ROCIN-ECO

System Requirements Document

(Lastenheft)

Systemanforderungen ROCIN-ECO

As of Q3 / 2024

Table of contents

I. References

Name	Content
User requirements_v1.0 FINAL.docx	Collection of user requirements for the Robot charger User Story 2a
UseCases_FINAL.xls	Defined use cases for robot charging at the beginning of the project

1. Introduction and Purpose

1.1 Purpose of the Document

This document provides recommended system requirements for ACD-S charging sites that has been gathered and evaluated during the ROCIN-ECO project. These requirements include user and system requirements for the charging site (e.g. parking space), the ACD-S unit and the EV. The document has the status of the current project progress and is subject to change during the project until the project's end.

1.2 Terms, Definitions and Abbreviations

The following in terms and definitions will be used throughout this document and are listed here for reference.

ACD-S Case	Automated charging from side socket
ACD-S	Automated Connection Device - Side
AVDS	Automated Valet Driving System
AVP(S)	Automated Valet Parking (System)
ASM	Area Site Manager
BLE	Bluetooth Low Energy
CCC	Connected Car Consortium
CCS	Combined Charging System
EV	Electric Vehicle
HMI	Human-Machine-Interface
Legacy Case	Use Case using EV that does not support EV-robot communication
Mating Zone	Area where the Robot is capable of mating with the EV
ODD	Operational design domain
OZ	Operation Zone
URD	User requirements Document
UWB	Ultra Wide Band

Table 1: List of abbreviations

2. Use Cases

The user requirements in this documents are based on four general use cases for side-robot charging (ACD-S) that have been identified by the ROCIN-ECO project team.

These use cases are:

1. Charging as a service (driver triggers charging) – LEGACY Case

2a. Charging as a service (EV/Robot triggers charging)

2b. Customer drives into Queueing situation manually (Robot is moving between Evs)

3 APVS

While the simples use case is a manually triggered robot by the user (legacy case) in future scenarios manual handling of the charging plug will be replaced by the autonomous handling by the robot.

In all cases the charging experience shall be as easy as possible for the user and hurdles shall be minimized.

While all use cases will be relevant for future charging scenarios, this User Requirement Documentation focuses on use case 2a. where the EV/Robot communication triggers charging and use case 3.for AVPS. Use case 1. And 2b. will be deviations of the other two use cases that are downwards compatible.

2.1 Use Case 1: Driver Triggers Charging manually (LEGACY Case)

The driver uses a legacy vehicle that does not support ACD-S communication, so he needs to park manually. The start and stop of the charging process and therefore docking and undocking process will be triggered manually by the user.

This can be done for example by:

- Pushing a dock/undocking button on the charging robot
- Triggering start/stop charging by using the HMI at the Charger that triggers the docking/undocking process
- Using an CPO-/third party APP
- Start via Backend
- Etc..

2.2 Use Case 2a: EV triggers Docking and Undocking

In this use case the vehicle will be capable of communicating with the robot on its own. The charging process and therefore the docking/undocking will be triggered by the EV and not by the driver.

The driver needs to position the EV correctly in the robot's mating zone. Live tests with volunteers showed that the driver wishes a support function for exact positioning of the EV. As the EV supports communication with the robot in this use case, the EV should provide a feedback system for the user to position the EV correctly.

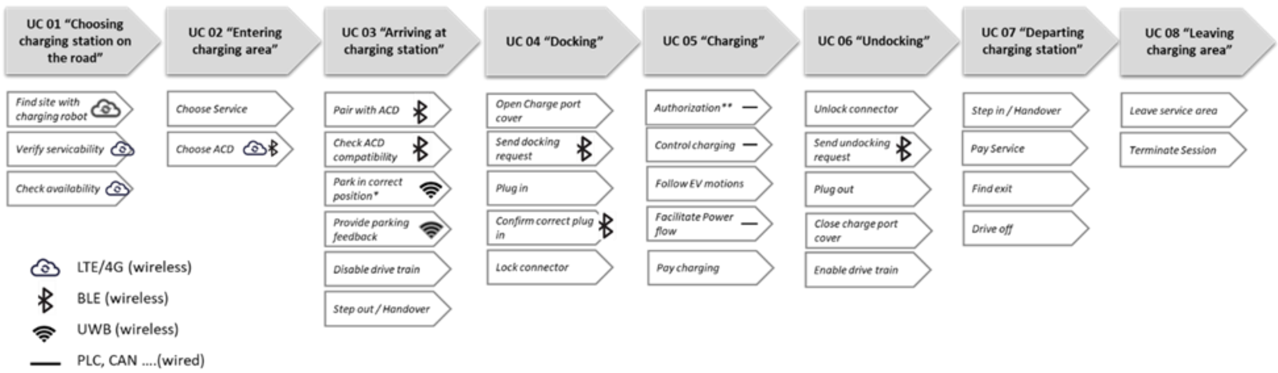


Figure 1: Use Case 2a - EV to trigger ACD

Figure 1: Use Case 2a - EV to trigger ACD shows the indicated process for this use case from the navigation to the site, the parking and charging process until the driver leaves the area.

2.3 Use Case 2b: Queuing Situation

Use Case 1b. refers to a queuing situation where the driver parks in a certain queuing area and the robot is able to serve at least two or more vehicles.

This use case will not be evaluated practically within the project. But could be worth evaluating in the future.

2.4 Use Case 3: AVPS

Automated Valet Parking System describes the fully autonomous use case and is the most complex one. The EV needs to fully support AVP.

The driver is responsible to drive the EV to a certain handover zone. From there the EV takes over and fully autonomously parks the vehicle. The full charging process is automized incl. authentication. When the driver requests the EV for handover it will automatically return to the handover zone.

This use case is only described theoretically and will not be evaluated or realized within the project

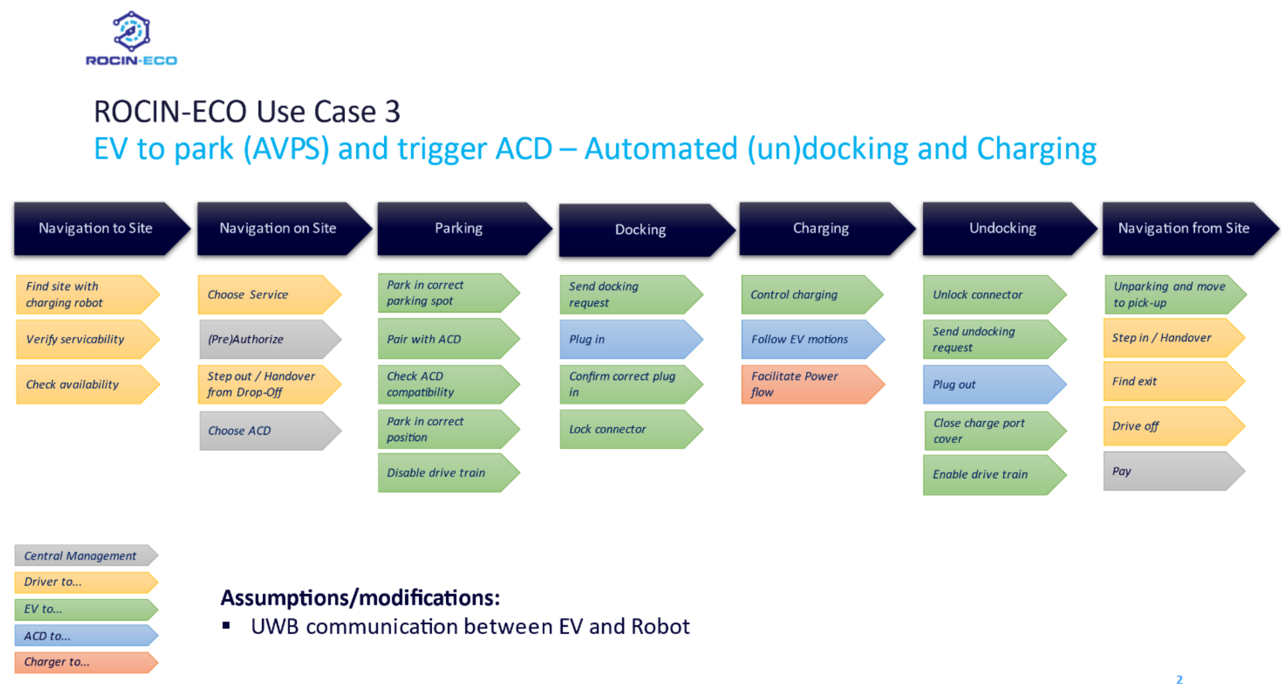


Figure 2: Use Case 3 - AVPS

3. User Requirements

Please see "User_Requirements_v1.0"

4. System Requirements

The following System Requirements are the results of theoretical discussions with the ROCIN ECO Project team in collaboration with real test scenarios that have been performed by the ROCIN ECO team.

At first assumption for each part of the ecosystem have been made by the project team. For assumption that have been already verified during the project by the project team, the results are included in the following sections.

4.1 Operational design domain

The following setup was used during the ROCIN-ECO project:

Component	Type
Charger	IONITY ABB Gen3 350kW CCS Charger
Cable	Huber&Suhner 500Amp Radox 400
Charging Robot	Rocsys ROC-1
UWB/BLE Communication device	Marquardt Keyfob integration with Rocsys on Audi eTron (Prototype)
Electric Vehicle	Audi eTron, additionally equipped with UWB/BLE Antennas and a ECU to implement a guidance/positioning system
Charging Site for Site evaluation and live tests	Rocsys, Rijswjik, NL IONITY Testsite, Unterschleissheim, DE

The following ODD is adapted from ISO 34503 (Road Vehicles — Test scenarios for automated driving systems — Specification for operational design domain) and merged to the requirements for ACD-S. This increases efficiency for harmonization:

Attribute	Requirements
Operation Zone (OZ)	See IEC 61851-1 – Operation area
Parking space for ACD-S charging	The drivable area shall have a paved surface with no abnormal bumps or holes. Road markings may be used where necessary in specific use.
Accuracy Requirement for parking position	see IEC 61851-27 – Mating Space requirements Parking accuracy is defined in AVDS ISO 12768-1
Outside air temperature	Temperature range between -20°C and +40°C. Local conditions may require extended ranges
Weather	ACD-S shall be able to master even difficult weather conditions like snowfall or ice during wintertime as well as heavy rain to provide a high availability for the user.

	The Area site manager (ASM) is responsible to maintain a drivable OZ (in case of AVP site)
Visibility	The visibility in the OZ between robot and vehicle inlet should be given and free from any obstacles. See package space requirements in IEC 61851-27 for details.
Illumination	All daylight conditions independent of cloud cover and position or elevation of the sun. At night and in buildings an illumination of at least 20 lx is required throughout the whole OZ. Additional light shall be provided by the ACD-S where needed.
Connectivity	Vehicle to infrastructure (V2I) and vehicle to network (V2N) communication shall be available throughout the whole OZ. The specific communication technology like 4G, 5G, Wi-Fi etc. depends on the local availability and is not prescribed in this document. ACD-S to vehicle communication must be available within the following ranges: < 30 meter from charge point (roboter) for BLE, < 10 meter from charge point (roboter) for UWB

4.2 High Level System Architecture

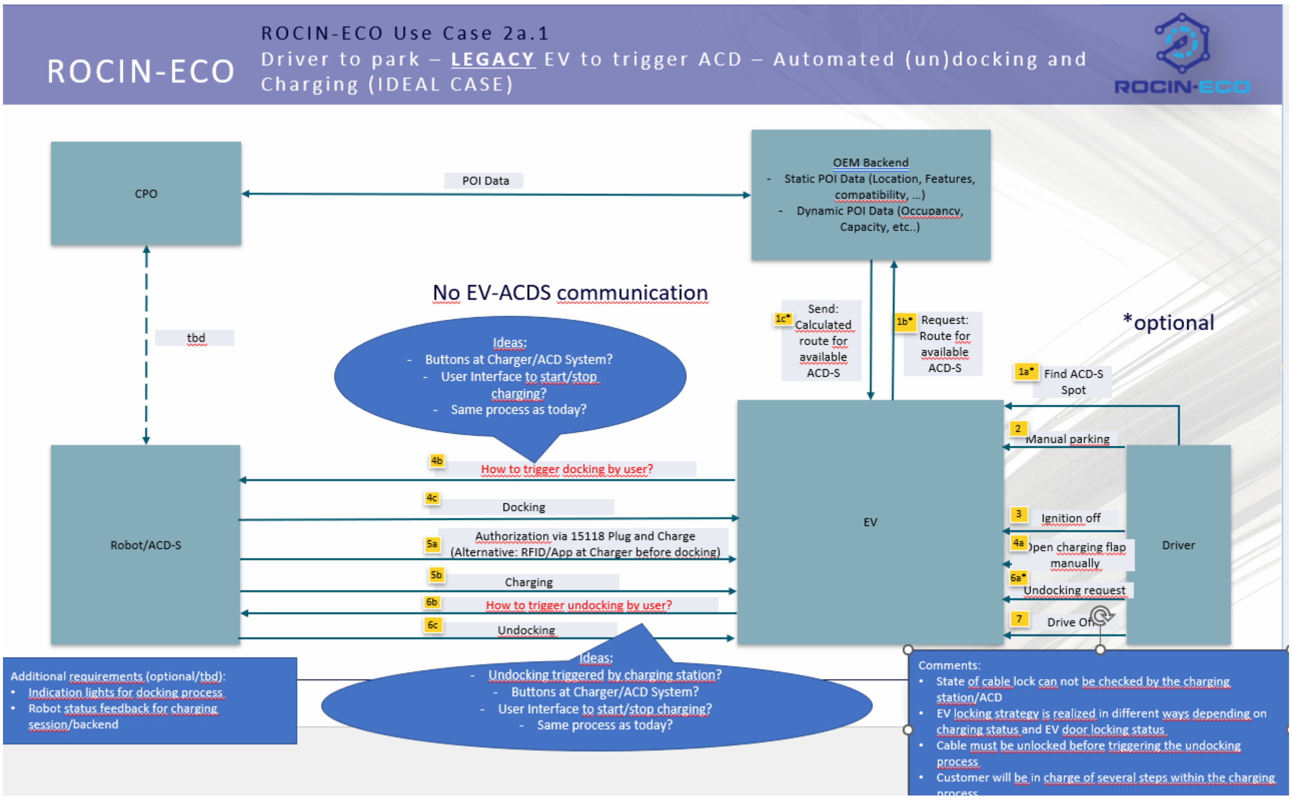
4.3 Project Assumptions

- For the evaluation and development of the system the complexity should be as simple as possible
- Focus on EV-robot communication and recycling of features and functionalities that are already available and implemented for “normal” charging (e.g. charging communication)
- Future architectures, e.g. a central management system, will be an implementation for later stages of the robot and charging ecosystem
- Where applicable approved and standardized components should be used.

4.3.1 Use Case 1: No ACD-S Communication (Legacy Case)

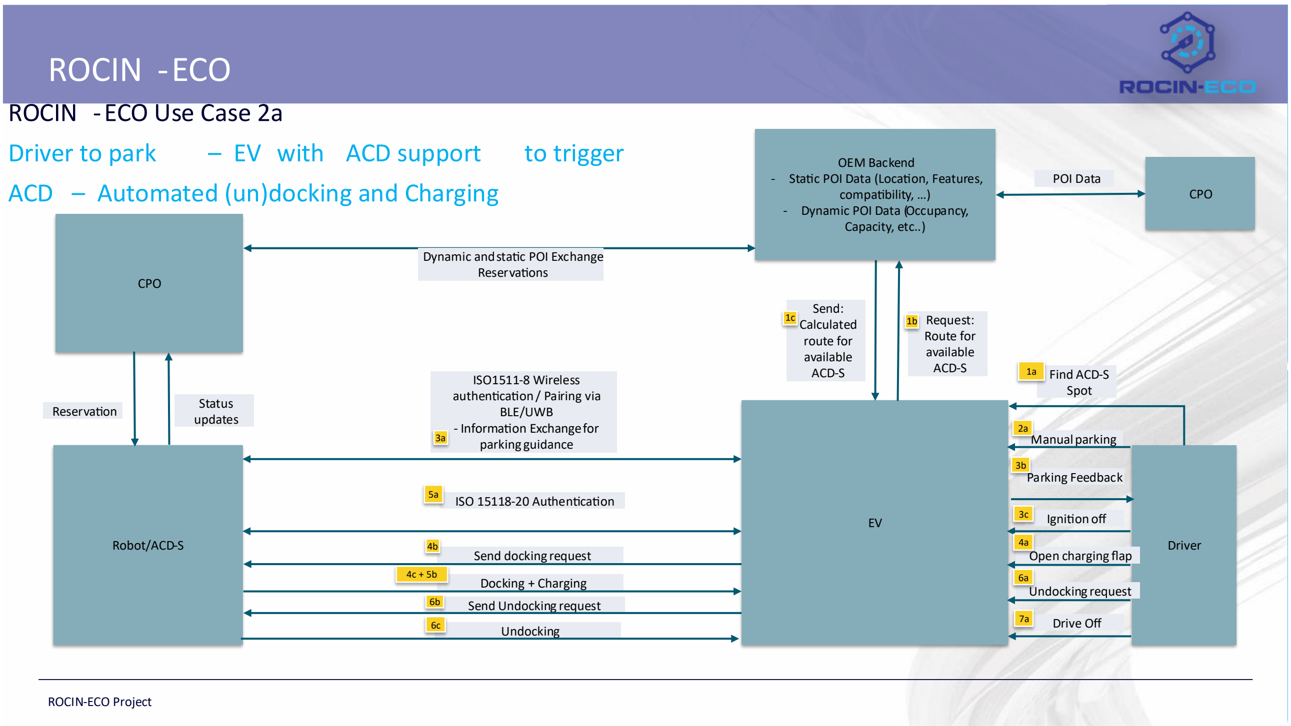
For the legacy case there will be no communication between EV and Charging Robot. The docking and undocking process needs to be triggered by the driver manually. This requires individual solutions for a legacy robot-charging system. The first approach how to handle this is documented in the follow graphic.

The project consortium was aligned on the opinion that the only sustainable way for ACD-S charging requires EV-robot communications as seen in Use Case 2a and further.

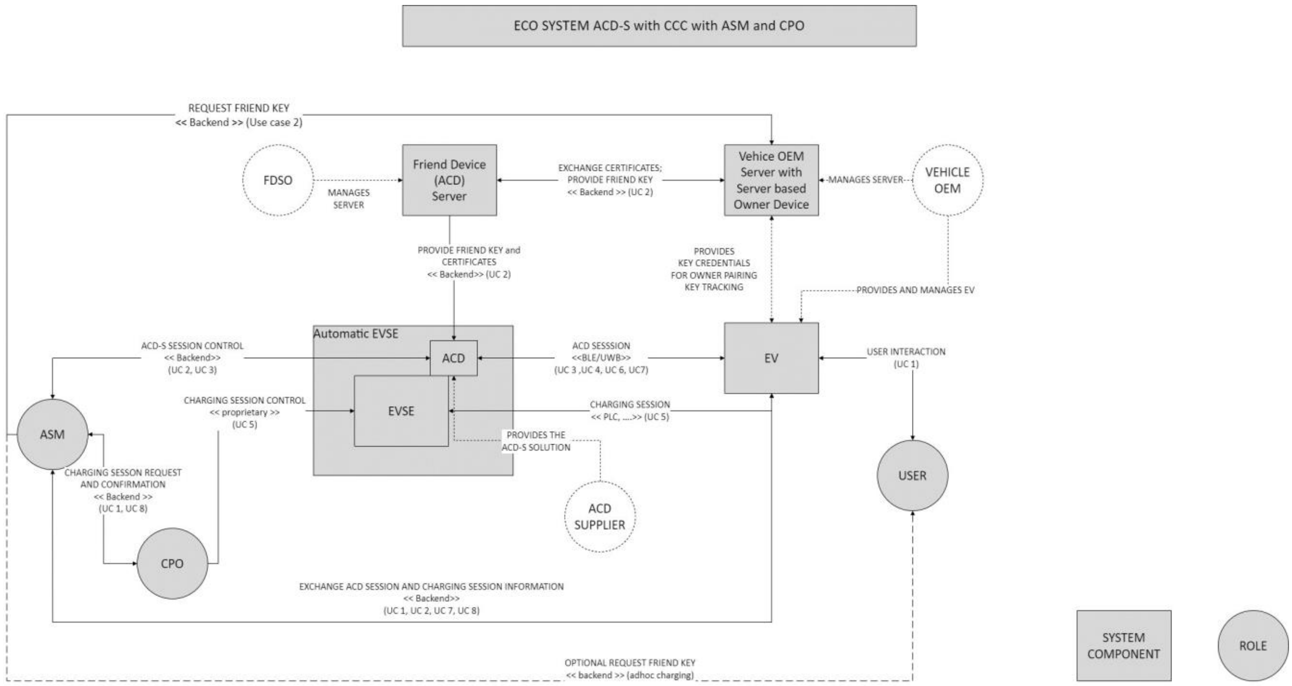


4.3.2 Use Case 2a: With ACD-S communication

This use case is the **main use case for the project**. It requires EV-Robot communication via BLE/UWB that is defined more in detailed within this document or within references.



The project partners worked within the standardization groups of ISO 5474-5 and IEC 61851-27 and -28 to standardize the ACD-S communication. This state of the standard development has been used for the project assumptions.



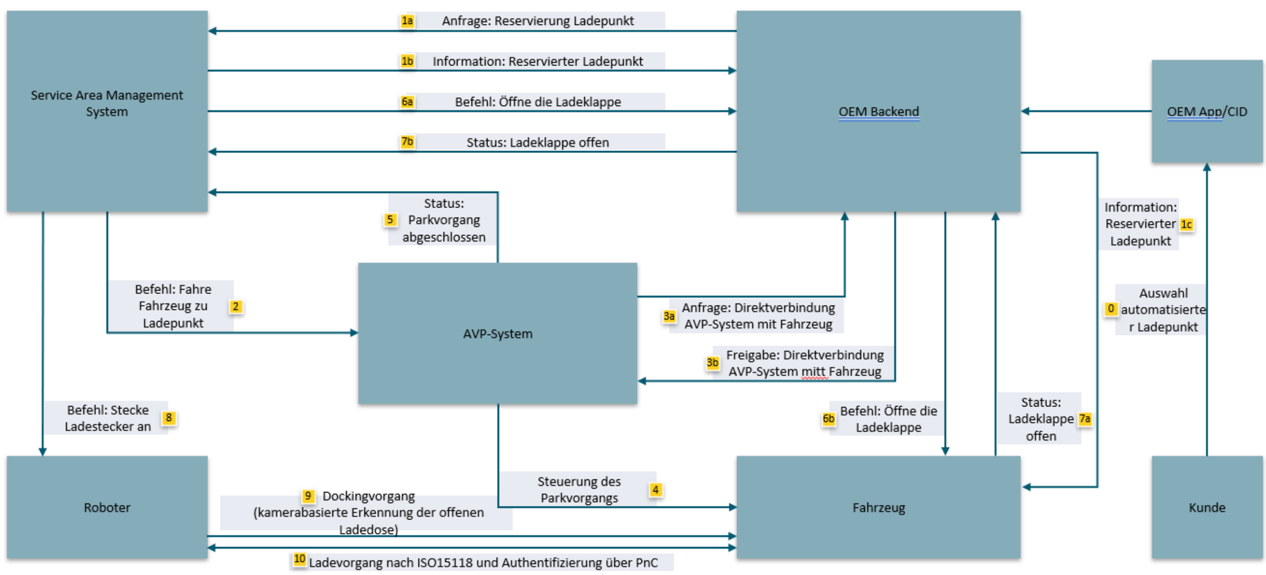
4.3.3 Use Case 2b: Queuing

Basis for Use Case 2b is Use Case 2a. This use case is not in scope of the project.

4.3.4 Use Case 3: AVPS

See Use Case 2a) for mandatory base features. AVPS development and standardization is still in development during the runtime of the ROCIN-ECO Project.

VOLLAUTONOMES PARKEN UND LADEN (AUTONOMES PARKEN TYP 2).



4.4 Site Layout

4.4.1 Project assumptions

- The parking layout should be designed in a away for the driver to park easily within the mating zone of the robot
- Compatibility between EVs with different charging socket (inlet) positions should be taken into account

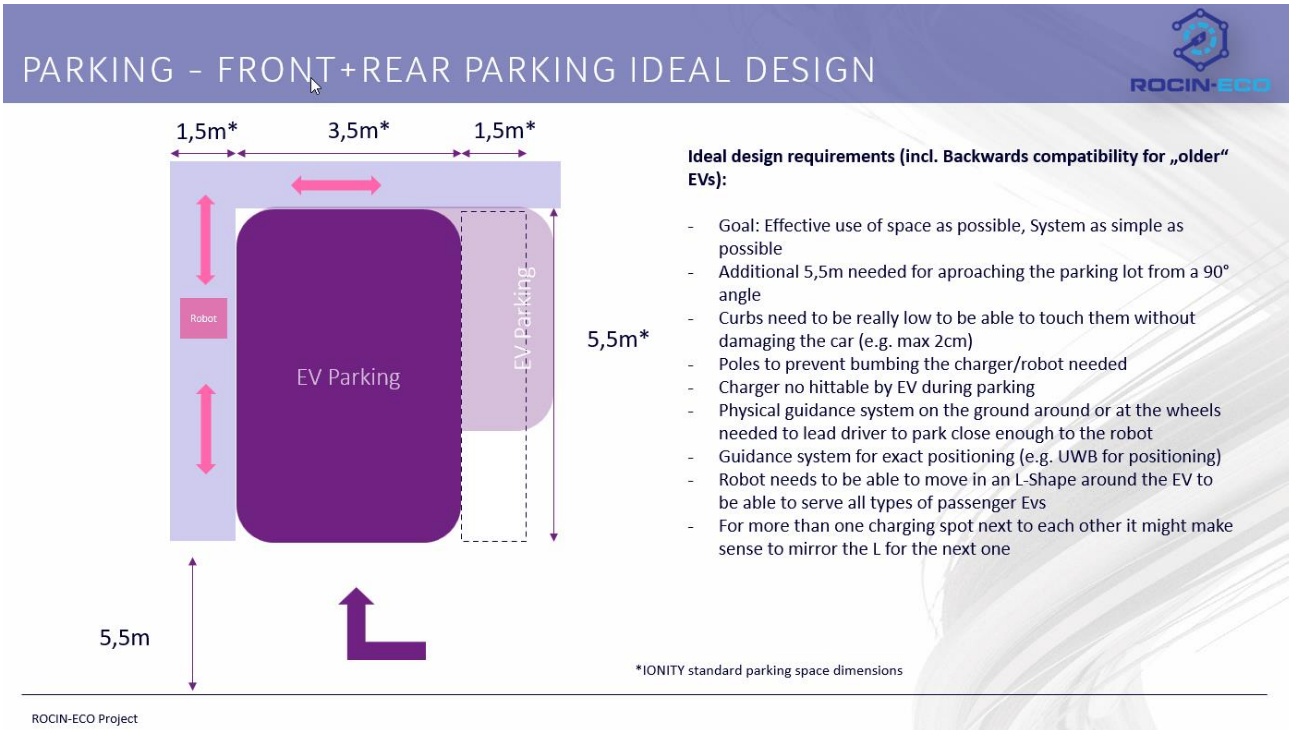
4.4.2 Overview

During the project we evaluated two different parking scenarios practically with user feedback and evaluated the technical feasibility.

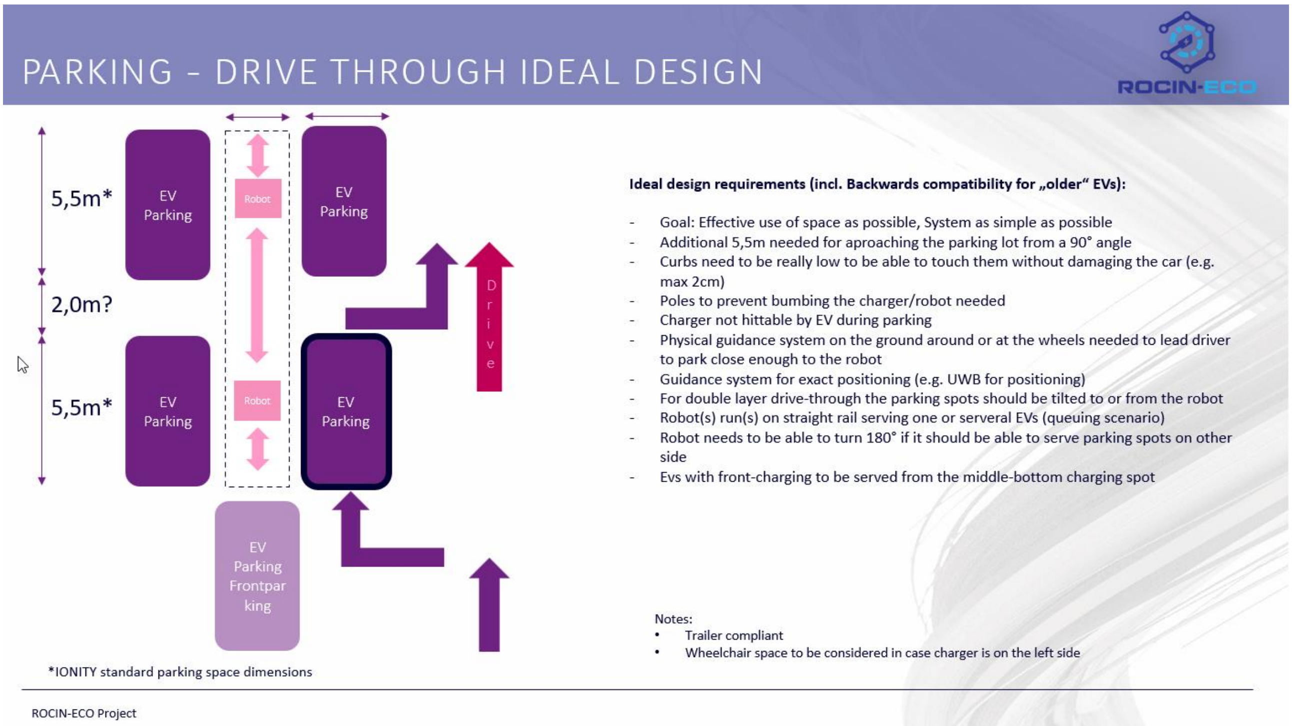
The tests showed that a front/backwards parking scenario leads to more complexity of the robot system to be able to serve different vehicles with all kind of charging socket positions.

4.4.3 Parking Space and Dimensions based on project

Test with different parking scenarios showed that the following two parking scenarios are preferred for the autonomous charging use cases of the ROCIN-ECO project.

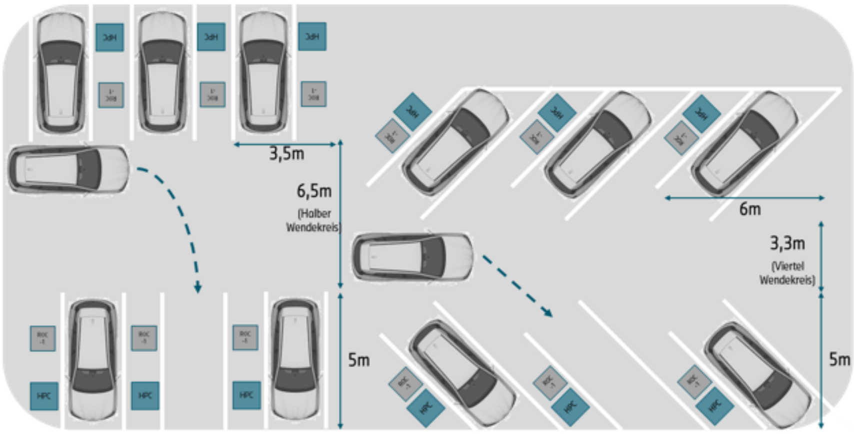


The drive through layout brings flexibility for the user to park his vehicle from both directions to improve general compatibility to the robot system.



4.4.4 Additional parking scenarios to consider

LADEPARK-SZENARIEN
FRONT- VS. SCHRÄGPARKER

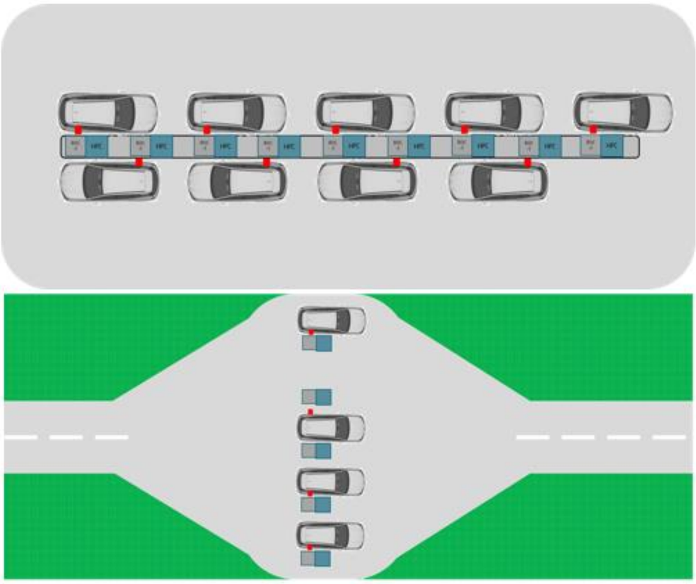


- Anlehnung an Raststätten-Parkplätze
- Ein Roboter neben jedem Parkplatz

→ Hoher Platzbedarf

- Abhängig von Ladedosenposition ist die richtige Positionierung im Parkplatz anspruchsvoll
- Schrägparker erleichtert Positionierung durch kleinere Lenkwinkel → leichter einzuschätzen (weniger Bedeutung für AVC)
- **Arbeitsbereich des Roboters muss in X-Richtung deutlich erweitert werden um fast alle Ladedosen zu erreichen**

LADEPARK
TANKSTELLEN-SITUATION



- Ladepark als „Strom-Tankstelle“
- Laderoboter aufgereiht wie Zapfsäulen
- Erweiterung des Roboters um 180° Drehbarkeit?
- Freie Positionierung des Fahrzeugs entsprechend der Ladedosen-Position
- Unterschiedliche Layouts um Anfahrt zu erleichtern (Reihe oder Parallel)

4.4.5 Placement of Robot and Charger, Spaces

Figure 3: Operating Space for ACD-S


ROCSYS
IONITY

4.4.6 Signs and Instructions

Instructions for compatibility reasons to be considered.

The parking space could be marked for the end user as autonomous charging spot. This includes also markings on the ground to define the parking position for a manually driven vehicle.

Optional: For parking guidance and autonomous driven vehicles a sign on the parking space shall be considered that can be read by the EV front camera. This sign then has an offset to the charging robot that can be used to generate parking guidance.

4.4.7 Parking guidance

The primary means of parking guidance is to get feedback about the correct parking position.

For a **manually driven vehicle** this can be realized by:

- Using physical means like curbs for the driver to get physical feedback
- Using marking of parking spot (with and without 360 degree vehicle camera)

- Using vehicle camera with additional marking on the screen and visual feedback
- Using UWB for positioning and dedicated in-car feedback for driver

For the **AVPS** case this can be realized by:

- Using the parking system according to AVDS standard (ISO 12768). In this case the AVDS system must have knowledge of the mating space.

4.4.8 Curbs

Depending on the site layout the curbs should be flat enough not to damage the vehicles rim's or underbody. Shallow curbs with an ~45 degree angle could help the user to park correctly within the parking zone and mating space for the robot.

4.4.9 Collision Protection

In case there are no further measures to protect the charger again collision with an vehicle, bumpers could be considered.

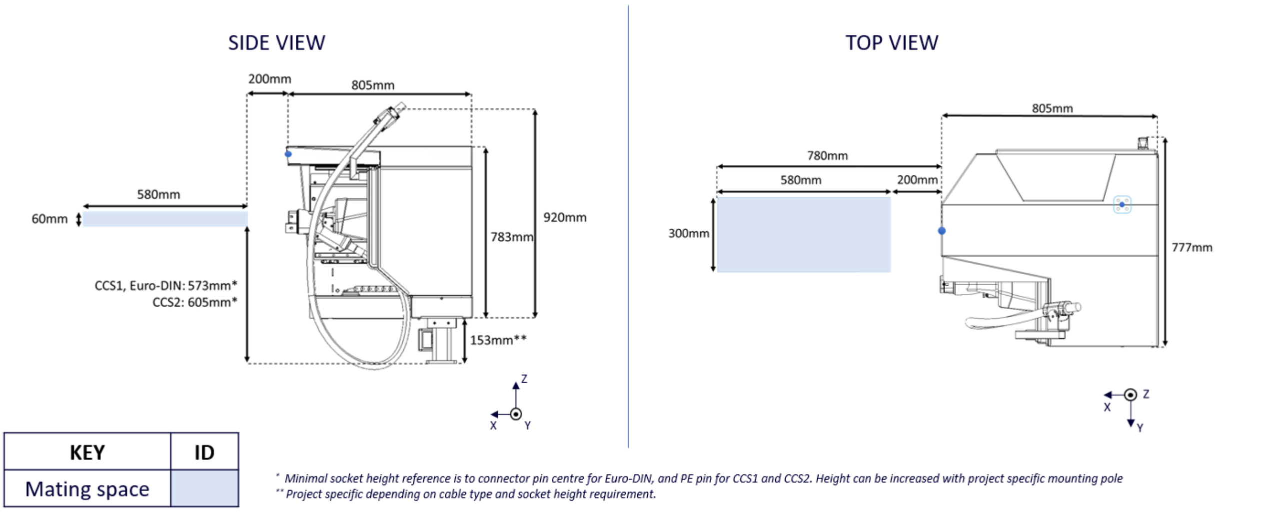
4.5 Charging Robot

4.5.1 Project assumptions

- The Charger should be viewed and deployed as a separate system to the ACD-S System
- CCS2 High Power only
- The robot can have a functional design (focus on technology evaluations)
- As the work range is limited this needs to be considered for the parking guidance and parking layout
- Safety requirements for charging robot not standardized yet and need to be evaluated during the project

4.5.2 Mating Spaces

Product specific experience has been generated during the project with the Rocsys ROC-1 robot.



Within standardization the mating space definition is subject of standardization group ISO 5474-5 and IEC 61851-27.

4.5.3 Cable Management

Cable should not influence the movement of the robot in a way of adding force on the connector.

Good practice: The cable management system should be designed in a way that the cable is not dragged over ground or other rough surfaces.

4.5.4 Exception handling

The following two exception flows have been created within the project consortium.

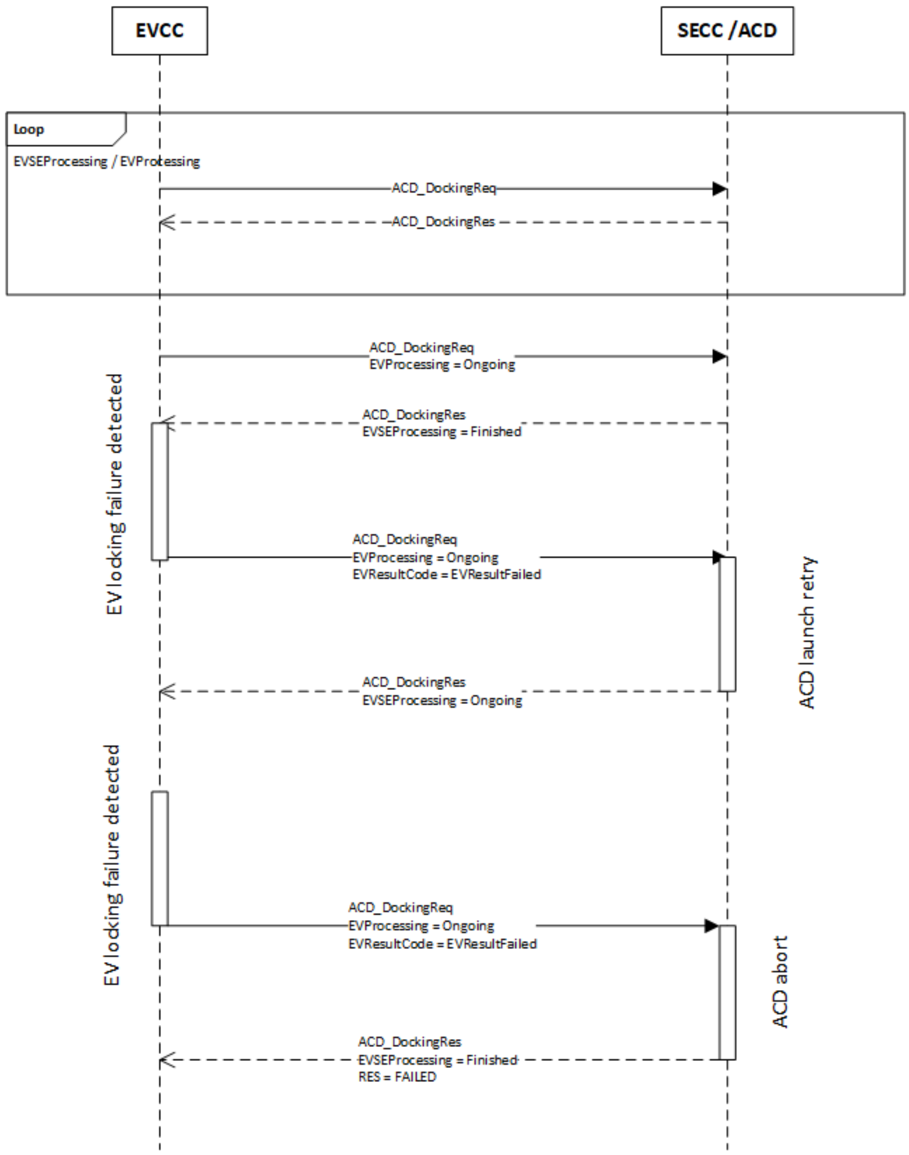


Figure 4: Exception handling for docking process

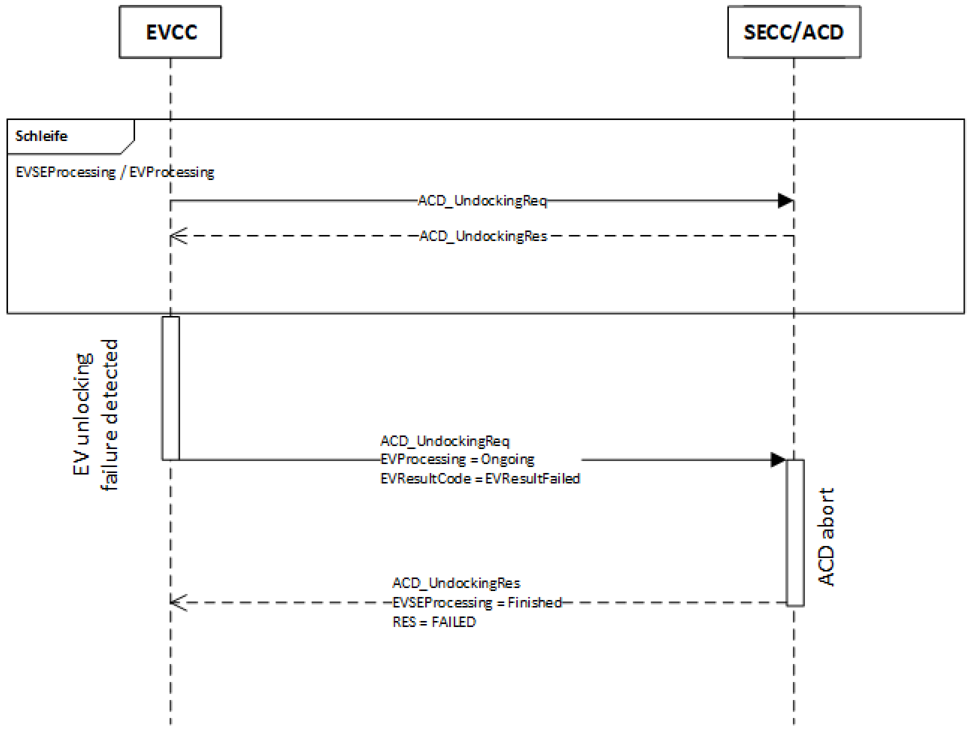


Figure 5: Exception handling for docking process

4.5.5 Emergency Button vs Emergency Stop

Differentiation:

Emergency switching off / emergency cut-off (Notaus bzw. Not-Ausschaltung)	**Emergency stop (Not-Halt)**
Deactivates all electric power sourcing for an electric system or immediate removal of electric power to the actuators of a machine (EN 60204 9.2.3.4.3).	Stops all movements of a machine and deactivates the power to its actuators.
Needs to be Stop-Category 0: uncontrolled deenergizing of a system	Stop-Category 0 (uncontrolled stop) or 1 (controlled stop with deenergization after reaching halt) allowed
System allowed to repower when emergency switch is brought back to "armed" position.	System is not allowed to automatically resume movement after deactivation of the emergency switch.
Example: emergency switch off at a lab test stand.	Example: Motorcycle killswitch.

Regarding the charging robot emergency switching off is not necessarily needed for the robot parts as long as the risk of electric shock is not given due to the usage of extra-low voltage components. The safety of the high voltage parts is ensured by using an IEC 61851-23 compliant charger incl. compliant charging cable and connector.

Open questions to be considered for robot-system safety:

- Is an emergency stop mandatory in every case? --> avoidable, if a site layout requirement can be added like "avoid obstacles which may lead to sheering in the surrounding of the robot" and the robot design prevents pinching.
- Specific national and local regulations differentiate between an emergency stop for the robot and emergency switch off for the charger. This needs to be evaluated for each specific case.
- How is the restart going to take place after emergency stop was triggered. Allowed by common user, need of qualified personnel (onsite or remote)? This is to be checked against the local and national requirements.

4.5.6 Miscellaneous requirements

Noise Emission

Loud noises during movement of the robot will limit the areas of use:

Area	Time	Loudness in dB(A)
Industrial areas	All day	70
Commercial areas	Day	65
	Night	50
Rural areas	Day	63
	Night	45
Core areas of cities, villages and mixed use areas	Day	60
	Night	45
Residential areas, dwelling areas	Day	55
	Night	40
Residential-only areas	Day	50
	Night	35
Hospitals, sanatoria, spa areas	Day	45
	Night	35

Single short time peak noise limit is +30dB(A) day and +20 dB(A) night.

Table is derived from TA Lärm (Sechste Allgemeine Verwaltungsvorschrift zum Bundes-Immissionsschutzgesetz) chapter 6. See requirements for each country.

Photobiologic safety

No blue lights currently intended to be used. No high intensity LEDs are intended to be used.

4.6 ACD-S / EV communication

4.6.1 Project assumptions

- Parking correctly in the mating zone becomes a challenge for the EV driver without any parking guidance
- Technology-supported parking guidance (following the parking sensor feedback) could ensure proper parking within the robot's mating zone
- Existing technology (UWB/BLE) for functions that are needed for the EV/ACD-S communication should be used within the project.

4.6.2 Problems to be solved with wireless communication

Communication between electric vehicle and ACD-S is needed to provide solution solutions for following functions:

Must have:

- Position pairing of EV and ACD-S (make sure that the correct EV is paired with the correct ACD-S)
- Start Docking when vehicle is ready and undocking after charging is finished.
- Trigger locking and unlocking of connector.
- Undocking Confirmation by ACD-S that manipulator is fully retracted and out of the mating zone.

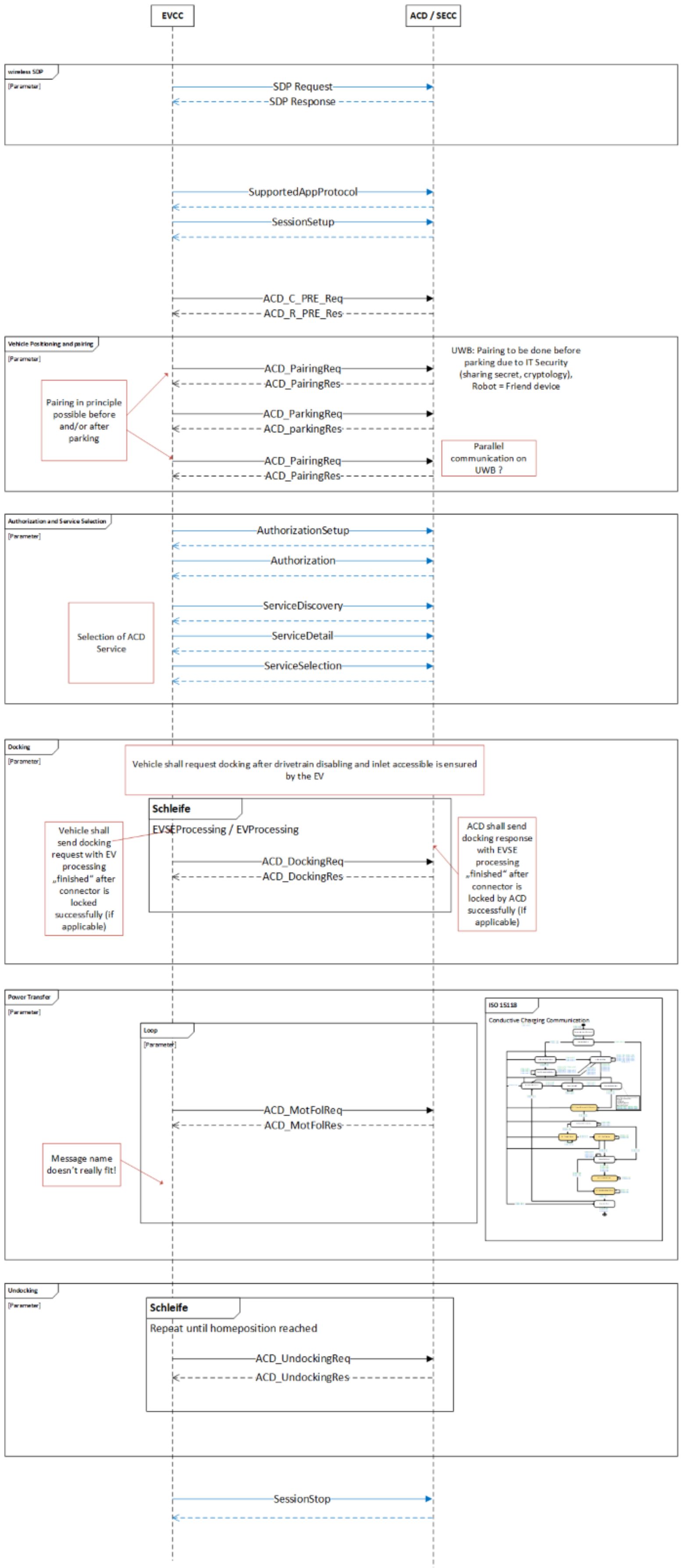
Recommended:

- Get status for user information
- Process optimization
- …

4.6.3 Communication Sequence

The sequence for “normal” dc charging will be fully integrated into the ACD-S communication sequence.

The following sequence is an example based on the ISO 15118 communication sequence.


EVCC
ACD / SECC
wireless SDP
[Parameter]
SDP Request
SDP Response
SupportedAppProtocol
SessionSetup
ACD_C_PRE_Req
ACD_R_PRE_Res
Vehicle Positioning and pairing
[Parameter]
ACD_PairingReq
ACD_PairingRes
UWB: Pairing to be done before parking due to IT Security (sharing secret, cryptology), Robot = Friend device
Pairing in principle possible before and/or after parking
ACD_ParkingReq
ACD_parkingRes
ACD_PairingReq
ACD_PairingRes
Parallel communication on UWB ?
Authorization and Service Selection
[Parameter]
AuthorizationSetup
Authorization
ServiceDiscovery
ServiceDetail
ServiceSelection
Selection of ACD Service
Docking
[Parameter]
Vehicle shall request docking after drivetrain disabling and inlet accessible is ensured by the EV
Schleife
EVSEProcessing / EVProcessing
Vehicle shall send docking request with EV processing „finished“ after connector is locked successfully (if applicable)
ACD_DockingReq
ACD_DockingRes
ACD shall send docking response with EVSE processing „finished“ after connector is locked by ACD successfully (if applicable)
Power Transfer
[Parameter]
Loop
[Parameter]
ACD_MotFolReq
ACD_MotFolRes
Message name doesn't really fit!
ISO 15118
Conductive Charging Communication
Undocking
[Parameter]
Schleife
Repeat until homeposition reached
ACD_UndockingReq
ACD_UndockingRes
SessionStop

4.6.4 UWB/BLE

Tests with UWB within the project have shown that a UWB system delivers good results for positioning feedback of the vehicle.

BLE has the benefit of low energy consumption and is a technology that has already been used with keyless entry systems.

Therefore the combination of UWB and BLE provided existing technology that can be used to handle communication between EV and robot.

See IEC 61851-28 annex for BLE/UWB communication.

4.6.5 Pairing Process

BLE pairing according to CCC standard.

See IEC 61851-28 annex for BLE/UWB communication.

4.7 EV Requirements

4.7.1 Project assumptions

- EV will be driven and parked by the driver manually
- An EV that is available (on the market) needs to be used for project evaluations (Audi eTron)
- Audi eTron is updated with proof of concept features for the project (BLE/UWB, guidance system, automatic flap, etc..)

4.7.2 Socket and Inlet positions

Requirements for socket and inlet positions have been provided by the project team to ISO 5474-5 “Mating Spaces”.

4.7.3 Communication by EV to automatic EVSE

This section describes the message exchanges the EV may initiate, and how the automatic EVSE shall respond.

A. Overviews

Table 1 contains an overview of the message exchanges the EV may initiate towards the automatic EVSE, and the response of the automatic EVSE, to facilitate the automated connection process. This section elaborates on these message exchanges by providing the conditions the EV shall fulfil when initiating the respective message exchanges, i.e. the assumptions the automatic EVSE makes or could verify when receiving the respective requests for action or information.

Table 1 – Overview of commands by EV to ACD station

Key	**Description**	**Actionable in state**	**Response**
ACD_C_COM (**B.3.1.2**)	Set up communication	ACD_S_STA	ACD_R_COM
ACD_C_TER (**B.3.1.3**)	Terminate communication	ACD_S_PRE	ACD_R_TER
		ACD_S_REA	
		ACD_S_EXC	
ACD_C_CAN (**B.3.1.4**)	Cancel	ACD_S_PRE	ACD_R_CAN ACD_R_CCO
		ACD_S_REA	
		ACD_S_DOC	
		ACD_S_LOC	
ACD_C_ABT (**B.3.1.5**)	Abort	ACD_S_DOC	ACD_R_ABT
		ACD_S_UND	
		ACD_S_LOC	
ACD_C_PRE (**B.3.1.6**)	Prepare for arrival	ACD_S_PRE ACD_S_REA	ACD_R_PRE ACD_R_PCO
ACD_C_DOC (**B.3.1.7**)	Request docking	ACD_S_REA ACD_S_PRE	ACD_R_DOC ACD_R_DCO
ACD_C_LOC (**B.3.1.8**)	Transition to latched	ACD_S_MAT	ACD_R_LOC
ACD_C_MAT (**B.3.1.9**)	Transition to mated	ACD_S_LOC	ACD_R_MAT
ACD_C_UND (**B.3.1.10**)	Request undocking	ACD_S_MAT	ACD_R_UND ACD_R_UCO
ACD_C_DIA (**B.3.1.11**)	Diagnostics information	All	ACD_R_DIA
ACD_C_CMP (**B.3.1.12**)	Compatibility information	All	ACD_R_CMP
ACD_C_PAI (**B.3.1.13**)	Pairing information	All	ACD_R_PAI
ACD_C_PAR (**B.3.1.14**)	Parking information	All	ACD_R_PAR
ACD_C_RAN (**B.3.1.15**)	In-range information	All	ACD_R_RAN

B. Set up communication (ACD_C_COM and ACD_R_COM)

Description

With this message exchange the EV initiates an ACD session with the automatic infrastructure, otherwise referred to as communication setup.

Behaviour

In normal operating behaviour communication is set up in the respective standby states (ACD_S_STA, and ACD_V_STA). Reinitiating the ACD session after a loss of communication is described in [TODO: refer to reinitiating communication].

Content

N/A

Sequence

[TODO: description of how to interpret ACD_R_COM]

The exact sequence to set up communication is described in [TODO: refer to sequence] .

C. Terminate communication (ACD_C_TER, and ACD_R_TER)

Description

With this message exchange the EV requests to terminate communication with the automatic EVSE. The automatic EVSE will respond positively with ACD_R_TER directly after having received the command.

Behaviour

This request can be sent in a state where the EV and automatic EVSE are not in physical contact, i.e. ACD_*_PRE, ACD_*_REA and ACD_*_EXC.

Content

N/A

Sequence

The exact sequence to terminate an ACD session is described in [TODO: refer to sequence].

D. Cancel (ACD_C_CAN, ACD_R_CAN, and ACD_R_CCO)

Description

With this message exchange the EV requests to undo an ongoing action, and let the manipulator return to its previous state.

Behaviour

The automatic EVSE will respond with ACD_R_CAN directly after having received the command: positively when it will act upon the cancellation, and negatively when it won't. Subsequently, it will respond with ACD_R_CCO after having completed the cancellation process.

[TODO: when does cancel apply, what behaviour follows; refer to process flow]

Content

ACD_C_CAN: None

ACD_R_CAN:

Type of information	Content
Will start with cancellation	True / False

ACD_R_CCO:

Type of information	Content

Cancellation completed successfully	True / False

Sequence

The exact sequence to cancel an action is described in [TODO: refer to sequence].

E. Abort (ACD_C_ABT, and ACD_R_ABT)

Description

With this message exchange the EV requests an immediate stop of all activities, and a transition to the exception state.

Behaviour

This message exchange may be initiated in any EV state. However, it is mainly relevant in states where the EV and manipulator are or might be physically interacting. The actions by the manipulator triggered by this command depend on the state the automatic EVSE is in:

[TODO: when does the manipulator do what when receiving an abort]

The automatic EVSE will respond with ACD_R_ABT after having initiated the relevant actions.

Content

ACD_C_ABT: None

ACD_R_ABT:

Type of information	**Content**
Will abort	True / False

Sequence

The exact sequence to abort an action is described in [TODO: refer to sequence].

F. Prepare for arrival (ACD_C_PRE, ACD_R_PRE, and ACD_R_PCO)

Description

With this message exchange, the EV notifies the automatic EVSE that it has decided to proceed with the automated connection process with that automatic EVSE following certain parameters. This message exchange is not required to execute the process, but it may reduce the time that the manipulator needs for docking.

Behaviour

This message exchange can be initiated in EV state ACD_V_PRE or ACD_V_REA. The automatic EVSE may perform actions to prepare for the arrival of the vehicle when the automatic EVSE is in ACD_S_PRE or in ACD_S_REA. For example, these actions could include moving the manipulator from a stored position to a position aligned with the intended mating space.

If the EV does not use this message exchange, but the automatic EVSE does need to perform preparations, receiving ACD_C_DOC in ACD_S_PRE has the effect of subsequently performing the default ACD_C_PRE and ACD_C_DOC message exchange.

The manipulator shall remain out of the clearance space while performing preparatory actions.

The automatic EVSE shall respond with ACD_R_PRE directly after receiving the request to prepare: positively when starting preparations, otherwise negatively. Subsequently, after having completed the necessary activities (which may be none), the automatic EVSE shall respond with ACD_R_PCO.

Content

The EV may send the following content in the ACD_C_PRE message exchange, to facilitate preparations:

ACD_C_PRE

Type of information	Content
Intended mating space	[TODO: refer to mating space description, and attribute a communication format to describe those mating spaces] Default: None
Inlet orientation	[TODO: consider how to describe this] Default: None
Charge door orientation	[TODO: consider how to describe this] Default: None
…	

ACD_R_PRE:

Type of information	**Content**
Will start with preparations	True / False

ACD_R_PCO:

Type of information	**Content**
Preparations completed successfully	True / False

Sequence

The exact sequence for the message exchange to start preparations is described in [TODO: refer to sequence].

G. Request docking (ACD_C_DOC, ACD_R_DOC, and ACD_R_DCO)

Description

With this message exchange the EV requests the automatic EVSE to start docking.

Behaviour

This command can be given in the EV state ACD_V_DOC. The **automatic EVSE** shall proceed towards ACD_S_DOC when receiving this command in ACD_S_REA, when the relevant conditions are met. When receiving this command in ACD_S_PRE while requiring ACD_C_PRE to continue, the **automatic EVSE** shall execute the actions and send responses as if triggered by ACD_C_PRE, and shall respond to ACD_C_DOC as if it were received in ACD_S_REA. In other **automatic EVSE** states, the **automatic EVSE** shall not take action.

The EV shall fulfil the following conditions before the ACD station takes action.

The EV has disabled the drive train.

The inlet is fully uncovered and accessible for the manipulator.

The automatic EVSE shall respond with ACD_R_DOC directly: positively when it expects to successfully perform docking, or negatively when it does not expect to successfully perform docking. Subsequently, the automatic EVSE shall respond positively with ACD_R_DCO when docking is done, or negatively when docking has failed.

Content

ACD_C_DOC: None

ACD_R_DOC:

Type of information	**Content**
Will start docking	True / False

ACD_R_DCO:

Type of information	**Content**
Docking completed successfully	True / False

Sequence

The exact sequence for the message exchange to perform docking is described in [TODO: refer to sequence].

H. Transition to latched (ACD_C_LOC, and ACD_R_LOC)

Description

With this message exchange the EV notifies the automatic EVSE that it has engaged the interlock.

Behaviour

This message exchange may be initiated in the EV state ACD_V_MAT. The automatic EVSE shall proceed towards ACD_S_LOC when receiving this message in ACD_S_MAT. In other states of the automatic EVSE, it will not take action.

The EV should fulfil the following conditions before initiating this message exchange:

The EV has engaged the interlock.

The automatic EVSE shall respond with ACD_R_LOC directly.

Content

ACD_C_LOC: None

ACD_R_LOC:

Type of information	Content
Will transition to ACD_S_LOC	True / False

Sequence

The exact sequence for the message exchange to notify engagement of the interlock is described in [TODO: refer to sequence].

I. Transition to mated (ACD_C_MAT, and ACD_R_MAT)

Description

With this message exchange the EV notifies the automatic EVSE that it has disengaged the interlock.

Behaviour

This message exchange can be initiated in the EV state ACD_V_LOC. The automatic EVSE shall proceed towards ACD_S_MAT when receiving this message in ACD_S_LOC. In other states, the automatic EVSE will not take action.

The EV should fulfil the following conditions before initiating this message exchange:

The EV has disengaged the interlock.

The automatic EVSE will respond with ACD_R_MAT directly after receiving the message.

Content

ACD_C_MAT: None

ACD_R_MAT:

Type of information	Content

Will transition to ACD_S_MAT	True / False

Sequence

The exact sequence for the message exchange to notify disengagement of the interlock is described in [TODO: refer to sequence].

J. Request undocking (ACD_C_UND, ACD_R_UND, and ACD_R_UCO)

Description

With this message exchange the EV requests the automatic EVSE to start undocking.

Behaviour

This command can be given in the EV state ACD_V_UND. The automatic EVSE shall proceed towards ACD_S_UND when receiving this command in ACD_S_MAT. In other states, the automatic EVSE shall not take action.

The EV should fulfil the following conditions before initiating this message exchange:

The EV has disengaged the interlock.

The automatic EVSE shall respond with ACD_R_UND directly after receiving the message.

Subsequently, the automatic EVSE shall positively respond with ACD_R_UCO after having completed undocking, or negatively when failing to successfully complete undocking.

Content

ACD_C_UND: None

ACD_R_UND:

Type of information	**Content**
Will start undocking	True / False

ACD_R_UCO:

Type of information	**Content**
Has successfully completed undocking	True / False

Sequence

The exact sequence for the message exchange to perform undocking is described in [TODO: refer to sequence].

K. Diagnostics information (ACD_C_DIA, and ACD_R_DIA)

Description

This message exchange may be used to inquire diagnostics information of the automatic EVSE.

Behaviour

When the EV sends this message, the automatic EVSE shall answer using ACD_R_DIA.

Content

ACD_C_DIA: None

ACD_R_DIA:

Type of information	**Content**
System operational	True / False
System state	[TODO: describe communicatable version of system states]
Manipulator moving	True / false
…	

Sequence

The exact sequence for the message exchange for diagnostics information described in [TODO: refer to sequence].

L. Request compatibility information (ACD_C_CMP, and ACD_R_CMP)

Description

This message exchange may be used to inquire about the compatibility parameters of the automatic EVSE.

Behaviour

When the EV initiates the message exchange, the automatic EVSE shall directly respond using ACD_R_CMP.

Content

ACD_C_CMP: None

ACD_R_CMP:

Type of information	**Content**

Available mating spaces	Set of [TODO: refer to mating space description, and attribute a communication format to describe those mating spaces]
Available connector interfaces	Set: [Type 1, Combo 1, Type 2, Combo 2, Chademo] [TODO: refer to IEC62196-2, and -3]
Available parking feedback / in range mechanisms	Set of: [ACD_C_PAR, ACD_C_RAN]
…	

Sequence

The exact sequence for the message exchange for compatibility information is described in [TODO: refer to sequence].

M. Inquire pairing information (ACD_C_PAI)

[TODO: describing /referring to position pairing behaviour]

N. Inquire parking information (ACD_C_PAR)

Description

This message exchange is used to provide information about the automatic EVSE to allow the EV to determine the offset between the inlet and the centre of the mating space the EV intends to use.

Note: be aware that the position and/or orientation of infra-side radio tags may change during the process.

Behaviour

When the EV initiates the message exchange, the automatic EVSE shall directly respond using ACD_R_PAR.

Content

ACD_C_PAR: None

ACD_R_PAR:

Type of information	Content
Position and optionally orientation of infra-side radio tags w.r.t. the common coordinate system.	List of [static tag: true/false, [x,y,z],[rx,ry,rz]], [static tag: true/false, [x,y,z],[rx,ry,rz]] [TODO: refer to common coordinate system, and to ranging description]

Sequence

The exact sequence for the message exchange for parking information is described in [TODO: refer to sequence].

O. Request in-range confirmation (ACD_C_RAN)

Description

This message exchange is used to provide the EV with information whether its inlet is within the intended mating space, without using interactive ranging.

Behaviour

When the EV initiates the message exchange, the automatic EVSE shall directly respond using ACD_R_RAN.

Content

ACD_C_RAN: None

ACD_R_RAN:

Type of information	Content
Inlet is in range	True/false

Sequence

The exact sequence for the message exchange for in-range information is described in [TODO: refer to sequence].

A. Broadcasting errors by EV to automatic EVSE

It is under consideration to broadcast errors from the EV to the automatic EVSE. This section describes the different error messages the EV shall transmit in which situation, and how the automatic EVSE shall react.

A proposal for this section is expected.

B.Communication to EV by automatic EVSE

A. General

This subclause describes the message exchanges with the EV that the automatic EVSE may initiate, and how the EV shall respond. B.3.3.2 gives an overview, and the communication is described in more detail in the subclauses that follow.

Generally, the automatic EVSE does not initiate a message exchange with the EV regarding the normal process flow. However, there may be situations where the automatic EVSE needs to start undocking, while the EV is not aware of that need. For example: a critical failure for the automatic EVSE is imminent, like power failure. To prevent blocking of the EV due to failure of the automatic EVSE, the automatic EVSE may request the EV to subsequently terminate charging, disengage the interlock mechanism, transition to the mated state, and request the automatic EVSE to perform undocking.

B. Overview

Table 2 contains an overview of the commands the automatic EVSE may send to the EV, to facilitate safe and convenient operation.

Table 2 – Overview of commands to EV by automatic EVSE

Key	Description	Actionable in state	Response
ACD_C_STO (**B.3.3.3**)	Request proceed to undocking	ACD_V_LOC	ACD_R_STO
		ACD_V_MAT	
ACD_C_DIA (**B.3.3.4**)	Diagnostics information	All	ACD_R_DIA

C. Request proceed to undocking (ACD_C_STO, and ACD_R_STO)

Description

With this message exchange, the automatic EVSE notifies the EV that it needs to undock.

Behaviour

This message exchange may be initiated in automatic EVSE states ACD_S_LOC and ACD_S_MAT, and received in EV states ACD_V_LOC and ACD_V_MAT. When receiving this request, the EV shall respond positively, and, depending on the precise point in the process, proceed to stop the charging session, disengage the interlock mechanism, transition to ACD_V_MAT, and proceed towards ACD_V_UND, and hence executing a normal undocking process flow on request of the automatic EVSE.

The EV will respond with ACD_R_STO directly after determining whether it is permissible to fulfil the request: positively when permissible, negatively when not permissible.

Content

ACD_C_STO: None

ACD_R_STO:

Type of information	Content
Will proceed towards undocking	True / False

Sequence

The exact sequence for the message exchange regarding proceeding towards undocking on request of the automatic EVSE is described in [TODO: refer to sequence].

D. Diagnostics information (ACD_C_DIA, and ACD_R_DIA)

Description

This message exchange may be used to inquire diagnostics information of the EV.

Behaviour

When the EV sends this message, the automatic EVSE shall answer using ACD_R_DIA.

Content

ACD_C_DIA: None

ACD_R_DIA:

Type of information	Content
System operational	True / False
System state	[TODO: describe communicatable version of system states]
Inlet accessible	True / false
Interlock engaged	True / false
Power transfer active	True / false
…	
…	

Sequence

The exact sequence for the message exchange for diagnostics information described in [TODO: refer to sequence].

C. Broadcasting errors to EV by automatic EVSE

It is under consideration to broadcast errors from the automatic EVSE to the EV. This section describes the different error messages the automatic EVSE shall transmit in which situation, and how the EV shall react.

4.7.4 Positioning Feedback

Several trials with voluntaries showed that a parking feedback within the EV is requested by the user to support the parking accuracy in the mating zone of the robot. The parking feedback for the EV to support correct parking should be implemented in all EVs that fully support ACD-S communication.

4.7.4.1 UWB Ranging

Positioning feedback or guidance can be realized via UWB Ranging, which takes over ranging technology from the digital smart key technology.

4.7.4.2 Optical Guidance

As an alternative, an easy way is to do optical guidance is using the front or back camera to identify markings on the ground and maneuver the vehicle to place the pilot lines in the camera into the marking.

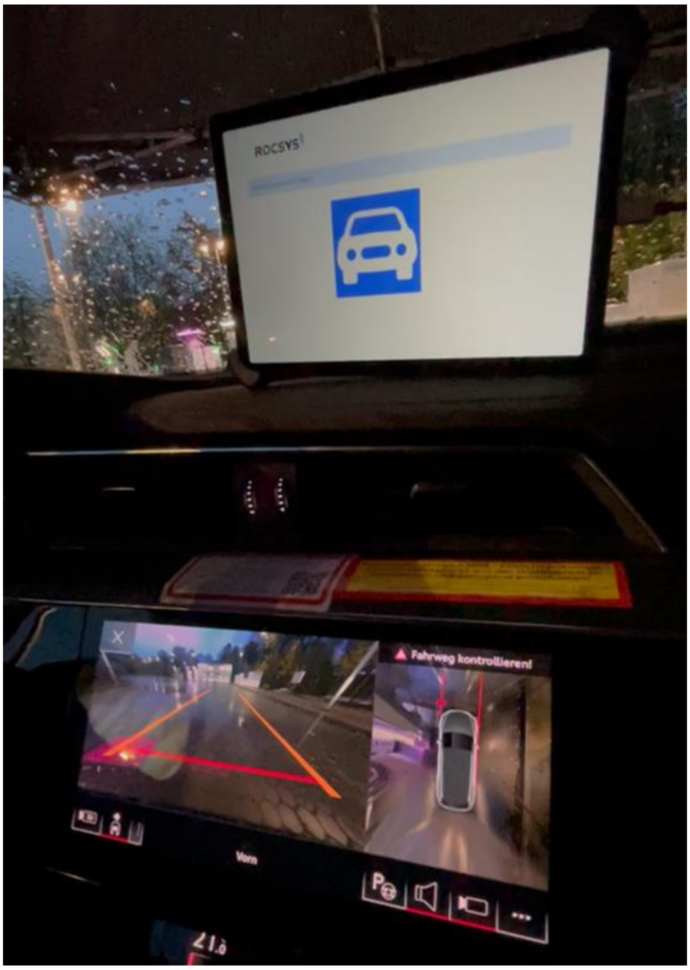


Figure 6: Sample for parking guidance feedback

4.7.5 Automated flap

The following requirements shall be considered for the automated flap:

- Mating Space according to IEC 61851-27

- Package Space according to ISO 5474-5
- Configurable speed for opening and closing the flap
- Flap status detection (e.g. fully opened, fully closed) within vehicle

4.7.6 Absence of DC cover

Many EVs use a dust-cover on the DC Pins that needs to be removed manually before CCS DC Charging. For robot charging this cover should be waived or open automatically. The mating space for the robot needs to be taken into consideration.

4.7.7 Cable locking requirements

1. Legacy Case: Depending on the mating speed of the robot, timing issues for cable locking need to be considered and handled. Unlock must be triggered manually.
2. With EV-Robot communication: EV applies lock after docking confirmation from robot. EV unlocks plug after unlocking signal via robot-EV communication.

4.8 Charger Requirements

4.8.1 Project assumptions

- The Charging station and robot are separate systems
- Charging system: CCS2 High Power only – 350kW
- For the HMI the standard User Flow and the standard charging processes will be used for the project
- Where applicable and possible standardized components should be used

4.8.2 Exception handling

Exceptions are defined and used from the standard process (e.g Plug and Charge).

4.8.3 Cable length and cable management

The cable shall not block or stand in the way of user acting range. It needs to be mounted in a way that it will not touch the ground and/or handled by a cable management system.

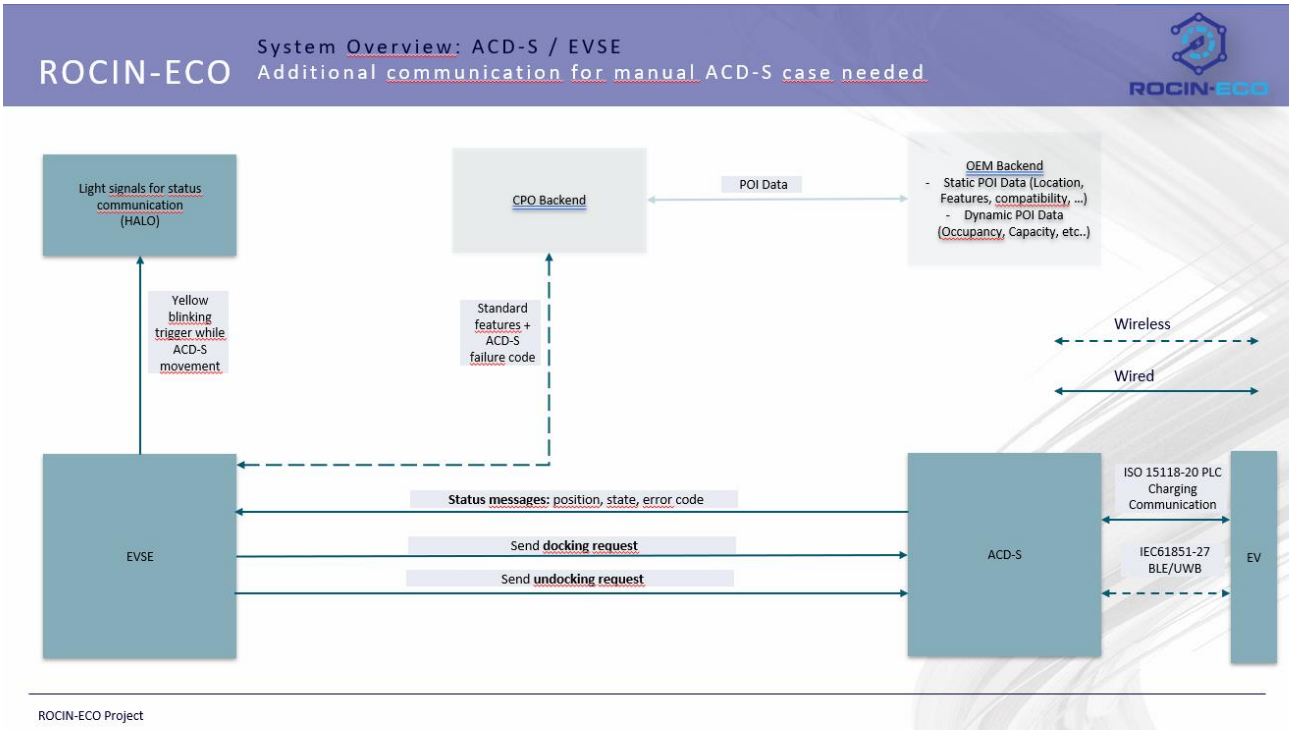
4.9 Robot-Charger-Backend Communication

4.9.1 Project assumptions

- The charger that will be used for project evaluations is connected to the IONITY backend via OCPP 1.6
- Within the project the robot will not be connected to any backend or having any communication to the charger. It will run as a standalone unit.
- The necessity and options for the robot to be connected to a backend or necessity for robot-charger communication need to be evaluated.

4.9.2 Use Cases for Backend Communication

- Manual trigger docking/undocking by Charger or Backend
- Exchange of status and failure codes
- Remote/Manual override and reset
- Key exchange for BLE/UWB communication

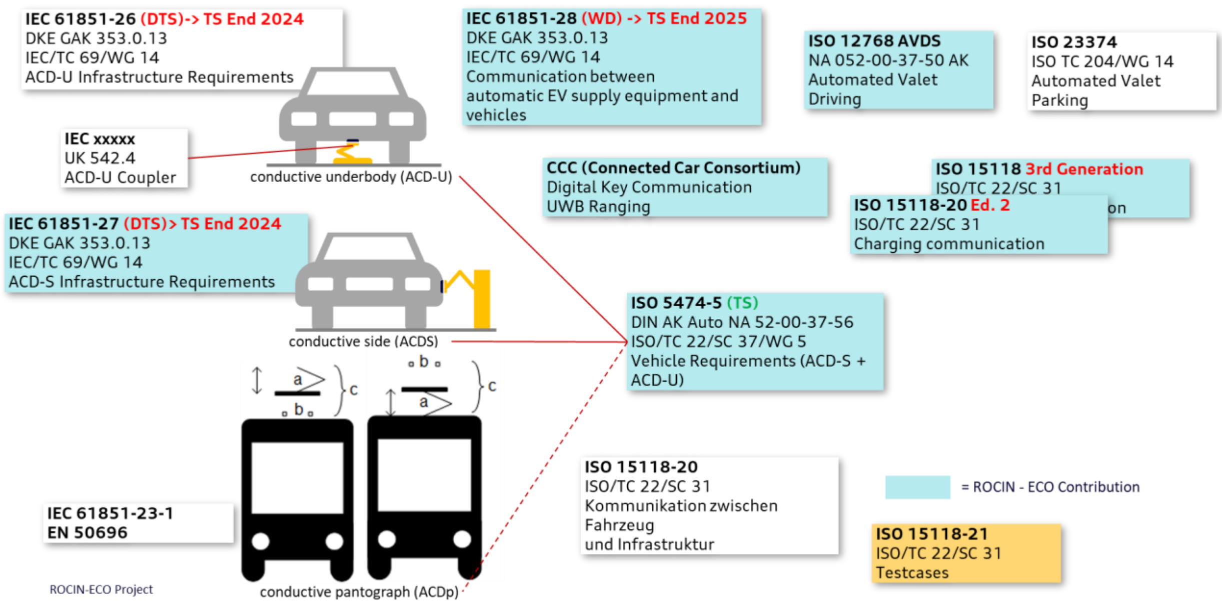


4.9.3 Roboter Backendintegration

Robot should be integrated directly into the backend as own OCPP Client to serve the use cases described above. The current OCPP version should be used. Custom messages needs to be defined.

5. Applicable standards

- IEC 61851-27
- IEC 61851-28
- ISO 5474-5
- ISO 12100 (for Installation)
- IEC 60364-7-722
- ISO15118-20
- ISO 12768 – AVDS
- CCC Version 3 – BLE/UWB



6. Annex

A – Message Structure

The following message structure have been created within the project as a proposal for ISO 15118:

```
<!--                              -->
<!-- ACD Prepare Arrival    -->
<!--                              -->
<xs:element name="ACD_C_PRE_Req" type="ACD_C_PRE_ReqType"/>
<xs:complexType name="ACD_PRE_ReqType">
        <xs:complexContent>
                <xs:extension base="v2gci_ct:V2GRequestType">
                        <xs:sequence>
                                <!-- Enter parameters here -->
                                <xs:element name="EV_InletPos" type="ACD_EV_InletPosType"
minOccurs="1" maxOccurs="4"/>   <!-- EV might have more inlets ... -->
                                <xs:element name="Reservation_IDCode" type="ACD_IDCodeType"
minOccurs="0"/>
                                <xs:element name="handicapped" type="xs:boolean"
minOccurs="0"/>
                                <!-- PnC ability -->


                                <!-- Pairing methods -->
                                <xs:element name="ACD_PairingMethodList"
type="ACD_PairingMethodListType"/>   <!-- should contain at least one entry (->external
confirmation)... -->
                                <xs:element name="Connector" type="ACD_ConnectorType" />
                                <xs:element name="VendorSpecificDataContainer"
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>
                        </xs:sequence>
                </xs:extension>
        </xs:complexContent>
</xs:complexType>

<xs:element name="ACD_R_PRE_Res" type="ACD_R_PRE_ResType"/>
<xs:complexType name="ACD_R_PRE_ResType">
        <xs:complexContent>
                <xs:extension base="v2gci_ct:V2GResponseType">
                        <xs:sequence>
                                <!-- Enter parameters here -->
```

```
                              <xs:element name="EVSEProcessing"
type="v2gci_ct:processingType"/>  <!-- finished -> ACD_R_PCO -->

                              <xs:element name="ACDStation_VacancyList"
type="ACD_ACDStationListType" minOccurs="0"/>

                              <xs:element name="VendorSpecificDataContainer"
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>

                        </xs:sequence>

                  </xs:extension>

            </xs:complexContent>

      </xs:complexType>



      <!--                            -->

      <!-- ACD Parking                -->

      <!--                            -->

      <xs:element name="ACD_ParkingReq" type="ACD_ParkingReqType"/>

      <xs:complexType name="ACD_ParkingReqType">

            <xs:complexContent>

                  <xs:extension base="v2gci_ct:V2GRequestType">

                        <xs:sequence>

                              <xs:element name="EVProcessing"
type="v2gci_ct:processingType"/>  <!-- finished -> ACD_R_DCO -->

                              <xs:element name="EVResultCode"
type="ACD_EVResultType"/>

                              <xs:element name="VendorSpecificDataContainer"
type="WPT_DataContainerType" minOccurs="0" maxOccurs="16"/>

                        </xs:sequence>

                  </xs:extension>

            </xs:complexContent>

      </xs:complexType>

      <xs:element name="ACD_ParkingRes" type="ACD_ParkingResType"/>

      <xs:complexType name="ACD_ParkingResType">

            <xs:complexContent>

                  <xs:extension base="v2gci_ct:V2GResponseType">

                        <xs:sequence>

                              <xs:element name="EVSEProcessing"
type="v2gci_ct:processingType"/>

                              <xs:element name="VendorSpecificDataContainer"
type="WPT_DataContainerType" minOccurs="0" maxOccurs="16"/>

                              <!-- add additional information depending on the parking
guidance method used -->
```

```
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <!--                                    -->
            <!-- ACD Pairing (ACD_I_PAI, ACD_R_PAI) -->
            <!--                                    -->
            <xs:element name="ACD_PairingReq" type="ACD_PairingReqType"/>
            <xs:complexType name="ACD_PairingReqType">
                <xs:complexContent>
                    <xs:extension base="v2gci_ct:V2GRequestType">
                        <xs:sequence>
                            <xs:element name="EVProcessing"
type="v2gci_ct:processingType"/>
                            <xs:element name="ObservedIDCode" type="ACD_IDCodeType"
minOccurs="0"/>
                            <xs:element name="EVResultCode"
type="ACD_EVResultType"/>
                            <xs:element name="VendorSpecificDataContainer"
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:element name="ACD_PairingRes" type="ACD_PairingResType"/>
            <xs:complexType name="ACD_PairingResType">
                <xs:complexContent>
                    <xs:extension base="v2gci_ct:V2GResponseType">
                        <xs:sequence>
                            <xs:element name="EVSEProcessing"
type="v2gci_ct:processingType"/>
                            <xs:element name="ObservedIDCode"
type="v2gci_ct:numericIDType" minOccurs="0"/>
                            <xs:element name="AlternativeSECCList"
type="AlternativeSECCListType" minOccurs="0"/>
                            <xs:element name="VendorSpecificDataContainer"
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>
                        </xs:sequence>
                    </xs:extension>
```

```
		</xs:complexContent>
	</xs:complexType>
	<!--                                                                  -->
	<!-- ACD Docking   (ACD_C_DOC, ACD_R_DOC)              -->
	<!--                                                                  -->
	<xs:element name="ACD_DockingReq" type="ACD_DockingReqType"/>
	<xs:complexType name="ACD_DockingReqType">
		<xs:complexContent>
			<xs:extension base="v2gci_ct:V2GRequestType">
				<xs:sequence>
					<xs:element name="EVProcessing"
type="v2gci_ct:processingType"/>
					<xs:element name="EV_InletPos"
type="ACD_EV_InletPosType" minOccurs="1"/>
					<xs:element name="EVResultCode"
type="ACD_EVResultType"/>
					<xs:element name="VendorSpecificDataContainer"
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>
				</xs:sequence>
			</xs:extension>
		</xs:complexContent>
	</xs:complexType>
	<xs:element name="ACD_DockingRes" type="ACD_DockingResType"/>
	<xs:complexType name="ACD_DockingResType">
		<xs:complexContent>
			<xs:extension base="v2gci_ct:V2GResponseType">
				<xs:sequence>
					<xs:element name="EVSEProcessing"
type="v2gci_ct:processingType"/>
					<xs:element name="VendorSpecificDataContainer"
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>
				</xs:sequence>
			</xs:extension>
		</xs:complexContent>
	</xs:complexType>
	<!--                          -->
	<!-- ACD Motion Following     -->
	<!--                          -->
```

```
	<!-- while mechanically connected the EV requests the ACD to follow little
movements of the EV		-->

	<!-- Request is periodically send as long as connector shall keep mated
						-->

	<xs:element name="ACD_MotFolReq" type="ACD_MotFolReqType"/>

	<xs:complexType name="ACD_MotFolReqType">

		<xs:complexContent>

			<xs:extension base="v2gci_ct:V2GRequestType">

				<xs:sequence>

					<xs:element name="EVProcessing"
type="v2gci_ct:processingType"/>

					<xs:element name="EV_PowerTransferStatus"
type="ACD_EV_PowerTransferStatusType" minOccurs="1"/>

					<xs:element name="EV_Locking_Status"
type="ACD_LockingStatusType" minOccurs = "1"/> <!-- in case locking is controlled by EV -
->

					<xs:element name="VendorSpecificDataContainer"
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>

				</xs:sequence>

			</xs:extension>

		</xs:complexContent>

	</xs:complexType>

	<xs:element name="ACD_MotFolRes" type="ACD_MotFolResType"/>

	<xs:complexType name="ACD_MotFolResType">

		<xs:complexContent>

			<xs:extension base="v2gci_ct:V2GResponseType">

				<xs:sequence>

					<xs:element name="EVSEProcessing"
type="v2gci_ct:processingType"/>

					<xs:element name="EVSE_Locking_Status"
type="ACD_LockingStatusType" minOccurs = "1"/>  <!-- in case locking is controlled by
EVSE -->

					<xs:element name="VendorSpecificDataContainer"
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>

				</xs:sequence>

			</xs:extension>

		</xs:complexContent>

	</xs:complexType>


	<!--										-->
```

```
<!-- ACD Undocking   (ACD_C_UND, ACD_R_UND)         -->
<!--                                                              -->
<xs:element name="ACD_UndockingReq" type="ACD_UndockingReqType"/>
<xs:complexType name="ACD_UndockingReqType">
	<xs:complexContent>
		<xs:extension base="v2gci_ct:V2GRequestType">
			<xs:sequence>
				<xs:element name="EVProcessing" 
type="v2gci_ct:processingType"/>
				<xs:element name="EVResultCode" 
type="ACD_EVResultType"/>
				<xs:element name="VendorSpecificDataContainer" 
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>
			</xs:sequence>
		</xs:extension>
	</xs:complexContent>
</xs:complexType>
<xs:element name="ACD_UndockingRes" type="ACD_UndockingResType"/>
<xs:complexType name="ACD_DockingResType">
	<xs:complexContent>
		<xs:extension base="v2gci_ct:V2GResponseType">
			<xs:sequence>
				<xs:element name="EVSEProcessing" 
type="v2gci_ct:processingType"/> <!-- "finished" when homeposition is reached -->
				<xs:element name="VendorSpecificDataContainer" 
type="ACD_DataContainerType" minOccurs="0" maxOccurs="16"/>
			</xs:sequence>
		</xs:extension>
	</xs:complexContent>
</xs:complexType>

<!--                        -->
<!-- ---------------------- -->
<!-- Message Specific Types -->
<!-- ---------------------- -->
```

```xml
<xs:simpleType name="ACD_EV_PowerTransferStatusType">
	<xs:restriction base="xs:string">
		<xs:enumeration value="PowertransferInit"/>
		<xs:enumeration value="PowertransferActive"/>

		<xs:enumeration value="PowertransferStopped"/>
	</xs:restriction>
</xs:simpleType>

<xs:simpleType name="ACD_LockingStatusType">
	<xs:restriction base="xs:string">
		<xs:enumeration value="Locked"/>
		<xs:enumeration value="Unlocked"/>
		<xs:enumeration value="NotApplicable"/>
		<xs:enumeration value="Failure"/>
	</xs:restriction>
</xs:simpleType>

<xs:simpleType name="ACD_EV_InletPosType">
	<xs:restriction base="xs:string">
		<xs:enumeration value="FL"/>  <!-- Front left   -->
		<xs:enumeration value="FR"/>  <!-- Front right  -->
		<xs:enumeration value="RL"/>  <!-- Rear left    -->
		<xs:enumeration value="RR"/>  <!-- Rear right   -->
		<xs:enumeration value="FC"/>  <!-- Front center -->
		<xs:enumeration value="RC"/>  <!-- Rear center  -->
	</xs:restriction>
</xs:simpleType>

<xs:simpleType name="ACD_EVResultType">
	<xs:restriction base="xs:string">
		<xs:enumeration value="EVResultUnknown"/>
		<xs:enumeration value="EVResultSuccess"/>
		<xs:enumeration value="EVResultFailed"/>
	</xs:restriction>
</xs:simpleType>
```

```
	<xs:complexType name="ACD_IDCodeType">
		<xs:sequence>
			<xs:element name="ACD_IDCode" type="v2gci_ct:numericIDType"/>
		</xs:sequence>
	</xs:complexType>

	<xs:complexType name="ACD_ACDStationListType">
		<xs:sequence>
			<xs:element name="ACD_ACDStationData" type="ACD_ACDStationDataType"/>
		</xs:sequence>
	</xs:complexType>
	<xs:complexType name="ACD_ACDStationDataType">
		<xs:sequence>
				<xs:element name="ACD_IDCode" type="ACD_IDCodeType"/>
				<!-- add more relevant data here -->
				<!-- handicapped accessible -->
				<xs:element name="ACD_handicapped" type="xs:boolean"/>
				<xs:element name="Connector" type="ACD_ConnectorType" /> <!--
CCS2, Chademo, Type2, .....  -->

				<!-- Parking Orientation: Depends on the Inlet Position of the
vehicle, which requests for an ACD Station -->
				<xs:element name="ParkingOrientation"
type="ACD_ParkingOrientationType" minOccurs="1" />
		</xs:sequence>
	</xs:complexType>

	<xs:simpleType name="ACD_ParkingOrientationType">
		<xs:restriction base="xs:string">
			<xs:enumeration value="forward"/>
			<xs:enumeration value="backward"/>
			<xs:enumeration value="NotApplicable"/>
		</xs:restriction>
	</xs:simpleType>

	<xs:simpleType name="ACD_ConnectorType">
```

```
            <xs:restriction base="xs:string">
                  <xs:enumeration value="CCS2"/>
                  <xs:enumeration value="Type2"/>
                  <xs:enumeration value="Chademo"/>
            </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="ACD_PairingMethodType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="External confirmation"/>      <!-- User must do
pairing interactively, no auotmated pairing -->
                  <xs:enumeration value="Optical"/>                             <!-
- vehicle has ability to read sign or marker -->
                  <xs:enumeration value="UWB"/>                                 <!-
- vehicle has ability to pair via UWB channel -->
                  <xs:enumeration value="Proprietary"/>
            </xs:restriction>
      </xs:simpleType>

      <xs:complexType name="ACD_PairingMethodListType">
            <xs:sequence>
                  <xs:element name="ACD_PairingMethod" type="ACD_PairingMethodType"
maxOccurs="8"/>
            </xs:sequence>
      </xs:complexType>

<!--                         -->
<!--   ACD Service selection -->
<!--                         -->

      <xs:element name="ACD_SelectedService" type="ACD_SelectedServiceType"
substitutionGroup="SelectedServiceType"/>
      <xs:complexType name="ACD_SelectedServiceType">
            <xs:complexContent>
                  <xs:extension base="SelectedServiceTypeType">
                        <xs:sequence>
                              <xs:element name="ServiceVoucher_IDCode"
type="ServiceVoucher_IDCodeType" minOccurs="0"/>
```

```
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:complexType name="ServiceVoucher_IDCodeType">
            <xs:sequence>
                <xs:element name="Voucher_IDCode" type="v2gci_ct:numericIDType"/>
            </xs:sequence>
        </xs:complexType>

        <!--AlternativeSECCTYpe is connection information for an alternative SECC-->
        <xs:complexType name="AlternativeSECCType">
            <xs:sequence>
                <xs:element name="SSID" type="v2gci_ct:identifierType"
minOccurs="0"/>
                <xs:element name="BSSID" type="bssidType" minOccurs="0"/>
                <xs:element name="IPAddress" type="ipaddressType" minOccurs="0"/>
                <xs:element name="Port" type="xs:unsignedShort" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>

        <!--BSSIDType: hexa-decimal string representation of the MAC address of the AP in
upper case-->
        <xs:simpleType name="bssidType">
            <xs:restriction base="xs:string">
                <xs:maxLength value="12"/>
            </xs:restriction>
        </xs:simpleType>

        <!--IPAddressType: hexa-decimal string representation of the IPv6 address of the
SECC
        including colons, in upper case-->
        <xs:simpleType name="ipaddressType">
            <xs:restriction base="xs:string">
                <xs:maxLength value="39"/>
            </xs:restriction>
        </xs:simpleType>
```

```
	<!-- --------------   -->
	<!-- Simple ACD Types -->
	<!-- --------------   -->
	<xs:simpleType name="ACD_DataContainerType">
		<xs:restriction base="xs:base64Binary">
			<xs:maxLength value="256"/>
		</xs:restriction>
	</xs:simpleType>
</xs:schema>
```