

INTERNATIONAL STANDARD



**Systems interface between customer energy management system and the power management system –
Part 10-1: Open automated demand response**

IECNORM.COM : Click to view the full PDF of IEC 62746-10-1:2018



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2018 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 21 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IECNORM.COM : Click to view the full text of IEC 6116-101:2018

INTERNATIONAL STANDARD



**Systems interface between customer energy management system and the
power management system –
Part 10-1: Open automated demand response**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 33.200

ISBN 978-2-8322-6228-3

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

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
1 Scope.....	9
2 Normative references	9
3 Terms, definitions and abbreviated terms	10
3.1 Terms and definitions.....	10
3.2 Abbreviated terms.....	11
4 Overview	12
4.1 General.....	12
4.2 Node and device types.....	13
4.3 IEC 62746-10-1 services.....	14
4.4 Assumptions	15
5 IEC 62746-10-1 features.....	15
5.1 General.....	15
5.2 Supported services	15
5.3 Report Only VENS.....	15
5.4 Transport mechanism	15
5.5 Security	15
6 Services and data model extensions.....	15
6.1 Event service	15
6.1.1 Event interactions.....	15
6.1.2 oadrEvent mechanism	19
6.2 Report service	23
6.2.1 General	23
6.2.2 Core reporting operations	25
6.3 Registration service.....	30
6.3.1 Service operations.....	30
6.3.2 Registration information.....	33
6.4 Opt service.....	34
6.4.1 Service operations.....	34
6.4.2 Detail requirements	35
6.5 Poll service	36
6.6 Application error codes	39
7 Transport protocol	40
7.1 General.....	40
7.2 Simple HTTP	40
7.2.1 General	40
7.2.2 PUSH and PULL implementation	40
7.2.3 Service endpoint URIs	41
7.2.4 HTTP methods.....	41
7.2.5 Failure conditions	41
7.2.6 HTTP response codes	41
7.2.7 Message timeouts.....	42
7.2.8 Message retry/quiesce behaviour	42
7.2.9 PULL timing.....	42
7.2.10 HTTP headers	43

7.3	Transport-specific security	44
7.3.1	General	44
7.3.2	TLS client certificate	44
7.4	XMPP	44
7.4.1	General	44
7.4.2	Exchange model implementation	44
7.4.3	Service endpoints	44
7.4.4	Service execution	45
7.4.5	Implementation of XMPP features	45
7.4.6	Security considerations	48
8	Cyber security	49
8.1	General	49
8.2	Architecture and certificate types	49
8.3	Certificate authorities	50
8.4	Certificate revocation	50
8.5	TLS and cipher suites	50
8.6	System registration process	50
8.6.1	General	50
8.6.2	Certificate fingerprints	50
8.7	Implementing XML signatures for message payloads	51
8.7.1	XML signature	51
8.7.2	Components of XML signatures	51
8.7.3	Creating XML signatures	52
8.7.4	Verifying XML signatures	53
9	Conformance	53
9.1	Conformance statement	53
9.2	Conformance rules	53
9.2.1	EiEvent	53
9.2.2	EiEvent – Additional 2.0b conformance rules	61
9.2.3	EiOpt	63
9.2.4	EiReport	65
9.2.5	EiRegisterParty	71
9.2.6	General conformance rules	73
9.3	Cardinality	76
Annex A (normative)	Detailed report description	77
Annex B (normative)	Profile extensions	78
B.1	Overview	78
B.2	Report extension	78
B.3	Event extension	78
B.4	Other extensions	78
Annex C (normative)	oadrPoll scenarios	79
C.1	Overview	79
C.2	Scenarios	79
Annex D (normative)	Definition of VEN, VTN, resource, and party	81
Annex E (normative)	IEC 62746-10-1 Schema	82
Bibliography		206

Figure 2 – EiEvent PUSH pattern	16
Figure 3 – EiEvent PULL pattern	17
Figure 4 – Time intervals of an event	19
Figure 5 – Report type	24
Figure 6 – Interaction diagram: Register reporting capabilities	26
Figure 7 – Interaction diagram: Request reports	27
Figure 8 – Interaction diagram: Send reports	29
Figure 9 – Interaction diagram: Cancel reports	30
Figure 10 – Interaction diagram: Query registration	31
Figure 11 – Interaction diagram: Create registration	31
Figure 12 – Interaction diagram: Request reregistration	32
Figure 13 – Interaction diagram: Cancel registration	32
Figure 14 – Interaction diagram: Create opt	35
Figure 15 – Interaction diagram: Cancel opt	35
Figure 16 – Interaction diagram: oadrPoll (nothing in queue)	37
Figure 17 – Interaction diagram: oadrPoll (oadrDistributeEvent reply)	38
Figure 18 – Interaction diagram: oadrPoll (oadrCreateReport reply)	38
Figure 19 – Interaction diagram: oadrPoll (request reregistration reply)	39
Figure 20 – XML signature example	52
Table 1 – IEC 62746-10-1 services support	13
Table 2 – Signals	21
Table 3 – EiRegisterParty payloads	30
Table 4 – VEN information in oadrCreatePartyRegistration payload	33
Table 5 – VTN information oadrCreatedPartyRegistration payload	34
Table 6 – EiOpt payloads	34
Table 7 – Conformance rules	54
Table 8 – Additional conformance rules	61
Table 9 – EiOpt conformance rules	63
Table 10 – EiReport conformance rules	65
Table 11 – EiRegisterParty conformance rules	71
Table 12 – General conformance rules	73
Table 13 – Cardinalities	76

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**SYSTEMS INTERFACE BETWEEN CUSTOMER ENERGY
MANAGEMENT SYSTEM AND THE POWER MANAGEMENT SYSTEM –****Part 10-1: Open automated demand response**

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.

International Standard IEC 62746-10-1 has been prepared by IEC project committee 118: Smart grid user interface.

The text of this International Standard is based on the following documents:

CDV	Report on voting
118/91/CDV	118/96B/RVC

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62746 series, published under the general title *Systems interface between customer energy management system and the power management system*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "<http://webstore.iec.ch>" in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

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

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

IECNORM.COM : Click to view the full PDF of IEC 62746-10-1:2018

INTRODUCTION

Development of the demand response (DR) market has resulted in a transition from manual DR to OpenADR (this document) in automated DR programs. DR is defined as an 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. This document was developed to support common auto-DR programs and energy policy objectives to move toward dynamic markets to improve the economics and reliability of the electricity grid. The recent developments have expanded the use of this document to meet diverse market needs, such as ancillary services, 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 document's communication has the following defining features:

- Continuous, secure, and reliable – Provides continuous, secure, and reliable two-way communications where the endpoints at the end-use site receive and acknowledge the receipt of DR signals from the energy service providers.
- Translation – Translates DR event information to 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.

This document is a communications data model, along with transport and security mechanisms, which facilitate information exchange between two end-points: the electricity service provider or DR program operator, and a customer-side resource. It is not a protocol that specifies "bit-structures" as some communications protocols do, but instead relies upon existing open standards such as Extensible Markup 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 this document.

IEC 62746-10-1 is designed to facilitate automation of DR actions at the customer location, whether it involves electric load decreases, load increases, or load shifting for various demand response markets. Many emergency or reliability DR events occur at specific times when the electricity grid is strained. The communications are designed to coordinate such signals with facility control systems (commercial, industrial, and residential). This document 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 within a facility. Several reports present the history of this document and the involved research. This document covers the signalling data models for price and reliability signals to both wholesale and retail markets, and can act as a complementary standard to a CIM-based grid control system.

This document provides the following benefits:

- Open specification – Provides a standardized DR communications and signalling 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 obsolescence in technology.
- Ease of integration – Facilitates integration of common energy management and control systems (EMCS), centralised lighting, and other end-use devices that can receive internet signals (such as XML).
- Supports a wide range of information complexity – Can express the information in the DR signals in a variety of ways to allow 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.

This document's purpose is to manage the growing demand for demand-side flexibility – the ability to modify the load profile of the consumers over time. This low-cost communications infrastructure is used to improve the reliability, repeatability, robustness, and cost-effectiveness of DR.

This technology has been field tested for over a decade and deployed in a number of DR programs worldwide. While the scope of this document focuses on signals for DR events and prices, significant supporting research work has been done to study DR strategies and techniques to automate DR within facilities. This document 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.

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

Part 10-1: Open automated demand response

1 Scope

This part of IEC 62746-10, OpenADR 2.0 (this document), specifies a minimal data model and services for demand response (DR), pricing, and distributed energy resource (DER) communications. This document can be leveraged to manage customer energy resources, including load, generation, and storage, via signals provided by grid and/or market operators. These resources can be identified and managed as individual resources with specific capabilities, or as virtual resources with an aggregated set of capabilities.

This document specifies how to implement a two-way signaling system to facilitate information exchange between electricity service providers, aggregators, and end users. The DR signalling system is described in terms of servers (virtual top nodes or VTNs), which publish information to automated clients (virtual end nodes, or VENs), which in turn subscribe to the information.

This document provides application-level service communication that can be used to incentivise a response from a customer-owned DER. Price and DR signals over the internet allow indirect control of customer-owned devices that otherwise would not be available.

This document's services are independent of transport mechanisms. For the purposes of interoperability, this document provides basic transport mechanisms and their relevant interaction patterns to address different stakeholder needs. In addition, this document specifies cyber security mechanisms required for data confidentiality, integrity, authentication and message-level security, in order to provide non-repudiation and mitigation of cyber security risks.

The services make no assumption of specific DR electric load control strategies that can be used within a DR resource or of any market-specific contractual or business agreements between electricity service providers and their customers.

This document provides a clear set of mandatory and optional attributes within each of the services to meet broader interoperability, testing and certification requirements. All necessary XML schema are included in Annex E.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

INTERNET ENGINEERING TASK FORCE (IETF). RFC 2616: *Hypertext Transfer Protocol – HTTP/1.1* [online]. Edited by R. Fielding et al. June 1999 [viewed 2018-08-02]. available at: <http://www.ietf.org/rfc/rfc2616.txt>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 3986: *Uniform Resource Identifier (URI): Generic Syntax* [online]. Edited by T. Berners-Lee et al. January 2005 [viewed 2018-08-02]. available at:
<http://www.ietf.org/rfc/rfc3986.txt>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2 [online]. Edited by T. Dierks et al. August 2008 [viewed 2018-08-02]. available at
<https://tools.ietf.org/html/rfc5246>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 6120: *Extensible Messaging and Presence Protocol (XMPP): Core* [online]. Edited by P. Saint-Andre. March 2011 [viewed 2018-08-02]. available at:
<http://www.ietf.org/rfc/rfc6120.txt>

INTERNET ENGINEERING TASK FORCE (IETF). RFC 6121: *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence* [online]. Edited by P. Saint-Andre. March 2011 [viewed 2018-08-02]. available at:
<http://www.ietf.org/rfc/rfc6121.txt>

XMPP Standards Foundation. *XEP-0030: Service Discovery* [online]. Edited by J. Hildebrand et al. October 2017 [viewed 2018-08-02]. available at:
<http://xmpp.org/extensions/xep-0030.html>

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1.1

OpenADR 2.0

specification that provides the basis for this document

3.1.2

PUSH and PULL operations

mode of operation by which this document can be used in either PULL mode (VEN pulling information from VTN) or in a PUSH mode (VTN pushing information to the VEN)

Note 1 to entry: The XMPP transport uses a PUSH model, although VENs can still make requests of the VTN, excluding the use of oadrPoll.

3.1.3

simple HTTP

HTTP implementation that uses HTTP POST over TLS to propagate payloads

3.1.4

virtual end node

VEN

technical role assumed by an actor where the actor is a consumer and/or producer of messages that are defined by this document

Note 1 to entry: A virtual end node (VEN) can be associated with zero or more resources. A VEN can receive messages pushed from a VTN or send requests or events to a VTN. A VEN may communicate with multiple VTNs, where each VTN is part of a different communication domain.

[SOURCE: IEC TS 62746-3:2015, 3.1.15, modified – Note 2 to entry has been omitted.]

3.1.5

virtual top node

VTN

technical role assumed by an actor that is assuming responsibility for the coordination of VENs within a communication domain

Note 1 to entry: This is a special case of a VEN, where a virtual top node (VTN) is effectively a parent of many VENs with the responsibility for coordination of those VENs. A VTN is responsible for pushing to or receiving messages from many VENs. A market operator, grid operator or aggregator are examples of actors which will typically implement a VTN interface.

[SOURCE: IEC TS 62746-3:2015, 3.1.17, modified – Note 2 to entry has been omitted.]

3.1.6

resource

demand-side commodity that is associated with a load profile

3.1.7

party

entity that enters into some sort of business relationship or contract

Note 1 to entry: A more detailed definition can be found in Annex D.

3.1.8

RSA

public-key crypto system placed into the public domain by RSA Data Security, Inc.

3.2 Abbreviated terms

CA	certificate authority
DER	distributed energy resources
DR	Demand Response
ECC	elliptic curve cryptography
EI	energy interoperation
HTTP	Hypertext Transfer Protocol
ISO	independent systems operator
JID	jabber identifier
OpenADR	Open Automated Demand Response
PICS	protocol implementation conformance statement
PKI	public key infrastructure
SASL	Simple Authentication and Security Layer
SOAP	Simple Object Access Protocol
TLS	Transport Layer Security
UCAIug	Utilities Communications Architecture International Users Group
VEN	virtual end node
VTN	virtual top node
XML	Extensible Markup Language
XMPP	XML Messaging and Presence Protocol

4 Overview

4.1 General

Clause 4 gives an overview of the message exchanges, the roles, and actors supported within this document. It contains the following elements that are used to develop test and certification frameworks for smart grid and customer system interoperability:

- a) a set of data models that describe information communicated in message payloads;
- b) a set of services for performing various functions and operations for the exchange of the data models;
- c) a set of 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);
- d) a set of security mechanisms for securing each of the transport mechanisms;
- e) a set of XML schemas (see Annex E).

Integration of IEC 62746-10-1 systems within the IEC's standards framework is done with a CIM adapter that may be produced in accordance with the methodology described in IEC 62746-10-3. Message exchanges in this document support services for communicating information about demand response events. Networks of nodes shall be able to query for active or pending events, register themselves, schedule events, and send reports. The nodes shall also be able to refine and update previously sent information. For instance, a node reporting DR events to nodes downstream shall 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 and then 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/VENs may also communicate over other transport mechanisms, such as XML messaging and presence protocol (XMPP).

This document supports end devices with a varying degree of capabilities. However, for interoperability, all protocol capabilities are mandatory.

IEC 62746-10-1 specifies the following services:

- 1) Register: Registration identifies entities in advance of interactions with other parties in various roles such as VEN and VTN.
- 2) Event: 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: reliability events, emergency events, price events, regulation events and possibly other types in the future.
- 3) Report: The report service enables feedback to the server in order to provide periodic or one-time information on the state of a resource.
- 4) Opt: Overrides the EiAvail and addresses short-term changes in availability to create and communicate opt-in and opt-out schedules from the VEN to the VTN.

Table 1 outlines the mandatory and optional implementation of this document.

Table 1 – IEC 62746-10-1 services support

	VTN	VEN	
		Full	(Energy Reporting only)
Services and Functions Support			
EiEvent			
Full Profile	M	M	NA
EiOpt			
Full Profile	M	M	NA
EiReport			
Full Profile	M	M*	M*
EiRegisterParty			
Full Profile	M	M	M
Transport Protocols			
Simple HTTP	M	O-1	O-1
XMRP	M	O-1	O-1
Security Levels			
Standard	M	M	M
High	O	O	O
M Mandatory O Optional O-1 Optional, but at least one option shall be supported			
NA Not available * Optional features available			

4.2 Node and device types

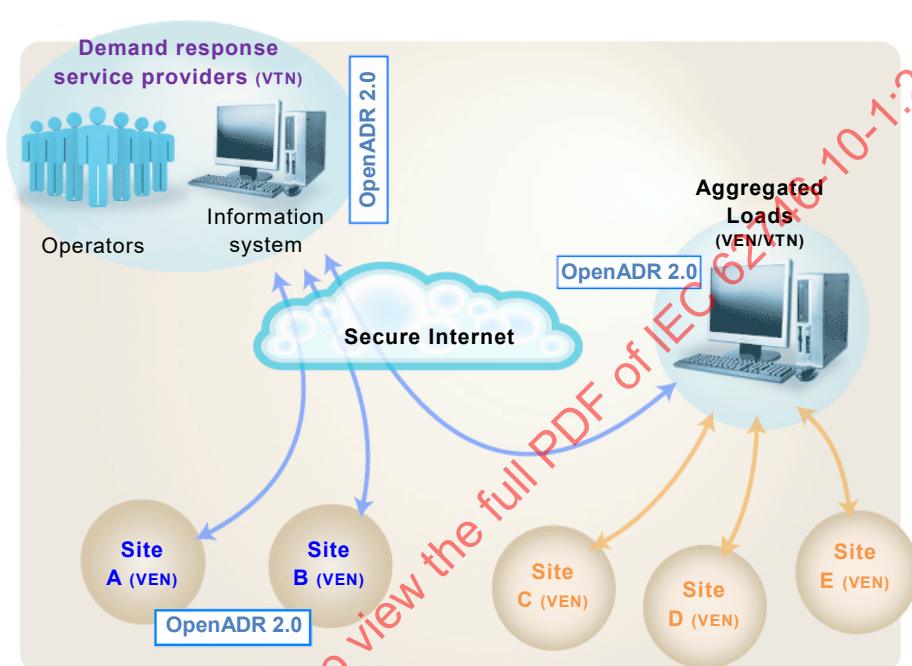
For any interaction between actors using this document to communicate, one actor is designated the virtual top node and the remainder are the virtual end nodes. All communications are between a VTN and one or more VENs. There is no peer-to-peer communication in this document, i.e. VTNs do not communicate directly with other VTNs, and, likewise, VENs do not communicate directly with other VENs.

The VTN generally acts as the server, providing information to the VEN, which responds to the information. For instance, a VTN would be the entity to announce a DR event; then, VENs hear about DR events and respond. The response can be to reduce power to some devices. The response could also be to propagate the signal further downstream to other VENs. In this case, the VEN would become the VTN for the new interaction (e.g. the aggregator in Figure 1).

For the purpose of device development, the manufacturer should always test the interface between a VTN and a VEN, where either node can be the device under test. Intelligence built into the systems not related to the message exchange is not part of this document.

The authoritative requirements for implementation of VENs and VTNs are defined in the schema and the conformance rules. Narrative descriptions in later clauses separate from the conformance rules provide context and implementation examples, but do not contain the full breadth of implementation requirements. Therefore, it is strongly recommended that implementers thoroughly understand all the conformance rules prior to coding.

Although within any interaction, one actor is designated the VTN and the remainder are VENs (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 VENs for some interactions and VTNs for others.



IEC

Figure 1 – Possible relationships of VTN and VEN

As illustrated in Figure 1, 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 this document does not define how the nodes react to the information. In nodes that 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 can 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.

4.3 IEC 62746-10-1 services

This document uses four defined services, each one profiled and tailored to meet the relevant market needs. These services are EiRegisterParty, EiEvent, EiReport, and EiOpt. For further information on these services, consult Clause 6.

4.4 Assumptions

Operating assumptions regarding correct functional behaviour between VENs and VTNs are as follows:

- Both the VTN and the VEN shall have awareness of the current time. Time synchronization requirements are determined by the specific implementation and demand response program.

5 IEC 62746-10-1 features

5.1 General

This document's 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.

5.2 Supported services

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

5.3 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. This document has a sub-profile for VENs called Report Only, which includes support for the following services:

- a) EiReport service;
- b) EiRegisterParty Service.

5.4 Transport mechanism

Supported transport mechanisms are as follows. The mechanisms are described in Clause 7.

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

5.5 Security

Supported security details are outlined in Clauses 7 and 8. The following security levels apply to this document:

- a) standard security – mandatory;
- b) high security – optional.

6 Services and data model extensions

6.1 Event service

6.1.1 Event interactions

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 opt-out disposition by responding with an oadrCreatedEvent payload containing eventResponse elements

matching each `oadrEvent`. If no response is required, the VEN shall not reply with `oadrCreatedEvents` (or `oadrCreateOpt`) payloads for this event.

Either a PUSH or PULL interaction pattern may be used. For push, the VTN shall deliver events to the VEN using an `oadrDistributeEvent` payload. In PULL mode, the `oadrDistributeEvent` shall be sent from the VTN to the VEN as response to an `oadrPoll` (refer to 6.5). 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. 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 2. 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).

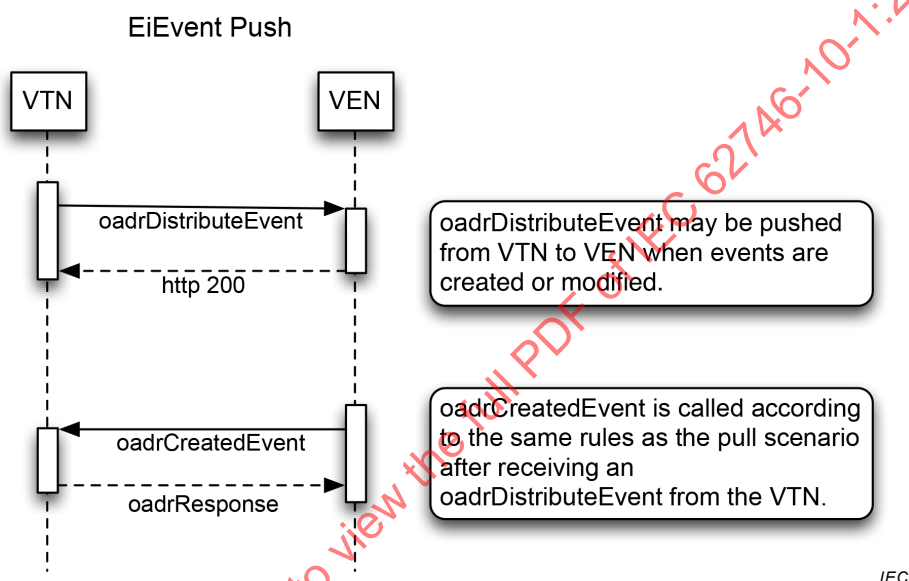


Figure 2 – EiEvent PUSH pattern

For the PULL case, the VEN requests events by sending an `oadrPoll` to the VTN (see 6.5). 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 3.

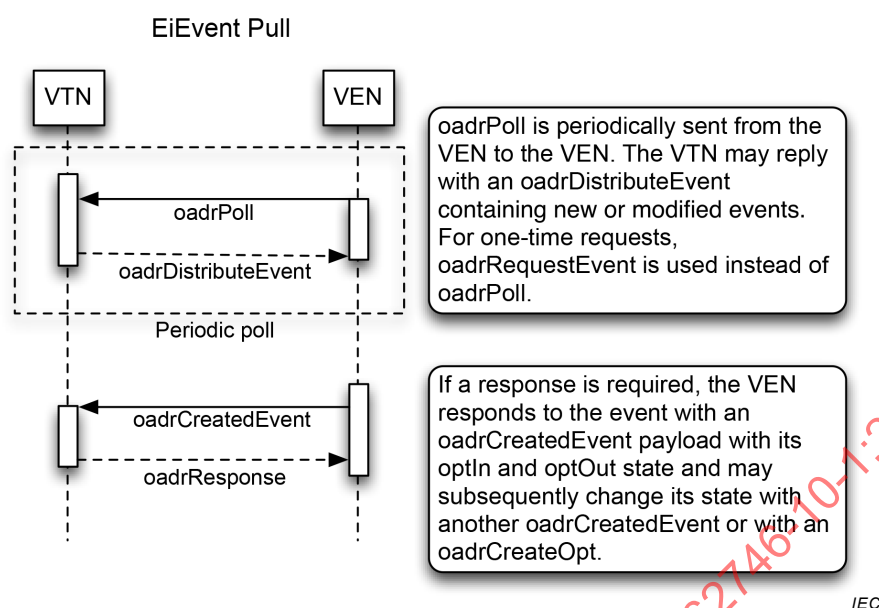


Figure 3 – EiEvent PULL pattern

The details of how these interactions are performed within the context of a specific transport mechanism are covered in Clause 7.

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 a 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 event;
- zero or more oadrEvent elements.

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

a) 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 shall respond with an oadrCreatedEvent.

The responseRequired field indicates how the VEN shall respond to contained events. A value of "always" indicates that the VEN shall 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 shall 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 VEN.
- 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). Refer to 5.4 for time synchronization requirements.
- 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 can be buffered and only recreated if the event is modified. Any change of the payload of the event requires an update of createdDateTime. Refer to 5.4 for time synchronization requirements.
- 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. Implementations are required to support the ISO 8601 representation (including support for fractional seconds).

Event start times can be randomized using the tolerance object in the eiActivePeriod. The sub-element startafter defines a randomization time window used by the VEN to select a random value that is added to the start time of the event. If the start time of a one-hour event is 3:00 and the randomization window is 5 min, if the VEN selected 3 min as the random value then the event would start at 3:03 and would end at 4:03.

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, each with a sequence of durations, the sum of which shall 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 2.

Figure 4 depicts the different time periods of an event. Two items on this figure not previously discussed are defined below.

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.

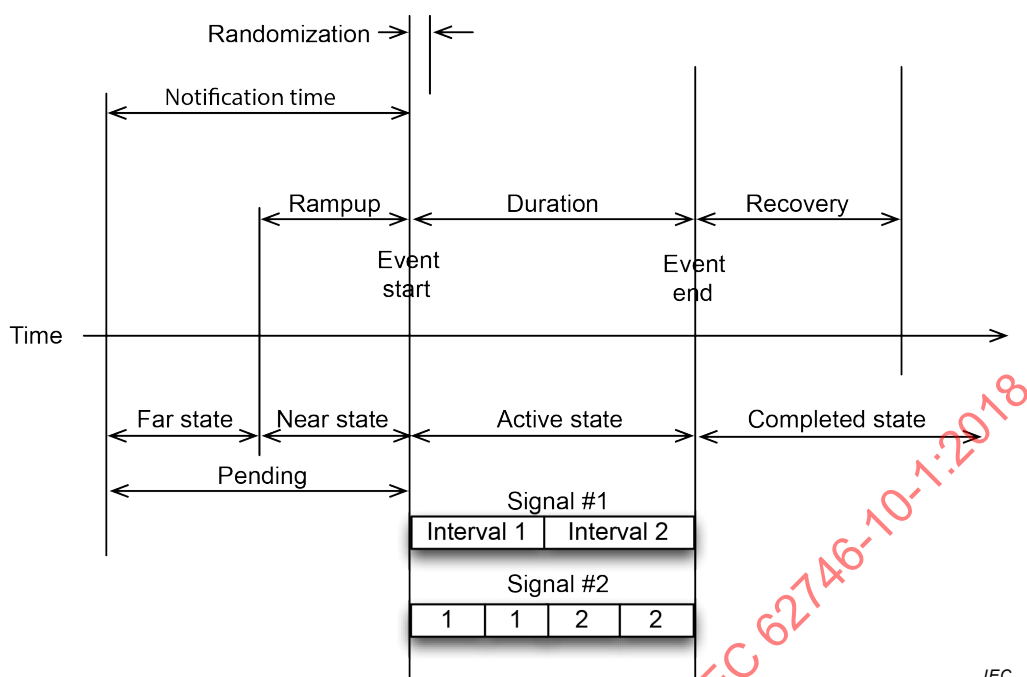


Figure 4 – Time intervals of an event

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 venID and one or more targets such as groupIDs, resourceIDs, or partyIDs (many others available). 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 document and shall be dealt with by provisioning at the VEN and VTN. If the EiTarget element is empty, the VEN shall assume that it is the primary and only resource targeted by the events.

b) 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 shall 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.

6.1.2 oadrEvent mechanism

6.1.2.1 General

- Event signal type can be SIMPLE or specific. The full set of signals described in 6.1.2.3 is supported.
- Events may contain multiple event signals.
- Event signals may contain EiTarget elements, distinct from the event-level EiTarget. However, the target types shall 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`.
- The `oadrPoll` mechanism is used in a simple HTTP PULL exchange to model the behaviour of a PUSH implementation. For more information, refer to 6.5.

6.1.2.2 Event targets and resources

This document includes several 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 sheddable 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 this document 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 6.6.

6.1.2.3 Event signal definitions

IEC 62746-10-1 allows for a diverse set of signals in the event messages. The `eiEventSignal:signalName`, `eiEventSignal:signalType`, and `eiEventSignal:itemBase` attributes are used to describe the signal. Table 2 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 a VEN support all the signals listed in Table 2.

Furthermore, specific deployments are free to define their own custom signals beyond what is defined in Table 2, but there are no requirements for compliance purposes that any VTNs or VENs support such signals.

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

Table 2 – 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 ^{b)}	any	This is the price that was bid by the resource
	BID_LOAD	setpoint	powerXXX ^{a)}	any	This is the amount of load reduction that was offered by a resource into a program. Positive BID_LOAD refers to curtailment, negative BID_LOAD refers to increase
	BID_ENERGY	setpoint	energyXXX ^{a)}	any	This is the amount of energy reduction that was offered by a resource into a program. Positive BID_ENERGY refers to curtailment, negative BID_ENERGY refers to increase

Signal category	Name (signalName)	Type (signalType)	Units (itemBase)	Allowed values	Description
Used to dispatch storage resources	CHARGE_STATE	setpoint	energyXXX ^{a)}	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 ^{a)}	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	0,0 < 1,0	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 can be expressed in terms of the desired load	LOAD_DISPATCH	setpoint	powerXXX ^{a)}	any	This is used to dispatch loads to a specific amount
	LOAD_DISPATCH	delta	powerXXX ^{a)}	any	This is used to dispatch loads to some offset from an agreed upon baseline. Note that the baseline may be the current power consumption
	LOAD_DISPATCH	multiplier	None	any	This is used to dispatch loads as some multiple of power against some agreed upon baseline. Note that the baseline may be the current power consumption
	LOAD_DISPATCH	level	powerXXX	integer value from -10 to +10	This is used to specify the load in terms of discrete levels
These instructions are used to set the load control to values that are relative to the load controller and its output capacity. This does not require the VTN or the VEN knowing precisely what the load consumption level is, but is expressed in ways that the VTN can know that the signal values will either increase or decrease the load consumption	LOAD_CONTROL	x-loadControlCapacity	None	0 % to 100 % (0,0 to 1,0)	This is an instruction for the load controller to operate at a level that is some percentage of its maximum load consumption capacity. This can be mapped to specific load controllers to do things like duty cycling. Note that 1,0 refers to 100 % consumption. In the case of simple ON/OFF type devices, then 0 = OFF and 1 = ON

Signal category	Name (signalName)	Type (signalType)	Units (itemBase)	Allowed values	Description
regardless of the specific type of device that is performing the load control. These instructions can be used for some aspects of direct load control by mapping these general instructions to specific load control commands in the VEN without the VTN needing to know precisely what device can be consuming the signal.	LOAD_CONTROL	x-loadControlLevel Offset	None	integer value, positive or negative	Discrete integer levels that are relative to normal operations where 0 is normal operations. There is no requirement to link the setpoints to specific load consumption values, but the intention is that the higher the set point the less load is consumed. Note that these are controller set points that can be mapped at the VEN side to something as simple as thermostat temperature set point offsets
	LOAD_CONTROL	x-loadControlSetpoint	None	any value	Load controller set points. There is no requirement to link the set points to specific load consumption values. Note that these are generic controller set points and can be mapped at the VEN side to something as simple as specific thermostat temperature set points.
	LOAD_CONTROL	x-loadControlPercent Offset	None	any percentage, both positive and negative	Percentage change from normal operations. The percentage does not refer to specific load consumptions values, but to load control operation levels. The lower the percentage, the less load is consumed.
a) The XXX in the table represents Real, Apparent, and Reactive versions of power or energy. b) The XX in the table represents the currency unit, such as USD.					

6.2 Report service

6.2.1 General

6.2.1.1 Overview

Subclause 6.2 describes the reporting service for IEC 62746-10-1. For detailed report content, see Annex A.

6.2.1.2 Report types

This document supports several well-known report types. Each report type is intended to represent a certain set of reporting functionality that is supported by either a VEN or a VTN that claims to support that type. Figure 5 gives the taxonomy of the different report types.

For compliance purposes, VTNs and VENs are not required to support all the different types of reports (see conformance rule 510 in Clause 9). Furthermore, for specific deployments, there can be reports used that do not conform to any of the standard report profiles, but these are not relevant for compliance purposes.

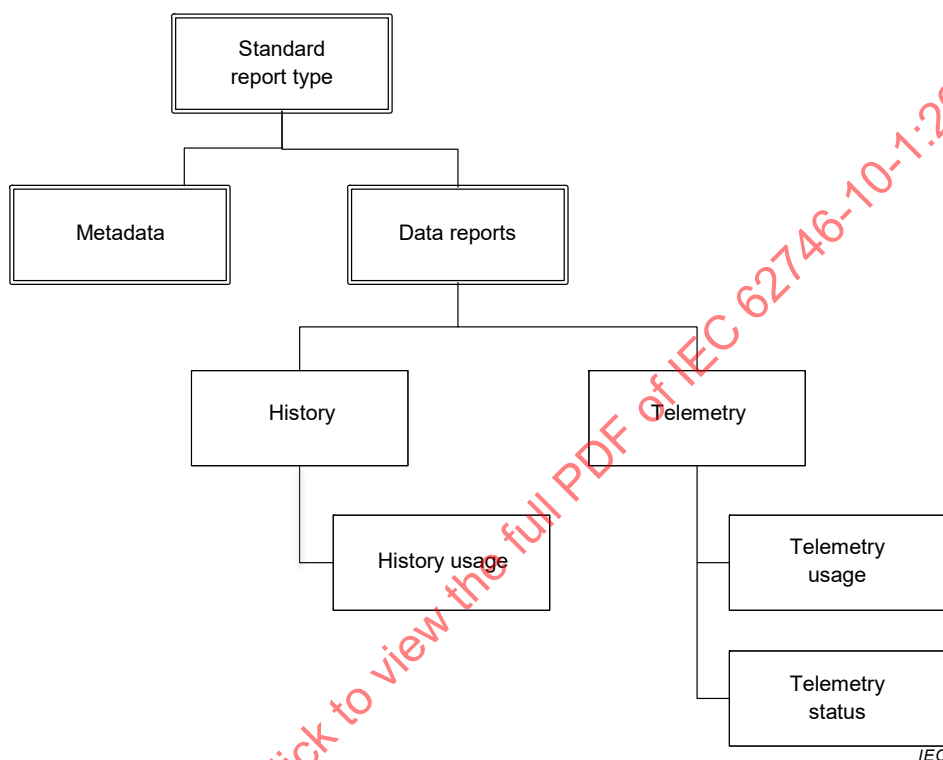


Figure 5 – Report type

Each of the report types is described in more detail below.

METADATA – This report type is used to specify reporting capabilities. It is exchanged as part of the report registration process that is described in 6.2.2.2. The METADATA report can contain a specification for one or more type of reports, each report having its own set of report descriptors and specifications. Each report in the METADATA report has a reportSpecifierID that is used in subsequent interactions to refer to that report specification. Each report specification in the METADATA report will use the reportName attribute to indicate which of the well-known report profiles it is referring to. All VTN and VEN implementations that conform to this document shall support the METADATA report and, in the case where a VTN or VEN does not support any reporting capabilities, for compliance purposes it shall still support sending the METADATA report message in order to portray the fact that it does not support any reporting capabilities.

DATA REPORTS (non-METADATA reports) – These reports are used to report actual data that can be measured or calculated. The core element of a DATA REPORT is the so-called "data point". A data point represents a certain quantity that can be measured or calculated as part of a report. Each data point has attributes such as units, etc. A DATA REPORT may contain one or more data points. The METADATA report will contain a report with a set of data points that can appear in that report and when an actual report is requested the set of data points that should appear in the report are specified by the requesting party. Each data point is represented in the schema with the rID attribute. For example, assume a VEN can measure both energy and power as two separate data points. The VEN has the choice of either specifying that it can generate two separate reports, each with a single data point or it can specify that it can generate a single DATA REPORT with two data points. The VTN would then make the appropriate report request to get the data.

There are several sub-categories of DATA REPORTS as described below.

- **HISTORY DATA REPORTS** – This is a type of DATA REPORT in which the history of the data point values is logged and can be subsequently requested. This includes the following specific types:
 - **HISTORY_USAGE** – These are logs of usage data that are typically logged by VENs and can be queried by the VTN.
- **TELEMETRY DATA REPORTS** – The term telemetry in the context of this standard refers to data that is reported periodically in real time and includes the following specific report types:
 - **TELEMETRY_USAGE** – This is usage data that is periodically reported from the VEN to the VTN in real time.
 - **TELEMETRY_STATUS** – This is the status of a resource, which may be periodically reported from the VEN to the VTN.

6.2.2 Core reporting operations

6.2.2.1 General

The reporting service supports the exchange of reports between the VEN and VTN, and vice versa. All report interactions between the VEN and the VTN are built upon the following core operations:

- registering reporting capabilities;
- requesting reports;
- sending reports;
- cancelling reports.

Figures 6 to 9 show the logical payload exchanges that support each of these operations.

6.2.2.2 Register reporting capabilities

This general use case describes how one party's reporting capabilities are registered with the other party. Report registration is performed after completion of party registration, both from the VTN to the VEN and from the VEN to the VTN. In addition, any party may send report registrations at any time after the initial registration.



Figure 6 – Interaction diagram: Register reporting capabilities

In this use case, depicted in Figure 6, the source party sends its reporting capabilities to the target party. The source party's reporting capabilities are specified using a special well-known report profile called the METADATA report, which is exchanged using the same schema as any other report.

The METADATA report contains a collection of all the different types of reports that can be sent by the source party (VEN or VTN) that is generating the report. Each reporting capability specified in the METADATA report is uniquely identified by using the reportSpecifierID attribute, which is generated by the source party when registering the report. The reportSpecifierID will allow the target party to make subsequent requests for that specific report.

Each report specified in the METADATA report is a specification for the elements and attributes that can appear in that report and is based upon one of the well-known report profiles. By specifying the metadata for the report as part of this registration process, the actual reports that can be exchanged in the future need only refer to the reportSpecifierID along with the actual data without the need to send all the other report description attributes such as units, etc.

The interaction proceeds with the following steps:

- 1) The source party first sends its reporting capabilities to the target party by using the oadrRegisterReport payload. In general, the oadrRegisterReport payload is the same as the oadrUpdateReport payload except that it contains a METADATA report. The source party sends the special well-known report of type METADATA using the oadrReport schema as described above.
- 2) The target party responds with the oadrRegisteredReport payload. The target party's response may contain an oadrReportRequest object requesting which reports the source party should generate. If there are reports that the target party knows that it wants to receive from the source party, then it should make those requests as part of this step. This is similar to requesting a report using oadrCreateReport as specified in 6.2.2.3.
- 3) If the target party requests that the source party create any reports as part of step 2), then the source party responds with the oadrCreatedReport payload.

In essence, steps 2) and 3) are equivalent to, and use the same data structures as, the interaction in which the target party requests reports from the source party using oadrCreateReport, as specified in 6.2.2.3.

Every exchange of the METADATA report shall supply all the reporting capabilities of the source party (VTN or VEN) and will therefore supplant any previously exchanged METADATA report. Furthermore, the sending of the METADATA report implicitly cancels the sending of all reports that the source party has previously requested from the target party.

Key elements included in the metadata report payloads include:

- reportSpecifierID: a unique identifier for the reporting capability, used in subsequent requests;
- rid: a unique identifier for each data point offered by the report;
- Duration: the amount of time that data can be collected (or has been collected thus far for history);
- SamplingRate.oadrMinPeriod = maximum sampling frequency;
- SamplingRate.oadrMaxPeriod = minimum sampling frequency;
- SamplingRate.onChange = whether or not data is sampled as it changes.

If the target party does not request any reports as part of step 2), then the source party should assume that there are no reports to be generated and sent to the target party. Therefore, if there are any reports that the target party has previously requested from the source party, then it shall make those requests again as part of step 2).

The source party can therefore use this interaction at any time to request that the target party notify the source party of any reports that it should create and send. This provides a mechanism for the source and target party to synchronize both the reporting capabilities of the source party and the report requests of the target party.

6.2.2.3 Request reports

This use case, depicted in Figure 7, shows how one party can request reports from the other party. Note that any reports that are being requested shall correspond to one of the reports that were previously specified in the METADATA report that was previously sent by the other party.

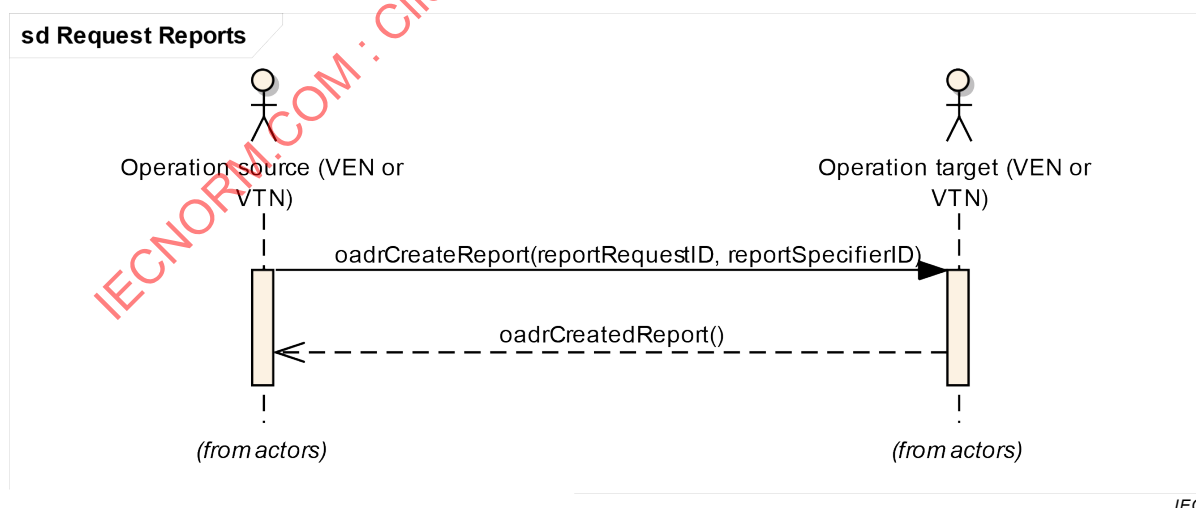


Figure 7 – Interaction diagram: Request reports

The source party requests a report from the target party by using the oadrCreateReport payload. That payload contains a set of reportSpecifierIDs that correspond to report capabilities in the METADATA report that was previously sent by the target party as part of the previously described oadrRegisterReport interaction (refer to 6.2.2.2). As part of the

report request, the source party specifies a set of reportRequestIDs that are used in subsequent operations on this report request.

In short, the reportSpecifierID is generated by the target during report registration (as specified in 6.2.2.2) to uniquely identify the report contents of a METADATA report, while the reportRequestID is generated by the source party of an oadrCreateReport (or oadrReportRequest object in an oadrRegisteredReport during registration) and associates the reportSpecifierID with a specific report request. For example, the source requesting a report can select only a certain subset of all data points (i.e. rIDs) that have been registered in the METADATA report identified by reportSpecifierID.

All subsequent reports sent from the target party that are in response to this request use the reportRequestID to identify the report.

The response to the oadrCreateReport payload is the oadrCreatedReport payload.

Note that the source party can only send the oadrCreateReport after the target party has sent its METADATA report as part of the reporting registration process. The exception to this is the METADATA report, which can be requested at any time by using the well-known string "METADATA" as the reportSpecifierID.

While the reportSpecifierID dictates the type of data that may appear in a report, there are some reporting parameters that are specified by the source party when making this request, including the following:

- The specific data items to appear in the report as specified by the rIDs from the corresponding METADATA report. The rID attribute is an identifier in the METADATA report that is associated with a specific type of value that may appear in the report. This is sometimes referred to as a data point and allows multiple data points to exist in a single report. The rIDs allow the source party to request a subset of possible data points that should be reported by the target party.
- Whether this is a one-time report or a recurring data stream.
- If it is a recurring data stream, the following temporal parameters are also specified: the time frame over which the report should be generated (i.e. next month, indefinitely, etc.), the time intervals at which the data points will be reported, and the frequency at which the report should be generated.
- If it is not a telemetry report, then the request specifies the span of time that the report should cover.

The meaning of dtstart in the report request defines the start time of the reporting period, while the dtstart in the returned report indicates the start time of the data collection period.

For periodic telemetry reports containing point data, if granularity and duration are the same, the report is sent immediately after the request. If granularity is smaller than duration, the delivery of the report will be delayed until enough data is collected.

Key elements included in the report request payloads include:

- Start time/Duration: when reporting should commence using granularity as sampling frequency. If Duration equals 0, it should report indefinitely.
- Granularity: Frequency of reported data. Value shall be between SamplingRate.oadrMinPeriod and SamplingRate.oadrMaxPeriod. If 0, the requester is requesting the report to use onChange (if available).
- reportBackDuration: the interval for sending reports (oadrUpdateReport or orderRegisterReport) back to the requester. If reportBackDuration == 0, then it's only a one-shot report.

See the example report request payloads in Annex E for more detailed information on the various attributes.

6.2.2.4 Send reports

This use case defines how the actual reports can be exchanged.



Figure 8 – Interaction diagram: Send reports

Figure 8 depicts the source party sending a report to the target party.

This operation can be performed by the source party only after a previous report request interaction is performed by the target party.

This operation uses the same `oadrReport` object as the report registration operation did, but is used to exchange a report with actual data elements as opposed to the METADATA report used in the registration process.

The reports sent in the `oadrUpdateReport` payload use the `EiReport` schema with the `reportRequestID` as defined by the target party in the previous report request interaction that prompted the sending of this report.

The response sent by the target party uses the `oadrUpdatedReport` payload to acknowledge receipt of the report. As part of the `oadrUpdatedReport` response, the target party may cancel the sending of any future reports using the optional `oadrCancelReport` object, which contains a list of `reportRequestIDs`. There is no confirmation of the cancellation if included in the `oadrUpdatedReport` payload, but if the report continues to be sent, the receiving party should use `oadrCancelReport` to cancel the report. In the case of periodic reports, if `reportToFollow` is set to true in the `oadrCancelReport` object by the target party, the source party is expected to send one final additional report.

See the various sample report payloads for more details on the report attributes.

6.2.2.5 Cancel reports

This interaction, depicted in Figure 9, is used by the source party to cancel ongoing (i.e. periodic) reports that are being generated by the target party.

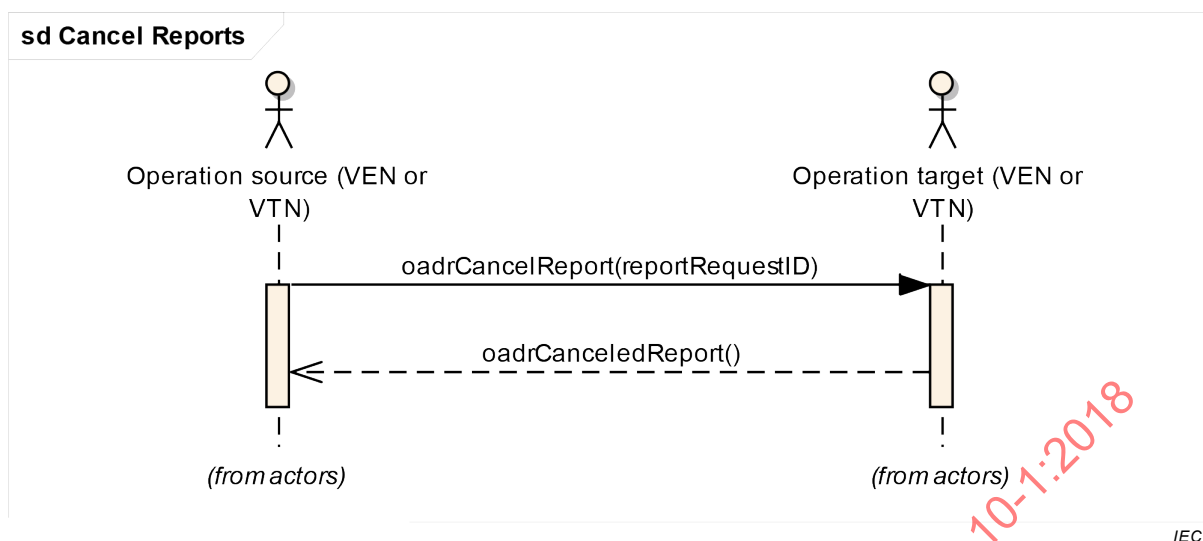


Figure 9 – Interaction diagram: Cancel reports

The source party uses the `oadrCancelReport` payload with the appropriate `reportRequestIDs` that were specified by the source party in a previous request interaction (see 6.2.2.3). Upon receiving the `oadrCancelReport` payload, the target party stops generating and sending the reports corresponding to the `reportRequestIDs`.

The response to the `oadrCancelReport` payload is the `oadrCanceledReport` payload that is sent by the target party to acknowledge the cancellation of the report.

Note that both `oadrCanceledReport` and `oadrCreatedReport` return a list of pending reports in the `oadrPendingReports` object, which includes all reports that are scheduled for future delivery.

When cancellation of the periodic report is requested by the target party, if `reportToFollow` is set to true in `oadrCancelReport`, the source party is expected to send one final additional report.

6.3 Registration service

6.3.1 Service operations

This document uses the `EiRegisterParty` service to support in-band registration of VENs with VTNs. The following service operations (listed in Table 3) are supported:

Table 3 – EiRegisterParty payloads

Request payload	Response payload	Requestor	Responder
<code>oadrQueryRegistration</code>	<code>oadrCreatedPartyRegistration</code>	VEN	VTN
<code>oadrCreatePartyRegistration</code>	<code>oadrCreatedPartyRegistration</code>	VEN	VTN
<code>oadrCancelPartyRegistration</code>	<code>oadrCanceledPartyRegistration</code>	VEN VTN	VTN VEN
<code>oadrRequestReregistration</code>	<code>oadrResponse</code>	VTN	VEN

Registration can optionally begin with the VEN querying the VTN to determine what profiles, transports, and extensions it may support using the `oadrQueryRegistration` payload (see Figure 10). This query operation can be initiated using any of the supported transports, however the VEN will need to be configured out-of-band with the address of the VTN in order

to initiate the query. The response to the query is the `oadrCreatedPartyRegistration` payload and contains information on all the profiles and transports supported by the VTN in addition to any supported extensions to the profile. The information received by the VEN can be used to determine the best configuration to use when formally registering as described in the balance of Clause 6.

Annex B provides normative information on schema extensions.

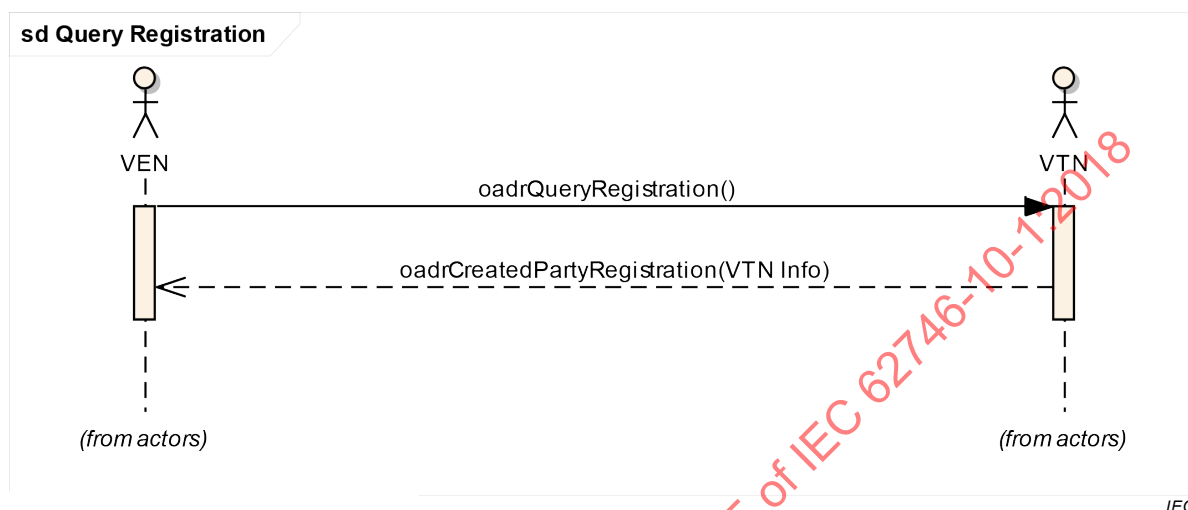


Figure 10 – Interaction diagram: Query registration

Registration is always initiated by the VEN with the `oadrCreatePartyRegistration` payload (see Figure 11). This payload provides the information on the profile and transport the VEN has decided to use for communication with the VTN, in addition to other registration-related information (see Table 4). The VTN responds with an `oadrCreatedPartyRegistration` containing all the profiles and transports supported by the VTN, IDs, and other registration-related information (see Table 5). The VTN returns a `registrationID` in its response payload, which is used for subsequent operations pertaining to this registration instance.

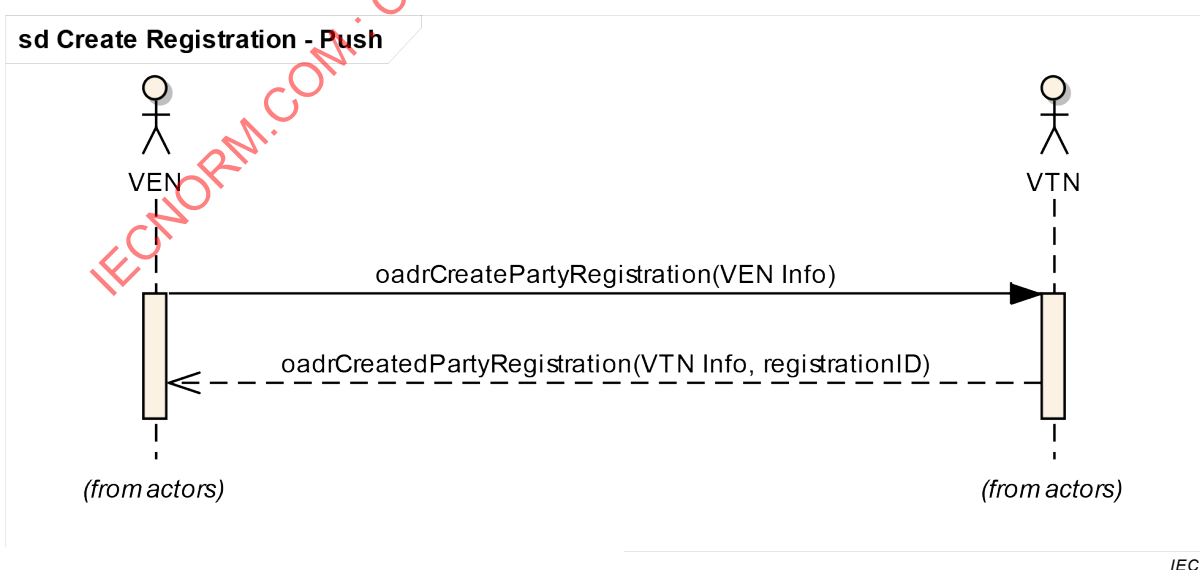


Figure 11 – Interaction diagram: Create registration

If the VEN's registration information changes, the VEN can reregister at any time using the `oadrCreatePartyRegistration` payload referencing the current `registrationID` (see Figure 12).

The same registrationID will be maintained across reregistrations until one of the parties cancels the registration.

If the VTN's registration information changes, the VTN can request the VEN to reregister using the `oadrRequestReregistration` payload. The response to this request is an `oadrResponse` acknowledgement followed by an asynchronous `oadrCreatePartyRegistration` request from the VEN.

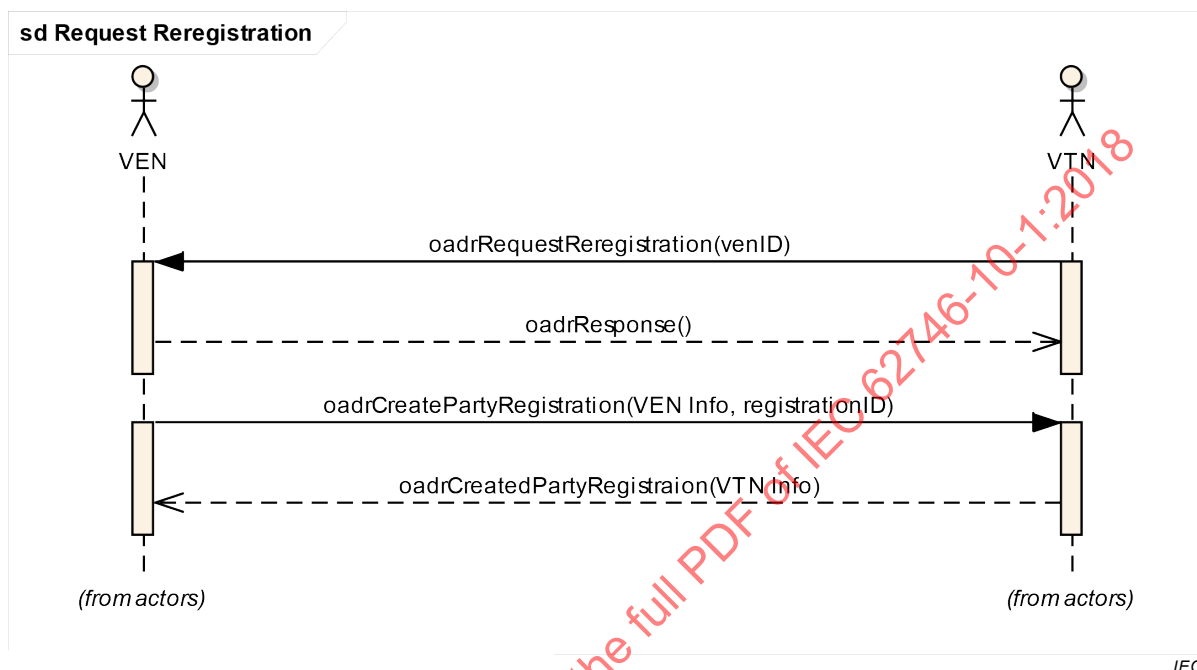


Figure 12 – Interaction diagram: Request reregistration

The VTN or VEN can cancel an active registration using the `oadrCancelPartyRegistration` payload, referencing the registrationID (see Figure 13). The other party responds with an `oadrCanceledRegistration` payload.

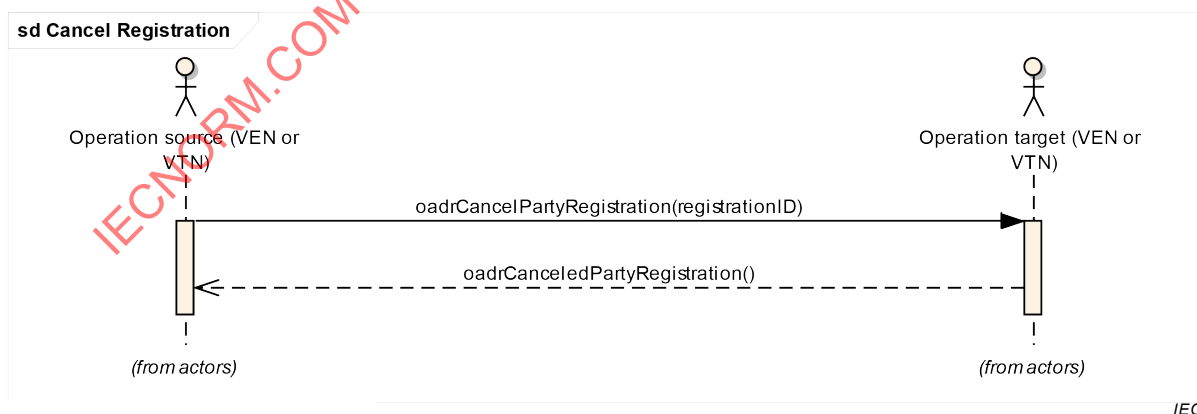


Figure 13 – Interaction diagram: Cancel registration

The sequence diagrams shown are for a PUSH interaction pattern. In an HTTP PULL model, the VEN shall periodically poll the VTN using `oadrPoll` to provide the VTN an opportunity to cancel the registration or request the VEN to reregister. Please refer to the documentation for `oadrPoll` in 6.5 for sample sequence diagrams of the PULL interaction pattern.

6.3.2 Registration information

Table 4 and Table 5 outline registration elements defined and whether the information appears in the VEN `oadrCreatePartyRegistration` request or the VTN `oadrCreatedPartyRegistration` response.

Table 4 – VEN information in `oadrCreatePartyRegistration` payload

Registration item	Comments
<code>requestID</code>	
<code>registrationID</code>	RegistrationID is only included in this payload when reregistering.
<code>venID</code>	Unless preconfigured, the <code>venID</code> is not included in the initial registration payload, but shall be present and correct for reregistration.
<code>oadrProfileName</code>	The profile the VEN selects to use for communication with the VTN. Either 2.0a or 2.0b. This document refers to 2.0b only.
<code>oadrTransportName</code>	The transport the VEN selects to use for communication with the VTN. Either XMPP or simpleHTTP.
<code>oadrTransportAddress</code>	If the selected transport is simpleHTTP or XMPP and the <code>oadrHttpPullModel</code> is false, this shall contain the address of the VEN server.
<code>oadrReportOnly</code>	This Boolean element is used to indicate the implementation is a Report Only instance of the B profile.
<code>oadrXmlSignature</code>	This Boolean element is used to indicate the implementation supports XmlSignatures.
<code>oadrVenName</code>	This is an optional human readable name for the VEN.
<code>oadrHttpPullModel</code>	This Boolean element is used to indicate the implementation uses the simpleHTTP PULL exchange model.

Table 5 – VTN information oadrCreatedPartyRegistration payload

Registration item	Comments
EiResponse Object	Contains standard payload response status information.
RegistrationID	A value used to identify the registration instance. Used for reregistration and cancelation.
venID	The venID is assigned by the VTN and returned in the registration response.
vtnID	Provided to the VEN for inclusion in payloads.
oadrProfiles:oadrProfile:oadrProfileName	A collection of one or more profiles and the associated profile name.
Nested within each profile... oadrTransports:oadrTransport:oadrTransportName	A collection of one or more transports and the associated transport name.
oadrRequestedOadrPollFreq:duration	The polling interval (i.e. the duration between two polls) using oadrPoll should be as requested by the VTN during registration in oadrRequestedOadrPollFreq. Conformance rule 500 defines the constraints for oadrRequestedOadrPollFreq.
oadrServiceSpecificInfo:oadrService:oadrServiceName oadrServiceSpecificInfo:oadrService:oadrInfo:oadrKey oadrServiceSpecificInfo:oadrService:oadrInfo:oadrValue	An object that can contain key-value pair information specific to a particular named service.
oadrExtensions:oadrExtension:oadrExtensionName oadrExtensions:oadrExtension:oadrInfo:oadrKey oadrExtensions:oadrExtension:oadrInfo:oadrValue	An object that can contain key-value pair information specific to a particular named extension.

6.4 Opt service

6.4.1 Service operations

This document specifies the EiOpt service to create and communicate Opt-In and Opt-Out schedules from the VEN to the VTN. These schedules define temporary changes in the availability, and may be combined with longer-term availability schedules and the Market Context requirements to give a complete picture of the willingness of the VEN to respond to EiEvents received by the VEN. The service operations listed in Table 6 are supported.

Table 6 – EiOpt payloads

Request payload	Response payload	Requestor	Responder
oadrCreateOpt	oadrCreatedOpt	VEN	VTN
oadrCancelOpt	oadrCanceledOpt	VEN	VTN

The VEN uses the oadrCreateOpt payload (see Figure 14) to accomplish one of the following objectives:

- to communicate to the VTN a period of temporary availability for a specific set of eiTargets,
- to communicate to the VTN a period of temporary un-availability for a specific set of eiTargets,
- to optIn or optOut of a previously acknowledged event for a specific set of eiTargets.

oadrCreateOpt payload includes an optID, which can be used in subsequent operations to reference this schedule. The VTN's response to oadrCreateOpt is an oadrCreatedOpt payload that includes an optID, which can be used in subsequent operations to reference this schedule.

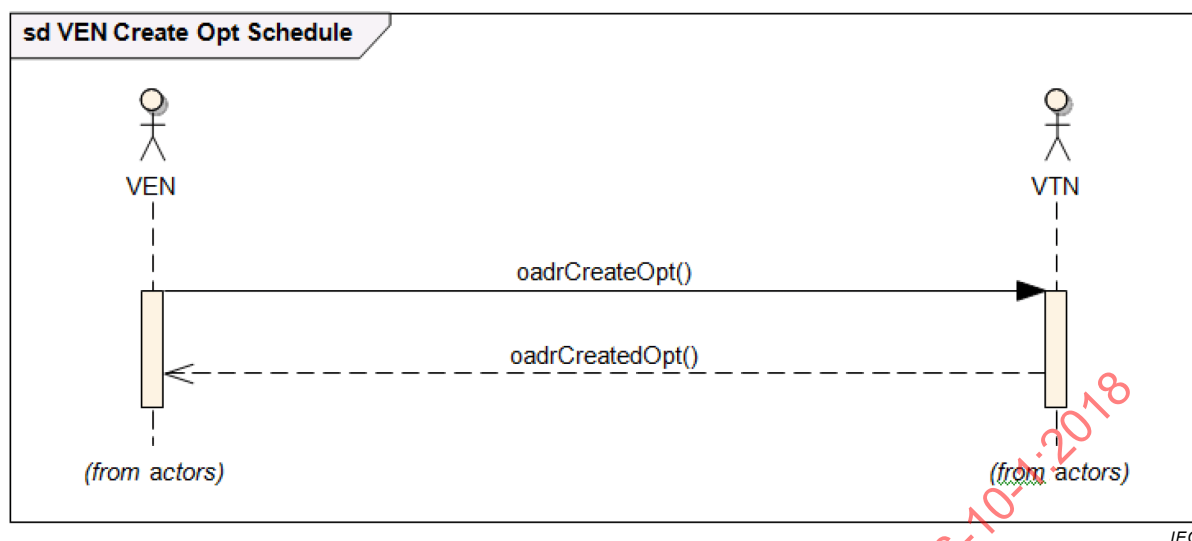


Figure 14 – Interaction diagram: Create opt

The VEN may at any time cancel a temporary availability schedule by using the oadrCancelOpt with an optID referencing the schedule to be cancelled (see Figure 15).

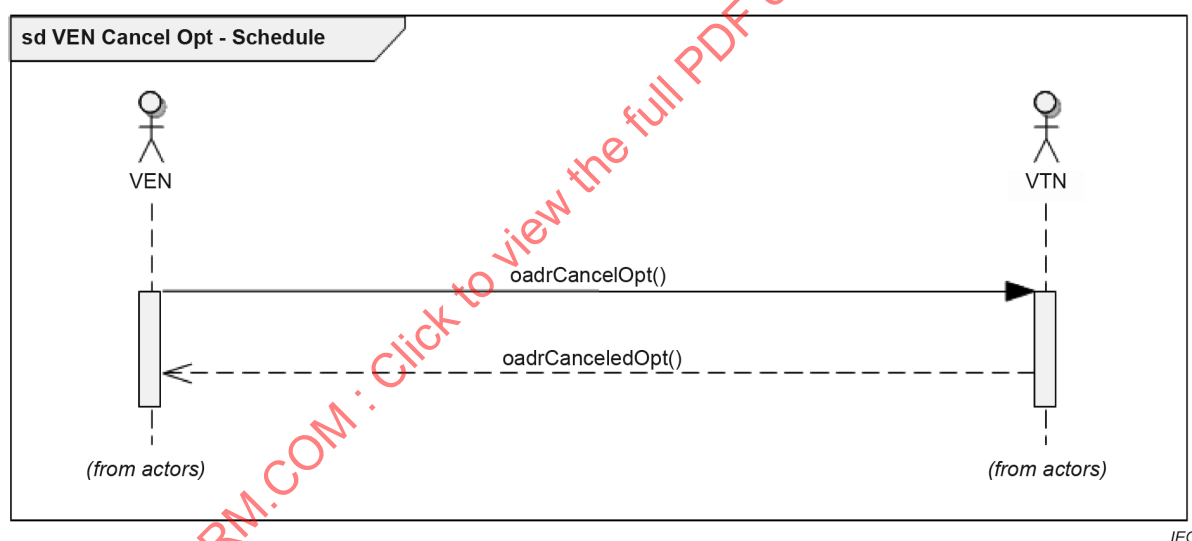


Figure 15 – Interaction diagram: Cancel opt

6.4.2 Detail requirements

If marketContext is present in the oadrCreateOpt payload, the opt schedule shall only apply to the VEN's availability with respect to events generated within the specified marketContext.

A VEN shall respond with an EiEvent oadrCreatedEvent payload in response to an oadrDistributeEvent. However, further qualifications of the optType state for the event can be accomplished by sending either an oadrCreatedEvent or an EiOpt oadrCreateOpt payload with the eventID specified. Once oadrCreateOpt is used, it will take precedence and subsequent changes to the opt state by oadrCreatedEvent will be ignored.

The oadrCreateOpt eiTarget shall contain the venID of the VEN initiating the opt schedule. If only the venID is specified in eiTarget, then the opt schedule applies to all of the resources associated with that VEN. If additional eiTarget sub-elements are defined in addition to venID, these sub-elements shall be OR'd together to define a subset of the VEN's resources that the

opt schedule shall apply. Note that the VEN may send multiple opt schedules for different sets of resources identified by the eiTarget element.

A new opt schedule (optIn or optOut) sent via oadrCreateOpt shall be handled as follows when a previously sent opt schedule is still active:

- If only the venID is specified in eiTarget, the previous opt schedule with respect to future availability is replaced in its entirety by the new opt schedule for all resources associated with the VEN.
- If sub-elements are specified in eiTarget in addition to the venID, then an opt schedule shall be generated for these resources, overwriting necessarily any previously defined schedules for specific resources.
- Previously defined opt schedules for resources that are not specified by eiTarget shall remain unchanged.

Many aspects of an opt schedule are program-specific, including:

- the type of schedules that a VEN is allowed to issue: optIn, optOut, or both;
- the default opt state (optIn or optOut) for time periods not defined by either an opt or avail schedule (if defined);
- whether optIn or optOut schedules have precedence in the situation where they have overlapping time frames;
- whether overlapping time frames are allowed within the same schedule (optIn or optOut) and if they are not allowed, the expected exception handling behaviour;
- whether the VTN is allowed to send events during an optOut period.

6.5 Poll service

As new services are added, there is a need for pull-only VENs to periodically poll the VTN, particularly for VTN-initiated use cases where the information is not necessarily periodic and the VEN cannot predict when a VTN might want to send some information. In the PUSH case, this is a non-issue. However, in the PULL case, this requires the VEN to periodically poll the VTN, possibly at multiple different service endpoints, to retrieve all the information that the VTN might want to provide. oadrPoll provides a solution that allows the PULL VEN to emulate the PUSH message exchange pattern from VTN to VEN.

oadrPoll is a service-independent polling mechanism used by VENs in a PULL model to request pending service operations from the VTN. The VEN queries the poll endpoint and the VTN responds with the same message that it would have "pushed" had it been a PUSH VEN. If there are multiple messages pending a "push," the VEN will continue to query the poll endpoint until there are no new messages and the VTN responds with an eiResponse payload.

The allowed response payloads sent by the VTN in response to an oadrPoll request are:

- oadrResponse,
- oadrDistributeEvent,
- oadrCreateReport,
- oadrRegisterReport,
- oadrCancelReport,
- oadrUpdateReport,
- oadrCancelPartyRegistration,
- oadrRequestReregistration.

The rules for which payloads are valid and how those payloads are delivered are the same as if the VTN had initiated the operations and pushed the payloads to the VEN. Only one operation payload may be sent by the VTN in response to the oadrPoll message.

If a logical response is required by the VTN to the received operational payload, the VEN shall send that logical response asynchronously via a transport request. The VTN should acknowledge this logical response with an oadrResponse payload.

The VTN can optionally ignore an oadrPoll if it has not received an expected logical response to a payload delivered as a response to a previous poll. The VTN should be coded such that after some timeout it gives up waiting for the expected response and resumes responding to oadrPoll.

The oadrPoll request has a single sub-element, the venID initiating the oadrPoll. Below is an example payload:

```
<oadr:oadrPoll ei:schemaVersion="2.0b">
  <ei:venID>VEN_123</ei:venID>
</oadr:oadrPoll>
```

The following sequence diagrams in Figures 16 to 19 illustrate typical patterns using oadrPoll:

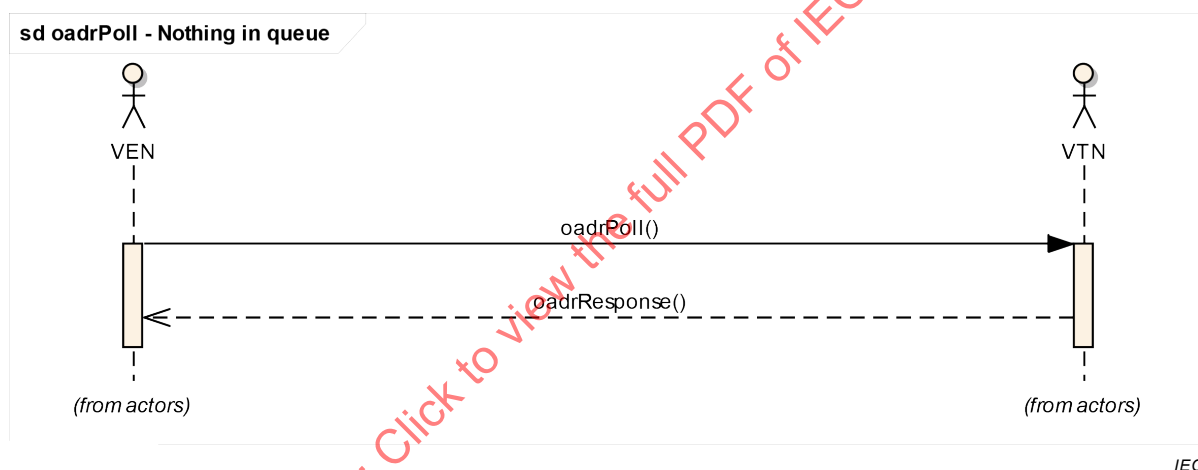


Figure 16 – Interaction diagram: oadrPoll (nothing in queue)

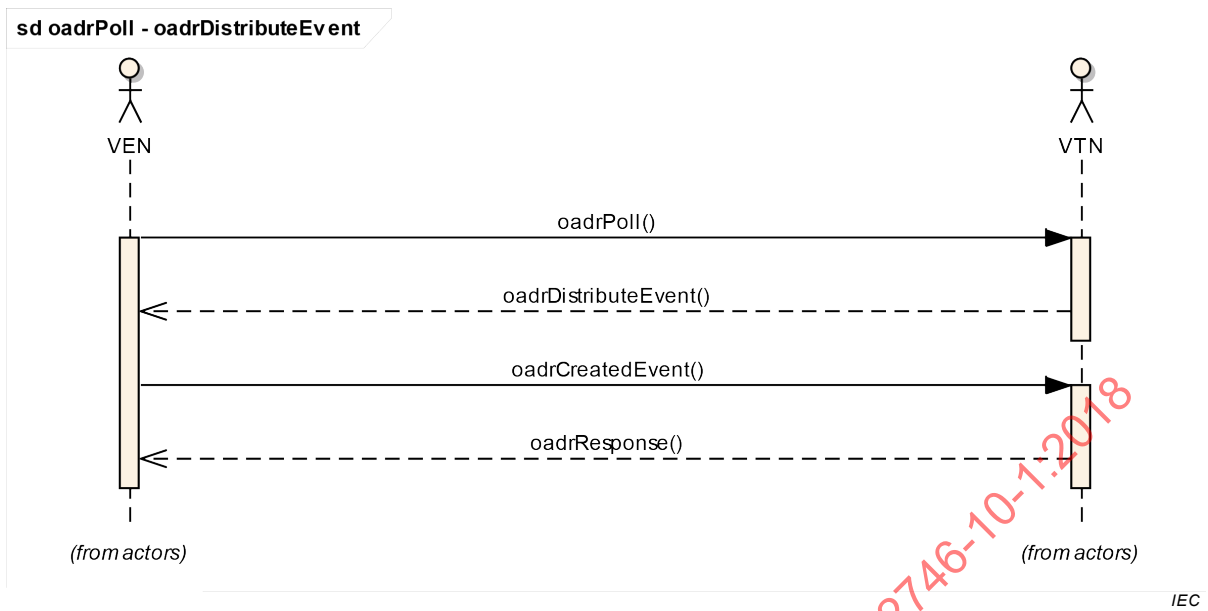


Figure 17 – Interaction diagram: oadrPoll (oadrDistributeEvent reply)

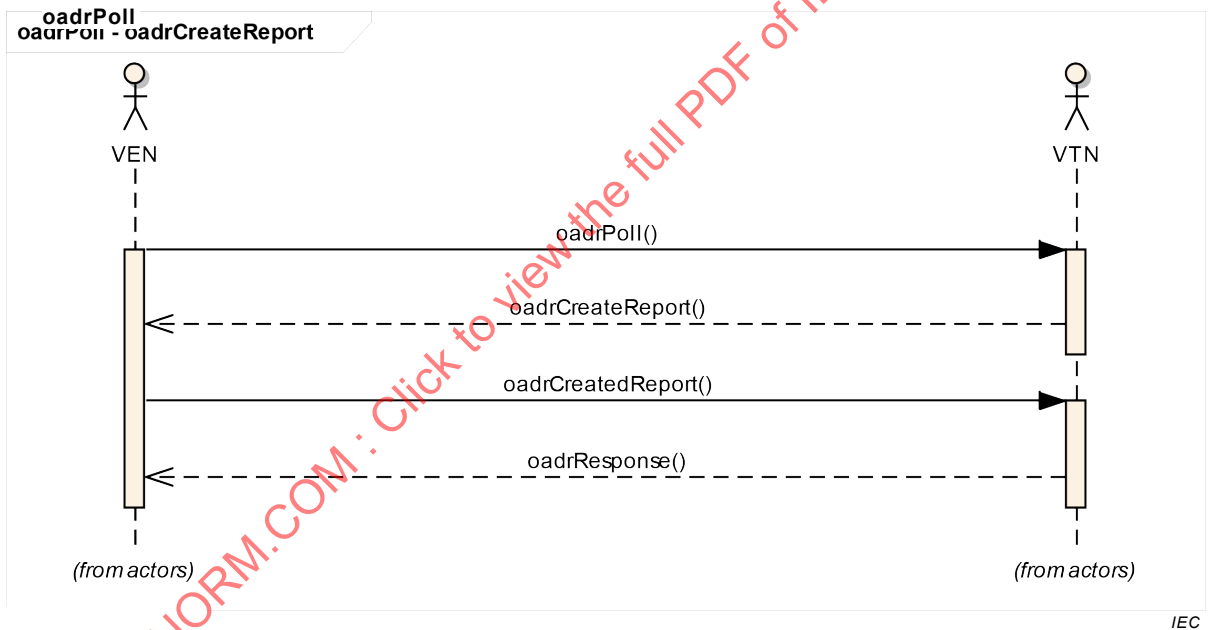


Figure 18 – Interaction diagram: oadrPoll (oadrCreateReport reply)



Figure 19 – Interaction diagram: oadrPoll (request reregistration reply)

Annex C provides several example scenarios that shall be supported by all implementations.

6.6 Application error codes

This document uses a three-digit numeric error code starting with the digit "4" to communicate application layer errors in the responseCode element of the various response objects in the schema. There are two general categories of errors: compliance errors and deployment errors. The second digit of the error code shall be used to differentiate between these two general categories with "5" indicating compliance errors and "6" indicating deployment errors. Error scenarios documented in the conformance rules will specify which of the error codes listed below should be used. For other error scenarios, the VEN or VTN implementer will have to use their best judgment to determine the appropriate application layer error code to use.

Compliance error codes

- 450 – Out of sequence (event ordering, uid ordering, modification number sequencing)
- 451 – Not allowed (changing an event in the past)
- 452 – Invalid ID (eventID, optID, requestID, registrationID, etc.)
- 453 – Not recognized (reportName, signalName, etc.)
- 454 – Invalid data (out of range signal or report data)
- 455 – Open
- 456 – Open
- 457 – Open
- 458 – Open
- 459 – Compliance error – Other

Deployment errors

- 460 – Signal not supported (recognized, but not supported)
- 461 – Report not supported (recognized, but not supported)
- 462 – Target mismatch (cannot resolve target/market context to VEN or its resources)
- 463 – Not registered/authorized
- 464 – Open
- 465 – Open
- 466 – Open
- 467 – Open
- 468 – Open
- 469 – Deployment error – Other

7 Transport protocol

7.1 General

This document supports a small number of transport protocols to accommodate different deployment scenarios. 2.0b VENs can either support HTTP or XMPP, or may support both. VTNs shall support both HTTP and XMPP.

7.2 Simple HTTP

7.2.1 General

Simple HTTP in this document refers to an HTTP implementation that uses HTTP POST over TLS to propagate the payloads.

7.2.2 PUSH and PULL implementation

7.2.2.1 PUSH definition

In PUSH mode, messages may be sent from the VTN to the VEN (pushed). In order to use push, the VEN shall expose HTTP URI endpoints (an HTTP server) to which the VTN may send requests such as `oadrDistributeEvent`. While this is the most efficient way to execute the communication over HTTP, it presents technical difficulties as the VEN can reside behind a network firewall.

7.2.2.2 PULL definition

In PULL mode, all operations are initiated by the VEN to the VTN. This can be thought of as a 'polling' mode, where the VEN periodically asks for updates from the VTN. The PULL mode removes the requirement for an HTTP server on the VEN, avoiding the technical limitation presented by the possibility of a network firewall in front of a VEN. However, the PULL mode has its own limitations, namely latency (due to limited polling frequency) and increased bandwidth requirements.

The PULL mode can involve a 'two-phase' execution to complete some operations. This is due to the nature of the VEN initiating the HTTP request. In a PUSH mode, the VTN can notify a VEN of a new event via the `oadrDistributeEvent` operation. The VTN would send a request with an `oadrDistributeEvent` payload, to which the VEN responds with HTTP 200 followed by an asynchronous `oadrCreatedEvent`.

However, in the PULL model, the VEN requests events from the VTN using `oadrPoll`, to which an `oadrDistributeEvent` payload is sent in the response. After parsing the response, the VEN still needs to acknowledge the creation of any new events by making a second request using the `oadrCreatedEvent` operation on the VEN.

7.2.3 Service endpoint URIs

The endpoint names will be of the form:

`https://<hostname>(:port)/(prefix/)OpenADR2/Simple/2.0b/<service>`

- "prefix" is an optional URI path prefix that can be used to separate IEC 62746-10-1 services from other services that can reside on the same HTTP server.
- "Simple" indicates the simple XML over HTTP protocol.
- <service> is the name of the service (e.g. "EiEvent", "EiReport", "EiOpt", "EiRegisterParty", "OadrPoll").

The "operation" portion of a service is defined by the XML payload sent in a request, for example, an oadrDistributeEvent payload root element specifies the oadrDistributeEvent operation.

VTN to VEN traffic via HTTP should flow through the same well-defined service endpoints used when communicating in the opposite direction. The oadrCreatePartyRegistration:oadrTransportAddress element should contain the base address for the VEN, such as "http://myaddress:8080/prefix/". When sending traffic to the VEN, the VTN should concatenate the base address to the well-known service endpoints to form a complete endpoint address such as `http://129.6.252.49:8080/prefix/OpenADR2/Simple/2.0b/EiReport`.

Note that for implementations that expose both a VEN and VTN (such as an aggregator), these implementations shall use different URI endpoints for their VTN and VEN interfaces, for example, `https://mycompany.com/myVTN/OpenADR2/Simple/2.0b/EiEvent` and `https://mycompany.com/myVEN/OpenADR2/Simple/2.0b/EiEvent`.

7.2.4 HTTP methods

All messages shall be sent using the HTTP POST method. This helps to avoid caching and allows all operations to contain a payload in the HTTP request body.

7.2.5 Failure conditions

The following failures can occur for a given operation:

- TCP (or below) fails;
- TLS negotiation fails;
- HTTP fails (HTTP error code);
- application acknowledgement fails (application error code);
- response failure (timeout or application error code).

The proper action for each failure condition depends upon the application and the operation being attempted. Since all operations are idempotent, it is safe to retry any operation.

In the case of TCP failure, it is always recommended to retry the operation.

Application-level errors are defined in 6.6.

7.2.6 HTTP response codes

The following HTTP status codes, defined in IETF RFC 2616, are used in this document:

200 OK – any response that the endpoint was able to handle completely and send a valid response payload. This includes responses that can indicate an error at the application level

(e.g. 'you gave me an invalid event ID'). Errors that indicate a failure at the transport level are handled by transport-level HTTP error codes.

404 Not Found – the VEN or VTN does not support the requested operation. The requestor shall not re-send the request.

406 Not Acceptable – if a payload is sent that does not validate against the schema, or if a request content-type is unsupported. The requestor shall not re-send the request without first modifying it.

501 Not Implemented – if any request is made with an unsupported HTTP method. The requestor shall not re-send the request without fixing the HTTP method.

503 Service Unavailable – indicates that the server is temporarily unavailable, possibly due to inability to handle the current request load. This error in particular shall indicate to the requestor that it shall execute quiesce logic in order to not put further strain on the server. The requestor shall retry the request after the proper quiesce period.

500 Internal Server Error – undefined or unexpected server error. The requestor can retry the request after a quiesce period.

For all error (non-200) codes, the content body of the response is undefined. The server can choose to send some informational message in the response, but the requestor is not obligated to parse or understand it.

All application-level error conditions are conveyed through the status code element of oadrResponse payload.

7.2.7 Message timeouts

There are no prescribed connect or response timeout thresholds for this document. Implementations shall allow configurable timeouts in order to handle IP networks with different latency characteristics. HTTP clients shall use a request timeout of at least 5 s.

7.2.8 Message retry/quiesce behaviour

When a request fails for any reason (either due to physical or network-level failure or a timeout), the requestor shall institute 'back-off' or quiesce logic to avoid flooding the network or receiver with requests.

Clients shall begin quiesce at some small interval (say, 1 s) plus or minus some random 'jitter,' which is a small percentage of that interval (say, 10 %). For example, the first quiesce interval for a device might be between 0,9 s and 1,1 s. Then the device may retry the request. If subsequent retries fail, the quiesce interval shall double from the prior interval, again adding a random jitter of $\pm 10\%$ of that interval. This doubling for subsequent failures shall continue up to some maximum, probably dependent on the poll interval in the case of a VEN polling a VTN. This is known as a "truncated binary exponential back off algorithm."

7.2.9 PULL timing

The ability to configure the poll interval for a PULL mode is not defined in this document, however VEN implementations shall allow the poll interval to be configurable at a millisecond resolution. This adjustment resolution is only mandatory for the communication protocol and this requirement does not infer that building systems require millisecond-level control. Responsiveness of building systems is program-dependent.

In this document, a VTN may configure the minimum poll interval for the VEN by setting oadrCreatedPartyRegistration:oadrRequestedOadrPollFreq accordingly.

VENs shall also be configurable to induce some 'jitter' – a random offset from the absolute poll interval – in order to avoid request spikes on the VTN caused by many VENs initiating a PULL request at the exact same instant within the poll interval.

7.2.10 HTTP headers

7.2.10.1 General

The HTTP headers in 7.2.10.2 to 7.2.10.8 shall appear in requests or responses (where indicated):

7.2.10.2 Accept

The accept request header specifies the expected content-type of a response. Since responses are always "application/xml", the accept header may be omitted. However, if it is included, the value of the accept request header shall always be "application/xml".

7.2.10.3 Accept-Encoding

This request header indicates if a client supports content compression of the response payload. A VEN may include this header in a request if it supports content compression such as gzip or deflate. If the VEN includes this header, the VTN shall honor it and compress the response content using one of the methods given in the request header.

7.2.10.4 Content-Encoding

If a VTN is responding to a request for which it has a compressed payload, it shall include a content-encoding response header indicating the correct encoding method, such as gzip or deflate.

PUSH operations from a VTN shall not utilize the content-encoding header in the request, since it would require the VTN to have a priori knowledge of which content-encodings are supported by each and every VEN.

7.2.10.5 Content-Length

The content-length header shall be used according to IETF RFC 2616 to indicate content body size of all request and response payloads.

Note that 'chunked' transfer encoding (where content-length is unknown) shall not be used in this document. Implementers shall assume that the total content body length is known when the response headers are sent, and shall not send chunked responses.

7.2.10.6 Content-Type

This shall be used for request and response messages, indicating payload MIME type. The appropriate value is "application/xml". The content-type may also specify a character encoding. For this document, the only supported character encoding is UTF-8. If a charset is included, the entire header value shall appear as "application/xml; charset=utf-8".

7.2.10.7 Host

The Host header shall be included in all requests in accordance with IETF RFC 2616, section 14.23 (3).

7.2.10.8 User-agent

The requestor may include the User-Agent header, but its presence shall not be relied upon, nor shall it materially affect the behaviour of the server handling the request.

7.3 Transport-specific security

7.3.1 General

TLS shall be used to encrypt all traffic. The client shall always validate the server's TLS certificate given during the handshake.

7.3.2 TLS client certificate

Client certificates shall be used for mutual HTTP client authentication. The entity initiating the request (the client) shall have an X.509 certificate that is validated by the server during the TLS handshake. If no client certificate is supplied, or if the certificate is not valid (e.g. it is not signed by a trusted CA, or it is expired), the server shall terminate the connection during the TLS handshake.

If the certificate appears valid during the TLS handshake, the connection is established and the HTTP request proceeds. Once the server receives the HTTP request, it shall perform authentication, given the credentials in the client certificate. The VTN shall use the certificate public key as the primary identification mechanism and not the identity in the certificate. The client credentials shall be compared to ensure they match the `venid` that appears in the payload of the client request. If credentials do not match, or if the server otherwise determines that the request is not authorized, it shall respond with an application layer 463 error in the `eiResponse` object of an appropriate message (e.g. in an `oadrCreatedPartyRegistration` response if the request from the VEN was an `oadrCreatePartyRegistration`).

To implement this document, any acceptable security certificate can be used as supported by the DR program operator. One option is to use security certificates that are regulated by the OpenADR Alliance Certificate Policy¹.

7.4 XMPP

7.4.1 General

XMPP is a stateful, bi-directional protocol ideal for transmitting messages in XML format. The core protocol is specified in IETF RFC 6120, and addressing is defined in IETF RFC 6122 and is also referenced in Annex G of 61968-100:2013. RFC 6120 section 13.7 describes certificate-based authentication handling. It should be noted that in the intended implementation of this document, the XMPP server is part of the VTN server infrastructure for security purposes. If the XMPP server resides outside the VTN's security framework, additional security measures may be necessary.

7.4.2 Exchange model implementation

By nature, XMPP is a bi-directional, stateful protocol. As such, any client using XMPP can implement both PUSH and PULL operations seamlessly, with the exception of `oadrPoll`, which is disallowed over XMPP. PULL operations are simply those initiated by the VEN, and PUSH operations are initiated by the VTN.

7.4.3 Service endpoints

A jabber identifier (JID), which is similar to an email address, defines both VTN and VEN service endpoints. The fully qualified JID performs the same function as an endpoint URI in an HTTP implementation. The definition of JIDs is further described in 7.4.5.1.

¹ OpenADR Alliance Certificate Policy is a publication of the OpenADR Alliance. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

7.4.4 Service execution

Because XMPP is a message-based protocol, execution of services occurs by passing XMPP messages that contain an XML payload.

7.4.5 Implementation of XMPP features

7.4.5.1 JIDs

While the prospect of using multiple resources per JID opens some interesting possibilities (e.g. exposing each building at a facility as a different resource), this strategy should be used with some caution, as the same mechanism is not supported by other transports such as HTTP.

VEN clients should always authenticate with a fully qualified JID (rather than allow the server to assign a random resource ID.)

VTN clients can choose to expose the operations either as an XMPP service or as a client JID. Although most service operations will be 'pushed' from VTN to VEN, a VEN can still need to send some requests to the VTN. Rather than use some configuration to define the JID for VTN services, services shall be discovered at runtime via service discovery (described in 7.4.5.9).

VTN to VEN traffic via XMPP shall flow through an endpoint defined by a single fully qualified JID. This VEN JID will be communicated to the VTN as part of the `oadrCreatePartyRegistration` payload in the `oadrTransportAddress` element. The resource name used for the JID can be arbitrary, such as "client". For instance, the contents of the `oadrTransportAddress` element might be something like "ven@domain/client".

VEN to VEN traffic is disallowed and shall be filtered by the XMPP server.

7.4.5.2 Use of the packet 'type' attribute

All operations shall use the 'set' IQ type attribute for all service operations.

7.4.5.3 Use of message versus IQ packets

Any service operations that require an application-level response (e.g. a response payload) or a transport-level response code shall use an IQ packet. The XMPP protocol then mandates a response IQ from the recipient.

Any service operations that do not require a response, such as a broadcast operation (or possibly feedback), may use a message packet to send the payload. Handling of the packet should be identical to if it were an IQ, except that the recipient sends no response. This is a bandwidth optimization for use cases where a response is not expected, and a response IQ would be unnecessary.

7.4.5.4 Error handling

7.4.5.4.1 Transport-level status codes

Under normal conditions, an IQ response will have a `type='result'` attribute. This is analogous to an HTTP 200 response. Service operations that define an empty HTTP 200 response will simply look like an IQ element with no child payload element:

```
<iq type='result'  
  to='ven1234@xmpp.myvo.net/client'  
  from='vtn.xmpp.somevtn.net'  
  id='1' />
```

7.4.5.4.2 Transport-level errors

Operations that result in a transport-level error will have a type='error' and a child <error> element that indicates the transport-level status code as described in IETF RFC 6120, section 8.3. For example:

```
<iq type='error'
  to='ven1234@xmpp.myco.net/client'
  from='vtn.xmpp.somevtn.net'
  id='2'>
  <error type='modify'>
    <bad-request xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

Since XMPP does not use numeric status codes, direct mapping of HTTP status codes is not possible, and logic for handling transport-level errors is outside the scope of this document. Guidelines for handling different types of XMPP errors can be found in IETF RFC 6120, section 8.3.3.

If a client receives a payload that does not adhere to the schema, a client should respond with a bad-request error as defined in IETF RFC 6120, section 8.3.3.1.

Note that application-level errors shall be handled the same as in HTTP. That is, a normal 'result' XMPP response should be returned with the application-level error details in the payload.

7.4.5.5 Presence

VEN clients shall use presence packets to notify subscribed entities of online and offline status as described in IETF RFC 6121, section 4. All other status broadcasts from a client should be considered informational.

After a VEN completes session negotiation, it shall send an 'available' presence stanza as defined in IETF RFC 6121, section 4.2.1:

```
<presence />
```

If a VEN deliberately terminates its XMPP connection (e.g. owing to a graceful shutdown, not an unexpected, sudden connectivity loss), it shall first send an 'unavailable' presence stanza as defined in IETF RFC 6121, section 4.5.1:

```
<presence type='unavailable' />
```

The VTN should not generally advertise its presence to the VENs. A VTN could subscribe to VENs and receive presence updates, but can cause scalability issues and should be used with caution.

7.4.5.6 XMPP ping

Support for XMPP ping (as specified in XMPP Standards Foundation specification XEP-0199 version 2) is mandated from the VEN to the XMPP server. The VEN can notice disconnection from the XMPP server and, for example, re-establish connection or send an alarm if one or more pings are not replied to within a certain time. XMPP ping intervals and other parameters related to XMPP ping may be set locally, and exchange of these parameters is not required as part of registration.

7.4.5.7 Authentication

All clients shall support SSL/TLS and authentication as defined in IETF RFC 6120, sections 13.8 and 13.9.4. Clients shall also implement simple authentication and security layer SASL EXTERNAL in order to use certificate authentication as defined in IETF RFC 6120.

The username of the VTN/VEN logging in to the XMPP server shall match the Common Name (CN) field of the x509 certificate that is used in the authentication process.

7.4.5.8 Rosters

While rosters are a core feature of XMPP and commonly occur during XMPP session initiation, they have no defined role within this document and, therefore, their use is undefined within the scope of this document. A roster request may be performed by a client; however, under most circumstances, the server can simply respond with an empty roster. Optimally, a client (particularly VEN clients) should not request a roster from the server for the purposes of this document.

More specifically, VEN clients should not typically add other entities to their roster, as this will result in additional network overhead owing to presence broadcasts.

VTN clients might choose to use the roster as a mechanism to track the presence of online VEN clients, however this is not strictly a feature or requirement of this document.

7.4.5.9 Service discovery

7.4.5.9.1 General

XMPP service discovery XEP-0030 shall be used to define the address for services on the VTN. This allows the VTN flexibility in how it implements services, while removing the need for additional configuration on the VEN, which would be outside the scope of this document.

7.4.5.9.2 Discovery of the features

After a VEN has completed session initiation, it may perform an XEP-0030 'info' query to the bare domain to which the VEN has authenticated. For example, if a VEN has connected as 'ven1@xmpp.myco.net/client' then its initial service discovery query would look like the following:

```
<iq type='get'
  from='ven1@xmpp.myco.net/client'
  to='xmpp.somevttn.net'
  id='info1'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

To which the server would respond:

```
<iq type='result'
  from='xmpp.somevttn.net'
  to='ven1@xmpp.myco.net/client'
  id='info1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <feature var='http://openadr.org/openadr2'/>
    <feature var='http://jabber.org/protocol/disco'/>
  </query>
</iq>
```

This indicates, in an XMPP-compliant fashion that the XMPP server supports the protocol.

7.4.5.9.3 Discovery of service endpoints

Then VEN may then perform an 'items' query, with the 'node' set to the namespace 'http://openadr.org/openadr2#services' as such:

```
<iq type='get'
  from='ven1@xmpp.myco.net/client'
  to='xmpp.somevtn.net'
  id='items1'>
  <query xmlns='http://jabber.org/protocol/disco#items'
    node='http://openadr.org/openadr2#services' />
</iq>
```

The VTN shall then respond with the JIDs used for each service, for example:

```
<iq type='result'
  from='xmpp.somevtn.net'
  to='ven1@xmpp.myco.net/client'
  id='items1'>
  <query xmlns='http://jabber.org/protocol/disco#items'
    node='http://openadr.org/OpenADR2#services'>
    <item jid='event.openadr2.xmpp.somevtn.net'
      node='http://openadr.org/OpenADR2/EiEvent' />
    <item jid='feedback.openadr2.xmpp.somevtn.net'
      node='http://openadr.org/OpenADR2/EiFeedback' />
  </query>
</iq>
```

Note that all services could use the same JID or different JIDs and JIDs may be subdomains (which are common for XMPP services) or fully-qualified with a 'localpart' and resource.

7.4.5.9.4 Service discovery on VEN clients

VEN clients may choose to implement service discovery to provide supplemental information about their client (such as client IP address and available disk space). However, such supplemental information is outside the scope of this document. At the bare minimum, if a VEN supports XEP-0030 queries, it shall indicate support for the 'http://openadr.org/openadr2' feature, as described in 7.4.5.9.2.

7.4.6 Security considerations

Beyond authentication (as defined in 7.4.5.7), additional steps are required to secure an XMPP network. In particular, VEN clients shall not be allowed to communicate with each other (at least, not within the scope of this document). Clients can opt to implement some sort of whitelist to control access from different JIDs, which would be similar to implementing a firewall in an HTTP PUSH scenario.

However, ultimate responsibility for controlling access between XMPP clients shall be held at the XMPP server level. In most cases, the XMPP server will be logically deployed alongside, or as an integrated part of, a VTN. The XMPP server can then implement access controls to keep VENs logically isolated so they can only communicate with the VTN.

It is presumed that the XMPP server and the VTN (server) are either collocated in the same server or in the same secure network. Securing the communication between the XMPP server and the VTN is the responsibility of the VTN implementer. The XMPP server is a proxy of the VTN, so from the VEN's standpoint, there is no difference between whether it is communicating directly with the VTN or through the XMPP server.

8 Cyber security

8.1 General

Conformance to this document requires systems to implement security functionality as outlined in 8.2 to 8.7.

It is expected that many implementations will make use of existing cloud computing services and platforms. It was therefore decided to allow currently available security protocols and cipher suites until they are deprecated for the intended use. All certified products shall be upgradable. The mechanism of the upgradability is left to the manufacturers' implementation.

This document uses common security mechanisms without any modification.

The demand response program operator (e.g. utility) will need to require the appropriate level of security for its system.

The cyber security requirements conform to IEC 62351-11.

8.2 Architecture and certificate types

To provide security services such as authentication, confidentiality and integrity, VTNs and VENs shall use public key infrastructure (PKI) certificates. Two levels of security are defined for this document, called 'standard' and 'high'. The 'standard' security uses TLS for establishing secure channels between a VTN and a VEN for communication. 'High' security additionally uses XML signatures providing non-repudiation for documentation purposes (e.g. a signed event may be stored in a database for later documentation that an event has actually been received).

This document adopts an open architecture for security and will not restrict itself to some specific or proprietary technologies. There are primarily two public key cryptography algorithm options for using PKI certificates, namely RSA and ECC. While RSA is more widely accepted, ECC provides the benefits of more efficient cryptographic operations, such as encryption and digital signatures. For the same cryptographic strength, ECC key sizes are much shorter than RSA keys. This is especially important for embedded devices that favour efficient cryptography. As mentioned, RSA is more widely accepted on the Internet and has multiple certificate providers making it easier to select a provider that best matches the requirements.

To retain the benefits of both options and for the purposes of interoperability, the VTNs shall support both ECC and RSA certificates, each from a well-known certificate authority (CA) – for instance the one governed by the OpenADR Certificate Policy, unless otherwise determined by the DR program operator. The final design may have more than two providers but it will be ensured that at least one ECC and one RSA certificate authorities are included. A VEN can choose to use one or more PKI certificates on the device. The only restriction is that a VEN shall implement at least one certificate from the approved list of certificates and CAs for the VTN. This list shall be decided upon in advance and shall be publicly available. To establish a secure communication channel between the VTNs and VENs, VTNs shall have to support all the certificates in the approved list of certificates types and CAs. Upon initiating the communication, a VEN shall have the choice to use any one of them.

The following parameters shall be used for the certificates:

- ECC – 256 bits or longer keys;
- RSA – 2 048 bits or longer keys;
- Certificate types – X.509v3.

8.3 Certificate authorities

Implementers of demand response programs can choose between international, national, or local cyber security certificate providers. Options include the use of the publicly available OpenADR Alliance Certificate Policy and the OpenADR Alliance specific security certificates.

8.4 Certificate revocation

Please refer to the OpenADR 2.0 Certificate Policy for details.

8.5 TLS and cipher suites

Manufacturers may review NIST SP 800-131A for deprecation dates and current security requirements. This document requires TLS1.2 and the corresponding cipher suites.

Requirements (mandatory for interoperability):

- Transport layer security: TLS 1.2
- Cipher suites:
 - ECC – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
 - RSA – TLS_RSA_WITH_AES_128_CBC_SHA256

TLS clients shall use the signature_algorithm extension with the values {ECDSA, SHA 256; RSA, SHA256} to ensure application of SHA 256 for signature operation in the handshake and also for the used certificates. If this extension is not sent, the server is required by IETF RFC 5246 to fall back to default, which is SHA1.

Note that a VTN or VEN may be configured to support any TLS version and cipher suite combination based on the needs of a specific deployment. However, in the absence of changes to the default configuration of the VTN or VEN, the behaviour of the devices shall be as noted above.

Expired certificates shall not be used. To avoid service outage caused by expired certificates, deployed VTN and VEN implementations should have appropriate processes or warning mechanisms in place to identify pending security certificate expiration.

To comply with IEC TS 62351-4, a manufacturer and DR program implementer shall use AES 256 as a minimum.

8.6 System registration process

8.6.1 General

Registration is defined as the VTN becoming "aware" of the VEN, for example, by exchange of credential and possibly other information so that the VTN can properly authenticate the VEN. Note that this specifically does not include enrolling in DR programs. This document includes a registration service, although specific deployments can elect to do some portions of the registration process out-of-band.

8.6.2 Certificate fingerprints

VENs shall facilitate registration by providing a "certificate fingerprint", which can be easily transmitted out-of-band to the VTN. On the VTN, there should be one-to-one mapping between venID and certificate fingerprint. The fingerprint can then be used by the VTN to identify a VEN when it first connects to the VTN.

The certificate fingerprint will be generated as follows:

- 1) Perform a SHA-256 hash on the bytes of the DER-encoded client certificate.

- 2) Take the last 10 bytes (from the 32-byte SHA-256 hash), represented as pairs of hexadecimal digits, separated by the colon (ASCII 58) character.

This fingerprint can be generated by the common 'openssl' command line tool from a PEM certificate as follows:

```
$ openssl x509 -in client_cert.pem -fingerprint -sha256
SHA256Finger
print=D8:40:F6:2B:9D:6D:91:E4:21:21:64:2B:A5:55:76:GB:9C:6C:6B:00:9B:B5:5E:71:FA:
E4:61:75:9C:EF:A4:D6
```

The last 29 characters of the SHA-256 hash (5E:71:FA:E4:61:75:9C:EF:A4:D6) shall be used as the "certificate fingerprint." This fingerprint shall be printed or otherwise distributed with the VEN so it can be transmitted out-of-band to the VTN during installation time.

The fingerprint could also be computed in python as follows:

```
import ssl, hashlib
bin_cert = ssl.PEM_cert_to_DER_cert( open('client_cert.pem').read() )
sha_hash = hashlib.sha256(bin_cert).digest()
print ''.join( '%02X' % ord(c) for c in sha_hash[-10:] )
```

In some implementation cases, an alternative to certificate fingerprints may be used.

8.7 Implementing XML signatures for message payloads

8.7.1 XML signature

An XML signature uses public key cryptography to digitally sign portions of an XML document for integrity protection of the signed portions of the XML document. A brief description of XML signing and signature verification is provided here for implementation guidance. The complete XML signature specification is available at <http://www.w3.org/Signature/>.

A fundamental feature of the XML signature is the ability to sign individual portions of the XML tree in the document. This is important when the integrity of certain parts of the XML tree need to be preserved, while the complete document may still be appended by different entities as required. XML signature can be applied to more than one type of resource like sections of XML file, XML encoded data, binary-encoded images, text, or character-encoded files.

XML signatures are of three types:

- 1) enveloped, where the signed element being signed is a parent and the signature element is a child;
- 2) enveloping, where the signature element is the parent of the element being signed; and
- 3) detached, where the signature element and the signed element do not have a parent-child relationship. Detached signatures can reside in the same XML document as the content being signed, essentially living as sibling element, or they can be in a different document that the content being signed.

For the "high security" profile in this document, detached signatures with siblings are used.

8.7.2 Components of XML signatures

Reference URI: this element provides a reference to the resource that is being signed using a URI.

Digest value: this element contains the digest of the resource that is being signed.

Signature value: this element contains the resource digest signed by the private key of the signing authority.

Key info: the key info element contains the information of the key that should be used for signature verification. This contains information on the public key of the signing authority.

8.7.3 Creating XML signatures

The first step in creating XML signatures is to identify the resource that needs to be signed. For this document, 'detached' signatures living in the same XML document as the sibling element to the data object being signed shall be used. The VTN and VEN devices shall use the cipher suites that are used for TLS connections. As such, there will not be any need for supporting additional cipher suites for creating and verifying message digests. SHA-256 is mandatory for creating message digests and X.509 shall be used for public key certificates. Refer to IETF RFC 3275 for further details on how to create XML signatures.

An example XML signature is shown in Figure 20.

```
<oadr:oadrPayload>
  <ds:Signature>
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <ds:SignatureMethod
        Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <ds:Reference URI="#signedObject">
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256" />
        <ds:DigestValue>5H8v6B/z+YMDBR61xKdUZVvCLvUnC/rVfEcrZ5IAFDY=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#prop">
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256" />
        <ds:DigestValue>7H7U/}kg%yHHFD7jb^*jvR80KHffdr%7hHFVdEttGJU</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>UjBsR09EbGhjZ0dTQvXNQUFBUNBRU1tQ1p0dU1GUXhEUzhi</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:KeyName>key123</ds:KeyName>
    </ds:KeyInfo>
    <ds:Object Id="prop">
      <ds:SignatureProperties xmlns:dsp="http://openadr.org/oadr-2.0b/2012/07/xmldsig-properties">
        <ds:SignatureProperty>
          <dsp:ReplayProtect>
            <dsp:timestamp>2006-05-04T18:13:51.0</timestamp>
            <dsp:nonce>nonce0</nonce>
          </dsp:ReplayProtect>
        </ds:SignatureProperty>
      </ds:SignatureProperties>
    </ds:Object>
  </ds:Signature>
  <oadr:oadrSignedObject Id="signedObject">
    <oadr:oadrDistributeEvent ei:schemaVersion="2.0b">
      ...
    </oadr:oadrDistributeEvent>
  </oadr:oadrSignedObject>
</oadr:oadrPayload>
```

IEC

Figure 20 – XML signature example

The different fields of the XML signature are explained below. The <oadrPayload> is the root element for the XML document. It contains two children, <Signature> and <oadrSignedObject>. <Signature> contains the signature element, whereas <oadrSignedObject> contains the payload that is being signed.

The <Signature> element includes the entire XML signature. The <Reference> element contains the URI pointing to the <oadrSignedObject> element. <DigestMethod> points to the algorithm for creating the digest of the resource being signed. In the example, SHA-256 is used for creating the message digest. <DigestValue> is the alpha-numeric value of the digest.

The element <SignedInfo> contains all this information and the canonicalization method that was used. Canonicalization is important to have the same verification of XML signatures with different textual representations.

The <KeyInfo> element contains the information about the public key of the signing authority that should be used to verify the signature. It contains information including <SubjectName> and the corresponding X.509 certificate of the signing authority that contains the public key.

The <Object> element contains a list of <SignatureProperties>. An implementation supporting XML signatures shall at least insert the <ReplayProtect> element as <SignatureProperty>. The <ReplayProtect> element shall contain the dateTime when the payload is sent to the other party (not when it is created), as well as a random nonce. Note that the <ReplayProtect> element was part of an earlier version of the W3C "XML Signature Properties" document², but has been removed because of lack of interoperable implementations. This document uses the same element, but in its own namespace.

A digest of the whole <SignedInfo> element is created and signed using the private key of the signing authority.

The allowable algorithms for the signature method are: RSA-SHA256, and ECDSA-SHA256. The allowable algorithm for the digest method is: SHA256. Refer to conformance rule 514 in Clause 9 for details.

8.7.4 Verifying XML signatures

The following steps should be performed to verify an XML signature (refer to IETF RFC 3275):

- 1) Create a digest of the <SignedInfo> element using the digest mechanism indicated in the <DigestMethod> element (i.e. SHA-256). Verify the <SignatureValue> element using the key pointed out in the <KeyInfo> element. If the verification value is the same as the digest, the first verification step is complete.
- 2) Calculate the digest of the <obj:SignedObject> field and make sure the calculated digest is the same as the value in the <DigestValue> field.
- 3) Verify if a <ReplayProtect> element is contained as <SignatureProperty>. Reject the payload if the current date and time on the device differ from the value in the <ReplayProtect> more than a predefined value. In addition, the nonce may be used for further protection against replay attacks.

9 Conformance

9.1 Conformance statement

In order to claim conformance to this profile specification, a VTN, VEN or VTN/VEN combination shall conform to all statements made in this document. Product variations can be found in the product matrix in Table 1.

9.2 Conformance rules

9.2.1 EiEvent

Tables 7, 8, 9, 10, 11 and 12 list all applicable conformance rules. Note that the numbering of the conformance rules are not necessarily consecutive due to placeholders or deletions.

² <http://www.w3.org/2008/xmlsec/Drafts/xmlsig-properties/Overview.html>

Table 7 – Conformance rules

Conformance rule	Requirement
1	VEN/VTN, date-time and duration The date-time values shall be specified using ISO 8601 utc-time (also called zulu time). Example: 2013-04-22T15:26:44.123Z. Fractional seconds in date-time values are allowed by ISO 8601, however it is the responsibility of the receiving device to truncate the fractional seconds if necessary. Although ISO 8601 allows representation of zulu time with a zero hour offset (e.g. 2013-04-22T15:26:44.123+0000), it shall not be used. Representation of duration also follows ISO 8601. However, use of decimal values shall not be used in this document.
2	VTN, EiEvent service, oadrDistributeEvent payload The uid element is required for each eiEventSignal interval. Within a single oadrDistributeEvent eiEventSignal, uid shall be expressed as an interval number with a base of 0 and an increment of 1 for each subsequent interval.
3	VTN, EiEvent service, oadrDistributeEvent payload oadrDistributeEvent priority element – This is the priority of this event relative to other events. The lower the number, the higher the priority. A value of zero (0) indicates no priority, which is the lowest priority by default.
4	VTN, EiEvent service, oadrDistributeEvent payload A new event shall start with a modificationNumber of 0.
5	VTN, EiEvent service, oadrDistributeEvent payload Each modification of the oadrDistributeEvent eiEvent object, excluding createdDateTime, eventStatus, and currentValue, shall cause the modificationNumber to increment by 1. Exception: an eventStatus change to "cancelled" shall cause the modification number to increment by 1.
6	VEN, EiEvent service, oadrDistributeEvent payload The presence of any string except "false" in the oadrDistributeEvent testEvent element shall be treated as a trigger for a test event.
7 A profile only	VTN, EiEvent service, oadrDistributeEvent payload The oadrDistributeEvent eiEvent object shall contain only one event signal and that signal shall have a signalName of "SIMPLE" (in either uppercase or lowercase, i.e. "simple").
8	VTN, EiEvent service, oadrDistributeEvent payload oadrDistributeEvent eventSignal interval durations for a given event shall add up to eiEvent eiActivePeriod duration.
9	VTN, EiEvent service, oadrDistributeEvent payload oadrDistributeEvent eiEventSignal's with a signalName of "SIMPLE" shall use signalPayload values of 0 = normal; 1 = moderate; 2 = high; 3 = special.
10	VTN, EiEvent service, oadrDistributeEvent payload The VTN shall change the oadrDistributeEvent eventStatus to "cancelled" when communicating an event cancellation to the VEN. Note that the modificationNumber is incremented by 1 when issuing a cancellation.
12	VEN, EiEvent service, oadrCreatedEvent payload The VEN shall respond with an oadrCreatedEvent to an event in oadrDistributeEvent based upon the value in each event's oadrResponseRequired element as follows: Always – The VEN shall respond to the event with an oadrCreatedEvent eventResponse. This includes unchanged, new, changed, and cancelled events. Never – The VEN shall not respond to the event with an oadrCreatedEvent eventResponse. Note that oadrCreatedEvent event responses should be returned in one message, but may be returned in separate messages.
13	VTN, EiEvent service, oadrDistributeEvent payload EventStatus shall always transition from "far" to "near" to "active". The transition to "near" occurs at the start of the x-eiRampUp period if defined. If x-eiRampUp is not defined, the transition will move from "far" to "active" at the dtstart time.

Conformance rule	Requirement
14	VTN, EiEvent service, oadrDistributeEvent payload If currentValue is included in the payload, it shall be set to 0 (normal) when the event status is not "active" for the SIMPLE signalName
15	VTN, EiEvent service, oadrDistributeEvent payload A list of events sent by the VTN in an oadrDistributeEvent payload shall be ordered as follows: 1) "active" events have priority over pending events. 2) Within events with "active" event status, priority is determined by Priority in the Event Descriptor. 3) Between "active" events with the same priority, the one with the earlier start time has the higher priority. 4) Between pending (i.e. "far" or "near") events, the one with the earlier start time has the higher priority. 5) After processing rules 1) to 4), if Priority is still indeterminate within a set of events and if the VTN is responding to an oadrRequestEvent with replyLimit set to a value less than the number of pending or "active" events, the VTN shall maintain a fixed order between successive replies to oadrRequestEvent while the ordering remains indeterminate. 6) A cancellation should be ordered according to whatever rules were applied before the event was cancelled.
16	VTN, EiEvent service, oadrDistributeEvent payload The VTN shall recognize the state of the eiCreatedEvent optType element, both optIn and optOut, except when a B profile VEN responds to an event with oadrCreateOpt, in which case conformance rule 206 shall apply.
17	VTN, EiEvent service, oadrDistributeEvent payload The VTN shall recognize an asynchronous eiCreatedEvent optOut for a previously acknowledged event, except when a B profile VEN had previously responded to an event with oadrCreateOpt, in which case conformance rule 206 shall apply.
18	VEN/VTN, EiEvent service VTNs may send overlapping events. The behaviour of VENs when processing overlapping events is undefined. DR deployments using overlapping events should define the expected behaviour and should ensure that VENs participating in the program support the expected behaviour. The following rules shall be used when determining if events are overlapping: Event 1 = first event in a sequence; Event 2 = second event in a sequence; Start Time = dtStart value in event payload; Duration = overall event duration value in payload; Randomization Window = startafter value in payload; Effective End Time = Event Start Time plus Event Duration plus Randomization Window. If Event 2 Start Time is equal to or greater than Event 1 Effective End Time, then Event 2 is not considered to be overlapping. If Event 2 Start Time is less than Event 1 Effective End Time, then Event 2 is overlapping. When an event is cancelled, the Effective End Time for the event becomes the time the cancellation is acknowledged plus the Randomization Window for the purpose of calculating if a subsequent event is overlapping. If events are adjacent, where a subsequent event's Start Time is equal to the current event's Effective End Time, the VEN shall treat the end times as non-inclusive and the start times as inclusive for the purpose of determining which of the two concurrent event signal interval values to use at the instant of transition between the two events.
19	VEN, EiEvent service, oadrDistributeEvent payload If an oadrDistributeEvent payload has as mix of valid and invalid events, the VEN implementation shall only respond to the relevant valid events and not reject the entire message.
20	VTN, EiEvent service, oadrDistributeEvent payload At any time, a VTN may change any element or attribute of a pending (i.e. "far" or "near") or "active" event as long as it does not pertain to the past.

Conformance rule	Requirement
21	VEN/VTN If venID, vtnID, or eventID value is included in the payload, the receiving entity shall validate that the ID value is as expected and generate an error if an unexpected value is received. Exception: a VEN shall not generate an error upon receipt of a cancelled event whose eventID is not previously known.
22	VEN, EiEvent service, oadrDistributeEvent payload If no sub-elements are present in oadrDistributeEvent eiTarget, the presumption is that the recipient is the intended target of the event. If multiple criteria are present in eiTarget sub-elements, the values are OR'd together to determine whether the VEN is a target for the event. However, the VEN's behaviour with respect to responding to an event when it matches one of the eiTarget criteria is implementation-dependent.
23	VEN/VTN, EiEvent service, oadrRequestEvent payload oadrRequestEvent shall only be sent in the VEN to VTN direction.
25	VTN, EiEvent service, oadrCreatedEvent payload The following rules shall be followed with respect to application level responses for multiple events in an oadrCreatedEvent: 1) If the oadrCreatedEvent:eiResponse indicates failure, there is no need to examine each element in the eventResponses (eventResponses is not required to be part of the oadrCreatedEvent payload in an error condition, as specified in rule 35). 2) If the oadrCreatedEvent:eiResponse indicates success, the VTN should evaluate each eventResponse:responseCode to discover which optType state and response codes the VEN recorded for each event.
27	VTN, EiEvent service, oadrRequestEvent payload If a value is provided in the oadrRequestEvent replyLimit element, the VTN shall only return the number of events specified by the value in its oadrDistributeEvent response.
29	VTN, EiEvent service, oadrDistributeEvent payload The oadrDistributeEvent currentValue, if included in the payload, for each eiEventSignal shall accurately reflect the signalPayload value for the active interval in an executing event.
30	VEN, EiEvent service, oadrDistributeEvent payload The VEN shall randomize the dtstart time of the event if a value is present in the startafter element. The randomized dtstart time should be between the requested dtstart and dtstart plus startafter. Event completion times are determined by adding the event duration to the randomized dtstart time. Modifications to an event should maintain the same random offset, unless the startafter element itself is modified.
31	VEN, EiEvent service, oadrDistributeEvent payload The VEN shall recognize and act upon values specified in the subelements of activePeriod including: • dtstart, • duration, • tolerance, • x-eiRampUp (positive and negative), • x-eiRecovery (positive and negative). NOTE x-eiRampup and x-eiRecovery are not testable requirements.
32	VEN/VTN, EiEvent service, oadrDistributeEvent payload The VEN shall recognize and act upon values specified in the sub- elements of intervals including: • duration, • signalPayload.

Conformance rule	Requirement
33	VEN/VTN The implementation shall provide an application layer error indication as a result of the following conditions: • Missing expected information (459); • Payload not of expected type (459); • ID not as expected (452); • Illogical request – Old date on new event, durations do not add up correctly, etc. Note that a schema validation error will most likely cause a transport rather than application layer error indication.
35	VEN, EiEvent service, oadrCreatedEvent payload The eventResponses element in oadrCreatedEvent is required, except when an error condition is reported in eiResponse.
36	VEN, EiEvent service, oadrCreatedEvent payload An event cancellation received by the VEN shall be acknowledged with an oadrCreatedEvent with the optType element set as follows, unless the oadrResponseRequired is set to "never": optIn = confirm to cancellation; optOut = cannot cancel. Once an event cancellation is acknowledged by the VEN, the event shall not be included in subsequent oadrCreatedEvent payloads unless the VTN includes this event in a subsequent oadrDistributeEvent payload.
37	VEN A simple HTTP VEN implementation shall support the PULL model and may optionally also support push.
38	VTN A VTN shall support all transports and exchange models including HTTP PUSH, HTTP PULL, and XMPP PUSH.
40	VTN, EiEvent Sservice, oadrDistributeEvent payload The optional eiResponse object in oadrDistributeEvent shall be included when responding to oadrRequestEvent or to an oadrPoll.
41	VTN, EiEvent service, oadrDistributeEvent payload The VTN shall send a requestID value as part of the oadrDistributeEvent payload. The requestID value is not required to be unique, and in fact may be the same for all oadrDistributeEvent payloads. That there are two requestID fields in oadrDistributeEvent. The field that shall be populated with a requestID is located at oadrDistributeEvent:requestID.
42	VEN/VTN, EiEvent service, oadrCreatedEvent payload A VEN receiving an event in an oadrDistributeEvent payload shall use the received requestID value in the eiCreatedEvent:eventResponse when responding to the event with the oadrCreatedEvent payload. This includes any and all subsequent EiCreatedEvent messages that may be sent to change the opt status of the VEN. The eiResponse:requestID in oadrCreatedEvent shall be left empty if the payload contains eventResponses. The VTN shall look inside each eventResponse for the relevant requestID.
43	VEN, EiEvent service, oadrDistributeEvent payload The VEN shall not make any assumptions regarding the uniqueness of requestID values received from the VTN in the oadrDistributePayload.
45	VEN/VTN Messages sent between VENs and VTNs shall not include a schemaLocation attribute.
46	VEN/VTN Optional elements do not need to be included in outbound payloads, but if they are, the VEN or VTN receiving the payload shall understand and act upon those optional elements. Empty or omitted elements are considered equal unless a conformance rule explicitly states that element tags shall be excluded from the payload.

Conformance rule	Requirement
47	VEN/VTN, EiEvent service, oadrDistributeEvent payload An event with an overall duration of 0 indicates an event with no defined end time and will remain active until explicitly cancelled.
48	VEN/VTN When a VTN or VEN receives a schema-compliant payload that has logical errors, the receiving device shall provide an application layer error indication of 4xx in the eiResponse element of the payload. The detailed error message number is informational and not a requirement for response to a specific scenario, unless otherwise stated in the conformance rules. If the error is in an event contained in an oadrDistributeEvent payload and there are otherwise no application errors in the oadrDistributeEvent, the oadrCreatedResponse should set the eiResponse:responseCode to 200 and should report the status code and opt status for each event in the eventResponse element of oadrCreatedEvent. If no logical error is detected for an event, the status code is set to 200 and the opt status included in the oadrResponse element. The following logical errors shall be detected by implementations: • VEN receives non-matching market context (462); • VEN receives non-matching eiTarget (462); • VEN receives unsupported signalName (460). In the case of a logical error for an event, optType is set to optOut. Exception: on a deployment-specific basis, a VEN may be configured so that it accepts any marketContext. In this case, "any" valid URI will be considered matching with respect to this requirement. When the VTN receives an oadrCreatedEvent with logical errors in one of the eventResponses, it should set the eiResponse:responseCode in the oadrResponse reply to: • VTN receives non-matching eventID in at least one of the oadrResponses in the oadrCreatedEvent (452); • VTN receives mismatched modificationNumber in at least one of the oadrResponses in the oadrCreatedEvent (450). If there are multiple errors (e.g. one non-matching eventID and one non-matching modificationNumber), any 4xx error code (e.g. 452) can be used. When the VEN receives the oadrResponse, it could resend oadrCreatedEvents, one for each event instead of several grouped into one oadrCreatedEvent.
50	VTN, EiEvent service, oadrDistributeEvent payload In both the PUSH and PULL model, oadrDistributeEvent shall contain all existing events that have the eventStatus element set to either FAR, NEAR, or ACTIVE. Events with an eventStatus of "cancelled" shall be included in the payload upon change to the modificationNumber, but once the cancellation is acknowledged by the VEN, it shall not be included in subsequent payloads as stated in rule 36. VTN may continue to send events that the VEN has opted out of.
51	VEN/VTN PULL implementations may only use payloads where the transport layer request is in the VEN to VTN direction. PUSH implementations, both HTTP and XMPP, may use any payload in a valid exchange sequence except for oadrPoll.
52	VTN, EiEvent service, oadrDistributeEvent payload If a VTN requests acknowledgment of a cancelled event with oadrResponseRequired set to always, the VTN shall continue to send the cancelled event to the VEN until the event is acknowledged, eventStatus transitions to the complete state, or some well-defined number of retries is attempted.
53	VEN/VTN VTNs and VENs that support simple HTTP mode shall support the following HTTP headers: • Host (Only required in HTTP request); • Content-Length; • Content-Type of application/xml.
56	VEN, EiEvent service, oadrDistributeEvent payload If the VTN sends an oadrEvent with an eventID that the VEN is not aware of, then the VEN should process the event and add it to its list of known events.

Conformance rule	Requirement
57	VEN/VTN, EiEvent service, oadrDistributeEvent payload If the VTN sends an oadrEvent with an eventID that the VEN is already aware of, but with a higher modification number, then the VEN should replace the previous event with the new one in its list of known events.
58	VEN, EiEvent service, oadrDistributeEvent payload If the VTN sends an oadrEvent with an eventID that the VEN is already aware of, but which has a lower modification number than one in which the VEN is already aware, then this is an error and the VEN should respond with the appropriate error code (450). Note that this is true regardless of the event state, including "cancelled".
59	VEN, EiEvent service, oadrDistributeEvent payload If the VTN sends an oadrEvent object with the eventStatus set to "cancelled" and has an eventID that the VEN is aware of, then the VEN should cancel the existing event and delete it from its list of known events.
60	VEN, EiEvent service, oadrDistributeEvent, oadrCreatedEvent payload If the VTN sends an oadrEvent object with the eventStatus set to "cancelled" and has an eventID that the VEN is not aware of, then the VEN should ignore the event since it is not currently in its list of known events, but still shall respond with the createdEvent if required to do so by oadrResponseRequired.
61	VEN, EiEvent service, oadrDistributeEvent payload If the VTN sends the oadrDistributeEvent payload and it does not contain an event of which the VEN is aware (i.e. in its list of known events), then the VEN shall delete it from its list of known events (i.e. implied cancel). Exception: a VEN that has an active event that cannot be immediately stopped for operational reasons may leave the event in its data store until the event expires or the event can be stopped.
62	VEN, EiEvent service, oadrDistributeEvent, oadrCreatedEvent payload The VEN shall process every oadrEvent event message (new, modified, cancelled, etc.) that it receives from the VTN in an oadrDistributeEvent payload and it shall reply with a createdEvent message for every EiEvent message in which the responseRequired is set to "always". Furthermore, if the responseRequired is set to "never", the VEN shall not respond with a createdEvent message. It is at the complete discretion of the VTN as to whether responses are required from the VEN. Note that this rule is universal and applies to all scenarios including the following: - the event is one of which the VEN is already aware; - the event is being cancelled and the VEN did not even know it existed; - it does not matter how the EiEvent payloads were delivered, i.e. PUSH, PULL or as the result of being delivered in an ALL payload.
63	VTN The VTN shall not include more than one venID in the oadrDistributeEvent eiTarget.
64	VEN, EiEvent service A profile only A PULL VEN shall respond to all received events before initiating another polling cycle.
65	VEN, EiEvent service, oadrDistributeEvent, oadrCreatedEvent payload When an event containing a randomization value in the startafter element is cancelled, either explicitly or implicitly, the VEN shall randomize its termination of the event. The randomization window should be between 0 and a duration equal to the value specified in startafter.
66	VEN/VTN, EiEvent service, oadrDistributeEvent payload If a VTN sends an oadrDistributeEvent payload containing an event with a startafter element with a value greater than zero, the VTN shall continue to include the event in oadrDistributeEvent payloads, even if the event is complete, until current time is equal to dstart plus duration plus startafter. The receipt of an eventStatus equal to completed shall not cause the VEN to change its operational status with respect to executing the event.

Conformance rule	Requirement
67	VEN/VTN VTN and VEN shall support TLS 1.2. The default cipher suite selection shall be as follows: • The VEN client shall offer at least one of the default cipher suites listed below. • The VEN server shall support at least one of the default cipher suites listed below and shall select one of the default cipher suites regardless of other cipher suites that can be offered by the VTN client. • The VTN client shall offer both default cipher suites listed below. • The VTN server shall support both default cipher suites listed below and shall select one of listed the default cipher suites regardless of other ciphers that can be offered by the VEN client. Default cipher suites for TLS1.2: • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256; • TLS_RSA_WITH_AES_128_CBC_SHA256. Note that a VTN or VEN may be configured to support any TLS version and cipher suite combination based on the needs of a specific deployment. However, in the absence of changes to the default configuration of the VTN or VEN, the behaviour of the devices shall be as noted above.
68	VEN/VTN Both VTNs and VENs shall support client and server X.509v3 certificates. A VTN shall support both an ECC and RSA certificate. A VEN shall support either an RSA or ECC certificate and may support both. RSA certificates shall be signed with a minimum key length of 2 048 bits. ECC certificates shall be signed with a minimum key length of 224 bits. ECC hybrid certificates shall be signed with a 256-bit key signed with an RSA 2 048-bit key.
69	VTN/VEN, EiEvent service, oadrDistributeEvent payload The signalType element contained in a SIMPLE signal may be any of the supported enumerated values. The value used is informational and provides a hint as to the nature of the relative values specified in the signalPayload.
70	VTN, EiEvent service, IDs VTN shall ensure that venID is unique in the scope of VTN, namely, all VENs are assigned unique IDs. VTN shall ensure that eventID is unique in the scope of VEN, namely, while VTN can distribute events with the same eventID to multiple VENs, it cannot send two distinct events to the same VEN with the same eventID.

9.2.2 EiEvent – Additional 2.0b conformance rules

Table 8 – Additional conformance rules

Conformance rule	Requirement
100	VTN, EiEvent service, oadrDistributeEvent payload For both eiEventSignal and eiEventBaseline the following substitution group items shall be used: - streamPayloadBase = signalPayload; - payloadBase = payloadFloat. The number of signalPayload elements in each interval shall be equal to 1.
101	VTN, EiEvent service, oadrDistributeEvent payload The uid element is required for each eiEventBaseline interval. Within a single eiEventBaseline, uid shall be expressed as an interval number with a base of 0 and an increment of 1 for each subsequent interval.
102	VTN, EiEvent service, oadrDistributeEvent payload For both eiEventSignal and eiEventBaseline, the interval duration element shall appear in each interval and the sum of interval durations shall add up to overall duration element specified in eiActivePeriod:properties:duration for event signals and eiEventSignal:eiEventBaseline:duration for baselines.
103	VTN, EiEvent service, oadrDistributeEvent payload For both eiEventSignal and eiEventBaseline, the dtstart element shall not be included in the interval specification.
104	VTN, EiEvent service, oadrDistributeEvent payload If the signalName is one of the well-known alliance signals, then signalType, Units, and allowable values shall be as shown in Table 2.
105	VTN, EiEvent service, oadrDistributeEvent payload The eiNotification element shall be included as a sub-element of activePeriod.
106	VTN, EiEvent service, oadrDistributeEvent payload B profile oadrDistributeEvent eiEvents may contain the "SIMPLE" event signal in addition to other signals, however the relationship between the SIMPLE signal and other signals in the event are deployment-specific.
107	VTN, EiEvent service, oadrDistributeEvent payload If multiple signals are present for an event, each signal shall have a signalID value that is unique within the scope of the event.
108	VTN, EiEvent service, oadrDistributeEvent payload If one or more resourceIDs are present as sub-elements of eiEventBaseline, the baseline represents the aggregate of the listed resources. No assumptions should be made regarding the relationship between the eiEvent:eiTarget object and the source of the baseline data.
109	VEN, EiEvent service, oadrDistributeEvent payload A VEN shall generate an oadrCreatedEvent application level 460 error if any requested signal in an event cannot be supported. Supported means that the requested combination of signalType and Unit cannot be currently supported.
110	VTN, EiEvent service, oadrDistributeEvent payload The value of currentValue when an event is not active is undefined for event signals other than SIMPLE. See conformance rule 14 for the currentValue requirements for the SIMPLE signal.

111	VTN/VEN, EiEvent service, oadrDistributeEvent payload eiTarget is defined at both the event and signal levels of the oadrDistributeEvent payload. At the signal level, only the endDeviceAsset may be used as an eiTarget sub-element. The allowable values for the eiEventSignal:eiTarget:endDeviceAsset:mrid are as follows: • Thermostat, • Strip_Heater, • Baseboard_Heater, • Water_Heater, • Pool_Pump, • Sauna, • Hot_tub, • Smart_Appliance, • Irrigation_Pump, • Managed_Commercial_and_Industrial_Loads, • Simple_Residential_On_Off_Loads, • Exterior_Lighting, • Interior_Lighting, • Electric_Vehicle, • Generation_Systems, • Load_Control_Switch, • Smart_Inverter, • EVSE, • RESU, • Energy_Management_System, • Smart_Energy_Module, • Storage, • x-{user Defined}. If more that one endDeviceAsset:mrid value is present at the signal level, the values are OR'd together to determine a match with the VEN's resources. If present, eiEventSignal:eiTarget:endDeviceAsset:mrid values are AND'd with any event level eiTarget values to determine the intended target for the specific signal. If no eiTarget:endDeviceAsset sub-element is present in the eiTarget, the eiEventSignal:eiTarget element shall be omitted from the payload.
112	VEN, EiEvent service, oadrDistributeEvent payload If a VEN receives an event where the nature of the signal type causes the VEN not to be able to resolve the intended resource to apply the event to, the VEN shall return a 469 error response to the VTN.
113	VTN, EiEvent service, oadrDistributeEvent payload The event currentValue element is optional for the B service, but shall be included in payloads sent to A Profile VENs.
114	VEN, EiEvent service, oadrDistribute payload A VEN shall have reasonably synchronized clocks with the VTN. It is up to the deployment to define the acceptable skew. It is out of scope of this document on how to synchronize clocks; mechanisms such as NTP may be used. When a 2.0b VEN receives an event, it may ignore the eventStatus contained in the event (with the exception of "cancelled") and calculate the eventStatus based on the current time, event start time, interval duration, etc.
115	VTN, EiEvent service, oadrDistribute payload createdDateTime shall be recreated each time any element of the event payload is changed. The eventStatus shall be updated before sending out the event when a VTN receives an oadrRequestEvent. It may update the status when it receives an oadrPoll, but this is not required, as the 2.0b VEN can calculate the eventStatus automatically. When oadrDistributeEvent payload contains multiple events, all createdDateTime found in the payload shall be identical.
116	VTN/VEN, EiEvent service, oadrDistributeEvent payload The signalType element contained in a SIMPLE signal shall be "level" for the 2.0b profile implementations. Refer to rule 69 for 2.0a profile VENs.

9.2.3 EiOpt

Table 9 – EiOpt conformance rules

200	VEN/VTN, EiOpt service, oadrCreateOpt payload oadrCreateOpt payload vavailability properties shall not include the following elements: • tolerance, • eiRampUp, • eiRecovery, • eiNotification.
201	VEN/VTN, EiOpt service, oadrCreateOpt payload The following aspects of an opt schedule are program-specific: • the type of schedules that a VEN is allowed to issue: optIn, optOut, or both; • the default opt state (optIn or optOut) for time periods not defined by either an opt or Avail schedule (if defined); • whether optIn or optOut schedules have precedence in the situation where they have overlapping time frames; • whether overlapping time frames are allowed within the same schedule (optIn or optOut) and if they are not allowed, the expected exception handling behaviour; • whether the VTN is allowed to send events during an optOut period.
202	VEN, EiOpt service, oadrCreateOpt payload If the oadrCreateOpt eiTarget is empty, then the opt schedule applies to all the resources associated with that VEN. If eiTarget sub-elements are defined, these sub-elements shall be OR'd together to define a subset of the VEN's resources that the opt schedule shall apply. Note that the VEN may send multiple opt schedule payloads for different sets of resources identified by the eiTarget element.
203	VTN, EiOpt service, oadrCreateOpt payload A new opt schedule (optIn or optOut) sent via oadrCreateOpt shall be handled by the VTN as follows when a previously sent opt schedule is still active: • if only the venID is specified in eiTarget, previous opt schedules with respect to future availability are replaced in their entirety by the new opt schedule for all resources associated with the VEN; • if sub-elements are specified in eiTarget in addition to the venID, then an opt schedule shall be generated for these resources superseding any previously defined schedules for specific resources; • previously defined opt schedules for resources that are not specified by eiTarget shall remain unchanged. Note that if both optIn and optOut schedules exist, they are treated independently with respect to this rule.
204	VEN, EiOpt service, oadrCreateOpt payload The VEN shall send an optID value as part of the oadrCreateOpt payload, however the optID value is not required to be unique, and in fact may be the same for all oadrCreateOpt payloads. Note that if the same optID is used defining opt schedules for multiple sets of resources, a cancellation referencing the optID shall cancel the opt schedule for all the resources.
205	VTN, EiOpt service, oadrCreateOpt payload If marketContext is present in the oadrCreateOpt payload, the opt schedule shall only apply to the VEN's availability with respect to events generated within the specified marketContext.

206	VTN/VEN, EiOpt service, oadrCreateOpt payload A VEN responds with an EiEvent oadrCreatedEvent payload each time it receives an oadrDistributeEvent payload if oadrResponseRequired is set to "always". However, further qualifications of the optType state for the event can be accomplished by sending either an oadrCreatedEvent or an EiOpt oadrCreateOpt payload with the eventID specified, provided that the oadrResponseRequired element was set to "always" when the event was last received from the VTN. Once the optType state for a certain resource is qualified with oadrCreateOpt, subsequent changes to the optType state for the specific resource can only be made by another oadrCreateOpt payload. optType states for resources not specifically targeted in the oadrCreateOpt payload would still be controlled by subsequent oadrCreateEvent optType changes. Modifications to an event do not change this rule. Note that oadrCreateOpt may not be sent to qualify an event if oadrResponseRequired is set to "never" in the most recently received oadrDistributeEvent. If oadrCreateOpt is used to modify the opt state of a pending or active event, the following rules shall apply with regards to the eiTarget and oadrDeviceClass targeting elements: • if no target is present in the oadrCreateOpt:eiTarget or if the only target is the venID, then all resources associated with the qualifiedEvent shall have its opt state altered; • if the oadrCreateOpt:eiTarget further qualifies any resources of the VEN for the given qualifiedEvent (in addition to the implicitly or explicitly targeted VEN), the included resources shall be a subset of the resources targeted in the qualifiedEvent, and the opt state will only be altered for the targeted subset; • if the VEN includes one or more targets in the oadrCreateOpt:eiTarget payload that are not part of the qualifiedEvent, the VTN shall generate a 4xx error in its oadrCreatedOpt:eiResponse; • the VTN may respond with a 4xx error in the oadrCreatedOpt response if it does not accept the requested opt state changes for the specified target resources. Note that the VEN may send multiple oadrCreateOpt payloads with eventID specified for different sets of resources identified by the eiTarget element. When eventID is specified in oadrCreateOpt, the VEN shall not include vavailability or marketContext in the payload. When eventID is not specified in oadrCreateOpt, the VEN shall include a vavailability and may include a marketContext in the payload. Note that oadrCreateOpt with an eventID specified may also include oadrDeviceClass to further qualify the resources targeted. Resolution of the targeted resources should follow conformance rule 209.
207	VTN/VEN, EiOpt service, oadrCreateOpt payload Although requestID is a mandatory payload element for oadrCreateOpt and oadrCancelOpt, it may be left as an empty string. However, if a value is specified for requestID, the VTN shall return that value in its oadrCreatedOpt or oadrCanceledOpt payload.
208	VEN, EiOpt service, IDs The following IDs shall be specified by the VEN and should be validated by the VTN receiving the payload. If the ID does not match the expected ID, the device should include a 452 error in eiResponse: • venID, • optID.
209	VEN/VTN, EiOpt service, oadrCreateOpt payload The oadrCreateOpt:oadrDeviceClass element, if present in a payload, shall contain only the endDeviceAsset sub-element whose values shall conform to the list of device classes shown in conformance rule 111. If no oadrCreateOpt:oadrDeviceClass:endDeviceAsset sub-element is present in the oadrCreateOpt, the oadrCreateOpt:oadrDeviceClass element shall be omitted from the payload. The device classes specified in oadrDeviceClass should be AND'd with the resource targets defined by oadrCreateOpt:eiTarget and marketContext (rule 205), to determine the resources whose availability is impacted by this opt schedule.
210	VEN/VTN, oadrCreateOpt payload If an opt availability schedule uses a duration of zero, it shall be treated as an open-ended opt state.
211	VEN/VTN, oadrCancelOpt A VEN shall not use oadrCancelOpt with an optID matching that of a oadrCreateOpt used to qualify an active or pending event. An opt cancellation in this context would be illogical. VTN shall not generate an error if it receives a cancellation for an opt schedule that has been superseded by a more recent opt schedule.

9.2.4 EiReport

Table 10 – EiReport conformance rules

300	VEN/VTN, EiReport service An alliance report profile is a subset of all the possible oadrReport object schema elements that define the characteristics (data points) of a specific type of report (history and telemetry usage, telemetry status). The metadata report defines the characteristics that a particular implementation is capable of reporting. This shall be a subset of the characteristics defined by a well-known alliance report profile if the reportName matches that defined by a well-known report. A report request (oadrReportRequest object) defines the specific characteristics that an implementation would like to receive in a report. This list of characteristics shall be a subset of the characteristics defined in the metadata report. A non-metadata report (oadrReport object) shall contain values for all the characteristics requested in the report request.
301	VEN/VTN, EiReport service Implementations shall at a minimum support the following payload interactions even if the implementation does not have any reportable data: • send oadrRegisterReport payload on power up (after initial registration), reset, or reregistration. For push implementations oadrRegisterReport would be returned by the VTN in response to an initial oadrPoll payload; • send oadrCreatedReport payload in response to an oadrCreateReport payload requesting a Metadata report; • send oadrRegisterReport payload containing a metadata report should an oadrCreateReport payload request Metadata reports. In each case where the implementation does not have any reportable data, the oadrReport object shall be omitted from the oadrRegisterReport payload noted above.
302	VEN/VTN, EiReport service All reports are the aggregate of the resources defined in the metadata report element reportDataSource, filtered by the optional inclusion of a marketContext in the Metadata report characteristics. If multiple target types are specified in reportDataSource, the resources targeted by these types are OR'd together.
303	VEN/VTN, EiReport service Although requestID element is a mandatory payload element for request payloads, it may be left as an empty string. However, if a value is specified for requestID in the request payload, the responding implementation shall return that value in its response payload.
304	VEN/VTN, EiReport service, IDs The following IDs should be validated by the implementations receiving the payload. If the ID does not match the expected ID, the device should include a 452 error in the eiResponse payload: • venID, • vtnID, • rID, • reportRequestID, • reportSpecifierID.
305	VEN/VTN, EiReport service, metadata oadrReport object Payloads containing a metadata report that describes reporting capabilities shall include the following optional payload elements: • reportDescription, • reportDescription:itemBase substitution group (Units) as indicated in conformance rule 331. For payloads containing a metadata report that depicts that no reporting capabilities are supported, the oadrReport object shall be omitted. Payloads containing a metadata report shall not include the following optional payload elements: – intervals, – dtstart, – ei.

306	VEN/VTN, EiReport service, metadata oadrReport object An implementation (VEN or VTN) sending a payload containing a metadata report shall supply all its reporting capabilities. An implementation receiving a metadata report shall use this report to replace any previously received Metadata reporting capabilities. It shall also implicitly cancel all previously scheduled reports, except for periodic requests for Metadata reports, which shall be explicitly cancelled. If implementations receiving a metadata report wish to continue to receive a previously requested report(s), it shall send a new report request after the receipt of the metadata report.
307	VEN/VTN, EiReport service, metadata oadrReport object The metadata oadrReportDescription:reportSubject element, if present in a payload, shall contain only the endDeviceAsset sub-element whose values shall conform to the list of device classes shown in conformance rule 111. If no oadrReportDescription:reportSubject:endDeviceAsset sub-element is present in the oadrReportDescription, the oadrReportDescription:reportSubject element shall be omitted from the payload. The device classes specified in reportSubject should be AND'd with the resource targets defined by reportDataSource and marketContext (rule 302), to determine the specific target whose data will be aggregated for the report.
308	VEN/VTN, EiReport service, metadata oadrReport object All reportSpecifierID element values shall be unique within a specific metadata report. If a reportDescription in the metadata report has a reportName element that contains one of the well-known alliance report profile names, the data points' characteristics and allowable values shall conform to the table shown in conformance rule 331. All rID element values shall be unique within a specific reportSpecifierID. When requesting a metadata report by sending an oadrCreateReport, rID shall be set to 0. The receiving party of the oadrCreateReport requesting a metadata report, shall ignore the rID value.
309	VEN/VTN, EiReport, metadata report – Power-up and reset Upon power-up or after a system reset (followed by an initial registration), PUSH and PULL implementations of both VTN and VEN shall send an oadrRegisterReport payload (i.e. metadata reports) to the other party prior to initiating other non-registration service operations except an oadrRequestEvent payload, which may be sent prior to the exchange of metadata reports.
311	VEN/VTN, EiReport service, metadata oadrReport Object The oadrRegisterReport.oadrReport.reportRequestID element in each contained metadata report shall be set to zero. If the oadrRegisterReport payload is being sent to as the result of a oadrCreateReport request, the optional oadrRegisterReport.reportRequestID element shall be present with a value that matches the reportRequestID used when the request was made. Otherwise, oadrRegisterReport.reportRequestID shall not be included in the oadrRegisterReport payload.
312	VEN/VTN, EiReport service, metadata oadrReport object The oadrReport:duration element communicates the amount of data that is available for reporting. For telemetry reporting, this may reflect the amount of data that can be buffered by the VEN. For reports, the amount of available data is more likely bounded by off-device storage. This element shall be included in metadata reports. Since duration is a time parameter, the assumption is that the implementation can store enough data at the fastest supported sampling rate (oadrSamplingRate:oadrMinPeriod) for that duration of time. The oadrSamplingRate:oadrMaxPeriod shall not be larger than the duration value noted above.
313	VEN/VTN, EiReport service, non-metadata oadrReport object Payloads containing a non-metadata report shall include the following optional payload elements: • dtstart, • Intervals, • reportName – the same value as used in the metadata report from which the report is derived, except for without the METADATA prefix. Payloads containing a non-metadata report shall not include the following optional payload element: • oadrReportDescription.

314	VEN/VTN, EiReport service, non-metadata oadrReport object oadrReport:dtstart element is the start time of the overall report and is inherited by the first interval in the report. If both the oadrReport:dtstart element and interval dtstart element values are present in the payload, then oadrReport:dtstart element shall equal the dtstart element value of the first interval. Note that oadrReport:dtstart shall be included in all oadrUpdateReport payloads. If one interval (oadrReport:intervals:interval) contains a dtstart element, all intervals shall contain a dtstart element. Each interval shall have an effective dtstart time, either inherited from oadrReport:dtstart or specified as part of the interval dtstart element. The sequence of interval (oadrReport:intervals:interval) available for reporting shall be viewed as a continuum of values from the past to the future. The first interval in a sequence shall always be the one closest to Epoch (1/1/1970), with subsequent intervals moving towards future relative to the time stamp of the first interval. For instance, an interval with a dtstart of 17:00 and a granularity of 10 min would cover the period from 17:00 to 17:10, followed by another interval with a dtstart 17:10 covering the period from 17:10 to 17:20.
315	VEN/VTN, EiReport service, non-metadata oadrReport object The oadrReport:duration element is the duration of the entire report and, if included, shall reflect the time period beginning of the first interval and the end of the last interval included in the report. If oadrReport:intervals:interval:duration element is absent from the payload, then the interval data contains data points that are bound to a specific point in time as opposed to a span of time. When the oadrReport:intervals:interval:duration element is present, the reported dtstart time shall represent the beginning of the data collection period for a given data point. For instance, a data sample with a dtstart of 17:00 and a granularity of 10 min would cover the period from 17:00 to 17:10. If one interval (oadrReport:intervals:interval) contains a duration element, all intervals shall contain a duration element. If intervals do not contain dtstart elements and there is more than one interval, duration shall be specified in oadrReport:intervals:interval:duration.
316	VEN/VTN, EiReport service, non-metadata oadrReport object The uid element is required in each interval if dtstart is not part of the interval specification. Within a single oadrReport object, when required, uid shall be expressed as an interval number with a base of 0 and an increment of 1 for each subsequent interval. Exception: green button reports are not required to have a uid element.
317	VEN/VTN, EiReport service, non-metadata oadrReportRequest object The reportSpecifier:granularity element value specified in the report request defines the requested interval reporting frequency. Example: telemetry report with a granularity of 10 min with a reportBackDuration of 60 min would result in a report with 6 intervals. The number of intervals in a report may not exactly match the relationship between granularity and reportBackDuration, although in general it should be within plus or minus one interval of this calculated value. The requested granularity shall not be less than oadrSamplingRate:oadrMinPeriod, if specified in the metadata report. At the same time, granularity shall not be more than reportBackDuration. If reportSpecifier:granularity is zero, then the requested data should be included in a report only when it changes from the previous value and not at regular intervals, while reports are to be sent at regular intervals defined by reportBackDuration. In no case will the duration between subsequent values in a report be less than oadrSamplingRate:oadrMinPeriod, even if they are being reported only on change. If the requested data has not changed since the last time a report has been sent out, the oadrDataQualityType shall be set to "no new value – previous value used". If reportBackDuration is set to zero in the same report request, regardless of whether granularity is zero or not, the expected behaviour is the one defined in conformance rule 324. For all history reports (HISTORY_XXX), granularity shall be set to 0 to signify that the data should be reported at whatever granularity it was recorded.
318	VEN/VTN, EiReport service, non-metadata oadrReport object If an implementation has no valid data for a data point that was included in its metadata report and it has been asked to report this value via a report request, the implementation shall include a placeholder value in the report and set the intervals:interval:oadrReportPayload:dataQuality element to one of the "quality bad" enumerated values.
319	VEN/VTN, EiReport service, non-metadata oadrReport object If included in a payload, the intervals:interval:oadrReportPayload:accuracy element shall be in same units as the payloadFloat value for the Interval. When present with Confidence, indicates the likely variability of the prediction. This rule is not applicable if the payloadResourceStatus is used to communicate the payload value.

321	VEN/VTN, EiReport service, oadrReportRequest object Payloads containing the oadrReportRequest object shall not include the following optional element: • reportSpecifier:reportInterval:properties:tolerance; • reportSpecifier:reportInterval:properties:eiNotification; • reportSpecifier:reportInterval:properties:eiReampUp; • reportSpecifier:reportInterval:properties:eiRecovery; • reportSpecifier:specifierPayload:itembase.
322	VEN/VTN, EiReport service, oadrReportRequest object The reportRequestID element value shall be unique within the scope of the VEN/VTN.
324	VEN/VTN, EiReport service, oadrReportRequest object The reportSpecifier:reportBackDuration element specifies how frequently the report should be sent. If reportBackDuration is 0, then the report shall be sent immediately and only once using the oadrUpdateReport or oadrRegisterReport payloads. History reports shall set reportBackDuration to 0. For telemetry reports with a reportBackDuration of 0, the granularity value shall be ignored and only one interval of data for each data point shall be returned in the report. If the reportSpecifier:reportBackDuration element is non-zero, the report shall be delivered periodically using the oadrUpdateReport or oadrRegisterReport payloads. Both metadata and non-metadata reports may be delivered periodically. The number of reports delivered over the lifetime of a periodic report may not exactly match the relationship between the requested overall reporting duration and reportBackDuration. HISTORY_USAGE and periodic TELEMETRY_USAGE and TELEMETRY_STATUS reports shall specify values for reportSpecifier:reportInterval dtstart and duration sub-elements. One-off telemetry reports with point data do not require these elements and they shall be ignored if included in the report request payload.
325	VEN/VTN, EiReport service, oadrReportRequest object For TELEMETRY_XXX reports, the reportSpecifier:reportInterval:properties:dtstart element value indicates when the first in a series of one or more reports shall begin. The reportSpecifier:reportInterval:properties:duration element value indicates the time span from the dtstart time that this report should cover. For all ongoing telemetry reports (TELEMETRY_XXX), this duration may be in the future and once this time span has expired the report should no longer be generated. If reportSpecifier:reportInterval:properties:duration value is 0, then the report shall be generated indefinitely. For all history reports (HISTORY_XXX), this duration is in the past and reflects the historical data requested. If this value is 0, then the report shall include all the history from dtstart time.
327	VEN/VTN, EiReport service, oadrReportRequest object Payloads containing the oadrReportRequest object shall use the well-known string "METADATA" as the reportSpecifierID element value to request a Metadata report.
328	VEN/VTN, EiReport service, oadrUpdatedReport payload VTN or VEN may include an oadrCancelReport object in an oadrUpdatedReport payload to enable the implementation receiving the report to cancel a report as part of its response. When receiving such a payload, VTN or VEN shall process it as defined in 6.2.2.4. The reportRequestIDs that appear in the oadrCancelReport object shall be a subset of the reportRequestIDs from the oadrUpdateReport that is being responded to.
329	VEN/VTN, EiReport, oadrCanceledReport and oadrCreatedReport oadrPendingReports element shall contain a list of reportRequestID element values that includes all reports that are scheduled for future delivery.
330	VEN/VTN, EiReport, oadrReport The rID element used in referencing data points is defined as a string in the schema. The Energy Interoperation schema restricts the use of rID to three numeric digits. Implementers not concerned with using descriptive rID values may wish to stay within the bounds of the Energy Interoperation restrictions.

331	VEN/VTN, EiReport, oadrReport This document has defined a number of well-known reports. These reports are identified by the well-known names shown under the reportName column in the table below. When the well-known report names are used as part of the oadrRegisterReport, they shall be prefixed with "METADATA_" and the values shown in the table shall be used for the specified report. <table><tr><th>reportName</th><th>reportType</th><th>Units</th><th>readingType</th><th>streamPayloadBase payloadBase</th></tr><tr><td>TELEMETRY_USAGE</td><td>usage</td><td>powerReal OR energyReal OR pulseCount</td><td>Direct Read</td><td>oadrReportPayload payloadFloat</td></tr><tr><td>TELEMETRY_STATUS</td><td>x-resourceStatus</td><td>None</td><td>x-notApplicable</td><td>oadrReportPayload oadrPayloadResourceStatus</td></tr><tr><td>HISTORY_USAGE</td><td>usage</td><td>powerReal OR energyReal OR pulseCount</td><td>Direct Read</td><td>oadrReportPayload payloadFloat</td></tr></table> Users may define their own custom reports by defining a reportName that is unique to their deployment and by using any of the schema-supported values to define the report.	reportName	reportType	Units	readingType	streamPayloadBase payloadBase	TELEMETRY_USAGE	usage	powerReal OR energyReal OR pulseCount	Direct Read	oadrReportPayload payloadFloat	TELEMETRY_STATUS	x-resourceStatus	None	x-notApplicable	oadrReportPayload oadrPayloadResourceStatus	HISTORY_USAGE	usage	powerReal OR energyReal OR pulseCount	Direct Read	oadrReportPayload payloadFloat
reportName	reportType	Units	readingType	streamPayloadBase payloadBase																	
TELEMETRY_USAGE	usage	powerReal OR energyReal OR pulseCount	Direct Read	oadrReportPayload payloadFloat																	
TELEMETRY_STATUS	x-resourceStatus	None	x-notApplicable	oadrReportPayload oadrPayloadResourceStatus																	
HISTORY_USAGE	usage	powerReal OR energyReal OR pulseCount	Direct Read	oadrReportPayload payloadFloat																	
333	VEN, EiReport service, Metadata oadrReport object The reportDescription:oadrSamplingRate element is required for any of the telemetry reports (TELEMETRY_XXX).																				
334	VEN, EiReport service, TELEMETRY_STATUS reports TELEMETRY_STATUS reports do not explicitly list a Unit in their metadata specification. VENS should include each of the following optional oadrPayloadResourceStatus sub-elements in TELEMETRY_STATUS reports if appropriate for the targeted resource: - oadrLoadControlState:oadrCapacity, - oadrLoadControlState:oadrLevelOffset, - oadrLoadControlState:oadrPercentOffset, - oadrLoadControlState:oadrSetPoint. All intervals shall have the same set of oadrLoadControlState child elements in a given report. The LoadControlState attributes are meant to mirror the LOAD_CONTROL signal attributes meaning that for each LOAD_CONTROL signal type, there is a corresponding attribute in the LoadControlState of a report. Their intended use is to enable a VEN to report on its load control state that corresponds to a LOAD_CONTROL signal that might be sent to a VEN. Such reports can be used to signify the VEN's current LOAD_CONTROL state and might be used by the VTN to determine how sending a LOAD_CONTROL signal to the VEN will affect the VEN. It can also be used by the VTN as a means to verify if previously sent LOAD_CONTROL signals have been effecting desired changes in the VEN. This is analogous to the correspondence between LOAD_DISPATCH signals and telemetry usage reports from VEN.																				
335	VEN/VTN, EiReport service, oadrUpdatedReport A report cancellation may be included in the oadrUpdatedReport response. If the other party continues to send reports after this cancellation, the party requesting the cancellation shall use oadrCancelReport to rerequest the cancellation.																				
336	VEN/VTN, EiReport service, oadrCanceledReport An oadrCancelReport may include multiple reportRequestID values. If the receiving party cannot successfully cancel all the reports listed or does not recognize one of the included reportRequestID values, it shall return a 4xx error in the responseCode element of oadrCanceledReport.																				
337	VEN/VTN, EiReport service If an implementation offers a TELEMETRY_USAGE report that contains interval data points (representing a value accumulated over time), the implementation shall sample this data periodically such that it can respond to a one-shot report request immediately with the most recent sample it has in its buffer. If the metadata report for the telemetry report offered a range of sampling frequencies, the report shall include the interval duration in the one-shot interval report.																				
338	VEN/VTN, EiReport service, oadrReportRequest object The reportSpecifier:specifierPayload:readingType values shall be set to x-notApplicable. The readingType element is required by the EI schema in this location, but is not functionally used by OpenADR.																				
339	VEN/VTN, EiReport service eiReportID is a unique identifier for a specific instance of a report. It is not used by OpenADR and should be ignored by VTN/VENS.																				

340	VEN/VTN, EiReport service A source party should not request more data than is offered. Example: if a target party indicates in oadrReport:duration that it maintains 1 day of data, a source party should not request 2 days of data. If a source party requests more data from the target party in an oadrReportRequest (e.g. based on the requested duration), the target party shall return an application error code 454 in an oadrCreatedReport:eiResponse.
341	VEN/VTN, EiReport service oadrCreateReport:reportInterval:dtstart shall reflect the report interval in the desired set of data that is closest to Epoch time. The dtstart time of the first interval (oadrReport:intervals:interval:dtstart) in the corresponding oadrUpdateReport payload may not exactly match the requested dtstart time (oadrCreateReport:reportInterval:dtstart), although in general the offset should not be greater than the value of granularity. In the case of a periodic report, it is possible that a final report can be delivered slightly beyond the effective end time of the requested reporting period (i.e. the sum of overall reporting dtstart and duration).
342	VEN/VTN, EiReport service The following combinations of dtstart and duration in an oadrUpdateReport intervals shall not be used (otherwise an implementation would have to derive sample time for point data or the time component of interval data): 1) interval data with just dtstart values, 2) point data with just duration values, 3) point data with dtstart and duration values. If intervals contain both point data and interval data, both dtstart and duration can be specified.
343	VEN/VTN, EiReport service If reportDataSource, reportSubject, and marketContext are omitted from the metadata report, the source of the report data is all resources associated with the VTN or VEN sending the metadata report.
344	VEN/VTN EiReportservice, oadrRegisteredReport payload VTN or VEN may include oadrReportRequest object in oadrRegisteredReport payload (Piggyback Report Request). VEN and VTNs are not required to implement sending Piggy-back Report Requests, but they shall be able to receive such a request and act upon it.
345	VEN/VTN EiReportservice, oadrCancelReport If reportToFollow is set to true by the report requester in oadrCancelReport object cancelling periodic reports, the report sender shall send one final additional report to the report requester.
346	VEN/VTN, EiReportservice, oadrUpdateReport VEN should send oadrUpdateReport payloads in a predictable sequence and with a consistent number of intervals per periodic report under normal operations. However, the VEN may send oadrUpdateReport payloads as needed (at a shorter interval than one specified by reportBackDuration) to provide corrected data (e.g. when previous value sent was of bad quality) or missing data (e.g. when measurement was delayed and thereby was not included in the previous report) to the VTN, provided that the number of intervals in the corrected data match the original report request and that all the data contained in the report replace any previously sent report values. It is not a protocol violation for the same report to be sent more than once.

9.2.5 EiRegisterParty

Table 11 – EiRegisterParty conformance rules

400	VTN/VEN, EiRegisterParty service, IDs The following IDs should be validated by the implementations receiving the payload. If the ID does not match the expected ID, the device should include a 452 error in eiResponse: • venID, • vtnID, • registrationID.
401	VEN/VTN, EiRegisterParty, oadrCreatePartyRegistration oadrQueryRegistration and oadrCreatePartyRegistration shall only be used in the VEN to VTN direction. oadrRequestReregistration shall only be used in the VTN to VEN direction. oadrCancelPartyRegistration may be used in either the VEN to VTN or VTN to VEN direction. A VEN whose registration has been cancelled may attempt to initiate a new registration at a deployment-specific interval. An unregistered VEN that is powered up or reset shall attempt to initiate a registration if appropriately configured to communicate with the VTN.
402	VEN/VTN, EiRegisterParty, oadrCreatePartyRegistration oadrCreatePartyRegistration shall include the following payload elements: • if the oadrTransportName is simpleHTTP, then oadrHttpPullModel shall be set to true or false; • if oadrHttpPullModel is set to false, indicating a HTTP PUSH exchange model, the oadrTransportAddress shall be included; • registrationID – when reregistering only; • venID – when reregistering. If the venID is included in the initial registration payload, the assumption shall be that the VEN has been preconfigured with this value and the VTN should validate the venID just as it would any other payload with a venID. oadrCreatePartyRegistration shall not include the following payload element: • registrationID – for a new registration.
403	VEN/VTN, EiRegisterParty, oadrCreatedPartyRegistration When responding to an oadrCreatePartyRegistration, oadrCreatedPartyRegistration shall include the following payload elements: • all supported profiles and transports in the oadrProfiles object; • if the VEN has registered with an HTTP PULL model, then the oadrRequestedOadrPollFreq shall be included in the payload; • any oadrServiceSpecificInfo or oadrExtensions required to insure interoperability over the profile and transport being utilized. In addition, oadrCreatedPartyRegistration shall include the following payload elements, unless the oadrCreatedPartyRegistration is sent as a response to an oadrQueryRegistration (in which case inclusion of the elements is optional): – registrationID, – venID.
404	VEN/VTN, EiRegisterParty, oadrCreatedPartyRegistration When responding to an oadrQueryRegistration, oadrCreatedPartyRegistration shall include the following payload elements: • all supported profiles and transports in the oadrProfiles object; • all relevant oadrServiceSpecificInfo or oadrExtensions that may influence the VENs choice of profile or transport. oadrCreatedPartyRegistration shall not include the following payload elements: – registrationID – if the VEN has not registered with the VTN yet; – venID – if the VEN has not registered with the VTN yet. It is not necessary for the VTN to include the oadrPoll polling information in the query response, although it may do so.

405	VTN/VEN, EiRegisterParty, oadrCreatePartyRegistration 2.0b VEN shall implement EiRegisterParty service. When a 2.0b VEN boots or is reset, it should initiate registration (using the EiRegisterParty service) before sending any other message to or accepting any message from a VTN. However, the VEN may be registered out-of-band instead of using EiRegisterParty. A VEN should ignore all payloads from a VTN when not in a registered state. Refer to rule 517 for behaviour when a VEN attempts to communicate with a VTN in an unregistered state. After a completed new registration, VENs and VTNs shall exchange their metadata reports, and VENs (both PULL and PUSH models) shall use oadrRequestEvent to obtain all relevant events prior to initiating other payload exchanges. The exchange of metadata reports and the use of oadrRequestEvents is not required for reregistration.
406	VTN/VEN, EiRegisterParty, oadrCreatePartyRegistration A VEN may send an oadrCreatePartyRegistration (without venID and registrationID) even if it is already registered. If a VTN receives such a new registration (that is not a requested reregistration), it shall erase all existing reporting and registration information and initiate a completely new registration. It may reuse the same venID and registrationID that the VEN had before this new registration.
407	VTN/VEN, EiRegisterParty, oadrCancelPartyRegistration If a device receives an oadrCancelPartyRegistration from the other party, it shall ignore all information about this device, including exchanged metadata reports, requested reports, and registration information for further protocol messaging. Administrative historic records of the device, events, and other exchanges may be retained.

9.2.6 General conformance rules

Table 12 – General conformance rules

500	VEN/VTN, oadrPoll oadrPoll is a service-independent polling mechanism used by VENs in a PULL model to request pending service operations from the VTN. oadrPoll shall not be used by VENs in a PUSH model. oadrPoll shall not be used in the VTN to VEN direction. When using oadrPoll, the rules for which payloads are valid and how those payloads are delivered are the same as if the VTN had initiated the operations and pushed the payloads to the VEN. Only one operational payload may be sent by the VTN in response to the oadrPoll message. When no additional operational payloads are available in the queue, the VTN will respond with an oadrResponse payload. At each polling, VENs should, without waiting until the next polling interval, continue sending oadrPoll messages and processing payloads until an oadrResponse payload is returned to signal an empty VTN message queue. If the oadrPoll contains an incorrect venID, the VTN shall respond with an oadrResponse containing a 452 error code. If a logical response is required by the VEN to the received operational payload, the VEN shall send that logical response asynchronously via a transport request. The VTN should acknowledge this logical response with an oadrResponse payload. The VTN may optionally ignore an oadrPoll if it has not received an expected logical response to a payload delivered as a response to a previous poll. The VTN should be coded such that after some timeout it gives up waiting for the expected response and resumes responding to oadrPoll. oadrPoll requests shall be used by a VEN to retrieve messages from the VTN. The polling interval (i.e. the amount of time between two successive polls, specified as ISO 8601 duration in oadrRequestedOadrPollFreq) should be as requested by the VTN during registration, but may be higher because of processing constraints of the VEN or other factors. The polling interval should not be lower than the requested duration in oadrRequestedOadrPollFreq, except when the VEN is emptying the VTN payload queue. Service-specific requests should only be used for re-synchronization where necessary. While it is not a protocol violation for a VEN to poll at an interval longer than oadrRequestedOadrPollFreq specified by the VTN, the actual polling interval used by the VEN should not be greater than 20 % of the average event duration. This will allow the VTN to cancel an event in a timely manner.
501	VEN/VTN, oadrPoll All OpenADR B profile VTN implementations as well as simple HTTP VENs shall support oadrPoll.
502	VEN/VTN, oadrPoll, EiEvent service A VTN that is polled with oadrPoll typically will return a oadrDistributeEvent only when a new event is generated or event content has changed since the last oadrDistributeEvent was transmitted to the VEN. Transition of eventStatus, except for cancellation, should not be considered as change of event content. However, it is not an error for a VTN to return an oadrDistributeEvent in response to oadrPoll with only events that it has previously communicated. It should also be noted that the oadrDistributeEvent payload shall contain all active and pending events, regardless of whether it is polled using oadrPoll or oadrRequestEvent. The behaviour described above is consistent with existing conformance rules and is provided here as clarification of expected behaviour.
506	VEN/VTN A VTN that supports the B profile shall also concurrently support the A profile. A VEN that supports the B profile may also support the A profile. In that case, the VEN can be configured to communicate with a 2.0a VTN using, for example, any of the following options: – manual configuration (e.g. as part of setting up the URL of the VTN); – automatic fallback (during EiRegisterParty) when receiving any reply from the VTN using the 2.0a namespace; – automatic configuration based on the URL of the VTN (which contains "2.0b" for B VTNs as per conformance rule 511).
507	VEN/VTN, transport B profile VTNs shall support XMPP in addition to the Simple HTTP transport. B profile VENs may support either HTTP or XMPP or both. B profile VEN and VTNs that implement XMPP shall support the PUSH model. XMPP VENs may still make requests of the VTN as in the PULL model, however they shall not use the oadrPoll request.

508	VEN, venID All payloads sent by a VEN to a VTN shall contain a valid venID. This venID typically appears just below the root payload element, but in a few payloads (e.g. oadrRequestEvent and oadrCreatedEvent), the venID is one layer deeper in the schema. If a payload containing one of these venID elements off the root is instead sent by the VTN, the venID (if optional) may be omitted from the payload contents.
509	VEN/VTN, schema version The optional schemaVersion attribute shall be included in all payloads exchanged. The value of the schemaVersion attribute shall be either "2.0a" or "2.0b" to indicate the version of the OpenADR profile being used.
510	VEN/VTN – Minimum B profile feature support In order to facilitate interoperability and validation of behaviour, the following features shall be supported: 1) A VTN shall be capable of sending and a VEN shall be capable of receiving the following standard event signals: – SIMPLE; – ELECTRICITY_PRICE with a signalType of price; – LOAD_DISPATCH with a signalType of setpoint. 2) A VEN shall be capable of producing the following standard report, in addition to the metadata report, which both VENs and VTNs shall support: – TELEMETRY_STATUS report with the mandatory data points of oadrOnline and oadrManualOverride for an attached resource identified with a resourceID target. A VEN shall be capable of producing TELEMETRY_USAGE reports, at least for certification (and may offer it in deployments). The device shall be able to send some telemetry data (i.e. in case it does not have any metering resources attached, it shall provide sample data). A VEN may in addition support HISTORY_USAGE or other reports. A VEN shall provide sufficient storage to store recent data, at least 100 data points. In case the last reading has not been received by the VTN (e.g. because of transitory communication problems), recent history can be requested by the VTN from the VEN. A VTN shall support report registration (i.e. exchange of metadata report), and may optionally support additional report types. VTNs are not required to support periodic reporting of metadata reports if they do not have any reports to offer. Should a VTN receive a request for a periodic metadata report and have no reports to offer, it should respond to the request as if it were a one-shot request with the reportBackDuration value set to 0. 3) A Report Only VEN shall be capable of supporting all the Alliance's defined standard reports including: – TELEMETRY_USAGE – TELEMETRY_STATUS – HISTORY_USAGE 4) A VTN shall be capable of utilizing the eiTarget sub-elements of venID and resourceID. 5) A VEN shall be capable of utilizing the EiOpt service to further qualify the opt state of an event. While there is the expectation that fully functional VENs will be able to generate opt schedules, and fully functional VTNs will be capable of generating baselines and more than just metadata reports, these capabilities are not explicitly defined in this rule. If these capabilities can be configured by implementations, they will be tested as part of certification.
511	VTN/VEN B Profile endpoints shall use the following template in the Simple HTTP mode, while accepting normalized URIs according to the rules specified in IETF RFC 3986 (Section 6.2.3). Port and prefix are optional. https://<hostname>(:port)/(prefix/)OpenADR2/Simple/2.0b/<service> The following strings are used for the service name: – EiEvent, – EiOpt, – EiReport, – EiRegisterParty, – OadrPoll (only on the VTN). When VTN and VEN are running on the same IP address, they shall be deployed under different prefix while using the path under it as described above.

512	VTN/VEN venID and vtnID shall be case-sensitive (e.g. a vtnID of "vtnID1" is different from "vtnid1" when parsing an incoming message). For marketContext, the normalization rules specified in IETF RFC 3986 (section 6.2.3) shall be applied, allowing for hostname and scheme (at least) to be case insensitive and still be equivalent.
514	VEN/VTN, XML signatures Implementations may optionally support XML signatures as defined in 8.7. If a device is configured to use XML signatures, it shall ignore incoming messages that do not contain a valid signature, as defined in 8.7. If supported, implementation shall select from the following list of methods when creating the XML signature: canonicalizationMethod: – http://www.w3.org/TR/2001/REC-xml-c14n-20010315 – http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments SignatureMethod: – http://www.w3.org/2000/09/xmldsig#rsa-sha1 – http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 – http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1 – http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256 DigestMethod: – http://www.w3.org/2000/09/xmldsig#sha1 – http://www.w3.org/2001/04/xmldsig-more#sha256 The SignatureMethod and DigestMethod selected should be consistent with the negotiated TLS cipher and should be base64 encoded. Note that a VTN or VEN may be configured to support any methods based on the needs of a specific deployment. However, in the absence of changes to the default configuration of the VTN or VEN, the behaviour of the devices shall be as noted above. A device using XML signatures shall include a ReplayProtect element as SignatureProperty (refer to 8.7 for an example). The ReplayProtect element shall contain the dateTime when the payload is sent to the other party (not when it is created), as well as a random nonce. A device receiving a payload in high-security mode shall verify if the ReplayProtect element is part of the SignatureProperties element and shall reject the payload if the current date and time on the device differ from the value in the ReplayProtect more than a predefined value. In addition, the nonce may be used for further protection against replay attacks.
515	VEN/VTN, XMPP VEN to VEN communication shall not be allowed by the XMPP server. An XMPP client (both VTN and VEN) shall support XMPP presence (refer to 7.4.5.5). The client shall implement XMPP ping and may use it in deployments (refer to 7.4.5.6). The VEN shall only use a single endpoint defined by a single JID. During authentication to the XMPP server, the common name (CN) of the x.509 certificate shall match the username of the client. VENs shall use the XMPP discovery mechanism to determine the service endpoints provided by the VTN.
516	VEN/VTN, expired certificate VENs and VTNs shall not communicate at the application layer with a counter party that has an expired certificate.
517	VTN, error codes For communication with VEN, a VTN shall return errors in the ways defined below. If TLS handshake fails (e.g. because of invalid or expired digital certificate), the VTN shall return a transport layer error code of 403 for HTTP or a <forbidden/> error for XMPP. If TLS handshake is successful but the VEN is not yet registered, return a 463 application error code in an appropriate (relative to the request) payload. If TLS handshake is successful and the VEN is registered but a wrong ID is specified in the payload from the VEN (e.g. venID different from the one set during registration), return a 452 error code in an appropriate (relative to the request) payload.

9.3 Cardinality

Cardinalities for the schema in this document are outlined in Table 13.

Table 13 – Cardinalities

Object	Element	OADR 2.0
EiEventType	eventDescriptor:eventStatus	1
EiEventType	eventDescriptor:createdDateTime	1
EiEventType	eiActivePeriod:properties	1
EiEventType	eiActivePeriod:properties:dtstart	1
EiEventType	eiActivePeriod: properties:duration	1
EiEventType	eiActivePeriod: properties:tolerance	0 – 1
EiEventType	eiActivePeriod:properties:x-eiNotification	1
EiEventType	eiActivePeriod: properties:x-eiRampUp	0 – 1
EiEventType	eiActivePeriod: properties:x-Recovery	0 – 1
EiEventType	eiEventSignals:eiEventSignal	1 (A profile) 1 – many (B profile)
EiEventType	eiEventSignals:eiEventSignal:intervals	1
EiEventType	eiEventSignals:eiEventSignal:currentValue	1 (A profile) 0 – 1 (B profile)
EiEventType	eiEventSignals:eiEventSignal:intervals:interval	1 – many
EiEventType	eiEventSignals:eiEventSignal:intervals: Interval:duration	1
EiEventType	eiEventSignals:eiEventSignal:intervals: Interval/uid	1
EiEventType	eiTarget	1

Annex A (normative)

Detailed report description

Report Profiles		29-Nov-12																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
-----------------	--	-----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Annex B (normative)

Profile extensions

B.1 Overview

The schema of this document provide a number of ways in which various enumerated values and standardized sets of values (signal and reports) can be extended without the need to modify the schema. However, utilizing the extension mechanisms can cause interoperability issues between VENs and VTNs unless both are aware of the extensions. In general, the extensions rely on a schema type called EiExtensionTokenType, which allows enumerated values to be extended with an "x-" prefix without causing schema validation errors.

B.2 Report extension

This document has defined a number of standardized reports that are intended to meet most needs. If necessary, custom reports can be added by creating a new reportName prefixed with "x-". Existing reportType, units (itemBase), and readingType values can be utilized to construct the new report. Custom reportType and readingTypes can also be defined if necessary by prefixing the new values with "x-".

B.3 Event extension

This document has defined a number of standardized signals that are intended to meet most needs. If necessary, custom signals can be added by creating a new signalName prefixed with "x-". Existing signalType and units (itemBase) values can be utilized to construct the new signal. Custom signalType can be defined if necessary by prefixing the new value with "x-".

B.4 Other extensions

Other enumerated schema elements that can be extended via the "x-" prefix mechanism include optReason, responseCode, oadrDataQuality, schemaVersion, and device classes defined in the endDeviceAsset element contained in the EiEvent, EiOpt, and EiReport schemas.

The VTN may also support additional extension mechanisms that can be communicated to the VEN using the oadrCreatedRegistration:oadrExtensions object. How these extensions are implemented and what functionality they can provide are outside the scope of this document. However, great caution should be exercised in the implementation of any extension to insure interoperability with the large ecosystem of VEN implementations.

Annex C (normative)

oadrPoll scenarios

C.1 Overview

The current conformance rules allow different scenarios when a VTN has multiple payloads in its queue and it receives oadrPoll from the VEN. Annex C exemplifies several scenarios that are in accordance with this document and thus have to be supported by implementations.

C.2 Scenarios

Scenario 1:

- 1) VEN sends oadrPoll.
- 2) VTN responds with application layer request (oadrCreateReport).
- 3) VEN sends application layer response (oadrCreatedReport).
- 4) VTN responds with acknowledgement (oadrResponse).
- 5) VEN sends another oadrPoll.
- 6) VTN responds with next item in the queue.

It is assumed that this would be the typical behaviour.

Scenario 2:

- 1) VEN sends oadrPoll.
- 2) VTN responds with application layer request (oadrCreateReport).
- 3) VEN sends another oadrPoll.
- 4) VTN ignores the oadrPoll and does not respond.
- 5) VEN sends application layer response (oadrCreatedReport).
- 6) VTN responds with acknowledgement (oadrResponse).
- 7) VEN sends another oadrPoll.
- 8) VTN responds with next item in the queue.

The conformance rules allow the VTN to optionally ignore oadrPoll if a pending application layer response is expected.

Scenario 3:

- 1) VEN sends oadrPoll.
- 2) VTN responds with application layer request (oadrCreateReport).
- 3) VEN sends another oadrPoll.
- 4) VTN responds with application layer request (oadrDistributeEvent).
- 5) VEN sends application layer response (oadrCreatedReport).
- 6) VTN responds with acknowledgement (oadrResponse).
- 7) VEN sends application layer response (oadrCreatedEvent).
- 8) VTN responds with acknowledgement (oadrResponse).
- 9) VEN sends another oadrPoll.
- 10) VTN responds with next item in the queue.

VEN implementations will likely behave like scenario 1 (although the other scenarios are also valid): the VEN will not poll again until it provides an application layer response to the most recent payload from the VTN.

IECNORM.COM : Click to view the full PDF of IEC 62746-10-1:2018

Annex D (normative)

Definition of VEN, VTN, resource, and party

In typical DR interactions, grid side entities (e.g. service providers such as utilities, ISOs, etc.) intentionally try to modify the load profiles of demand-side entities (e.g. customers) through a logical construct known as a Resource.

A Resource is a demand side commodity that is associated with a load profile. Examples of Resources include the following:

- a single load such as a water heater;
- a collection of loads within a building such that the entire building is the Resource;
- a collection of buildings across a campus;
- a collection of disparate customers as in the case of aggregated Resources.

The key is that a Resource is associated with a load profile that can be influenced by grid-side service providers. Resources may be composed of other Resources in a hierarchical fashion. An *Asset* is a special type of a Resource that does not contain any sub-Resources and typically refers to specific physical loads.

A service provider influences the load profile of a Resource by sending out "DR signals" that are "applied" to Resources. Note that this does not imply that a service provider communicates with resources.

A Party is an entity that enters into some sort of business relationship or contract. Service providers and customers are Parties. Resources are commodities that are "owned" by Parties.

A VEN is a demand-side entity that is associated with Parties and Resources and forms the communications endpoint in the form of services that can be used to facilitate communications concerning the demand-side Resources and Parties that are associated with that VEN. Note that when Resources and Parties are associated with a VEN, it may take on "meaning" at a system level within the context of those associations, but the primary function of the VEN is a means of communicating about the Parties and Resources associated with the VEN.

A VTN is a grid-side entity that is associated with service providers and forms the communications endpoint in the form of services that can be used to facilitate communications with demand-side entities concerning Resources and Parties.

Annex E (normative)

IEC 62746-10-1 Schema

This annex contains listings of the 16 schema definition files that collectively comprise the IEC 62746-10-1 schema. The root schema file is oadr_20b.xsd. All other schema files are imported as required.

oadr_20b.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:dsig11="http://www.w3.org/2009/xmldsig11#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:oadr="http://openadr.org/oadr-2.0b/2012/07"
xmlns:clm5ISO42173A="urn:un:unece:uncefact:odelist:standard:5:ISO42173A:2010-04-07"
xmlns:pyld="http://docs.oasis-open.org/ns/energyinterop/201110/payloads" xmlns:ei="http://docs.oasis-
open.org/ns/energyinterop/201110" xmlns:scale="http://docs.oasis-open.org/ns/emix/2011/06/siscale"
xmlns:emix="http://docs.oasis-open.org/ns/emix/2011/06" xmlns:strm="urn:ietf:params:xml:ns:icalendar-2.0:stream"
xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0" xmlns:power="http://docs.oasis-
open.org/ns/emix/2011/06/power" xmlns:gb="http://naesb.org/espi" xmlns:atom="http://www.w3.org/2005/Atom"
targetNamespace="http://openadr.org/oadr-2.0b/2012/07" elementFormDefault="qualified"
attributeFormDefault="qualified">
  <xs:import namespace="urn:ietf:params:xml:ns:icalendar-2.0:stream" schemaLocation="oadr_strm_20b.xsd"/>
  <xs:import namespace="http://docs.oasis-open.org/ns/energyinterop/201110"
schemaLocation="oadr_ei_20b.xsd"/>
  <xs:import namespace="http://docs.oasis-open.org/ns/energyinterop/201110/payloads"
schemaLocation="oadr_pyld_20b.xsd"/>
  <xs:import namespace="http://docs.oasis-open.org/ns/emix/2011/06" schemaLocation="oadr_emix_20b.xsd"/>
  <xs:import namespace="http://docs.oasis-open.org/ns/emix/2011/06/siscale"
schemaLocation="oadr_siscale_20b.xsd"/>
  <xs:import namespace="urn:ietf:params:xml:ns:icalendar-2.0" schemaLocation="oadr_xcal_20b.xsd"/>
  <xs:import namespace="urn:un:unece:uncefact:odelist:standard:5:ISO42173A:2010-04-07"
schemaLocation="oadr_ISO_ISO3AlphaCurrencyCode_20100407.xsd"/>
  <xs:import namespace="http://naesb.org/espi" schemaLocation="oadr_greenbutton.xsd"/>
  <xs:import namespace="http://www.w3.org/2005/Atom" schemaLocation="oadr_atom.xsd"/>
  <xs:import namespace="http://www.w3.org/2009/xmldsig11#" schemaLocation="oadr_xmldsig11.xsd"/>
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="oadr_xmldsig.xsd"/>
  <xs:import namespace="http://openadr.org/oadr-2.0b/2012/07/xmldsig-properties"
schemaLocation="oadr_xmldsig-properties.schema.xsd"/>
  <xs:element name="oadrPayload">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ds:Signature" minOccurs="0"/>
        <xs:element ref="oadr:oadrSignedObject"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="oadrSignedObject">
    <xs:complexType>
      <xs:choice>
        <xs:element ref="oadr:oadrDistributeEvent"/>
        <xs:element ref="oadr:oadrCreatedEvent"/>
        <xs:element ref="oadr:oadrRequestEvent"/>
        <xs:element ref="oadr:oadrResponse"/>
        <xs:element ref="oadr:oadrCancelOpt"/>
        <xs:element ref="oadr:oadrCanceledOpt"/>
        <xs:element ref="oadr:oadrCreateOpt"/>
        <xs:element ref="oadr:oadrCreatedOpt"/>
        <xs:element ref="oadr:oadrCancelReport"/>
        <xs:element ref="oadr:oadrCanceledReport"/>
        <xs:element ref="oadr:oadrCreateReport"/>
        <xs:element ref="oadr:oadrCreatedReport"/>
        <xs:element ref="oadr:oadrRegisterReport"/>
        <xs:element ref="oadr:oadrRegisteredReport"/>
        <xs:element ref="oadr:oadrUpdateReport"/>
        <xs:element ref="oadr:oadrUpdatedReport"/>
        <xs:element ref="oadr:oadrCancelPartyRegistration"/>
        <xs:element ref="oadr:oadrCanceledPartyRegistration"/>
        <xs:element ref="oadr:oadrCreatePartyRegistration"/>
        <xs:element ref="oadr:oadrCreatedPartyRegistration"/>
        <xs:element ref="oadr:oadrRequestReregistration"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <xs:element ref="oadr:oadrQueryRegistration"/>
        <xs:element ref="oadr:oadrPoll"/>
      </xs:choice>
      <xs:attribute name="Id" type="xs:ID" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="oadrDistributeEvent" type="oadr:oadrDistributeEventType">
    <xs:annotation>
      <xs:documentation>Send DR Events to a VEN</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="oadrDistributeEventType">
    <xs:sequence>
      <xs:element ref="ei:eiResponse" minOccurs="0"/>
      <xs:element ref="pyld:requestID"/>
      <xs:element ref="ei:vtID"/>
      <xs:element name="oadrEvent" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>An object containing a demand response event</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="ei:eiEvent"/>
            <xs:element ref="oadr:oadrResponseRequired"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
  </xs:complexType>
  <xs:element name="oadrCreatedEvent" type="oadr:oadrCreatedEventType"/>
  <xs:complexType name="oadrCreatedEventType">
    <xs:sequence>
      <xs:element ref="pyld:eiCreatedEvent"/>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
  </xs:complexType>
  <xs:element name="oadrRequestEvent" type="oadr:oadrRequestEventType"/>
  <xs:complexType name="oadrRequestEventType">
    <xs:sequence>
      <xs:element ref="pyld:eiRequestEvent"/>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
  </xs:complexType>
  <xs:element name="oadrResponse" type="oadr:oadrResponseType"/>
  <xs:complexType name="oadrResponseType">
    <xs:sequence>
      <xs:element ref="ei:eiResponse"/>
      <xs:element ref="ei:venID" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
  </xs:complexType>
  <xs:element name="oadrCancelOpt" type="oadr:oadrCancelOptType">
    <xs:annotation>
      <xs:documentation>Cancel an opt schedule</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="oadrCancelOptType">
    <xs:sequence>
      <xs:element ref="pyld:requestID"/>
      <xs:element ref="ei:optID"/>
      <xs:element ref="ei:venID"/>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
  </xs:complexType>
  <xs:element name="oadrCanceledOpt" type="oadr:oadrCanceledOptType">
    <xs:annotation>
      <xs:documentation>Acknowledge cancelation of an opt schedule</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="oadrCanceledOptType">
    <xs:sequence>
      <xs:element ref="ei:eiResponse"/>
      <xs:element ref="ei:optID" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
  </xs:complexType>

```

```

<xs:element name="oadrCreateOpt" type="oadr:oadrCreateOptType">
  <xs:annotation>
    <xs:documentation>Create an optIn or optOut schedule</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrCreateOptType">
  <xs:complexContent>
    <xs:extension base="ei:EiOptType">
      <xs:sequence>
        <xs:element ref="pyId:requestID"/>
        <xs:element ref="ei:qualifiedEventID" minOccurs="0"/>
        <xs:element ref="ei:eiTarget"/>
        <xs:element ref="oadr:oadrDeviceClass" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="oadrCreatedOpt" type="oadr:oadrCreatedOptType">
  <xs:annotation>
    <xs:documentation>Acknowledge receipt of an opt schedule</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrCreatedOptType">
  <xs:sequence>
    <xs:element ref="ei:eiResponse"/>
    <xs:element ref="ei:optID"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrCancelReport" type="oadr:oadrCancelReportType">
  <xs:annotation>
    <xs:documentation>Cancel a report</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrCancelReportType">
  <xs:sequence>
    <xs:element ref="pyId:requestID"/>
    <xs:element ref="ei:reportRequestID" maxOccurs="unbounded"/>
    <xs:element ref="pyId:reportToFollow"/>
    <xs:element ref="ei:venID" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrCanceledReport" type="oadr:oadrCanceledReportType"/>
<xs:complexType name="oadrCanceledReportType">
  <xs:sequence>
    <xs:element ref="ei:eiResponse"/>
    <xs:element ref="oadr:oadrPendingReports"/>
    <xs:element ref="ei:venID" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrCreateReport" type="oadr:oadrCreateReportType">
  <xs:annotation>
    <xs:documentation>Request report from other party</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrCreateReportType">
  <xs:sequence>
    <xs:element ref="pyId:requestID"/>
    <xs:element ref="oadr:oadrReportRequest" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Request report</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="ei:venID" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrCreatedReport" type="oadr:oadrCreatedReportType">
  <xs:annotation>
    <xs:documentation>Acknowledge the request for report was received</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrCreatedReportType">
  <xs:sequence>
    <xs:element ref="ei:eiResponse"/>

```

```

    <xs:element ref="oadr:oadrPendingReports">
      <xs:annotation>
        <xs:documentation>List of periodic reports that have not yet been delivered</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="ei:venID" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrRegisterReport" type="oadr:oadrRegisterReportType">
  <xs:annotation>
    <xs:documentation>Register Metadata report settings with other party</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrRegisterReportType">
  <xs:sequence>
    <xs:element ref="pyId:requestID"/>
    <xs:element ref="oadr:oadrReport" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="ei:venID" minOccurs="0"/>
    <xs:element ref="ei:reportRequestID" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrRegisteredReport" type="oadr:oadrRegisteredReportType">
  <xs:annotation>
    <xs:documentation>Acknowledge registration of Metadata report by other party</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrRegisteredReportType">
  <xs:sequence>
    <xs:element ref="ei:eiResponse"/>
    <xs:element ref="oadr:oadrReportRequest" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="ei:venID" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrUpdateReport" type="oadr:oadrUpdateReportType">
  <xs:annotation>
    <xs:documentation>Send a previously requested report</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrUpdateReportType">
  <xs:sequence>
    <xs:element ref="pyId:requestID"/>
    <xs:element ref="oadr:oadrReport" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="ei:venID" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrUpdatedReport" type="oadr:oadrUpdatedReportType">
  <xs:annotation>
    <xs:documentation>Acknowledge receipt of a report</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrUpdatedReportType">
  <xs:sequence>
    <xs:element ref="ei:eiResponse"/>
    <xs:element ref="oadr:oadrCancelReport" minOccurs="0"/>
    <xs:element ref="ei:venID" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrCancelPartyRegistration" type="oadr:oadrCancelPartyRegistrationType">
  <xs:annotation>
    <xs:documentation>Cancel a registration</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrCancelPartyRegistrationType">
  <xs:sequence>
    <xs:element ref="pyId:requestID"/>
    <xs:element ref="ei:registrationID"/>
    <xs:element ref="ei:venID" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrCanceledPartyRegistration" type="oadr:oadrCanceledPartyRegistrationType">
  <xs:annotation>

```

```

        <xs:documentation>Acknowledge cancelation of registration</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="oadrCanceledPartyRegistrationType">
    <xs:sequence>
        <xs:element ref="ei:eiResponse"/>
        <xs:element ref="ei:registrationID" minOccurs="0"/>
        <xs:element ref="ei:venID" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrCreatePartyRegistration" type="oadr:oadrCreatePartyRegistrationType">
    <xs:annotation>
        <xs:documentation>Used by VEN to initiate registration with VTN</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="oadrCreatePartyRegistrationType">
    <xs:sequence>
        <xs:element ref="pyld:requestID"/>
        <xs:element ref="ei:registrationID" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Used for reregistering an existing registration</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="ei:venID" minOccurs="0"/>
        <xs:element ref="oadr:oadrProfileName"/>
        <xs:element ref="oadr:oadrTransportName"/>
        <xs:element ref="oadr:oadrTransportAddress" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Address of this VEN. Not required if http pull model</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="oadr:oadrReportOnly">
            <xs:annotation>
                <xs:documentation>ReportOnlyDeviceFlag – True or False</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="oadr:oadrXmlSignature">
            <xs:annotation>
                <xs:documentation>Implementation supports XML signatures – True or False</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="oadr:oadrVenName" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Human readable name for VEN</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="oadr:oadrHttpPullModel" minOccurs="0">
            <xs:annotation>
                <xs:documentation>If transport is simpleHttp indicate if VEN is operating in pull exchange model –
true or false</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrCreatedPartyRegistration" type="oadr:oadrCreatedPartyRegistrationType">
    <xs:annotation>
        <xs:documentation>Acknowledge receipt of VEN registration, provide VTN registration
info</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="oadrCreatedPartyRegistrationType">
    <xs:sequence>
        <xs:element ref="ei:eiResponse"/>
        <xs:element ref="ei:registrationID" minOccurs="0"/>
        <xs:element ref="ei:venID" minOccurs="0">
            <xs:annotation>
                <xs:documentation>venID not included in query unless already registered</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="ei:vtnID"/>
        <xs:element ref="oadr:oadrProfiles">
            <xs:annotation>
                <xs:documentation>VTN response to query registration returns all supported. This element is not
required for a registration response</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>

```

```

</xs:element>
<xs:element ref="oadr:oadrRequestedOadrPollFreq" minOccurs="0">
  <xs:annotation>
    <xs:documentation>HTTP Pull Only – The VEN shall send an oadrPoll payload to the VTN at most
once for each duration specified by this element</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="oadr:oadrServiceSpecificInfo" minOccurs="0"/>
<xs:element name="oadrExtensions" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="oadrExtension" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="oadrExtensionName" type="xs:string"/>
            <xs:element ref="oadr:oadrInfo" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrRequestReregistration" type="oadr:oadrRequestReregistrationType">
  <xs:annotation>
    <xs:documentation>Used by VTN to request that the VEN reregister</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrRequestReregistrationType">
  <xs:sequence>
    <xs:element ref="ei:venID"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrQueryRegistration" type="oadr:oadrQueryRegistrationType">
  <xs:annotation>
    <xs:documentation>Query VTN for registration information without actually registering</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrQueryRegistrationType">
  <xs:sequence>
    <xs:element ref="pyId:requestID"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrPoll" type="oadr:oadrPollType">
  <xs:annotation>
    <xs:documentation>Query pull VTN for payloads with new or modified information</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrPollType">
  <xs:sequence>
    <xs:element ref="ei:venID"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="oadrVenName" type="xs:string">
  <xs:annotation>
    <xs:documentation>VEN name. May be used in VTN GUI</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="oadrProfiles">
  <xs:annotation>
    <xs:documentation>OpenADR profiles supported by the implementation</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="oadrProfile" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="oadr:oadrProfileName"/>
            <xs:element ref="oadr:oadrTransports"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="oadrProfileName" type="oadr:oadrProfileType">
  <xs:annotation>
    <xs:documentation>OpenADR profile name such as 2.0a or 2.0b.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:simpleType name="oadrProfileType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="2.0a"/>
    <xs:enumeration value="2.0b"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="oadrTransports">
  <xs:annotation>
    <xs:documentation>OpenADR transports supported by implementation</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="oadrTransport" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="oadr:oadrTransportName"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="oadrTransportName" type="oadr:oadrTransportType">
  <xs:annotation>
    <xs:documentation>OpenADR transport name such as simpleHttp or xmpp</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:simpleType name="oadrTransportType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="simpleHttp"/>
    <xs:enumeration value="xmpp"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="oadrHttpPullModel" type="xs:boolean"/>
<xs:element name="oadrServiceSpecificInfo">
  <xs:annotation>
    <xs:documentation>Service specific registration information</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="oadrService" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="oadr:oadrServiceName"/>
            <xs:element ref="oadr:oadrInfo" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="oadrServiceName" type="oadr:oadrServiceNameType"/>
<xs:simpleType name="oadrServiceNameType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="EiEvent"/>
    <xs:enumeration value="EiOpt"/>
    <xs:enumeration value="EiReport"/>
    <xs:enumeration value="EiRegisterParty"/>
    <xs:enumeration value="OadrPoll"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="oadrInfo">
  <xs:annotation>
    <xs:documentation>A key value pair of service specific registration information</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="oadrKey" type="xs:string"/>
      <xs:element name="oadrValue" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="oadrXmlSignature" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>Implementation supports XML signature</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="oadrReportOnly" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>ReportOnlyDeviceFlag</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="oadrTransportAddress" type="xs:string">
  <xs:annotation>
    <xs:documentation>Root address used to communicate with other party. Should include port if
required</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="oadrRequestedOadrPollFreq" type="xcal:DurationPropType">
  <xs:annotation>
    <xs:documentation>The VEN shall send an oadrPoll payload to the VTN at most once for each duration
specified by this element</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="oadrResponseRequired" type="oadr:ResponseRequiredType">
  <xs:annotation>
    <xs:documentation>Controls when optIn/optOut response is required. Can be always or
never</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:simpleType name="ResponseRequiredType">
  <xs:annotation>
    <xs:documentation>Defines what type of response is required</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="always">
      <xs:annotation>
        <xs:documentation>Always send a response for every event received.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="never">
      <xs:annotation>
        <xs:documentation>Never respond.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
<xs:element name="oadrPendingReports" type="oadr:oadrPendingReportsType"/>
<xs:complexType name="oadrPendingReportsType">
  <xs:sequence>
    <xs:element ref="ei:reportRequestID" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="customUnit" type="oadr:BaseUnitType" substitutionGroup="emix:itemBase"/>
<xs:complexType name="BaseUnitType" mixed="false">
  <xs:annotation>
    <xs:documentation>Custom Units</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="emix:ItemBaseType">
      <xs:sequence>
        <xs:element name="itemDescription" type="xs:string"/>
        <xs:element name="itemUnits" type="xs:string"/>
        <xs:element ref="scale:siScaleCode"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="current" type="oadr:CurrentType" substitutionGroup="emix:itemBase"/>
<xs:complexType name="CurrentType" mixed="false">
  <xs:annotation>
    <xs:documentation>Current</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="emix:ItemBaseType">
      <xs:sequence>

```

```

        <xs:element name="itemDescription" type="xs:string" fixed="Current"/>
        <xs:element name="itemUnits" type="xs:string" fixed="A"/>
        <xs:element ref="scale:siScaleCode"/>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="currency" type="oadr:currencyType" substitutionGroup="emix:itemBase"/>
<xs:element name="currencyPerKWh" type="oadr:currencyType" substitutionGroup="emix:itemBase"/>
<xs:element name="currencyPerKW" type="oadr:currencyType" substitutionGroup="emix:itemBase"/>
<xs:element name="currencyPerThm" type="oadr:currencyType" substitutionGroup="emix:itemBase"/>
<xs:complexType name="currencyType" mixed="false">
    <xs:annotation>
        <xs:documentation>currency</xs:documentation>
    </xs:annotation>
    <xs:complexContent mixed="false">
        <xs:extension base="emix:ItemBaseType">
            <xs:sequence>
                <xs:element name="itemDescription" type="oadr:currencyItemDescriptionType"/>
                <xs:element name="itemUnits" type="clm5ISO42173A:ISO3AlphaCurrencyCodeContentType">
                    <xs:annotation>
                        <xs:documentation>ISO enumeration of currency types, such as USD</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element ref="scale:siScaleCode"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="currencyItemDescriptionType">
    <xs:restriction base="xs:token">
        <xs:enumeration value="currency"/>
        <xs:enumeration value="currencyPerKW"/>
        <xs:enumeration value="currencyPerKWh"/>
    </xs:restriction>
</xs:simpleType>
<xs:element name="frequency" type="oadr:FrequencyType" substitutionGroup="emix:itemBase"/>
<xs:complexType name="FrequencyType" mixed="false">
    <xs:annotation>
        <xs:documentation>Frequency</xs:documentation>
    </xs:annotation>
    <xs:complexContent mixed="false">
        <xs:extension base="emix:ItemBaseType">
            <xs:sequence>
                <xs:element name="itemDescription" type="xs:string" fixed="Frequency"/>
                <xs:element name="itemUnits" type="xs:string" fixed="Hz"/>
                <xs:element ref="scale:siScaleCode"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="Therm" type="oadr:ThermType" substitutionGroup="emix:itemBase"/>
<xs:complexType name="ThermType" mixed="false">
    <xs:annotation>
        <xs:documentation>Therm</xs:documentation>
    </xs:annotation>
    <xs:complexContent mixed="false">
        <xs:extension base="emix:ItemBaseType">
            <xs:sequence>
                <xs:element name="itemDescription" type="xs:string" fixed="Therm"/>
                <xs:element name="itemUnits" type="xs:string" fixed="thm"/>
                <xs:element ref="scale:siScaleCode"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="temperature" type="oadr:temperatureType" substitutionGroup="emix:itemBase"/>
<xs:complexType name="temperatureType" mixed="false">
    <xs:annotation>
        <xs:documentation>temperature</xs:documentation>
    </xs:annotation>
    <xs:complexContent mixed="false">
        <xs:extension base="emix:ItemBaseType">
            <xs:sequence>
                <xs:element name="itemDescription" type="xs:string" fixed="temperature"/>
                <xs:element name="itemUnits" type="oadr:temperatureUnitType">
                    <xs:annotation>

```

```

        <xs:documentation>Temperature in Celsius or Fahrenheit</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="scale:siScaleCode"/>
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:simpleType name="temperatureUnitType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="celsius"/>
    <xs:enumeration value="fahrenheit"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="pulseCount" type="oadr:pulseCountType" substitutionGroup="emix:itemBase"/>
<xs:element name="pulseFactor" type="xs:float">
  <xs:annotation>
    <xs:documentation>kWh per count</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="pulseCountType" mixed="false">
  <xs:annotation>
    <xs:documentation>Pulse Count</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="emix:ItemBaseType">
      <xs:sequence>
        <xs:element name="itemDescription" type="xs:string" fixed="pulse count"/>
        <xs:element name="itemUnits" type="xs:string" fixed="count">
          <xs:annotation>
            <xs:documentation>Pulse count from meter</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element ref="oadr:pulseFactor"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="oadrReportRequest" type="oadr:oadrReportRequestType"/>
<xs:complexType name="oadrReportRequestType">
  <xs:annotation>
    <xs:documentation>This type is used to request an EiReport</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="ei:reportRequestID"/>
    <xs:element ref="ei:reportSpecifier"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="oadrReport" type="oadr:oadrReportType"/>
<xs:complexType name="oadrReportType">
  <xs:annotation>
    <xs:documentation>eiReport is a Stream of [measurements] recorded over time and delivered to the
requestor periodically. The readings may be actual, computed, summed if derived in some other
manner.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="strm:StreamBaseType">
      <xs:sequence>
        <xs:element ref="ei:eiReportID" minOccurs="0">
          <xs:annotation>
            <xs:documentation>reference ID to this report.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element ref="oadr:oadrReportDescription" minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>Define data points the implementation is capable of reporting on. Only used
in Metadata report</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element ref="ei:reportRequestID">
          <xs:annotation>
            <xs:documentation>Reference to the oadrCreateReport request that defined this
report.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element ref="ei:reportSpecifierID">
          <xs:annotation>

```

```

        <xs:documentation>Reference to Metadata report from which this report was
derived.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element ref="ei:reportName" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Name possibly for use in a user interface.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element ref="ei:createdDateTime"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="oadrReportDescription" type="oadr:oadrReportDescriptionType"/>
<xs:complexType name="oadrReportDescriptionType">
    <xs:annotation>
        <xs:documentation>Describes the subject and attributes of a report.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element ref="ei:rID"/>
        <xs:element ref="ei:reportSubject" minOccurs="0"/>
        <xs:element ref="ei:reportDataSource" minOccurs="0"/>
        <xs:element ref="ei:reportType"/>
        <xs:element ref="emix:itemBase" minOccurs="0">
            <xs:annotation>
                <xs:documentation>What is measured or tracked in this report (Units).</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="ei:readingType"/>
        <xs:element ref="emix:marketContext" minOccurs="0"/>
        <xs:element ref="oadr:oadrSamplingRate" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="oadrSamplingRateType">
    <xs:sequence>
        <xs:element name="oadrMinPeriod" type="xcal:DurationValueType">
            <xs:annotation>
                <xs:documentation>Minimum sampling period</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="oadrMaxPeriod" type="xcal:DurationValueType">
            <xs:annotation>
                <xs:documentation>Maximum sampling period</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="oadrOnChange" type="xs:boolean">
            <xs:annotation>
                <xs:documentation>If true then the data will be recorded when it changes, but at no greater a
frequency than that specified by minPeriod.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:element name="oadrSamplingRate" type="oadr:oadrSamplingRateType">
    <xs:annotation>
        <xs:documentation>Sampling rate for telemetry type data</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element
    name="oadrPayloadResourceStatus"
    type="oadr:oadrPayloadResourceStatusType"
    substitutionGroup="ei:payloadBase">
    <xs:annotation>
        <xs:documentation>Current resource status information</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="oadrPayloadResourceStatusType">
    <xs:annotation>
        <xs:documentation>This is the payload for reports that require a status.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="ei:PayloadBaseType">
            <xs:sequence>
                <xs:element name="oadrOnline" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>If true then resource/asset is online, if false then offline.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>

```

```

        <xs:element name="oadrManualOverride" type="xs:boolean">
          <xs:annotation>
            <xs:documentation>If true then the control of the load has been manually
overridden</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element ref="oadr:oadrLoadControlState" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="oadrLoadControlState" type="oadr:oadrLoadControlStateType"/>
<xs:complexType name="oadrLoadControlStateType">
  <xs:sequence>
    <xs:element name="oadrCapacity" type="oadr:oadrLoadControlStateTypeType" minOccurs="0"/>
    <xs:element name="oadrLevelOffset" type="oadr:oadrLoadControlStateTypeType" minOccurs="0"/>
    <xs:element name="oadrPercentOffset" type="oadr:oadrLoadControlStateTypeType" minOccurs="0"/>
    <xs:element name="oadrSetPoint" type="oadr:oadrLoadControlStateTypeType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="oadrLoadControlStateTypeType">
  <xs:sequence>
    <xs:element name="oadrMin" type="xs:float" minOccurs="0"/>
    <xs:element name="oadrMax" type="xs:float" minOccurs="0"/>
    <xs:element name="oadrCurrent" type="xs:float"/>
    <xs:element name="oadrNormal" type="xs:float" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="oadrGBItemBase">
  <xs:complexContent>
    <xs:extension base="emix:itemBaseType">
      <xs:sequence>
        <xs:element ref="atom:feed"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="oadrGBStreamPayloadBase">
  <xs:complexContent>
    <xs:extension base="strm:StreamPayloadBaseType">
      <xs:sequence>
        <xs:element ref="atom:feed"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element
  name="oadrGBPayload"
  type="oadr:oadrGBStreamPayloadBase"
  substitutionGroup="strm:streamPayloadBase"/>
<xs:element name="oadrGBDataDescription" type="oadr:oadrGBItemBase" substitutionGroup="emix:itemBase"/>
<xs:element name="oadrDeviceClass" type="ei:EiTargetType">
  <xs:annotation>
    <xs:documentation>Device Class target – use only endDeviceAsset.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element
  name="oadrReportPayload"
  type="oadr:oadrReportPayloadType"
  substitutionGroup="strm:streamPayloadBase">
  <xs:annotation>
    <xs:documentation>Data point values for reports</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="oadrReportPayloadType">
  <xs:annotation>
    <xs:documentation>Report payload for use in reports.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="ei:ReportPayloadType">
      <xs:sequence>
        <xs:element ref="oadr:oadrDataQuality" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Enumerated value for the quality of this data item</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="oadrDataQuality" type="oadr:oadrDataQualityTypeType"/>

```

```

<xs:simpleType name="oadrDataQualityType">
  <xs:union memberTypes="oadr:oadrDataQualityType ei:EiExtensionTokenType"/>
</xs:simpleType>
<xs:simpleType name="oadrDataQualityType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="No Quality – No Value"/>
    <xs:enumeration value="No New Value – Previous Value Used"/>
    <xs:enumeration value="Quality Bad – Non Specific"/>
    <xs:enumeration value="Quality Bad – Configuration Error"/>
    <xs:enumeration value="Quality Bad – Not Connected"/>
    <xs:enumeration value="Quality Bad – Device Failure"/>
    <xs:enumeration value="Quality Bad – Sensor Failure"/>
    <xs:enumeration value="Quality Bad – Last Known Value"/>
    <xs:enumeration value="Quality Bad – Comm Failure"/>
    <xs:enumeration value="Quality Bad – Out of Service"/>
    <xs:enumeration value="Quality Uncertain – Non Specific"/>
    <xs:enumeration value="Quality Uncertain – Last Usable Value"/>
    <xs:enumeration value="Quality Uncertain – Sensor Not Accurate"/>
    <xs:enumeration value="Quality Uncertain – EU Units Exceeded"/>
    <xs:enumeration value="Quality Uncertain – Sub Normal"/>
    <xs:enumeration value="Quality Good – Non Specific"/>
    <xs:enumeration value="Quality Good – Local Override"/>
    <xs:enumeration value="Quality Limit – Field/Not"/>
    <xs:enumeration value="Quality Limit – Field/Low"/>
    <xs:enumeration value="Quality Limit – Field/High"/>
    <xs:enumeration value="Quality Limit – Field/Constant"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

IECNORM.COM : Click to view the full PDF of IEC 62746-10-1:2018

oadr_atom.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:atom="http://www.w3.org/2005/Atom" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xhtml="http://www.w3.org/XML/1998/namespace" targetNamespace="http://www.w3.org/2005/Atom"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation>
      This version of the Atom schema is based on version 1.0 of the format specifications,
      found here http://www.atomenabled.org/developers/syndication/atom-format-spec.php.
    </xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="oadr_xml.xsd"/>
  <xs:annotation>
    <xs:documentation>
      An Atom document may have two root elements, feed and entry, as defined in section 2.
    </xs:documentation>
  </xs:annotation>
  <xs:element name="feed" type="atom:feedType"/>
  <xs:element name="entry" type="atom:entryType"/>
  <xs:complexType name="textType" mixed="true">
    <xs:annotation>
      <xs:documentation>
        The Atom text construct is defined in section 3.1 of the format spec.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:any namespace="http://www.w3.org/1999/xhtml" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="type">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="text"/>
          <xs:enumeration value="html"/>
          <xs:enumeration value="xhtml"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attributeGroup ref="atom:commonAttributes"/>
  </xs:complexType>
  <xs:complexType name="personType">
    <xs:annotation>
      <xs:documentation>
        The Atom person construct is defined in section 3.2 of the format spec.
      </xs:documentation>
    </xs:annotation>
    <xs:choice maxOccurs="unbounded">
      <xs:element name="name" type="xs:string"/>
      <xs:element name="uri" type="atom:uriType" minOccurs="0"/>
      <xs:element name="email" type="atom:emailType" minOccurs="0"/>
      <xs:any namespace="##other"/>
    </xs:choice>
    <xs:attributeGroup ref="atom:commonAttributes"/>
  </xs:complexType>
  <xs:simpleType name="emailType">
    <xs:annotation>
      <xs:documentation>
        Schema definition for an email address.
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:normalizedString">
      <xs:pattern value="\w+@(\w+\.)+\w+"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="feedType">
    <xs:annotation>
      <xs:documentation>
        The Atom feed construct is defined in section 4.1.1 of the format spec.
      </xs:documentation>
    </xs:annotation>
    <xs:choice minOccurs="3" maxOccurs="unbounded">
      <xs:element name="author" type="atom:personType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="category" type="atom:categoryType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="contributor" type="atom:personType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="generator" type="atom:generatorType" minOccurs="0"/>
    </xs:choice>
  </xs:complexType>

```

```

<xs:element name="icon" type="atom:iconType" minOccurs="0"/>
<xs:element name="id" type="atom:idType"/>
<xs:element name="link" type="atom:linkType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="logo" type="atom:logoType" minOccurs="0"/>
<xs:element name="rights" type="atom:textType" minOccurs="0"/>
<xs:element name="subtitle" type="atom:textType" minOccurs="0"/>
<xs:element name="title" type="atom:textType"/>
<xs:element name="updated" type="atom:dateTimeType"/>
<xs:element name="entry" type="atom:entryType" minOccurs="0" maxOccurs="unbounded"/>
<xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
</xs:choice>
<xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="entryType">
  <xs:annotation>
    <xs:documentation>
      The Atom entry construct is defined in section 4.1.2 of the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:choice maxOccurs="unbounded">
    <xs:element name="author" type="atom:personType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="category" type="atom:categoryType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="content" type="atom:contentType" minOccurs="0"/>
    <xs:element name="contributor" type="atom:personType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="id" type="atom:idType"/>
    <xs:element name="link" type="atom:linkType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="published" type="atom:dateTimeType" minOccurs="0"/>
    <xs:element name="rights" type="atom:textType" minOccurs="0"/>
    <xs:element name="source" type="atom:textType" minOccurs="0"/>
    <xs:element name="summary" type="atom:textType" minOccurs="0"/>
    <xs:element name="title" type="atom:textType"/>
    <xs:element name="updated" type="atom:dateTimeType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>
  <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="contentType" mixed="true">
  <xs:annotation>
    <xs:documentation>
      The Atom content construct is defined in section 4.1.3 of the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="type" type="xs:string"/>
  <xs:attribute name="src" type="xs:anyURI"/>
  <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="categoryType">
  <xs:annotation>
    <xs:documentation>
      The Atom category construct is defined in section 4.2.2 of the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="term" type="xs:string" use="required"/>
  <xs:attribute name="scheme" type="xs:anyURI" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="generatorType">
  <xs:annotation>
    <xs:documentation>
      The Atom generator element is defined in section 4.2.4 of the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="uri" type="xs:anyURI" use="optional"/>
      <xs:attribute name="version" type="xs:string" use="optional"/>
      <xs:attributeGroup ref="atom:commonAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="iconType">
  <xs:annotation>
    <xs:documentation>

```

The Atom icon construct is defined in section 4.2.5 of the format spec.

```

</xs:documentation>
</xs:annotation>
<xs:simpleContent>
  <xs:extension base="xs:anyURI">
    <xs:attributeGroup ref="atom:commonAttributes"/>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="idType">
  <xs:annotation>
    <xs:documentation>
      The Atom id construct is defined in section 4.2.6 of the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:anyURI">
      <xs:attributeGroup ref="atom:commonAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="linkType" mixed="true">
  <xs:annotation>
    <xs:documentation>
      The Atom link construct is defined in section 3.4 of the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="href" type="xs:anyURI" use="required"/>
  <xs:attribute name="rel" type="xs:string" use="optional"/>
  <xs:attribute name="type" type="xs:string" use="optional"/>
  <xs:attribute name="hreflang" type="xs:NMTOKEN" use="optional"/>
  <xs:attribute name="title" type="xs:string" use="optional"/>
  <xs:attribute name="length" type="xs:positiveInteger" use="optional"/>
  <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="logoType">
  <xs:annotation>
    <xs:documentation>
      The Atom logo construct is defined in section 4.2.8 of the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:anyURI">
      <xs:attributeGroup ref="atom:commonAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="sourceType">
  <xs:annotation>
    <xs:documentation>
      The Atom source construct is defined in section 4.2.11 of the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:choice maxOccurs="unbounded">
    <xs:element name="author" type="atom:personType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="category" type="atom:categoryType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="contributor" type="atom:personType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="generator" type="atom:generatorType" minOccurs="0"/>
    <xs:element name="icon" type="atom:iconType" minOccurs="0"/>
    <xs:element name="id" type="atom:idType" minOccurs="0"/>
    <xs:element name="link" type="atom:linkType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="logo" type="atom:logoType" minOccurs="0"/>
    <xs:element name="rights" type="atom:textType" minOccurs="0"/>
    <xs:element name="subtitle" type="atom:textType" minOccurs="0"/>
    <xs:element name="title" type="atom:textType" minOccurs="0"/>
    <xs:element name="updated" type="atom:dateTimeType" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>
  <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="uriType">
  <xs:simpleContent>
    <xs:extension base="xs:anyURI">
      <xs:attributeGroup ref="atom:commonAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

```
<xs:complexType name="dateTimeType">
  <xs:simpleContent>
    <xs:extension base="xs:dateTime">
      <xs:attributeGroup ref="atom:commonAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:attributeGroup name="commonAttributes">
  <xs:attribute ref="xml:base"/>
  <xs:attribute ref="xml:lang"/>
  <xs:anyAttribute namespace="##other"/>
</xs:attributeGroup>
</xs:schema>
```

IECNORM.COM : Click to view the full PDF of IEC 62746-10-1:2018

oadr_ei_20b.xsd

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ei="http://docs.oasis-
open.org/ns/energyinterop/201110" xmlns:pyld="http://docs.oasis-open.org/ns/energyinterop/201110/payloads"
xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0" xmlns:emix="http://docs.oasis-open.org/ns/emix/2011/06"
xmlns:strm="urn:ietf:params:xml:ns:icalendar-2.0:stream" xmlns:power="http://docs.oasis-
open.org/ns/emix/2011/06/power" targetNamespace="http://docs.oasis-open.org/ns/energyinterop/201110"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xs:import namespace="http://docs.oasis-open.org/ns/emix/2011/06/power"
schemaLocation="oadr_power_20b.xsd"/>
  <xs:import namespace="urn:ietf:params:xml:ns:icalendar-2.0" schemaLocation="oadr_xcal_20b.xsd"/>
  <xs:import namespace="urn:ietf:params:xml:ns:icalendar-2.0:stream" schemaLocation="oadr_strm_20b.xsd"/>
  <xs:import namespace="http://docs.oasis-open.org/ns/emix/2011/06" schemaLocation="oadr_emix_20b.xsd"/>
  <xs:import namespace="http://docs.oasis-open.org/ns/energyinterop/201110/payloads"
schemaLocation="oadr_pyld_20b.xsd"/>
  <xs:element name="eventStatus" type="ei:EventStatusEnumeratedType"/>
  <xs:simpleType name="EventStatusEnumeratedType">
    <xs:restriction base="xs:token">
      <xs:enumeration value="none">
        <xs:annotation>
          <xs:documentation>No event pending</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="far">
        <xs:annotation>
          <xs:documentation>Event pending in the far future. The exact definition of how far in the future this
refers is dependent upon the market context, but typically means the next day.</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="near">
        <xs:annotation>
          <xs:documentation>Event pending in the near future. The exact definition of how near in the future
the pending event is active is dependent on the market context.</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="active">
        <xs:annotation>
          <xs:documentation>The event has been initiated and is currently active.</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="completed">
        <xs:annotation>
          <xs:documentation>The event has completed.</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="cancelled">
        <xs:annotation>
          <xs:documentation>The event has been canceled.</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="resourceID" type="xs:string"/>
  <xs:element name="groupID" type="xs:string"/>
  <xs:element name="partyID" type="xs:string"/>
  <xs:element name="groupName" type="xs:string"/>
  <xs:simpleType name="EiExtensionTokenType">
    <xs:annotation>
      <xs:documentation>Pattern used for extending string enumeration, where allowed.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:token">
      <xs:pattern value="x-\\S.*"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="venID" type="xs:string"/>
  <xs:element name="vtnID" type="xs:string"/>
  <xs:element name="eventID" type="xs:string">
    <xs:annotation>
      <xs:documentation>An ID value that identifies a specific DR event instance.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="modificationNumber" type="xs:unsignedInt">
    <xs:annotation>
      <xs:documentation>Incremented each time an event is modified.</xs:documentation>
    </xs:annotation>
  </xs:element>

```

```

    </xs:annotation>
  </xs:element>
  <xs:element name="qualifiedEventID" type="ei:QualifiedEventIDType"/>
  <xs:complexType name="QualifiedEventIDType">
    <xs:annotation>
      <xs:documentation>Fully qualified event ID includes the eventID and the
modificationNumber.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="ei:eventID"/>
      <xs:element ref="ei:modificationNumber"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="x-eiNotification" type="xcal:DurationPropType">
    <xs:annotation>
      <xs:documentation>The VEN should receive the DR event payload prior to dtstart minus this
duration.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="x-eiRampUp" type="xcal:DurationPropType">
    <xs:annotation>
      <xs:documentation>A duration before or after the event start time during which load shed should
transit.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="x-eiRecovery" type="xcal:DurationPropType">
    <xs:annotation>
      <xs:documentation>A duration before or after the event end time during which load shed should
transit.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="interval" type="ei:IntervalType"/>
  <xs:complexType name="IntervalType">
    <xs:sequence>
      <xs:element ref="xcal:dtstart" minOccurs="0"/>
      <xs:element ref="xcal:duration" minOccurs="0"/>
      <xs:element ref="xcal:uid" minOccurs="0"/>
      <xs:element ref="strm:streamPayloadBase" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="currentValue" type="ei:currentValueType">
    <xs:annotation>
      <xs:documentation>The payloadFloat value of the event interval currently executing.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="currentValueType">
    <xs:choice>
      <xs:element ref="ei:payloadFloat"/>
    </xs:choice>
  </xs:complexType>
  <xs:element name="payloadBase" type="ei:PayloadBaseType" abstract="true"/>
  <xs:complexType name="PayloadBaseType" abstract="true">
    <xs:annotation>
      <xs:documentation>Base for information in signal / baseline / report payloads</xs:documentation>
    </xs:annotation>
  </xs:complexType>
  <xs:element name="payloadFloat" type="ei:PayloadFloatType" substitutionGroup="ei:payloadBase">
    <xs:annotation>
      <xs:documentation>Data point value for event signals or for reporting current or historical
values.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="PayloadFloatType">
    <xs:annotation>
      <xs:documentation>This is the payload for signals that require a quantity.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="ei:PayloadBaseType">
        <xs:sequence>
          <xs:element name="value" type="xs:float"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="responseCode" type="ei:ResponseCodeType">
    <xs:annotation>
      <xs:documentation>A 3 digit response code</xs:documentation>
    </xs:annotation>
  </xs:element>

```

```

</xs:annotation>
</xs:element>
<xs:simpleType name="ResponseCodeType">
  <xs:annotation>
    <xs:documentation>Similar to HTTP 1.1 Error Pattern, 1st digit sufficient for most error processing
    – 1xx: Informational – Request received, continuing process
    – 2xx: Success – The action was successfully received, understood, and accepted
    – 3xx: Pending – Further action must be taken in order to complete the request
    – 4xx: Requester Error – The request contains bad syntax or cannot be fulfilled
    – 5xx: Responder Error – The responder failed to fulfill an apparently valid request
    xx is used for defining more fine grained errors
  </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9][0-9][0-9]"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="responseDescription" type="xs:string">
  <xs:annotation>
    <xs:documentation>Narrative description of response status</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="optType" type="ei:OptTypeType">
  <xs:annotation>
    <xs:documentation>optIn or optOut of an event</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:simpleType name="OptTypeType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="optIn"/>
    <xs:enumeration value="optOut"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="eiResponse" type="ei:EiResponseType">
  <xs:annotation>
    <xs:documentation>Indicate whether received payload is acceptable</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="EiResponseType">
  <xs:sequence>
    <xs:element ref="ei:responseCode"/>
    <xs:element ref="ei:responseDescription" minOccurs="0"/>
    <xs:element ref="pyId:requestID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="eventResponses">
  <xs:annotation>
    <xs:documentation>optIn or optOut responses for received events</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="eventResponse" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="ei:responseCode"/>
            <xs:element ref="ei:responseDescription" minOccurs="0"/>
            <xs:element ref="pyId:requestID"/>
            <xs:element ref="ei:qualifiedEventID"/>
            <xs:element ref="ei:optType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="eiEvent" type="ei:EiEventType"/>
<xs:complexType name="EiEventType">
  <xs:sequence>
    <xs:element ref="ei:eventDescriptor"/>
    <xs:element ref="ei:eiActivePeriod"/>
    <xs:element ref="ei:eiEventSignals"/>
    <xs:element ref="ei:eiTarget"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="eiActivePeriod" type="ei:eiActivePeriodType">
  <xs:annotation>
    <xs:documentation>Time frames relevant to the overall event</xs:documentation>
  </xs:annotation>

```

```

</xs:annotation>
</xs:element>
<xs:complexType name="eiActivePeriodType">
  <xs:sequence>
    <xs:element ref="xcal:properties"/>
    <xs:element ref="xcal:components"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="signalType" type="ei:SignalTypeEnumType">
  <xs:annotation>
    <xs:documentation>An enumerated value describing the type of signal such as level or
price</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:simpleType name="SignalTypeEnumType">
  <xs:annotation>
    <xs:documentation>SignalTypeEnumType lists the pre-defined types used to specify the payload types
and conformance in a stream</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:enumeration value="delta">
      <xs:annotation>
        <xs:documentation>Signal indicates the amount to change from what one would have used without
the signal.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="level">
      <xs:annotation>
        <xs:documentation>Signal indicates a program level.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="multiplier">
      <xs:annotation>
        <xs:documentation>Signal indicates a multiplier applied to the current rate of delivery or usage from
what one would have used without the signal.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="price">
      <xs:annotation>
        <xs:documentation>Signal indicates the price.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="priceMultiplier">
      <xs:annotation>
        <xs:documentation>Signal indicates the price multiplier. Extended price is the computed price value
multiplied by the number of units.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="priceRelative">
      <xs:annotation>
        <xs:documentation>Signal indicates the relative price.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="setpoint">
      <xs:annotation>
        <xs:documentation>Signal indicates a target amount of units.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="x-loadControlCapacity">
      <xs:annotation>
        <xs:documentation>This is an instruction for the load controller to operate at a level that is some
percentage of its maximum load consumption capacity. This can be mapped to specific load controllers to do things
like duty cycling. Note that 1.0 refers to 100% consumption. In the case of simple ON/OFF type devices then 0 =
OFF and 1 = ON.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="x-loadControlLevelOffset">
      <xs:annotation>
        <xs:documentation>Discrete integer levels that are relative to normal operations where 0 is normal
operations.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="x-loadControlPercentOffset">
      <xs:annotation>
        <xs:documentation>Percentage change from normal load control operations.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>

```

```

    <xs:enumeration value="x-loadControlSetpoint">
      <xs:annotation>
        <xs:documentation>Load controller set points.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
<xs:element name="responses" type="ei:ArrayofResponses"/>
<xs:complexType name="ArrayofResponses">
  <xs:annotation>
    <xs:documentation>Collection of Responses. When a service operation regards multiple referenceable
    items, each referenced item may have its own response. Always accompanied by an overall Response
    Type.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="response" type="ei:EiResponseType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="eventDescriptor" type="ei:eventDescriptorType">
  <xs:annotation>
    <xs:documentation>Information about the event</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="eventDescriptorType">
  <xs:sequence>
    <xs:element ref="ei:eventID"/>
    <xs:element ref="ei:modificationNumber"/>
    <xs:element name="modificationDateTime" type="xcal:DateTimeType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>When an event is modified</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="modificationReason" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Why an event was modified</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="priority" type="xs:unsignedInt" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The priority of the event in relation to other events (The lower the number higher
        the priority. A value of zero (0) indicates no priority, which is the lowest priority by default).</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="eiMarketContext">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="emix:marketContext"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element ref="ei:createdDateTime"/>
    <xs:element ref="ei:eventStatus">
      <xs:annotation>
        <xs:documentation>An indication of the event state: far, near, active, canceled,
        completed</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="testEvent" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Anything other than false indicates a test event</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="vtnComment" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Any text</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="signalPayload" type="ei:signalPayloadType" substitutionGroup="strm:streamPayloadBase">
  <xs:annotation>
    <xs:documentation>Signal values for events and baselines</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="signalPayloadType">
  <xs:complexContent>
    <xs:extension base="strm:StreamPayloadBaseType">

```

```

        <xs:choice>
            <xs:element ref="ei:payloadBase"/>
        </xs:choice>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="createdDateTime" type="xcal:DateTimeType">
    <xs:annotation>
        <xs:documentation>The dateTime the payload was created</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="eiTarget" type="ei:EiTargetType">
    <xs:annotation>
        <xs:documentation>Identifies the resources associated with the logical VEN interface. For events, the
values specified are the target for the event</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="EiTargetType">
    <xs:sequence>
        <xs:element ref="power:aggregatedPnode" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="power:endDeviceAsset" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="power:meterAsset" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="power:pnode" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="emix:serviceArea" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="power:serviceDeliveryPoint" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="power:serviceLocation" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="power:transportInterface" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ei:groupID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ei:groupName" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ei:resourceID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ei:venID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ei:partyID" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="eiEventSignal" type="ei:eiEventSignalType"/>
<xs:complexType name="eiEventSignalType">
    <xs:sequence>
        <xs:element ref="strm:intervals"/>
        <xs:element ref="ei:eiTarget" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Optionally identifies the device class associated with the signal. Only the
endDeviceAsset subelement is used</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="ei:signalName">
            <xs:annotation>
                <xs:documentation>Descriptive name for signal.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="ei:signalType"/>
        <xs:element name="signalID" type="xs:string">
            <xs:annotation>
                <xs:documentation>unique Identifier for a specific event signal</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="emix:itemBase" minOccurs="0">
            <xs:annotation>
                <xs:documentation>This is the unit of the signal.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element ref="ei:currentValue" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="signalName" type="ei:signalNameType"/>
<xs:simpleType name="signalNameType">
    <xs:annotation>
        <xs:documentation>Signal name.</xs:documentation>
    </xs:annotation>
    <xs:union memberTypes="ei:SignalNameEnumeratedType ei:EiExtensionTokenType"/>
</xs:simpleType>
<xs:element name="SignalNameEnumerated" type="ei:SignalNameEnumeratedType"/>
<xs:simpleType name="SignalNameEnumeratedType">
    <xs:restriction base="xs:token">
        <xs:enumeration value="SIMPLE">
            <xs:annotation>
                <xs:documentation>Simple levels (OpenADR 2.0a compliant)</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>

```

```

</xs:enumeration>
<xs:enumeration value="simple">
  <xs:annotation>
    <xs:documentation>deprecated – for backwards compatibility with A profile</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="ELECTRICITY_PRICE">
  <xs:annotation>
    <xs:documentation>This is the cost of electricity</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="ENERGY_PRICE">
  <xs:annotation>
    <xs:documentation>This is the cost of energy</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="DEMAND_CHARGE">
  <xs:annotation>
    <xs:documentation>This is the demand charge</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="BID_PRICE">
  <xs:annotation>
    <xs:documentation>This is the price that was bid by the resource</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="BID_LOAD">
  <xs:annotation>
    <xs:documentation>This is the amount of load that was bid by a resource into a
program</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="BID_ENERGY">
  <xs:annotation>
    <xs:documentation>This is the amount of energy from a resource that was bid into a
program</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="CHARGE_STATE">
  <xs:annotation>
    <xs:documentation>State of energy storage resource</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="LOAD_DISPATCH">
  <xs:annotation>
    <xs:documentation>This is used to dispatch load</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="LOAD_CONTROL">
  <xs:annotation>
    <xs:documentation>Set load output to relative values</xs:documentation>
  </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
<xs:element name="eiEventSignals" type="ei:eiEventSignalsType">
  <xs:annotation>
    <xs:documentation>Interval data for one or more event signals and/or baselines</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="eiEventSignalsType">
  <xs:sequence>
    <xs:element ref="ei:eiEventSignal" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Interval data for an event</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="ei:eiEventBaseline" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Interval data for a baseline</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="eiEventBaseline" type="ei:eiEventBaselineType">
  <xs:annotation>
    <xs:documentation>B profile</xs:documentation>
  </xs:annotation>

```

```

</xs:annotation>
</xs:element>
<xs:complexType name="eiEventBaselineType">
  <xs:sequence>
    <xs:element ref="xcal:dtstart"/>
    <xs:element ref="xcal:duration"/>
    <xs:element ref="strm:intervals"/>
    <xs:element name="baselineID" type="xs:string">
      <xs:annotation>
        <xs:documentation>Unique ID for a specific baseline</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="ei:resourceID" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="baselineName" type="xs:string">
      <xs:annotation>
        <xs:documentation>Descriptive name for baseline</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="emix:itemBase" minOccurs="0">
      <xs:annotation>
        <xs:documentation>This is the unit of the signal.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EiOptType">
  <xs:annotation>
    <xs:documentation>Opts are used by the VEN to temporarily override the pre-existing agreement. For
    example, a VEN may opt in to events during the evening, or opt out from events during the world
    series.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="ei:optID"/>
    <xs:element ref="ei:optType"/>
    <xs:element ref="ei:optReason"/>
    <xs:element ref="emix:marketContext" minOccurs="0"/>
    <xs:element ref="ei:venID"/>
    <xs:element ref="xcal:vavailability" minOccurs="0"/>
    <xs:element ref="ei:createdDateTime"/>
  </xs:sequence>
  <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<xs:element name="optReason" type="ei:OptReasonType">
  <xs:annotation>
    <xs:documentation>Enumerated value for the opt reason such as x-schedule</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:simpleType name="OptReasonType">
  <xs:annotation>
    <xs:documentation>Reason for opting.</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="ei:OptReasonEnumeratedType ei:EiExtensionTokenType"/>
</xs:simpleType>
<xs:element name="optReasonEnumerated" type="ei:OptReasonEnumeratedType"/>
<xs:simpleType name="OptReasonEnumeratedType">
  <xs:annotation>
    <xs:documentation>Enumerated reasons for opting.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:enumeration value="economic"/>
    <xs:enumeration value="emergency"/>
    <xs:enumeration value="mustRun"/>
    <xs:enumeration value="notParticipating"/>
    <xs:enumeration value="outageRunStatus"/>
    <xs:enumeration value="overrideStatus"/>
    <xs:enumeration value="participating"/>
    <xs:enumeration value="x-schedule"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="optID" type="xs:string">
  <xs:annotation>
    <xs:documentation>Identifier for an opt interaction</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="eiReportID" type="xs:string">
  <xs:annotation>
    <xs:documentation>Reference ID for a report</xs:documentation>
  </xs:annotation>
</xs:element>

```

```

    </xs:annotation>
  </xs:element>
  <xs:element name="reportRequestID" type="xs:string">
    <xs:annotation>
      <xs:documentation>Identifier for a particular report request</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="reportSpecifierID" type="xs:string">
    <xs:annotation>
      <xs:documentation>Identifier for a particular Metadata report specification</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="reportName" type="ei:reportNameType">
    <xs:annotation>
      <xs:documentation>Optional name for a report.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:simpleType name="reportNameType">
    <xs:union memberTypes="ei:reportNameEnumeratedType ei:EiExtensionTokenType"/>
  </xs:simpleType>
  <xs:simpleType name="reportNameEnumeratedType">
    <xs:restriction base="xs:token">
      <xs:enumeration value="METADATA_HISTORY_USAGE"/>
      <xs:enumeration value="HISTORY_USAGE"/>
      <xs:enumeration value="METADATA_HISTORY_GREENBUTTON"/>
      <xs:enumeration value="HISTORY_GREENBUTTON"/>
      <xs:enumeration value="METADATA_TELEMETRY_USAGE"/>
      <xs:enumeration value="TELEMETRY_USAGE"/>
      <xs:enumeration value="METADATA_TELEMETRY_STATUS"/>
      <xs:enumeration value="TELEMETRY_STATUS"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="rID" type="xs:string">
    <xs:annotation>
      <xs:documentation>ReferenceID for this data point</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="reportSubject" type="ei:EiTargetType">
    <xs:annotation>
      <xs:documentation>Device Class target – use only endDeviceAsset.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="reportDataSource" type="ei:EiTargetType">
    <xs:annotation>
      <xs:documentation>Sources for data in this report. Examples include meters or submeters. For example, if
a meter is capable of providing two different types of measurements, then each measurement stream would be
separately identified.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="statusDateTime" type="xcal:DateTimeType">
    <xs:annotation>
      <xs:documentation>Date and time this artifact references.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="reportType" type="ei:ReportTypeType"/>
  <xs:simpleType name="ReportTypeType">
    <xs:annotation>
      <xs:documentation>An enumerated value that gives the type of report being provided.</xs:documentation>
    </xs:annotation>
    <xs:union memberTypes="ei:ReportEnumeratedType ei:EiExtensionTokenType"/>
  </xs:simpleType>
  <xs:element name="reportEnumerated" type="ei:ReportEnumeratedType"/>
  <xs:simpleType name="ReportEnumeratedType">
    <xs:annotation>
      <xs:documentation>Enumerated report types</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:token">
      <xs:enumeration value="reading">
        <xs:annotation>
          <xs:documentation>Report indicates a reading, as from a meter. Readings are moments in time-
changes over time can be computed from the difference between successive readings. Payload type is
float</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="usage">
        <xs:annotation>

```

<xs:documentation>Report indicates an amount of units (denominated in ItemBase or in the EMIX Product) over a period. Payload type is Quantity. A typical ItemBase is Real Energy.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="demand">
 <xs:annotation>
 <xs:documentation>Report indicates an amount of units (denominated in ItemBase or in the EMIX Product). Payload type is Quantity. A typical ItemBase is Real Power.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="setPoint">
 <xs:annotation>
 <xs:documentation>Report indicates the amount (denominated in ItemBase or in the EMIX Product) currently set. May be a confirmation/return of the setpoint control value sent from the VTN. Payload type is Quantity. A typical ItemBase is Real Power.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="deltaUsage">
 <xs:annotation>
 <xs:documentation>Change in usage as compared to the baseline. See usage for more information</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="deltaSetPoint">
 <xs:annotation>
 <xs:documentation>Changes in setpoint from previous schedule.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="deltaDemand">
 <xs:annotation>
 <xs:documentation>Change in demand as compared to the baseline. See demand for more information</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="baseline">
 <xs:annotation>
 <xs:documentation>Can be demand or usage, as indicated by ItemBase. Indicates what [measurement] would be if not for the event or regulation. Report is of the format Baseline.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="deviation">
 <xs:annotation>
 <xs:documentation>Difference between some instruction and actual state.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="avgUsage">
 <xs:annotation>
 <xs:documentation>Average usage over the duration indicated by the Granularity. See usage for more information.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="avgDemand">
 <xs:annotation>
 <xs:documentation>Average usage over the duration indicated by the Granularity. See demand for more information.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="operatingState">
 <xs:annotation>
 <xs:documentation>Generalized state of a resource such as on/off, occupancy of building, etc. No ItemBase is relevant. Requires an Application Specific Payload Extension.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="upRegulationCapacityAvailable">
 <xs:annotation>
 <xs:documentation>Up Regulation capacity available for dispatch, expressed in EMIX Real Power. Payload is always expressed as positive Quantity.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="downRegulationCapacityAvailable">
 <xs:annotation>
 <xs:documentation>Down Regulation capacity available for dispatch, expressed in EMIX Real Power. Payload is always expressed as positive Quantity.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="regulationSetpoint">
 <xs:annotation>

```

        <xs:documentation>Regulation setpoint as instructed as part of regulation
services</xs:documentation>
    </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="storedEnergy">
        <xs:annotation>
            <xs:documentation>Stored Energy is expressed as Real Energy and Payload is expressed as a
Quantity.</xs:documentation>
        </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="targetEnergyStorage">
        <xs:annotation>
            <xs:documentation>Target Energy is expressed as Real Energy and Payload is expressed as a
Quantity.</xs:documentation>
        </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="availableEnergyStorage">
        <xs:annotation>
            <xs:documentation>Capacity available for further energy storage, perhaps to get to Target Energy
Storage</xs:documentation>
        </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="price">
        <xs:annotation>
            <xs:documentation>Price per ItemBase at each Interval</xs:documentation>
        </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="level">
        <xs:annotation>
            <xs:documentation>Simple level from market at each Interval. Itembase is not
relevant.</xs:documentation>
        </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="powerFactor">
        <xs:annotation>
            <xs:documentation>Power factor for the resource.</xs:documentation>
        </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="percentUsage">
        <xs:annotation>
            <xs:documentation>Percentage of usage.</xs:documentation>
        </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="percentDemand">
        <xs:annotation>
            <xs:documentation>Percentage of demand</xs:documentation>
        </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="x-resourceStatus">
        <xs:annotation>
            <xs:documentation>Percentage of demand</xs:documentation>
        </xs:annotation>
    </xs:enumeration>
    </xs:restriction>
</xs:simpleType>
<xs:element name="readingType" type="ei:ReadingTypeType">
    <xs:annotation>
        <xs:documentation>Metadata about the Readings, such as mean or derived</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:simpleType name="ReadingTypeType">
    <xs:annotation>
        <xs:documentation>Type of Reading.</xs:documentation>
    </xs:annotation>
    <xs:union memberTypes="ei:ReadingTypeEnumeratedType ei:EiExtensionTokenType"/>
</xs:simpleType>
<xs:element name="readingTypeEnumerated" type="ei:ReadingTypeEnumeratedType"/>
<xs:simpleType name="ReadingTypeEnumeratedType">
    <xs:restriction base="xs:token">
        <xs:enumeration value="Direct Read">
            <xs:annotation>
                <xs:documentation>Reading is read from a device that increases monotonically, and usage must be
computed from pairs of start and stop readings.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Net">
            <xs:annotation>

```

<xs:documentation>Meter or [resource] prepares its own calculation of total use over time.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="Allocated">
 <xs:annotation>
 <xs:documentation>Meter covers several [resources] and usage is inferred through some sort of pro data computation.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="Estimated">
 <xs:annotation>
 <xs:documentation>Used when a reading is absent in a series in which most readings are present.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="Summed">
 <xs:annotation>
 <xs:documentation>Several meters together provide the reading for this [resource]. This is specifically a different than aggregated, which refers to multiple [resources] in the same payload. See also Hybrid.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="Derived">
 <xs:annotation>
 <xs:documentation>Usage is inferred through knowledge of run-time, normal operation, etc.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="Mean">
 <xs:annotation>
 <xs:documentation>Reading is the mean value over the period indicated in Granularity</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="Peak">
 <xs:annotation>
 <xs:documentation>Reading is Peak (highest) value over the period indicated in granularity. For some measurements, it may make more sense as the lowest value. May not be consistent with aggregate readings. Only valid for flow-rate Item Bases, i.e. Power not Energy.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="Hybrid">
 <xs:annotation>
 <xs:documentation>If aggregated, refers to different reading types in the aggregate number.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="Contract">
 <xs:annotation>
 <xs:documentation>Indicates reading is pro forma, i.e. is reported at agreed upon rates</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="Projected">
 <xs:annotation>
 <xs:documentation>Indicates reading is in the future, and has not yet been measured.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="x-RMS">
 <xs:annotation>
 <xs:documentation>Root Mean Square</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="x-notApplicable">
 <xs:annotation>
 <xs:documentation>Not Applicable</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 </xs:restriction>
 </xs:simpleType>
 <xs:element name="confidence" type="ei:ConfidenceType"/>
 <xs:simpleType name="ConfidenceType">
 <xs:restriction base="xs:unsignedInt">
 <xs:minInclusive value="0"/>
 <xs:maxInclusive value="100"/>

```

    </xs:restriction>
  </xs:simpleType>
  <xs:element name="accuracy" type="ei:AccuracyType"/>
  <xs:simpleType name="AccuracyType">
    <xs:annotation>
      <xs:documentation>Number is in same units as the payload variable for an Interval. When present with
      Confidence, indicates the likely variability of the prediction. When present with ReadingType, indicates likely error
      of Reading.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:float"/>
  </xs:simpleType>
  <xs:complexType name="ReportPayloadType">
    <xs:annotation>
      <xs:documentation>Report Payload for use in Reports, snaps, and projections.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="strm:StreamPayloadBaseType">
        <xs:sequence>
          <xs:element ref="ei:rID">
            <xs:annotation>
              <xs:documentation>A reference to a metadata data point description</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element ref="ei:confidence" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Likely variability of prediction: 0-100</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element ref="ei:accuracy" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Accuracy in same units as interval payload value</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element ref="ei:payloadBase"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="numDataSources" type="xs:unsignedInt"/>
  <xs:element name="reportSpecifier" type="ei:ReportSpecifierType">
    <xs:annotation>
      <xs:documentation>Specify data points desired in a particular report instance</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="ReportSpecifierType">
    <xs:annotation>
      <xs:documentation>Parameters that define the content of a Report Stream</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="ei:reportSpecifierID"/>
      <xs:element ref="xcal:granularity">
        <xs:annotation>
          <xs:documentation>How frequently the [measurement] is to be recorded.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="reportBackDuration" type="xcal:DurationPropType">
        <xs:annotation>
          <xs:documentation>Report back with the Report-To-Date for each passing of this
          Duration.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="reportInterval" type="xcal:WsCalendarIntervalType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>This is the overall period of reporting.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element ref="ei:specifierPayload" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="specifierPayload" type="ei:SpecifierPayloadType"/>
  <xs:complexType name="SpecifierPayloadType">
    <xs:annotation>
      <xs:documentation>Payload for use in Report Specifiers.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="ei:rID"/>
      <xs:element ref="emix:itemBase" minOccurs="0">

```

```

        <xs:annotation>
          <xs:documentation>What is measured or tracked in this report (Units).</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element ref="ei:readingType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="registrationID" substitutionGroup="ei:refID">
    <xs:annotation>
      <xs:documentation>Identifier for Registration transaction. Not included in response to query registration
unless already registered</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="refID" substitutionGroup="ei:uid">
    <xs:annotation>
      <xs:documentation>Reference ID for a particular instance, transmittal, or artifact. Note: not the same as
the native ID of the object being transmitted or shared.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="uid" type="ei:UidType" abstract="true"/>
  <xs:simpleType name="UidType">
    <xs:annotation>
      <xs:documentation>Unique Identifier</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:attribute name="schemaVersion" type="ei:schemaVersionType"/>
  <xs:simpleType name="schemaVersionType">
    <xs:union memberTypes="ei:schemaVersionEnumeratedType ei:EiExtensionTokenType"/>
  </xs:simpleType>
  <xs:simpleType name="schemaVersionEnumeratedType">
    <xs:restriction base="xs:token">
      <xs:enumeration value="2.0a"/>
      <xs:enumeration value="2.0b"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

IECNORM.COM : Click to view the full PDF of IEC 62746-10-1:2018

oadr_emix_20b.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:emix="http://docs.oasis-
open.org/ns/emix/2011/06" xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:gmlsf="http://www.opengis.net/gmlsf/2.0"
xmlns:power="http://docs.oasis-open.org/ns/emix/2011/06/power" xmlns:ei="http://docs.oasis-
open.org/ns/energyinterop/201110" targetNamespace="http://docs.oasis-open.org/ns/emix/2011/06"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xs:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="oadr_gml_20b.xsd"/>
  <xs:element name="marketContext" type="emix:MarketContextType">
    <xs:annotation>
      <xs:documentation>A URI identifying a DR Program</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:simpleType name="MarketContextType">
    <xs:restriction base="xs:anyURI"/>
  </xs:simpleType>
  <xs:element name="serviceArea" type="emix:ServiceAreaType"/>
  <xs:complexType name="ServiceAreaType">
    <xs:annotation>
      <xs:documentation>The Service Area is the geographic region that is affected by the EMIX market
condition</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="gml:FeatureCollection"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="itemBase" type="emix:ItemBaseType" abstract="true"/>
  <xs:complexType name="ItemBaseType" abstract="true" mixed="false">
    <xs:annotation>
      <xs:documentation>Abstract base type for units for EMIX Product delivery, measurement, and
warrants.</xs:documentation>
    </xs:annotation>
  </xs:complexType>
</xs:schema>
```

IECNORM.COM : Click to view the full PDF of IEC 62746-10-1:2018

oadr_gml_20b.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema      xmlns:xs="http://www.w3.org/2001/XMLSchema"      xmlns:gml="http://www.opengis.net/gml/3.2"
targetNamespace="http://www.opengis.net/gml/3.2"      elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
  <xs:element name="FeatureCollection">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="location">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Polygon">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="exterior">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="LinearRing">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element ref="gml:posList"/>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:attribute ref="gml:id"/>
</xs:schema>

```

IECNORM.COM : Click to view the full PDF of IEC 62746-10-1:2018

oadr_greenbutton.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://naesb.org/espi" xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://naesb.org/espi"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="0.7.20121113">
  <xs:import namespace="http://www.w3.org/2005/Atom" schemaLocation="oadr_atom.xsd"/>
  <xs:complexType name="IntervalBlock">
    <xs:annotation>
      <xs:documentation>Time sequence of Readings of the same ReadingType.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="IdentifiedObject">
        <xs:sequence>
          <xs:element name="interval" type="DateTimeInterval" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Specifies the time period during which the contained readings were
taken.</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="IntervalReading" type="IntervalReading" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="MeterReading">
    <xs:annotation>
      <xs:documentation>Set of values obtained from the meter.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="IdentifiedObject">
        </xs:complexContent>
      </xs:complexType>
    </xs:complexType>
    <xs:complexType name="ReadingType">
      <xs:annotation>
        <xs:documentation>Characteristics associated with all Readings included in a
MeterReading.</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
        <xs:extension base="IdentifiedObject">
          <xs:sequence>
            <xs:element name="accumulationBehaviour" type="AccumulationKind" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Code indicating how value is accumulated over time for Readings of
ReadingType.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="commodity" type="CommodityKind" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Code for commodity classification of Readings of
ReadingType.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="consumptionTier" type="Int16" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Code for consumption tier associated with a Reading of
ReadingType.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="currency" type="Currency" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Code for the currency for costs associated with this ReadingType. The valid
values per the standard are defined in CurrencyCode.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="dataQualifier" type="DataQualifierKind" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Code describing a salient attribute of Readings of
ReadingType.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="defaultQuality" type="QualityOfReading" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Default value to be used if no value of ReadingQuality.quality is provided.

```

Specific format and valid values per the standard are specified in QualityOfReading.</xs:documentation>

```

</xs:annotation>
</xs:element>
<xs:element name="flowDirection" type="FlowDirectionKind" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Direction associated with current related Readings.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="intervalLength" type="UInt32" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Default interval length specified in seconds for Readings of
ReadingType.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="kind" type="MeasurementKind" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Code for general classification of a Reading of ReadingType.
</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="phase" type="PhaseCodeKind" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Code for phase information associated with Readings of
ReadingType.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="powerOfTenMultiplier" type="UnitMultiplierKind" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Code for the power of ten multiplier which, when used in combination with
the uom, specifies the actual unit of measure for Readings of ReadingType.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="timeAttribute" type="TimePeriodOfInterest" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Code used to specify a particular type of time interval method for Readings
of ReadingType.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="tou" type="Int16" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Code for the TOU type of Readings of ReadingType.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="uom" type="UnitSymbolKind" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Code for the base unit of measure for Readings of ReadingType. Used in
combination with the powerOfTenMultiplier to specify the actual unit of measure</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="cpp" type="Int16" minOccurs="0">
  <xs:annotation>
    <xs:documentation>[extension] Critical peak period (CPP) bucket the reading value is attributed
to. Value 0 means not applicable. Even though CPP is usually considered a specialized form of time of use 'tou',
this attribute is defined explicitly for flexibility.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="interharmonic" type="ReadingInterharmonic" minOccurs="0">
  <xs:annotation>
    <xs:documentation>[extension] Indication of a "harmonic" or "interharmonic" basis for the
measurement. Value 0 in 'numerator' and 'denominator' means not applicable.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="measuringPeriod" type="TimeAttributeKind" minOccurs="0">
  <xs:annotation>
    <xs:documentation>[extension] Time attribute inherent or fundamental to the reading value (as
opposed to 'macroPeriod' that supplies an "adjective" to describe aspects of a time period with regard to the
measurement). It refers to the way the value was originally measured and not to the frequency at which it is
reported or presented. For example, an hourly interval of consumption data would have value 'hourly' as an
attribute. However in the case of an hourly sampled voltage value, the meterReadings schema would carry the
'hourly' interval size information.
It is common for meters to report demand in a form that is measured over the course of a portion of an hour, while
enterprise applications however commonly assume the demand (in kW or kVAr) normalised to 1 hour. The sytem
that receives readings directly from the meter therefore must perform this transformation before publishing
readings for use by the other enterprise systems. The scalar used is chosen based on the block size (not any sub-
interval size).</xs:documentation>
  </xs:annotation>
</xs:element>

```

```

    <xs:element name="argument" type="RationalNumber" minOccurs="0">
      <xs:annotation>
        <xs:documentation>[extension] Argument used to introduce numbers into the unit of measure
        description where they are needed (e.g. 4 where the measure needs an argument such as CEMI(n=4)). Most
        arguments used in practice however will be integers (i.e. 'denominator'=1).
        Value 0 in 'numerator' and 'denominator' means not applicable.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="UsagePoint">
  <xs:annotation>
    <xs:documentation>Logical point on a network at which consumption or production is either physically
    measured (e.g. metered) or estimated (e.g. unmetered street lights).</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="IdentifiedObject">
      <xs:sequence>
        <xs:element name="roleFlags" type="HexBinary16" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Specifies the roles that this usage point has been assigned. Bit 1 – isMirror
            Bit 2 – isPremisesAggregationPoint Bit 3 – isPEV Bit 4 – isDER Bit 5 – isRevenueQuality Bit 6 – isDC Bit 7-16 –
            Reserved</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="ServiceCategory" type="ServiceCategory" minOccurs="0"/>
        <xs:element name="status" type="UInt8" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Specifies the current status of this usage point. Valid values include: 0 = off
            1 = on</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="ServiceDeliveryPoint" type="ServiceDeliveryPoint" minOccurs="0">
          <xs:annotation>
            <xs:documentation>[extension] Contains service delivery point information about the UsagePoint
            if it does represent the service delivery point.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ElectricPowerQualitySummary">
  <xs:annotation>
    <xs:documentation>A summary of power quality events. This information represents a summary of power
    quality information typically required by customer facility energy management systems. It is not intended to satisfy
    the detailed requirements of power quality monitoring. All values are as defined by measurementProtocol during
    the period. The standards typically also give ranges of allowed values; the information attributes are the raw
    measurements, not the &quot;yes/no&quot; determination by the various standards. See referenced standards for
    definition, measurement protocol and period.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="IdentifiedObject">
      <xs:sequence>
        <xs:element name="flickerPlt" type="Int48" minOccurs="0">
          <xs:annotation>
            <xs:documentation>A measurement of long term Rapid Voltage Change in hundredths of a Volt.
            flickerPlt is derived from 2 hours of Pst values (12 values combined in cubic relationship).</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="flickerPst" type="Int48" minOccurs="0">
          <xs:annotation>
            <xs:documentation>flickerPst is a value measured over 10 minutes that characterizes the
            likelihood that the voltage fluctuations would result in perceptible light flicker. A value of 1.0 is designed to
            represent the level that 50% of people would perceive flicker in a 60 watt incandescent bulb. The value reported is
            represented as an integer in hundredths.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="harmonicVoltage" type="Int48" minOccurs="0">
          <xs:annotation>
            <xs:documentation>A measurement of the Harmonic Voltage during the period. For DC,
            distortion is with respect to a signal of zero Hz.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="longInterruptions" type="Int48" minOccurs="0">

```

```

        <xs:annotation>
            <xs:documentation>A count of Long Interruption events (as defined by measurementProtocol)
            during the summary interval period.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="mainsVoltage" type="Int48" minOccurs="0">
        <xs:annotation>
            <xs:documentation>A measurement of the Mains [Signaling] Voltage during the summary
            interval period in uV.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="measurementProtocol" type="UInt8" minOccurs="0">
        <xs:annotation>
            <xs:documentation>A reference to the source standard used as the measurement protocol
            definition. Examples are: 0 = &quot;IEEE1519-2009&quot;; 1 = &quot;EN50160&quot;</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="powerFrequency" type="Int48" minOccurs="0">
        <xs:annotation>
            <xs:documentation>A measurement of the power frequency during the summary interval period
            in uHz.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="rapidVoltageChanges" type="Int48" minOccurs="0">
        <xs:annotation>
            <xs:documentation>A count of Rapid Voltage Change events during the summary interval
            period</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="shortInterruptions" type="Int48" minOccurs="0">
        <xs:annotation>
            <xs:documentation>A count of Short Interruption events during the summary interval
            period</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="summaryInterval" type="DateTimeInterval">
        <xs:annotation>
            <xs:documentation>Interval of summary period</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="supplyVoltageDips" type="Int48" minOccurs="0">
        <xs:annotation>
            <xs:documentation>A count of Supply Voltage Dip events during the summary interval
            period</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="supplyVoltageImbalance" type="Int48" minOccurs="0">
        <xs:annotation>
            <xs:documentation>A count of Supply Voltage Imbalance events during the summary interval
            period</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="supplyVoltageVariations" type="Int48" minOccurs="0">
        <xs:annotation>
            <xs:documentation>A count of Supply Voltage Variations during the summary interval
            period</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="tempOvervoltage" type="Int48" minOccurs="0">
        <xs:annotation>
            <xs:documentation>A count of Temporary Overvoltage events (as defined by
            measurementProtocol) during the summary interval period</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexType>
<xs:complexType name="ElectricPowerUsageSummary">
    <xs:annotation>
        <xs:documentation>Summary of usage for a billing period</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="IdentifiedObject">
            <xs:sequence>
                <xs:element name="billingPeriod" type="DateTimeInterval" minOccurs="0">
                    <xs:annotation>

```

<xs:documentation>The billing period to which the included measurements apply</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="billLastPeriod" type="Int48" minOccurs="0">
 <xs:annotation>
 <xs:documentation>The amount of the bill for the previous billing period, in hundred-thousandths of the currency specified in the ReadingType for this reading (e.g. 840 = USD, US dollar).</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="billToDate" type="Int48" minOccurs="0">
 <xs:annotation>
 <xs:documentation>The bill amount related to the billing period as of the date received, in hundred-thousandths of the currency specified in the ReadingType for this reading. (e.g. 840 = USD, US dollar).</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="costAdditionalLastPeriod" type="Int48" minOccurs="0">
 <xs:annotation>
 <xs:documentation>Additional charges from the last billing period, in hundred-thousandths of the currency specified in the ReadingType for this reading. (e.g. 840 = USD, US dollar).</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="costAdditionalDetailLastPeriod" type="LineItem" minOccurs="0" maxOccurs="unbounded">
 <xs:annotation>
 <xs:documentation>[extension] Additional charges from the last billing period which in total add up to costAdditionalLastPeriod</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="currency" type="Currency" minOccurs="0">
 <xs:annotation>
 <xs:documentation>The ISO 4217 code indicating the currency applicable to the bill amounts in the summary. See list at <http://tiny.cc/4217></xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="overallConsumptionLastPeriod" type="SummaryMeasurement" minOccurs="0">
 <xs:annotation>
 <xs:documentation>[extension] The amount of energy consumed in the last billing period.</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="currentBillingPeriodOverAllConsumption" type="SummaryMeasurement" minOccurs="0">
 <xs:annotation>
 <xs:documentation>The total consumption for the billing period</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="currentDayLastYearNetConsumption" type="SummaryMeasurement" minOccurs="0">
 <xs:annotation>
 <xs:documentation>The amount of energy consumed one year ago interpreted as same day of week same week of year (see ISO 8601).</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="currentDayNetConsumption" type="SummaryMeasurement" minOccurs="0">
 <xs:annotation>
 <xs:documentation>Net consumption for the current day (delivered – received)</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="currentDayOverallConsumption" type="SummaryMeasurement" minOccurs="0">
 <xs:annotation>
 <xs:documentation>Overall energy consumption for the current day</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="peakDemand" type="SummaryMeasurement" minOccurs="0">
 <xs:annotation>
 <xs:documentation>Peak demand recorded for the current period</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="previousDayLastYearOverallConsumption" type="SummaryMeasurement" minOccurs="0">
 <xs:annotation>
 <xs:documentation>The amount of energy consumed on the previous day one year ago interpreted as same day of week same week of year (see ISO 8601).</xs:documentation>

```

        </xs:annotation>
    </xs:element>
    <xs:element name="previousDayNetConsumption" type="SummaryMeasurement" minOccurs="0">
        <xs:annotation>
            <xs:documentation>Net consumption for the previous day</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="previousDayOverallConsumption" type="SummaryMeasurement" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The total consumption for the previous day</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="qualityOfReading" type="QualityOfReading" minOccurs="0">
        <xs:annotation>
            <xs:documentation>Indication of the quality of the summary readings</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="ratchetDemand" type="SummaryMeasurement" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The current ratchet demand value for the ratchet demand
period</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="ratchetDemandPeriod" type="DateTimeInterval" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The period over which the ratchet demand applies</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="statusTimeStamp" type="TimeType">
        <xs:annotation>
            <xs:documentation>Date/Time status of this UsageSummary</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="TimeConfiguration">
    <xs:annotation>
        <xs:documentation>[extension] Contains attributes related to the configuration of the time
service.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="IdentifiedObject">
            <xs:sequence>
                <xs:element name="dstEndRule" type="DstRuleType">
                    <xs:annotation>
                        <xs:documentation>Rule to calculate end of daylight savings time in the current year. Result of
dstEndRule must be greater than result of dstStartRule.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="dstOffset" type="TimeType">
                    <xs:annotation>
                        <xs:documentation>Daylight savings time offset from local standard time.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="dstStartRule" type="DstRuleType">
                    <xs:annotation>
                        <xs:documentation>Rule to calculate start of daylight savings time in the current year. Result of
dstEndRule must be greater than result of dstStartRule.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="tzOffset" type="TimeType">
                    <xs:annotation>
                        <xs:documentation>Local time zone offset from UTCTime. Does not include any daylight savings
time offsets.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="IntervalReading">
    <xs:annotation>
        <xs:documentation>Specific value measured by a meter or other asset. Each Reading is associated with a
specific ReadingType.</xs:documentation>
    </xs:annotation>

```

```

<xs:complexContent>
  <xs:extension base="Object">
    <xs:sequence>
      <xs:element name="cost" type="Int48" minOccurs="0">
        <xs:annotation>
          <xs:documentation>[correction] Specifies a cost associated with this reading, in hundred-
thousandths of the currency specified in the ReadingType for this reading. (e.g. 840 = USD, US
dollar)</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ReadingQuality" type="ReadingQuality" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="timePeriod" type="DateTimeInterval" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The date time and duration of a reading. If not specified, readings for each
          &quot;intervalLength&quot; in ReadingType are present.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="value" type="Int48" minOccurs="0">
        <xs:annotation>
          <xs:documentation>[correction] Value in units specified by ReadingType</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ReadingQuality">
  <xs:annotation>
    <xs:documentation>Quality of a specific reading value or interval reading value. Note that more than one
Quality may be applicable to a given Reading. Typically not used unless problems or unusual conditions occur (i.e.
quality for each Reading is assumed to be &#39;Good&#39; (valid) unless stated otherwise in associated
ReadingQuality).</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Object">
      <xs:sequence>
        <xs:element name="quality" type="QualityOfReading">
          <xs:annotation>
            <xs:documentation>Quality, to be specified if different than ReadingType.defaultQuality. The
specific format is specified per the standard is defined in QualityOfReading.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ServiceCategory">
  <xs:annotation>
    <xs:documentation>Category of service provided to the customer.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Object">
      <xs:sequence>
        <xs:element name="kind" type="ServiceKind">
          <xs:annotation>
            <xs:documentation>Service classification Examples are: 0 – electricity 1 – gas The list of
specific valid values per the standard are itemized in ServiceKind.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SummaryMeasurement">
  <xs:annotation>
    <xs:documentation>An aggregated summary measurement reading.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Object">
      <xs:sequence>
        <xs:element name="powerOfTenMultiplier" type="UnitMultiplierKind" minOccurs="0">
          <xs:annotation>
            <xs:documentation>The multiplier part of the unit of measure, e.g. &quot;kilo&quot;
(k)</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="timeStamp" type="TimeType" minOccurs="0">

```

```

        <xs:annotation>
          <xs:documentation>The date and time (if needed) of the summary
measurement.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="uom" type="UnitSymbolKind" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The units of the reading, e.g. &quot;Wh&quot;</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="value" type="Int48" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The value of the summary measurement.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="BatchItemInfo">
  <xs:annotation>
    <xs:documentation>Includes elements that make it possible to include multiple transactions in a single
(batch) request.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Object">
      <xs:sequence>
        <xs:element name="name" type="HexBinary16" minOccurs="0">
          <xs:annotation>
            <xs:documentation>An identifier for this object that is only unique within the containing
collection.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="operation" type="CRUDOperation" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Specifies the operation requested of this item.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="statusCode" type="StatusCode" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Indicates the status code of the associated transaction.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="statusReason" type="String256" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Indicates the reason for the indicated status code.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ServiceDeliveryPoint">
  <xs:annotation>
    <xs:documentation>[extension] Service Delivery Point is representation of revenue UsagePoint
attributes</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Object">
      <xs:sequence>
        <xs:element name="name" type="String256" minOccurs="0">
          <xs:annotation>
            <xs:documentation>The name is any free human readable and possibly non unique text naming
the object.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="tariffProfile" type="String256" minOccurs="0">
          <xs:annotation>
            <xs:documentation>A schedule of charges; structure associated with Tariff that allows the
definition of complex tariff structures such as step and time of use.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="customerAgreement" type="String256" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Agreement between the customer and the ServiceSupplier to pay for service
at a specific service location. It provides for the recording of certain billing information about the type of service

```

provided at the service location and is used during charge creation to determine the type of service.

```

</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:simpleType name="HexBinary128">
  <xs:annotation>
    <xs:documentation>A 128-bit field encoded as a hex string (32 characters / 16 octets)</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:hexBinary">
    <xs:maxLength value="16"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="HexBinary32">
  <xs:annotation>
    <xs:documentation>A 32-bit field encoded as a hex string (8 characters / 4 octets)</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:hexBinary">
    <xs:maxLength value="4"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="HexBinary16">
  <xs:annotation>
    <xs:documentation>A 16-bit field encoded as a hex string (4 characters / 2 octets)</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:hexBinary">
    <xs:maxLength value="2"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="HexBinary8">
  <xs:annotation>
    <xs:documentation>An 8-bit field encoded as a hex string (2 characters / 1 octets)</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:hexBinary">
    <xs:maxLength value="1"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="String256">
  <xs:annotation>
    <xs:documentation>Character string of max length 256</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:maxLength value="256"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="String32">
  <xs:annotation>
    <xs:documentation>Character string of max length 32</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:maxLength value="32"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="String64">
  <xs:annotation>
    <xs:documentation>Character string of max length 64</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:maxLength value="64"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="UInt16">
  <xs:annotation>
    <xs:documentation>Unsigned integer, max inclusive 65535 (2^16-1), same as
xs:unsignedShort</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:unsignedShort"/>
</xs:simpleType>
<xs:simpleType name="UInt32">
  <xs:annotation>
    <xs:documentation>Unsigned integer, max inclusive 4294967295 (2^32-1), same as
xs:unsignedInt</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:unsignedInt"/>

```

```

</xs:simpleType>
<xs:simpleType name="UInt48">
  <xs:annotation>
    <xs:documentation>Unsigned integer, max inclusive 281474976710655 (2^48-1), restriction of
xs:unsignedLong</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:unsignedLong">
    <xs:maxInclusive value="281474976710655"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="UInt8">
  <xs:annotation>
    <xs:documentation>Unsigned integer, max inclusive 255 (2^8-1), same as
xs:unsignedByte</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:unsignedByte"/>
</xs:simpleType>
<xs:simpleType name="Int48">
  <xs:annotation>
    <xs:documentation>Signed integer, max inclusive -140737488355327 to +140737488355327 (2^47-1),
restriction of xs:long</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:long">
    <xs:minInclusive value="-140737488355328"/>
    <xs:maxInclusive value="140737488355328"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Int16">
  <xs:annotation>
    <xs:documentation>Signed integer, max inclusive (2^16-1), restriction of xs:short</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:short"/>
</xs:simpleType>
<xs:simpleType name="TimeType">
  <xs:annotation>
    <xs:documentation>Time is a signed 64-bit value representing the number of seconds since 0 hours, 0
minutes, 0 seconds, on the 1st of January, 1970.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:long"/>
</xs:simpleType>
<xs:complexType name="DateTimeInterval">
  <xs:annotation>
    <xs:documentation>Interval of date and time. End is not included because it can be derived from the start
and the duration.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Object">
      <xs:sequence>
        <xs:element name="duration" type="UInt32" minOccurs="0">
          <xs:annotation>
            <xs:documentation>[correction] Duration of the interval, in seconds.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="start" type="TimeType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>[correction] Date and time that this interval started.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="IdentifiedObject">
  <xs:annotation>
    <xs:documentation>This is a root class to provide common naming attributes for all classes needing
naming attributes</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Object">
      <xs:sequence>
        <xs:element name="batchItemInfo" type="BatchItemInfo" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="UUIDType">
  <xs:annotation>

```

```

    <xs:documentation>This pattern defines a UUID </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="Object">
  <xs:annotation>
    <xs:documentation>Superclass of all object classes to allow extensions.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="extension" type="xs:anyType" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Contains an extension.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceStatus">
  <xs:annotation>
    <xs:documentation>Contains the current status of the service.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Object">
      <xs:sequence>
        <xs:element name="currentStatus" type="ESPIServiceStatus">
          <xs:annotation>
            <xs:documentation>The current status of the service.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="RationalNumber">
  <xs:annotation>
    <xs:documentation>[extension] Rational number = 'numerator' / 'denominator'.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="numerator" type="xs:integer" minOccurs="0"/>
    <xs:element name="denominator" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ReadingInterharmonic">
  <xs:annotation>
    <xs:documentation>[extension] Interharmonics are represented as a rational number 'numerator' /
'denominator', and harmonics are represented using the same mechanism and identified by
'denominator'=1.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="numerator" type="xs:integer" minOccurs="0"/>
    <xs:element name="denominator" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="LineItem">
  <xs:annotation>
    <xs:documentation>[extension] Line item of detail for additional cost</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="amount" type="Int48"/>
    <xs:element name="rounding" type="Int48" minOccurs="0"/>
    <xs:element name="dateTime" type="TimeType"/>
    <xs:element name="note" type="String256"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="AccumulationKind">
  <xs:annotation>
    <xs:documentation>Code indicating how value is accumulated over time for Readings of ReadingType. The
list of valid values per the standard are defined in AccumulationBehaviorType.</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="0">
          <xs:annotation>
            <xs:appinfo>none</xs:appinfo>
            <xs:documentation>Not Applicable, or implied by the unit of measure.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>

```

```

</xs:annotation>
</xs:enumeration>
<xs:enumeration value="1">
  <xs:annotation>
    <xs:appinfo>bulkQuantity</xs:appinfo>
    <xs:documentation>A value from a register which represents the bulk quantity of a commodity.
    This quantity is computed as the integral of the commodity usage rate. This value is typically used as the basis for
    the dial reading at the meter, and as a result, will roll over upon reaching a maximum dial value. Note 1: With the
    metering system, the roll-over behaviour typically implies a roll-under behaviour so that the value presented is
    always a positive value (e.g. unsigned integer or positive decimal.) However, when communicating data between
    enterprise applications a negative value might occur in a case such as net metering. Note 2: A BulkQuantity refers
    primarily to the dial reading and not the consumption over a specific period of time.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="2">
  <xs:annotation>
    <xs:appinfo>continuousCumulative</xs:appinfo>
    <xs:documentation>The sum of the previous billing period values and the present period value.
    Note: "ContinuousCumulative" is commonly used in conjunction with "demand." The "ContinuousCumulative
    Demand" would be the cumulative sum of the previous billing period maximum demand values (as occurring with
    each demand reset) summed with the present period maximum demand value (which has yet to be
    reset.)</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="3">
  <xs:annotation>
    <xs:appinfo>cumulative</xs:appinfo>
    <xs:documentation>The sum of the previous billing period values. Note: "Cumulative" is
    commonly used in conjunction with "demand." Each demand reset causes the maximum demand value for the
    present billing period (since the last demand reset) to accumulate as an accumulative total of all maximum
    demands. So instead of "zeroing" the demand register, a demand reset has the affect of adding the present
    maximum demand to this accumulating total.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="4">
  <xs:annotation>
    <xs:appinfo>deltaData</xs:appinfo>
    <xs:documentation>The difference between the value at the end of the prescribed interval and
    the beginning of the interval. This is used for incremental interval data. Note: One common application would be for
    load profile data, another use might be to report the number of events within an interval (such as the number of
    equipment energizations within the specified period of time.)</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="6">
  <xs:annotation>
    <xs:appinfo>indicating</xs:appinfo>
    <xs:documentation>As if a needle is swung out on the meter face to a value to indicate the
    current value. (Note: An "indicating" value is typically measured over hundreds of milliseconds or greater, or may
    imply a "pusher" mechanism to capture a value. Compare this to "instantaneous" which is measured over a shorter
    period of time.)</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="9">
  <xs:annotation>
    <xs:appinfo>summation</xs:appinfo>
    <xs:documentation>A form of accumulation which is selective with respect to time. Note:
    "Summation" could be considered a specialization of "Bulk Quantity" according to the rules of inheritance where
    "Summation" selectively accumulates pulses over a timing pattern, and "BulkQuantity" accumulates pulses all of
    the time.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="10">
  <xs:annotation>
    <xs:appinfo>timeDelay</xs:appinfo>
    <xs:documentation>A form of computation which introduces a time delay characteristic to the
    data value</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="12">
  <xs:annotation>
    <xs:appinfo>instantaneous</xs:appinfo>
    <xs:documentation>Typically measured over the fastest period of time allowed by the definition
    of the metric (usually milliseconds or tens of milliseconds.) (Note: "Instantaneous" was moved to attribute #3 in
    61968-9Ed2 from attribute #1 in 61968-9Ed1.)</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="13">

```

```

    <xs:annotation>
      <xs:appinfo>latchingQuantity</xs:appinfo>
      <xs:documentation>When this description is applied to a metered value, it implies that the value
is a time-independent cumulative quantity much a BulkQuantity, except that it latches upon the maximum value
upon reaching that value. Any additional accumulation (positive or negative) is discarded until a reset occurs. Note:
A LatchingQuantity may also occur in the downward direction – upon reaching a minimum value. The terms
"maximum" or "minimum" will usually be included as an attribute when this type of accumulation behaviour is
present. When this description is applied to an encoded value (UOM= "Code"), it implies that the value has one or
more bits which are latching. The condition that caused the bit to be set may have long since evaporated. In either
case, the timestamp that accompanies the value may not coincide with the moment the value was initially set. In
both cases a system will need to perform an operation to clear the latched value.</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="14">
    <xs:annotation>
      <xs:appinfo>boundedQuantity</xs:appinfo>
      <xs:documentation>A time-independent cumulative quantity much a BulkQuantity or a
LatchingQuantity, except that the accumulation stops at the maximum or minimum values. When the maximum is
reached, any additional positive accumulation is discarded, but negative accumulation may be accepted (thus
lowering the counter.) Likewise, when the negative bound is reached, any additional negative accumulation is
discarded, but positive accumulation is accepted (thus increasing the counter.)</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="CommodityKind">
  <xs:annotation>
    <xs:documentation>Code for commodity classification of Readings of ReadingType. The valid values per
the standard are defined in CommodityType.</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="0">
          <xs:annotation>
            <xs:appinfo>none</xs:appinfo>
            <xs:documentation>Not Applicable</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="1">
          <xs:annotation>
            <xs:appinfo>electricity SecondaryMetered</xs:appinfo>
            <xs:documentation>All types of metered quantities. This type of reading comes from the meter
and represents a "secondary" metered value.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
          <xs:annotation>
            <xs:appinfo>electricity PrimaryMeter</xs:appinfo>
            <xs:documentation>It is possible for a meter to be outfitted with an external VT and/or CT. The
meter might not be aware of these devices, and the display not compensate for their presence. Ultimately, when
these scalars are applied, the value that represents the service value is called the "primary metered" value. The
"index" in sub-category 3 mirrors those of sub-category 0.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="3">
          <xs:annotation>
            <xs:appinfo>communication</xs:appinfo>
            <xs:documentation>A measurement of the communication infrastructure
itself.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="4">
          <xs:annotation>
            <xs:appinfo>air</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="5">
          <xs:annotation>
            <xs:appinfo>insulativeGas</xs:appinfo>
            <xs:documentation>(SF6 is found separately below.)</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="6">

```

```

<xs:annotation>
  <xs:appinfo>insulativeOil</xs:appinfo>
  <xs:documentation/>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="7">
  <xs:annotation>
    <xs:appinfo>naturalGas</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="8">
  <xs:annotation>
    <xs:appinfo>propane</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="9">
  <xs:annotation>
    <xs:appinfo>potableWater</xs:appinfo>
    <xs:documentation>Drinkable water</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="10">
  <xs:annotation>
    <xs:appinfo>steam</xs:appinfo>
    <xs:documentation>Water in steam form, usually used for heating.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="11">
  <xs:annotation>
    <xs:appinfo>wasteWater</xs:appinfo>
    <xs:documentation>(Sewerage)</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="12">
  <xs:annotation>
    <xs:appinfo>heatingFluid</xs:appinfo>
    <xs:documentation>This fluid is likely in liquid form. It is not necessarily water or water based.
The warm fluid returns cooler than when it was sent. The heat conveyed may be metered.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="13">
  <xs:annotation>
    <xs:appinfo>coolingFluid</xs:appinfo>
    <xs:documentation>The cool fluid returns warmer than when it was sent. The heat conveyed
may be metered.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="14">
  <xs:annotation>
    <xs:appinfo>nonpotableWater</xs:appinfo>
    <xs:documentation>Reclaimed water – possibly used for irrigation but not sufficiently treated to
be considered safe for drinking.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="15">
  <xs:annotation>
    <xs:appinfo>nox</xs:appinfo>
    <xs:documentation>Nitrous Oxides NOX</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="16">
  <xs:annotation>
    <xs:appinfo>so2</xs:appinfo>
    <xs:documentation>Sulfur Dioxide SO2</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="17">
  <xs:annotation>
    <xs:appinfo>ch4</xs:appinfo>
    <xs:documentation>Methane CH4</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="18">
  <xs:annotation>
    <xs:appinfo>co2</xs:appinfo>

```

```

        <xs:documentation>Carbon Dioxide CO2</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="19">
      <xs:annotation>
        <xs:appinfo>carbon</xs:appinfo>
        <xs:documentation></xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="20">
      <xs:annotation>
        <xs:appinfo>hch</xs:appinfo>
        <xs:documentation>Hexachlorocyclohexane HCH</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="21">
      <xs:annotation>
        <xs:appinfo>pfc</xs:appinfo>
        <xs:documentation>Perfluorocarbons PFC</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="22">
      <xs:annotation>
        <xs:appinfo>sf6</xs:appinfo>
        <xs:documentation>Sulfurhexafluoride SF6</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="23">
      <xs:annotation>
        <xs:appinfo>tvLicence</xs:appinfo>
        <xs:documentation>Television</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="24">
      <xs:annotation>
        <xs:appinfo>internet</xs:appinfo>
        <xs:documentation>Internet service</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="25">
      <xs:annotation>
        <xs:appinfo>refuse</xs:appinfo>
        <xs:documentation>trash</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="Currency">
  <xs:annotation>
    <xs:documentation>Code for the currency for costs associated with this ReadingType. The valid values
per the standard are defined in CurrencyCode.</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="840">
          <xs:annotation>
            <xs:appinfo>USD</xs:appinfo>
            <xs:documentation>US dollar</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="978">
          <xs:annotation>
            <xs:appinfo>EUR</xs:appinfo>
            <xs:documentation>European euro</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="36">
          <xs:annotation>
            <xs:appinfo>AUD</xs:appinfo>
            <xs:documentation>Australian dollar</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="124">
          <xs:annotation>

```

```

        <xs:appinfo>CAD</xs:appinfo>
        <xs:documentation>Canadian dollar</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="756">
    <xs:annotation>
        <xs:appinfo>CHF</xs:appinfo>
        <xs:documentation>Swiss francs</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="156">
    <xs:annotation>
        <xs:appinfo>CNY</xs:appinfo>
        <xs:documentation>Chinese yuan renminbi</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="208">
    <xs:annotation>
        <xs:appinfo>DKK</xs:appinfo>
        <xs:documentation>Danish crown</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="826">
    <xs:annotation>
        <xs:appinfo>GBP</xs:appinfo>
        <xs:documentation>British pound</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="392">
    <xs:annotation>
        <xs:appinfo>JPY</xs:appinfo>
        <xs:documentation>Japanese yen</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="578">
    <xs:annotation>
        <xs:appinfo>NOK</xs:appinfo>
        <xs:documentation>Norwegian crown</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="643">
    <xs:annotation>
        <xs:appinfo>RUB</xs:appinfo>
        <xs:documentation>Russian ruble</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="752">
    <xs:annotation>
        <xs:appinfo>SEK</xs:appinfo>
        <xs:documentation>Swedish crown</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="356">
    <xs:annotation>
        <xs:appinfo>INR</xs:appinfo>
        <xs:documentation>India rupees</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="0">
    <xs:annotation>
        <xs:appinfo>other</xs:appinfo>
        <xs:documentation>Another type of currency.</xs:documentation>
    </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="DataQualifierKind">
    <xs:annotation>
        <xs:documentation>Code describing a salient attribute of Readings of ReadingType. Valid values per the
standard are defined in DataQualifierType.</xs:documentation>
    </xs:annotation>
    <xs:union memberTypes="UInt16">
        <xs:simpleType>
            <xs:restriction base="UInt16">
                <xs:enumeration value="0">

```

```

    <xs:annotation>
      <xs:appinfo>none</xs:appinfo>
      <xs:documentation>Not Applicable</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="2">
    <xs:annotation>
      <xs:appinfo>average</xs:appinfo>
      <xs:documentation>Average value</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="4">
    <xs:annotation>
      <xs:appinfo>excess</xs:appinfo>
      <xs:documentation>The value represents an amount over which a threshold was
exceeded.</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="5">
    <xs:annotation>
      <xs:appinfo>highThreshold</xs:appinfo>
      <xs:documentation>The value represents a programmed threshold.</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="7">
    <xs:annotation>
      <xs:appinfo>lowThreshold</xs:appinfo>
      <xs:documentation>The value represents a programmed threshold.</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="8">
    <xs:annotation>
      <xs:appinfo>maximum</xs:appinfo>
      <xs:documentation>The highest value observed</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="9">
    <xs:annotation>
      <xs:appinfo>minimum</xs:appinfo>
      <xs:documentation>The smallest value observed</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="11">
    <xs:annotation>
      <xs:appinfo>nominal</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="12">
    <xs:annotation>
      <xs:appinfo>normal</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="16">
    <xs:annotation>
      <xs:appinfo>secondMaximum</xs:appinfo>
      <xs:documentation>The second highest value observed</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="17">
    <xs:annotation>
      <xs:appinfo>secondMinimum</xs:appinfo>
      <xs:documentation>The second smallest value observed</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="23">
    <xs:annotation>
      <xs:appinfo>thirdMaximum</xs:appinfo>
      <xs:documentation>The third highest value observed</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="24">
    <xs:annotation>
      <xs:appinfo>fourthMaximum</xs:appinfo>
      <xs:documentation>The fourth highest value observed</xs:documentation>
    </xs:annotation>
  </xs:enumeration>

```

```

</xs:enumeration>
<xs:enumeration value="25">
  <xs:annotation>
    <xs:appinfo>fifthMaximum</xs:appinfo>
    <xs:documentation>The fifth highest value observed</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="26">
  <xs:annotation>
    <xs:appinfo>sum</xs:appinfo>
    <xs:documentation>The accumulated sum</xs:documentation>
  </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="FlowDirectionKind">
  <xs:annotation>
    <xs:documentation>Direction associated with current related Readings. valid values per the standard are
defined in FlowDirectionType.</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="0">
          <xs:annotation>
            <xs:appinfo>none</xs:appinfo>
            <xs:documentation>Not Applicable (N/A)</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="1">
          <xs:annotation>
            <xs:appinfo>forward</xs:appinfo>
            <xs:documentation>"Delivered," or "Imported" as defined 61968-2.Forward Active Energy is a
positive kWh value as one would naturally expect to find as energy is supplied by the utility and consumed at the
service.Forward Reactive Energy is a positive VARh value as one would naturally expect to find in the presence of
inductive loading.In polyphase metering, the forward energy register is incremented when the sum of the phase
energies is greater than zero.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
          <xs:annotation>
            <xs:appinfo>lagging</xs:appinfo>
            <xs:documentation>Typically used to describe that a power factor is lagging the reference value.
Note 1: When used to describe VA, "lagging" describes a form of measurement where reactive power is considered
in all four quadrants, but real power is considered only in quadrants I and IV.Note 2: When used to describe power
factor, the term "Lagging" implies that the PF is negative. The term "lagging" in this case takes the place of the
negative sign. If a signed PF value is to be passed by the data producer, then the direction of flow enumeration
zero (none) should be used in order to avoid the possibility of creating an expression that employs a double
negative. The data consumer should be able to tell from the sign of the PF is leading or lagging. This
principle is analogous to the concept that "Reverse" energy is an implied negative value, and to publish a negative
reverse value would be ambiguous.Note 3: Lagging power factors typically indicate inductive
loading.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="3">
          <xs:annotation>
            <xs:appinfo>leading</xs:appinfo>
            <xs:documentation>Typically used to describe that a power factor is leading the reference
value.Note: Leading power factors typically indicate capacitive loading.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="4">
          <xs:annotation>
            <xs:appinfo>net</xs:appinfo>
            <xs:documentation>|Forward| – |Reverse|, See 61968-2.Note: In some systems, the value
passed as a "net" value could become negative. In other systems the value passed as a "net" value is always a
positive number, and rolls-over and rolls-under as needed.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="5">
          <xs:annotation>
            <xs:appinfo>q1plusQ2</xs:appinfo>
            <xs:documentation>Reactive positive quadrants. (The term "lagging" is
preferred.)</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>

```

```

</xs:enumeration>
<xs:enumeration value="7">
  <xs:annotation>
    <xs:appinfo>q1plusQ3</xs:appinfo>
    <xs:documentation>Quadrants 1 and 3</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="8">
  <xs:annotation>
    <xs:appinfo>q1plusQ4</xs:appinfo>
    <xs:documentation>Quadrants 1 and 4 usually represent forward active
energy</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="9">
  <xs:annotation>
    <xs:appinfo>q1minusQ4</xs:appinfo>
    <xs:documentation>Q1 minus Q4</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="10">
  <xs:annotation>
    <xs:appinfo>q2plusQ3</xs:appinfo>
    <xs:documentation>Quadrants 2 and 3 usually represent reverse active
energy</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="11">
  <xs:annotation>
    <xs:appinfo>q2plusQ4</xs:appinfo>
    <xs:documentation>Quadrants 2 and 4</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="12">
  <xs:annotation>
    <xs:appinfo>q2minusQ3</xs:appinfo>
    <xs:documentation>Q2 minus Q3</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="13">
  <xs:annotation>
    <xs:appinfo>q3plusQ4</xs:appinfo>
    <xs:documentation>Reactive negative quadrants. (The term "leading" is
preferred.)</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="14">
  <xs:annotation>
    <xs:appinfo>q3minusQ2</xs:appinfo>
    <xs:documentation>Q3 minus Q2</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="15">
  <xs:annotation>
    <xs:appinfo>quadrant1</xs:appinfo>
    <xs:documentation>Q1 only</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="16">
  <xs:annotation>
    <xs:appinfo>quadrant2</xs:appinfo>
    <xs:documentation>Q2 only</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="17">
  <xs:annotation>
    <xs:appinfo>quadrant3</xs:appinfo>
    <xs:documentation>Q3 only</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="18">
  <xs:annotation>
    <xs:appinfo>quadrant4</xs:appinfo>
    <xs:documentation>Q4 only</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="19">

```

```

<xs:annotation>
  <xs:appinfo>reverse</xs:appinfo>
  <xs:documentation>Reverse Active Energy is equivalent to "Received," or "Exported" as defined
in 61968-2.Reverse Active Energy is a positive kWh value as one would expect to find when energy is backed by
the service onto the utility network.Reverse Reactive Energy is a positive VARh value as one would expect to find in
the presence of capacitive loading and a leading Power Factor.In polyphase metering, the reverse energy register
is incremented when the sum of the phase energies is less than zero.Note: The value passed as a reverse value is
always a positive value. It is understood by the label "reverse" that it represents negative flow.</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="20">
  <xs:annotation>
    <xs:appinfo>total</xs:appinfo>
    <xs:documentation>|Forward| + |Reverse|, See 61968-2.The sum of the commodity in all
quadrants Q1+Q2+Q3+Q4.In polyphase metering, the total energy register is incremented when the absolute value
of the sum of the phase energies is greater than zero.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="21">
  <xs:annotation>
    <xs:appinfo>totalByPhase</xs:appinfo>
    <xs:documentation>In polyphase metering, the total by phase energy register is incremented
when the sum of the absolute values of the phase energies is greater than zero.In single phase metering, the
formulas for "Total" and "Total by phase" collapse to the same expression. For communication purposes however,
the "Total" enumeration should be used with single phase meter data.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="MeasurementKind">
  <xs:annotation>
    <xs:documentation>Name of physical measurement</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="0">
          <xs:annotation>
            <xs:appinfo>none</xs:appinfo>
            <xs:documentation>Not Applicable</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
          <xs:annotation>
            <xs:appinfo>apparentPowerFactor</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="3">
          <xs:annotation>
            <xs:appinfo>currency</xs:appinfo>
            <xs:documentation>funds</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="4">
          <xs:annotation>
            <xs:appinfo>current</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="5">
          <xs:annotation>
            <xs:appinfo>currentAngle</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="6">
          <xs:annotation>
            <xs:appinfo>currentImbalance</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="7">
          <xs:annotation>
            <xs:appinfo>date</xs:appinfo>

```

```
<xs:documentation/>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="8">
  <xs:annotation>
    <xs:appinfo>demand</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="9">
  <xs:annotation>
    <xs:appinfo>distance</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="10">
  <xs:annotation>
    <xs:appinfo>distortionVoltAmperes</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="11">
  <xs:annotation>
    <xs:appinfo>energization</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="12">
  <xs:annotation>
    <xs:appinfo>energy</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="13">
  <xs:annotation>
    <xs:appinfo>energizationLoadSide</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="14">
  <xs:annotation>
    <xs:appinfo>fan</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="15">
  <xs:annotation>
    <xs:appinfo>frequency</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="16">
  <xs:annotation>
    <xs:appinfo>Funds</xs:appinfo>
    <xs:documentation>Dup with "currency"</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="17">
  <xs:annotation>
    <xs:appinfo>ieee1366ASAI</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="18">
  <xs:annotation>
    <xs:appinfo>ieee1366ASIDI</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="19">
  <xs:annotation>
    <xs:appinfo>ieee1366ASIFI</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="20">
  <xs:annotation>
```

```

        <xs:appinfo>ieee1366CAIDI</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="21">
    <xs:annotation>
        <xs:appinfo>ieee1366CAIFI</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="22">
    <xs:annotation>
        <xs:appinfo>ieee1366CEMIn</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="23">
    <xs:annotation>
        <xs:appinfo>ieee1366CEMSMIn</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="24">
    <xs:annotation>
        <xs:appinfo>ieee1366CTAIDI</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="25">
    <xs:annotation>
        <xs:appinfo>ieee1366MAIFI</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="26">
    <xs:annotation>
        <xs:appinfo>ieee1366MAIFle</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="27">
    <xs:annotation>
        <xs:appinfo>ieee1366SAIDI</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="28">
    <xs:annotation>
        <xs:appinfo>ieee1366SAIFI</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="31">
    <xs:annotation>
        <xs:appinfo>lineLosses</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="32">
    <xs:annotation>
        <xs:appinfo>losses</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="33">
    <xs:annotation>
        <xs:appinfo>negativeSequence</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="34">
    <xs:annotation>
        <xs:appinfo>phasorPowerFactor</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="35">

```

```
<xs:annotation>
  <xs:appinfo>phasorReactivePower</xs:appinfo>
  <xs:documentation/>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="36">
  <xs:annotation>
    <xs:appinfo>positiveSequence</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="37">
  <xs:annotation>
    <xs:appinfo>power</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="38">
  <xs:annotation>
    <xs:appinfo>powerFactor</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="40">
  <xs:annotation>
    <xs:appinfo>quantityPower</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="41">
  <xs:annotation>
    <xs:appinfo>sag</xs:appinfo>
    <xs:documentation>or Voltage Dip</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="42">
  <xs:annotation>
    <xs:appinfo>swell</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="43">
  <xs:annotation>
    <xs:appinfo>switchPosition</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="44">
  <xs:annotation>
    <xs:appinfo>tapPosition</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="45">
  <xs:annotation>
    <xs:appinfo>tariffRate</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="46">
  <xs:annotation>
    <xs:appinfo>temperature</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="47">
  <xs:annotation>
    <xs:appinfo>totalHarmonicDistortion</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="48">
  <xs:annotation>
    <xs:appinfo>transformerLosses</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
```

```

<xs:enumeration value="49">
  <xs:annotation>
    <xs:appinfo>unipedeVoltageDip10to15</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="50">
  <xs:annotation>
    <xs:appinfo>unipedeVoltageDip15to30</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="51">
  <xs:annotation>
    <xs:appinfo>unipedeVoltageDip30to60</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="52">
  <xs:annotation>
    <xs:appinfo>unipedeVoltageDip60to90</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="53">
  <xs:annotation>
    <xs:appinfo>unipedeVoltageDip90to100</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="54">
  <xs:annotation>
    <xs:appinfo>voltage</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="55">
  <xs:annotation>
    <xs:appinfo>voltageAngle</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="56">
  <xs:annotation>
    <xs:appinfo>voltageExcursion</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="57">
  <xs:annotation>
    <xs:appinfo>voltageImbalance</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="58">
  <xs:annotation>
    <xs:appinfo>volume</xs:appinfo>
    <xs:documentation>Clarified from Ed. 1. to indicate fluid volume</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="59">
  <xs:annotation>
    <xs:appinfo>zeroFlowDuration</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="60">
  <xs:annotation>
    <xs:appinfo>zeroSequence</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="64">
  <xs:annotation>
    <xs:appinfo>distortionPowerFactor</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>

```

```
</xs:enumeration>
<xs:enumeration value="81">
  <xs:annotation>
    <xs:appinfo>frequencyExcursion</xs:appinfo>
    <xs:documentation>Usually expressed as a "count"</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="90">
  <xs:annotation>
    <xs:appinfo>applicationContext</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="91">
  <xs:annotation>
    <xs:appinfo>apTitle</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="92">
  <xs:annotation>
    <xs:appinfo>assetNumber</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="93">
  <xs:annotation>
    <xs:appinfo>bandwidth</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="94">
  <xs:annotation>
    <xs:appinfo>batteryVoltage</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="95">
  <xs:annotation>
    <xs:appinfo>broadcastAddress</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="96">
  <xs:annotation>
    <xs:appinfo>deviceAddressType1</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="97">
  <xs:annotation>
    <xs:appinfo>deviceAddressType2</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="98">
  <xs:annotation>
    <xs:appinfo>deviceAddressType3</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="99">
  <xs:annotation>
    <xs:appinfo>deviceAddressType4</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="100">
  <xs:annotation>
    <xs:appinfo>deviceClass</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="101">
  <xs:annotation>
    <xs:appinfo>electronicSerialNumber</xs:appinfo>
    <xs:documentation/>
```

```

    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="102">
    <xs:annotation>
      <xs:appinfo>endDeviceID</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="103">
    <xs:annotation>
      <xs:appinfo>groupAddressType1</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="104">
    <xs:annotation>
      <xs:appinfo>groupAddressType2</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="105">
    <xs:annotation>
      <xs:appinfo>groupAddressType3</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="106">
    <xs:annotation>
      <xs:appinfo>groupAddressType4</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="107">
    <xs:annotation>
      <xs:appinfo>ipAddress</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="108">
    <xs:annotation>
      <xs:appinfo>macAddress</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="109">
    <xs:annotation>
      <xs:appinfo>mfgAssignedConfigurationID</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="110">
    <xs:annotation>
      <xs:appinfo>mfgAssignedPhysicalSerialNumber</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="111">
    <xs:annotation>
      <xs:appinfo>mfgAssignedProductNumber</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="112">
    <xs:annotation>
      <xs:appinfo>mfgAssignedUniqueCommunicationAddress</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="113">
    <xs:annotation>
      <xs:appinfo>multiCastAddress</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="114">
    <xs:annotation>
      <xs:appinfo>oneWayAddress</xs:appinfo>

```

```
<xs:documentation/>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="115">
  <xs:annotation>
    <xs:appinfo>signalStrength</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="116">
  <xs:annotation>
    <xs:appinfo>twoWayAddress</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="117">
  <xs:annotation>
    <xs:appinfo>signaltoNoiseRatio</xs:appinfo>
    <xs:documentation>Moved here from Attribute #9 UOM</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="118">
  <xs:annotation>
    <xs:appinfo>alarm</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="119">
  <xs:annotation>
    <xs:appinfo>batteryCarryover</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="120">
  <xs:annotation>
    <xs:appinfo>dataOverflowAlarm</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="121">
  <xs:annotation>
    <xs:appinfo>demandLimit</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="122">
  <xs:annotation>
    <xs:appinfo>demandReset</xs:appinfo>
    <xs:documentation>Usually expressed as a count as part of a billing cycle</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="123">
  <xs:annotation>
    <xs:appinfo>diagnostic</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="124">
  <xs:annotation>
    <xs:appinfo>emergencyLimit</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="125">
  <xs:annotation>
    <xs:appinfo>encoderTamper</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="126">
  <xs:annotation>
    <xs:appinfo>ieee1366MomentaryInterruption</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="127">
  <xs:annotation>
```

```

        <xs:appinfo>ieee1366MomentaryInterruptionEvent</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="128">
    <xs:annotation>
        <xs:appinfo>ieee1366SustainedInterruption</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="129">
    <xs:annotation>
        <xs:appinfo>interruptionBehaviour</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="130">
    <xs:annotation>
        <xs:appinfo>inversionTamper</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="131">
    <xs:annotation>
        <xs:appinfo>loadInterrupt</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="132">
    <xs:annotation>
        <xs:appinfo>loadShed</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="133">
    <xs:annotation>
        <xs:appinfo>maintenance</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="134">
    <xs:annotation>
        <xs:appinfo>physicalTamper</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="135">
    <xs:annotation>
        <xs:appinfo>powerLossTamper</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="136">
    <xs:annotation>
        <xs:appinfo>powerOutage</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="137">
    <xs:annotation>
        <xs:appinfo>powerQuality</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="138">
    <xs:annotation>
        <xs:appinfo>powerRestoration</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="139">
    <xs:annotation>
        <xs:appinfo>programmed</xs:appinfo>
        <xs:documentation/>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="140">

```

```

    <xs:annotation>
      <xs:appinfo>pushbutton</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="141">
    <xs:annotation>
      <xs:appinfo>relayActivation</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="142">
    <xs:annotation>
      <xs:appinfo>relayCycle</xs:appinfo>
      <xs:documentation>Usually expressed as a count</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="143">
    <xs:annotation>
      <xs:appinfo>removalTamper</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="144">
    <xs:annotation>
      <xs:appinfo>reprogrammingTamper</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="145">
    <xs:annotation>
      <xs:appinfo>reverseRotationTamper</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="146">
    <xs:annotation>
      <xs:appinfo>switchArmed</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="147">
    <xs:annotation>
      <xs:appinfo>switchDisabled</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="148">
    <xs:annotation>
      <xs:appinfo>tamper</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="149">
    <xs:annotation>
      <xs:appinfo>watchdogTimeout</xs:appinfo>
      <xs:documentation/>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="150">
    <xs:annotation>
      <xs:appinfo>billLastPeriod</xs:appinfo>
      <xs:documentation>Customer's bill for the previous billing period (Currency)</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="151">
    <xs:annotation>
      <xs:appinfo>billToDate</xs:appinfo>
      <xs:documentation>Customer's bill, as known thus far within the present billing period
(Currency)</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="152">
    <xs:annotation>
      <xs:appinfo>billCarryover</xs:appinfo>
      <xs:documentation>Customer's bill for the (Currency)</xs:documentation>
    </xs:annotation>
  </xs:enumeration>

```

```

</xs:enumeration>
<xs:enumeration value="153">
  <xs:annotation>
    <xs:appinfo>connectionFee</xs:appinfo>
    <xs:documentation>Monthly fee for connection to commodity.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="154">
  <xs:annotation>
    <xs:appinfo>audibleVolume</xs:appinfo>
    <xs:documentation>Sound</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="155">
  <xs:annotation>
    <xs:appinfo>volumetricFlow</xs:appinfo>
    <xs:documentation/>
  </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="PhaseCodeKind">
  <xs:annotation>
    <xs:documentation>Enumeration of phase identifiers. Allows designation of phases for both transmission
and distribution equipment, circuits and loads. Residential and small commercial loads are often served from single-
phase, or split-phase, secondary circuits. Phases 1 and 2 refer to hot wires that are 180 degrees out of phase,
while N refers to the neutral wire. Through single-phase transformer connections, these secondary circuits may be
served from one or two of the primary phases A, B, and C. For three-phase loads, use the A, B, C phase codes
instead of s12N.</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="225">
          <xs:annotation>
            <xs:appinfo>ABCN</xs:appinfo>
            <xs:documentation>ABC to Neutral</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="224">
          <xs:annotation>
            <xs:appinfo>ABC</xs:appinfo>
            <xs:documentation>Involving all phases</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="193">
          <xs:annotation>
            <xs:appinfo>ABN</xs:appinfo>
            <xs:documentation>AB to Neutral</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="97">
          <xs:annotation>
            <xs:appinfo>ACN</xs:appinfo>
            <xs:documentation>Phases A, C and neutral.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="97">
          <xs:annotation>
            <xs:appinfo>BCN</xs:appinfo>
            <xs:documentation>BC to neutral.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="132">
          <xs:annotation>
            <xs:appinfo>AB</xs:appinfo>
            <xs:documentation>Phases A to B</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="96">
          <xs:annotation>
            <xs:appinfo>AC</xs:appinfo>
            <xs:documentation>Phases A and C</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>

```

```
<xs:enumeration value="66">
  <xs:annotation>
    <xs:appinfo>BC</xs:appinfo>
    <xs:documentation>Phases B to C</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="129">
  <xs:annotation>
    <xs:appinfo>AN</xs:appinfo>
    <xs:documentation>Phases A to neutral.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="65">
  <xs:annotation>
    <xs:appinfo>BN</xs:appinfo>
    <xs:documentation>Phases B to neutral.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="33">
  <xs:annotation>
    <xs:appinfo>CN</xs:appinfo>
    <xs:documentation>Phases C to neutral.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="128">
  <xs:annotation>
    <xs:appinfo>A</xs:appinfo>
    <xs:documentation>Phase A.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="64">
  <xs:annotation>
    <xs:appinfo>B</xs:appinfo>
    <xs:documentation>Phase B.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="32">
  <xs:annotation>
    <xs:appinfo>C</xs:appinfo>
    <xs:documentation>Phase C.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="16">
  <xs:annotation>
    <xs:appinfo>N</xs:appinfo>
    <xs:documentation>Neutral</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="272">
  <xs:annotation>
    <xs:appinfo>S2N</xs:appinfo>
    <xs:documentation>Phase S2 to neutral.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="784">
  <xs:annotation>
    <xs:appinfo>S12N</xs:appinfo>
    <xs:documentation>Phase S1, S2 to neutral.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="528">
  <xs:annotation>
    <xs:appinfo>S1N</xs:appinfo>
    <xs:documentation>Phase S1 to Neutral</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="256">
  <xs:annotation>
    <xs:appinfo>S2</xs:appinfo>
    <xs:documentation>Phase S2.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="768">
  <xs:annotation>
    <xs:appinfo>S12</xs:appinfo>
    <xs:documentation>Phase S1 to S2</xs:documentation>
  </xs:annotation>
</xs:enumeration>
```

```

</xs:enumeration>
<xs:enumeration value="0">
  <xs:annotation>
    <xs:appinfo>none</xs:appinfo>
    <xs:documentation>Not applicable to any phase</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="136">
  <xs:annotation>
    <xs:appinfo>AtoAv</xs:appinfo>
    <xs:documentation>Phase A current relative to Phase A voltage</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="72">
  <xs:annotation>
    <xs:appinfo>BAv</xs:appinfo>
    <xs:documentation>Phase B current or voltage relative to Phase A voltage</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="41">
  <xs:annotation>
    <xs:appinfo>CAN</xs:appinfo>
    <xs:documentation>CA to Neutral</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="40">
  <xs:annotation>
    <xs:appinfo>CAv</xs:appinfo>
    <xs:documentation>Phase C current or voltage relative to Phase A voltage</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="17">
  <xs:annotation>
    <xs:appinfo>NG</xs:appinfo>
    <xs:documentation>Neutral to ground</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="512">
  <xs:annotation>
    <xs:appinfo>S1</xs:appinfo>
    <xs:documentation>Phase S1</xs:documentation>
  </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="UnitMultiplierKind">
  <xs:annotation>
    <xs:documentation>The power of ten unit multipliers.</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="Int16">
    <xs:simpleType>
      <xs:restriction base="Int16">
        <xs:enumeration value="-12">
          <xs:annotation>
            <xs:appinfo>p</xs:appinfo>
            <xs:documentation>Pico 10**-12</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="-9">
          <xs:annotation>
            <xs:appinfo>n</xs:appinfo>
            <xs:documentation>Nano 10**-9</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="-6">
          <xs:annotation>
            <xs:appinfo>micro</xs:appinfo>
            <xs:documentation>Micro 10**-6</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="-3">
          <xs:annotation>
            <xs:appinfo>m</xs:appinfo>
            <xs:documentation>Milli 10**-3</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>

```

```

</xs:enumeration>
<xs:enumeration value="-2">
  <xs:annotation>
    <xs:appinfo>c</xs:appinfo>
    <xs:documentation>Centi 10**-2</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="-1">
  <xs:annotation>
    <xs:appinfo>d</xs:appinfo>
    <xs:documentation>Deci 10**-1</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="3">
  <xs:annotation>
    <xs:appinfo>k</xs:appinfo>
    <xs:documentation>Kilo 10**3</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="6">
  <xs:annotation>
    <xs:appinfo>M</xs:appinfo>
    <xs:documentation>Mega 10**6</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="9">
  <xs:annotation>
    <xs:appinfo>G</xs:appinfo>
    <xs:documentation>Giga 10**9</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="12">
  <xs:annotation>
    <xs:appinfo>T</xs:appinfo>
    <xs:documentation>Tera 10**12</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="0">
  <xs:annotation>
    <xs:appinfo>none</xs:appinfo>
    <xs:documentation>Not Applicable or "x1"</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="1">
  <xs:annotation>
    <xs:appinfo>da</xs:appinfo>
    <xs:documentation>deca 10**1</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="2">
  <xs:annotation>
    <xs:appinfo>h</xs:appinfo>
    <xs:documentation>hecto 10**2</xs:documentation>
  </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="QualityOfReading">
  <xs:annotation>
    <xs:documentation>List of codes indicating the quality of the reading</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="0">
          <xs:annotation>
            <xs:appinfo>valid</xs:appinfo>
            <xs:documentation>data that has gone through all required validation checks and either passed
them all or has been verified</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="7">
          <xs:annotation>
            <xs:appinfo>manually edited</xs:appinfo>
            <xs:documentation>Replaced or approved by a human</xs:documentation>

```

</xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="8">
 <xs:annotation>
 <xs:appinfo>estimated using reference day</xs:appinfo>
 <xs:documentation>data value was replaced by a machine computed value based on analysis of
 historical data using the same type of measurement.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="9">
 <xs:annotation>
 <xs:appinfo>estimated using linear interpolation</xs:appinfo>
 <xs:documentation>data value was computed using linear interpolation based on the readings
 before and after it</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="10">
 <xs:annotation>
 <xs:appinfo>questionable</xs:appinfo>
 <xs:documentation>data that has failed one or more checks</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="11">
 <xs:annotation>
 <xs:appinfo>derived</xs:appinfo>
 <xs:documentation>data that has been calculated (using logic or mathematical
 operations)</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="12">
 <xs:annotation>
 <xs:appinfo>projected (forecast)</xs:appinfo>
 <xs:documentation>data that has been calculated as a projection or forecast of future
 readings</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="13">
 <xs:annotation>
 <xs:appinfo>mixed</xs:appinfo>
 <xs:documentation>indicates that the quality of this reading has mixed
 characteristics</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="14">
 <xs:annotation>
 <xs:appinfo>raw</xs:appinfo>
 <xs:documentation>data that has not gone through the validation</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="15">
 <xs:annotation>
 <xs:appinfo>normalized for weather</xs:appinfo>
 <xs:documentation>the values have been adjusted to account for weather</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="16">
 <xs:annotation>
 <xs:appinfo>other</xs:appinfo>
 <xs:documentation>specifies that a characteristic applies other than those
 defined</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="17">
 <xs:annotation>
 <xs:appinfo>validated</xs:appinfo>
 <xs:documentation>data that has been validated and possibly edited and/or estimated in
 accordance with approved procedures</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="18">
 <xs:annotation>
 <xs:appinfo>verified</xs:appinfo>
 <xs:documentation>data that failed at least one of the required validation checks but was
 determined to represent actual usage</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 </xs:restriction>

```

    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:simpleType name="ServiceKind">
  <xs:annotation>
    <xs:documentation>Kind of service represented by the UsagePoint</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="0">
          <xs:annotation>
            <xs:appinfo>electricity</xs:appinfo>
            <xs:documentation>Electricity service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="1">
          <xs:annotation>
            <xs:appinfo>gas</xs:appinfo>
            <xs:documentation>Gas service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
          <xs:annotation>
            <xs:appinfo>water</xs:appinfo>
            <xs:documentation>Water service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="3">
          <xs:annotation>
            <xs:appinfo>time</xs:appinfo>
            <xs:documentation>Time service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="4">
          <xs:annotation>
            <xs:appinfo>heat</xs:appinfo>
            <xs:documentation>Heat service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="5">
          <xs:annotation>
            <xs:appinfo>refuse</xs:appinfo>
            <xs:documentation>Refuse (waster) service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="6">
          <xs:annotation>
            <xs:appinfo>sewerage</xs:appinfo>
            <xs:documentation>Sewerage service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="7">
          <xs:annotation>
            <xs:appinfo>rates</xs:appinfo>
            <xs:documentation>Rates (e.g. tax, charge, toll, duty, tariff, etc.) service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="8">
          <xs:annotation>
            <xs:appinfo>tvLicence</xs:appinfo>
            <xs:documentation>TV license service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="9">
          <xs:annotation>
            <xs:appinfo>internet</xs:appinfo>
            <xs:documentation>Internet service.</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:simpleType name="TimeAttributeKind">
  <xs:annotation>

```

<xs:documentation>Code used to specify a particular type of time interval method for Readings of ReadingType. Valid values per the standard are defined in TimeAttributeType.</xs:documentation>

</xs:annotation>

<xs:union memberTypes="UInt16">

<xs:simpleType>

<xs:restriction base="UInt16">

<xs:enumeration value="0">

<xs:annotation>

<xs:appinfo>none</xs:appinfo>

<xs:documentation>Not Applicable</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="1">

<xs:annotation>

<xs:appinfo>tenMinute</xs:appinfo>

<xs:documentation>10-minute</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="2">

<xs:annotation>

<xs:appinfo>fifteenMinute</xs:appinfo>

<xs:documentation>15-minute</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="3">

<xs:annotation>

<xs:appinfo>oneMinute</xs:appinfo>

<xs:documentation>1-minute</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="4">

<xs:annotation>

<xs:appinfo>twentyfourHour</xs:appinfo>

<xs:documentation>24-hour</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="5">

<xs:annotation>

<xs:appinfo>thirtyMinute</xs:appinfo>

<xs:documentation>30-minute</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="6">

<xs:annotation>

<xs:appinfo>fiveMinute</xs:appinfo>

<xs:documentation>5-minute</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="7">

<xs:annotation>

<xs:appinfo>sixtyMinute</xs:appinfo>

<xs:documentation>60-minute</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="10">

<xs:annotation>

<xs:appinfo>twoMinute</xs:appinfo>

<xs:documentation>2-minute</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="14">

<xs:annotation>

<xs:appinfo>threeMinute</xs:appinfo>

<xs:documentation>3-minute</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="15">

<xs:annotation>

<xs:appinfo>present</xs:appinfo>

<xs:documentation>Within the present period of time</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="16">

<xs:annotation>

<xs:appinfo>previous</xs:appinfo>

<xs:documentation>Shifted within the previous monthly cycle and data set</xs:documentation>

</xs:annotation>

IEC 62746-10-1:2018. Click to view the full PDF of IEC 62746-10-1:2018

```
</xs:enumeration>
<xs:enumeration value="31">
  <xs:annotation>
    <xs:appinfo>twentyMinute</xs:appinfo>
    <xs:documentation>20-minute interval</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="50">
  <xs:annotation>
    <xs:appinfo>fixedBlock60Min</xs:appinfo>
    <xs:documentation>60-minute Fixed Block</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="51">
  <xs:annotation>
    <xs:appinfo>fixedBlock30Min</xs:appinfo>
    <xs:documentation>30-minute Fixed Block</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="52">
  <xs:annotation>
    <xs:appinfo>fixedBlock20Min</xs:appinfo>
    <xs:documentation>20-minute Fixed Block</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="53">
  <xs:annotation>
    <xs:appinfo>fixedBlock15Min</xs:appinfo>
    <xs:documentation>15-minute Fixed Block</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="54">
  <xs:annotation>
    <xs:appinfo>fixedBlock10Min</xs:appinfo>
    <xs:documentation>10-minute Fixed Block</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="55">
  <xs:annotation>
    <xs:appinfo>fixedBlock5Min</xs:appinfo>
    <xs:documentation>5-minute Fixed Block</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="56">
  <xs:annotation>
    <xs:appinfo>fixedBlock1Min</xs:appinfo>
    <xs:documentation>1-minute Fixed Block</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="57">
  <xs:annotation>
    <xs:appinfo>rollingBlock60MinIntvl30MinSubIntvl</xs:appinfo>
    <xs:documentation>60-minute Rolling Block with 30-minute sub-intervals</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="58">
  <xs:annotation>
    <xs:appinfo>rollingBlock60MinIntvl20MinSubIntvl</xs:appinfo>
    <xs:documentation>60-minute Rolling Block with 20-minute sub-intervals</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="59">
  <xs:annotation>
    <xs:appinfo>rollingBlock60MinIntvl15MinSubIntvl</xs:appinfo>
    <xs:documentation>60-minute Rolling Block with 15-minute sub-intervals</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="60">
  <xs:annotation>
    <xs:appinfo>rollingBlock60MinIntvl12MinSubIntvl</xs:appinfo>
    <xs:documentation>60-minute Rolling Block with 12-minute sub-intervals</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="61">
  <xs:annotation>
    <xs:appinfo>rollingBlock60MinIntvl10MinSubIntvl</xs:appinfo>
    <xs:documentation>60-minute Rolling Block with 10-minute sub-intervals</xs:documentation>
  </xs:annotation>
</xs:enumeration>
```

[illegible]

```

        <xs:documentation>10-minute Rolling Block with 5-minute sub-intervals</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="75">
      <xs:annotation>
        <xs:appinfo>rollingBlock10MinIntvl2MinSubIntvl</xs:appinfo>
        <xs:documentation>10-minute Rolling Block with 2-minute sub-intervals</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="76">
      <xs:annotation>
        <xs:appinfo>rollingBlock10MinIntvl1MinSubIntvl</xs:appinfo>
        <xs:documentation>10-minute Rolling Block with 1-minute sub-intervals</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="77">
      <xs:annotation>
        <xs:appinfo>rollingBlock5MinIntvl1MinSubIntvl</xs:appinfo>
        <xs:documentation>5-minute Rolling Block with 1-minute sub-intervals</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="TimePeriodOfInterest">
  <xs:annotation>
    <xs:documentation>[extension]</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="0">
          <xs:annotation>
            <xs:appinfo>none</xs:appinfo>
            <xs:documentation>Not Applicable</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="8">
          <xs:annotation>
            <xs:appinfo>billingPeriod</xs:appinfo>
            <xs:documentation>Captured during the billing period starting at midnight of the first day of the
            billing period (as defined by the billing cycle day). If during the current billing period, it specifies a period from the
            start of the current billing period until "now." </xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="11">
          <xs:annotation>
            <xs:appinfo>daily</xs:appinfo>
            <xs:documentation>Daily Period starting at midnight. If for the current day, this specifies the
            time from midnight to "now." </xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="13">
          <xs:annotation>
            <xs:appinfo>monthly</xs:appinfo>
            <xs:documentation>Monthly period starting at midnight on the first day of the month. If within the
            current month, this specifies the period from the start of the month until "now." </xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="22">
          <xs:annotation>
            <xs:appinfo>seasonal</xs:appinfo>
            <xs:documentation>A season of time spanning multiple months. E.g. "Summer," "Spring," "Fall,"
            and "Winter" based cycle. If within the current season, it specifies the period from the start of the current season
            until "now." </xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="24">
          <xs:annotation>
            <xs:appinfo>weekly</xs:appinfo>
            <xs:documentation>Weekly period starting at midnight on the first day of the week and ending
            the instant before midnight the last day of the week. If within the current week, it specifies the period from the start
            of the week until "now." </xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="32">

```

<xs:annotation>
 <xs:appinfo>specifiedPeriod</xs:appinfo>
 <xs:documentation>For the period defined by the start and end of the TimePeriod element in the message.</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 </xs:restriction>
 </xs:simpleType>
 </xs:union>
 </xs:simpleType>
 <xs:simpleType name="UnitSymbolKind">
 <xs:annotation>
 <xs:documentation>Code for the base unit of measure for Readings of ReadingType. Used in combination with the powerOfTenMultiplier to specify the actual unit of measure. Valid values per the standard are defined in UomType.
 </xs:documentation>
 </xs:annotation>
 <xs:union memberTypes="UInt16">
 <xs:simpleType>
 <xs:restriction base="UInt16">
 <xs:enumeration value="61">
 <xs:annotation>
 <xs:appinfo>VA</xs:appinfo>
 <xs:documentation>Apparent power, Volt Ampere (See also real power and reactive power.), VA</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="38">
 <xs:annotation>
 <xs:appinfo>W</xs:appinfo>
 <xs:documentation>Real power, Watt. By definition, one Watt equals oneJoule per second. Electrical power may have real and reactive components. The real portion of electrical power (I^2R) or $VI\cos\phi$, is expressed in Watts. (See also apparent power and reactive power.), W</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="63">
 <xs:annotation>
 <xs:appinfo>VAR</xs:appinfo>
 <xs:documentation>Reactive power, Volt Ampere reactive. The "reactive" or "imaginary" component of electrical power ($VISin\phi$). (See also real power and apparent power.), VAR</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="71">
 <xs:annotation>
 <xs:appinfo>VAh</xs:appinfo>
 <xs:documentation>Apparent energy, Volt Ampere hours, VAh</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="72">
 <xs:annotation>
 <xs:appinfo>Wh</xs:appinfo>
 <xs:documentation>Real energy, Watt hours, Wh</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="73">
 <xs:annotation>
 <xs:appinfo>VARh</xs:appinfo>
 <xs:documentation>Reactive energy, Volt Ampere reactive hours, VARh</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="29">
 <xs:annotation>
 <xs:appinfo>V</xs:appinfo>
 <xs:documentation>Electric potential, Volt (W/A), V</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="30">
 <xs:annotation>
 <xs:appinfo>ohm</xs:appinfo>
 <xs:documentation>Electric resistance, Ohm (V/A), O</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="5">
 <xs:annotation>
 <xs:appinfo>A</xs:appinfo>
 <xs:documentation>Current, ampere, A</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 </xs:simpleType>
 </xs:union>
 </xs:simpleType>

```

</xs:enumeration>
<xs:enumeration value="25">
  <xs:annotation>
    <xs:appinfo>F</xs:appinfo>
    <xs:documentation>Electric capacitance, Farad (C/V), °C</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="28">
  <xs:annotation>
    <xs:appinfo>H</xs:appinfo>
    <xs:documentation>Electric inductance, Henry (Wb/A), H</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="23">
  <xs:annotation>
    <xs:appinfo>degC</xs:appinfo>
    <xs:documentation>Relative temperature in degrees Celsius. In the SI unit system the symbol is
°C. Electric charge is measured in coulomb that has the unit symbol C. To distinguish degree Celsius from
coulomb the symbol used in the UML is degC. Reason for not using °C is the special character ° is difficult to
manage in software.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="27">
  <xs:annotation>
    <xs:appinfo>sec</xs:appinfo>
    <xs:documentation>Time, seconds, s</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="159">
  <xs:annotation>
    <xs:appinfo>min</xs:appinfo>
    <xs:documentation>Time, minute = s * 60, min</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="160">
  <xs:annotation>
    <xs:appinfo>h</xs:appinfo>
    <xs:documentation>Time, hour = minute * 60, h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="9">
  <xs:annotation>
    <xs:appinfo>deg</xs:appinfo>
    <xs:documentation>Plane angle, degrees, deg</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="10">
  <xs:annotation>
    <xs:appinfo>rad</xs:appinfo>
    <xs:documentation>Plane angle, Radian (m/m), rad</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="31">
  <xs:annotation>
    <xs:appinfo>J</xs:appinfo>
    <xs:documentation>Energy joule, (N·m = C·V = W·s), J</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="32">
  <xs:annotation>
    <xs:appinfo>n</xs:appinfo>
    <xs:documentation>Force newton, (kg m/s²), N</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="53">
  <xs:annotation>
    <xs:appinfo>siemens</xs:appinfo>
    <xs:documentation>Electric conductance, Siemens (A / V = 1 / Ω), S</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="0">
  <xs:annotation>
    <xs:appinfo>none</xs:appinfo>
    <xs:documentation>N/A, None</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="33">

```

```

<xs:annotation>
  <xs:appinfo>Hz</xs:appinfo>
  <xs:documentation>Frequency hertz, (1/s), Hz</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="3">
  <xs:annotation>
    <xs:appinfo>g</xs:appinfo>
    <xs:documentation>Mass in gram, g</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="39">
  <xs:annotation>
    <xs:appinfo>pa</xs:appinfo>
    <xs:documentation>Pressure, Pascal (N/m2)(Note: the absolute or relative measurement of
pressure is implied with this entry. See below for more explicit forms.), Pa</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="2">
  <xs:annotation>
    <xs:appinfo>m</xs:appinfo>
    <xs:documentation>Length, meter, m</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="41">
  <xs:annotation>
    <xs:appinfo>m2</xs:appinfo>
    <xs:documentation>Area, square meter, m2</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="42">
  <xs:annotation>
    <xs:appinfo>m3</xs:appinfo>
    <xs:documentation>Volume, cubic meter, m3</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="69">
  <xs:annotation>
    <xs:appinfo>A2</xs:appinfo>
    <xs:documentation>Amps squared, amp squared, A2</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="105">
  <xs:annotation>
    <xs:appinfo>A2h</xs:appinfo>
    <xs:documentation>ampere-squared, Ampere-squared hour, A2h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="70">
  <xs:annotation>
    <xs:appinfo>A2s</xs:appinfo>
    <xs:documentation>Amps squared time, square amp second, A2s</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="106">
  <xs:annotation>
    <xs:appinfo>Ah</xs:appinfo>
    <xs:documentation>Ampere-hours, Ampere-hours, Ah</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="152">
  <xs:annotation>
    <xs:appinfo>APerA</xs:appinfo>
    <xs:documentation>Current, Ratio of Amperages, A/A</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="103">
  <xs:annotation>
    <xs:appinfo>aPerM</xs:appinfo>
    <xs:documentation>A/m, magnetic field strength, Ampere per metre, A/m</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="68">
  <xs:annotation>
    <xs:appinfo>As</xs:appinfo>
    <xs:documentation>Amp seconds, amp seconds, As</xs:documentation>
  </xs:annotation>
</xs:enumeration>

```

```

</xs:enumeration>
<xs:enumeration value="79">
  <xs:annotation>
    <xs:appinfo>b</xs:appinfo>
    <xs:documentation>Sound pressure level, Bel, acoustic, Combine with multiplier prefix "d" to
form decibels of Sound Pressure Level"dB (SPL).", B (SPL)</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="113">
  <xs:annotation>
    <xs:appinfo>bm</xs:appinfo>
    <xs:documentation>Signal Strength, Bel-mW, normalized to 1mW. Note: to form "dBm" combine
"Bm" with multiplier "d". Bm</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="22">
  <xs:annotation>
    <xs:appinfo>bq</xs:appinfo>
    <xs:documentation>Radioactivity, Becquerel (1/s), Bq</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="132">
  <xs:annotation>
    <xs:appinfo>btu</xs:appinfo>
    <xs:documentation>Energy, British Thermal Units, BTU</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="133">
  <xs:annotation>
    <xs:appinfo>btuPerH</xs:appinfo>
    <xs:documentation>Power, BTU per hour, BTU/h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="8">
  <xs:annotation>
    <xs:appinfo>cd</xs:appinfo>
    <xs:documentation>Luminous intensity, candela, cd</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="76">
  <xs:annotation>
    <xs:appinfo>char</xs:appinfo>
    <xs:documentation>Number of characters, characters, char</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="75">
  <xs:annotation>
    <xs:appinfo>HzPerSec</xs:appinfo>
    <xs:documentation>Rate of change of frequency, hertz per second, Hz/s</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="114">
  <xs:annotation>
    <xs:appinfo>code</xs:appinfo>
    <xs:documentation>Application Value, encoded value, code</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="65">
  <xs:annotation>
    <xs:appinfo>cosTheta</xs:appinfo>
    <xs:documentation>Power factor, Dimensionless , cos?</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="111">
  <xs:annotation>
    <xs:appinfo>count</xs:appinfo>
    <xs:documentation>Amount of substance, counter value, count</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="119">
  <xs:annotation>
    <xs:appinfo>ft3</xs:appinfo>
    <xs:documentation>Volume, cubic feet, ft³</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="120">

```

```

<xs:annotation>
  <xs:appinfo>ft3compensated</xs:appinfo>
  <xs:documentation>Volume, cubic feet, ft3(compensated)</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="123">
  <xs:annotation>
    <xs:appinfo>ft3compensatedPerH</xs:appinfo>
    <xs:documentation>Volumetric flow rate, compensated cubic feet per hour,
ft3(compensated)/h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="78">
  <xs:annotation>
    <xs:appinfo>gM2</xs:appinfo>
    <xs:documentation>Turbine inertia, gram·meter2 (Combine with multiplier prefix "k" to form
kg·m2.), gm2</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="144">
  <xs:annotation>
    <xs:appinfo>gPerG</xs:appinfo>
    <xs:documentation>Concentration, The ratio of the mass of a solute divided by the mass of the
solution., g/g</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="21">
  <xs:annotation>
    <xs:appinfo>gy</xs:appinfo>
    <xs:documentation>Absorbed dose, Gray (J/kg), GY</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="150">
  <xs:annotation>
    <xs:appinfo>HzPerHz</xs:appinfo>
    <xs:documentation>Frequency, Rate of frequency change, Hz/Hz</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="77">
  <xs:annotation>
    <xs:appinfo>charPerSec</xs:appinfo>
    <xs:documentation>Data rate, characters per second, char/s</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="130">
  <xs:annotation>
    <xs:appinfo>imperialGal</xs:appinfo>
    <xs:documentation>Volume, imperial gallons, ImperialGal</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="131">
  <xs:annotation>
    <xs:appinfo>imperialGalPerH</xs:appinfo>
    <xs:documentation>Volumetric flow rate, Imperial gallons per hour,
ImperialGal/h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="51">
  <xs:annotation>
    <xs:appinfo>jPerK</xs:appinfo>
    <xs:documentation>Heat capacity, Joule/Kelvin, J/K</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="165">
  <xs:annotation>
    <xs:appinfo>jPerKg</xs:appinfo>
    <xs:documentation>Specific energy, Joules / kg, J/kg</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="6">
  <xs:annotation>
    <xs:appinfo>K</xs:appinfo>
    <xs:documentation>Temperature, Kelvin, K</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="158">
  <xs:annotation>

```

<xs:appinfo>kat</xs:appinfo>
 <xs:documentation>Catalytic activity, katal = mol / s, kat</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="47">
 <xs:annotation>
 <xs:appinfo>kgM</xs:appinfo>
 <xs:documentation>Moment of mass ,kilogram meter (kg·m), M</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="48">
 <xs:annotation>
 <xs:appinfo>kgPerM3</xs:appinfo>
 <xs:documentation>Density, gram/cubic meter (combine with prefix multiplier "k" to form kg/ m³),
 g/m³</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="134">
 <xs:annotation>
 <xs:appinfo>litre</xs:appinfo>
 <xs:documentation>Volume, litre = dm³ = m³/1000., L</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="157">
 <xs:annotation>
 <xs:appinfo>litreCompensated</xs:appinfo>
 <xs:documentation>Volume, litre, with the value compensated for weather effects,
 L(compensated)</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="138">
 <xs:annotation>
 <xs:appinfo>litreCompensatedPerH</xs:appinfo>
 <xs:documentation>Volumetric flow rate, litres (compensated) per hour,
 L(compensated)/h</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="137">
 <xs:annotation>
 <xs:appinfo>litrePerH</xs:appinfo>
 <xs:documentation>Volumetric flow rate, litres per hour, L/h</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="143">
 <xs:annotation>
 <xs:appinfo>litrePerLitre</xs:appinfo>
 <xs:documentation>Concentration, The ratio of the volume of a solute divided by the volume of
 the solution., L/L</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="82">
 <xs:annotation>
 <xs:appinfo>litrePerSec</xs:appinfo>
 <xs:documentation>Volumetric flow rate, Volumetric flow rate, L/s</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="156">
 <xs:annotation>
 <xs:appinfo>litreUncompensated</xs:appinfo>
 <xs:documentation>Volume, litre, with the value uncompensated for weather effects.,
 L(uncompensated)</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="139">
 <xs:annotation>
 <xs:appinfo>litreUncompensatedPerH</xs:appinfo>
 <xs:documentation>Volumetric flow rate, litres (uncompensated) per hour,
 L(uncompensated)/h</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="35">
 <xs:annotation>
 <xs:appinfo>lm</xs:appinfo>
 <xs:documentation>Luminous flux, lumen (cd sr), Lm</xs:documentation>
 </xs:annotation>
 </xs:enumeration>
 <xs:enumeration value="34">

```

<xs:annotation>
  <xs:appinfo>lx</xs:appinfo>
  <xs:documentation>Illuminance lux, (lm/m2), L(uncompensated)/h</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="49">
  <xs:annotation>
    <xs:appinfo>m2PerSec</xs:appinfo>
    <xs:documentation>Viscosity, meter squared / second, m2/s</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="167">
  <xs:annotation>
    <xs:appinfo>m3compensated</xs:appinfo>
    <xs:documentation>Volume, cubic meter, with the value compensated for weather effects.,
m3(compensated)</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="126">
  <xs:annotation>
    <xs:appinfo>m3compensatedPerH</xs:appinfo>
    <xs:documentation>Volumetric flow rate, compensated cubic meters per hour,
3(compensated)/h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="125">
  <xs:annotation>
    <xs:appinfo>m3PerH</xs:appinfo>
    <xs:documentation>Volumetric flow rate, cubic meters per hour, m3/h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="45">
  <xs:annotation>
    <xs:appinfo>m3PerSec</xs:appinfo>
    <xs:documentation>m3PerSec, cubic meters per second, m3/s</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="166">
  <xs:annotation>
    <xs:appinfo>m3uncompensated</xs:appinfo>
    <xs:documentation>m3uncompensated, cubic meter, with the value uncompensated for weather
effects., m3(uncompensated)</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="127">
  <xs:annotation>
    <xs:appinfo>m3uncompensatedPerH</xs:appinfo>
    <xs:documentation>Volumetric flow rate, uncompensated cubic meters per hour,
m3(uncompensated)/h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="118">
  <xs:annotation>
    <xs:appinfo>meCode</xs:appinfo>
    <xs:documentation>EndDeviceEvent, value to be interpreted as a EndDeviceEventCode,
meCode</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="7">
  <xs:annotation>
    <xs:appinfo>mol</xs:appinfo>
    <xs:documentation>Amount of substance, mole, mol</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="147">
  <xs:annotation>
    <xs:appinfo>molPerKg</xs:appinfo>
    <xs:documentation>Concentration, Molality, the amount of solute in moles and the amount of
solvent in kilograms., mol/kg</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="145">
  <xs:annotation>
    <xs:appinfo>molPerM3</xs:appinfo>
    <xs:documentation>Concentration, The amount of substance concentration, (c), the amount of
solvent in moles divided by the volume of solution in m3., mol/ m3</xs:documentation>
  </xs:annotation>

```

```

</xs:enumeration>
<xs:enumeration value="146">
  <xs:annotation>
    <xs:appinfo>molPerMol</xs:appinfo>
    <xs:documentation>Concentration, Molar fraction (?), the ratio of the molar amount of a solute
divided by the molar amount of the solution., mol/mol</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="80">
  <xs:annotation>
    <xs:appinfo>money</xs:appinfo>
    <xs:documentation>Monetary unit, Generic money (Note: Specific monetary units are identified
the currency class)., ¤</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="148">
  <xs:annotation>
    <xs:appinfo>mPerM</xs:appinfo>
    <xs:documentation>Length, Ratio of length, m/m</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="46">
  <xs:annotation>
    <xs:appinfo>mPerM3</xs:appinfo>
    <xs:documentation>Fuel efficiency, meters / cubic meter, m/m³</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="43">
  <xs:annotation>
    <xs:appinfo>mPerSec</xs:appinfo>
    <xs:documentation>Velocity, meters per second (m/s), m/s</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="44">
  <xs:annotation>
    <xs:appinfo>mPerSec2</xs:appinfo>
    <xs:documentation>Acceleration, meters per second squared, m/s²</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="102">
  <xs:annotation>
    <xs:appinfo>ohmM</xs:appinfo>
    <xs:documentation>resistivity, ? (rho), ?m</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="155">
  <xs:annotation>
    <xs:appinfo>paA</xs:appinfo>
    <xs:documentation>Pressure, Pascal, absolute pressure, PaA</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="140">
  <xs:annotation>
    <xs:appinfo>paG</xs:appinfo>
    <xs:documentation>Pressure, Pascal, gauge pressure, PaG</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="141">
  <xs:annotation>
    <xs:appinfo>psiA</xs:appinfo>
    <xs:documentation>Pressure, Pounds per square inch, absolute, psiA</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="142">
  <xs:annotation>
    <xs:appinfo>psiG</xs:appinfo>
    <xs:documentation>Pressure, Pounds per square inch, gauge, psiG</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="100">
  <xs:annotation>
    <xs:appinfo>q</xs:appinfo>
    <xs:documentation>Quantity power, Q, Q</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="161">
  <xs:annotation>

```

```

        <xs:appinfo>q45</xs:appinfo>
        <xs:documentation>Quantity power, Q measured at 45°, Q45</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="163">
    <xs:annotation>
        <xs:appinfo>q45h</xs:appinfo>
        <xs:documentation>Quantity energy, Q measured at 45°, Q45h</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="162">
    <xs:annotation>
        <xs:appinfo>q60</xs:appinfo>
        <xs:documentation>Quantity power, Q measured at 60°, Q60</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="164">
    <xs:annotation>
        <xs:appinfo>q60h</xs:appinfo>
        <xs:documentation>Quantity energy, Qh measured at 60°, Q60h</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="101">
    <xs:annotation>
        <xs:appinfo>qh</xs:appinfo>
        <xs:documentation>Quantity energy, Qh, Qh</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="54">
    <xs:annotation>
        <xs:appinfo>radPerSec</xs:appinfo>
        <xs:documentation>Angular velocity, radians per second, rad/s</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="154">
    <xs:annotation>
        <xs:appinfo>rev</xs:appinfo>
        <xs:documentation>Amount of rotation, Revolutions, rev</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="4">
    <xs:annotation>
        <xs:appinfo>revPerSec</xs:appinfo>
        <xs:documentation>Rotational speed, Rotations per second, rev/s</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="149">
    <xs:annotation>
        <xs:appinfo>secPerSec</xs:appinfo>
        <xs:documentation>Time, Ratio of time (can be combined with an multiplier prefix to show rates
such as a clock drift rate, e.g. "µs/s"), s/s</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="11">
    <xs:annotation>
        <xs:appinfo>sr</xs:appinfo>
        <xs:documentation>Solid angle, Steradian (m2/m2), sr</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="109">
    <xs:annotation>
        <xs:appinfo>status</xs:appinfo>
        <xs:documentation>State, "1" = "true", "live", "on", "high", "set"; "0" = "false", "dead", "off", "low",
"cleared"Note: A Boolean value is preferred but other values may be supported, status</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="24">
    <xs:annotation>
        <xs:appinfo>sv</xs:appinfo>
        <xs:documentation>Doe equivalent, Sievert (J/kg), Sv</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="37">
    <xs:annotation>
        <xs:appinfo>t</xs:appinfo>
        <xs:documentation>Magnetic flux density, Tesla (Wb/m2), T</xs:documentation>
    </xs:annotation>

```

```

</xs:enumeration>
<xs:enumeration value="169">
  <xs:annotation>
    <xs:appinfo>therm</xs:appinfo>
    <xs:documentation>Energy, Therm, therm</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="108">
  <xs:annotation>
    <xs:appinfo>timeStamp</xs:appinfo>
    <xs:documentation>Timestamp, time and date per ISO 8601 format,
timeStamp</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="128">
  <xs:annotation>
    <xs:appinfo>usGal</xs:appinfo>
    <xs:documentation>Volume, US gallons, Gal</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="129">
  <xs:annotation>
    <xs:appinfo>usGalPerH</xs:appinfo>
    <xs:documentation>Volumetric flow rate, US gallons per hour, USGal/h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="67">
  <xs:annotation>
    <xs:appinfo>V2</xs:appinfo>
    <xs:documentation>Volts squared, Volt squared (W2/A2), V2</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="104">
  <xs:annotation>
    <xs:appinfo>V2h</xs:appinfo>
    <xs:documentation>volt-squared hour, Volt-squared-hours, V2h</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="117">
  <xs:annotation>
    <xs:appinfo>VAhPerRev</xs:appinfo>
    <xs:documentation>Kh-Vah, apparent energy metering constant, VAh/rev</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="116">
  <xs:annotation>
    <xs:appinfo>VARhPerRev</xs:appinfo>
    <xs:documentation>Kh-VArh, reactive energy metering constant, VARh/rev</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="74">
  <xs:annotation>
    <xs:appinfo>VPerHz</xs:appinfo>
    <xs:documentation>Magnetic flux, Volts per Hertz, V/Hz</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="151">
  <xs:annotation>
    <xs:appinfo>VPerV</xs:appinfo>
    <xs:documentation>Voltage, Ratio of voltages (e.g. mV/V), V/V</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="66">
  <xs:annotation>
    <xs:appinfo>Vs</xs:appinfo>
    <xs:documentation>Volt seconds, Volt seconds (Ws/A), Vs</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="36">
  <xs:annotation>
    <xs:appinfo>wb</xs:appinfo>
    <xs:documentation>Magnetic flux, Weber (V s), Wb</xs:documentation>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="107">
  <xs:annotation>
    <xs:appinfo>WhPerM3</xs:appinfo>

```

```

        <xs:documentation>Wh/m3, energy per volume, Wh/m³</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="115">
      <xs:annotation>
        <xs:appinfo>WhPerRev</xs:appinfo>
        <xs:documentation>Kh-Wh, active energy metering constant, Wh/rev</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="50">
      <xs:annotation>
        <xs:appinfo>wPerMK</xs:appinfo>
        <xs:documentation>Thermal conductivity, Watt/meter Kelvin, W/m K</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="81">
      <xs:annotation>
        <xs:appinfo>WPerSec</xs:appinfo>
        <xs:documentation>Ramp rate, Watts per second, W/s</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="153">
      <xs:annotation>
        <xs:appinfo>WPerVA</xs:appinfo>
        <xs:documentation>Power Factor, PF, W/VA</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="168">
      <xs:annotation>
        <xs:appinfo>WPerW</xs:appinfo>
        <xs:documentation>Signal Strength, Ratio of power, W/W</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="StatusCode">
  <xs:annotation>
    <xs:documentation>indicates the status code of the associated transaction</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="200">
          <xs:annotation>
            <xs:appinfo>Ok</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="201">
          <xs:annotation>
            <xs:appinfo>Created</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="204">
          <xs:annotation>
            <xs:appinfo>No Content</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="301">
          <xs:annotation>
            <xs:appinfo>Moved Permanently</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="302">
          <xs:annotation>
            <xs:appinfo>Redirect</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="304">
          <xs:annotation>
            <xs:appinfo>Not Modified</xs:appinfo>

```

```

        <xs:documentation/>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="400">
      <xs:annotation>
        <xs:appinfo>Bad Request</xs:appinfo>
        <xs:documentation/>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="401">
      <xs:annotation>
        <xs:appinfo>Unauthorized</xs:appinfo>
        <xs:documentation/>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="403">
      <xs:annotation>
        <xs:appinfo>Forbidden</xs:appinfo>
        <xs:documentation/>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="404">
      <xs:annotation>
        <xs:appinfo>Not Found</xs:appinfo>
        <xs:documentation/>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="405">
      <xs:annotation>
        <xs:appinfo>Method Not Allowed</xs:appinfo>
        <xs:documentation/>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="410">
      <xs:annotation>
        <xs:appinfo>Gone</xs:appinfo>
        <xs:documentation/>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="500">
      <xs:annotation>
        <xs:appinfo>Internal Server Error</xs:appinfo>
        <xs:documentation/>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="CRUDOperation">
  <xs:annotation>
    <xs:documentation>Specifies the operation requested of this item</xs:documentation>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="0">
          <xs:annotation>
            <xs:appinfo>Create</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="1">
          <xs:annotation>
            <xs:appinfo>Read</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
          <xs:annotation>
            <xs:appinfo>Update</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="3">
          <xs:annotation>
            <xs:appinfo>Delete</xs:appinfo>

```

```

        <xs:documentation/>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType name="DataCustodianApplicationStatus">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="1">
          <xs:annotation>
            <xs:appinfo>Review</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
          <xs:annotation>
            <xs:appinfo>Production (Live)</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="3">
          <xs:annotation>
            <xs:appinfo>On Hold</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="4">
          <xs:annotation>
            <xs:appinfo>Revoked</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:simpleType name="ThirdPartyApplicationStatus">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="1">
          <xs:annotation>
            <xs:appinfo>Development</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
          <xs:annotation>
            <xs:appinfo>ReviewTest</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="3">
          <xs:annotation>
            <xs:appinfo>Production</xs:appinfo>
            <xs:documentation>Live</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="4">
          <xs:annotation>
            <xs:appinfo>Retired</xs:appinfo>
            <xs:documentation>Remove</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>

```

```

<xs:simpleType name="ThirdPartyApplicationType">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="1">
          <xs:annotation>
            <xs:appinfo>Web</xs:appinfo>
            <xs:documentation>The application is on the web</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
          <xs:annotation>
            <xs:appinfo>Desktop</xs:appinfo>
            <xs:documentation>The application is on a desktop</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="3">
          <xs:annotation>
            <xs:appinfo>Mobile</xs:appinfo>
            <xs:documentation>The application is on a mobil device</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="4">
          <xs:annotation>
            <xs:appinfo>Device</xs:appinfo>
            <xs:documentation>The application is on another device</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:simpleType name="ThirdPartyApplicationUse">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:union memberTypes="UInt16">
    <xs:simpleType>
      <xs:restriction base="UInt16">
        <xs:enumeration value="1">
          <xs:annotation>
            <xs:appinfo>EnergyManagement</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
          <xs:annotation>
            <xs:appinfo>Comparisons</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="3">
          <xs:annotation>
            <xs:appinfo>Government</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="4">
          <xs:annotation>
            <xs:appinfo>Academic</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="5">
          <xs:annotation>
            <xs:appinfo>LawEnforcement</xs:appinfo>
            <xs:documentation/>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:simpleType name="AuthorizationStatus">

```