bodies and public review for additional comments. This revision will include test results, aspects of PLC products as they complete testing and have continued to mature and comments from other organizations.

The following information was developed as a combined effort between the SAE J2847-2 committee and the DIN SPEC 70121 team.

TO PLACE A DOCUMENT ORDER: **Tel: 877-606-7323 (inside USA and Canada)**
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
SAE WEB ADDRESS: http://www.sae.org

SAE values your input. To provide feedback on this Technical Report, please visit http://www.sae.org/technical/standards/J2847/2_201208

TABLE OF CONTENTS

1. SCOPE

This SAE Recommended Practice SAE J2847-2 establishes requirements and specifications for communication between PLUG-IN ELECTRIC VEHICLE (EVCC)s and the DC Off-board charger. Where relevant, this document notes, but does not formally specify, interactions between the vehicle and vehicle operator.

This document applies to the off-board DC charger for conductive charging, which supplies DC current to the RESS of the electric vehicle through a SAE J1772™ coupler. Communications will be on the J1772 Pilot line for PLC communication. The details of PLC communications are found in SAE J2931/4.

The specification supports DC energy transfer via Forward Power Flow (FPF) from source to vehicle.

1.1 Purpose

The primary purpose of SAE J2847-2 is to provide the communication to achieve RESS charging control irrespective of RESS variations or energy storage technology.

SAE J2847-1 identifies the functional messaging for the PEV to connect to the utility or service provider and participate in available programs. This document, J2847-2, identifies the additional messages for DC energy transfer to the PEV.

The specification supports DC energy transfer via Forward Power Flow (FPF) from grid to vehicle, and DC Reverse Power Flow (RPF) from vehicle to grid is included in SAE J2847-3.[1] DC Forward Power Flow is used to charge the vehicle's rechargeable energy storage system (RESS).

2. REFERENCES

2.1 Applicable Documents

The following publications form a part of this specification to the extent specified herein. Unless otherwise specified, the latest issue of SAE publications shall apply.

2.1.1 SAE PUBLICATIONS

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or 724-776-4970 (outside USA), www.sae.org.

[1] In this specification, we use the terms FPF and RPF as precisely defined, and avoid the term V2G because its meaning has become ambiguous through divergent common usages.

2.1.2 Related Publications (Optional)

The following publications are provided for information purposes only and are not a required part of this SAE Technical Report.

J1715	Hybrid Electric Vehicle (HEV) & Electric Vehicle (EV) Terminology
J1772™	SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler
J2836-1™	Use Cases for Communication Between Plug-in Vehicles and the Utility Grid
J2836-2™	Use Cases for Communication Between Plug-in Vehicles and Off-Board DC Charger
J2847-1	Communication between Plug-in Vehicles and the Utility Grid
J2931-1	Digital Communications for Plug-in Electric Vehicles
J1939	Serial Control and Communications Heavy Duty Vehicle Network - Top Level Document
J2894-1	Power Quality Requirements for Plug-In Vehicle Chargers

2.2 ISO/IEC PUBLICATIONS

IEC61851-1	Electric vehicle conductive charging system - Part 1: General requirements (Under revision)
IEC61851-23	Electric vehicle conductive charging system - Part 23: D.C. electric vehicle charging station (Under development)
IEC61851-24	Digital communication between a d.c. EV charging station and an electric vehicle for control of d.c. charging (Under development)
IEC62196-3	Plugs, socket-outlets, and vehicle couplers – Conductive charging of electric vehicles - Part 3: Dimensional interchangeability requirements for d.c. pin and contact-tube vehicle couplers (Under development)

IEC Publications are also available from the American National Standards Institute, 25 West 43rd Street, New York, NY 10036-8002, Tel: 212-642-4900, www.ansi.org.

2.3 OTHER PUBLICATIONS

DIN SPEC 70121	Electromobility - Digital communication between a d.c. EV charging station and an electric vehicle for control of d.c. charging in the Combined Charging System

3. DEFINITIONS

3.1 ADVANCED METERING INFRASTRUCTURE (AMI)

AMI or Advanced Metering Infrastructure typically refers to the full measurement and collection system that includes meters at the customer site, communication networks between the customer and a service provider, such as an electric, gas, or water utility, and data reception and management systems that make the information available to the service provider.

3.2 BATTERY

See Electric Vehicle Storage Battery.

3.3 BATTERY ELECTRIC VEHICLE (BEV)

The BEV is a vehicle that receives its power solely from batteries, unlike a hybrid vehicle that may receive a portion of its power from an internal combustion engine (ICE). See also PEV.

3.4 BRANCH CIRCUIT

The circuit conductors between the final overcurrent device protecting the circuit and the equipment supplied by the circuit. It is typically an unswitched circuit from the service equipment (fuse box) to an appliance. For this application, the appliance is the Electric Vehicle Supply Equipment (EVSE).

3.5 COMPATIBILITY CHECK

The PEV and the off-board charger should check if they are compatible. For example, the PEV RESS voltage may be higher than what the DC supply can support. Another example is where the PEV RESS maximum voltage is lower than what the DC supply can regulate down to. The vehicle will evaluate the DC supply limit parameters and the vehicle decides whether to proceed with charging.

3.6 DC SUPPLY

It converts AC energy from the grid to the DC energy for the rechargeable energy storage system (RESS). It is also referred as the off-board charger.

3.7 ELECTRIC VEHICLE STORAGE BATTERY (BATTERY)

A group of electrochemical cells electrically connected in a series and/or parallel arrangement, the principal purpose of which is to provide DC electrical energy to propel the EV. May be called Rechargeble Energy Storage System (RESS), or Energy Storage System (ESS)

3.8 ELECTRIC VEHICLE SUPPLY EQUIPMENT (EVSE)

The equipment from the branch circuit to, and including, the connector that couples to the electric vehicle inlet, the purpose of which is to transfer electric energy to an EV. This equipment is located off-board the vehicle.

3.9 END-USE-MEASUREMENT-DEVICE (EUMD)

3.10 ENERGY SERVICE COMMUNICATION INTERFACE (ESCI)

3.11 EVCC

Electric Vehicle Communication Controller

3.12 FORWARD POWER FLOW (FPF)

Forward Power Flow means the direction of energy for charging a Vehicle from the source to the vehicle.

3.13 HANDSHAKING COMPLETE

This is not a signal but it is a state from a combination of several signals between the PEV and the off-board charger. After initiating high level communications, the DC off-board charger must receive: vehicle ready, vehicle maximum current limit, vehicle voltage upper limit and vehicle error code. The PEV must receive: charger maximum current limit, charger maximum voltage limit, charger minimum current limit, and charger minimum voltage limit and charger status code. The vehicle must confirm the connector lock status and will implicitly reflect the lock state in the vehicle ready status. The lock status is not explicitly sent from the vehicle to the DC off-board charger. After all of these signals have been exchanged between the PEV and the off-board DC charger then handshaking complete state is reached.

3.14 HOME AREA NETWORK (HAN)

3.15 MESSAGE

A message may include several signals packaged together in a group such that the entire group can be sent together over a physical layer interface. The physical layer will be defined in J2931 documents.

3.16 NO DATA PARAMETER

This defines a means for the Vehicle or DC Supply to indicate that a signal is not available (i.e., hasn't been read yet).

3.17 PLUG-IN ELECTRIC VEHICLE (EVCC)

Any class of vehicle BEV, PHEV, Electric Tug, etc. which can be plugged in to receive power from the Electrical Grid where this power is then used to apply traction to the vehicle wheels.

3.18 POWER FLOW

See Forward Power Flow and Reverse Power Flow.

3.19 POWER LINE COMMUNICATION (PLC)

Power line communication (PLC), also called power line carrier, mains communication, power line telecom (PLT), or power line networking (PLN), are terms describing several different systems for using electric power lines to carry information over the power line.

Electrical power is transmitted over high voltage transmission lines, distributed over medium voltage, and used inside buildings at lower voltages. Power line communications can be applied at each stage. Most PLC technologies limit themselves to one set of wires (for example, premises wiring), but some can cross between two levels (for example, both the distribution network and premises wiring).

All power line communications systems operate by impressing a modulated carrier signal on the wiring system. Different types of power line communications use different frequency bands, depending on the signal transmission characteristics of the power wiring used. Since the power wiring system was originally intended for transmission of AC power, the power wire circuits have only a limited ability to carry higher frequencies. The propagation problem is a limiting factor for each type of power line communications.

Data rates over a power line communication system vary widely. Low-frequency (about 100-200 kHz) carriers impressed on high-voltage transmission lines may carry one or two analog voice circuits, or telemetry and control circuits with an equivalent data rate of a few hundred bits per second; however, these circuits may be many miles (kilometers) long. Higher data rates generally imply shorter ranges; a local area network operating at millions of bits per second may only cover one floor of an office building, but eliminates installation of dedicated network cabling.

3.20 PRE-CHARGE

Precharging the DC bus ensures that the voltage supplied by the EVSE is nearly matched to the Vehicle RESS voltage prior to closing the vehicle contactors. This ensures that there is no possibility of a high in-rush current at the time of contactor closure.

3.21 RECHARGEABLE ENERGY STORAGE SYSTEM (RESS)

A rechargeable system that stores energy for delivery of electric energy. May also be called Electric Vehicle Storage Battery or Battery.

3.22 REVERSE POWER FLOW (RPF)

Reverse Power Flow means the direction of energy for discharging a Vehicle from the vehicle to the load.

3.23 SECC

Supply Equipment Communication Controller

3.24 SIGNAL

A signal refers to the individual data elements that are communicated between the Plug In Vehicle and the Off-Board charger. The signals are listed in Section 4.2 and in the Table in Appendix B.

3.25 SIGNAL BIT SIZE

The bit size defines how many digital bits are needed to represent the full range signal, including no data

3.26 SIGNAL DEFAULT VALUE

The default value is what is assumed if a signal has not yet been received over the communications bus.

3.27 SIGNAL OFF-SET

Signal Offset defines how an integer is adjusted to achieve the full range at the defined resolution. Offset can be positive or negative.

3.28 SIGNAL RANGE

The Range defines the signal limits in engineering units.

3.29 SIGNAL RESOLUTION

The resolution determines how much a signal can change with each bit step in the integer representation.

3.30 SIGNAL UPDATE RATE

This defines the expected frequency at which a periodic signal must be received over the communications bus for proper system control. If a particular signal is missing for more than ten full rate cycles, (based on section 5.6.2) then the system should enter a faulted state, and proceed to shutdown energy transfer.

DC off-board charger must receive: vehicle ready, vehicle maximum current limit, vehicle voltage upper limit and vehicle error code.

3.31 V2G

Vehicle to Grid Communication, for the scope of this document the grid is the EVSE.

4. TECHNICAL REQUIREMENTS

4.1 Charging Phases

Reference Appendix F in the J1772 document for Details about the charging phases. A normal charging session is broken up into four phases: Initialization, Pre-charge, Energy Transfer, and Shutdown. Additionally, there are three types of Emergency shutdown: Vehicle Initiated, Supply Initiated, and Cord Disconnection.

4.1.1 Message Sequences

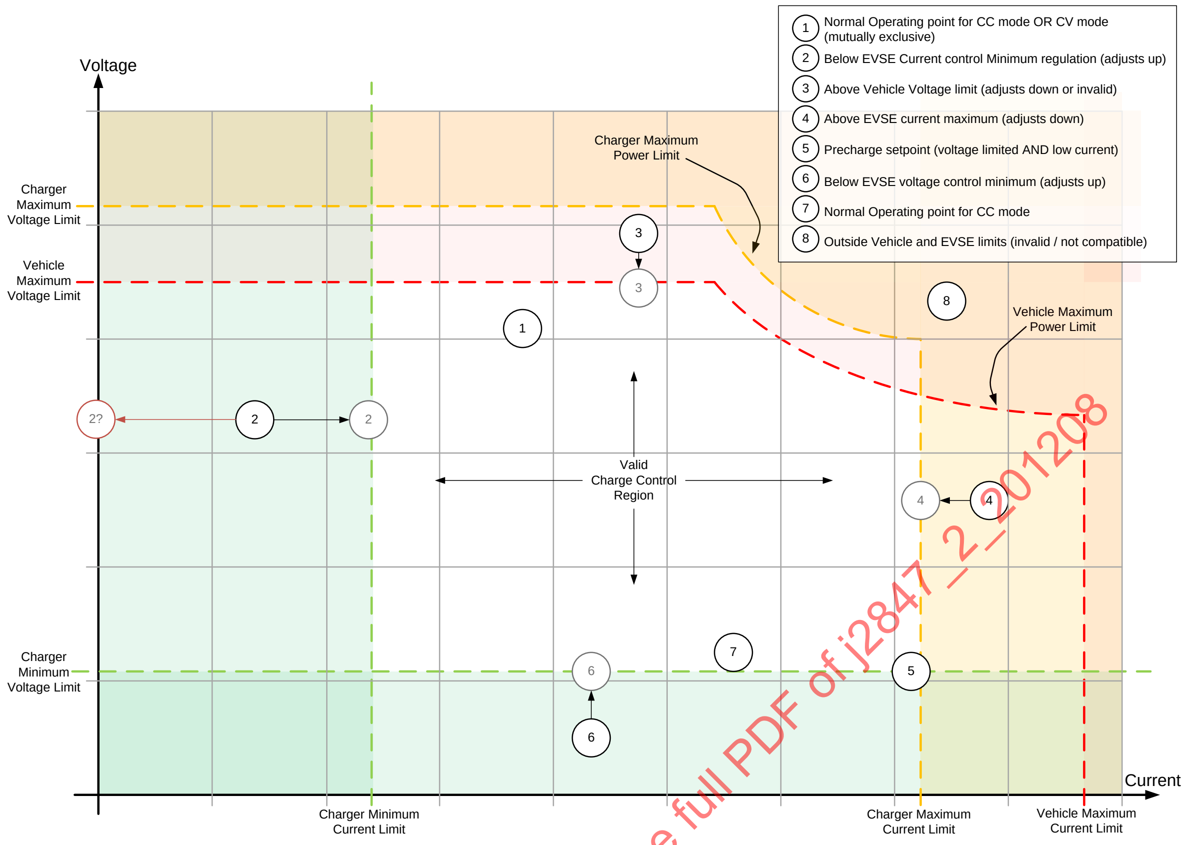
Message Sequences are defined for several scenarios

- Normal Startup (Figure A1)
- Normal Shutdown (Figure A2)
- Vehicle Initiated Shutdown (Figure A3)
- Charger Initiated Shutdown (Figure A4)
- Loss of Pilot Shutdown (Figure A5)

4.1.2 Charging Limits

Throughout the charging session the vehicle and DC charger will both communicate their respective operation limits. These limits include maximum and minimum for current, maximum voltage and maximum power. The charger will regulate its output control to maintain these limits.

The diagram below shows how those limits are to be maintained at a few sample operating points:


1 Normal Operating point for CC mode OR CV mode (mutually exclusive)
2 Below EVSE Current control Minimum regulation (adjusts up)
3 Above Vehicle Voltage limit (adjusts down or invalid)
4 Above EVSE current maximum (adjusts down)
5 Precharge setpoint (voltage limited AND low current)
6 Below EVSE voltage control minimum (adjusts up)
7 Normal Operating point for CC mode
8 Outside Vehicle and EVSE limits (invalid / not compatible)
Voltage
Current
Charger Maximum Voltage Limit
Vehicle Maximum Voltage Limit
Charger Minimum Voltage Limit
Charger Maximum Power Limit
Vehicle Maximum Power Limit
Valid Charge Control Region
Charger Minimum Current Limit
Charger Maximum Current Limit
Vehicle Maximum Current Limit
2?

4.2 Signal Definitions

This section describes the details of the signals used to control an off-board charger. Several signals will be combined together to form EXI encoded XML schema (see section 6). The Message packaging structure is outside the scope of this document in this current revision.

Refer to table in section 6.1 for the message signal content. Refer to table in Appendix B for signal details.

4.2.1 Bulk Charging Complete

Source: EVCC
Destination:SECC

This is the signal that indicates that the RESS has reached the bulk charge level (generally up to 80 to 100% SoC). This is optional, and is used for display purposes on the charging station. See "Charging Phases" above for more details. Indicates that the RESS system has completed the High Power charging

4.2.2 Charging Complete

Source: EVCC
Destination: SECC

This is the signal that indicates that the RESS has reached the full charge level (up to 100% SoC). The status is reported to the DC Supply for presentation in a display. Note that the charge process may continue with energy transfer to support vehicle accessory loads (cabin conditioning, RESS conditioning, etc.) after this signal transitions to TRUE.

4.2.3 Vehicle Energy Capacity

Source: EVCC
Destination: SECC

This signal indicates the maximum designed energy capacity allowed by the vehicle manufacturer. The DC Supply could use this signal for safety mitigation from the charger side. Charger may optionally stop after delivering this amount of energy the vehicle would accept. That value includes RESS capacity and vehicle auxiliary energy.

4.2.4 Vehicle RESS SOC

Source: EVCC
Destination: SECC

This signal is defined by the vehicle manufacture to represent the relative charge level as a percentage of the full energy storage of the Vehicle RESS. It's primarily used for display on the charging supply. If the vehicle uses a battery for energy storage, then this would be Battery SOC.

4.2.5 Vehicle Maximum Current Limit

Source: EVCC
Destination: SECC

It's a signal from the vehicle to the DC Supply indicating its maximum allowed current. In other words, it's the maximum current limit allowed by the vehicle. It may depend on RESS temperature, SoC, etc. The charger shall not exceed the minimum of this signal and the Charger Maximum Current Limit. The primary purpose of the Current Limit signal is for the DC charger to regulate its fast internal (ms) regulation loops. The DC Supply hardware and software shall control the output so that there will be no violations of this limit. The vehicle may optionally change this signal dynamically throughout the charging session.

4.2.6 Vehicle Maximum Power Limit

Source: EVCC
Destination: SECC

It's the maximum power limit allowed by the vehicle. It may depend on RESS temperature, SoC, etc. The charger shall not exceed the minimum of this signal and the Charger Maximum Power Limit. The primary purpose of the Power Limit signal is for the DC charger to regulate its fast internal (ms) regulation loops. The vehicle may optionally change this signal dynamically throughout the charging session.

4.2.7 Vehicle Maximum Voltage Limit

Source: EVCC
Destination: SECC

It's the maximum voltage limit allowed by the vehicle. The value may depend on Inlet rating, RESS temperature, SoC, etc. The charger shall not exceed the minimum of this signal and the Charger Maximum Voltage Limit. The primary purpose of the Voltage Limit signal is for the DC charger to regulate its fast internal (ms) regulation loops. The vehicle may optionally change this signal dynamically throughout the charging session.

When vehicle needs to use constant voltage mode, this signal will be adjusted by the vehicle RESS controls to maintain a constant voltage around a low current.

4.2.8 Charger Maximum Power Limit

Source: SECC
Destination: EVCC

It's the maximum power that can be delivered to the PEV. The charger should regulate this value. The charger shall not exceed the minimum of this signal and the Vehicle Maximum Power Limit. The DC supply can dynamically change this value throughout the charging session (as in the case where a utility demand reduces the available power).

4.2.9 Charger Maximum Current Limit

Source: SECC
Destination: EVCC

It's the maximum current that can be delivered to the PEV. The charger should regulate this value. The charger shall not exceed the minimum of this signal and the Vehicle Maximum Current Limit. This value can change dynamically through the charging session.

4.2.10 Charger Maximum Voltage Limit

Source: SECC
Destination: EVCC

It's the maximum voltage that can be maintained by the charger. The value may depend on Coupler rating, ambient temperature, etc. The charger should regulate this value. The charger shall not exceed the minimum of this signal and the Vehicle Maximum Voltage Limit. This value can change dynamically through the charging session.

4.2.11 Charge Current Request

Source: EVCC
Destination: SECC

This signal is the instantaneous charging current request from the vehicle. The request should be within the established charger and vehicle limits from handshaking. In accordance with J1772, the current request is allowed to be higher than the Charger Maximum Current Limit (4.2.9), but the DC Supply will reduce the output to maintain the limit. The DC supply is allowed to overshoot or undershoot around this target as long as it doesn't violate 4.2.5, and the requirements specified in SAE J1772™. In the case where any of the charging limits have been achieved (4.2.23, 4.2.24, 4.2.25), the charge current request may be higher than the actual measured current reading. Since the vehicle is controlling charge current and reading on-vehicle current sensors, the vehicle will ensure that the request is adjusted to achieve the vehicle control target.

Since it is not always possible to achieve the PEV Charge Current Request and PEV Target Voltage (section 0) simultaneously, the DC Supply will only regulate to one of the requests. The PEV should not issue a fault when both cannot be achieved simultaneously.

4.2.12 Vehicle Ready

Source: EVCC
Destination: SECC

It informs the charger that the PEV is ready to be charged. It is needed if there is no control pilot signal. It's redundant to control pilot signal.

4.2.13 Vehicle Error Code

Source: EVCC
Destination: SECC

It indicates the PEV internal error state that will inhibit/abort charging. This is for display purposes only to guide the user through the charging process for the particular vehicle. Not all vehicle manufacturers will have this feature.

In situations where a vehicle might have more than one state from the list below, the vehicle (or OEM) will determine the highest priority item to send to the charger for display. Only one item from the list can be sent in this signal.

This signal may contain any value defined by the enumerated type DC_EVErrorCodeType. The value and meaning of each enumeration is described in the table below.

XML Enumeration	**Comments / Description**
NO_ERROR	Default value, when EV has no Error detected
FAILED_RESSTemperatureInhibit	Battery Temperature Inhibit, Battery too hot/cold to accept charge
FAILED_EVShiftPosition	Vehicle Shift Position, Vehicle is not in Park
FAILED_ChargerConnectorLockFault	Charger Connector Lock Fault, Vehicle has not detected the Charge cord connector locked into the inlet or failure where connector cannot be unlocked from the charging inlet.
FAILED_EVRESSMalfunction	Vehicle RESS Malfunction, Any non-recoverable fault or error condition of the Vehicle RESS.
FAILED_ChargingCurrentdifferential	Charging Current Differential, Indication that vehicle has stopped the charging session after detecting that the charging station is not able to maintain the current request.
FAILED_ChargingVoltageOutOfRange	Charging voltage out of range, Indication that vehicle has stopped the charging session after detecting that the RESS is either under or above normal operating voltage range.
Reserved_A	Reserved for future use.
Reserved_B	Reserved for future use.
Reserved_C	Reserved for future use.
FAILED_ChargingSystemIncompatibility	Charging System Incompatibility, If the vehicle determines that the charging station is incompatible.
NoData	No Data, Only used when vehicle has not yet determined its operating state.

4.2.14 Charger Status Code

Source: SECC
Destination: EVCC

It indicates the charger internal charging sequence state. That could be charger malfunction, Utility Interrupt Event, charger ready, Pre-charge, charger standby, etc.

In situations where a charger might have more than one state from the list below, the charger will determine the highest priority item to send to the vehicle for display.

This signal may contain any value defined by the enumerated type DC_EVSEStatusCodeType. The value and meaning of each enumeration is described in the table below.

XML Enumeration	Comments / Description
EVSE_NotReady	Charger Standby, Not authorized, On maintenance, Initialization state etc.
/EVSE_Ready	Charge Station is ready to proceed/continue with the charge process, Charger is charging
EVSE_Shutdown	Customer/EVSE Initiated normal Shutdown
EVSE_UtilityInterruptEvent	Utility or Equipment operator has requested a temporary reduction in load.
EVSE_Precharge	Indicates the EVSE is ready for or is currently in the pre-charge phase of the charge process
EVSE_IsolationMonitoringActive	Indicates the EVSE is in a state where it is performing an internal isolation check or cable isolation check.
EVSE_EmergencyShutdown	Emergency Shutdown or 'E-Stop' button pressed at charging station.
EVSE_Malfunction	Any type of non-recoverable charger fault. Ex: Coupler overtemperature, Isolation Failure, etc.
Reserved_8	Reserved
Reserved_9	Reserved
Reserved_A	Reserved
Reserved_B	Reserved
Reserved_C	Reserved

4.2.15 Remaining Charging Time to Bulk SoC

Source: EVCC
Destination: SECC

This signal from the vehicle allows the charging station to display a time remaining count. This value is the Remaining time to charge PEV up to the bulk SoC. The time represents how much time it will take to get from the actual SoC to the bulk SoC. The DC Supply may optionally use this periodic signal from the vehicle in order to display information to the user. The DC Supply will not stop the charging session based on this time; the vehicle always controls the charging session stop time. OEM will decide bulk level (80% for example)

4.2.16 Bulk SoC

Source: EVCC
Destination: SECC

It’s the end point (SoC) where optimum fast charge ends. Each vehicle OEM will decide the bulk level value (80% for example). The displayed SoC could indicate this from actual SoC to bulk percent if desired.

4.2.17 Remaining Time to Full SoC

Source: EVCC
Destination: SECC

Remaining time to charge PEV to full SoC; that is, the interval in time when no more energy is needed to charge the RESS. Display time to charge from actual SoC to full SoC. The DC Supply may optionally use this periodic signal from the vehicle in order to display information to the user. The DC Supply will not stop the charging session based on this time; the vehicle always controls the charging session stop time.

4.2.18 Full SoC

Source: EVCC
Destination: SECC

This signal defines the SOC where the vehicle considers the RESS is full. It is defined by each vehicle OEM and allows the charger to display the full charge status.

4.2.19 Vehicle Energy Request

Source: EVCC
Destination: SECC

This signal provides the EVSE with an indication of the amount of energy that will be requested during the current charging session.

4.2.20 Charger Energy to be delivered

Source: SECC
Destination: EVCC

It's the allowed or requested units of energy to be delivered during charging (a charging target). This allows public charging stations to limit charging session by limiting the energy transfer (i.e., for a prepaid amount).

4.2.21 Voltage Output

Source: SECC
Destination: EVCC

It's the measured High Voltage DC voltage at the output of the DC Supply. It's used by the PEV during charging for diagnosis of on board sensors. It may be opportunistically used by Vehicle during charge as a redundant check to diagnose on-board sensors.

4.2.22 Current Output

Source: SECC
Destination: EVCC

It's the measured High Voltage DC current at the output of the DC Supply through the charge coupler.

NOTE: The range allows for negative current measurements in the event that the energy is flowing out of the vehicle's charge receptacle.

4.2.23 Charger Current Limit Achieved

Source: SECC
Destination: EVCC

The charger is at the limit of its source capability (saturation). The charger becomes saturated when either the charger limits (4.2.9) are achieved or the vehicle limits (4.2.5) from handshaking have been achieved. It's used by the PEV control to compensate for reaching the limit. This would allow vehicle to refrain from setting a fault when the charger is providing current to the limit of the present operating conditions.

The default state for this signal is FALSE.

4.2.24 Charger Voltage Limit Achieved

Source: SECC
Destination: EVCC

The charger is at the limit of its source capability (saturation). The charger becomes saturated when either the charger limits are achieved or the vehicle limits (4.2.7) from handshaking have been achieved. It's used by the PEV control to compensate for reaching the limit. This would allow vehicle to refrain from setting a fault when the charger is providing voltage to the limit of the present operating conditions.

4.2.25 Charger Power Limit Achieved

Source: SECC
Destination: EVCC

The charger is at the limit of its source capability (saturation). The charger becomes saturated when either the charger limits (4.2.8) are achieved or the vehicle limits (4.2.6) from handshaking have been achieved. It's used by the PEV control to compensate for reaching the limit. This would allow vehicle to refrain from setting a fault when the charger is providing power to the limit of the present operating conditions.

4.2.26 Charger Minimum Voltage Limit

Source: SECC
Destination: EVCC

Minimum Voltage available from the charger to the vehicle. It informs the PEV that the DC supply can only regulate its voltage output down to this level. If a vehicle has an RESS that must operate below this level, then the vehicle will determine that the charging station is incompatible. The vehicle will not start charging or will stop a charging session if the target RESS voltage falls below this value. The charger should regulate down to this value. The vehicle will decide if this is compatible with its RESS. A vehicle with an RESS below this value would not be compatible with the charger.

4.2.27 Charger Minimum Current Limit

Source: SECC
Destination: EVCC

Minimum Current available from the charger to the vehicle. It informs the PEV that the DC supply can only regulate its current output down to this level. If a vehicle has RESS that must operate below this level, then the vehicle will determine that the charging station is incompatible. The vehicle will not start charging or will stop a charging session if the target RESS current falls below this value. The charger should regulate down to this value while still meeting the accuracy requirements. All chargers should be able to respond when the vehicle requests zero, that the output of the charger should be shut off to zero current.

4.2.28 Charger Peak Current Ripple

Source: SECC
Destination: EVCC

The DC Supply's peak to peak ripple current amplitude of the regulated output at any current request. This signal informs the PEV what peak-to-peak magnitude of the DC Supply current ripple can be expected at any charge level. This is dependent on the charger filter and control design. It is expected that quick chargers will have higher ripple, and may not be suitable for low current topoff charging (Bulk level to Full level). The vehicle may modify its charging strategy using this information. The vehicle may chose not to initiate the charging session if the value is not compatible with the RESS charging strategy. The vehicle may not support topoff charging if it cannot accept this amplitude of ripple current. The vehicle manufacturer can use this signal to optimize the charging algorithm.

4.2.29 Charger Current Regulation Tolerance

Source: SECC
Destination: EVCC

The DC Supply's ability to regulate current with respect to the current request. This signal informs the PEV what absolute magnitude of the DC Supply regulation tolerance can be expected at any charge level. This is dependent on the charger filter and control design. It is expected that quick chargers will have larger tolerance, and may not be suitable for low current topoff charging (Bulk level to Full level). The vehicle may modify its charging strategy using this information. The vehicle may chose not to initiate the charging session if the value is not compatible with the RESS charging strategy. The vehicle may not support topoff charging if it cannot accept this tolerance of current. The vehicle manufacturer can use this signal to optimize the charging algorithm.

4.2.30 Vehicle Requested Energy Transfer Type

Source: EVCC
Destination: SECC

This signal informs the DC supply of the type of energy transfer that the EVCC wishes to perform..This allows a vehicle to connect to an EVSE that can do either DC or AC energy transfer.

This signal allows the Vehicle to select its desired energy transfer type in the case that both the EVSE and Vehicle support multiple energy transfer types and J1772™ connector variations (AC, DC type 1, DC Combo). The charging station should confirm that it is capable of supporting the requested type, and should only complete handshaking if it is compatible.

This signal may contain any value defined by the enumerated type EVRequestedEnergyTransferType. The value and meaning of each enumeration is described in the table below.This signal may contain any value defined by the enumerated type EVSESupportedEnergyTransferType. The value and meaning of each enumeration is described in the table below.

XML Enumeration	Comments / Description
AC_single_phase_core	Vehicle is requesting single phase AC energy transfer on core pins as specified through the SAE J1772 type 1 or type 2 connector.
AC_three_phase_core	Vehicle is requesting three phase AC energy transfer as specified through the SAE 1772 type 1 or type 2 connector on Type 2 core pins.
DC_core	Vehicle is requesting DC energy transfer as specified through the SAE 1772 type 1 or type 2 connector using the standard core connector pins. (couple type C1)
DC_extended	Vehicle is requesting DC energy transfer as specified through the SAE 1772 type 1 or type 2 combo connector over the additional pins outside the core (coupler type C2)
DC_combo_core	Vehicle is requesting DC energy transfer as specified through the SAE 1772 type 1 or type 2 combo connector over the core pins (coupler type C2)
DC_unique	In the case of a dedicated coupler for Level 3 DC charging (coupler type C3)

4.2.31 Charger Supported Energy Transfer Type

Source: SECC
Destination: EVCC

This signal informs the Vehicle of all Energy transfer types supported by the EVSE. Since an EVSE may support multiple types of energy transfer (AC, DC type 1, DC combo) this signal allows a vehicle know all options of charging that are supported by the EVSE to determine if it is compatible and select the desired energy transfer type. Vehicle will only complete handshaking if one of the supported types matches the vehicle configuration. If the vehicle cannot use any of the supported charging types from the DC supply, it shall respond with Vehicle Error Code enumeration FAILED_ChargingSystemIncopatibility.

If a EV is connected to an EVSE that supports Digital Communications (5% pilot duty cycle) and the EV supports Digital Communication only for DC charging, the EV shall be able to request the Charge Station to revert to a non-Digital Communication method of AC charging in order to proceed with the charge process. If the EV does not initiate digital communications from the EVSE, after a timeout period the EVSE might revert to normal AC charging mode by updating the duty cycle as per SAE J1772 to proceed with the charge process.

Once the EV receives the "Charger Supported Energy Transfer Type" of the Charge Station, and does not want to continue AC Charging with Digital Communications, the EV shall request to terminate the charge process and send a "PEV Error Code" enumeration of FAILED_ChargingSystemIncompatibility. The Charge Station shall then abort digital communications and revert to the lower level version of AC charging using the pilot duty cycle to communicate the EVSE current limit.

This is an enumerated type signal defined by:

XML Enumeration	Comments / Description
AC_single_phase_core	The EVSE supports single phase AC energy transfer as specified through the SAE 1772 type 1 or IEC type 2 connector.
AC_three_phase_core	The EVSE supports three phase AC energy transfer as specified through the IEC type 2 connector.
DC_core	The EVSE supports DC energy transfer as specified through the SAE 1772 type 1 or type 2 connector on the core pins.
DC_extended	The EVSE supports DC energy transfer as specified through the SAE 1772 type 1 or type 2 combo connector on the extended pins.
DC_combo_core	The EVSE supports DC energy transfer on a combo connector core pins (but not on the extended pins)
DC_dual	The EVSE supports DC energy transfer on a combo connector core pins and on the extended pins (but not at the same time)
AC_core1p_DC_extended	The EVSE supports AC energy transfer on a combo connector core pins and DC energy transfer on the extended pins (but not at the same time)
AC_single_DC_core	The EVSE supports single phase AC or DC energy transfer on Type 1/Type 2 connector core pins (but not at the same time)
AC_single_phase_three_phase_core_DC_extended	The EVSE supports single phase AND three phase AC energy transfer on a combo connector using combination of core pins and DC on extended pins (but not both at the same time)
AC_core3p_DC_extended	The EVSE supports three phase AC energy transfer on a combo connector using combination of core pins and DC on extended pins (but not both at the same time)

4.2.32 Charge Station Isolation Status

Source: SECC
Destination: EVCC

This signal informs the PEV the status of the isolation monitoring on the DC Supply. This signal is sent to the vehicle periodically throughout the charging session. Consult the SAE J1772™ and IEC 61851 part 23 documents for further details regarding the isolation thresholds. Until the first reading has been measured, the DC supply must report the Invalid state.

This signal may contain any value defined by the enumerated type EVSESupportedEnergyTransferType. The value and meaning of each enumeration is described in the table below.

XML Enumeration	Comments / Description
Valid	Isolation measurement confirms the system is within the Valid region above thresholds.
Warning	Isolation has detected a measurement below the warning level threshold
Fault	Isolation has detected a measurement below the Fault level threshold
Invalid	DC Supply may transmit this enumeration at startup before measurement has been confirmed

4.2.33 PEV Target Voltage

Source: EVCC
Destination:SECC

This signal tells the Charger the target voltage request of the vehicle. This signal shall not exceed the Vehicle Maximum Voltage Limit. During the charge process, the EVSE must regulate the current output to close loop around the target voltage as it may be used by the vehicle for voltage control charging. This signal shall be used during the pre-charge Phase of Initialization to command the DC Supply voltage to match the RESS.

Since it is not always possible to achieve the PEV Charge Current Request (section 4.2.11) and PEV Target Voltage simultaneously, the DC Supply will only regulate to one of the requests. The PEV should not issue a fault when both cannot be achieved simultaneously.

4.2.34 Ready To Charge State

Source: EVCC
Destination:SECC

This signal tells the EVSE that the vehicle is ready for full energy transfer (i.e., any contactors are closed and Pre-charge is complete) and the EVSE can enable its full high voltage output regulation. Any circuits related to inrush current limiting and HV bus precharging can be disabled in the EVSE to allow regulated control of the output power. This commands the DC Supply to switch on the High Power conversion circuit.

During Shutdown, this signal tells the EVSE that the vehicle has completed charging, and that the EVSE should discharge the HV output bus.

4.2.35 PEV Cabin Conditioning

Source: EVCC
Destination:SECC

A Boolean state that informs the charger that the EVCC is performing some type of cabin conditioning. This can be used for display purposes on the charge station.

4.2.36 PEV RESS Conditioning

Source: EVCC
Destination:SECC

A Boolean state that informs the charger that the EVCC is performing some type of HV battery thermal conditioning. This can be used for display purposes on the charge station.

4.2.37 Response Code

Source: SECC
Destination: EVCC

This response code is included in all response messages except Supported Application Protocol (see below).

This enumerated state encoded signal indicates how the charge station has responded to a specific request message from the vehicle. Depending on the value in the Response Code the vehicle decides if it can proceed with the charge process or if it has to handle an error. This signal may contain any value defined by the enumerated type responseCodeType. The value and meaning of each enumeration is described in the table below.

Any Value starting with “FAILED” indicates an negative response. Detailed information may be provided by FAILED_<additional info>. This information may be used to differentiate the reaction on the negative response.

Any Value starting with “OK” indicates a positive response. Detailed information may be provided by OK_<additional info>. This information may be used to differentiate the reaction on the positive response.

XML Response Code Enumeration	Comments / Description
OK	Indicates if the processing of the request message was successful and no specific ResponseCodeType is defined for the current state.
OK_NewSessionEstablished	Indicates processing of the SessionSetupReq message was successful and a different SessionID is contained in the response message than the SessionID in the request message.
OK_OldSessionJoined	Indicates processing of the SessionSetupReq message was successful and the same SessionID as used in the request message is contained in the response message
OK_CertificateExpiresSoon	Not used for DC charging
FAILED	Indicates the processing of the request message was not successful and no specific ‘responseCodeType’ is defined for the current error case.
FAILED_SequenceError	Indicates the EVSE has received an unexpected request message
FAILED_ServiceIDInvalid	Not used for DC charging
FAILED_UnknownSession	Indicates the SessionID in the request message does not fit to the EVSE provided SessionID during SessionSetupRes.
FAILED_ServiceSelectionInvalid	Indicates the SelectedServiceList contained in the ServicePaymentSelectionReq message contains a ServiceID which was not contained in the offered ServiceList of ServiceDiscoveryRes
FAILED_PaymentSelectionInvalid	Indicates the SelectedPaymentOption contained in the ServicePaymentSelectionReq message was not part of the offered PaymentOptions of ServiceDiscoveryRes.
FAILED_CertificateExpired	Not used for DC charging
FAILED_SignatureError	Indicates the validation of the Security element in the message header failed.
FAILED_NoCertificateAvailable	Not used for DC charging
FAILED_CertChainError	Not used for DC charging
FAILED_ChallengeInvalid	Not used for DC charging
FAILED_ContractCanceled	Not used for DC charging
FAILED_WrongChargeParameter	Indicates if the contents of ChargeParameterDiscoveryReq message is not valid, e.g., wrong parameter set is provided, one or multiple parameters cannot be interpreted
FAILED_PowerDeliveryNotApplied	Indicates the EVSE is not able to deliver energy
FAILED_TariffSelectionInvalid	Indicates the charging profile in the PowerDeliveryReq message contains a SAtupleID which was not contained in the 'SASchedules' attribute provided in 'ChargeParameterDiscoveryRes'
FAILED_ChargingProfileInvalid	Indicates the charging profile in the PowerDeliveryReq message violates a power limitation provided in 'ChargeParameterDiscoveryRes'.
FAILED_EVSEPresentVoltageToLow	Not used for DC charging
FAILED_MeteringSignatureNotValid	Not used for DC charging
FAILED_WrongEnergyTransferType	The vehicle requested energy transfer does not match what the DC Supply is able to deliver.

4.2.38 Supported Application Protocol Response Code

Source: SECC
Destination: EVCC

This enumerated state encoded signal indicates how the charge station has responded to the supportedAppProtocolReq message from the vehicle.

XML Response Code enumeration	**Comments / Description**
OK_SuccessfulNegotiation	successful negotiation of a application protocol
OK_SuccessfulNegotiationWithMinorDeviation	successful negotiation of a protocol, with minor deviation
Failed_NoNegotiation	failed negotiation of protocols

4.2.39 EVSE ID

Source: SECC
Destination: EVCC

This signal tells the Vehicle the unique identification number of an EVSE.

4.2.40 PEV ID

Source: EVCC
Destination: SECC

This signal tells the EVSE the unique identification number of a vehicle. It contains the HPGP Ethernet MAC address of the vehicle.

4.2.41 Priority

Source: EVCC
Destination: SECC

This signal is used by the vehicle for indicating the protocol priority of a specific protocol allowing the EVSE to select a protocol based on priorities.

4.2.42 Protocol Namespace

Source: EVCC
Destination: SECC

This signal is used by the vehicle to uniquely identify the Namespace URI of a specific protocol supported by the vehicle, i.e., this is the protocol name of the related protocol.

4.2.43 Schema ID

Source: EVCC
Destination: SECC

This signal is used by the vehicle to indicate the schema ID assigned by the vehicle to the protocol indicated in the signals ProtocolNamespace, VersionNumberMajor and VersionNumberMinor. This message element is used by the EVSE to reference one of the vehicle supported protocols received in the request message. This identifier allows also for referring to a particular protocol later on in the communication process (EXI Option schemaID).

4.2.44 Version Number Major

Source: EVCC
Destination: SECC

This signal is used by the vehicle to indicate the major version number of the protocol indicated in the message element ProtocolNamespace.

This is sent at the beginning of the charging session in order to coordinate which signals can be used. This signal was included to support the existence of future protocols It is expected that if a new protocol is developed, that devices will fall back to support the lower level protocol.

4.2.45 Version Number Minor

Source: EVCC
Destination: SECC

This signal is used by the vehicle to indicate the minor version number of the protocol indicated in the message element ProtocolNamespace.

Minor version changes shall be backward compatible as long as the Major version number has not changed.

4.2.46 Date Time Now

Source: SECC
Destination: EVCC

This signal is the timestamp of the current time using to the Unix Time Stamp format. This signal is used by the vehicle to check the validity of the certificates for contract based charging and as external time reference. Based on this information the vehicle might implement a strategy when certificate updates are required. Using this signal avoids requiring the time base of the vehicle and EVSE to be synchronized.

4.2.47 Session ID

Source: EVCC
Destination:SECC

This signal is used by the vehicle and EVSE for uniquely identifying a V2G communication session.

4.2.48 Payment Options

Source: SECC
Destination: EVCC

This signal includes the list of payment options an EVSE offers to the vehicle indicating what method could be chosen to pay for the services. The vehicle can only select one payment method for all services used.

NOTE: Only the payment option “external payment” shall be used, since detailed payment options are not yet defined.

XML Response Code enumeration	**Comments / Description**
Contract	Not used with DC charging
ExternalPayment	Indicates that external payment methods are being used to enable the charge process.

4.2.49 Selected Payment Option

Source: EVCC
Destination:SECC

This enumerated state signal is used for indicating the payment type selected for the use of all selected services in the selectedServiceList

XML Response Code enumeration	Comments / Description
Contract	Not used with DC charging
ExternalPayment	Indicates that external payment methods are being used to enable the charge process.

NOTE: Only the payment option “external payment” shall be used, since detailed payment options are not yet defined.

4.2.50 Charge Service

Source: EVCC
Destination: SECC

Available charging services supported by the EVSE.

4.2.51 SAS Schedule List

Source: EVCC
Destination: SECC

Optional Element: Includes several tuples of schedules from secondary actors

5. MESSAGES

5.1 Message Summary

5.1.1 Session Setup Request

By using the SessionSetupReq message the EV establishes a V2G communication session.

5.1.2 Session Setup Response

By using the SessionSetupRes the SECC responds to an SessionSetupReq. With the SessionSetup Response the EVSE notifies the EV with an enclosed response code, whether establishing a new session or joining a previous Session was successful or not.

5.1.3 Service Discovery Request

By sending the ServiceDiscoveryReq message the EVCC triggers the SECC to send information about all services offered by the SECC. Furthermore, the EVCC can limit for particular services by using the service scope and service type elements.

5.1.4 Service Discovery Response

After receiving the ServiceDisoveryReq message of the EVCC the SECC sends the ServiceDiscoveryRes message. In case of a successful service discovery, the response lists all available services of the SECC for the defined criteria. In case the service discovery failed the service list is empty and the response code indicates potential reasons.

5.1.5 Service and Payment Selection Request

This request message transports the information on the selected services and on how the all the selected services are paid.

5.1.6 Service and Payment Selection Response

With this message the SECC informs the EVCC whether the selected services and payment option were accepted.

5.1.7 Charge Parameter Discovery Request

By sending the Charge Parameter Discovery Request message the EVCC provides its charging parameters to the SECC. This message provides status information about the EV and additional charging parameters, like estimated energy amounts for recharge and the point in time for the end of charge.

5.1.8 Charge Parameter Discovery Response

With the Charge Parameter Discovery Response message the SECC provides applicable charge parameters from the grid's perspective. Next to general charge parameters of the EVSE this optionally includes further information on cost over time, cost over demand, cost over consumption or a combination of these. The term cost refers to any kind of cost specified in this version of the standard and is not limited to monetary costs. Based on this cost information the EV may optimize its charge for the requested amount of energy.

5.1.9 Power Delivery Request

By sending the Power Delivery Request the EVCC requests the EVSE to switch power on and transmits the charging profile it will follow during the charging process.

5.1.10 Power Delivery Response

After receiving the Power Delivery Request message of the EVCC the SECC sends the Power Delivery Response message including information if power will be available.

5.1.11 Cable Check Request

With the cable check request, the EV asks the EVSE to perform a cable check, which includes an isolation test, before charging.

5.1.12 Cable Check Response

After receiving the Cable Check Request of the EV the EVSE sends the Cable Check Response informing the EV about result of cable check and EVSE status.

5.1.13 Pre Charging Request

With the Pre Charging Request the EV asks the EVSE to apply certain values for output voltage and output current. Since the contactors of the EV are open during Pre Charging, the actual current flow from the EVSE to the EV will be very small, i.e., in most cases smaller than the requested output current. The EV may use several Pre Charging Request/Response message pairs in order to precisely adjust the EVSE output voltage to the EV RESS voltage measured inside the EV.

5.1.14 Pre Charging Response

After receiving the Pre Charge Request of the EV the EVSE sends the Pre Charge Response informing the EV about EVSE status and present EVSE output voltage.

5.1.15 Current Demand Request

By sending the Current Demand Request the EV requests a certain current from EVSE. Also the target voltage and current are transferred.

5.1.16 Current Demand Response

After receiving the Current Demand Request of the EV the EVSE sends the Current Demand Response informing the EV about EVSE status and present EVSE output voltage and current.

5.1.17 Welding Detection Request

The EV sends the Welding Detection Request to obtain from the EVSE the voltage value measured by the EVSE at its output.

5.1.18 Welding Detection Response

After receiving the Welding Detection Request of the EV, the EVSE sends the Welding Detection Response informing the EV about the EVSE status and the present EVSE output voltage.

5.1.19 Session Stop Request

By sending the Session Stop Request the EV requests termination of the charging process.

5.1.20 Session Stop Response

After receiving the Session Stop Request of the EVCC the SECC sends the Session Stop Response informing the EV if terminating the charging process was successful.

5.2 Message List Table

The Message ID in the table shown below directly aligns with the Request/Response messages identified in the sequence diagrams in Figures A1 – A3. Refer to XML Schema in Appendix D

Msg ID	XML Message Name	Corresponding XML Signals *			
0a	supportedAppProtocolReq	**AppProtocol**			
			ProtocolNamespace		
			VersionNumberMajor		
			VersionNumberMinor		
			SchemaID		
			Priority		
0b	supportedAppProtocolRes	ResponseCode			
		SchemaID			
1a	SessionSetupReq	EVCCID			
1b	SessionSetupRes	SessionSetupRes			
		ResponseCode			
		EVSEID			
		DateTimeNow			
2a	ServiceDiscoveryReq	ServiceScope			
		ServiceCategory			
2b	ServiceDiscoveryRes	ResponseCode			
		PaymentOptions			
			PaymentOption		
		ChargeService			
			ServiceTag		
				ServiceID	
				ServiceName	
				ServiceCategory	
				ServiceScope	
			FreeService		
			EnergyTransferType		
		ServiceList			
			Service		
				ServiceTag	
					ServiceID
					ServiceName
					ServiceCategory
					ServiceScope
				FreeService	

Msg ID	XML Message Name	Corresponding XML Signals *
Xa	ServicePaymentSelectionReq	ServicePaymentSelectionReqType SelectedPaymentOption Contract ExternalPayment SelectedServiceList ServiceID ParameterSetID
Xb	ServicePaymentSelectionRes	ResponseCode
3a	ChargeParameterDiscoveryReq	EVRequestedEnergyTransferType **DC_EVChargeParameter** **DC_EVStatus** EVReady EVCabinConditioning EVRESSConditioning EVErrorCode EVRESSSOC EVMaximumCurrentLimit EVMaximumPowerLimit EVMaximumVoltageLimit EVEnergyCapacity EVEnergyRequest FullSOC BulkSOC
3b	ChargeParameterDiscoveryRes	ResponseCode **SAScheduleList** **SAScheduleTuple** SAScheduleTupleID **PMaxSchedule** PMaxScheduleID PMaxScheduleEntry **RelativeTimeInterval** start duration PMax **SalesTariff** SalesTariffID SalesTariffDescription NumEPriceLevels

Msg ID	XML Message Name	Corresponding XML Signals *						
					SalesTariffEntry			
						RelativeTimeInterval		
							start	
							duration	
						EPriceLevel		
						ConsumptionCost		
							startValue	
							Cost	
								costKind
								amount
								amountMultiplier
		DC_EVSEChargeParameter						
			DC_EVSEStatus					
				EVSEIsolationStatus				
				EVSEStatusCode				
			EVSEMaximumCurrentLimit					
			EVSEMaximumPowerLimit					
			EVSEMaximumVoltageLimit					
			EVSEMinimumCurrentLimit					
			EVSEMinimumVoltageLimit					
			EVSECurrentRegulationTolerance					
			EVSEPeakCurrentRipple					
			EVSEEnergyToBeDelivered					
4a	CableCheckReq	**DC_EVStatus**						
			EVReady					
			EVCabinConditioning					
			EVRESSConditioning					
			EVErrorCode					
			EVRESSSOC					
4b	CableCheckRes	ResponseCode						
		DC_EVSEStatus						
			EVSEIsolationStatus					
			EVSEStatusCode					
5a	Pre-chargeReq	**DC_EVStatus**						
			EVReady					
			EVCabinConditioning					
			EVRESSConditioning					
			EVErrorCode					
			EVRESSSOC					

Msg ID	XML Message Name	Corresponding XML Signals *
		EVTargetVoltage EVTargetCurrent
5b	Pre-chargeRes	PreChargeRes ResponseCode **DC_EVSEStatus** EVSEIsolationStatus EVSEStatusCode EVSEPresentVoltage
6a	PowerDeliveryReq	ReadyToChargeState **ChargingProfile** SAScheduleTupleID **ProfileEntry** ChargingProfileEntryStart ChargingProfileEntryMaxPower **DC_EVPowerDeliveryParameter** **DC_EVStatus** EVReady EVCabinConditioning EVRESSConditioning EVErrorCode EVRESSSOC BulkChargingComplete ChargingComplete
6b	PowerDeliveryRes	ResponseCode **DC_EVSEStatus** EVSEIsolationStatus EVSEStatusCode
7a	CurrentDemandReq	**DC_EVStatus** EVReady EVCabinConditioning EVRESSConditioning EVErrorCode EVRESSSOC EVTargetCurrent

Msg ID	XML Message Name	Corresponding XML Signals *
		EVMaximumVoltageLimit EVMaximumCurrentLimit EVMaximumPowerLimit BulkChargingComplete ChargingComplete RemainingTimeToFullSoC RemainingTimeToBulkSoC EVTargetVoltage
7b	CurrentDemandRes	ResponseCode **DC_EVSEStatus** EVSEIsolationStatus EVSEStatusCode EVSEPresentVoltage EVSEPresentCurrent EVSECurrentLimitAchieved EVSEVoltageLimitAchieved EVSEPowerLimitAchieved EVSEMaximumVoltageLimit EVSEMaximumCurrentLimit EVSEMaximumPowerLimit
8a	PowerDeliveryReq	ReadyToChargeState **ChargingProfile** SAScheduleTupleID **ProfileEntry** ChargingProfileEntryStart ChargingProfileEntryMaxPower **DC_EVPowerDeliveryParameter** **DC_EVStatus** EVReady EVCabinConditioning EVRESSConditioning EVErrorCode EVRESSSOC BulkChargingComplete ChargingComplete
8b	PowerDeliveryRes	ResponseCode **DC_EVSEStatus** EVSEIsolationStatus

Msg ID	XML Message Name	Corresponding XML Signals *	
			EVSEStatusCode
9a	WeldingDetectionReq	**DC_EVStatus**	
			EVReady
			EVCabinConditioning
			EVRESSConditioning
			EVErrorCode
			EVRESSSOC
9b	WeldingDetectionRes	ResponseCode	
		DC_EVSEStatus	
			EVSEIsolationStatus
			EVSEStatusCode
		EVSEPresentVoltage	
10a	SessionStopReq	(none)	
10b	SessionStopRes	ResponseCode	

NOTE: Signals may be embedded in signal structures identified in bold. The signals contained in a structure are indented.

5.3 Message Sequences

5.3.1 Protocol flow stages and associated messages

After connecting EV and EVSE a network connection is established. Then initialisation of this communication session (charge session) has to be done. After getting all services offered by the EVSE, charging the battery can start. At the end of the charging process power is switched off. The following list describes the sequence messages, which are used to control the flow:

Initialisation of communication session: Check protocol compatibility, establish communication session, exchange client-/server IDs. Messages for this activity:

- supportedAppProtocolRequest/Response
- Session Setup Request/Response

Service Discovery: Discover the services offered by the EVSE, agreement on billing parameters. Messages for this activity:

- Service Discovery Request/Response
- Service and Payment Selection Request/Response

Charge vehicle: Charging the EV is one possible service offered by the EVSE. It is divided into three phases:

Set up charging process: Agreement on technical charging parameters, agreement on billing parameters, lock charge cord, start charging process. Messages for this activity:

- Charge Parameter Discovery Request/Response
- Cable Check Request/Response
- PreCharge Request/Response
- Power Delivery Request/Response

Charging process: do energy transfer validation in a loop and perform a new charge profile scheduling if requested by EVSE

- Current Demand Request/Response
- Power Delivery Request/Response

Finalise charging process: Stop charging process, check welding, unlock charge cord

- Welding Check Request/Response
- Stop Session Request/Response

After finalising the charging process the charge cord can be disconnected from EV and EVSE.

5.3.2 Basic Definitions for Error Handling

The basic error handling for a Request-Response-Message Pair and a Request-Response Message Sequence is based on the Response Code included in the Response Message of the SECC. Depending on the value in the Response Code the EVCC decides if it can proceed with the standard Request-Response Message Sequence or if it has to handle an error.

In this standard, the Response Code as defined in section 7.1.3 is interpreted by the EVCC as follows:

OK:Any Value starting with “OK” or “OK_” indicates a positive response. Detailed information may be provided by OK_<additional info>. This information may be used to differentiate the reaction on the positive response.

FAIL:Any Value starting with “FAIL” or “FAIL_” indicates a negative response. Detailed information may be provided by FAIL_<additional info>. This information may be used to differentiate the reaction on the negative response.

5.3.3 Response Code Handling

The responsecode defines the error indication from the EVSE for the EV.

In general each response message can contain two types of response codes 'OK' or 'FAILED'.

5.3.3.1 A response message shall contain the ResponseCode 'OK' in the 'ResponseCode' attribute if the processing of the request message was successful. If later on a specific positive 'ResponseCode' is defined for a dedicated situation, this ResponseCode shall be used.

5.3.3.2 A response message shall contain the ResponseCode 'FAILED' in the 'ResponseCode' attribute if the processing of the request message was not successful and no specific 'ResponseCodeType' is defined for the concrete error case.

5.3.3.3 The response message shall contain the ResponseCode 'FAILED_SequenceError' if the SECC has received an unexpected request message.

5.3.3.4 The response message shall contain the ResponseCode 'FAILED_UnknownSession' if the SessionID in the request message does not fit to the SECC provided SessionID during SessionSetupRes.

5.3.3.5 The response message shall contain the ResponseCode 'FAILED_SignatureError' if the validation of the Security element in the message header failed.

5.3.3.6 The message 'SessionSetupRes' shall contain the specific ResponseCode 'OK_NewSessionEstablished' if processing of the SessionSetupReq message was successful and a different SessionID is contained in the response message than the SessionID in the request message.

5.3.3.7 The message 'SessionSetupRes' shall contain the specific ResponseCode 'OK_OldSessionJoined' if processing of the SessionSetupReq message was successful and the same SessionID as used in the request message is contained in the response message.

5.3.3.8 The message 'ServicePaymentSelectionRes' shall contain the ResponseCode 'FAILED_PaymentSelectionInvalid' if the SelectedPaymentOption contained in the ServicePaymentSelectionReq message was not part of the offered PaymentOptions of ServiceDiscoveryRes.

5.3.3.9 The message 'ServicePaymentSelectionRes' shall contain the ResponseCode 'FAILED_ServiceSelectionInvalid' if the SelectedServiceList contained in the ServicePaymentSelectionReq message contains a ServiceID which was not contained in the offered ServiceList of ServiceDiscoveryRes.

5.3.3.10 The message 'ChargeParameterDiscoveryRes' shall contain the ResponseCode 'FAILED_WrongEnergyTransferType' if the content of attribute 'EVRequestedEnergyTransferType' in the ChargeParameterDiscoveryReq message is not valid, or does not fit to the content of attribute EVChargeParameter.

5.3.3.11 The message 'ChargeParameterDiscoveryRes' shall contain the ResponseCode 'FAILED_WrongChargeParameter' if the content of attribute ' EVChargeParameter' in the ChargeParameterDiscoveryReq message is not valid, e.g., wrong parameter set is provided, one or multiple parameters can not be interpreted, etc.

5.3.3.12 The message 'PowerDeliveryRes' shall contain the ResponseCode 'FAILED_ChargingProfileInvalid' if the content of attribute 'ChargingProfile' in the PowerDeliveryReq message violates a power limitation provided in 'ChargeParameterDiscoveryRes'.

5.3.3.13 The message 'PowerDeliveryRes' shall contain the ResponseCode 'FAILED_TariffSelectionInvalid' if the content of attribute 'ChargingProfile' in the PowerDeliveryReq message contains a SAtupleID which was not contained in the 'SASchedules' attribute provided in 'ChargeParameterDiscoveryRes'.

5.3.3.14 The message 'PowerDeliveryRes' shall contain the ResponseCode 'FAILED_PowerDeliveryNotApplied' if the EVSE is not able to deliver energy.

NOTE: Response codes that are not defined in this subsection can be used implementation specific.

5.3.4 Request-Response Message Sequence Requirements

5.3.4.1 EVCC Message Sequence Requirements
The EVCC behavior defining all valid Request-Response Message Sequences is shown in Figure A1.

5.3.4.2 The EVCC shall stop the V2G Communication Session whenever it receives a response message that does not correspond to the last request message sent.

NOTE: This means for example that the EVCC shall only accept a SessionSetupRes if the message sent before was a SessionSetupReq message.

5.3.4.3 The EVCC shall send a supportedAppProtocolReq as the first message of a V2G Communication Session.

5.3.4.4 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout of 'supportedAppProtocolRes' according to Table 2.

5.3.4.5 After receiving the supportedAppProtocolRes, the EVCC shall send a SessionSetupReq while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time.

5.3.4.6 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' of 'SessionSetupRes' according to Table 2.

5.3.4.7 After receiving the SessionSetupRes, the EVCC shall send a ServiceDiscoveryReq while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time.

5.3.4.8 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' of ' ServiceDiscoveryRes' according to Table 2.

5.3.4.9 After receiving the ServiceDiscoveryRes, the EVCC shall send a ServicePaymentSelectionReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time, since no service details are necessary for the remaining process.

5.3.4.10 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' of 'ServicePaymentSelectionRes' according to Table 2.

5.3.4.11 After receiving the ServicePaymentSelectionRes, the EVCC shall send a ChargeParameterDiscoveryReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time, since no service details are necessary for the remaining process.

5.3.4.12 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' of 'ChargeParameterDiscoveryRes' according to Table 2.

5.3.4.13 After receiving the ChargeParameterDiscoveryRes, the EVCC shall send a CableCheckReq while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time.

5.3.4.14 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' or 'IsolationsStatus = Fault' of 'CableCheckRes' according to Table 2.

5.3.4.15 After receiving the CableCheckRes, the EVCC shall send a PreChargeReq while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time and the parameter EVSEIsolationStatus is equal to 'Valid' or 'Warning'.when the isolation status is safe.

5.3.4.16 After receiving the CableCheckRes, the EVCC shall send a CableCheckReq while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time as long as the parameter EVSEIsolationStatus is equal to 'Invalid'.the isolation status is invalid.

5.3.4.17 The SECC shall measure state C or D as defined in IEC 61851.1(IO-SET_CPSTATE.indication (CPState=C or D)) before receiving a Cable Check Request for sending a Cable Check Response Message with parameter ResponseCode set to 'OK' otherwise it shall send ResponseCode set to 'FAILED'.

5.3.4.18 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' of 'PreChargeReq' according to Table 2.

5.3.4.19 After receiving the PreChargeRes, the EVCC shall send a PowerDeliveryReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time when the vehicle internal sensor voltage measurements match the EV RESS pack voltage.

5.3.4.20 After receiving the PreChargeRes, the EVCC shall send a PreChargeReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time until the vehicle internal sensor voltage measurements match the EV RESS pack voltage.

NOTE: See IEC61851-23 for Pre Charge details

5.3.4.21 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' of 'PowerDeliveryRes' according to Table 2.

5.3.4.22 After receiving the PowerDeliveryRes, the EVCC shall send a CurrentDemandReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time.

5.3.4.23 After receiving the CurrentDemandRes, the EVCC shall send a CurrentDemandReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time, if the charge process is continued.

5.3.4.24 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' of 'CurrentDemandRes' according to Table 2.

5.3.4.25 After receiving the CurrentDemandRes, the EVCC shall send a PowerDeliveryReq with parameter ‘ReadyToChargeState = FALSE’, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time, if the charge process shall be stopped.

5.3.4.26 After receiving the PowerDeliveryRes, the EVCC may send a WeldingDetectionReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time, if a PowerDeliveryReq with parameter ‘ReadyToChargeState = FALSE’ was sent.

5.3.4.27 After receiving the PowerDeliveryRes, the EVCC shall send a SessionStopReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time, if a PowerDeliveryReq with parameter ‘ReadyToChargeState = FALSE’ was sent and no welding detection shall be performed.

5.3.4.28 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' of 'WeldingDetectionRes' according to Table 2.

5.3.4.29 After receiving the WeldingDetectionRes, the EVCC shall send a SessionStopReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time and Welding Detection function has finished on EV side.

5.3.4.30 After receiving the WeldingDetectionRes, the EVCC shall send a WeldingDetectionReq, while V2G_EVCC_Sequence_Timer is smaller than V2G_EVCC_Sequence_Performance_Time and Welding Detection function has not finished on EV side.

NOTE: See IEC61851-23 for welding detection details.

5.3.4.31 The EVCC shall stop the V2G Communication Session when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout or 'ResponseCode = FAIL' of 'SessionStopRes' according to Table 2.

5.3.4.32 After receiving the SessionStopRes, the EVCC shall terminate the V2G Communication.

5.3.4.33 SECC Message Sequence Requirements

The SECC behavior stating all valid Request-Response message sequences is shown in Figure A2.

5.3.4.34 The SECC shall enter a wait state for supportedAppProtocolReq, set the timeout V2G_SECC_Sequence_Timeout to the value MessageType as defined in Table 2 reset the V2G_SECC_Sequence_Timer and start monitoring the V2G_SECC_Sequence_Timer.

NOTE: Before the first message the SECC did not send any response message. Therefore the SECC has to start its Sequence Timer when starting to wait for the first message.

5.3.4.35 The SECC shall stop the V2G Communication Session when V2G_SECC_Sequence_Timer is equal or larger than V2G_SECC_Sequence_Timeout according to Table 2.

5.3.4.36 The SECC shall respond with the corresponding response message containing a “ResponseCode = FAILED_SequenceError' within V2G_SECC_Msg_Performance_Time according to Table 2 if a request message was received which the SECC does not expect in the respective wait state.

5.3.4.37 After having received an unexpected request message, the SECC shall send the corresponding response message for the request message and shall stop the V2G Communication Session with an error.

5.3.4.38 After receiving a supportedAppProtocolReq, the SECC shall process the received information.

5.3.4.39 The SECC shall respond with a supportedAppProtocolRes within V2G_SECC_Msg_Performance_Time according to Table 2. The wait state for SessionSetupReq shall be entered and the V2G_SECC_Sequence_Timeout shall beis set according to Table 2.

5.3.4.40 After receiving a SessionSetupReq, the SECC shall process the received information.

5.3.4.41 The SECC shall respond with a SessionSetupRes containing “ResponseCode = OK' within V2G_SECC_Msg_Performance_Time according to Table 81. The allowed next request shall be ServiceDiscoveryReq and the V2G_SECC_Sequence_Timeout shall beis set according to Table 81.

5.3.4.42 After receiving a ServiceDiscoveryReq, the SECC shall process the received information.

5.3.4.43 The SECC shall respond with a ServiceDiscoveryRes containing “ResponseCode = OK' within V2G_SECC_Msg_Performance_Time according to Table 81, if the processing of the information is passed successfully. The allowed next request shall be ServiceAndPaymentSelectionReq and the V2G_SECC_Sequence_Timeout shall beis set according to Table 2.

5.3.4.44 The SECC shall respond with ServiceDiscoveryRes containing “ResponseCode=FAIL” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is not successful.

5.3.4.45 After receiving a ServicePaymentSelectionReq, the SECC shall process the received information.

5.3.4.46 The SECC shall respond with ServicePaymentSelectionRes containing “ResponseCode = OK' within V2G_SECC_Msg_Performance_Time according to Table 81, if the processing of the information is successfully passed. The allowed next request shall be ChargeParameterDiscoveryReq and the V2G_SECC_Sequence_Timeout shall beis set according to Table 2.

5.3.4.47 The SECC shall respond with ServicePaymentSelectionRes containing “ResponseCode = FAIL” within V2G_SECC_Msg_Performance_Time according to Table 81, if the processing of the information is not successful

5.3.4.48 After receiving a ChargeParameterDiscoveryReq, the SECC shall process the received information.

5.3.4.49 The SECC shall respond with ChargeParameterDiscoveryRes containing “ResponseCode = OK” within V2G_SECC_Msg_Performance_Time according to Table 81, if the processing of the information is successfully passed. The allowed next request shall be CableCheckReq and the V2G_SECC_Sequence_Timeout shall beis set according to Table 2.

5.3.4.50 The SECC shall respond with ChargeParameterDiscoveryRes containing “ResponseCode = FAIL” within V2G_SECC_Msg_Performance_Time according to Table 81, if the processing of the information is not successful.

5.3.4.51 After receiving a CableCheckReq, the SECC shall process the received information.

5.3.4.52 The SECC shall respond with CableCheckRes containing “ResponseCode = OK” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is successfully passed. The allowed next request shall be PreChargeReq if the parameter EVSEIsolationStatus is set to "Valid" or "Warning" or "Fault" and the V2G_SECC_Sequence_Timeout shall beis set according toThe allowed next requests shall be PrechargeReq and CableCheckReq and the V2G_SECC_Sequence_Timeout is set according to Table 2.

5.3.4.53 The SECC shall respond with CableCheckRes containing “ResponseCode = FAIL” within V2G_SECC_Msg_Performance_Time according to Table 81, if the processing of the information is not successful.

5.3.4.54 After receiving a PrechargeReq, the SECC shall process the received information and start the V2G_SECC_Msg_Performance_Timer.

5.3.4.55 The SECC shall respond with PreChargeRes containing “ResponseCode = OK” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is successfully passed. The allowed next requests shall be PrechargeReq and PowerDeliveryReq and the V2G_SECC_Sequence_Timeout shall beis set according to Table 81.

5.3.4.56 The SECC shall respond with PreChargeRes containing “ResponseCode = FAIL” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is not successful.

5.3.4.57 After receiving a PowerDeliveryReq, the SECC shall process the received information and start the V2G_SECC_Msg_Performance_Timer.

5.3.4.58 The SECC shall respond with PowerDeliveryRes containing “ResponseCode = OK” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is successfully passed and the request contained “ReadyToChargeState = TRUE”. The allowed next request shall be CurrentDemandReq and the V2G_SECC_Sequence_Timeout shall beis set according to Table 2.

5.3.4.59 The SECC shall respond with PowerDeliveryRes containing “ResponseCode = FAIL” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is not successful.

5.3.4.60 After receiving a CurrentDemandReq, the SECC shall process the received information and start the V2G_SECC_Msg_Performance_Timer.

5.3.4.61 The SECC shall respond with CurrentDemandRes containing “ResponseCode = OK” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is successfully passed. The allowed next requests shall be CurrentDemandReq and PowerDeliveryReq and the V2G_SECC_Sequence_Timeout shall beis set according to Table 81.

5.3.4.62 The SECC shall respond with CurrentDemandRes containing “ResponseCode = FAIL” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is not successful.

5.3.4.63 After Receiving a PowerDeliveryReq containing “ReadyToChargeState = FALSE”, the SECC shall process the received information and start the V2G_SECC_Msg_Performance_Timer.

5.3.4.64 The SECC shall respond with PowerDeliveryRes containing “ResponseCode = OK” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is successfully passed and the request contained “ReadyToChargeState = FALSE”. The allowed next requests shall be WeldingDetectionReq and SessionStopReq and the V2G_SECC_Sequence_Timeout shall beis set according to Table 81.

5.3.4.65 After receiving a WeldingDetectionReq, the SECC shall process the received information and start the V2G_SECC_Msg_Performance_Timer.

5.3.4.66 The SECC shall respond with WeldingDetectionRes containing “ResponseCode = OK” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is successfully passed. The allowed next requests shall be WeldingDetectionReq and SessionStopReq and the V2G_SECC_Sequence_Timeout shall beis set according to Table 81.

5.3.4.67 The SECC shall respond with WeldingDetectionyRes containing “ResponseCode = FAIL” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is not successful.

5.3.4.68 After receiving a SessionStopReq, the SECC shall process the received information and start the V2G_SECC_Msg_Performance_Timer.

5.3.4.69 The SECC shall respond with SessionStopRes containing “ResponseCode = OK” within V2G_SECC_Msg_Performance_Time according to Table 2, if the processing of the information is successfully passed. The V2G Communication Session is then stopped without error.

5.3.4.70 The SECC shall respond with SessionStopRes containing “ResponseCode = FAIL” within V2G_SECC_Msg_Performance_Time according to Table 81, if the processing of the information is not successful.

5.4 Request Message Parameters (required/optional) Modes

	SupportedAppProtocolReq	SessionSetupReq	ServiceDiscoveryReq	ServicePaymentSelectionReq	ChargeParameterDiscoveryReq	CableCheckReq	Pre-chargeReq	PowerDeliveryReq	CurrentDemandReq	WeldingDetectionReq	SessionStopReq
Charging Complete				O					R		
Ready to Charge State				O				R			
Vehicle Ready				O	R	R	R		R	R	
PEV Target Voltage							R		R		
Bulk Charge Complete									O		
Vehicle RESS SOC					R	R	R		R	R	
Remaining Charge Time to bulk SOC									O		
Remaining Charge Time to Full SOC									O		
Charge Current Request							R		R		
Vehicle Maximum Current Limit					R				O		
Vehicle Maximum Voltage Limit					R				O		
Vehicle Requested Energy Transfer Type					R						
Vehicle Maximum Power Limit					O				O		
Vehicle Energy Capacity					O						
Bulk SOC					O						
Full SOC					O						
PEV Cabin Conditioning					O	O	O		O	O	
PEV RESS Conditioning					O	O	O		O	O	
Vehicle Error Code					R	R	R		R	R	
PEV ID		R									
Priority	R										
Protocol Namespace	R										
Schema ID	R										
Service Scope			O	R							
Version Number Major	R										
Version Number Minor	R										
Selected Payment Option				R							
Vehicle Energy Request					O						

NOTES:

- The above tables show what signals are used in particular message structures.
- The signal is noted as optional ('O') or required ('R') to be transmitted within specific messages as show in the table.
- In some cases, a signal is required to send, but the receiver does not have to do anything to handle that signal
- Optional signals do not need to be transmitted, and the receiver should not issue a fault in the absence of optional signals.

5.5 Response Message Parameters (required/optional) Modes

	supportedAppProtocolRes	SessionSetupRes	ServiceDiscoveryRes	ServicePaymentSelectionRes	ChargeParameterDiscoveryRes	CableCheckRes	Pre-chargeRes	PowerDeliveryRes	CurrentDemandRes	WeldingDetectionRes	SessionStopRes
Charger Status Code						R	R	R	R	R	
Charge Station Isolation Status						R	R	R	R	R	
Charger Current Limit Achieved									R		
Charger Voltage Limit Achieved									R		
Charger Power Limit Achieved									R		
Voltage output							R		R	R	
Current output									R		
Isolation Monitoring Reading											
Charger Maximum Current Limit					R				O		
Charger Maximum Voltage Limit					R				O		
Charger Minimum Current Limit					R						
Charger Minimum Voltage Limit					R						
Charger Supported Energy Transfer Type			R								
Charger Peak Current Ripple					R						
Charger Maximum Power Limit					O				O		
Charger Energy to be Delivered					O						
Charger Current Regulation Tolerance					O						
EVSE ID		R									
Date Time Now		O									
Payment Options			R								
Schema ID	O										
Response Code	R		R	R	R	R	R	R	R	R	R
Session Setup Response Code		R									

NOTES:

- The above tables show what signals are used in particular message structures.
- The signal is noted as optional ('O') or required ('R') to be transmitted within specific messages as show in the table.
- In some cases, a signal is required to send, but the receiver does not have to do anything to handle that signal
- Optional signals do not need to be transmitted, and the receiver should not issue a fault in the absence of optional signals.

5.6 Session Timing and Error handling

5.6.1 Overview

This sub-section describes the timing and error handling for the V2G Communication Session. The error handling is based on timers enabling the EVCC and the SECC to monitor the Charging message exchange. For the detection of missing or delayed messages the EVCC and the SECC use predefined timeout values as error criteria. Whenever a timer is equal to or larger than the related timeout the related error handling is processed.

A timer counts the duration from the last time it was reset. The value of a timer is the duration from the last reset to the present time. The monitoring of a V2G Communication message is based on two timer categories:

- Message Timer: Monitors the exchange of a request message and the corresponding response message (Request-Response-Pair);

- Sequence Timer: Monitors a Response-Request Message Sequence.

To enable error handling for a V2G Communication Session setup the EVCC monitors the time between plug-in and the reception of the Session Setup Response and the Power Delivery Response, respectively. This allows the EVCC to decide about a successful or failed charging session by using the defined timeouts.

The monitoring of a V2G Communication Session is based on two timer categories:

- Communication Setup Timer: Monitors the time from plug-in until the Session Setup message. It allows deciding if the communication setup was successful;

- Ready to Charge Timer: Monitors the time from plug-in until the first Power Delivery message. It allows deciding if the request for power from the SECC was successful.

The timers are compared to predefined time values as decision criteria. The EVCC and the SECC decide between two categories:

Timeout: If the specified time is exceeded the related error handling is initiated;

- Performance time: If the specified time is exceeded the performance requirement is not fulfilled.

NOTE: While exceeding a timeout always causes an error handling, the performance time does not necessarily cause error handling. Depending on the system behavior (e.g., transmission time) no error may occur if the corresponding communication partner does not detect a timeout but the probability for causing a timeout is high.

5.6.2 Message sequence and performance timing definitions

Message Timers, Sequence Timers, Timeouts, and Performance Times are defined for the EVCC and SECC separately and summarized in Table 1. Timeouts and Performance Times are parameterized for messages separately to describe different processing times.

Table 2 defines the values for each Charging message type.

TABLE 1 - EVCC AND SECC TIMERS, TIMEOUTS, PERFORMANCE TIMES

Name	**Type**	**Applicable for**	
		EVCC	**SECC**
V2G_EVCC_Msg_Timer	Message Timer in the EVCC	x	
V2G_SECC_Msg_Timer	Message Timer in the SECC		x
V2G_EVCC_Msg_Timeout(MessageType)	Timeout for the Message Timer The value is defined by the parameter MessageType as defined in Table .	x	
V2G_SECC_Msg_Performance_Time(MessageType)	Performance Time for the Message Timer The value is defined by the parameter MessageType as defined in Table .		x
V2G_EVCC_Sequence_Performance_Time	Performance Time for the Sequence Timer as defined in Table .	x	
V2G_SECC_Sequence_Timeout	Timeout for the Sequence Timer as defined in Table .		x

TABLE 2 - EVCC AND SECC MESSAGE TIMEOUTS AND PERFORMANCE TIMES

Name	**MessageType**	**Value [s]**
V2G_EVCC_Msg_Timeout(MessageType)	SupportedAppProtocol	2
	SessionSetup	2
	ServiceDiscovery	2
	ServicePaymentSelection	2
	ChargeParameterDiscovery	2
	PowerDelivery	2
	CableCheck	2
	Pre-charge	2
	CurrentDemand	0.25
	WeldingDetection	2
	SessionStop	2
V2G_SECC_Msg_Performance_Time(MessageType)	SupportedAppProtocol	1.5
	SessionSetup	1.5
	ServiceDiscovery	1.5
	ServicePaymentSelection	1.5
	ChargeParameterDiscovery	1.5
	PowerDelivery	1.5
	CableCheck	1.5
	Pre-charge	1.5
	CurrentDemand	0.025
	WeldingDetection	1.5
	SessionStop	1.5
V2G_EVCC_Sequence_Performance_Time	(any messages)	56
V2G_SECC_Sequence_Timeout	(any messages)	60

Figure 1 illustrates how the message timers, sequence timers, timeouts and performance times are applied in the EVCC and the SECC.

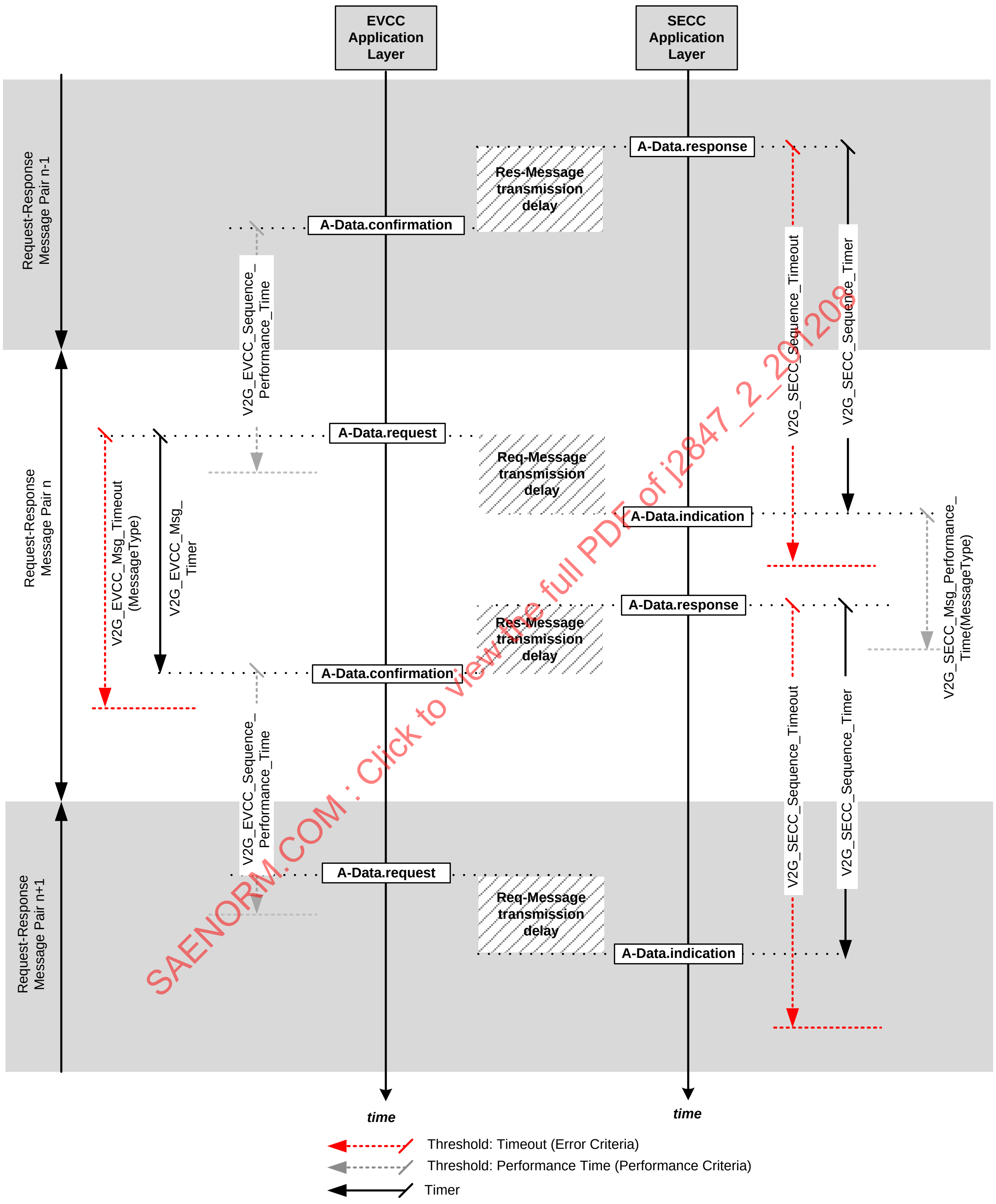

EVCC Application Layer
SECC Application Layer
Request-Response Message Pair n-1
Request-Response Message Pair n
Request-Response Message Pair n+1
A-Data.response
Res-Message transmission delay
A-Data.confirmation
V2G_EVCC_Sequence_Performance_Time
V2G_SECC_Sequence_Timeout
V2G_SECC_Sequence_Timer
A-Data.request
Req-Message transmission delay
V2G_EVCC_Msg_Timeout (MessageType)
V2G_EVCC_Msg_Timer
A-Data.indication
A-Data.response
V2G_SECC_Msg_Performance_Time(MessageType)
Res-Message transmission delay
A-Data.confirmation
V2G_EVCC_Sequence_Performance_Time
V2G_SECC_Sequence_Timeout
V2G_SECC_Sequence_Timer
A-Data.request
Req-Message transmission delay
A-Data.indication
time
time
Threshold: Timeout (Error Criteria)
Threshold: Performance Time (Performance Criteria)
Timer

FIGURE 1 - MESSAGE SEQUENCE AND PERFORMANCE TIMING

The EVCC shall implement the EVCC specific timeouts and performance times defined in Table 1 and Table 2.

The SECC shall implement the SECC specific timeouts and adhere to performance times defined in Table 1 and Table 2.

5.6.3 EVCC Timing and Error handling for Request-Response Message Pairs:

5.6.3.1 The EVCC shall set the timeout V2G_EVCC_Msg_Timeout to the value MessageType as defined in Table 2, reset the V2G_EVCC_Msg_Timer and start monitoring the V2G_EVCC_Msg_Timer when it sends a request message.

NOTE: In this document sending a request message is described by A-DATA.request.

5.6.3.2 The EVCC shall wait for the response message corresponding to the request message sent before transmitting another request.

5.6.3.3 The EVCC shall stop waiting for the response message and stop monitoring the V2G_EVCC_Msg_Timer when V2G_EVCC_Msg_Timer is equal or larger than V2G_EVCC_Msg_Timeout(MessageType) and no response message was received. It shall then apply the error handling as defined in section 5.4.

NOTE: In this document receiving a response message is described by A-DATA.confirmation.

5.6.3.4 The EVCC shall stop waiting for the response message and stop monitoring the V2G_EVCC_Msg_Timer when V2G_EVCC_Msg_Timer is smaller than V2G_EVCC_Msg_Timeout(MessageType) and it received a response message. It shall then process the response message as defined in 6.4.

NOTE: In this document receiving a response message is described by A-DATA.confirmation.

5.6.3.5 The EVCC shall ignore any message that is not a valid response message.

5.6.4 SECC Timing and Error handling for Response-Request Message Sequence:

5.6.4.1 The SECC shall set the timeout V2G_SECC_Sequence_Timeout to the value MessageType as defined in Table 2, reset the V2G_SECC_Sequence_Timer and start monitoring the V2G_SECC_Sequence_Timer when it sends a response message.

NOTE: In this document receiving a response message is described by A-DATA.confirmation.

5.6.4.2 The SECC shall wait for a request message.

5.6.4.3 The SECC shall stop waiting for a request message and stop monitoring the V2G_SECC_Sequence_Timer when V2G_SECC_Sequence_Timer is equal or larger than V2G_SECC_Sequence_Timeout and no request message was received. It shall then apply the error handling as defined in section 6.4.

NOTE: In this document receiving a response message is described by A-DATA.confirmation.

5.6.4.4 The SECC shall stop waiting for a request message and stop monitoring the V2G_SECC_Sequence_Timer when V2G_SECC_Sequence_Timer is smaller than V2G_SECC_Sequence_Timeout and it received a request message. It shall then process the request message as defined in section 6.4.

NOTE: In this document receiving a response message is described by A-DATA.confirmation.

5.6.4.5 The SECC shall ignore any message that is not a valid request message.

5.6.5 V2G Communication Session Timing Definitions

Table 3 shows timing parameters applicable to several phases of the communication session as defined in this standard. Table 4 defines the values for the related Performance Times and the Timeouts.

5.6.5.1 EVCC Timing and Error handling for communication session setup timeout:

5.6.5.1.1 The EVCC shall set the timeout V2G_EVCC_CommunicationSetup_Timeout to the value as defined in Table 4, reset the V2G_EVCC_CommunicationSetup_Timer and start monitoring the V2G_EVCC_CommunicationSetup_Timer when state B is indicated. EVCC shall send the Session Setup Message request

NOTE: In this document state B is indicated by Control Pilot=State B.

5.6.5.1.2 The EVCC shall wait for the Session Setup Response message.

5.6.5.1.3 The EVCC shall stop waiting for the Session Setup Response message and stop monitoring the V2G_EVCC_CommunicationSetup_Timer when V2G_EVCC_CommuncationSetup_Timer is equal or larger than V2G_EVCC_CommunicationSetup_Timeout and no Session Setup Response message was received. It shall then apply the error handling as defined in section 6.4.

NOTE: In this document receiving the response message “Session Setup Response” is described by A-DATA.confirmation(SessionSetupRes).

5.6.5.1.4 The EVCC shall stop waiting for the Session Setup Response message and stop monitoring the V2G_EVCC_CommunicationSetup_Timer when V2G_EVCC_CommuncationSetup_Timer is small than V2G_EVCC_CommunicationSetup_Timeout and a Session Setup Response message was received. It shall then process the response message as defined in 6.4.

NOTE: In this document receiving the response message “Session Setup Response” is described by A-DATA.confirmation(SessionSetupRes).

5.6.5.2 EVCC Timing and Error handling for ready to charge timeout:

5.6.5.2.1 The EVCC shall set the timeout V2G_EVCC_ReadyToCharge_Timeout to the value as defined in, reset the V2G_EVCC_ReadyToCharge_Timer and start monitoring the V2G_EVCC_ReadyToCharge_Timer when state B is indicated.

NOTE: In this document state B is indicated by Control Pilot=State B.

5.6.5.2.2 The EVCC shall wait for the Power Delivery Response message.

5.6.5.2.3 The EVCC shall stop waiting for the Power Delivery Response message and stop monitoring the V2G_EVCC_ReadyToCharge_Timer when V2G_EVCC_ReadyToCharge_Timer is equal or larger than V2G_EVCC_ReadyToCharge_Timeout and no Power Delivery Response message was received. It shall then apply the error handling as defined in section 6.4.

NOTE: In this document receiving the response message “Power Delivery Response” is described by A-DATA.confirmation(PowerDeliveryRes).

5.6.5.2.4 The EVCC shall stop waiting for the Power Delivery Response message and stop monitoring the V2G_EVCC_ReadyToCharge_Timer when V2G_EVCC_ReadyToCharge_Timer is smaller than V2G_EVCC_ReadyToCharge_Timeout and a Power Delivery Response message was received. It shall then process the response message as defined in section 6.4.

NOTE: In this document receiving the response message “Power Delivery Response” is described by A-DATA.confirmation(PowerDeliveryRes).

5.6.5.3 EVCC Timing and Error handling for cable check timeout:

5.6.5.3.1 The EVCC shall set the timeout V2G_EVCC_CableCheck_Timeout to the value as defined in, reset the V2G_EVCC_CableCheck_Timer and start monitoring the V2G_EVCC_CableCheck_Timer when sending message CableCheckReq for the first time in a charging session.

NOTE: In this document sending a request message is described by A-DATA.request.

5.6.5.3.2 The EVCC shall wait for the Cable Check of the EVSE to finish indicated by the reception of a Cable Check Response message with DC_EVSEStatus.EVSEIsolationStatus is not equal to Invalid.

NOTE: In this document receiving a response message is described by A-DATA.confirmation.

5.6.5.3.3 Upon reception of a Cable Check Response message with DC_EVSEStatus.EVSEIsolationStatus = Invalid, while V2G_EVCC_CableCheck_Timer is smaller than V2G_EVCC_CableCheck_Timeout, the EVCC shall send another Cable Check request.

NOTE: In this document sending a request message is described by A-DATA.request.

5.6.5.3.4 The EVCC shall stop waiting for the Cable Check of the EVSE to finish and stop monitoring the V2G_EVCC_CableCheck_Timer when V2G_EVCC_CableCheck_Timer is equal or larger than V2G_EVCC_CableCheck_Timeout. It shall then apply the error handling as defined in section 6.4.

5.6.5.3.5 The EVCC shall stop waiting for the Cable Check of the EVSE to finish and stop monitoring the V2G_EVCC_CableCheck_Timer when V2G_EVCC_CableCheck_Timer is smaller than V2G_EVCC_CableCheck_Timeout and a Cable Check Response message with DC_EVSEStatus.EVSEIsolationStatus Not Equal To Invalid was received. It shall then process the response message as defined in 6.4.

NOTE: In this document receiving a response message is described by A-DATA.confirmation.

5.6.5.4 EVCC Timing and Error handling for pre charging timeout:

5.6.5.4.1 The EVCC shall set the timeout V2G_EVCC_Pre-charge_Timeout to the value as defined in Table 4, reset the V2G_EVCC_Pre-charge_Timer and start monitoring the V2G_EVCC_Pre-charge_Timer when sending message Pre-chargeReq for the first time in a charging session.

NOTE: In this document sending a request message is described by A-DATA.request.

5.6.5.4.2 The EVCC shall wait for the Pre Charging to finish, indicated by the EVCC receiving a Pre Charge Response and EV determining that the SECC output voltage, as measured inside the EV, has sufficiently been adjusted to the EV RESS voltage..

5.6.5.4.3 Upon reception of a Pre-charge Response message while V2G_EVCC_Pre-charge_Timer is smaller than V2G_EVCC_Pre-charge_Timeout, if the EV determines that the EVSE output voltage, as measured inside the EV, has not sufficiently been adjusted to the EV RESS voltage, the EVCC shall send another Pre-charge request.

NOTE: In this document sending a request message is described by A-DATA.request.

5.6.5.4.4 The EVCC shall stop waiting for the Pre Charging to finish and stop monitoring the V2G_EVCC_Pre-charge_Timer when V2G_EVCC_Pre-charge_Timer is equal or larger than V2G_EVCC_Pre-charge_Timeout. It shall then apply the error handling as defined in 6.4.

5.6.5.4.5 EVCC shall stop monitoring the V2G_EVCC_Pre-charge_Timer when V2G_EVCC_Pre-charge_Timer is smaller than V2G_EVCC_Pre-charge_Timeout and Pre Charging has finished, indicated by the EV determining that the EVSE output voltage, as measured inside the EV, has sufficiently been adjusted to the EV RESS voltage. It shall then process the response message as defined in 6.4.

TABLE 3 - EVCC AND SECC COMMUNICATION SESSION TIMING PARAMETERS

Parameter name	**Definition**	**Implementation**	
		EVCC	**SECC**
V2G_EVCC_CommunicationSetup_Timer	Communication Setup Timer in the EVCC	x	
V2G_SECC_CommunicationSetup_Timer	Communication Setup Timer in the SECC		x
V2G_EVCC_ReadyToCharge_Timer	Ready to Charge Timer in the EVCC	x	
V2G_SECC_ReadyToCharge_Timer	Ready to Charge Timer in the SECC		x
V2G_EVCC_CableCheck_Timer	Cable Check Timer in the EVCC	x	
V2G_SECC_CableCheck_Timer	Cable Check Timer in the SECC		x
V2G_EVCC_Pre-charge_Timer	Pre-charge Timer in the EVCC	x	
V2G_SECC_Pre-charge_Timer	Pre-charge Timer in the SECC		x
V2G_EVCC_CommunicationSetup_ Timeout	Timeout for the Communication Setup Timer in the EVCC as defined in Table .	X	
V2G_SECC_CommunicationSetup_ Performance_Time	Performance Time for the Communication Setup Timer in the SECC as defined in Table .		X
V2G_EVCC_ReadyToCharge_Timeout	Timeout for the Ready to Charge Timer in the EVCC as defined in Table .	X	
V2G_SECC_ReadyToCharge_ Performance_Time	Performance Time for the Ready to Charge Timer in the SECC as defined in Table .		X
V2G_EVCC_CableCheck_Timeout	Timeout for the CableCheck Timer in the EVCC as defined in Table .	X	
V2G_SECC_CableCheck_Performance_Time	Performance Time for the CableCheck Timer in the SECC as defined in Table .		x
V2G_EVCC_Pre-charge_Timeout	Timeout for the Pre-charge Timer in the EVCC as defined in Table .	x	
V2G_SECC_ Pre-charge_Performance_Time	Performance Time for the Pre-charge Timer in the SECC as defined in Table .		x

The EVCC and SECC shall implement the timing parameter values defined in Table 4.

TABLE 4 - EVCC AND SECC COMMUNICATION SESSION TIMING PARAMETER VALUES

Parameter name	**Value [s]**	**Implementation**	
		EVCC	**SECC**
V2G_SECC_ReadyToCharge_Performance_Time	40		x
V2G_EVCC_ReadyToCharge_Timeout	45	x	
V2G_SECC_CommunicationSetup_Performance_Time	18		x
V2G_EVCC_CommunicationSetup_Timeout	20	x	
V2G_SECC_CableCheck_Performance_Time	30		x
V2G_EVCC_CableCheck_Timeout	40	x	
V2G_SECC_Pre-charge_Performance_Time	5		x
V2G_EVCC_Pre-charge_Timeout	6	x	

Figure 2 illustrates how some of the timing parameters defined in Table 3 are applied.

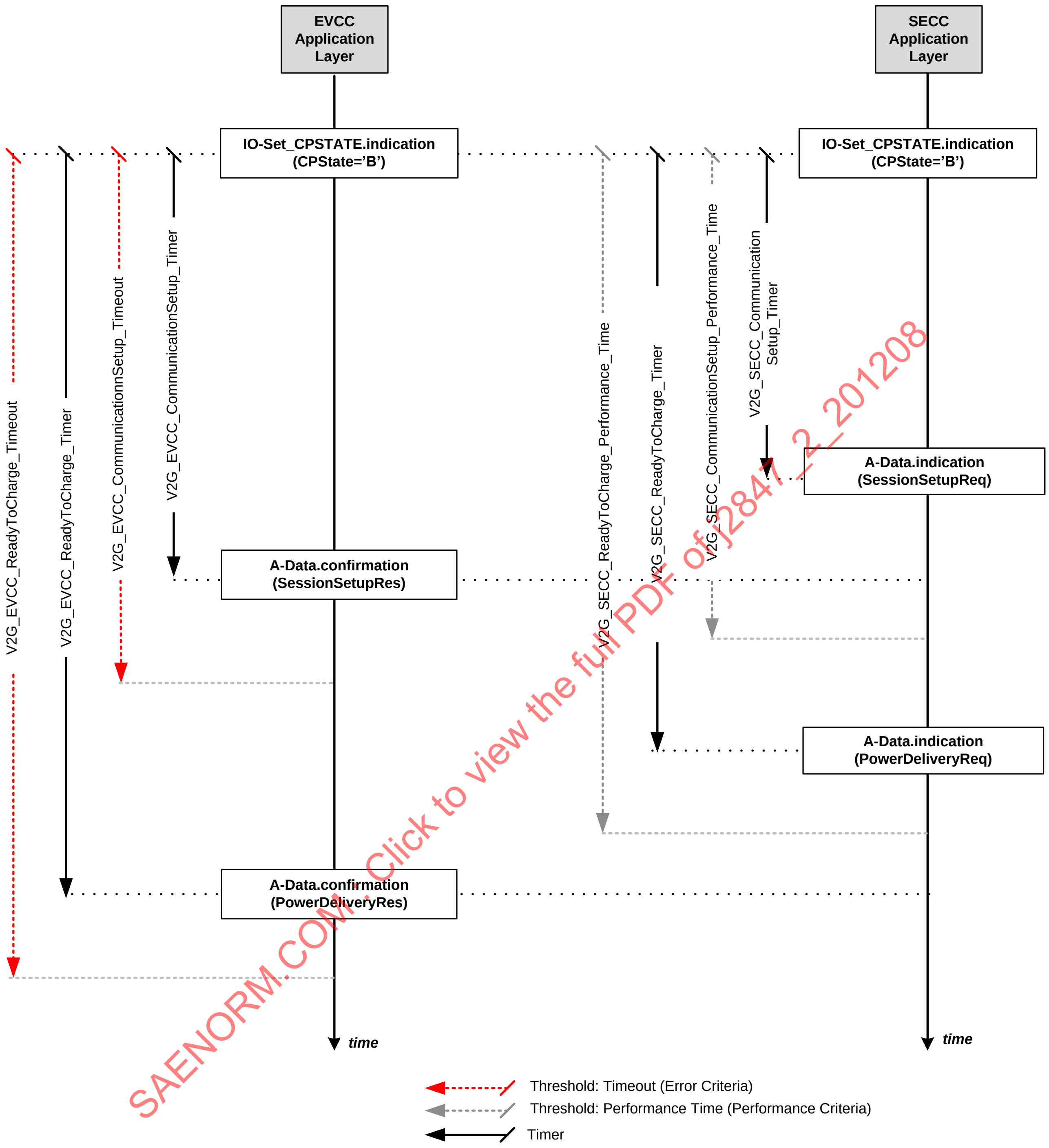

EVCC Application Layer
SECC Application Layer
IO-Set_CPSTATE.indication (CPState='B')
IO-Set_CPSTATE.indication (CPState='B')
V2G_EVCC_ReadyToCharge_Timeout
V2G_EVCC_ReadyToCharge_Timer
V2G_EVCC_CommunicationnSetup_Timeout
V2G_EVCC_CommunicationSetup_Timer
V2G_SECC_ReadyToCharge_Performance_Time
V2G_SECC_ReadyToCharge_Timer
V2G_SECC_CommunicationSetup_Performance_Time
V2G_SECC_Communication Setup_Timer
A-Data.indication (SessionSetupReq)
A-Data.confirmation (SessionSetupRes)
A-Data.indication (PowerDeliveryReq)
A-Data.confirmation (PowerDeliveryRes)
time
time
Threshold: Timeout (Error Criteria)
Threshold: Performance Time (Performance Criteria)
Timer

FIGURE 2 - V2G COMMUNICATION SESSION TIMING

6. PROTOCOL

6.1 General information and definitions

A V2G message uses the EXI-based Presentation Layer as described in subclause 1.1.1. The communication between EVCC and SECC at application layer level is based on a client/server architecture. The EVCC always acts as a client (service requester) during the entire charging process, whereas the SECC always acts as a server (service responder). Hence the EVCC always initates communication by sending a request message to the SECC which then returns the corresponding response message. All messages exchanged between EVCC and SECC are described with their syntax and their semantics in subclauses 6.2, 6.3 and 6.4. The entire XML Schema definition describing both V2G message set is included in Appendix A.

Subclause 5.6.2 defines message timing and error handling for the V2G communication message exchange.

An example for a typical message sequence is shown in Appendix A.

V2G communication consists of two different message sets:

- V2G application layer protocol handshake messages (refer to subclause 6.2);
- V2G application layer messages (refer to J2931/4).

6.2 Protocol Handshake

6.2.1 Handshake Request-Response Message Pair

6.2.1.1 Before starting the application layer message exchange, an appropriate application layer protocol including its version shall be negotiated between the EVCC and the SECC.

In order to negotiate the protocol between the EVCC and the SECC the following application layer protocol handshake is performed.

6.2.1.2 The EVCC shall initiate the handshake sending a SupportedAppProtocolReq message as depicted in Figure 3 to the SECC. This request message provides a list of charging protocols supported by the EVCC.

6.2.1.3 Each entry in the list of supported EVCC protocols shall include the ProtocolNamespace, the VersionNumberMajor and VersionNumberMinor, the SchemaID dynamically assigned by the EVCC and the Priority of the protocol entry. The Priority in the EVCC request message enables the EVCC to announce the preferred application layer protocol where Pritory equal to 1 indicates the highest priority and Priority equal to 20 indicates the lowest priority. The number of protocols included in the request message is limited to 20.

6.2.1.4 The SECC shall respond with a supportedAppProtocolRes message as depicted in Figure 4 indicating the protocol to be used for the subsequent message exchange by both the EVCC and the SECC.

6.2.1.5 The response message shall include a ResponseCode and the SchemaID of the protocol/schema which is agreed as application protocol for the following communication session. Thereby, the SECC shall select from its own list of supported protocols the protocol with highest Priority indicated by the EVCC.

6.2.1.6 The SECC shall confirm (positively respond) an EVCC supported protocol even if the values of the VersionNumberMinor in EVCC request message does not match with the VersionNumberMinor of an SECC supported protocol where the VersionNumberMajor matches.

NOTE: A higher value in the VersionNumberMinor indicates that (in comparison to a lower value) additional data elements will be transmitted from either the EVCC or SECC. Implementations only supporting the lower VersionNumberMinor value may not be able to process the data and may have to ignore this data, however a difference in the VersionNumberMinor value between EVCC and SECC does not lead to an incompatibility. Refer to section 6.4 showing examples for successful protocol negotiation.

6.2.1.7 All additional data element defined by the respective minor version shall be encoded as schema deviated case by the EXI coder (see also EXI option settings in J2931/4).

6.2.1.8 Usually it is expected that the SECC is able to support the relevant application layer protocols indicated by the EVCC. However when none of the application layer protocols included the list received from the EVCC is supported by the SECC, the ResponseCode in the response message shall be equal to Failed_NoNegotiation indicating that the protocol negotiation was not successful. In this error scenario the response message shall not include a SchemaID.

6.2.1.9 If no successful protocol negotiation can be achieved the EVCC shall not initialize a communication session.

6.2.1.10 This protocol handshake between EVCC and SECC shall be peformed prior the actual V2G application layer message exchange. Only the message set defined in the agreed protocol shall be used in the V2G message flow except for minor deviations.

6.2.1.11 In the scope of SAE 2847-2 and DIN 70121, only the namespace “urn:din:70121:2012:MsgDef” and the version number 2.0 shall be used to indicate DIN 70121.

6.2.2 Message definition SupportedAppProtocolReq and supportedAppProtocolRes

6.2.2.1 The SECC shall implement the message and message elements as in defined in Figure 3.

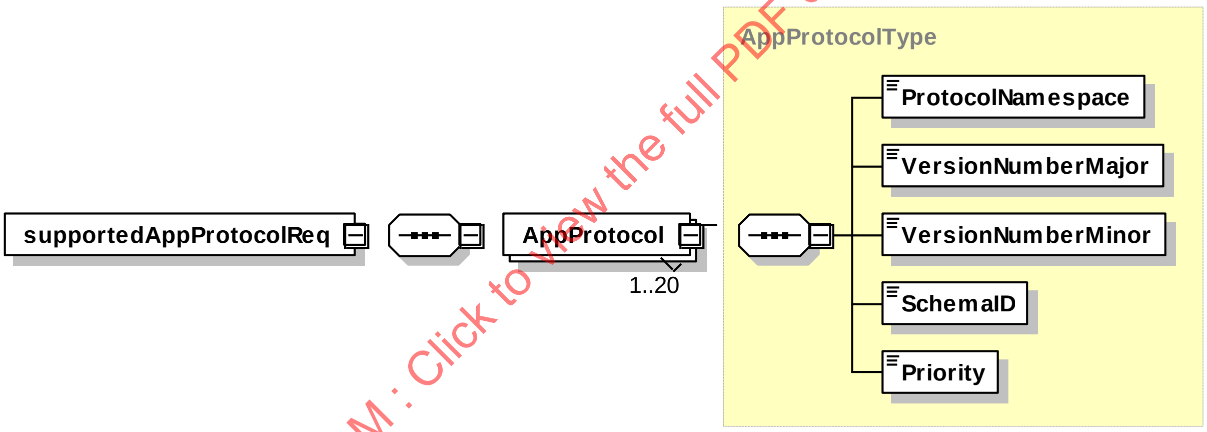


FIGURE 3 - SCHEMA DIAGRAM - SUPPORTEDAPPPROTOCOLREQ

6.2.2.2 The SECC shall implement the message and message elements as defined and Figure 4.

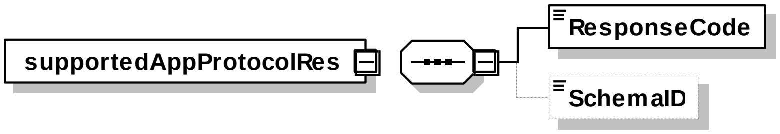


FIGURE 4 - SCHEMA DIAGRAM - SUPPORTEDAPPPROTOCOLRES

NOTE: Refer to 7.1.1 for the XML schema code.

6.2.2.3 Semantics decription supportedAppProtocol messages

6.2.2.4 The message elements defined in Figure 3 and Figure 4 of the supportedAppProtocol messages shall be used as defined in Table 5.

6.2.2.5 The message elements of the messages defined in Figure 3 and Figure 4 shall be used as defined in Table 5.

TABLE 5 - SEMANTICS AND TYPE DEFINITION FOR SUPPORTEDAPPPROTOCOL MESSAGE ELEMENTS

Element/Attribute Name	**Type**	**Semantics**
AppProtocol	complexType: includes the message elements defined in this table	This message element is used by the EVCC for transmitting the list of supported protcols. Each protocol with a particular version supported by the EVCC is represented by one AppProtocol entry in the request message (maximum number of entries: 20)
ProtocolNamespace	simpleType: protocolNamespaceType string (max length: 100) refer to 7.1.1 for the type definition	This message element is used by the EVCC to uniquely identify the Namespace URI of a specific protocol supported by the EVCC, i.e., this is the protocol name of the related protocol.
VersionNumberMajor	simpleType unsignedInt refer to 7.1.1 for the type definition	This message element is used by the EVCC to indicate the major version number of the protocol indicated in the message element ProtocolNamespace.
VersionNumberMinor	simpleType unsignedInt refer to 7.1.1 for the type definition	This message element is used by the EVCC to indicate the minor version number of the protocol indicated in the message element ProtocolNamespace.
SchemaID	simpleType: unsignedByte refer to 7.1.1 for the type definition	This message element is used by the EVCC to indicate the schemaID assigned by the EVCC to the protocol indicated in the message element ProtocolNamespace, VersionNumberMajor and VersionNumberMinor. This message element is used by the SECC to reference one of the EVCC supported protocols received in the request message. This identifier allows also for referring to a particular protocol later on in the communication process (EXI Option schemaID).
Priority	simpleType: priorityType unsignedByte (range 1..20) refer to 7.1.1 for the type definition	This message element is used by the EVCC for indicating the protocol priority of a specific protocol allowing the SECC to select a protocol based on priorities.
ResponseCode	simpleType: responseCodeType enumeration refer to 7.1.1 for the type definition	This message element is used by the SECC for indicating wether the list of protocols received from the EVCC includes at least one protocol matching with the protocols supported by the SECC. Signals a response codes with the following meaning: - OK_SuccessfulNegotiation: successful negotiation of a application protocol - OK_SuccessfulNegotiationWithMinorDeviation: successful negotiation of a protocol, with minor deviation - Failed_NoNegotiation: failed negotiation of protocols

6.3 Protocol prioritization

V2G message example 1 and V2G message example 2 illustrate the exchange of suppAppProtocol messages between the EVCC and the SECC. In the request message, the EVCC sends a prioritized list of supported application layer protocols (SAE_J2847-2 with version 1.0, din:7012115118:2:20120 with version 1.0, iso:15118:2:2010 with version 1.0) to the SECC. In the response message the SECC confirms protocol (din:7012115118:2:20120 with version 1.0) using a ResponseCode equal to 'OK_SuccessfulNegotiation' and a schemaID equal to ten (10).

In the scope of dc charging, only the namespace “SAE_J2847-2” and the major version number 1.0 and minor verion number 1.0 shall be used to indicate compatiblity with this verion of the J2847-2 document.

V2G message example 1 – supportedAppProtocolReq: protocol prioritization

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolReq xmlns:ns0="urn:iso:15118:2:2010:AppProtocol"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <AppProtocol>
        <ProtocolNamespace>urn:din:70121:2012:MsgDef</ProtocolNamespace>
        <VersionNumberMajor>1</VersionNumberMajor>
        <VersionNumberMinor>0</VersionNumberMinor>
        <SchemaID>10</SchemaID>
        <Priority>1</Priority>
    </AppProtocol>
    <AppProtocol>
        <ProtocolNamespace>urn:iso:15118:2:2010:MsgDef</ProtocolNamespace>
        <VersionNumberMajor>1</VersionNumberMajor>
        <VersionNumberMinor>0</VersionNumberMinor>
        <SchemaID>20</SchemaID>
        <Priority>5</Priority>
    </AppProtocol>
</ns0:supportedAppProtocolReq>
```

V2G message example 2 – supportedAppProtocolRes: protocol prioritization

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolRes xmlns:ns0="urn:iso:15118:2:2010:AppProtocol"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ResponseCode>OK_SuccessfulNegotiation</ResponseCode>
    <SchemaID>10</SchemaID>
</ns0:supportedAppProtocolRes>
```

V2G message example 3 – supportedAppProtocolReq: protocol prioritization

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolReq xmlns:ns0="urn:sae:SAE_J2847-2:2:2012:AppProtocol"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <AppProtocol>
        <ProtocolNamespace>urn:sae:SAE_J2847-2:2012:MsgDef</ProtocolNamespace>
        <VersionNumberMajor>1</VersionNumberMajor>
        <VersionNumberMinor>0</VersionNumberMinor>
        <SchemaID>10</SchemaID>
        <Priority>1</Priority>
    </AppProtocol>
    <AppProtocol>
        <ProtocolNamespace>urn:iso:15118:2:2010:MsgDef</ProtocolNamespace>
        <VersionNumberMajor>1</VersionNumberMajor>
        <VersionNumberMinor>0</VersionNumberMinor>
        <SchemaID>20</SchemaID>
        <Priority>5</Priority>
    </AppProtocol>
</ns0:supportedAppProtocolReq>
```

V2G message example 4 – supportedAppProtocolRes: protocol prioritization

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolRes xmlns:ns0="urn:sae:SAE_J2847-2:2:2012:AppProtocol"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ResponseCode>OK_SuccessfulNegotiation</ResponseCode>
    <SchemaID>10</SchemaID>
</ns0:supportedAppProtocolRes>
```

6.4 Minor Deviation

V2G message example 3 and V2G message example 4 illustrate the exchange of suppAppProtocol messages between the EVCC and the SECC. In the request message, the EVCC sends just one supported application layer protocol (din:70121:2012 with version 1.0) to the SECC. The SECC supports protocol version 1.1 only. In the response message the SECC confirms protocol (din:70121:2012) with VersionNumberMajor equal to one (1) using a schemaID equal to one (1). However, the ResponseCode is equal to OK_SuccessfulNegotiationWithMinorDeviation signalling that a minor version deviation applies. The EVCC may now expect message elements which aren't known by the EVCC but can be ignored.

V2G message example 3 – supportedAppProtocolReq: deviation in minor version

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolReq xmlns:ns0="urn:iso:15118:2:2010:AppProtocol"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <AppProtocol>
        <ProtocolNamespace>urn:din:70121:2012:MsgDef</ProtocolNamespace>
        <VersionNumberMajor>1</VersionNumberMajor>
        <VersionNumberMinor>0</VersionNumberMinor>
        <SchemaID>1</SchemaID>
        <Priority>1</Priority>
    </AppProtocol>
</ns0:supportedAppProtocolReq>
```

V2G message example 4 – supportedAppProtocolRes: deviation in minor version

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:supportedAppProtocolRes xmlns:ns0="urn:iso:15118:2:2010:AppProtocol"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ResponseCode>OK_SuccessfulNegotiationWithMinorDeviation</ResponseCode>
    <SchemaID>1</SchemaID>
</ns0:supportedAppProtocolRes>
```

7. NOTES

7.1 Marginal Indicia

A change bar (l) located in the left margin is for the convenience of the user in locating areas where technical revisions, not editorial changes, have been made to the previous issue of this document. An (R) symbol to the left of the document title indicates a complete revision of the document, including technical revisions. Change bars and (R) are not used in original publications, nor in documents that contain editorial changes only.

8. ACKNOWLEDGEMENT

SAE International acknowledges that original DIN SPEC 70121, with DIN permission, was used in relevant parts of this SAE J2847/2 standard

PREPARED BY SAE HYBRID - EV COMMITTEE

APPENDIX A – MESSAGE SEQUENCE EXAMPLES

A.1 MESSAGE SEQUENCE EXAMPLE DC CHARGING

This section gives an overview of the message flow between EV and EVSE during normal operation for a DC charging scenario. It does not provide an in depth view on message timing and other constraints. Figure A give a basic overview of the different communication phases and the respective message sequence. The content of all messages is described in section 6 "Messages". For reasons of clarity and readability, the individual messages are represented just by the message name rather than including the complete data structure and data types.

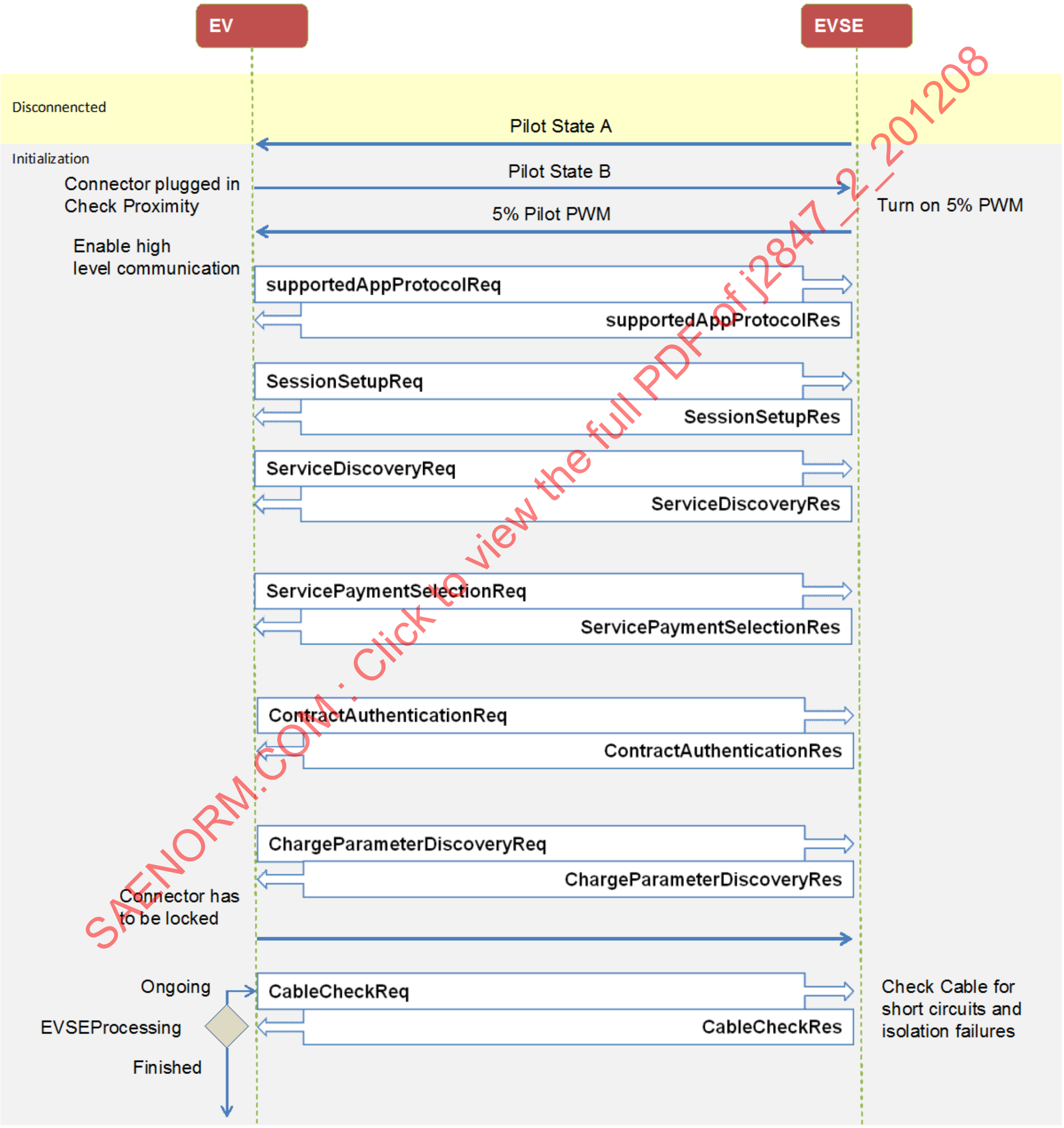


FIGURE A1 - MESSAGE EXCHANGE BETWEEN EV AND EVSE DURING STARTUP

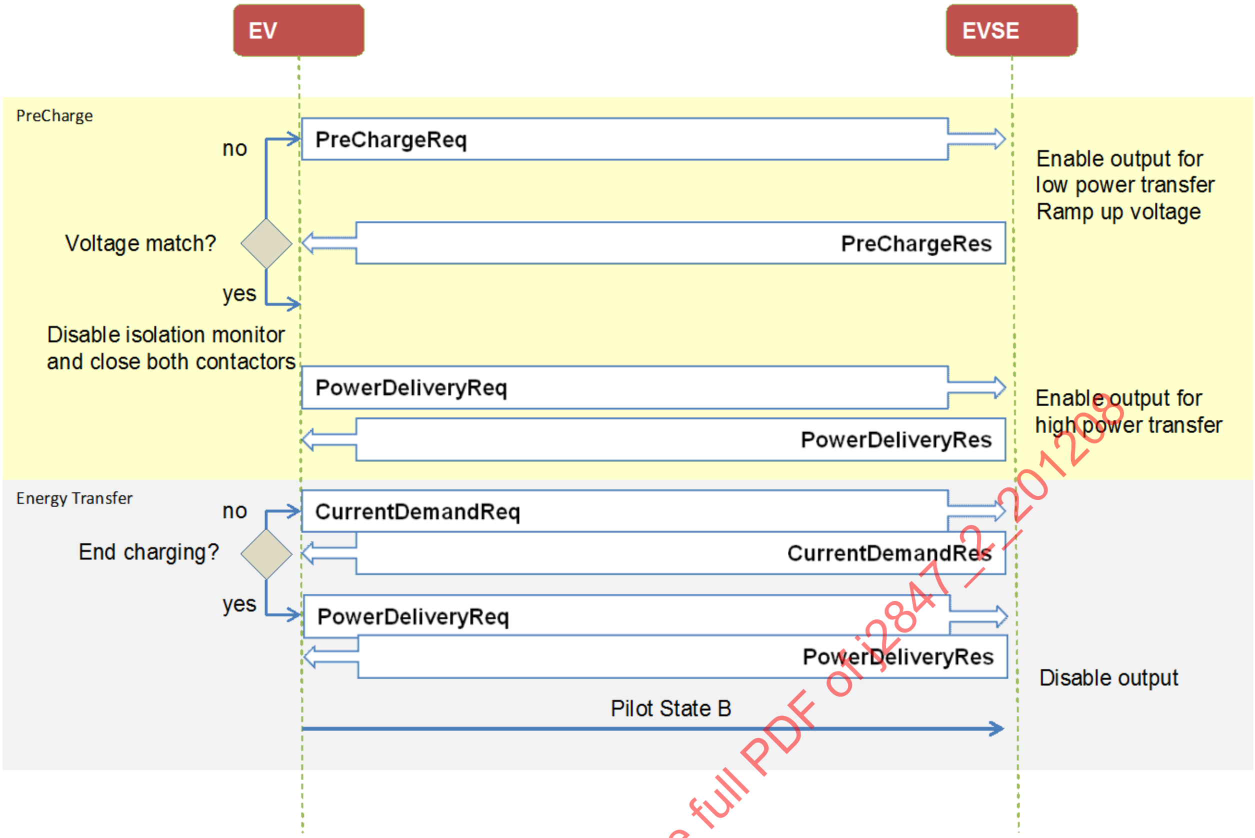


FIGURE A2 - MESSAGE EXCHANGE BETWEEN EV AND EVSE DURING NORMAL SHUTDOWN

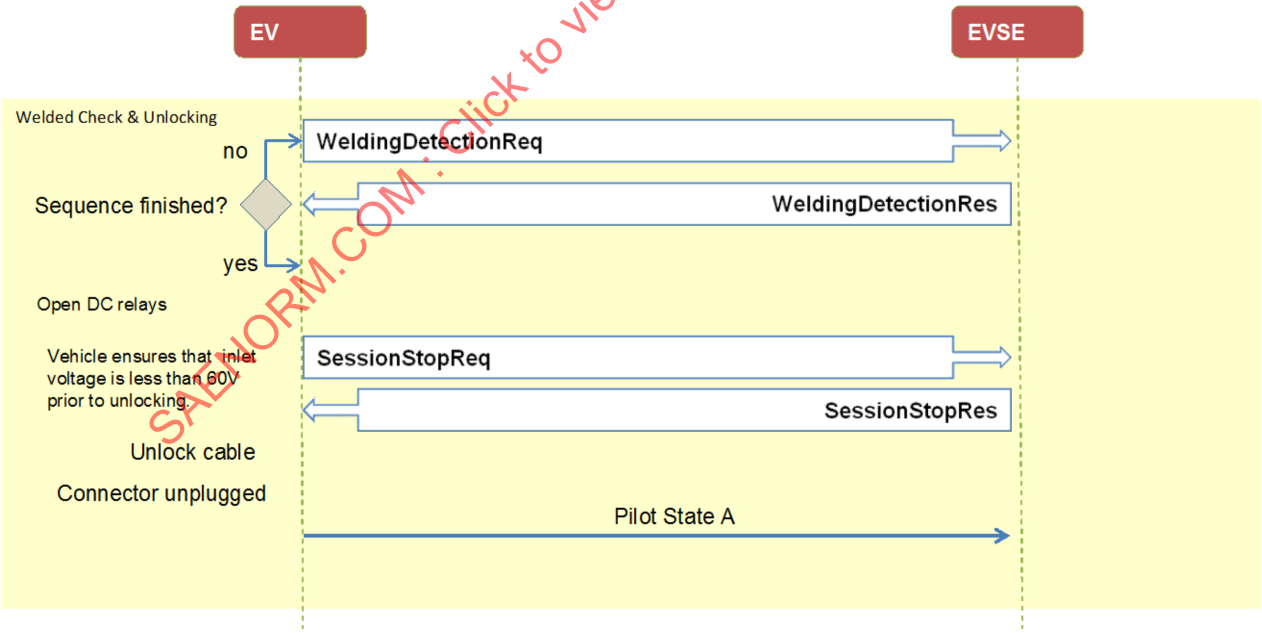


FIGURE A3 - MESSAGE EXCHANGE BETWEEN EV AND EVSE DURING NORMAL DC OPERATION

APPENDIX B - SIGNALS TABLE

Index No.	Signal	Source	Destination	Unit	Purpose	Default value	xml Name	xml Type
4.2.1	Bulk Charging Complete	EVCC	SECC	Boolean	Display	false	BulkChargingComplete	Boolean
4.2.2	Charging Complete	EVCC	SECC	Boolean	Display	False	ChargingComplete	Boolean
4.2.3	Vehicle Energy Capacity	EVCC	SECC	Wh	Display	1000 Wh	EVEnergyCapacity	PhysicalValueType
4.2.4	Vehicle RESS SOC	EVCC	SECC	%	Display	100%	EVRESSSOC	percentValueType
4.2.5	Vehicle Maximum Current Limit	EVCC	SECC	A	Safety	0 A	EVMaximumCurrentLimit	PhysicalValueType
4.2.6	Vehicle Maximum Power Limit	EVCC	SECC	W	Initialization	0 W	EVMaximumPowerLimit	PhysicalValueType
4.2.7	Vehicle Maximum Voltage Limit	EVCC	SECC	V	Safety	0 V	EVMaximumVoltageLimit	PhysicalValueType
4.2.8	Charger Maximum Power Limit	SECC	EVCC	W	Initialization	0 W	EVSEMaximumPowerLimit	PhysicalValueType
4.2.9	Charger Maximum Current Limit	SECC	EVCC	A	Safety	0 A	EVSEMaximumCurrentLimit	PhysicalValueType
4.2.10	Charger Maximum Voltage Limit	SECC	EVCC	V	Safety	0 V	EVSEMaximumVoltageLimit	PhysicalValueType
4.2.11	Charge Current Request	EVCC	SECC	A	Safety	0 A	EVTargetCurrent	PhysicalValueType
4.2.12	Vehicle Ready	EVCC	SECC	Boolean	Display	0	EVReady	Boolean
4.2.13	Vehicle Error Code	EVCC	SECC	Enumeration	Display	NoData	EVErrorCode	DC_EVErrorCodeType
4.2.14	Charger Status Code	SECC	EVCC	Complex	Safety & Display	EVSE_ NotReady	DC_EVSEStatus	DC_EVSEStatusType
4.2.15	Remaining Charging Time to Bulk SoC	EVCC	SECC	seconds	Display	0 sec	RemainingTimeToBulkSoC	PhysicalValueType

Index No.	Signal	Source	Destination	Unit	Purpose	Default value	xml Name	xml Type
4.2.16	Bulk SoC	EVCC	SECC	%	Display	100%	BulkSOC	percentValueType
4.2.17	Remaining Time to Full SoC	EVCC	SECC	seconds	Display	0 sec	RemainingTimeToFullSoC	PhysicalValueType
4.2.18	Full SoC	EVCC	SECC	%	Display	100%	FullSOC	percentValueType
4.2.19	Vehicle Energy Request	EVCC	SECC	Wh	Display	1000 Wh	EVEnergyRequest	PhysicalValueType
4.2.20	Charger Energy to be delivered	SECC	EVCC	Wh	Display	1000 Wh	EVSEEnergyToBeDelivered	PhysicalValueType
4.2.21	Voltage Output	SECC	EVCC	V	Diagnostics	0	EVSEPresentVoltage	PhysicalValueType
4.2.22	Current Output	SECC	EVCC	A	Diagnostics	false	EVSEPresentCurrent	PhysicalValueType
4.2.23	Charger Current Limit Achieved	SECC	EVCC	Boolean	Control	false	EVSECurrentLimitAchieved	Boolean
4.2.24	Charger Voltage Limit Achieved	SECC	EVCC	Boolean	Control	false	EVSEVoltageLimitAchieved	Boolean
4.2.25	Charger Power Limit Achieved	SECC	EVCC	Boolean	Control	false	EVSEPowerLimitAchieved	Boolean
4.2.26	Charger Minimum Voltage Limit	SECC	EVCC	V	Safety	0 V	EVSEMinimumVoltageLimit	PhysicalValueType
4.2.27	Charger Minimum Current Limit	SECC	EVCC	A	Safety	0 A	EVSEMinimumCurrentLimit	PhysicalValueType
4.2.28	Charger Peak Current Ripple	SECC	EVCC	A	Initialization	0 A	EVSEPeakCurrentRipple	PhysicalValueType
4.2.29	Charger Current Regulation Tolerance	SECC	EVCC	A	Initialization	0 A	EVSECurrentRegulationTolerance	PhysicalValueType
4.2.30	Vehicle Requested Energy Transfer Type	EVCC	SECC	Enumeration	Initialization	DC_ Extended	EVRequestedEnergyTransfer	EVRequestedEnergyTransferType
4.2.31	Charger Supported Energy Transfer Type	SECC	EVCC	Enumeration	Initialization	DC_ Extended	EnergyTransferType	EVSESupportedEnergyTransferType

Index No.	Signal	Source	Destination	Unit	Purpose	Default value	xml Name	xml Type
4.2.32	Charge Station Isolation Status	SECC	EVCC	Enumeration	Initialization	Invalid	EVSEIsolationStatus	isolationLevelType
4.2.33	EVCC Target Voltage	EVCC	SECC	V	Initialization	0 V	EVTargetVoltage	PhysicalValueType
4.2.34	Ready to Charge State	EVCC	SECC	Boolean	Control	false	ReadyToChargeState	Boolean
0	Isolation Monitoring Reading	SECC	EVCC	Ohms	Information	n/a	To be defined in a later revision of this document	TBD (not already defined in DIN spec)
4.2.35	PEV Cabin Conditioning	EVCC	SECC	Boolean	Display	false	EVCabinConditioning	Boolean
4.2.36	PEV RESS Conditioning	EVCC	SECC	Boolean	Display	false	EVRESSConditioning	Boolean
4.2.37	Response Code	SECC	EVCC	Enumeration	Control	NoData	EVErrorCode	DC_EVErrorCodeType
4.2.38	Session Setup Response Code	SECC	EVCC	Complex	Initialization	(9)	SessionSetupRes	SessionSetupResType
4.2.39	EVSE ID	SECC	EVCC	value	Initialization	0	EVSEID	evseIDType
4.2.40	PEV ID	EVCC	SECC	value	Initialization	0	EVCCID	evccIDType
4.2.41	Priority	EVCC	SECC	value	Initialization	0	Priority	priorityType
4.2.42	Protocol Namespace	EVCC	SECC	anyURI	Initialization	url	ProtocolNamespace	protocolNamespaceType
4.2.43	Schema ID	SECC & EVCC	EVCC & SECC	value	Initialization	0	SchemaID	idType
4.2.44	Version Number Major	EVCC	SECC	value	Initialization	0	VersionNumberMajor	unsignedInt

Index No.	Signal	Source	Destination	Unit	Purpose	Default value	xml Name	xml Type
4.2.45	Version Number Minor	EVCC	SECC	value	Initialization	0	VersionNumberMinor	unsignedInt
4.2.46	Date Time Now	SECC	EVCC	value	Initialization	0	DateTimeNow	Long
4.2.47	Session ID	EVCC	SECC	value	Initialization	0	SessionID	sessionIDType
4.2.48	Payment Options	SECC	EVCC	String	Initialization	Contract	SelectedPaymentOption	paymentOptionType
4.2.49	Selected Payment Option	EVCC	SECC	String	Initialization	Contract	SelectedPaymentOption	paymentOptionType

APPENDIX C - RESPONSE CODE CORRELATION MATRIX.

	supportedAppProtocolRes	SessionSetupRes	ServiceDiscoveryRes	ServiceDetailRes	ServiceandPaymentSelectionRes	PaymentDetailsRes	ContractAuthenticationRes	ChargeParameterDiscoveryRes	PowerDeliveryRes	ChargingStatusRes	MeteringReceiptRes	CertificateupdateRes	CertificateinstallationRes	CableCheckRes	PreChargingRes	CurrentDemandRes	WeldingDetectionRes	SessionStopRes
OK	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
OK_CertificateExpiresSoon																		
OK_NewSessionEstablished		x																
OK_OldSessionJoined		x																
FAILED	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
FAILED_SequenceError	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
FAILED_SignatureError	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
FAILED_UnknownSession			x	x	x	x	x	x	x	x	x	x	x					x
FAILED_ServiceIDInvalid				x														
FAILED_PaymentSelectionInvalid					x													
FAILED_ServiceSelectionInvalid					x													
FAILED_CertificateExpired						x						x	x					
FAILED_NoCertificateAvailable													x					
FAILED_CertChainError												x						
FAILED_ContractCanceled												x						
FAILED_ChallengeInvalid							x											
FAILED_WrongEnergyTransferType								x										
FAILED_WrongChargeParameter								x										
FAILED_ChargingProfileInvalid									x									
FAILED_TariffSelectionInvalid									x									
FAILED_PowerDeliveryNotApplied									x									
FAILED_MeteringSignatureNotValid											x							

APPENDIX D - XML SCHEMA (FOR REFERENCE BACK TO DIN DOCUMENT)

The XML schema definitions types used in this standard can be distinguished as follows:

- complex type use capitalized first letters
- simple types use non-capitalized first letters.

D.1 V2G_CI_APPPROTOCOL.XSD

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="urn:iso:15118:2:2010:AppProtocol"
    targetNamespace="urn:iso:15118:2:2010:AppProtocol">

    <xs:element name="supportedAppProtocolReq">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AppProtocol" type="AppProtocolType" maxOccurs="20"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="supportedAppProtocolRes">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="ResponseCode" type="responseCodeType"/>
                <xs:element name="SchemaID" type="idType" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="AppProtocolType">
        <xs:sequence>
            <xs:element name="ProtocolNamespace" type="protocolNamespaceType"/>
            <xs:element name="VersionNumberMajor" type="xs:unsignedInt"/>
            <xs:element name="VersionNumberMinor" type="xs:unsignedInt"/>
            <xs:element name="SchemaID" type="idType"/>
            <xs:element name="Priority" type="priorityType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="idType">
        <xs:restriction base="xs:unsignedByte"/>
    </xs:simpleType>
    <xs:simpleType name="protocolNameType">
        <xs:restriction base="xs:string">
            <xs:maxLength value="30"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="protocolNamespaceType">
        <xs:restriction base="xs:anyURI">
            <xs:maxLength value="100"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="priorityType">
        <xs:restriction base="xs:unsignedByte">
            <xs:minInclusive value="1"/>
            <xs:maxInclusive value="20"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="responseCodeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="OK_SuccessfulNegotiation"/>
            <xs:enumeration value="OK_SuccessfulNegotiationWithMinorDeviation"/>
            <xs:enumeration value="Failed_NoNegotiation"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

D.2 V2G_CI_MSGBODY.XSD

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="urn:din:70121:2012:MsgBody"
     xmlns:v2gci_d="urn:din:70121:2012:MsgDef"
     xmlns:v2gci_t="urn:din:70121:2012:MsgDataTypes"
     targetNamespace="urn:din:70121:2012:MsgBody"
     elementFormDefault="qualified"
     attributeFormDefault="unqualified">

     <xs:import namespace="urn:din:70121:2012:MsgDef" schemaLocation="V2G_CI_MsgDef.xsd"/>
     <xs:import namespace="urn:din:70121:2012:MsgDataTypes"
schemaLocation="V2G_CI_MsgDataTypes.xsd"/>

     <!-- ....................... -->
     <!-- Common Messages (AC/DC) -->
     <!-- ....................... -->

     <!--               -->
     <!-- Session Setup -->
     <!--               -->
     <xs:element name="SessionSetupReq" type="SessionSetupReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="SessionSetupReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="EVCCID" type="v2gci_t:evccIDType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="SessionSetupRes" type="SessionSetupResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="SessionSetupResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="EVSEID" type="v2gci_t:evseIDType"/>
                         <xs:element name="DateTimeNow" type="xs:long" minOccurs="0"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                   -->
     <!-- Service Discovery -->
     <!--                   -->
     <xs:element name="ServiceDiscoveryReq" type="ServiceDiscoveryReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ServiceDiscoveryReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ServiceScope" type="v2gci_t:serviceScopeType"
minOccurs="0"/>
                         <xs:element name="ServiceCategory" type="v2gci_t:serviceCategoryType"
minOccurs="0"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="ServiceDiscoveryRes" type="ServiceDiscoveryResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ServiceDiscoveryResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="PaymentOptions" type="v2gci_t:PaymentOptionsType"/>
                         <xs:element name="ChargeService" type="v2gci_t:ServiceChargeType"/>
                         <xs:element name="ServiceList" type="v2gci_t:ServiceTagListType"
minOccurs="0"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                   -->
     <!-- Service Detail    -->
     <!--                   -->
     <xs:element name="ServiceDetailReq" type="ServiceDetailReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ServiceDetailReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ServiceID" type="v2gci_t:serviceIDType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="ServiceDetailRes" type="ServiceDetailResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ServiceDetailResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="ServiceID" type="v2gci_t:serviceIDType"/>
                         <xs:element name="ServiceParameterList"
type="v2gci_t:ServiceParameterListType" minOccurs="0"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                             -->
     <!-- Service Payment & Selection -->
     <!--                             -->
     <xs:element name="ServicePaymentSelectionReq" type="ServicePaymentSelectionReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ServicePaymentSelectionReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="SelectedPaymentOption" type="v2gci_t:paymentOptionType"/>
                         <xs:element name="SelectedServiceList"
type="v2gci_t:SelectedServiceListType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="ServicePaymentSelectionRes" type="ServicePaymentSelectionResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ServicePaymentSelectionResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                 -->
     <!-- Payment Details -->
     <!--                 -->
     <xs:element name="PaymentDetailsReq" type="PaymentDetailsReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="PaymentDetailsReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ContractID" type="v2gci_t:contractIDType"/>
                         <xs:element name="ContractSignatureCertChain"
type="v2gci_t:CertificateChainType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="PaymentDetailsRes" type="PaymentDetailsResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="PaymentDetailsResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="GenChallenge" type="v2gci_t:genChallengeType"/>
                         <xs:element name="DateTimeNow" type="xs:long"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                           -->
     <!-- Contract Authentification -->
     <!--                           -->
     <xs:element name="ContractAuthenticationReq" type="ContractAuthenticationReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ContractAuthenticationReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="GenChallenge" type="v2gci_t:genChallengeType"
minOccurs="0"/>
                    </xs:sequence>
                    <xs:attribute name="Id" type="xs:IDREF"/>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="ContractAuthenticationRes" type="ContractAuthenticationResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ContractAuthenticationResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="EVSEProcessing" type="v2gci_t:EVSEProcessingType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                            -->
     <!-- Charge Parameter Discovery -->
     <!--                            -->
     <xs:element name="ChargeParameterDiscoveryReq" type="ChargeParameterDiscoveryReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ChargeParameterDiscoveryReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="EVRequestedEnergyTransferType"
type="v2gci_t:EVRequestedEnergyTransferType"/>
```

```
                         <xs:element ref="v2gci_t:EVChargeParameter"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="ChargeParameterDiscoveryRes" type="ChargeParameterDiscoveryResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ChargeParameterDiscoveryResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="EVSEProcessing" type="v2gci_t:EVSEProcessingType"/>
                         <xs:element ref="v2gci_t:SASchedules"/>
                         <xs:element ref="v2gci_t:EVSEChargeParameter"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                            -->
     <!-- Power Delivery             -->
     <!--                            -->
     <xs:element name="PowerDeliveryReq" type="PowerDeliveryReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="PowerDeliveryReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ReadyToChargeState" type="xs:boolean"/>
                         <xs:element name="ChargingProfile" type="v2gci_t:ChargingProfileType"
minOccurs="0"/>
                         <xs:element ref="v2gci_t:EVPowerDeliveryParameter" minOccurs="0"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="PowerDeliveryRes" type="PowerDeliveryResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="PowerDeliveryResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element ref="v2gci_t:EVSEStatus"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                            -->
     <!-- Charging Status            -->
     <!--                            -->
     <xs:element name="ChargingStatusReq" type="ChargingStatusReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ChargingStatusReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence/>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="ChargingStatusRes" type="ChargingStatusResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="ChargingStatusResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="EVSEID" type="v2gci_t:evseIDType"/>
                         <xs:element name="SAScheduleTupleID" type="v2gci_t:SAIDType"/>
                         <xs:element name="EVSEMaxCurrent" type="v2gci_t:PhysicalValueType"
minOccurs="0"/>
                         <xs:element name="MeterInfo" type="v2gci_t:MeterInfoType" minOccurs="0"/>
                         <xs:element name="ReceiptRequired" type="xs:boolean"/>
                         <xs:element name="AC_EVSEStatus" type="v2gci_t:AC_EVSEStatusType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                            -->
     <!-- Metering Receipt           -->
     <!--                            -->
     <xs:element name="MeteringReceiptReq" type="MeteringReceiptReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="MeteringReceiptReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="SessionID" type="v2gci_t:sessionIDType"/>
                         <xs:element name="SAScheduleTupleID" type="v2gci_t:SAIDType"
minOccurs="0"/>
                         <xs:element name="MeterInfo" type="v2gci_t:MeterInfoType"/>
                    </xs:sequence>
                    <xs:attribute name="Id" type="xs:IDREF"/>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="MeteringReceiptRes" type="MeteringReceiptResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="MeteringReceiptResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="AC_EVSEStatus" type="v2gci_t:AC_EVSEStatusType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                    -->
     <!-- SessionStop        -->
     <!--                    -->
     <xs:element name="SessionStopReq" type="SessionStopType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="SessionStopType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence/>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="SessionStopRes" type="SessionStopResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="SessionStopResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                        -->
     <!-- Certificate Update     -->
     <!--                        -->
     <xs:element name="CertificateUpdateReq" type="CertificateUpdateReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="CertificateUpdateReqType">
          <xs:complexContent>
```

```
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ContractSignatureCertChain"
type="v2gci_t:CertificateChainType"/>
                         <xs:element name="ContractID" type="v2gci_t:contractIDType"/>
                         <xs:element name="ListOfRootCertificateIDs"
type="v2gci_t:ListOfRootCertificateIDsType"/>
                         <xs:element name="DHParams" type="v2gci_t:dHParamsType"/>
                    </xs:sequence>
                    <xs:attribute name="Id" type="xs:IDREF"/>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="CertificateUpdateRes" type="CertificateUpdateResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="CertificateUpdateResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="ContractSignatureCertChain"
type="v2gci_t:CertificateChainType"/>
                         <xs:element name="ContractSignatureEncryptedPrivateKey"
type="v2gci_t:privateKeyType"/>
                         <xs:element name="DHParams" type="v2gci_t:dHParamsType"/>
                         <xs:element name="ContractID" type="v2gci_t:contractIDType"/>
                         <xs:element name="RetryCounter" type="xs:short"/>
                    </xs:sequence>
                    <xs:attribute name="Id" type="xs:IDREF" use="required"/>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                              -->
     <!-- Certificate Installation     -->
     <!--                              -->
     <xs:element name="CertificateInstallationReq" type="CertificateInstallationReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="CertificateInstallationReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="OEMProvisioningCert" type="v2gci_t:certificateType"/>
                         <xs:element name="ListOfRootCertificateIDs"
type="v2gci_t:ListOfRootCertificateIDsType"/>
                         <xs:element name="DHParams" type="v2gci_t:dHParamsType"/>
                    </xs:sequence>
                    <xs:attribute name="Id" type="xs:IDREF" />
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="CertificateInstallationRes" type="CertificateInstallationResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="CertificateInstallationResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="ContractSignatureCertChain"
type="v2gci_t:CertificateChainType"/>
                         <xs:element name="ContractSignatureEncryptedPrivateKey"
type="v2gci_t:privateKeyType"/>
                         <xs:element name="DHParams" type="v2gci_t:dHParamsType"/>
                         <xs:element name="ContractID" type="v2gci_t:contractIDType"/>
                    </xs:sequence>
                    <xs:attribute name="Id" type="xs:IDREF" use="required"/>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <!-- ....................... -->
     <!-- DC-Messages             -->
     <!-- ....................... -->

     <!--               -->
     <!-- Cable Check   -->
     <!--               -->
     <xs:element name="CableCheckReq" type="CableCheckReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="CableCheckReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="DC_EVStatus" type="v2gci_t:DC_EVStatusType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="CableCheckRes" type="CableCheckResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="CableCheckResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="DC_EVSEStatus" type="v2gci_t:DC_EVSEStatusType"/>
                         <xs:element name="EVSEProcessing" type="v2gci_t:EVSEProcessingType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--               -->
     <!-- Pre-Charge    -->
     <!--               -->
     <xs:element name="PreChargeReq" type="PreChargeReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="PreChargeReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="DC_EVStatus" type="v2gci_t:DC_EVStatusType"/>
                         <xs:element name="EVTargetVoltage" type="v2gci_t:PhysicalValueType"/>
                         <xs:element name="EVTargetCurrent" type="v2gci_t:PhysicalValueType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="PreChargeRes" type="PreChargeResType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="PreChargeResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="DC_EVSEStatus" type="v2gci_t:DC_EVSEStatusType"/>
                         <xs:element name="EVSEPresentVoltage" type="v2gci_t:PhysicalValueType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                   -->
     <!-- Current Demand    -->
     <!--                   -->
     <xs:element name="CurrentDemandReq" type="CurrentDemandReqType"
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="CurrentDemandReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="DC_EVStatus" type="v2gci_t:DC_EVStatusType"/>
                         <xs:element name="EVTargetCurrent" type="v2gci_t:PhysicalValueType"/>
                         <xs:element name="EVMaximumVoltageLimit" type="v2gci_t:PhysicalValueType" 
minOccurs="0"/>
                         <xs:element name="EVMaximumCurrentLimit" type="v2gci_t:PhysicalValueType" 
minOccurs="0"/>
                         <xs:element name="EVMaximumPowerLimit" type="v2gci_t:PhysicalValueType" 
minOccurs="0"/>
                         <xs:element name="BulkChargingComplete" type="xs:boolean" minOccurs="0"/>
                         <xs:element name="ChargingComplete" type="xs:boolean"/>
                         <xs:element name="RemainingTimeToFullSoC" type="v2gci_t:PhysicalValueType" 
minOccurs="0"/>
                         <xs:element name="RemainingTimeToBulkSoC" type="v2gci_t:PhysicalValueType" 
minOccurs="0"/>
                         <xs:element name="EVTargetVoltage" type="v2gci_t:PhysicalValueType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="CurrentDemandRes" type="CurrentDemandResType" 
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="CurrentDemandResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="DC_EVSEStatus" type="v2gci_t:DC_EVSEStatusType"/>
                         <xs:element name="EVSEPresentVoltage" type="v2gci_t:PhysicalValueType"/>
                         <xs:element name="EVSEPresentCurrent" type="v2gci_t:PhysicalValueType"/>
                         <xs:element name="EVSECurrentLimitAchieved" type="xs:boolean"/>
                         <xs:element name="EVSEVoltageLimitAchieved" type="xs:boolean"/>
                         <xs:element name="EVSEPowerLimitAchieved" type="xs:boolean"/>
                         <xs:element name="EVSEMaximumVoltageLimit" type="v2gci_t:PhysicalValueType" 
minOccurs="0"/>
                         <xs:element name="EVSEMaximumCurrentLimit" type="v2gci_t:PhysicalValueType" 
minOccurs="0"/>
                         <xs:element name="EVSEMaximumPowerLimit" type="v2gci_t:PhysicalValueType" 
minOccurs="0"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>

     <!--                      -->
     <!-- Welding Detection -->
     <!--                      -->
     <xs:element name="WeldingDetectionReq" type="WeldingDetectionReqType" 
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="WeldingDetectionReqType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="DC_EVStatus" type="v2gci_t:DC_EVStatusType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:element name="WeldingDetectionRes" type="WeldingDetectionResType" 
substitutionGroup="v2gci_d:BodyElement"/>
     <xs:complexType name="WeldingDetectionResType">
          <xs:complexContent>
               <xs:extension base="v2gci_d:BodyBaseType">
                    <xs:sequence>
                         <xs:element name="ResponseCode" type="v2gci_t:responseCodeType"/>
                         <xs:element name="DC_EVSEStatus" type="v2gci_t:DC_EVSEStatusType"/>
                         <xs:element name="EVSEPresentVoltage" type="v2gci_t:PhysicalValueType"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
</xs:schema>
```