Smart Electric Power Alliance

Application Guide: Interoperability Profile for Electric Vehicle Fleet Managed Charging

Utilizing IEEE 2030.5-2018

February 2025

This work was performed under the financial assistance award 70NANB20H138 from the U.S. Department of Commerce, National Institute of Standards and Technology.

Application Guide: Interoperability Profile for Electric Vehicle Fleet Managed Charging

Table of Contents

Executive Summary 4

Introduction 5

Architectural Overview 6

- Scenario 1: Direct Exchange 6
- Scenario 2: Aggregator-Mediated 8

IEEE 2030.5-2018 10

- End Device Function Set 10
- Function Set Assignments Function Set 11
- Subscription/Notification Function Set 11
- Response Function Set 12
- Distributed Energy Resources Function Set 12
- Pricing Function Set 13
- Metering Mirror Function Set 13
- Objects, Resources, Lists, and Their Attributes 13

Scenarios 1 and 2 Step-by-Step Exchanges in IEEE 2030.5 24

- Scenario 1: Direct Exchange 24
- Scenario 2: Aggregator-Mediated 27

Conclusion 30

List of Figures

Figure 1. Scenario 1—Direct Exchange 7

Figure 2. Scenario 2—Aggregator-Mediated 8

Figure 3. Relationships Between List, Resources, Objects 22

Figure 4. Relation Between List, Resources, Objects for Metering and Pricing Function Sets 23

Figure 5. Subscription Process 25

Figure 6. Activity Diagram for Scenario 1 26

Figure 7. Pricing Process 28

Figure 8. Activity Diagram for Scenario 2 29

List of Tables

Table 1: Acronyms and Abbreviations 5

Table 2. Terminology 5

Table 3. Step-by-Step Analysis of Scenario 1 7

Table 4. Step-by-Step Analysis of Scenario 2 9

Table 5. Event Object Attributes 13

Table 6. EventStatus Object Attributes 14

Table 7. Respondable Resource Attributes 14

Table 8. Respondable Subscribable Identified Object Attributes 14

Table 9. EndDevice Function Set Resources and Their Attributes 15

Table 10. Attributes of Function Set Assignments 16

Table 11. Response Function Set Resources and Their Attributes 16
Table 12. DER Function Set Lists, Resources, and Their Attributes 17
Table 13. Device Capability Object 18
Table 14. Pricing Function Set Attributes 19
Table 15. Metering Mirror Function Set Attributes 21

Copyright

© Smart Electric Power Alliance, 2025. All rights reserved
This material may not be published, reproduced, broadcast, rewritten, or redistributed without permission.

Authors

Aaron Smallwood, Vice President, Industry Solutions, Smart Electric Power Alliance

Daisy Chung, Technical Program Manager, Smart Electric Power Alliance

Cuong Nguyen, Lead, Smart Grid Testing and Certification, National Institute of Standards and Technology

Sean Morash, Principal Consultant, EnerNex

Aaron Snyder, Director of Grid Technology Consulting, EnerNex

Phillip Court, Product and Company Strategy, Ecogy Energy

Gustavo Alvarez, Founder and Principal Consultant, Green Planet Lab

James Mater, General Manager, Smart Grid, QualityLogic, Inc.

Jack Lacy, Senior Vice President Standards and Community Initiatives, InterTrust Technologies

John Gorman, Product Lead, Ecogy Energy

Rose Kallio, Solution Architect, Schneider Electric

Cedric Bassil, Consultant, EnerNex

Disclaimer

All content, including, without limitation, any documents provided on or linked to the SEPA website is provided "as is" and may contain errors or misprints. SEPA and the companies who contribute content to the website and to SEPA publications ("contributing companies") make no warranties, representations or conditions of any kind, express or implied, including, but not limited to any warranty of title or ownership, of merchantability, of fitness for a particular purpose or use, or against infringement, with respect to the content of this website or any SEPA publications. SEPA and the contributing companies make no representations, warranties, or guarantees, or conditions as to the quality, suitability, truth, or accuracy, or completeness of any materials contained on the website.

About SEPA

The Smart Electric Power Alliance (SEPA) helps all electric power stakeholders accelerate the transformation to a carbon free electricity system. SEPA concentrates our focus on the following areas to maximize impact: Transportation, Storage, Resilience, Emerging Technology, Policy, and Energy Equity.

SEPA delivers value to our members through research, education, events, working groups, peer engagements, and member projects. We facilitate collaboration, develop innovative strategies and guidance for regulatory and business innovation, and provide actionable solutions for our members and partner organizations. For more information, visit www.sepapower.org.

Executive Summary

Roughly 20% of America's greenhouse gas emissions are produced by cars and trucks, all of which must be eliminated to achieve the federal target of net-zero emissions by 2050. Not only can electric vehicles reduce greenhouse gas emissions, but they can also play a significant role in demand management programs. These twin goals of decarbonizing transportation while meeting and managing increasing electrical demand has created the need to develop a reliable and widely available electric vehicle charging infrastructure.

Numerous proposed smart grid standards for EV integration and charging infrastructure have emerged. However, these proposed standards typically include several mandatory and optional functions and are technically complex, which makes their implementation difficult. As customer expectations mature, an application guide specifically describing the implementation of a particular use case utilizing a particular standard will prove useful to a broad stakeholder community.

The goal of this Application Guide is to a map the requirements from a use case to a standard, and then to execute an interoperability profile. The report describes the required procedures to follow to perform a use case. The employed standard is IEEE 2030.5-2018, an available open smart grid standard. The use case addresses the information exchanges needed between a distribution system operator (DSO) and EV charging systems (charging station operator, charging station management system, and EV supply equipment ["EVSE"]) to execute demand management or demand response programs. The use case primarily covers DSO and charging station management system (system-to-system) data exchanges, and only calls for charging station operator and EV fleet operator intervention when necessary.

Introduction

This Application Guide ("Guide") describes how to use an open industry standard communication protocol, IEEE Std 2030.5™-2018 ("IEEE 2030.5"), to implement the Electric Vehicle Fleet Managed Charging Use Case ("Use Case") developed by the SEPA Interoperability Profile Task Force, a subgroup of the SEPA Testing and Certification Working Group. This Guide is built upon the Use Case and the *SEPA interoperability profile for EV Fleet Managed Charging Utilizing IEEE 2030.5*.

The EV Fleet Managed Charging Use Case outlines the information exchanges between Electric Vehicle (EV) charging station management systems (CSMSs), distribution system operators (DSOs), and any human in the loop required to ensure appropriate action(s) to fulfill the requirements of the Use Case, which includes EVSE participation in DSO demand management events. DSOs and EV charging systems managed by charge station operators (CSOs) via CSMS and EVSE need to exchange information to enable charging of EVs as well as EVSE participation in DSO demand management programs. The use case addresses two of the multiple potential implementation scenarios for fleet-managed charging operations: 1) direct CSMS and DSO bidirectional information exchanges and 2) CSMS and DSO information exchanges via an aggregator.

Table 1: Acronyms and Abbreviations

Abbreviations	Full Term
EV	Electric Vehicle
CSMS	Charging Station Management System
EVSE	Electric Vehicle Supply Equipment
CSO	Charge Station Operator
DSO	Distribution System Operator

Source: SEPA (2025).

Table 2. Terminology

Term	Description
Electric Vehicle	An electric-powered vehicle that uses one or more electric motors for propulsion and does not use an internal combustion engine. For the purposes of this guide, EVs are considered plug-in electric vehicles that are primarily charged through electrical sources external to the EV, such as grid power.
Electric Vehicle Supply Equipment	An independently operated and managed part of the charging station that can deliver energy to or receive energy from one or more EVs.
Charging Station Management System	Manages EVSEs. The CSMS transforms EVSEs from electrical supply equipment to smart grid actors. The CSMS has the information for authorizing users to use charging stations and monitors charging processes, gives charging commands (smart charging), and integrates information from other sources (e.g., CSO, DSO, Aggregator, EV Fleet Operator). The CSMS may be operated by the CSO directly or by Aggregators and may be separate from or integrated with EVSE.
Charge Station Operator	The person or organization tasked with operating the charging station, which includes maintenance and coordinating with the DSO and EV Fleet Operator to meet the charging needs of the EVs while optionally participating in DSO programs. The CSO may be a homeowner with a single EV.
Distribution System Operator	The DSO is the legal entity responsible for operating, ensuring the maintenance of, and, if necessary, developing the distribution system in a defined area and, where applicable, its interconnections with other systems and for ensuring the ability of the system to meet demands for the distribution of electricity. The DSO has many subsystems and further actors that could be spelled out, such as meters and management systems, but is considered at a high level for the purposes of this use case.

Source: SEPA (2025).

This Guide describes the step-by-step instructions to implement both Scenario 1 (direct DSO to CSMS connectivity) and Scenario 2 (DSO to Aggregator connectivity) of the use case in terms of IEEE 2030.5.

Note further that the key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this Guide are to be interpreted as described in RFC 2119.[1]

Architectural Overview

The Use Case, which is the basis of this work, addresses the data exchanges between the DSO and an Aggregator and between the DSO and a CSMS directly. The information exchanges between the DSO and charging stations have been modeled on two scenarios, the primary difference between the scenarios being the addition of an aggregator in the second scenario. The two scenarios are as follows:

- **Scenario 1:** Direct DSO and CSMS bidirectional information exchange ("Direct Exchange").
- **Scenario 2:** DSO to Aggregator, Aggregator to CSMS information exchange ("Aggregator Mediated")

In both cases, the information exchanges are governed by regulatory and DSO requirements. The defined information exchanges within the Use Case allow for multiple opt-outs throughout the process that end the need for continued information exchange.

Scenario 1: Direct Exchange

In the direct DSO and CSMS bidirectional information exchange scenario, the CSMS may take multiple forms. The CSMS may control multiple EVSEs at multiple facilities, or it may be integrated within an EVSE at a residential home. The Use Case establishes the precondition that the CSMS is already "working with DSO input," which for the purposes of this profile will mean that the CSMS:

- Has been discovered within the DSO's established communication network.
- Has implemented appropriate cybersecurity measures consistent with DSO requirements.
- Manages the charging of EVSEs that are interconnected to the DSO grid at known and stationary locations.

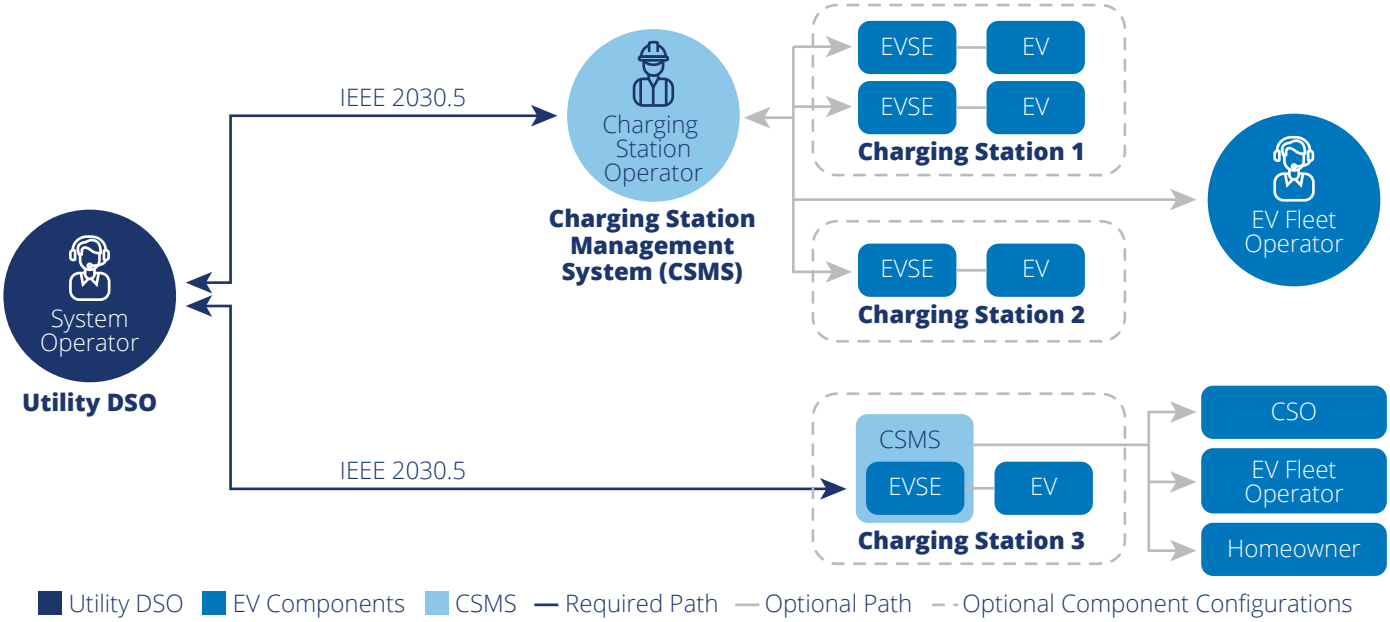
This scenario model is represented in Figure 1.

The DSO provides the CSMS with information about historic usage and future demand management events. The CSMS may elicit opt-outs from the EV fleet operator and the charge station operator. However, the mechanisms and data exchanges required for those opt-outs are considered vendor-proprietary.

Table 3 details the steps of a demand management event from start to finish for Scenario 1.

1 Request for Comment 2119.

Figure 1. Scenario 1—Direct Exchange



Source: SEPA (2022).

Table 3. Step-by-Step Analysis of Scenario 1

Step	Event or Name of Process/Activity	Description	Info Producer (Actor)	Info Receiver (Actor)	Information Exchanged
1	DSO identifies need for demand management.	DSO uses available information (including EV charging profiles) to identify need for demand management on specific substation(s) or circuit(s).	DSO	DSO	DSO internal
2	DSO informs CSMS of demand management opportunity.	DSO informs the appropriate CSMS of demand management.	DSO	CSMS	Start Time, Duration, Set Point
3	(Optional) CSMS solicits human authorization of opt-in.	(Optional) CSMS informs CSO and EV fleet operator of demand management particulars, providing an opportunity for participation override. Note that the CSMS will have the logic to skip this step and automatically opt in or out depending on established rules configured by the CSO and/or EV fleet operator.	CSMS	EV Fleet Operator / CSO	Start Time, Duration, Set Point
4	CSMS opts into Demand Management.	CSMS informs DSO of intent to participate in the opportunity as specified.	CSMS	DSO	Opt-in notification
5	DSO dispatches Demand Management.	DSO dispatches demand management via System Operator.	DSO	CSMS	Start Time, Duration, Set Point
6	CSMS controls EVSE.	Based on demand management commands, CSMS controls EVSE.	CSMS	EVSE	Start Time, Duration, Set Point
7	CSMS confirms performance.	CSMS confirms that it attempted to meet demand management need and EVSE command was successful.	CSMS	DSO	Confirmation of participation

Source: SEPA (2025).

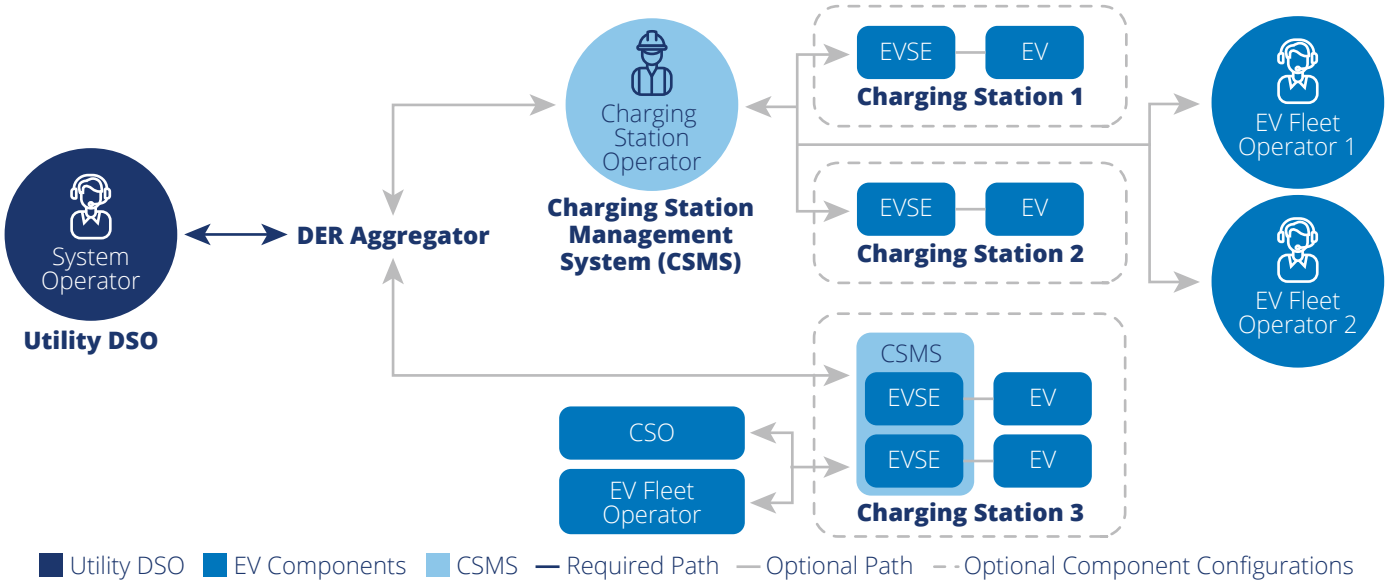
Scenario 2: Aggregator-Mediated

Scenario 2 includes the DSO interacting with a DER Aggregator to meet the needs of the distribution system. Within this profile, the DER Aggregator's EV Fleet is of the most interest, but the Aggregator may have additional DER that could be combined to meet the needs of the DSO in the most cost-effective manner. The aggregator manages the information exchange on behalf of the CSMS (CSO) and communicates the results of the information exchanges between the DSO and the CSMS.

This scenario model is represented in Figure 2.

Table 4 details the steps of a demand management event from start to finish for Scenario 2.

Figure 2. Scenario 2—Aggregator-Mediated



Source: SEPA (2022).

Table 4. Step-by-Step Analysis of Scenario 2

Step	Event or Name of Process/Activity	Description	Info Producer (Actor)	Info Receiver (Actor)	Information Exchanged
1	DSO identifies need for demand management.	DSO uses available information (including EV charging profiles) to identify need for demand management on specific substation(s) or circuit(s).	DSO	DSO	DSO internal
2	DSO informs Aggregator of demand management opportunity.	DSO informs the appropriate Aggregator of demand management.	DSO	Aggregator	Start Time, Duration, Set Point
3	Aggregator informs CSMS of demand management opportunity.	Aggregator informs CSMS of demand management opportunity. CSO has access to CSMS.	Aggregator	CSMS	Start Time, Duration, Set Point
4	(Optional) CSMS solicits human authorization of opt-in.	(Optional) CSMS informs CSO and EV fleet operator of demand management particulars, providing an opportunity for participation override. Note that the CSMS will have the logic to skip this step and automatically opt in or out depending on established rules configured by the CSO and/or EV fleet operator.	CSMS	EV Fleet Operator / CSO	Start Time, Duration, Set Point
5	CSMS opts into Demand Management.	CSMS informs Aggregator of intent to participate in the opportunity as specified.	CSMS	Aggregator	Opt-in notification
6	Aggregator opts into Demand Management.	CSMS informs DSO of intent to participate in the opportunity as specified	Aggregator	DSO	Opt-in notification
7	DSO dispatches Demand Management.	DSO dispatches demand management via System Operator	DSO	Aggregator	Start Time, Duration, Set Point
8	Aggregator dispatches EVSE through CSMS.	Aggregator dispatches EVSE demand management through CSMS.	Aggregator	CSMS	Start Time, Duration, Set Point
9	CSMS controls EVSE.	Based on demand management commands, CSMS controls EVSE.	CSMS	EVSE	Start Time, Duration, Set Point
10	CSMS confirms performance.	CSMS confirms that it attempted to meet demand management need and EVSE command was successful.	CSMS	Aggregator	Confirmation of participation
11	Aggregator confirms performance.	Aggregator confirms performance.	Aggregator	CSMS	Confirmation of participation
12	DSO measures performance & calculates payments.	DSO subsystems perform demand management measurement and validation.	DSO	DSO	EVSE energy usage across historic and demand management time periods

Source: SEPA (2025).

IEEE 2030.5-2018

As described by the IEEE Standards Association, the IEEE 2030.5 protocol defines "the application layer with TCP/IP providing functions in the transport and Internet layers to enable utility management of the end user energy environment, including demand response, load control, time of day pricing, management of distributed generation, electric vehicles, etc."[2] In other words, IEEE 2030.5 defines a framework to support communications between the smart grid and the consumers in a secure way. IEEE 2030.5 is based on the REST (Representation State Transfer) architecture.[3] REST is an architectural style for implementing web services in a stateless manner to promote simplicity and interoperability. REST uses Client-Server architecture, where Servers host resources and Clients manipulate resources on Servers. Servers host a collection of resources. A resource can be thought of as an object that has an address (URL). Each resource has its own identifier called a Uniform Resource Identifier (URI). The Client initiates most interactions. The Server passively waits for clients to manipulate its resources. Clients interact with Server resources using four HTTP operations: GET, PUT, POST, DELETE.

- **GET** is used to read the resource. The Server returns the resource for the given URI. GET only fetches the resource and does not modify it in any way.
- **POST** is used to transfer data to the Server. Thus, POST is usually associated with creating a resource. After the resource is created, the Server should respond with its URI.
- **PUT** is used to update the existing resource. It overrides the whole resource with transferred data. PUT points to an existing resource; POST creates a new one.
- The final basic operation is **DELETE**. As the name suggests, DELETE is used to remove an existing resource from the Server.

An Event refers to any instance of a resource with a defined valid duration for which a Client should act. Many resources within this standard are derived from the <List> object. These resources will collectively be referred to as list resources.

The 2030.5 protocol is organized into multiple function sets. Of these, the following function sets are relevant to the Use Case.

End Device Function Set

An End Device is a Client Device recognized by the Server. End Device resources allow the Server to manage individual Clients. The Server can send controls to individual Clients and can receive information from individual Clients. The Server represents a Client device by creating an End Device instance resource for the device. This may include both general information about a Client (e.g., spatial frequency domain imaging (SFDI), software version numbers) and information specific to the relationship between a Client and the Server where the End Device resource is hosted (e.g., subscriptions, Function Set Assignments, registration). The major resources are EndDeviceList and EndDevice, where:

- **EndDeviceList** provides the list of all EndDevice resources hosted by the given Server.
- **EndDevice** resources may also be created on a POST to the End Device List.

General Guidelines

- Servers SHOULD delete or prevent duplicate EndDevice instances for the same Client from appearing in its EndDeviceList.
- Servers MAY remove records after 72 hours of no interaction from the corresponding Client. It is RECOMMENDED that Clients interact with the Server at least once every 48 hours.
- Servers MAY remove EndDevice instances and/or their subordinate resources at any time, for any reason. Servers SHALL NOT remove EndDevice instances before marking them as disabled (using the enabled attribute) and allowing time for Clients to query this change.

2 IEEE (2018). Standard for Smart Energy Profile Application Protocol.

3 Roy Thomas Fielding (2000). Architectural Styles and the Design of Network-based Architecture.

Function Set Assignments Function Set

Function Set Assignments ("FSAs") are the set of functions by which a Server assigns a Client to one or more groups. A Server assigns a Client to a set of groups by providing the Client Function Set Assignments via the EndDevice Resource.

The Function Set Assignments function set defines collections of references to function set instances. These collections are used by the EndDevice resource to indicate to devices the resources that are to be used for specific purposes. For example, a service provider may wish that all the participants of a particular program use a given DER function set instance, Time resource, and a Pricing function set instance. A FunctionSetAssignments resource contains references to the function set instances that are to be used by the assigned device(s).

General Guidelines

- Clients that act on events SHALL determine if they are assigned into Function Set Assignments.
- Clients MAY be assigned multiple Function Set Assignments.
- Multiple Clients MAY be assigned the same Function Set Assignment.
- If a Server supports Function Set Assignments, it SHALL support a minimum of one Function Set Assignment for each device registered to the Server.
- Clients SHALL support at least six function set instances assigned through one or more Function Set Assignments.
- Clients SHALL identify which Function Set Assignments apply to them by querying the FunctionSetAssignments resource within their End Device function set instance.
- Clients SHALL periodically poll their group assignments under their EndDevice resource and the corresponding Function Set Assignments resource, or they SHALL subscribe to them to monitor for changes. Client devices that do not subscribe SHALL query at least once every 24 hours but SHALL NOT query more than once per hour.

Subscription/Notification Function Set

This mechanism may be useful when a Client wishes to quickly learn of changes to a resource on a Server.

A Subscription resource is realized as a resource for an individual Client, providing interfaces to all Subscriptions for the given Client. Resources that support Subscriptions indicate that via the subscribable attribute. Some resources can have conditional Subscriptions.

The Notification resource is used to receive Notifications that a resource to which a host is subscribed has changed. The location of the Notification resource is passed to the Subscription Server in the body of the Subscription. As such, a given Client (Notification resource Server) may have one Notification resource for multiple different Notifications or may have a different Notification resource for different Notifications. The resource representation returned in a Notification SHALL be identical to that which would be returned via a GET request to the resource.

General Guidelines

Servers MAY remove Subscriptions at any time.

- Clients SHOULD send Subscription renewals every 24 hours, and no more often than every 1 hour to sustain a Subscription.
- Servers SHOULD remove Subscriptions if a Client has not renewed in 36 hours.
- Subscriptions SHALL be maintained on Servers through power loss.
- Servers SHALL use the Subscription parameters from the latest Subscription renewal.
- Clients SHOULD check the status of their Subscriptions after perceived loss of connectivity.
- Following recovery from a perceived loss of connectivity, Clients SHOULD poll their resources of interest (including those to which they are subscribed) in case those resources changed during the loss of connectivity.
- If the URI of a resource changes, then Subscriptions to that resource SHALL be terminated.
- To prevent overwhelming network resources, Notifications SHOULD be sent to a given Client for a given resource no more than once every 30 seconds.
- Servers implementing the Subscription resource SHALL be capable of maintaining a minimum of one Subscription and Servers implementing the Subscription resource SHOULD support at least one Subscription per device per subscribable resource.

- If a Server implementing the Subscription resource is unable to accept a Subscription, the Server SHALL return an error resource representation indicating the specific error (e.g., element does not support conditional Subscription) with an HTTP response code of 400.
- If a Client receives a Notification for an undesired Subscription, the Client SHALL return an HTTP 400 error. Upon receipt of such an error, the Server SHALL remove the Subscription without Notification.

Response Function Set

This function set provides an interface for capturing responses from all events, including demand response and load control (DRLC), distributed energy resources (DERs), pricing, and messaging.

The originating event will indicate if a response is required. The destination of a Response is controlled by the replyTo attribute of the originating event.

General Guidelines

- Clients of any function set that can utilize the Response function set (e.g., DRLC, DER, Price, Messaging, Flow Reservation) SHALL implement a Response function set Client.
- The typical use for this function set is that end devices will POST their responses to a Response resource identified in events. It is not expected that devices will use discovery to locate these resources.
- If a response to an event is desired, then the event SHALL provide, in the replyTo field, a URI indicating the location of where the responses are to be posted.
- The Client SHALL POST the responses to the indicated URI based on the rules indicated by the responseRequired bitfield of the event.
- If a Server hosts events that specify a response is required, then that Server is NOT REQUIRED to host the response Server.

Distributed Energy Resources Function Set

This function set provides an interface to manage distributed energy resources (DERs). There are two main types of Client devices of this function set: generation and storage. Examples of generation devices include fuel cells, intelligent solar inverters, and backup generation units. Examples of storage devices include battery storage systems and electric vehicles.

Servers host one or more DERPrograms (Resource), which in turn expose DERControl events to DER Clients. DERControl instances contain attributes that allow DER Clients to respond to events that are explicitly targeted at specific kinds of devices. A DERControl instance also includes scheduling attributes that allow DER Clients to store and process future events.

A DER Client will typically discover its associated DERProgramList through Function Set Assignments.

A DERProgram exposes control parameters to a DER Client via DERControl Events.

General Guidelines

- DERProgram Server devices SHALL be capable of internally storing and supporting at least one DERProgram instance.
- DERProgram Server devices SHOULD be capable of internally storing and supporting three unique DERProgram instances.
- DER Client devices SHALL be capable of internally storing and supporting at least one DERProgram instance.
- DER Client devices SHOULD be capable of internally storing and supporting two unique DERProgram instances.
- At any point in time a DER Client is managed by a single DERControl Event, which SHALL supersede any previous Event.
- DERProgram Server devices SHALL be capable of internally storing and supporting at least five unique DERControl instances, which MAY be distributed among multiple programs.
- DERProgram Server devices SHOULD be capable of internally storing and supporting a total of 10 unique DERControl instances.
- Each DERProgram SHALL internally store a single DefaultDERControl instance that defines the default behavior of associated DER Clients when no applicable active Events are available.

- DER Client devices SHALL be capable of internally storing and supporting at least one DERControl instance and a single DefaultDERControl instance for each stored DERProgram.
- DER Client devices SHOULD be capable of internally storing and supporting three DERControl instances.

Pricing Function Set

The Pricing function set provides the tariff structures communicated by the Server and is designed to support a variety of tariff types, including flat-rate pricing, time-of-use tiers, hourly day-ahead pricing, and real-time pricing.

The Pricing function set can be paired with the Metering Mirror function set.

Metering Mirror Function Set

The Metering Mirror function set reports meter data such as real power, apparent power, reactive power, power factor, voltage, and current. Metering data is PUT/POSTed to the Utility Server via the MirrorUsagePointListLink in the End Device's FSA group. The Utility Server enables/disables the PUT/POSTing of meter data by including or not including the MirrorUsagePointListLink.

General Guidelines

- The Metering Mirror Server SHOULD only accept POSTs to a given MirrorUsagePoint from the Client that created the mirror.
- If a Client POSTs more data than the Metering Mirror Server is willing to accept, the Server SHALL respond with a response code of 413 (Request Entity Too Large).
- The Metering Mirror Server MAY decide when to remove data from the related UsagePoint resource structure.
- When a MirrorUsagePoint is deleted, the associated UsagePoint SHALL also be deleted.

Objects, Resources, Lists, and Their Attributes

IEEE 2030.5 contains objects that are used by function sets. The following tables show the objects, resources, and attributes that are used by the function sets described above to implement the Use Case.

Table 5. Event Object Attributes

Attribute	Example	Description
creationtime	1234556	The time at which the Event was created.
Interval	<duration>86400</duration> <start>1341446400</start>	The period during which the Event applies.

Source: SEPA (2025).

Table 6. EventStatus Object Attributes

Attribute	Example	Description
Currentstatus	0	0 = Scheduled 1 = Active 2 = Cancelled 3 = Cancelled with Randomization 4 = Superseded
dateTime	1234960	The dateTime attribute will provide a timestamp of when the current status was defined. dateTime MUST be set to the time at which the status change occurred.
potentiallySuperseded	false	Set to true by a Server of this event if there are events that overlap this event in time and also overlap in some, but not all, deviceCategory's (if applicable) and DERControl controls (e.g., opModTargetW) (if applicable) in the same function set instance.
potentiallySuperseded Time		Indicates the time that the potentiallySuperseded flag was set.
Reason	Overload	The Reason attribute allows a Service provider to provide a textual explanation of the status.

Source: SEPA (2025).

Table 7. Respondable Resource Attributes

Attribute	Example	Description
responseRequired	1	Indicates whether or not a response is required upon receipt, creation, or update of this resource. Responses shall be posted to the collection specified in "replyTo." 0 = End device shall indicate that message was received 1 = End device shall indicate specific response 2 = End user/customer response is required
replyTo		A reference to the response resource address (URI). Required on a response to a GET if responseRequired is "true."

Source: SEPA (2025).

Table 8. Respondable Subscribable Identified Object Attributes

Attribute	Examples	Description
description	Operation X	The description is a human-readable text describing or naming the object.
mRID	0ED30F5A0000	The global identifier of the object.
Version		Contains the version number of the object.
Subscribable	Subscribable = 1 Not Subscribable = 0	Indicates whether or not subscriptions are supported for this resource, as well as whether conditional (thresholds) are supported. If not specified, is "not subscribable."

Source: SEPA (2025).

Table 9. EndDevice Function Set Resources and Their Attributes		
Attribute	**Example**	**Description**
EndDeviceList (Resource)		Is a subscribable list and provides the list of all EndDevice resources hosted by the given Server. It contains the following attributes:
pollRate	900	The default polling rate for this function set in seconds. If not specified, a default of 900 seconds is used. It is RECOMMENDED that a Client poll the resources of this function set every pollRate seconds.
*****EndDeviceList attributes end here*****		
EndDevice (Resource)		Asset container that performs one or more end device functions. Contains information about individual devices in the network. Contains the following attributes:
ChangedTime	1379905200	The time at which this resource was last modified or created.
Enabled	True	This attribute indicates whether an EndDevice is enabled, or registered, on the Server. If a Server set this attribute to false, the device is no longer registered.
postRate		How often EndDevice and subordinate resources should be POSTed, in seconds. A Client MAY indicate a preferred postRate when POSTing EndDevice. A Server MAY add or modify postRate to indicate its preferred posting rate.
*****EndDevice attributes end here*****		
AbstractDevice (Resource)		Is a subscribable resource that contains the following attributes:
deviceCategory	08	This field is for use in devices that can adjust energy usage (e.g., demand response, DERs). For devices that do not respond to EndDeviceControls or DERControls, this field should not have any bits set.
LFDI	C0FFEE00	Long form of device identifier.
SFDI	987654321005	Short form of device identifier.
***** AbstractDevice attributes end here*****		

Source: SEPA (2025).

Table 10. Attributes of Function Set Assignments		
Attribute	**Example**	**Description**
FunctionSet Assignments List		A List element to hold FunctionSetAssignments objects. It is a subscribablelist and contains the following attribute:
pollRate	900	The default polling rate for this function set in seconds. If not specified, a default of 900 seconds is used. It is RECOMMENDED that a Client poll the resources of this function set every pollRate seconds.
***** FunctionSetAssignmentsList attributes end here*****		
FunctionSet Assignments (Object)		Provides an identifiable, subscribable collection of resources for a particular device to consume. It contains the following attributes:
description	Operation X	The description is a human-readable text describing or naming the object.
mRID	0ED30F5A0000	The global identifier of the object.
Subscribable	Subscribable=1 Not Subscribable=0	Indicates whether or not subscriptions are supported for this resource, and whether or not conditional (thresholds) are supported. If not specified, is "not subscribable."
Version		Contains the version number of the object.
***** FunctionSetAssignment attributes end here*****		
FunctionSetAssignmentsBase object (Resource)		Defines a collection of function set instances that are to be used by one or more devices as indicated by the EndDevice object(s) of the Server.

Source: SEPA (2025).

Table 11. Response Function Set Resources and Their Attributes		
Attribute	**Example**	**Description**
Response (Resource)		Generic response data repository that is extended for specific function sets. It contains the following attributes:
createdDateTime	1234900	The createdDateTime field contains the date and time when the acknowledgement/status occurred in the Client. The Client will provide the timestamp to ensure the proper time is captured in case the response is delayed in reaching the Server. The time reported from the Client should be relative to the time Server indicated by the FunctionSetAssignment that also indicated the event resource; if no FunctionSetAssignment exists, the time reported from the Client should be relative to the time of the Server where the event resource was hosted.
endDeviceLFDI	C0FFEE00	Contains the long-form device identifier (LFDI) of the device providing the response.
status	1	The status field contains the acknowledgement or status. Each event type (DRLC, DER, Price) can return different status information (e.g., an Acknowledge will be returned for a Price event where a Demand Response Load Control (DRLC) event can return Event Received, Event Started, and Event Completed.
subject	OED30F5A0000	The subject field provides a method to match the response with the originating event. It is populated with the mRID of the original object.

Source: SEPA (2025).

Table 12. DER Function Set Lists, Resources, and Their Attributes		
Attribute	**Example**	**Description**
DERProgramList		A List element to hold DERProgram objects. These DERProgramList objects are used to define the possible controls for groups. It contains the following attribute:
PollRate	900	The default polling rate for this function set in seconds. If not specified, a default of 900 seconds is used. It is RECOMMENDED that a Client poll the resources of this function set every pollRate seconds.
***** DERProgramList attributes end here*****		
DERProgram		Distributed energy resource programs. Is a resource and contains the following attributes:
Primacy	2	Indicates the relative primacy of the provider of this program. Lower values of primacy represent higher priority levels.
mRID	01BE7A7E57	The global identifier of the object.
description	An example of a DER program	The description is a human-readable text describing or naming the object.
version		Contains the version number of the object.
subscribable	Subscribable = 1 Not Subscribable = 0	Indicates whether or not subscriptions are supported for this resource and whether or not conditional (thresholds) are supported. If not specified, is "not subscribable."
href	/edev/3/fsal	A reference to the resource address (URI).
***** DERProgram attributes end here*****		
DERControl		Distributed energy resource (DER) time/event-based control. It contains the following attribute:
deviceCategory		Specifies the bitmap indicating the categories of devices that SHOULD respond. Devices SHOULD ignore events that do not indicate their device category. If not present, all devices SHOULD respond.
***** DERControl attribute ends here*****		
DefaultDERControl		Contains control mode information to be used if no active DERControl is found. It contains the following attributes:
setESDelay		Enter service delay, in hundredths of a second.
setESHighFreq		Enter service frequency high. Specified in hundredths of hertz.
setESHighVolt		Enter service voltage high. Specified as an effective percent voltage.
setESLowFreq		Enter service frequency low. Specified in hundredths of hertz.
setESLowVolt		Enter service voltage low. Specified as an effective percent voltage.
setESRampTms		Enter service ramp time, in hundredths of a second.
setESRandomDelay		Enter service randomized delay, in hundredths of a second.
setGradW		Set default rate of change (ramp rate) of active power output due to command or internal action, defined in %setWMax/second. Resolution is in hundredths of a percent/second. A value of 0 means there is no limit.

Table 12. DER Function Set Lists, Resources, and Their Attributes		
Attribute	**Example**	**Description**
setSoftGradW		Set soft-start rate of change (soft-start ramp rate) of active power output due to command or internal action, defined in %setWMax/second. Resolution is in hundredths of a percent/second. A value of 0 means there is no limit.
DERControlBase		Distributed energy resource (DER) control values. It contains the following attributes (not all attributes have been listed):
opModConnect		Set DER as connected (true) or disconnected (false).
opModEnergize		Set DER as energized (true) or de-energized (false).
opModFixedPF AbsorbW		The opModFixedPFAbsorbW function specifies a requested fixed power factor (PF) setting for when active power is being absorbed.
opModFixedPF InjectW		The opModFixedPFInjectW function specifies a requested fixed power factor (PF) setting for when active power is being injected.
opModFixedVar		The opModFixedVar function specifies the delivered or received reactive power setpoint.
rampTms		Requested ramp time, in hundredths of a second, for the device to transition from the current DERControl mode setting(s) to the new mode setting(s). If absent, use default ramp rate (setGradW). Resolution is 1/100 sec.
***** DERControl attribute ends here*** (not all attributes have been listed)**		

Source: SEPA (2025).

Table 13. Device Capability Object		
Attribute	**Example**	**Description**
pollRate	900	Allows Clients to find the URIs to the resources in which they are interested. The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED that a Client poll the resources of this function set every pollRate seconds.

Source: SEPA (2025).

Table 14. Pricing Function Set Attributes		
Attribute	**Example**	**Description**
TariffProfile		A schedule of charges; structure that allows the definition of tariff structures such as step (block) and time of use (tier) when used in conjunction with TimeTariffInterval and ConsumptionTariffInterval.
Currency	840	The currency code indicating the currency for this TariffProfile.
pricePowerOfTenMultiplier	-6	Indicates the power of ten multiplier for the price attribute.
primacy	0	Indicates the relative primacy of the provider of this program.
Ratecode	TOU-D-PEV Baseline 6	The rate code for this tariff profile. Provided by the Pricing service provider per its internal business needs and practices, it provides a method to identify the specific rate code for the TariffProfile instance. This would typically not be communicated to the user except to facilitate troubleshooting due to its service provider–specific technical nature.
serviceCategorykind		The kind of service provided by this usage point.
mRID	799794f4620b17e00000e566	The global identifier of the object.
description	PEV TOU Rate	The description is a human-readable text describing or naming the object.
version		Contains the version number of the object.
href	/edev/3/fsal	A reference to the resource address (URI).
***** MirrorMeterReading attributes end here*****		
RateComponent		Specifies the applicable charges for a single component of the rate, which could be generation price or consumption price, for example.
flowRateEndLimit		Specifies the maximum flow rate (e.g., kW for electricity) for which this RateComponent applies, for the usage point and given rate/tariff.
flowRateStartLimit:		Specifies the minimum flow rate (e.g., kW for electricity) for which this RateComponent applies, for the usage point and given rate/tariff.
roleFlags	12	Specifies the roles that this usage point has been assigned.
mRID	799794f4620b17e00000e566	The global identifier of the object.
description	PEV TOU Rate	The description is a human-readable text describing or naming the object.
version		Contains the version number of the object.
href	/edev/3/fsal	A reference to the resource address (URI).
***** RateComponent attributes end here*****		
TimeTariffInterval		Describes the time-differentiated portion of the RateComponent, if applicable, and provides the ability to specify multiple time intervals, each with its own consumption-based components and other attributes.
touTier		Indicates the time of use tier related to the reading. If not specified, is assumed to be "0 - N/A."
creationTime:	1357430400	The time at which the object was created

Table 14. Pricing Function Set Attributes		
Attribute	**Example**	**Description**
Interval		The period during which the Event applies.
mRID	799794f4620b17e00000e566	The global identifier of the object.
description	PEV TOU Rate	The description is a human-readable text describing or naming the object.
version		Contains the version number of the object.
href	/edev/3/fsal	A reference to the resource address (URI).
subscribable	0	0 = Resource does not support subscriptions 1 = Resource supports non-conditional subscriptions 2 = Resource supports conditional subscriptions 3 = Resource supports both conditional and non-conditional subscriptions
replyTo		Indicates whether or not a response is required upon receipt, creation, or update of this resource. Responses shall be posted to the collection specified in "replyTo." 0 = End device shall indicate that message was received 1 = End device shall indicate specific response 2 = End user/customer response is required
responseRequired		A reference to the response resource address (URI). Required on a response to a GET if responseRequired is "true."
***** TimeTariffInterval attributes end here*****		
ConsumptionTariffInterval (Resource)		One of a sequence of thresholds defined in terms of consumption quantity of a service such as electricity, water, gas, etc. It defines the steps or blocks in a step tariff structure, where startValue simultaneously defines the entry value of this step and the closing value of the previous step. Where consumption is greater than startValue, it falls within this block; and where consumption is less than or equal to startValue, it falls within one of the previous blocks. It contains the following attributes:
Consumptionblock		Indicates the consumption block related to the reading. If not specified, is assumed to be "0 - N/A."
Price	113000	The charge for this rate component, per unit of measure defined by the associated ReadingType, in currency specified in TariffProfile.
startValue	0	The lowest level of consumption that defines the starting point of this consumption step or block.
href		
***** ConsumptionTimeTariffInterval attributes end here*****		

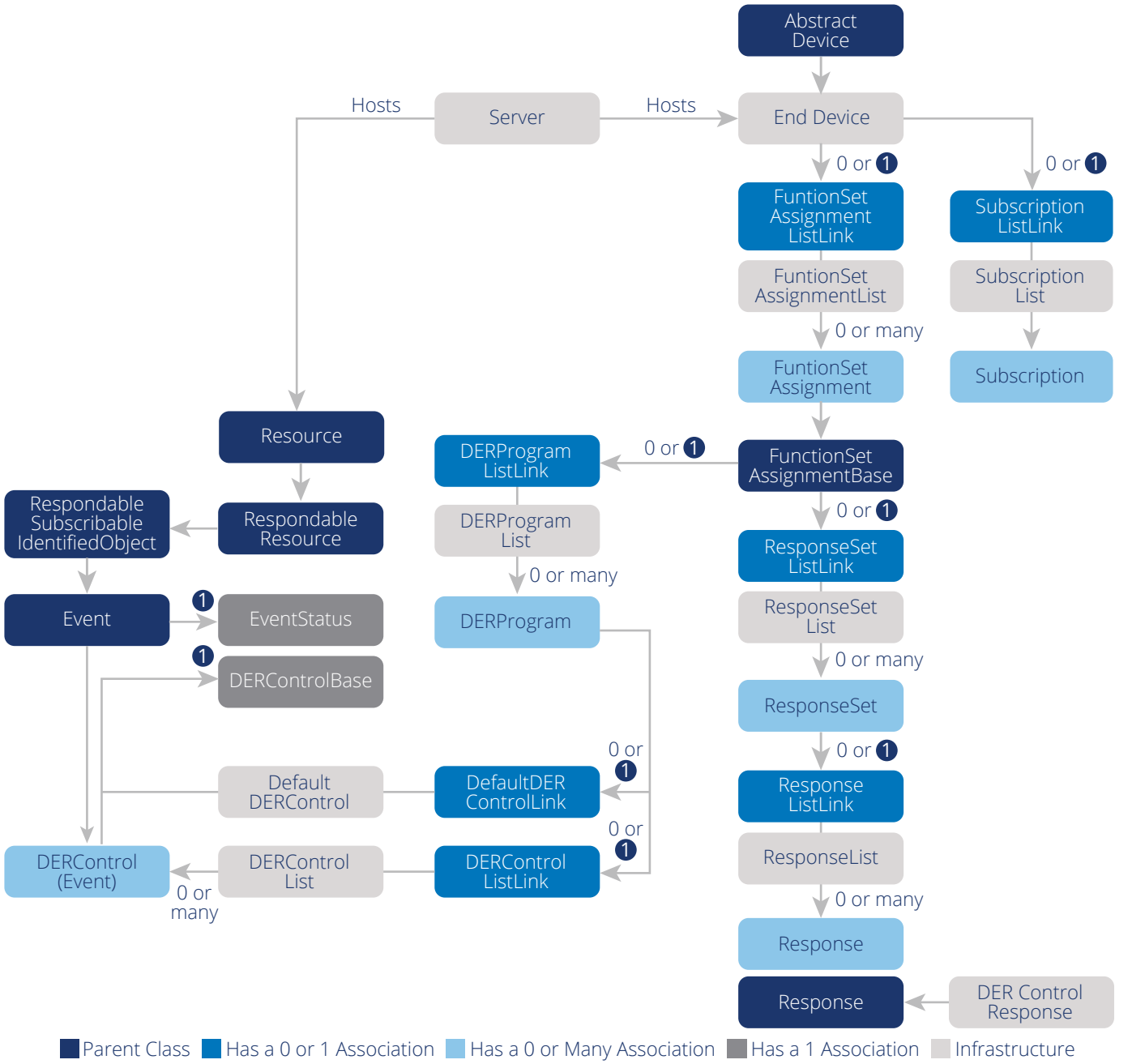
Source: SEPA (2025).

Table 15. Metering Mirror Function Set Attributes		
Attribute	**Example**	**Description**
MirrorUsagePoint		Object, contains the following attributes:
DeviceLFDI		The LFDI of the device being mirrored.
PollRate		POST rate, or how often mirrored data should be POSTed, in seconds. A Client MAY indicate a preferred postRate when POSTing MirrorUsagePoint. A Server MAY add or modify postRate to indicate its preferred posting rate.
***** MirrorUsagePoint attributes end here*****		
MirrorMeterReading		Object, contains the following attributes:
lastUpdateTime		The date and time of the last update.
nextUpdateTime		The date and time of the next planned update.
mRID		The global identifier of the object.
description		The description is a human-readable text describing or naming the object.
version		Contains the version number of the object.
href	/edev/3/fsal	A reference to the resource address (URI).
***** MirrorMeterReading attributes end here*****		

Source: SEPA (2025).

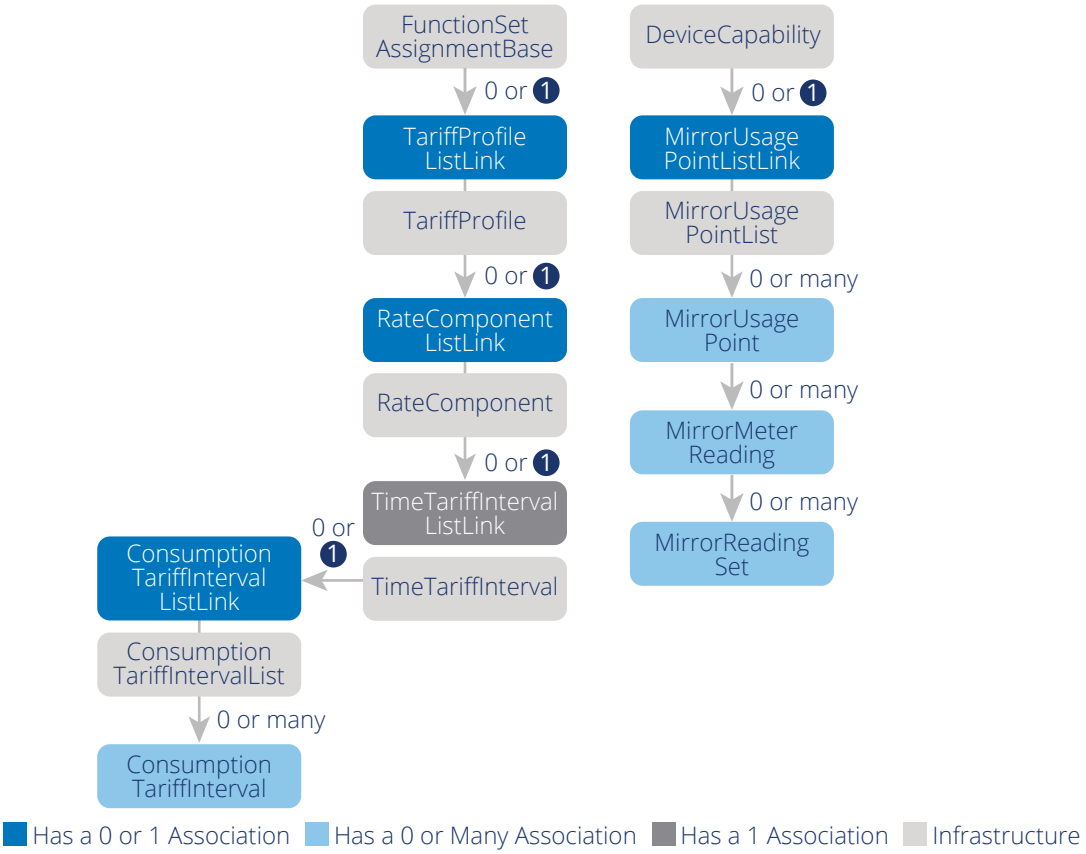
Figure 3 and Figure 4 illustrate the relationship between the resources of the function sets mentioned in the preceding tables.

Figure 3. Relationships Between List, Resources, Objects



Source: SEPA (2025).

Figure 4. Relation Between List, Resources, Objects for Metering and Pricing Function Sets



Source: SEPA (2025).

Scenarios 1 and 2 Step-by-Step Exchanges in IEEE 2030.5

This section details how to implement the Use Case using IEEE 2030.5.

Scenario 1: Direct Exchange

In Scenario 1, the DSO is the Server, and the Client is the CSMS.

Before implementing the steps shown in Table 3, the following steps must be accomplished:

1. **DSO should create a DERProgram and DERProgramList.**
2. **DSO should create the Groups (i.e., FunctionSetAssignments).**

   FunctionSetAssignments (FSAs) are used to bundle specific DER programs for usage. The basic approach is to store the appropriate DERProgramList link with the FSA.
3. **DSO should create an FSA List for each EVSE.**

   FunctionSetAssignmentList objects are used to define the relationship between individual EVSE and the groups created in the previous step.
4. **DSO should create an EndDevice instance for each EVSE.**

   The appropriate FSA List to each EndDevice instance is assigned.
5. **DSO should create an EndDevice instance for the CSMS.**

   The CSMS instance does not have an FSA List link but does have a Subscription List link. The Subscription List link allows the CSMS to POST subscriptions for resources for which it wishes to receive notifications related to its managed EVSEs. Because this is a global resource for the CSMS, the utility will only create a single end device instance.
6. **DSO should create an EndDevice List for the CSMS.**

   The EndDeviceList contains the CSMS's EndDevice instance as well as all the instances for the EVSE under the CSMS's control. Each CSMS managed by the utility will receive a uniquely tailored EndDeviceList reflecting all its currently enrolled systems with the utility.
7. **CSMS should find the EndDeviceList.**

   From the DeviceCapability response, the CSMS knows the URL of the EndDeviceList resource. It then GETs the contents of the complete list (EndDeviceList). The list contains entries for the CSMS as well as all the EVSEs under the CSMS's control.
8. **CSMS should find the group assignments of each EVSE.**

   This group retrieval process involves the following steps:

   - Get the FSA List Link. For each EVSE, get the FunctionSetAssignmentsListLink. This link is the location where the group assignments for this EVSE can be found.
   - Get the FSA List. Using the link, get the FunctionSetAssignmentsList for each EVSE. Each entry in the list is an FSA (i.e., a group).
9. **CSMS should find the DER Programs associated with each group's assignments.**

   The DER Program retrieval process involves the following steps:

   - Get the DER Program. For each FSA, use the DERProgramListLink to get the DERProgramList and all the DERPrograms in the list.
   - Get the DER Information for each Program. For each DERProgram, get the relevant information, like primacy and links to the DERControlList, and the DefaultDERControl.
10. **CSMS should subscribe to DER Programs.**

    By subscribing, the DSO pushes a Notification to the CSMS when there is a change to the DER Control List. The CSMS then gets the DER Control List and finds the new DER Control event.

The CSMS shall subscribe to at least the following resources:

- EndDeviceList—to receive notification of the addition of an inverter
- EndDevice—to receive notification of the deletion or change to an existing inverter
- FunctionSetAssignmentsList—to receive notification of group changes
- DERControlList—to receive notifications of new DER Controls or changes to an existing DER Control
- DERProgramList—to receive notifications of new DER Programs
- DERProgram—to receive notifications when program metadata like primacy changes

Step-by-Step Guide to Creating a Subscription for Scenario 1

- **POST the Subscription.** After discovering the URI of the DSO Server's SubscriptionList and desired resource, the CSMS appends new subscription entries to the list. It POSTs the URI of the Notification resource, together with the DERControlListLink it read from the DERProgram, to the subscription list of its EndDevice instance.
- **Get the Subscription Confirmation.** If the subscription was successful, the DSO Server returns a confirmation with the location of the confirmed subscription. The DSO Server will now notify the CSMS if there are any changes to the subscribed resource.

Figure 5 illustrates the subscription process.

If there is a change to the subscribed resource, the DSO notifies the CSMS by POSTing to the CSMS notification resource.

On behalf of every EVSE under CSMS control, the CSMS subscribes to each EVSE End Device instance.

After completing the steps mentioned above, the Use Case steps in Table 3 can be applied via the following:

- **Steps 1 and 2:** To create a DER event for a group, create a DER Control with the appropriate Start time, Duration, and DER control type.

  This control is then added to the DER Control List for the DER Program group.

  For each DERControl with ResponseRequired Bit 0 set, CSMS confirms the receipt of the notification by POSTing a response with a status of "Message Received" to the response resource specified by the replyTo field.
- **Step 3:** [Not covered by IEEE 2030.5]
- **Step 4:** CSMS opts into Demand Management by POSTing the response to the indicated URI in bitfield 1 of the event. The destination of the Responses is controlled by the replyTo attribute.
- **Step 5:** If the CSMS opts in, it begins the event at the specified start time and POSTs a response with a status of "Event Started" to the Response resource specified by the replyTo attribute.
- **Step 6:** [Not covered by IEEE 2030.5]
- **Step 7:** CSMS completes the event and POSTs a response with a status of "Event Completed" to the response resource specified by the replyTo attribute.

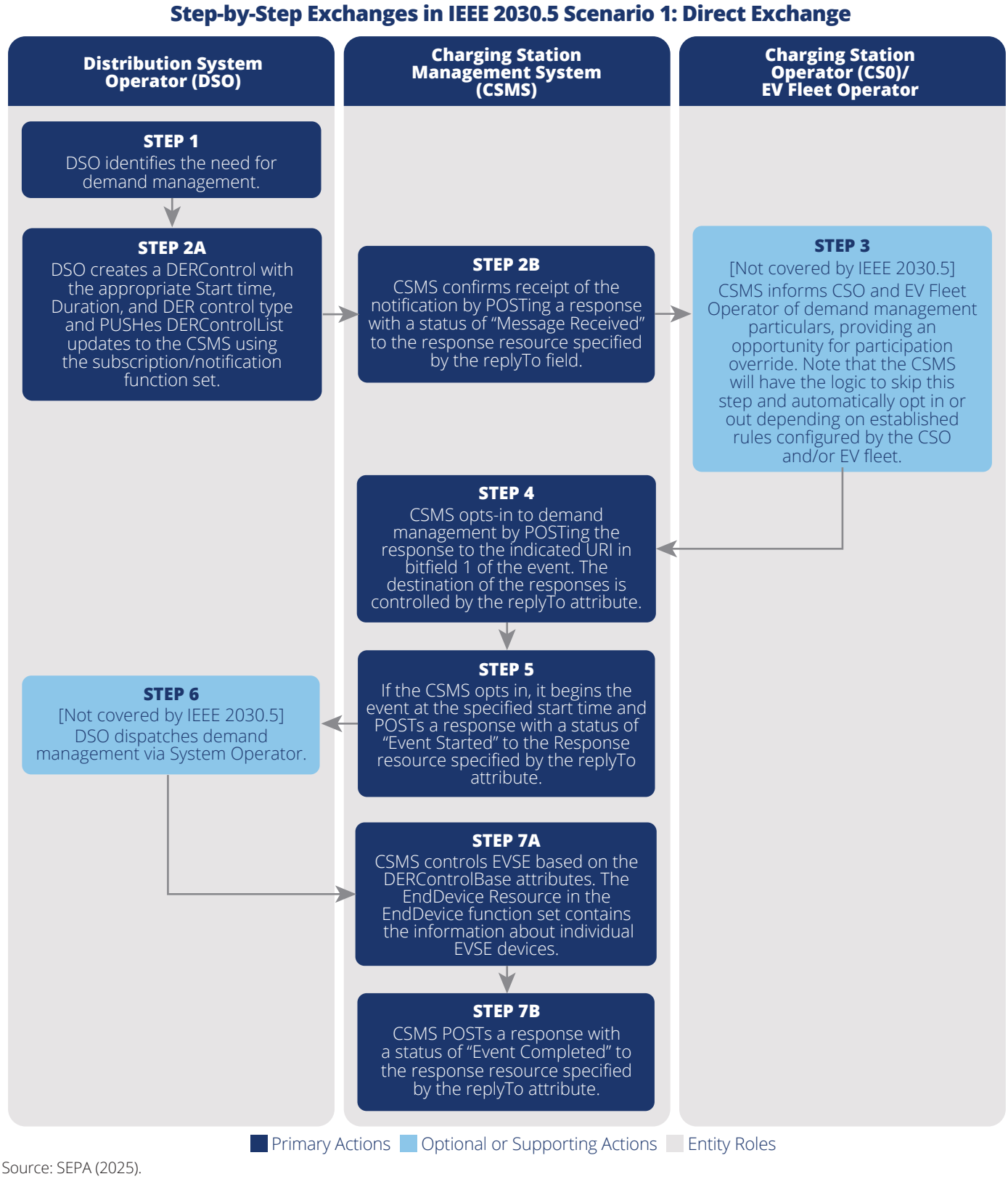
The activity diagram (Figure 6) represents all the steps discussed above.

Figure 5. Subscription Process



Source: SEPA (2025).

Figure 6. Activity Diagram for Scenario 1



Source: SEPA (2025).

Scenario 2: Aggregator-Mediated

In Scenario 2, the DSO is the Server, and the Client is now the Aggregator.

Before implementing the steps shown in Table 4, the following steps must be accomplished:

1. **DSO should create a DERProgram and DERProgramList.**
2. **DSO should create the Groups (i.e., FunctionSetAssignments).**

   FunctionSetAssignments (FSAs) are used to bundle specific DER programs for usage. The basic approach is to store the appropriate DERProgramList link with the FSA.
3. **DSO should create an FSA List for each EVSE.**

   FunctionSetAssignmentList objects are used to define the relationship between individual EVSE, and the groups created in the previous step.
4. **DSO should create an EndDevice instance for each EVSE.**

   The appropriate FSA List to each EndDevice instance is assigned.
5. **DSO should create an EndDevice instance for the CSMS.**
6. **DSO should create an EndDevice List for the CSMS.**
7. **DSO should create an EndDevice instance for the Aggregator.**

   The CSMS instance does not have an FSA List link but does have a Subscription List link. The Subscription List link allows the Aggregator to POST subscriptions for resources it wishes to receive notifications related to its managed CSMS and their EVSEs. Because this is a global resource for the Aggregator, the utility will only create a single end device instance.
8. **DSO should create an EndDevice List for the Aggregator.**

   The EndDevice List contains the Aggregator's EndDevice instance as well as all the instances for the CSMS and the EVSE under its control. Each Aggregator managed by the utility will receive a uniquely tailored EndDevice List reflecting all its currently enrolled systems with the utility.
9. **Aggregator should find the EndDeviceList.**

   From the DeviceCapability response, the Aggregator knows the URL of the EndDeviceList resource. It then GETs the contents of the complete list (EndDeviceList). The list contains entries for the CSMS as well as all the EVSEs under the CSMS's control.
10. **Aggregator should find the group assignments of each CSMS and their EVSEs.**

    This group retrieval process involves the following steps:

    - Get the FSA List Link. For each EVSE, get the FunctionSetAssignmentsListLink. This link is the location where the group assignments for this EVSE can be found.
    - Get the FSA List. Using the link, get the FunctionSetAssignmentsList for each EVSE. Each entry in the list is an FSA (i.e., a group).
11. **Aggregator should find the DER Programs associated with each group's assignments.**

    The DER Program retrieval process involves the following steps:

    - Get the DER Program. For each FSA, use the DERProgramListLink to get the DERProgramList and all the DERPrograms in the list.
    - Get the DER Information for each Program. For each DERProgram, get the relevant information, like primacy and links to the DERControlList, and the DefaultDERControl.
12. **Aggregator should subscribe to DER Programs.**

    By subscribing, the DSO pushes a Notification to the Aggregator when there is a change to the DER Control List. The Aggregator then gets the DER Control List and finds the new DER Control event.

    The Aggregator shall subscribe to at least the following resources:

    - EndDeviceList—to receive notification of the addition of an inverter
    - EndDevice—to receive notification of the deletion or change to an existing inverter
    - FunctionSetAssignmentsList—to receive notification of group changes
    - DERControlList—to receive notifications of new DER Controls or changes to an existing DER Control
    - DERProgramList—to receive notifications of new DER Programs
    - DERProgram—to receive notifications when program metadata like primacy changes

Step-by-Step Guide to Creating a Subscription for Scenario 2

- **POST the Subscription.** After discovering the URI of the DSO Server's SubscriptionList and desired resource, the Aggregator appends new subscription entries to the list. It POSTs the URI of the Notification resource, together with the DERControlListLink it read from the DERProgram, to the subscription list of its EndDevice instance.
- **Get the Subscription Confirmation.** If the subscription was successful, the DSO Server returns a confirmation with the location of the confirmed subscription. The DSO Server will now notify the Aggregator if there are any changes to the subscribed resource.

Figure 5 in the previous section illustrates the subscription process.

If there is a change to the subscribed resource, the DSO notifies the Aggregator by POSTing to the Aggregator notification resource.

The Aggregator subscribes to each CSMS and EVSE End Device instance.

After completing the steps mentioned above, the Use Case steps in Table 4 can be applied as follows:

- **Steps 1 and 2:** To create a DER event for a group, create a DER Control with the appropriate Start time, Duration, and DER control type.

  This control is then added to the DER Control List for the DER Program group.

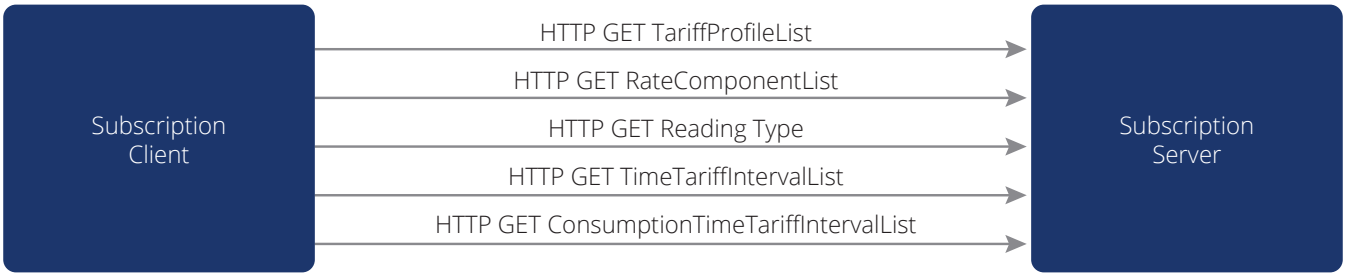
  For each DERControl with ResponseRequired Bit 0 set, Aggregator confirms the receipt of the notification by POSTing a response with a status of "Message Received" to the response resource specified by the replyTo field in the DERControl.
- **Step 3:** [Not covered by IEEE 2030.5]
- **Step 4:** [Not covered by IEEE 2030.5]
- **Step 5:** [Not covered by IEEE 2030.5]
- **Step 6:** Aggregator opts into Demand Management by POSTing the response to the indicated URI in bitfield 1 of the event. The destination of the Responses is controlled by the replyTo attribute.
- **Step 7:** If the Aggregator opts-in, it begins the event at the specified start time and POSTs a response with a status of "Event Started" to the Response resource specified by the replyTo attribute.
- **Step 8:** [Not covered by IEEE 2030.5]
- **Step 9:** [Not covered by IEEE 2030.5]
- **Step 10:** [Not covered by IEEE 2030.5]
- **Step 11:** Aggregator completes the event and POSTs a response with a status of "Event Completed" to the response resource specified by the replyTo attribute.
- **Step 12:** The Metering Mirror and Pricing function sets will be used to measure performance and calculate payments.

  The Utility Server can identify the source of the End Device meter data in two ways. First, it can provide a unique MirrorUsagePointListLink to each End Device such that each End Device PUT/POSTs to a different location. Second, the LFDI of the End Device is contained in the MeterUsagePoint that is POSTed to the Utility Server.

Figure 7 illustrates the pricing process. The subscription Server responds with HTTP 200 to all interactions.

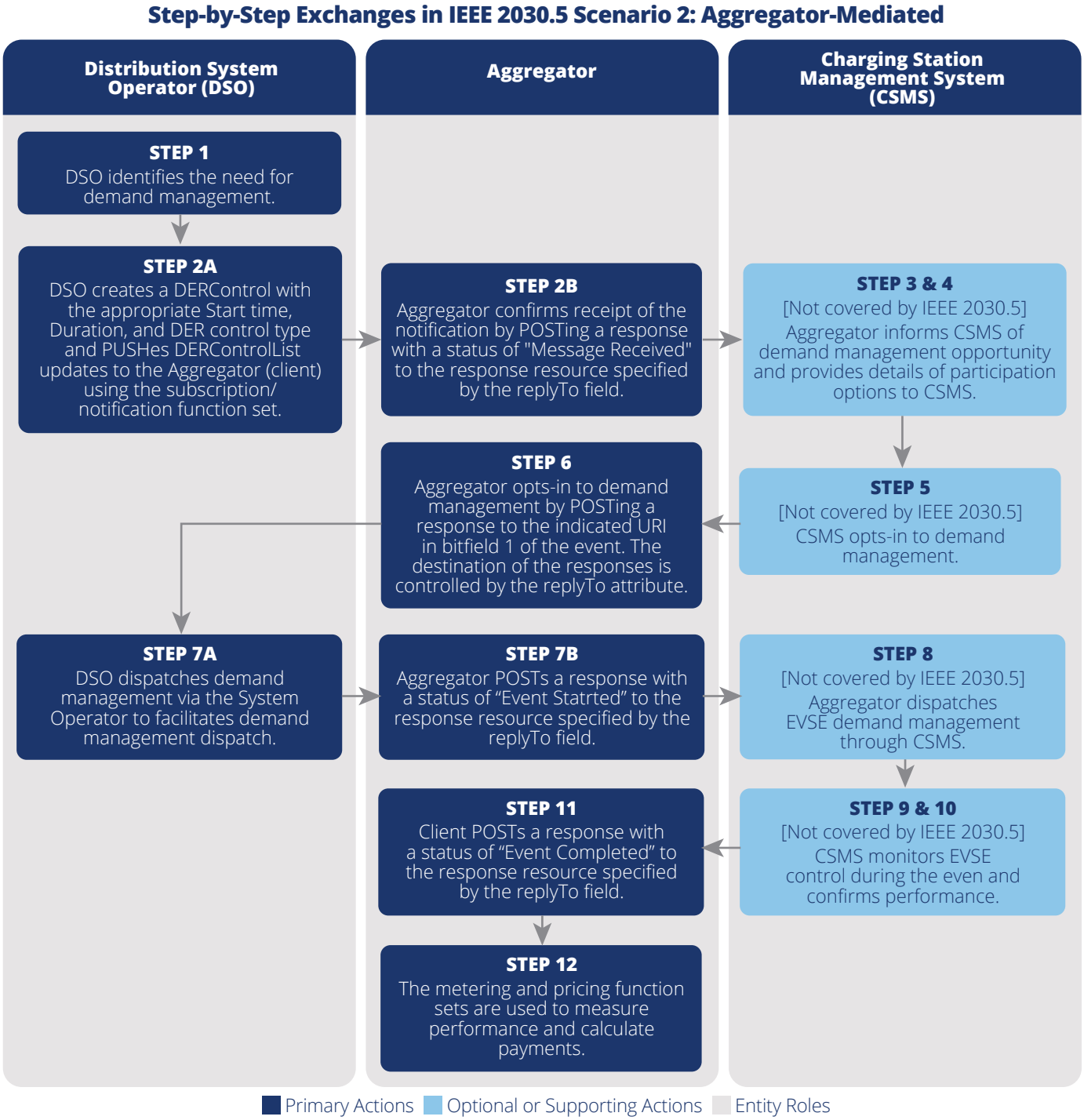
The activity diagram (Figure 8) represents all the steps discussed above.

Figure 7. Pricing Process



Source: SEPA (2025).

Figure 8. Activity Diagram for Scenario 2



Source: SEPA (2025).

Conclusion

This guide describes the specific instructions for implementing Scenario 1 (direct DSO to CSMS connectivity) and Scenario 2 (DSO to Aggregator connectivity) of the Electric Vehicle Fleet Managed Charging Use Case in terms of the IEEE 2030.5 standard. The authors hope that this document will make the implementation of smart grid standards for EV integration and charging infrastructure less complicated, less challenging, and clearer.

Analyzing the results of this application guide to identify any gaps in the Use Case and in this document could be a relevant next step in maturing these processes in the electric power sector.



Smart Electric Power Alliance

1800 M STREET, NW FRONT 1
#33159
WASHINGTON, DC 20036
202-857-0898

©2025 Smart Electric Power Alliance. All Rights Reserved.