Common Smart Inverter Profile

IEEE 2030.5 Implementation Guide for Smart Inverters

March 2018

Version 2.1

Contents

1 Introduction

This guide serves to assist manufacturers, Distributed Energy Resources (DER) operators, system integrators and DER aggregators to implement the Common Smart Inverter Profile (CSIP) implementation guide for IEEE 2030.5. CSIP was developed as an outgrowth of the California Rule 21 Smart Inverter process to create common communication profile for inverter communications that could be relied on by all parties to foster "plug and play" communications-level interoperability (outside of out-of-band commissioning) between the California IOU's and 3rd party operated smart inverters or the systems/service providers managing those inverters. Rule 21 Smart Inverter proceedings segregated smart inverter functionality and implementations into three progressive phases: Phase 1, which comprises the *Autonomous* functionality and related settings which inverters must support when interconnected to California Investor Owned Utility's (IOU) distributions system; Phase 2, which prescribes the communications between the IOUs and DER aggregators, DER management systems, and DERs themselves; and Phase 3 which details the use of Phase 2 communications for monitoring and control and other necessary uses. This implementation guides was a required outcome of Phase 2, which prescribed IEEE 2030.5 as the default protocol for Rule 21 Smart Inverter communications. This guide, along with the IEEE 2030.5 specifications, is also intended to be used to develop an IEEE 2030.5 Client conformance test plan and certification program which is required in California.

While the impetus and scope of this profile of 2030.5 was to meet the needs of the California IOU's requirements for communications, the profile implements widely applicable use cases making CSIP generic and likely applicable to other regulatory jurisdictions beyond California's borders. With this in mind, the California Rule 21 specific terminology is genericized throughout this document. Additionally, it is important to note that while this guide intends to describe the full set of Rule 21 and IEEE 2030.5 DER Client requirements, much of the actual implementation details and requirements are expected to be derived from utility interconnection tariffs (e.g., Rule 21), Utility Handbooks, contracts or other regulatory or program-related vehicles. Where this is so, it is denoted throughout this guide.

2 Guiding Principles

The following principles have been used to help guide the development of CSIP. From a communications perspective

1. All smart inverters require communications to achieve their full value as distributed energy resources.
2. Establish a complete profile – To achieve complete interoperability a complete profile is required including a data model, messaging model, communication protocol and security. Without a complete profile specification it would be impossible to achieve communications interoperability without additional systems integration activities.
3. Leverage existing standards and models from both engineering (e.g., IEEE 1547) and communications (e.g., IEEE 2030.5) standards – The development of a new, stand-alone

standard would create additional burden on all parties and only serve to raise costs of both development and maintenance.

4. Assume that future revisions will be necessary – The use of DERs will continue to evolve and utilities and other DER stakeholders anticipate the emergence of additional use cases to the near future (5 years). But, attempting to anticipate all future use cases will add complexity to the specification without commensurate value. As such, extensibility of the specification through future revisions is required.
5. Eliminate optionality and keep to a single base specification – Optionality in the specification can serve to hinder interoperability when parties chose to implement.
6. Create a minimal specification – A simple interface serves to lower costs and improve quality.
7. Strictly focus on utility to DER owner/operators and aggregators. All other communications are out of scope from the perspective of CSIP.
8. Strictly focus on inverter management, such as monitoring, setting group changes and basic on/off functions, rather than explicit real-time control.
9. Implementation of the interface infers no proprietary advantage to any party – Smart Inverter communications between the utility and 3rd parties provides a critical, but non-differentiating service. As such, the costs to all parties should be minimized to drive proliferation of DER in California.
10. Provide alternate models of implementation around a single common standard to provide customer choice, 3rd party business models and utility needs.

3 Communications Architecture Overview

3.1 Scope of Communications

CSIP addresses the communications path between the utility and the Aggregator, the utility and a Generating Facility Management System (GFEMS), and the utility and the Smart Inverter Control Unit (SMCU). Communications between the Aggregator/GFEMS and its managed DERs or communications within the DER are out of scope.

3.2 Scenarios

The two scenarios envisioned for communications between the utility and DER systems are Direct DER Communications and Aggregator Mediated Communications. In both cases, the communications path to the utility is governed by regulatory and utility requirements, and the IEEE 2030.5 protocol.

1. **Scenario 1: Direct DER Communications** – In this scenario, the utility communicates with the DER system directly. This scenario applies when the DER owner wishes to interact directly with the utility for managing their DER or when the utility needs to control the DER for proper system operations. The DER system itself can be architected in many ways. In this guide, the term "DER Client" is used generically to refer to any of the client devices shown in Figure 1.

a. **DER with Embedded or Separate Smart Inverter Control Unit (SMCU)**[1] – In this architecture, a Smart Inverter Control Unit is used to provide the communications component for a single DER and appears as a single IEEE 2030.5 *EndDevice* to the utility server. The SMCU can be integrated with the DER, or it can reside external to the DER. The communications path between the SMCU and DER is outside the scope of this guide.
b. **DER with Generating Facility Energy Management System (GFEMS)** – In this architecture, a Generating Facility Energy Management System mediates communications between the utility and one or more local DERs under its control. The GFEMS appears as a single IEEE 2030.5 *EndDevice* to the utility server and optimizes energy in the context of the overall energy. The communications path between the GFEMS and its DERs is outside the scope of this guide. The likely applicability for this architecture is for future residential, commercial or DER plant operations at a single point of common coupling, each of which will have differing requirements. This model may also be used to represent micro-inverters managed by a central controller.

[1] SMCU and GFEMS are terms used in Rule 21 Regulatory Documents

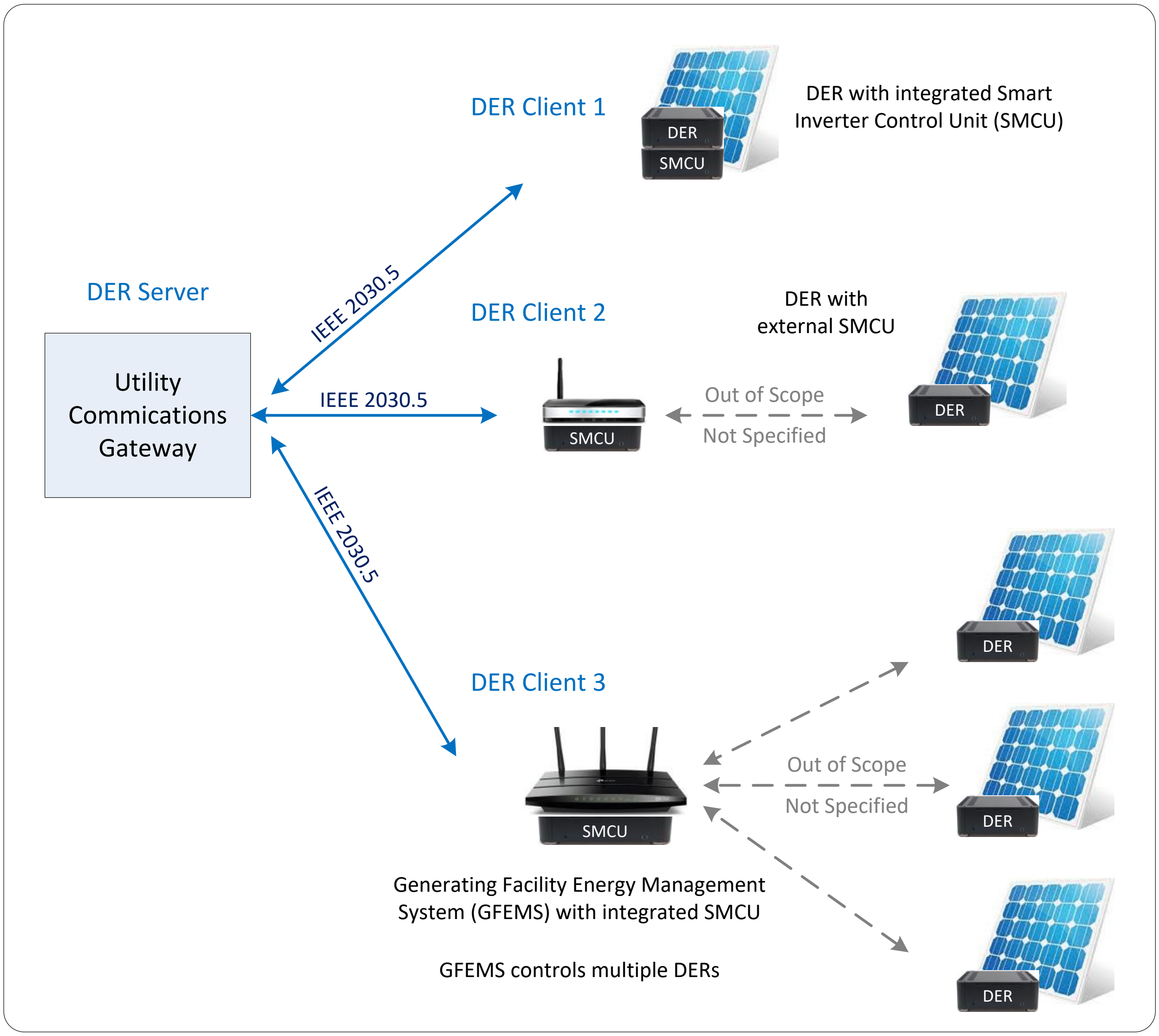

Figure 1 - Scenario 1: Direct DER Communications to IEEE 2030.5 DER Clients

Note that the notion of a DER in CSIP is logical concept generally thought of as one or more physical inverters organized and operating as a single system with a common point of aggregation behind a single point of common coupling (PCC) with the utility. This allows the management of a plant/system possessing a single PCC regardless of whether it is composed of a single inverter or many. It is the responsibility of the aggregator system to manage the underlying inverters to meet the requirements of the settings provided by the utility server. The specific interpretation of the DER being a single entity or a related group is established at the time of interconnection with the utility.

2. **Scenario 2: Aggregator Mediated Communications** – In this scenario the utility communicates with an Aggregator back end management system rather than directly with individual DERs. The Aggregator is assumed to be managing a fleet of inverters that are distributed across the utility's service territory rather than having a single point of common coupling. The Aggregator is then responsible for relaying any requirements for DER operational changes or data requests to the

affected systems and returning any required information to the utility. Each DER controlled by the Aggregator appears as an IEEE 2030.5 *EndDevice* to the utility server. The likely applicability is for fleet operators and aggregation service providers.

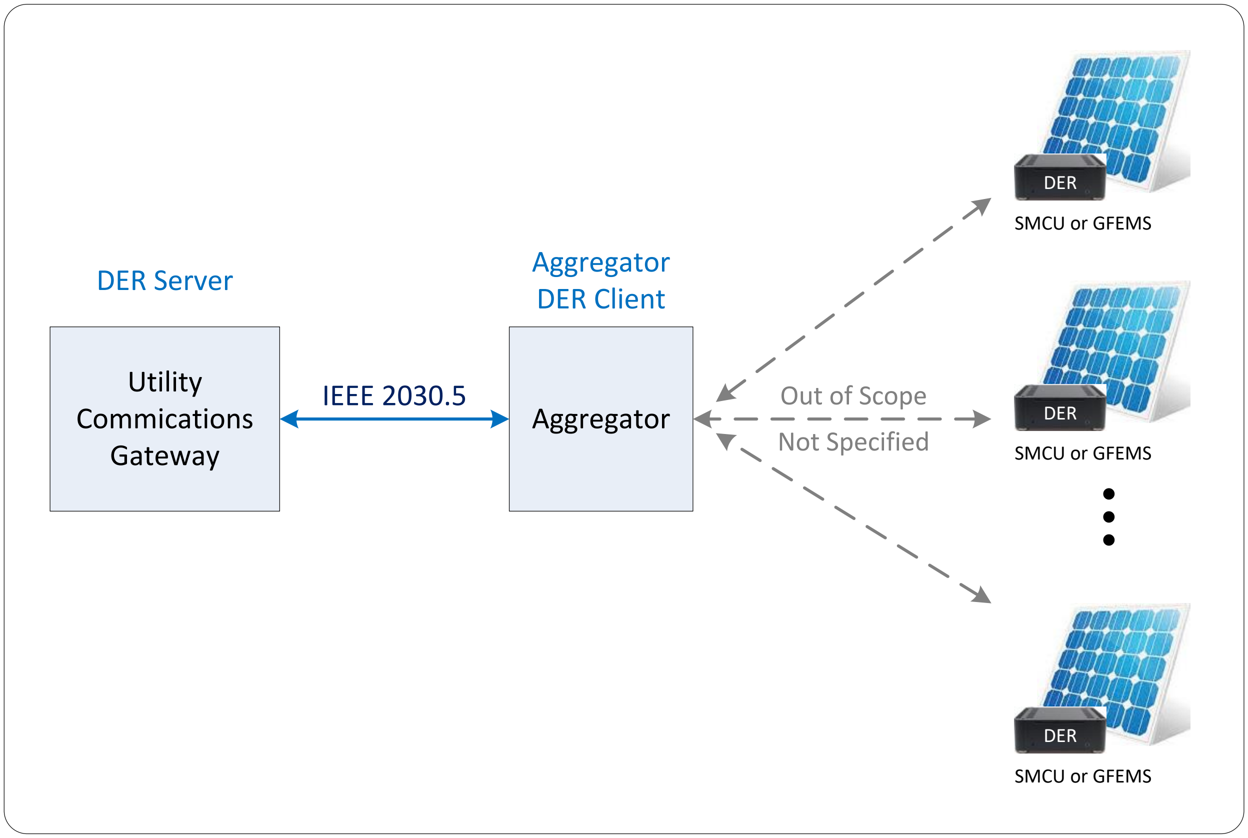

Figure 2 - Scenario 2: Aggregator Mediated Communications

Each DER SHALL[2] connect to the utility in one and only one scenario. The utility will designate the scenario of communications according to the utility's Interconnection Handbook requirements.

4 General CSIP Requirements

This section provides general requirements[3] related to implementing all grid support DER utility interactions. The related IEEE 2030.5 specific requirements can be found in Section 5.

4.1 Security Requirements

IEEE 2030.5 security requirements are covered in section 5.2.1. Although outside the scope of CSIP, security SHOULD be used in all non-IEEE 2030.5 interactions between the Aggregators, site hosts, GFEMS, and DERs and other entities receiving or transmitting DER related communications. Security includes data in motion (e.g. encryption of communications), data at rest, the authentication of clients

[2] The full set of requirements can be found in appendix B
[3] The key words "SHALL", "SHOULD", "MUST" and "MAY" in this document when capitalized constitute normative text and are to be interpreted as described in [RFC 2119].

and services, as well as the authorization of all requests. The composition of any Aggregator or DER access to utility servers is managed via contractual relationships. As such, the specific permissible actions across different utility servers may be different. See utility handbooks or programs/contracts for further cyber security requirements.

4.2 Registration and Identification of DERs

The registration of DER Clients is utility specific and is assumed to be outside the scope of CSIP. The registration process may result in the delivery of a globally unique identifier (GUID) associated with a particular DER. The GUID provides a shared name between the utility and the other party to ensure that operations and data are routed appropriately. The GUID is used to guarantee its authenticity and uniqueness within the scope of a single utility's CSIP server. For DER Clients that have an IEEE 2030.5 certificate, the GUID SHALL be derived from this certificate (see section 5.2.1.2). Implementers SHALL refer to each utility's Interconnection Handbook for requirements related to the creation, use or management of this identifier.

4.3 Group Management

Effective utility management of DERs requires that their location from an electrical system perspective be known. As a result, a special management function is required to align DERs operated by Aggregators to the utility system topology or other utility defined grouping. In certain cases, settings or commands can be sent to the entire system under a specific Aggregator's control. In other cases, the settings or commands will be targeted to limited numbers of DERs due to differences in needs across the utilities distribution system. For the purposes of this specification, DERs can be assigned to a minimum of one group and a maximum of 15 groups.

Although topological grouping is expected to be the primary use case, any type of grouping is allowed. A group consisting of DERs from a specific vendor or a group of DERs enrolled on a special program can be implemented. Each utility will apply the grouping levels as it sees fit to meet its own operational needs. For example, distribution transformer-level grouping is likely to be a future rather than a near term requirement. Likewise, other utilities may want to apply these group constructs in support of other distribution system network models.

Group membership may change over the life of the inverter being interconnected to the utility's system. These changes can be the result of system configuration or changes in segmentation or equipment. Aggregators and DER Clients SHALL support IEEE 2030.5 based grouping and full lifecycle management of group relationships as defined within Section 5.2.3 and within each utility's Interconnection Handbook or program/contract requirements.

Finally, a key concept of grouping is that DER can exist in multiple groups to support utility management at differing levels of the system. In all cases, the utility is responsible for maintaining these groups over time and to deliver any changes to groups to the impacted DERs.

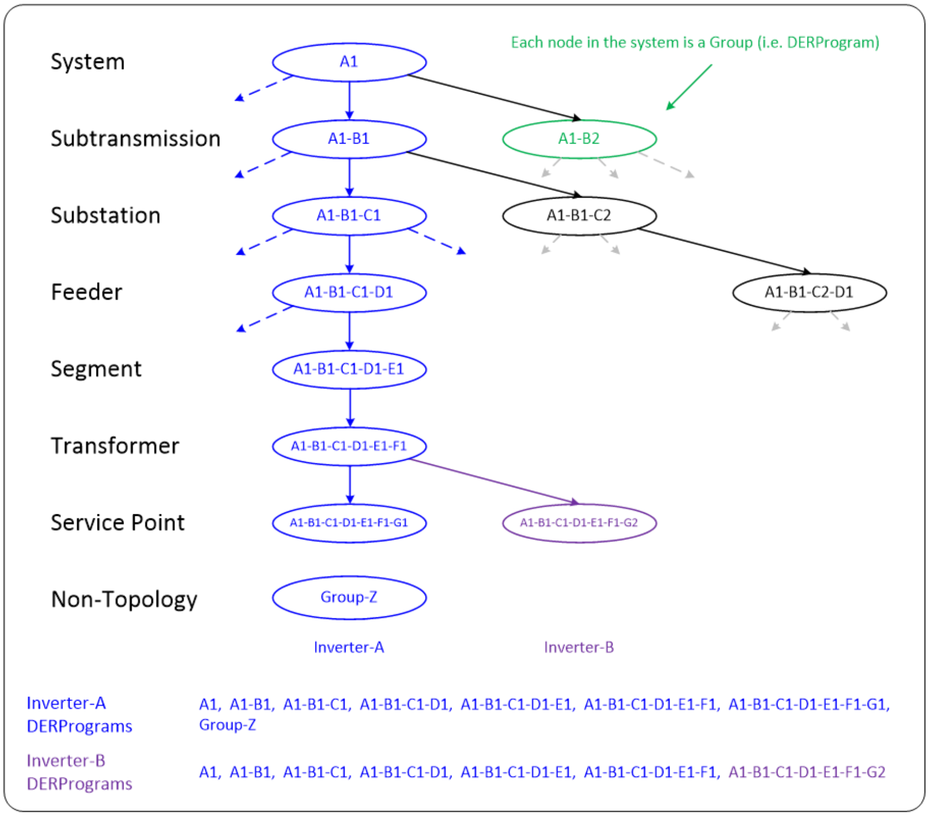

Figure 3 - Sample Grouping with Topology and Non-Topology Groups

1. System – refers to the utility service territory in total. All inverters are assigned to this group. It is expected that an inverter's membership will never change.
2. Sub-transmission – refers to a section of a utility's service territory where the transmission grid is managed directly by the utility
3. Substation – refers to the substation from which the inverter is electrically connected. Note that this group assignment can change as the electric system topology changes.
4. Feeder– refers to the feeder that the inverter is attached to. Note that this group assignment can change as the electric system topology changes.
5. Segment – refers to a section of a distribution feeder/circuit that cannot be further isolated or modified via switching or other sectionalizing device.
6. Service Transformer – refers to the collection of service points that are electrically connected to a single service transformer.
7. Service Point – refers to the point of common coupling between the utility and a 3rd party facility where one or more smart inverters are present.
8. Non-Topology- refers to a DER that has been placed in a group based on utility system needs

4.4 DER Control Events and Settings

4.4.1 Definition and Usage

Before listing the requirements, some terms that are used in this guide need to be defined and explained.

- A DER control is a generic term for a grid control function (e.g. fixed power factor or connect/disconnect).
- A *DERControl* is an IEEE 2030.5 control event that contains a start time, a duration, and a control parameter value. An example of a *DERControl* resource is the fixed power factor control event *DERControl*:*opModFixedPF*.
- A *DefaultDERControl* is an IEEE 2030.5 control resource that is in effect if there are no active *DERControls* for that resource. For example, the *DefaultDERControl:opModFixedPF* resource is in effect when there are no *DERControl:opModFixedPF* events active.

For most DER controls, there are two ways to issue the control: using *DERControl* events or using *DefaultDERControls*.

When the start time and duration of the control is known, the typical way to issue the control is to create a *DERControl* event for the control. Like any IEEE 2030.5 event, *DERControl* events can be scheduled, superseded, cancelled, etc. If configured, the utility DER server can receive the event status responses (e.g. received, started, completed, superseded, etc.) of the *DERControl* from each DER.

When the DER control is intended to be used to modify a setting (i.e. start time is “now” and the duration is indefinite), the most natural way to issue the control is to create or update the *DefaultDERControl*. The *DefaultDERControl* will be in effect until it is changed or a *DERControl* event occurs. In many use cases, the utility server may simply use *DefaultDERControls* and never issue a *DERControl* event for the controls. One limitation of using *DefaultDERControls* is there are no status responses associated with *DefaultDERControls*.

If status responses for modification of settings are needed, the utility server can use *DERControl* events. To accomplish this, the start time of the *DERControl* is “now”, and the duration is set to a very large, effectively infinite, number. To change the *DERControl* setting, a new *DERControl* is issued to supersede or cancel the existing *DERControl*.

4.4.2 Requirements

All DERs and related communications will support the Autonomous and Advanced functionality and controls as shown below.

Grid Support DER Functions	
Autonomous Functions	**Advanced Function**
Anti-Islanding	Connect/Disconnect
Low/High Voltage Ride Through	Limit Maximum Active Power Mode
Low/High Frequency Ride Through	Scheduling Power Values and Modes

Ramp Rate Setting	Monitor Key Data including Alarms, DER Status and Output
Dynamic Volt-Var	Volt-Watt Control
Fixed Power Factor Control	Frequency-Watt Control
	Set Active Power Mode

Table 1 – Grid DER Functions

Default settings or modes for Autonomous Functions, including which are activated and deactivated at deployment, will be specified in the applicable interconnection tariff and/or the utility's Interconnection Handbook. Autonomous functions' default settings SHALL be changeable via IEEE 2030.5 *DefaultDERControl* communications. Modifications to default settings SHALL occur immediately upon receipt and have an indefinite duration.

Scheduling Autonomous and Advanced Power Values and Modes SHALL be controllable via IEEE 2030.5 *DERControl* events. As opposed to modification of default settings, these events allow the server to schedule operations for single or groups of DERs at a future point in time for a specific duration. Through events, the utility can send one or more operations as a sequence to the DERs for processing and implementation. In this way, the utility can schedule and sequence DER control events.

Aggregators and DER Clients SHALL be responsible for assuring that all operations received from the utility are processed in the appropriate time sequence as specified by the utility.

An Aggregator acting for its DERs and DER Clients SHALL be able to store at least 24 scheduled DER control events for each DER.

In the absence of scheduled controls, DERs SHALL revert to a default control setting specified by interconnection tariffs, the utility Interconnection Handbook or as specified by the last *DefaultDERControl.*

Should there be a loss of communications, DERs SHALL complete any scheduled event and then revert to default settings or other settings as determined by the site host or tariffs/contracts.

4.4.3 Prioritization

When commanded in a manner where two or more operations are in conflict, the interpreting system SHALL operate against the control operation which has the highest priority subject to the systems capability, contracts and self-protection requirements.

In setting up commands for groups of DERs, it is expected that commands for lower level groups will typically have precedence over higher level groups (i.e. commands at the System level are trumped by commands at a more local level Feeder). In this manner, multiple needs can be managed. For example, a system level group operation might call for a voltage-watt mode of operation with a set of curve parameters at the same time as several circuits might require a voltage-watt mode with a different set of curve parameters.

The utility will avoid creating situations where there can be conflicting commands of the same priority. If avoidance of conflicting commands is not possible, the more recently received command SHOULD have precedence over the older command. In either case, it SHALL be the responsibility of the Aggregator or DER Client to decide how to handle two simultaneous controls.

4.5 Communication Interactions

For Aggregator communications, notifications and call backs (subscription/notification) SHALL be used to limit system polling to the greatest extent practical.

To simplify communication requirements for Direct DER Communications scenarios, unless specified otherwise in utility Interconnection Handbooks or programs/contracts, all communications SHALL be initiated by the DER Client (i.e., client-side initiation). This model of communication eliminates the need for unsophisticated parties to make changes in networking security based on the needs of CSIP.   In Direct DER communication scenarios, the client system SHALL initiate communications with the utility according to pre-defined polling and posting intervals to ensure the DER has up to date settings and the utility understands the operational state of the DER. Unless specified in each utility's Interconnection Handbook or programs/contracts, default polling and posting rates SHALL be as follows:

- Polling of *DERControls* and *DefaultDERControls* (Direct DER Communication)– every 10 minutes
- Posting monitoring information (Direct and Aggregator Mediated Communications)– every 5 minutes

For DERs with an external SMCU, the SMCU SHALL transfer the DER control to the DER within 10 minutes of receiving the control from the server.

For DERs with a GFEMS, the GFEMS SHALL transfer the DER control to the DERs within 10 minutes of receiving the control from the server.

For DERs mediated by Aggregators, the Aggregator SHALL transfer the DER control to the DERs within 15 minutes of receiving the control from the server.

4.6 Reporting DER Data

4.6.1 Monitor Data

Aggregators acting for its DERs and DER Clients SHALL have the capability to report the monitoring data in Table 2. Aggregators acting for its DERs and DER Clients SHALL have the capability to include the data qualifiers in Table 3. All measurement SHALL include a date-time stamp. Unless otherwise specified in each utility's Interconnection Handbook or programs/contracts, Aggregators acting for its DERs and DER Clients SHALL report the monitoring data in Table 2 and MAY include the data qualifiers in Table 3.

Monitoring Data
Real (Active) Power
Reactive Power
Frequency
Voltage per Phase

Table 2 - Monitoring Data

Data Qualifiers
Instantaneous (Latest)
Maximum over the Interval
Average over the Interval (the last posting)
Minimum over the Interval

Table 3 - Data Qualifiers

Note that some DERs may be capable of only reporting instantaneous measurements and cannot report minimum, maximum, or average values. For those situations where the DERs cannot provide Monitoring Data, the Aggregator acting for its DERs and DER Clients SHALL not send the data.

4.6.2 Status Information

4.6.2.1 Ratings and Settings

Aggregators acting for its DERs and DER Clients SHALL have the capability to report the Nameplate Ratings and Adjusted Settings information shown in Table 4. Nameplate Ratings and Adjusted Settings SHOULD be reported once at start-up and whenever there is a change in value. This information is not expected to change during normal operation. The Nameplate Rating is the value of the item as manufactured. The Adjusted Setting is the modified value of the Nameplate Rating to account site-specific deviations, degradations over time, or other factors. Specific requirements related to when Nameplate Ratings and Adjusted Setting must be provided will be found in each utility's Interconnection Handbook or contracts/programs.

Nameplate Ratings and Adjusted Settings
Maximum rate of energy transfer received
Maximum rate of energy transfer delivered
Maximum apparent power
Maximum reactive power delivered
Maximum reactive power received
Maximum active power
Minimum power factor displacement

Table 4 - Nameplate Ratings and Adjusted Settings

4.6.2.2 Operational Status Information

Aggregators acting for its DERs and DER Clients SHALL have the capability to report the dynamic Operational Status Information shown in Table 5. The frequency of reporting will be specified in each utility's Interconnection Handbook or contracts/programs.

Operational Status Information
Operational State
Connection Status
Alarm Status
Operational Energy Storage Capacity

Table 5 – Operational Status Information

4.6.3 Alarms

Aggregators acting for its DERs and DER Clients SHALL have the capability to report the alarm data shown in Table 6 as they occur. For each alarm, there is a corresponding “return to normal” message. All alarms and their “return to normal” messages SHALL include a date-time stamp along with the alarm type. The frequency of reporting of alarms will be specified in each utility’s Interconnection Handbook or contracts/programs.

Alarms
Over Current
Over Voltage
Under Voltage
Over Frequency
Under Frequency
Voltage Imbalance
Current Imbalance
Local Emergency
Remote Emergency
Low Input Power
Phase Rotation

Table 6 – Alarms

By design, low-level equipment health and status information is not part of this interface as the utility does not have maintenance responsibility for these 3rd party operated systems.

5 IEEE 2030.5 Implementation and Requirements

This section defines IEEE 2030.5 implementation requirements and maps them to the CSIP and necessary Grid DER Support capabilities. The specific version of the protocol implemented SHALL be IEEE 2030.5-2018.

While it is assumed that the reader has a working knowledge of IEEE 2030.5 concepts and operations, a brief overview of IEEE 2030.5 is provided below to help the reader understand the detailed requirements.

5.1 Overview

5.1.1 High-Level Architecture

The IEEE 2030.5 protocol implements a client/server model based on a representational state transfer (REST) architecture utilizing the core HTTP methods of GET, HEAD, PUT, POST, and DELETE. In the REST model, the server hosts resources, and the client uses the HTTP methods to act on those resources. In this guide, the server is implemented at the utility communications gateway, and the client is then implemented at the Aggregator system or the SMCU or GFEMs (aka DER Clients). The client typically initiates the action, but the protocol does provide a lightweight subscription mechanism for the server to push resources to the client.

5.1.2 Resources and Function Sets

In IEEE 2030.5, a resource is a piece of information that a server exposes. These resources are used to represent aspects of a physical asset such as a smart inverter, attributes relating to the control of those assets (e.g., Volt-VAr curve), and general constructs for organizing those assets. IEEE 2030.5 resources are defined in the IEEE 2030.5 XML schema and access methods are defined in the Web Application Description Language (WADL). The schema is generally organized by Function Sets, a logical grouping of resources that cooperate to implement IEEE 2030.5 features. IEEE 2030.5 provides a rich set of Function Sets (e.g. Demand Response Load Control, Pricing, Messaging, Metering, etc.) to support a variety of use cases. This guide only requires the subset required to meet the required Grid DER support functionality. Utility servers, Aggregators, and DER Clients SHALL support all CSIP required IEEE 2030.5 function sets and resources in Table 7. Any additional function set specific requirements will be detailed in the sections below.

Function Set	Utility Server	Aggregator	DER Client
Time	MUST	MUST	MUST
Device Capability	MUST	MUST	MUST
End Device	MUST	MUST	MUST
FSA	MUST	MUST	MUST
DER	MUST	MUST	MUST
Response	MAY	MUST	MUST
Meter/Mirror Meter	MAY	MUST	MUST
Log Event	MUST	MUST	MUST
Subscription/Notification	MUST	MUST	MAY
Security	MUST	MUST	MUST

Table 7 - Required Function Sets and Resources

5.1.2.1 Time

The utility server uses the *Time* function set to distribute the current time to clients. Time is expressed in Coordinated Universal Time (UTC). Server event timing is based on this time resource. Unless otherwise specified in the utility's Implementation Handbook, coordination of this time and rates for updating this time SHALL conform to the requirements of IEEE 2030.5-2018.

5.1.2.2 Device Capability

The utility server uses the *DeviceCapability* resource to enumerate the function sets it supports. Clients use this function set to discover the location (URL) of the enumerated function sets.

5.1.2.3 End Device

The *EndDevice* function set provides interfaces to exchange information related to specific client or EndDevice. In the Direct DER Communications scenario, the SMCU and the GFEMS are *EndDevices*. In the Aggregator scenario, the Aggregator itself and all the DERs it manages are all *EndDevices*. The *EndDevice* resource can contain the *EndDevice:DER* resource. This resource contains links for DERs to report their status.

Aggregators acting for its DERs and DER Clients SHALL support the *EndDevice:DER* resources in Table 8 if the utility server makes them available.

***EndDevice:DER* Resource**
DERCapability
DERSettings
DERStatus
DERAvailability

Table 8 - Required EndDevice:DER Resources

5.1.2.4 Function Set Assignments (FSA)

The *FunctionSetAssignments* function set provides the mechanism to convey the grouping assignments of each DER. Grouping with FSAs can be implemented in many ways. Section 5.2.3.1 explains the required method for CSIP.

5.1.2.5 Distributed Energy Resource (DER)

The *DER* function set provides an interface to manage Distributed Energy Resources (*DER*). It is the primary function set for issuing DER controls.

5.1.2.5.1 DERProgram

The top-level resource for organizing DER controls is the *DERProgram*. In CSIP, the *DERProgram* is the resource used to convey controls to a group (i.e. each controllable group has an associated *DERProgram* for issuing controls to that group). The *DERProgram* contains an *mRID* to identify the resource, a *primacy* value to specify the priority of the *DERProgram* relative to other *DERPrograms*, and links to the *DefaultDERControl*, *DERControlList*, and the *DERCurveList*.

5.1.2.5.2 DefaultDERControl

Each *DERProgram* can have a *DefaultDERControl* that specifies the control values that are in effect in the absence of an active *DERControl*. *DefaultDERControls* can be used as “settings” for controls that are expected to be in effect for long durations without a definite end time (see section 4.4.1). The server can from time to time modify the *DefaultDERControls*. As with DERContols, clients periodically monitor the *DefaultDERControls* for changes (See Section 4.5).

5.1.2.5.3 DERControl

DERControls are events that specify the control value(s) to be used at a specific time for a specific duration. For example, a *DERControl* can specify a fixed power factor value be used at a certain time for a certain duration. When a *DERControl* is active, it overrides any existing *DefaultDERControl* for that specific control. *DERControls* are typically when the start time and stop time are known, but they can also be used when the end time is unknown. In this case, the *DERControl* is created with a very long duration and is cancelled or superseded when the control is no longer in effect.

5.1.2.5.4 DERCurve

DERCurves are a type of *DERControl* that define behavior based on an X-Y curve instead of a fixed value. *DERCurves* are used to define the behavior of a DER in response to a sensed grid condition. These curves are already embedded in the DER. The curve management functionality is used to update the set points on a specific curve and determine which curves are active at a point in time. While only one curve per curve-type can be active at the same time, different curve-types can be active at the same time if they do not conflict.

These curves are used to provide autonomous control in a predictable fashion. For example, assuming a volt-watt curve is active; if the inverter senses an over voltage situation a volt-watt curve would direct the inverter to lower its power output. Likewise, in an under-voltage situation, the same curve would likely direct the DER to increase its output (if possible).

5.1.2.6 Response

The *Response* function set provides the resources needed for the Aggregator or DER Client to report the status of *DERControl* events. Typical response information includes event reception, event start, event completion, event cancellation, etc.

5.1.2.7 Metering and Metering Mirror

The *Metering* function set provides the resources needed to support metrology measurements (e.g. real power, reactive power, voltage, etc.) The *Metering Mirror* function set provides the resources needed for an Aggregator or DER to send metrology data to the utility server.

5.1.2.8 Log Event

The *LogEvent* function set provides the resources needed for the Aggregator or DER to send alarms to the utility server.

5.1.2.9 Subscription/Notification

In the Aggregator scenario, the utility server provides resources to support subscriptions that allow rapid notification of a change in the resource. For example, the utility might change a Volt-VAr curve to reflect new tolerances based on the level of solar penetration on a feeder. The Aggregator implements a notification resource to receive the notifications sent by the utility server.

5.2 IEEE 2030.5 Requirements

Aggregators and DER Clients SHALL meet all IEEE 2030.5 mandatory requirements that are described in the standard for each of these sections/functions unless otherwise specified in utility Interconnection Handbooks or programs/contracts.

5.2.1 Security Requirements

HTTPS SHALL be used in all Direct and Aggregator Mediated communications scenarios.

IEEE 2030.5 defines a specific security framework (i.e. PKI infrastructure). However, this framework may not be compatible with the utility's security and IT infrastructure requirements. Therefore, the utility has the option of mandating the implementation and use of other security frameworks as defined in this section or in utility Interconnection Handbooks or programs/contracts (e.g., Site to Site VPNs, Cipher Suites, Certificates, etc.).

Aggregators and DER Clients SHALL support the required IEEE 2030.5 security framework and other security frameworks as required by the utility Interconnection Handbook or programs/contracts.

In all cases, including Aggregator Mediated communications scenarios and Direct Communication scenarios, the utility should specify the security framework based on its security and IT requirements. Possible PKI options include:

- Use of the IEEE 2030.5 or CSIP defined Certificate Authority (CA)
- Contracting with a commercial, third-party certificate authority chain to generate certificates
- Implementing their own private certificate authority chain to generate certificates
- Using self-signed certificates

5.2.1.1 TLS and Cipher Suites

TLS version 1.2 SHALL be used for all HTTPS transactions.

IEEE 2030.5 specifies a single cipher suite for HTTPS communications, namely: *TLS_ECDHE_ECDSA_WITH _AES_128_CCM_8* using the elliptic curve *secp256r1*. DER Clients SHALL support the IEEE 2030.5 cipher suite.

Aggregators SHALL also support the TLS_RSA_WITH_AES_256_CBC_SHA256 cipher suite or other cipher suites as specified by the utility Interconnection Handbook or programs/contracts.

5.2.1.2 Certificates

Certificates provide a mechanism to authenticate identities during the TLS handshake. All utility servers, Aggregators, and DER Clients SHALL have a valid certificate. A valid certificate is a certificate that conforms to the IEEE 2030.5 security framework or the security framework specified by the utility Interconnection Handbook or programs/contracts. A valid certificate SHALL be used in all IEEE 2030.5 TLS transactions.

Certificates for Aggregators and DER Clients SHALL only be provisioned upon completion of Conformance Testing.

Conformance testing and certificate provisioning and usage requirements will be specified in interconnections tariffs or utility Interconnection Handbooks or programs/contracts.

The GUID for both Aggregators and DERs SHALL be the IEEE 2030.5 Long Form Device Identifier (LFDI) which is based on the 20-byte SHA-256 hash of the device's certificate.

5.2.1.3 Authentication

The utility server, Aggregator, and DER Clients perform mutual authentication during the TLS handshake by exchanging and authenticating each other's certificate. The certificates specified by each utility SHALL be used for authentication. Authentication consists of verifying the integrity of the received certificate, checking the certificate has not expired, and verifying the certificate chains back to the correct root certificate authority. If authentication fails, the authenticator SHOULD issue a TLS Alert – Bad Certificate and close the connection.

5.2.1.4 Authorization

The utility server maintains a list of authorized devices (i.e. Aggregators and DERs) that are permitted to communicate with the server. For Aggregators and DER Clients, the authorization list SHALL be based on the LFDI since the SFDI may not provide enough collision protection for a large population (e.g. 1 million) of devices. If the device is not on the authorization list, the utility server SHOULD return an HTTP error code (e.g. 404 – Not Found) to terminate the transaction.

5.2.1.5 Access Control

Once a device (i.e. Aggregators or DER Clients) has been authenticated and authorized, it potentially has access to resources on the utility server. The utility server controls access to resources based on Access Control Lists (ACL). In theory, every resource on the utility server can have its own ACL. The utility SHALL establish the permissions for read, write, control, and other interactions, based on agreements on which interactions are authorized between each DER and the utility. For example, role-based access control may be used to establish these permissions for different roles.

Another aspect of Access Control is that the utility server may present different resource information based on the identity of the device making the request. This is done for both efficiency and/or privacy reasons.

When an Aggregator accesses the *EndDeviceList*, the utility server SHALL only present *EndDevices* that are under the management of that Aggregator. This means the utility server will present each Aggregator with a different *EndDeviceList*. This is done for both efficiency (Aggregators know that all DERs in the list are under its control), and privacy (Aggregators do not see any information related to DERs not under its control).

5.2.2 Commissioning and Identification of DER Requirements

IEEE 2030.5 uses two identifiers, both of which are hashes of the device certificate. The Short-Form Device Identifier (SFDI) is based on a 36-bit SHA256 hash of the device certificate and is expressed as 12 decimal digits. The Long-Form Device Identifier (LFDI) is the first 20 bytes of the SHA256 hash of the device certificate. In the Direct DER Communications scenario, the GUID used to identify the DER SHALL be the DER's LFDI.

In the Aggregator scenario, the DERs under the management of the Aggregator may not be IEEE 2030.5 devices – that is, they may not have a device certificate. In this case, the utility or the Aggregator will produce the LFDI (see section 5.2.1.2). In all cases, this identity and the associated LFDI are returned to the Aggregator for their uses in ensuring communications are routed correctly. Implementers SHOULD refer to each utility's Interconnection Handbook or programs/contracts for more information needed to establish the LFDI.

In the rare event that an LFDI collision is detected (i.e. two unique certificates or IDs hash to the same LFDI value), the utility will replace the certificates or IDs of the offending DERs. This may require returning the DERs to the manufacturer for certificate replacement. Note that the probability of a LFDI collision is infinitesimally small. It is much more likely the collision was caused by an accidental duplication of the certificate or ID.

5.2.3 Group Management Requirements

The primary function of groups is the ability to target DER controls to members of those groups. In IEEE 2030.5, DER controls exist within *DERPrograms*, so effectively, each controllable group has one associated *DERProgram* to receive the group's DER controls. Aggregators acting for its DERs and DER Clients SHALL track the *DERProgram* associated with that group. CSIP allows DERs to be a member of up to 15 groups. Aggregators acting for its DERs and DER Clients SHALL support up to 15 *DERPrograms* simultaneously for each DER.

Figure 3 shows an example grouping structure containing both topological and non-topological groups and the associated *DERPrograms* being tracked by two DERs.

Note that the utility server does not need to associate a *DERProgram* for each group. It only needs to associate a *DERProgram* to those groups it intends to send controls to. For example, if the utility does not intend to send controls at the Substation level, it does not need to create a *DERProgram* for the Substation groups. To minimize resource requirements for the utility server, Aggregators, and DERs, the utility server SHOULD only create *DERPrograms* for groups that are intended to receive controls.

5.2.3.1 FSA Architecture

In IEEE 2030.5, group membership is conveyed to an Aggregator or Directly Communicated to a DER using the *FunctionSetAssignmentsListLink* in the DER's *EndDevice* instance. This link points to a *FunctionSetAssignmentsList (FSAList)* that is usually unique to each DER. This list contains one or more *FunctionSetAssignments (FSA).* Each *FSA* can contain a link to a *DERProgramList* which contain link to a *DERProgram* the DER is required to track. Aggregators acting for its DERs and DER Clients SHALL traverse all these links and lists to discover all *DERPrograms* the DER is required to track.

The utility server can structure the FSAs to achieve its grouping objectives in many ways. CSIP has chosen the model shown in Figure 4 to promote efficiency and interoperability.

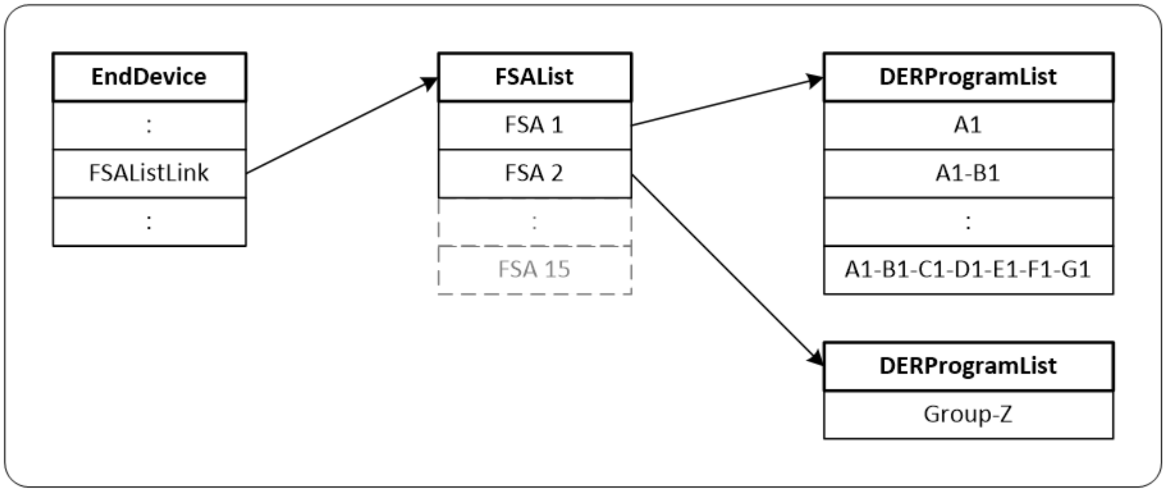

Figure 4 - CSIP FSA Model

In the above model, the *FSA 1* points to a *DERProgramList* that contains all *DERPrograms* for topology groups. *FSA 2* points to a *DERProgramList* containing a *DERProgram* for a non-topology group.

For each DER *EndDevice*, the utility server SHALL use one FSA to point to a *DERProgramList* containing all topology-based *DERPrograms* and MAY use additional FSAs to point to a *DERProgramList* containing non-topology-based *DERPrograms*. DER Clients SHALL be capable of supporting 15 FSAs.

For the CSIP Direct Communication scenario, the DER Client SHALL only receive function set assignments for a single energy connection point reflecting the aggregate capabilities of the plant at its point of common coupling with the utility.

5.2.4 DER Controls and DER Default Control Requirements

DER Clients SHALL use the IEEE 2030.5 mappings for the Grid DER Support Functions shown in Table 9.

Grid DER Support Functions	IEEE 2030.5 DERControls	IEEE 2030.5 DefaultDERControls
Low/High Voltage Ride Through	*opModLVRTMUSTTrip* *opModLVRTMAYTrip* *opModLVRTMomentaryCessation* *opModHVRTMUSTTrip* *opModHVRTMAYTrip* *opModHVRTMomentaryCessation*	*opModLVRTMUSTTrip* *opModLVRTMAYTrip* *opModLVRTMomentaryCessation* *opModHVRTMUSTTrip* *opModHVRTMAYTrip* *opModHVRTMomentaryCessation*
Low/High Frequency Ride Through	*opModLFRTMUSTTrip* *opModLFRTMAYTrip* *opModHFRTMUSTTrip* *opModHFRTMAYTrip*	*opModLFRTMUSTTrip* *opModLFRTMAYTrip* *opModHFRTMUSTTrip* *opModHFRTMAYTrip*
Ramp Rate Setting		*setGradW* *setSoftGradW*
Connect/Disconnect	*opModEnergize*	*opModEnergize*

Dynamic Volt-VAr	*opModVoltVar*	*opModVoltVar*
Fixed Power Factor Control	*opModFixedPF*	*opModFixedPF*
Real Power Output Limit Control	*opModMaxLimW*	*opModMaxLimW*
Volt-Watt Control	*opModVoltWatt*	*opModVoltWatt*
Frequency-Watt Control	*opModFreqWatt*	*opModFreqWatt*
Set Active Power Mode (in percentage of Max power) (in Watts)	*opModFixedW* *opModTargetW*	*opModFixedW* *opModTargetW*

Table 9 – Grid DER Support Functions to IEEE 2030.5 Control Mapping.

Usage of *DERControls* and *DefaultDERControls* was described in section 4.4.1. Note that the Ramp Rate Settings function maps to a *DefaultDERControl* and not a *DERControl*. This means they cannot be scheduled and can only be changed by changing the *DefaultDERControl* object.

5.2.4.1 Scheduling of Controls

DERControls are IEEE 2030.5 events and SHALL conform to all the event rules in Section 12.1.3 of IEEE 2030.5-2018.

Aggregators SHALL subscribe to each *DERProgramList* assigned to its DERs to discover changes in *DERProgram*:*primacy*.

Aggregators SHALL subscribe to the *DERControlList* of each *DERProgram* assigned to its DERs to discover new controls or changes to existing controls.

Aggregators SHALL subscribe to the *DefaultDERControl* of each *DERProgram* assigned to its DERs to discover changes to the default controls.

Unless otherwise specified in utility Interconnection Handbooks or programs/contracts to allow subscriptions, DER Clients SHALL poll to each *DERProgram* assigned to it to discover changes in *DERProgram*:*primacy*.

Unless otherwise specified in utility Interconnection Handbooks or programs/contracts to allow subscriptions, DER Clients SHALL poll to the *DERControlList* of each *DERProgram* assigned to it to discover new controls or changes to existing controls.

Unless otherwise specified in utility Interconnection Handbooks or programs/contracts to allow subscriptions, DER Clients SHALL poll to the *DefaultDERControl* of each *DERProgram* assigned to it to discover changes to the default controls.

The utility MAY optionally specify a recommended polling rate for these resources using the *DERProgramList:pollRate* resource. If the polling rate is specified, DERs SHOULD poll at this rate.

5.2.4.2 Prioritization

Prioritization of events is achieved using the *DERProgram:primacy* resource. Priority is assigned at the group (i.e. *DERProgram*) level.

Note that *DERControls* only conflict if they affect the same control. For example, if a power factor control issued at the Service Point level overlaps with a power factor control issued at the Feeder level, these controls are the same and conflict. In this case, the Service Point control with lower primacy takes precedence subject to the normal IEEE 2030.5 event rules. However, if a power factor control issued at the Service Point level overlaps with a limit real power control issued at the Feeder level, these controls are different and do not conflict. Both are in effect subject to the normal IEEE 2030.5 event rules.

5.2.5 Communication Interactions Requirements

In the Aggregator scenario, use of the IEEE 2030.5 subscription/notification function set is required to reduce unnecessary communications traffic.

Aggregators SHALL subscribe to the following lists:

- *EndDeviceList*
- *FunctionSetAssignmentsList* of each of the DERs under its management
- *DERProgramList* of each of the DERs under its management
- *DERControlList* of each of the DERs under its management
- *DefaultDERControls* of each of the DERs under its management

Aggregators MAY subscribe to other lists and resources, such as EndDevice, DERProgram, DERControl instances and others.

5.2.5.1 Monitor Data

Aggregators acting for its DERs and DER Clients SHALL use the IEEE 2030.5 Metering Mirror function set to report metrology data. Each of the monitoring data in Table 10 maps to a *MirrorMeterReading* with a *ReadingType* specifying the unit of measure (*uom*) and *dataQualifier*. The *dataQualifier* enumeration codes are shown in Table 11. For “instantaneous” data, *dataQualifier* need not be sent as the *ReadingType* already identifies the data as “instantaneous”.

Monitoring Data	ReadingType uom
Real (Active) Power	*38 (Watts)*
Reactive Power	*63 (VArs)*
Frequency	*33 (Hertz)*
Voltage	*29 (Voltage)*

Table 10 – Monitoring Data Mapping

Data Qualifiers	Qualifier Enumeration
Not Specified	*0*
Minimum	*9*
Maximum	*8*
Average	*2*

Table 11 - Data Qualifier Enumeration Codes

Aggregators acting for its DERs and DER Clients SHOULD post readings based on the *MirrorUsagePoint*:*postRate* resource.

5.2.5.2 Status Information

5.2.5.2.1 Ratings and Settings

Aggregators acting for its DERs and DER Clients SHALL be able to report the information shown in Table 12.

DER Data	Nameplate Mapping	Settings Mapping
Max rate of energy transfer received by the storage DER	*rtgMaxChargeRateW*	*setMaxChargeRateW*
Max rate of energy transfer delivered by the storage DER	*rtgMaxDischargeRateW*	*setMaxDischargeRateW*
Max apparent power	*rtgMaxVA*	*setMaxVA*
Max reactive power delivered by DER	*rtgMaxVar*	*setMaxVar*
Max reactive power received by DER	*rtgMaxVarNeg*	*setMaxVarNeg*
Max active power output	*rtgMaxW*	*setMaxW*
Min power factor when injecting reactive power	*rtgMinPFOverExcited*	*setMinPFOverExcited*
Min power factor when absorbing reactive power	*rtgMinPFUnderExcited*	*setMinPFUnderExcited*
Max energy storage capacity	*rtgMaxWh*	*setMaxWh*

Table 12 - Nameplate Ratings and Adjusted Settings Mapping

5.2.5.2.2 Operational Status Information

Aggregators acting for its DERs and DER Client SHALL be able to report the dynamic status information shown in Table 13. The frequency of reporting is specified by the utility.

Operational Status Information	DERStatus Mapping
Operational State	*operationalModeStatus* *inverterStatus*
Connection Status	*genConnectStatus*
Alarm Status	*alarmStatus*
Operational Energy Storage Capacity	*stateOfChargeStatus*

Table 13 – Operational Status Information Mapping

5.2.5.3 Alarms

The *LogEvent* function set is used to report the DER alarms using the *LogEvent:functionSet* enumeration of 11 (Distributed Energy Resource). DER Clients SHALL be able to report alarm data shown in Table 14. Alarms and their corresponding RTN "return to normal" messages are reported as they occur.

Alarms	LogEvent Name	LogEvent Code
Over Current	*DER_FAULT_OVER_CURRENT*	*0*
Over Current RTN	*DER_FAULT_OVER_CURRENT_RTN*	*1*
Over Voltage	*DER_FAULT_OVER_VOLTAGE*	*2*
Over Voltage RTN	*DER_FAULT_OVER_VOLTAGE_RTN*	*3*

Under Voltage	*DER_FAULT_UNDER_VOLTAGE*	*4*
Under Voltage RTN	*DER_FAULT_UNDER_VOLTAGE_RTN*	*5*
Over Frequency	*DER_FAULT_OVER_FREQUENCY*	*6*
Over Frequency RTN	*DER_FAULT_OVER_FREQUENCY_RTN*	*7*
Under Frequency	*DER_FAULT_UNDER_FREQUENCY*	*8*
Under Frequency RTN	*DER_FAULT_UNDER_FREQUENCY_RTN*	*9*
Voltage Imbalance	*DER_FAULT_VOLTAGE_IMBALANCE*	*10*
Voltage Imbalance RTN	*DER_FAULT_VOLTAGE_IMBALANCE_RTN*	*11*
Current Imbalance	*DER_FAULT_CURRENT_IMBALANCE*	*12*
Current Imbalance RTN	*DER_FAULT_CURRENT_IMBALANCE_RTN*	*13*
Local Emergency	*DER_FAULT_EMERGENCY_LOCAL*	*14*
Local Emergency RTN	*DER_FAULT_EMERGENCY_LOCAL_RTN*	*15*
Remote Emergency	*DER_FAULT_EMERGENCY_REMOTE*	*16*
Remote Emergency RTN	*DER_FAULT_EMERGENCY_REMOTE_RTN*	*17*
Low Input Power	*DER_FAULT_LOW_POWER_INPUT*	*18*
Low Input Power RTN	*DER_FAULT_LOW_POWER_INPUT_RTN*	*19*
Phase Rotation	*DER_FAULT_PHASE_ROTATION*	*20*
Phase Rotation RTN	*DER_FAULT_PHASE_ROTATION_RTN*	*21*

Table 14 – Alarms Mapping

Note the active alarms are available in the bit-mapped resource *DERStatus:alarmStatus* described in section 5.2.5.2.2.

5.3 Maintenance

It is assumed that the model of smart inverters will require maintenance over time. The managed population of smart inverters will most certainly grow as customers decide to install or upgrade DER systems. Likewise, utilities are likely to evolve their distribution systems requiring the changing of inverter grouping and management strategies.

This section describes how the model is updated and maintained over time via subscriptions to reflect changes. The following items are included:

- Inverters
- Groups
- Controls
- Programs
- Subscriptions

5.3.1 Maintenance of Inverters (EndDevice, EndDeviceList)

As part of the initial set up of the Utility Server, CSIP assumes the Aggregator has provided a list of inverters to the utility. The utility uses this list to construct and populate the initial EndDevice list for that Aggregator. Over time, this list will change as new inverters are added to the list and others are removed from the list.

5.3.1.1 Out-Of-Band Updates

The utility adds/removes EndDevice instances from the *EndDeviceList* using an out-of-band mechanism. If the Aggregator wants to add/remove an inverter from service, it communicates the request to the utility by some out-of-band mechanism (e.g. phone call, email, FTP, etc.). If the utility agrees to this request, the Utility Server adds/removes the corresponding *EndDevice* instance from the *EndDeviceList*. The Aggregator SHOULD subscribe to the *EndDeviceList* to receive notifications for any additions or changes to the list. The Aggregator SHOULD subscribe to each *EndDevice* instance under its control to receive notifications for any deletions of that instance.

5.3.1.2 In-Band Updates

The utility allows the Aggregator to directly add/remove *EndDevice* instances from the *EndDeviceList*. If the Aggregator wants to add a new inverter, it POSTs this proposed new instance to the *EndDeviceList*. If the Utility Server accepts and approves this addition, it returns a HTTP 201 – Created response along with the location of the newly created instance. Otherwise, the Utility Server returns an HTTP 4XX error response. If the Aggregator wants to delete an inverter, it tries to DELETE the corresponding *EndDevice* instance. If the Utility Server accepts and approves this deletion, it returns a HTTP 200 – OK response. Otherwise, the Utility Server returns an HTTP 4XX error response.

5.3.2 Maintenance of Groups (Function Set Assignments)

The utility may from time to time make changes to the system topology. This topology change typically results in a change in one or more inverter's group assignments. The group assignments for each inverter is located at the resource pointed to by the *FunctionSetAssignmentsListLink* within the *EndDevice* instance for that inverter. For every inverter under its control, the Aggregator SHOULD subscribe to the list pointed to by *EndDevice: FunctionSetAssignmentsListLink* to receive notifications for any changes in the inverter's group assignments.

5.3.3 Maintenance of Controls (DERControl, DERControlList)

The *DERControlList* hosts the scheduled and active *DERControl* events for the parent *DERProgram*. Since an inverter typically belongs to many groups, and each group may have one or more *DERPrograms*, an inverter or its controlling Aggregator needs to track many *DERControlLists*. For every inverter under its control, the Aggregator SHOULD subscribe to all of the *DERControlLists* associated with its FSA groups and *DERProgram* assignments to receive notifications for any new or changed *DERControl* events.

5.3.4 Maintenance of Programs (DERProgram, DERProgramList)

The *DERProgram* is a container for the *DERControlList*. It also contains some meta-data associated with the program. One important piece of meta-data is the *primacy* object, which determines the relative priority of the *DERControls* under this program. From time to time, the utility may want to adjust the e priority levels of *DERControls* by changing the primacy value. For every inverter under its control, the Aggregator SHOULD subscribe to all of the *DERPrograms* associated with its FSA groups to receive notifications for changes to the *DERProgram* meta-data. For every inverter under its control, the Aggregator SHOULD subscribe to all of the *DERProgramLists* associated with its FSA groups to receive notifications for additions, deletions, or changes to the list.

5.3.5 Maintenance of Subscriptions

Maintenance of various aspects of the CSIP model depends heavily on the proper operation of the Subscription/Notification function set. Maintenance of subscriptions is described previously for the IEEE 2030.5 Specification. In particular:

- The Aggregator Client SHOULD renew its subscriptions periodically (e.g. every 24 hours) with the Utility Server.
- The Aggregator Client SHOULD fall back to polling on perceived communications errors.

6 CSIP IEEE 2030.5 Implementation

6.1 Utility Server Operation

This section describes the operation of the IEEE 2030.5 utility server. For the most part, the operations of the utility sever are the same whether communicating with an Aggregator or a DER. Where there are differences, sub-sections for Aggregator operation vs. DER operation will be provided.

6.1.1 Server and Resource Discovery

6.1.2 Registration

In IEEE 2030.5, registration is the process of creating an *EndDevice* instance for the device being registered. The utility server SHOULD only register authorized devices. The utility server SHOULD NOT allow access to critical resources to un-registered devices. The utility SHOULD return an HTTP error code (e.g. 404 – Not Found) for un-authorized accesses by un-registered devices.

6.1.3 Out-Of-Band DER Registration

In the out-of-band registration model, the utility server creates *EndDevice* instances corresponding to authorized devices at start-up, prior to any client device connecting to the server. The utility server receives a list of authorized devices via an out-of-band process. For example, a utility may generate this list internally, or receive it from an Aggregator. Each utility may have their own procedure for creating and updating this list, and these procedures are outside the scope of CSIP.

When an authorized DER queries the utility server's *EndDeviceList*, the utility server SHOULD return an *EndDeviceList* containing 1 entry – the *EndDevice* instance of the authorized device making the query.

When an unauthorized DER queries the utility server's *EndDeviceList*, the utility server SHOULD return an HTTP error code (e.g. 404 – Not Found).

If a device tries to perform in-band registration by POSTing to the *EndDeviceList*, the server SHOULD return an HTTP error code (e.g. 403 – Forbidden).

6.1.4 In-Band DER Registration

In the in-band registration model, the utility server has a list of authorized devices, but does not create and *EndDevice* instance for the authorized devices at start-up. The utility server receives a list of authorized devices via an out-of-band process. For example, a utility may generate this list internally, or receive it from an Aggregator. Each utility may have their own procedure for creating and updating this list, and these procedures are outside the scope of CSIP.

When an authorized DER initially queries the utility server's *EndDeviceList*, the utility server returns an empty *EndDeviceList*. The authorized DER tries to perform in-band registration by POSTing its *EndDevice* instance to the *EndDeviceList*. The utility server receives this POST and verifies the DER is in the authorized devices list. If it is, the utility server creates the *EndDevice* instance and returns an HTTP success code of 201 – Created, along with the location (i.e. URL) of the created *EndDevice* instance. Once created, any subsequent GETs of the *EndDeviceList* by this device returns an *EndDeviceList* containing a single entry – the *EndDevice* instance of the authorized device making the query.

If the device is not on the authorized list, the utility server SHOULD return an HTTP error code (e.g. 403 – Forbidden).

The in-band registration model may be more convenient for installers that have a pool of authorized devices in its inventory and only needs to register them when they are installed at the customer site.

6.1.5 Aggregator Registration

An Aggregator is a special *EndDevice* to the utility server. The utility will likely use the out-of-band method to register an Aggregator though in-band registration is possible. As described in 6.1.3 and 6.1.4, the utility has a list of all the authorized devices managed by the Aggregator and has created *EndDevice* instances for those devices.

When the Aggregator starts up, it initially queries the server's *EndDeviceList*. The server returns an *EndDeviceList* consisting of the Aggregator's instance as well as the instances of all the authorized DERs under the Aggregator's management. The Aggregator gets all the *EndDevice* instances to discover the group assignments of each *EndDevice* under its management.

6.1.6 Group Assignment of Inverters

The utility server is responsible for assigning an *EndDevice* to its groups. As a reminder, a group ultimately maps to a *DERProgram*. The *DERProgram* provides a reference to the controls and curves associated with a specific DER management program. The key components of the *DERProgram* are the *primacy* value (which sets the priority of this program) and the links to the *DefaultDERControl*, the *DERControlList*, and the *DERCurveList*. The utility server creates a *DERProgram* for each group in the system. Figure 5 shows an example DERProgram for the System A1 group of Figure 3.

```
<DERProgram href="/sep2/A1/derp/1">
  <mRID>B01000000</mRID>
  <description>SYS-A1</description>
  <ActiveDERControlListLink href="/sep2/A1/derp/1/actderc" all="0"/>
  <DefaultDERControlLink href="/sep2/A1/derp/1/dderc"/>
  <DERControlListLink href="/sep2/A1/derp/1/derc" all="0"/>
  <DERCurveListLink href="/sep2/A1/derp/1/dc" all="0"/>
  <primacy>89</primacy>
</DERProgram>
```

Figure 5 - Example DERProgram

Once all the *DERPrograms* have been created, each *EndDevice* needs to be assigned to their appropriate groups. CSIP uses one FSA for topology groups and a second FSA for non-topology groups as described in section 5.2.3.1 to create the group assignments for each *EndDevice*. A *DERProgramList* is created for each of the FSAs, and group assignment simply consists of populating these lists with all the topology and non-topology *DERPrograms* (i.e. group assignments) for that *EndDevice*. An example of the *DERProgramLists* for Inverter-A of Figure 3 is shown in Figure 6.

```xml
<DERProgramList href="/sep2/edev/1/derpF1" subscribable="1" pollRate="3600" all="7" results="7"
xmlns="urn:ieee:std:2030.5:ns">
  <DERProgram href="/sep2/A1/derp/1">
    <mRID>B01000000</mRID>
    <description>SYS-A1</description>
    <ActiveDERControlListLink href="/sep2/A1/derp/1/actderc" all="0"/>
    <DefaultDERControlLink href="/sep2/A1/derp/1/dderc"/>
    <DERControlListLink href="/sep2/A1/derp/1/derc" all="0"/>
    <DERCurveListLink href="/sep2/A1/derp/1/dc" all="0"/>
    <primacy>89</primacy>
  </DERProgram>
  <DERProgram href="/sep2/A1-B1/derp/1">
    <mRID>B01100000</mRID>
    <description>SUBTX-A1-B1</description>
    <ActiveDERControlListLink href="/sep2/A1-B1/derp/1/actderc" all="0"/>
    <DefaultDERControlLink href="/sep2/A1-B1/derp/1/dderc"/>
    <DERControlListLink href="/sep2/A1-B1/derp/1/derc" all="0"/>
    <DERCurveListLink href="/sep2/A1-B1/derp/1/dc" all="0"/>
    <primacy>88</primacy>
  </DERProgram>
    :    :    :    :    :    :    :    :    :    :    :
    :    :    :    :    :    :    :    :    :    :    :
  <DERProgram href="/sep2/A1-B1-C1-D1-E1-F1/derp/1">
    <mRID>B01111110</mRID>
    <description>TRANS-A1-B1-C1-D1-E1-F1</description>
    <ActiveDERControlListLink href="/sep2/A1-B1-C1-D1-E1-F1/derp/1/actderc" all="0"/>
    <DefaultDERControlLink href="/sep2/A1-B1-C1-D1-E1-F1/derp/1/dderc"/>
    <DERControlListLink href="/sep2/A1-B1-C1-D1-E1-F1/derp/1/derc" all="0"/>
    <DERCurveListLink href="/sep2/A1-B1-C1-D1-E1-F1/derp/1/dc" all="0"/>
    <primacy>84</primacy>
  </DERProgram>
  <DERProgram href="/sep2/A1-B1-C1-D1-E1-F1-G1/derp/1">
    <mRID>B01111111</mRID>
    <description>SP-A1-B1-C1-D1-E1-F1-G1</description>
    <ActiveDERControlListLink href="/sep2/A1-B1-C1-D1-E1-F1-G1/derp/1/actderc" all="0"/>
    <DefaultDERControlLink href="/sep2/A1-B1-C1-D1-E1-F1-G1/derp/1/dderc"/>
    <DERControlListLink href="/sep2/A1-B1-C1-D1-E1-F1-G1/derp/1/derc" all="0"/>
    <DERCurveListLink href="/sep2/A1-B1-C1-D1-E1-F1-G1/derp/1/dc" all="0"/>
    <primacy>83</primacy>
  </DERProgram>
</DERProgramList>

<DERProgramList href="/sep2/edev/1/derpF2" subscribable="1" pollRate="3600" all="1" results="1"
xmlns="urn:ieee:std:2030.5:ns">
  <DERProgram href="/sep2/Z/derp/1">
    <mRID>B10000000</mRID>
    <description>Group-Z</description>
    <ActiveDERControlListLink href="/sep2/Z/derp/1/actderc" all="0"/>
    <DefaultDERControlLink href="/sep2/Z/derp/1/dderc"/>
    <DERControlListLink href="/sep2/Z/derp/1/derc" all="0"/>
    <DERCurveListLink href="/sep2/Z/derp/1/dc" all="0"/>
    <primacy>81</primacy>
  </DERProgram>
</DERProgramList>
```

Figure 6 - Example DERProgramLists for Inverter-A

The utility server then creates the *FunctionSetAssignmentsList* to link the *DERProgramList* with the appropriate *EndDevice*. An example *FunctionSetAssignmentsList* for Inverter-A shown in Figure 7. The *EndDevice* instance for the device will contain a link to this *FunctionSetAssignmentsList*.

```xml
<FunctionSetAssignmentsList href="/sep2/edev/1/fsa" subscribable="1" all="2" results="2"
xmlns="urn:ieee:std:2030.5:ns">
  <FunctionSetAssignments href="/sep2/edev/1/fsa/1">
    <mRID>A1000000</mRID>
    <description>Inverter-A Topology FSA</description>
    <DERProgramListLink href="/sep2/edev/1/derpF1" all="7"/>
    <TimeLink href="/sep2/tm"/>
  </FunctionSetAssignments>
  <FunctionSetAssignments href="/sep2/edev/1/fsa/2">
    <mRID>A1000001</mRID>
    <description>Inverter-A Non-Topology FSA</description>
    <DERProgramListLink href="/sep2/edev/1/derpF2" all="1"/>
    <TimeLink href="/sep2/tm"/>
  </FunctionSetAssignments>
</FunctionSetAssignmentsList>
```

Figure 7 - Example FunctionSetAssignmentsList for Inverter-A

6.1.7 EndDevice Creation

The utility server then creates the *EndDevice* instance that links to the appropriate *FunctionSetAssignmentsList*. An example *EndDevice* instance for Inverter-A is shown in Figure 8. Note that the *FunctionSetAssignmentsListLink* points to the list in Figure 7.

```xml
<EndDevice href="/sep2/edev/1">
  <lFDI>12a4a4b406ad102e7421019135ffa2805235a21c</lFDI>
  <LogEventListLink href="/sep2/edev/1/log" all="0"/>
  <sFDI>050044792964</sFDI>
  <changedTime>1514836800</changedTime>
  <enabled>1</enabled>
  <FunctionSetAssignmentsListLink href="/sep2/edev/1/fsa" all="2"/>
  <RegistrationLink href="/sep2/edev/1/rg"/>
</EndDevice>
```

Figure 8 - Example EndDevice for Inverter-A

If the utility is using an Aggregator, an *EndDevice* instance for the Aggregator also needs to be created. There are a couple of differences between an Aggregator *EndDevice* instance and a DER *EndDevice* instance. First, the Aggregator uses IEEE 2030.5 subscription/notification, so it needs a *SubscriptionListLink*, and second, the Aggregator itself is not a DER, so it does not need a *FunctionSetAssignmentsListLink*. An example of an Aggregator *EndDevice* instance is shown in Figure 9.

```xml
<EndDevice href="/sep2/edev/1000">
  <lFDI>9dfdd56f6128cdc894a1e42c690cab197184a8e9</lFDI>
  <LogEventListLink href="/sep2/edev/1000/log" all="0"/>
  <sFDI>424105305501</sFDI>
  <changedTime>1514837000</changedTime>
  <enabled>1</enabled>
  <RegistrationLink href="/sep2/edev/1000/rg"/>
  <SubscriptionListLink href="/sep2/edev/1000/sub" all="0"/>
</EndDevice>
```

Figure 9 - Example EndDevice for Aggregator

6.1.7.1 EndDevice Access

DERs or Aggregators get access to *EndDevices* through an *EndDeviceListLink* that is available via the server's *DeviceCapability* resource. The utility server should return a custom *EndDeviceList* for each device making the request. If the querying device is a DER, the server should return an *EndDeviceList* consisting of a single entry – the *EndDevice* instance of the requesting DER. An example of a DER *EndDeviceList* is shown in Figure 10.


```xml
<EndDeviceList href="/sep2/edev" subscribable="1" pollRate="86400" all="1" results="1"
xmlns="urn:ieee:std:2030.5:ns">
  <EndDevice href="/sep2/edev/1">
    <DERListLink href="/sep2/edev/1/der" all="0"/>
    <lFDI>12a4a4b406ad102e7421019135ffa2805235a21c</lFDI>
    <LogEventListLink href="/sep2/edev/1/log" all="0"/>
    <sFDI>050044792964</sFDI>
    <changedTime>1514836800</changedTime>
    <enabled>1</enabled>
    <FunctionSetAssignmentsListLink href="/sep2/edev/1/fsa" all="1"/>
    <RegistrationLink href="/sep2/edev/1/rg"/>
  </EndDevice>
</EndDeviceList>
```


Figure 10 - Example EndDeviceList for Inverter-A

If the querying device is an Aggregator, the server should return an *EndDeviceList* consisting of an entry for the Aggregator and entries for each DER under the Aggregator's control. An example of an Aggregator *EndDeviceList* is shown in Figure 11.


```xml
<EndDeviceList href="/sep2/edev" subscribable="1" all="3" results="3"
xmlns="urn:ieee:std:2030.5:ns">
  <EndDevice href="/sep2/edev/1000">
    <lFDI>9dfdd56f6128cdc894a1e42c690cab197184a8e9</lFDI>
    <LogEventListLink href="/sep2/edev/1000/log" all="0"/>
    <sFDI>424105305501</sFDI>
    <changedTime>1514837000</changedTime>
    <enabled>1</enabled>
    <RegistrationLink href="/sep2/edev/1000/rg"/>
    <SubscriptionListLink href="/sep2/edev/1000/sub" all="0"/>
  </EndDevice>
  <EndDevice href="/sep2/edev/1">
    <DERListLink href="/sep2/edev/1/der" all="0"/>
    <lFDI>12a4a4b406ad102e7421019135ffa2805235a21c</lFDI>
    <LogEventListLink href="/sep2/edev/1/log" all="0"/>
    <sFDI>050044792964</sFDI>
    <changedTime>1514836800</changedTime>
    <enabled>1</enabled>
    <FunctionSetAssignmentsListLink href="/sep2/edev/1/fsa" all="2"/>
    <RegistrationLink href="/sep2/edev/1/rg"/>
  </EndDevice>
  <EndDevice href="/sep2/edev/2">
    <DERListLink href="/sep2/edev/2/der" all="0"/>
    <lFDI>5509d69f8b353595206ad71b47e27906318ea367</lFDI>
    <LogEventListLink href="/sep2/edev/2/log" all="0"/>
    <sFDI>228273300409</sFDI>
    <changedTime>1514836800</changedTime>
    <enabled>1</enabled>
    <FunctionSetAssignmentsListLink href="/sep2/edev/2/fsa" all="2"/>
    <RegistrationLink href="/sep2/edev/2/rg"/>
  </EndDevice>
</EndDeviceList>
```


Figure 11 - Example EndDeviceList for Aggregator

If the querying device is not authorized, the server should return an HTTP error code of (404 – Not Found) or (403 – Forbidden).

6.1.8 DER Control Management

The utility server sends controls to groups by creating a new *DERControl* and adding it to the *DERControlList* of the group's *DERProgram*. CSIP uses three types of controls: immediate controls, default-only controls, and curve controls.

6.1.8.1 Immediate Controls

An immediate control is an IEEE 2030.5 DER event used to change the value of a control at a scheduled time for a scheduled duration. Table 9 shows the list of CSIP immediate controls. Immediate controls may have an optional default value that is applied when there are no events active. Figure 12 shows an example *DERControlList* containing a *DERControl* with the *opModMaxLimW* immediate control along with the *opModVoltVar* Curve control.


```xml
<DERControlList href="/sep2/A1/derp/1/derc" subscribable="1" all="1" results="1"
xmlns="urn:ieee:std:2030.5:ns">
  <DERControl href="/sep2/A1/derp/1/derc/1" replyTo="/rsps/1/rsp" responseRequired="03">
    <mRID>D0000001</mRID>
    <description>Scheduled DERC</description>
    <creationTime>1514838000</creationTime>
    <EventStatus>
      <currentStatus>0</currentStatus>
      <dateTime>1514838000</dateTime>
      <potentiallySuperseded>false</potentiallySuperseded>
    </EventStatus>
    <interval>
      <duration>3600</duration>
      <start>1514926800</start>
    </interval>
    <DERControlBase>
      <opModMaxLimW>9500</opModMaxLimW>
      <opModVoltVar href="/sep2/dc/1"/>
    </DERControlBase>
  </DERControl>
</DERControlList>
```


Figure 12 - Example Immediate and Curve DER Control

6.1.8.2 Default-Only Controls

A default-only control is a control that cannot be scheduled – it only exists in the *DefaultDERControl* of the *DERProgram*. This type of control is intended for settings that have an indefinite duration and are not expected to change often. Table 9 shows the list of CSIP default-only controls. Figure 13 shows an example *DefaultDERControl* with the *setGradW* and *setSoftGradW* default-only controls.


```xml
<DefaultDERControl href="/sep2/A1/derp/1/dderc" xmlns="urn:ieee:std:2030.5:ns">
  <mRID>E0000001</mRID>
  <description>Default DERC</description>
  <DERControlBase>
    <opModMaxLimW>10000</opModMaxLimW>
    <opModVoltVar href="/sep2/dc/1"/>
    <setGradW>0</setGradW>
    <setSoftGradW>0</setSoftGradW>
  </DERControlBase>
</DefaultDERControl>
```


Figure 13 - Example of DefaultDERControl

6.1.8.3 Curve Controls

A curve control uses a series of (x, y) points to define the behavior of a dependent variable (y) based on the value of the independent variable (x). Table 9 shows the list of CSIP curve controls. A Curve control is an IEEE 2030.5 DER event which can be scheduled and can have an optional default curve that is applied when there are no events active. Figure 14 shows an example of *DERCurveList* containing two Volt-VAr curves. Figure 12 shows an example of *DERControl* scheduling Volt-VAr Curve 1.


```xml
<DERCurveList href="/sep2/A1/derp/1/dc" all="2" results="2" xmlns="urn:ieee:std:2030.5:ns">
  <DERCurve href="/sep2/dc/2">
    <mRID>C0000002</mRID>
    <description>Volt-VAr Curve 2</description>
    <creationTime>1514836801</creationTime>
    <CurveData> <xvalue>90</xvalue> <yvalue>60</yvalue> </CurveData>
    <CurveData> <xvalue>93</xvalue> <yvalue>0</yvalue> </CurveData>
    <CurveData> <xvalue>107</xvalue> <yvalue>0</yvalue> </CurveData>
    <CurveData> <xvalue>110</xvalue> <yvalue>-60</yvalue> </CurveData>
    <curveType>0</curveType>
    <xMultiplier>0</xMultiplier> <yMultiplier>0</yMultiplier>
    <yRefType>3</yRefType>
  </DERCurve>
  <DERCurve href="/sep2/dc/1">
    <mRID>C0000001</mRID>
    <description>Volt-VAr Curve 1</description>
    <creationTime>1514836800</creationTime>
    <CurveData> <xvalue>91</xvalue> <yvalue>61</yvalue> </CurveData>
    <CurveData> <xvalue>94</xvalue> <yvalue>1</yvalue> </CurveData>
    <CurveData> <xvalue>108</xvalue> <yvalue>1</yvalue> </CurveData>
    <CurveData> <xvalue>111</xvalue> <yvalue>-61</yvalue> </CurveData>
    <curveType>0</curveType>
    <xMultiplier>0</xMultiplier> <yMultiplier>0</yMultiplier>
    <yRefType>3</yRefType>
  </DERCurve>
</DERCurveList>
```


Figure 14 - Example DER Curve List

6.2 Aggregator Operations

This informative section describes the typical operations of an Aggregator. Keep in mind that CSIP only addresses the utility to Aggregator communications. Communications between the Aggregator and its DERs is outside the scope of this document.

6.2.1 Host and Service Discovery

For this section, discovery is the process whereby the Aggregator obtains enough information to get the utility server's *DeviceCapability* resource. There are many methods for the Aggregator to get this information, and the exact method to use is determined by the utility. Two methods are presented, but other methods could be used by mutual consent of the utility and Aggregator.

6.2.1.1 Out-Of-Band Discovery

In this model, the Aggregator is provisioned with all the information below by some out-of-band method (e.g. configuration file, webpage, user interface, etc.):

- The IP Address or DNS name of the utility server. If a DNS name is provided, the Aggregator performs a name resolution using unicast DNS.
- The HTTPS port to use. HTTP is not permitted for utility to Aggregator communications.
- The path to the *DeviceCapability* resource. This URL is the starting point for the Aggregator to discover the utility server's resources.

6.2.1.2 Unicast-DNS and DNS-SD

In this mode, the Aggregator is provisioned with the DNS name of the utility server. The Aggregator performs name resolution using unicast DNS to obtain the server's IP address. The Aggregator uses DNS-based service discovery (DNS-SD) to obtain the port, scheme (HTTP or HTTPS), and the path to the *DeviceCapability* resource. Refer to the IEEE 2030.5 specification for more details pertaining to DNS-SD.

6.2.2 Security, Authentication, and Authorization

Once the Aggregator has determined the location (URL) of the utility server's *DeviceCapability* resource, the Aggregator performs an HTTP GET of this resource. This action initiates a TLS handshake to establish a secure connection. Certificates are exchanged between the utility server and the Aggregator during the handshake. The utility server authenticates the Aggregator's certificate and verifies whether it is authorized.

Once the utility server authenticates and authorizes the Aggregator, it returns the contents of the DCAP resource with an HTTP response code of 200 – OK. If the Aggregator fails to authenticate or is not authorized, the utility server can abort the TLS connection by sending a TLS Alert message, or it can complete the TLS connection but return an HTTP response code of 403 – Forbidden.

6.2.3 Getting Server Resources

Once a secure connection has been established, the Aggregator can get resources from the utility server.

6.2.3.1 DeviceCapability

The *DeviceCapability* resource is the starting point for discovering resources on the server. It provides links to the entry point of function sets supported by the server. An example *DeviceCapability* resource is shown in Figure 15.

```xml
<DeviceCapability href="/sep2/dcap" xmlns="urn:ieee:std:2030.5:ns">
  <ResponseSetListLink href="/sep2/rsps" all="0"/>
  <TimeLink href="/sep2/tm"/>
  <UsagePointListLink href="/sep2/upt" all="0"/>
  <EndDeviceListLink href="/sep2/edev" all="3"/>
  <MirrorUsagePointListLink href="/sep2/mup" all="0"/>
</DeviceCapability>
```

Figure 15 - Example Aggregator DeviceCapability

6.2.3.2 EndDeviceList

Once the Aggregator obtains *DeviceCapability*, it then gets the *EndDeviceListLink* to get its *EndDevice* instance along with the *EndDevice* instances of all the DERs under its control. An example of this *EndDeviceList* was shown in Figure 11. An example of the Aggregator instance was shown in Figure 9, and an example of a DER instance was shown in Figure 8.

6.2.3.3 Subscriptions

The Aggregator instance contains the *SubscriptionListLink*. The Aggregator posts to this link to create subscriptions to resources for which it wants to receive notifications. The Aggregator subscribes to the following resources:

- *EndDeviceList* – to detect additions/deletions and enabling/disabling of DERs
- *DERProgramList* – to detect changes to the group assignments of each DER and to detect changes in the priority of each DERProgram
- *DERControlList* – to detect the creation of a DERControl and changes to its status
- *DefaultDERControl* – to detect changes in the default controls of each *DERProgram*

The Aggregator may subscribe to other resources if allowed by the server. Figure 16 shows and example subscription to the *EndDeviceList* resource requesting a list *limit* of up to 1 entries.

```
<Subscription xmlns="urn:ieee:std:2030.5:ns">
  <subscribedResource>/sep2/edev</subscribedResource>
  <encoding>0</encoding>
  <level>+S1</level>
  <limit>1</limit>
  <notificationURI>https://12.34.56.78:443/ntfy</notificationURI>
</Subscription>
```

Figure 16 - Example EndDeviceList Subscription

The Aggregator acts on behalf of all the DERs it manages. It is highly likely these DERs belong to many of the same groups, and there are significant overlaps in the resources the Aggregator is monitoring on behalf of the DERs. The Aggregator needs to keep track of these overlaps so that it only subscribes to a shared resource once.

6.2.3.4 Notifications

When a subscribed resource changes, the utility server posts a *Notification* to the Aggregator. For list resources, the *Notification* payload may contain entries from the list, depending on the *limit* setting of the requested by the Aggregator and the policy of the server. The Aggregator may need to perform additional list queries to get all changes to the list. Refer to the IEEE 2030.5 specification for details about subscription/notification behavior. An example of a *Notification* to an *EndDeviceList* subscription is shown in Figure 17.

```xml
<Notification xmlns="urn:ieee:std:2030.5:ns">
  <subscribedResource>/sep2/edev</subscribedResource>
  <status>0</status>
  <subscriptionURI>https://98.76.54.32/sep2/edev/1000/sub/1</subscriptionURI>
  <EndDeviceList href="/sep2/edev" subscribable="1" all="4" results="1">
    <EndDevice href="/sep2/edev/3">
      <DERListLink href="/sep2/edev/3/der" all="0"/>
      <lFDI>bdd7bb2babe673a3fc603d433125291971a88ac0</lFDI>
      <LogEventListLink href="/sep2/edev/3/log" all="0"/>
      <sFDI>509605116746</sFDI>
      <changedTime>1514837100</changedTime>
      <enabled>1</enabled>
      <FunctionSetAssignmentsListLink href="/sep2/edev/3/fsa" all="2"/>
      <RegistrationLink href="/sep2/edev/3/rg"/>
    </EndDevice>
  </EndDeviceList>
</Notification>
```


Figure 17 - Example EndDeviceList Notification

6.2.4 Acting on DER Controls

Once the Aggregator has retrieved and/or subscribed to the necessary DER resources, it waits for *Notifications* of new *DERControl* events. The new *DERControl* may be sent with the *Notification*. Otherwise, the Aggregator uses the *Notification* to trigger a GET of the *DERControlList* containing the new *DERControl*. At the start time of the event, the Aggregator activates the control for all the targeted DERs, and at the end of the event, the Aggregator de-activates the control returning the control to its default value, if a default was specified. How the Aggregator activates/de-activates the control for all the targeted DERs is outside the scope of CSIP.

If *Responses* are enabled for the *DERControl*, the Aggregator must post the appropriate *Responses* on behalf of each targeted DER.

6.2.5 Reporting DER Data

6.2.5.1 Reporting Monitor Data

For every DER under its control, the Aggregator reports monitor data described in 5.2.5.1. For each DER, the Aggregator creates a *MirrorUsagePoint* (*MUP*) instance for the DER by posting to the utility server's *MirrorUsagePointListLink* specified in the *DeviceCapability* resource. The location of this newly created instance is returned in the server response (e.g. /mup/1). An example of the contents of a *MUP* post for Inverter A is shown in Figure 18. This *MUP* post contains the definition of a *MirrorMeterReading* for reporting a Real Power set. Every 24 hours, the Aggregator posts a new Real Power reading set for each DER. An example of this reading set post is shown in Figure 19. The Aggregator makes similar posts for all type of metrology specified in Table 2.

```xml
<MirrorUsagePoint xmlns="urn:ieee:std:2030.5:ns">
  <mRID>5509D69F8B3535950000000000009182</mRID>
  <description>DER [Inverter A]</description>
  <roleFlags>49</roleFlags>
  <serviceCategoryKind>0</serviceCategoryKind >
  <status>1</status>
  <deviceLFDI>12a4a4b406ad102e7421019135ffa2805235a21c</deviceLFDI>
  <MirrorMeterReading>
    <mRID>5509D69F8B3535950001000000009182</mRID>
    <description>Real Power(W) Set</description>
    <ReadingType>
      <accumulationBehaviour>4</accumulationBehaviour>
      <dataQualifier>2</dataQualifier>
      <intervalLength>300</intervalLength>
      <powerOfTenMultiplier>0</powerOfTenMultiplier>
      <uom>38</uom>
    </ReadingType>
  </MirrorMeterReading>
</MirrorUsagePoint>
```


Figure 18 - Example MirrorUsagePoint POST


```xml
<MirrorMeterReading xmlns="urn:ieee:std:2030.5:ns">
  <mRID>5509D69F8B3535950001000000009182</mRID>
  <description>Real Power(W) Set</description>
  <MirrorReadingSet>
    <mRID>5509D69F8B3535950001100000009182</mRID>
    <timePeriod>
      <duration>86400</duration>
      <start>1514880000</start>
    </timePeriod>
    <Reading> <value>1</value> </Reading>
    <Reading> <value>2</value> </Reading>
    <Reading> <value>3</value> </Reading>
         :
    <Reading> <value>288</value> </Reading>
  </MirrorReadingSet>
</MirrorMeterReading>
```


Figure 19 - Example MirrorMeterReading POST

6.2.5.2 Reporting Status Information

For every DER under its control, the Aggregator reports status data described in 5.2.5.2. Figure 20 shows an example *DERCapability* post, Figure 21Figure 21 shows an example *DERSettings* post, and Figure 22 shows an example of a *DERStatus* post. For *DERCapability* and *DERSettings*, the Aggregator posts these resources at device start-up and on any changes. For *DERStatus*, the Aggregator posts at the rate specified in *DERList:pollRate*.

```xml
<DERCapability xmlns="urn:ieee:std:2030.5:ns">
  <modesSupported>3FFFFF</modesSupported>
  <rtgA> <multiplier>0</multiplier> <value>20</value> </rtgA>
  <rtgW> <multiplier>0</multiplier> <value>5000</value> </rtgW>
  <type>4</type>
</DERCapability>
```


Figure 20 - Example DERCapability POST


```xml
<DERSettings xmlns="urn:ieee:std:2030.5:ns">
  <setGradW>0</setGradW>
  <setMaxA> <multiplier>0</multiplier> <value>20</value> </setMaxA>
  <setMaxW> <multiplier>0</multiplier> <value>5000</value> </setMaxW>
  <updatedTime>1483257600</updatedTime>
</DERSettings>
```


Figure 21 - Example DERSettings POST


```xml
<DERStatus xmlns="urn:ieee:std:2030.5:ns">
  <genConnectStatus>
    <dateTime>1456345000</dateTime>
    <value>0</value>
  </genConnectStatus>
  <readingTime>1456345000</readingTime>
</DERStatus>
```


Figure 22 - Example DERStatus POST

6.2.5.3 Reporting Alarms

For every DER under its control, the Aggregator reports alarm data using the *LogEvent* function set described in 5.2.5.3 as they occur. Figure 23 shows an example *LogEvent* post for an over-current fault condition.


```xml
<LogEvent xmlns="urn:ieee:std:2030.5:ns">
  <createdDateTime>1514934000</createdDateTime>
  <details>Over-Current-Alarm</details>
  <functionSet>11</functionSet>
  <logEventCode>0</logEventCode>
  <logEventID>1</logEventID>
  <logEventPEN>40732</logEventPEN>
  <profileID>2</profileID>
</LogEvent>
```


Figure 23 – Example LogEvent POST

6.3 DER Device Operations

This informative section describes the typical operations of a DER Client when communicating directly with the utility server.

6.3.1 Host and Service Discovery

For this section, discovery is the process whereby the DER Client obtains enough information to get the utility server's *DeviceCapability* resource. There are many methods for the DER Client to get this information, and the exact method to use is determined by the utility. Two methods are presented, but other methods could be used by mutual consent of the utility and DER.

6.3.1.1 Out-Of-Band Discovery

In this model, the DER Client is provisioned with all the information below by some out-of-band method (e.g. configuration file, webpage, user interface, …):

- The IP Address or DNS name of the utility server. If a DNS name is provided, the DER performs a name resolution using unicast DNS.
- The HTTPS port to use. HTTP is not permitted for utility to DER communications.
- The path to the *DeviceCapability* resource. This URL is the starting point for the DER to discover the utility server's resources.

6.3.1.2 Unicast-DNS and DNS-SD

In this mode, the DER Client is provisioned with the DNS name of the utility server. The DER Client performs name resolution using unicast DNS to obtain the server's IP address. The DER Client uses DNS-based service discovery (DNS-SD) to obtain the port, scheme (HTTP or HTTPS), and the path to the *DeviceCapability* resource. Refer to the IEEE 2030.5 specification for more details pertaining to DNS-SD.

6.3.2 Security, Authentication, and Authorization

Once the DER Client has determined the location (URL) of the utility server's *DeviceCapability* resource, the DER Client performs an HTTP GET of this resource. This action initiates a TLS handshake to establish a secure connection. Certificates are exchanged between the utility server and the DER Client during the handshake. The utility server authenticates the DER Client's certificate and verifies whether it is authorized.

Once the utility server authenticates and authorizes the DER Client, it returns the contents of the *DeviceCapability* resource with an HTTP response code of 200 – OK. If the DER fails to authenticate or is not authorized, the utility server can abort the TLS connection by sending a TLS Alert message, or it can complete the TLS connection but return an HTTP response code of 403 – Forbidden.

6.3.3 Getting Server Resources

Once a secure connection has been established, the DER Client can get resources from the utility server.

6.3.3.1 DeviceCapability

The *DeviceCapability* resource is the starting point for discovering resources on the server. It provides links to the entry point of function sets supported by the server. An example *DeviceCapability* resource is shown in Figure 24. It is similar to the Aggregator version shown in Figure 15 except the length of the *EndDeviceList* is 1.

```xml
<DeviceCapability href="/sep2/dcap" xmlns="urn:ieee:std:2030.5:ns">
  <ResponseSetListLink href="/sep2/rsps" all="0"/>
  <UsagePointListLink href="/sep2/upt" all="0"/>
  <TimeLink href="/sep2/tm"/>
  <EndDeviceListLink href="/sep2/edev" all="1"/>
  <MirrorUsagePointListLink href="/sep2/mup" all="0"/>
</DeviceCapability>
```


Figure 24 - Example DER Client DeviceCapability

6.3.3.2 EndDeviceList

Once the DER Client obtains *DeviceCapability*, it then gets the *EndDeviceListLink* to get its *EndDevice* instance. An example of this *EndDeviceList* was shown in Figure 10Figure 11.

6.3.3.3 Polling for Resources

Once the DER Client gets its EndDevice instance, it finds its group assignments by following the *FunctionSetAssignmentsListLink*. From there, the DER finds the *DERProgramListLink*, the *DERProgramList*, all its assigned *DERPrograms*, *DERControlLists*, *DefaultDERControls, DERCurveLists*, etc. The DER Client periodically polls these resources at a rate specified by the *DERProgramList:pollRate* setting.

6.3.4 Acting on DER Controls

Once the DER Client has retrieved the necessary DER resources, it waits for new *DERControl* events. At the start time of the event, the DER Client activates the control, and at the end of the event, the DER Client de-activates the control returning the control to its default value, if a default was specified.

If *Responses* are enabled for the *DERControl*, the DER Client posts the appropriate *Responses.*

6.3.5 Reporting DER Data

6.3.5.1 Reporting Monitor Data

The DER Client reports monitor data described in 5.2.5.1. The DER Client creates a *MirrorUsagePoint* (*MUP*) instance by posting to the utility server's *MirrorUsagePointListLink* specified in the *DeviceCapability* resource. The location of this newly created instance is returned in the server response (e.g. /mup/1). An example of the contents of a *MUP* post for Inverter A is shown in Figure 19. This *MUP* post contains the definition of a *MirrorMeterReading* for reporting a Real Power set. Every 24 hours, the DER posts a new Real Power reading set. An example of this reading set post is shown in Figure 19. The DER makes similar posts for all type of metrology specified in Table 10.

6.3.5.2 Reporting Status Information

The DER Client reports status data described in 5.2.5.2. Figure 20 shows an example *DERCapability* post, Figure 21 shows an example *DERSettings* post, and Figure 22 shows an example of a *DERStatus* post. For *DERCapability* and *DERSettings*, the DER posts these resources at device start-up and on any changes. For *DERStatus*, the DER posts at the rate specified in *DERList:pollRate*.

6.3.5.3 Reporting Alarms

The DER reports alarm data using the *LogEvent* function set described in 5.2.5.3 as they occur. Figure 23 shows an example *LogEvent* post for an over-voltage fault condition.

7 Examples

7.1 Discovery, DeviceCapability, EndDeviceList

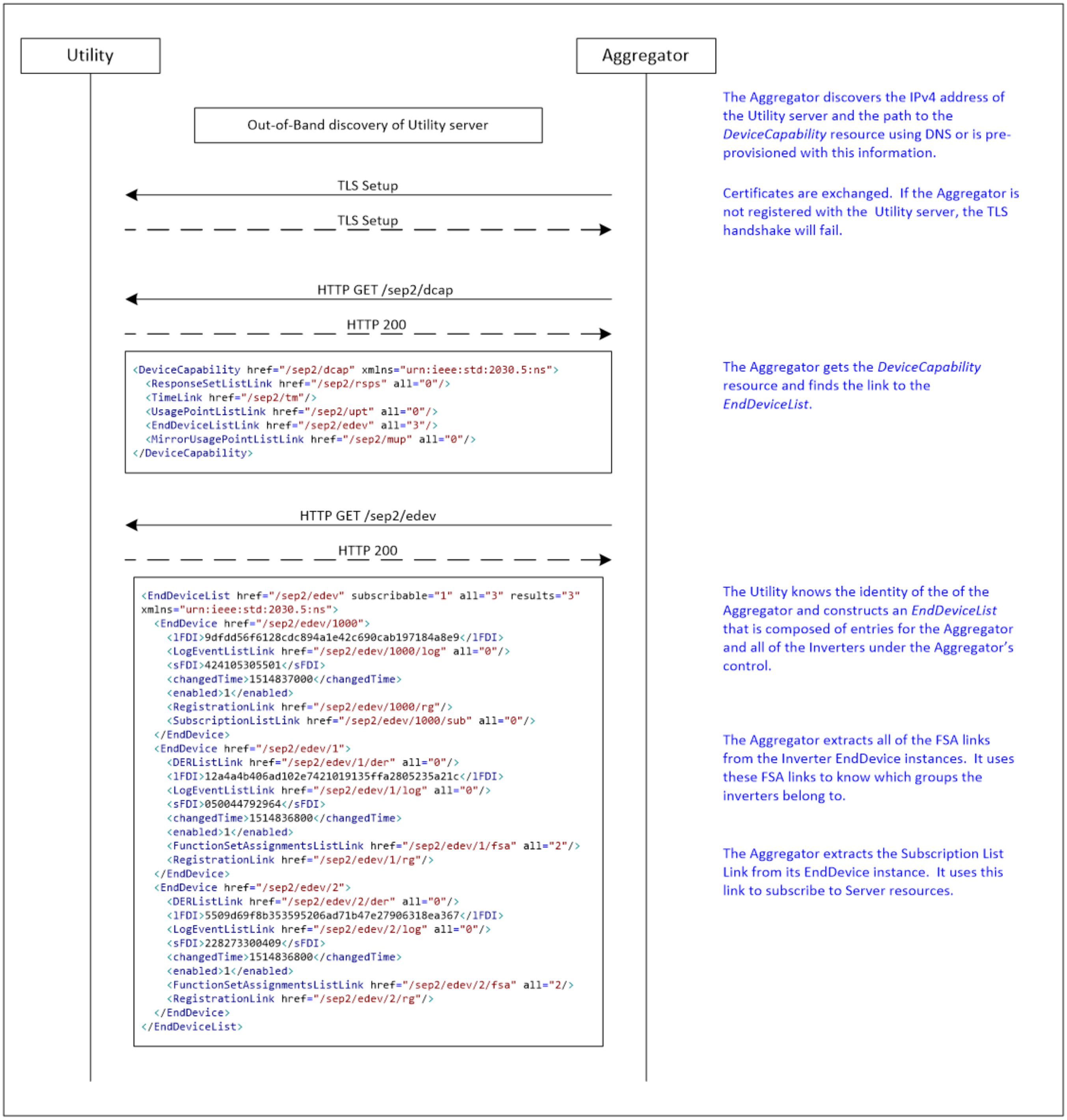

Figure 25 – Example: Discovery, DeviceCapability, EndDeviceList

7.2 FunctionSetAssignments

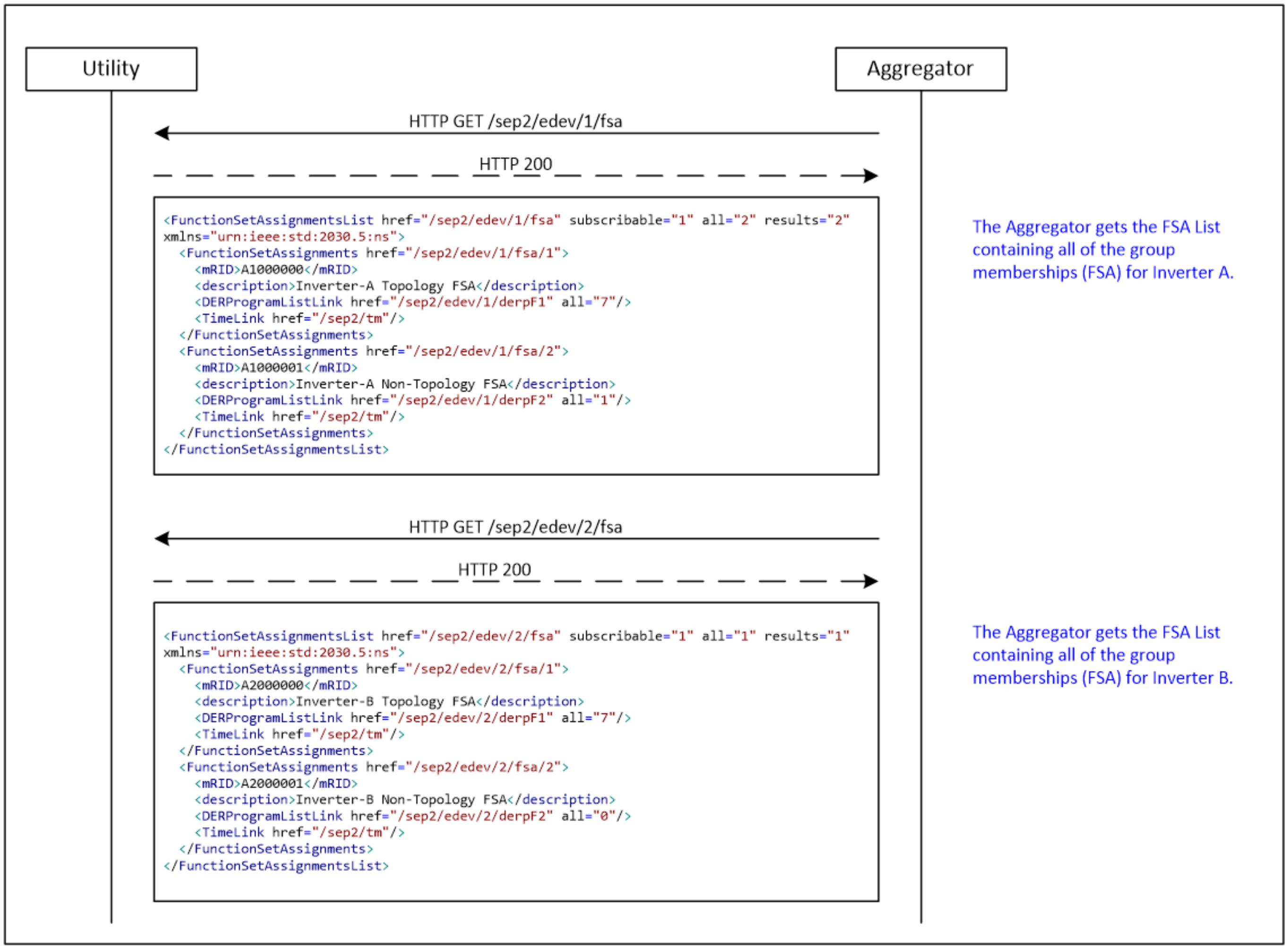

Figure 26 - Example: FunctionSetAssignments

7.3 DERProgramList, DERPrograms

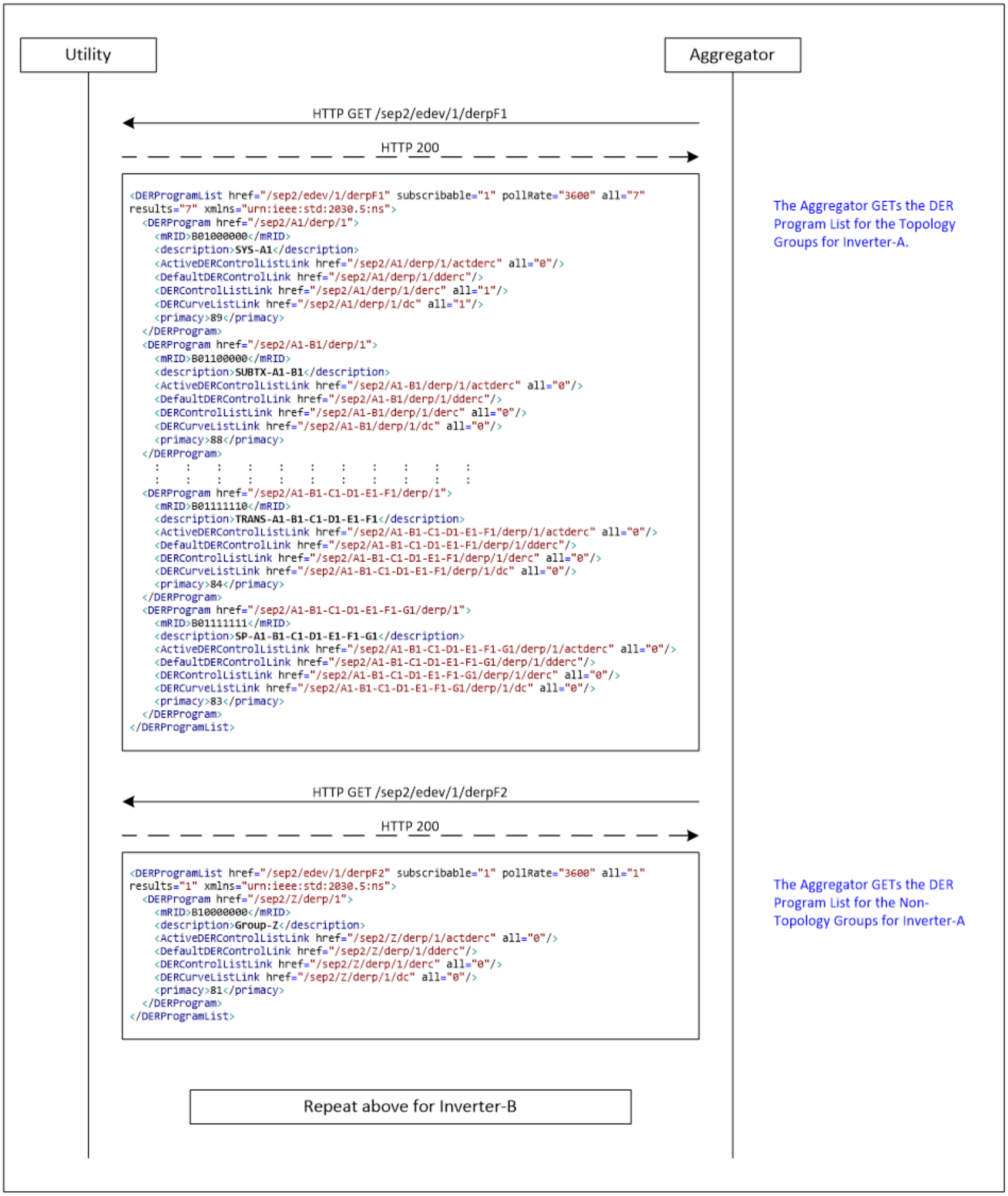

Figure 27 - Example: DERProgramList, DERPrograms

7.4 DERControlList, DERCurveList, DefaultDERControl

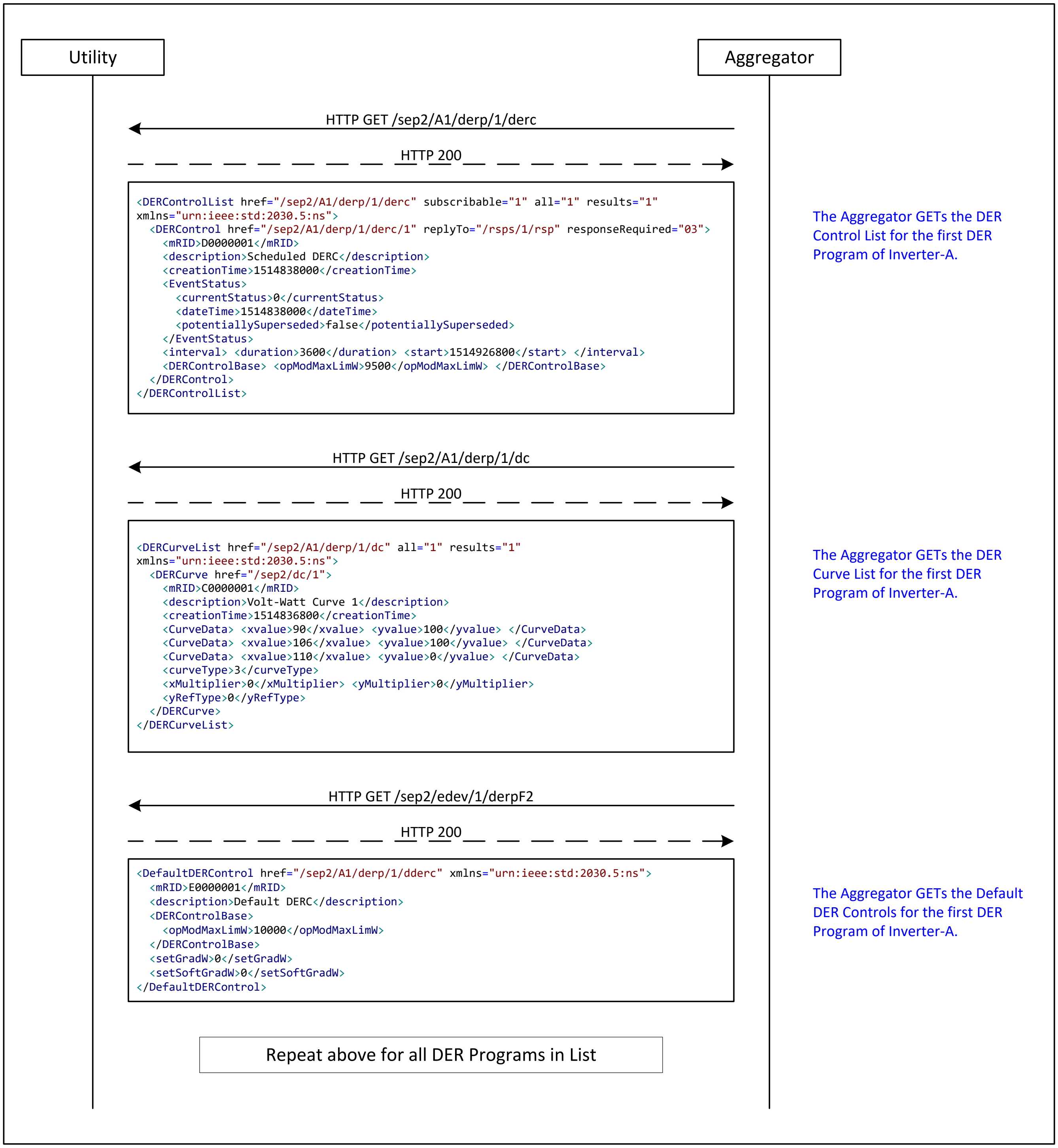

Figure 28 - Example: DERControlList, DERCurveList, DefaultDERControl

7.5 Subscription/Notification – EndDeviceList

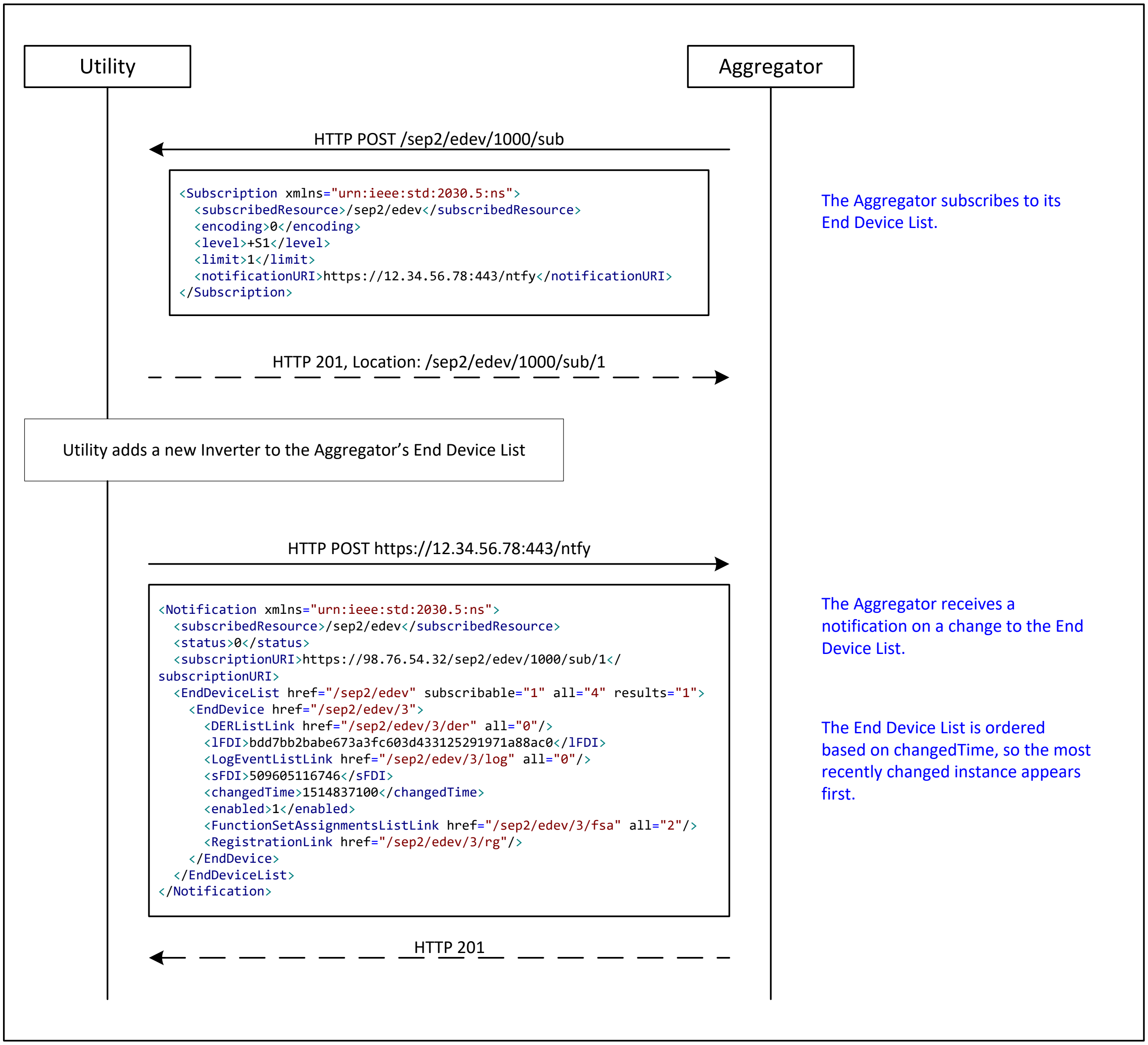

Figure 29 - Example: Subscription/Notification – EndDeviceList

7.6 Subscription/Notification – DERControlList

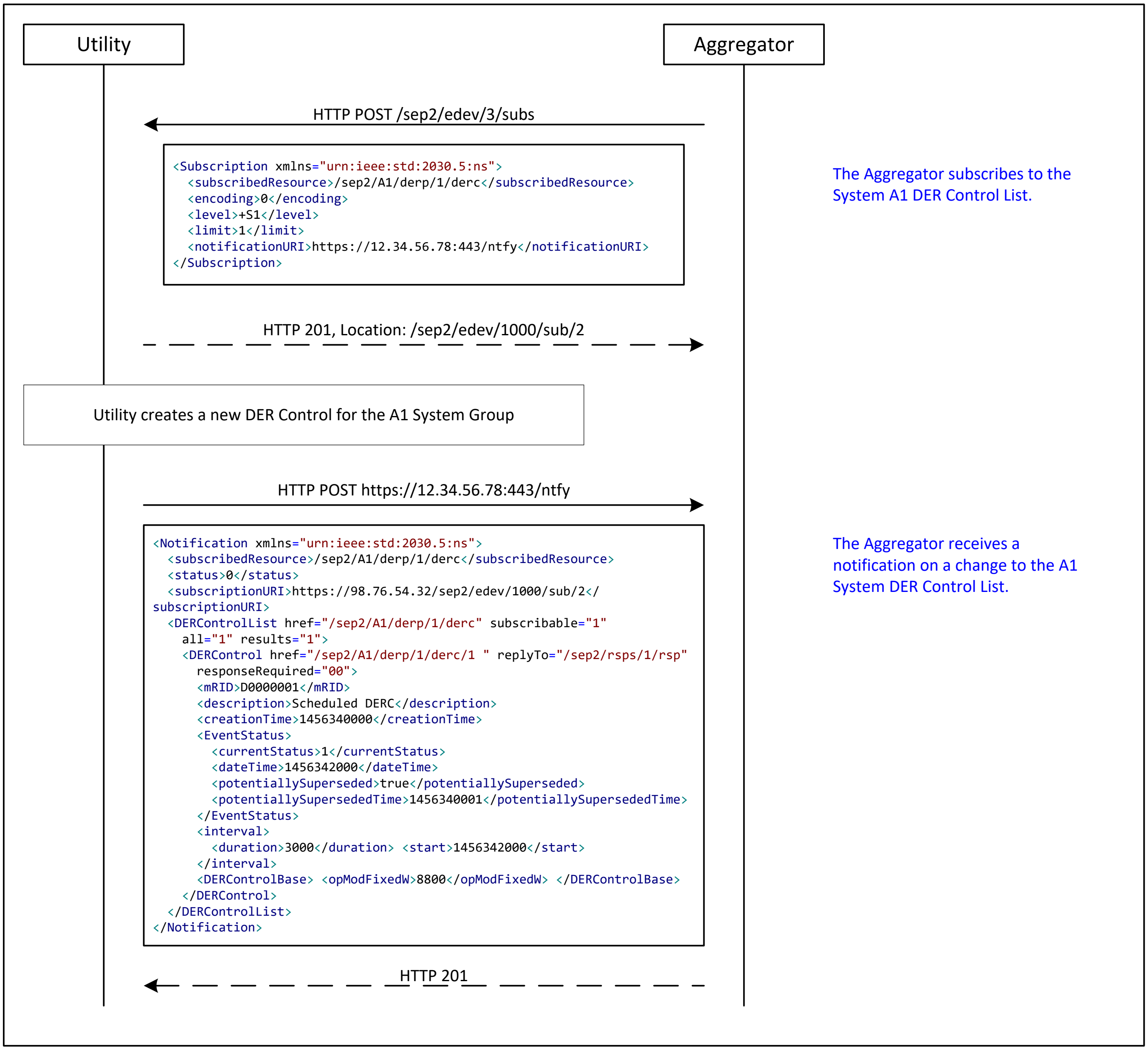

Figure 30 - Example: Subscription/Notification – DERControlList

7.7 Sending DER Status Information

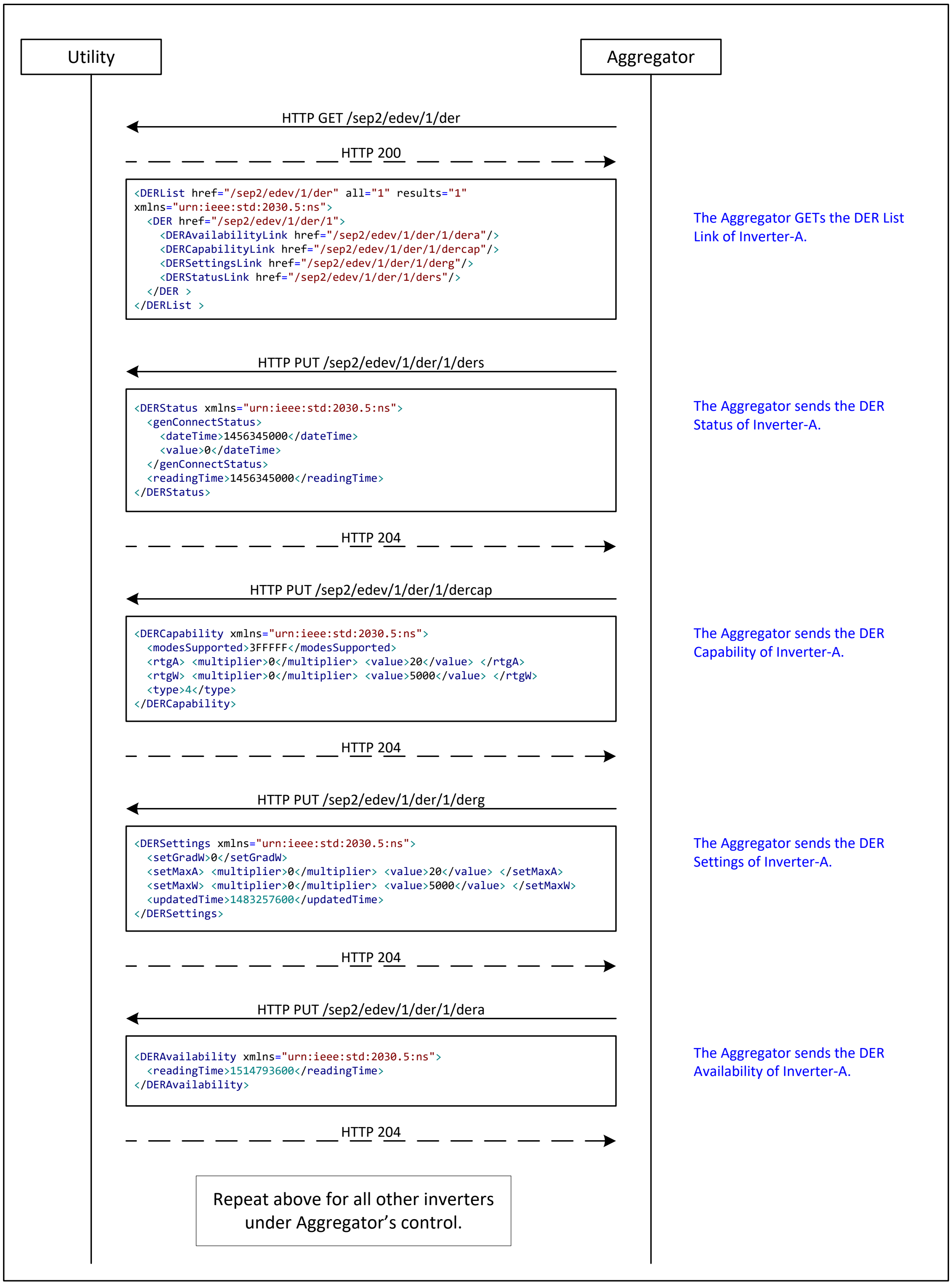

Figure 31 - Example: Sending DER Status Information

7.8 Sending Monitor Data

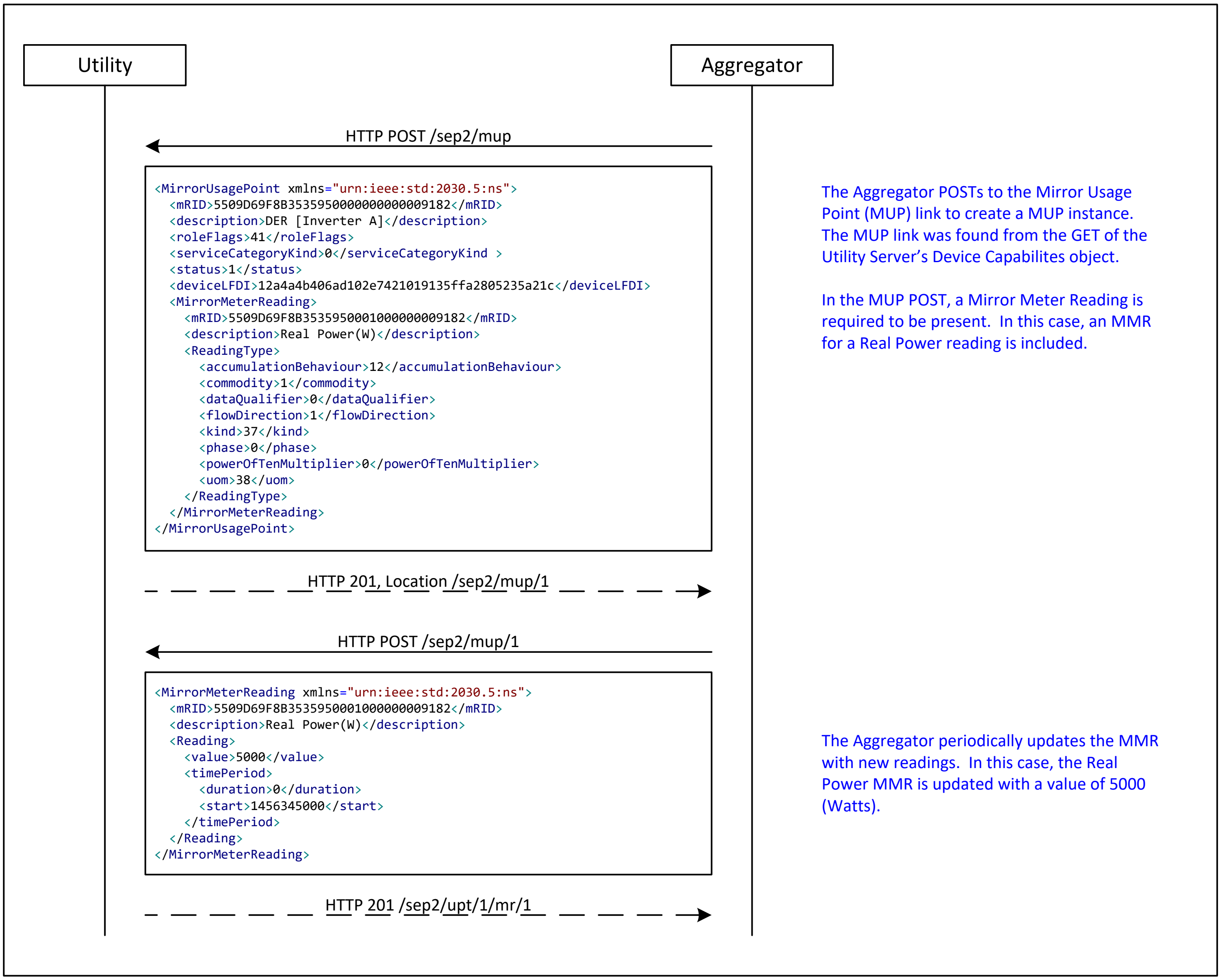

Figure 32 - Example: Sending Monitor Data

7.9 Sending Alarms

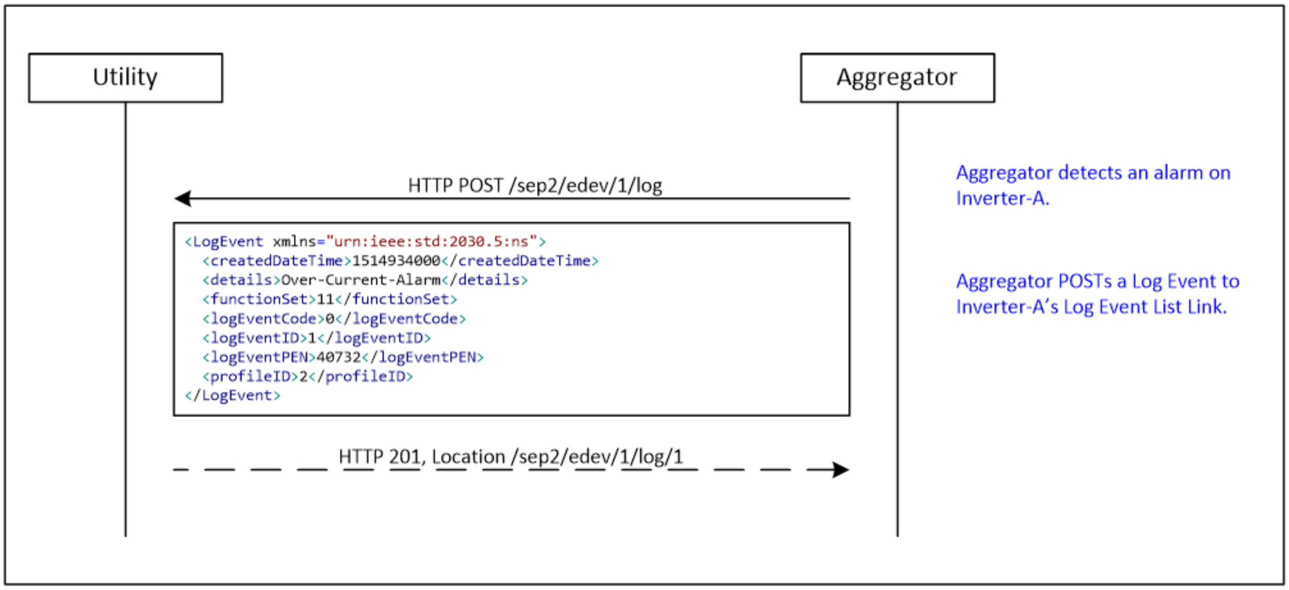

Figure 33 - Example: Sending Alarms

7.10 Event Prioritization

Aggregators acting for its DERs and DER Clients subscribe to or poll for new *DERControl* events from all the *DERProgram* groups they belong to. It is possible, and probably quite common, for a DER to receive overlapping events from different groups. How a DER handles these situations is determined by the *Event Rules and Guidelines* of section 12.1.3 of the IEEE 2030.5 specification. This document will highlight some of the important rules.

The priority of a *DERControl* is determined by the *primacy* setting of its containing *DERProgram* with a lower *primacy* value indicating higher priority. In the absence of any active events, the inverter executes the *DefaultDERControl* of the *DERProgram* with the highest priority (i.e. lowest *primacy* value).

When a DER receives overlapping *DERControl* events, the *DERControl* whose *DERProgram* has the higher priority (i.e. lower *primacy* value) takes precedence. The following examples describe two very similar overlapping event scenarios that only differ in when the DER receives the events. These examples assume the DER has discovered the *DERPrograms* and has subscribed to the *DERControlLists*. The process of discovering and subscribing to these resources was discussed earlier in this document.

In the first case, the DER receives both *DERControl* events prior to the start of either. In this case, the DER does not execute the lower priority (superseded) event. It only executes the higher priority event as shown in the figure below.

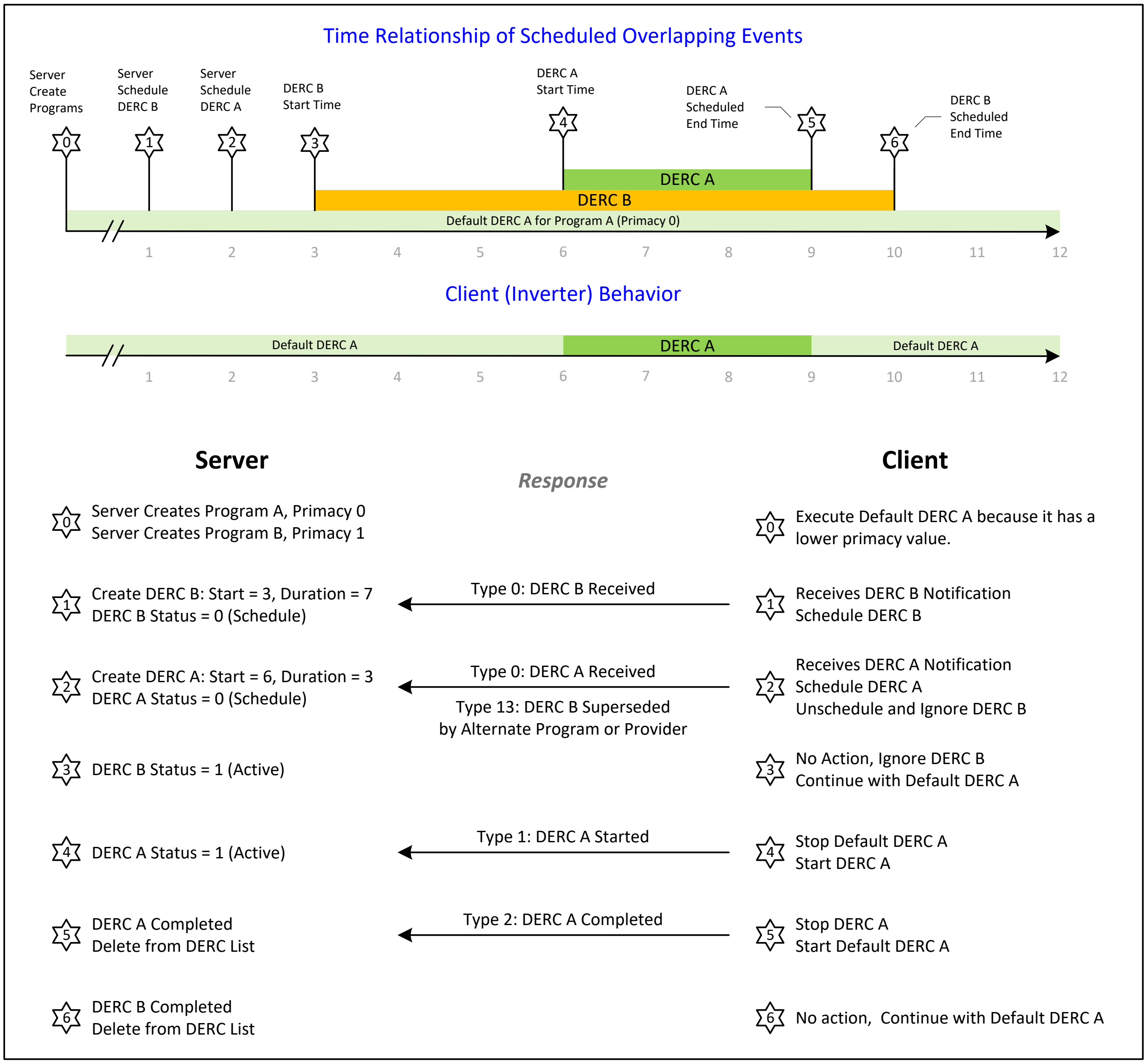

Figure 34 - Example: Supersede before Start of DERControl Event

In the second case, the DER receives the higher priority event while executing lower priority event. In this case, the DER continues with the lower priority event until the start time of the higher priority event. It then superseded the lower priority event and switches to executing the higher priority event as shown in the figure below.

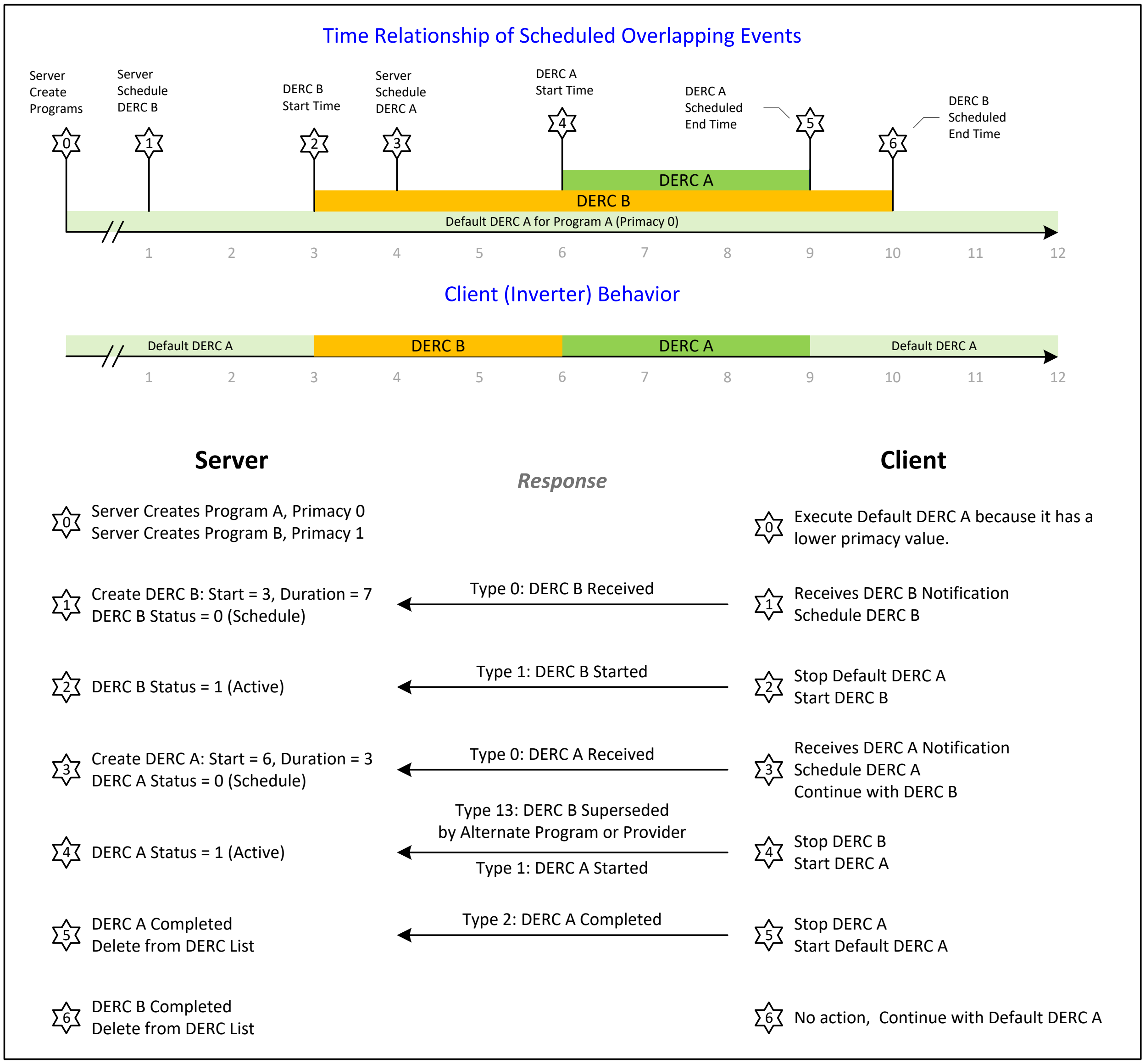

Figure 35 - Example: Supersede after Start of DERControl Event

Please note that in both scenarios the DER **DOES NOT** resume execution of the lower priority (superseded) event after completing the higher priority event.

Appendix A- Requirements

General CSIP Requirement	
ID	Requirements
G1.	Each DER Client SHALL connect to the utility in one and only one scenarios.
G2.	Although outside the scope of CSIP, security SHOULD be used in all non-IEEE 2030.5 interactions between the Aggregators, site hosts, GFEMS, and DERs and other entities receiving or transmitting DER related communications
G3.	For DER Clients that have an IEEE 2030.5 certificate, the GUID SHALL be derived from this certificate (see section 5.2.1.2).
G4.	Implementers SHALL refer to each utility's Interconnection Handbook for requirements related to the creation, use or management of this identifier.
G5.	Aggregators and DER Clients SHALL support IEEE 2030.5 based grouping and full lifecycle management of group relationships as defined within Section 5.2.3 and within each utility's Interconnection Handbook or program/contract requirements.
G6.	Autonomous functions' default settings SHALL be changeable via IEEE 2030.5 *DefaultDERControl* communications.
G7.	Modifications to default settings SHALL occur immediately upon receipt and have an indefinite duration.
G8.	Scheduling Autonomous and Advanced Power Values and Modes SHALL be controllable via IEEE 2030.5 *DERControl* events
G9.	Aggregators and DER Clients SHALL be responsible for assuring that all operations received from the utility are processed in the appropriate time sequence as specified by the utility.
G10.	An Aggregator acting for its DERs and DER Clients SHALL be able to store at least 24 scheduled DER control events for each DER.
G11.	In the absence of scheduled controls, DERs SHALL maintain a default control setting specified by interconnection tariffs or the utility Interconnection Handbook.
G12.	Should there be a loss of communications, DERs SHALL complete any scheduled event and then revert to default settings or as determined by the site host or tariffs/contracts.
G13.	When commanded in a manner where two or more operations are possibly in conflict, the interpreting system SHALL operate against the control operation which has the highest priority subject to the systems capability, contracts and self-protection requirements.
G14.	If avoidance of conflicting commands is not possible, the more recently received command SHOULD have precedence over the older command.
G15.	In either case, it SHALL be the responsibility of the aggregator or DER Client to decide how to handle these two simultaneous controls.
G16.	For Aggregators communications, notifications and call backs (subscription/notification) SHALL be used to limit system polling to the greatest extent practical.
G17.	To simplify communication requirements for Direct DER Communications scenarios, unless specified otherwise in utility Interconnection Handbooks or programs/contracts, all communications SHALL be initiated by the DER Client (i.e., client-side initiation).
G18.	In Direct DER communication scenarios, the DER Client SHALL initiate communications with the utility according to a pre-defined polling and posting interval to ensure the DER has up to date settings and the utility understands the operational state of the DER.
G19.	Unless specified in each utility's Interconnection Handbook, default polling and posting rates SHALL be as follows:

	-Polling of *DERControls* and *DefaultDERControls* (Direct DER Communication)– every 10 minutes -Posting monitoring information (Direct and Aggregator Mediated Communications)– every 5 minutes
G20.	For DERs with an external SMCU, the SMCU SHALL transfer the DER control to the generating facility within 10 minutes of receiving the control from the server.
G21.	For DERs with a GFEMS, the GFEMS SHALL transfer the DER control to the DERs within 10 minutes of receiving the control from the server.
G22.	For DERs mediated by Aggregators, the Aggregator SHALL transfer the DER control to the DERs within 15 minutes of receiving the control from the server.
G23.	Aggregators acting for its DERs and DER Clients SHALL have the capability to report the monitoring data in Table 2.
G24.	Aggregators acting for its DERs and DER Clients SHALL have the capability to include the data qualifiers in Table 3.
G25.	All measurement SHALL include a date-time stamp.
G26.	Unless otherwise specified in each utility's Interconnection Handbook or programs/contracts, Aggregators acting for its DERs and DER Clients SHALL report the monitoring data in Table 2 and MAY include the data qualifiers in Table 3
G27.	For those situations where the DERs cannot provide Monitoring Data, the Aggregator acting for its DERs and DER Clients SHALL not send the data.
G28.	Aggregators acting for its DERs and DER Clients SHALL have the capability to report the Nameplate Ratings and Adjusted Settings information shown in Table 4.
G29.	Nameplate Ratings and Adjusted Settings SHOULD be reported once at start-up and whenever there is a change in value.
G30.	Aggregators acting for its DERs and DER Clients SHALL have the capability to report the dynamic Operational Status Information shown in Table 5.
G31.	Aggregators acting for its DERs and DER Clients SHALL have the capability to report the alarm data shown in Table 6 as they occur.
G32.	All alarms and their "return to normal" messages SHALL include a date-time stamp along with the alarm type.
IEEE 2030.5 Protocol Requirements	
P1.	The specific version of the protocol implemented SHALL be IEEE 2030.5-2018.
P2.	Utility servers, Aggregators, and DER Clients SHALL support all CSIP required IEEE 2030.5 function sets and resources in Table 7.
P3.	Unless otherwise specified in the utility's Implementation Handbook, coordination of this time and rates for updating this time SHALL conform to the requirements of IEEE 2030.5-2018.
P4.	Aggregators acting for its DERs and DER Clients SHALL support the *EndDevice:DER* resources in Table 8 if the utility server makes them available.
P5.	Aggregators and DER Clients SHALL meet all IEEE 2030.5 mandatory requirements that are described in the standard for each of these sections/functions unless otherwise specified in utility Interconnection Handbooks or programs/contracts.
P6.	HTTPS SHALL be used in all Direct and Aggregated communications scenarios.
P7.	Aggregators and DER Clients SHALL support the required IEEE 2030.5 security framework and other security frameworks as required by the utility Interconnection Handbook or programs/contracts.
P8.	TLS version 1.2 SHALL be used for all HTTPS transactions.
P9.	DER Clients SHALL support the IEEE 2030.5 cipher suite.

P10.	Aggregators SHALL also support the TLS_RSA_WITH_AES_256_CBC_SHA256 cipher suite or other cipher suites as specified by the utility Interconnection Handbook or programs/contracts.
P11.	All utility servers, Aggregators, and DER Clients SHALL have a valid certificate.
P12.	A valid certificate SHALL be used in all IEEE 2030.5 TLS transactions.
P13.	Certificates for Aggregators and DER Clients SHALL only be provisioned upon completion of Conformance Testing.
P14.	The GUID for both Aggregators and DERs SHALL be the IEEE 2030.5 Long Form Device Identifier (LFDI) which is based on the 20-byte SHA-256 hash of the device's certificate.
P15.	The certificates specified by each utility SHALL be used for authentication.
P16.	If authentication fails, the authenticator SHOULD issue a TLS Alert – Bad Certificate and close the connection.
P17.	For Aggregators and DER Clients, the authorization list SHALL be based on the LFDI since the SFDI may not provide enough collision protection for a large population (e.g. 1 million) of devices.
P18.	If the device is not on the authorization list, the utility server SHOULD return an HTTP error code (e.g. 404 – Not Found) to terminate the transaction.
P19.	The utility SHALL establish the permissions for read, write, control, and other interactions, based on agreements on which interactions are authorized between each DER and the utility.
P20.	When an Aggregator accesses the *EndDeviceList*, the utility server SHALL only present *EndDevices* that are under the management of that Aggregator.
P21.	In the Direct DER Communications scenario, the GUID used to identify the DER Client SHALL be the DER's LFDI.
P22.	Implementers SHOULD refer to each utility's Interconnection Handbook or programs/contracts for more information needed to establish the LFDI.
P23.	Aggregators acting for its DERs and DER Clients SHALL track the *DERProgram* associated with that group.
P24.	Aggregators acting for its DERs and DER Clients SHALL support up to 15 *DERPrograms* simultaneously for each DER.
P25.	Aggregators acting for its DERs and DER Clients SHALL traverse all these links and lists to discover all *DERPrograms* the DER is required to track.
P26.	For each DER *EndDevice*, the utility server SHALL use one FSA to point to a *DERProgramList* containing all topology-based *DERPrograms* and MAY use additional FSAs to point to a *DERProgramList* containing non-topology-based *DERPrograms*.
P27.	DER Clients SHALL be capable of supporting 15 FSAs.
P28.	For the CSIP Direct Communication scenario, the DER Client SHALL only receive function set assignments for a single energy connection point reflecting the aggregate capabilities of the plant at its point of common coupling with the utility.
P29.	DER Clients SHALL use the IEEE 2030.5 mappings for the Grid DER Support Functions shown in Table 9.
P30.	*DERControls* are IEEE 2030.5 events and SHALL conform to all the event rules in Section 12.1.3 of IEEE 2030.5-2018.
P31.	Aggregators SHALL subscribe to each *DERProgramList* assigned to its DERs to discover changes in *DERProgram*:*primacy*.
P32.	Aggregators SHALL subscribe to the *DERControlList* of each *DERProgram* assigned to its DERs to discover new controls or changes to existing controls.
P33.	Aggregators SHALL subscribe to the *DefaultDERControl* of each *DERProgram* assigned to its DERs to discover changes to the default controls.

P34.	Unless otherwise specified in utility Interconnection Handbooks or programs/contracts to allow subscriptions, DER Clients SHALL poll to each *DERProgram* assigned to it to discover changes in *DERProgram*:*primacy*.
P35.	Unless otherwise specified in utility Interconnection Handbooks or programs/contracts to allow subscriptions, DER Clients SHALL poll to the *DERControlList* of each *DERProgram* assigned to it to discover new controls or changes to existing controls.
P36.	Unless otherwise specified in utility Interconnection Handbooks or programs/contracts to allow subscriptions, DER Clients SHALL poll to the *DefaultDERControl* of each *DERProgram* assigned to it to discover changes to the default controls.
P37.	The utility MAY optionally specify a recommended polling rate for these resources using the *DERProgramList:pollRate* resource.
P38.	If the polling rate is specified, DER Clients SHOULD poll at this rate.
P39.	Aggregators SHALL subscribe to the following lists: • *EndDeviceList* • *FunctionSetAssignmentsList* of each of the DERs under its management • *DERProgramList* of each of the DERs under its management • *DERControlList* of each of the DERs under its management • *DefaultDERControls* of each of the DERs under its management
P40.	Aggregators MAY subscribe to other lists and instances, such as EndDevice, DERProgram, DERControl instances and others
P41.	Aggregators acting for its DERs and DER Clients SHALL use the IEEE 2030.5 Metering Mirror function set to report metrology data.
P42.	Aggregators acting for its DERs and DER Clients SHOULD post readings based on the *MirrorUsagePoint*:*postRate* resource.
P43.	Aggregators acting for its DERs and DER Clients SHALL be able to report the information shown in Table 12.
P44.	Aggregators acting for its DERs and DER Client SHALL be able to report the dynamic status information shown in Table 13.
P45.	DER Clients SHALL be able to report alarm data shown in 14.
P46.	The Aggregator SHOULD subscribe to the *EndDeviceList* to receive notifications for any additions or changes to the list.
P47.	The Aggregator SHOULD subscribe to each *EndDevice* instance under its control to receive notifications for any deletions of that instance.
P48.	For every inverter under its control, the Aggregator SHOULD subscribe to the list pointed to by *EndDevice: FunctionSetAssignmentsListLink* to receive notifications for any changes in the inverter's group assignments.
P49.	For every inverter under its control, the Aggregator SHOULD subscribe to all of the *DERControlLists* associated with its FSA groups and *DERProgram* assignments to receive notifications for any new or changed *DERControl* events.
P50.	For every inverter under its control, the Aggregator SHOULD subscribe to all of the *DERPrograms* associated with its FSA groups to receive notifications for changes to the *DERProgram* meta-data.

P51.	For every inverter under its control, the Aggregator SHOULD subscribe to all of the *DERProgramLists* associated with its FSA groups to receive notifications for additions, deletions, or changes to the list.
P52.	Maintenance of subscriptions is described previously for the IEEE 2030.5 Specification. In particular: • The Aggregator Client SHOULD renew its subscriptions periodically (e.g. every 24 hours) with the Utility Server. • The Aggregator Client SHOULD fall back to polling on perceived communications errors.

Appendix B – Table of Acronyms.

ACL – access control list

AES – advanced encryption standard

CA – certificate authority

CSIP – Common Smart Inverter Profile

DER – distributed energy resource

DNS – domain name service

ECDHE – elliptic curve Diffie-Helman

ECDSA – elliptic curve digital signature algorithm

EMS – energy management system

FSA – function set assignment

GUID – global unique identifier

HTTP – Hypertext Transfer Protocol

ID – identity

IEC - International Electrotechnical Commission

IEEE – Institute of Electrical and Electronics Engineers

IOU – investor owned utility

IP – internet protocol

LAN – local area network

LFDI – long form device identifier

PCC – point of common coupling

REST - representational state transfer

SIWG – Smart Inverter Working Group

SFDI – short form device identifier

TLS – transport layer security

UTC – coordinated universal time

VAr – volt-ampere reactive

WADL – web application description language

XML - extensible markup language