



EEBus UC Technical Specification

Coordinated EV Charging

Version 1.0.1

Cologne, 2019-07-01

EEBus Initiative e.V.

Butzweilerhof Allee 4
50829 Cologne
GERMANY

Rue d'Arlon 25
1050 Brussels
BELGIUM

Phone: +49 221 / 47 44 12 - 20
Fax: +49 221 / 47 44 12 - 1822

info@eebus.org
www.eebus.org

District court: Cologne
VR 17275

Terms of use for publications of EEBus Initiative e.V.**General information**

The specifications, particulars, documents, publications and other information provided by the EEBus Initiative e.V. are solely for general informational purposes. Particularly specifications that have not been submitted to national or international standardisation organisations by EEBus Initiative e.V. (such as DKE/DIN-VDE or IEC/CENELEC/ETSI) are versions that have not yet undergone complete testing and can therefore only be considered as preliminary information. Even versions that have already been published via standardisation organisations can contain errors and will undergo further improvements and updates in future.

Liability

EEBus Initiative e.V. does not assume liability or provide a guarantee for the accuracy, completeness or up-to-date status of any specifications, data, documents, publications or other information provided and particularly for the functionality of any developments based on the above.

Copyright, rights of use and exploitation

The specifications provided are protected by copyright. Parts of the specifications have been submitted to national or international standardisation organisations by EEBus Initiative e.V., such as DKE/DIN-VDE or IEC/CENELEC/ETSI, etc. Furthermore, all rights to use and/or exploit the specifications belong to the EEBus Initiative e.V., Butzweilerhof-Allee 4, 50829 Cologne, Germany and can be used in accordance with the following regulations:

The use of the specifications for informational purposes is allowed. It is therefore permitted to use information evident from the contents of the specifications. In particular, the user is permitted to offer products, developments and/or services based on the specifications.

Any respective use relating to standardisation measures by the user or third parties is prohibited. In fact, the specifications may only be used by EEBus Initiative e.V. for standardisation purposes. The same applies to their use within the framework of alliances and/or cooperations that pursue the aim of determining uniform standards.

Any use not in accordance with the purpose intended by EEBus Initiative e.V. is also prohibited.

Furthermore, it is prohibited to edit, change or falsify the content of the specifications. The dissemination of the specifications in a changed, revised or falsified form is also prohibited. The same applies to the publication of extracts if they distort the literal meaning of the specifications as a whole.

It is prohibited to pass on the specifications to third parties without reference to these rights of use and exploitation.

It is also prohibited to pass on the specifications to third parties without informing them of the authorship or source.

Without the prior consent of EEBus Initiative e.V., all forms of use and exploitation not explicitly stated above are prohibited.

Table of contents

Table of contents.....	3
List of figures	4
List of tables	5
1 Scope of the document	7
1.1 References.....	7
1.1.1 EEBUS documents	7
1.1.2 Normative references.....	7
1.2 Terms and definitions.....	7
1.3 Requirements	8
1.3.1 Requirements wording.....	8
1.3.2 Mapping of High-Level requirements.....	8
2 High-Level description.....	9
2.1 Introduction.....	9
2.2 User Story as an example	10
2.3 Actors	10
2.3.1 EV.....	10
2.3.2 Energy Guard	11
2.3.3 Energy Broker	11
2.4 Scenarios	11
2.4.1 Scenario 1 - EV sends charging energy demand to Energy Broker and Energy Guard..	12
2.4.2 Scenario 2 - Energy Guard sends maximum power limitation curve to EV.....	14
2.4.3 Scenario 3 - Energy Broker sends incentive table to EV.....	15
2.4.4 Scenario 4 - EV sends charging plan curve to Energy Broker and Energy Guard	18
2.4.5 Scenario 5 - Energy Guard heartbeat	20
2.4.6 Scenario 6 - Energy Broker heartbeat	20
2.4.7 Scenario 7 - Energy Guard error state.....	21
2.4.8 Scenario 8 - Energy Broker error state	21
2.5 Dependencies to other Use Cases.....	22
2.5.1 "EV Commissioning and Configuration"	22
2.5.2 "EV Charging Electricity Measurement"	22
2.5.3 "Overload Protection by EV Charging Current Curtailment"	23
2.6 Assumptions and Prerequisites.....	23
3 Technical SPINE solution	24
3.1 General rules and information	24
3.1.1 Underlying technology documents	24
3.1.2 Use Case Discovery rules.....	24
3.1.3 Rules for "Content of Specialization..." tables and "Content of Function..." tables	25
3.1.4 Rules for "Feature Types and Functions..." tables	31
3.1.5 "Actor ... overview" diagram rules	32
3.1.6 Specializations	33
3.1.7 Order of messages within the sequence diagrams	34
3.1.8 Further information and rules.....	34
3.2 Actors	34
3.2.1 EV.....	34
3.2.2 Energy Guard	52

49	3.2.3	Energy Broker	63
50	3.3	Pre-Scenario communication	78
51	3.3.1	General information	78
52	3.3.2	Detailed discovery	79
53	3.3.3	Binding.....	81
54	3.3.4	Subscription.....	81
55	3.3.5	Dynamic behaviour.....	82
56	3.4	Scenarios	82
57	3.4.1	Scenario 1 - EV sends charging energy demand to Energy Broker and Energy Guard..	82
58	3.4.2	Scenario 2 - Energy Guard sends maximum power limitation curve to EV.....	86
59	3.4.3	Scenario 3 - Energy Broker sends incentive table to EV.....	90
60	3.4.4	Scenario 4 - EV sends charging plan curve to Energy Broker and Energy Guard	93
61	3.4.5	Scenario 5 - Energy Guard heartbeat	97
62	3.4.6	Scenario 6 - Energy Broker heartbeat	99
63	3.4.7	Scenario 7 - Energy Guard error state	101
64	3.4.8	Scenario 8 - Energy Broker error state	103
65			

66 List of figures

67	Figure 1: High-Level Use Case functionality overview	9
68	Figure 2: Charging session example	10
69	Figure 3: Scenario overview	11
70	Figure 4: Energy demand example.....	13
71	Figure 5: Maximum power limitation curve example	14
72	Figure 6: Incentive table example with consideration of the P_{\max} curve from Scenario 2.....	16
73	Figure 7: IncentiveTable example (single incentiveSlot) for different tiers	17
74	Figure 8: Charging plan curve example	19
75	Figure 9: Actor overview example.....	33
76	Figure 10: Actor "EV" overview	35
77	Figure 11: Actor "Energy Guard" overview	52
78	Figure 12: Actor "Energy Broker" overview	63
79	Figure 13: Pre-Scenario communication - Detailed discovery sequence diagram.....	80
80	Figure 14: Pre-Scenario communication - Binding sequence diagram	81
81	Figure 15: Pre-Scenario communication - Subscription sequence diagram	82
82	Figure 16: Scenario 1 - Initial Scenario communication sequence diagram	83
83	Figure 17: Scenario 1 - Runtime Scenario communication sequence diagram.....	85
84	Figure 18: Scenario 2 - Initial Scenario communication sequence diagram	87
85	Figure 19: Scenario 2 - Runtime Scenario communication sequence diagram.....	89
86	Figure 20: Scenario 3 - Initial Scenario communication sequence diagram	91
87	Figure 21: Scenario 3 - Runtime Scenario communication sequence diagram.....	92
88	Figure 22: Scenario 4 - Initial Scenario communication sequence diagram	94
89	Figure 23: Scenario 4 - Runtime Scenario communication sequence diagram.....	96
90	Figure 24: Scenario 5 - Initial Scenario communication sequence diagram	98
91	Figure 25: Scenario 5 - Runtime Scenario communication sequence diagram.....	98
92	Figure 26: Scenario 6 - Initial Scenario communication sequence diagram	100

93	Figure 27: Scenario 6 - Runtime Scenario communication sequence diagram.....	101
94	Figure 28: Scenario 7 - Initial Scenario communication sequence diagram	102
95	Figure 29: Scenario 7 - Runtime Scenario communication sequence diagram.....	103
96	Figure 30: Scenario 8 - Initial Scenario communication sequence diagram	104
97	Figure 31: Scenario 8 - Runtime Scenario communication sequence diagram.....	105

98

99 List of tables

100	Table 1: Scenario implementation requirement for Actors	12
101	Table 2: Presence indication description	25
102	Table 3: Example table for cardinality indications	27
103	Table 4: Presence indication of Feature Types and Functions support	31
104	Table 5: Feature Types and Functions used within this Use Case by the Actor EV	36
105	Table 6: Content of Function "timeSeriesDescriptionListData" at Actor EV	38
106	Table 7: Content of Function "timeSeriesConstraintsListData" at Actor EV	41
107	Table 8: Content of Function "timeSeriesListData" at Actor EV.....	45
108	Table 9: Content of Function "incentiveTableDescriptionData" at Actor EV	48
109	Table 10: Content of Function "incentiveTableConstraintsData" at Actor EV	48
110	Table 11: Content of Function "incentiveTableData" at Actor EV	51
111	Table 12: Content of Specialization "DeviceDiagnosis_FailureState" at Actor EV	51
112	Table 13: Content of Specialization "Heartbeat_Timeout60Seconds" at Actor EV	52
113	Table 14: Feature Types and Functions used within this Use Case by the Actor Energy Guard.....	53
114	Table 15: Content of Function "deviceDiagnosisHeartbeatData" at Actor Energy Guard	54
115	Table 16: Content of Function "deviceDiagnosisStateData" at Actor Energy Guard	54
116	Table 17: Content of Specialization "TimeSeries_AcEnergySingleDemand" at Actor Energy Guard	57
117	Table 18: Content of Specialization "TimeSeries_AcPowerMaxWriteable" at Actor Energy Guard	60
118	Table 19: Content of Specialization "TimeSeries_AcMaxPowerPlan" at Actor Energy Guard	63
119	Table 20: Feature Types and Functions used within this Use Case by the Actor Energy Broker	64
120	Table 21: Content of Function "deviceDiagnosisHeartbeatData" at Actor Energy Broker	65
121	Table 22: Content of Function "deviceDiagnosisStateData" at Actor Energy Broker	65
122	Table 23: Content of Specialization "TimeSeries_AcEnergySingleDemand" at Actor Energy Broker ...	68
123	Table 24: Content of Specialization "TimeSeries_AcPowerMaxWriteable" at Actor Energy Broker	69
124	Table 25: Content of Specialization "TimeSeries_AcMaxPowerPlan" at Actor Energy Broker	72
125	Table 26: Content of Specialization "IncentiveTable_SimpleIncentiveTableWriteable" at Actor Energy	
126	Broker	78
127	Table 27: Initial Scenario communication content references for Scenario 1	84
128	Table 28: Runtime Scenario communication content references for Scenario 1	86
129	Table 29: Initial Scenario communication content references for Scenario 2	88
130	Table 30: Runtime Scenario communication content references for Scenario 2	90
131	Table 31: Initial Scenario communication content references for Scenario 3	91
132	Table 32: Runtime Scenario communication content references for Scenario 3	93
133	Table 33: Initial Scenario communication content references for Scenario 4	95
134	Table 34: Runtime Scenario communication content references for Scenario 4	97
135	Table 35: Initial Scenario communication content references for Scenario 5	98
136	Table 36: Runtime Scenario communication content references for Scenario 5	99

137	Table 37: Initial Scenario communication content references for Scenario 6	100
138	Table 38: Runtime Scenario communication content references for Scenario 6	101
139	Table 39: Initial Scenario communication content references for Scenario 7	102
140	Table 40: Runtime Scenario communication content references for Scenario 7	103
141	Table 41: Initial Scenario communication content references for Scenario 8	104
142	Table 42: Runtime Scenario communication content references for Scenario 8	105
143		
144		

1 Scope of the document

This document describes the Use Case "Coordinated EV Charging" (short-name: CEVC). Chapter 2 specifies the High-Level Use Case. Chapter 3 describes the technical solution for SPINE for this Use Case in detail. Within this document a top-down approach is used to derive the requirements for the technical solution from the High-Level description.

1.1 References

1.1.1 EEBUS documents

[UseCaseBaseSpecification] EEBus_UC_TS_UseCaseBaseSpecification.pdf

[ProtocolSpecification] EEBus_SPINE_TS_ProtocolSpecification.pdf

[ResourceSpecification] EEBus_SPINE_TS_ResourceSpecification.pdf

[SHIP] SHIP_Specification_v1.0.0.pdf

1.1.2 Normative references

[RFC2119] IETF RFC 2119: 1997, Key words for use in RFCs to indicate requirement levels
Please see section 1.3 for details.

[ISO15118] Standard for communication protocol between EVSE and EV

[IEC61851-1] Standard for PWM communication between EVSE and EV

1.2 Terms and definitions

Actor

An Actor models a role within a Use Case definition (e.g. an energy manager or an electric vehicle).

CEM

Abbreviation for Customer Energy Manager. The CEM is an energy manager located at the home or premises of the user or in a cloud application. The energy manager enables energy-optimized operation of the connected devices by harmonizing energy demand and availability.

CEVC

Coordinated EV Charging (short name of this Use Case)

DSO

Abbreviation for (Energy) Distribution System Operator

EV

Abbreviation for Electric Vehicle

EVSE

Abbreviation for Electric Vehicle Supply Equipment

179 PV

180 Abbreviation for Photovoltaic (system)

181 PWM

182 Abbreviation for Pulse Width Modulation

183 Scenario

184 Part of the Use Case. Splitting a Use Case in Scenarios helps to understand the Use Case more
185 quickly. Some Scenarios are mandatory for a Use Case, whereas others may be recommended or
186 optional.

187 Specialization

188 Reusable data collection for a specific functionality.

189 SPINE

190 **S**mart **P**remises **I**nteroperable **N**eutral-message **E**xchange: Technical Specification of EEBus Initiative
191 e.V.

192

193 1.3 Requirements**194 1.3.1 Requirements wording**

195 The following keywords are used:

- 196 - SHALL
- 197 - SHALL NOT
- 198 - SHOULD
- 199 - SHOULD NOT
- 200 - MAY

201 Note: They apply only if written in capital letters.

202 For the meaning of the keywords, please refer to [RFC2119].

203

204 1.3.2 Mapping of High-Level requirements

205 Within the High-Level Use Case description, the following abbreviation is used:

206 [CEVC-xyz]

207 e.g.: [CEVC-007]

208 The abbreviation is used to mark High-Level requirements or rules of this Use Case with a unique
209 number xyz. Those requirements are referenced throughout the technical solution to show how each
210 High-Level requirement is realised in the technical part.

211

2 High-Level description

2.1 Introduction

This Use Case focuses on the adjustment of the charging process of an electric vehicle (EV) such that limitations of the customer's local grid or optimization goals are met.

Examples for optimization goals are:

- the reduction of the electricity costs associated with a charging process
- the reduction of the CO₂-footprint associated with a charging process
- the compliance with constraints of the energy supplier or DSO
- the coordinated realization of demand response setpoints from grid-level aggregators

To achieve the goal an Actor called "Energy Broker" can influence the consumption of an EV with incentives. For this the Energy Broker sends an incentive table that defines different tiers with power levels and incentives. Different kinds of incentives are possible, like price, CO₂ emission or renewable energy percentage. Additionally, a maximum power limitation (P_{\max}) is provided by an Actor called "Energy Guard" as a time-based curve, to avoid in advance overload situations during charging. To handle unexpected overload situations, please refer to the "Overload Protection" Use Case.

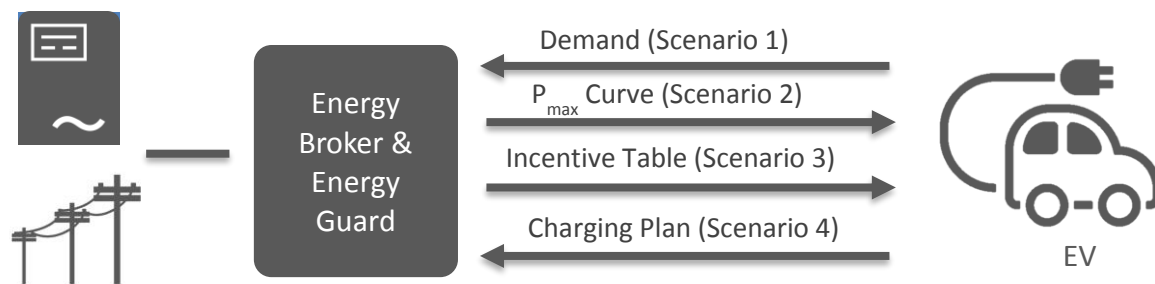


Figure 1: High-Level Use Case functionality overview

This Use Case mainly focuses on private charging infrastructure with a limited number of cars and other consumers. The incentive table and maximum power limitation (P_{\max}) curve can be mapped on the ISO 15118 tariff and P_{\max} schedule information by the EVSE to inform the EV over the charging cable. In case of PWM communication (IEC 61851-1) the EVSE has to calculate the charging plan for the EV (as PWM can only be used to communicate the maximum charging current from EVSE to EV). If PWM communication is used, the EVSE may use additional input from the driver (for example an EVSE user interface or customer app) to generate a charging energy demand.

The Energy Guard and Energy Broker are logical Actors that can be integrated into the same device (e.g. a Customer Energy Manager (CEM) that can provide both the incentives as well as the maximum power limitation), but also can be on different devices. However, it is recommended to implement both Actors on one device, as both Actors need similar knowledge. To generate an incentive table, the Energy Broker needs to have access to forecasts for the local electricity production, the electrical load and the electricity costs. To generate the maximum power limitation (P_{\max}) curve, the Energy Guard also needs to have access to forecasts for local electricity production and load.

Based on the incentive table from the Energy Broker and the maximum power limitation (P_{\max}) curve from the Energy Guard, the EV creates a charging plan, while considering its own constraints resulting from the user-configuration and other technical constraints. The charging plan contains a

maximum charging power curve of the EV and is then sent to the Energy Broker and Guard, to reserve the necessary charging power for the EV.

The EV, the Energy Guard or Energy Broker can trigger a renegotiation by sending updates:

- a) The EV can send updates of its demand and trigger an update of the incentive table and the power limitation curve (e.g. if the EV departure time is shifted).
- b) The Energy Guard can send an adjusted power limitation curve (e.g. if energy consumption or production has changed).
- c) The Energy Broker can send updated incentives (e.g. if a local photovoltaic system (PV) suddenly provides more energy as expected).

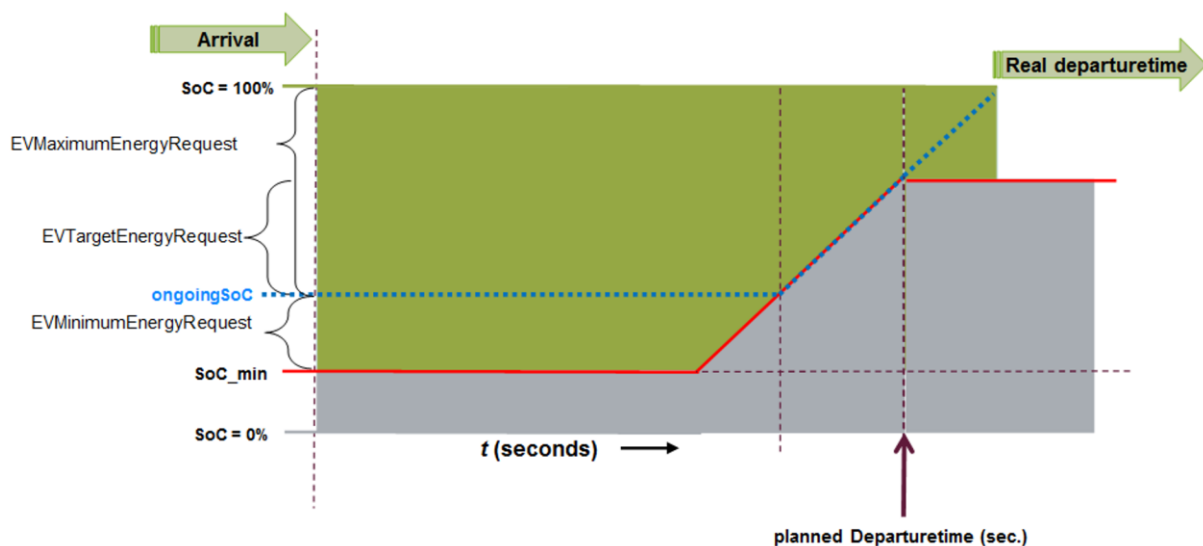


Figure 2: Charging session example

Added value: With coordinated charging an EV knows when green, cheap or costless energy is available and can charge accordingly. In the special case of PV curtailment (energy that is not consumed but also cannot be fed into the grid), surplus costless energy can be used to charge the EV for free.

2.2 User Story as an example

The driver of the EV wants to charge his EV until the next morning with minimum costs.

2.3 Actors

2.3.1 EV

The Actor EV is the electric vehicle that wants to charge with low environmental or monetary costs. Within this Use Case only one Energy Guard and Energy Broker SHALL be connected to an EV, while multiple EVs MAY be connected to an Energy Guard or Energy Broker.

2.3.2 Energy Guard

The Actor Energy Guard protects against overload situations. Multiple EVs MAY be connected to an Energy Guard. However, only one Energy Guard SHALL be connected to an EV within this Use Case.

2.3.3 Energy Broker

The Actor Energy Broker delivers incentives. Multiple EVs MAY be connected to an Energy Broker. However, only one Energy Broker SHALL be connected to an EV within this Use Case.

2.4 Scenarios

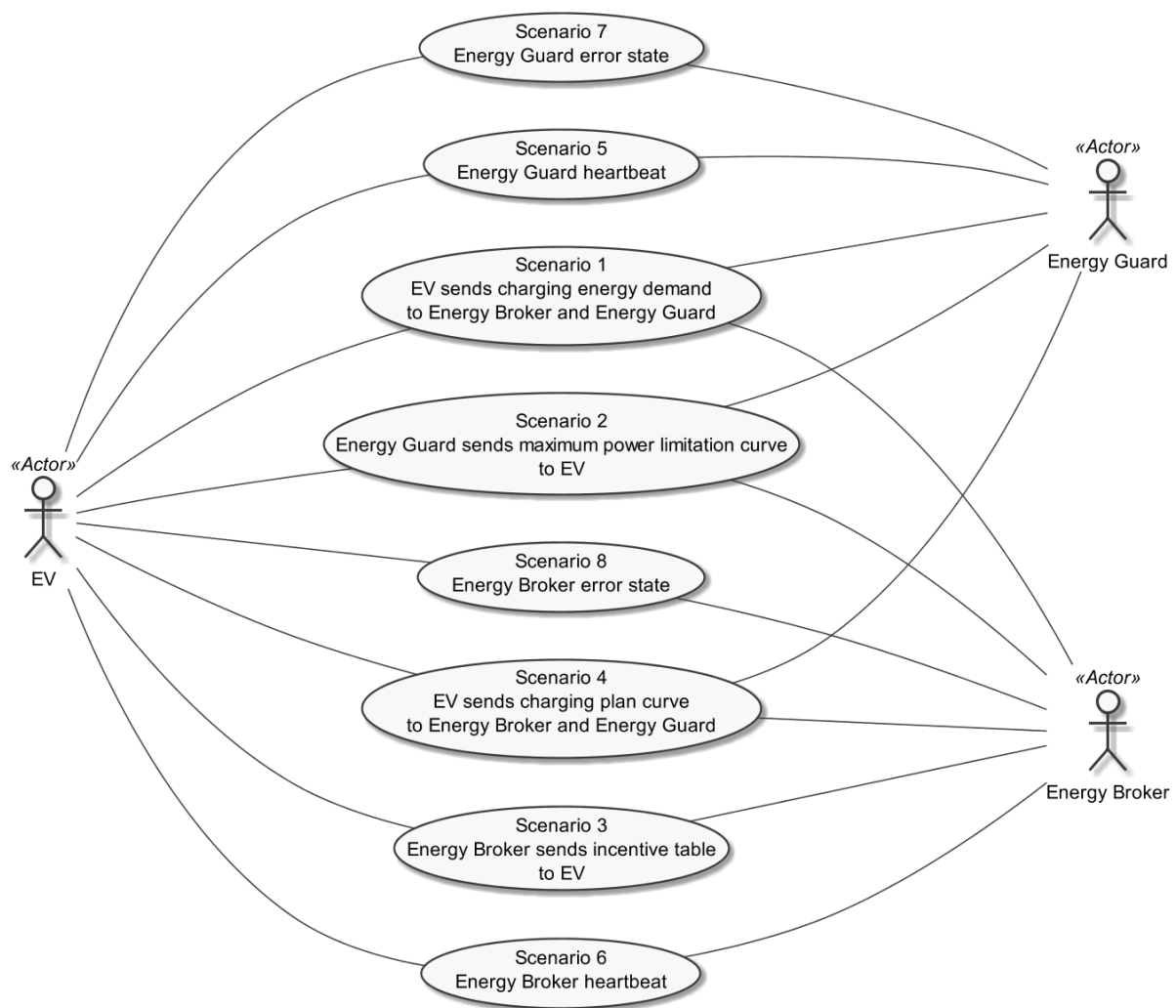


Figure 3: Scenario overview

Scenario number	Scenario name	EV	Energy Guard	Energy Broker
1	EV sends charging energy demand to Energy Broker and Energy Guard	R	M	M
2	Energy Guard sends maximum power limitation curve to EV	M	M	O
3	Energy Broker sends incentive table to EV	M	-	M
4	EV sends charging plan curve to Energy Broker and Energy Guard	M	M	M
5	Energy Guard heartbeat	M	M	-
6	Energy Broker heartbeat	M	-	M
7	Energy Guard error state	M	M	-
8	Energy Broker error state	M	-	M

Table 1: Scenario implementation requirement for Actors

The order in which the Scenarios occur is defined by triggering events specified in the Scenarios. For some Scenarios data of other Scenarios is relevant, while other Scenarios can run in parallel.

2.4.1 Scenario 1 - EV sends charging energy demand to Energy Broker and Energy Guard

2.4.1.1 Description

The EV demands energy to charge the car of the customer. Within the demand the EV may specify an arrival time [CEVC-001], a departure time [CEVC-002], the minimum energy required E_{\min} [CEVC-003], the recommended energy E_{opt} [CEVC-004] and the uncharged energy capacity E_{\max} [CEVC-005] of the EV. The Scenario itself and all communicated values are optional and can depend on the implementation of the EV as well as the driver settings. Of course, at least one value SHALL be communicated, though it is recommended to deliver as much information as possible to the Energy Broker to achieve optimal energy management results.

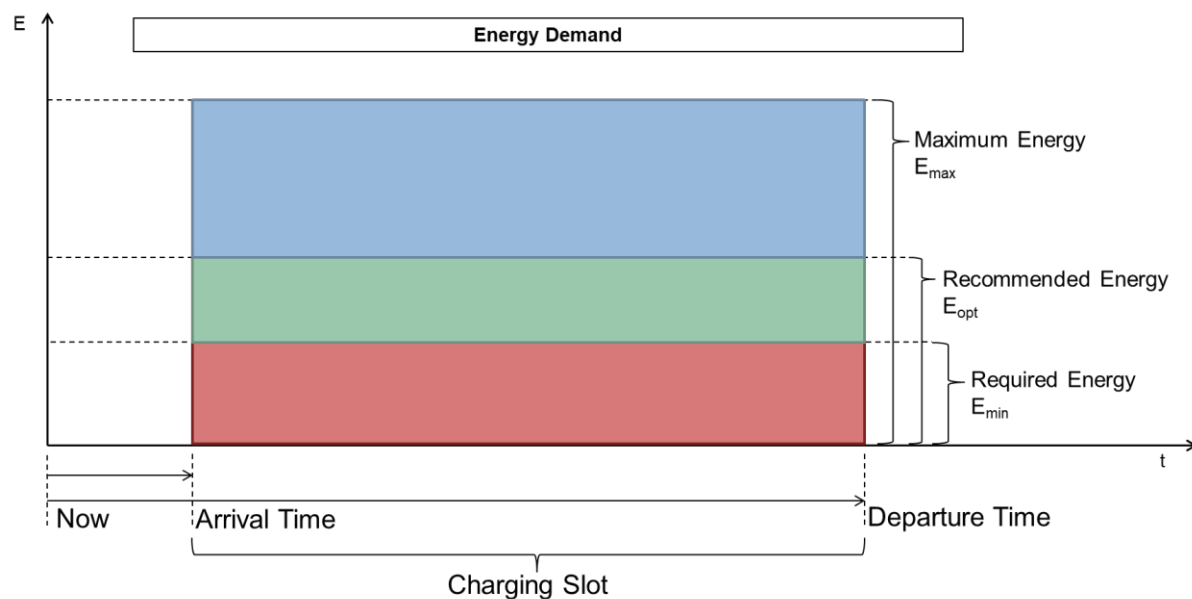


Figure 4: Energy demand example

E.g. E_{\min} could be set to an energy level sufficient to reach the next hospital, E_{opt} could be set to an energy level to reach the workplace and E_{\max} should be set to the remaining battery capacity of the EV. E_{\max} is especially valuable when e.g. a lot of costless PV surplus energy (energy above a certain curtailment factor) is available. In such a case the EV can be charged for free.

2.4.1.2 Conditions

Triggering Event:

The Scenario SHALL be triggered by connecting the EV to the EVSE and the EV wants to charge [CEVC-006]. The Scenario MAY also be triggered later during the charging process if the charging demand changes [CEVC-006]. In this case the EV SHOULD also trigger Scenario 2 and 3 after the demand was updated.

Pre-condition:

The Energy Broker and Energy Guard do not know the current energy demand, arrival time and departure time of the EV.

Post-condition:

The Energy Broker and Energy Guard know the energy demand, arrival time and departure time of the EV that shall be used in the calculation of the maximum power limitation curve and incentive table later on, when triggered. After sending the energy demand, the EV SHALL request an update of Scenario 2 and 3.

2.4.2 Scenario 2 - Energy Guard sends maximum power limitation curve to EV

2.4.2.1 Description

The Energy Guard sends a maximum charging power limitation (P_{\max}) curve [CEVC-007] that serves as limit for the charging process. The EV charging consumption SHALL NOT surpass the maximum charging power limitation (P_{\max}) curve [CEVC-008].

If a departure time was transmitted within Scenario 1, the Energy Guard SHALL deliver a maximum power limitation curve that spans at least until the departure time [CEVC-009]. If the energy provided in the maximum charging power limitation (P_{\max}) curve is not high enough to fulfil the demand of the EV (also taking into account the maximum charge power of the EV, transmitted in the Use Case "EV Commissioning and Configuration"), the Energy Guard SHALL deliver a longer maximum charging power limitation (P_{\max}) curve that MAY exceed the departure time but is able to fulfil the energy demand of the EV [CEVC-010]. If no departure time was delivered in Scenario 1, the Energy Guard SHALL deliver a maximum charging power limitation curve for at least the next 48 hours [CEVC-011].

The maximum charging power limitation (P_{\max}) curve SHOULD NOT be longer than 7 days (168h) [CEVC-012], also if the departure time is bigger than 168h. A maximum charging power limitation (P_{\max}) curve that covers a long time period MAY use a higher time resolution in the near future and a lower time resolution in the farer future. As soon as the time moves closer to the part of the curve with the lower time resolution, the curve SHOULD be updated [CEVC-013].

In general, the maximum charging power limitation (P_{\max}) curve SHALL choose a time resolution that allows to express all considerable changes of P_{\max} [CEVC-014]. E.g. if P_{\max} does not change over the complete time span of the maximum charging power limitation (P_{\max}) curve, only a single time slot is needed.

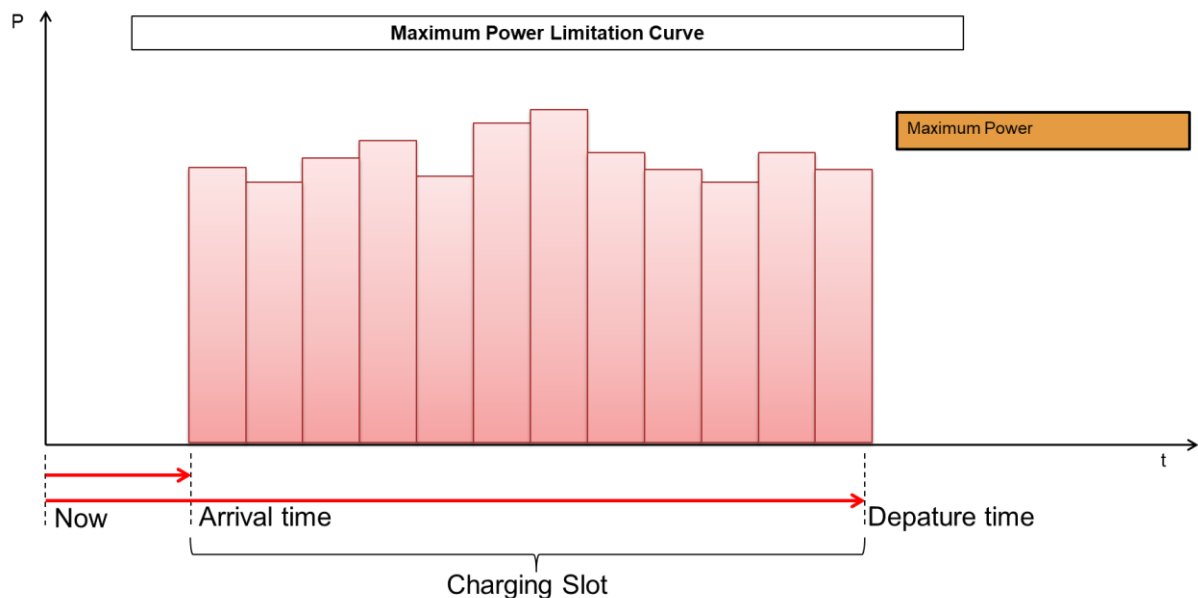


Figure 5: Maximum power limitation curve example

2.4.2.2 Conditions**Triggering Event:**

The Scenario is triggered when the EV requests an update. The EV SHALL request an update of the maximum charging power limitation (P_{\max}) curve [CEVC-015] after the EV has transmitted or updated its demand in Scenario 1. The EV MAY also request an update without transmitting a demand. This Scenario MAY also be triggered by the Energy Guard, e.g. if the maximum power limitation (P_{\max}) changes in critical ways [CEVC-016], that may result in a changed charging plan of an EV. In this case the Energy Guard SHALL send an update of the maximum power limitation (P_{\max}) curve to the corresponding EV, to trigger a new calculation of the charging plan [CEVC-017].

After the EV has requested an update the Energy Guard SHALL send a maximum power limitation (P_{\max}) curve within 55 seconds [CEVC-018], but ideally within 15 seconds. Otherwise the EV MAY run into a timeout and the EV MAY start charging with initial parameters.

Pre-condition:

The EV has no current information on a maximum power limitation (P_{\max}) curve yet or the EV has no maximum power limitation (P_{\max}) curve for a changed demand and cannot adjust its charging plan accordingly.

Post-condition:

The EV knows the maximum power limitation curve and can adjust its charging plan accordingly.

2.4.3 Scenario 3 - Energy Broker sends incentive table to EV**2.4.3.1 Description**

The Energy Broker sends a table with incentives [CEVC-054] to the EV that allows to create a cost optimized charging plan. The incentive table communicates up to 3 different power levels [CEVC-055] (tiers) over time. For each power level the Energy Broker MAY communicate 3 different incentives over time [CEVC-056]:

- Price (either relative [CEVC-019] or absolute [CEVC-020])
- Renewable energy percentage [CEVC-021]
- CO₂ emission [CEVC-022]

All power levels SHALL include the same combination of incentives [CEVC-023]. E.g. if a power level has a price incentive with a certain currency, the price incentive SHALL be used with the same currency for all power levels within the incentive table.

If a departure time was transmitted within Scenario 1, the Energy Broker SHALL deliver an incentive table that spans at least until the departure time [CEVC-024]. The Energy Broker MAY wait until the maximum power limitation curve from the Energy Guard is known and adjust the time span of the incentive table accordingly, if the maximum power limitation defines a timespan that exceeds the departure time.

If no departure time was delivered in Scenario 1, the Energy Broker SHALL deliver an incentive table for at least the next 48 hours [CEVC-025].

An incentive table SHOULD NOT be longer than 7 days (168h) [CEVC-026], even if the departure time is bigger than 168h. An incentive table that covers a longer time period MAY use a higher time resolution in the near future and a lower time resolution in the farer future. As soon as the time moves closer to the part of the incentive table with the lower time resolution, the incentive table SHOULD be updated [CEVC-027].

In general, the incentive table SHALL have a time resolution that allows to express all considerable changes in the power levels [CEVC-028]. E.g. if the power levels and incentives do not change over the time span of the incentive table, only a single time slot is needed.

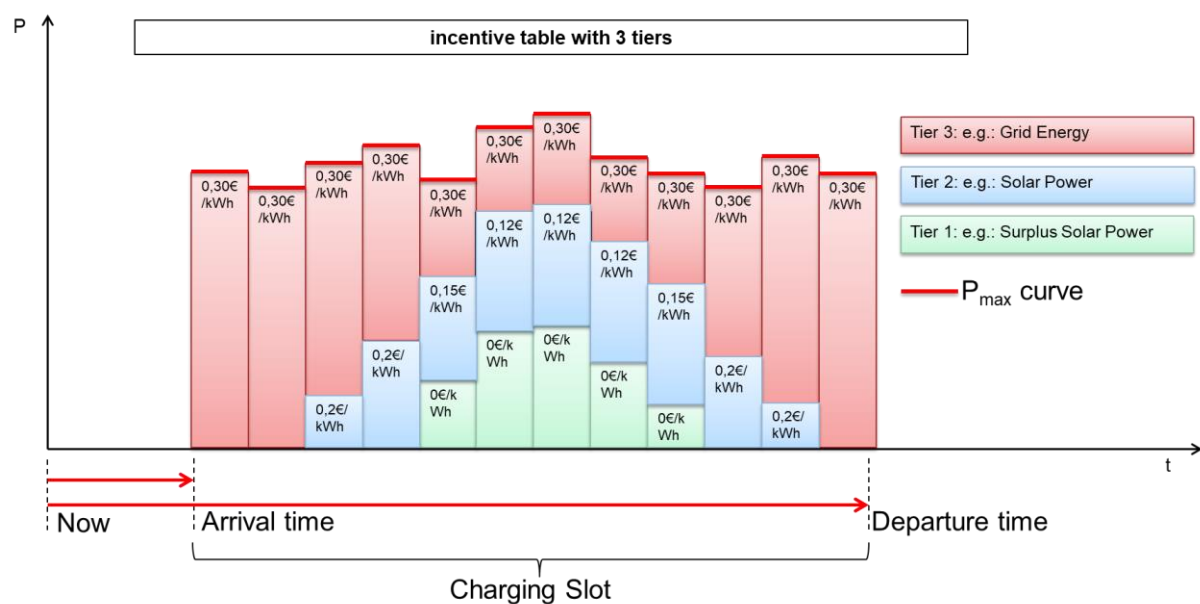
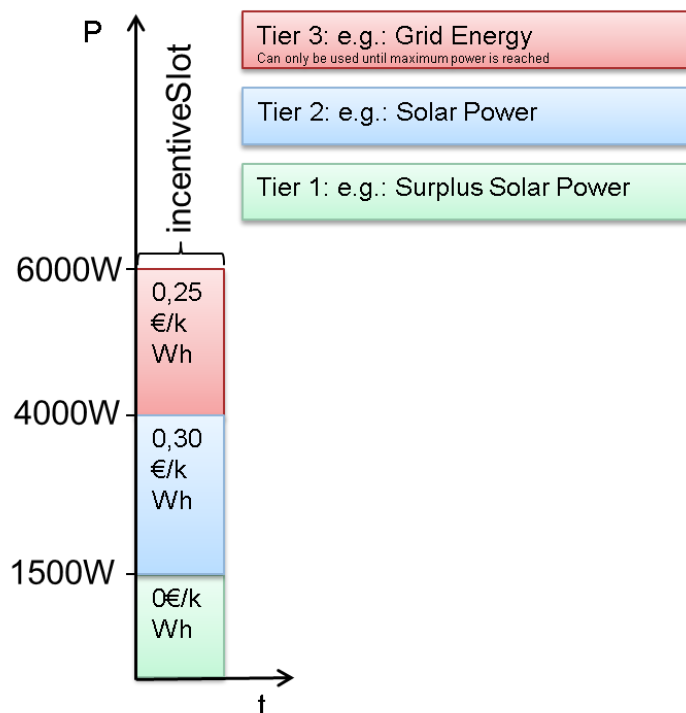


Figure 6: Incentive table example with consideration of the P_{max} curve from Scenario 2

Each power level defines its supply (difference between a power level and the next higher power level) that can be consumed for the incentive(s) (e.g. price) given by the power level. The power levels within the incentive table shall provide a realistic picture of the physical installation (e.g. PV power is consumed before power from the grid can be consumed). This means the power of the lowest power level will be consumed first before the power of the next higher power level can be consumed. Even if the power of a higher power level may be cheaper, physically it is not possible to consume that cheap power without consuming the power of the lower power level first. Only if physically possible the power levels can be ordered after price.

The following figure and examples show how the supply of each power level can be consumed and how the incentives are applied.

403



404

405 *Figure 7: IncentiveTable example (single incentiveSlot) for different tiers*

406 Example 1: If the actual power value is 5000W:

- 407 - 1500W are for free
- 408 - 2500W cost 0,30€/kWh
- 409 - and the last 1000W cost 0,25€/kWh.

410 Example 2: If the actual power value is 1000W:

- 411 - Only the lowest tier is used
- 412 - the 1000W are for free.

413 Example 3: If the actual power value is 2000W:

- 414 - 1500W are for free
- 415 - and the last 500W cost 0,30€/kWh.

416 Note: Negative prices or incentives may be used if feasible. Negative power values are not relevant in
 417 this Use Case, as this Use Case only covers the EV charging consumption and not bi-directional
 418 charging or similar cases.

419

420 **2.4.3.2 Conditions**421 **Triggering Event:**

422 The EV SHALL request an update of the incentive table after the EV has transmitted or updated its
 423 demand in Scenario 1 [CEVC-030] but MAY also request an update without transmitting a demand.

424 This MAY also be used to request an update later during charging to look for better power conditions
425 [CEVC-031].

426 This Scenario MAY also be triggered if the incentive table changes in ways that would affect the
427 charging costs of a charging plan of a certain EV. In this case the Energy Broker SHALL send an update
428 [CEVC-032] of the incentive table to the corresponding EV to trigger a new calculation of the charging
429 plan.

430 After the Scenario is triggered the Energy Broker SHALL send an incentive table within 55 seconds
431 [CEVC-033], but ideally within 15 seconds. Otherwise the EV MAY run into a timeout and the EV MAY
432 start charging with initial parameters.

433 **Pre-condition:**

434 The EV does not know the current incentive table and cannot create a cost optimized charging plan.

435 **Post-condition:**

436 The EV knows the incentive table and can create a cost optimized charging plan.

437

438 **2.4.4 Scenario 4 - EV sends charging plan curve to Energy Broker and Energy Guard**

439 **2.4.4.1 Description**

440 If the EV has received the maximum power limitation curve and the incentive table, the EV can create
441 a cost optimized charging plan to reach its charging demand. The charging plan SHALL be a simple
442 power curve, that communicates the planned charging power over time [CEVC-034]. A minimum
443 (P_{\min}) [CEVC-035] and expected charging power (P_{opt}) [CEVC-036] MAY be included into the charging
444 power curve. A maximum power (P_{\max}) [CEVC-037] SHALL always be set in the charging power curve
445 to allow the Energy Guard and the Energy Broker to calculate with the worst case charging
446 consumption of the EV. The charging plan is then transmitted to the Energy Guard and the Energy
447 Broker, so they can reserve the necessary charging power for the EV [CEVC-038]. The EV SHALL
448 charge according to its plan [CEVC-039] within its reported minimum (P_{\min}) and maximum (P_{\max})
449 power.

450 The charging plan represents all power needed by the EV. This includes charging the battery and
451 other loads of the EV, e.g. climatization and conditioning.

452 Note: The EV chooses the best available power slots for its charging plan depending on its charging
453 strategy. E.g. if the goal of the EV is to charge for the cheapest price, the EV will choose the power
454 slots with the cheapest price.

455

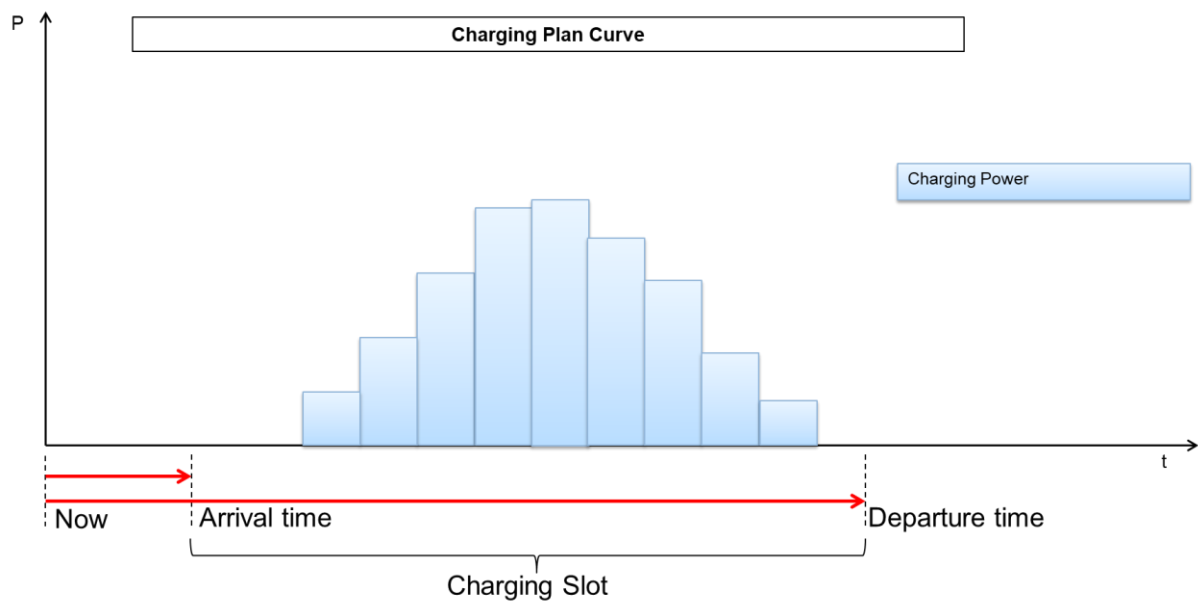


Figure 8: Charging plan curve example

2.4.4.2 Conditions

Triggering Event:

The Scenario SHALL initially be triggered by Scenario 2 [CEVC-040]. An EV SHOULD wait until it has also received the incentive table via Scenario 3 or until the timeouts in Scenario 2 or 3 occur, before it starts with the charging plan calculation [CEVC-041], to avoid unnecessary re-calculation of the charging plan.

If Scenarios 2 or 3 are used later on to send updates of either the incentive table or the maximum power limitation curve, this SHALL also trigger Scenario 4 to adjust the charging plan accordingly [CEVC-042]. If Scenario 2 or 3 sends an update, the EV SHOULD wait at least a short period (e.g. 5 seconds) of time, whether there is also an update provided from the other Scenario before charging plan calculation starts [CEVC-043], to avoid unnecessary re-calculation of the charging plan. However, it is also possible that only one Scenario sends an update, in this case the EV just uses the old information of the other Scenario.

After the Scenario is triggered the EV SHALL send the charging plan within 55 seconds, but ideally within 15 seconds [CEVC-044].

Pre-condition:

The charging plan is not known to the Energy Broker and Energy Guard. Therefore, the power used for charging cannot be reserved for the EV and it cannot be ensured that the needed power is available when the EV is planned to charge.

Post-condition:

The charging plan is known to the Energy Broker and Energy Guard, so they can reserve the necessary charging power for the EV.

2.4.5 Scenario 5 - Energy Guard heartbeat

2.4.5.1 Description

The Energy Guard SHALL send a regular heartbeat signal [CEVC-045], at least each 60 seconds [CEVC-046], which allows to detect whether the Energy Guard is available. In case of a heartbeat timeout, updates from the Energy Guard may no longer be received. Therefore, the EV SHOULD consider that the data from the Energy Guard may no longer be up to date.

2.4.5.2 Conditions

Triggering Event:

The Scenario SHALL be triggered as soon as the EV is connected to the Energy Guard [CEVC-047].

Pre-condition:

The EV does not know if the Energy Guard is currently available.

Post-condition:

The EV does know if the Energy Guard is currently available or not and can react as specified.

2.4.6 Scenario 6 - Energy Broker heartbeat

2.4.6.1 Description

The Energy Broker SHALL send a regular heartbeat signal [CEVC-048], at least each 60 seconds [CEVC-049], which allows to detect whether the Energy Broker is available. In case of a heartbeat timeout, updates from the Energy Broker may no longer be received. Therefore, the EV SHOULD consider that the data from the Energy Broker may no longer be up to date.

2.4.6.2 Conditions

Triggering Event:

The Scenario SHALL be triggered as soon as the EV is connected to the Energy Broker [CEVC-050].

Pre-condition:

The EV does not know if the Energy Broker is currently available.

Post-condition:

The EV does know if the Energy Broker is currently available or not and can react as specified.

511 2.4.7 Scenario 7 - Energy Guard error state

512 2.4.7.1 Description

513 The Energy Guard SHALL indicate errors [CEVC-051], so the EV can visualize this to the user. If the
514 Energy Guard has an error, updates from the Energy Guard may no longer contain valid data and the
515 Energy Guard may no longer be able to reserve the charging power for the EV correctly.

516

517 2.4.7.2 Conditions

518 Triggering Event:

519 The Scenario SHALL be triggered as soon as the Energy Guard has an error [CEVC-052].

520 Pre-condition:

521 The EV does not know if the Energy Guard has an error.

522 Post-condition:

523 The EV does know if the Energy Guard has an error and can react as specified.

524

525 2.4.8 Scenario 8 - Energy Broker error state

526 2.4.8.1 Description

527 The Energy Broker SHALL indicate errors [CEVC-053], so the EV can visualize this to the user. If the
528 Energy Broker has an error, updates from the Energy Broker may no longer contain valid data and
529 the Energy Broker may no longer be able to reserve the charging power for the EV correctly.

530

531 2.4.8.2 Conditions

532 Triggering Event:

533 The Scenario SHALL be triggered as soon as the Energy Broker has an error [CEVC-054].

534 Pre-condition:

535 The EV does not know if the Energy Broker has an error.

536 Post-condition:

537 The EV does know if the Energy Broker has an error and can react as specified.

538

2.5 Dependencies to other Use Cases

2.5.1 "EV Commissioning and Configuration"

The Actor EV of this Use Case acts as Actor EV within "EV Commissioning and Configuration".

The Actors Energy Broker and Energy Guard of this Use Case act as Actor CEM within "EV Commissioning and Configuration".

The following Scenarios of "EV Commissioning and Configuration" Use Case are relevant in this Use Case.

2.5.1.1 "Scenario 1 - EV connected"

If this Use Case relates to connecting the EV to the EVSE the referenced Scenario is meant. Therefore, this Scenario SHALL be supported by the Actors EV, Energy Guard and Energy Broker.

2.5.1.2 "Scenario 6 - EV sends charging power limits"

The Actors Energy Guard and Energy Broker can use the EV charging power limits during calculations for Scenario 2 and 3 of this Use Case. Therefore, the referenced Scenario SHOULD be supported by the Actor EV and SHALL be supported by the Actors Energy Guard and Energy Broker.

2.5.1.3 "Scenario 7 - EV sleep mode"

If the Actor EV is in sleep mode the EV will not charge. To reduce the EV wake up cycles to a minimum, the Energy Guard and Energy Broker shall only send updates in case of significant changes. Therefore, the referenced Scenario SHALL be supported by the Actor EV if it supports a sleep mode and the Scenario SHALL be supported by the Actors Energy Guard and Energy Broker.

2.5.1.4 "Scenario 8 - EV disconnected"

If the EV gets disconnected the EV will no longer be charged. The EV can be removed from the power management calculations of the Actors Energy Guard and Energy Broker. Therefore, the referenced Scenario SHALL be supported by the Actor EV and it SHALL be supported by the Actors Energy Guard and Energy Broker.

2.5.2 "EV Charging Electricity Measurement"

The Actor EV of this Use Case acts as Actor EV within the Use Case "EV Charging Electricity Measurement".

The Actors Energy Broker and Energy Guard of this Use Case act as Actor CEM within the Use Case "EV Charging Electricity Measurement".

At least Scenario 1 or Scenario 2 or Scenario 3 of the Use Case "EV Charging Electricity Measurement" SHOULD be supported, otherwise the Energy Guard and Energy Broker cannot

evaluate the progress of the charging process and divergences from the charging plan. This may be especially helpful in case PWM communication is used between EV and EVSE. If the EV charges with less power than reserved by the Energy Guard and Energy Broker for the EV, this can lower the efficient use of power for all consumers within the installation.

2.5.3 "Overload Protection by EV Charging Current Curtailment"

The Actor EV of this Use Case acts as Actor EV within the Use Case "Overload Protection by EV Charging Current Curtailment".

The Actor Energy Guard of this Use Case acts as Actor Energy Guard within the Use Case "Overload Protection by EV Charging Current Curtailment".

Even if a power limitation curve is provided within this Use Case, the currents provided in the Use Case "Overload Protection by EV Charging Current Curtailment" are more significant especially in case of asymmetric charging with phase specific currents. In any case, the EV shall neither consume more than the power limitation value provided by the "Overload Protection by EV Charging Current Curtailment" Use Case nor the power limitation curve provided by this Use Case.

2.6 Assumptions and Prerequisites

None.

3 Technical SPINE solution

3.1 General rules and information

3.1.1 Underlying technology documents

This technical solution relies on the SPINE Resources Specification version 1.1.1 [ResourceSpecification].

For interoperable connectivity this technical solution relies on:

- SPINE Protocol Specification version 1.1.1 [ProtocolSpecification] as application protocol.
- SHIP Specification version 1.0 [SHIP] as transport protocol.

Further applicable documents:

- EEBUS Use Case Base Specification version 1.0.0 [UseCaseBaseSpecification].

3.1.2 Use Case Discovery rules

The Use Case Discovery SHALL be supported by each Actor and the following rules SHALL apply:

- The string content for the Element "nodeManagementUseCaseData. useCaseInformation. useCaseSupport. useCaseName" within the Use Case Discovery (please refer to [ProtocolSpecification]) SHALL be "coordinatedEvCharging". The string content SHALL only be defined by this Use Case (regardless of the Use Case version).
- The string content of the Element "nodeManagementUseCaseData. useCaseInformation. actor" within the Use Case Discovery (please refer to [ProtocolSpecification]) SHALL be set to the according value stated within the corresponding Actor's section.
- An Actor A that is implemented to support this Use Case specification SHALL set the Element "nodeManagementUseCaseData. useCaseInformation. useCaseSupport. useCaseVersion" within the Use Case discovery (please refer to [ProtocolSpecification]) to "1.0.1" (for details on the structure of the Use Case version number please refer to [UseCaseBaseSpecification]).
- If an Actor A supports multiple versions of this Use Case with the same major version number, only the highest one SHOULD be set within the Use Case discovery.
- If an Actor A finds a proper counterpart Actor B for this Use Case that supports multiple versions of this Use Case with the same major version number as supported by Actor A, the Actor A SHOULD evaluate from these versions of Actor B only the highest version number.
- If an Actor A supports multiple versions of this Use Case with different major version numbers, for each major version number only the highest version number SHOULD be set within the Use Case discovery.
- If an Actor A finds a proper counterpart Actor B for this Use Case that supports only versions with a major version number not implemented by Actor A, it still might be possible to run the Use Case or parts of the Use Case. Therefore, the Actor A should try to evaluate the Actor B as a valid partner for this Use Case.

3.1.3 Rules for "Content of Specialization..." tables and "Content of Function..." tables

3.1.3.1 General presence indication definitions

Abbreviations for the presence indication of Elements listed in the tables are defined as follows:

Abbreviation	Meaning	Link to requirement keywords
M	Mandatory	SHALL
R	Recommended	SHOULD
O	Optional	MAY

Table 2: Presence indication description

An Actor MAY support Elements that are not listed in the tables. However, another Actor MAY ignore these Elements.

The presence indications "M", "R" and "O" are always meant relative to the respective parent Element. I.e. if a parent Element is optional ("O") and a child is mandatory ("M") the child Element can only be present if the parent Element is present as well.

Note: The indications and the aforementioned rules apply for "complete messages" (so-called "full function exchange", please refer to [ProtocolSpecification]). In contrast, the so-called "restricted function exchange" is designed to permit exchange of specific excerpts of data, i.e. fewer Elements than potentially available from the data owner (partially even not all "mandatory" Elements).

3.1.3.2 Presence indications for "Content of Specialization..." tables

This section only defines rules for the client side.

Elements that are marked with "M" SHALL be supported by the client in case of readable as well as writeable data. This Element may be optional on the server side.

The following applies for readable data that is exchanged in a "read/reply" or "notify" operation:

- "R" means that the data SHOULD be supported by the client. In other words: If the server responds with the according Element, the client SHOULD be able to interpret the according Elements.
- "O" means that the data MAY be supported by the client. In other words: If the server responds with the according Element, the client MAY be able to interpret the according Elements.

The following applies for writeable data that is exchanged in a "write" operation:

- "R" means that the data SHOULD be written by the client.
- "O" means that the data MAY be written by the client.
- "F" means that the data SHALL NOT be written by the client.

The following applies for Elements that are not listed in the Actor section:

- In case of a received "reply" message: The client MAY ignore the Element.
- In case of a "write" operation to be created: The client MAY set the Element but SHALL consider that the server may ignore the Element.

- In case of a received "notify" message: The client MAY ignore the Element.

M, R or O may be combined with the suffix "(event)" to express that a supported Element or value only has to be supported during a certain event and hence does not need to be present at all times. If the event is not active the Element may be omitted or another value may be set. In most cases a High-Level requirement reference for the event is given in the rules column.

3.1.3.3 Presence indications for "Content of Function..." tables

This section only defines rules for the server side.

Elements that are marked with "M" SHALL be supported by the server in case of readable as well as writeable data. In case of writeable data (marked with "M \W") the server does not need to set the Element, because the Element is set only by the client.

The following applies for readable data that is exchanged in a "read/reply" or "notify" operation:

- "R" means that the data SHOULD be provided by the server.
- "O" means that the data MAY be provided by the server.
- "F" means that the data SHALL NOT be provided by the server.

The following applies for writeable data that is exchanged in a "write" operation:

- "R" means that the data SHOULD be supported. In other words: If the client writes the Element, the server SHOULD accept those messages and the contained Elements.
- "O" means that the data MAY be supported. In other words: If the client writes the Element, the server MAY accept those messages and the contained Elements.

The following applies for Elements that are not listed in the Actor section:

- In case of a received "read" request: The according Element MAY be set in the reply.
- In case of a received "write" operation: The server MAY ignore the Element.
- In case of a "notify" operation to be created: The server MAY set the Element.

Note: The server will only accept write operations if the result fulfils the server Function requirements (permitted values, e.g.). Write operations on Elements that are not writeable MAY result in an error message.

M, R or O may be combined with the suffix "(event)" to express that a supported Element or value only has to be supported during a certain event and hence does not need to be present at all times. If the event is not active the Element may be omitted or another value may be set. In most cases a High-Level requirement reference for the event is given in the rules column.

3.1.3.4 Cardinality indications - Permitted number of occurrences

A cardinality indication expresses constraints on the number of occurrences of a given Element or data set. In this section we use "X" as representation for such an Element or data set. Furthermore, "a" and "b" represent constraints. The following rules apply for the occurrence of "X" and its content related to a specific Scenario (see note underneath the list):

1. X
No cardinality indication.
2. X (a..b)
This means "X" SHALL occur at least "a" times and at maximum "b" times.
3. X (a..
This means "X" SHALL occur at least "a" times and MAY occur more than "a" times.
4. X (..b)
This means "X" SHALL occur at maximum "b" times and MAY occur less than "b" times (even zero occurrences are permissive).

Note: These rules apply only under consideration of presence indications and with regards to the given Scenario or Function definition for this Use Case.

The following table is an example to explain this for two different placements.

Scenario [{...}]: M/R/O [W][C]	Element	Value	[High Level Mapping] Element and value rules
1: O
2: M \W	xFeatureType. xListData. xData. [UC-002] (1..3)		
2: M \W	xId	<g7> [<g8>] [<g9>]	PRIMARY IDENTIFIER of x
2: M \W	timePeriod		...
2: M \W	timePeriod. startTime	<xs:duration>	
2: M \W	xSlot. (1..)		
2: M \W	xSlot. xSlotId		...
2: M \W	xSlot. duration	<xs:duration>	...
2: M \W	qId	<h3>(-><g7>) [<h4>(-><g8>)] [<h5>(-><g9>)]	FOREIGN IDENTIFIER.
...

Table 3: Example table for cardinality indications

The field

xFeatureType. xListData. xData. [UC-002] (1..3)

introduces a data pattern (required Elements and values) for "xData" instances used for Scenario 2. The field itself specifies that such an "xData" instance SHALL occur at least 1 time and at maximum 3 times within "xListData" of Feature Type "xFeatureType". However, this holds only for Scenario 2 and only if such "xData" are required. In this case, they are required, as the left field

2: M \W

denotes that this data set is mandatory for Scenario 2. The "Value" definition

<g7> [<g8>] [<g9>]

of the Element "xId" specifies that this is the reason for the cardinality: There must be at least one "xData" instance and the corresponding "Value" placeholder is "<g7>" (see section 3.1.3.6 for the definition of "Value" placeholders). The second and third instance of "xData" are optional, as the corresponding placeholders "<g8>" and "<g9>" are put in brackets. Of course, the placeholders SHALL then have distinct values.

The "Value" definition of the Element "qId" contains the expression

```
<h3>(-><g7>) [<h4>(-><g8>)] [<h5>(-><g9>)]
```

This means that the placeholder "<h3>" is to be used with "<g7>". Likewise, "<h4>" is associated with "<g8>" and "<h5>" is associated with "<g9>".

Some Scenarios may require the association to two or more placeholders. As an example, we consider an expression

```
<t2>(-><v1>,<k3>)
```

In this case the placeholder "<t2>" is to be used with the pair of "<v1>" and "<k3>".

The field

```
xSlot. (1..)
```

expresses that the Element "xSlot" SHALL occur at least one time within its "xData", but MAY occur more than one time.

The remaining fields do not have an explicit cardinality indication.

3.1.3.5 Writability and changeability indication

In the same column where the presence indications are denoted, a mark is used to distinguish between writeable, changeable or readable Elements:

- Elements that are marked with "\W" are written by a client and SHALL be writeable at the server according to their presence indications. The client is not obliged to read the according data. Received notifications do not need to be evaluated.
- Elements that are marked with "\C" are changed by a client and SHALL be changeable at the server according to their presence indications. The client is not obliged to read the according data. Received notifications do not need to be evaluated.
- Elements that are marked with "\RW" are read and written by a client and SHALL be writeable and provided by the server according to their presence indications. Received notifications SHALL be evaluated according to their presence indications.
- Elements that are marked with "\RC" are read and changed by a client and SHALL be changeable and provided by the server according to their presence indications. Received notifications SHALL be evaluated according to their presence indications.
- Elements that are not marked are only read by a client and SHALL be provided by the server according to their presence indications. Received notifications SHALL be evaluated according to their presence indications.

"Writeable" means that the Element and its value may be written by a client. This includes the possibility to modify (if the Element is already present), create (if the Element is not present yet), and delete the Element. The server SHALL adjust its Function according to the received "write" operation (unless the server cannot accept the "write" operation according to section 3.1.3.3).

"Changeable" means that the Element's value may be changed by a client. If the Element is not present at the resource before, it probably **cannot** be created by the client via the "write" operation. In this case the server MAY decline such a message.

Note: "\W" includes "\C" already.

Note: Depending on the resource a client might need to request a proper binding before the server accepts a "write" operation.

3.1.3.6 Rules for "Value" placeholders

If the "Value" column contains values for identifiers they are always written as placeholder variable (i.e. placeholder for the real value of the Element) in angle brackets, e.g. <x1>. This means all Elements used within a Scenario that have <x1> (e.g.) in the "Value" column SHALL have set the same content of the Element.

A placeholder variable <xY> (e.g. <x1>) for Scenario A is, in general, independent from a placeholder variable <xY> for Scenario B. However, the server SHOULD combine datasets if possible. If there is the requirement that the same value SHALL be used for different stated Scenarios, the according Scenario numbers in column "Scenario" are put in curly brackets (" {... }") for the Element containing the variable. Several curly bracket groups may exist.

Example: An Element with variable <x1> contains in the column "Scenario" the following expression:
{2, 3}, {4, 5}

This means that Scenario 2 and 3 SHALL use the same value for the variable (e.g. 5) as well as Scenario 4 and 5 SHALL use the same value for the variable (e.g. 12). The variable values MAY differ between the two groups ({2, 3} and {4, 5}).

3.1.3.7 Rules for content of "Value" column

For a given Scenario the "Value" column may restrict the permitted content of a Function's Element to one or more particular values. This means that Elements with values deviating from the restriction (i.e. from the permitted values) do not belong to the respective Scenario and need to be considered as if the Element is not set. If more than one particular value is permitted for an Element the values are in a single line each.

If a presence indication is set for the value (in an additional column before the value) the following rules SHALL be applied:

- "M" means that the value SHALL be supported. This means the value needs to be set at a certain point in time (depending on the value rules) or for a certain Element within a list entry.
- "R" means that the value SHOULD be supported.

- "O" means that the value MAY be supported.

If all possible values of a given mandatory Element are optional or recommended and this Element is used for the purpose of the respective Scenario, one of the values SHALL be set. If all possible values of a given optional or recommended Element are optional or recommended, this Element MAY contain also other values, but then this Element SHALL NOT be considered as part of the respective Scenario.

M, R or O may be combined with the suffix "(event)" to express that a supported value only has to be supported during a certain event and hence does not need to be present at all times. If the event is not active another value may be set. In most cases a High-Level requirement reference for the event is given in the rules column.

If no presence indication is set for the value, the following rules SHALL be applied:

- In case of Elements where the server may set or change an Element on its own (see section 3.1.3.5):
 - within the tables in the "Server data - Resources" sections:
 - the server SHALL support at least one of the listed values.
 - within the tables in the "Client data - Specializations" sections:
 - the client SHALL support all listed values.
- In case of Elements that are writable or changeable (see section 3.1.3.5):
 - within the tables in the "Server data - Resources" sections:
 - the server SHALL support all listed values.
 - within the tables in the "Client data - Specializations" sections:
 - the client SHALL support at least one of the listed values.

Depending on the Element, different values may be used during runtime. If this is the case, those rules are described within the value rules.

If a value is placed in parenthesis, the corresponding value is a recommendation. The actual value MAY deviate from this, e.g. "(1024)".

3.1.3.8 General information on how to interpret the "Content of Function..." and "Content of Specialization..." tables

Within the "Client data - Specializations" sections each Specialization is described in an own sub-section with the name "Specialization "<name of the Specialization>" (e.g. "Specialization "Measurement_GridFeedInEnergy"). It contains only one table that includes all Elements needed for this Specialization. The different Functions are mentioned in a continuous row, highlighted with grey background colour. This row contains the following parts:

<Feature Type>. <Function>.[<list entry instance name>.]

The <list entry instance name> is only included if the <Function> is a list-based Function. An example could be:

DeviceConfiguration. deviceConfigurationKeyValueDescriptionListData.
deviceConfigurationKeyValueDescriptionData.

In the following rows, only the names of the Elements are stated, without the prefix described above.

Within the "Server data - Resources" sections each Feature Type is described in an own sub-section with the name "Feature Type "<name of the Feature Type>" (e.g. "Feature Type "Measurement"). It contains sub-sections for each Function named "Function "<name of the Function>" (e.g. "Function "measurementListData"). These sections contain one table with all Elements needed for this resource. The list entries are mentioned in a continuous row, highlighted with grey background colour. This row contains the following parts:

<Feature Type>. <Function>.[<list entry instance name>.]

The <list entry instance name> is only included if the <Function> is a list-based Function. An example could be:

Measurement. measurementDescriptionListData. measurementDescriptionData.

In the following rows, only the names of the Elements are stated, without the prefix described above.

For both kinds of tables, the following applies:

- Parent Elements are marked with a dot at the end of the name:

<parent Element>.

E.g.:

value.

- If there are sub-Elements, they are described in own rows with the name of the parent Element as prefix, separated by a dot and a blank space:

<parent Element>. <sub-Element>

E.g.:

value. number

3.1.4 Rules for "Feature Types and Functions..." tables

3.1.4.1 Presence indications for "Feature Types and Functions..." tables

The following presence indications are used:

Abbreviation	Meaning	Link to requirement keywords
M	Mandatory	SHALL
R	Recommended	SHOULD
O	Optional	MAY

Table 4: Presence indication of Feature Types and Functions support

If at least one Function of a Feature has the presence indication "M", it is mandatory to support the Feature.

870 3.1.4.2 Rules for "Possible operations" column

871 Within the "Feature Types and Functions..." tables the column "Possible operations" state whether
872 the Function is read- or writeable (as defined in the detailed discovery mechanism, see
873 [ProtocolSpecification]).

874 If the "partial" concept (also called "restricted function exchange") SHALL be supported, the
875 following notation is used (separated for read and write access):

876 read (M). partial (M)

877 write (M). partial (M)

878 If the "partial" concept SHOULD be supported, the following notation is used:

879 read (M). partial (R)

880 write (M). partial (R)

881 If the "partial" concept MAY be supported, the following notation is used:

882 read (M). partial (O)

883 write (M). partial (O)

884 The server can decide whether a notification is submitted complete or partial (as described in
885 [ProtocolSpecification]) if not defined differently within this Use Case Specification.

886

887 3.1.5 "Actor ... overview" diagram rules

888 Within the "Actor [...] overview" diagrams in the "Actors" sub-sections the complete functionality of
889 this Use Case is provided, including optional Scenarios. Which Scenarios are optional can be found in
890 Table 1. The Actor MAY have more functionality implemented than needed for this Use Case.

891 For the following Actor overview example, a brief description of the graphical symbols will be
892 described.

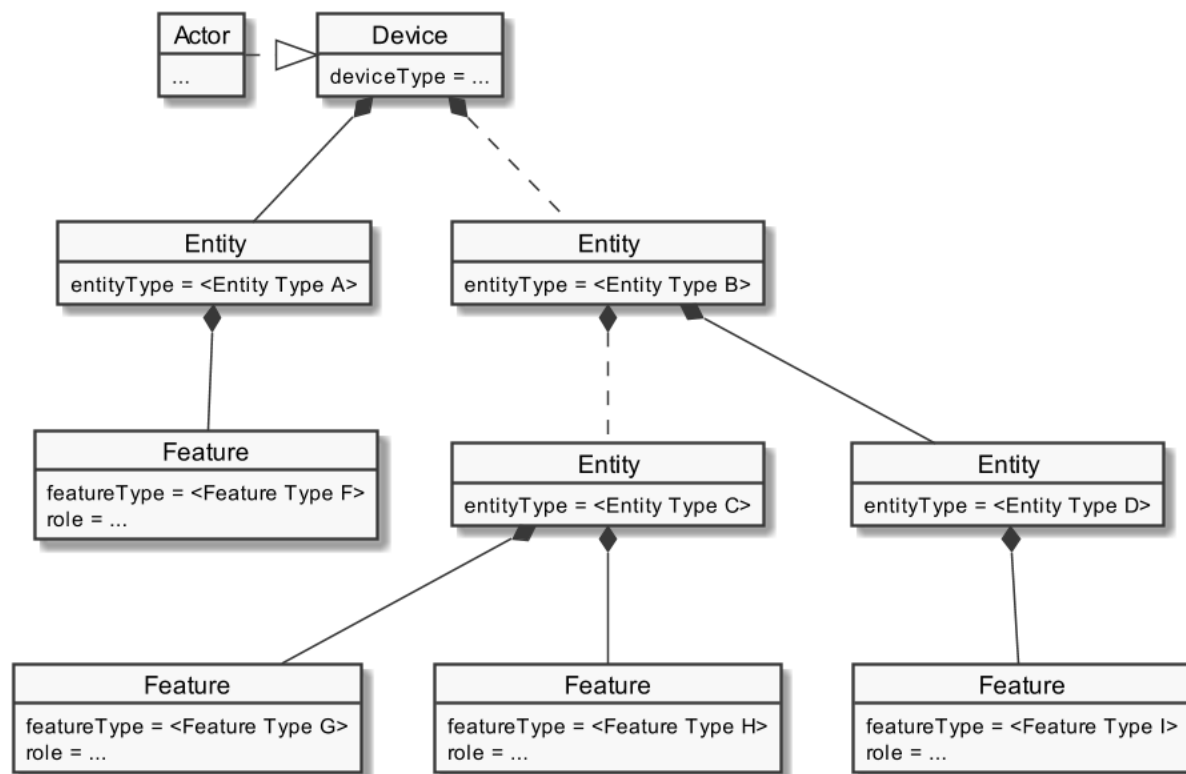


Figure 9: Actor overview example

The solid lines in the figure represent an immediate parent-childhood relation: The Entity with "<Entity Type A>" is a direct child of "Device". The Entity with "<Entity Type D>" is a direct child of the Entity with "<Entity Type B>". All Features are immediate child of the respective Entity.

The dashed lines in the figure express that there MAY be additional Entities between the shown Entities: A vendor's implementation MAY have one or more Entities between "Device" and the Entity with "<Entity Type B>". Likewise, a vendor's implementation MAY have one or more Entities between the Entity with "<Entity Type B>" and the Entity with "<Entity Type D>".

3.1.6 Specializations

Within the "Actors" sub-sections Specializations are referenced. A Specialization describes a dataset necessary to fulfil the specific requirements of a High-Level Use Case and its Scenarios. Often data from multiple different Features and Functions are needed to fulfil the requirements. Therefore, a Specialization defines a dataset that may encompass multiple related Functions from one or more different Features.

As different Use Cases sometimes share similar requirements, Specializations are also important from a re-usability perspective. This approach is used to improve consistency across Use Cases and avoid multiple variances of basically the same dataset. This is especially important in the case when an implementation supports multiple Use Cases. E.g. if a power measurement is necessary in two different Use Cases, both Use Cases could define slightly different datasets. In this case the server as well as the client functionality would have to implement both variances if both Use Cases are supported. This means, depending on the number of Use Cases, two or more datasets need to be

generated, transmitted and stored instead of one. Therefore, already existing Specializations specified within [UseCaseBaseSpecification] are used in this Use Case to avoid such problems.

If a Feature server can provide the data of a Specialization, the data does not necessarily always need to be available at the Feature server. There might be situations where the user deactivates a Use Case. There may also be other reasons why Use Case data cannot be provided currently. Therefore, a client always needs to be subscribed (as described in section 3.3.4) on the corresponding dataset to stay updated.

The SPINE resource description given in the "SPINE resources of the Actor" sections are derived from the Specializations given in the Actor's overview diagram. Please refer to [UseCaseBaseSpecification] for a detailed description of all Specializations.

3.1.7 Order of messages within the sequence diagrams

There are several sequence diagrams in this document describing message flows. The order of the messages SHOULD be kept by the communications partners, but there might be cases where a different order makes sense. The communications partners SHALL be able to handle the Scenario functionalities even if the messages are transmitted in a different order by the other Actor(s). The sequence diagrams can be seen as examples.

3.1.8 Further information and rules

None.

3.2 Actors

3.2.1 EV

3.2.1.1 Resource hierarchy

Within the Use Case discovery this Actor SHALL be denoted as "EV" in the Element "nodeManagementUseCaseData. useCaseInformation. actor".

The following diagram provides an overview of the Actor EV resource hierarchy.

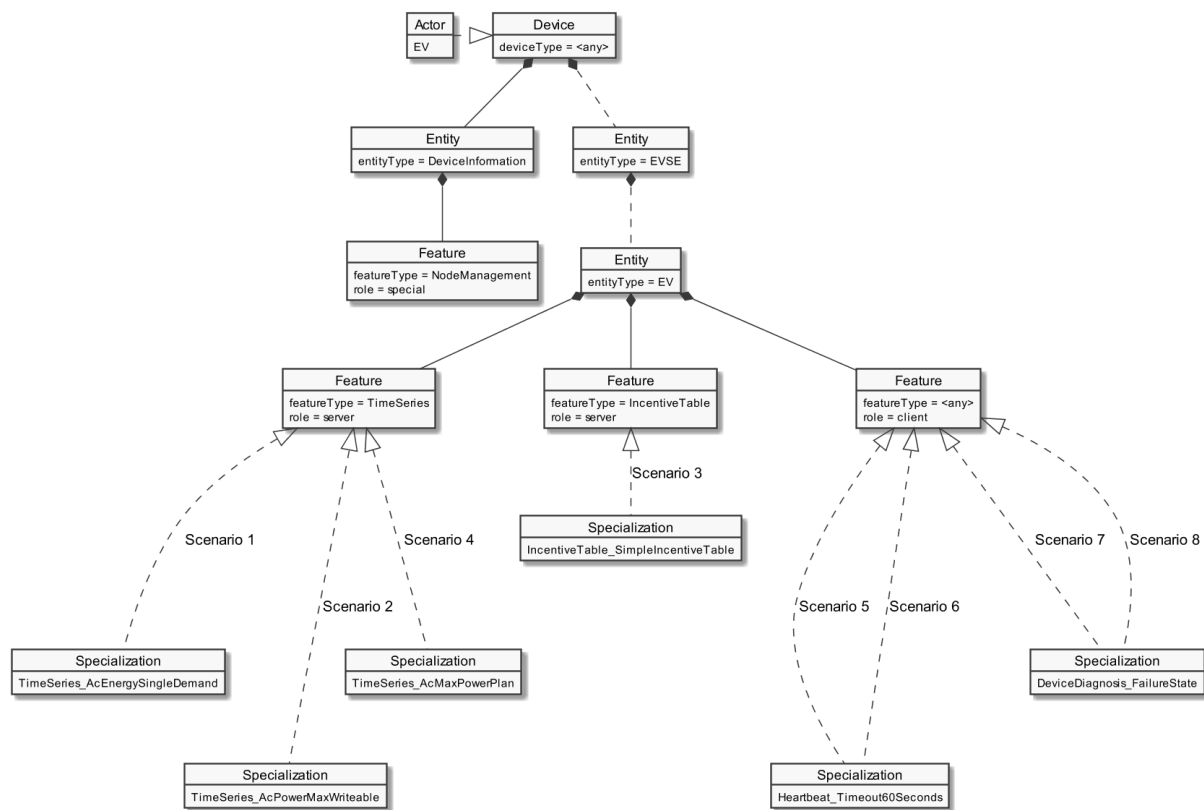


Figure 10: Actor "EV" overview

The "Actor ... overview" diagram rules section describes how to interpret the diagram above. See the "Specializations" section for more information regarding the Specializations given in the diagram above.

Note: The entityType "DeviceInformation" with the featureType "NodeManagement" is required by the SPINE protocol and therefore SHALL be supported. Both types are added in the figure for completeness but are not directly linked to the Use Case.

The Use Case specific data follows behind the entityType "EV" which is a sub-Entity of the "EVSE" Entity. The Specializations represent the Scenario specific data that has to be supported for each Scenario and are realized with the according featureTypes.

If a Specialization is connected to a Feature with the role "client", the Actor has a client role for this data. This means the Actor accesses the data set described by the Specialization at a corresponding server Feature. Further details are described in the sub-section "Client data - Specializations".

If a Specialization is connected to a Feature with the role "server", the Actor has the server role for this data. This means the Actor must provide the corresponding data set of the Specialization on its Features. Further details are described in the sub-section "Server data - Resources".

3.2.1.2 Server data - Resources

3.2.1.2.1 Overview

Behind the entityType "EV" the Actor EV SHALL offer the Feature Types and Functions given in the table below.

Feature Type	Scenario: M/R/O	Function	Possible operations
TimeSeries	1: M 2: M 4: M	timeSeriesDescriptionListData	read (M). partial (R)
	1: O 2: R 4: O	timeSeriesConstraintsListData	read (M). partial (R)
	1: M 2: M 4: M	timeSeriesListData	read (M). partial (R) write (M). partial (R)
IncentiveTable	3: M	incentiveTableDescriptionData	read (M). partial (R) write (M). partial (M)
	3: M	incentiveTableConstraintsData	read (M). partial (R)
	3: M	incentiveTableData	read (M). partial (R) write (M). partial (M)

Table 5: Feature Types and Functions used within this Use Case by the Actor EV

For each of these Feature Types the following rule applies: There SHALL be at maximum one Feature with the Feature Type in the Entity.

Note: As a consequence of the previous rule, an implementation may need to have Feature data from different Scenarios/Specializations or even Use Cases in a given Feature.

The Scenario number shows in which Scenarios the Energy Guard acts as server and which Feature Types and Functions are relevant in each Scenario.

A detailed definition of the Elements and values that shall be supported in each Function is given in the following sub-sections.

Note: If in the table above "partial" read is not mentioned or is only optional, it still might be mandatory to support partial notifications. The details of "partial" support are described within the Scenario sections.

Note: The presence indications stated above are meant relative to the ones of the according Scenario stated in Table 1. I.e. if a Scenario is optional ("O") and a Feature Type is mandatory ("M") the Feature Type must only be supported if the Scenario is supported, too.

Note: Further Features MAY be implemented on the same Entities, as well as further Functions MAY be implemented in the used Entities.

983 3.2.1.2.2 Feature Type "TimeSeries"

984 3.2.1.2.2.1 Function "timeSeriesDescriptionListData"

Scenario [...]: M/R/O [W][C]	Element	Value	[High Level Mapping] Element and value rules
1: M	TimeSeries. timeSeriesDescriptionListData. timeSeriesDescriptionData.		
1: M	timeSeriesId	<m1>	PRIMARY IDENTIFIER of timeSeries.
1: M	timeSeriesType	"singleDemand"	"singleDemand" SHALL be used to model a single demand. Only one timeslot SHALL be used.
1: M	timeSeriesWriteable	"false"	If set to false or omitted, this timeSeriesData SHALL NOT be writeable.
1: M	unit	"Wh"	The unit SHALL be applied to the value in timeSeriesData
2: M	TimeSeries. timeSeriesDescriptionListData. timeSeriesDescriptionData.		
2: M	timeSeriesId	<m2>	PRIMARY IDENTIFIER of timeSeries.
2: M	timeSeriesType	"constraints"	"constraints" SHALL be used to model certain value constraints or limits over time.
2: M	timeSeriesWriteable	"true"	If set to false or omitted, this timeSeriesData SHALL NOT be writeable.
2: M(event)	updateRequired		[CEVC-015] With updateRequired the server can request an update of writeable or changeable data related to the same PRIMARY IDENTIFIER from a client. The server SHALL ensure that only one responsible client is able to update the related data. To request an update the server SHALL set updateRequired to "true". Note: In this case, the server expects the responsible client to update the writeable or changeable data related to the same PRIMARY IDENTIFIER. However, also if updateRequired is set to "false" a server SHOULD in general allow updates of the data from the responsible client. The server SHALL set the updateRequired back to "false", as soon as "timeSeriesDescriptionListData" was updated successfully (if writeable or changeable) OR the update of the other writeable or changeable data

			related to the same PRIMARY IDENTIFIER was successful. Note: The client does not need to stop an ongoing update process (e.g. if multiple Functions are written), when updateRequired is set back to "false". The server MAY choose to withdraw the update request at any time by setting updateRequired back to "false".
2: M	unit	"W"	The unit SHALL be applied to the value in timeSeriesData
4: M	TimeSeries. timeSeriesDescriptionListData. timeSeriesDescriptionData.		
4: M	timeSeriesId	<m3>	PRIMARY IDENTIFIER of timeSeries.
4: M	timeSeriesType	"plan"	"plan" SHALL be used to model how a certain value will change over time.
4: M	timeSeriesWriteable	"false"	If set to false or omitted, this timeSeriesData SHALL NOT be writeable.
4: R	measurementId	<k1>	FOREIGN IDENTIFIER, links data of other Features to the timeSeries.
4: M	unit	"W"	The unit SHALL be applied to the value in timeSeriesData

Table 6: Content of Function "timeSeriesDescriptionListData" at Actor EV

3.2.1.2.2.2 Function "timeSeriesConstraintsListData"

Scenario {...}: M/R/O [W]/[C]	Element	Value	[High Level Mapping] Element and value rules
1: R	TimeSeries. timeSeriesConstraintsListData. timeSeriesConstraintsData.		
1: M	timeSeriesId	<m1>	SHALL be set as PRIMARY IDENTIFIER.
1: O	slotCountMin		If set and a corresponding timeSeriesData exists, it SHALL NOT have less identifiable slots.
1: R	slotCountMax	1	If set, the corresponding timeSeriesData SHALL NOT have more slots.
1: O	slotDurationMin		If set, the corresponding timeSeriesData SHALL NOT have slots with shorter duration.
1: O	slotDurationMax		If set, the corresponding timeSeriesData SHALL NOT have slots with longer duration.
1: O	slotDurationStepSize		If set, the duration of a slot SHALL be a multiple of slotDurationStepSize and also slotDurationMin and slotDurationMax SHALL be multiples of slotDurationStepSize.

1: O	earliestTimeSeriesStartTime		If set the start time of a timeSeries SHALL be later than or equal to earliestTimeSeriesStartTime.
1: R	latestTimeSeriesEndTime		If set, the end time of a timeSeries SHALL be earlier than or equal to latestTimeSeriesEndTime.
1: R	slotValueMin.		The value, maxValu or minValu of a slot SHALL be equal to or higher than slotValueMin. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	slotValueMin. number		SHALL be used.
1: O	slotValueMin. scale		MAY be used. If absent, a default value of "0" applies.
1: R	slotValueMax.		The value, maxValu or minValu of a slot SHALL be equal to or lower than slotValueMax. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	slotValueMax. number		SHALL be used.
1: O	slotValueMax. scale		MAY be used. If absent, a default value of "0" applies.
1: O	slotValueStepSize.		If slotValueStepSize is set and slotValueMin as well as slotValueMax is not set, the elements value, maxValu and minValu of a slot SHALL be a multiple of slotValueStepSize. If slotValueStepSize is set and slotValueMin is also set, the elements value, maxValu and minValu of a slot SHALL be slotValueMin plus a multiple of slotValueStepSize. If slotValueStepSize is set and slotValueMax is also set, the elements value, maxValu and minValu of a slot SHALL be slotValueMax minus a multiple of slotValueStepSize. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	slotValueStepSize. number		SHALL be used.
1: O	slotValueStepSize. scale		MAY be used. If absent, a default value of "0" applies.
2: R	TimeSeries. timeSeriesConstraintsListData. timeSeriesConstraintsData.		
2: M	timeSeriesId	<m2>	SHALL be set as PRIMARY IDENTIFIER.
2: F	slotCountMin		Element SHALL NOT be used.
2: R	slotCountMax	(1024)	If set, the corresponding timeSeriesData SHALL NOT have more slots.
2: F	slotDurationMin		Element SHALL NOT be used.
2: F	slotDurationMax		Element SHALL NOT be used.

2: F	slotDurationStepSize		Element SHALL NOT be used.
2: O	earliestTimeSeriesStartTime		If set the start time of a timeSeries SHALL be later than or equal to earliestTimeSeriesStartTime.
2: R	latestTimeSeriesEndTime	(PT168H)	[CEVC-012] If set, the end time of a timeSeries SHALL be earlier than or equal to latestTimeSeriesEndTime.
2: F	slotValueMin.		Element SHALL NOT be used.
2: F	slotValueMax.		Element SHALL NOT be used.
2: O	slotValueStepSize.		<p>If slotValueStepSize is set and slotValueMin as well as slotValueMax is not set, the elements value, maxValue and minValue of a slot SHALL be a multiple of slotValueStepSize.</p> <p>If slotValueStepSize is set and slotValueMin is also set, the elements value, maxValue and minValue of a slot SHALL be slotValueMin plus a multiple of slotValueStepSize.</p> <p>If slotValueStepSize is set and slotValueMax is also set, the elements value, maxValue and minValue of a slot SHALL be slotValueMax minus a multiple of slotValueStepSize.</p> <p>The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
2: M	slotValueStepSize. number		SHALL be used.
2: O	slotValueStepSize. scale		MAY be used. If absent, a default value of "0" applies.
4: O	TimeSeries. timeSeriesConstraintsListData. timeSeriesConstraintsData.		
4: M	timeSeriesId	<m1>	SHALL be set as PRIMARY IDENTIFIER.
4: O	slotValueMin.		<p>The value, maxValue or minValue of a slot SHALL be equal to or higher than slotValueMin.</p> <p>The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
4: M	slotValueMin. number		SHALL be used.
4: O	slotValueMin. scale		MAY be used. If absent, a default value of "0" applies.
4: O	slotValueMax.		<p>The value, maxValue or minValue of a slot SHALL be equal to or lower than slotValueMax.</p> <p>The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
4: M	slotValueMax. number		SHALL be used.
4: O	slotValueMax. scale		MAY be used. If absent, a default value of "0" applies.

4: O	slotValueStepSize.		<p>If slotValueStepSize is set and slotValueMin as well as slotValueMax is not set, the elements value, maxValue and minValue of a slot SHALL be a multiple of slotValueStepSize.</p> <p>If slotValueStepSize is set and slotValueMin is also set, the elements value, maxValue and minValue of a slot SHALL be slotValueMin plus a multiple of slotValueStepSize.</p> <p>If slotValueStepSize is set and slotValueMax is also set, the elements value, maxValue and minValue of a slot SHALL be slotValueMax minus a multiple of slotValueStepSize.</p> <p>The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
4: M	slotValueStepSize. number		SHALL be used.
4: O	slotValueStepSize. scale		MAY be used. If absent, a default value of "0" applies.

Table 7: Content of Function "timeSeriesConstraintsListData" at Actor EV

3.2.1.2.2.3 Function "timeSeriesListData"

Scenario [{...}]: M/R/O [W]\C]	Element	Value	[High Level Mapping] Element and value rules
1: M	TimeSeries. timeSeriesListData. timeSeriesData.		
1: M	timeSeriesId	<m1>	PRIMARY IDENTIFIER of timeSeries
1: O	timePeriod		If set, neither timeSeriesSlot.duration nor timeSeriesSlot.timePeriod SHALL define times out of the bound of the overall timeSeries timePeriod. Only relative times SHALL be used (xs:duration). Relative times SHALL be interpreted relative to "now".
1: M	timePeriod. startTime	<xs:duration>	[CEVC-001]
1: M	timeSeriesSlot.		
1: O	timeSeriesSlot. timeSeriesSlotId		If timeSeriesSlotId is omitted it SHALL be interpreted as zero.
1: O	timeSeriesSlot. duration		<p>[CEVC-002] SHALL only contain values greater than zero seconds.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in the first timeSeriesSlot (slot with the lowest timeSeriesSlotId), the</p>

			<p>timePeriod. startTime of the overall timeSeries SHALL be set and the duration of the first timeSeriesSlot starts from there.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in a slot that is not the first slot, the duration starts after the previous slot is finished.</p>
1: O	timeSeriesSlot. value.		<p>[CEVC-004]</p> <p>A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If minValue is set the content of value SHALL be greater or equal. If maxValue is set the content of value SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
1: M	timeSeriesSlot. value. number	≥0	SHALL be used.
1: O	timeSeriesSlot. value. scale		MAY be used. If absent, a default value of "0" applies.
1: O	timeSeriesSlot. minValue.		<p>[CEVC-003]</p> <p>A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of minValue SHALL be smaller or equal. If value is not set but maxValue is set the content of minValue SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
1: M	timeSeriesSlot. minValue. number	≥0	SHALL be used.
1: O	timeSeriesSlot. minValue. scale		MAY be used. If absent, a default value of "0" applies.
1: O	timeSeriesSlot. maxValue.		<p>[CEVC-005]</p> <p>A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of maxValue SHALL be greater or equal. If value is not set but minValue is set the content of maxValue SHALL be greater or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
1: M	timeSeriesSlot. maxValue. number	≥0	SHALL be used.

1: O	timeSeriesSlot. maxValue. scale		MAY be used. If absent, a default value of "0" applies.
2: M \W	TimeSeries. timeSeriesListData. timeSeriesData. [CEVC-007]		
2: M \W	timeSeriesId	<m2>	PRIMARY IDENTIFIER of timeSeries
2: M \W	timePeriod		If set, neither timeSeriesSlot.duration nor timeSeriesSlot.timePeriod SHALL define times out of the bound of the overall timeSeries timePeriod. Only relative times SHALL be used (xs:duration). Relative times SHALL be interpreted relative to "now".
2: M \W	timePeriod. startTime	<xs:duration>	
2: M \W	timeSeriesSlot.		
2: M \W	timeSeriesSlot. timeSeriesSlotId		If timeSeriesSlotId is omitted it SHALL be interpreted as zero.
2: M \W	timeSeriesSlot. duration		<p>[CEVC-009], [CEVC-010] SHALL only contain values greater than zero seconds.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in the first timeSeriesSlot (slot with the lowest timeSeriesSlotId), the timePeriod. startTime of the overall timeSeries SHALL be set and the duration of the first timeSeriesSlot starts from there.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in a slot that is not the first slot, the duration starts after the previous slot is finished.</p> <p>The sum of timePeriod. startTime and the durations of all slots SHALL equal timePeriod. endTime if set.</p>
2: M \W	timeSeriesSlot. maxValue.		<p>[CEVC-010], [CEVC-014] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of maxValue SHALL be greater or equal. If value is not set but minValue is set the content of maxValue SHALL be greater or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
2: M \W	timeSeriesSlot. maxValue. number	≥ 0	SHALL be used.
2: M \W	timeSeriesSlot. maxValue. scale		MAY be used. If absent, a default value of "0" applies.
4: M	TimeSeries. timeSeriesListData. timeSeriesData.		
4: M	timeSeriesId	<m3>	PRIMARY IDENTIFIER of timeSeries

4: M	timePeriod		If set, neither timeSeriesSlot.duration nor timeSeriesSlot.timePeriod SHALL define times out of the bound of the overall timeSeries timePeriod. Only relative times SHALL be used (xs:duration). Relative times SHALL be interpreted relative to "now".
4: M	timePeriod. startTime	<xs:duration>	
4: M	timeSeriesSlot.		
4: M	timeSeriesSlot. timeSeriesSlotId		If timeSeriesSlotId is omitted it SHALL be interpreted as zero.
4: M	timeSeriesSlot. duration		SHALL only contain values greater than zero seconds. If timeSeriesSlot. timePeriod. startTime is not set in the first timeSeriesSlot (slot with the lowest timeSeriesSlotId), the timePeriod. startTime of the overall timeSeries SHALL be set and the duration of the first timeSeriesSlot starts from there. If timeSeriesSlot. timePeriod. startTime is not set in a slot that is not the first slot, the duration starts after the previous slot is finished.
4: O	timeSeriesSlot. value.		A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If minValue is set the content of value SHALL be greater or equal. If maxValue is set the content of value SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	timeSeriesSlot. value. number	≥ 0	SHALL be used.
4: O	timeSeriesSlot. value. scale		MAY be used. If absent, a default value of "0" applies.
4: O	timeSeriesSlot. minValue.		A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of minValue SHALL be smaller or equal. If value is not set but maxValue is set the content of minValue SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	timeSeriesSlot. minValue. number	≥ 0	SHALL be used.

4: O	timeSeriesSlot. minValue. scale		MAY be used. If absent, a default value of "0" applies.
4: M	timeSeriesSlot. maxValue.		[CEVC-008] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of maxValue SHALL be greater or equal. If value is not set but minValue is set the content of maxValue SHALL be greater or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	timeSeriesSlot. maxValue. number	≥ 0	SHALL be used.
4: O	timeSeriesSlot. maxValue. scale		MAY be used. If absent, a default value of "0" applies.

Table 8: Content of Function "timeSeriesListData" at Actor EV

3.2.1.2.3 Feature Type "IncentiveTable"

3.2.1.2.3.1 Function "incentiveTableDescriptionData"

Scenario {...}: M/R/O [W][C]	Element	Value	[High Level Mapping] Element and value rules
3: M	IncentiveTable. incentiveTableDescriptionData. incentiveTableDescription.		
3: M	tariffDescription.		
3: M	tariffDescription. tariffId	<x1>	SHALL be set as PRIMARY IDENTIFIER of an incentiveTable, if more than one incentiveTable is defined within IncentiveTable.
3: M	tariffDescription. tariffWriteable	true	SHALL be set to "true" if the incentive table is writeable by a client. Otherwise it may be omitted.
3: M (event)	tariffDescription. updateRequired	true	[CEVC-030], [CEVC-031] With updateRequired the server can request an update of writeable or changeable data related to the same PRIMARY IDENTIFIER from a client. The server SHALL ensure that only one responsible client is able to update the related data. To request an update the server SHALL set updateRequired to "true". Note: In this case, the server expects the responsible client to update the writeable or changeable data related

			to the same PRIMARY IDENTIFIER. However, also if updateRequired is set to "false" a server SHOULD in general allow updates of the data from the responsible client. The server SHALL set the updateRequired back to "false", as soon as "incentiveTableDescriptionData" was updated successfully (if writeable or changeable) OR the update of the other writeable or changeable data related to the same PRIMARY IDENTIFIER was successful. Note: The client does not need to stop an ongoing update process (e.g. if multiple Functions are written), when updateRequired is set back to "false". The server MAY choose to withdraw the update request at any time by setting updateRequired back to "false".
3: M	tariffDescription. scopeType	"simpleIncentiveTable"	
3: O	tariffDescription. slotIdSupport		If omitted, a default of "false" SHALL be applied. If set to "false" no incentiveSlot SHALL contain a timeSlotId. If set to "true" timeSlotId MAY be used within the incentiveSlot as described for the element timeSlotId within the Function incentiveTableData.
3: M \W	tier.		
3: M \W	tier. tierDescription.		
3: M \W	tier. tierDescription. tierId	<y1>	SHALL be set as SUB IDENTIFIER of tariffId, if the incentiveTable contains more than one tier. If the tierId is omitted, a default value of "1" SHALL be applied.
3: M \W	tier. tierDescription. tierType	"dynamicCost"	The tier has a cost incentive that MAY vary over time.
3: M \W	tier. tierDescription. boundaryDescription.		
3: M \W	tier. boundaryDescription. boundaryId	<z1>	SHALL be set as SUB IDENTIFIER of tierId, if the tier contains more than one boundary entry. If the boundaryId is omitted, a default value of "1" SHALL be applied.
3: M \W	tier. boundaryDescription. boundaryType	"powerBoundary"	The boundary SHALL be an electrical power. E.g. if the surplus power is consumed, more expensive power has to be used. Positive powerBoundary values describe

				which power can be consumed with certain incentives, while negative powerBoundary values describe which power can be produced with certain incentives.
3: M \W	tier. boundaryDescription. boundaryUnit	"W"		If set, the unit SHALL be applied to the lowerBoundaryValue and upperBoundaryValue.
3: M \W	tier. incentiveDescription.			[CEVC-023]
3: M \W	tier. incentiveDescription. incentiveId	<j1>		SHALL be set as SUB IDENTIFIER of tierId, if the tier contains more than one incentive entry. If the incentiveId is omitted, a default value of "1" SHALL be applied.
3: M \W	tier. incentiveDescription. incentiveType	3: M \W	"absoluteCost"	[CEVC-020] Each related value SHALL be an absolute cost with a specific currency. Positive values SHALL be interpreted as costs. Negative values SHALL be interpreted as profits. The currency SHALL be set and relate to Wh. E.g. EUR/Wh. For this incentive the Element tier. incentiveDescription. unit SHALL be ignored and only the Element tier. incentiveDescription. currency is relevant.
		3: M \W	"relativeCost"	[CEVC-019] The values SHALL have no currency and the unit SHALL be percentage (pct). The value SHALL be in the range from 0% to 100%. With 0% representing lowest costs and 100% highest costs. For this incentive the Elements tier. incentiveDescription. unit and tier. incentiveDescription. currency SHALL be ignored.
		3: M \W	"renewableEnergyPercentage"	[CEVC-021] The unit of the incentive SHALL relate to percentage (pct) in this case. The value SHALL be in the range from 0% to 100%. For this incentive the Elements tier. incentiveDescription. unit and tier. incentiveDescription. currency SHALL be ignored.
		3: M \W	"co2Emission"	[CEVC-022] The unit of the incentive SHALL relate to "kg/Wh". Only positive values SHALL be used.

				For this incentive the Elements tier. incentiveDescription. unit and tier. incentiveDescription. currency SHALL be ignored.
3: M \W	tier. incentiveDescription. currency			If set, the currency SHALL be applied to the value of the incentive

Table 9: Content of Function "incentiveTableDescriptionData" at Actor EV

3.2.1.2.3.2 Function "incentiveTableConstraintsData"

Scenario [...]: M/R/O [W][\C]	Element	Value	[High Level Mapping] Element and value rules
3: R	IncentiveTable. incentiveTableConstraintsData. incentiveTableConstraints.		
3: M	tariff.		
3: M	tariff. tariffId	<x1>	SHALL be set as PRIMARY IDENTIFIER of a incentiveTable, if more than one incentiveTable is defined within IncentiveTable.
3: R	tariffConstraints.		
3: R	tariffConstraints. maxTiersPerTariff	≥3	[CEVC-055] If set the incentiveTable SHALL NOT include more tiers than given in maxTiersPerTariff.
3: R	tariffConstraints. maxBoundariesPerTier	≥1	[CEVC-055] If set the tier within the incentiveTable SHALL NOT include more boundaries in a tier than given in maxBoundariesPerTier.
3: R	tariffConstraints. maxIncentivesPerTier	≥3	[CEVC-056] If set the tier within the incentiveTable SHALL NOT include more incentives than given in maxIncentivesPerTier.
3: R	incentiveSlotConstraints.		
3: R	incentiveSlotConstraints. slotCountMax		If set the incentiveTable SHALL NOT include more slots than given in slotCountMax.

Table 10: Content of Function "incentiveTableConstraintsData" at Actor EV

1000 3.2.1.2.3.3 Function "incentiveTableData"

Scenario [{...}]: M/R/O [W][C]	Element	Value	[High Level Mapping] Element and value rules
3: M \W	IncentiveTable. incentiveTableData. incentiveTable.		
3: M \W	tariff.		
3: M \W	tariff. tariffId	<x1>	SHALL be set as PRIMARY IDENTIFIER.
3: M \W	incentiveSlot.		
3: M \W	incentiveSlot. timeInterval.		SHALL be set if an incentive has a time restriction or the incentiveTable contains more than one incentiveSlot entry. The timeInterval of different incentiveSlots within an incentiveTable SHALL NOT overlap in time.
3: M \W	incentiveSlot. timeInterval. startTime.		
3: M \W	incentiveSlot. timeInterval. startTime. relative		
3: M \W	incentiveSlot. timeInterval. endTime.		SHALL be set for the last slot. SHALL be set if a time gap follows after this slot. MAY be set for all other slots.
3: M \W	incentiveSlot. timeInterval. endTime. relative		
3: M \W	incentiveSlot. tier.		[CEVC-028]
3: M \W	incentiveSlot. tier. tier.		
3: M \W	incentiveSlot. tier. tier. tierId	<y1>	SUB IDENTIFIER of tariffId to identify a tier of a tariff. SHALL be set if the incentiveTable contains more than one tier. If the tierId is omitted, a default value of "1" SHALL be applied.
3: M \W	incentiveSlot. tier. boundary.		The boundary list defines the boundaries of a tier. The boundary range of different tiers within an incentiveSlot defined by lowerBoundaryValue and upperBoundaryValue SHALL NOT overlap for boundaries with the same boundaryId. The boundary range of a tier includes all values that are equal or greater than lowerBoundaryValue and lower than upperBoundaryValue.
3: M \W	incentiveSlot. tier. boundary. boundaryId	<z1>	SHALL be used as SUB IDENTIFIER of tierId. SHALL be set if the tier contains more than one boundary entry. If the boundaryId is omitted, a default value of "1" SHALL be applied.

3: M \W	incentiveSlot. tier. boundary. lowerBoundaryValue.		<p>[CEVC-028] Within a boundary lowerBoundaryValue SHALL be smaller than upperBoundaryValue.</p> <p>Lower and upper boundaries of adjacent tiers: If lowerBoundaryValue is omitted and upperBoundaryValue is set, the lowerBoundaryValue SHALL be interpreted as if it is equal (without a gap) to the upperBoundaryValue of the next lower tier within the incentiveSlot. Gaps between boundaries of adjacent "tier" instances can only be modelled by setting the upperBoundaryValue of the lower "tier" AND the lowerBoundaryValue of the higher "tier" instance explicitly.</p> <p>Minus infinity lowerBoundary rule: If lowerBoundaryValue is omitted and there exists no lower tier with a lowerBoundaryValue or upperBoundaryValue within the incentiveSlot, the omitted lowerBoundaryValue SHALL be interpreted as minus infinity.</p> <p>The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
3: M \W	incentiveSlot. tier. boundary. lowerBoundaryValue. number	≥0	SHALL be used.
3: M \W	incentiveSlot. tier. boundary. lowerBoundaryValue. scale		MAY be used. If absent, a default value of "0" applies.
3: M \W	incentiveSlot. tier. incentive.		
3: M \W	incentiveSlot. tier. incentive. incentiveld	<j1>	SHALL be used as SUB IDENTIFIER of tierId. SHALL be set if the tier contains more than one incentive entry. If the incentiveld is omitted, a default value of "1" SHALL be applied.
3: M \W	incentiveSlot. tier. incentive. value.		<p>[CEVC-054] The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
3: M \W	incentiveSlot. tier. incentive. value. number		SHALL be used.

3: M \W	incentiveSlot. tier. incentive. value. scale		MAY be used. If absent, a default value of "0" applies.
---------	---	--	---

Table 11: Content of Function "incentiveTableData" at Actor EV

3.2.1.3 Client data - Specializations

3.2.1.3.1 Topic "DeviceDiagnosis"

3.2.1.3.1.1 Specialization "DeviceDiagnosis_FailureState"

Scenario [...]: M/R/O [\W][\C]	Element	Value	[High Level Mapping] Element and value rules
7: M 8: M	DeviceDiagnosis. deviceDiagnosisStateData.		
7: M 8: M	operatingState	"normalOperation"	
		"failure"	
7: M 8: M	lastErrorCode		[CEVC-051], [CEVC-053]

Table 12: Content of Specialization "DeviceDiagnosis_FailureState" at Actor EV

3.2.1.3.2 Topic "Heartbeat"

3.2.1.3.2.1 Specialization "Heartbeat_Timeout60Seconds"

Scenario [...]: M/R/O [\W][\C]	Element	Value	[High Level Mapping] Element and value rules
5: M 6: M	DeviceDiagnosis. deviceDiagnosisStateData.		
5: M 6: M	timestamp		SHALL hold the time of creation
5: M 6: M	heartbeatCounter		The value of the heartbeatCounter element SHALL be increased after every <i>heartbeatTimeout</i> (NOT with every sending of this Function). The <i>deviceDiagnosisHeartbeatData</i> Function can not only be sent initially by the device itself, but can be requested by another device, too. In this case, the element <i>heartbeatCounter</i>

			SHALL NOT be incremented and the heartbeatTimeout has (as always) its fixed value (i.e. not the remaining time to the next (automatic) notification by the device).
5: M 6: M	heartbeatTimeout	≤60 seconds	[CEVC-046], [CEVC-049] deviceDiagnosisHeartbeatData SHALL be sent at least each heartbeatTimeout period.

Table 13: Content of Specialization "Heartbeat_Timeout60Seconds" at Actor EV

3.2.2 Energy Guard

3.2.2.1 Resource hierarchy

The Energy Broker and Energy Guard MAY be represented by the same CEM Entity.

Definition of the Element "nodeManagementUseCaseData. useCaseInformation. actor" within the Use Case Discovery (if supported; please refer to [ProtocolSpecification]):

- "CEM" (without the surrounding quotes) SHALL be used if Energy Guard and Energy Broker are represented by the same CEM Entity.
- "EnergyGuard" (without the surrounding quotes) SHALL be used if Energy Guard and Energy Broker are represented by different CEM Entities.

The following diagram provides an overview of the Actor Energy Guard resource hierarchy.

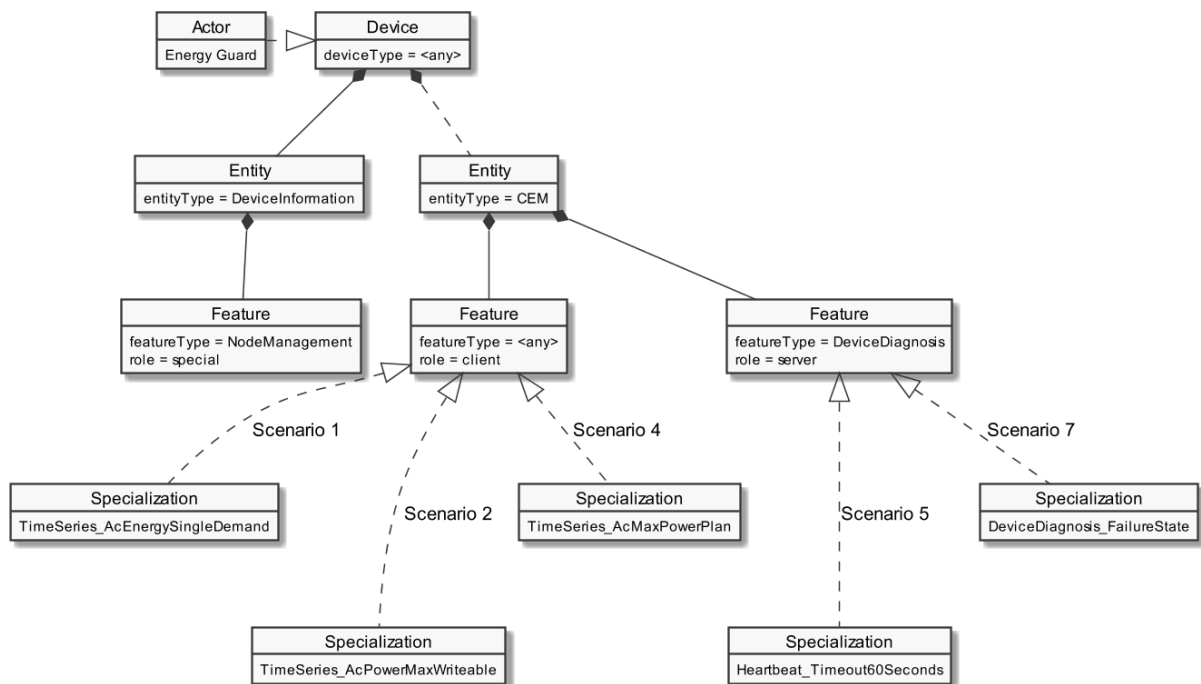


Figure 11: Actor "Energy Guard" overview

The ""Actor ... overview" diagram rules" section describes how to interpret the diagram above. See the "Specializations" section for more information regarding the Specializations given in the diagram above.

Note: The entityType "DeviceInformation" with the featureType "NodeManagement" is required by the SPINE protocol and therefore SHALL be supported. Both types are added in the figure for completeness but are not directly linked to the Use Case.

The Use Case specific data follow behind the entityType "CEM". The Specializations represent the Scenario specific data that has to be supported for each Scenario and are realized with the according featureTypes.

If a Specialization is connected to a Feature with the role "client", the Actor has a client role for this data. This means the Actor accesses the data set described by the Specialization at a corresponding server Feature. Further details are described in the sub-section "Client data - Specializations".

If a Specialization is connected to a Feature with the role "server", the Actor has the server role for this data. This means the Actor must provide the corresponding data set of the Specialization on its Features. Further details are described in the sub-section "Server data - Resources".

3.2.2.2 Server data - Resources

3.2.2.2.1 Overview

Behind the entityType "CEM" the Actor Energy Guard SHALL offer the Feature Types and Functions given in the table below.

Feature Type	Scenario: M/R/O	Function	Possible operations
DeviceDiagnosis	5: M	deviceDiagnosisHeartbeatData	read (M). partial (O)
	7: M	deviceDiagnosisStateData	read (M). partial (O)

Table 14: Feature Types and Functions used within this Use Case by the Actor Energy Guard

For each of these Feature Types the following rule applies: There SHALL be at maximum one Feature with the Feature Type in the Entity.

Note: As a consequence of the previous rule, an implementation may need to have Feature data from different Scenarios/Specializations or even Use Cases in a given Feature.

The Scenario number shows in which Scenarios the Energy Guard acts as server and which Feature Types and Functions are relevant in each Scenario.

A detailed definition of the Elements and values that shall be supported in each Function is given in the following sub-sections.

Note: If in the table above "partial" read is not mentioned or is only optional, it still might be mandatory to support partial notifications. The details of "partial" support are described within the Scenario sections.

Note: The presence indications stated above are meant relative to the ones of the according Scenario stated in Table 1. I.e. if a Scenario is optional ("O") and a Feature Type is mandatory ("M") the Feature Type must only be supported if the Scenario is supported, too.

Note: Further Features MAY be implemented on the same Entities, as well as further Functions MAY be implemented in the used Entities.

3.2.2.2.2 Feature Type "DeviceDiagnosis"

3.2.2.2.2.1 Function "deviceDiagnosisHeartbeatData"

Scenario [...]: M/R/O [W][C]	Element	Value	[High Level Mapping] Element and value rules
5: M	DeviceDiagnosis. deviceDiagnosisHeartbeatData.		
5: M	timestamp		SHALL hold the time of creation
5: M	heartbeatCounter		The value of the heartbeatCounter element SHALL be increased after every <i>heartbeatTimeout</i> (NOT with every sending of this Function). The <i>deviceDiagnosisHeartbeatData</i> Function can not only be sent initially by the device itself, but can be requested by another device, too. In this case, the element <i>heartbeatCounter</i> SHALL NOT be incremented and the <i>heartbeatTimeout</i> has (as always) its fixed value (i.e. not the remaining time to the next (automatic) notification by the device).
5: M	heartbeatTimeout	≤60 seconds	[CEVC-046] deviceDiagnosisHeartbeatData SHALL be sent at least each heartbeatTimeout period.

Table 15: Content of Function "deviceDiagnosisHeartbeatData" at Actor Energy Guard

3.2.2.2.2.2 Function "deviceDiagnosisStateData"

Scenario [...]: M/R/O [W][C]	Element	Value	[High Level Mapping] Element and value rules
7: M	DeviceDiagnosis. deviceDiagnosisStateData.		
7: M	operatingState	"normalOperation"	
		"failure"	
7: M	lastErrorCode		[CEVC-051]

Table 16: Content of Function "deviceDiagnosisStateData" at Actor Energy Guard

1068 **3.2.2.3 Client data - Specializations**

1069 3.2.2.3.1 Topic "TimeSeries"

1070 3.2.2.3.1.1 Specialization "TimeSeries_AcEnergySingleDemand"

Scenario {...}: M/R/O [W][C]	Element	Value	[High Level Mapping] Element and value rules
1: M	TimeSeries. timeSeriesDescriptionListData. timeSeriesDescriptionData.		
1: M	timeSeriesId	<m1>	PRIMARY IDENTIFIER of timeSeries.
1: M	timeSeriesType	"singleDemand"	"singleDemand" SHALL be used to model a single demand. Only one timeslot SHALL be used.
1: M	timeSeriesWriteable	"false"	If set to false or omitted, this timeSeriesData SHALL NOT be writeable.
1: M	unit	"Wh"	The unit SHALL be applied to the value in timeSeriesData
1: M	TimeSeries. timeSeriesListData. timeSeriesData.		
1: M	timeSeriesId	<m1>	PRIMARY IDENTIFIER of timeSeries
1: M	timePeriod		If set, neither timeSeriesSlot.duration nor timeSeriesSlot.timePeriod SHALL define times out of the bound of the overall timeSeries timePeriod. Only relative times SHALL be used (xs:duration). Relative times SHALL be interpreted relative to "now".
1: M	timePeriod. startTime	<xs:duration>	[CEVC-001]
1: M	timeSeriesSlot.		
1: M	timeSeriesSlot. timeSeriesSlotId		If timeSeriesSlotId is omitted it SHALL be interpreted as zero.
1: M	timeSeriesSlot. duration		<p>[CEVC-002] SHALL only contain values greater than zero seconds.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in the first timeSeriesSlot (slot with the lowest timeSeriesSlotId), the timePeriod. startTime of the overall timeSeries SHALL be set and the duration of the first timeSeriesSlot starts from there.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in a slot that is not the first slot, the duration starts after the previous slot is finished.</p>

1: M	timeSeriesSlot. value.		[CEVC-004] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If minValue is set the content of value SHALL be greater or equal. If maxVale is set the content of value SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	timeSeriesSlot. value. number	≥ 0	SHALL be used.
1: M	timeSeriesSlot. value. scale		MAY be used. If absent, a default value of "0" applies.
1: M	timeSeriesSlot. minVale.		[CEVC-003] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of minVale SHALL be smaller or equal. If value is not set but maxVale is set the content of minVale SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	timeSeriesSlot. minVale. number	≥ 0	SHALL be used.
1: M	timeSeriesSlot. minVale. scale		MAY be used. If absent, a default value of "0" applies.
1: M	timeSeriesSlot. maxVale.		[CEVC-005] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of maxVale SHALL be greater or equal. If value is not set but minVale is set the content of maxVale SHALL be greater or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	timeSeriesSlot. maxVale. number	≥ 0	SHALL be used.
1: M	timeSeriesSlot. maxVale. scale		MAY be used. If absent, a default value of "0" applies.
1: R	TimeSeries. timeSeriesConstraintsListData. timeSeriesConstraintsData.		
1: M	timeSeriesId	<m1>	SHALL be set as PRIMARY IDENTIFIER.
1: R	slotValueMin.		The value, maxVale or minVale of a slot SHALL be equal to or higher than slotValueMin.

			The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	slotValueMin. number		SHALL be used.
1: M	slotValueMin. scale		MAY be used. If absent, a default value of "0" applies.
1: R	slotValueMax.		The value, max Value or min Value of a slot SHALL be equal to or lower than slotValueMax. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	slotValueMax. number		SHALL be used.
1: M	slotValueMax. scale		MAY be used. If absent, a default value of "0" applies.
1: R	slotValueStepSize.		If slotValueStepSize is set and slotValueMin as well as slotValueMax is not set, the elements value, max Value and min Value of a slot SHALL be a multiple of slotValueStepSize. If slotValueStepSize is set and slotValueMin is also set, the elements value, max Value and min Value of a slot SHALL be slotValueMin plus a multiple of slotValueStepSize. If slotValueStepSize is set and slotValueMax is also set, the elements value, max Value and min Value of a slot SHALL be slotValueMax minus a multiple of slotValueStepSize. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	slotValueStepSize. number		SHALL be used.
1: M	slotValueStepSize. scale		MAY be used. If absent, a default value of "0" applies.

Table 17: Content of Specialization "TimeSeries_AcEnergySingleDemand" at Actor Energy Guard

1073 3.2.2.3.1.2 Specialization "TimeSeries_AcPowerMaxWriteable"

Scenario [{...}]: M/R/O [\W][\C]	Element	Value	[High Level Mapping] Element and value rules
2: M	TimeSeries. timeSeriesDescriptionListData. timeSeriesDescriptionData.		
2: M	timeSeriesId	<m2>	PRIMARY IDENTIFIER of timeSeries.
2: M	timeSeriesType	"constraints"	"constraints" SHALL be used to model certain value constraints or limits over time.
2: M	timeSeriesWriteable	"true"	If set to false or omitted, this timeSeriesData SHALL NOT be writeable.
2: M	updateRequired		[CEVC-015] With updateRequired the server can request an update of writeable or changeable data related to the same PRIMARY IDENTIFIER from a client. The server SHALL ensure that only one responsible client is able to update the related data. To request an update the server SHALL set updateRequired to "true". Note: In this case, the server expects the responsible client to update the writeable or changeable data related to the same PRIMARY IDENTIFIER. However, also if updateRequired is set to "false" a server SHOULD in general allow updates of the data from the responsible client. The server SHALL set the updateRequired back to "false", as soon as "timeSeriesDescriptionListData" was updated successfully (if writeable or changeable) OR the update of the other writeable or changeable data related to the same PRIMARY IDENTIFIER was successful. Note: The client does not need to stop an ongoing update process (e.g. if multiple Functions are written), when updateRequired is set back to "false". The server MAY choose to withdraw the update request at any time by setting updateRequired back to "false".
2: M	unit	"W"	The unit SHALL be applied to the value in timeSeriesData
2: M \W	TimeSeries. timeSeriesListData. timeSeriesData. [CEVC-007]		
2: M \W	timeSeriesId	<m2>	PRIMARY IDENTIFIER of timeSeries
2: M \W	timePeriod		If set, neither timeSeriesSlot.duration nor timeSeriesSlot.timePeriod SHALL define

			times out of the bound of the overall timeSeries timePeriod. Only relative times SHALL be used (xs:duration). Relative times SHALL be interpreted relative to "now".
2: M \W	timePeriod. startTime	<xs:duration>	
2: M \W	timeSeriesSlot.		
2: M \W	timeSeriesSlot. timeSeriesSlotId		If timeSeriesSlotId is omitted it SHALL be interpreted as zero.
2: M \W	timeSeriesSlot. duration		<p>[CEVC-008], [CEVC-010] SHALL only contain values greater than zero seconds.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in the first timeSeriesSlot (slot with the lowest timeSeriesSlotId), the timePeriod. startTime of the overall timeSeries SHALL be set and the duration of the first timeSeriesSlot starts from there.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in a slot that is not the first slot, the duration starts after the previous slot is finished.</p> <p>The sum of timePeriod. startTime and the durations of all slots SHALL equal timePeriod. endTime if set.</p>
2: M \W	timeSeriesSlot. maxValue.		<p>[CEVC-010], [CEVC-014] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of maxValue SHALL be greater or equal. If value is not set but minValue is set the content of maxValue SHALL be greater or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
2: M \W	timeSeriesSlot. maxValue. number	≥0	SHALL be used.
2: M \W	timeSeriesSlot. maxValue. scale		MAY be used. If absent, a default value of "0" applies.
2: M	TimeSeries. timeSeriesConstraintsListData. timeSeriesConstraintsData.		
2: M	timeSeriesId	<m2>	SHALL be set as PRIMARY IDENTIFIER.
2: M	slotCountMax	(1024)	If set, the corresponding timeSeriesData SHALL NOT have more slots.
2: R	earliestTimeSeriesStartTime		If set the start time of a timeSeries SHALL be later than or equal to earliestTimeSeriesStartTime.
2: R	latestTimeSeriesEndTime	(PT168H)	[CEVC-012]

			If set, the end time of a timeSeries SHALL be earlier than or equal to latestTimeSeriesEndTime.
2: R	slotValueStepSize.		<p>If slotValueStepSize is set and slotValueMin as well as slotValueMax is not set, the elements value, maxVale and minVale of a slot SHALL be a multiple of slotValueStepSize.</p> <p>If slotValueStepSize is set and slotValueMin is also set, the elements value, maxVale and minVale of a slot SHALL be slotValueMin plus a multiple of slotValueStepSize.</p> <p>If slotValueStepSize is set and slotValueMax is also set, the elements value, maxVale and minVale of a slot SHALL be slotValueMax minus a multiple of slotValueStepSize.</p> <p>The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
2: M	slotValueStepSize. number		SHALL be used.
2: M	slotValueStepSize. scale		MAY be used. If absent, a default value of "0" applies.

Table 18: Content of Specialization "TimeSeries_AcPowerMaxWriteable" at Actor Energy Guard

3.2.2.3.1.3 Specialization "TimeSeries_AcMaxPowerPlan"

Scenario [...]: M/R/O [W][\C]	Element	Value	[High Level Mapping] Rule and description of elements and values
4: M	TimeSeries. timeSeriesDescriptionListData. timeSeriesDescriptionData.		
4: M	timeSeriesId	<m3>	PRIMARY IDENTIFIER of timeSeries.
4: M	timeSeriesType	"plan"	"plan" SHALL be used to model how a certain value will change over time.
4: M	timeSeriesWriteable	"false"	If set to false or omitted, this timeSeriesData SHALL NOT be writeable.
4: M	measurementId	<k1>	FOREIGN IDENTIFIER, links data of other Features to the timeSeries.
4: M	unit	"W"	The unit SHALL be applied to the value in timeSeriesData
4: M	TimeSeries. timeSeriesListData. timeSeriesData.		
4: M	timeSeriesId	<m3>	PRIMARY IDENTIFIER of timeSeries

4: M	timePeriod		If set, neither timeSeriesSlot.duration nor timeSeriesSlot.timePeriod SHALL define times out of the bound of the overall timeSeries timePeriod. Only relative times SHALL be used (xs:duration). Relative times SHALL be interpreted relative to "now".
4: M	timePeriod. startTime	<xs:duration>	
4: M	timeSeriesSlot.		
4: M	timeSeriesSlot. timeSeriesSlotId		If timeSeriesSlotId is omitted it SHALL be interpreted as zero.
4: M	timeSeriesSlot. duration		SHALL only contain values greater than zero seconds. If timeSeriesSlot. timePeriod. startTime is not set in the first timeSeriesSlot (slot with the lowest timeSeriesSlotId), the timePeriod. startTime of the overall timeSeries SHALL be set and the duration of the first timeSeriesSlot starts from there. If timeSeriesSlot. timePeriod. startTime is not set in a slot that is not the first slot, the duration starts after the previous slot is finished.
4: M	timeSeriesSlot. value.		A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If minValue is set the content of value SHALL be greater or equal. If maxValue is set the content of value SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	timeSeriesSlot. value. number	≥0	SHALL be used.
4: M	timeSeriesSlot. value. scale		MAY be used. If absent, a default value of "0" applies.
4: M	timeSeriesSlot. minValue.		A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of minValue SHALL be smaller or equal. If value is not set but maxValue is set the content of minValue SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	timeSeriesSlot. minValue. number	≥0	SHALL be used.
4: M	timeSeriesSlot. minValue. scale		MAY be used. If absent, a default value of "0" applies.

4: M	timeSeriesSlot. maxValue.		[CEVC-008] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of maxValue SHALL be greater or equal. If value is not set but minValue is set the content of maxValue SHALL be greater or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	timeSeriesSlot. maxValue. number	≥0	SHALL be used.
4: M	timeSeriesSlot. maxValue. scale		MAY be used. If absent, a default value of "0" applies.
4: R	TimeSeries. timeSeriesConstraintsListData. timeSeriesConstraintsData.		
4: R	timeSeriesId	<m3>	SHALL be set as PRIMARY IDENTIFIER.
4: R	slotValueMin.		The value, maxValue or minValue of a slot SHALL be equal to or higher than slotValueMin. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	slotValueMin. number		SHALL be used.
4: M	slotValueMin. scale		MAY be used. If absent, a default value of "0" applies.
4: R	slotValueMax.		The value, maxValue or minValue of a slot SHALL be equal to or lower than slotValueMax. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	slotValueMax. number		SHALL be used.
4: M	slotValueMax. scale		MAY be used. If absent, a default value of "0" applies.
4: R	slotValueStepSize.		If slotValueStepSize is set and slotValueMin as well as slotValueMax is not set, the elements value, maxValue and minValue of a slot SHALL be a multiple of slotValueStepSize. If slotValueStepSize is set and slotValueMin is also set, the elements value, maxValue and minValue of a slot SHALL be slotValueMin plus a multiple of slotValueStepSize. If slotValueStepSize is set and slotValueMax is also set, the elements value, maxValue and minValue of a slot SHALL be slotValueMax minus a multiple of slotValueStepSize.

			The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	slotValueStepSize. number		SHALL be used.
4: M	slotValueStepSize. scale		MAY be used. If absent, a default value of "0" applies.

Table 19: Content of Specialization "TimeSeries_AcMaxPowerPlan" at Actor Energy Guard

3.2.3 Energy Broker

3.2.3.1 Resource hierarchy

The Energy Broker and Energy Guard MAY be represented by the same CEM Entity.

Definition of the Element "nodeManagementUseCaseData. useCaseInformation. actor" within the Use Case Discovery (if supported; please refer to [ProtocolSpecification]):

- "CEM" (without the surrounding quotes) SHALL be used if Energy Guard and Energy Broker are represented by the same CEM Entity.
- "EnergyBroker" (without the surrounding quotes) SHALL be used if Energy Guard and Energy Broker are represented by different CEM Entities.

The following diagram provides an overview of the Actor Energy Broker resource hierarchy.

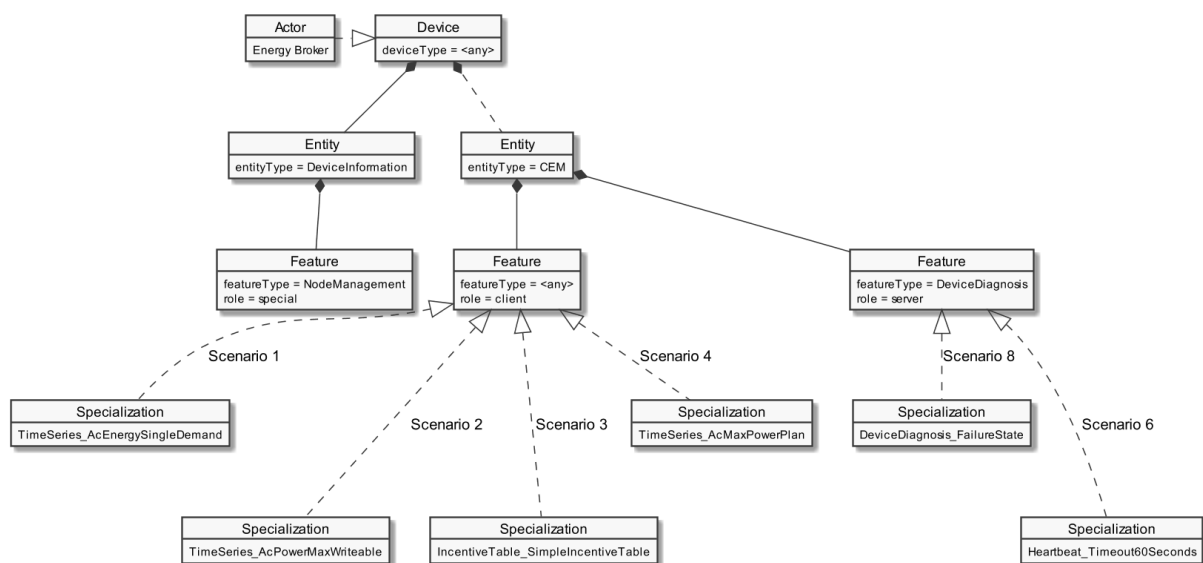


Figure 12: Actor "Energy Broker" overview

The "Actor ... overview" diagram rules" section describes how to interpret the diagram above. See the "Specializations" section for more information regarding the Specializations given in the diagram above.

Note: The entityType "DeviceInformation" with the featureType "NodeManagement" is required by the SPINE protocol and therefore SHALL be supported. Both types are added in the figure for completeness but are not directly linked to the Use Case.

1097 The Use Case specific data follow behind the entityType "CEM". The Specializations represent the
 1098 Scenario specific data that has to be supported for each Scenario and are realized with the according
 1099 featureTypes.

1100 If a Specialization is connected to a Feature with the role "client", the Actor has a client role for this
 1101 data. This means the Actor accesses the data set described by the Specialization at a corresponding
 1102 server Feature. Further details are described in the sub-section "Client data - Specializations".

1103 If a Specialization is connected to a Feature with the role "server", the Actor has the server role for
 1104 this data. This means the Actor must provide the corresponding data set described by the
 1105 Specialization on its Features. Further details are described in the sub-section "Server data -
 1106 Resources".

1107

1108 **3.2.3.2 Server data - Resources**

1109 **3.2.3.2.1 Overview**

1110 Behind the entityType "CEM" the Actor Energy Broker SHALL offer the Feature Types and Functions
 1111 given in the table below.

Feature Type	Scenario: M/R/O	Function	Possible operations
DeviceDiagnosis	6: M	deviceDiagnosisHeartbeatData	read (M). partial (O)
	8: M	deviceDiagnosisStateData	read (M). partial (O)

1112 *Table 20: Feature Types and Functions used within this Use Case by the Actor Energy Broker*

1113 For each of these Feature Types the following rule applies: There SHALL be at maximum one Feature
 1114 with the Feature Type in the Entity.

1115 Note: As a consequence of the previous rule, an implementation may need to have Feature data
 1116 from different Scenarios/Specializations or even Use Cases in a given Feature.

1117 The Scenario number shows in which Scenarios the Energy Broker acts as server and which Feature
 1118 Types and Functions are relevant in each Scenario.

1119 A detailed definition of the Elements and values that shall be supported in each Function is given in
 1120 the following sub-sections.

1121 Note: If in the table above "partial" read is not mentioned or is only optional, it still might be
 1122 mandatory to support partial notifications. The details of "partial" support are described within the
 1123 Scenario sections.

1124 Note: The presence indications stated above are meant relative to the ones of the according Scenario
 1125 stated in Table 1. I.e. if a Scenario is optional ("O") and a Feature Type is mandatory ("M") the
 1126 Feature Type must only be supported if the Scenario is supported, too.

1127 Note: Further Features MAY be implemented on the same Entities, as well as further Functions MAY
 1128 be implemented in the used Entities.

1129 3.2.3.2.2 Feature Type "DeviceDiagnosis"

1130 3.2.3.2.2.1 Function "deviceDiagnosisHeartbeatData"

Scenario [{...}]: M/R/O [\W][\C]	Element	Value	[High Level Mapping] Element and value rules
6: M	DeviceDiagnosis. deviceDiagnosisHeartbeatData.		
6: M	timestamp		SHALL hold the time of creation
6: M	heartbeatCounter		The value of the heartbeatCounter element SHALL be increased after every <i>heartbeatTimeout</i> (NOT with every sending of this Function). The <i>deviceDiagnosisHeartbeatData</i> Function can not only be sent initially by the device itself, but can be requested by another device, too. In this case, the element <i>heartbeatCounter</i> SHALL NOT be incremented and the <i>heartbeatTimeout</i> has (as always) its fixed value (i.e. not the remaining time to the next (automatic) notification by the device).
6: M	heartbeatTimeout	≤60 seconds	[CEVC-049] deviceDiagnosisHeartbeatData SHALL be sent at least each heartbeatTimeout period.

1131 Table 21: Content of Function "deviceDiagnosisHeartbeatData" at Actor Energy Broker

1132

1133 3.2.3.2.2.2 Function "deviceDiagnosisStateData"

Scenario [{...}]: M/R/O [\W][\C]	Element	Value	[High Level Mapping] Element and value rules
8: M	DeviceDiagnosis. deviceDiagnosisStateData.		
8: M	operatingState	"normalOperation"	
		"failure"	
8: M	lastErrorCode		[CEVC-053]

1134 Table 22: Content of Function "deviceDiagnosisStateData" at Actor Energy Broker

1135

1136 **3.2.3.3 Client data - Specializations**

1137 3.2.3.3.1 Topic "TimeSeries"

1138 3.2.3.3.1.1 Specialization "TimeSeries_AcEnergySingleDemand"

Scenario {...}: M/R/O [W][C]	Element	Value	[High Level Mapping] Element and value rules
1: M	TimeSeries. timeSeriesDescriptionListData. timeSeriesDescriptionData.		
1: M	timeSeriesId	<m1>	PRIMARY IDENTIFIER of timeSeries.
1: M	timeSeriesType	"singleDemand"	"singleDemand" SHALL be used to model a single demand. Only one timeslot SHALL be used.
1: M	timeSeriesWriteable	"false"	If set to false or omitted, this timeSeriesData SHALL NOT be writeable.
1: M	unit	"Wh"	The unit SHALL be applied to the value in timeSeriesData
1: M	TimeSeries. timeSeriesListData. timeSeriesData.		
1: M	timeSeriesId	<m1>	PRIMARY IDENTIFIER of timeSeries
1: M	timePeriod		If set, neither timeSeriesSlot.duration nor timeSeriesSlot.timePeriod SHALL define times out of the bound of the overall timeSeries timePeriod. Only relative times SHALL be used (xs:duration). Relative times SHALL be interpreted relative to "now".
1: M	timePeriod. startTime	<xs:duration>	[CEVC-001]
1: M	timeSeriesSlot.		
1: M	timeSeriesSlot. timeSeriesSlotId		If timeSeriesSlotId is omitted it SHALL be interpreted as zero.
1: M	timeSeriesSlot. duration		<p>[CEVC-002] SHALL only contain values greater than zero seconds.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in the first timeSeriesSlot (slot with the lowest timeSeriesSlotId), the timePeriod. startTime of the overall timeSeries SHALL be set and the duration of the first timeSeriesSlot starts from there.</p> <p>If timeSeriesSlot. timePeriod. startTime is not set in a slot that is not the first slot, the duration starts after the previous slot is finished.</p>

1: M	timeSeriesSlot. value.		[CEVC-004] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If minValue is set the content of value SHALL be greater or equal. If maxVale is set the content of value SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	timeSeriesSlot. value. number	≥ 0	SHALL be used.
1: M	timeSeriesSlot. value. scale		MAY be used. If absent, a default value of "0" applies.
1: M	timeSeriesSlot. minVale.		[CEVC-003] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of minVale SHALL be smaller or equal. If value is not set but maxVale is set the content of minVale SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	timeSeriesSlot. minVale. number	≥ 0	SHALL be used.
1: M	timeSeriesSlot. minVale. scale		MAY be used. If absent, a default value of "0" applies.
1: M	timeSeriesSlot. maxVale.		[CEVC-005] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of maxVale SHALL be greater or equal. If value is not set but minVale is set the content of maxVale SHALL be greater or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	timeSeriesSlot. maxVale. number	≥ 0	SHALL be used.
1: M	timeSeriesSlot. maxVale. scale		MAY be used. If absent, a default value of "0" applies.
1: M	TimeSeries. timeSeriesConstraintsListData. timeSeriesConstraintsData.		
1: M	timeSeriesId	<m1>	SHALL be set as PRIMARY IDENTIFIER.
1: M	slotValueMin.		The value, maxVale or minVale of a slot SHALL be equal to or higher than slotValueMin.

			The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	slotValueMin. number		SHALL be used.
1: M	slotValueMin. scale		MAY be used. If absent, a default value of "0" applies.
1: M	slotValueMax.		The value, max Value or min Value of a slot SHALL be equal to or lower than slotValueMax. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	slotValueMax. number		SHALL be used.
1: M	slotValueMax. scale		MAY be used. If absent, a default value of "0" applies.
1: M	slotValueStepSize.		If slotValueStepSize is set and slotValueMin as well as slotValueMax is not set, the elements value, max Value and min Value of a slot SHALL be a multiple of slotValueStepSize. If slotValueStepSize is set and slotValueMin is also set, the elements value, max Value and min Value of a slot SHALL be slotValueMin plus a multiple of slotValueStepSize. If slotValueStepSize is set and slotValueMax is also set, the elements value, max Value and min Value of a slot SHALL be slotValueMax minus a multiple of slotValueStepSize. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
1: M	slotValueStepSize. number		SHALL be used.
1: M	slotValueStepSize. scale		MAY be used. If absent, a default value of "0" applies.

Table 23: Content of Specialization "TimeSeries_AcEnergySingleDemand" at Actor Energy Broker

1141 3.2.3.3.1.2 Specialization "TimeSeries_AcPowerMaxWriteable"

Scenario [{...}]: M/R/O [\W][\C]	Element	Value	[High Level Mapping] Element and value rules
2: M	TimeSeries. timeSeriesDescriptionListData. timeSeriesDescriptionData.		
2: M	timeSeriesId	<m2>	PRIMARY IDENTIFIER of timeSeries.
2: M	timeSeriesType	"constraints"	"constraints" SHALL be used to model certain value constraints or limits over time.
2: M	timeSeriesWriteable	"true"	If set to false or omitted, this timeSeriesData SHALL NOT be writeable.
2: M	unit	"W"	The unit SHALL be applied to the value in timeSeriesData
2: M	TimeSeries. timeSeriesListData. timeSeriesData.		
2: M	timeSeriesId	<m2>	PRIMARY IDENTIFIER of timeSeries
2: M	timePeriod.		If set, neither timeSeriesSlot.duration nor timeSeriesSlot.timePeriod SHALL define times out of the bound of the overall timeSeries timePeriod. Only relative times SHALL be used (xs:duration). Relative times SHALL be interpreted relative to "now".
2: M	timePeriod. startTime	<xs:duration>	
2: M	timeSeriesSlot.		
2: M	timeSeriesSlot. timeSeriesSlotId		If timeSeriesSlotId is omitted it SHALL be interpreted as zero.
2: M	timeSeriesSlot. duration		SHALL only contain values greater than zero seconds. If timeSeriesSlot. timePeriod. startTime is not set in the first timeSeriesSlot (slot with the lowest timeSeriesSlotId), the timePeriod. startTime of the overall timeSeries SHALL be set and the duration of the first timeSeriesSlot starts from there. If timeSeriesSlot. timePeriod. startTime is not set in a slot that is not the first slot, the duration starts after the previous slot is finished. The sum of timePeriod. startTime and the durations of all slots SHALL equal timePeriod. endTime if set.

1142 Table 24: Content of Specialization "TimeSeries_AcPowerMaxWriteable" at Actor Energy Broker

1143

1144 3.2.3.3.1.3 Specialization "TimeSeries_AcMaxPowerPlan"

Scenario [{...}]: M/R/O [W][V]	Element	Value	[High Level Mapping] Element and value rules
4: M	TimeSeries. timeSeriesDescriptionListData. timeSeriesDescriptionData.		
4: M	timeSeriesId	<m3>	PRIMARY IDENTIFIER of timeSeries.
4: M	timeSeriesType	"plan"	"plan" SHALL be used to model how a certain value will change over time.
4: M	timeSeriesWriteable	"false"	If set to false or omitted, this timeSeriesData SHALL NOT be writeable.
4: M	measurementId	<k1>	FOREIGN IDENTIFIER, links data of other Features to the timeSeries.
4: M	unit	"W"	The unit SHALL be applied to the value in timeSeriesData
4: M	TimeSeries. timeSeriesListData. timeSeriesData.		
4: M	timeSeriesId	<m3>	PRIMARY IDENTIFIER of timeSeries
4: M	timePeriod		If set, neither timeSeriesSlot.duration nor timeSeriesSlot.timePeriod SHALL define times out of the bound of the overall timeSeries timePeriod. Only relative times SHALL be used (xs:duration). Relative times SHALL be interpreted relative to "now".
4: M	timePeriod. startTime	<xs:duration>	
4: M	timeSeriesSlot.		
4: M	timeSeriesSlot. timeSeriesSlotId		If timeSeriesSlotId is omitted it SHALL be interpreted as zero.
4: M	timeSeriesSlot. duration		SHALL only contain values greater than zero seconds. If timeSeriesSlot. timePeriod. startTime is not set in the first timeSeriesSlot (slot with the lowest timeSeriesSlotId), the timePeriod. startTime of the overall timeSeries SHALL be set and the duration of the first timeSeriesSlot starts from there. If timeSeriesSlot. timePeriod. startTime is not set in a slot that is not the first slot, the duration starts after the previous slot is finished.
4: M	timeSeriesSlot. value.		A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If minValue is set the content of value SHALL be greater or equal. If maxValue is set the content of value SHALL be smaller or equal.

			The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	timeSeriesSlot. value. number	≥ 0	SHALL be used.
4: M	timeSeriesSlot. value. scale		MAY be used. If absent, a default value of "0" applies.
4: M	timeSeriesSlot. minValue.		A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of minValue SHALL be smaller or equal. If value is not set but maxValue is set the content of minValue SHALL be smaller or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	timeSeriesSlot. minValue. number	≥ 0	SHALL be used.
4: M	timeSeriesSlot. minValue. scale		MAY be used. If absent, a default value of "0" applies.
4: M	timeSeriesSlot. maxValue.		[CEVC-008] A positive value SHALL relate to consumption and a negative value SHALL relate to production of the server. If value is set the content of maxValue SHALL be greater or equal. If value is not set but minValue is set the content of maxValue SHALL be greater or equal. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	timeSeriesSlot. maxValue. number	≥ 0	SHALL be used.
4: M	timeSeriesSlot. maxValue. scale		MAY be used. If absent, a default value of "0" applies.
4: M	TimeSeries. timeSeriesConstraintsListData. timeSeriesConstraintsData.		
4: M	timeSeriesId	<m3>	SHALL be set as PRIMARY IDENTIFIER.
4: M	slotValueMin.		The value, maxValue or minValue of a slot SHALL be equal to or higher than slotValueMin. The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	slotValueMin. number		SHALL be used.
4: M	slotValueMin. scale		MAY be used. If absent, a default value of "0" applies.
4: M	slotValueMax.		The value, maxValue or minValue of a slot SHALL be equal to or lower than slotValueMax.

			The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
4: M	slotValueMax. number		SHALL be used.
4: M	slotValueMax. scale		MAY be used. If absent, a default value of "0" applies.
4: M	slotValueStepSize.		<p>If slotValueStepSize is set and slotValueMin as well as slotValueMax is not set, the elements value, maxValue and minValue of a slot SHALL be a multiple of slotValueStepSize.</p> <p>If slotValueStepSize is set and slotValueMin is also set, the elements value, maxValue and minValue of a slot SHALL be slotValueMin plus a multiple of slotValueStepSize.</p> <p>If slotValueStepSize is set and slotValueMax is also set, the elements value, maxValue and minValue of a slot SHALL be slotValueMax minus a multiple of slotValueStepSize.</p> <p>The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
4: M	slotValueStepSize. number		SHALL be used.
4: M	slotValueStepSize. scale		MAY be used. If absent, a default value of "0" applies.

Table 25: Content of Specialization "TimeSeries_AcMaxPowerPlan" at Actor Energy Broker

3.2.3.3.2 Topic "IncentiveTable"

3.2.3.3.2.1 Specialization "IncentiveTable_SimpleIncentiveTableWriteable"

Scenario [{...}]: M/R/O [W][C]	Element	Value	[High Level Mapping] Element and value rules
3: M \W	IncentiveTable. incentiveTableData. incentiveTable.		
3: M \W	tariff.		
3: M \W	tariff. tariffId	<x1>	SHALL be set as PRIMARY IDENTIFIER.
3: M \W	incentiveSlot.		
3: M \W	incentiveSlot. timeInterval.		SHALL be set if an incentive has a time restriction or the incentiveTable contains more than one incentiveSlot entry. The timeInterval of different incentiveSlots within an

			incentiveTable SHALL NOT overlap in time.
3: M \W	incentiveSlot. timeInterval. startTime.		
3: M \W	incentiveSlot. timeInterval. startTime. relative		
3: M \W	incentiveSlot. timeInterval. endTime.		SHALL be set for the last slot. SHALL be set if a time gap follows after this slot. MAY be set for all other slots.
3: M \W	incentiveSlot. timeInterval. endTime. relative		
3: M \W	incentiveSlot. tier.		[CEVC-028]
3: M \W	incentiveSlot. tier. tier.		
3: M \W	incentiveSlot. tier. tier. tierId	<y1>	SUB IDENTIFIER of tariffId to identify a tier of a tariff. SHALL be set if the incentiveTable contains more than one tier. If the tierId is omitted, a default value of "1" SHALL be applied.
3: M \W	incentiveSlot. tier. boundary.		The boundary list defines the boundaries of a tier. The boundary range of different tiers within an incentiveSlot defined by lowerBoundaryValue and upperBoundaryValue SHALL NOT overlap for boundaries with the same boundaryId. The boundary range of a tier includes all values that are equal or greater than lowerBoundaryValue and lower than upperBoundaryValue.
3: M \W	incentiveSlot. tier. boundary. boundaryId	<z1>	SHALL be used as SUB IDENTIFIER of tierId. SHALL be set if the tier contains more than one boundary entry. If the boundaryId is omitted, a default value of "1" SHALL be applied.
3: M \W	incentiveSlot. tier. boundary. lowerBoundaryValue.		[CEVC-028] Within a boundary lowerBoundaryValue SHALL be smaller than upperBoundaryValue. Lower and upper boundaries of adjacent tiers: If lowerBoundaryValue is omitted and upperBoundaryValue is set, the lowerBoundaryValue SHALL

			<p>be interpreted as if it is equal (without a gap) to the upperBoundaryValue of the next lower tier within the incentiveSlot. Gaps between boundaries of adjacent "tier" instances can only be modelled by setting the upperBoundaryValue of the lower "tier" AND the lowerBoundaryValue of the higher "tier" instance explicitly.</p> <p>Minus infinity lowerBoundary rule: If lowerBoundaryValue is omitted and there exists no lower tier with a lowerBoundaryValue or upperBoundaryValue within the incentiveSlot, the omitted lowerBoundaryValue SHALL be interpreted as minus infinity.</p> <p>The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$</p>
3: M \W	incentiveSlot. tier. boundary. lowerBoundaryValue. number	≥ 0	SHALL be used.
3: M \W	incentiveSlot. tier. boundary. lowerBoundaryValue. scale		MAY be used. If absent, a default value of "0" applies.
3: M \W	incentiveSlot. tier. incentive.		
3: M \W	incentiveSlot. tier. incentive. incentiveld	<j1>	SHALL be used as SUB IDENTIFIER of tierId. SHALL be set if the tier contains more than one incentive entry. If the incentiveld is omitted, a default value of "1" SHALL be applied.
3: M \W	incentiveSlot. tier. incentive. value.		[CEVC-054] The sub-elements "number" and "scale" represent a value according to the following formula: $\text{number} * 10^{\text{scale}}$
3: M \W	incentiveSlot. tier. incentive. value. number		SHALL be used.
3: M \W	incentiveSlot. tier. incentive. value. scale		MAY be used. If absent, a default value of "0" applies.
3: M	IncentiveTable. incentiveTableDescriptionData. incentiveTableDescription.		
3: M	tariffDescription.		

3: M	tariffDescription. tariffId	<x1>	SHALL be set as PRIMARY IDENTIFIER of an incentiveTable, if more than one incentiveTable is defined within IncentiveTable.
3: M	tariffDescription. tariffWriteable	true	SHALL be set to "true" if the incentive table is writeable by a client. Otherwise it may be omitted.
3: M	tariffDescription. updateRequired	true	[CEVC-030], [CEVC-031] With updateRequired the server can request an update of writeable or changeable data related to the same PRIMARY IDENTIFIER from a client. The server SHALL ensure that only one responsible client is able to update the related data. To request an update the server SHALL set updateRequired to "true". Note: In this case, the server expects the responsible client to update the writeable or changeable data related to the same PRIMARY IDENTIFIER. However, also if updateRequired is set to "false" a server SHOULD in general allow updates of the data from the responsible client. The server SHALL set the updateRequired back to "false", as soon as "incentiveTableDescriptionData" was updated successfully (if writeable or changeable) OR the update of the other writeable or changeable data related to the same PRIMARY IDENTIFIER was successful. Note: The client does not need to stop an ongoing update process (e.g. if multiple Functions are written), when updateRequired is set back to "false". The server MAY choose to withdraw the update request at any time by setting updateRequired back to "false".
3: M	tariffDescription. scopeType	"simpleIncentiveTable"	
3: O	tariffDescription. slotIdSupport		If omitted, a default of "false" SHALL be applied. If set to "false" no incentiveSlot SHALL contain a timeSlotId. If set to "true"

				timeSlotId MAY be used within the incentiveSlot as described for the element timeSlotId within the Function incentiveTableData.
3: M \W	tier.			
3: M \W	tier. tierDescription.			
3: M \W	tier. tierDescription. tierId	<y1>		SHALL be set as SUB IDENTIFIER of tariffId, if the incentiveTable contains more than one tier. If the tierId is omitted, a default value of "1" SHALL be applied.
3: M \W	tier. tierDescription. tierType	"dynamicCost"		The tier has a cost incentive that MAY vary over time.
3: M \W	tier. boundaryDescription.			
3: M \W	tier. boundaryDescription. boundaryId	<z1>		SHALL be set as SUB IDENTIFIER of tierId, if the tier contains more than one boundary entry. If the boundaryId is omitted, a default value of "1" SHALL be applied.
3: M \W	tier. boundaryDescription. boundaryType	"powerBoundary"		The boundary SHALL be an electrical power. E.g. if the surplus power is consumed, more expensive power has to be used. Positive powerBoundary values describe which power can be consumed with certain incentives, while negative powerBoundary values describe which power can be produced with certain incentives.
3: M \W	tier. boundaryDescription. boundaryUnit	"W"		If set, the unit SHALL be applied to the lowerBoundaryValue and upperBoundaryValue.
3: M \W	tier. incentiveDescription.			[CEVC-023]
3: M \W	tier. incentiveDescription. incentiveId	<j1>		SHALL be set as SUB IDENTIFIER of tierId, if the tier contains more than one incentive entry. If the incentiveId is omitted, a default value of "1" SHALL be applied.
3: M \W	tier. incentiveDescription. incentiveType Note: At least one of the values SHALL be used.	3: O \W	"absoluteCost"	[CEVC-020] Each related value SHALL be an absolute cost with a specific currency. Positive values SHALL be interpreted as costs. Negative values SHALL be interpreted as profits. The currency SHALL be set and relate to Wh. E.g. EUR/Wh. For this incentive the Element tier. incentiveDescription. unit SHALL be ignored and only the

				Element tier. incentiveDescription. currency is relevant.
		3: O \W	"relativeCost"	[CEVC-019] The values SHALL have no currency and the unit SHALL be percentage (pct). The value SHALL be in the range from 0% to 100%. With 0% representing lowest costs and 100% highest costs. For this incentive the Elements tier. incentiveDescription. unit and tier. incentiveDescription. currency SHALL be ignored.
		3: O \W	"renewableEnergyPercentage"	[CEVC-021] The unit of the incentive SHALL relate to percentage (pct) in this case. The value SHALL be in the range from 0% to 100%. For this incentive the Elements tier. incentiveDescription. unit and tier. incentiveDescription. currency SHALL be ignored.
		3: O \W	"co2Emission"	[CEVC-022] The unit of the incentive SHALL relate to "kg/Wh". Only positive values SHALL be used. For this incentive the Elements tier. incentiveDescription. unit and tier. incentiveDescription. currency SHALL be ignored.
3: O \W	tier. incentiveDescription. currency			If set, the currency SHALL be applied to the value of the incentive
3: M	IncentiveTable. incentiveTableConstraintsData. incentiveTableConstraints.			
3: M	tariff.			
3: M	tariff. tariffId	<x1>		SHALL be set as PRIMARY IDENTIFIER of a incentiveTable, if more than one incentiveTable is defined within IncentiveTable.
3: M	tariffConstraints.			
3: M	tariffConstraints. maxTiersPerTariff	≥3		[CEVC-055] If set the incentiveTable SHALL NOT include more tiers than given in maxTiersPerTariff.
3: M	tariffConstraints. maxBoundariesPerTier	≥1		[CEVC-055] If set the tier within the incentiveTable SHALL NOT include more boundaries in a tier than given in maxBoundariesPerTier.
3: M	tariffConstraints. maxIncentivesPerTier	≥3		[CEVC-056]

			If set the tier within the incentiveTable SHALL NOT include more incentives than given in maxIncentivesPerTier.
3: M	incentiveSlotConstraints.		
3: M	incentiveSlotConstraints.slotCountMax		If set the incentiveTable SHALL NOT include more slots than given in slotCountMax.

Table 26: Content of Specialization "IncentiveTable_SimpleIncentiveTableWriteable" at Actor Energy Broker

3.3 Pre-Scenario communication

3.3.1 General information

The Pre-Scenario communication is needed if a client does not know the corresponding addresses on the server or if the required subscriptions or bindings are not active. In this case certain general communication mechanisms SHALL be used within SPINE:

- a) Detailed discovery: allows to discover resource addresses.
- b) Binding: allows to bind to resource address, which is frequently necessary to obtain write privileges.
- c) Subscription: allows to subscribe to resource addresses, which is necessary to receive unsolicited notifications if a resource changes during runtime.

It is possible to combine those steps for multiple Scenarios or also multiple Use Cases:

- E.g. if multiple Scenarios in multiple Use Cases use the same Feature, only one subscription needs to occur.
- E.g. a complete detailed discovery or a subscription to the NodeManagement Feature needs to occur only once for all Use Cases.

Depending on which Entity, Feature and Functions are used within a Scenario the payload of the corresponding messages may slightly differ, but the basic principles and messages used stay the same.

The subsequent messages SHALL be exchanged for those parts that have not already been performed since the current connection is established or if those parts are outdated for another reason (e.g. if the detailed discovery is needed, but the bindings and subscriptions are still active from a previous connection only the detailed discovery messages need to be exchanged). If all Pre-Scenario communication parts are up-to-date, this section MAY be skipped, and the implementation can proceed as described in the corresponding "Scenario communication" sections.

After the connection is re-established (e.g. due to a power loss or a firmware update) the Pre-Scenario communication SHALL be performed as well. There may be circumstances where messages from the Pre-Scenario communication may be exchanged again.

Often the necessary messages of different Scenarios can be combined, so that only one single message is needed instead of multiple messages for the different Scenarios. This also is the case for the Pre-Scenario communication. In most cases only one "read" operation on the detailed discovery

1181 is necessary, as well as only one subscription request or binding request is needed for each Feature.
1182 Often multiple Scenarios within a Use Case access the same Feature, so only one subscription or
1183 binding is necessary.

1184

1185 **3.3.2 Detailed discovery**

1186 For the functionality where a client already has current detailed discovery information (i.e.
1187 independent of this Use Case or any Scenario of it) the remainder of this section SHOULD be skipped.

1188 Otherwise, the following procedure SHALL be performed in the given order:

- 1189 1. If a client is not subscribed to the primary NodeManagement instance, the client SHALL
1190 acquire a subscription according to the figure provided within this sub-section.
- 1191 2. A client SHALL read the detailed discovery information according to the figure provided
1192 within this sub-section. It SHALL keep the received information as far as it concerns
1193 mandatory and supported optional Entity Types, Feature Types and Functions of this Use
1194 Case that are needed by the client. This means that a client may choose how to store the
1195 necessary information. E.g. a client Actor can store the information how to address the
1196 necessary Features of the implemented Scenarios but may discard the Entity information.
- 1197 3. If and as long as a client has a subscription to the detailed discovery information of an Actor
1198 and receives proper notifications, it SHALL consider (i.e. integrate into the kept detailed
1199 discovery information) the received information as far as it concerns mandatory and
1200 supported optional Entity Types, Feature Types and Functions of this Use Case.

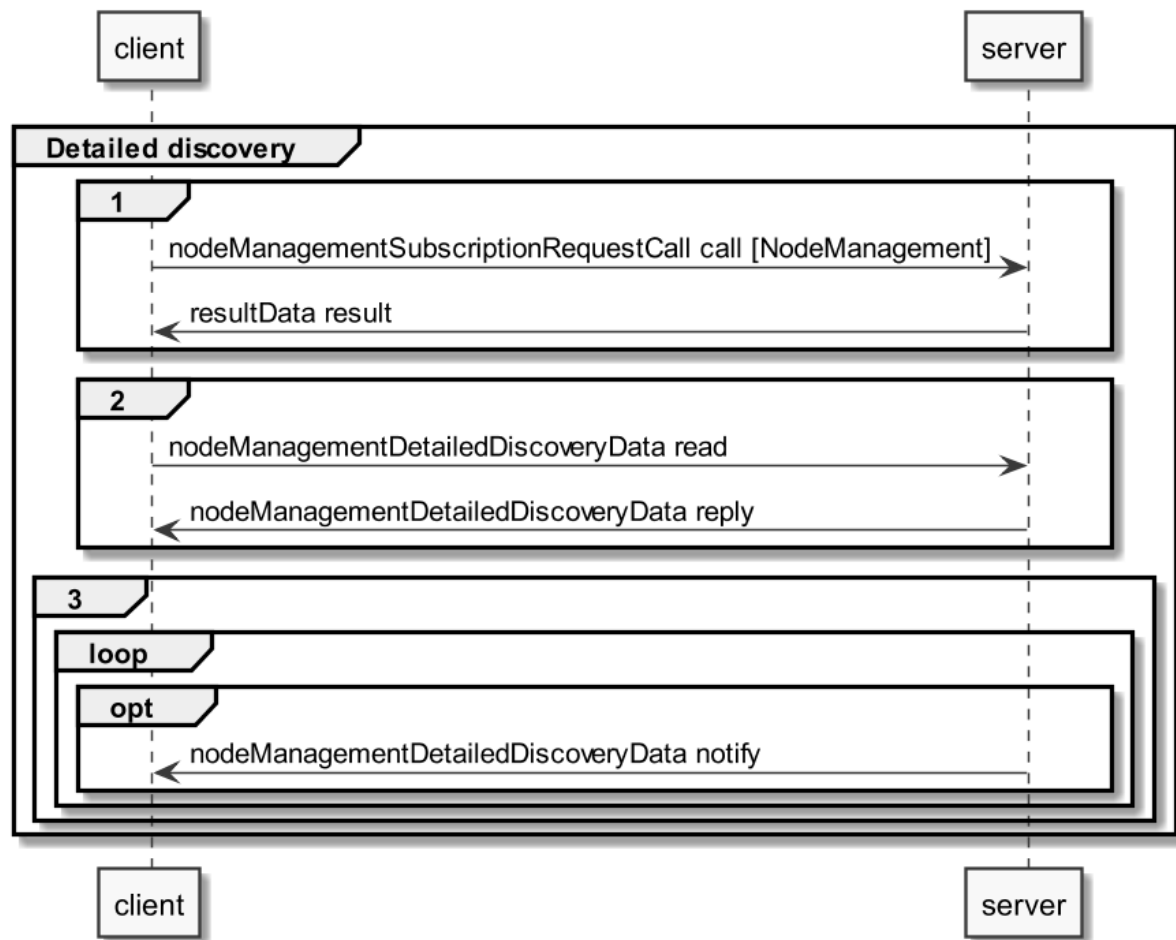


Figure 13: Pre-Scenario communication - Detailed discovery sequence diagram

If the "nodeManagementDetailedDiscoveryData read" fails, the client SHOULD retry to read the detailed discovery information until the "nodeManagementDetailedDiscoveryData reply" message was received successfully.

If all functionality is present at all times: The "nodeManagementDetailedDiscoveryData reply" message contains at least the mandatory Entities and Features given in the "Actor [...] overview" diagrams as well as the used Functions and their "possible operations" described in section 3.2 and its sub-sections.

If functionality is added or removed dynamically: The "nodeManagementDetailedDiscoveryData reply" message does not need to contain all mandatory Entities and Features given in the "Actor [...] overview" diagrams as well as all needed Functions and their "possible operations" described in section 3.2 and its sub-sections. However, as soon as the functionality is available it will be announced via a "nodeManagementDetailedDiscoveryData notify" message.

For the nodeManagementDetailedDiscoveryData read Function it is recommended to use a partial read with separated Selectors that may use one of the following Elements:

- entityType
- featureType

Note: Even with the usage of Selectors Features and Entities that are not relevant for this Use Case may be discovered. However, only Features and Entities that fulfil the hierarchical order as described within the Actors' sections shall be considered for this Use Case.

A "partial" notify SHALL be supported without using Selectors and Elements. Partial "delete" notify SHOULD also be supported with separated Selectors that may use one of the following Elements:

- entityAddress
- featureAddress

3.3.3 Binding

A server SHALL support binding for all Features that contain writeable or changeable data. Before a write on a Function of a Feature occurs, the client SHALL create a binding to the corresponding Feature. For this the nodeManagementBindingRequestCall Function is used as shown in the following sequence diagram:

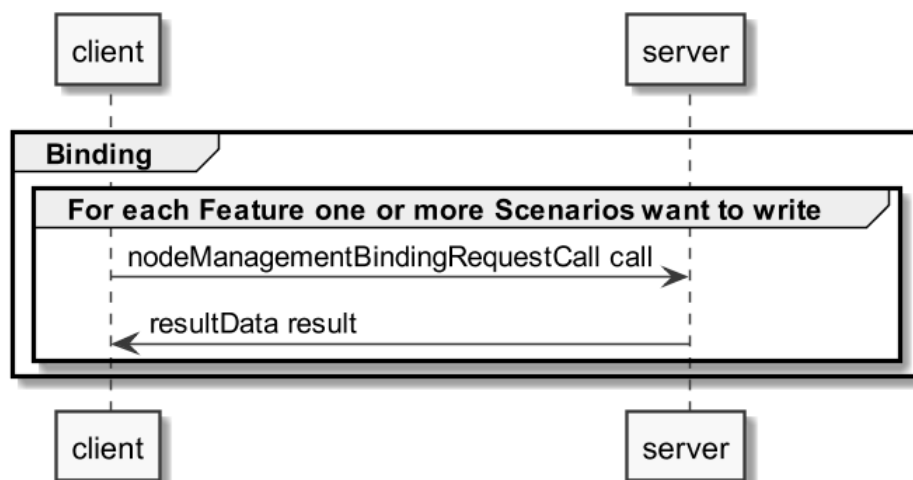


Figure 14: Pre-Scenario communication - Binding sequence diagram

If functionality is added or removed dynamically, binding may not be possible at all times on the required Functions. A client SHALL retry to create a binding again when receiving according updated detailed discovery information.

3.3.4 Subscription

A server SHALL support subscription for all Features that contain readable data that may change during runtime. The client SHALL create a subscription for all Features that the client wants to read. For this the nodeManagementSubscriptionRequestCall Function is used as shown in the following sequence diagram:

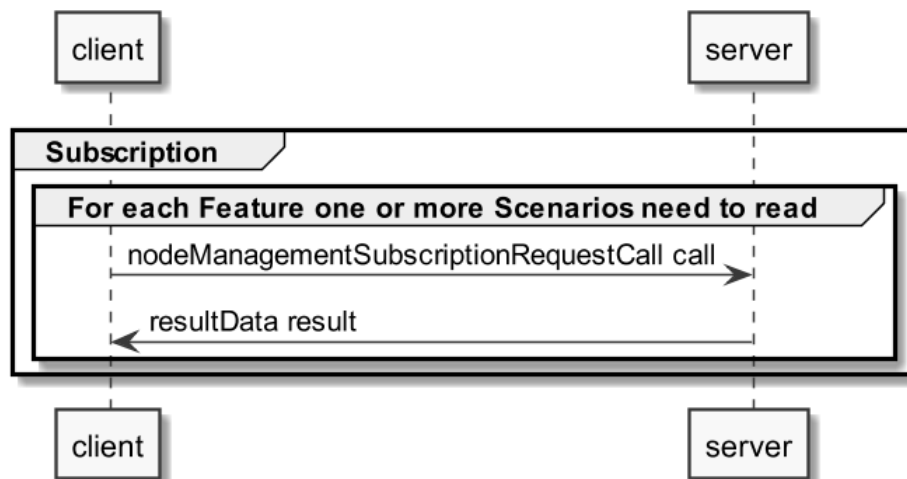


Figure 15: Pre-Scenario communication - Subscription sequence diagram

If the subscription request fails (e.g. because it is not supported by the server or the maximum number of possible subscriptions is reached), the client **SHOULD** read the data periodically (so-called "polling").

If functionality is added or removed dynamically, subscription may not be possible at all times on the required Functions. A client **SHALL** retry its subscription procedure again when receiving according updated detailed discovery information.

3.3.5 Dynamic behaviour

In case Entities or Features are removed, a nodeManagementDetailedDiscoveryData "notify" is transmitted that informs about the deleted Entities and Features. All existing binding or subscription entries on the deleted Features **SHALL** be deleted by each device.

In case Entities or Features are added the Pre-Scenario communication starts with transmitting a nodeManagementDetailedDiscoveryData "notify" that contains the added Entities and Features.

3.4 Scenarios

3.4.1 Scenario 1 - EV sends charging energy demand to Energy Broker and Energy Guard

3.4.1.1 Pre-Scenario communication

1. **Detailed Discovery:** Actors that act as client within this Scenario, need to know the addresses of the server Features used in the Initial Scenario communication. If an address of a particular server Feature is not known, the detailed discovery has to be used, as described in section 3.3.2.
2. **Binding:** Binding **SHOULD NOT** be used for this Scenario.
3. **Subscription:** Actors **SHALL** create a subscription for each server Feature that is relevant for the corresponding Actor within this Scenario, as described in section 3.3.4.

The Initial Scenario communication **SHALL** start at the latest when the required resources on an Actor are known and the necessary binding and subscription procedures have been finished. However, as

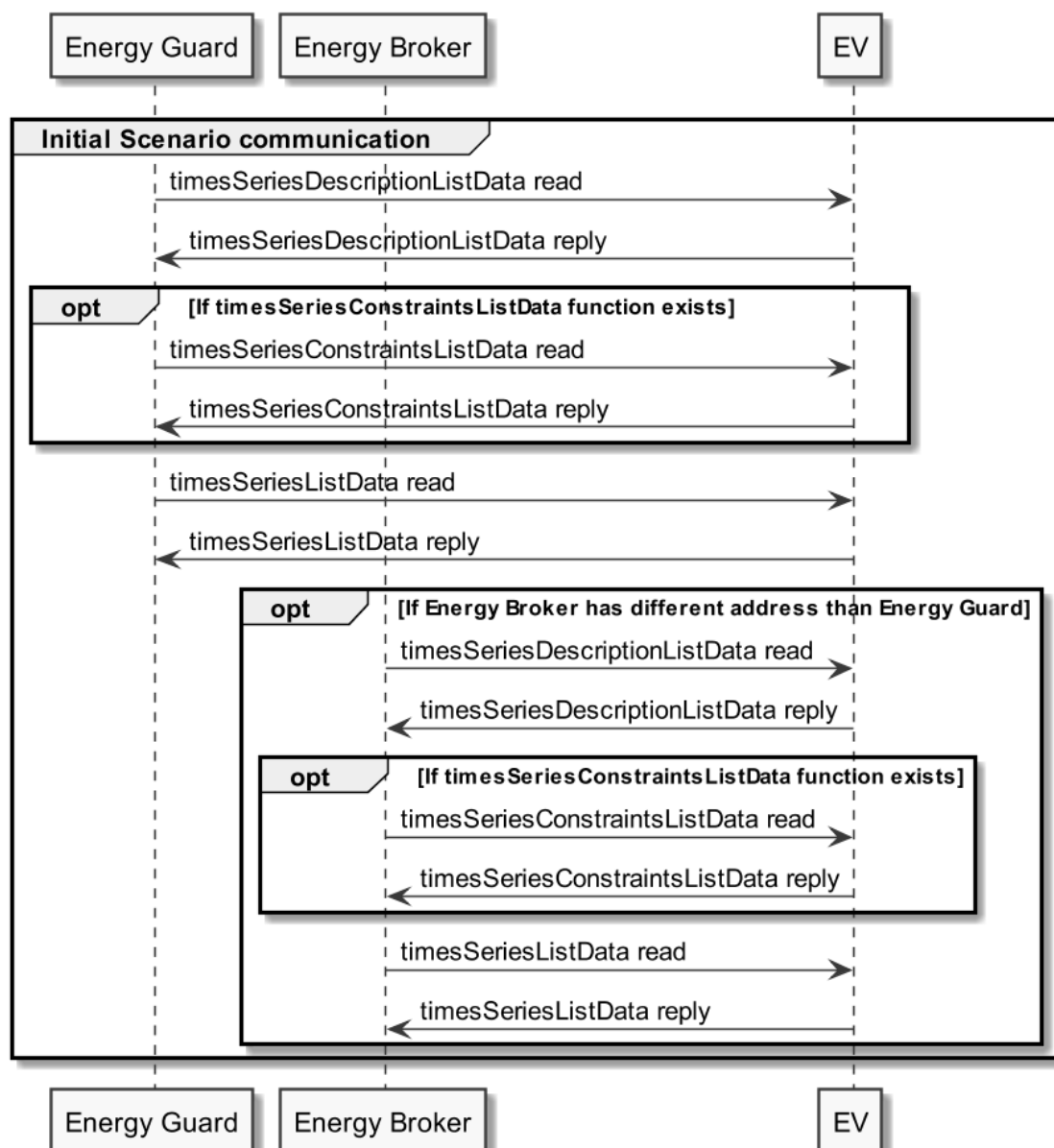
1271 soon as an address of a required resource is known, the Initial Scenario communication for this
 1272 resource MAY start already, even if addresses of other required resources are not known yet.

1273 If required resources are removed and added again, they are re-discovered, and the Initial Scenario
 1274 communication is triggered again for those resources.

1275

1276 **3.4.1.2 Initial Scenario communication**

1277 Each time a (re-)connection is established, even if the Pre-Scenario communication phase is skipped,
 1278 the messages as shown in the following sequence diagram SHALL be exchanged [CEVC-006], as the
 1279 corresponding resources may have changed in the meantime:



1280

1281 *Figure 16: Scenario 1 - Initial Scenario communication sequence diagram*

1282 The `timesSeriesDescriptionListData` read SHOULD be a "partial" read with the following Selectors:

1283 - `timeSeriesType = "singleDemand"`

1284 The timeSeriesConstraintsListData read and timeSeriesListData read SHOULD be "partial" read
 1285 operations with the following Selectors:

1286 - timeSeriesId (value taken from the timeSeriesDescriptionListData reply)

1287 Note: If partial read is not supported a full read SHALL be performed.

1288 Note: If the Energy Guard and Energy Broker are implemented on the same device and client Feature
 1289 each read operation only needs to be performed once.

1290 Note: If the Energy Guard and Energy Broker are implemented on different devices or client Features
 1291 the order of the messages may switch depending on who discovers the EV first.

1292

1293 The following table shows where the necessary content of the messages from the sequence diagram
 1294 is described:

Message name from sequence diagram	Content description in table	Scenario number in table
timeSeriesDescriptionListData reply	Table 6	1
timeSeriesConstraintsListData reply	Table 7	1
timeSeriesListData reply	Table 8	1

1295 *Table 27: Initial Scenario communication content references for Scenario 1*

1296 Note: Within the Initial Scenario communication the content required by this Scenario MAY not be
 1297 provided completely but later on during Runtime Scenario communication.

1298

1299 **3.4.1.3 Runtime Scenario communication**

1300 Based on the Initial Scenario communication the Runtime Scenario communication provides updates
 1301 during runtime [CEVC-006].

1302 If one of the referenced server Functions' data change, the server SHALL submit the change as shown
 1303 in the following figure:

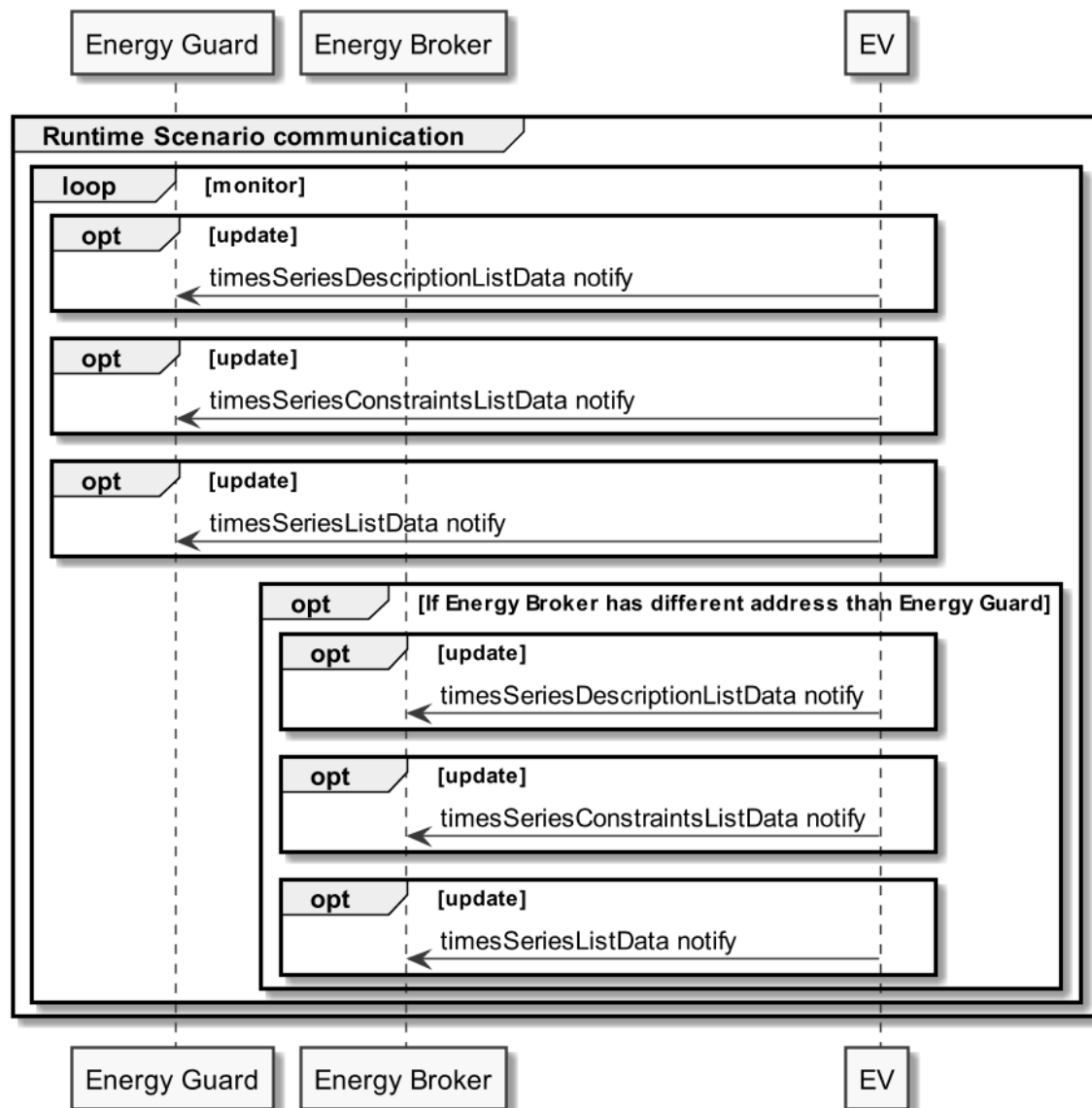


Figure 17: Scenario 1 - Runtime Scenario communication sequence diagram

Partial notifications without Selectors or Elements SHALL be supported for all Functions used in this Scenario.

Partial delete notification SHOULD also be supported with the Selector:

- timeSeriesId
- timeSeriesSlotId (only for timeSeriesListData Function)

Note: To interpret partial notification messages correctly the information obtained during the Initial Scenario communication phase is necessary.

Note: A read operation ("polling") on all Functions is possible at any time, e.g. if a notification could not be evaluated.

The following table shows where the necessary content of the messages of the sequence diagram is described:

Message name from sequence diagram	Content description in table	Scenario number in table
timeSeriesDescriptionListData notify	Table 6	1
timeSeriesConstraintsListData notify	Table 7	1
timeSeriesListData notify	Table 8	1

Table 28: Runtime Scenario communication content references for Scenario 1

3.4.1.1 Additional information

None.

3.4.2 Scenario 2 - Energy Guard sends maximum power limitation curve to EV

3.4.2.1 Pre-Scenario Communication

1. **Detailed Discovery:** Actors that act as client within this Scenario, need to know the addresses of the server Features used in the Initial Scenario communication. If an address of a particular server Feature is not known, the detailed discovery has to be used, as described in section 3.3.2.
2. **Binding:** Actors that write parts of a Feature within this Scenario, need to create a binding, as described in section 3.3.3. Only one binding partner is allowed to write the data specified in this Scenario, according to the "updateRequired" Element rule.
3. **Subscription:** Actors SHALL create a subscription for each server Feature that is relevant for the corresponding Actor within this Scenario, as described in section 3.3.4.

The Initial Scenario communication SHALL start at the latest when the required resources on an Actor are known and the necessary binding and subscription procedures have been finished. However, as soon as an address of a required resource is known, the Initial Scenario communication for this resource MAY start already, even if addresses of other required resources are not known yet.

If required resources are removed and added again, they are re-discovered, and the Initial Scenario communication is triggered again for those resources.

3.4.2.2 Initial Scenario communication

Each time a (re-)connection is established, even if the Pre-Scenario communication phase is skipped, the messages as shown in the following sequence diagram SHALL be exchanged, as the corresponding resources may have changed in the meantime:

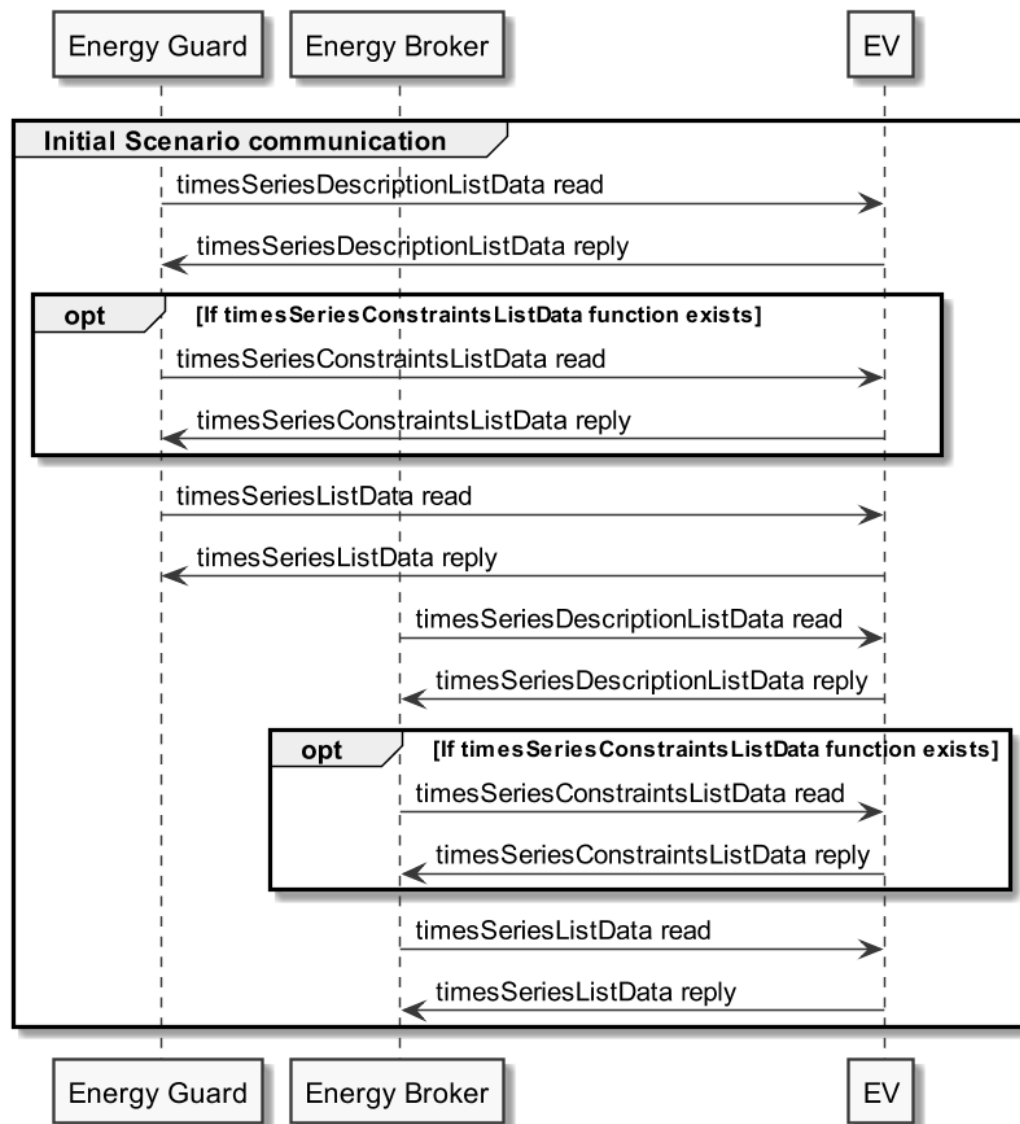


Figure 18: Scenario 2 - Initial Scenario communication sequence diagram

The `timesSeriesDescriptionListData` read SHOULD be a "partial" read with the following Selectors:

- `timeSeriesType = "constraints"`

The `timesSeriesConstraintsListData` and `timesSeriesListData` read SHOULD be "partial" read operations with the following Selectors:

- `timeSeriesId` (value taken from the `timesSeriesDescriptionListData` reply)

Note: If partial read is not supported a full read SHALL occur.

Note: If the Energy Guard and Energy Broker are implemented on the same device and client Feature each read only need to occur once.

Note: If the Energy Guard and Energy Broker are implemented on different devices or client Features the order of the messages may switch depending on who discovers the EV first.

1358 The following table shows where the necessary content of the messages from the sequence diagram
1359 is described:

Message name from sequence diagram	Content description in table	Scenario number in table
timeSeriesDescriptionListData reply	Table 6	2
timeSeriesConstraintsListData reply	Table 7	2
timeSeriesListData reply	Table 8	2

1360 *Table 29: Initial Scenario communication content references for Scenario 2*

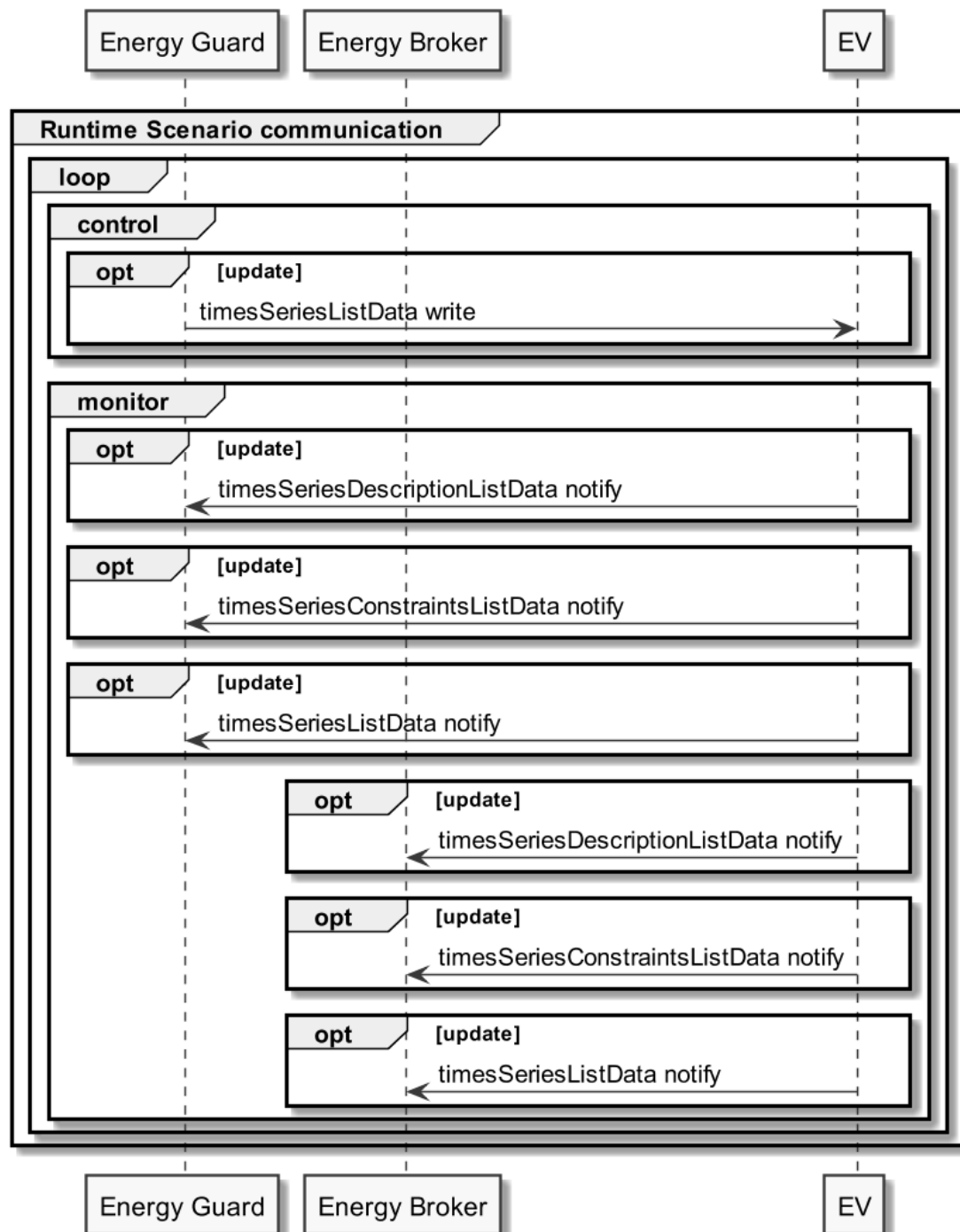
1361 Note: Within the Initial Scenario communication the content required by this Scenario MAY not be
1362 provided completely but later on during Runtime Scenario communication.

1363

1364 **3.4.2.3 Runtime Scenario communication**

1365 Based on the Initial Scenario communication the Runtime Scenario communication provides updates
1366 during runtime.

1367 If one of the referenced server Functions' data change, the server SHALL submit the changes as
1368 shown in the following figure:



1369

1370 *Figure 19: Scenario 2 - Runtime Scenario communication sequence diagram*

1371 Energy Guard and EV SHALL support partial notifications without Selectors or Elements for all
 1372 Functions used in this Scenario.

1373 Energy Guard and EV SHOULD also support partial delete notification with the Selectors:

- 1374 - timeSeriesId
- 1375 - timeSeriesSlotId (only for timeSeriesListData Function)

1376 Note: To interpret partial notification messages correctly the information obtained during the Initial
 1377 Scenario communication phase is necessary.

1378 Note: A read operation ("polling") on all Functions is possible at any time, e.g. if a notification could
1379 not be evaluated.

1380 The Energy Guard and EV SHALL support partial write for the timeSeriesListData Function without
1381 Selectors or Elements.

1382 Partial delete write SHALL also be supported for the timeSeriesListData Function with the Selectors:

- 1383 - timeSeriesId
- 1384 - timeSeriesSlotId

1385

1386 The following table shows where the necessary content of the messages of the sequence diagram is
1387 described:

Message name from sequence diagram	Content description in table	Scenario number in table
timeSeriesListData write [CEVC-013], [CEVC-016]	Table 8	2
timeSeriesDescriptionListData notify	Table 6	2
timeSeriesConstraintsListData notify	Table 7	2
timeSeriesListData notify	Table 8	2

1388 *Table 30: Runtime Scenario communication content references for Scenario 2*

1389

1390 **3.4.2.4 Additional information**

1391 None.

1392

1393 **3.4.3 Scenario 3 - Energy Broker sends incentive table to EV**

1394 **3.4.3.1 Pre-Scenario Communication**

- 1395 1. **Detailed Discovery:** Actors that act as client within this Scenario, need to know the addresses
1396 of the server Features used in the Initial Scenario communication. If an address of a
1397 particular server Feature is not known, the detailed discovery has to be used, as described in
1398 section 3.3.2.
- 1399 2. **Binding:** Actors that write parts of a Feature within this Scenario, need to create a binding, as
1400 described in section 3.3.3. Only one binding partner is allowed to write the data specified in
1401 this Scenario, according to the "updateRequired" Element rule.
- 1402 3. **Subscription:** Actors SHALL create a subscription for each server Feature that is relevant for
1403 the corresponding Actor within this Scenario, as described in section 3.3.4.

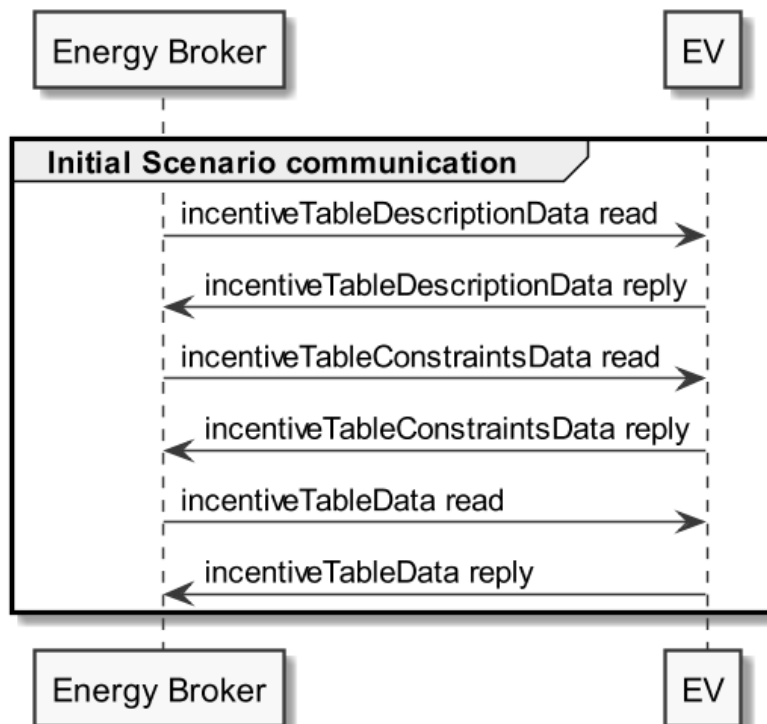
1404 The Initial Scenario communication SHALL start at the latest when the required resources on an Actor
1405 are known and the necessary binding and subscription procedures have been finished. However, as
1406 soon as an address of a required resource is known, the Initial Scenario communication for this
1407 resource MAY start already, even if addresses of other required resources are not known yet.

1408 If required resources are removed and added again, they are re-discovered, and the Initial Scenario
 1409 communication is triggered again for those resources.

1410

1411 **3.4.3.2 Initial Scenario communication**

1412 Each time a (re-)connection is established, even if the Pre-Scenario communication phase is skipped,
 1413 the messages as shown in the following sequence diagram SHALL be exchanged, as the
 1414 corresponding resources may have changed in the meantime:



1415

1416 *Figure 20: Scenario 3 - Initial Scenario communication sequence diagram*

1417 The incentiveTableDescriptionData read SHOULD be a "partial" read with the following Selectors:

- 1418 - scopeType = "simpleIncentiveTable"

1419 The incentiveTableConstraintsData read and incentiveTableData read SHOULD be "partial" read
 1420 operations with the following Selectors:

- 1421 - tariffId (value taken from the incentiveTableDescriptionData reply)

1422 Note: If partial read is not supported a full read SHALL occur.

1423 The following table shows where the necessary content of the messages from the sequence diagram
 1424 is described:

Message name from sequence diagram	Content description in table	Scenario number in table
incentiveTableDescriptionData reply	Table 9	3
incentiveTableConstraintsData reply	Table 10	3
incentiveTableData reply	Table 11	3

1425 *Table 31: Initial Scenario communication content references for Scenario 3*

Note: Within the Initial Scenario communication the content required by this Scenario MAY not be provided completely but later on during Runtime Scenario communication.

3.4.3.3 Runtime Scenario communication

Based on the Initial Scenario communication the Runtime Scenario communication provides updates during runtime.

If one of the referenced server Functions' data change, the server SHALL submit the changes as shown in the following figure:

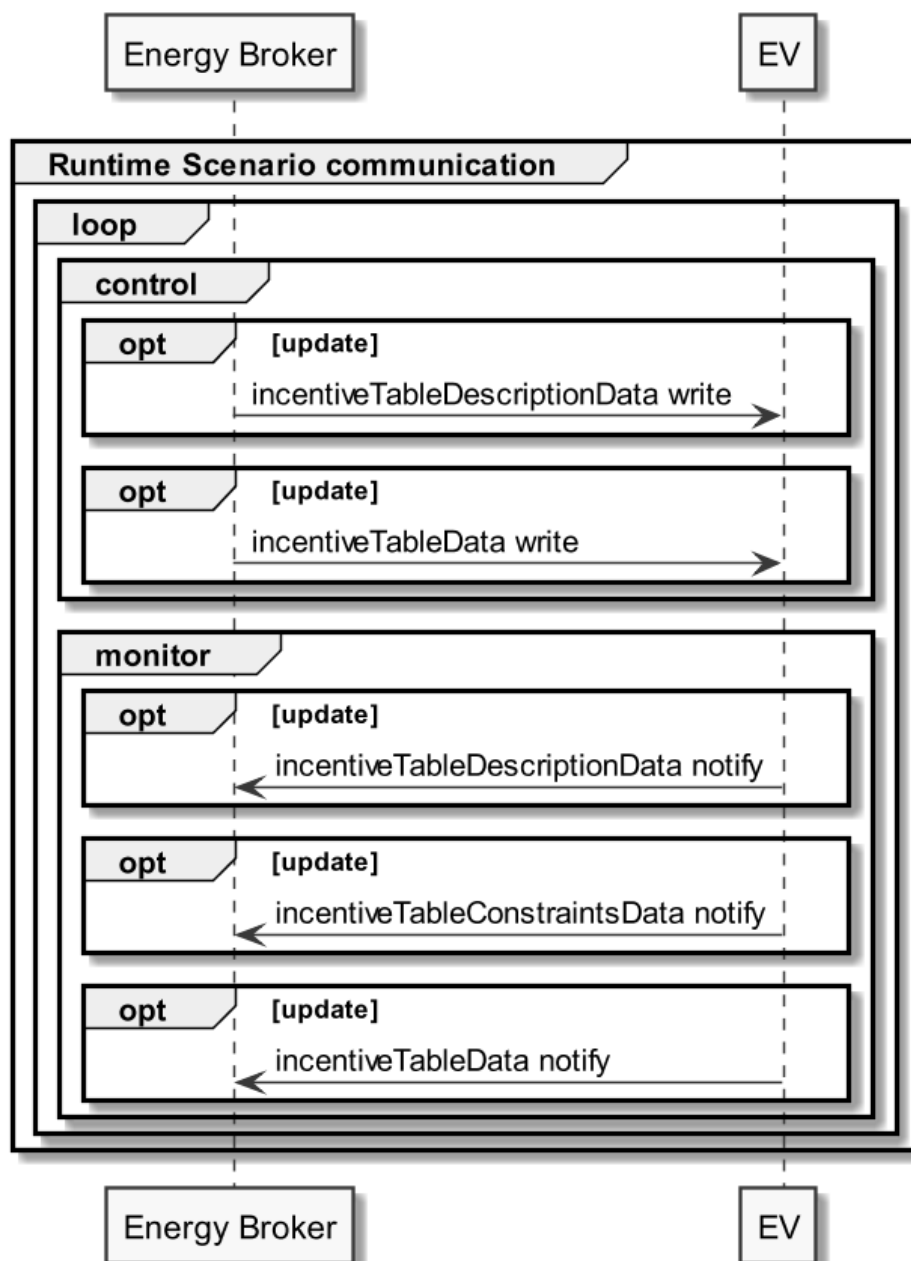


Figure 21: Scenario 3 - Runtime Scenario communication sequence diagram

Partial notifications without Selectors or Elements SHALL be supported for all Functions used in this Scenario.

1438 Partial delete notification SHOULD also be supported with the Selector:

1439 - tariffId

1440 Note: To interpret partial notification messages correctly the information obtained during the Initial
1441 Scenario communication phase is necessary.

1442 Note: A read operation ("polling") on all Functions is possible at any time, e.g. if a notification could
1443 not be evaluated.

1444 The Energy Broker and EV SHALL support partial write for the incentiveTableDescriptionData and
1445 incentiveTableData Function without Selectors or Elements.

1446 Partial delete write SHALL also be supported for the incentiveTableData Function with the Selectors:

1447 - tariffId

1448

1449 The following table shows where the necessary content of the messages of the sequence diagram is
1450 described:

Message name from sequence diagram	Content description in table	Scenario number in table
incentiveTableDescriptionData write	Table 10	3
incentiveTableData write [CEVC-027], [CEVC-032]	Table 11	3
incentiveTableDescriptionData notify	Table 9	3
incentiveTableConstraintsData notify	Table 10	3
incentiveTableData notify	Table 11	3

1451 *Table 32: Runtime Scenario communication content references for Scenario 3*

1452

1453 **3.4.3.4 Additional information**

1454 None.

1455

1456 **3.4.4 Scenario 4 - EV sends charging plan curve to Energy Broker and Energy Guard**

1457 **3.4.4.1 Pre-Scenario Communication**

- 1458 1. **Detailed Discovery:** Actors that act as client within this Scenario, need to know the addresses
1459 of the server Features used in the Initial Scenario communication. If an address of a
1460 particular server Feature is not known, the detailed discovery has to be used, as described in
1461 section 3.3.2.
- 1462 2. **Binding:** Binding SHOULD NOT be used for this Scenario.
- 1463 3. **Subscription:** Actors SHALL create a subscription for each server Feature that is relevant for
1464 the corresponding Actor within this Scenario, as described in section 3.3.4.

1465 The Initial Scenario communication SHALL start at the latest when the required resources on an Actor
1466 are known and the necessary binding and subscription procedures have been finished. However, as

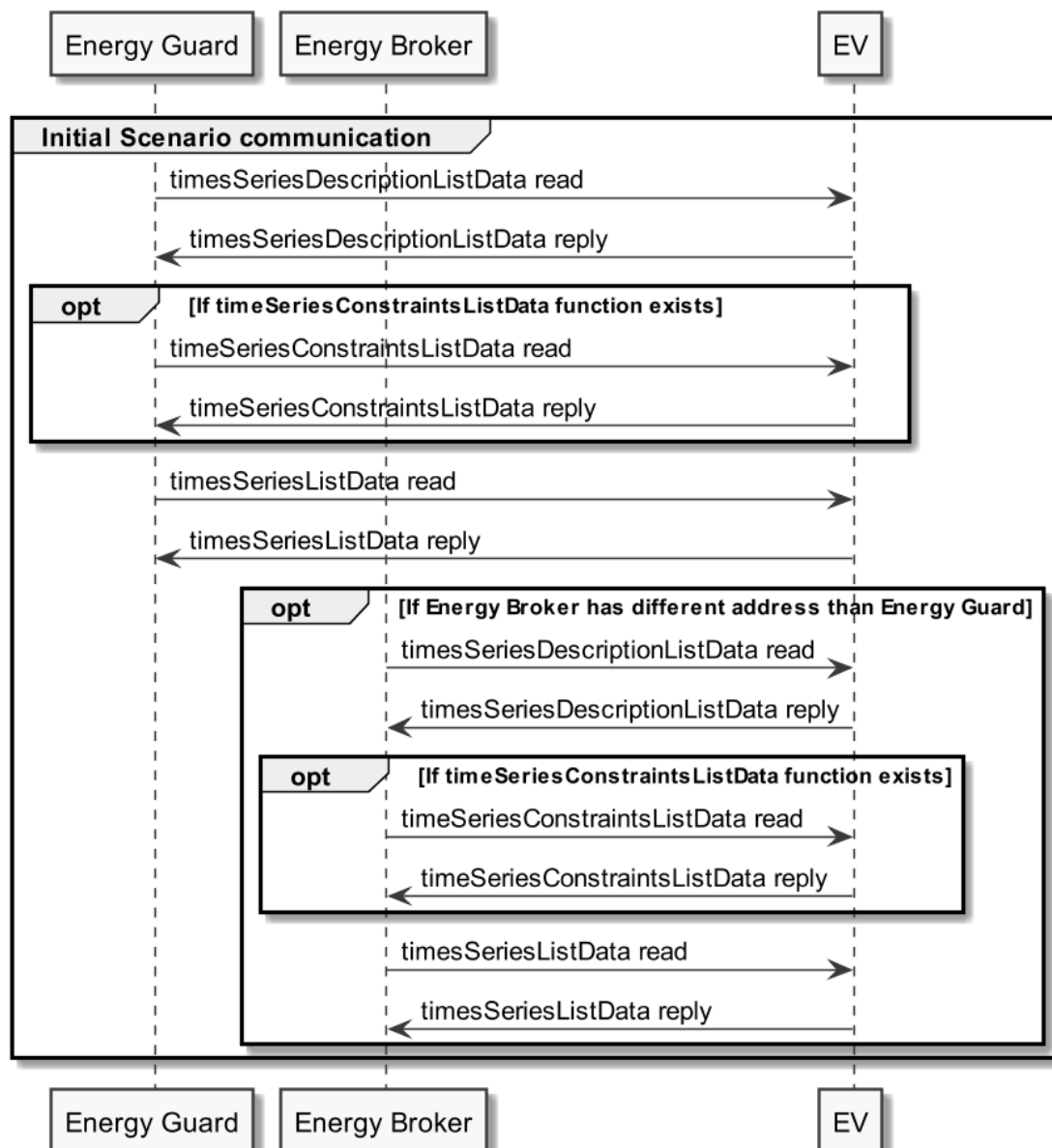
1467 soon as an address of a required resource is known, the Initial Scenario communication for this
 1468 resource MAY start already, even if addresses of other required resources are not known yet.

1469 If required resources are removed and added again, they are re-discovered, and the Initial Scenario
 1470 communication is triggered again for those resources.

1471

1472 **3.4.4.2 Initial Scenario communication**

1473 Each time a (re-)connection is established, even if the Pre-Scenario communication phase is skipped,
 1474 the messages as shown in the following sequence diagram SHALL be exchanged, as the
 1475 corresponding resources may have changed in the meantime:



1476

1477 *Figure 22: Scenario 4 - Initial Scenario communication sequence diagram*

1478 The `timeseriesDescriptionListData` read SHOULD be a "partial" read with the following Selectors:

- 1479 - `timeSeriesType = "plan"`

1480 The timeSeriesConstraintsListData read and timeSeriesListData read SHOULD be "partial" read
 1481 operations with the following Selectors:

1482 - timeSeriesId

1483 Note: If partial read is not supported a full read SHALL occur.

1484 Note: If the Energy Guard and Energy Broker are implemented on the same device and client Feature
 1485 each read only need to occur once.

1486 Note: If the Energy Guard and Energy Broker are implemented on different devices or client Features
 1487 the order of the messages may switch depending on who discovers the EV first.

1488 The following table shows where the necessary content of the messages from the sequence diagram
 1489 is described:

Message name from sequence diagram	Content description in table	Scenario number in table
timeSeriesDescriptionListData reply	Table 6	4
timeSeriesConstraintsListData reply	Table 7	4
timeSeriesListData reply	Table 8	4

1490 *Table 33: Initial Scenario communication content references for Scenario 4*

1491 Note: Within the Initial Scenario communication the content required by this Scenario MAY not be
 1492 provided completely but later on during Runtime Scenario communication.

1493

1494 **3.4.4.3 Runtime Scenario communication**

1495 Based on the Initial Scenario communication the Runtime Scenario communication provides updates
 1496 during runtime.

1497 If one of the referenced server Functions' data change, the server SHALL submit the changes as
 1498 shown in the following figure:

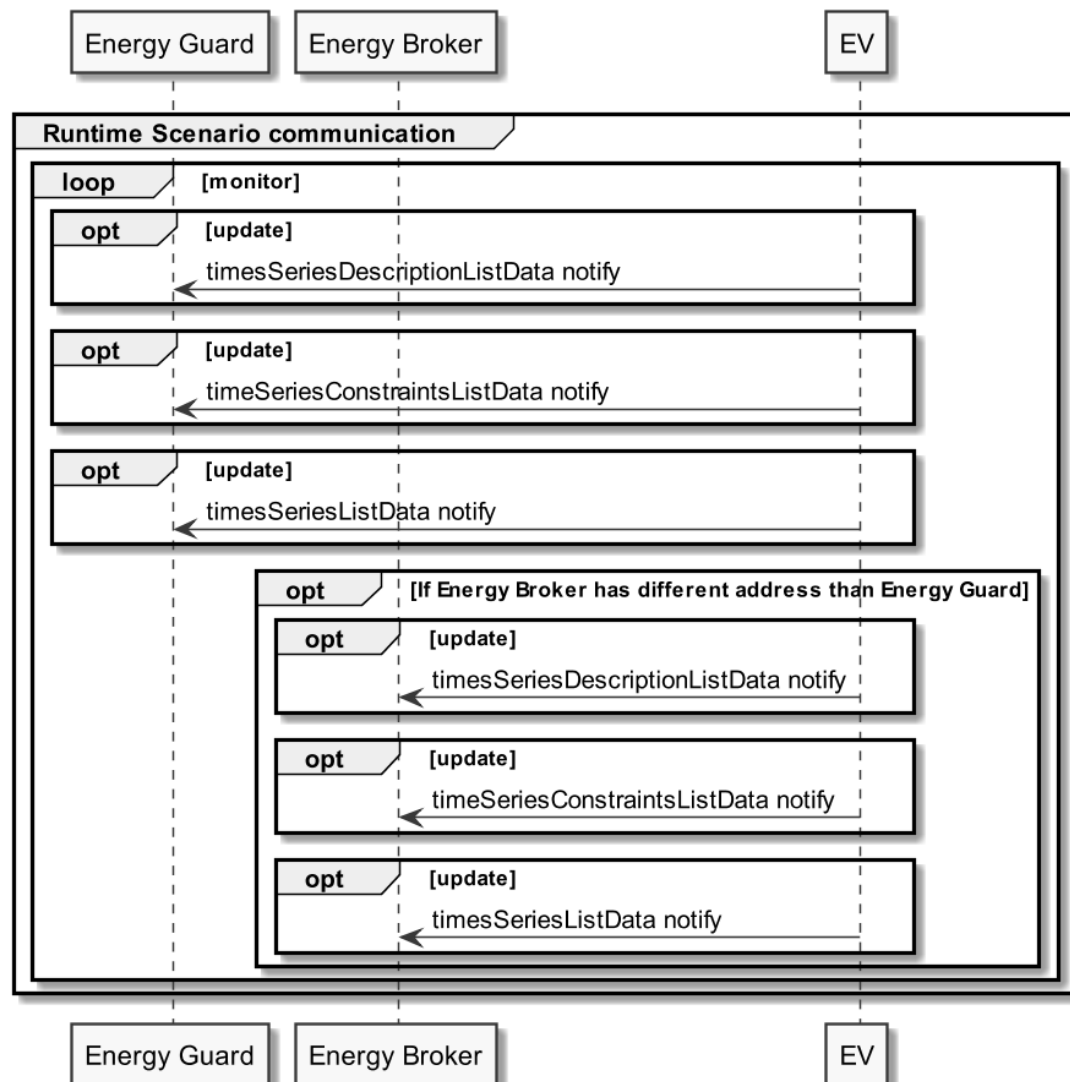


Figure 23: Scenario 4 - Runtime Scenario communication sequence diagram

Energy Guard, Energy Broker and EV SHALL support partial notifications without Selectors or Elements for all Functions used in this Scenario.

Partial delete notification SHOULD also be supported with the selector:

- timeSeriesId
- timeSeriesSlotId (only for timeSeriesListData Function)

Note: To interpret partial notification messages correctly the information obtained during the Initial Scenario communication phase is necessary.

Note: A read operation ("polling") on all Functions is possible at any time, e.g. if a notification could not be evaluated.

The following table shows where the necessary content of the messages of the sequence diagram is described:

Message name from sequence diagram	Content description in table	Scenario number in table
timeSeriesDescriptionListData notify	Table 6	4
timeSeriesConstraintsListData notify	Table 7	4
timeSeriesListData notify	Table 8	4

Table 34: Runtime Scenario communication content references for Scenario 4

3.4.4.4 Additional information

None.

3.4.5 Scenario 5 - Energy Guard heartbeat

3.4.5.1 Pre-Scenario Communication

1. **Detailed Discovery:** Actors that act as client within this Scenario, need to know the addresses of the server Features used in the Initial Scenario communication. If an address of a particular server Feature is not known, the detailed discovery has to be used, as described in section 3.3.2.
2. **Binding:** Binding SHOULD NOT be used for this Scenario.
3. **Subscription:** Actors SHALL create a subscription for each server Feature that is relevant for the corresponding Actor within this Scenario, as described in section 3.3.4.

The Initial Scenario communication SHALL start at the latest when the required resources on an Actor are known and the necessary binding and subscription procedures have been finished. However, as soon as an address of a required resource is known, the Initial Scenario communication for this resource MAY start already, even if addresses of other required resources are not known yet.

If required resources are removed and added again, they are re-discovered, and the Initial Scenario communication is triggered again for those resources.

3.4.5.2 Initial Scenario communication

Each time a (re-)connection is established, even if the Pre-Scenario communication phase is skipped, the messages as shown in the following sequence diagram SHALL be exchanged, as the corresponding resources may have changed in the meantime:

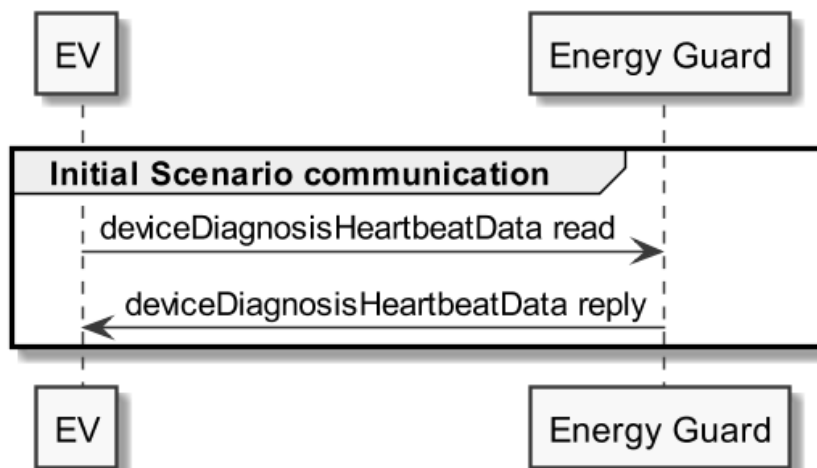


Figure 24: Scenario 5 - Initial Scenario communication sequence diagram

The following table shows where the necessary content of the messages from the sequence diagram is described:

Message name from sequence diagram	Content description in table	Scenario number in table
deviceDiagnosisHeartbeatData reply [CEVC-047]	Table 15	5

Table 35: Initial Scenario communication content references for Scenario 5

Note: Within the Initial Scenario communication the content required by this Scenario MAY not be provided completely but later on during Runtime Scenario communication.

3.4.5.3 Runtime Scenario communication

Based on the Initial Scenario communication the Runtime Scenario communication provides updates during runtime.

If one of the referenced server Functions' data change, the server SHALL submit the change as shown in the following figure:

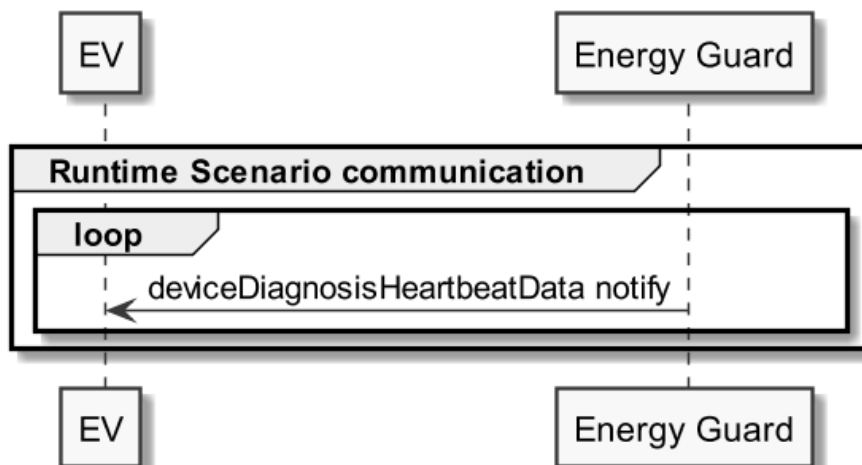


Figure 25: Scenario 5 - Runtime Scenario communication sequence diagram

1554 Note: To interpret partial notification messages correctly the information obtained during the Initial
1555 Scenario communication phase is necessary.

1556 Note: A read operation ("polling") on all Functions is possible at any time, e.g. if a notification could
1557 not be evaluated.

1558

1559 The following table shows where the necessary content of the messages of the sequence diagram is
1560 described:

Message name from sequence diagram	Content description in table	Scenario number in table
deviceDiagnosisHeartbeatData notify [CEVC-045]	Table 15	5

1561 *Table 36: Runtime Scenario communication content references for Scenario 5*

1562

1563 **3.4.5.4 Additional information**

1564 None.

1565

1566 **3.4.6 Scenario 6 - Energy Broker heartbeat**

1567 **3.4.6.1 Pre-Scenario Communication**

- 1568 1. **Detailed Discovery:** Actors that act as client within this Scenario, need to know the addresses
1569 of the server Features used in the Initial Scenario communication. If an address of a
1570 particular server Feature is not known, the detailed discovery has to be used, as described in
1571 section 3.3.2.
- 1572 2. **Binding:** Binding SHOULD NOT be used for this Scenario.
- 1573 3. **Subscription:** Actors SHALL create a subscription for each server Feature that is relevant for
1574 the corresponding Actor within this Scenario, as described in section 3.3.4.

1575 The Initial Scenario communication SHALL start at the latest when the required resources on an Actor
1576 are known and the necessary binding and subscription procedures have been finished. However, as
1577 soon as an address of a required resource is known, the Initial Scenario communication for this
1578 resource MAY start already, even if addresses of other required resources are not known yet.

1579 If required resources are removed and added again, they are re-discovered, and the Initial Scenario
1580 communication is triggered again for those resources.

1581

1582 **3.4.6.2 Initial Scenario communication**

1583 Each time a (re-)connection is established, even if the Pre-Scenario communication phase is skipped,
1584 the messages as shown in the following sequence diagram SHALL be exchanged, as the
1585 corresponding resources may have changed in the meantime:

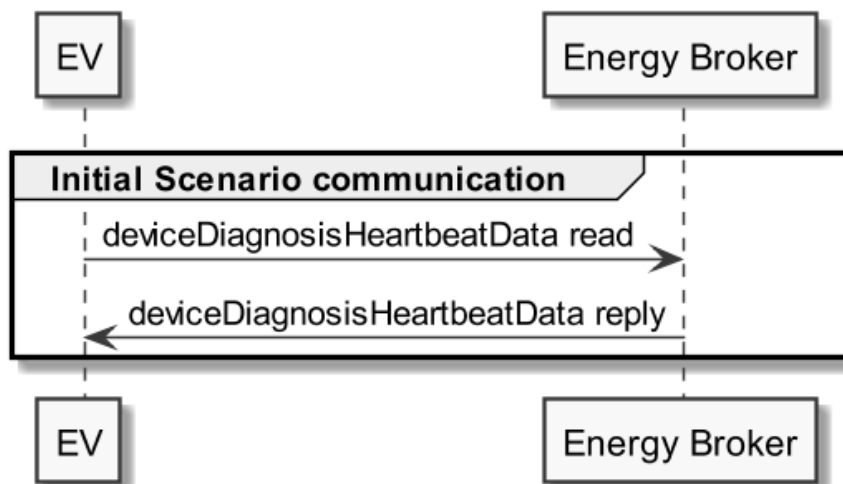


Figure 26: Scenario 6 - Initial Scenario communication sequence diagram

The following table shows where the necessary content of the messages from the sequence diagram is described:

Message name from sequence diagram	Content description in table	Scenario number in table
deviceDiagnosisHeartbeatData reply [CEVC-050]	Table 21	6

Table 37: Initial Scenario communication content references for Scenario 6

Note: Within the Initial Scenario communication the content required by this Scenario MAY not be provided completely but later on during Runtime Scenario communication.

3.4.6.3 Runtime Scenario communication

Based on the Initial Scenario communication the Runtime Scenario communication provides updates during runtime.

If one of the referenced server Functions' data change, the server SHALL submit the changes as shown in the following figure:

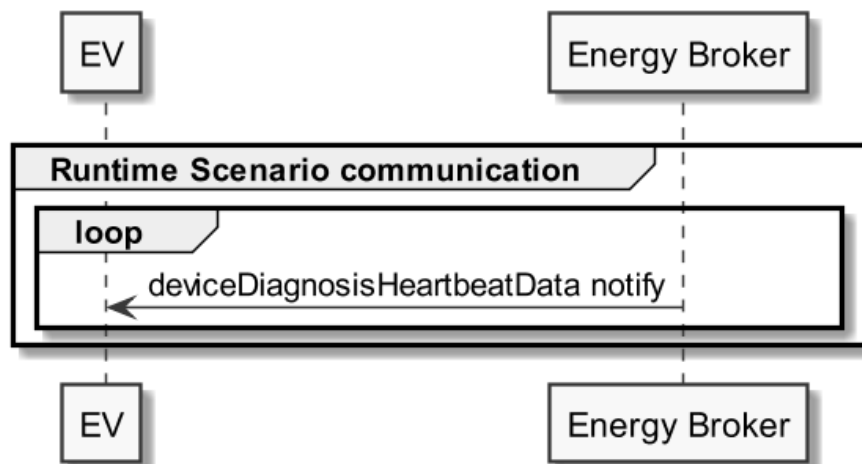


Figure 27: Scenario 6 - Runtime Scenario communication sequence diagram

Note: To interpret partial notification messages correctly the information obtained during the Initial Scenario communication phase is necessary.

Note: A read operation ("polling") on all Functions is possible at any time, e.g. if a notification could not be evaluated.

The following table shows where the necessary content of the messages of the sequence diagram is described:

Message name from sequence diagram	Content description in table	Scenario number in table
deviceDiagnosisHeartbeatData notify [CEVC-048]	Table 21	6

Table 38: Runtime Scenario communication content references for Scenario 6

3.4.6.4 Additional information

None.

3.4.7 Scenario 7 - Energy Guard error state

3.4.7.1 Pre-Scenario Communication

- Detailed Discovery:** Actors that act as client within this Scenario, need to know the addresses of the server Features used in the Initial Scenario communication. If an address of a particular server Feature is not known, the detailed discovery has to be used, as described in section 3.3.2.
- Binding:** Binding SHOULD NOT be used for this Scenario.
- Subscription:** Actors SHALL create a subscription for each server Feature that is relevant for the corresponding Actor within this Scenario, as described in section 3.3.4.

The Initial Scenario communication SHALL start at the latest when the required resources on an Actor are known and the necessary binding and subscription procedures have been finished. However, as soon as an address of a required resource is known, the Initial Scenario communication for this resource MAY start already, even if addresses of other required resources are not known yet.

If required resources are removed and added again, they are re-discovered, and the Initial Scenario communication is triggered again for those resources.

3.4.7.2 Initial Scenario communication

Each time a (re-)connection is established, even if the Pre-Scenario communication phase is skipped, the messages as shown in the following sequence diagram SHALL be exchanged, as the corresponding resources may have changed in the meantime:

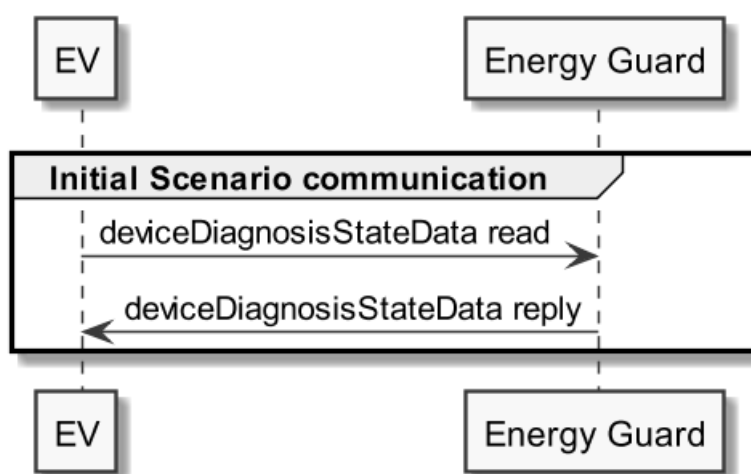


Figure 28: Scenario 7 - Initial Scenario communication sequence diagram

The following table shows where the necessary content of the messages from the sequence diagram is described:

Message name from sequence diagram	Content description in table	Scenario number in table
deviceDiagnosisStateData reply	Table 16	7

Table 39: Initial Scenario communication content references for Scenario 7

Note: Within the Initial Scenario communication the content required by this Scenario MAY not be provided completely but later on during Runtime Scenario communication.

3.4.7.3 Runtime Scenario communication

Based on the Initial Scenario communication the Runtime Scenario communication provides updates during runtime.

If one of the referenced server Functions' data change, the server SHALL submit the change as shown in the following figure:

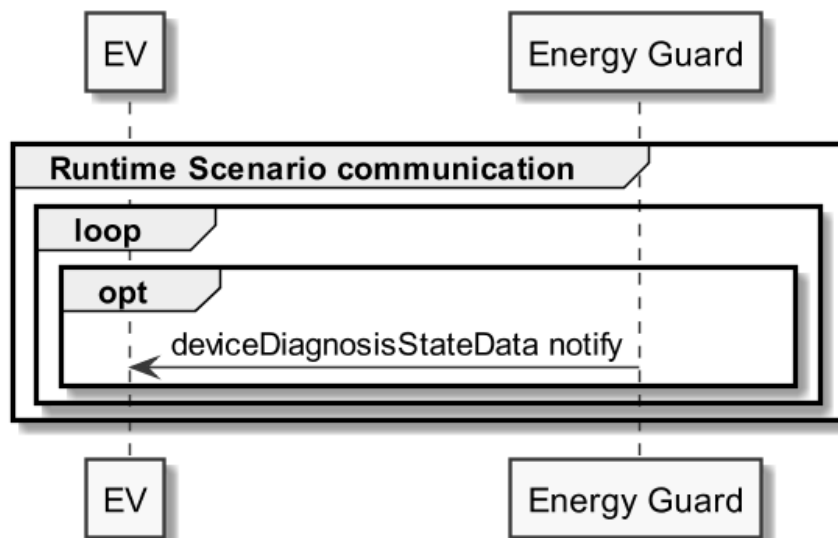


Figure 29: Scenario 7 - Runtime Scenario communication sequence diagram

Note: To interpret partial notification messages correctly the information obtained during the Initial Scenario communication phase is necessary.

Note: A read operation ("polling") on all Functions is possible at any time, e.g. if a notification could not be evaluated.

The following table shows where the necessary content of the messages of the sequence diagram is described:

Message name from sequence diagram	Content description in table	Scenario number in table
deviceDiagnosisStateData notify [CEVC-052]	Table 16	7

Table 40: Runtime Scenario communication content references for Scenario 7

3.4.7.4 Additional information

None.

3.4.8 Scenario 8 - Energy Broker error state

3.4.8.1 Pre-Scenario Communication

- Detailed Discovery:** Actors that act as client within this Scenario, need to know the addresses of the server Features used in the Initial Scenario communication. If an address of a particular server Feature is not known, the detailed discovery has to be used, as described in section 3.3.2.
- Binding:** Binding SHOULD NOT be used for this Scenario.
- Subscription:** Actors SHALL create a subscription for each server Feature that is relevant for the corresponding Actor within this Scenario, as described in section 3.3.4.

The Initial Scenario communication SHALL start at the latest when the required resources on an Actor are known and the necessary binding and subscription procedures have been finished. However, as soon as an address of a required resource is known, the Initial Scenario communication for this resource MAY start already, even if addresses of other required resources are not known yet.

If required resources are removed and added again, they are re-discovered, and the Initial Scenario communication is triggered again for those resources.

3.4.8.2 Initial Scenario communication

Each time a (re-)connection is established, even if the Pre-Scenario communication phase is skipped, the messages as shown in the following sequence diagram SHALL be exchanged, as the corresponding resources may have changed in the meantime:

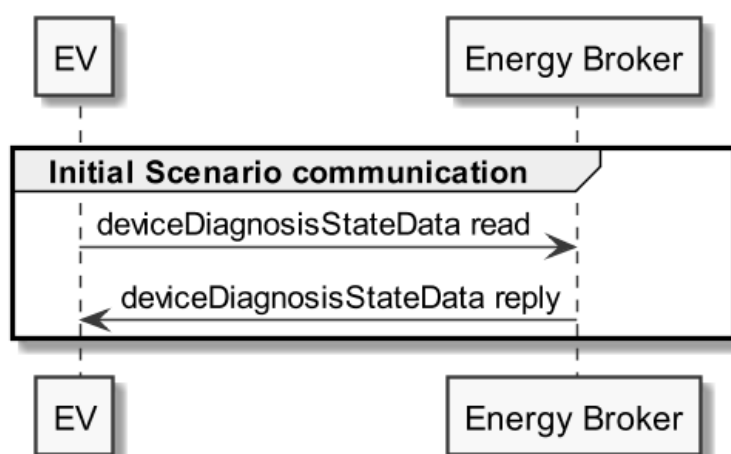


Figure 30: Scenario 8 - Initial Scenario communication sequence diagram

The following table shows where the necessary content of the messages from the sequence diagram is described:

Message name from sequence diagram	Content description in table	Scenario number in table
deviceDiagnosisStateData reply	Table 22	8

Table 41: Initial Scenario communication content references for Scenario 8

Note: Within the Initial Scenario communication the content required by this Scenario MAY not be provided completely but later on during Runtime Scenario communication.

3.4.8.3 Runtime Scenario communication

Based on the Initial Scenario communication the Runtime Scenario communication provides updates during runtime.

If one of the referenced server Functions' data change, the server SHALL submit the change as shown in the following figure:

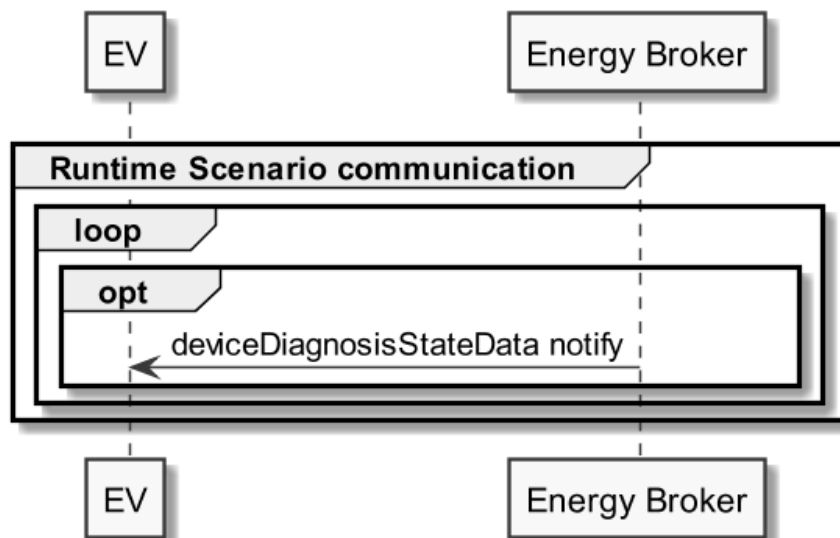


Figure 31: Scenario 8 - Runtime Scenario communication sequence diagram

Note: To interpret partial notification messages correctly the information obtained during the Initial Scenario communication phase is necessary.

Note: A read operation ("polling") on all Functions is possible at any time, e.g. if a notification could not be evaluated.

The following table shows where the necessary content of the messages of the sequence diagram is described:

Message name from sequence diagram	Content description in table	Scenario number in table
deviceDiagnosisStateData notify [CEVC-054]	Table 22	8

Table 42: Runtime Scenario communication content references for Scenario 8

3.4.8.4 Additional information

None.