OpenADR 3 for DER Grid Code Management and Building Operating Envelopes

In relation to IEEE 2030.5, Rule 21 CSIP, Matter, and other standards

Revision Number: 1.0

Document Status: First Release

Document Number: 20250701-01

Table of Contents

Introduction

A grid code is a set of technical rules and requirements that govern how power generation and other facilities connect to and operate within an electrical grid. As electric grids incorporate increasing levels of distributed energy resources (DERs) like solar PV, batteries, and electric vehicles, utilities face new challenges in maintaining voltage stability, frequency regulation, and power quality. Smart inverters now include advanced functions (e.g. Volt-VAR, frequency-watt control) mandated by standards such as California's Rule 21 and IEEE 1547. Rule 21 requires DERs to support an open communication interface (currently defaulting to IEEE 2030.5) to enable utilities (or aggregators) to adjust inverter grid settings and receive telemetry. To date, to our knowledge in the US, direct utility-to-inverter 2030.5 control appears limited in broad production deployment; most operational models rely on gateway/aggregator mediation.

Dynamic Operating Envelopes in contrast govern the maximal import and export limits of a building. Initially aimed at inverters to limit solar exports, we can also apply this to a more holistic view of a building or consumer in general. This is a very clear, objective-based approach that ultimately narrows down the building's consumption curve. While this document focuses more broadly on inverter management, we will outline later how dynamic operating envelopes can be communicated to any control system.

This whitepaper overall proposes an OpenADR 3 based architecture where grid requirements communicate with a customer's Energy & Power Coordination Entity (EPCE) to dynamically control the customer's Point of Common Coupling (PCC) via OpenADR events. Examples of grid requirements include "provide reactive power support", "adjust export capacity to x", or "limit import capacity to y". For the purposes of this paper an EPCE has a one-to-one relationship to a service provider and a one-to-many relationship with downstream DERs. EPCEs may be connected hierarchically to support the aggregator role. All EPCEs act as OpenADR Virtual End Nodes (VENs), and EPCEs coordinating downstream EPCEs also act as OpenADR Virtual Top Nodes (VTNs). EPCEs controlling a customer's PCC with any number of BTM DERs implements these grid requirements using device-level protocols (e.g. IEEE 2030.5, Sunspec, Modbus, Matter, OpenADR 3) to control the customer's fleet of smart inverters and to enforce the dynamic operating envelope. An EPCE may be a cloud-based aggregator that coordinates with many customer EPCEs.

This approach unifies Demand Response, highly dynamic price communications, and direct customer PCC control. OpenADR 3 can be employed as a 'uniform interface' for all grid coordination with customers. Grid code control of the customer PCC is focused on power management; highly dynamic pricing and DR events focus primarily on energy shifting.

Adopting a uniform interface at the utility to EPCEs reduces the costs of implementation, maintenance, and troubleshooting for the utility.

We begin by clarifying the distinct roles of OpenADR vs. IEEE 2030.5 in DER (specifically inverter) management, especially under California's Rule 21 Common Smart Inverter Profile (CSIP) guidelines. We then present a layered control architecture using OpenADR 3 for upstream signaling and IEEE 2030.5 for downstream enforcement, with references to how this aligns with Rule 21/CSIP functionalities. Realistic use cases are explored to demonstrate how OpenADR "programs" and "events" can encapsulate grid support commands, which are then translated into inverter controls defined by IEEE 2030.5 (or equivalent local protocols). Finally, we provide implementation guidance and highlight the benefits of this approach including modularity, improved cybersecurity and customer experience, reduced utility liability, and faster deployment of DER programs.

OpenADR vs IEEE 2030.5 in the Context of Rule 21

OpenADR originated as a messaging standard for demand response (DR) and load flexibility, coordination between grid operators and downstream participants, sending market-based signals and event notifications (pricing changes or load reduction requests) as the Virtual Top Node (VTN) to many clients (aggregators, charge point operators, or facility EMS'es), the Virtual End Nodes (VENs). OpenADR 3 is the latest version of this standard, offering a modernized REST/JSON API, subscriptions, and "push" notifications, while retaining core concepts from OpenADR 2.0 (events, reports, etc.). In OpenADR 3, grid coordination occurs through Programs (which define a context, e.g. tariff or DR program) and Events (which carry time-bound signals or instructions). This architecture is intentionally and inherently hierarchical and scalable: a VTN typically communicates high-level requirements to multiple VENs (aggregators or EPCEs), each of which then orchestrates the actual load or generation response of one or more devices.

IEEE 2030.5 plays a very different role, as a device-level communication standard supporting direct monitoring and control of a single DER device (or, when controlling a fleet of DERs, the fleet is controlled "as if" it were a single DER). California's Rule 21 for CSIP, IEEE 2030.5 was specified as the default protocol (unless another protocol is mutually agreed upon by the utility & vendor) to provide for "visible and controllable" smart inverters. It provides a rich data model (derived from IEC 61850-7-420) covering numerous DER functions – for example, configuring Volt-VAR curves, retrieving meter readings, scheduling output limits, and so on. In other words, IEEE 2030.5 can specify detailed inverter settings and read status information, whereas in our proposed architecture OpenADR focuses on sending more abstract grid requirements or incentives. While OpenADR is capable of encapsulating the detailed syntax of device-level communications, the utility or ISO may not be optimally positioned to implement the logic required to interpret and generate device-level messages.

To illustrate the distinction, consider California Rule 21 Phase 2 and 3 functions. Rule 21 (through the CSIP guide) specifies that utilities or aggregators must be able to remotely adjust inverter functions like: activating Volt-Var or Volt-Watt mode, setting an export power limit, initiating demand response modes, and obtaining inverter telemetry. IEEE 2030.5 natively defines messages for these (e.g. DERControl, DERProgram, etc.), allowing a utility or aggregator to send a command directly to an inverter or energy management system to activate a specific Volt-VAR, Volt-Watt, or Frequency-Watt curve. OpenADR 3, while capable of device control, typically communicates events with objective messages. For example, an OpenADR event could be crafted to convey an objective like "provide reactive power support at 0.95 power factor leading" or "curtail solar output by 20%" to an EPCE, which then translates that objective into actual device instructions using any protocol the device supports, e.g. IEEE 2030.5 or other protocols. In essence, OpenADR, while also able to carry device-level messages, could act as the "messenger" of grid needs, curves, and other information, while other protocols can be used to execute the "device-level actions" to fulfill those needs.

California Rule 21 and CSIP Alignment

Rule 21 explicitly requires (as of 2019) that inverter-based DERs must be capable of receiving remote commands using a "default communication protocol of IEEE 2030.5," unless another protocol is mutually agreed upon. The investor-owned utilities (IOUs) developed the Common Smart Inverter Profile (CSIP) to detail how to implement IEEE 2030.5 for Rule 21 compliance. CSIP defines the functions and security requirements needed so that any CSIP-certified DER or aggregator can securely interact with utility systems. Notably, CSIP's scope assumes that either a utility or an aggregator will be communicating via IEEE 2030.5 with either individual inverters or with a local gateway at the customer site. In fact, Rule 21 allows an "aggregator or gateway" approach explicitly: rather than mandating that every inverter itself speak IEEE 2030.5, the DER site can use a gateway device or cloud aggregator that translates between the utility's IEEE 2030.5 messages and the inverter's native protocol. Common alternatives include SunSpec Modbus for local inverter communication, or proprietary cloud APIs, with a gateway performing the IEEE 2030.5 interface role for utility connectivity. The California Solar & Storage Association (CALSSA) successfully advocated for this approach, creating a pathway for non-2030.5-capable inverters to comply via certified gateway/aggregator solutions.

In summary, OpenADR and IEEE 2030.5 can be seen as complementary: OpenADR excels in sending broad, programmatic signals (often price- or event-based) to aggregators or end customers, whereas IEEE 2030.5 excels in precise, device-level control and telemetry of DERs. The next sections describe how leveraging both, OpenADR 3 for upstream communications and IEEE 2030.5 (with CSIP) for downstream enforcement, yields a powerful, scalable architecture for DER coordination that aligns with Rule 21 requirements while minimizing the burden on utilities.

Challenges with Direct IEEE 2030.5 Control and the Case for an Aggregator Approach

Although IEEE 2030.5 was mandated as the default inverter communications protocol in California, utility adoption of actual day-to-day direct control with IEEE 2030.5 seems to be limited to date. Several factors seem to have contributed to this slow uptake:

- Scalability and Integration Complexity: Direct utility-to-inverter communication implies that the utility (or its DERMS - DER management system) must manage potentially millions of individual device connections. Each inverter or site would need networking, authentication, and ongoing data exchanges with the utility's servers. This scenario assumes utilities deal directly with millions of customer-owned inverters, which creates major challenges in terms of customer account management, network infrastructure, data processing, and maintenance. In practice, most utilities are likely to find this impractical to implement, especially given legacy IT systems.

- Vendor and Protocol Fragmentation: While Rule 21 defaulted to IEEE 2030.5, it allowed alternate protocols by mutual agreement. Many DER vendors (inverter manufacturers, energy storage providers, etc.) were initially hesitant to embed full IEEE 2030.5 client capabilities in each device, given the complexity and certification overhead. Instead, rather than designing inverters that contain IEEE 2030.5 communications, vendors plan to use another communications protocol that is translated to IEEE 2030.5 by a gateway device or an aggregator. For example, an aggregator might communicate with its inverters via simpler means (Modbus, MQTT, proprietary cloud API) and only expose an IEEE 2030.5 interface at the fleet level for the utility. This means the utility seldom sees a one-to-one 2030.5 link to each inverter; instead, it interacts with a handful of aggregator systems.

- Liability and Support Costs Concerns: Direct control of customer-sited equipment raises questions of responsibility and risk. Utilities are naturally cautious about sending commands that could inadvertently destabilize a customer's system or, for example, reduce their solar output without clear consent. While some more specific control is possible, OpenADR's main philosophy is that the customer should remain in control as the utility asserts control via motivation, providing a demarcation point that reduces potential customer complaints. In contrast, directly operating inverters via IEEE 2030.5 could blur that line, making the utility directly responsible for device behavior. Many utilities prefer a model where aggregators shield them from device-level intricacies; the aggregator ensures the utility's requests are implemented safely and handles any customer issues, thereby limiting the utility's liability and customer management costs.

- Cybersecurity: CSIP mandates a PKI model for device identity at scale. OpenADR 3 supports multiple modern authorization patterns (incl. OAuth2, TLS/mTLS) suited to B2B and B2C integrations. Both approaches can be operated securely; operational overhead differs by deployment model.

- Slow Rollout and Testing: The Rule 21 IEEE 2030.5 mandate took effect around 2019-2020, but creating the necessary public key infrastructure (PKI), certification program, and utility head-end systems has been a multi-year process. By 2020, analyses showed that implementing IEEE 2030.5 for Rule 21 required supporting 18 out of 30 function sets in the standard – a significant undertaking. Utilities and vendors have spent time in pilot tests and workshops to iron out interoperability (e.g., via the SunSpec CSIP test harness), delaying broad deployment. In the interim, many DER aggregators have continued using existing methods (like OpenADR for demand response events, or proprietary APIs) to fulfill grid requests.

Given these challenges, the industry trend has been toward an aggregator-mediated model for DER communication. Aggregators (or DER operators) act as intermediaries that handle the complexity of device control, presenting a simpler interface to utilities. This is explicitly recognized in the CSIP implementation: a utility may treat an aggregator as a single IEEE 2030.5 "Server" representing dozens or hundreds of inverters behind it. In such a model, the utility doesn't address each inverter individually; instead, it sends commands to the aggregator, which in turn controls its inverters as discrete end- devices within its own network. The aggregator knows the capabilities and status of each inverter in its fleet and can optimize the response to meet the utility's request.

OpenADR's role in this picture is to further streamline and standardize the utility–aggregator link. Rather than the utility building a proprietary API or custom integration for every aggregator, the utility can use its existing OpenADR 2.0/3 infrastructure to publish grid requests as standard events. In fact, OpenADR is already widely deployed by utilities globally for DR programs and is mostly used between grid entities (e.g. a utility and an aggregator). Leveraging OpenADR for DER signals is a natural extension. The OpenADR Alliance notes that utilities worldwide have invested in OpenADR infrastructure precisely because it keeps a better demarcation between utility and customer owned equipment sets.

To summarize, the lack of widespread direct IEEE 2030.5 usage by utilities is not a failure of the standard, but rather a reflection of operational prudence. A layered approach where OpenADR serves as the high-level communication layer and IEEE 2030.5 (or equivalent) serves as the local control layer addresses the pain points: it scales better, simplifies utility integration, and maintains clear responsibility boundaries. The next section describes the proposed architecture in detail.

Note that this is identical in spirit to how the telecom industry has evolved towards 'Software Defined Networks' to allow an operator to support a single, simple RESTful interface to control all manner of standardized and non-standardized networking gear via a layered controller architecture.

Layered Control Architecture: OpenADR 3 Upstream, IEEE 2030.5 Downstream

In the proposed architecture, OpenADR 3 is used for upstream signaling from the utility/ISO to DER aggregators, and IEEE 2030.5 (CSIP) or Modbus etc. is used by the aggregators for downstream control of individual inverters. This creates a two-tier control scheme, with a logical demarcation at the aggregator (VEN). The utility's signals are high-level grid requirements or objectives, which the aggregator then translates into specific device actions. Figure 1 illustrates this layered control approach:

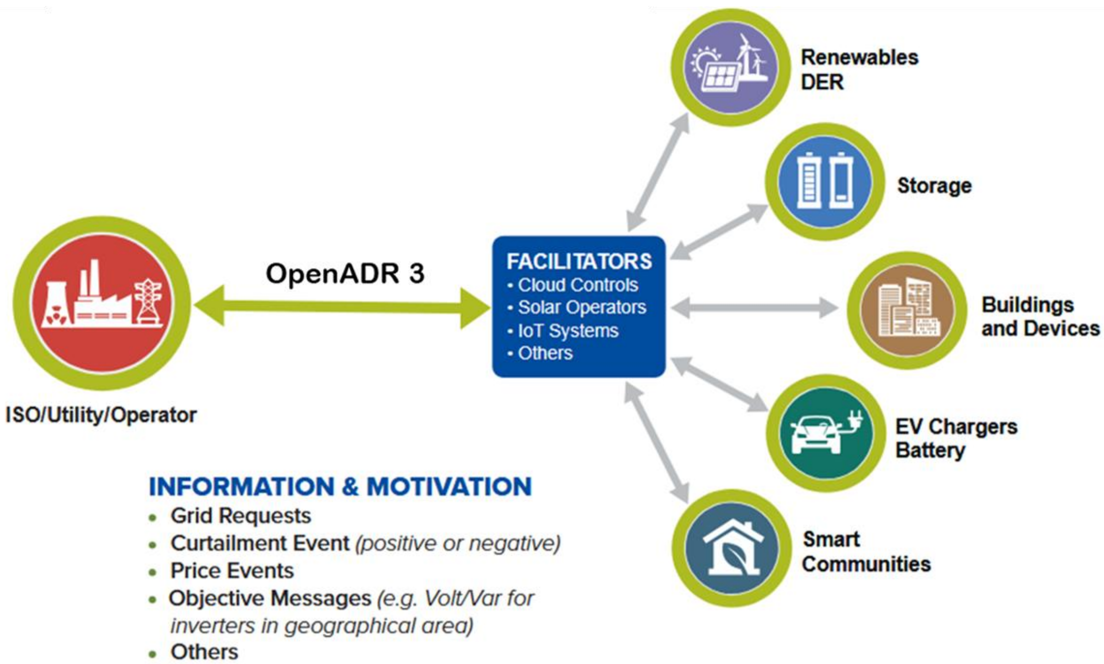


Figure 1: DER control architecture.

A utility or ISO (acting as an OpenADR VTN) publishes high-level grid requirement events to aggregator VENs. These events carry "information and motivation" such as grid requests, curtailment levels, or objective functions (e.g. a Volt-VAR profile to activate). The DER aggregator's system then enforces these requirements on individual inverters using local protocols like IEEE 2030.5, Modbus, Sunspec, Matter, or even OpenADR 3 as well. This control model allows the utility to influence many DERs through one interface, without directly controlling each device.

Control Flow and Components

• Utility / Grid Operator (VTN): The utility (or a distribution system operator or an ISO managing DER aggregations) runs an OpenADR Virtual Top Node server. This VTN is configured with one or more Programs that correspond to DER services or grid support functions. For example, there might be a program for "Voltage Support", another for "Emergency Solar Curtailment", and so on. Each program defines the context and default parameters, while actual dispatches are sent as Events. The VTN may issue events manually (in response to grid conditions) or automatically via a DERMS platform. Crucially, while events could be targeted to individual inverters, in this scenario, events do not micromanage each inverter; they instead communicate requirements or setpoints that the aggregator must achieve (aggregator's responsibility to figure out how).

- Aggregator / DER Operator / Facilitator (VEN): On the other side, each DER aggregator (e.g. a large solar provider, a virtual power plant operator, or an EV charging aggregator with V2G capability) runs an OpenADR Virtual End Node client. The VEN listens for events from the utility's VTN for the programs it is enrolled in. In essence, the aggregator's VEN represents an aggregation point for possibly hundreds or thousands of devices with specific target groups. Once an OpenADR event is received, the aggregator's control system (often called a DERMS or fleet management system) interprets the event's signals and translates them into device-level controls. The aggregator might use an IEEE 2030.5 server to send commands to its inverters (if the inverters have IEEE 2030.5 clients per CSIP), or it might use Modbus, Matter, or other connections.

- Smart Inverters / DER Devices: These are the end devices providing the grid support functions. Each inverter is typically configured to accept commands via some interface – it could be directly via IEEE 2030.5 (common for newer, Rule 21-compliant inverters), or via a local interface such as Modbus/RS485 to a site controller, or via the vendor's cloud. Under Rule 21 CSIP, if an aggregator is used, the aggregator assumes the role of the IEEE 2030.5 "DER Server" and the individual inverter (or site energy management system) is a "DER Client" to that aggregator. Regardless of the method, the aggregator issues the low-level instructions: for instance, set Volt-VAR mode to enabled with curve X, or limit real power to 50% for the next 1 hour, or report current status. IEEE 2030.5 provides a standard way to do this with function sets for DER control, scheduling, and DER telemetry, whereas Modbus would involve writing specific registers for those settings. It should be noted that the latest Matter specification also includes messages to manage residential inverters to some degree.

A key aspect of this layering is that OpenADR events can carry just enough information to express the grid's needs, without dictating the exact per-device command. The aggregator has freedom (within contractual/program limits) to decide how to meet the request. For example, if the utility issues an OpenADR event for voltage support requesting absorption of reactive power (i.e. lower voltage in an over- voltage scenario), the payload might indicate "provide reactive power up to 0.3 per-unit as needed to hold voltage at 1.04 p.u." (if such a custom signal is defined). The aggregator could then select which inverters in which locations should absorb VARs and by how much, possibly prioritizing those with more headroom or closer to voltage violation. OpenADR 3's flexible data model allows defining such program-specific signals – the OpenADR 3 User Guide explicitly envisions using "specific event signal definitions for communicating advanced inverter functions" as a best practice. Indeed, the OpenADR Alliance has previously drafted a "DER Addendum" that demonstrates how to map IEEE 2030.5's inverter function models already into OpenADR 2.0 event payloads for Rule 21 use cases. In those examples, the OpenADR event's signal payloads mirror IEEE 2030.5 data models (derived from IEC 61850), so that the translation at the aggregator can be straightforward. For instance, an OpenADR event could include an "objective" signal indicating a desired Volt-VAR curve number to activate, which corresponds to the IEEE 2030.5 DER settings for Volt-VAR mode. OpenADR 3 makes this even simpler now.

Program and Event Object Construction: Using OpenADR 3, a utility can construct a Program that encapsulates a particular grid support service. Each Program has a unique ID and metadata (e.g., "VoltageRegulationProgram1" with descriptions). Events under that program will carry one or more signals. A signal in OpenADR 3 is essentially a data field conveying a value or level for a certain metric, over one or more time intervals. OpenADR 2.0 already defined standard signal types like ENERGY_PRICE, LOAD_CONTROL, POWER_LEVEL etc., and it also allowed custom signals. OpenADR 3 continues this concept and simplifies it with JSON.

For example:

- A Voltage Support Event might have a signal like voltageReq with a value of, say, 1.04 per-unit (indicating target voltage) or a specific reactive power dispatch in VARs. The target might apply immediately and until further notice, or for a set duration. The targeting of the event could specify a group of sites (VENs) in a certain region of the grid experiencing high voltage.

- A Frequency Response Event could be a grid-wide instruction during an over-frequency condition. It might use a signal like genCurtail with a percentage value or watt value indicating how much generation to drop, or vice-versa (similar to a contingency DR event). The aggregator would then send appropriate frequency-watt controls to inverters (e.g., requesting them to curtail output according to a frequency threshold).

- A Capacity Relief Event (for peak load reduction or emergency) might simply request all solar to curtail to X% of capacity. This resembles a DR event and can be handled by a Load_Control type signal in OpenADR (which many VENs already support for shedding load or generation).

- Dynamic Operating Envelopes and Voltage Feedback. More and more areas are confronted with local overproduction from solar systems and general energy storage solutions. On the other hand, the locations also have a very wide range of power consumption levels due to new heat pumps, EV chargers, and similar consumers. To reduce overproduction and curtail maximum consumption, the operating envelopes can set limits on both ends. OpenADR 3 already supports this in the user guide. Local voltage levels could also be reported back to the utility.

From the IEEE 2030.5 side, the aggregator would translate these into standard DER controls. For instance, if an OpenADR event indicates activation of a Volt-Watt curve, the aggregator's IEEE 2030.5 server would issue a DERControl with the Volt-Watt mode enabled and reference to the predefined curve parameters. If the OpenADR event asks for a certain power factor or VAR output, the aggregator might use the DER IEEE2030.5 function set (if implemented) to set a fixed reactive power or power factor command on each inverter. The EPRI demonstration of OpenADR for smart inverters showed this concept: an OpenADR server event can trigger an inverter to activate a specific Volt-VAR or Volt-Watt curve, essentially by mapping the OpenADR signal to the corresponding IEEE 2030.5 function call. In practice, the aggregator's control software contains this mapping logic.

Example Use Case: Volt-VAR Dispatch via OpenADR

To make this concrete, imagine a scenario where midday solar production in a neighborhood is causing over-voltage at the distribution transformer. The utility's DERMS detects this and decides to mitigate it by adjusting the inverters' reactive power. The steps might be as follows:

1. Utility issues OpenADR Event: The utility VTN creates an event under the "Voltage Support" program targeting the VEN for the solar aggregator that operates in that neighborhood. The event might have an OBJECTIVE signal with a value like "Activate Volt-VAR Mode 3" (where "Mode 3" is understood via program documentation to be a specific curve with aggressive VAR absorption at high voltage). The event can include a textual message or instruction as well (OpenADR allows including a human-readable payload to ensure clarity). The event is effective immediately and remains until canceled.

2. Aggregator receives and acknowledges: The aggregator VEN receives this event (OpenADR events have an eventID , start time, etc.) and sends a confirmation if required. The aggregator's DERMS now knows the utility is requesting Volt-VAR support mode 3 on all relevant inverters.

3. Aggregator executes via IEEE 2030.5: The aggregator's system communicates with each inverter or site controller. If using IEEE 2030.5, the aggregator (as server) sends a SetData or DERProgram message to each DER client enabling Volt-VAR Mode 3. If using Modbus, it writes the registers that enable that mode in the inverter. The aggregator might do this only for inverters in the affected region or perhaps all in its fleet if broadly targeted. Each inverter then

autonomously starts modulating VAR output per the commanded curve (e.g., absorbing VARs once voltage exceeds 1.02 p.u., etc., per Rule 21 default curves).

4. Feedback and verification: The inverters or the aggregator send telemetry back – e.g., var output, voltage readings. The aggregator could compile an aggregated OpenADR report back to the utility summarizing the response (OpenADR supports reporting from VEN to VTN). Rule 21 requires certain status reporting, which an aggregator can fulfill on behalf of its fleet (for instance, providing minimum and maximum voltage seen, or confirming the mode status).

5. Event completion: Later, the utility clears the event (voltage normalized) by sending an OpenADR cancel or a new event restoring normal settings. The aggregator then returns inverters to autonomous default mode (perhaps Volt-VAR Mode 1, the default moderate curve).

This use case highlights how OpenADR can transport a high-level functional requirement ("activate Volt-VAR profile X") which the aggregator enforces via local control. At no point did the utility need to know the details of each inverter's make/model or IP address – that's the aggregator's job. Conversely, the aggregator did not need to interpret why the event was called – it simply responded to a standard signal in a standardized way. This division of responsibility is key to scalability.

Additional Architectural Considerations

- Addressing and Targeting: OpenADR events can target VENs by IDs or groups, location, or any other application-defined attributes. In a DER aggregator program, each VEN could correspond to a region, a customer type, or a specific fleet. For example, one VEN might represent "Solar Aggregator A's fleet in Distribution Circuit 123". The utility can issue events to all VENs in a region or a single VEN if a localized issue. IEEE 2030.5, on the other hand, has its own addressing (it knows each inverter or site under the aggregator by a unique ID in the EndDeviceList). The aggregator maps the higher-level target to the correct subset of devices. If a utility knows a specific inverter or site needs control, it would typically still address via aggregator (e.g., include that site's ID in the OpenADR event targeting if the program was set up that way). However, generally the utility will operate at the aggregator level for simplicity.

- Timing and Latency: RESTful services, like OpenADR 3, typically resolve requests in the sub-second range. Because messages traverse the Internet, lower latency is simply not practical. IEEE 2030.5 is bound by the same Internet constraints. OpenADR 3 performance may be enhanced by communicating commands in advance of their intended effect, or, in the case where the Utility cannot know to send a message in advance, a VEN client may be notified immediately upon creation of the event (using webhook or message queue (MQTT) notification). The message latency between the aggregator and a specific inverter is likely of similar duration, but could vary depending on the nature of the interface.

- Security: OpenADR 2.0/3 and IEEE 2030.5 both specify robust security models. In this layered approach, there are two secure links: one between the utility VTN and aggregator VEN (secured per OpenADR specs), and another between the aggregator and its devices (secured per IEEE 2030.5 CSIP or via VPN/other means for Modbus). The utility and DER provider each manage their own security credentials, which can simplify compliance. Notably, OpenADR 3 supports modern security practices and simpler certificates (leveraging JSON web tokens in some cases), and IEEE 2030.5 in CSIP uses a California-specific PKI. By keeping the utility interface at the OpenADR level, the utility only has to manage certificates for the aggregators, not for thousands of individual inverters – a significant reduction in cost, attack surface and administrative load. This addresses some cybersecurity concerns, as noted earlier, by providing a "clear demarcation point" between utility systems and customer devices.

- Standards Alignment: This approach aligns well with emerging regulatory models. For example, FERC Order 2222 (in the U.S.) envisions DER aggregators participating in wholesale markets. Those aggregators will likely use market signals (possibly via OpenADR or similar) on one side and device controls on the other. Our architecture dovetails with that paradigm: OpenADR for market and grid coordination, local protocols for device execution. In fact, researchers have explicitly recommended "OpenADR 3 for energy and capacity signals; IEEE 2030.5 for inverter management" as a combined standard solution (CalFlex Hub). By standardizing how these protocols interwork, we can achieve greater interoperability. (For instance, ensuring that an OpenADR signal for "limit export to X kW" corresponds exactly to an IEEE 2030.5 DER Control command that sets an Absolute Export Limit of X kW on each site.)

Implementation Guidance for Utilities and Aggregators

Implementing the layered OpenADR/IEEE 2030.5 approach requires coordination on both organizational and technical levels. Below is guidance for each stakeholder:

For Utilities / Program Operators (VTN side):

Define Clear Programs and Requirements: Begin by defining the DER grid services you want to support (voltage regulation, frequency response, emergency curtailment, capacity market, etc.). For each, decide what metrics or signals will be sent. Consult the OpenADR 3 User Guide and Definition to utilize existing signal types when possible (e.g., use powerReal for real power limits in watts, powerReactive for reactive power in VARs, as defined in OpenADR schemas). If new signals are needed (e.g., a signal to select a predefined inverter mode), coordinate with the OpenADR Alliance or at least document these signals clearly for aggregators. The OpenADR 3 framework is quite flexible – it allows multiple payload elements per event interval, so systems can send compound instructions if needed (e.g., one signal for active power limit, another for reactive power setpoint, in the same event).

- Leverage OpenADR 3 Features: OpenADR 3 introduced a simpler RESTful interface and improved reporting. Use the Program object to your advantage – include metadata like the Program name/ID in all events so that VENs know which service is being invoked. For example, a “Rule21VoltVar” program could consistently carry events about volt-var adjustments. This reduces ambiguity. Also utilize OpenADR’s event priority field to manage conflicts (e.g., an emergency safety shutdown event might override a normal voltage support event, using priority = 1 as highest). Ensure your VTN software is certified to OpenADR 3 for basic compliance.

- Coordinate with Aggregators on CSIP: Even though you won’t send IEEE 2030.5 commands directly, you should understand the CSIP functions that aggregators must perform. For instance, Rule 21 may require periodic status telemetry. Many utilities may prefer to get an aggregated telemetry report via OpenADR (simpler IT integration). OpenADR 3 VENs can be configured to send reports upstream – e.g., total solar output, or confirmation that settings were applied. Alternatively, some data (like inverter status) might be available via the IEEE 2030.5 link to aggregators – in which case the utility could request it as needed (if they maintain a IEEE 2030.5 client just for monitoring). In any case, clearly specify in your program rules what data the aggregator must provide, and how.

- Testing and Pilots: Before full deployment, run pilot tests with one or two aggregators. For example, send a test OpenADR event for a moderate curtailment on a mild day, and verify the aggregator’s inverters respond correctly (perhaps by comparing SCADA voltage readings or requiring the aggregator to provide a before/after performance report). Fine-tune the event parameters and timing (e.g., how much notice to give, or whether the events should be "modifiable" after issuance). The OpenADR 3 reference implementation and various test harnesses (QualityLogic, etc.) for OpenADR 2.0b can simulate VEN/VTN behavior for validation.

For DER Aggregators / Operators (VEN side):

OpenADR VEN Integration: If not already OpenADR-capable, deploy an OpenADR 2.0b/3 client. Many vendors offer VEN software libraries or cloud services. Ensure the VEN can handle multiple simultaneous events and can parse custom signal types if the utility uses them. The VEN should support opt-in/opt-out as well (OpenADR allows VENs to send opt-out signals (reports in OpenADR 3) if, say, a resource is unavailable). As an aggregator, you likely won't opt-out often (that would mean refusing a grid request), but the mechanism is there for emergency or maintenance situations.

- Mapping OpenADR to Device Controls: Develop a mapping dictionary or logic engine that converts each OpenADR event signal into one or more device-level actions. The OpenADR Alliance's DER Addendum draft provides examples mapping CSIP/IEEE 2030.5 to OpenADR signals – use these as a starting point. For instance, if you receive an OpenADR signal "VoltVarMode=3" for a certain time interval, your system should translate that into the appropriate API calls or register writes that put all relevant inverters into Volt-Var curve #3 during that interval. This may involve caching or knowing each inverter's preset curves. If an OpenADR event specifies a numeric level (e.g., "limit to 50% output"), ensure you distribute that curtailment appropriately across your fleet (some inverters might need to curtail more if others are at minimum, etc., but collectively achieving ~50% reduction). In effect, your system should function as a simplified DERMS where OpenADR event data is an input, and device commands are the output.

- Compliance with Rule 21 (CSIP) – see also next section on mappings: Even though the utility isn't directly commanding your devices via IEEE 2030.5, you as an aggregator still must meet Rule 21's requirements. That means your fleet's inverters must behave as if they're Rule 21-compliant. For example, they must have the autonomous functions enabled by default (Volt-Var etc.), and you must be able to adjust settings within the allowed ranges. If using IEEE 2030.5 internally, you should get your system CSIP certified (SunSpec certification) as an aggregator server or gateway. If you are using a proprietary or Modbus approach internally, ensure you've followed the CALSSA pathway (i.e. using certified gateway devices, and attesting to capability). The utility might require proof – for instance, demonstrating that you can achieve the control points requested, and providing data logs.

- Telemetry and Reporting: Build in monitoring so you know the state of each inverter (on/off, current P and Q output, voltage, etc.). This is important for you to respond effectively and also to provide data back to the utility. Under CSIP, an aggregator is typically expected to be able to report status on each inverter (via IEEE 2030.5's DERStatus objects). The utility may not poll you for each device's status, but they might ask for an aggregate or exception report. Using OpenADR reporting, you could configure periodic summary reports – e.g., "solar production reduced vs requested" or "voltage at key nodes". These can be sent as telemetry reports to the VTN. This not only satisfies Rule 21's intent for visibility but also builds trust with the utility that your aggregator is performing as expected.

- Fail-safes and Overrides: Decide how to handle conflicts or safety situations. For instance, if your VEN receives overlapping events (say a price schedule and a voltage support request), you'll need to prioritize or combine them. Typically, grid reliability signals should override economic signals. OpenADR provides an eventPriority and an eventType (e.g., EMERGENCY vs PRICE) to help identify critical events. Implement logic so that, for example, an emergency curtailment (to prevent grid outage) will take precedence over a normal price response program for that period. Also, have local fail-safes: if communications to some inverters fail, those devices should ideally revert to default safe behavior (which for smart inverters might be their autonomous Volt-Var default, per Rule 21 Phase 1). Communicate with the utility if a portion of your fleet couldn't execute a command (OpenADR allows sending an "optOut" status for specific events, or a report showing shortfall).

- User Experience: As an aggregator, you interface with both the utility and the end-customers hosting the DERs. Use the OpenADR event information to also inform your customers in a friendly way. For example, if a voltage support event is active, you might display on customer apps or portals that "your solar system is providing grid support by absorbing reactive power, which may slightly reduce real power output – as per utility request". This can preempt customer questions about why their system output has changed. A better customer experience reduces complaints and reinforces the value of participating in these programs.

OpenADR 3 Implementation of Rule 21 (CSIP) Requirements

OpenADR 3 can fulfill the high-level requirements outlined in California Rule 21's Common Smart Inverter Profile (CSIP) guidelines, using a flexible event-based approach. The following best practices map key CSIP requirements to OpenADR 3 capabilities, aligning with the structure of earlier OpenADR 2.0b guidelines:

- Single Communication Model: Unlike IEEE 2030.5 which distinguishes between direct vs. aggregator-mediated communication, OpenADR uses a unified model. A Virtual End Node (VEN) can represent a single site or an aggregator without protocol changes. In practice, most deployments use a gateway/EMS or aggregator VEN to interface with many inverters, but a VEN could also be embedded in a device for direct utility communication. OpenADR's flexibility thus supports both scenarios under one model.

- Secure Interfaces End-to-End: All OpenADR 3 interactions occur over secure transport (HTTPS with TLS 1.2+ by default). Mutual authentication (e.g. validating client and server certificates or tokens) and encryption are optionally required just as in IEEE 2030.5. Any "downstream" link – for example, between an OpenADR VEN (aggregator or gateway) and the actual DER devices – is outside the OpenADR protocol scope but should also be secured per utility guidelines. In other words, OpenADR ensures secure utility-to-aggregator messaging, and utilities typically mandate similar security on the aggregator-to-device communications.

- Client Identification and Certificates: OpenADR 3 supports robust identification of VEN clients. Each VEN has a unique ID (venID) which can be pre-provisioned or assigned at registration. In OpenADR 2.0b this VEN ID was often derived from an X.509 certificate fingerprint, and OpenADR 3 continues to support TLS1.3 as well as OAuth2 client credentials. This approach aligns with CSIP's requirement for unique client GUIDs – the VEN's certificate or credentials serve as the basis for a unique identifier.

- Resource Grouping and Hierarchies: To mirror IEEE 2030.5's DER grouping (e.g. site, circuit, substation groupings), OpenADR provides a groupID mechanism for event targeting. A VEN can assign its managed DER resources to one or many group IDs (logical identifiers) representing categories or grid locations. The VTN (utility) can then issue a single OpenADR event targeted to a groupID (e.g. all DERs on a feeder), and all VENs with resources in that group will respond. It is recommended to use a consistent naming convention for groupIDs (for example, feeder or circuit identifiers) and map each physical DER to all relevant groupIDs along its grid topology. This allows OpenADR to achieve the intent of IEEE 2030.5's group communications using simpler event targeting.

- Time Synchronization: IEEE 2030.5 includes an explicit time service and strict requirements for clock sync. OpenADR 3 does not define a dedicated time-sync protocol; instead, it assumes the VEN and VTN system clocks are reasonably synchronized (e.g. via NTP) . The OpenADR 3 events carry absolute times (ISO 8601) which the VEN interprets using its local clock. In practice, this has been sufficient, but implementers should ensure device clocks are NTP-synced to meet Rule 21 timing accuracy. There is no separate time poll in OpenADR – the normal HTTPS requests inherently use current time for scheduling.

- Monitoring and Telemetry: CSIP expects DER clients (or aggregators) to report status and telemetry (voltage, power, etc.) for each resource. OpenADR's reporting mechanism allows VENs to deliver telemetry upstream, at the aggregator or resource level. For instance, an aggregator VEN might periodically report the total real power output of a fleet of inverters, or a summary of status, rather than each inverter's data unless configured to do so. OpenADR 3 supports configurable report descriptors so that the utility VTN can request detailed data if needed (e.g. per-device metrics) or just high-level metrics. In short, OpenADR can meet Rule 21 telemetry needs, but it emphasizes flexibility – granular device reporting is possible, though not always required for every deployment.

- CSIP Function Set Support: Rule 21 mandates support for various DER functions (volt/var, volt/watt, etc.). OpenADR 3 does not implement inverter control functions as separate "resources" the way IEEE 2030.5 does; instead, it uses events and programs to convey commands for these functions. Essentially all functions in Table 7 of CSIP (the required IEEE 2030.5 function set) can be mapped to OpenADR event signals or report requests. For example, scheduling, direct load control, price signaling, and DER disconnects are all achievable via OpenADR event payloads (illustrated in a later section). The OpenADR 3 Definitions document and OpenAPI schema define payload types that cover these needs (e.g. CURVE for volt-var curves, DISPATCH_SETPOINT for direct power levels, etc.), ensuring that the intent of each smart inverter function can be realized within an OpenADR 3 message. The next sections describe in detail how OpenADR 3 signals map to specific inverter functions.

In summary, OpenADR 3's flexible publish/subscribe event model, combined with secure authentication and grouping capabilities, can satisfy Rule 21's communication requirements. Differences in approach – such as relying on an aggregator VEN rather than direct device links – are by design and come with benefits in scalability and cybersecurity as noted in the whitepaper's earlier sections.

Event Rules and Overlapping Event Handling

IEEE 2030.5 (CSIP) defines strict Event Rules to handle overlapping or conflicting DER commands (ensuring, for example, that a new volt-watt command doesn't improperly override a volt-var setting unless intended). OpenADR 3's event model is more free-form – multiple events can be active and even overlap in time for a given VEN or resource. To maintain deterministic control similar to IEEE 2030.5, implementers should follow these guidelines for event handling:

- Overlapping Events: OpenADR does allow overlapping or nested events in time. There is no built-in protocol rule that automatically resolves conflicts if two events affecting the same DER parameter overlap. Best practice is to avoid sending overlapping OpenADR events for the same control function and resource. If overlaps are necessary (e.g. an emergency curtailment during an ongoing time-of-use event), then use event priorities to manage precedence. OpenADR 3 supports an integer priority field in each event (where a lower number denotes higher priority). The VEN should be configured to always honor the highest-priority event when two are active concurrently. For example, a priority "0" Emergency Shutoff event could temporarily override a priority "1" Volt-Var schedule. Once the higher priority event expires, the VEN can revert to the lower priority event's settings for the remaining time. This approach mimics the CSIP rule that a higher priority DER schedule preempts a lower priority one.

• Sequencing and Nested Events: If an OpenADR VEN receives a new event whose active period falls entirely inside the active period of an existing event (a "nested" event), the VEN's client logic should treat the new event as an override for that interval if it's of higher priority. The VEN may need to resume or reapply the original event after the nested event ends. For example, a battery might be following a daily export limit schedule via one event but then receives a short 15-minute event to respond to a frequency deviation. After that 15-minute frequency-watt response event completes, the VEN should resume abiding by the original export limit schedule. OpenADR does not automate this, so the VEN application must track suspended events and reinstate them as needed. Implementers are encouraged to follow IEEE 2030.5's event layering guidelines (CSIP Event Rule "c" and related rules) in their VEN logic to ensure smooth transitions.

• Event Modification and Cancellation: OpenADR events can be modified or canceled by the VTN, analogous to IEEE 2030.5's event cancellation mechanism. In OpenADR 2.0b this was done via a cancellation message (oadrCancelEvent); in OpenADR 3 the VTN can perform an HTTP DELETE on the event resource or send an updated event with a modified temporal window rendering it completed in the distant past. VENs should be prepared to handle event cancellations – for instance, if an ongoing DER dispatch event is canceled, the VEN should promptly stop that dispatch and possibly revert to a default or previous state. Likewise, if a running event is updated (modification number incremented), the VEN should apply the new parameters. This ensures that utility operators can gracefully stop or adjust DER controls mid-course, echoing the intent of CSIP's update/cancel provisions.

• Idle State and Conflicting Signals: If two active OpenADR events demand truly conflicting actions on the same resource (e.g. one event requests maximum export while another requests minimum export at overlapping times), and priority is the same or not assigned, the VEN cannot satisfy both. In such cases, the VEN or the resource's local controller should have a defined conflict resolution strategy – generally, highest priority wins, or if priorities are equal, a predefined hierarchy of signal types (for example, safety or emergency signals override economic signals). It's recommended that program design prevents such conflicts by not enrolling a single resource in incompatible programs simultaneously. If using an aggregator, the aggregator's business logic (BL) client should coordinate to prevent sending contradictory OpenADR events to the same VEN. By designing DR programs carefully and using the priority field, one can avoid ambiguity in resource behavior. A utility's Business Logic generating events for the VTN may include a rules engine to prevent sending nonsense events.

- Event Confirmation and Compliance: OpenADR 3 maintains the concept of event responses – a VEN can be requested to send a confirmation report back. The VEN's compliance with event instructions (e.g. did the inverter actually reduce output?) is verified via reports rather than protocol-enforced rules. To align with Rule 21 expectations, utilities can require VENs to send follow-up reports of actual DER performance during events. For instance, after a curtailment event, the VEN might report the achieved power reduction. This provides a closed-loop confirmation similar to IEEE 2030.5's "feedback" via resource readings. OpenADR 3's reporting service can be used to automate this: the VTN can include a reportRequest within an event (via reportDescriptor) so that the VEN will return specific telemetry (e.g. real power, voltage) during or after the event. Ensuring that such feedback is in place will help meet the Rule 21 verification and responsiveness requirements, even though OpenADR's core spec leaves verification to program rules and not protocol rules.

By following these guidelines, OpenADR 3 deployments can enforce orderly DER behavior during multiple DR events, effectively emulating the event management rules of IEEE 2030.5. The key is to use the tools OpenADR provides – event priority, careful scheduling, and targeted program design – to prevent conflicts and ensure that higher-importance grid control signals always take precedence.

OpenADR 3 Example Payloads for DER Functions

To illustrate how OpenADR 3 can emulate the IEEE 2030.5 / CSIP functional controls for smart inverters, this section provides example JSON payloads for key DER control functions. These examples use OpenADR 3 event and program constructs to achieve typical smart inverter behaviors such as Volt-Var and Frequency-Watt responses, as well as scheduled dispatch. Each example assumes that an appropriate OpenADR Program has been set up on the VTN with the necessary payload descriptors (defining the type of event payload). The VEN (aggregator or device) enrolled in that program will interpret the event's payload accordingly.

Note: In OpenADR 3, the payloadType and associated data schema define the meaning of the event's payload. For inverter controls, the OpenADR Alliance has introduced payload types like CURVE for multi-point curves and DISPATCH_SETPOINT for direct setpoints. A payloadDescriptor in the Program tells the VEN what kind of payload to expect (curve, setpoint, etc.), and optionally units. Below, we show how these are used in context. (All JSON is representative and formatted for clarity.)

The OpenADR Alliance intends to publish a series of Program Descriptions that serve as models for programs that embody certain functionality, such as Continuous Pricing. A Program Description could be developed for this use case.

Volt-VAR Control (Dynamic Reactive Power Curve)

Volt-Var Functionality: A “volt-var” function directs the inverter to provide reactive power (VAR) support based on the local voltage. Typically this is defined as a piecewise linear curve of % reactive power vs. voltage. In IEEE 2030.5, volt-var is set via the DERSettings and DERCurve. In OpenADR 3, we can realize the same by sending a CURVE payload in an event.

OpenADR 3 Approach: The utility VTN defines a Program (e.g. InverterVoltVar) with payloadType:”CURVE” to indicate that events under this program will carry curve data. The VTN then issues an Event with a list of (x, y) points that represent the voltage vs VAR curve. The VEN, upon receiving the event, will apply these points to the inverter’s control logic

For example, suppose we want a simple volt-var profile where at 0.95 pu voltage the inverter provides 50% reactive power (capacitive, injecting vars), and at 1.05 pu the inverter absorbs 50% reactive power (inductive). We would send an OpenADR event like this:


```
Event =
{
 "programID": "9835839",
 "eventName": "voltVarCurveEvent",
 "intervalPeriod":
 {
  "start": "2025-08-21T12:00:00Z",
  "duration": "PT1H"
 },
 "intervals":[
  {
   "id": 0,
   "payloads": [
    {
     "type":"CURVE",
     "values": [{0.95, 0.5}, {1.05, -0.5}]
    }
   ]
  }
 ]
}
```

In this JSON payload: - The values array contains pairs of floats representing a 2D curve. Here each pair is a point (Voltage, VAR_fraction). We use per-unit (p.u.) voltage on the x-axis (0.95 and 1.05 p.u.), and a fractional reactive power (y-axis) where 0.5 = +50% (vars injecting) and -0.5 = -50% (vars absorbing). The curve is interpreted as linear between these points. - The VEN receiving this event will configure the inverter's volt-var function accordingly. For instance, at nominal voltage (1.0 p.u.) the expected VAR output would interpolate between the given points (around 0 VAR in this example), and at 0.95 or below it would supply ~50% reactive power (capacitive). - The event is scheduled to start at 12:00 UTC and last 1 hour. After one hour, the volt-var setting could revert to default or be updated by a new event, depending on program design. (If a continuous volt-var setting is desired, the event could be sent with a far-future end time or re-issued periodically.)

This approach shows how an IEEE 2030.5 volt-var curve (which might be expressed as a VoltVarCurve with multiple points) is rendered in OpenADR 3 as a simple event carrying an array of points. The OpenADR 3 Definitions explicitly allow arbitrary curves for such use cases.

Frequency-Watt Response (Over-frequency Curtailment)

Frequency-Watt Functionality: This function reduces inverter output when grid frequency rises above nominal, helping to stabilize frequency. In Rule 21, an inverter might be required to curtail power according to a curve (e.g. 100% output at 60.0 Hz, dropping to 0% at 61.5 Hz). IEEE 2030.5 handles this via DERSettings parameters for Frequency-Watt. With OpenADR 3, we can send an event with a CURVE payload to define the output derating vs. frequency.

OpenADR 3 Approach: Similar to volt-var, we use a curve. We define a Program (e.g. InverterFreqWatt) with payload type CURVE. The event's VALUES will list frequency (Hz) on the x-axis and relative active power output on the y-axis.

For example, consider a simple frequency-watt requirement: If frequency rises to 61.5 Hz or above, the inverter should cut output to 0%; at 60.0 Hz (nominal) it should maintain 100% output. An OpenADR event to achieve this:

Event =


```
{
 "programID": "9835839",
 "eventName": "freqWattCurveEvent",
 "intervalPeriod":
 {
  "start": "2025-08-21T12:00:00Z",
  "duration": "PT2H"
```

```
 },
 "intervals":[
   {
     "id": 0,
     "payloads": [
       {
         "type":"CURVE",
         "values": [{60.0, 1.0}, {61.5, 0.0}]
       }
     ]
   }
 ]
}
```


Here the curve points denote frequency (Hz) vs power output fraction. The VEN will interpret that between 60 Hz and 61.5 Hz, it should ramp down the inverter's power linearly. At 60 Hz, y:1.0 means 100% of available output (no curtailment). At 61.5 Hz,y:0.0 means 0% output (full curtailment). If the frequency goes above 60 Hz, the inverter should start reducing power according to this line. In practice, a more detailed curve with multiple points could be used to shape the response (e.g. begin curtailment at 60.3 Hz, etc.), but this two-point example captures the essential behavior.

The duration of 2 hours indicates this frequency-watt override is a temporary setting (maybe during a grid stress period). After 2 hours, the event ends – the VEN might restore the default frequency-watt settings of the inverter (often inverters have built-in default curves per standards like IEEE 1547). If needed, the VTN could send another event to extend or adjust the curve.

Using OpenADR events for frequency-watt has the benefit of being dynamic – the utility can tighten or relax the curve in real time by sending new or modified events (unlike a static configuration). It leverages the same CURVE payload mechanism as volt-var, demonstrating reuse of the OpenADR 3 semantic for any 2D curve-based control. The VEN's inverter control logic just needs to know that in the FreqWatt program, the x represents Hz and y represents power percentage. (This could be conveyed by program documentation, event.payloadDescriptors, or standardized program names.)

Scheduled Active Power Limiting (Dispatch Schedule)

Scheduled Dispatch Functionality: Another common DER control is scheduling an inverter's output or consumption over time. For example, an energy storage system might be instructed to output a certain kW during peak hours and less during off-peak, or a solar inverter could be given an export limit that changes by time of day. In IEEE 2030.5, this is handled via the DERSchedule / DERProgram objects. In OpenADR, the analogous concept is simply an event with multiple intervals, each interval carrying a setpoint for a different time window.

OpenADR 3 Approach: We use the payload type DISPATCH_SETPOINT to send an absolute power limit or setpoint, and we leverage the ability to include multiple intervals in one event. The OpenADR program (e.g. DERDispatch) would declare a payload of type DISPATCH_SETPOINT and a unit (e.g. kW or %). The event can then enumerate a series of intervals, each with a value. OpenADR 3 treats these intervals sequentially.r example, suppose a battery is to discharge at 5 kW from 2:00–3:00 PM, then taper down to 2 kW from 3:00–4:00 PM. The VTN could send one event with two intervals:

Event =

```
{
 "programID": "9835839",
 "eventName": "twoStepDispatchEvent",
 "intervalPeriod":
 {
  "start": "2025-08-21T14:00:00-07:00",
  "duration": "PT1H"
 },
 "payloadDescriptors": [
  {
   "payloadType": "DISPATCH_SETPOINT",
   "units": "KW"
  }
 ]
 "intervals":[
  {
   "id": 0,
   "payloads": [
    {
     "type":"DISPATCH_SETPOINT",
     "values": [5]
```

```
        }
      ]
    },
    {
      "id": 0,
      "payloads": [
        {
          "type":"DISPATCH_SETPOINT",
          "values": [2]
        }
      ]
    }
  ]
}
```


Key details of this JSON schedule:

- The event starts at 14:00 (2 PM) local time (PDT in the example) and runs for 2 hours total. We set DURATION of the default intervalPeriod to PT1H and such that each interval's duration is 1 hour.
- Interval 0 has a duration of 1 hour and carries a payload value of 5 kW. Interval 1 immediately follows, also 1 hour, with a payload of 2 kW.
- The VEN receiving this will interpret: from 2:00–3:00 PM, maintain a 5 kW discharge setpoint; from 3:00–4:00 PM, transition to 2 kW. Because the intervals are consecutive, no explicit start time is needed for the second interval – it begins when the previous one ends.
- The payloadDescriptors section clarifies that the payload is in kilowatts (kW). This helps the VEN apply the value properly (some programs might use percent of capacity instead, but here we choose an absolute kW).
- This example uses an absolute setpoint. OpenADR 3 also supports relative adjustments if needed. For instance, DISPATCH_SETPOINT_RELATIVE could be used to request a change of +5 kW or -5 kW from current level 15. In a schedule, relative values would adjust the prior baseline. Absolute setpoints (as shown above) are simpler when you want explicit control levels.

Under the hood, this scheduled dispatch event in OpenADR achieves what an IEEE 2030.5 DERSchedule would: it defines power levels for specific times. The advantage in OpenADR is that it's part of the standard event framework – no separate object model for schedules is needed. The VEN executes the schedule as a series of setpoints. If the schedule needs to change (say, the utility updates the 3:00 PM value from 2 kW to 3 kW), the VTN can either modify the event (if far in advance) or cancel and send a new event with the revised intervals.

Verification: The VEN can be asked to report the actual battery output during this period using OpenADR reporting (e.g. a telemetry report for the 2-4 PM window). This could be accomplished by adding a reportDescriptor for power in the event, but for brevity we omit that here. The main point is that OpenADR's event with multiple intervals provides an elegant way to implement time-varying inverter or DER commands.

Other DER Functions and Considerations

The above examples cover three primary functions (volt-var, freq-watt, and scheduled dispatch). OpenADR 3 can handle many other Rule 21 functions in a similar fashion:

- Volt-Watt: This is another curve-based function (active power reduction as voltage rises). It would be implemented with a CURVE payload just like Volt-Var, except the y values would represent permitted real power (perhaps in per-unit of max power) versus voltage (x). For example, a volt- watt curve could be sent where at 1.06 p.u. voltage the allowed output is 50%, etc. The VEN would apply that curve to limit the inverter output based on measured voltage.

- Fixed Power Factor: An event could carry a simple setpoint for power factor (or equivalently a specific VAR dispatch) for the inverter. OpenADR has not yet defined a direct "power factor" payload type, but you can achieve it by introducing a new payloadtype or sending either a CURVE with a flat line (if PF is to change at a certain threshold) or by using a combination of real and reactive power setpoints. For instance, instructing an inverter to operate at 0.9 PF could be done by calculating the necessary VAR level for the present real power and sending that as a dispatch value.

- Connect/Disconnect (Energize): While IEEE 2030.5 can remotely connect or disconnect a DER, OpenADR maintains the philosophy of sending a request rather than a direct binary command. To disconnect a DER via OpenADR, the typical method is to send a zero output limit. For example, a DISPATCH_SETPOINT of 0 kW effectively tells the inverter to stop exporting (which is equivalent to a disconnect from an energy flow perspective). Similarly, a resume or reconnect can be signaled by a non-zero setpoint or simply the end of the zero-output event. Some implementations may define a custom signal (or use a boolean in a report request) to explicitly request disconnection, but generally using the standard setpoint achieves the goal. The VEN or site controller might then open a relay or command the inverter off internally. In all cases, OpenADR's role is to send the signal – the site is responsible for executing a safe disconnect.

- Ride-Through Settings: Functions like Low/High Voltage Ride-Through (LVRT/HVRT) and Low/High Frequency Ride-Through are more configuration than dynamic control – they set how an inverter behaves during abnormal grid conditions. OpenADR can convey new settings for these via events as well (as the OpenADR 2.0b guidelines demonstrated using custom signal names). For example, a utility could issue an OpenADR event with a payload that encodes updated trip settings (perhaps as part of a program for DER configuration updates). However, these settings changes are infrequent (often set once during interconnection). It may be more practical to handle ride-through configuration via commissioning processes or management systems rather than via OpenADR events. Still, if needed, OpenADR 3's flexible payload format could carry ride-through parameters (e.g. a set of curves or key values) using either multiple payload fields or a custom payload schema.

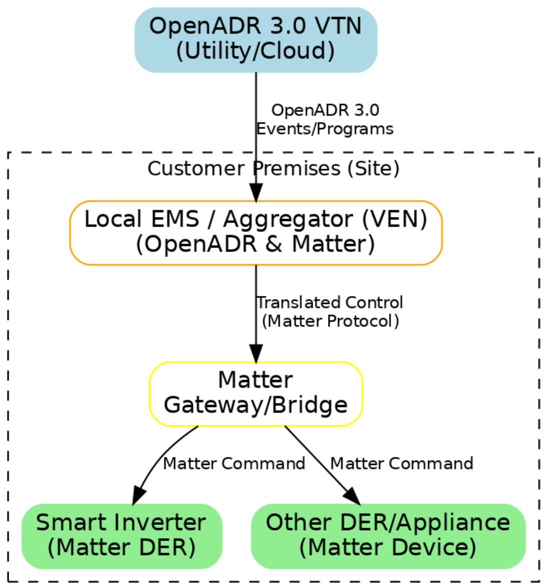
Each of the above can be integrated into the OpenADR 3 framework without requiring protocol modifications – they are a matter of defining the right payload descriptors and event semantics. The OpenADR 3 User Guide includes an Inverter Management use case that exemplifies how a combination of events and reports can manage DER. For instance, one can create an "Inverter Management" program that uses a CURVE payload for all sorts of curve-based controls (volt-var, volt-watt, etc.), and separate programs or signals for other controls, depending on the utility's preference. OpenADR 3 essentially provides a toolkit; as shown, that toolkit is capable of implementing the standard smart inverter functions through carefully constructed events.

The JSON examples above demonstrate that OpenADR 3 messages can encapsulate the same information as IEEE 2030.5's DER controls, albeit in a more abstracted form. This allows utilities to leverage existing OpenADR infrastructure to send high-level requests to DERs, which aggregators or controllers then translate into direct inverter settings. The next section explores a hybrid approach where OpenADR signals could even interface with inverters on local networks via a consumer IoT standard, Matter, blending wholesale DR signals with device-level control.

Direct Inverter Control via OpenADR and Matter Networks

As an intermezzo, we would also like to introduce an alternative pathway to connect to inverters, in particular for residential settings. Increasingly we are seeing EV charging, heat pumps, solar inverters, V2G, and other domestic systems become part of a home energy management system. This offers unique opportunities to connect multiple resources to the grid.

Conceptual architecture: An OpenADR 3 signal from the utility or aggregator (VTN in the cloud) is delivered to a local Energy Management System (EMS) or gateway in the home. The EMS acts as a dual translator – it is a VEN to the utility and a Matter controller on the local network. The EMS receives OpenADR events (e.g. a request to limit power or adjust settings) and then issues corresponding commands to the smart inverter (and/or other DER devices) over the Matter protocol within the home.



The diagram above illustrates an innovative approach to direct smart inverter control that marries OpenADR 3 with the emerging Matter smart home standard. As stated before in this document, traditionally, OpenADR is used to send signals to an aggregator or facility EMS, which then motivates devices to act (without reaching into the device's native control protocol). Matter, on the other hand, is a local communication protocol (IP-based) that allows smart home devices from different manufacturers to interoperate securely on a home LAN. By using a Matter-based gateway, an OpenADR signal can effectively reach into the home and directly affect inverter settings, in a standardized way.

Concept and Feasibility: In this hybrid model, the utility or aggregator runs an OpenADR VTN in the cloud, which sends events to a VEN located in the home or building (this could be a physical hub, an energy management system, or even a smart inverter's controller if it's capable). That VEN is simultaneously a Matter controller on the home network. Matter is designed for local control of devices – it does not inherently have cloud/dr grid interaction – but it can be bridged easily. The VEN/EMS effectively uses OpenADR event payloads to derive Matter commands that the inverter (as a Matter device) can understand. For example, if the OpenADR event payload says “limit output to 5 kW”, the EMS would use Matter’s Device Energy Management cluster (or a similar control cluster for the inverter) to send a 5 kW limit command to the inverter over the LAN.

This approach is conceptually feasible because it leverages each standard in the domain it excels at: - OpenADR 3 handles cloud-to-site communication, enrollment, and high-level DR program logic securely over the internet. - Matter handles local device communication, using a common language to talk to IoT devices (inverters, thermostats, EV chargers, etc.) within the home, over technologies like Wi-Fi or Thread. Matter is IP-based and designed for interoperability, with strong device authentication at the local level.

As noted in a reference design for OpenADR-Matter interworking, “Matter deals with devices in the home, and is only designed to work on a local LAN… The EMS is also an OpenADR VEN which allows the EMS to communicate with the grid and translates the OpenADR requests into Matter commands.” This encapsulates the concept: the EMS/gateway is the linchpin that speaks both languages.

Benefits:

- Standardized Local Control: By using Matter as the intermediary, the need for proprietary protocols or custom integrations to each inverter brand is reduced. Matter aims to be universal for smart home/DER devices. If an inverter (or its connected home battery system) supports Matter, the utility’s signals can reach it in a manufacturer-independent way. This could simplify integration, as utilities send OpenADR events and don’t need to worry about the low-level command specifics – the EMS will handle that via Matter’s standardized device clusters.
- Cybersecurity and Safety: Matter includes robust security (devices are authenticated with certificates at the network layer, and communication is encrypted). Combining this with OpenADR’s secure communication means end-to-end security: cloud to gateway (TLS/ OAuth2 via OpenADR) and gateway to device (Matter’s secure session). Moreover, keeping the final control local (EMS to device) provides a clear demarcation – the utility issues requests, but the customer’s local system has ultimate control to accept and distribute those commands to devices . This can alleviate concerns about utilities directly “reaching into” devices. The customer can always have an override or safety check in the EMS (consistent with OpenADR’s philosophy of the customer being in control via the VEN).
- Reduced Latency for Device Control: Local Matter commands can be very fast (millisecond range on a LAN), which is beneficial for fast-responding functions like frequency-watt or voltage support. The OpenADR signal from cloud to EMS might carry

an instruction like "if frequency exceeds X, respond with Y", and the actual detection and response loop can happen locally via the EMS and inverter, without cloud round-trips. Matter is designed for real-time or near-real-time control in homes, so it complements OpenADR which operates on below seconds to minutes signals timescale. Together, they can achieve a hierarchy: high-level triggers from the grid via OpenADR, immediate execution via Matter.

Leveraging Consumer Ecosystems: Matter is backed by major industry players and is rapidly becoming common in smart home products. If smart inverters, EV chargers, thermostats, etc. join the Matter ecosystem, a utility can tap into a broad range of devices through one integration. The homeowner might buy a Matter-certified inverter or battery, which out-of-the-box can talk to a Matter controller. The utility or aggregator just provides the EMS software (could be an app on a home hub or even integrated into a Wi-Fi router or EV charger station). This drastically lowers the barrier to demand response/DER integration – no specialized vendor-specific gateway needed if the home already has a Matter hub. The OpenADR VEN functionality could even be embedded in software on existing hubs (e.g. SmartThings, Home Assistant with Matter support, etc.), turning them into grid gateways.

Limitations and Considerations:

- Standard Maturity: As of 2025, Matter's energy management features (sometimes referred to as Matter DEM - Device Energy Management) are still evolving. While basic capabilities (on/off, simple load control) exist, more complex inverter-specific controls (like volt-var curves or frequency- watt) might not yet be fully standardized in Matter. This means that in the near term, the EMS may still need custom logic to convert an OpenADR curve or command into the appropriate sequence of device actions. However, ongoing work in CSA (Connectivity Standards Alliance) is defining more advanced energy management clusters. The interworking specification mentioned above envisions exactly this integration.

- Deployment Complexity: Introducing an EMS/gateway that must handle both OpenADR and Matter adds a device (or software service) to the architecture. This gateway must be maintained and kept online. If it fails or if the Matter network in the home is misconfigured, signals won't reach the inverter. There's an operational consideration for utilities: supporting customers in maintaining these gateways (troubleshooting connectivity, updates, etc.). This is somewhat analogous to supporting Wi-Fi thermostats in DR programs today.

- Device Support: Not all inverter manufacturers currently support Matter – many use proprietary apps or protocols. It will take time for Matter to permeate DER devices. Interim solutions might involve add-on Matter bridges that translate Matter commands to whatever protocol the inverter uses (Modbus, Wi-Fi API, etc.). This is feasible – for example, a bridge device on the LAN could receive a Matter command “set output to 5 kW” and then send the appropriate Modbus write to the inverter. However, it adds another layer and might be manufacturer-specific unless/until native Matter support arrives in the inverter firmware.

- Scaling and Addressing: A home might have multiple Matter-compatible DERs (solar inverter, battery, EV charger). The OpenADR-to-Matter gateway (EMS) needs to know which device(s) to control for a given event. Matter uses concepts of Endpoints and Group addressing as well. The EMS could address a specific inverter or a group of devices through Matter. Managing this mapping (for instance, linking an OpenADR “resourceID” or group to a Matter device ID) is a configuration step. It’s not particularly difficult, but it requires careful setup so that, say, “Site Solar Inverter 1” in OpenADR maps to the correct Matter node in the home. Fortunately, Matter has a device discovery mechanism that the EMS can use to find available devices and their capabilities when it connects to the home network.

Future Outlook – A Unified Ecosystem: If this OpenADR+Matter approach proves out, it presents a compelling vision: utilities send price signals, grid constraint signals, or emergency DR signals via OpenADR; homes receive them through an intelligent gateway that coordinates all appliances (not just inverters) via Matter. For instance, the same OpenADR event could request load reduction, and the EMS might then dim thermostats, pause EV charging, and up solar export all in concert, by issuing multiple Matter commands. This bridges grid-level coordination with device-level action in a standardized, secure way. Early implementations are promising – the reference spec by some industry collaborators outlines how an OpenADR-to-Matter bridge can be built to ensure interoperability across many device types in DER programs.

In summary, direct inverter control via OpenADR and Matter is an extension of the OpenADR paradigm deeper into the customer premises. It retains the advantages of OpenADR (open, transport-neutral, scalable for cloud integration) and gains the advantages of Matter (manufacturer-agnostic local control, strong security, ease of use). Utilities benefit by being able to leverage existing consumer smart home infrastructure for DR and DER management, rather than deploying proprietary boxes. Customers benefit because their devices can participate in grid programs with minimal friction and with assurances that control is local and secure. There are still practical hurdles to overcome (device support, standard alignment), but as both OpenADR and Matter continue to mature and converge on energy management use cases, this approach could significantly accelerate DER integration into grid operations in a customer-friendly way.

Benefits of the OpenADR + IEEE 2030.5 Layered Approach

Adopting this two-layer control architecture offers numerous technical and operational benefits for all parties involved:

- Scalability and Modular Integration: By using OpenADR at the top layer, a utility can manage many aggregators through one standardized interface. Each new aggregator (be it a solar provider, a battery aggregator, or a fleet of EVs) can be onboarded simply by configuring a VEN registration – no need for the utility to custom-integrate that aggregator's proprietary API. This hub- and-spoke model is inherently scalable. The aggregator layer, in turn, can handle growth of devices by vertically scaling their own systems or adding more gateway hardware, without affecting the utility's communications. Essentially, it's a modular plug-in architecture: the utility's DERMS "speaks" OpenADR to any number of clients, and each client module handles its devices internally. This was a conscious design choice in OpenADR: "the interactions are between a server (VTN) and a client (VEN)... The core data model uses events flowing to DER and reports back. The server is kept simple, with business logic pushed to clients" – meaning the intelligence about how to execute an event resides with the VEN/aggregator.

- Faster Deployment and Adoption: Utilities can leverage existing OpenADR infrastructure which is already deployed for DR programs in many regions (OpenADR is "the most widely used DR protocol globally" (link)). Extending it to DER control requires minimal incremental investment – mostly in planning the signals and ensuring aggregators join the program. There is no need to wait for full IEEE 2030.5 integration into every back-office system. Meanwhile, DER aggregators find it easier to integrate one OpenADR client than to individually negotiate protocols with each utility. This accelerates market participation. In effect, OpenADR becomes a common language or translation layer between diverse utility DERMS implementations and diverse aggregator control platforms. Evidence from demonstrations (like LBNL's CalFlexHub project) suggests that using OpenADR 3 in front of IEEE 2030.5 significantly streamlines communications and can make new grid services operational in months rather than years.

- Improved Cybersecurity and Clear Demarcation: With this architecture, the utility's control actions stop at the aggregator; they do not directly reach into customer networks. This provides a strong security demarcation. The utility only manages secure communication with trusted aggregator entities, each of which can be vetted and contracted. The aggregators in turn handle security for their devices. This division limits the blast radius of any cyber incident – a breach at an aggregator affects that aggregator's fleet, but cannot directly penetrate the utility's network, and vice versa. It also avoids the utility needing to manage thousands of digital certificates for individual inverters (which would be the case if every inverter was directly on the utility's IEEE 2030.5 system). The OpenADR Alliance specifically touts this as an advantage, noting it "provides the utility with a clear demarcation point to reduce cybersecurity risks" when compared to direct control. Furthermore, OpenADR's "pull" model (VEN can poll or be pushed events) often sits in the DMZ of utility IT, which is a known and protected interface, rather than opening many inbound connections.

- Better Customer Experience and Engagement: Customers generally prefer transparent, incentive-based programs over feeling that the utility can "press a button" to control their assets at will. OpenADR's design enshrines an opt-in principle – customers enroll in programs voluntarily and can opt out of specific events if absolutely needed. The customers can of course manage all this without constant active input but rather by once defining presets and then the aggregator manages the system. Even though in practice an aggregator might auto-accept events, the philosophy is one of collaboration rather than control. As a result, customers (and third-party aggregators representing them) tend to be more receptive to grid requests delivered via OpenADR signals (which often come with incentives or compliance payments) as opposed to direct curtailment commands. This reduces pushback and fosters a more positive relationship. Additionally, because the aggregator can moderate the impact on the customer (e.g., by spreading curtailment across many units or timing responses optimally), the end-user experience is optimized. For instance, an EV charging aggregator receiving a "grid emergency – shed 50% load" signal via OpenADR can choose to delay or throttle charging for a subset of vehicles, rather than disconnecting all charging sessions abruptly, thereby meeting the grid request with minimal user disturbance.

- Utility Focus on Outcomes, Aggregator Innovation: The utility can focus on what needs to happen (the outcome), and the aggregators focus on how to make it happen. This division allows each party to excel. Utilities and grid operators are experts in grid reliability – they know how much load or generation needs adjusting and when. They package that knowledge into an OpenADR event (perhaps even automated by AI or grid analytics). Aggregators are experts in their specific DER technologies and customer behavior – they can innovate in how they fulfill the request. For example, a battery aggregator might use a mix of battery discharge and solar curtailment to meet a "provide 1 MW power" request, optimizing for efficiency and battery life. The utility doesn't need to micromanage those choices, as long as the outcome (1 MW net injection) is achieved. This flexibility encourages innovation in the aggregator space and could lead to better overall efficiency and reliability. It also reduces the engineering burden on utilities; they do not have to develop and test dozens of device-specific control strategies – they simply tell aggregators the performance required, often in terms of standard grid quantities like watts or VARs.

- Alignment with Regulatory and Market Structures: Using OpenADR for upstream communication aligns well with emerging market mechanisms. For instance, California's Demand Response Auction Mechanism (DRAM) and other capacity programs already use OpenADR for dispatching aggregator portfolios. Extending the same interface to encompass smart inverter functionality blurs the line between "demand response" and "distributed energy resource" programs, which is a good thing – it creates a unified approach to flexibility. The layered model also sets a foundation for transactive energy or distribution-level markets. One can imagine in the future the utility VTN sends price signals or grid conditions via OpenADR, and multiple aggregators respond with their resources, maintaining grid balance through quasi-market responses. Indeed, the combination of price-based coordination and reliability signals is explicitly supported in OpenADR 3 (it can send real-time prices as intervals, as well as emergency alerts). Meanwhile, aggregators ensure compliance with IEEE 1547 and Rule 21 on the ground. This could expedite policy goals like California's push for utilizing smart inverters for grid services, without waiting for every inverter to be individually integrated.

In closing, OpenADR 3 and IEEE 2030.5 together provide a powerful toolkit for integrating DERs. OpenADR offers the broad reach, standardization, and flexibility to get the right signals to the right entities (aggregators) at the right time. IEEE 2030.5 (with CSIP in California) offers the fine-grained control and status monitoring needed to ensure the DERs actually perform those signals. Rather than viewing them as competing standards, this whitepaper shows they are complementary layers of a cohesive DER communication strategy. By adopting a layered approach, utilities can accelerate DER integration and harness services like voltage support, frequency response, and capacity provision from customer-owned resources – all while keeping customers engaged and technology vendors innovating. The result is a more modular, resilient grid architecture where responsibilities are well-demarcated: the utility sets the destination, and the aggregator drives the vehicle (DER fleet) to get there. This approach not only meets the letter of regulations like Rule 21, but also achieves their spirit – a future where high DER penetration strengthens rather than jeopardizes grid reliability, through coordination and intelligent control.