



OpenADR 3.0 – Introduction

June 6, 2023



Why a new OpenADR?

- 2.0b is nearly a decade old (and ideas designed well before that)
- 2.0b is widely used globally - no existing implementations need to change anything
- 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

- What we have set out to do
 - Make OpenADR future proof
 - Update functionalities that were not optimal
 - Add more use case specific mechanisms without changing the spirit of OpenADR (inform & motivate and decouple utility network space & assets from customer systems)
 - Create an appropriate security function set for receiving-only/one-way systems (price distribution, etc.)
 - Facilitate new usage models – in-building coordination and microgrid operation
 - Greatly simplify implementation for manufacturers
 - Improve “off the shelf” interoperability
 - Reduce perceived complexity and actual complexity
 - Maintain 2.0a and 2.0b viability

What's happening

- Task team finished draft and sent to working groups – April 19
- Start discussions and commenting – April 27
- Weekly calls as needed
- Goal: Get all comments by May 19
- Final draft after comment resolution and discussions – Aiming for mid June
- Discuss testing and certification – ongoing
 - Verify reference implementation
 - Other activities TBD
- Standard will be publicly available after the alliance comment resolution

Some clarifications

- 3.0 will not make 2.0 obsolete
 - We will need to review wording and messaging
 - Fully expect 2.0 to continue to be deployed
 - We do not expect deployed VTNs to change mid term
 - 2.0a and 2.0b will remain unchanged
- New VTNs will need to support 2.0 and 3.0 for certification after a grace period
 - May consider dropping the mandate to support 2.0a. For discussion

Now the meat – Core Documents

- F - OpenADR 3.0 Specification.yml
 - The core standard
- A - Introducing OpenADR 3.0.pdf - <https://www.dropbox.com/s/uvxqzcru5hghzut/A%20-%20Introducing%20OpenADR%203.0.pdf?dl=0>
 - Background, Process, Structure, Branding, Comparison with 2.0b, ...
- B - OpenADR 3.0 Definition.pdf
 - Core principles, summary of data elements, and summary of endpoints
- C - OpenADR 3.0 Enumerations.pdf
 - Defined strings for Events, Reports, etc.
- D - OpenADR 3.0 User Guide.pdf
 - Explanations and examples of numerous features and use cases
- E - OpenADR 3.0 Security Model.pdf
 - How security is applied to OpenADR 3.0
- G - Reference Implementation (available soon)
 - Open source code for a VTN

Some snippets

Price Communication Example

Metadata

```
"program": {
  "ID": "73536392",
  "createdDateTime": "2020-01-08T18:52:50",
  "programID": "ResHDPL5",
  "retailerName": "Pacific Gas and Electric",
  "retailerID": "PGE",
  "programName": "Residential Highly Dynamic Price Location 5",
  "country": "US",
  "principalSubdivision": "CA",
  "timeZoneOffset": 8,
  "activePeriod": {
    "start": "2023-01-01T09:30:47Z",
    "duration": "PT4H"
  },
  "programDescription": "http://www.pge.com/tariffs/ResHDPL5",
  "bindingEvents": False,
  "payloadTable": [
    {"payloadType": "PRICE", "units": "KWH", "currency": "USD"},
    {"payloadType": "GHG", "units": "GHG"}
  ]
}
```

Dynamic Data

```
"event": {
  "ID": 73536392,
  "createdDateTime": "2020-01-08T18:52:50",
  "programID": "ResHDPL5",
  "intervalPeriod": {
    "start": "2023-01-01T09:00:00Z",
    "duration": "PT1H"
  },
  "payloadDescriptors": [
    {"payloadType": "PRICE", "units": "KWH", "currency": "USD"}
  ],
  "intervals": [
    {"ID": 123456789, "payloads": [{"payloadType": "PRICE", "values": [15.1]}]},
    {"ID": 123456790, "payloads": [{"payloadType": "PRICE", "values": [18.4]}]},
    {"ID": 123456791, "payloads": [{"payloadType": "PRICE", "values": [21.1]}]},
    {"ID": 123456792, "payloads": [{"payloadType": "PRICE", "values": [19.8]}]}
  ]
}
```

To include GHG, *payloadDescriptors* adds:

```
{ "payloadType": "GHG", "units": "GHG" }
```

and each interval looks like:

```
{ "ID": 123456789, "payloads": [ {"payloadType": "PRICE", "values": [15.1]}, {"payloadType": "GHG", "values": "383"} ] },
```

The same example in 2.0B

Price Communication Example

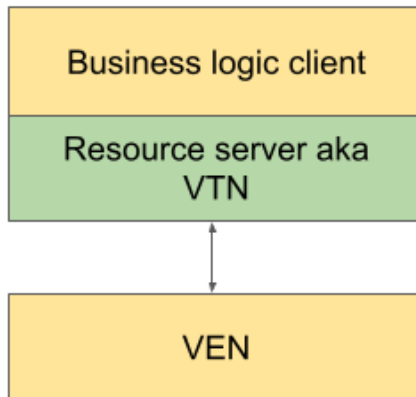
```
<oadr:oadrPayload>
  <oadr:oadrSignedObject>
    <oadr:oadrDistributeEvent ei:schemaVersion="2.0b">
      <pyId:requestID>OadrDisReq091214_043740_513</pyId:requestID>
      <ei:vtnID>TH_VTN</ei:vtnID>
      <oadr:oadrEvent>
        <ei:eiEvent>
          <ei:eventDescriptor>
            <ei:eventID>Event091214_043741_028_0</ei:eventID>
            <ei:modificationNumber>0</ei:modificationNumber>
            <ei:priority>0</ei:priority>
            <ei:eiMarketContext>
              <emix:marketContext>http://MarketContext1</emix:marketContext>
            </ei:eiMarketContext>
            <ei:createdDateTime>2023-05-09T12:37:40Z</ei:createdDateTime>
            <ei:eventStatus>far</ei:eventStatus>
          </ei:eventDescriptor>
          <ei:eiActivePeriod>
            <xcal:properties>
              <xcal:dtstart>
                <xcal:date-time>2023-05-09T13:00:00Z</xcal:date-time>
              </xcal:dtstart>
              <xcal:duration>
                <xcal:duration>PT4H</xcal:duration>
              </xcal:duration>
              <ei:x-eiNotification>
                <xcal:duration>PT24H</xcal:duration>
              </ei:x-eiNotification>
            </xcal:properties>
            <xcal:components/>
          </ei:eiActivePeriod>
          <ei:eiEventSignals>
            <ei:eiEventSignal>
              <strm:intervals>
                <ei:interval>
                  <xcal:duration>
                    <xcal:duration>PT1H</xcal:duration>
                  </xcal:duration>
                  <xcal:uid>
                    <xcal:text>0</xcal:text>
                  </xcal:uid>
                  <ei:signalPayload>
                    <ei:payloadFloat>
                      <ei:value>15.1</ei:value>
                    </ei:payloadFloat>
                  </ei:signalPayload>
                </ei:interval>
              </strm:intervals>
            </ei:eiEventSignal>
          </ei:eiEventSignals>
        </ei:eiEvent>
      </oadr:oadrEvent>
    </oadr:oadrDistributeEvent>
  </oadr:oadrSignedObject>
</oadr:oadrPayload>
```

```
      <xcal:uid>
        <xcal:text>0</xcal:text>
      </xcal:uid>
      <ei:signalPayload>
        <ei:payloadFloat>
          <ei:value>18.4</ei:value>
        </ei:payloadFloat>
      </ei:signalPayload>
    </ei:interval>
  </strm:intervals>
</ei:eiEventSignal>
</ei:eiEvent>
<oadr:oadrResponseRequired>always</oadr:oadrResponseRequired>
</oadr:oadrEvent>
</oadr:oadrDistributeEvent>
</oadr:oadrSignedObject>
</oadr:oadrPayload>
```

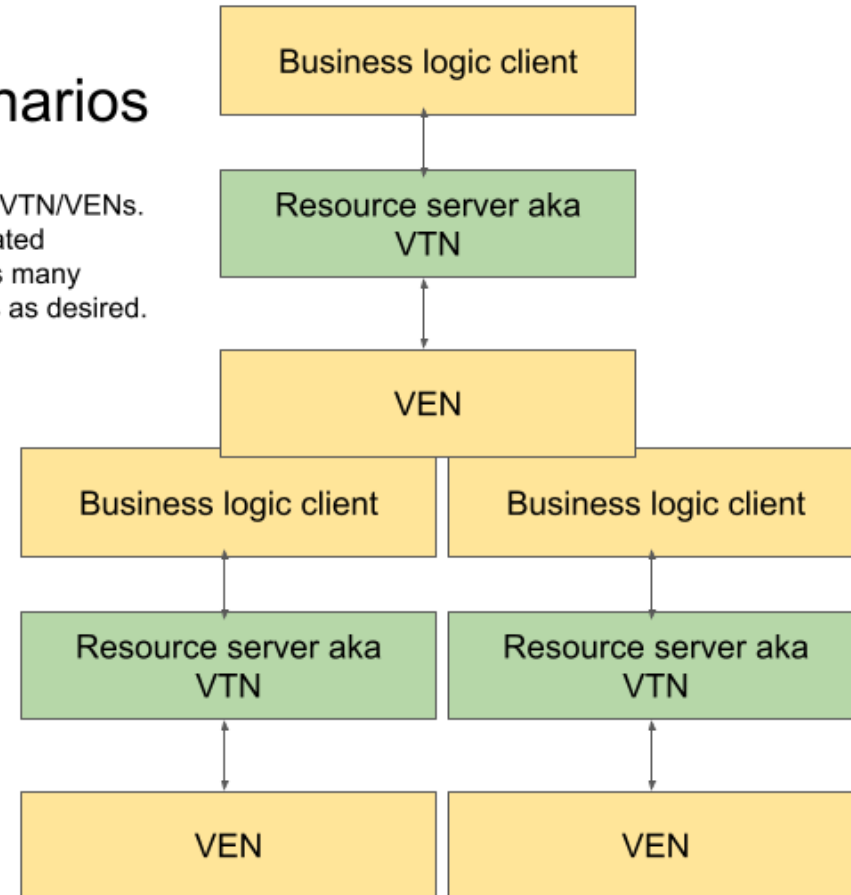

Some snippets (2)

OA 3.0 Implementation scenarios

Below is an implementation in which the the business logic and VTN are hosted by the same system and do not have communication between them (or use proprietary communication. This usage is equally valid.



Right is a stack of VTN/VENs. This can be replicated indefinitely, with as many layers or branches as desired.



Thank you!

Contact:

Rolf Bienert

Managing & Technical Director OpenADR Alliance

rolf@openadr.org

Education Programs

Don Dulchinos

don@openadr.org

Marketing

Shannon Mayette

shannon@openadr.org