

PD IEC/PAS 62746-10-1:2014



BSI Standards Publication

Systems interface between customer energy management system and the power management system

Part 10-1: Open Automated Demand Response
(OpenADR 2.0b Profile Specification)

bsi.

...making excellence a habit.
AVA Documents™
Document Sharing Hub

National foreword

This Published Document is the UK implementation of IEC/PAS 62746-10-1:2014.

The UK participation in its preparation was entrusted to Technical Committee PEL/57, Power systems management and associated information exchange.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2014.
Published by BSI Standards Limited 2014

ISBN 978 0 580 85223 7
ICS 33.200

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 March 2014.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------



PUBLICLY AVAILABLE SPECIFICATION PRE-STANDARD



**Systems interface between customer energy management system and the power management system –
Part 10-1: Open Automated Demand Response (OpenADR 2.0b Profile Specification)**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE **XD**

ICS 33.200

ISBN 978-2-8322-1435-0

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD.....	4
Original introductory material	6
1 Scope.....	9
2 Normative References	11
3 Non-Normative References.....	13
4 Terms and Definitions.....	14
5 Abbreviations	15
6 Overview	16
6.1 Node and Device Types	17
6.2 Energy Interoperation Services	18
6.3 Feature Sets	18
6.4 Assumptions.....	19
7 OpenADR 2.0 Feature Set Profiles	20
7.1 Differences between OpenADR 2.0a and OpenADR 2.0b	20
7.2 OpenADR 2.0b Feature Set Profile	21
7.2.1 Supported Services	21
7.2.2 Report Only VENs	21
7.2.3 Transport Mechanism	22
7.2.4 Security	22
8 OpenADR 2.0b Services and Data Models Extensions.....	23
8.1 OpenADR 2.0b EiEvent Service	23
8.1.1 Data Model.....	26
8.1.2 UML Models	27
8.2 Differences between OpenADR2.0a and 2.0b Event Mechanism	29
8.2.1 Event Targets and Resources.....	29
8.2.2 OpenADR 2.0b Signal Definitions	29
8.3 OpenADR 2.0b Report Service	33
8.3.1 Introduction	33
8.3.2 Core Reporting Operations	34
8.4 OpenADR 2.0b Registration Service	40
8.4.1 Service Operations	40
8.4.2 Registration Information	43
8.5 OpenADR 2.0b EiOpt Service	44
8.5.1 Service Operations	44
8.5.2 Detail Requirements	46
8.6 OpenADR Poll	47
8.7 Application Error Codes.....	50
9 Transport Protocol.....	51
9.1 Simple HTTP	51
9.1.1 PUSH and PULL implementation	51
9.1.2 Service Endpoint URIs.....	51
9.1.3 HTTP Methods	52
9.1.4 Failure Conditions	52
9.1.5 HTTP Response Codes	52

9.1.6	Message Timeouts	53
9.1.7	Message Retry / Quiesce Behavior	53
9.1.8	PULL Timing.....	53
9.1.9	HTTP Headers.....	53
9.2	Transport Specific Security.....	54
9.3	XMPP.....	54
9.3.1	PUSH and PULL implementation	54
9.3.2	Service Endpoints.....	54
9.3.3	Service Execution.....	55
9.3.4	Implementation of XMPP Features for OpenADR.....	55
9.3.5	Security Considerations.....	58
10	OpenADR 2.0 Security	59
10.1	Architecture and Certificate Types.....	59
10.2	Certificate Authorities	60
10.3	Certificate Revocation	60
10.4	TLS and Cipher Suites	60
10.5	System Registration Process.....	61
10.5.1	Certificate Fingerprints	61
10.6	Implementing XML Signatures for OpenADR 2.0 Message Payloads	61
10.6.1	Introduction to XML Signature	61
10.6.2	Components of XML Signatures	62
10.6.3	Creating XML Signatures.....	62
10.6.4	Verifying XML Signatures	63
11	Conformance.....	64
11.1	OpenADR 2.0 conformance statement.....	64
11.2	OpenADR 2.0b Profile Conformance Rules.....	64
11.2.1	EiEvent – from 2.0a	64
11.2.2	EiEvent – Additional 2.0b Conformance Rules	73
11.2.3	EiOpt.....	75
11.2.4	EiReport	78
11.2.5	EiRegisterParty	87
11.2.6	General Conformance Rules.....	89
11.3	Cardinality	95
11.4	Services used from OASIS Energy Interoperation V1.0 Standard	95
11.5	Services not currently used from OASIS EI	96
Annex A	– Detailed Report Description	97
Annex B	– B Profile Extensions.....	98
B.1	Overview	98
B.2	Report Extension.....	98
B.3	Signal Extensions.....	98
B.4	Other Extensions.....	98
Annex C	– oadrPoll Scenarios	100
C.1	Overview	100
C.2	Scenarios	100

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**SYSTEMS INTERFACE BETWEEN CUSTOMER ENERGY
MANAGEMENT SYSTEM AND THE POWER MANAGEMENT SYSTEM –**

**Part 10-1: Open Automated Demand Response
(OpenADR 2.0b Profile Specification)**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

A PAS is a technical specification not fulfilling the requirements for a standard, but made available to the public.

IEC-PAS 62746-10-1, submitted by Open ADR Alliance, has been processed by IEC project committee 118: Smart grid user interface. It is based on Open Automated Demand Response Communications Specification, also called OpenADR 2.0 Specification B Profile. The structure and editorial rules used in this PAS reflect the practice of the organization which submitted it.

The text of this PAS is based on the following document:

This PAS was approved for publication by the P-members of the committee concerned as indicated in the following document

Draft PAS	Report on voting
118/29/DPAS	118/32/RVD

Following publication of this PAS, the technical committee or subcommittee concerned may transform it into an International Standard.

This PAS shall remain valid for an initial maximum period of 3 years starting from the publication date. The validity may be extended for a single period up to a maximum of 3 years, at the end of which it shall be published as another type of normative document, or shall be withdrawn.

One of the tasks of PC 118 is to develop IEC standards on Demand Response.

This document has been approved as an IEC Publicly Available Specification (PAS) following the decision made by PC 118 at its second plenary meeting (March 2013).

It has been submitted by the OpenADR Alliance which has a category D liaison with PC 118 working groups.

The intent of this publication is to present a widely used technical solution for Demand Response and a promising trend of the industry. It has a large but partial compatibility with IEC CIM (IEC 61970 and IEC 61968). Considering the urgent need of industry that is starting to move ahead and cannot wait, the PAS can temporarily be used as a reference, and gives time for IEC to develop a formal technical specification (TS) or an international standard (IS) on Demand Response based on OpenADR 2.0b, fully compatible with IEC CIM which will then replace this IEC PAS.

The content of this document is based on the Open Automated Demand Response Communications Specification also called OpenADR 2.0 Specification B Profile. OpenADR is designed to facilitate automated DR actions at the customer location and has been field tested and deployed for several years in a number of DR programs in the world.

This standard contains attached Profile B Schema files. These files are intended to be used as a complement and do not form an integral part of the PAS.

A bilingual version of this publication may be issued at a later date.

Original introductory material

FOREWORD

The development of the **Open Automated Demand Response Communications Specification**, also called OpenADR, began in 2002 following the California electricity crisis. The California Energy Commission Public Interest Energy Research Program funded an OpenADR research program through the Demand Response Research Center (DRRC) at Lawrence Berkeley National Laboratory (LBNL). OpenADR development began in 2002 to support California's energy policy objectives to move toward dynamic pricing to improve the economics and reliability of the electric grid. Initial field tests focused on automating a number of event-based DR utility programs for commercial and industrial (C&I) customers. The DRCC research set out to determine if today's communications and information technologies could be used to automate Demand Response (DR) operations using standardized electricity price and reliability signals. This research, development, and deployment have led to commercial adoption of OpenADR. Today, utilities and governments worldwide are using OpenADR to manage the growing demand for electricity and peak capacity of the electric systems. This low cost communications infrastructure is used to improve the reliability, repeatability, robustness, and cost-effectiveness of DR.

OpenADR is a fundamental element of U.S. Smart Grid interoperability standards being developed to improve optimization between electric supply and demand. OpenADR is designed to facilitate automated DR actions at the customer location, whether it involves electric load shedding or shifting. OpenADR is also designed to provide continuous dynamic price signals such as hourly day-ahead or day-of real time pricing. OpenADR has been field tested and deployed in a number of DR programs in U.S and worldwide. While the scope of OpenADR focuses on signals for DR events and prices, significant work focuses on DR strategies and techniques to automate DR within facilities. OpenADR interacts with facility control systems that are pre-programmed to take action based on a DR signal, enabling a response to a DR event or a price to be fully automated, with no manual intervention.

The DRCC OpenADR 1.0 specification was donated to the Organization of Structured Information Standards (OASIS) to create a national standard for OpenADR. The OASIS' Energy Interoperation (EI) Technical Committee (TC) developed a standard to describe "an information model and a communication model to enable collaborative and transactive use of energy, service definitions consistent with the OASIS SOA Reference Model [SOA-RM], and XML vocabularies for the interoperable and standard exchange of dynamic price signals, reliability signals, emergency signals, communication of market participation information such as bids, load predictability and generation information." Considering that the goal of OASIS EI TC was more than DR and Distributed Energy Resources (DER), the EI TC created profiles within the EI Version 1.0 standard for specific applications within the Smart Grid. The OpenADR Alliance used the EI OpenADR profile as the basis for the OpenADR 2.0 Profile Specification defined in this document. OpenADR 2.0 defines profiles for DR and Distributed Energy Resources (DER), while keeping in mind the requirements of the diverse market and stakeholder needs.

INTRODUCTION

Development of the Demand Response (DR) market has resulted in a transition from manual DR to OpenADR in Automated DR (Auto-DR) programs. As of 2013, over 250 MW was enrolled in California commercial and industrial customers Auto-DR programs using OpenADR 1.0.¹ DR is defined as “...action taken to reduce electricity demand in response to price, monetary incentives, or utility directives so as to maintain reliable electric service or avoid high electricity prices.”² OpenADR 1.0 was developed to support Auto-DR programs and California’s energy policy objectives to move toward dynamic pricing to improve the economics and reliability of the electric grid. The recent developments have expanded the use of OpenADR to meet diverse market needs such as ancillary services (Fast DR), dynamic prices, intermittent renewable resources, supplement grid-scale storage, electric vehicles, and load as generation. For example, with real-time price information, an automated client within the customer facility can be designed to continuously monitor these prices and translate this information into continuous automated control and response strategies. This rationale is a fundamental element of the United States (U.S.) Smart Grid interoperability standards, which are developed to improve dynamic optimization of electric supply and demand.

OpenADR Communications have the following defining features:

- **Continuous, Secure, and Reliable** - Provides continuous, secure, and reliable two-way communications infrastructures where the end points at the end-use site receive and acknowledge the receipt of DR signals from the energy service providers.
- **Translation** - Translates DR event information to continuous Internet signals to facilitate DR automation. These signals are designed to interoperate with energy management and control systems, lighting, or other end-use controls.
- **Automation** - Receipt of the external signal is designed to initiate automation through the use of pre-programmed demand response strategies determined and controlled by the end-use participant.
- **Opt-Out** - Provides opt-out or override function to any participants for a DR event if the event comes at a time when changes in end-use services are not desirable.
- **Complete Data Model** - Describes a rich data model and architecture to communicate price, reliability, and other DR activation signals.
- **Scalable Architecture** - Provides scalable communications architecture to different forms of DR programs, end-use buildings, and dynamic pricing.
- **Open Standards** - Open standards-based technology such as Internet Protocol (IP) and web services form the basis of the communications model.

OpenADR is a communications data model, along with transport and security mechanisms, which facilitate information exchange between two end-points, the electricity service provider and the customer. It is not a protocol that specifies “bit-structures” as some communications protocols do, but instead relies upon existing open standards such as eXtensible Mark-up Language (XML) and Internet Protocol (IP) as the framework for exchanging DR signals. In some references the term “system,” “technology,” or “service” is used to refer to the features of OpenADR.

OpenADR is designed to facilitate automation of DR actions at the customer location, whether it involves electric load shedding or load shifting. We are often asked if the communications data model can be used for continuous operations. The answer is **yes**. Many emergency or

¹ Piette, Mary Ann, Girish Ghatikar, Sila Kiliccote, Ed Koch, Dan Hennage, Peter Palensky, and Charles McParland. 2009. Open Automated Demand Response Communications Specification (Version 1.0). California Energy Commission, PIER Program. CEC- 500- 2009- 063.

² U.S. Federal Energy Regulatory Commission (FERC), 2007 Assessment of Demand Response and Advanced Metering, Staff Report, available: <http://www.ferc.gov/legal/staff-reports/09-07-demand-response.pdf>

reliability DR events occur at specific times when the electric grid is strained. The OpenADR communications are designed to coordinate such signals with facility control systems (commercial, industrial, and residential). OpenADR is also designed to provide continuous dynamic price signals such as hourly day-ahead or day-of real time pricing. With such price information an automated client can be configured to continuously monitor these prices and translate this information into continuous automated control and response strategies within a facility. Several reports present the history of OpenADR 1.0 research.³ This OpenADR 2.0 profile specification covers the signaling data models for price and reliability signals to both wholesale and retail markets in the U.S.

OpenADR provides the following benefits:

- **Open Specification**—Provides a standardized DR communications and signaling infrastructure using open, non-proprietary, industry-approved data models that can be implemented for both dynamic prices and DR emergency or reliability events.
- **Flexibility**—Provides open communications interfaces and protocols that are flexible, platform-independent, interoperable, and transparent to end-to-end technologies and software systems.
- **Innovation and Interoperability**—Encourages open innovation and interoperability, and allows controls and communications within a facility or enterprise to build on existing strategies to reduce technology operation and maintenance costs, stranded assets, and obsolesce in technology.
- **Ease of Integration**—Facilitates integration of common Energy Management and Control Systems (EMCS), centralized lighting, and other end-use devices that can receive Internet signals (such as XML).
- **Supports Wide Range of Information Complexity** – Can express the information in the DR signals in a variety of ways to allows for systems ranging from simple end devices (e.g., thermostats) to sophisticated intermediaries (e.g., aggregators) to receive the DR information that is best suited for its operations.
- **Remote Access**— Facilitates opt-out or override functions for participants to manage standardized DR-related operation modes to DR strategies and control systems.

The OpenADR Alliance is the primary authority for the development and adoption of OpenADR, leveraging the OpenADR 1.0 activities and OASIS Energy Interoperation (EI) Technical Committee's Version 1.0 standard.⁴ The OpenADR profile within OASIS EI Version 1.0 standard is the basis for the OpenADR 2.0 profile specification and is referenced as appropriate in this document.

³ These reports are available at <http://drrc.lbl.gov/drrc-pubsall.html>:

Piette, M.A., S. Kiliccote, G. Ghatikar, Design and Implementation of an Open, Interoperable Automated Demand Response Infrastructure, Proceedings of the Grid-Interop Forum, October 2007, LBNL-63665.

Koch, E., M.A. Piette, Architecture Concepts and Technical Issues for an Open, Interoperable Automated Demand Response Infrastructure. Proceedings of the Grid-Interop Forum, October 2007. LBNL-63664.

Piette, M.A, D. Watson, N. Motegi, S. Kiliccote Automated Critical Peak Pricing Field Tests: 2006 Pilot Program Description and Results, August, 2007. LBNL-62218.

Motegi, N., M.A. Piette, D.S. Watson, S. Kiliccote, P. Xu. Introduction to Commercial Building Control Strategies and Techniques for Demand Response, May 2007. LBNL-59975.

⁴ Energy Interoperation OASIS Committee Specification, Energy Interoperation Version 1.0, December 2011. <http://www.oasis-open.org/committees/download.php/44364/energyinterop-v1.0-csprd03.zip>

1 Scope

The OpenADR 2.0 profile specification is a flexible data model to facilitate common information exchange between electricity service providers, aggregators, and end users. The concept of an open specification is intended to allow anyone to implement the two-way signaling systems, providing the servers, which publish information (Virtual Top Nodes or VTNs) to the automated clients, which subscribe the information (Virtual End Nodes, or VENs).

This OpenADR 2.0 profile specification covers the signaling data models between VTN and VEN (or VTN/VEN pairs) and does include information related to specific DR electric reduction or shifting strategies, which are taken at the facility. In particular, OpenADR 2.0 supports the following services from OASIS EI Version 1.0 standard or subset thereof. Extensions to these services are included to meet the DR stakeholder and market requirements:

1. **Registration (EiRegisterParty):** Register is used to identify entities such as VEN's and parties. This is necessary in advance of an actor interacting with other parties in various roles such as VEN, VTN, tenderer, and so forth.
2. **Enrollment (EiEnroll):** Used to enroll a Resource for participation in DR programs. This establishes a relationship between two actors as a basis for further interactions. (Planned for future releases)
3. **Market Contexts (EiMarketContext):** Used to discover program rules, standard reports, etc. Market contexts are used to express market information that rarely changes, and thereafter need not be communicated with each message. (Planned for future releases)
4. **Event (EiEvent):** The core DR event functions and information models for price-responsive DR. This service is used to call for performance under a transaction. The service parameters and event information distinguish different types of events. Event types include reliability events, emergency events, and more – and events MAY be defined for other actions under a transaction.
5. **Quote or Dynamic Prices (EiQuote):** EiDistributeQuote for distributing complex dynamic prices such as block and tier tariff communication. These are sometimes referred to as *price signals*; such signals are indications of a possible tender price – they are not themselves actionable. Such services can be used to implement the functionality for energy market interactions or transactional energy. (Planned for future releases)
6. **Reporting or Feedback (EiReport):** The ability to set periodic or one-time information on the state of a Resource (response).
7. **Availability (EiAvail):** Constraints on the availability of Resources. This information is set by the end node and indicates when an event may or may not be accepted and executed by the VEN with respect to a Market Context. Knowing the Availability and Opt information for its VENs improves the ability of the VTN to estimate response to an event or request. (Planned for future releases)
8. **Opt or Override (EiOpt):** Overrides the EiAvail; addresses short-term changes in availability to create and communicate Opt-in and Opt-out schedules from the VEN to the VTN.

These OpenADR 2.0 services in this specification provide information that is pertinent to DR, pricing, and DER communication requirements. These services make no assumption on specific DR electric load control strategies within the resource or market-specific contractual or business agreements between electricity service providers and their customers.

OpenADR uses an application-level data model, which is independent of transport mechanisms. For the purposes of interoperability, OpenADR 2.0 provides basic transport mechanisms and their relevant interaction patterns (e.g., PUSH information vs. PULL information) to address different stakeholder needs.

OpenADR 2.0 specifies the necessary level of security that is essential to meet the U.S. Cyber Security requirements for such purposes as data confidentiality, integrity, authentication and message-level security. Such security requirements are essential for non-repudiation and to mitigate any resulting Cyber Security risks.

OpenADR 2.0 provides a clear set of mandatory and optional attributes within each of the services to meet the broader interoperability, testing and certification requirements, while creating feature-sets with different product profiles to address today's market needs as well

as future requirements that are closely aligned to meet OpenADR goals and national interoperability requirements for Smart Grid standards.

The different product certification levels for OpenADR include OpenADR 2.0a, OpenADR 2.0b, and OpenADR 2.0b “Energy Reporting only” VENs (depicted in Figure 1). VTN certification for 2.0a will end with publication of this document, and existing implementations of 2.0a VTNs must upgrade to the OpenADR2.0b standard. For this reason, Figure 1 has no column for 2.0a VTN. 2.0b VTNs must support 2.0a VENs (and therefore comply with the OpenADR2.0a standard). VENs can be certified using the 2.0a, the 2.0b, and a 2.0b “Energy reporting only” profile. An OpenADR 2.0c or new market-specific profiles may be specified in the future. This profile specification describes OpenADR 2.0b. For the final 2.0a features, please refer to the respective specification, which is available on the OpenADR Alliance’s website – <http://www.openadr.org/>.

	VTN	VEN		
	B	A	B	B (Energy Reporting only)
Services and Functions Support				
EiEvent				
Limited Profile (2.0a specification)	M	M	NA	NA
Full Profile	M	NA	M	NA
EiOpt				
Full Profile	M	NA	M	NA
EiReport				
Full Profile	M	NA	M*	M*
EiRegisterParty				
Full Profile	M	NA	M	M
Transport Protocols				
Simple HTTP	M	M	O-1	O-1
XMPP	M	O	O-1	O-1
Security Levels				
Standard	M	M	M	M
High	O	NA	O	O
M - Mandatory O - Optional O-1 - Optional, but at least one of them must be supported NA - Not available for profile * Optional features available				

Figure 1 – OpenADR 2.0 Certification Levels

2 Normative References

- **[OpenADR 2.0 PICS]** – Source: OpenADR website, <http://www.openadr.org>
- **[OpenADR 2.0 Certificate Policy]** – Source: OpenADR website, <http://www.openadr.org>
- **[OASIS EI 1.0]** Energy Interoperation OASIS Committee Specification 02, *Energy Interoperation Version 1.0*, <http://docs.oasis-open.org/energyinterop/ei/v1.0/cs02/energyinterop-v1.0-cs02.html>, February 2012.
- **[OASIS EMIX 1.0]** EMIX OASIS Committee Specification 02, *Energy Market Information Exchange 1.0*, <http://docs.oasis-open.org/emix/emix/v1.0/cs02/emix-v1.0-cs02.html>, January 2012.
- **[OASIS WS-Calendar]** WS-Calendar OASIS Committee Specification 1.0, *WS-Calendar*, <http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.html>, July 2011.
- **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- **[RFC2246]** T. Dierks, C. Allen, *The TLS Protocol Version 1.0*, <http://www.ietf.org/rfc/rfc2246.txt>, IETF RFC 2246, January 1999.
- **[RFC2616]** R. Fielding et. al., *Hypertext Transfer Protocol -- HTTP/1.1*, <http://www.ietf.org/rfc/rfc2616.txt>, IETF RFC 2616, June 1999.
- **[RFC3275]** D. Eastlake, J. Reagle, D. Solo, *(Extensible Markup Language) XML-Signature Syntax and Processing*, <http://www.ietf.org/rfc/rfc3275.txt>, IETF RFC 3275, March 2002.
- **[RFC3986]** T. Berners-Lee et. al., *Uniform Resource Identifier (URI): Generic Syntax*, <http://www.ietf.org/rfc/rfc3986.txt>, IETF RFC 3986, June 1999.
- **[RFC4346]** T. Dierks, E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.1*, <http://www.ietf.org/rfc/rfc4346.txt>, IETF RFC 4346, April 2006.
- **[RFC5246]** T. Dierks, E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, <http://www.ietf.org/rfc/rfc5246.txt>, IETF RFC 5246, April 2008.
- **[RFC6120]** P. Saint-Andre, *Extensible Messaging and Presence Protocol (XMPP): Core Version 1.0*, <http://www.ietf.org/rfc/rfc6120.txt>, IETF RFC 6120, March 2011.
- **[RFC6121]** P. Saint-Andre, *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence*, <http://www.ietf.org/rfc/rfc6121.txt>, IETF RFC 6121, March 2011.
- **[RFC6122]** P. Saint-Andre, *Extensible Messaging and Presence Protocol (XMPP): Address Format*, <http://www.ietf.org/rfc/rfc6122.txt>, IETF RFC 6122, March 2011.
- **[SOA-RM]** SOA-RM OASIS Standard, *OASIS Reference Model for Service Oriented Architecture 1.0*, <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html>, October 2006.

- **[XEP-0030]** Joe Hildebrand et. al., *XEP-0030: Service Discovery*,
<http://xmpp.org/extensions/xep-0030.html>, June 2006.

3 Non-Normative References

- OpenADR 2.0a Profile Specification
- UCA OpenSG OpenADR security profile
- IRC and NAESB requirements/use-cases
- NIST Special Publication 800-131A
- NAESB REQ.21 Energy Services Provider Interface (ESPI), Version 1.0 October 2011(Green Button)

4 Terms and Definitions

The OpenADR Alliance: The OpenADR Alliance is comprised of industry stakeholders that are interested in fostering the deployment of low-cost price- and reliability-based demand response communication protocol by facilitating and accelerating the development and adoption of OpenADR standards and compliance with those standards. These include de facto standards based on specifications published by LBNL in April 2009, as well as Smart Grid-related standards emerging from OASIS, UCAIug, NAESB, and IRC.

OpenADR 2.0 Profile Specification: The OpenADR 2.0a and 2.0b Profile Specifications provide specific implementation related information in order to build an OpenADR enabled device or system. Developers shall use the Profile Specification in conjunction with the schemas, sample payloads, PICS and test plans.

OASIS Energy Interoperation (EI): Energy Interoperation standard describes information and communication model to coordinate energy supply, transmission, distribution, and use, including power and ancillary services, between any two parties, such as energy suppliers and customers, markets and service providers, in any of the domains defined in the Smart Grid. The EI 1.0 standard was used as a basis for OpenADR 2.0 Profile Specification.

Demand Response: A mechanism to manage customer load demand in response to supply conditions, such as prices or availability signals.

Slow DR: Demand Response where the signals are sent significantly before the events are called, such as day-ahead.

Fast DR: Fast Demand Response or Fast DR refers to programs that require a (much) faster than usual response time. While typical peak shaving DR programs have minutes, if not hours or days, of lead-time, these programs have lead times of seconds (e.g., 4 second response time) used for load balancing and frequency stabilization (e.g., ancillary services and regulation services)

PUSH/PULL operations: OpenADR 2.0 can be used in either PULL mode (VEN pulling information from VTN) or in a PUSH mode in the simple HTTP transport. The XMPP transport uses a PUSH model, although VENs can still make requests of the VEN, excluding the use of oadrPoll.

Simple HTTP: Simple HTTP in OpenADR 2.0 (a/b) refers to an HTTP implementation that uses HTTP POST over TLS to propagate OpenADR payloads.

The upper-case key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “NOT RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

5 Abbreviations

AutoDR: Automated Demand Response

DER: Distributed Energy Resources

DR: Demand Response

DRRC: Demand Response Research Center

DUT: Device Under Test

EI: Energy Interoperation

HTTP: Hyper Text Transfer Protocol

IRC: ISO/RTO Council

ISO: Independent Systems Operator

JID: Jabber Identifiers

LBNL: Lawrence Berkeley National Laboratory

NAESB: North American Energy Standards Board

OASIS: Organization of Structured Information Standards

OpenADR: Open Automated Demand Response

PICS: Protocol Implementation Conformance Statement

RTO: Regional Transmission Operators

SASL: Simple Authentication and Security Layer

Simple HTTP: Limited REST transport protocol

SOAP: Simple Object Access Protocol

TC: Technical Committee

UCAIug: Utilities Communications Architecture International Users Group

VEN: Virtual End Node

VTN: Virtual Top Node

XMPP: XML Messaging and Presence Protocol

6 Overview

This section gives an overview of the message exchanges, the roles, and actors supported within OpenADR 2.0 Profile Specifications. It contains the following elements, used to develop test and certification framework for Smart Grid and customer systems interoperability:

1. A set of data models derived from the OASIS Energy Interoperation 1.0 standard.
2. A set of services for performing various functions and operations for the exchange of the data models, also derived from the OASIS Energy Interoperation 1.0 standard.
3. A set transport mechanisms for implementing the services. The transport mechanisms rely upon standard-based IP communications such as HTTP and XML Messaging and Presence Protocol (XMPP).
4. A set of security mechanisms for securing each of the transport mechanisms.
5. OpenADR 2.0 Schemas (separate document)

The OASIS Energy Interoperation (EI) 1.0 standard describes the coordination of energy supplies, transmission, distribution, and usage. However, as shown in Figure 2, the features required within the OpenADR 2.0 Profile Specifications are a subset of this Energy Interoperation 1.0 standard. The reason for this development is twofold – 1) By having a strict and clear subset of features to support OpenADR, devices can be clear about what features they must support in order to participate in the OpenADR ecosystem; 2) by being a referenced subset, devices can validate against the Energy Interoperation 1.0 standard, and participate in that ecosystem as well. These requirements are critical to maintain interoperability and reference the original standard.

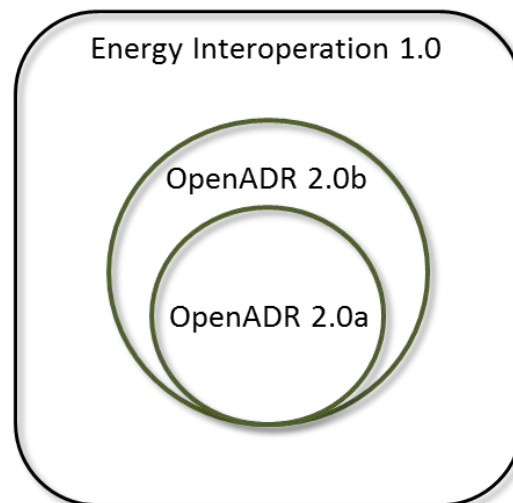


Figure 2 – Relationship between OASIS Energy Interoperation 1.0 Standard and OpenADR 2.0 Profiles

The above Figure 2 shows the relative relationship between the complete data model and services features of the Energy Interoperability 1.0 standard and OpenADR 2.0 profiles. The diagram also shows how the different profile subsets of OpenADR 2.0 relate to the complete OpenADR 2.0 feature set.

Message exchanges in OpenADR 2.0 support services related to communicating information about Demand Response events. Networks of OpenADR nodes must be able to query for active or pending events, register themselves, schedule events, and send reports. OpenADR nodes must also be able to refine and update previously sent information. For instance, an OpenADR node reporting DR events to nodes downstream must be able to cancel a previously scheduled event if this becomes necessary.

Nodes in these networks are divided into two groups: nodes that publish and transmit information about events to other nodes (e.g., utilities), and nodes that receive the communications respond to that information (e.g., end users). The upstream nodes that publish information about upcoming events are called Virtual Top Nodes (VTNs); the downstream nodes that receive this information are called Virtual End Nodes (VENS).

These nodes may communicate using a variety of protocols. They may communicate using HTTP in either PUSH mode (where the VTN initiates communication) or in a PULL mode (the VEN requests information from the VTN to begin a series of message exchanges). The VTNs/VEs may also communicate over other transport mechanisms such as XML Messaging and Presence Protocol (XMPP).

The profiles support end devices with varying degree of capabilities. The profile levels are 2.0a, for simple devices, and 2.0b, for more sophisticated devices. An OpenADR 2.0c profile may be specified in the future. Profile 2.0b VEN devices support a sub-profile referred to as Report Only. Each profile has a defined set of optional and mandatory functionality. In general, this functionality is only optional for VEN, as VTN must support all functionality for a given profile in order to maintain interoperability.

6.1 Node and Device Types

Following the notion from the OASIS Energy Interoperability 1.0 standard, OpenADR 2.0 uses the definitions of Virtual Top Nodes (VTNs) and Virtual End Nodes (VEs). For any interaction between actors using OpenADR 2.0 to communicate, one actor is designated the Virtual Top Node and the remainder are the Virtual End Nodes. All communications are between a VTN and one or more VEN's. There is no peer-to-peer communication in OpenADR 2.0, i.e., VTNs do not communicate directly with other VTNs and likewise VEs do not communicate directly with other VEs.

Generally in an interaction, the VTN acts as the server, providing information to the VEN, which themselves respond to the information. For instance, a VTN would be the entity to announce a DR event; VEs hear about DR events and respond. The response may be to reduce power to some devices. The response could also be to propagate the signal further downstream to other VEN's. In this case, the VEN would become the VTN for the new interaction (as depicted in Figure 3).

For the purpose of device development, the OpenADR Alliance always tests the interface between a VTN and a VEN, where either node can be the Device Under Test (DUT). Intelligence build into the systems not related to the OpenADR 2.0 message exchange is not part of the OpenADR Alliance testing program.

Virtual Top Node (VTN): An entity that is responsible for communicating grid conditions (e.g., prices, reliability events, etc.) to other entities (i.e., VEs) that control demand side resources. The VTN is able to communicate with both the Grid and the VEN devices or systems in its domain. A VTN may take the role of a VEN interacting with another VTN.

Virtual End Node (VEN): The VEN has operational control of a set of resources and/or processes and is able to control the electrical energy demand of these in response to an understood set of Smart Grid messages (i.e., DR signals). The VEN may be either a producer or consumer of energy. The VEN is able to communicate (2-way) with a VTN receiving and transmitting Smart Grid messages that relay grid situations, conditions, or events. A VEN may take the role of a VTN in other interactions.

Although within any interaction, one actor is designated the VTN and the remainder are VEs (moreover, most interactions have exactly one VTN and one VEN), sets of actors can be arranged in any hierarchy, by allowing actors to act as VEs for some interactions and VTNs for others.

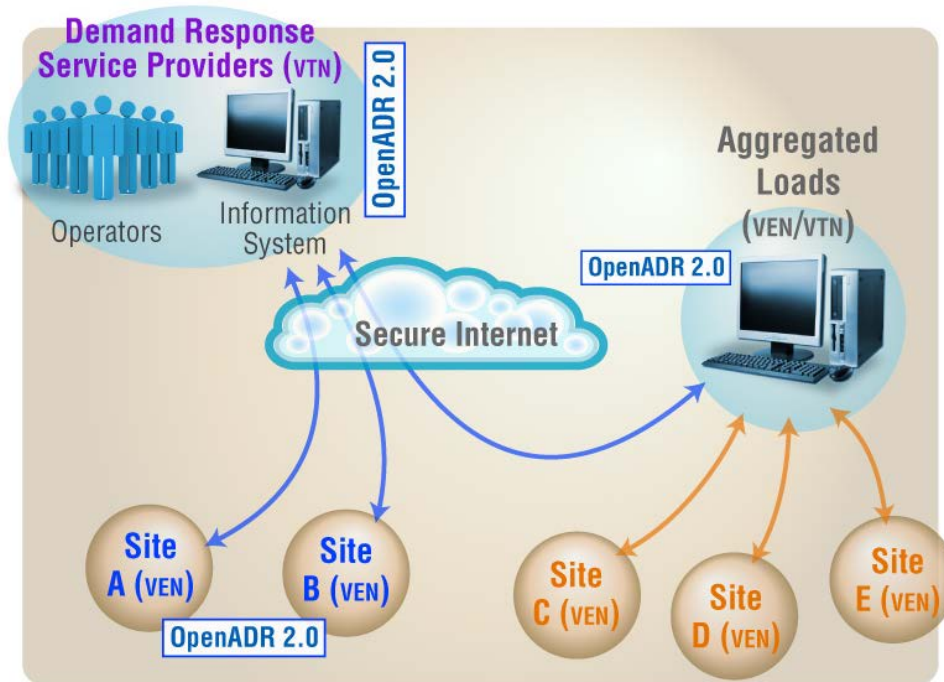


Figure 3 – Possible relationships of VTN and VEN

As illustrated in Figure 3, any combination of VTN and VEN is possible through a utility/ISO (service provider or server) to sites (customers). Also, as shown above, systems can function as a VEN to a VTN in a higher layer of the hierarchy and as a VTN to subordinate VENs. In either of these architectural scenarios, an operation can be initiated by the VTN to a VEN (PUSH pattern) or a VEN can request it from a VTN (PULL pattern). The exchanged events in either direction can be independent from each other and the OpenADR Alliance does not define how the nodes react to the information. In nodes, which support both the VTN and VEN interfaces (e.g., aggregators) there are no specifications or constraints on how messages arriving at the VEN interface are coupled or translated into any subsequent messages that may be sent from the VTN interface and vice versa. They are treated as completely independent interfaces and both will be evaluated and tested independently to assure adherence to the profile specification and interoperability. A specific deployment scenario depends on an agreement between the utility/ISO and the participating sites.

6.2 Energy Interoperation Services

The OASIS Energy Interoperation specification encompasses a number of services for collaborative and transactive use of energy of which only 4 services are applicable for OpenADR (in the current B profile). OpenADR 2.0 uses a profiled subset of the Energy Interoperation services tailored to meet the OpenADR needs, but still conforming to the Energy Interoperation Specification.

These services are EiRegisterParty, EiEvent, EiReport, and EiOpt. For further information on these services, consult Section 7.

6.3 Feature Sets

The OpenADR 2.0 specification has been designed to support a variety of end-use devices, from simple to more sophisticated. Accordingly, not all devices need to support all OpenADR capabilities. For example, although OpenADR describes how VTNs may report lists of active and pending future events, a simple residential programmable thermostat may only need information about the next pending event. In order to prevent onerous requirements on all devices, OpenADR currently defines two feature sets, each of which represent a subset of OpenADR functionality. The feature sets are 2.0a (or simply A), and 2.0b (or simply B), and

possibly a future specification of a 2.0c profile. The purpose of the profiles is to create a range of functionality that can be supported by devices as simple as a thermostat (profile A) to more complex IT based systems such as might be used by aggregators (profile B). Additional feature sets can be established to accommodate additional market requirements.

6.4 Assumptions

This is a list of operating assumptions regarding correct functional behavior between VENs and VTNs:

- Both the VTN and VEN have a reasonably accurate awareness of the current time. How that is accomplished is device and/or deployment specific.

7 OpenADR 2.0 Feature Set Profiles

The OpenADR Alliance has defined several feature sets to accommodate a variety of different devices for varying applications. Each Feature Set Profile describes the required services to be implemented as well as the accompanying operations and attributes. Please refer to the OpenADR 2.0 Protocol Implementation Conformance Statement ([OpenADR 2.0 PICS] – refer to separate document for alliance members only) for more information regarding the required services. Additional profiles will be added if required by the market participants.

The OpenADR 2.0 services discussed in the Feature Set Profiles can be found in section 8. All services are subsets of the OASIS Energy Interoperation Specification and validate individually with the relevant schemas.

This document outlines the OpenADR 2.0b profile. Any following profiles will have their own sets of documents and will follow. Summary information for these profiles is described in this document.

7.1 Differences between OpenADR 2.0a and OpenADR 2.0b

The only service supported by the A profile is the EiEvent service. The EiEvent object is simplified in the A profile with the following constraints:

- Only one signal per event is allowed and that signal must be the OpenADR well-known signal SIMPLE.
- There is a limited event targeting with only venID, groupID, resourceID, and partyID supported. (eiEvent:eiTarget).
- Targeting at the signal level with device classes is not supported (eiEventSignal:eiTarget:endDeviceAsset).
- Baselines are not supported (eiEvent:eiEventSignals:eiEventBaseline).
- modificationDateTime and modificationReason are not supported.
- The endpoint URL for simple HTTP in 2.0b is:
`https://<hostname>(:port)/(prefix/)OpenADR2/Simple/2.0b/<service>`
(note the additional “2.0b/” prefix compared to the 2.0a specification).

B profile VTNs must be able to interoperate with both A and B VENs. B VENs may optionally support the A profile, but are not required to. If a B profile VEN also supports the A profile and communicates with an A profile VTN, it must constrain its behavior as follows:

- Use only the EiEvent service
- Do not send oadrPoll
- Root Payload Element schemaVersion attribute must not be sent as part of any payload
- oadrResponse:venID must not be sent as part of the oadrResponse payload

With publication of this document, no 2.0a VTNs will be certified any more, and after a grace period, 2.0a VTN implementations must update to the 2.0b standard. This means that a certified VTN must support both 2.0a and 2.0b VEN implementations.

When a B profile VTN communicates with an A profile VEN, it must constrain its behavior as follows:

- Use only the EiEvent service
- Events must contain no more than one EiEventSignal
- Event signalName must be “SIMPLE” as defined in the A profile conformance rule 7
- The event signalPayload values must be limited to 0,1,2,3 as defined in A conformance rule 9
- The following optional B profile eiEvent schema elements must not be sent as part of the oadrDistributeEvent payload to the VEN:
 - eiEvent:eventDescriptor:modificationDateTime
 - eiEvent:eventDescriptor:modificationReason
 - eiEvent:eiTarget:aggregatedPnode
 - eiEvent:eiTarget:endDeviceAsset
 - eiEvent:eiTarget:meterAsset
 - eiEvent:eiTarget:pnode
 - eiEvent:eiTarget:serviceArea
 - eiEvent:eiTarget:serviceDeliveryPoint
 - eiEvent:eiTarget:serviceLocation
 - eiEvent:eiTarget:transport Interface
 - eiEvent:eiEventSignals:eiEventSignal:eiTarget
 - eiEvent:eiEventSignals:eiBaseline
 - eiEvent:eiEventSignals:eiEventSignal:itemBase Substitution Group
- Root Payload Element schemaVersion attribute must not be sent as part of any payload
- The currentValue eiEvent element must be included in the oadrDistributeEvent payload

The OpenADR 2.0b profile adds the functionality excluded above for the 2.0a profile.

In addition, the 2.0b profile optionally supports XML signatures, which requires a different XML root element for each payload (refer to Section 10.6). With the changes to the B schema to support XML signatures, the changes required for A and B devices to interoperate will be much more extensive and may not be practical using a common code base. The B devices would need to strip the root oadrPayload, Signature, and oadrSignedObject elements from every payload.

7.2 OpenADR 2.0b Feature Set Profile

The OpenADR 2.0b Feature Set was developed for advanced demand response systems and markets (e.g., wholesale and retail DR markets). It includes an extended EiEvent Service as well as several additional services usable in Demand Response programs and for ancillary services.

7.2.1 Supported Services

- a) EiEvent Service
- b) EiReport Service
- c) EiRegisterParty Service
- d) EiOpt Service

7.2.2 Report Only VENs

Some devices, such as meters, do not have the ability to shed load. However, these types of devices can provide valuable reporting information to the VTN. The B profile has a sub-profile for VENs called Report Only, which includes support for the following services:

- a) EiReport Service

- b) EiRegisterParty Service

7.2.3 Transport Mechanism

Supported transport mechanisms are as follows. The mechanisms are described in section 9.

- a) HTTP is mandatory for VTNs; VENs must either support HTTP or XMPP
- b) XMPP is mandatory for VTNs; VENs must either support HTTP or XMPP

7.2.4 Security

Supported security details are outlined in sections 9 and 10. The following security levels apply to OpenADR 2.0b.

- a) Standard Security – mandatory
- b) High Security – optional

8 OpenADR 2.0b Services and Data Models Extensions

8.1 OpenADR 2.0b EiEvent Service

Events are generated by the VTN and sent to the VEN using the `oadrDistributeEvent` payload containing one or more events described by the `oadrEvent` element. Some events require a response and others do not as indicated by the `oadrResponseRequired` element in the event description. If a response is required, the VEN acknowledges its opt-in or out-out disposition by responding with an `oadrCreatedEvent` payload containing `eventResponse` elements matching each `oadrEvent`. If no response is required, the VEN must not reply with `oadrCreatedEvents` (or `oadrCreateOpt`) payloads for this event.

Either a PUSH or PULL interaction pattern may be used. For push, the VTN will deliver events to the VEN using an `oadrDistributeEvent` payload. In PULL mode, the `oadrDistributeEvent` will be sent from the VTN to the VEN as response to an `oadrPoll` (refer to section 8.6). In addition to periodically sending `oadrPoll`, a VEN may also send one-time `oadrRequestEvent` payloads to the VTN in order to acquire events from the VTN. Note that 2.0A VENs will use `oadrRequestEvent` for periodic pulling of events instead of `oadrPoll`. If an application level response is required, the VEN asynchronously sends an `oadrCreatedEvent` back to the VTN in a second message. These sequences are illustrated in Figure 4. Note that in simpleHTTP PUSH mode, the VEN's response to `oadrDistributeEvent` is a transport level acknowledgement if required (in the case of HTTP a 200 response code, in XMPP an empty *iq* stanza).

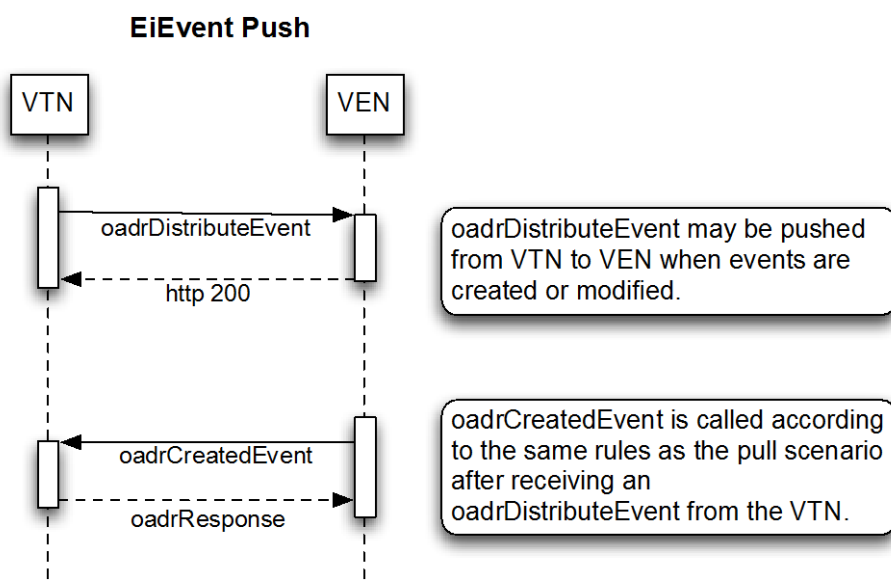


Figure 4 – EiEvent PUSH Pattern

For the PULL case the VEN requests events by sending an `oadrPoll` to the VTN (see section 8.6). The VTN responds with an `oadrDistributeEvent`. From this point the VEN response is exactly the same as in the PUSH interaction. These sequences are illustrated in Figure 5.

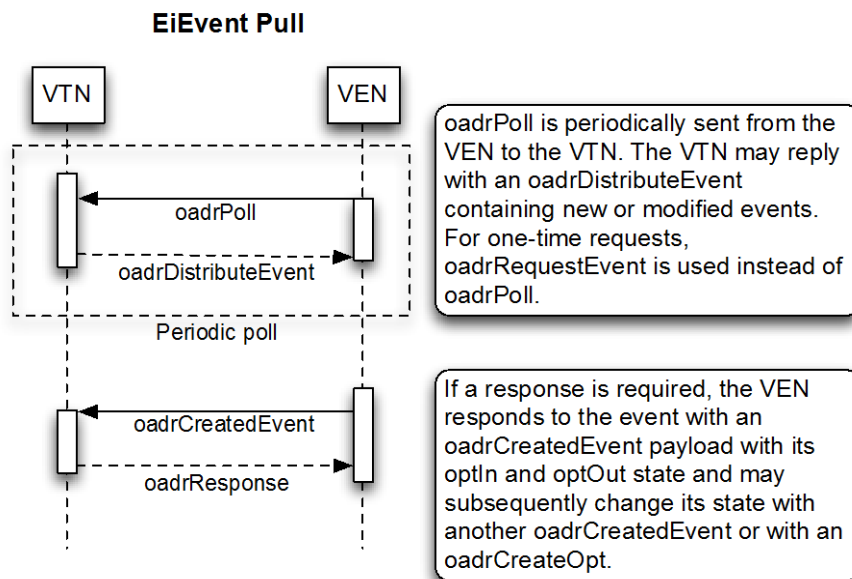


Figure 5 – EiEvent PULL Pattern

The details of how these interactions are performed within the context of a specific transport mechanism are covered in Section 9.

When an event requires a response, an initial oadrCreatedEvent is always sent from the VEN to the VTN. If a given program allows a VEN to later change its opt-state during an event it may do so by issuing subsequent oadrCreatedEvent (or oadrCreateOpt) message containing the new state for a given event. Detailed descriptions of VTN and VEN event processing are given in the following paragraphs.

Note that for both PUSH and PULL operations, an oadrDistributeEvent payload will always contain all events applicable to the VEN it is communicating with.

Events are conveyed in the oadrDistributeEvent payload using one or more oadrEvent elements. The oadrDistributeEvent payload has the following components:

- A requestID to uniquely identify this request and any contained events.
- A vtnID identifying the VTN sending the request.
- Zero or more oadrEvent elements.

The requestID uniquely identifies the request and any contained events. Its value is set by the VTN using whatever scheme they desire, including using the same value in every request if its use is not needed by the VTN. The receiving VEN must use this requestID in the oadrCreatedEvent event responses.

oadrEvent Description

oadrEvent elements describe individual events, signal values, and time periods that apply to signals. Each oadrEvent has an eiEvent element containing detailed event information and an oadrResponseRequired element that controls whether a VEN must respond with an oadrCreatedEvent.

The responseRequired field indicates how the VEN must respond to contained events. A value of “always” indicates that the VEN must reply to each event whether or not the event state (eventID, modificationNumber) has changed. A value of “never” implies a “broadcast” event and the VEN must not send any responses in this case.

The eiEvent eventDescriptor has the following fields:

- **eventID** – a unique ID for this event. The ID is unique within the context of a VTN.
- **modificationNumber** – A sequence that starts at zero and is incremented by 1 each time the VTN modifies the event.
- **modificationDateTime** – The time the event has been modified (only in 2.0b payloads)
- **modificationReason** – The reason why the event has been modified (only in 2.0b payloads)
- **priority** – An indication of the event priority with 0 being no priority
- **marketContext** – Identifies a particular program or application defined grouping that pertains to an event.
- **createdDateTime** – The time the payload containing the event was created. Note that payloads are not necessarily created each time they are transmitted as they may be buffered and only recreated if the event is modified. Any change of the payload of the event requires an update of createdDateTime.
- **eventStatus** – The status of the event, indicating if the event is “near”, “far”, “active” or “cancelled”.
- **testEvent** – If not false, indicates this is a test event.
- **vtnComment** – Arbitrary comment provided by the VTN.

A single **eiActivePeriod** defines the start time and duration of the event. The start time is defined by an ISO 8601 time descriptor and an ISO 8601 duration string specifies the duration. Note that RFC 5545 (iCalendar) does not support the representation of years and months in the duration string that is otherwise similar to the one specified in ISO 8601. OpenADR 2.0 implementations are required to support the ISO 8601 representation (including support for fractional seconds).

The event signals that get applied over the entire active period are defined in an **eiEventSignals** element. This element contains one or more **eiEventSignal** elements (for OpenADR 2.0a at most one), each with a sequence of durations, the sum of which must equal the full duration of the active period. Each signal element contains a **signalType** such as level or price. The **signalPayload** contains the value of the signal according to table Table 1 (for the B profile), and relative values of “normal”, “moderate”, “high”, or “special” (for the A profile) for each duration.

Figure 6 depicts the different time periods of an event. Elv1.0 specifies the following elements as:

Ramp Up Period:

Period at the beginning of the Active Interval expressed as a Duration, during which a VEN moves from its former state to its requested state. If negative, then the Ramp Up occurs within the bounds of the Active Interval, i.e., it starts at the same moment as the Active Interval. If there is no Ramp Up Period, then all other rules are processed as if there were a Ramp Up Period of zero length.

Recovery Period:

Period at the end of the Active Interval expressed as a Duration during which the effect of the response may be reversed while the system returns to its base state. For example, a system that reduces energy use during an Event by raising the air temperature may use additional energy during the recovery period while cooling the air to the normal setting. If negative, then the Recovery Period occurs within the bounds of the Active Interval, i.e., it ends at the same moment as does the Active Interval.

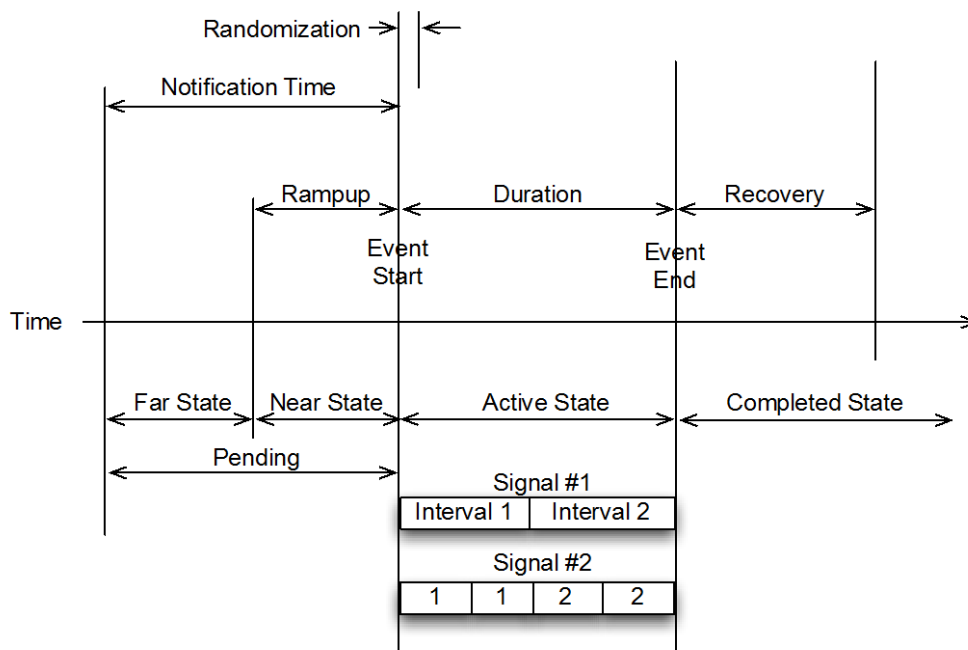


Figure 6 – Time intervals of an event

The EiTarget may be used to provide information to explicitly specify the entities that apply to events. The B profile also supports signal specific EiTargets. An EiTarget may contain one or more targets such as venIDs, groupIDs, resourceIDs, or partyIDs (many others available in the B profile). These may be used, for example, when a VEN is acting as an aggregator and has multiple resources behind it. Exactly how these IDs are handled is beyond the scope of this specification and must be dealt with by provisioning at the VEN and VTN. If the EiTarget element is empty, the VEN must assume that it is the primary and only resource targeted by the events.

oadrCreatedEvent Description

When one or more received events require a response, the VEN creates and populates an oadrCreatedEvent element and posts it to the VTN. The eiResponse element contains an application level responseCode and responseDescription and a requestID. An eiResponses element contains one or more eventResponse elements corresponding to each event. These are matched to specific events using the qualifiedEventID, which contains an eventID and modificationNumber. The optType may have a value of “optIn” or “optOut” to indicate the VENs disposition for a given event.

An initial oadrCreatedEvent response must be sent for each event requiring a response. Subsequent oadrCreatedEvent messages may also be sent to change the opt-state of a VEN when this is allowed for a given marketContext.

The grouping of events in an oadrCreatedEvent is completely up to the VEN and does not necessarily correspond to the grouping of events in an oadrDistributeEvent. The VEN is free to send its opt-statuses by transmitting one oadrCreatedEvent payload per event or group multiple event responses into a single oadrCreatedEvent payload.

8.1.1 Data Model

The OpenADR Alliance has its own schema that defines Alliance specific extensions and references elements from Energy Interoperation and related schemas.

For each schema referenced by Energy Interoperation and used by OpenADR profiles, the OpenADR Alliance created a separate subset schema XSD file. These schema subsets will maintain the same schema hierarchical element relationships, ordering of elements

mandatory cardinality, type definitions, and restrictions. However, the subset schemas may not use the same structural elements, such as abstract types and substitution groups, as the Energy Interoperation and related schemas. The Alliance will associate namespace prefixes with the same URIs used by the Energy Interoperation and related schemas.

URIs referenced in the OpenADR schema can be mapped to either the Alliance subset schema files or the Energy Interoperation 1.0 standard and related schema files. Individual components of the OpenADR payloads will successfully validate against the OpenADR schema and the OASIS Energy Interoperation 1.0

8.1.2 UML Models

Figure 7 shows the OpenADR 2.0 EiEvent Service using the OASIS EI 1.0 standard. The green highlights are the elements that apply to OpenADR 2.0a with an indication if these elements are mandatory (M) or optional (O), and the additional element that are needed for OpenADR 2.0b in red.

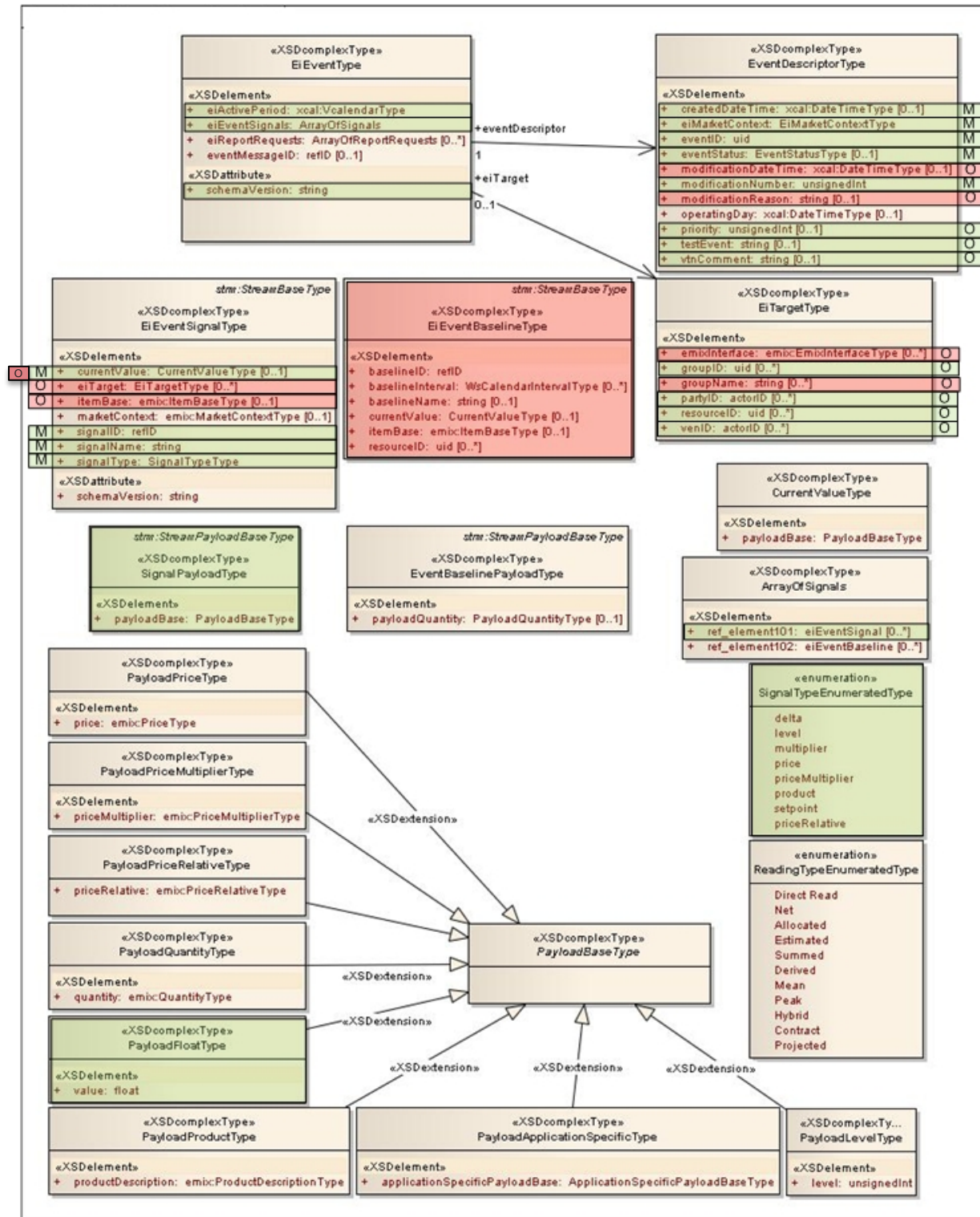


Figure 7 – EiEvent UML (green – A profile; red – B profile additions)

8.2 Differences between OpenADR2.0a and 2.0b Event Mechanism

The OpenADR 2.0b profile supports additional functionalities not supported by A (see also section 7.1). The B profile adds support for the following capabilities, mostly relating to event signals:

- Event signal type is not limited to SIMPLE. The full set of signals described in section 8.2.2 is supported.
- Events may contain multiple event signals.
- Event signals may contain EiTarget elements, distinct from the event-level EiTarget. However, the target types must be constrained to endDeviceAsset with the enumerated values for device type shown in the conformance rules.
- Event signals may contain baseline elements. The rules for baseline overall duration and intervals are the same as those between the event active period and event signal intervals.
- The event description may contain modificationDateTime and modificationReason elements.
- The EiOpt service can be used to send schedules from the VEN to the VTN, in addition to oadrCreatedEvent from 2.0a.
- The oadrPoll mechanism is used in a simple HTTP PULL exchange to model the behavior of a PUSH implementation. For more information refer to section 8.6.

8.2.1 Event Targets and Resources

While the A Profile supported event targets (EiTarget), the B Profile includes several more target types, and also allows device class targeting to be applied at the signal level. A VEN is a communication endpoint that represents one or more logical resources (individual shedable loads, endpoint equipment, meters, etc.). Event targets select which specific VEN resources the event applies to. If an event target is not specified, the VEN should assume that it applies to all of its resources. How resources are assigned properties (location, pnode associations, resourceIDs, groupIDs, etc.) is outside the scope of the specification and is up to deployment configurations in the VTN and VEN. However, if a VEN receives an event target that it is not configured for, it should reject the message with the appropriate error code described in section 8.7.

8.2.2 OpenADR 2.0b Signal Definitions

Profile B allows for a more diverse set of signals in the event messages. The eiEventSignal:signalName, eiEventSignal:signalType, and eiEventSignal:itemBase attributes are used to describe the signal. Table 1 lists standard pre-defined signals that may be used. The purpose of the pre-defined signals is to establish a common set of signals and their attributes for the purposes of interoperability. For compliance purposes it is not a requirement that a VTN or VEN support all the signals listed in the table below.

Furthermore specific deployments are free to define their own custom signals beyond what are defined in the table below, but there are no requirements for compliance purposes that any VTN's or VEN's support such signals.

The value of the eiEventSignal:signalID attribute is deployment specific and primarily used to differentiate signals in cases where there may be multiple signals of the same type.

Table 1 – Signals

Signal Category	Name (signalName)	Type (signalType)	units (itemBase)	Allowed Values	Description
Simple levels	SIMPLE	level	None	0,1,2,3	Simple levels
Price of electricity	ELECTRICITY_PRICE	price	currency/kWh	any	This is the cost of electricity expressed in absolute terms
	ELECTRICITY_PRICE	priceRelative	currency/kWh	any	This is a delta change to the existing price of electricity
	ELECTRICITY_PRICE	priceMultiplier	None	any	This is a multiplier to the existing cost of electricity
Price of energy	ENERGY_PRICE	price	currency/kWh	any	This is the cost of energy expressed in absolute terms
	ENERGY_PRICE	priceRelative	currency/kWh	any	This is a delta change to the existing price of energy
	ENERGY_PRICE	priceMultiplier	None	any	This is a multiplier to the existing cost of energy
Demand charge	DEMAND_CHARGE	price	currency/kW	any	This is the demand charge expressed in absolute terms
	DEMAND_CHARGE	priceRelative	currency/kW	any	This is a delta change to the existing demand charge
	DEMAND_CHARGE	priceMultiplier	None	any	This is a multiplier to the existing demand charge
Customer bid levels	BID_PRICE	price	currency/XX	any	This is the price that was bid by the resource
	BID_LOAD	setpoint	powerXXX	any	This is the amount of load that was bid by a resource into a program
	BID_ENERGY	setpoint	energyXXX	any	This is the amount of energy from a resource that was bid into a program
Used to dispatch storage resources	CHARGE_STATE	setpoint	energyXXX	any	This is used to either charge or discharge a certain amount of energy from a storage resource until its charge state reaches a certain level.
	CHARGE_STATE	delta	energyXXX	any	This is the delta amount of energy that should be contained in a storage resource from where it currently is.
	CHARGE_STATE	multiplier	None	0 < 1	This is the percentage of full charge that the storage resource should be at.
These instructions are used to set the load to values that	LOAD_DISPATCH	setpoint	powerXXX	any	This is used to dispatch loads to a specific amount