OpenADR Webinar: Intro to OpenADR 3

Rolf Bienert
Frank Sandoval
Bruce Nordman

May 2025

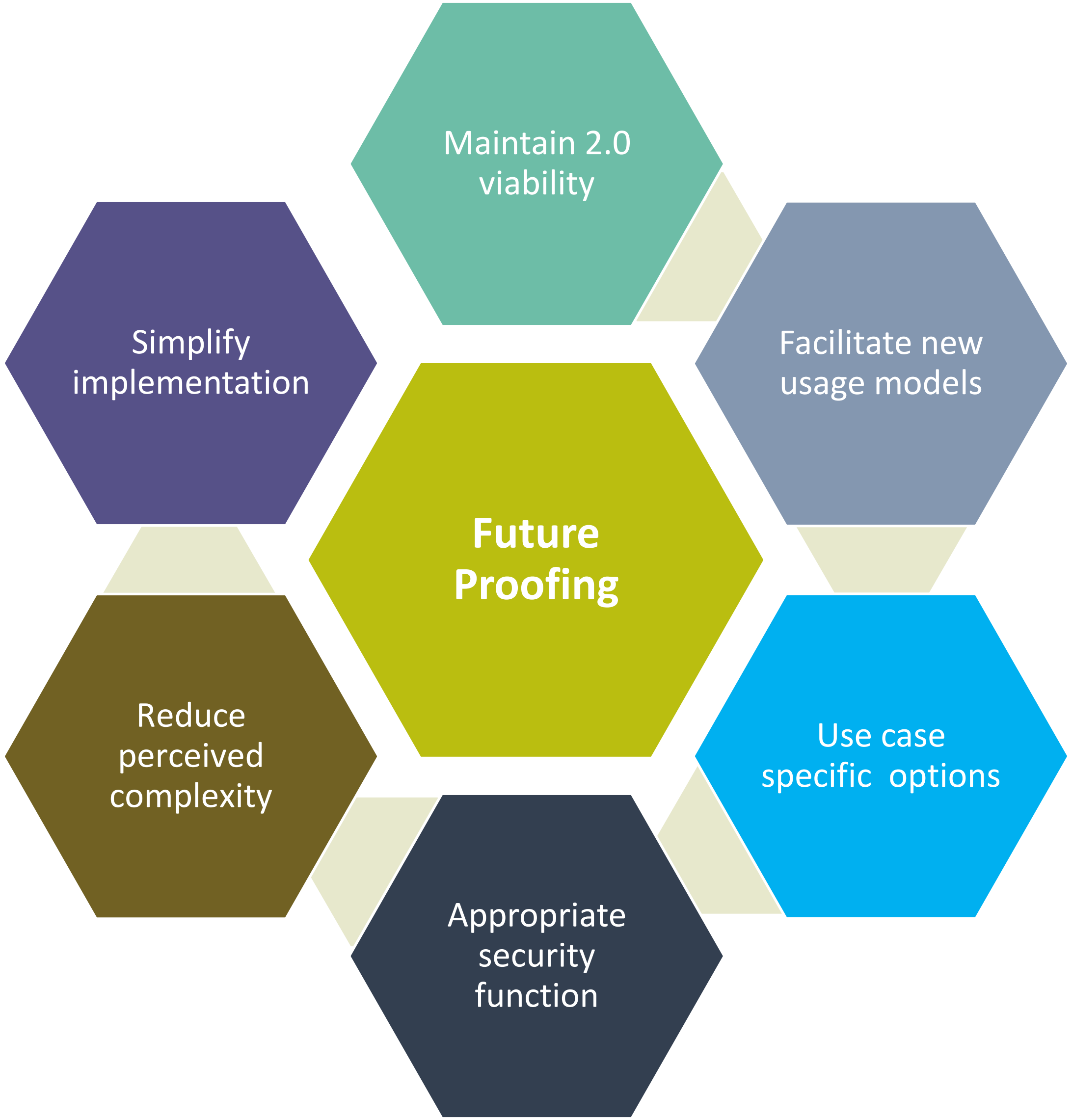
Housekeeping

- The webinar is being recorded
- Slides and Recording will be made available on https://www.openadr.org/webinar-series
- All attendees are in listen only mode
- To ask questions, please enter them in the Questions tab of the Webinar Tool
  - We will field as many questions as possible at the end of the presentations

Why a new OpenADR version?

- 2.0b is nearly a decade old (and the ideas were designed well before that – although still valid)
- 2.0b is widely used globally - no existing implementations need to change anything
- 3.0 will not make 2.0 obsolete
- 2+ years of discussions around improvements, updates, new tech
- 3.0 is built on modern IT technologies and principles
  - REST model for API
  - Clear separation of VTN as server of data from Business Logic that determines that data
- 2.0b oriented to VENs in cloud entities
  - An increasing number of VENs will be in individual flexible loads and other in-building devices
  - These VENs will implement a small subset of OpenADR capabilities
  - Implementation burden should be minimized - 3.0 does this

What we wanted


Maintain 2.0 viability
Simplify implementation
Facilitate new usage models
Future Proofing
Reduce perceived complexity
Use case specific options
Appropriate security function

Survey

The OpenADR Alliance, in partnership with Tanergy, invites you to participate in a short but impactful market research survey on OpenADR 3.

This study will help quantify adoption, understand industry challenges, and influence the evolution of the OpenADR 3 standard.

- ✓ You'll receive early access to the summary of findings.
- ✓ Help drive innovation across energy management systems.
- ✓ Support a future-ready demand response ecosystem.
- ✓ Share your needs—so vendors, regulators, and tech partners can act.

Take the 5-minute survey here

(https://docs.google.com/forms/d/e/1FAIpQLSdw7u-ezfIjPDzKlCQ81rxV6qtenPgBwkyaJwIhXgBzDJ5tyA/viewform?usp=send_form)

OpenADR 3 Intro

Frank Sandoval

Principal - Pajarito Technologies LLC

email: francisrsandoval@gmail.com

Agenda: Questions welcome!

- Executive overview
- API
- Definitions and User Guide
- Reference Implementation and Test Tool
- GitHub projects
- Open source
- Testing and Certification

Executive Overview - Vision

OpenADR 3 is a streamlined version of the OpenADR protocol intended to **lower technical barriers** to adoption across **the broadest set of users**.

As with 2.0b, Utilities may communicate a wide range of signals, such as simple demand response commands, continuous dynamic pricing, and many more.

OpenADR 3 provides a more modern and easy to use mechanism more appropriate for home users and IoT devices.

- As well as supporting traditional Demand Response clients such as Industrial and Commercial users and Aggregators.

OpenADR 3 is a ‘functional equivalent’ of OpenADR 2.0b

- The set of Use Cases and Requirements addressed by 2.0b are all addressed by OpenADR 3.

Executive Overview - Basic Use Case

In general terms, OpenADR is quite simple:

- A utility defines a Demand Response ‘program’ to address a business goal, such as providing dynamic pricing to time-shift load and smooth peak usage.
- It transmits ‘events’ that are consumed by any number of VEN clients
- VENs may produce ‘reports’ that are transmitted back to the utility

OpenADR provides a standard so that all VEN clients can **interoperate** with all VTNs

VEN is ADR speak for an entity that controls one or more ‘loads’ (system or device that consumes energy)

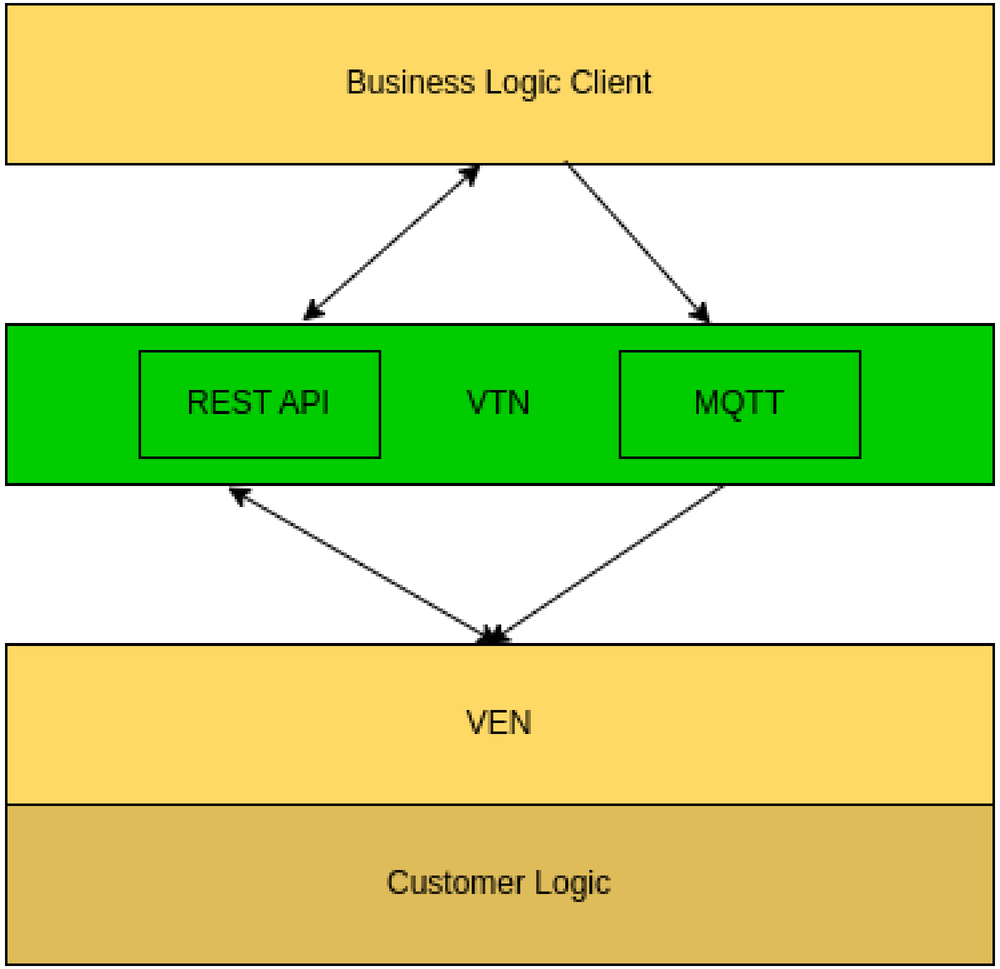
VTN is ADR speak for the element that VENs communicate with to receive events and publish reports

Executive Overview - differences between 2.0b and 3

OpenADR 3 defines a RESTful API, whereas 2.0b defines a SOAP web service

- SOAP is no longer a common technology choice
  - Requires complex back and forth message exchanges
  - Typically uses XML formatting which is verbose and therefore difficult for humans to read and debug
- RESTful APIs are extremely common today
  - **Re**presentational **S**tate **T**ransfer - a style of **A**pplication **P**rogramming **I**nterface used for Internet applications
  - Uses common HTTP verbs (GET, etc) to allow clients (e.g. VENs) to read or change the contents on a server (e.g. VTN)
  - Literally millions of developers worldwide are very familiar with how they work

Executive Overview - RESTful paradigm



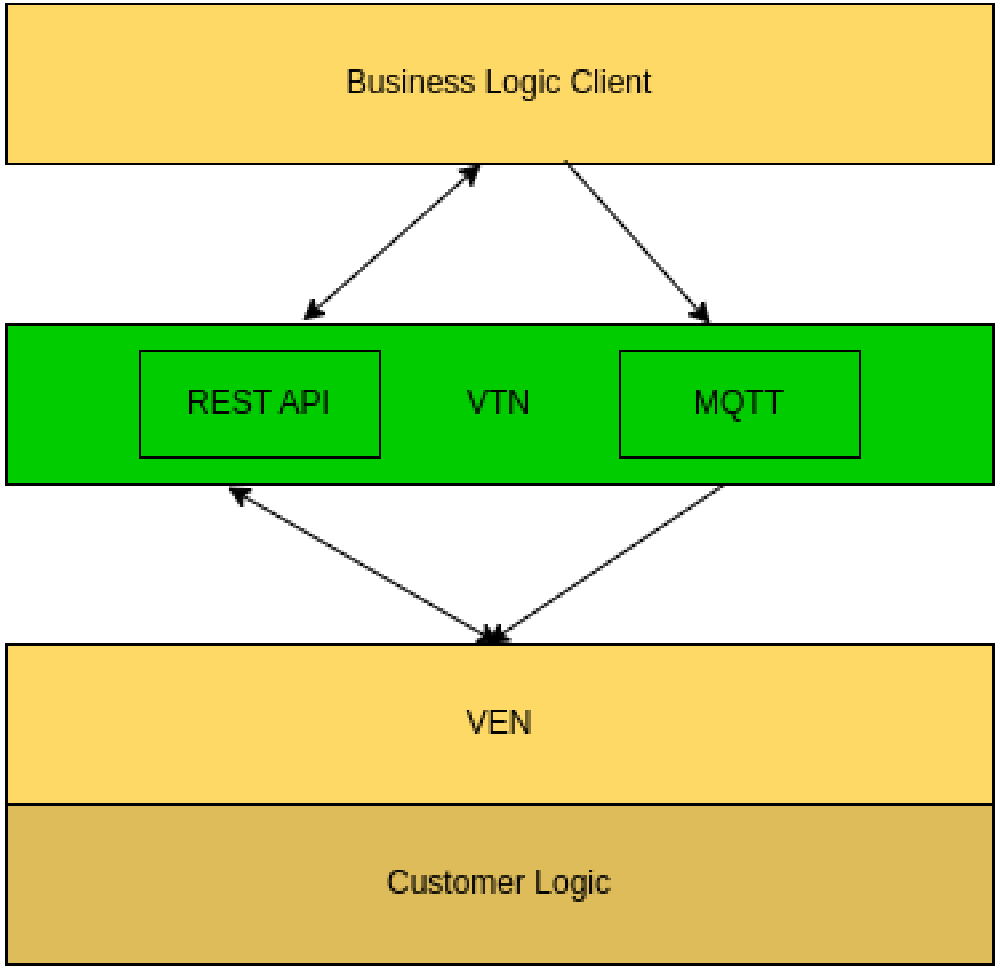
Business Logic (utility) clients and User (VEN) clients use the same API

Clients Create, Read, Update, Delete (CRUD) ‘Resources’ (e.g event) on the server (VTN)

The VTN has no logic, it’s just a storehouse for resources that clients manage

MQTT is a messaging queue to notify VENs when a resource has changed

Executive Overview - RESTful paradigm: example



Example:

A Utilities Business Logic client creates a ‘program’ resource and a number of ‘event’ resources on the VTN.

Any number of VEN clients can read those resources and react to them, or be notified via MQTT when they’re created or modified.

Executive Overview - RESTful paradigm: JSON

RESTful APIs generally use JSON formatted messages. Easy for humans and computers to read.

Example event (simplified):

```
{
        "eventName": "minimalEvent",
        "programID": "0",
        "intervalPeriod": {
                "start": "0000-00-00"
        },
        "intervals": [{
                "id": 0,
                "payloads": [{
                        "type": "SIMPLE",
                        "values": [1]
                }]
        }]
}
```

Executive Overview - Security

OpenADR 3 Security Model is consistent with today’s IT best practices

- OAuth2 client credential flow for application level security, TLS for on-the-wire security
- 2.0b uses Public Key Infrastructure (PKI) certificates which are expensive and only appropriate for systems managed by technical experts
  - Still available if usage is desired

API - Jargon Alert!!

OpenADR 3 defines a RESTful API with the following elements:

Information Model:

- ‘schemas’ defining ‘resources’ like programs, events, reports, etc

Endpoints and operations:

- ‘endpoints’ (URLs) and ‘operations’ (Create, Read, etc) allow clients to manage resources on the server
- Uses simple HTTP commands, much like a browser.
  - READ: HTTP GET <baseUrl>/openadr3/3.0.1/programs
  - CREATE: HTTP POST @program_data.json <baseUrl>/openadr3/3.0.1/programs

API - openapi format

The API is entirely defined in one file: openadr3.yaml.

- This file conforms to the openapi specification, a common format for defining APIs
- YAML is a Domain Specific Language (DSM) - kind of like JSON

The openapi ecosystem provides many handy tools to help make our API definition robust, complete, testable

- Online navigators
- Code generators - auto-generate servers and clients

API - YAML snippets

Example endpoint/operation

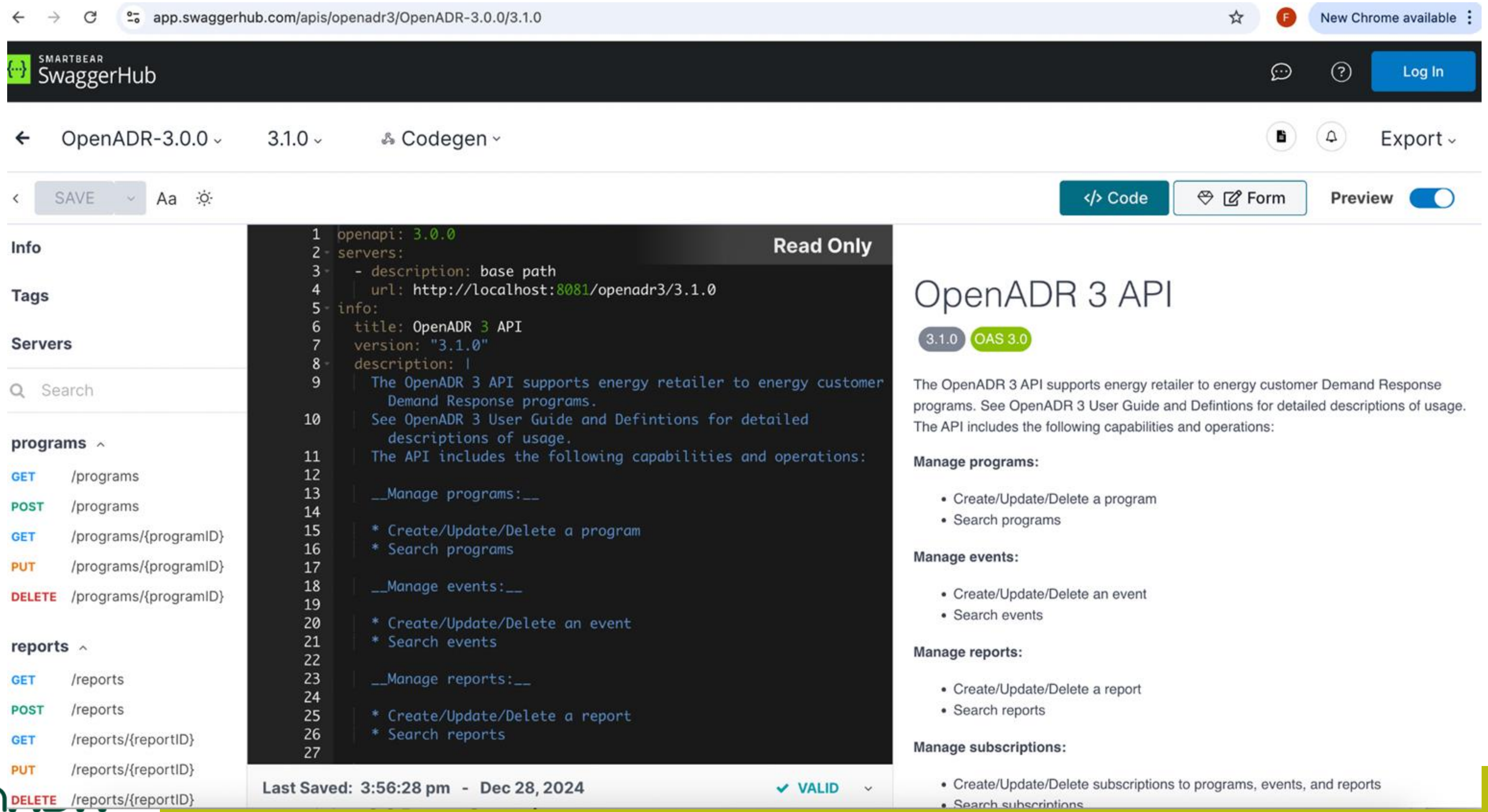
```
paths:
 /programs:
   get:
     tags:
       - programs
     summary: searches all programs
     operationId: searchAllPrograms
     description: |

More ….
```

Example Information Model element

```
components:
 schemas:
   # examples are provided at the element level to
aid readability and to support Reference
Implementation UI
   program:
     type: object
     description: Provides program specific
metadata from VTN to VEN.
     required:
       - programName
     properties:
       id:
         $ref: '#/components/schemas/objectID'
More ….
```

API - swaggerhub viewer



programs	
GET	/programs
POST	/programs
GET	/programs/{programID}
PUT	/programs/{programID}
DELETE	/programs/{programID}
reports	
GET	/reports
POST	/reports
GET	/reports/{reportID}
PUT	/reports/{reportID}
DELETE	/reports/{reportID}

```
openapi: 3.0.0
servers:
  - description: base path
    url: http://localhost:8081/openadr3/3.1.0
info:
  title: OpenADR 3 API
  version: "3.1.0"
  description: |
    The OpenADR 3 API supports energy retailer to energy customer Demand Response programs.
    See OpenADR 3 User Guide and Defintions for detailed descriptions of usage.
    The API includes the following capabilities and operations:

    __Manage programs:__

    * Create/Update/Delete a program
    * Search programs

    __Manage events:__

    * Create/Update/Delete an event
    * Search events

    __Manage reports:__

    * Create/Update/Delete a report
    * Search reports

```

Last Saved: 3:56:28 pm - Dec 28, 2024

OpenADR 3 API

3.1.0 OAS 3.0

The OpenADR 3 API supports energy retailer to energy customer Demand Response programs. See OpenADR 3 User Guide and Defintions for detailed descriptions of usage. The API includes the following capabilities and operations:

Manage programs:

- Create/Update/Delete a program
- Search programs

Manage events:

- Create/Update/Delete an event
- Search events

Manage reports:

- Create/Update/Delete a report
- Search reports

Manage subscriptions:

- Create/Update/Delete subscriptions to programs, events, and reports
- Search subscriptions

API - Summary

- Information Model includes:
  - Program
  - Event
  - Report
  - Ven
  - Resource (a load managed by a VEN)
  - Secondary schemas e.g. DateTime, etc
- Endpoints and Operations
  - All of the above (except secondaries) have URLs and operations to Create, Read, Update, and Delete

More detail out-of-scope for a slide deck! See https://app.swaggerhub.com/apis/openadr3/OpenADR-3.0.0/3.1.0

Definitions and User Guide

In addition to opanadr3.yaml, the specification includes two other documents:

Definitions (normative - aka includes “MUST” statements)

- Includes a number of sections describing various aspects such as security
- Includes ‘enumerations’ or strings with defined meanings that can be embedded in resources, e.g. “PRICE” in an event
- Note! We distinguish between the ‘protocol’ and ‘content’
    - Openadr3.yaml defines a protocol, i.e. what is the format of an event resource
    - Enumerations define content conventions, i.e. how to interpret values in an event

Definitions and User Guide 2

User Guide (informative)

- Describes Use Cases, scenarios and a lengthy set of examples
  - Conventions for how to define programs, events, and reports for a continuous pricing program
  - Load shed
  - How to use the targeting feature
  - Control timing and periodicity of reports
  - Sending Alerts
  - Inverter management
  - State of Charge reporting
  - And so on…

Reference Implementation (RI) and Test Tool

Borrowing from the successful Java Community Process (JCP) the Alliance has adopted the ‘three legged stool” approach to specification development:

- Specification documents (lots of groups stop here)
- Reference Implementation (VTN)
  - Serves an existence proof that the spec is implementable
  - Serves as a target to develop tests against
  - Serves as a model for other developers
- Tests
  - Used to evaluate any VTN and VEN for specification compliance

GitHub projects

GitHub is very common cloud platform for maintaining code or other artifacts (like documents). Allows teams to collaborate on incremental improvements. Project = repository = repo

- The OpenADR Alliance maintains several projects on GitHub:
  - Specification: openadr2.yaml, Definitions, User Guide
  - VTN Reference Implementation
  - Test Tool
  - …A few other secondary repos

Alliance members are encouraged to participate in maintaining repos

- Feature enhancements
- Bug fixes

Open source

The Alliance recognizes the value of open sourcing some repos. Broadens the scope of contributors:

- Leads to higher quality
- Leads to higher visibility and broader adoption

The Alliance may open source the specification and Reference Implementation

- Need to define governance and maintenance structure
  - Efficient mechanism to gain consensus among members on proposed changes
  - A small set of expert reviewers to ensure contributions are of high quality
  - Needs funding to operate

Testing and Certification

The Alliance provides testing and certification services.

Same Test Tool is used by implementers during product development and by Alliance for certification

- Test Tool can be executed locally and included in development workflow to ensure product is compliant at every stage of development.
- Test Tool is also available online for testing and certification.
  - To test or certify online, a team purchases a license from the Alliance
  - An Alliance certification manager monitors a certification session
    - Instead of multiple separate testing organization as in 2.0b, OpenADR 3 provides for 3rd party cert managers but uses the identical tool for all testing to ensure uniformity.

What’s next?

- 3.1.0
  - Includes notifications message queue (MQTT) as alternative to webhooks notifications
  - Compact payloads e.g. series of PRICES transmitted in a single payload
  - Object privacy. Targeting authorization enforced for VENs
  - Other minor-ish patches and features
- Open Source (see slide 19)

Price Communication Use Case

Bruce Nordman

Lawrence Berkeley National Laboratory

email: bnordman@lbl.gov

Price Communication

- More and more electricity customers have access to Highly Dynamic Prices
  - At least 20 countries and 180 retailers (priicer.com)
- More and more pay such prices
  - Most people in some countries
- Pricing is higher performing and lower cost than other coordination mechanisms

- Supporting price communication is a central use case for OpenADR 3

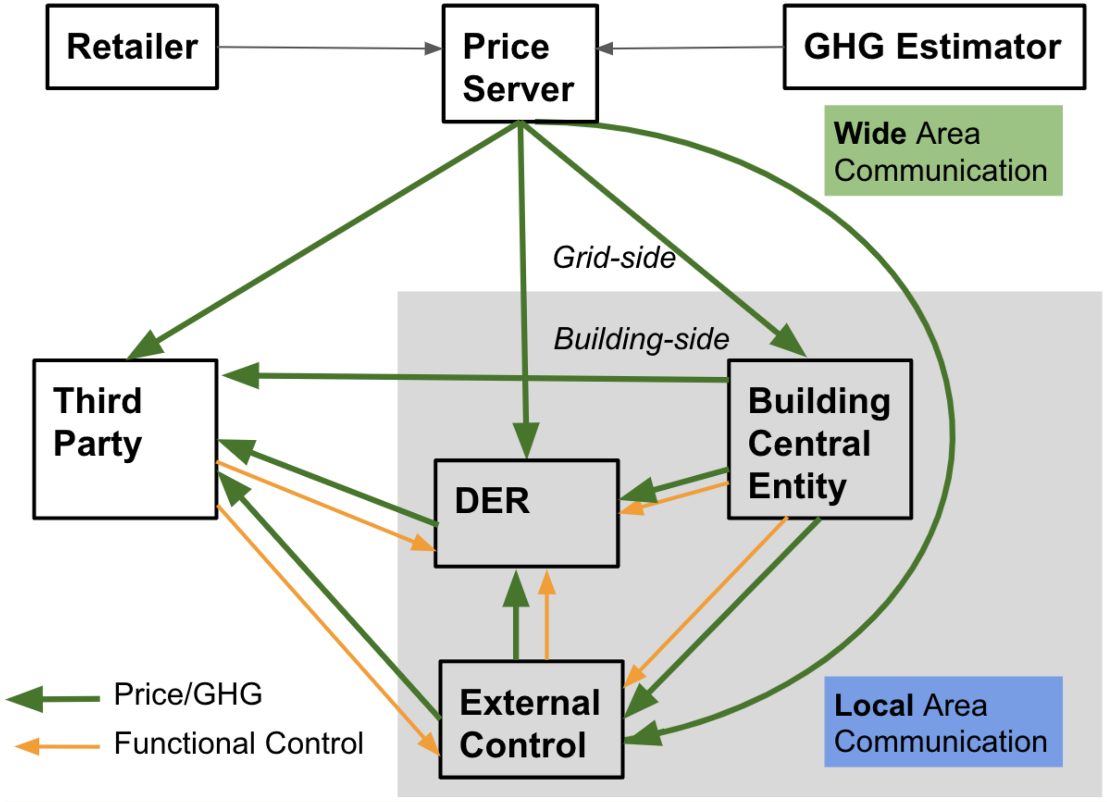
- Need to constrain ***maximum complexity of tariff***

Price Communication - System Architecture

- Price server (OpenADR 3 VTN) distributes prices
- Any of the four entities in the lower half can optimize control to prices

- OpenADR 3 is suitable for use intra-grid grid-customer, and intra-customer
  - Local communication (intra-customer) is a new concept for OpenADR
  - Gateway device ('Building Central Entity') is a VEN to grid price server and a VTN to customer DER (loads, storage, EVSE, etc.)
- 'Local Price' (within customer site) is determined by gateway and sent to loads

- This model doesn't constrain how prices are *set*, or how they are *used*



Highly Dynamic Prices (HDP)

Definition

- Intervals between **hourly** and 5 minutes
- Set no farther in advance than the day before
- **Different every day**

Necessary conditions to be responsive to actual grid conditions

NOT "RTP"

Components

- Electricity **prices** for a forecast period, usually 24 hours
  - May be guaranteed or a forecast
  - May include a different set of **export** price intervals

Non-Financial

- Marginal **GHG** emissions forecast (also intervals)
- Very occasional **alerts**
  - e.g. grid emergency, possible grid outage, fire, tsunami, air quality, …

That's it!

OpenADR 2.0b / 3.1 Comparison - Sending 3 hours of prices

```
<?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>
```

```
{
   "eventName": "pricingEvent",
   "programID": "44",
   "intervalPeriod": {
      "start": "2023-02-10T00:00:00.000Z",
      "duration": "PT1H"
   },
   "payloadDescriptors": [{
         "payloadType": "PRICES",
         "units": "KWH",
         "currency": "USD"
      }
   ],
   "intervals": [{
      "id": 0,
      "payloads": [{
         "type": "PRICES",
         "values": [
            0.17, 0.23, 0.08]
         }
      ]
   },
   ]
}
```

Price Server Functions

- Publishes prices for one or more tariffs from one or more retailers
- Once a day for day-ahead prices; once an hour (e.g.) for day-of prices
- No authentication required - prices are public information
  - No return data from customer site to VTN
- PUSH with MQTT encouraged

Local Price Server (local to customer site; not defined in OpenADR 3 standard)

- Functions
  - Receives prices from grid price server; Modifies as appropriate; Rebroadcasts price
  - Advertises its presence with mDNS for automatic device discovery
- Benefits
  - Only one device in building needs to know retailer/tariff
  - Easy to set local price
  - Facilitates other use cases (e.g. microgrid operation)

OpenADR 3.1 Price Example (17 hours)

```
{
    "eventName": "pricingEvent",
    "programID": "44",
    "intervalPeriod": {
        "start": "2023-02-10T00:00:00.000Z",
        "duration": "PT1H"
    },
    "payloadDescriptors": [{
            "payloadType": "PRICES",
            "units": "KWH",
            "currency": "USD"
        }
    ],
    "intervals": [{
        "id": 0,
        "payloads": [{
                "type": "PRICES",
                "values": [
                    0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23]
            }, {
                "type": "EXPORT_PRICES",
                "values": [
                    0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23]
            }, {
                "type": "GHGS",
                "values": [
                    0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23, 0.08, 0.17, 0.23]
            }
        ]
    },
    ] }
```

Thank you for participating!

Q & A

Q&A

Contact:

Rolf Bienert

Managing & Technical Director OpenADR Alliance

rolf@openadr.org

Bus Dev & Education Programs

Don Dulchinos

don@openadr.org

Marketing

Shannon Mayette

shannon@openadr.org