CalFlexHub
Advancing Dynamic Energy Management

Transforming Demand Response using Open ADR 3.0

Bruce Nordman, Lauren Parker, Anand Krishnan Prakash, Mary Ann Piette

Lawrence Berkeley National Laboratory

Energy Technologies Area
August 2024

BERKELEY LAB

CalFlexHub's work is made possible by the California Energy Commission and funded through the Electric Program Investment Charge (EPIC) Program. It is administered by Lawrence Berkeley National Laboratory (Berkeley Lab), a Department of Energy (DOE) sponsored Office of Science laboratory.

Acknowledgments:

This work was supported by the California Energy Commission under the Electric Program Investment Charge (EPIC) Program. Solicitation GFO 19-309 entitled California Load Flexibility Research and Deployment Hub (CalFlexHub) Project, that was awarded to Lawrence Berkeley National Lab for the work herein.

Transforming Demand Response using OpenADR 3.0

Bruce Nordman, Lauren Parker, Anand Krishnan Prakash, Mary Ann Piette
Lawrence Berkeley National Laboratory[1]

ABSTRACT

Demand Response (DR) is transitioning from traditional event-based DR to continuous demand flexibility (DF). This transformation is bringing about changes in several dimensions of building-grid interaction: from static DR events on a few days per year to continuous flexibility every hour of every day, from a few customers who sign up for DR programs to all utility customers, and from a few devices such as a smart thermostat to all customer devices. This shift can also lead to a major part of the system load becoming “flexible”. With this, expensive and cumbersome building-grid integration processes are detrimental to achieving climate and electric utility resilience goals. Scalable DR solutions are needed that are continuous and ubiquitous and that facilitate new capabilities to customers and the grid, such as inexpensive microgrid operation and maximizing the benefits from all grid resources.

Central to making this work is a simple coordination model—price and capacity—both inside the customer site and at the customer/grid interface. Also critical is a universal mechanism for communicating this information—for all contexts and scales; with its release in late 2023, the mechanism that best achieves these goals is OpenADR 3.0. This paper will present a system architecture that enables this transformation and explain how all of the above dimensions are facilitated by OpenADR 3.0. It will also describe the nature of the implementation details to incorporate OpenADR 3.0 functionality into building loads, through building gateways, and other devices. Finally, it describes a sample implementation that demonstrates how it can easily be implemented and integrated.

Introduction

Demand Response (DR) began many decades ago with utility direct control (cycling off and on) of devices such as air conditioners and water heaters, and curtailing of industrial processes. Around the turn of the century, this evolved to more formal “event-based” demand response to abstract the desired result from the individual devices involved. Accompanying this was the development of communication protocols to facilitate standard integration of relevant end-use systems. At that time, many aspects of demand response were uncertain and/or changing: the end uses, organizations, markets, communication protocols, coordination with the grid, and more. With this, the OpenADR 2.0b protocol planned to readily facilitate a wide variety of ‘coordination architectures’ (see background section below for definition and discussion).

Even at this early stage, demand response mechanisms were clustering in a few categories, notably event-based and price-based. More recently, two mechanisms have gained increasing traction: aggregators/VPPs (DOE 2023), and pricing (Nordman et al. 2022);

---

[1] This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Office, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This work was also supported by the California Energy Commission under the Electric Program Investment Charge (EPIC) Program, Solicitation Grant Funding Opportunity Number: GFO-19-039, entitled, California Flexible Load Research and Deployment Hub, that was awarded to Lawrence Berkeley National Lab for the work herein.

event-based DR is falling out of favor. This creates the possibility of focusing interoperability efforts in technology and policy on just these two methods.

The core concern of DR is to balance supply and demand over large areas; this has been well addressed by various grid mechanisms including DR. However, distribution system capacity constraints have emerged that need to be addressed. In addition, the technical feasibility and policy desire to dramatically scale up clean energy solutions indicates that the pace of change needs to dramatically change. Such scaling requires a technology basis that is low-cost and standard.

OpenADR has responded to these trends with a new version of the standard, OpenADR 3.0 that is much more suitable as a central tool than its predecessor. Other new needs have also emerged that need to be supported, such as resiliency and microgrids.

This paper is organized as follows. The first section provides background on the history of demand response, its state today, the resulting 'coordination architectures', the issue of capacity management, and the essential importance of Interoperability. Following this is a review of OpenADR 3.0 structure and capabilities. The paper then describes experience with implementing OpenADR 3.0 and lessons learned. The paper ends with some next steps and conclusions.

Background

Demand Response Past

The first widespread use of Demand Response (DR) for residential and commercial buildings began many decades ago with utilities directly controlling devices. Specifically there were dedicated control mechanisms (usually based on radio broadcast or power line communication) to cycle off water heaters and air conditioners on peak consumption days[2]. Direct load control has fallen out of favor over time.

Around the turn of the century, this evolved to more formal "event-based" demand response to decouple the desired result from the individual devices involved. Accompanying this was the development of a communication protocol, OpenADR, to facilitate more standard integration of relevant systems. That DR needs to be automated was assumed from the beginning—and is embodied as the "A" in OpenADR. At that time, DR was understood to cover both price-based and event-based coordination. The other relevant protocol in wide use is IEEE 2030.5; while it supports many of the price- and event-based DR mechanisms that OpenADR does, today IEEE 2030.5 is mostly used to manage inverters.

In parallel to all this is the evolution of retail pricing. Time-of-day pricing was proposed in 1898 (Faruqui 2019) and real-time-pricing in 1971 (Vickrey 1971). The potential of automation to facilitate response to dynamic pricing dates back at least 1978 (Schweppe 1978). Schweppe spoke of both day-ahead hourly and real-time 5-minute pricing. Another important proposal for dynamic pricing was in 2002 (Borenstein, Jaske, and Rosenfeld 2002). The concept of a "price server" dates back at least to 2004 (Piette, Sezgen and Watson 2004).

Important to this were NIST efforts on smart grid architecture and technology (Gopstein et al. 2020), IETF outlining of relevant protocols (Baker and Meyer 2011), the Smart Energy

---

[2] This paper focuses on mechanisms for commercial and residential customers; arrangements that utilities have had with industrial customers are not considered here.

Profile developed by the Zigbee Alliance (Zigbee Alliance 2017), OpenADR 2.0b (OpenADR 2015), and IEEE 2030.5 (IEEE 2023).

After years of development and field testing in California and several other states, OpenADR 1.0 was published to provide a standard way to communicate energy price and demand response events (Piette et al. 2009). Soon after its publication the US developed a national smart grid standards effort, and the 1.0 standard was provided to the standards development organizations (SDOs) to go through a formal consensus-based SDO process. The NIST process supported the development of two standards that became the foundation of OpenADR 2.0: EMIX and Energy Interop; these provided structures for communicating sophisticated grid coordination mechanisms in a standard way. At that time, many aspects of demand response were in a state of flux: the end uses involved, the organizations involved, the market opportunities, communication protocols, and more. With this, the OpenADR protocol planned to readily facilitate a wide variety of 'coordination architectures. EMIX and Energy Interop standards were leveraged to formalize the mechanisms.

Demand Response Today

From early on, demand response was characterized as having two major types: event-based and price-based (DOE 2006). That early DOE definition of DR is:

> *Changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized.*

Pricing operates every day and so is continuous. Event-based DR is utilized only occasionally, e.g. a dozen times a year. Event-based DR performance is usually evaluated by comparing 'event' days with 'normal' or 'baseline' days and observing the statistical difference. While event-based DR can be coordinated directly by a utility, aggregator companies emerged to take on the recruitment and management of customers and customer devices. More recently, aggregators were rebranded as Virtual Power Plants (VPPs) (Downing et al. 2023). VPPs can integrate with utilities and wholesale markets to provide either event-based DR or more continuous optimization. Today, only VPPs and highly dynamic pricing (Nordman et al. 2022) are growing DR mechanisms, both within the US and internationally. This creates the possibility of focusing interoperability effort on specific methods for just these two.

This paper is focused only on issues related to using pricing as the coordination mechanism. However, as when price optimization is done by a device other than the flexible load itself, it is necessary to send functional control signals to the load, which is also needed for VPP operation. This is a point of common need between the two methods.

Capacity management

Until recently, customer loads were such that capacity constraints in the grid were driven only by system peaks, e.g. on hot summer days when many air conditioners were operating at the same time. This grew slowly and could be addressed by efforts targeting this problem. Today and in the future, two new in-building devices raise new issues for system capacity: excess PV power from on-site production, and EV charging (and possibly peaks from electrified heating).

Highly dynamic pricing will reduce capacity issues by shifting load from high-price peak times to low-price times. This should be particularly true for transmission and medium voltage lines. For more local capacity constraints, as with individual feeders or transformers, the problem still remains, even as it is reduced. Pricing can be locational to some degree, but it seems unlikely that prices will become hyper-local, and even if they were would still not guarantee that capacity constraints would not be violated. Dynamic pricing may increase some capacity problems, as when there are peak power times at low prices when renewable generation is so plentiful as to be otherwise curtailed. Thus, a capacity management mechanism is needed.

Capacity management directly in coordination with customers is being done today in Australia with the Dynamic Operating Envelopes mechanism; this limits each customer's maximum export to the grid to enable more customers to export at all. They began with research (ARENA 2021), and are deploying it in Queensland (Energex 2024) and South Australia (SAPN 2024). The mechanism was added to IEEE 2030.5 (Energex 2023) which is used for this in Australia; it also is included in OpenADR 3.0 (OpenADR 2023). This mechanism is limit-based with one-way communication. This is reasonable as it is to address excess PV which is highly correlated across all customers and readily forecasted.

A second mechanism is included in OpenADR 3.0 which is permission-based, designed around the issues that EV charging raises. It involves customer subscription to a capacity level for their typical use (without high levels of EV charging) that they normally never exceed. At any time, a customer can charge at a level between their current use of all other end uses and the subscription. If the customer wants to charge faster, then a digital and automated request can be made to the utility for extra capacity for a specific time duration. The request may be just granted; it may be available but for a fee (if it is a peak time for local capacity); or if there is no capacity available then it will not be granted. Modulating charging by tracking meter status to stay under a specified limit is a capability available on the market today from multiple vendors. The OpenADR 3.0 User Guide describes both mechanisms, as does (Nordman 2024).

What mechanism is the best choice remains to be determined; research and experimentation are needed. However, some mechanism is clearly warranted. Without one, utilities will need to increasingly deny customer requests to add load, and/or spend large amounts of money on increasing capacity.

Ultimately, grid/customer coordination can be decomposed into three domains: power, energy, and capacity[3]. "Power" covers capabilities of inverters such as for reactive power, power quality, etc.; these can be completely disjoint from other coordination, or nearly so. "Energy" is managed with pricing. "Capacity" is the third element, and with the exception of Australia's innovations is very rarely managed at the local level.

Coordination architectures

Any mechanism between the customer and the grid is embodied in a 'coordination architecture'. A Coordination Architecture is an overall framework of entities, their communication and control relationships, financial relationships, and interaction patterns. Often, a specific organization can participate in more than one system (defined by a Coordination Architecture) at the same time. Most coordination architectures have a single central mechanism by which they operate.

For DR, there are the two emerging central architectures of VPPs and highly dynamic pricing. Others that have been used include:

---

[3] This framing does not cover niche ancillary services such as regulation signals. These are unlikely to be significant in coordination with retail customers.

- Direct Load Control: Utilities directly changing the behavior of customer devices, e.g. through cycling devices off periodically during peak times, or changing an operating level (e.g. a thermostat setpoint).
- Event-based Demand Response: Utility programs in which customers agree in advance to reduce load (compared to a 'normal' day) for a defined period of time when called on.
- Limited Time-varying Prices: Examples of this include time-of-use tariffs and critical peak price (or variable peak price) tariffs.
- Two-way Transactive Energy: Systems in which customers provide bids for consumption or flexibility and must then operate in accordance with the acceptance or not of such bids.

Most of these mechanisms have been used at significant scale, except for two-way transactive.

Interoperability

Interoperability is the ability of two or more devices or systems to successfully connect and exchange information to accomplish the desired result, with little or no integration effort. Interoperability is a concern in many electricity contexts, from phone charging connectors to EV charging to AC power plugs/outlets. IT systems demonstrate this even more clearly. For demand response (DR), interoperability has had several dimensions, such as:

- The structure of how coordination occurs between the grid, customers, and third parties.
- Whether the DR is embodied in a tariff, or in an optional program.
- Which communication protocol is used.
- How the protocol is used.
- Whether the coordination is with the customer site as a whole, or with an individual device.

Grid entities have created diverse models selecting among these dimensions. These differences make it difficult for manufacturers to embed DR capabilities into products and for utilities to gain wide uptake of DR by customers.

In general in IT systems the ideal number of mechanisms used for a particular problem to use is one, and are the same across utilities, states, and countries. This is familiar in many technologies used every day such as email addressing, web browsing, and Wi-Fi. In cases where there are multiple mechanisms, such as in phone charging, there is pressure to converge to one. In other cases such as personal computer operating systems, there are only three in wide use globally.

Demand Response has suffered from the use of diverse mechanisms for the same problem, fracturing the market. The presence of aggregator models[4] has further exacerbated this, with diverse arrangements between grids and aggregators, and often proprietary mechanisms between the aggregator and the device. In addition, the diversity of functional control[5] mechanisms in use in buildings further complicates creating standard demand response controls.

---

[4] An aggregator is an entity that controls individual loads in large numbers of buildings and has changes in this operation valued by the utility or a wholesale market pay the aggregator for inducing the change in operation. Thus, the aggregator model values some of the electricity in a building different from what the rest of the electricity is valued at by the meter and tariff with the retail market.

[5] Functional controls relate to the operation of a device to deliver its service, e.g. turning something on or off, changing an operating level, changing a setpoint, etc. Power distribution technologies are limited to the availability and control of electricity flows, over space and time, so are independent of the function of any particular load.

OpenADR 3.0 addresses the interoperability issue in several ways. OpenADR 2.0b is already the most widely used protocol for DR globally but is mostly used between grid entities (e.g. a utility and an aggregator). OpenADR 3.0:

- Is simple and straightforward to implement; it can be readily incorporated into *any* relevant building device.
- Enables a server to be created at the building site (e.g. a gateway device) to decouple grid communication (wide area) with communication to individual building devices (local area). Devices can begin by getting prices directly from the grid, then transition to a gateway device when that becomes available.
- Includes a User Guide that provides example content for common DR mechanisms and tools so that implementers can design their work to be interoperable with others. There is a need for testing and certification of implementations; these are being implemented by the OpenADR Alliance.
- Provides a clear path for implementing standard price communication.
- Is the logical choice to become the *de facto* standard for DR communication globally.
- Provides support for programs in addition to tariffs, and for a diverse range of other applications such as reporting and capacity management, and the User Guide.

Such a combination is unlike anything seen before and should be highly compelling for utilities, product manufacturers, and others to adopt and implement.

Gateways

IT technology transformations often begin with individual devices coordinating directly with the outside world, but moving to having infrastructure devices at the customer site to shift to indirect coordination. When cable TV was introduced, cables were connected directly to television sets, but over time, set-top boxes became ubiquitous as an intermediary. When Internet access began, it was used with dial-up modems connecting an individual PC to the Internet. Again, society moved to using an infrastructure device, a modem/router, to decouple the technology and more efficiently facilitate multiple devices on the customer side.

Electricity should be expected to follow the same path. With the digitalization of power distribution management, there will likely first be individual devices getting prices directly from the grid, but later move to most customers having a device performing gateway functionality to receive prices from the grid, possibly modify them, then rebroadcast locally. Such a device can also host price optimization algorithms for legacy devices that are unable to use prices themselves, and also implement capacity management. A gateway can also host useful functions such as relaying alerts to the user (e.g. device no longer communicating or signaling an issue), and receive energy reporting data.

OpenADR 3.0

OpenADR 2.0b was released over 10 years ago but has technology roots even farther into the past. It was designed around formal mechanisms (EMIX (OASIS 2012)) and Energy Interop (OASIS 2015)) that facilitated highly complex interaction patterns. While the resulting capabilities were extensive, it also resulted in a standard that was complex to understand and complex to implement. OpenADR 2.0b is the most widely used protocol for DR today, and is used globally.

Over the last several years, it became clear that some evolution in OpenADR was needed, to provide a simpler implementation path for applications with simple needs—such as pricing. After consideration of modifying 2.0b, it became clear that today's needs could only be met with a new version of the standard, which became 3.0. While core constructs of 2.0b were retained—such as events, reports, and intervals—the technology basis was redone from the ground up. XML was replaced with JSON, and SOAP interaction with a REST API. Terminology and capabilities were retained when they were known to be used and still suitable.

Structure

OpenADR 3.0 is designed as a REST API—a modern style of IT design that emphasizes simplicity in several ways. This facilitates defining the standard with a machine-readable file (written in YAML) to automate creation of much of the software code needed. This approach can reduce implementation time, reduce errors, and facilitate updating to new versions. While the YAML file is readable by humans, it is not convenient so a Definitions document contains the same information in a more user-friendly format (plus some additional content). The third part of the standard is an informative "User Guide" that provides examples of how to do common demand response activities. The core data model only requires three pages of text to describe.

The primary structures of information transfer are events—that flow towards customer DER—and reports that flow back up from them. Both are composed of a set of time intervals; each interval can contain zero or more data elements. The overall structure that events (and reports) exist within is a program; a tariff is a type of program. Those are the key structural concepts within OpenADR 3.0. All but 'program' are adaptations of the same concept in OpenADR 2.0b.

For price-based coordination, the program object has metadata about the tariff (e.g. name of retailer, name of tariff, effective date, etc.), and periodic events that contain pricing intervals. Each price interval can have a price, an export price (left out if the same or not applicable), and a marginal GHG signal. A separate set of very sparse events contains emergency alerts when they occur. A likely common future example would be interval coverage of approximately 24 hours, with a current price and a forecast of future prices. It could be 25 hourly intervals, or 2 hours of 5-minute intervals followed by 23 hours of hourly intervals. Such coordination requires no customer information to be sent to the grid as a report.

OpenADR interactions are between a server (VTN; Virtual Top Node) and a client (VEN; Virtual End Node). Because of the complex message structure inherent in OpenADR 2.0b, a considerable amount of 'business logic' needed to be present in the server entity. By contrast, OpenADR 3.0 has a much simpler messaging structure which allows the business logic to be separated from the server and be a separate client that places event data into the server for the regular clients to retrieve from it (reports flow in the reverse direction). This separation allows the server itself to be much simpler; Figure 1 illustrates this separation.

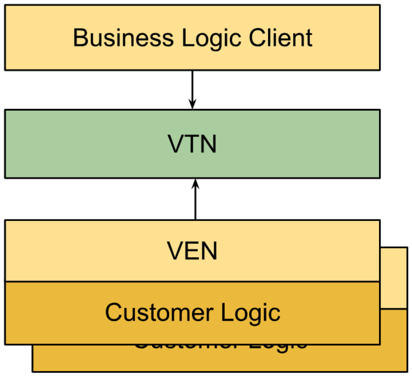


Figure 1. OpenADR 3.0 structure (Source: OpenADR 3.0 Definitions (OpenADR, 2023))

As an example of what OpenADR 3.0 looks like, Figure 2 shows an event to send out three hourly prices. In both Figures 2 and 3, actually useful data are highlighted in blue—the rest of the content is packaging.

```
{"eventName": "pricingEvent",
  "programID": "44",
  "intervalPeriod": {
     "start": "2023-02-10T00:00:00.000Z",
     "duration": "PT1H"
  },
 “payloadDescriptors”: [
    {"payloadType": "PRICE", “units”: “KWH”, “currency”: “USD”}
  ],
  "intervals": [
    { "ID": 123456789, "payloads": [ {"payloadType": "PRICE",  “values”: [.151] } ] },
    { "ID": 123456790, "payloads": [ {"payloadType": "PRICE",  "values": [.184] } ] },
    { "ID": 123456791, "payloads": [ {"payloadType": "PRICE",  "values": [.211] } ] },
  ]
}}
```

Figure 2. Price communication in OpenADR 3.0 (3 hours of prices)

By contrast, to do the same in OpenADR 2.0b looks like that shown in Figure 2. The amount of “packaging” for the core content is much higher for 2.0b than for 3.0.

```xml
<?xml version="1.0" encoding="utf-8"?>
<ei:eiEvent xmlns:emix="http://docs.oasis-open.org/ns/emix/2011/06"
xmlns:scale="http://docs.oasis-open.org/ns/emix/2011/06/siscale"
xmlns:oadr="http://openadr.org/oadr-2.0b/2012/07"
xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0"
xmlns:strm="urn:ietf:params:xml:ns:icalendar-2.0:stream"
xmlns:ei="http://docs.oasis-open.org/ns/energyinterop/201110">
    <ei:eventDescriptor>
        <ei:eventID>pge-pge-etou-b-2022-04-29</ei:eventID>
        <ei:modificationNumber>0</ei:modificationNumber>
        <ei:eiMarketContext>
            <emix:marketContext>http://www.example.org/tariffs/current/TOUB.html
                    </emix:marketContext>
        </ei:eiMarketContext>
        <ei:createdDateTime>2022-04-29T21:55:18.598114Z</ei:createdDateTime>
        <ei:eventStatus>active</ei:eventStatus>
        <ei:vtnComment>BindingPrices:True;LocalPrice:False;
                    RetailerLong:Pacific Edison;RateNameLong:E-TOU Option B;
                    DateAnnounced:2019-01-01;DateStart:2020-06-01</ei:vtnComment>
    </ei:eventDescriptor>
    <ei:eiActivePeriod>
        <xcal:properties>
            <xcal:dtstart>
                <xcal:date-time>2022-04-29T21:00:00.0000000Z</xcal:date-time>
            </xcal:dtstart>
            <xcal:duration>
                <xcal:duration>P1DT</xcal:duration>
            </xcal:duration>
            <ei:x-eiNotification>
                <xcal:duration>PT10H</xcal:duration>
            </ei:x-eiNotification>
        </xcal:properties>
        <xcal:components d3p1:nil="true"
xmlns:d3p1="http://www.w3.org/2001/XMLSchema-instance" />
    </ei:eiActivePeriod>
    <ei:eiEventSignals>
        <ei:eiEventSignal>
            <strm:intervals>
                <ei:interval>
                    <xcal:duration>
                        <xcal:duration>PT2H</xcal:duration>
                    </xcal:duration>
                    <xcal:uid>
                        <xcal:text>0</xcal:text>
                    </xcal:uid>
                    <ei:signalPayload>
                        <ei:payloadFloat>
                            <ei:value>0.25791</ei:value>
                        </ei:payloadFloat>
                    </ei:signalPayload>
                </ei:interval>
                <ei:interval>
                    <xcal:duration>
                        <xcal:duration>PT5H</xcal:duration>
                    </xcal:duration>
                    <xcal:uid>
                        <xcal:text>1</xcal:text>
                    </xcal:uid>
                    <ei:signalPayload>
                        <ei:payloadFloat>
                            <ei:value>0.27671</ei:value>
                        </ei:payloadFloat>
                    </ei:signalPayload>
                </ei:interval>
                <ei:interval>
                    <xcal:duration>
                        <xcal:duration>PT17H</xcal:duration>
                    </xcal:duration>
                    <xcal:uid>
                        <xcal:text>2</xcal:text>
                    </xcal:uid>
                    <ei:signalPayload>
                        <ei:payloadFloat>
                            <ei:value>0.25791</ei:value>
                        </ei:payloadFloat>
                    </ei:signalPayload>
                </ei:interval>
            </strm:intervals>
            <ei:signalName>ELECTRICITY_PRICE</ei:signalName>
            <ei:signalType>price</ei:signalType>
            <ei:signalID>Dynamic_Electricity_Price</ei:signalID>
            <oadr:currencyPerKWh>
                <oadr:itemDescription>currencyPerKWh</oadr:itemDescription>
                <oadr:itemUnits>USD</oadr:itemUnits>
                <scale:siScaleCode>none</scale:siScaleCode>
            </oadr:currencyPerKWh>
        </ei:eiEventSignal>
    </ei:eiEventSignals>
    <ei:eiTarget />
</ei:eiEvent>
```

Figure 3. Price communication in OpenADR 2.0b (3 hours of prices)

As is clear, the new standard is much more concise and much easier for the ordinary person to understand. It is also easier for devices to implement.

Relevant Capabilities

While communicating prices and coordinating capacity are the features of interest for this paper, the standard can do much more, as with:

- Subscriptions, to allow a VTN to 'push' data to a VTN (instead of the VTN having to poll)
- Program (or tariff) metadata
- Capabilities for targeting events and reports to particular resources
- Sophisticated reporting mechanisms including arrays and data quality characteristics
- Emergency alerts
- Event-based demand response support
- Tunneling CTA-2045[6] data to and from a module (or appliance)

Despite all this apparent complexity, devices that do not need a feature do not need to implement it. Devices will be able to implement defined subsets suitable for the applications they are used in.

---

[6] CTA-2045 defines an external hardware device for interfacing between a flexible load and other devices (in building or in the cloud) to decouple the load from any particular physical or application layer standard.

Example Implementation

This section describes the implementation of OpenADR 3.0 in a demonstration software and hardware setup. It covers the design, implementation process, and lessons learned.

General System

The OpenADR Alliance provides member companies an open-source Virtual Top Node (VTN) Reference Implementation (RI) server, which is a vital resource in quickly implementing the standard. The protocol is defined in the 'openapi' format for which online tools such as 'swagger hub' are available to auto-generate a server and clients in many programming languages (SmartBear 2024; see also Specification). One may create a VEN client from scratch or by building upon an auto-generated code base.

Design and Rationale

For the example implementation, customization in functionality of the client was crucial, so new Business Logic (BL) and Virtual End Node (VEN) clients were created from scratch. However, the RI VTN contains all necessary functionality for any customized system, so the provided server code was used as the VTN.

The RI code is entirely written in Python, as was the example implementation. Using the popular "requests" library[7] for Representational State Transfer Application Programming Interfaces (REST APIs) in Python, communicating with the Reference Implementation VTN was made simple, and applications could be readily built around these communications.

Although the communications between the BL Client, VTN Server, and VEN Client are standard using the OpenADR 3.0 protocol, custom client code was created for the data sourcing and final device communications. Figure 4 shows the flow of data between each entity during the example implementation.

The data to be posted as events to the VTN server were price schedules and greenhouse gas levels. The Lawrence Berkeley National Laboratory CalFlexHub Project has a Price Server with four sample 24-hour price schedules. To supplement these, a local price server was created that used California Independent System Operator (CAISO) wholesale energy pricing[8] based on geographical area for the upcoming 24-hour-period at any given time. Additionally, marginal greenhouse gas emission levels were sourced for this implementation from WattTime[9], which has real-time, geographically scoped forecast levels for the upcoming 24-hour period. The BL Client had options to choose either price server and include greenhouse gas levels when posting data to the VTN Server.

The VEN Client needed to interpret the data received from the VTN Server to transform the prices to functional control before communicating with the end-use devices (EUDs). Algorithms to combine price data with greenhouse gas data were integrated into the VEN Client, which then sent the EUDs a power level at which to operate. The two EUDs that were connected to the VEN were a simulated EUD, which displayed a consumed power percentage based on the

---

[7] "requests" python library (code and documentation): https://pypi.org/project/requests/

[8] CAISO data portal: http://oasis.caiso.com/mrioasis/logon.do

[9] WattTime Data API (V3), https://docs.watttime.org/ .

values it receives from the VEN Client, as well as a real smart bulb which adjusts brightness based on the power consumption level it receives from the VEN Client.

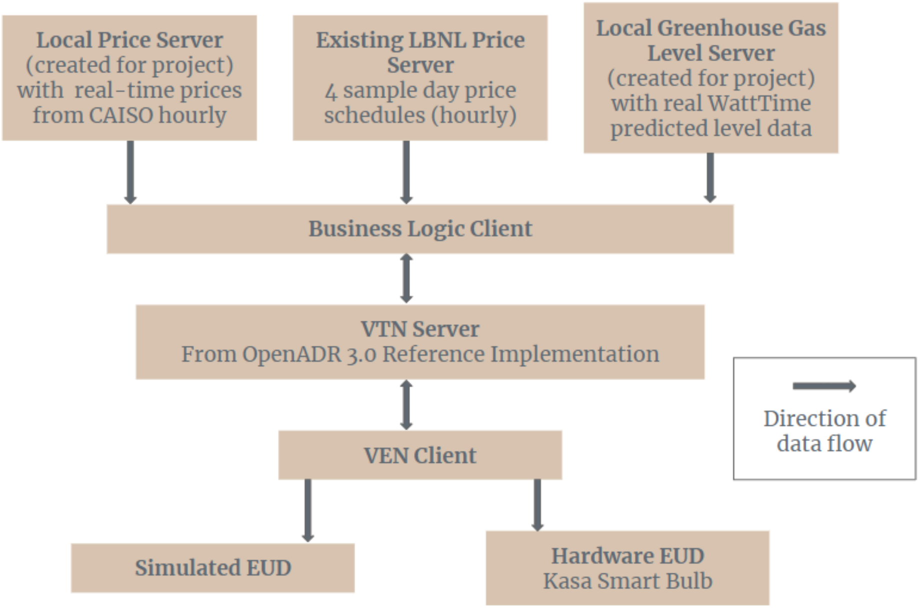


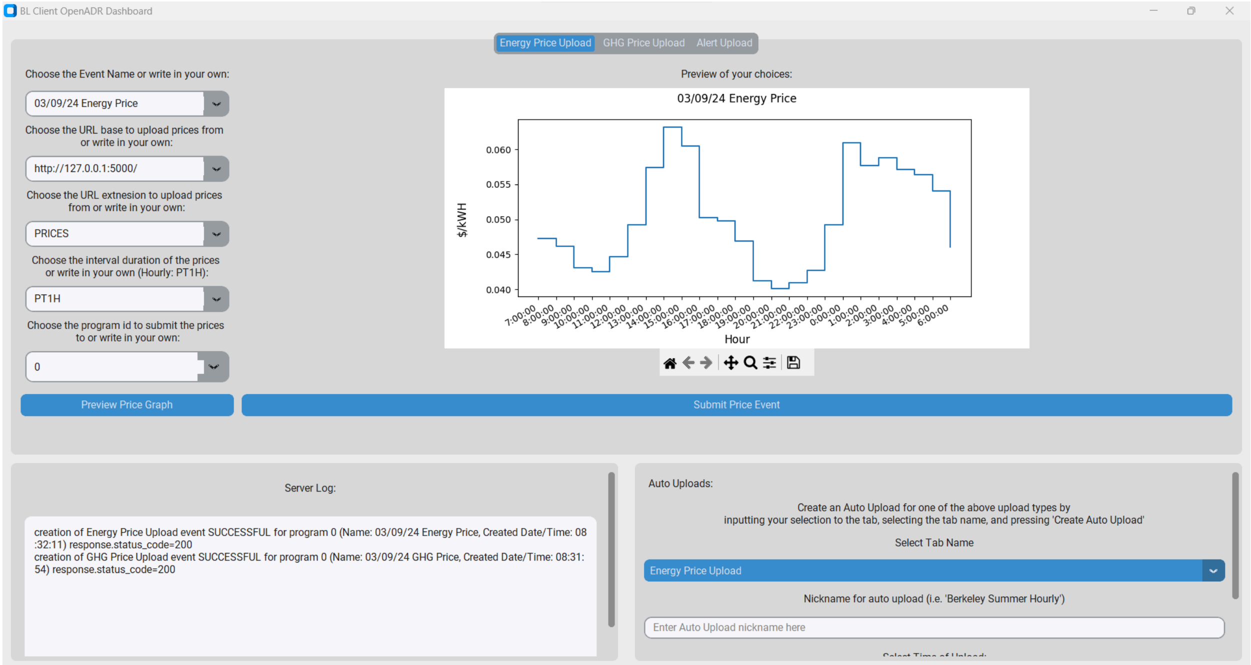
Figure 4. Example implementation architecture diagram

The events passed between the OpenADR 3.0 BL Client, VTN Server, and VEN Client allowed the BL Client and VEN Client to specify a program ID (e.g. a tariff ID), which ensures that the correct data are transmitted to the corresponding end user. It also allows specification of hourly or five minute pricing through the duration data, as well as the payload type and start time of the provided data.

Implementation Process

The first implementation step was to clone and run the RI VTN Server code. Any client, BL or VEN, needs to have the correct authorization token (found in using the RI VTN package tools). Once this has been done, the commands to interact with the server are simple. Based on the permissions given with a client authorization from the VTN Server, a client uses standard REST commands—GET, POST, PUT, and DELETE—to add to, retrieve, and manipulate the data held by the VTN Server. Authorization with the server, as well as a GET command to retrieve events, can all be done in less than ten lines of Python code.

To show the operation of the initial implementation, user interfacing applications were created to serve as a BL Client and VEN client. The applications show a display of the data that is being posted or has been retrieved from the VTN Server. The BL Client application also can retrieve data from the price servers and greenhouse gas server, and then has options to decide what data to post and when. It also allows emergency notification postings ("alerts") to the VTN Server for grid emergency or other such events. Each event posting can be automated to occur at specific times. These options can be selected in the UI application as shown in Figure 5.



Choose the Event Name or write in your own: 03/09/24 Energy Price

Choose the URL base to upload prices from or write in your own: http://127.0.0.1:5000/

Choose the URL extnesion to upload prices from or write in your own: PRICES

Choose the interval duration of the prices or write in your own (Hourly: PT1H): PT1H

Choose the program id to submit the prices to or write in your own: 0

```
creation of Energy Price Upload event SUCCESSFUL for program 0 (Name: 03/09/24 Energy Price, Created Date/Time: 08
:32:11) response.status_code=200
creation of GHG Price Upload event SUCCESSFUL for program 0 (Name: 03/09/24 GHG Price, Created Date/Time: 08:31:
54) response.status_code=200
```

Create an Auto Upload for one of the above upload types by inputting your selection to the tab, selecting the tab name, and pressing 'Create Auto Upload'

Select Tab Name: Energy Price Upload

Nickname for auto upload (i.e. 'Berkeley Summer Hourly')

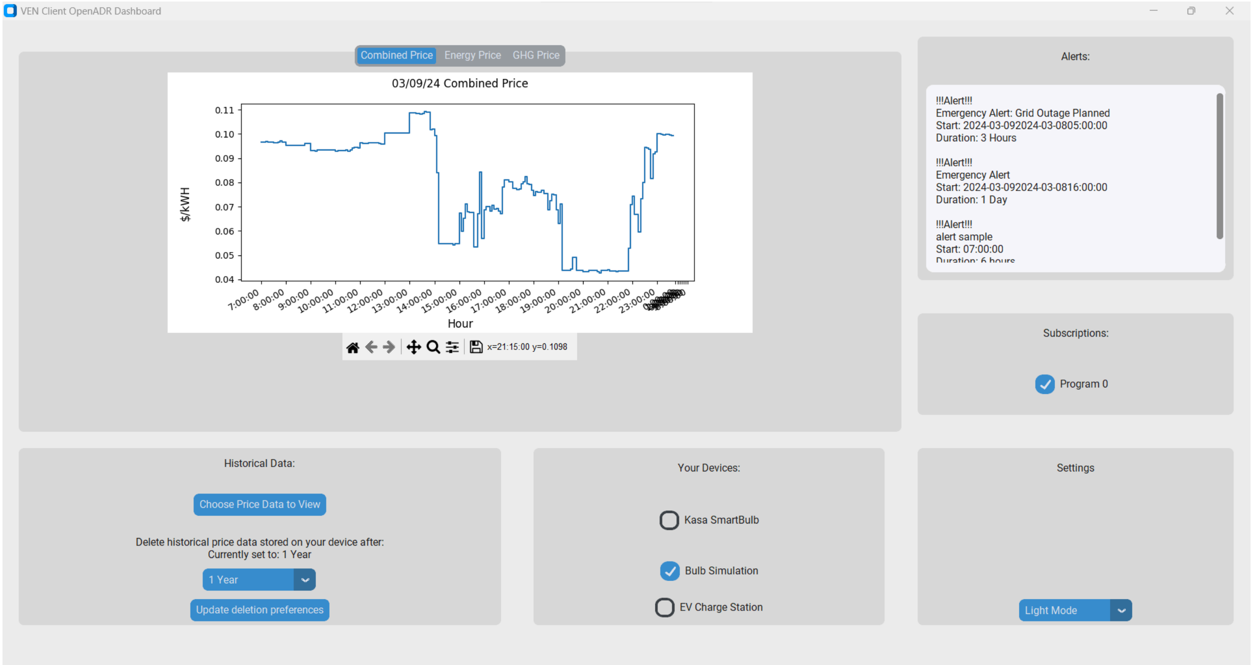
Figure 5. Screenshot of the BL client application

The VEN Client application automatically receives events for the program it is subscribed to (in this case a tariff). When new data are received, the client accepts the new prices, and displays them in the application. The end user has the option to combine the greenhouse gas levels and prices into a new "local price". Any of these values can be selected as the data by which to operate the EUDs. The example implementation VEN Client application showcasing these features can be seen in Figure 6. The original price changed only once an hour but the greenhouse gas signal has 5-minute intervals and is added to the retail price.

Lessons Learned

One of the biggest takeaways from the example implementation project was the simplicity and speed with which an OpenADR 3.0 VTN Server was implemented, along with low-level BL and VEN client code to communicate with it. The clear and concise documentation provided by the OpenADR Alliance created a strong foundation for understanding the code and communication protocols before implementing them in Python. The data flow is intuitive and consistent throughout scenarios, which makes testing the clients and server consistent as well.

Due to the intuitive nature of the data flow, the UI apps are able to clearly illustrate the processes by which separate entities might operate in a real-world scenario. For example, the BL Client application could be from a utility company providing prices for a tariff within a geographic area, and the VEN client application could serve a customer site as a whole, or an individual device. This can facilitate the likely normal set of grid signals: a 24-hour price forecast, marginal greenhouse gas levels, and occasional emergency alerts.



Alerts:

!!!Alert!!!
Emergency Alert: Grid Outage Planned
Start: 2024-03-092024-03-0805:00:00
Duration: 3 Hours

!!!Alert!!!
Emergency Alert
Start: 2024-03-092024-03-0816:00:00
Duration: 1 Day

!!!Alert!!!
alert sample
Start: 07:00:00
Duration: 6 hours

Subscriptions:

☑ Program 0

Historical Data:

Delete historical price data stored on your device after:
Currently set to: 1 Year

1 Year

Your Devices:

○ Kasa SmartBulb

☑ Bulb Simulation

○ EV Charge Station

Settings

Light Mode

Figure 6. Screenshot of the VEN Client Application with relevant alerts and a combined graph displayed, showing a Local Price combining GHG and Retail Prices

After demonstrating how simple it is to implement clients and servers that use the OpenADR 3.0 protocol, it is clear that many applications can utilize a sample implementation like this. Simple versions of clients, custom or generated, alongside the Reference Implementation VTN could also serve as a cohesive testing platform for smart device manufacturers so that the devices automatically respond to OpenADR messages coming in directly from the grid, or from a price server local to the building.

Next Steps

Demand flexibility has a bright future ahead. Many institutions could help accelerate the progress—government entities, utilities, advocacy groups, and others. Among steps needed soon are:

- Identification of a small number of coordination architectures to focus on
  - Highly dynamic pricing and Virtual Power Plants (aggregators)
- Agreement on a small number of communication protocols to use
  - OpenADR 3.0 for energy and capacity; IEEE 2030.5 for inverter management
- Standardization of how these protocols are used
  - For grid signals (e.g. prices) and for functional controls, a few modern protocols such as Matter (CSA, 2024).
- Utilities beginning to offer attractive tariffs with highly dynamic prices
  - Customers should expect that these routinely result in lower bills
- Definition of a mechanism for customers to have choice between tariffs and VPP optimization, separately for each flexible device
- Agreement that capacity management coordination between the customer and grid is needed

    - Research on what this mechanism should be
- Global cooperation and harmonization on all of the above

Notable for this is the tight linkage between policy and IT technology. Such a situation is almost novel for the energy efficiency community. A good source of insight here is the Energy Star program, which has addressed this often, since it has deep and long-standing activity with electronics devices for which IT technology is central. Even 32 years ago, the very first Energy Star specification—for computers and monitors—required one specific technology standard (VESA/DPMS).

Another priority should be the application of this approach to complementary topics, such as microgrids. The vast majority of microgrids today cover just a single customer site. However, microgrids of neighborhood or community scale are highly desired by many. Having load flexibility technology that is equally useful when grid and Internet connected as it is for when both are lost would significantly reduce the cost—and increase the performance—of microgrids of both types.

Progress in this area requires both understanding the ideal long-term state as well as the various methods used to efficiently transition. An increasing driver of the need for flexibility—and capacity management—is electric vehicle charging. This can be used as an introduction of the above to customers, utilities, and other stakeholders, even making short-term use of systems in which EV charging is done at a more dynamic tariff than the rest of a building's load.

Conclusions

The U.S., and much of the rest of the world, is in the midst of a transition from Demand Response to Demand Flexibility. Their technical definition may be the same, but flexibility as a framing leads people to have a much more expansive view of what it covers and can do. This transformation of what is considered reasonable and feasible is an essential part of making flexibility succeed.

In the coming few years, as the technology, policies and markets mature, a rich ecosystem will likely emerge that can provide grid services when needed and at scale. While some regional variation is inevitable, to the degree that the technologies involved are adopted globally will be to the benefit of all.

Flexibility is more and more clearly becoming implemented by various combinations of pricing and VPPs. This allows focusing implementations on a small number of mechanisms and protocols, to provide a reasonable path to accomplishing widespread interoperability. Central to all this can be simple, standard, and consistent use of OpenADR 3.0. This can lead to flexibility which is widespread, effective, low-cost, and valuable to utilities. Flexibility and dynamic capacity management can also enable more customer electrification and more integration of storage and renewables on the grid.

References

ARENA (Australian Renewable Energy Agency), Dynamic Operating Envelopes Workstream, https://arena.gov.au/knowledge-innovation/distributed-energy-integration-program/dynamic-operating-envelopes-workstream/, December 2021.

Baker, Fred, D. Meyer, Internet Protocols for the Smart Grid, RFC 6272, June 2011, https://datatracker.ietf.org/doc/html/rfc6272 .

Borenstein, Severin, Michael Jaske, and Arthur Rosenfeld, *Dynamic Pricing, Advanced Metering, and Demand Response in Electricity Markets*, CSEM WP 105, 2002. https://escholarship.org/uc/item/11w8d6m4

CSA, Connectivity Standards Alliance, Matter Specification Version 1.3, April 17, 2024.

CTA, Consumer Technology Association, Modular Communications Interface for Energy Management, CTA-2045-B, November 2020.

DOE, Benefits Of Demand Response In Electricity Markets And Recommendations For Achieving Them: A Report To The United States Congress Pursuant To Section 1252 Of The Energy Policy Act Of 2005, Department of Energy, February 2006.

Downing, Jennifer et al., Pathways to Commercial Liftoff: Virtual Power Plants, September 2023, Department of Energy, https://liftoff.energy.gov/vpp/ .

Energex, SEP2 Client Handbook, https://www.energex.com.au/__data/assets/pdf_file/0007/1072618/SEP2-Client-Handbook-13436740.pdf , September 2023.

Energex, Dynamic Connections for Energy Exports, https://www.energex.com.au/our-services/connections/residential-and-commercial-connections/solar-connections-and-other-technologies/dynamic-connections-for-energy-exports, May 2024.

Gopstein, Avi, Cuong Nguyen, Cheyney O'Fallon, Nelson Hastings, David Wollman, NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 4.0 (DRAFT), National Institute of Standards and Technology https://www.nist.gov/system/files/documents/2020/07/24/Smart%20Grid%20Draft%20Framework.pdf

OASIS, Energy Market Information Exchange (EMIX) Version 1.0, January 2012 https://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html .

OASIS, Energy Interoperation Version 1.0, June 2014, https://docs.oasis-open.org/energyinterop/ei/v1.0/os/energyinterop-v1.0-os.html .

OpenADR Alliance, OpenADR 2.0 Profile Specification: B Profile, 2015 Revision Number: 1.1.

OpenADR Alliance, OpenADR 3.0, Revision 3.0.0. https://www.openadr.org/openadr-3-0, September 2023.

Nordman, Bruce, Marco Pritoni, Mary Ann Piette, and Anand Krishnan Prakash, Communication Requirements for Price-Based Grid, ACEEE Summer Study on Energy Efficiency in Buildings, August 2022 https://escholarship.org/uc/item/4j30s17j .

Piette, Mary Ann, Osman Sezgen, David Watson, David, Development and evaluation of fully automated demand response in large facilities, Lawrence Berkeley National Laboratory, March 2004. https://escholarship.org/uc/item/4r45b9zt

Piette, Mary Ann, Ghatikar, Girish, Kiliccote, Sila, Koch, Ed, Hennage, Dan, Palensky, Peter, and McParland, Charles. Open Automated Demand Response Communications Specification (Version 1.0). United States: N. p., 2009. Web. doi:10.2172/951952.

SAPN (South Australia Power Networks), Flexible Exports, https://www.sapowernetworks.com.au/industry/flexible-exports/, May 2024.

Schweppe, Fred, Power Systems 2000: Hierarchical Control Strategies , IEEE Spectrum, Volume 15, Number 7, July 1978.

SmartBear, SwaggerHub, https://swagger.io/tools/swaggerhub/, May 2024.

Vickrey, William, Responsive Pricing of Public Utility Services, The Bell Journal of Economics and Management Science, Spring, 1971, Vol. 2, No. 1, pp. 337-346, RAND Corporation https://www.jstor.org/stable/3003171 .

Zigbee Alliance, Zigbee Smart Energy Standard, Version 1.4, June 2017.