



BSI Standards Publication

Household appliances network and grid connectivity

Part 3-1: Specific Data Model Mapping: SPINE and SPINE-IoT

National foreword

This British Standard is the UK implementation of EN 50631-3-1:2023.

The UK participation in its preparation was entrusted to Technical Committee CPL/59, Performance of household electrical appliances.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2023
Published by BSI Standards Limited 2023

ISBN 978 0 539 16886 0

ICS 97.120

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2023.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English Version

Household appliances network and grid connectivity - Part 3-1: Specific Data Model Mapping: SPINE and SPINE-IoT

Appareils domestiques connectés au réseau et réseau
intelligent - Partie 3-1: Mapping avec des modèles de
données spécifiques: SPINE et SPINE-IoT

Netzwerk- und Stromnetz-Konnektivität von
Haushaltsgeräten - Teil 3-1: Mapping auf spezifische
Datenmodelle: SPINE und SPINE-IoT

This European Standard was approved by CENELEC on 2023-02-13. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.



European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Contents

Page

European foreword	3
Introduction	3
1 Scope	5
2 Normative references	5
3 Terms and definitions	5
4 Reader's guide	10
4.1 Reading the graphics	10
4.1.1 General	10
4.1.2 Hierarchy diagram	10
4.1.3 Sequence diagram	11
4.2 Finding the right information	12
5 Use Case Function (UCF) details	12
5.1 General	12
5.2 Mapping to SPINE	12
5.2.1 Concepts	12
5.2.2 UCF_AC_Measurement	27
5.2.3 UCF_Characteristics	50
5.2.4 UCF_Configure_Current_Power_Sequence	55
5.2.5 UCF_Consumption_Curve	57
5.2.6 UCF_Device_Configuration	69
5.2.7 UCF_Device_Connected	81
5.2.8 UCF_Heartbeat	82
5.2.9 UCF_Incentive_Table	86
5.2.10 UCF_Overrun	117
5.2.11 UCF_Plan_With_Power_Sequences	124
5.2.12 UCF_Power_Limit	133
5.2.13 UCF_Report_Status_Of_Power_Sequence	139
5.2.14 UCF_Select_Power_Sequence	145
5.2.15 UCF_Shift_Preferred_Power_Sequence	147
5.2.16 UCF_System_Function	150
5.3 Mapping to SPINE-IoT	155
5.3.1 Concepts	155
5.3.2 UCF_Configure_Current_Power_Sequence	157
5.3.3 UCF_Device_Connected	159
5.3.4 UCF_Plan_With_Power_Sequences	159
5.3.5 UCF_Select_Power_Sequence	165
5.3.6 UCF_Shift_Preferred_Power_Sequence	165
5.3.7 YAML models for SPINE-IoT	167
Bibliography	199

European foreword

This document (EN 50631-3-1:2023) has been prepared by WG 07 “Smart Household Appliances” of CLC/TC 59X “Performance of household and similar electrical appliances”.

The following dates are fixed:

- latest date by which this document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2024-02-13
- latest date by which the national standards conflicting with this document have to be withdrawn (dow) 2026-02-13

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC shall not be held responsible for identifying any or all such patent rights.

Any feedback and questions on this document should be directed to the users' national committee. A complete listing of these bodies can be found on the CENELEC website.

Introduction

Energy management systems will more and more become necessary due to change from fossil and nuclear to renewable production and the associated decentralization. Since an appropriate standard for a home and building management is in preparation this document specifies how sets of products from multiple manufacturers can exchange information with Home and Building / Customer Energy Management Systems, located in a home network or in the cloud.

This document focuses on interoperability of household appliances and describes the necessary control and monitoring. It defines a set of functions of household and similar electrical appliances. The functions in this document cover next to energy-management main remote-control and – monitoring use cases.

This document does not deal with safety and security requirements. Safety requirements have been set in the EN 60335 series [1]. EN 50631 will provide interoperability on information exchange among various appliances in the home. The EN 50631 document series will be re-arranged regarding the further development and will be split into 6 parts:

- EN 50631-1, *Household appliances network and grid connectivity — Part 1: General Requirements, Generic Data Modelling and Neutral Messages*
- EN 50631-2, *Household appliances network and grid connectivity — Part 2: Product Specific mappings, details, requirements and deviations*
- EN 50631-3-x, *Household appliances network and grid connectivity — Part 3: Specific Data Model Mapping*
- EN 50631-4-x, *Household appliances network and grid connectivity — Part 4: Communication Protocol Specific Aspects*
- EN 50631-5, *Household appliances network and grid connectivity — Part 5: General Test-Requirements and - Specification*
- EN 50631-6, *Household appliances network and grid connectivity — Part 6: SPINE Data Model Toolbox*

Data communication heavily depends on the environment of appliances. Sometimes low bitrate or energy efficient communication puts strict requirements to selected communication technologies. Therefore, popular and de facto standards had been and will be developed by the industry to fulfil such requirements. To not influence common data modelling for appliances because of such restrictions, the standardized data models and neutral message structures need to be applied to communication technologies.

This standard series therefore is intended to separate data modelling and neutral message structure from the attached communication.

Part 1 defines general requirements, generic data modelling and generic neutral messages without relation to any specific communication technology or any product specific layout.

Part 2 lists and specifies product specific requirements and implementation guidance based on the generic data model and generic neutral messages.

Part 3 defines the mapping of neutral messages to examples of typical data models like SPINE, SPINE-IoT, OCF, and so forth. These data models are neither mandatory nor to be seen as complete spectrum of data models.

Part 4 defines the mapping of neutral messages to examples of typical communication protocols. These communication protocols are neither mandatory, nor do they provide an exhaustive list of communication protocols.

Part 5 defines testing requirements and testing specifications. This part will be covered in the future by a New Work Item Proposal.

Part 6 provides the technical reference specification for the SPINE data model. This part will be covered in the future by a New Work Item Proposal.

1 Scope

This document maps the generic use case functions and data models defined in EN 50631-1:2023 to specific languages; in this case, SPINE and SPINE-IoT.

This document is part of the EN 50631 series, which defines the information exchange between Smart Appliances and management systems in homes and buildings including energy management.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1

appliance

electrical apparatus intended for household or similar use

EXAMPLES Refrigerators, dishwashers, clothes washers, clothes dryers, air conditioners, water heaters, circulation pumps, heat pumps, etc.

3.2

Appliance Energy Flexibility

ability of an appliance to change power consumption in response to an external stimulus

3.3

binding

concept for connecting functionally matching features

3.4

classifier

role that specifies whether a message serves to read, reply, write, etc

3.5

client

role that specifies that a node uses data from a “server” or can request for change

3.6

command

functional part of a Message

3.7

CCM

Customer Connectivity Manager

component or set of functions with the capability to:

1. Receive and process Grid Information, Appliance Information and User Instructions, and
2. Manage one or more Smart Appliances

Note 1 to entry: A CCM may be integrated with a Smart Appliance or may be physically separate.

Note 2 to entry: A CCM manages the energy-using behaviour as well as other aspects of device behaviour (e.g. setting of job status like starting, stopping, pausing, parameters like temperature, notifications...) of one or more Smart Appliances.

Note 3 to entry: In other documentation, CCM is often called Customer Energy Manager (CEM) with a dedicated focus on energy management or called Energy Management System (EMS) with a dedicated focus on energy management.

3.8

data model

definition of possible data (data structures, values) for the exchange of information (especially for communications systems)

[SOURCE: IEC 60050, IEV] [2]

3.9

Demand Response

DR

action resulting from management of the electricity demand in response to supply conditions

[SOURCE: IEC 60050, IEV] [2]

3.10

Demand Side Management

DSM

process that is intended to influence the quantity or patterns of use of electric energy consumed by end-use customers

3.11

device

<specific node> SPINE node that can include a set of entities

Note 1 to entry: It has a "deviceType". With regard to the hierarchy of SPINE nodes a device is a root node for all functionalities offered by a device.

3.12

device

<address information> SPINE address part for the (physical) device

Note 1 to entry: The address part shall be written between quotation marks, e.g., "device".

3.13

deviceType

specific type of physical device (e.g., "WashingMachine", "HeatPump", "FridgeFreezer", etc.)

3.14

discovery

process of finding appropriate partners for communication.

Note 1 to entry: Dependent on the context this can be either finding other devices or examination of a device's potential functionalities.

3.15

element

item (or "attribute") of a SPINE function which holds one information (e.g. "timestamp", "value", etc.) or contains further sub-elements

3.16**entity**

<specific node> SPINE node that can include a set of (sub-)entities or features

Note 1 to entry: It has an "entityType". With regard to the hierarchy of SPINE nodes an entity is a sub-element of a device.

3.17**entity**

<address information> SPINE address part for the (logical) entity

Note 1 to entry: The address part shall be written between quotation marks, e.g., "entity".

3.18**EntityType**

specific type of logical device (e.g. "Freezer" is one logical part of a physical device "FridgeFreezer")

3.19**feature**

<specific node> SPINE node that can include a set of functions (of a class)

Note 1 to entry: It has a "featureType". With regard to the hierarchy of SPINE nodes a feature is a sub-element (i.e. "child") of an entity.

3.20**feature**

<address information> SPINE address part for one feature

Note 1 to entry: The address part shall be written between quotation marks, e.g., "feature".

3.21**FeatureType**

type that defines optional or mandatory rules and a general behaviour of the underlying Class (standard or complex)

3.22**function**

smallest structure to model "actual data" ("functional data"), i.e. functions usually consist of child elements that each hold an information (e.g. "timestamp", "value", etc.)

Note 1 to entry: Information between communication partners is exchanged via the exchange of a function (as part of a so-called "payload").

3.23**message**

one SPINE transfer from a sender to a receiver

3.24**namespace****XML namespace**

set of names that provides a simple method for qualifying element and attribute names used in XML documents by associating them with namespaces identified by URI references [3]

3.25**neutral message**

information exchange that is independent of any specific communication solution

Note 1 to entry: This document describes the mapping of neutral messages to examples of typical communication protocols.

3.26

node

common term for a SPINE instance that has a SPINE address

Note 1 to entry: Dependent on the situation a node can be either a device or an entity (of a specific device) or a feature (of a specific device-entity).

3.27

payload

SPINE Payload, containing the functional SPINE data

3.28

power sequence

expected power consumption over time (i.e., represented as a “curve” of power over time) including options on its flexibility

3.29

RMS

<electricity measurement> Root Mean Square of a set of numbers

3.30

role

each Feature has a functional role, usually either “server” (data owner) or “client”

Note 1 to entry: For some special features (e.g. NodeManagement) the role “special” is defined.

3.31

scope type

feature type that defines scope types for identifying specific functionalities unambiguously (e.g., outsideAirTemperature)

3.32

server

role that specifies that a node offers own data to be read or written by a node with role client

Note 1 to entry: A server can notify its data to other nodes (with role client).

3.33

smart appliance

appliance that is capable of Smart Operation

Note 1 to entry: Notwithstanding the possibly broader concept related to the term “smart appliance”, a smart appliance under the framework of this document needs to be understood as follows:

- 1) It is an appliance that can respond to an external stimulus initiated by a CCM and/or Remote Agent to provide activities such as
 - a. Support Appliance Energy Flexibility
 - b. job status related functions such as starting, stopping, pausing,
 - c. content or level related functions such as temperature, door status.
- 2) The appliance will respond when the user sets conditions and its status allows for a response,

3) The response is a change of the appliance's behaviour like electricity consumption, job status and/or level or content pattern, or a notification thereof,

4) The specific technical smart capabilities need not be activated when the product is placed on the market; the activation can be done at a later point in time by the consumer or a service provider.

Note 2 to entry: Smart appliances in this context can communicate through a Customer Connectivity Manager function processing external signals, such as price information or availability of Renewable Energy Sources (demand response), or direct control signals (demand side management), being able to consider households' preferences or the behaviour of the other home appliances.

3.34

smart operation

operation of an Appliance where the CCM has been set to modify operation automatically in response to Trigger Criteria

Note 1 to entry: Smart operation may be initiated by a CCM.

3.35

SPINE

Smart Premises Interoperable Neutral-message Exchange

3.36

SPINE-IoT

Smart Premises Interoperable Neutral-message Exchange for Internet of Things

3.37

SPINE Data Model

data model which describes the concepts and data model to ensure information exchange between devices like Smart Appliance and CCM, including

- 1) SPINE Protocol, defining a neutral message structure and neutral message exchange
- 2) SPINE Feature Types, describing the specific information to be exchanged

3.38

subscription

concept that enables the receiving of messages of interest from another device without polling it

3.39

Use Case

textual description of a re-usable functionality, consisting of one or more messages from one or more participating actors, which may be visualized with a sequence diagram, e.g. "A CCM shifts the energy usage of a washing machine"

3.40

Use Case Functions

Functions which group basic functionalities that had been derived from use cases

Note 1 to entry: These functions provide the entire information exchange required to implement the considered use cases and user stories.

3.41

XML

Extensible Markup Language

simple, very flexible text format derived from SGML (ISO 8879)

Note 1 to entry: See [3]. Extensible Mark-up Language (XML) defines a set of rules for encoding documents in a format that is both human-readable and machine-readable, enabling multiple publishing options and other applications for a variety of different purposes. Used to model SPINE messages.

3.42

XSD

XML Schema Definition

schema to define and describe a class of XML documents by using schema components to constrain and document the meaning, usage and relationships of their constituent parts: datatypes, elements and their content and attributes and their values, see [3]

Note 1 to entry: The SPINE data model is defined in XSD and supplementary documents (as not every rule can be specified with XSD only). Other formats than XML can be derived from an XSD, too (e.g. JSON).

3.43

YAML

YAML Human-Readable Data-Serialization Language

schema to define and describe a class of XML documents by using schema components to constrain and document the meaning, usage and relationships of their constituent parts: datatypes, elements and their content and attributes and their values

Note 1 to entry: The SPINE-IoT data model is defined in YAML and supplementary documents (as not every rule can be specified with YAML only). Other formats than YAML can be derived from an YAML, too (e.g. JSON).

4 Reader's guide

4.1 Reading the graphics

4.1.1 General

This document contains several graphical representations based on plantUML [4], especially in Clause 6 and later. In order to facilitate their interpretation by readers unfamiliar with UML, the following provides an introduction.

4.1.2 Hierarchy diagram

Within the "Actor [...] overview" diagrams in the "Actors" sub-sections the complete functionality of this Use Case is provided. The Actor MAY have more functionality implemented than needed for this Use Case.

For the following Actor overview example (see Figure 1), a brief description of the graphical symbols will be described.

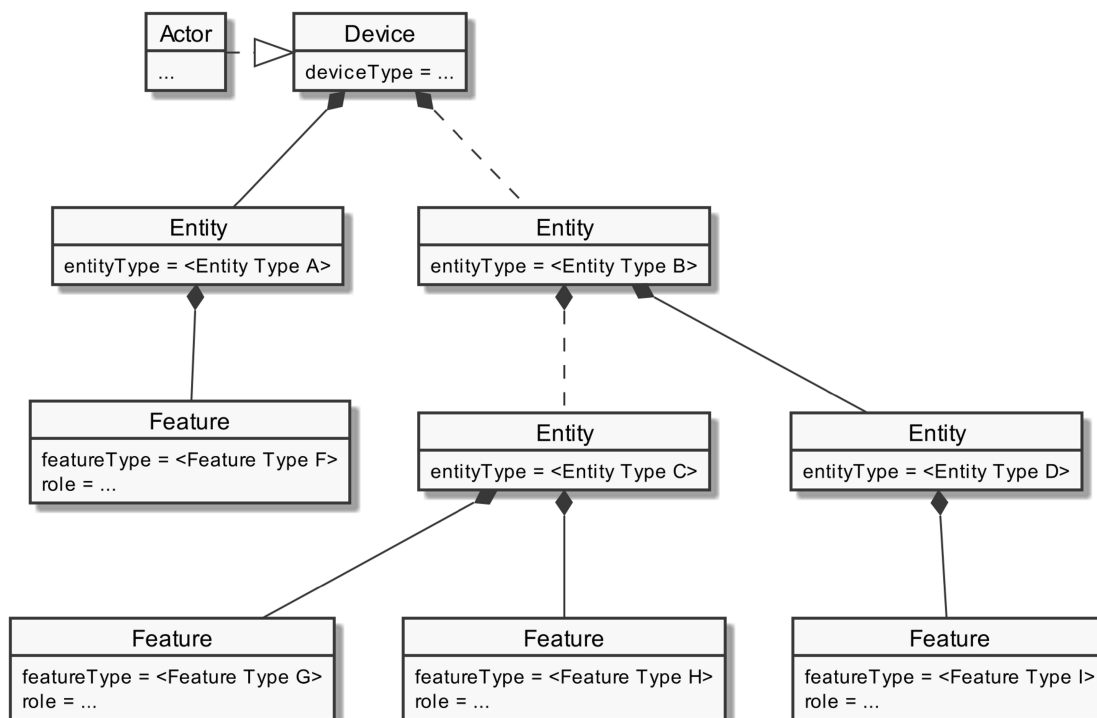


Figure 1 — Actor overview example

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>”.

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.

4.1.3 Sequence diagram

Communication sequence diagrams visualize the messages sent between actors as part of a Use Case function or scenario, as shown in this example in Figure 2:

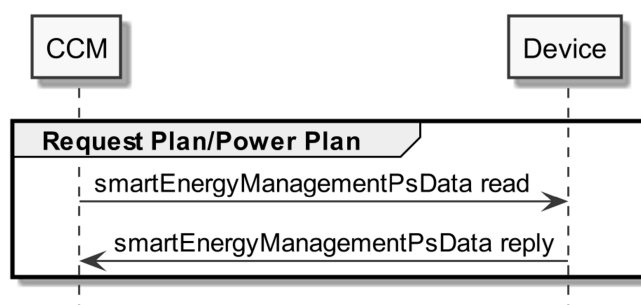


Figure 2 — Example communication sequence diagram

The participating actors are represented by rectangles with their names at the top of the diagram (and optionally at the bottom), as well as by a connecting vertical line.

Horizontal arrows indicate the types of messages sent as well as their direction. In the above example, the first message is a read command of the data structure called “smartEnergyManagementPsData” initiated by the CCM and sent to the Device; the second message is the Device’s reply. Note that time always flows from top to bottom.

4.2 Finding the right information

The Use Case Functions are described in Clause 5. Each Use Case Function is documented in detail as a SPINE and SPINE-IoT reference solution.

5 Use Case Function (UCF) details

5.1 General

In the following, the mapping of the Use Case functions from EN 50631-1 to SPINE and SPINE-IoT is described.

This document defines the mapping of neutral messages to examples of typical data models like SPINE, and SPINE-IoT. Specifically, it describes SPINE-Resources and SPINE-IoT-Resources mapped from the the high-level Use Case Functions as shown in Figure 3.

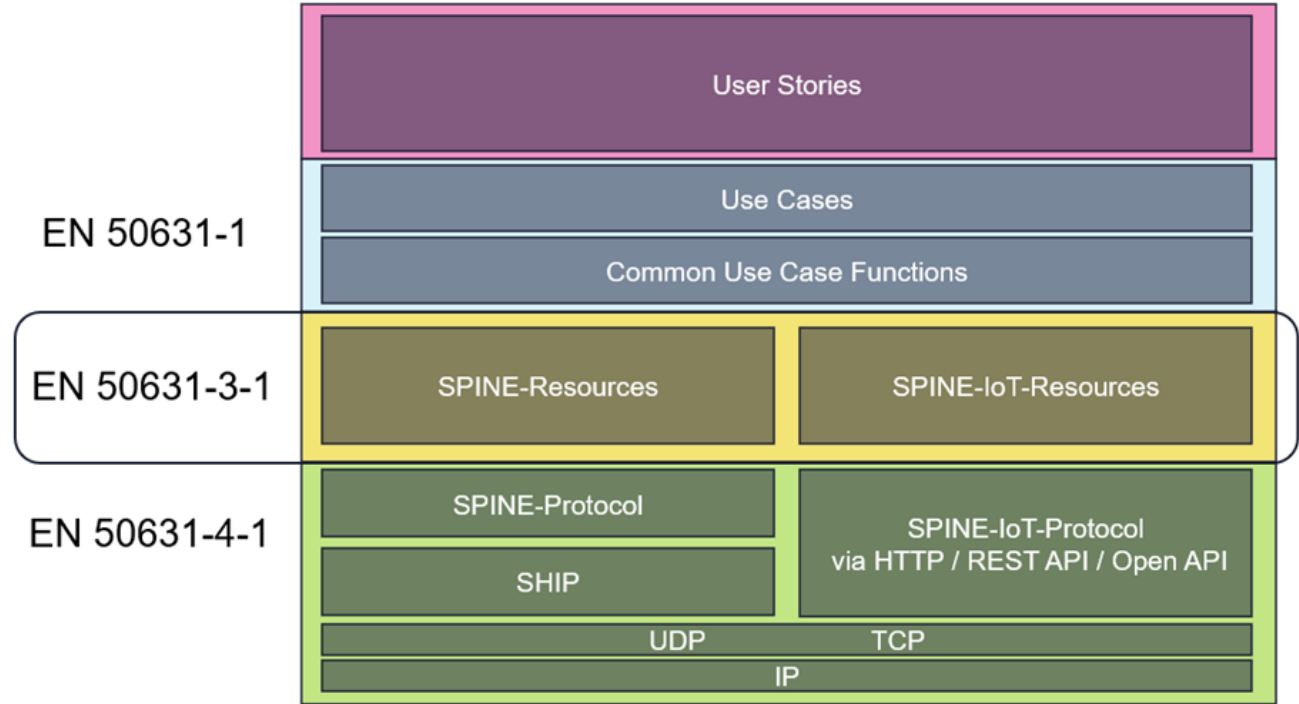


Figure 3 — Description of EN 50631-3

5.2 Mapping to SPINE

5.2.1 Concepts

5.2.1.1 General rules and information

5.2.1.1.1 Underlying technology documents

This technical solution relies on the SPINE Protocol Specification (see EN 50631-4-1) and the SHIP Specification as transport protocol (see EN 50631-4-1).

5.2.1.1.2 Use Case discovery rules

5.2.1.1.2.1 Incentive Table-based Power Consumption Management

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 (refer to the SPINE Protocol Specification EN 50631-4-1) SHALL be “incentiveTableBasedPowerConsumptionManagement”. 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 (refer to the SPINE Protocol Specification EN 50631-4-1) 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 (refer to the SPINE Protocol Specification EN 50631-4-1) to “1.0.0”.
- 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.

5.2.1.1.2.2 Limitation of Power Consumption

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 (refer to the SPINE Protocol Specification EN 50631-4-1) SHALL be “limitationOfPowerConsumption”. 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 (refer to the SPINE Protocol Specification EN 50631-4-1) 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 (refer to the SPINE Protocol Specification EN 50631-4-1) to “1.0.0”.
- 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.

5.2.1.1.2.3 Monitoring and Control of Smart Grid Ready Conditions

Use Case discovery SHOULD be supported by each Actor. If Use Case discovery is supported, the following rules SHALL apply:

- The string content for the Element “nodeManagementUseCaseData. useCaseInformation. useCaseSupport. useCaseName” within the Use Case discovery (refer to the SPINE Protocol Specification EN 50631-4-1) SHALL be “monitoringAndControlOfSmartGridReadyConditions”. 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 (refer to the SPINE Protocol Specification EN 50631-4-1) 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 (refer to the SPINE Protocol Specification EN 50631-4-1) to “1.0.0”.
- 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.

5.2.1.1.2.4 Monitoring of Power Consumption

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 (refer to SPINE Protocol Specification EN 50631-4-1) SHALL be “monitoringOfPowerConsumption”. 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 (refer to SPINE Protocol Specification EN 50631-4-1) 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 (refer to SPINE Protocol Specification EN 50631-4-1) to “1.0.0”.
- 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.

5.2.1.1.2.5 Optimization of Self Consumption by Heat Pump Compressor Flexibility

Use Case discovery SHOULD be supported by each Actor. If Use Case discovery is supported, the following rules SHALL apply:

- The string content for the Element “nodeManagementUseCaseData. useCaseInformation. useCaseSupport. useCaseName” within the Use Case discovery (refer to the SPINE Protocol Specification EN 50631-4-1) SHALL be “optimizationOfSelfConsumptionByHeatPumpCompressorFlexibility”. 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 (refer to the SPINE Protocol Specification EN 50631-4-1) 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 (refer to the SPINE Protocol Specification EN 50631-4-1) to “1.0.0”.
- 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.

5.2.1.1.3 Rules for “Information content” tables

5.2.1.1.3.1 General presence indication definitions

Abbreviations for the presence indication of Elements listed in the tables are defined as follows (see Table 1):

Table 1 — Presence indication description

Abbreviation	Meaning	Link to requirement keywords
M	Mandatory	SHALL
R	Recommended	SHOULD
O	Optional	MAY
F	Forbidden	SHALL NOT

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”; refer to SPINE Protocol Specification EN 50631-4-1). 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).

In some cases, an Element may have a double presence indication, separated by a colon.

Examples:

M, O

M \W, O \W

The description (rules) of the Element details in such a case the presence requirement. A typical example is a list-based Element where the “first” (or “last”) Element has a different presence requirement than the other list Elements.

5.2.1.1.3.2 Cardinality indications on Elements and list items

A cardinality indication on an Element or list item expresses constraints on the number of occurrences of a given Element or data set. In this subclause 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 regard to the given Scenario or Function definition for this Use Case.

Table 2 is an example to explain this for two different placements.

Table 2 — Example table for cardinality indications on Elements and list items

Scenario [...] : M/R/O [W][C]	Element	Value	Element and value rules
..	...		...
2: M W	xFeatureType. xListData. xData. (1..3)		
2: M W	xId	< *#(1..) >	PRIMARY IDENTIFIER
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>	...
...	...	...	...

The field

xFeatureType. xListData. xData. (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 constraint 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 field

xSlot. (1..)

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

For the expression “< *#(1..) >” of Element “xId”, see 5.2.1.1.3.4.

The remaining fields do not have an explicit cardinality indication.

NOTE Cardinality expressions are also used within placeholder expressions as defined in 5.2.1.1.3.4. In many cases such placeholder expressions define the number of required or permitted Elements or list items as they explicitly define how many different values for a given Identifier are required or permitted for a given Scenario.

5.2.1.1.3.3 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).

“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 1 “\W” includes “\C” already.

NOTE 2 Depending on the resource a client might need to request a proper binding before the server accepts a “write” operation.

5.2.1.1.3.4 “Value” placeholders

Introduction

Lists of message content may use placeholders to model relations between different Elements or even list items of different Functions. The main purpose is to declare which Identifier values relate to each other. As a Use Case does not prescribe specific values to be used for a given Identifier, a placeholder like “<x1>” can be used in “Value” columns to express the intended relations.

There are two styles to identify a placeholder that can be referenced:

1. <xM>
2. < xM#S >

where

1. “x” is any alphabetical prefix like “m”, “t”, “ec”, “cc”, etc.
2. “M” is a (major) number like “1”, “2”, “15”, etc.
3. “S” is a sub-number like “1”, “7”, “10”, etc.

Examples for the first style are “<ec1>”, “<z12>”. Examples for the second style are “<p4#2>”, “<m22#3>”. For a given placeholder, only one of the styles can be used.

In addition, there are also styles for placeholders that do not need to be referenced:

1. < * >
2. < *#S >

The second style is only used with so-called cardinality expressions.

Uniqueness of placeholders

A given placeholder $\langle xM \rangle$ or $\langle xM\#S \rangle$ represents the same value throughout a given Use Case specification for a given set of its parent Identifier values. This shall be explained in a brief example:

We assume a list item with PRIMARY IDENTIFIER “pId”. It also has a SUB IDENTIFIER “sId” with placeholder “ $\langle s1 \rangle$ ”. Then, each occurrence of “ $\langle s1 \rangle$ ” represents the same value for a given value of pId. This means that “ $\langle s1 \rangle$ ” of a list item with pId = 1 can differ from “ $\langle s1 \rangle$ ” of a list item with pId = 2. But it also means that “ $\langle s1 \rangle$ ” represents the same value in all list items with pId = 1.

NOTE Typically, parent Identifiers like “pId” will also be expressed with a placeholder like “ $\langle p5 \rangle$ ”, e.g. In this case, the uniqueness rule applies for “ $\langle p5 \rangle$ ” likewise.

NOTE The uniqueness also applies for placeholders used as FOREIGN IDENTIFIER.

Placeholder expressions with cardinalities

For some Identifiers, more than one placeholder is needed. Several notations are used for this purpose, which make use of cardinality expressions. The general notation is as follows:

1. $\langle xM\#(a..b) \rangle$

This is equivalent to this explicit definition:

At least a and at maximum b placeholders of this list: $\langle xM\#1 \rangle \langle xM\#2 \rangle \dots \langle xM\#b \rangle$

This means that the implementation of a given Use Case (or Scenario) requires a minimum of “a” distinct values of the respective Identifier. In total, there can be up to “b” distinct values of the respective Identifier.

Additionally, the following notations may occur:

2. $\langle xM\#(a..) \rangle$

This is equivalent to “ $\langle xM\#(a..b) \rangle$ ” with “b” equal to infinity.

3. $\langle xM\#(..b) \rangle$

This is equivalent to “ $\langle xM\#(a..b) \rangle$ ” with “a” equal to zero.

“ $\langle xM\#(a..) \rangle$ ” has only a lower bound of “a” distinct values, but no upper bound. “ $\langle xM\#(..b) \rangle$ ”, on the other hand, expresses that the Identifier may not be required at all, but it is permitted to have up to “b” distinct values.

Similarly, the cardinality can be used for placeholders that are not referenced, i.e. $\langle * \#(a..b) \rangle$ etc.

NOTE The cardinality does NOT express which of the sub-numbers have to be used! I.e. it does NOT mean that the Identifiers $\langle xM\#1 \rangle \dots \langle xM\#a \rangle$ are always used and just those with larger sub-numbers ($\langle xM\#a+1 \rangle \dots \langle xM\#b \rangle$) are optional. If, for instance, a placeholder expression “ $\langle xM\#(3..5) \rangle$ ” is given, it may well happen that an implementation makes use of $\langle xM\#2 \rangle$, $\langle xM\#4 \rangle$, and $\langle xM\#5 \rangle$ (i.e. it does NOT use $\langle xM\#1 \rangle$, $\langle xM\#3 \rangle$). Which sub-numbers are used usually depends on other parts of a Message and their references to required placeholders, which is explained in 5.2.1.3.4.4.

5.2.1.1.3.4.1 References to placeholders and relations

According to the styles for placeholders that can be referenced, an enumeration value “e” can refer to a particular placeholder:

1. $e(- \rangle \langle xM \rangle$

2. $e(- > < xM\#S >)$

This denotes that “e” is to be used with “<xM>” or “<xM#S>”, resp.

Example message contains the Elements “mld” and “phase”. “mld” is an Identifier with placeholder definition $< m2\#(1..3) >$. “phase” is a string that permits the values “a”, “b”, and “c” using this expression:

“a”($->< m2\#1 >$)

“b”($->< m2\#2 >$)

“c”($->< m2\#3 >$)

This expresses that the enumeration value “a” is to be used with the placeholder $< m2\#1 >$, “b” is to be used with $< m2\#2 >$ and “c” with $< m2\#3 >$.

Similarly, a placeholder “yN” can refer to a particular placeholder:

3. $< yN->xM >$

4. $< yN->xM\#S >$

5. $< yN\#T- > xM >$

6. $< yN\#T- > xM\#S >$

where “T” is a sub-number of “yN”.

It is also feasible to associate placeholders with cardinalities:

7. $< yN\#(a..b)- > xM\#(a..b) >$

denotes that $< yN\#1 >$ is to be used with $< xM\#1 >$, $< yN\#2 >$ is to be used with $< xM\#2 >$, etc.

In this case, the placeholder expressions of yN and xM shall have the same cardinality.

In some cases, there is a need to express that multiple list items with similar values are feasible or required, but only particular combinations of these different data are permitted. The following example shows that several “fData” list items with different “phase” value are required, but that these list items may only express either the “phase” value combination { “a”, “b”, “c” } or the “phase” value combination { “a”, “abb”, “neutral” }. The permitted combinations are defined in a note below Table 3:

Table 3 — Content of an example table

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
2: M	F. fListData. fData.		
2: M	zId	< z3#(3..5) >	
2: M	phase	"a"(- > < z3#1 >)	
		"b"(- > < z3#2 >)	
		"c"(- > < z3#3 >)	
		"abc"(- > < z3#4 >)	
		"neutral"(- > < z3#5 >)	

One of the following combinations SHALL be used at least: { < z3#1 > , < z3#2 > , < z3#3 > } or { < z3#1 > , < z3#4 > , < z3#5 > }.

5.2.1.1.3.5 Rules for content of "Value" column

For a given Scenario, the "Value" column may restrict the permitted content of an 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 5.2.1.1.3.3):
 - the server SHALL support at least one of the listed values.
 - the client SHALL support all listed values.
- In case of Elements that are writeable or changeable (see 5.2.1.1.3.3):

- the server SHALL support all listed values.
- 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)”.

5.2.1.1.4 “Actor ... overview” diagram rules

Within the “Actor [...] overview” diagrams in the “Actors” sub-sections the complete functionality of this Use Case is provided, including optional Scenarios. Which Scenarios are optional can be found in the table under „Scenario Support Requirements”. The Actor MAY have more functionality implemented than needed for this Use Case.

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

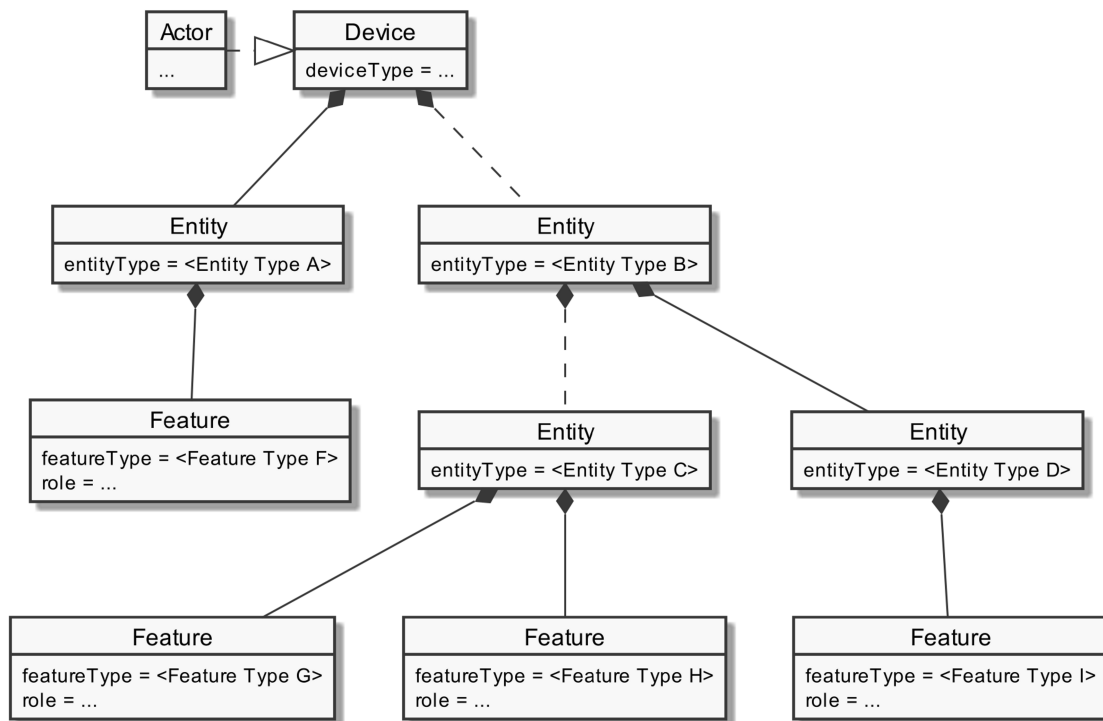


Figure 4 — Actor overview example

The solid lines in Figure 4 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 Figure 4 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 C>”.

5.2.1.1.5 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.

5.2.1.1.6 Further information and rules

5.2.1.1.6.1 Frequently used Element rules

This subclause serves as a collection of rules frequently used by the tables of the subsequent sections. Each rule applies only where referenced explicitly in the tables.

NOTE The purpose of this collection is just to reduce the size of the tables. As such, no rule has a meaning without a reference indicating the required rule. A reference looks like “See [Measurement value rules]”, e.g.

[Measurement value rules]:

SHALL be set if a value is available. Otherwise the whole list entry SHALL be omitted or the Element *valueState* SHALL be set to “error”.

If *valueState* is set to “error”, but *value* is set, the content of *value* SHALL be ignored.

If *valueState* is set to “outOfRange”, but *value* is set, the content of *value* SHALL be interpreted as being out of range.

If *valueState* is set to “outOfRange”, *measurementConstraintsListData.valueRangeMax* is set and *value* is equal or bigger than *valueRangeMax*, *value* SHALL be interpreted as above *valueRangeMax*.

If *valueState* is set to “outOfRange”, *measurementConstraintsListData.valueRangeMin* is set and *value* is equal or smaller than *valueRangeMin*, *value* SHALL be interpreted as below *valueRangeMin*.

If set, *measurementDescriptionListData.measurementType* SHALL be set, too.

[Value state rules]:

The Element *valueState* SHALL be set if its content differs from “normal”. This means, if the state of the value is “outOfRange” or “error” this SHALL be denoted in the *valueState* Element. A client SHALL always consider the content of *valueState*, if set. If omitted, a value of “normal” is assumed.

[String-length rules]:

The string-length SHOULD NOT be longer than 256 characters. If it is longer, the sender SHALL consider the possibility that the receiver will shorten the string to 256 characters.

[Scaled number rules]:

The sub-Elements “number” and “scale” represent a value according to the following formula: $\text{number} \cdot 10^{\text{scale}}$.

[Evaluation period rules]:

The “evaluationPeriod” expresses the analysis period for the according value.

5.2.1.1.6.2 Rules regarding the usage of time-related information

For all Use Cases but “Optimization of Self Consumption by Heat Pump Compressor Flexibility”: All time-related information within the Use Cases SHALL be presented as absolute times. This means relative times SHALL NOT be used, unless the data point only supports relative times (e.g. for a “duration”).

This means the Actors need to have a common understanding of these absolute times. It is strongly recommended that each Actor is “sufficiently synchronised” with the official “Coordinated Universal Time” (UTC).

For Use Case “Optimization of Self Consumption by Heat Pump Compressor Flexibility”: The time-related Elements “alternatives. powerSequence. schedule. startTime”, “alternatives. powerSequence. scheduleConstraints. earliestStartTime”, and “alternatives. powerSequence. scheduleConstraints. latestEndTime” SHALL be presented as relative times. A current relative time is expressed as duration relative

to the current (absolute) time (positive durations refer to the future, negative durations refer to the past). This means absolute times SHALL NOT be used.

5.2.1.1.6.3 Applied sign convention

Throughout the whole Use Case the “load convention” (i.e. “passive sign convention”) is applied for measurement values. This means measured electrical current, active power and exchanged energy are expressed with positive values in case of energy consumption whereas negative values are used in case of energy production. Voltages are measured independent of the energy direction.

5.2.1.2 Pre-Scenario communication

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

5.2.1.2.2 Detailed discovery

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

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

1. If a client is not subscribed to the primary NodeManagement instance, the client SHALL acquire a subscription according to Figure 5 provided within this subclause.

2. A client SHALL read the detailed discovery information according to Figure 5 provided within this subclause. It SHALL keep the received information as far as it concerns mandatory and supported optional Entity Types, Feature Types and Functions of this Use Case that are needed by the client. This means that a client may choose how to store the necessary information. E.g. a client Actor can store the information how to address the necessary Features of the implemented Scenarios but may discard the Entity information.
3. If and as long as a client has a subscription to the detailed discovery information of an Actor and receives proper notifications, it SHALL consider (i.e. integrate into the kept detailed discovery information) the received information as far as it concerns mandatory and supported optional Entity Types, Feature Types and Functions of this Use Case.

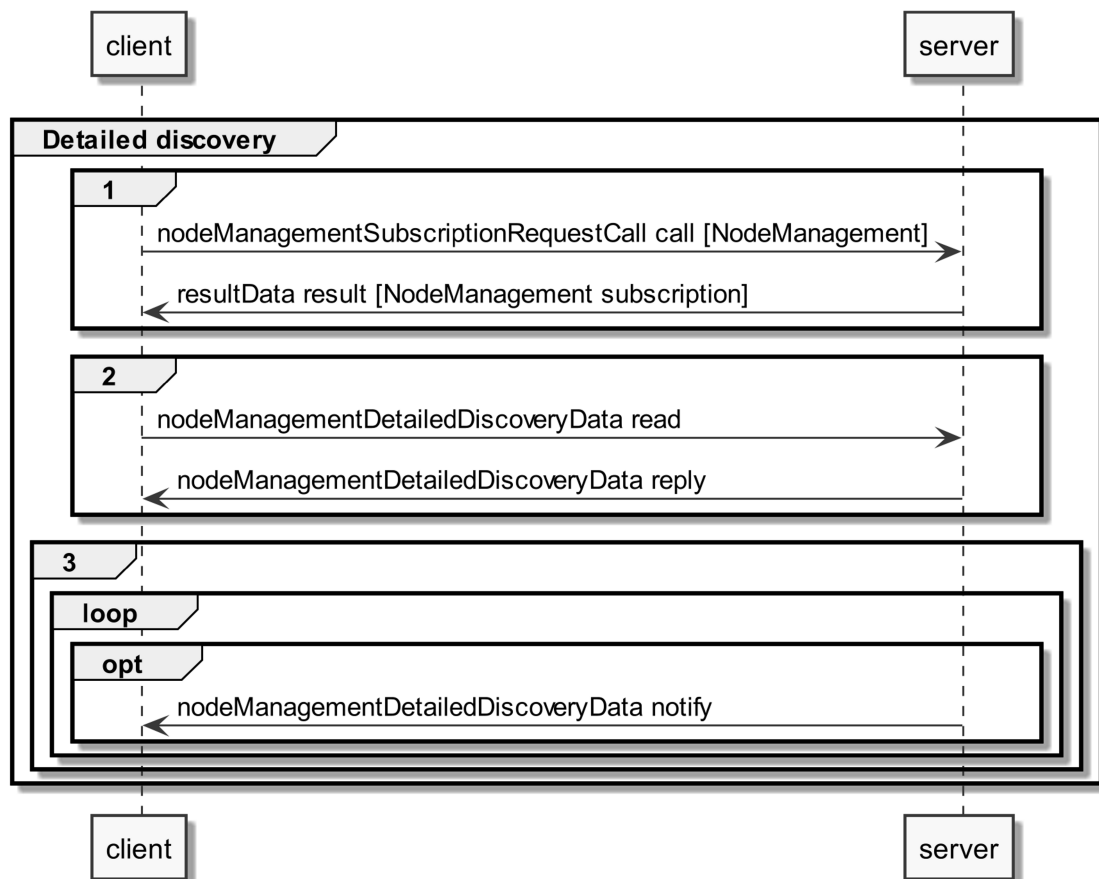


Figure 5 — Pre-Scenario communication - Detailed discovery sequence diagram

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.

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”. 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

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

5.2.1.2.3 Binding

If binding is required by a Scenario that uses Features with writeable or changeable data, the server SHALL support binding for the respective Features. 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 6):

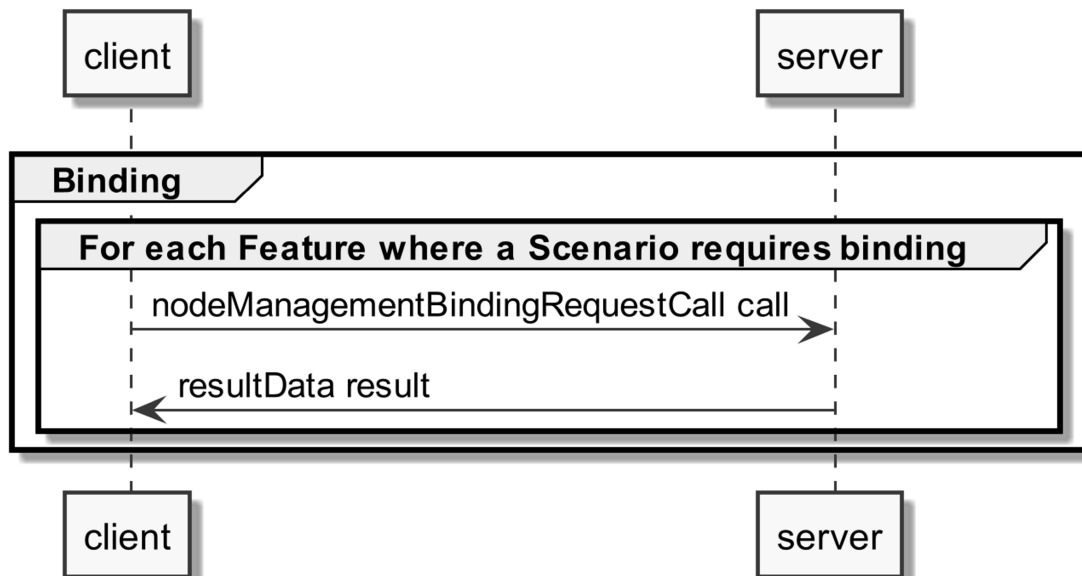


Figure 6 — Pre-Scenario communication - Binding sequence diagram

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.

5.2.1.2.4 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 7):

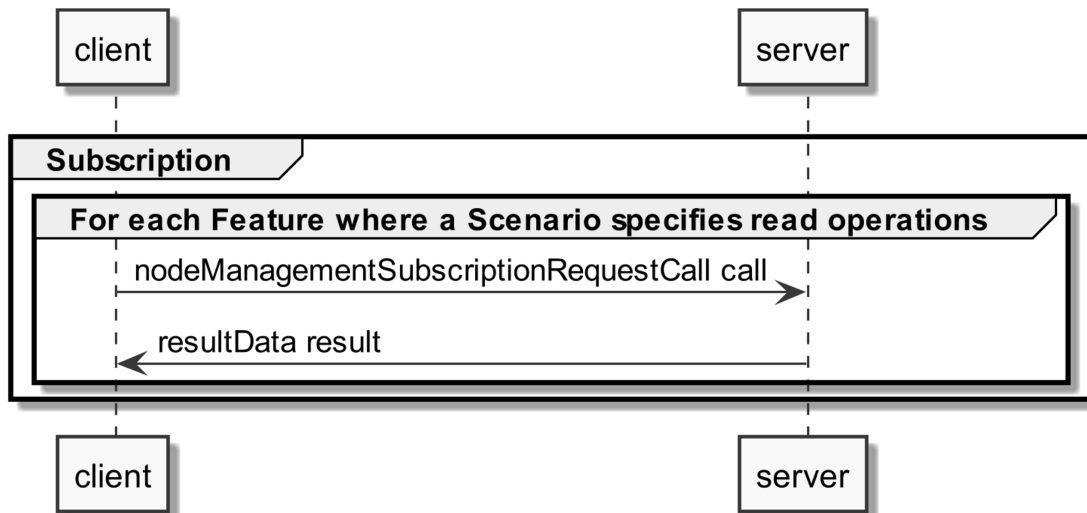


Figure 7 — Pre-Scenario communication - Subscription sequence diagram

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.

5.2.1.2.5 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.

5.2.2 UCF_AC_Measurement

5.2.2.1 Generic Sequence Diagram

Figure 8 describes a generic sequence diagram of UCF_AC_Measurement.

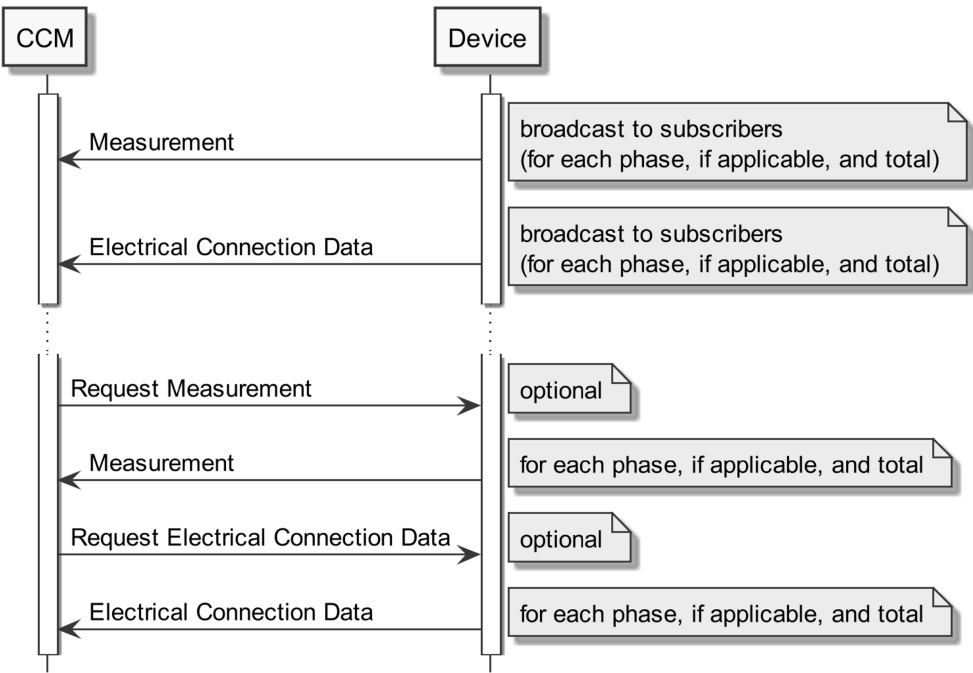


Figure 8 — Generic Sequence Diagram for UCF_AC_Measurement

5.2.2.2 SPINE Mapping

5.2.2.2.1 Request Measurement

5.2.2.2.1.1 General

This is a set of three SPINE partial “read” messages. It is considered best practice to send them in the following sequence, as shown in Figure 9.

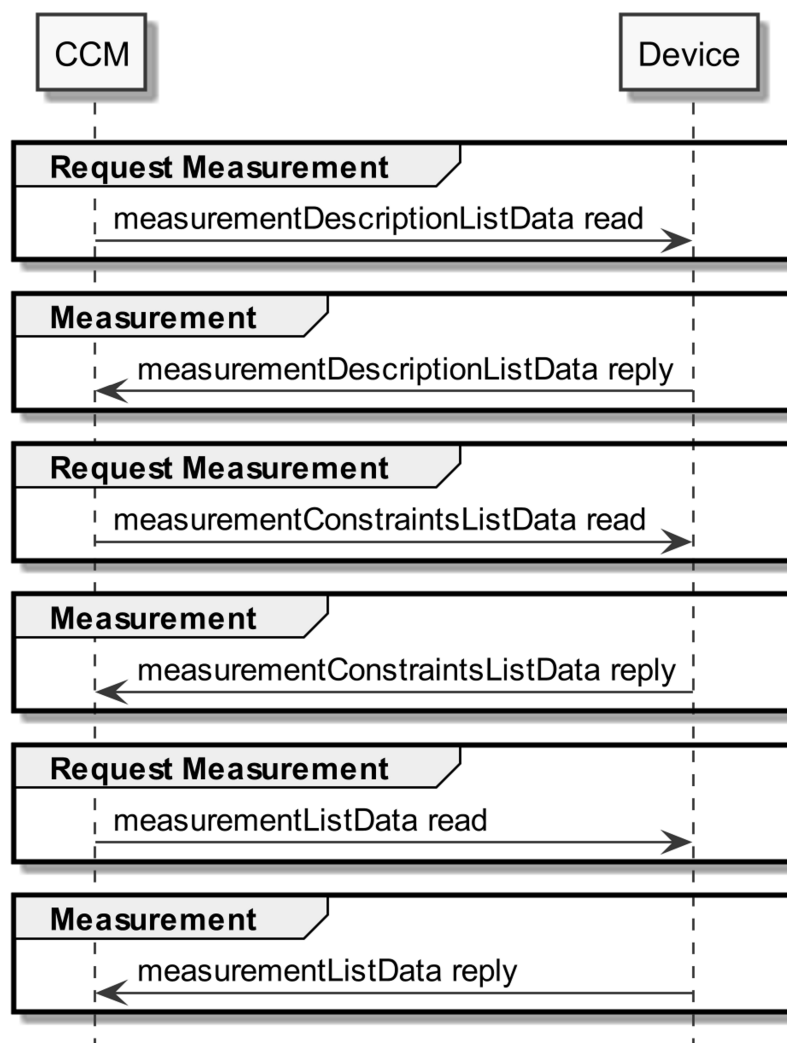


Figure 9 — UCF_AC_Measurement: Request Measurement/Measurement communication sequence diagram

The messages have the following payload (see Tables 4 to 33):

5.2.2.2.1.2 measurementDescriptionListData read

Table 4 — Content of measurementDescriptionListData read at Actor CCM (for “Monitoring of Power Consumption”)

Element		Value	
function		measurementDescriptionListData	
filter			
	cmdControl	partial	
measurementDescriptionDataSelectors			
	scopeType	Scope Type acPowerTotal acPower acEnergyConsumed acCurrent acVoltage acFrequency	Used for Measurement of: AC Power (non-phase specific) AC power (phase-specific) Consumed AC energy AC current. AC voltage. AC Frequency

5.2.2.2.1.3 measurementConstraintListData read

Table 5 — Content of measurementConstraintListData read at Actor CCM

Element		Value
function		measurementConstraintListData
filter		
cmdControl		partial
measurementConstraintListDataSelectors		
	measurementId	measurementId (derived from the measurementDescriptionListData reply)

5.2.2.2.1.4 measurementListData read

Table 6 — Content of measurementListData read at Actor CCM

Element		Value
function		measurementListData
filter		
cmdControl		partial
measurementListDataSelectors		
	measurementId	measurementId (derived from the measurementDescriptionListData reply)

These messages are sent during the initial communication phase, as well as on (re-)connect.

5.2.2.2.2 Measurement

For each read from “Request Measurement” above, there is a SPINE “reply” with the following payload:

5.2.2.2.2.1 For Power

measurementDescriptionListData reply

Non-phase specific:

Table 7 — Content of measurementDescriptionListData reply or notify at Actor Device – Power, non-phase specific (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
1: M	Measurement. measurementDescriptionListData. measurementDescriptionData.		
1: M	measurementId	< m1#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
1: M	measurementType	“power”	
1: M	commodityType	“electricity”	
1: M	unit	“W”	
1: M	scopeType	“acPowerTotal”	

Phase-specific:

Table 8 — Content of measurementDescriptionListData reply or notify at Actor Device – Power, phase-specific (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
1: M	Measurement. measurementDescriptionListData. measurementDescriptionData.		
1: M	measurementId	< m2#(1..3) >	SHALL be set as PRIMARY IDENTIFIER.
1: M	measurementType	“power”	
1: M	commodityType	“electricity”	
1: M	unit	“W”	
1: M	scopeType	“acPower”	

measurementConstraintsListData reply

non-phase specific:

Table 9 — Content of measurementConstraintsListData reply or notify at Actor Device – Power, non-phase specific (for “Monitoring of Power Consumption”)

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
1: R	Measurement. measurementConstraintsListData. measurementConstraintsData.		
1: M	measurementId	< m1#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
1: R	valueRangeMin.		SHOULD be used. [Scaled number rules]
1: M	valueRangeMin. number		SHALL be used.
1: O	valueRangeMin. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
1: R	valueRangeMax.		SHOULD be used. [Scaled number rules]
1: M	valueRangeMax. number		SHALL be used.
1: O	valueRangeMax. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
1: R	valueStepSize.		SHOULD be used. [Scaled number rules]
1: M	valueStepSize. number		SHALL be used.
1: O	valueStepSize. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.

phase-specific:

Table 10 — Content of measurementConstraintsListData reply or notify at Actor Device – Power, phase-specific (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
1: R	Measurement. measurementConstraintsListData. measurementConstraintsData.		
1: M	measurementId	< m2#(1..3) >	SHALL be set as PRIMARY IDENTIFIER.
1: R	valueRangeMin.		SHOULD be used. [Scaled number rules]
1: M	valueRangeMin. number		SHALL be used.
1: O	valueRangeMin. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
1: R	valueRangeMax.		SHOULD be used. [Scaled number rules]
1: M	valueRangeMax. number		SHALL be used.
1: O	valueRangeMax. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
1: R	valueStepSize.		SHOULD be used. [Scaled number rules]
1: M	valueStepSize. number		SHALL be used.
1: O	valueStepSize. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.

measurementListData reply

non-phase specific:

Table 11 — Content of measurementListData reply or notify at Actor Device– Power, non-phase specific (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
1: M	Measurement. measurementListData. measurementData.		
1: M	measurementId	< m1#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
1: M	valueType	“value”	SHALL be set as SUB IDENTIFIER.
1: O	timestamp	< t#(1..1)- > m1#1 >	MAY be used. Within this Use Case, only the newest measurement value SHALL be stated. Additional historical values are

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
			forbidden.
1: M	value.		See [Measurement value rules]. See [Scaled number rules].
1: M	value. number		SHALL be used.
1: O	value. scale		"SHALL be interpreted". If absent, a default value of "0" applies.
1: M	valueSource	"measuredValue"	
		"calculatedValue"	
		"empiricalValue"	
1: R	valueState		[Value state rules]

phase-specific:

**Table 12 — Content of measurementListData reply or notify at Actor Device – Power, phase-specific
(for "Monitoring of Power Consumption")**

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
1: M	Measurement. measurementListData. measurementData.		
1: M	measurementId	< m2#(1..3) >	SHALL be set as PRIMARY IDENTIFIER.
1: M	valueType	"value"	SHALL be set as SUB IDENTIFIER.
1: O	timestamp	< t#(1..1)- > m2#(1..3) >	MAY be used. Within this Use Case, only the newest measurement value SHALL be stated. Additional historical values are forbidden.
1: M	value.		See [Measurement value rules]. See [Scaled number rules].
1: M	value. number		SHALL be used.
1: O	value. scale		"SHALL be interpreted". If absent, a default value of "0" applies.
1: M	valueSource	"measuredValue"	
		"calculatedValue"	
		"empiricalValue"	
1: R	valueState		[Value state rules]

5.2.2.2.2.2 For Energy

measurementDescriptionListData reply

Energy Consumed:

Table 13 — Content of measurementDescriptionListData reply or notify at Actor Device – Energy consumed (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [WJ][C]	Element	Value	Element and value rules
2: M	Measurement. measurementDescriptionListData. measurementDescriptionData.		
2: M	measurementId	< m3#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
2: M	measurementType	“energy”	
2: M	commodityType	“electricity”	
2: M	unit	“Wh”	
2: M	scopeType	“acEnergyConsumed”	

measurementConstraintsListData reply

Energy Consumed:

Table 14 — Content of measurementConstraintsListData reply or notify at Actor Device – Energy consumed (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [WJ][C]	Element	Value	Element and value rules
2: R	Measurement. measurementConstraintsListData. measurementConstraintsData.		
2: M	measurementId	< m3#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
2: R	valueRangeMin.		SHOULD be used. [Scaled number rules]
2: M	valueRangeMin. number		SHALL be used.
2: O	valueRangeMin. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
2: R	valueRangeMax.		SHOULD be used. [Scaled number rules]
2: M	valueRangeMax. number		SHALL be used.
2: O	valueRangeMax. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
2: R	valueStepSize.		SHOULD be used. [Scaled number rules]
2: M	valueStepSize. number		SHALL be used.
2: O	valueStepSize. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.

measurementListData reply

Energy Consumed:

Table 15 — Content of measurementListData reply or notify at Actor Device – Energy consumed (for “Monitoring of Power Consumption”)

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
2: M	Measurement. measurementListData. measurementData.		
2: M	measurementId	< m3#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
2: M	valueType	“value”	SHALL be set as SUB IDENTIFIER.
2: O	timestamp	< t#(1..1)- > m3#1 >	MAY be used. Only the newest measurement value SHALL be stated. Additional historical values are forbidden.
2: M	value.		[Measurement value rules] [Scaled number rules]
2: M	value. number		SHALL be used.
2: O	value. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
2: O	evaluationPeriod		[Evaluation period rules]
2: M	evaluationPeriod. startTime		
2: M	evaluationPeriod. endTime		
2: M	valueSource	“measuredValue”	
		“calculatedValue”	
		“empiricalValue”	
2: R	valueState		[Value state rules]

5.2.2.2.3 For Current

measurementDescriptionListData reply

Table 16 — Content of measurementDescriptionListData reply or notify at Actor Device – Current (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
3: M	Measurement. measurementDescriptionListData. measurementDescriptionData.		
3: M	measurementId	< m5#(1..3) >	SHALL be set as PRIMARY IDENTIFIER.
3: M	measurementType	“current”	
3: M	commodityType	“electricity”	
3: M	unit	“A”	
3: M	scopeType	“acCurrent”	

measurementConstraintsListData reply

Table 17 — Content of measurementConstraintsListData reply or notify at Actor Device – Current (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
3: R	Measurement. measurementConstraintsListData. measurementConstraintsData.		
3: M	measurementId	< m5#(1..3) >	SHALL be set as PRIMARY IDENTIFIER.
3: R	valueRangeMin.		SHOULD be used. [Scaled number rules]
3: M	valueRangeMin. number		SHALL be used.
3: O	valueRangeMin. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
3: R	valueRangeMax.		SHOULD be used. [Scaled number rules]
3: M	valueRangeMax. number		SHALL be used.
3: O	valueRangeMax. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
3: R	valueStepSize.		SHOULD be used. [Scaled number rules]

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
3: M	valueStepSize. number		SHALL be used.
3: O	valueStepSize. scale		"SHALL be interpreted". If absent, a default value of "0" applies.

measurementListData reply

Table 18 — Content of measurementListData reply or notify at Actor Device – Current (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
3: M	Measurement. measurementListData. measurementData.		
3: M	measurementId	< m5#(1..3) >	SHALL be set as PRIMARY IDENTIFIER.
3: M	valueType	"value"	SHALL be set as SUB IDENTIFIER.
3: O	timestamp	< t#(1..1)- > m5#(1..3) >	MAY be used. Only the newest measurement value SHALL be stated. Additional historical values are forbidden.
3: M	value.		[Measurement value rules] [Scaled number rules]
3: M	value. number		SHALL be used.
3: O	value. scale		"SHALL be interpreted". If absent, a default value of "0" applies.
3: M	valueSource	"measuredValue"	
		"calculatedValue"	
		"empiricalValue"	
3: R	valueState		[Value state rules]

5.2.2.2.2.4 For Voltage

measurementDescriptionListData reply

Table 19 — Content of measurementDescriptionListData reply or notify at Actor Device – Voltage (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
4: M	Measurement. measurementDescriptionListData. measurementDescriptionData.		
4: M	measurementId	< m6#(1..6) >	SHALL be set as PRIMARY IDENTIFIER.
4: M	measurementType	“voltage”	
4: M	commodityType	“electricity”	
4: M	unit	“V”	
4: M	scopeType	“acVoltage”	

measurementConstraintsListData reply

Table 20 — Content of measurementConstraintsListData reply or notify at Actor Device – Voltage (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
4: R	Measurement. measurementConstraintsListData. measurementConstraintsData.		
4: M	measurementId	< m6#(1..6) >	SHALL be set as PRIMARY IDENTIFIER.
4: R	valueRangeMin.		SHOULD be used. [Scaled number rules]
4: M	valueRangeMin. number		SHALL be used.
4: O	valueRangeMin. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
4: R	valueRangeMax.		SHOULD be used. [Scaled number rules]
4: M	valueRangeMax. number		SHALL be used.
4: O	valueRangeMax. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
4: R	valueStepSize.		SHOULD be used. [Scaled number rules]
4: M	valueStepSize. number		SHALL be used.

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
4: O	valueStepSize. scale		"SHALL be interpreted". If absent, a default value of "0" applies.

measurementListData reply

Table 21 — Content of measurementListData reply or notify at Actor Device – Voltage (for “Monitoring of Power Consumption”)

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
4: M	Measurement. measurementListData. measurementData.		
4: M	measurementId	< m6#(1..6) >	SHALL be set as PRIMARY IDENTIFIER.
4: M	valueType	"value"	SHALL be set as SUB IDENTIFIER.
4: O	timestamp	< t#(1..1)- > m6#(1..6) >	MAY be used. Only the newest measurement value SHALL be stated. Additional historical values are forbidden.
4: M	value.		[Measurement value rules] [Scaled number rules]
4: M	value. number		SHALL be used.
4: O	value. scale		"SHALL be interpreted". If absent, a default value of "0" applies.
4: M	valueSource	"measuredValue"	
		"calculatedValue"	
		"empiricalValue"	
4: R	valueState		[Value state rules]

5.2.2.2.2.5 For Frequency

measurementDescriptionListData reply

Table 22 — Content of measurementDescriptionListData reply or notify at Actor Device – Frequency (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [WJ][C]	Element	Value	Element and value rules
5: M	Measurement. measurementDescriptionListData. measurementDescriptionData.		
5: M	measurementId	< m7#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
5: M	measurementType	“frequency”	
5: M	commodityType	“electricity”	
5: M	unit	“Hz”	
5: M	scopeType	“acFrequency”	

measurementConstraintsListData reply

Table 23 — Content of measurementConstraintsListData reply or notify at Actor Device – Frequency (for “Monitoring of Power Consumption”)

Scenario [...]: M/R/O [WJ][C]	Element	Value	Element and value rules
5: R	Measurement. measurementConstraintsListData. measurementConstraintsData.		
5: M	measurementId	< m7#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
5: R	valueRangeMin.		SHOULD be used. [Scaled number rules]
5: M	valueRangeMin. number		SHALL be used.
5: O	valueRangeMin. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
5: R	valueRangeMax.		SHOULD be used. [Scaled number rules]
5: M	valueRangeMax. number		SHALL be used.
5: O	valueRangeMax. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.
5: R	valueStepSize.		SHOULD be used. [Scaled number rules]
5: M	valueStepSize. number		SHALL be used.
5: O	valueStepSize. scale		“SHALL be interpreted”. If absent, a default value of “0” applies.

measurementListData reply

Table 24 — Content of measurementListData reply or notify at Actor Device – Frequency (for “Monitoring of Power Consumption”)

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
5: M	Measurement. measurementListData. measurementData.		
5: M	measurementId	< m6#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
5: M	valueType	“value”	SHALL be set as SUB IDENTIFIER.
5: O	timestamp	< t#(1..1)- > m7#(1..1) >	MAY be used. Only the newest measurement value SHALL be stated. Additional historical values are forbidden.
5: M	value.		[Measurement value rules] [Scaled number rules]
5: M	value. number		SHALL be used.
5: O	value. scale		SHALL be interpreted. If absent, a default value of “0” applies.
5: M	valueSource	“measuredValue”	
		“calculatedValue”	
		“empiricalValue”	
5: R	valueState		[Value state rules]

5.2.2.2.3 Request Electrical Connection Data

5.2.2.2.3.1 General

This is a set of two SPINE partial “read” messages. As some of these messages require input from previous requests, they shall be sent in sequence, as shown in Figure 10.

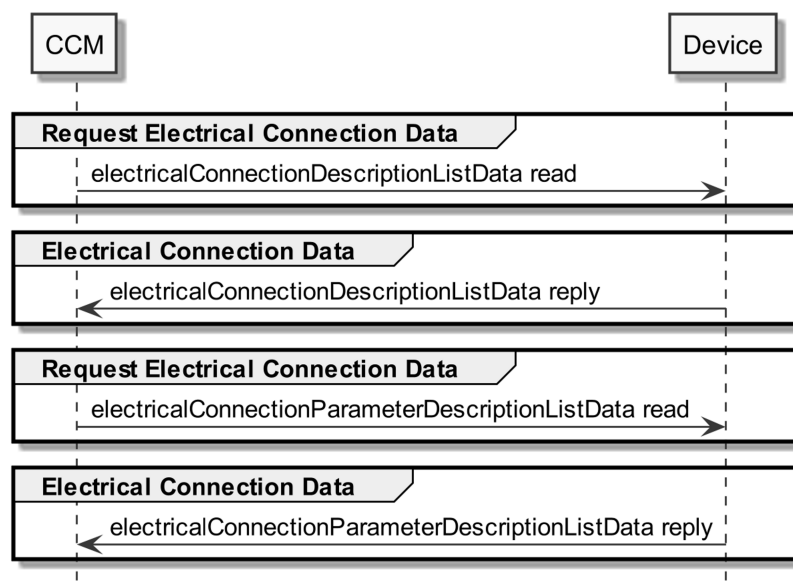


Figure 10 — UCF_AC_Measurement: Request Electrical Connection Data/Electrical Connection Data communication sequence diagram

The messages have the following payload:

5.2.2.2.3.2 electricalConnectionParameterDescriptionListData read

Table 25 — Content of electricalConnectionParameterListData read at Actor CCM

Element		Value
function		electricalConnectionParameterDescriptionListData
filter		
	cmdControl	partial
	electricalConnectionParameterDescriptionListDataSelectors	
	measurementId	measurementId (derived from the measurementDescriptionListData reply)

5.2.2.2.3.3 electricalConnectionDescriptionListData read

Table 26 — Content of electricalConnectionDescriptionListData read at Actor CCM

Element		Value
function		electricalConnectionDescriptionListData
filter		
	cmdControl	partial
	electricalConnectionDescriptionListDataSelectors	
	electricalConnectionId	electricalConnectionId (derived from the electricalConnectionParameterDescriptionListData reply)

These messages are sent during the initial communication phase, as well as on (re-)connect.

5.2.2.2.4 Electrical Connection Data

5.2.2.2.4.1 General

For each read from “Request Electrical Connection Data” above, there is a SPINE “reply” with the following payload:

5.2.2.2.4.2 for Power, Energy, Voltage, Current, Frequency

electricalConnectionDescriptionListData reply (for Power, Energy, Voltage, Current, Frequency)

Table 27 — Content of electricalConnectionDescriptionListData reply or notify at Actor Device (for “Monitoring of Power Consumption”)

M/R/O [W][C]	Element	Value
1–5: M	ElectricalConnection. electricalConnectionDescriptionData.	electricalConnectionDescriptionListData.
1–5: M	electricalConnectionId	< ec1#(1..1) >
1–5: M	powerSupplyType	“ac”
1–5: M	positiveEnergyDirection	“consume”

5.2.2.2.4.3 For Power

electricalConnectionParameterDescriptionListData reply

non-phase specific:

Table 28 — Content of electricalConnectionParameterDescriptionListData reply or notify at Actor Device – Power, non-phase specific

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
1: M	ElectricalConnection. electricalConnectionParameterDescriptionListData. electricalConnectionParameterDescriptionData.		
1: M	electricalConnectionId	< ec1#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
1: M	parameterId	< p1#(1..1)- > ec1#1 >	SHALL be set as SUB IDENTIFIER.
1: M	measurementId	< m1#1- > p1#1 >	SHALL be set as FOREIGN IDENTIFIER.
1: M	voltageType	“ac”	

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
1: O	acMeasuredPhases	"abc" "ab" "bc" "ac" "a" "b" "c" (- > p1#1)	If the Monitored Unit is connected to less than three phases, one of the other combinations like "a" or "ab" are allowed instead of "abc". The values "a", "b", and "c" are permitted if and only if only one phase is connected to the Monitored Unit.
1: O	acMeasuredInReferenceTo	"neutral"	
1: M	acMeasurementType	"real"	
1: O	acMeasurementVariant	"rms"	

phase-specific:

Table 29 — Content of electricalConnectionParameterDescriptionListData reply or notify at Actor Device – Power, phase-specific

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
1: M	ElectricalConnection. electricalConnectionParameterDescriptionListData. electricalConnectionParameterDescriptionData.		
1: M	electricalConnectionId	< ec1#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
1: M	parameterId	< p2#(1..3)- > ec1#1 >	SHALL be set as SUB IDENTIFIER.
1: M	measurementId	< m2#(1..3)- > p2#(1..3) >	SHALL be set as FOREIGN IDENTIFIER.
1: M	voltageType	"ac"	
1: M	acMeasuredPhases	"a" (- > p2#1)	
		"b" (- > p2#2)	
		"c" (- > p2#3)	
1: M	acMeasuredInReferenceTo	"neutral"	
1: M	acMeasurementType	"real"	
1: M	acMeasurementVariant	"rms"	

5.2.2.2.4.4 For Energy

electricalConnectionParameterDescriptionListData reply

Energy Consumed:

Table 30 — Content of electricalConnectionParameterDescriptionListData reply or notify at Actor Device – Energy consumed

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
2: M	ElectricalConnection. electricalConnectionParameterDescriptionListData. electricalConnectionParameterDescriptionData.		
2: M	electricalConnectionId	< ec1#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
2: M	parameterId	< p3#(1..1)- > ec1#1 >	SHALL be set as SUB IDENTIFIER.
2: M	measurementId	< m3#1- > p3#1 >	SHALL be set as FOREIGN IDENTIFIER.
2: M	voltageType	“ac”	
2: M	acMeasurementType	“real”	

5.2.2.2.4.5 For Current

electricalConnectionParameterDescriptionListData reply

Table 31 — Content of electricalConnectionParameterDescriptionListData reply or notify at Actor Device – Current

Scenario [...]: M/R/O [W][C]	Element	Value	Element and value rules
3: M	ElectricalConnection. electricalConnectionParameterDescriptionListData. electricalConnectionParameterDescriptionData.		
3: M	electricalConnectionId	< ec1#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
3: M	parameterId	< p5#(1..3)- > ec1#1 >	SHALL be set as SUB IDENTIFIER.
3: M	measurementId	< m5#(1..3)- > p5#(1..3) >	SHALL be set as FOREIGN IDENTIFIER.
3: M	voltageType	“ac”	

Scenario [...] : M/R/O [W][C]	Element	Value	Element and value rules
3: M	acMeasuredPhases ^a	"a" (- > < p5#1 >)	
		"b" (- > < p5#2 >)	
		"c" (- > < p5#3 >)	
3: M	acMeasurementType	"real"	
3: M	acMeasurementVariant	"rms"	
^a Only the connected phases SHALL be delivered. If only one phase is connected, only one of the values is provided by the Monitored Unit. This is not necessarily phase A, but could also be phase B or phase C.			

5.2.2.2.4.6 For Voltage

electricalConnectionParameterDescriptionListData reply

Table 32 — Content of electricalConnectionParameterDescriptionListData reply or notify at Actor Device – Voltage

Scenario [...] : M/R/O [W][C]	Element	Value	Element and value rules
4: M	ElectricalConnection. electricalConnectionParameterDescriptionListData. electricalConnectionParameterDescriptionData.		
4: M	electricalConnectionId	< ec1#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
4: M	parameterId	< p6#(1..6)- > ec1#1 >	SHALL be set as SUB IDENTIFIER.
4: M	measurementId	< m6#(1..6)- > p6#(1..6) >	SHALL be set as FOREIGN IDENTIFIER.
4: M	voltageType	"ac"	
4: M	acMeasuredPhases	"a" (- > < p6#1 >)	
		"a" (- > < p6#4 >)	
		"b" (- > < p6#2 >)	
		"b" (- > < p6#5 >)	
		"c" (- > < p6#3 >)	
		"c" (- > < p6#6 >)	

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
4: M	acMeasuredInReferenceTo	"a" (- > < p6#6 >)	
		"b" (- > < p6#4 >)	
		"c" (- > < p6#5 >)	
		"neutral" (- > < p6#1 >)	
		"neutral" (- > < p6#2 >)	
		"neutral" (- > < p6#3 >)	
4: M	acMeasurementType	"apparent"	
4: M	acMeasurementVariant	"rms"	

5.2.2.2.4.7 For Frequency

electricalConnectionParameterDescriptionListData reply

Table 33 — Content of electricalConnectionParameterDescriptionListData reply or notify at Actor Device – Frequency

Scenario [{...}]: M/R/O [W][C]	Element	Value	Element and value rules
5: M	ElectricalConnection. electricalConnectionParameterDescriptionListData. electricalConnectionParameterDescriptionData.		
5: M	electricalConnectionId	< ec1#(1..1) >	SHALL be set as PRIMARY IDENTIFIER.
5: M	parameterId	< p7#(1..1)- > ec1#(1..1) >	SHALL be set as SUB IDENTIFIER.
5: M	measurementId	< m7#(1..1)- > p7#(1..1) >	SHALL be set as FOREIGN IDENTIFIER.
5: M	voltageType	"ac"	

"Notify" messages as described above may be sent to subscribers by the Smart Appliance as a result of the changes.

5.2.2.2.5 Pre-Scenario communication for Use Case "Monitoring of Power Consumption" (all scenarios)

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 5.2.1.2.2.