Energy Interoperation Version 1.0

Committee Specification 01

16 December 2011

Specification URIs

This version:

http://docs.oasis-open.org/energyinterop/ei/v1.0/cs01/energyinterop-v1.0-cs01.pdf (Authoritative)
http://docs.oasis-open.org/energyinterop/ei/v1.0/cs01/energyinterop-v1.0-cs01.html
http://docs.oasis-open.org/energyinterop/ei/v1.0/cs01/energyinterop-v1.0-cs01.doc

Previous version:

http://docs.oasis-open.org/energyinterop/ei/v1.0/csprd02/energyinterop-v1.0-csprd02.doc (Authoritative)
http://docs.oasis-open.org/energyinterop/ei/v1.0/csprd02/energyinterop-v1.0-csprd02.html
http://docs.oasis-open.org/energyinterop/ei/v1.0/csprd02/energyinterop-v1.0-csprd02.pdf

Latest version:

http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.pdf (Authoritative)
http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html
http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.doc

Technical Committee:

OASIS Energy Interoperation TC

Chairs:

David Holmberg (david.holmberg@nist.gov), NIST
William T. Cox (wtcox@coxsoftwarearchitects.com), Individual

Editors:

Toby Considine (toby.considine@unc.edu), University of North Carolina at Chapel Hill

Additional artifacts:

This prose specification is one component of a Work Product which also includes:

- XML schemas: http://docs.oasis-open.org/energyinterop/ei/v1.0/cs01/xsd/
- WSDL files: http://docs.oasis-open.org/energyinterop/ei/v1.0/cs01/wsdl/

Related work:

This specification is related to:

- *Energy Market Information Exchange (EMIX) Version 1.0*. Latest version. http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html
- *WS-Calendar Version 1.0*. Latest version. http://docs.oasis-open.org/ws-calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.html
- NAESB Actors for DR

Declared XML namespaces:

- http://docs.oasis-open.org/ns/energyinterop/201110
- http://docs.oasis-open.org/ns/energyinterop/201110/enroll
- http://docs.oasis-open.org/ns/energyinterop/201110/payloads
- http://docs.oasis-open.org/ns/energyinterop/201110/wsdl

Abstract:

OASIS Energy Interoperation describes an information model and a communication model to enable collaborative and transactive use of energy, service definitions consistent with the OASIS SOA Reference Model [SOA-RM], and XML vocabularies for the interoperable and standard exchange of:

- Dynamic price signals
- Reliability signals
- Emergency signals
- Communication of market participation information such as bids
- Load predictability and generation information

This work facilitates enterprise interaction with energy markets, which:

- Allows effective response to emergency and reliability events
- Allows taking advantage of lower energy costs by deferring or accelerating usage
- Enables trading of curtailment and generation
- Supports symmetry of interaction between providers and consumers of energy
- Provides for aggregation of provision, curtailment, and use

The definition of a price and of reliability information depends on the market context in which it exists. It is not in scope for this TC to define specifications for markets or for pricing models, but the TC has coordinated with others to ensure that commonly used market and pricing models are supported.

While this specification uses Web Services to describe the services, no requirement or expectation of specific messaging implementation is assumed.

Status:

This document was last revised or approved by the OASIS Energy Interoperation TC on the above date and includes an inline edit to correct a normative reference made on 14 March 2012. The original publication is available on request from OASIS TC Administration. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at http://www.oasis-open.org/committees/energyinterop/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/energyinterop/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[ENERGYINTEROP-v1.0]

Energy Interoperation Version 1.0. 16 December 2011. OASIS Committee Specification 01. http://docs.oasis-open.org/energyinterop/ei/v1.0/cs01/energyinterop-v1.0-cs01.html.

Notices

Table of Contents

Tables, Figures & Examples

Index to Figures

Index to Tables

1 Introduction

Energy Interoperation describes an information and communication model to coordinate energy supply, transmission, distribution, and use, including power and ancillary services, between any two parties, such as energy suppliers and customers, markets and service providers, in any of the domains indicated in Figure 2.1 below. Energy Interoperation makes no assumptions about which entities will enter those markets, or as to what those market roles will be called in the future. Energy Interoperation supports each of the secure communications interfaces in Figure 1-1, but is not limited to those interfaces.

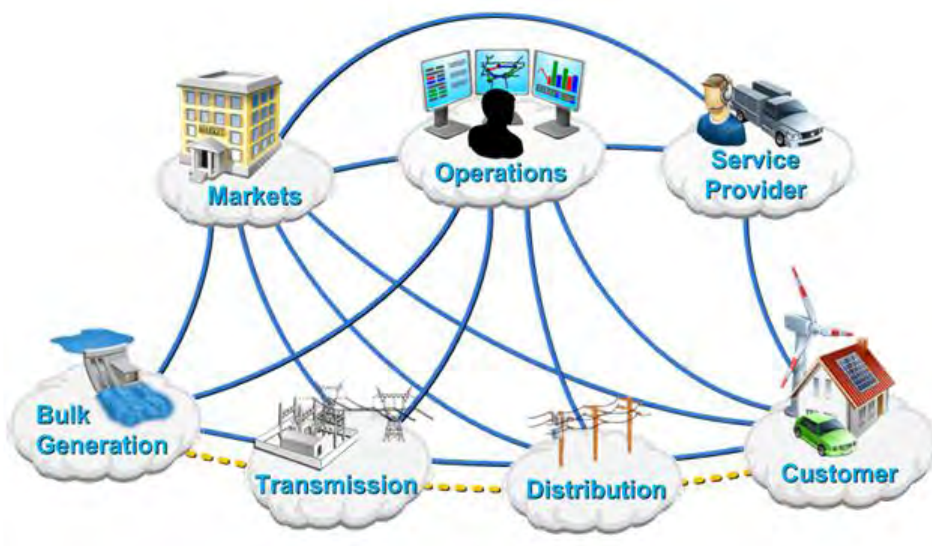


*Figure 1-1: Conceptual model for smart Grid from **[NIST]** showing communications requirements*

Energy Interoperation defines messages to communicate price, reliability, and emergency conditions over communications interfaces. Energy Interoperation is agnostic as to the technology that a communications interface may use to carry these messages.

Energy Interoperation messages can concern real time interactions, forward projections, or historical reporting. Energy Interoperation is intended to support market-based balancing of energy supply and demand while increasing fluidity of transactions. Increased deployment of distributed and intermittent energy sources will require greater fluidity in both wholesale and retail markets. In retail markets, Energy Interoperation is meant to support greater consumer choice as to energy source.

Energy supplies are becoming more volatile due to the introduction of renewable energy sources. The introduction of distributed energy resources may create localized, volatile surpluses and shortages. These changes will create more granular energy transactions, require more granularity in temporal price changes , and more granularity in service territory.

Balancing local energy resources brings more kinds of resources into the mix. Natural gas markets share many characteristics with electricity markets. Local thermal energy distribution systems can balance electricity markets while having their own surpluses and shortages. Nothing in Energy Interoperation restricts its use to electricity-based markets.

Energy consumers will need technologies to manage their local energy supply, including curtailment, storage, generation, and time-of-use load shaping and shifting. In particular, consumers will respond to Energy Interoperation messages for emergency and reliability events, or price messages to take advantage of lower energy costs by deferring or accelerating usage, and to trade curtailment, local generation and energy supply rights. Energy Interoperation does not specify which technologies consumers will use; rather it defines a technology agnostic interface to enable accelerated market development of such technologies.

To balance supply and demand, energy suppliers must be able to schedule resources, manage aggregation, and communicate both the scarcity and surplus of energy supply over time. Suppliers will use Energy Interoperation to inform customers of emergency and reliability events, to trade curtailment and supply of energy, and to provide intermediation services including aggregation of provision, curtailment, and use.

Energy Interoperation relies on standard format for communication of time and interval [WS-Calendar] and for Energy Price and Product Definition [EMIX]. This document assumes that there is a high degree of symmetry of interaction at any Energy Interoperation interface, i.e., that providers and customers may reverse roles during any period.

The OASIS Energy Interoperation Technical Committee is developing this specification in support of the National Institute of Standards and Technology (NIST) Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0 [Framework] in support of the US Department of Energy (DOE) as described in the Energy Independence and Security Act of 2007 [EISA2007].

Under the Framework and Roadmap, the North American Energy Standards Board (NAESB) surveyed the electricity industry and prepared a consensus statement of requirements and vocabulary. This work was submitted to the Energy Interoperation Committee in April 2010 and subsequently updated and delivered in January 2011.

All examples and all Appendices are non-normative.

1.1 Terminology

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

1.2 Normative References

[EMIX] *Energy Market Information Exchange (EMIX) Version 1.0.* 11 January 2012. OASIS Committee Specification. http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[RFC2246] T. Dierks, C. Allen *Transport Layer Security (TLS) Protocol Version 1.0*, http://www.ietf.org/rfc/rfc2246.txt, IETF RFC 2246, January 1999.

[SOA-RM] SOA-RM OASIS Standard, *OASIS Reference Model for Service Oriented Architecture 1.0*, October 2006 http://docs.oasis-open.org/soa-rm/v1.0/

[Vavailability] C. Daboo, B. Desruisseaux, Calendar Availability, http://tools.ietf.org/html/draft-daboo-calendar-availability-02, IETF Internet Draft, April 2011

[WS-Calendar] WS-Calendar OASIS Committee Specification 1.0, WS-Calendar, July 2011, http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.pdf

1.3 Non-Normative References

[BACnet/WS] Addendum C to ANSI/ASHRAE Standard 135-2004, *BACnet Web Services Interface.*

[ebXML-MS] OASIS Standard, *Electronic Business XML (ebXML) Message Service Specification v3.0: Part 1, Core Features*, October 2007. http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/ebms_core-3.0-spec-os.pdf

[EISA2007] Energy Independence and Security Act of 2007, http://nist.gov/smartgrid/upload/EISA-Energy-bill-110-140-TITLE-XIII.pdf

[EPRI] Concepts to Enable Advancement of Distributed Energy Resources, February 2010, http://my.epri.com/portal/server.pt?Abstract_id=000000000001020432

[Framework] National Institute of Standards and Technology, *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0*, January 2010, http://nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf

[Galvin] Galvin Electricity Initiative, *Perfect Power*, http://www.galvinpower.org/perfect-power/what-is-perfect-power

[ID-CLOUD] OASIS Identity in the Cloud Technical Committee http://www.oasis-open.org/committees/id-cloud

[IEC 61968] Application integration at electric utilities - System interfaces for distribution management - Part 9: Interfaces for meter reading and control

[IEC 61970-301] Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base

[KMIP] OASIS Standard, *Key Management Interoperability Protocol Specification Version 1.0*, October 2010 http://docs.oasis-open.org/kmip/spec/v1.0/kmip-spec-1.0.pdf

[OpenADR] Mary Ann Piette, Girish Ghatikar, Sila Kiliccote, Ed Koch, Dan Hennage, Peter Palensky, and Charles McParland. 2009. Open Automated Demand Response Communications Specification (Version 1.0). California Energy Commission, PIER Program. CEC-500-2009-063. [NAESB-SG] NAESB Smart Grid Subcommittee, http://www.naesb.org/smart_grid_standards_strategies_development.asp

[OASIS SCA] OASIS Service Component Architecture Member Section http://www.oasis-opencsa.org/sca

[OASIS PMRM] OASIS Privacy Management Reference Model (PMRM) Technical Committee, http://www.oasis-open.org/committees/pmrm

[SAML] OASIS Standard, *Security Assertion Markup Language 2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SOA-RA] OASIS Public Review Draft 01, *Reference Architecture for Service Oriented Architecture Version 1.0,* April 2008 http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf

[SPML] OASIS Standard, *Service Provisioning Markup Language (SPML) v2 - DSML v2 Profile*, April 2006. http://www.oasis-open.org/committees/download.php/17708/pstc-spml-2.0-os.zip [TC57CIM] IEC Technical Committee 57 Common Information Model (IEC 61968 and IEC 61970, various dates)

[TEMIX] TeMIX *Transactive Energy Market Information Exchange [TeMIX] an approved Note of the EMIX TC*. Ed Cazalet et al. http://www.oasis-open.org/committees/download.php/37954/TeMIX-20100523.pdf

[Vavailability] C. Daboo, B. Desruisseaux, Calendar Availability, http://tools.ietf.org/html/draft-daboo-calendar-availability-02, IETF Internet Draft, April 2011

[WS-Addr] Web Services Addressing (WS-Addressing) 1.0, W3C Recommendation, http://www.w3.org/2005/08/addressing.

[WSFED] OASIS Standard, *Web Services Federation Language (WS-Federation) Version 1.2*, 01 May 2009 http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.doc

[WSI-Basic] R Chumbley, J Durand, G Pilz, T Rutt , *Basic Profile Version 2.0,* http://ws-i.org/profiles/BasicProfile-2.0-2010-11-09.html, The Web Services-Interoperability Organization, November 2010

[WSRM] OASIS Standard, *WS-Reliable Messaging 1.1*, November 2004. http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf

[WS-SecureConversation] OASIS Standard, *WS-SecureConversation 1.3,* March 2007. http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.pdf

[WS-Security] OASIS Standard, *WS-Security 2004 1.1*, February 2006. http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

[WS-SX] OASIS Web Services Secure Exchange (WS-SX) Technical Committee http://www.oasis-open.org/committees/ws-sx

[XACML] OASIS Standard, *eXtensible Access Control Markup Language 2.0*, February 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

1.4 Contributions

The NIST Roadmap for Smart Grid Interoperability Standards described in the **[Framework]** requested that many standards development organizations (SDOs) and trade associations work together closely in unprecedented ways. An extraordinary number of groups came together and contributed effort, time, requirements, and documents. Each of these groups further gathered together, repeatedly, to review the work products of this committee and submit detailed comments. These groups contributed large numbers of documents to the Technical Committee. These efforts intersected with this specification in ways almost impossible to unravel, and the committee acknowledges the invaluable works below which are essential to understanding the North American Grid and its operation today, as well as its potential futures.

NAESB Smart Grid Standards Development Subcommittee [NAESB-SG]:

The following documents are password protected. For information about obtaining access to these documents, please visit www.naesb.org or contact the NAESB office at (713) 356 0060.

[NAESB EUI] NAESB REQ Energy Usage Information Model: http://www.naesb.org/member_login_check.asp?doc=req_rat102910_req_2010_ap_9d_rec.doc

[NAESB EUI] NAESB WEQ Energy Usage Information Model: http://www.naesb.org/member_login_check.asp?doc=weq_rat102910_weq_2010_ap_6d_rec.doc

The following documents are under development and subject to change.

[NAESB PAP 09] Phase Two Requirements Specification for Wholesale Standard DR Signals – for NIST PAP09: http://www.naesb.org/member_login_check.asp?doc=fa_2010_weq_api_6_c_ii.doc

[NAESB PAP 09] Phase Two Requirements Specification for Retail Standard DR Signals – for NIST PAP09: http://www.naesb.org/member_login_check.asp?doc=fa_2010_retail_api_9_c.doc

The ISO / RTO Council Smart Grid Standards Project:

Information Model – HTML: http://www.isorto.org/atf/cf/%7B5B4E85C6-7EAC-40A0-8DC3-003829518EBD%7D/IRC-DR-InformationModel-HTML-Condensed_Rev1_20101014.zip

Information Model – EAP: http://www.isorto.org/atf/cf/%7B5B4E85C6-7EAC-40A0-8DC3-003829518EBD%7D/IRC-DR-InformationModel-EAP-Condensed_Rev1_20101014.zip

XML Schemas: http://www.isorto.org/atf/cf/%7B5B4E85C6-7EAC-40A0-8DC3-003829518EBD%7D/IRC-DR-XML_Schemas_Rev1_20101014.zip

Eclipse CIMTool Project: http://www.isorto.org/atf/cf/%7B5B4E85C6-7EAC-40A0-8DC3-003829518EBD%7D/IRC-DR-CIMTool-Project-Workspace_Rev1_20101014.zip

Interactions - Enrollment and Qualification: http://www.isorto.org/atf/cf/%7B5B4E85C6-7EAC-40A0-8DC3-003829518EBD%7D/IRC-DR-Interactions-HTML_Enrollment_And_Qualification_Rev1_20101014.zip

Interactions - Scheduling and Award Notification: http://www.isorto.org/atf/cf/%7B5B4E85C6-7EAC-40A0-8DC3-003829518EBD%7D/IRC-DR-Interactions-HTML_Scheduling_And_Award_Notification_Rev1_20101014.zip

Interactions - Deployment and Real Time Notifications: http://www.isorto.org/atf/cf/%7B5B4E85C6-7EAC-40A0-8DC3-003829518EBD%7D/IRC-DR-Interactions-HTML_Deployment_And_RealTime_Communications_Rev1_20101014.zip

Interactions - Measurement and Performance: http://www.isorto.org/atf/cf/%7B5B4E85C6-7EAC-40A0-8DC3-003829518EBD%7D/IRC-DR-Interactions-HTML_Measurement_And_Performance_Rev1_20101014.zip

Interactions Non-Functional Requirements: http://www.isorto.org/atf/cf/%7B5B4E85C6-7EAC-40A0-8DC3-003829518EBD%7D/IRC-DR-Non-Functional_Requirements_Rev1_20100930.pdf

UCAIug OpenSG OpenADR Task Force:

OpenADR 1.0 System Requirements Specification v1.0 http://osgug.ucaiug.org/sgsystems/OpenADR/Shared%20Documents/SRS/OpenSG%20OpenADR%201.0%20SRS%20v1.0.pdf

OpenADR 1.0 Service Definition - Common Version :R0.91 http://osgug.ucaiug.org/sgsystems/OpenADR/Shared%20Documents/Services/OpenSG%20OpenADR%20SD%20-%20Common%20r0.91.doc

OpenADR 1.0 Service Definition – Web Services Implementation Profile Version: v0.91 http://osgug.ucaiug.org/sgsystems/OpenADR/Shared%20Documents/Services/OpenSG%20OpenADR%20SD%20-%20WS%20r0.91.doc

1.5 Namespace

The XML namespace [XML-ns] URI that MUST be used by implementations of this specification is:

```
http://docs.oasis-open.org/ns/energyinterop
```

Dereferencing the above URI will produce the Resource Directory Description Language [RDDL 2.0] document that describes this namespace.

Table 1 lists the XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Table 1-1: Namespaces Used in this Specification

Prefix	Namespace
xs	http://www.w3.org/2001/XMLSchema
gml	http://www.opengis.net/gml/3.2
xcal	urn:ietf:params:xml:ns:icalendar-2.0
strm	urn:ietf:params:xml:ns:icalendar-2.0:stream
emix	http://docs.oasis-open.org/ns/emix/2011/06
power	http://docs.oasis-open.org/ns/emix/2011/06/power
resource	http://docs.oasis-open.org/ns/emix/2011/06/power/resource
ei	http://docs.oasis-open.org/ns/energyinterop/201110
enrl	http://docs.oasis-open.org/ns/energyinterop/201110/enroll
pyld	http://docs.oasis-open.org/ns/energyinterop/201110/payloads
wsdl	http://docs.oasis-open.org/ns/energyinterop/201110/wsdl

The normative schemas for EMIX can be found linked from the namespace document that is located at the namespace URI specified above.

1.6 Naming Conventions

This specification follows some naming conventions for artifacts defined by the specification, as follows:

For the names of elements and the names of attributes within XSD files, the names follow the lowerCamelCase convention, with all names starting with a lower case letter. For example,

```
<element name="componentType" type="ei:ComponentType"/>
```

For the names of types within XSD files, the names follow the UpperCamelCase convention with all names starting with a lower case letter prefixed by “type-“. For example,

```
<complexType name="ComponentServiceType">
```

For the names of intents, the names follow the lowerCamelCase convention, with all names starting with a lower case letter, EXCEPT for cases where the intent represents an established acronym, in which case the entire name is in upper case.

An example of an intent that is an acronym is the "SOAP" intent.

1.7 Editing Conventions

For readability, element names in tables appear as separate words. The actual names are lowerCamelCase, as specified above, and as they appear in the XML schemas.

All elements in the tables not marked as “optional” are mandatory.

Information in the “Specification” column of the tables is normative. Information appearing in the note column is explanatory and non-normative.

All sections explicitly noted as examples are informational and are not to be considered normative.

1.8 Architectural Background

Energy Interoperability defines a service-oriented approach to energy interactions. Accordingly, it assumes a certain amount of definitions of roles, names, and interaction patterns. This document relies heavily on roles and interactions as defined in the OASIS Standard *Reference Model for Service Oriented Architecture* **[SOA-RA]**.

Service orientation focuses on the desired results rather than the requested processes. Service orientation complements loose integration. Service orientation organizes distributed capabilities that may be in different ownership domains.

The SOA paradigm concerns itself with visibility, interaction, and effect. Visibility refers to the capacity for those with needs and those with capabilities to be able to see each other. Interaction is the activity of using a capability. A service provides a decision point for any policies and transactions without delving into the process on either side of the interface

Services are concerned with the public actions of each interoperating system. Service interactions consider private actions, e.g., those on either side of the interface, to be inherently unknowable by other parties. A service is used without needing to know all the details of its implementation. Services are generally paid for results, not effort.

While loosely coupled it is important to understand some typical message exchange patterns to understand how business processes are tied together through an SOA. [SOA-RA] section 4.3.2.1 describes how message exchange patterns (MEP) are leveraged for this purpose. While [SOA-RA] describes two types of MEPs, event notification and request response it also notes that, "This is by no means a complete list of all possible MEPs used for inter- or intra-enterprise messaging".

Three types of MEPs can inform the discussion on energy-interop integration; a one way MEP, which differs somewhat from an event notification MEP in that no response is required or expected from the service provider, although the service consumer may receive appropriate http messages, e.g. 404 error.

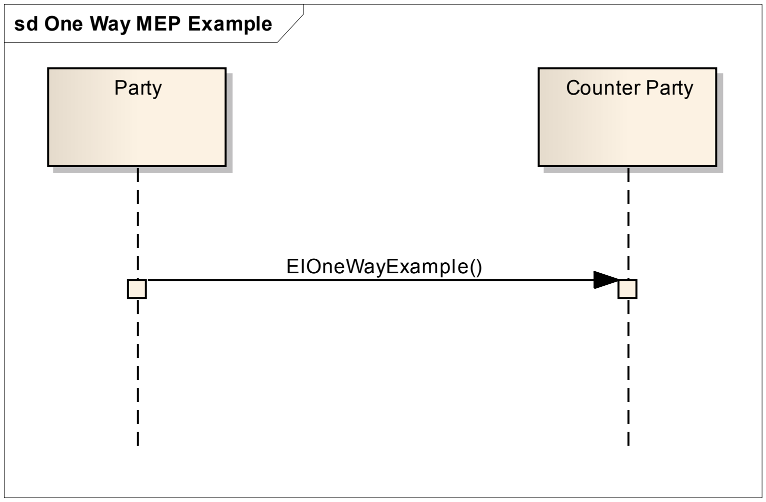


Figure 1-2: One-way MEP where no return is expected

Additionally a two-way MEP and a callback MEP are specific types of request/response MEPs described in [SOA-RA] that are used in Energy Interoperation. A two way MEP exchange pattern assumes that after a service is consumed an acknowledgement is sent. This acknowledgement is made up of the message header of the returning service, and may include a standardized acknowledgement payload, i.e., for capturing errors, (or no errors is the service was called successfully).

The callback MEP is similar to the request/response pattern described in [SOA-RA] except that it is more specific. In a callback MEP the service provider will send an acknowledgement upon receiving a request, however, once the service provider completes the corresponding business process, it will become a service consumer, by calling a service of the previous consumer, where it turn it will receive its own acknowledgement.

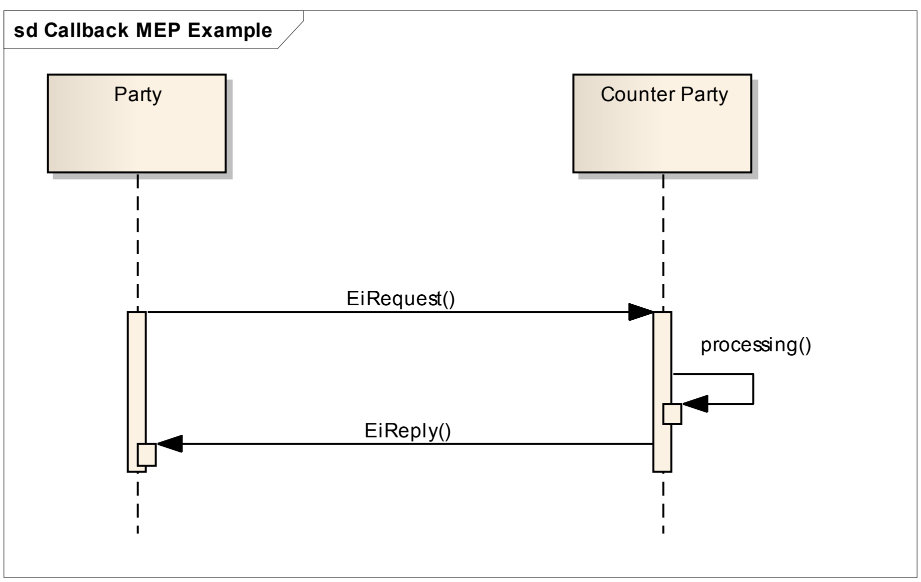


Figure 1-3: Callback MEP where a service provider sends an acknowledgement to the service consumer, performs a corresponding activity to act on the service request, then in turn makes a service request to the original initiating service consumer and receiving an acknowledgement in return.

Note: Acknowledgements are normally shown as a dashed arrow return but have been omitted from the figures of this specification for brevity. Appropriate returns should be assumed.

Most figures that illustrate a service interaction assume a PUSH paradigm, however that is not a requirement. A PULL paradigm may also be employed using energy-interop services. However, the PULL pattern differs slightly. A request is made, responded to, and then once the requestor has the information required, then it acts using a final operation as shown in the following figure.

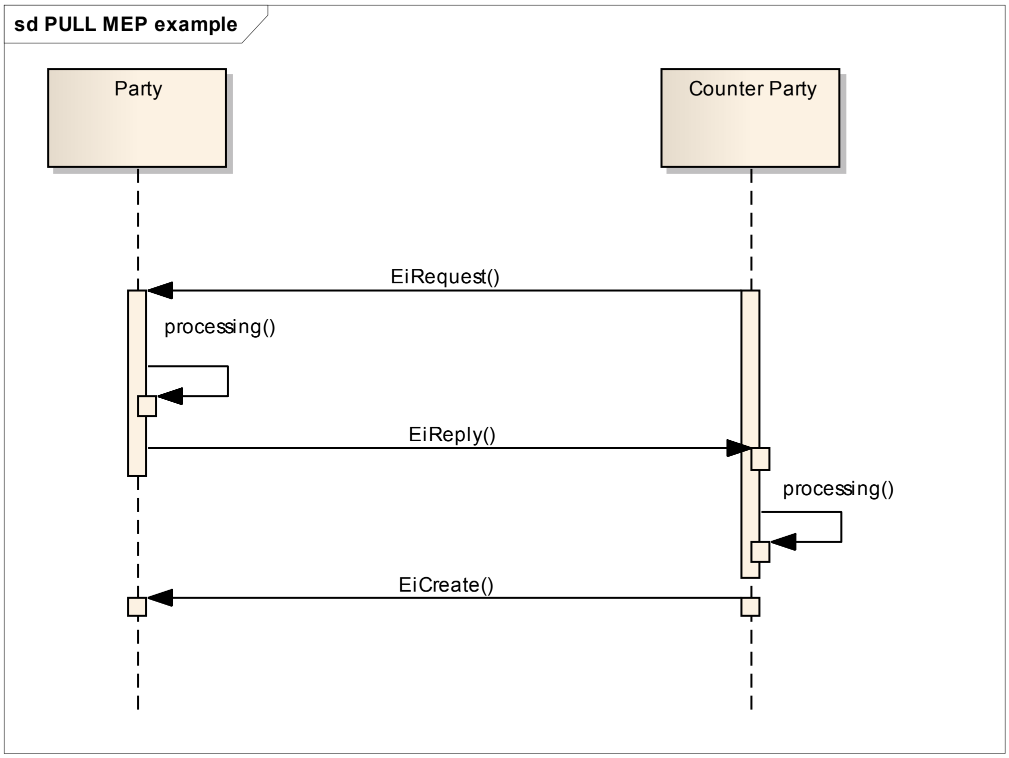


Figure 1-4: PULL MEP where a request is made, responded to, processed and then acted upon. Nominally this could be considered a combination of a callback MEP, followed by a two-way MEP

Loose integration using the SOA style assumes careful definition of security requirements between partners. Size of transactions, costs of failure to perform, confidentiality agreements, information stewardship, and even changing regulatory requirements can require similar transactions be expressed within quite different security contexts. It is a feature of the SOA approach that security is composed in to meet the specific and evolving needs of different markets and transactions. Security implementation must be free to evolve over time and to support different needs. Energy Interoperation allows for this composition, without prescribing any particular security implementation.

2 Overview of Energy Interoperation

2.1 Scope of Energy Interoperation

Energy Interoperation (EI) supports the following:

- Transactive Energy
- Distribution of dynamic and contract prices
- Demand response approaches ranging from dispatch of load resources to price levels embedded in an event.
- Measurement and confirmation of response.
- Projected price, demand, and energy

EI engages Distributed Energy Resources (DER) while making no assumptions as to their processes or technology.

While this specification supports agreements and transactional obligations, this specification offers flexibility of implementation to support specific programs, regional requirements, and goals of the various participants including the utility industry, aggregators, suppliers, and device manufacturers.

It is not the intent of the Energy Interoperation Technical Committee to imply that any particular agreements are endorsed, proposed, or required in order to implement this specification. Energy market operations are beyond the scope of this specification although the interactions that enable management of the actual delivery and acceptance are within scope. Energy Interoperation defines interfaces for use throughout the transport chain of electricity as well as supporting today's intermediation services and those that may arise tomorrow.

2.2 Specific scope statements

Interaction patterns and service definitions to support the following are in scope for Energy Interoperation:

- Market communications to support transactive energy. (see **[TEMIX]**)
- Specific offerings by end nodes to alter energy use.
- Measurement and confirmation of actions taken, including but not limited to curtailment, generation, and storage, including load and usage information, historical, present, and projected.
- Notifications requesting performance on transactions offered or executed.
- Information models for price and product communication.
- Service definitions for Energy Interoperation

The following are out of scope for Energy Interoperation:

- Requirements specifying the type of agreement, or tariff used by a particular market.
- Validation and verification of performance, except for feedback on curtailment and generation.
- Communication (e.g. transport method) other than Web services to carry the messages from one point to another. The messages specified in Energy Interoperation can be transmitted via a variety of transports.

2.3 Goals & Guidelines for Signals and Price and Product Communication

1. There are at least four market types, and signals and price and product standardization must support all four, while allowing for the key differences that exist and will continue to exist in them. The four market types are:

- no open wholesale and no retail competition
- open wholesale market only
- open retail competition only
- open wholesale and open retail competition.

2. Wholesale market DR signals and price and product communication have different characteristics than retail market DR signals and price and product communication, although Energy Interoperation defines a commonality in format.
3. It is likely that most end users, with some exceptions among Commercial and Industrial (C&I) customers, will not interact directly with wholesale markets.
4. Retail pricing models are complex, due to the numerous tariff rate structures that exist in both regulated and un-regulated markets. Attempts to standardize DR control and pricing signals must not hinder regulatory changes or market innovations when it comes to future tariff or pricing models.
5. New business entities such as Energy Service Providers (ESP), Demand Response Providers (DRP), DR Aggregators, and Energy Information Service Providers (EISP), will play an increasing role in DR implementation. Energy Interoperation supports these and new as yet unnamed intermediation services.
6. DER may play an increasingly important role in DR, yet the development of tariff and/or pricing models that support DER's role in DR are still in early stages of development.
7. The Customer's perspective and ability to react to DR control and price signals must be a key driver during the development of standards to support DR programs.

In addition, it is the policy of the Energy Interoperation Technical Committee that

8. Where feasible, customer interfaces and the presentation of energy information to the customer should be left in the hands of the market, systems, and product developers enabled by these specifications.

The NAESB Smart Grid Committee **[NAESB-SG]** provided guidance on the Demand Response and the electricity market customer interactions, as a required input under NIST Smart Grid Priority Action Plan 9 (PAP09). Energy Interoperation relied on this guidance. The service and class definitions relied on the information developed to support the NAESB effort in the wholesale **[IRC]** and retail **[OpenSG]** markets.

2.4 Scope of Energy Interoperation Communications

While the bulk of examples describe the purchase of real power, emerging energy markets must exchange economic information about other time-sensitive services.

For example, delivery of power is often constrained by delivery bottlenecks. The emergence of distributed generation and Plugin Electric Vehicles (PEV) will exacerbate this problem. EMIX includes product definitions for tradable congestion charges and transmission rights. Locational market prices in distribution may come to mirror those already seen in transmission markets.

Other services address the direct effects of distribution congestion, including phase imbalances, voltage violations, overloads, etc.

These markets introduce different market products, yet the roles and interactions remain the same. Intelligent distribution elements, up to an intelligent transformer take roles in these interactions.

A description of the tariffs or market rules to support these interactions is outside the scope of this specification. However, interaction patterns in this specification are defined to provide additional information for markets in which tariffs or market rules are required.

2.5 Collaborative Energy [Not Normative]

Collaborative Energy, in this specification, refers to the transactions and management of energy using collaborative approaches, including but not limited to markets, requests for decrease of net demand, while addressing the business goals of the respective parties in arms-length interactions.

Transactive energy describes the established process of parties buying and selling energy based on tenders (buy or sell offers) that may lead to transactions among parties. In open wholesale forward energy markets, a generator may tender a quantity of energy at a price over a future delivery interval of time to a customer. Acceptance of a tender results in a binding transaction. In some cases, the transaction requires physical delivery of energy. In other cases, the transaction is settled for cash at a price determined by a prescribed price index. The use of Energy Interoperation to enable present and future wholesale and retail energy markets and retail tariffs, including dynamic and multi-part tariffs is described in [EMIX]. This section reviews the generic roles and interactions of parties involved in energy transactions.

In this specification, the information exchanged and the services needed to implement smart energy are defined.

Today's markets are not necessarily tomorrow's. Today's retail markets have grown up around conflicting market restrictions, tariffs that are contrary to the goals of Collaborative Energy, and historical practices that pre-date automated metering and e-commerce. Today's wholesale market applications, designed, built and deployed in the absence of standards, has resulted in little or no interchangeability among vendor products, complex integration techniques, and duplicated product development. The Technical Committee opted to avoid direct engagement with these problems. Energy Interoperation aims for future flexibility while it addresses the problems of today.

While the focus today is on on-demand load reduction, on-demand load increase is just as critical for Collaborative Energy interactions. Any large component of intermittent energy sources will create temporary surpluses as well as surfeits. Interactions between different smart grids and between smart grids and end nodes must maximize load shifting to reflect changing surpluses or shortages of electricity. Responsibilities and benefits must accrue together to the participants most willing and able to adapt.

The Committee, working with the **[EMIX]** Technical Committee developed a component model of an idealized market for electricity transactions. This model assumes timely automated interval metering and an e-commerce infrastructure. TEMIX describes electricity in this normal market context. This model was explained in the **[TEMIX]** paper, an approved work product of the EMIX committee. Using the components in this model, the authors were then able to go back and simulate the market operations of today.

Energy Interoperation supports four essential market activities:

1. There is an **indication of interest** (trying to find tenders to buy or sell) when a Party is seeking partner Parties for a demand response transaction or for an energy source or sale.
2. There is a **tender** (offer or bid) to buy or sell a service, e.g. production of energy or curtailment of use.
3. There is a **transaction** to purchase or supply, generally from the acceptance of a tender.
4. For some transactions, such as Demand Response, there is an **execution** for delivery of the subject of a transaction at the agreed-upon price, time, and place.

Version 1.0 of Energy Interoperation does not define the critical fifth market activity, **measurement and verification** (M&V). A NAESB task force (Demand Side Management and Energy Efficiency Working Groups) is continuing work to define the business requirements for M&V.

Other business models may combine services in novel ways. An aggregator can publish an indication of interest to buy curtailment at a given price. A business willing to respond would offer an agreement to shed load for a specific price. The aggregator may accept some or all of these offers. The performance in this case could be called at the same time as the tender acceptance or later.

Communication of price in transactions is at the core of the Energy Interoperation services. Five types of prices are identified in this specification:

1. Priced Offer: a forward offer to buy or sell a quantity of an energy product for a specified future interval of time, the acceptance of which by a counterparty results in a binding agreement. This includes tariff priced offers where the quantity may be limited only by the service connection and DR prices.
2. Ex-Post Price: A price assigned to energy purchased or sold that is calculated or assigned after delivery. Price may be set based on market indices, centralized market clearing, tariff calculation or any other process.
3. Priced Indication of Interest: the same as a Priced Offer except that no binding agreement is immediately intended.
4. Historical Price: A current price, past transaction price, past offered price, and statistics about historical price such as high and low prices, averages and volatility.
5. Price Forecast: A forecast by a party of future prices that are not a Priced Indication of Interest or Priced Offer. The quality of a price forecast will depend on the source and future market conditions

A grid price service is able to answer the following sorts of questions:

1. What is the price of Electricity now?
2. What will it be in 5 minutes?
3. What price will electricity have for each hour of the day tomorrow? What is the confidence level about these predictions?
4. What will it be at other times in the future?
5. What was the highest or lowest price for electricity in the last day? Month? Year?
6. What was the high price for the day the last time it was this hot?

Each answer carries with it varying degrees of certainty. The prices may be fixed by contracts or tariffs that change infrequently if at all. The prices may be fixed tariffs, “unless a DR event is called.” The prices may even represent wild guesses about open markets. With a standardized price service, technology providers can develop solutions to help grid operators and grid customers manage their energy use portfolios.

This specification also encompasses Emergency or "Grid Reliability" events. Grid Reliability events require mandatory participation in today’s markets. These events are described as standing pre-executed option agreements. A grid operator need merely call for performance as in any other event.

2.6 Assumptions

2.6.1 Availability of Interval Metering

Energy Interoperation for many actions presumes a capability of interval metering where the interval might be smaller than the billing cycle. Intervals are typically one hour or less. Interval metering may be required for settlement or operations for measurement and verification of curtailment, distributed energy resources, and for other Energy Interoperation interactions.

2.6.2 Use of EMIX

This specification uses the OASIS Energy Market Information Exchange [EMIX] to communicate product definitions, quantities, and prices. EMIX provides a succinct way to indicate how prices, quantities, or both vary over time.

2.6.3 Use of WS-Calendar

This specification uses the OASIS [WS-Calendar] specification to communicate schedules and intervals. WS-Calendar is the standard under the NIST Smart Grid Roadmap for all such communication.

WS-Calendar expresses a general approach to communications of sequences and schedules, and their gradual complete instantiation during the transactive process. Despite its name, WS-Calendar does not require that communications use web services.

2.6.4 Energy Services Interface

The Energy Services Interface (ESI) is the external face of the energy-consuming node. The ESI may be directly on an energy management system in the end node, or it may be mediated by other business systems. The ESI is the point of communication whereby the entities (e.g. utilities, ISOs) that produce and distribute electricity interact with the entities (e.g. facilities and aggregators) that manage the consumption of electricity. An ESI may be in front of one system or several, one building or several, or even in front of a microgrid.

This work assumes that there is no direct interaction across the ESI.

3 Energy Interoperation Architecture

The following sections provide an overview of the interaction structure, and define the roles and actors in electricity markets. Later sections will define the interactions more carefully as services. The section first addresses Transactive Energy Interactions and then addresses Event Interactions for Demand and Generation Resources.

The Energy Interoperation (EI) architecture describes interactions between pairs of actors, and in a deployment, relationships are established among actors. Actors may perform in a chains of pairs of actors.

3.1 Transactive Energy Interactions

Transactive Energy refers to the communication of prospective and completed transactions of energy whether market-based, bilateral or, contract-, agreement-, or tariff-based, and whether of energy or options on energy. The terminology used by Transactive Energy is most evident today in the buying and selling of wholesale energy in bilateral and exchange transactions. This section reviews and interactions of Parties involved in energy transactions.

The actor for all Transactive EI interactions is a Party. A Party can be an end-use customer, a generator, a retail service provider, a demand response provider, a marketer, a distribution system operator, a transmission system operator, a system operator such as an ISO or RTO, a microgrid operator, or any party engaging in transactions for energy or the transport of energy.

Parties may participate in interactions concurrently as well as over time. In theory, any Party can transact with any other Party subject to applicable regulatory restrictions. In practice, markets will establish interactions between Parties based on regulation, convenience, economics, credit, network structure, locations, and other factors.

3.1.1 Transaction Side

A Party can take one of two Sides in a given Transaction:

- Buy, or
- Sell

At any moment, a Party has a position resulting from any previous Transactions. A Party selling power relative to its current position takes the Buy Side of the Transaction. A Party buying power relative to its current position takes the Sell Side of the Transaction.

A generator typically takes the Sell Side of a Transaction, but can also take the Buy Side of a Transaction. A generator may take the Buy Side of a Buyer in order to reduce generation because of a change in generator or market conditions.

An end-use customer typically takes the Sell Side of a Transaction, but if tendered an attractive price may curtail usage and thereby take the Sell Side of a Transaction.

A distributed generator also can take the Side of Buyer or Seller in a Transaction. For example, if a distributed generator sells 2 MW for an hour forward of a given interval, it may decide to buy back all or a portion of the 2 MW for that hour if the price is low enough. A distributed storage device may take the Buy side of a Transaction to store energy and the Sell Side of Buyer in a Transaction at a different time to release energy from storage.

3.1.2 Transactive Interactions among Parties

Parties may interact using Tenders for Transactions as illustrated in Figure 3-1.

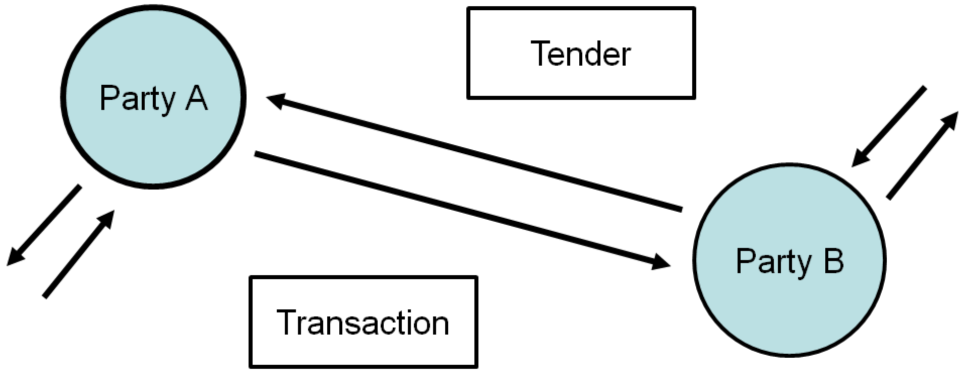


Figure 3-1: Parties Interacting Using Tenders for Transactions

Suppose Party B takes the Buy Side in initiating a Tender to a CounterParty, Party A. Party A has the Sell Side of that Tender. If the Tender is accepted by Party A, Party A takes the Sell Side and Party B takes the Buy Side of the Transaction.

Any Party can initiate a Tender to any CounterParty and take on either the Buy or Sell Side. The CounterParty can accept or reject Tenders from Parties and itself initiate Tenders as Party to any CounterParty to the extent allowed by market rules and regulations.

Two parties can also engage in an option Transaction. An option is a promise granted by a Party (Option Writer) to a CounterParty (Option Holder) usually for a premium payment. The Option Holder is granted a right to invoke specific Transactions for energy that the Option Writer promises to deliver. Demand response, ancillary services, and price cap Transactions are forms of options. Any Party may take the Buy Side or Sell of Sell Side a Tender for an option Transaction either as the Holder or Writer of the option.

3.1.3 Retail Service Interactions

Retail Customers interact with either tariffed cost-of-service retail providers or competitive retail providers with various service plans. Either way the price of the service must be clearly communicated to the customer. With the introduction of interval metering and dynamic pricing, clear communication of price and the purchasing decisions by customers is essential.

EI provides services to communicate both the tendered prices by retailers to customers and the purchase transactions by customers. Customers with distributed energy resources (DER) or storage may often be a seller to retailer or other parties. Transactions may also include call options on customers by a retailer to reduce deliveries and call options by customers on a retailer to provide price insurance.

3.1.4 Wholesale Power Interactions

Retail Energy Providers, Aggregators, Power Marketers, Brokers, Exchanges, System Operators and Generators all interact in the wholesale market for deliveries on the high voltage transmission grid. Transactions include forward transactions for delivery, near-real time transaction and cash settled futures transactions for hedging risks.

EI mirrors the tender and transaction interaction patterns of open forward wholesale power markets. Near real-time wholesale markets for resources provided by independent system operators are also provided for in EI design with work ongoing.

3.1.5 Transport Interactions

Transmission and Distribution services transport energy from one location to another. Transport is the common term used by EI and EMIX to refer to both Transmission and Distribution. Prices for Transport are dynamic and need careful communication. EI models tenders and transactions for Transport products using the same interactions as for Energy products.

EI makes no assumptions about how prices for Transport are determined.

3.2 Event Interactions for Demand and Generation Resources

In partial contrast to the transactive model described above, another common interaction model is based on event-based dispatch of resources by Parties. Resources include both generation resources and curtailment resources. Curtailment resources provide reductions in delivery to a customer from a baseline amount; such resources are typically treated as generation resources, usually in the context of events where shortages may occur. Curtailment resources are also called demand response (DR) resources. For DR resources the determination of the baseline is outside the scope of EI.

3.2.1 VTN and VEN Party Roles

Similar to the Party interactions of transactive energy, event interactions also have an interoperation model between two or more Actors. One designated Actor (for that given interaction) is called the **Virtual Top Node (VTN)** and the remaining one or more actors are called **Virtual End Node(s), or VEN(s).**[1].

Parties may participate in many interactions concurrently as well as over time. For example, a particular Actor may participate in multiple Demand Response programs, receive price communication from multiple sources, and may in turn distribute signals to additional sets of Parties.

The VTN / VEN Interactions combine and compose multiple sets of pairwise interactions to implement more complex structures. By using simple pairwise interactions, the computational and business complexity for each set of Parties is limited, but the complexity of the overall interaction is not limited.

The VTN and VEN Roles are useful beyond event-based interactions because they provide stereotyping of a wide range of behaviors and interactions in energy markets.

3.2.2 VTN/VEN Interactions

In this section the terminology for roles in VTN/VEN Energy Interoperation interaction patterns is clarified. The description and approach is consistent with the Service-Oriented Architecture Reference Model **[SOA-RM]**. The role of a Party as a VTN or VEN only has meaning within the context of a particular service interaction.

At this level of description the presence of application level acknowledgement of invocations is ignored, as reliable and confirmed delivery would typically be implemented by composition with **[WS-RM]**, **[WS-Reliability]**, **[WS-SecureConversation]** or a similar mechanism. For similar reasons, an actual deployment would compose the necessary security, e.g., **[WS-Security]**, **[SAML]**, **[XACML]**, or **[WS-SecureConversation**]. See Section 13 for a discussion of compositional security.

At this level the typical push or pull patterns for interactions are also ignored but are covered in later sections.

[1] We are indebted to EPRI for the Virtual End Node term [EPRI]

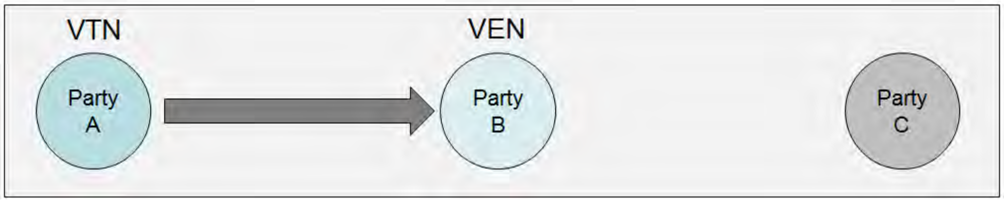


Figure 3-2: Example DR Interaction One

In Figure 3-2, Party A is the VTN with respect to Party B, which acts as the VEN in this interaction. Party C is not a party to this interaction.

Subsequently, as shown Figure 3-3, Party B may act as the VTN for an interaction with Party C, which is acting as the VEN for interaction two. Party A is not a party to this interaction.

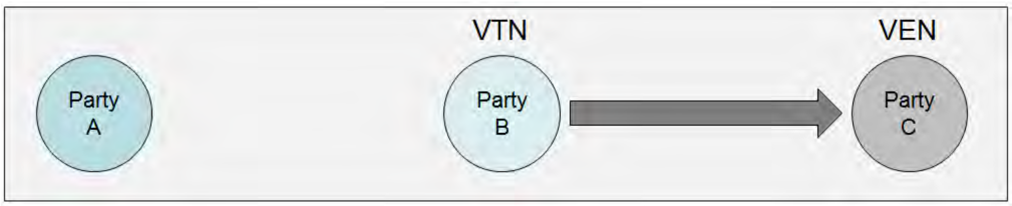


Figure 3-3: Example DR Interaction Two

Moreover, the directionality and the roles of the interaction can change as shown in Figure 3-4.

Again, Party A is not a party to this interaction, but now Party C is the VTN and Party B is the VEN.

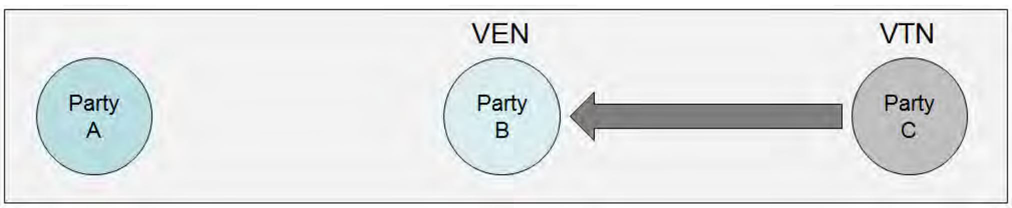


Figure 3-4: Example DR Interaction Three

There is no hierarchy implied by these examples. The examples are used to show how the pairwise interaction patterns and the respective roles that entities play can be composed in ways that are limited only by business needs and feasibility, and not by the architecture. From these simple interactions, one can construct more complex interactions such as those shown in Figure 3-5.

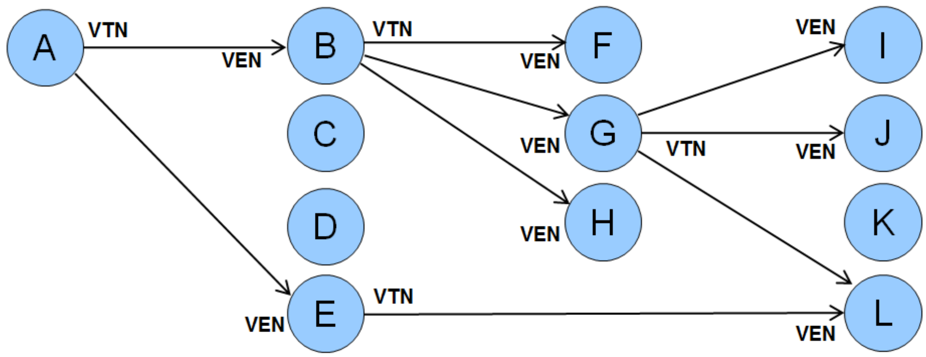


Figure 3-5: Web of Example DR Interactions

In this figure, certain Parties (B, E, and G) act as both VTN and VEN. This directed graph with arrows from VTN to its VENs could model a Reliability DR Event initiated by the Independent System Operator[2] A who would invoke an operation on its second level VTNs B-E, which could be a group of aggregators. The second level VTN B, in turn invokes the same service on its VENs FGH, who may represent their customers or Transactive resources. Those customers might be industrial parks with multiple facilities, real estate developments with multiple tenants, or a company headquarters with facilities in many different geographical areas, who would invoke the same operation on their VENs.

Each interaction can have its own security and reliability composed as needed—the requirements vary for specific interactions.

The following table has sample functional names for selected nodes. *(Note: wrt means "with respect to")*

Table 3-1: Interactions and Actors

Label	Structure Role	Possible Actor Names
A	VTN	System Operator, DR Event Initiator, Microgrid controller, landlord
B	VEN (wrt A), VTN (wrt F, G, H)	Aggregator, microgrid element, tenant, floor, building, factory
G	VEN (wrt B), VTN (wrt I, J, L)	Microgrid controller, building, floor, office suite, process controller, machine
L	VEN (wrt G and wrt E)	Microgrid element, floor, HVAC unit, machine

[2] Using North American Terminology.

3.2.3 VTN/VEN Roles and Services

Two structured roles have been defined for each interaction, the Virtual Top Node (VTN) and the Virtual End Node (VEN). A ***VTN*** has one or more associated ***VENs***.[3]

Considering service interactions for Energy Interoperation, each ***VTN*** may invoke services implemented by one or more of its associated ***VENs***, and each ***VEN*** may invoke services implemented by its associated ***VTN***.

In later sections abstract services that address common transactions are detailed; Demand Response, price distribution, and other use cases.

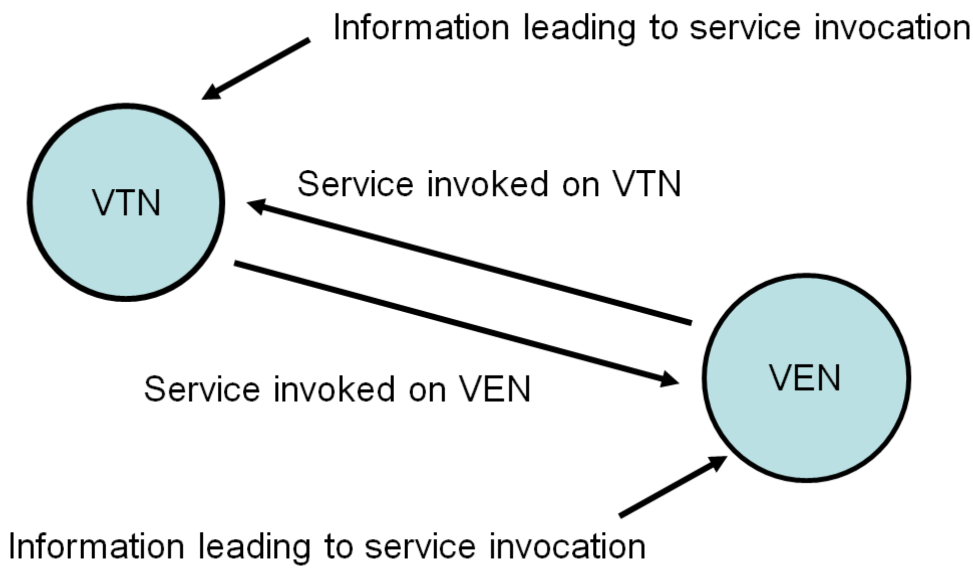


*Figure 3-6: Service Interactions between a **VTN** and a **VEN***

The interacting pairs can be connected into a more complex structure as shown in Figure 3-5.

The relationship of one or more ***VENs*** to a ***VTN*** mirrors common configurations where a VTN (e.g. an aggregator) has many VENs (say its resources under contract) and each VEN works with one VTN for a particular interaction.[4]

Second, as we have seen, each ***VEN*** can implement the ***VTN*** interface for another interaction.

Third, the pattern is recursive as we showed above in Figure 3-3 and allows for more complex structures.[5]

[3] The case of a VTN with zero VENs may be theoretically interesting but has little practical value, hence in a later section VENs having cardinality 1..n are described

[4] The model allows e.g. Demand Resources to participate in more than one interaction, that is, in more than one Demand Response program or offer or with more than one aggregator.

[5] For example, **[OpenADR1.0]** has four actors (the Utility, Demand Response Application Server, the Participant, and the Client (of the Participant). The Energy Interoperation architecture maps clearly to the

Finally, the Parties of the directed interaction graph can be of varying types or classes. In a Reliability DR Event, a System Operator as a VTN may initiate the event with the service invoked on its next level (highest) VENs, and so forth. But the same picture can be used to describe many other kinds of interaction, e.g. interactions to, from, or within a microgrid **[Galvin]**, price and product definition distribution, or distribution and aggregation of projected load and usage.

In some cases the structure graph may permit cycles, in others not.

3.2.4 Demand Response Interactions

In this section the interaction patterns of the services for demand response respectively invoked by an ***VTN*** on one or all of its associated ***VENs*** and vice versa, are described. Figure 3-6: above shows the generic interaction pattern; Figure 3-7 below is specific to Demand Response Events.

By applying the recursive definitions of VTN and VEN specific services will be defined in the following sections (See Figure 3-7)

The VTN invokes operations on its VENs such as Initiate DR Event and Cancel DR Event, while the VEN invokes operations on its VTN such as Create Tender and Create Feedback.

Note not all DR works this way. A customer may be sent a curtailment tender by the DR provider with a price and then can decide to respond. If the customer has agreed to a capacity payment then there may be a loss of payments if he does not respond.

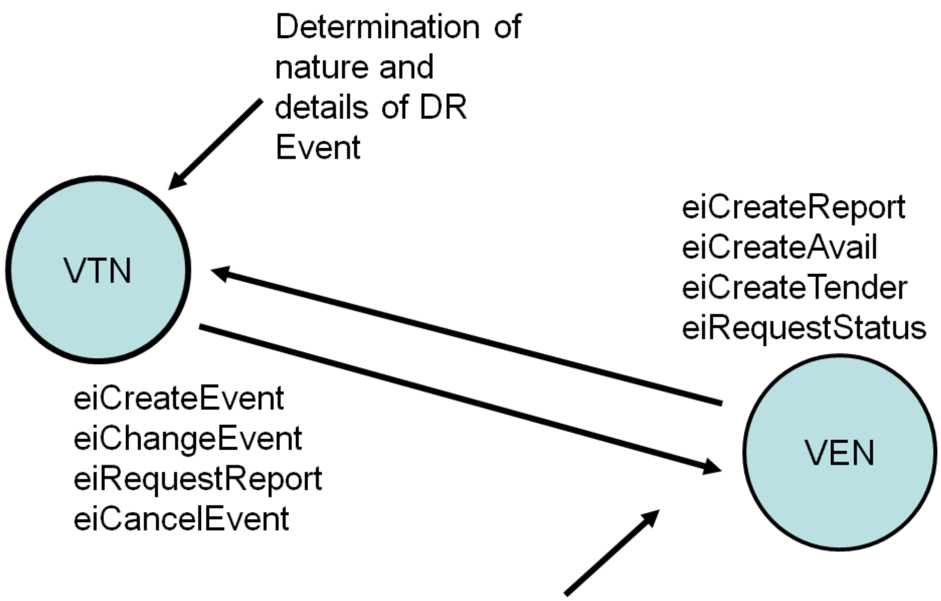


Figure 3-7: Demand Response Interaction Pattern Example

DRAS-Participant interface, and models the Participant-Client interface as an additional VTN-VEN relationship.

3.3 Roles, Resources and Interactions (Non-Normative)

There are many deployments possible, including many not described here. The Committee has striven to make Energy Interoperation agnostic about business processes or business relationships.

3.3.1 Choosing a Role

An Actor finds, discovers, or is configured to use a particular Registrar. By using the EiREgisterParty service, that application obtains a PartyID. With that PartyID, the application can implement and interact using the Party Role in the Transactive Services.

One interaction a Party may participate in is Enrollment. An application may, when it has a PartyID and is identified, Enroll. There are a number of Enrollee Types, reflecting different business roles and enrollments, which are out of scope for this specification—only the names are defined. AN exception is the Resource which extends the EMIX Resource Description Type.

The information required for Enrollment varies across Enroll Administrators. For example in North American wholesale markets, each ISO may potentially require different information or documentation than another. Since that information is out of scope, a deployment or profile would specify what information is required, and convey that information in an extension of the Enrollee types.

Once Enrolled, a Party may have other capabilities, the definition and description of which is also out of scope. The service operations supported are listed in Section 8 *“Enroll Service”.*

The operations for Party Registration and Enrollment are designed, as are all other operations and data types, to be both extensible and evolvable over time to add new or extended functionality to future versions of Energy Interoperation, or by extension of information definitions in specific profiles.

3.3.2 The relationship between Actors and Resources

There is no definitive way to classify an Actor, or a set of capabilities, as an Actor or a Resource. A VEN that is also a VTN may bundle the VENS its interacts with to offer as Resources. In another business model, that VEN may interact with its internal partners through transactive services. Different business structures will drive different technical deployments.

First, an Actor, representing application code, may assume the Virtual End Node (VEN) role. The same application code may also support the Virtual Top Node (VTN) role. This is how the graph of VTNs and VENs in Figure 3-5 is constructed. In that figure, Actor G implements the role of VEN with respect to Actor B, and the role of VTN with respect to Actors I, J, and L.

A Party interacts in transactive environments; the distinction is that a market may have many relationships. While it might seem attractive to make the Actor that interacts with a market take on the VEN role (with the market taking on the VTN role), this is too restrictive. An Actor offer, view, and transact regardless of the VEN/VTN relationships that it maintains--and so the transactive interfaces use Party and CounterParty.

In a deployment one must make decisions about how the roles are selected, discovered, or assigned; this is out of scope of this specification.

In contrast, a Resource is treated as a thing, rather than an Actor. A resource does not participate in relationships such as the Actor/application interfaces in the figure. It could be tempting to require that a Resource is related to (or possibly "managed by") exactly one Actor, a VEN in the Energy Interoperation architecture. It could seem clearest to assert a one-to-one relation between this VEN and the Resource. This would allow requests, reports, and other interactions to and from a single VEN which is uniquely related to that Resource.

But other business cases would be simpler with potentially many Resources managed by a single VEN. In a transactive environment, that VEN may offer capabilities of its individual or groups of Resources to a market (as a Party), and without requiring the defined structure of collaborating VENs and VTNs.

For example, a distributed application conforming to this specification MIGHT deploy in one of the following ways:

(a) assign a single Actor presenting the VEN role to each floor of a building, and a VTN related to them. For external interactions, that VTN for the building would present the VEN interface to receive and interact with the Energy Interoperation Services, and could present the Party role to tender, buy, and sell in a market,
(b) assign a single Actor presenting the VEN role to the building controller, and use other services to manage or convey information to the floor controllers
(c) assign a single Actor presenting the VEN role at the building controller, have that same Actor present the VTN role to the individual floor controllers. The floor controllers present the VEN role to the building controller, while presenting the VTN role to its devices, each of which presents the VEN role to the floor controller.

Were this specification to require exactly one Resource to one VEN, such multiplicity of deployment would not be possible.

4 Message Composition & Services

Energy Interoperation relies on two other standards, Energy Market Information eXchange (**[EMIX]**) and **[WS-Calendar]** to express intents.

- EMIX describes price and product for electricity markets.
- WS-Calendar communicates schedules and sequences of operations.
- Energy Interoperation uses the vocabulary and information models defined by those specifications to describe many of the services that it provides.

4.1 WS-Calendar in Energy Interoperation

[WS-Calendar] defines how to use the semantics of the enterprise calendar communications within service communications. Energy Interoperation is conformant with the **[WS-Calendar]** specification for communicating duration and time to define a Schedule. **[WS-Calendar]** itself extends the well-known semantics of **[RFC5545]**. The communication of a commonly understood Schedule is essential to Energy Interoperation.

Energy Interoperation also relies on **[EMIX]**, which defines schedules and types conforming to WS-Calendar. Energy Interoperation is conformant with the **[WS-Calendar]** specification for communicating duration and time to define a Schedule.

4.1.1 Schedule Semantics from WS-Calendar (Non-Normative)

Without an understanding of certain terms defined in **[WS-Calendar]**, the reader may have difficulty achieving complete understanding of their use in this standard. The table below provides summary descriptions of certain key terms from that specification. This specification does not redefine these terms; they are listed here solely as a convenience to the reader.

Table 4-1: Core Semantics from WS-Calendar

WS-Calendar Term	Description
Component	In **[iCalendar]**, the primary information structure is a Component, also referred to as a "vcomponent." A Component is refined by Parameters and can itself contain Components. Several RFCs have extended iCalendar by defining new Components using the common semantics defined in that specification. In the list below, Interval, Gluon, and Availability are Components. Duration, Link, and Relationship are Parameters. A Sequence is set of Components, primarily Intervals and Gluons, but is not itself a Type.
Duration	Duration is the length of time for an event scheduled using iCalendar or any of its derivatives. The **[XCAL]** duration is a data type using the string representation defined in the iCalendar (**[RFC5545]**) Duration.
Interval	The Interval is a single discrete segment, an element of a Sequence, and expressed with a Duration. The Interval is derived from the common calendar Components. An Interval is part of a Sequence.
Sequence	A set of Intervals with defined temporal relationships. Sequences may have gaps between Intervals, or even simultaneous activities. A Sequence is re-locatable, i.e., it does not have a specific date and time. A Sequence may consist of a single Interval, and can be scheduled by scheduling that single Interval in that Sequence.

WS-Calendar Term	Description
Gluon	A Gluon influences the serialization of Intervals in a Sequence, through inheritance and through schedule setting. The Gluon is similar to the Interval, but has no service or schedule effects until applied to an Interval or Sequence.
Artifact	The placeholder in an Component that holds that thing that occurs during an Interval. **[**EMIX Product Descriptions populate Schedules as Artifacts inside Intervals. In Streams, this specification refers to the Payload conveyed by an Interval.
Link	A reference to an internal object within the same calendar, or an external object in a remote system. The Link is used by one **[WS-Calendar]** Component to reference another.
Relationship	Links between Components.
Availability	Availability in this specification refers to the Vavailability Component, itself a collection of recurring Availability parameters each of which expresses set of Availability Windows. In this specification, these Windows may indicate when an Interval or Sequence can be Scheduled, or when a partner can be notified, or even when it cannot be Scheduled.

Normative descriptions of the terms in the table above are in **[WS-Calendar]**.

4.1.2 Schedules and Inheritance

Nearly every response, every event, and every interaction in Energy Interoperation (with the exception of all single interval TeMIX profile interactions) can have payloads with values that vary over time, i.e., it is described using a sequence of intervals. Many communications, particularly in today's retail market, involve information about or a request for power delivered over a single interval of time. Simplicity and parsimony of expression must coexist with complexity and syntactical richness.

The simplest power description in **[EMIX]** is Transactive power. The simplest demand response is to reduce power. The power object in EMIX can include specification of voltage, and Hertz and quality and other features. There are market interactions where each all of those are necessary. Reduced to its simplest, though, the EMIX Power information consists of Power Units and Power Quantity: as in

Units:	KW	Quantity	10

Figure 4-1: Basic Power Object from EMIX

At its simplest, though, WS-Calendar expresses repeating intervals of the same duration, one after the other, and something that changes over the course of the schedule

Start:	8:00	Duration:	1Hour		
		Duration:	1Hour		
		Duration:	1Hour		
		Duration:	1Hour		
		Duration:	1Hour		

Figure 4-2: WS-Calendar Partition, a simple sequence of 5 intervals

The WS-Calendar specification defines how to spread an object like the first over the schedule. The information that is true for every interval is expressed once only. The information that changes during each interval, is expressed as part of each interval.*

Units	KW	Start:	8:00	Duration:	1Hour	Quantity	10
				Duration:	1Hour	Quantity	10
				Duration:	1Hour	Quantity	15
				Duration:	1Hour	Quantity	25
				Duration:	1Hour	Quantity	10 *

Figure 4-3: Applying Basic Power to a Sequence

Many communications communicate requirements for a single interval. When expressing market information about a single interval, the market object (Power) and the single interval collapse to a simple model:

Units	KW	Start:	8:00	Duration:	1Hour	Quantity	10

Figure 4-4: Simplifying back to Power in a Single Interval

WS-Calendar calls this pattern Inheritance and specifies a number of rules that govern Inheritance. Table 4-2 summarizes those terms defined in WS-Calendar to describe Inheritance that are used in this specification as well. This specification does not redefine these terms; they are listed here solely as a convenience to the reader.

Table 4-2: WS-Calendar Semantics: Inheritance

Term	Definition
Lineage	The ordered set of Parents that results in a given inheritance or execution context for a Sequence.
Inherit	A Child Inherits attributes (Inheritance) from its Parent.
Inheritance	A pattern by which information in Sequence is completed or modified by information from a Gluon. Information specified in one informational object is considered present in another that is itself lacking expression of that information.
Bequeath	A Parent Bequeaths attributes (Inheritance) to its Children.

This specification extends the use of Inheritance as defined in WS-Calendar. Most interactions specify a schedule, whether for price Quote or for Demand Response event. These schedules are expressed in Streams (see Section 4.3). Each Interval in the Schedule contains an information payload. Each of these payloads is completed through inheriting information from the Stream as if from a Gluon. The Stream itself inherits information from the context of the interaction, especially from the Market Context, as if from Gluon.

A Market Context Bequeaths essential information to a Stream, which in turn its information to each Interval in the Stream. This specification uses this pattern of expression throughout.

4.1.3 Availability and Schedules

The WS-Calendar component Availability is used throughout Energy Interoperation. Availability expresses recurring patterns of schedule within a bounded period of time. This specification uses Availability in market definitions and in a number of inter-party commitments and communications. Availability is used to define windows for Demand Response, to define when during a given day a Party may receive requests, and for expressing the desire of a Party to place or remove services from markets.

While the expression of Availability is defined in WS-Calendar, the Committee recommends the informative discussion of Availability found in **[Vavailability]**.

4.1.4 Smoothing Response

Precision of communication and response causes special problems for large collections of entities and systems, as well as for switching of high electrical demand as in substations or with large electric motors. When devices interact at high speeds to change demand, they can create sharp spikes up or down in demand. These spikes can affect other nodes on a grid, cause a grid to crash, or even destroy equipment.

WS-Calendar defines Tolerance as an optional Property of Intervals that expresses allowable imprecision. Tolerance may have up to 5 parameters: Start Before Tolerance, Start After Tolerance, End Before Tolerance, End After Tolerance, and Precision.

For example, Start Before Tolerance may have a value of ten minutes. In the same Interval, Start After Tolerance may have a value of five minutes. Let us further specify that the Interval starts at 3:00 PM with a Duration of two hours. WS-Calendar then has expressed that the recipient begin its response at 3:00 and continue for two hours, but that a response that begins any time between 2:50 pm and 3:05 pm is acceptable.

For convenience, this specification refers to the Tolerance Interval as either the sum of the starting tolerances (Start Before Tolerance and Start After Tolerance) or the sum of the ending tolerances (End Before Tolerance and End After Tolerance).

Because Sequences are constructed of linked intervals expressed as Durations, Tolerance applied only to the Designated Interval in a Sequence can change the interpretation of the entire Sequence. If the Designated Interval begins five minutes late and lasts one hour, then the second Interval, which is anchored by the first, will also begin five minutes late, and so on.

The Smart Grid is a system of systems, and each system provides its respective class of application. Some systems are aggregates of hundreds or thousands of similar systems. Other Systems contain many internal systems with their own dependencies and interactions. Still others may consist of a single large system. Each of these represents a different application.

- Applications managing small loads may be required to randomize their start time within the Tolerance Interval. Conformance requirements for a deployment must specify how this randomization is demonstrated or evaluated for a particular application.
- Applications internally managing collections of smaller loads may be required to spread the starts and stops of each internal system to produce a load that moves in steps over the Tolerance Interval. Different systems may do this differently. Integrated systems will sequence their internal loads to manage internal cross-dependencies. Less integrated systems may randomize the starts of their internal systems. Conformance for these applications may include a minimum spread of steps or a maximum quantum change of load.
- Applications that front single large loads may be required to gradually ramp between the initial state and the requested response across the Tolerance Interval.

Conformance to these deployment scenarios is outside the scope of this specification.

4.2 EMIX in Energy Interoperation

Energy Interoperation uses EMIX to express the semantics of Power and Energy Markets.

In **[EMIX]** Product Descriptions define Energy and Power. Product Descriptions are applied to Sequences to create Schedules. Schedules conform to the inheritance pattern defined in **[WS-Calendar]** to reduce repetition of these descriptive elements. **[EMIX]** Products include an entire Schedule along with transactive information. **[EMIX]** Options use Availability to describe market information for the right to acquire Energy during certain periods at specified Rates. TeMIX defines communications for transactions of energy delivered at specified rates over specific intervals.

Each of the elements above is associated with a Market Context. A Market Context may be associated with Standard Terms which may define an overriding set of information for products therein. An **[EMIX]** Schedule can inherit information from the Standard Terms in a Market just as a WS-Calendar Sequence inherits from a Gluon.

Every Energy Interoperation interaction MAY convey an EMIX Type. Often they convey simplified derivations of **[EMIX]** types that use conformance and inheritance to reduce to a bare minimum, while still using EMIX semantics.

Energy Interoperation defines Parties which enroll with Counter-Parties. These Parties may then participate directly in energy transactions, using the Semantics from TEMIX. Others enroll as Resources with certain capabilities. Some of these Resources may share detailed capability and response information with their counter-party using the EMIX Resource semantics.

4.2.1 Core Semantics from EMIX

The terms in Table 4-3 are normatively defined in **[EMIX]**. Summary descriptions are provided here for the convenience of the reader only.

Table 4-3: EMIX Essential Semantics

EMIX Term	Description
Item Base	Abstract base type for units for EMIX Products. Item Base does not include Quantity or Price, because a single Product may have multiple quantities or prices associated with each Interval.
Schedule	EMIX Products are delivered for a Duration, at a particular time. EMIX relies on the Interval and the Gluon as defined in **[WS-Calendar].**
Product Description	The Product Description is the payload inside each Interval of the Schedule. The Product Description conveys the characteristics of the Power or Resource or Transport Product. Each Interval may hold an incomplete Product Description, one that can completed using the rules of Inheritance described in WS-Calendar.
EMIX Base	The EMIX Base conveys a Schedule populated with Product Descriptions and is intended to express additional market information sufficient to define Products.
Price Base	The PriceBase conveys a Price, a Relative Price, or a Price Multiplier.
EMIX Interface	Abstract base class for the interfaces for EMIX Product delivery, measurement, and/or pricing. The PNode and the Service Area are examples of the EMIX Interface.
Market Context	A URI uniquely identifying a source for market terms, market rules, market prices, etc.
EMIX Product	A Product Description applied to a Schedule. Using the Gluon / Sequence pattern of inheritance, there may be a nearly complete Product Description in the element that acts as a Gluon, and only elements that change in each interval.
EMIX Option	A Type of Product in which for a defined price, a party agrees to make Product available during a schedule (Availability) to be delivered at the counterparty’s request, in accord with agreed upon terms and at an agreed upon price.
Transactive State	An indicator included in EMIX Base derived types to aid in processing. The enumerated Transactive States are: Indication Of Interest, Tender, Transaction, Exercise, Delivery, Transport Commitment, and Publication.
Terms	Terms are used in EMIX to describe when and how a product is available. Minimum Notification Duration, Maximum Run Duration, and Minimum Remuneration per Event are all Terms.

EMIX Term	Description
Service Area	The Service Area is the only Interface defined for all derived schemas. The Service Area expresses locations or geographic regions relevant to price communication. For example, a change in price for a power product could apply to all customers in an urban area.
Power	The EMIX Power schema defines products related to the exchange of Electrical Power using the EMIX semantics.
Resource	The EMIX Resource schema defines the capabilities that a node has to deliver Power products.
Ancillary Service	Ancillary Services are typically products provided by a Resource contracted to stand by for a request to deliver changes in power to balance the grid on short notice.

The terms in Table 4-3 are defined normatively in EMIX and nothing in this specification changes or overrides those definitions.

4.2.2 Putting EMIX in Context

EMIX specifies that information that does not change can be summarized using standard Terms associated with a Market Context.

Table 4-4: EMIX Market Context

Expectations and Contexts	Description
Market Context	Defines the product, performance expectations and rules for interactions. All Events Signals, and Transactions occur within a market context. A Market Context acts as a Gluon for all sequences described in the EI Types. Market Contexts are described using the semantics of EMIX Standard Terms.
Availability	Describes when a Resource is available to respond relative to a particular VTN and Market Context
Market Expectations	Market Expectations are associated with a Market Context and consist of a number of Rule Sets.
Standard Terms	Standard Terms apply to all transactions in a Market Context. When they are conveyed as Standard Terms, they do not need to be repeated in individual interactions. A product references a Market Context and all Standard Terms associated with that Market Context.
Granularity	Granularity is the units of time used in operating a market transacts power in one hour increments. A One hour market is for one-hour purchases of Power with each interval in a modulo offset from the beginning of the business schedule
Non-Standard Terms Handling	Non-Standard terms handling defines what Parties should do with any Term not listed in the Market Rule Sets.
Market Rule-Set	A collection of Terms and how they are processed within this market. A Rule Set includes a Purpose to guide its interpretation.

Expectations and Contexts	Description
Rule Set Purpose	Defines the purpose of a Rule Set, i.e., to define minimum performance, maximum performance, etc.

The terms in Table 4-4 are defined normatively in EMIX and nothing in this specification is changes or overrides those definitions.

4.3 Streams: Adaptations of WS-Calendar for Energy Interoperation

Streams use WS-Calendar Sequences to convey a time sequence of prices, usage, demand, response, or anything else that varies over time. Streams are used both for projections of the future and for reports about the past; event signals and reports are each instances of Streams.

WS-Calendar specifies that Sequences that describe a Service be expressed as Duration within each Interval, Temporal Relations between those intervals, and a single Start or End time for the Sequence. WS-Calendar specifies that each Interval have a unique identifier (UID). WS-Calendar further specifies that each Interval include a Temporal Relation, either direct or transitive, with all other Intervals in a Sequence. A Temporal Relation consists of the Relationship, the UID of the related Interval, and the optional Gap between Intervals.

[WS-Calendar] defines a Partition as a Sequence of consecutive Intervals.

All Streams follow the Gluon-Sequence pattern from WS-Calendar, i.e., the Stream acts a Gluon that optionally contains a degenerate Sequence. Information valid for the entire stream is indicated in the Gluon, i.e., external to the Intervals of the Sequence. Only information that changes over time is contained within each interval. This changing information is referred to herein as the Payload.

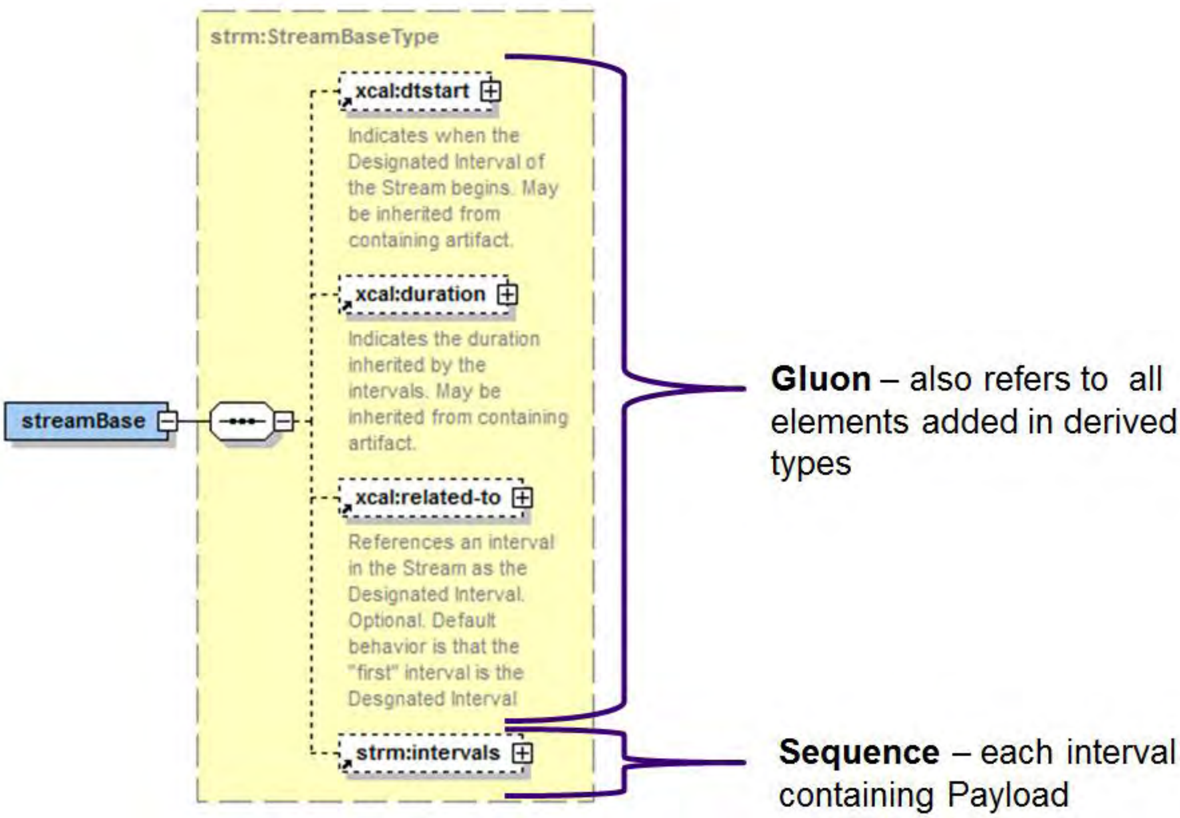


Figure 4-5: Stream as Gluon and Sequence

For example, an Event establishes in a context specified by Enrollment and each Signal arises within a Market Context and within that Event Base. The information contained in the Event Base MAY inherit information in the Market Context as an Interval or Gluon inherits information from a Gluon. WS-Calendar calls this the *lineage* of the information.

That Market Context may include Standard Terms, Product Description, Time Zone Identifier (TZID), and Simple Level Definition. The Market Context enters the Lineage (as described in WS-Calendar) of the Schedule as if the Market Context were contained in a Gluon. Product Description, TZID, Program Definition, Terms, et al. can be inherited in this manner. Again, following the WS-Calendar inheritance pattern, each Interval in the Sequence inherits from the Lineage described above.

4.3.1 UML Diagram of Stream

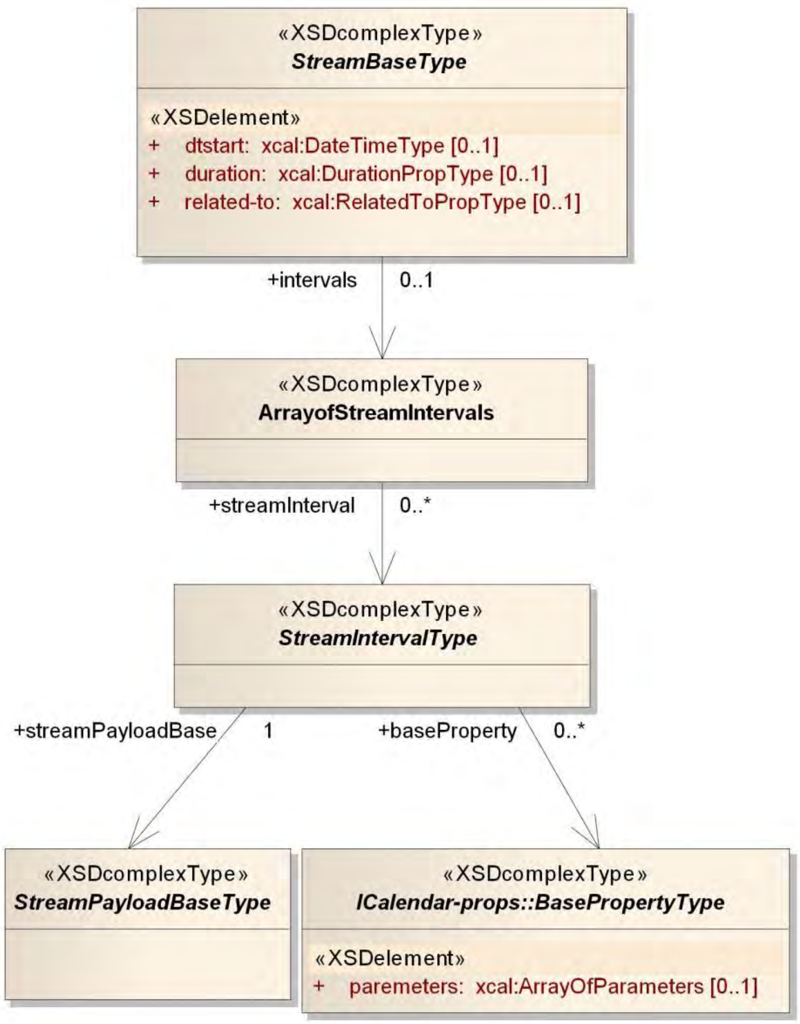


Figure 4-6: UML Class Diagram of abstract StreamBase class

4.3.2 Conformance of Streams to WS-Calendar

If it is necessary to process a Stream through standard Calendar communications, the Stream's GUID is the key and the Stream is processed as if a Gluon. All Sequence information MAY remain internal to that Gluon. If it is necessary to instantiate Interval in the Sequence as a WS-Calendar Interval, the GUID for each is derived by appending the Sequence ID to the Stream's GUID.

4.3.2.1 Stream expression of Intervals expressed as Durations

While conformant communications can include anything expressible in [WS-Calendar], this specification further defines standard profiles of Sequences and Intervals for use in Streams.

Streams describe Partitions. Within a Stream expressed using Durations, a virtual UID for each Interval MAY be constructed by concatenating the Stream Identifier, which may include the identity of the source or recipient, and a sequence number. Within a Stream, this UID can be expressed within each interval by the sequence number alone.

If the Designated Interval in a Sequence within a Stream omits a Temporal Relationship, then all Intervals in the Sequence MAY NOT include a Temporal Relation. Such intervals are sorted by increasing sequence number (expressed in the UID), and each Interval is treated as if it contained an implied FinishToStart relation to the next Interval with a Gap of zero Duration.

Partitions expressed in this way consist of Intervals containing only a Sequence Number, the Duration of the Interval (if not inherited), and the Market Signal Payload. The effect of this is that Stream Intervals are ordered as a Partition in order of increasing UID.

WS-Calendar inheritance defines a Lineage whereby Intervals inherit information from Gluons. In Energy Interoperation, Streams are contained in larger messages. A Stream MAY inherit information from its containing message as if from a Gluon. A Stream-derived Type may contain information external to the Sequence. This information inherits acts as if it were a Gluon, inheriting from the containing message, and Bequeathing information to the designated interval in the Sequence.

The first (in time and in sequence number) Interval in the Sequence in a Stream is the Designated Interval unless another Interval is explicitly so designated in the Stream Event. Signals, Reports, and many other messages use this pattern of expression. For example, the Active Period of an Event Bequeaths its start date and time to an Event Signal which Bequeaths that to the Designated Interval in the sequence. These terms are defined below.

4.3.2.2 Observational Data expressed as Streams

Observed information may be best communicated as raw data without interpretation. A single set of Observations may be re-purposed or re-processed for multiple uses. For example, a measurement recorded at 3:15 may be a point in both a 5 minute series and a 15 minute series. Observational data may have known errors that can be lost in processing. Low-end sensor systems may not update instantly. For example, a reading taken at 4:30 may be known to actually have been recorded at 4:27. Streams expressing a series of observations MAY use the date and times rather than the duration as their primary temporal element.

When the boundaries of Intervals in a Stream are expressed with Date and Time, then all Intervals in that Sequence SHALL be expressed with a Date and Time and that boundary selected SHALL be the Same, i.e., all Intervals MAY be expressed with a Begin Date and Time OR with an End Date and Time. For observations, use the End Date and Time.

Within a Stream expressed using Dates and Times, a virtual UID for each Interval MAY be constructed by concatenating the Signal Identifier, the PartyID (which may be the VEN ID), and the Date and Time. Within an Observational Stream, this UID can be expressed within each interval by the End Date and Time alone. Intervals in a Sequence expressed this way are treated as if each contains an implied FinishToStart relation to the next Interval with a Gap of zero duration. The Duration of each Interval can be computed by using the Date(s) and Time(s) of adjacent Intervals.

4.3.3 Payload Optimization in Streams

As defined in WS-Calendar and in EMIX, each Interval in a Sequence potentially contains any artifact that inherits/extends the EMIX Product Description Type as a payload. As used in Streams, the EMIX Artifact is expressed once or inherited from the Market Context. Each Interval in a Stream expresses only the common subset of facts that varies within the context of the Stream. For efficient communication and processing, Streams use these explicit processing rules:

1. Unless each interval includes a full EMIX payload, each Interval in a Stream expresses only the defined subset of the payload that varies over time.
2. Each Interval in a Stream uses the same payload subset as all other intervals in that stream.

All streams in this specification share a common Payload base. This commonality is derived from the commonality of a request for performance (Signal), a report of performance (Report and Delivery), projections of performance (Projection), and a baseline of performance (Baseline).

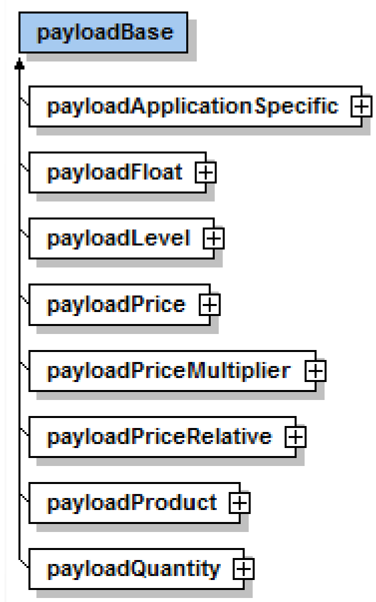


Figure 4-7: Payload Base

4.3.4 Other elements in Stream Payloads

It may be necessary to qualify information about intervals in the future. The element Interval Qualification extends the WS-Calendar Property. [All Intervals have a collection of Properties]. Energy Interoperation uses Qualifications to indicate the originator's indications as to how the sender should rely on the information in the Payload.

Qualifications MAY be used in Quotes, in Load and Response projections, and in Observations. They MAY NOT be used in other transactive states.

It may be necessary to qualify measurements delivered in a report. Devices have known accuracies. Several Measurements MAY be added together to create a single quantity. To support these uncertainties different payloads are defined for different services.

Each of streams in Energy Interoperation, Signals, Baselines, Reports, and Delivery is discussed below. All four payloads are shown together in Figure 4-8: Comparing Payloads for Signals, Baselines, Reports, and Delivery.

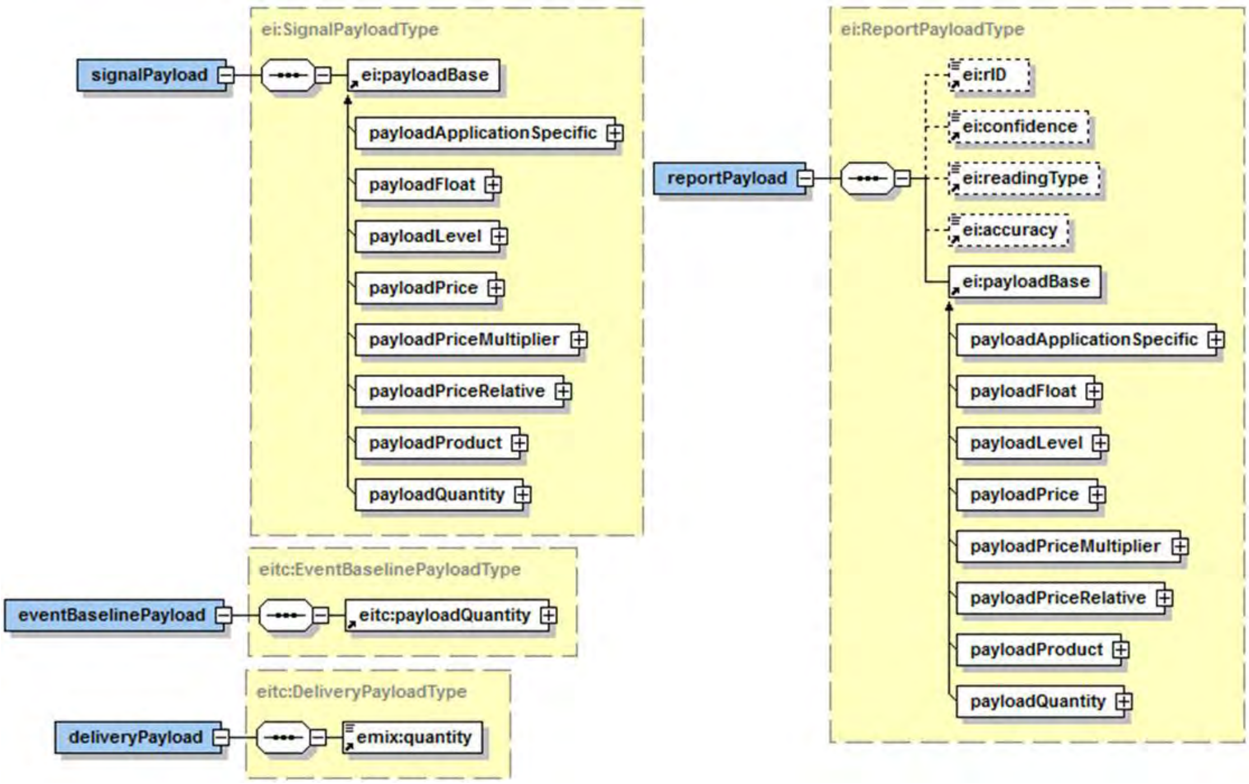


Figure 4-8: Comparing Payloads for Signals, Baselines, Reports, and Delivery

4.4 Applying EMIX and WS-Calendar to a Power Event

Consider the event in Figure 4-9. This event illustrates the potential complexity of marshaling a load response from a VEN, perhaps a commercial building.

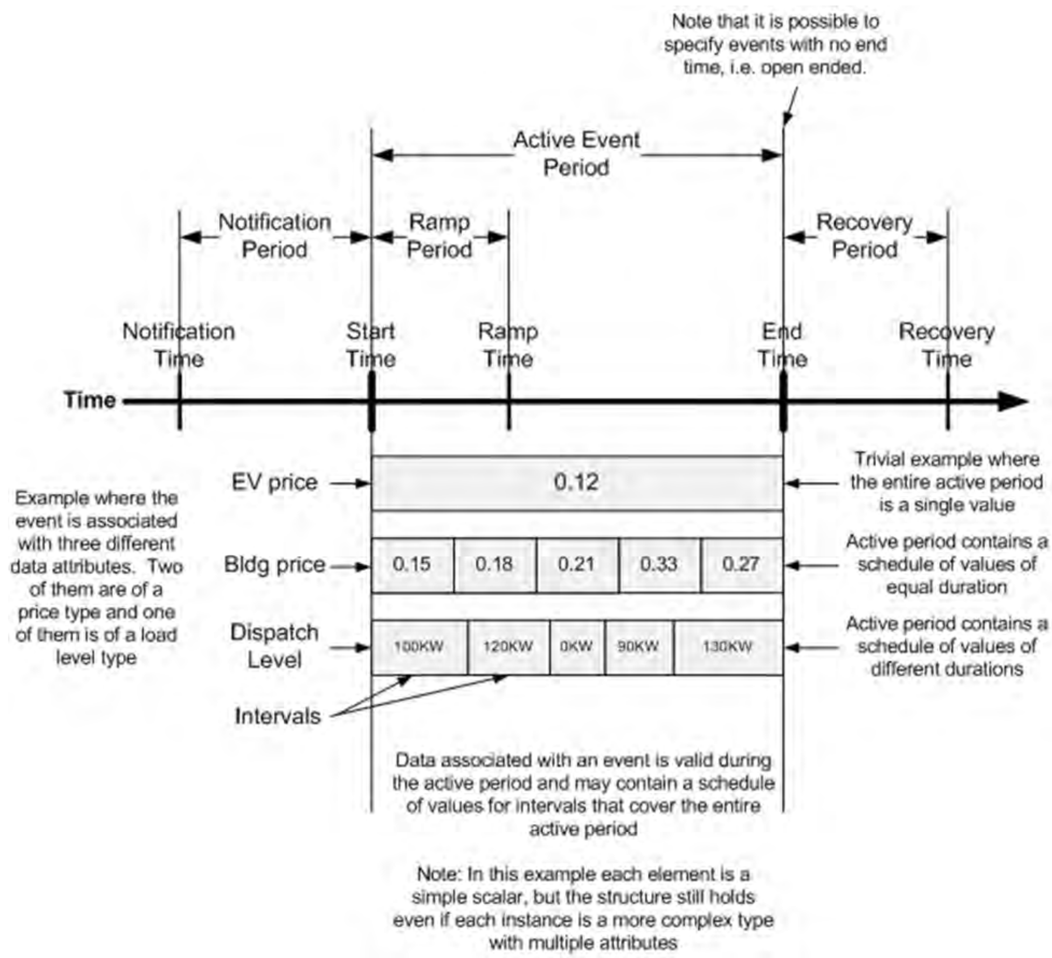


Figure 4-9 Demand Response Event and associated Streams

Note first that there are two schedules of prices. The price of electricity for the building “bldg price” is rising to more than double its original price of $0.15 during the interval. The price for Electric Vehicles (EV) is fixed at the lower-than-market rate of $0.12, perhaps because public policy is set to encourage their use. Each of those price curves has an EMIX description.

In the language of EMIX and WS-Calendar, this Event contains two Resources and three Schedules. The Resources are the Electric Vehicle and the Building. The Vehicle receives one schedule of Prices. The Building receives two schedules, one dispatch based, and one price based. Both resources are located within the VEN, and any decisions about how to respond to the event are made within the VEN which is the sole point of communication for the VTN.

The duration that encompasses the event is known as the Active Period for the event. Before and after the event, there is a notification period and a recovery period, respectively. These are fixed durations communicated from the VEN to the VTN, which then must respect them in transactions it awards the VEN.

4.4.1 Streams in a DR Event

The three schedules above are conveyed using Signals which are expressed as Streams as defined above.

The dispatch level, i.e., the load reduction made by the building, varies over time. This may be tied to building capabilities, or to maintaining essential services for the occupants. It is not important to the VTN why it is constrained, only that it is.

Note that the reductions in Figure 4-9 do not line up with the price intervals on the bar above. In this example, the dispatch level is applied to its own WS-Calendar sequence. There is no requirement that intervals in separate streams in an event align.

An Event may be associated with Observational Streams to report back to the requester information measured or derived during the event.

4.4.2 The Active Period

The Active Period is a special schedule for the overall description of an Event. The Active Period may have commercial and regulatory meaning, such as a rule requiring that an Event not be longer than two hours. While an Event as described below may have many schedules as expressed in Streams, it has one Active Period.

The Active period of an event typically includes intervals in which the receiving system prepares for the event, begins its response, maintains its response, and recovers from the response. The schedules for these activities MAY be expressed using EMIX artifacts. For Power communications these can be expressed using artifacts based on EMIX Resources. The schedule for an Event MAY be expressed as can any other Sequence.

More commonly, the Active Period is expressed through a single Interval. The properties of WS-Calendar are extended in this specification to include durations to indicate the notification, ram, and recovery periods. These are interpreted as if they are a normal sequence, constructed as indicated in Table 4-5.

Table 4-5: Semantics of the Active Period

Active Period elements	Description
Active Period	The nominal period of the Event. Expressed as a Vcalendar containing the Active Interval and supporting schedule information.
Active Interval	Interval within the Active Period whose Start Time and Duration define the period. The Active Interval may be the Designated Interval in the Sequence in the Active Period or it may be a specialized Interval as described above.
Notification Period	Nominally, the period expressed as a Duration between notification of the event and the commencement of the Active Interval. In distributed scenarios, a VEN may receive notification before or after this moment. Constrained devices may increase energy use during the Notification Period so as to be able to reduce energy use during the Active Interval.
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.

Active Period elements	Description
Tolerance	A collection of parameters that indicate whether there is a range of acceptable starting and ending times for the Active Period. Tolerance is used to smooth the response so that thousands of systems do not change state at the same moment.

5 Semantics of Energy Interoperation

As stated in in Section 4, much of the core vocabulary for this specification comes from **[EMIX]** and **[WS-Calendar]**. This section introduces the remaining vocabulary for Energy Interoperation and then defines the use of that vocabulary in the higher level types.

The services of Energy Interoperation are built around exchanges of and references to these standard information artifacts.

Table 5-1: Energy Interoperation Identities

Identity Types	Description
Party	As described in Section 3, all interactions are between two Parties. A Party consists of a Party Id, a Party Name, and a Party Role. The Party ID is a sub-type of the UID.
Resource	Identifies a discrete set of capabilities that a Party may offer to a counterparty. Resources may represent specific equipment, collections of market interactions, or a detailed promise to perform. Resources are associated with a VEN during Enrollment
Market	When used in this specification, a Market is a set of agreed upon assumptions and business practices. Tariffs and utility programs are examples of Markets. Each negotiation and transaction occurs within the named context of a Market.
Market Context	A collection of machine readable Market rules and assumptions. A Market Context is uniquely identified by a URI as defined by the EMIX Market Context. This URI can be used to retrieve the Context.
UID	Unique Identifier for every party, role, message, event, etc.

The elements above are used throughout the messages of this specification.

5.1 Dramatis Personae: identifying the Actors

As described in Section 3, each interaction is an interaction between two parties.

Low Level Identity Types	Description
VEN	As described in section 3 above, A Virtual End Node is a Party acting in a specific role in a market managed by a VTN.
VTN	As described in section 3 above, A Virtual Top Node is a Party acting in a specific role that sends events market information to a VEN.
Group	Resources and VENs may be the target of an Event. How group membership is identified or recognized is out of scope.
Target	A set of elements to that collectively name which Parties should participate in an event. A Target can include Service Areas, named Groups, VENs, and Resources and other standard identifiers. The Target can be used by VEN's that are also VTN's and must relay event information downstream to other VENs.

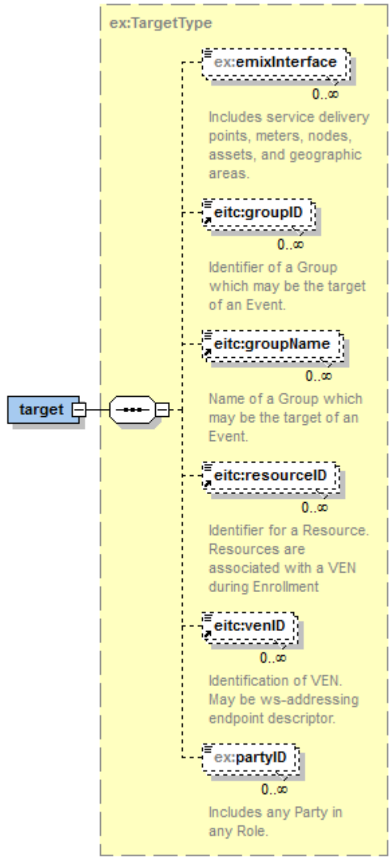


Figure 5-1: EI Target

5.1.1 Actor IDs and Roles

There is a certain fungibility of the Actor IDs in the service payloads. A Party may participate in many interactions, yet it is necessary to distinguish each Party by the role it is playing the current interaction. Accordingly, there are named derivatives of the Actor ID for use in each situation.

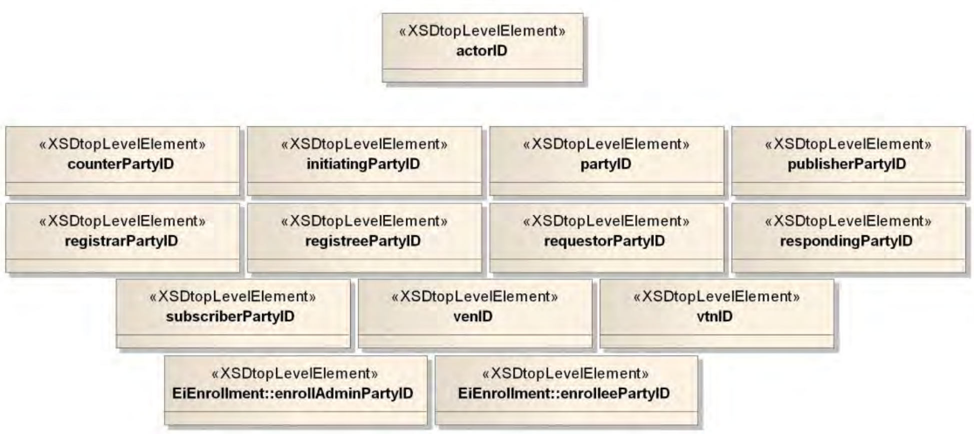


Figure 5-2: UML Class Diagram of Party ID and its derivatives

5.2 Market Context

As defined in **[EMIX]**, a Market Context is a URI, and it can be used to reference Standard Terms. This specification describes the expanded set of context information that is part of the EI Market Context.

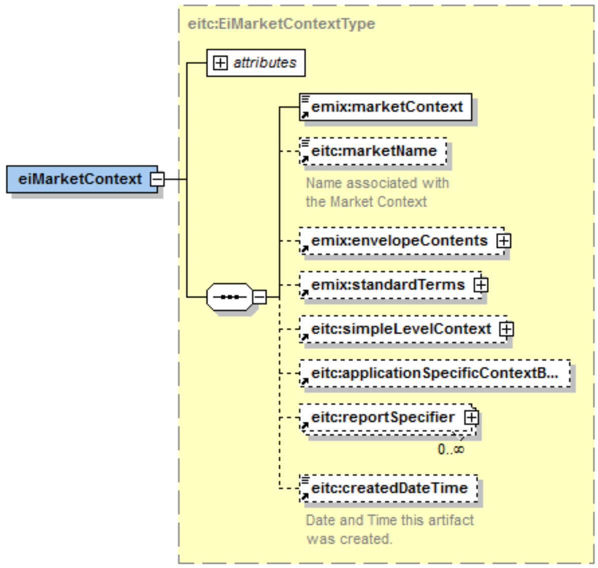


Figure 5-3: EI Market Context

The Elements of the EI Market Context are, for the most part, defined in **[EMIX]**. The Market Name conveys a human-readable text, perhaps for display in a user interface. As in EMIX, the Envelope contains warrants and certificates. For example, if a Market is purported to convey Green Power, however defined, that information would be conveyed in the Envelope. Two elements, Simple Levels and Application Specific Extensions bear discussion here.

5.2.1 Simple Levels

The Simple Level Context is an agreement-based interaction abstracted away from expressions of value or actual amounts. Simple Levels define levels of energy scarcity and abundance, at an agreed upon granularity. A VEN can discover Specific Levels within a Market Context.

Table 5-2: Simple Levels

Level Information	Description
Simple Level Context	Simple Levels are a set of simple indicators about scarcity and value, in which an ordered set of values indicate energy scarcity is above normal, normal, or below normal. Presumably, at higher levels, the VEN will use less.
Upper Limit	The upper level for this Context. If the Upper Limit is 5, the levels are 1-5, where 5 indicates the greatest scarcity.
Normal Value	The "normal" level indicating normal energy availability. Levels below normal indicate surplus, levels above normal indicate increasing scarcity. If the Upper Limit is 7, the levels are 1-7, and the Normal Value might be 3.
Level	Payload used in Signals to convey Simple Level to a VEN

For example, a simple program may have the levels Normal, High, and Critical. The Simple Level Context would indicate three levels with a normal value of one.

How a VEN associates particular activities and responses to the Simple Levels is out of scope for this specification.

5.2.2 Application Specific Extensions

A VTN may wish to communicate with, and a VEN may wish to allow communication with a specific Application operating within the VEN. Operating such an Application MAY be part of a specific Market Context. This specification provides explicit support for these Application Specific Extensions by means of 4 abstract types.

Table 5-3: Application Specific Extensions

Extensions	Description
Application Specific Extension Base	An abstract Base Type for all other Application Specific Extensions. Application Extensions are used to provide hints to or interactions with Applications running on the other side of an interaction. They are not defined in Energy Interoperation, although there are specific conformance rules that must be followed
Application Specific Context Base	An abstract class to exchange invariant or setup information with an Application running on the other side of an interaction. The Context Base is exchanged as part of a Market Context
Application Specific Signal Base	An abstract class to exchange current information and varying information with an Application running on the other side of an interaction. The Signal Base is exchanged by means of an Event Signal.
Application Specific Report Base	An abstract class to exchange Reports with an Application running on the other side of an interaction. The Report Base is exchanged by means of an Event Report r by the Report Service.

The primary concern of the conformance rules for Application Specific Extensions is that they avoid redefinition of the semantics of Energy Interoperation. Prices SHALL be communicated as defines in EMIX Price Base. Schedules SHALL be communicated using the semantics of WS-Calendar. Products and things to be measured SHALL be expressed using the EMIX Item Base.

Parties wishing to exchange Application Specific Extensions SHALL extend the Signal Types and Report Types to indicate they are using their specific Payloads.

5.2.3 Response Smoothing

Precision of communication and response causes new problems for collections of entities and systems. With WS-Calendar and Energy Interoperation, thousands of systems and devices could respond at the same moment, causing grid instabilities or even equipment damage.

To avoid these problems, Energy Interoperation uses WS-Calendar Tolerances (Start Before, Start After, End Before, and End After) to specify a Duration in which response smoothing MAY be requested.

To further refine the expectation surrounding Smoothing, this specification defines a new Term, i.e., an extension of the EMIX Base Term, to convey expectations for smoothing the aggregate response. Because it is a Term, is can be communicated as part of a Market Context, or as part of an individual Event.

The Smoothing Term provides actionable information; of course the degree of adherence to what is an application or deployment performance characteristic is out of scope for this specification. See also Section 4.1.4.

Table 5-4: Smoothing Terms

Response Smoothing	Description

Response Smoothing	Description
Smoothing	Response Smoothing defines a Term that indicates that the recipient is to ensure that the response is not in a single step. Response Smoothing is applied to the tolerance interval[s] indicated by the Start Before, Start After, End Before, and End After tolerances. The enumerated values of Smoothing are below.
Ramp	A smooth or uniform step ramp is indicated between the initial and end values in the respective Tolerance Interval
Uniform	A uniform distribution is indicated over the entire respective Tolerance Interval.
None	No specific smoothing is indicated. Applications need not react in a stepwise manner, so some degree of smoothing MAY occur in response to this request. If the Smoothing Term is absent, the behavior requested is the same as None.

5.3 Event-based Interactions

Events are stylized business interactions that are used in formal demand response environments. As described in Section 3, Events are used in communications between a VTN and a VEN. An Event consists of the time periods, deadlines, and transitions during which Demand Resources perform. The VTN specifies the duration and applicability of an Event. Some deadlines, time periods, and transitions may not be not applicable to all products or services.

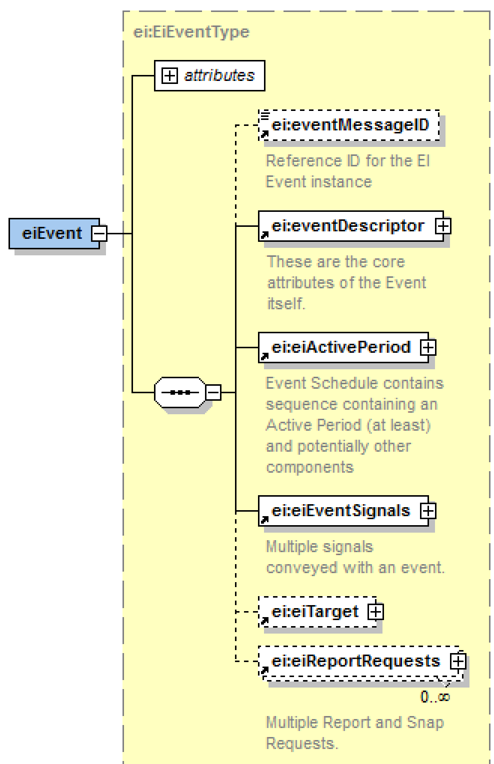


Figure 5-4: Event Overview

5.3.1 The Event Descriptor

The Event descriptor contains metadata about the event itself.

Table 5-5: The Event Descriptor

Event Descriptor Elements	Description
Event Descriptor	A collection of meta-data about an Event
Event ID	Identifier assigned to the Event Descriptor
Modification Number	If present, indicates that the event has been modified. Incremented each time the event is modified.
Modification Date and Time	The date and time a modification takes effect.
Modification Reason	Reason describing why the event is being modified. The values for reason are not specified or restricted.
Priority	Optional indication of the priority of an event. A given VEN or Resource may be eligible for more than one event at the same time.
Market Context	The overall market or program rules that govern this event.
Created Date Time	Indicates when this artifact was created.
Event Status	Indicates the current status of an event as of the descriptor generation. Enumerated values are: • Far: Event is 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. • Near: Event is in the near future. The exact definition of how near in the future the pending event is active is dependent on the market context. • Active: Event has been initiated and is currently active. • Completed: Event has completed. • Cancelled: Event has been canceled. These values are similar but not identical to those used by the Event Filter as described in Section 9.2 *"Special Semantics of the Event Request Operations".* The value is present in Energy Interoperation to support backward compatibility with OpenADR 1.0.
Operating Day	Indicates the nominal date for the event. Important for some market contexts.
Test Event	If present, can indicate that this event is a test event rather than an actual event.
Comment	Free-form information provided by the VTN

5.3.2 The Active Period

The Active Period is a Sequence that describes the overall schedule for an Event. The Active period is a Vcalendar type that contain a Sequence and MAY have its own properties. The Sequence of an Active Period generally falls into a common Interval pattern of Notification, Ramp-up, Active, and Recovery. The Designated Interval of the Sequence is also referred to as the Active Interval.

This stereotypic pattern can be collapsed with the Intervals for Notification, Ramp-up, and Recovery expressed as Properties of the Active Interval. Notwithstanding this common pattern, the Active Period can contain any valid Sequence, as long as the meaning conveyed is understood by both parties.

A single Event may be broadcast to many VENs with similar performance characteristics. If the VENs all perform in unison, it can create spikes (or sudden drops) in energy use that can be harmful to the distribution system. It is necessary for a VEN to be able to ameliorate this issue by requesting response smoothing as described in Section 4.1.4.

A smoothing request is indicated through the WS-Calendar Tolerance Property. This property is applied to the overall Active Period so its meaning is the same whether the simplified common pattern or a full Sequence is conveyed.

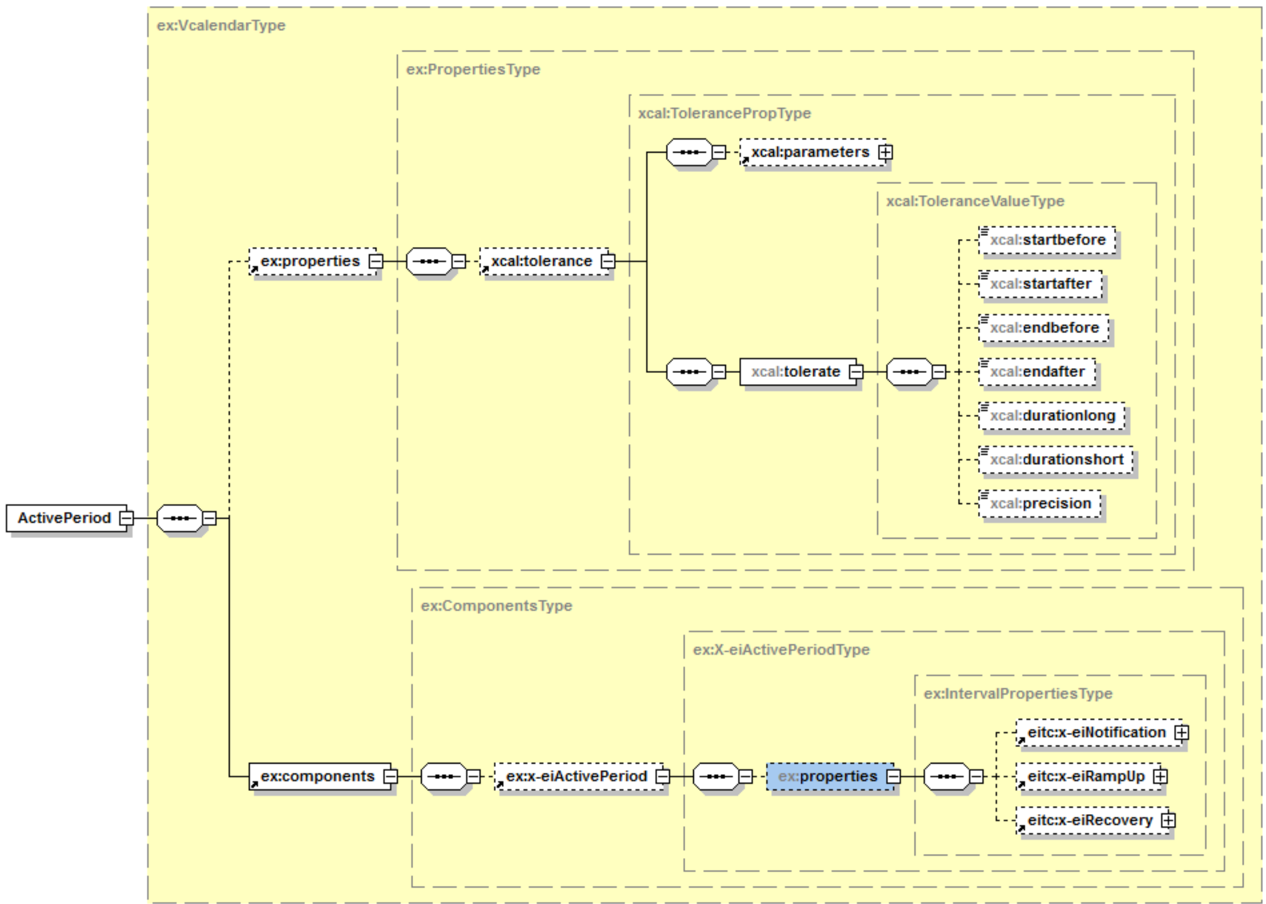


Figure 5-5: Active Period Elements

5.3.3 The Event Signals

Event Signals convey the detailed information about the schedule for an event. Signals are conveyed using Streams as described in Section 4.3. When an Event conveys multiple signals, they may be aimed at different target resources in different Market Contexts, or they may use different semantics, i.e., one use Price and another use Simple Level semantics. All Event Signals have a common form.

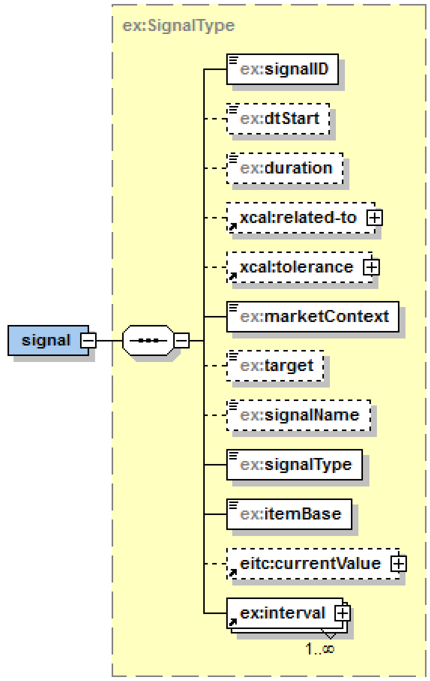


Figure 5-6: Event Signal Overview

As do all Streams, each Event Signal has a starting time, and a Tolerance (for smoothing); if absent, these are inherited from the Active Interval as if the Active Interval were a Gluon. The Time Zone is inherited from the Market Context. Each Event Signal includes a Related-To parameter to name the Designated Interval; if there is none, the first Interval is the Designated Interval. The Designated Interval has specific meaning for Sequence scheduling as defined in WS-Calendar.

5.3.3.1 Details of the Signal

Each signal includes a Market Context and optionally a Target. The Market Context and Target are used by the VEN to select which Signal, if any, to respond to. The Signal Name provides the VEN with a human-friendly description of the Signal, perhaps for display in a user interface. An EMIX Item Base enumerates what is being measured, and perhaps paid for, by the Signal. A Signal Type defines what Payload must be used throughout the signal; all Payloads in a signal MUST be of the same type. Each Interval contains a Payload, as specified by the Signal Type. An optional element, Current Value caches the current value (as of the signal creation) of the Payload.

Table 5-6: Signal Types

Signal Types	Description
Delta	The Payload in each Interval indicates a request to change the amount [used] by the amount in the signal as denominated by the Item Base.
Multiplier	The Payload in each Interval indicates a request to change the amount [used] to an amount computed by the amount in the signal times the Baseline as denominated by the Item Base.
Level	The Payload in each Interval indicates the Level during each Interval. See Section 5.2.1 for a description of Simple Levels.

Signal Types	Description
Price	The Payload in each Interval indicates indicate a price per unit as denominated by the Item Base. Price is conveyed as an EMIX Price, either a Price, a Price Multiplier, or a Price Relative. Each Payload in a Stream must contain the same type of Price. The Currency for each Price is inherited from the Market Context. In EMIX, both Price Multipliers and Prices Relatives include a Market Context; in a Payload in Signal, these are inherited from the Signal's Market Context.
Product	Signal indicates the Product for each interval. Payload Type is an EMIX Product Description.
Set-point	The Payload in each Interval indicates a requested amount [to use] as denominated by the Item Base. The amount may be more or less than the amount in the Baseline.

Parties may choose to exchange application specific payloads in signals as well. Prior to doing so, they MUST extend the Application Specific Signal Base and agree upon the Signal Type they will use. The Signal Type MUST conform to the EI Extension pattern. See Appendix C for a discussion of conforming extension.

5.3.4 Baselines

Baselines are streams that can incorporate signals and share many of the same elements. As some signals indicate the performance requested is relative to that in another interval, Baselines indicate the performance in that Interval.

The Baseline is a signal that expresses the amount point as denominated by the Item Base that is the starting for the signal types above. The computational basis for the Baseline is not in scope for this specification. The Baseline is compared to the actual metered consumption during the Event to determine the value of the Response. Depending on the type of product or service, Baseline calculations may be performed in real time or after the fact.

Another form of the Baseline merely indicates the comparable period that is used for comparison. This enables the sender to indicate when the Baseline is drawn from without indicating the values for that Baseline period, which may not yet be known.

5.3.5 Opt – Making Choices

When a VEN enrolls in an event-oriented Market Context, it makes itself Available to respond to events on a given schedule. The Availability schedule may be simple (all day, all the time) or complex (weekday afternoons, on weekends with a long notice, and not on Thursday mornings during biweekly payroll). No matter how simple or complex the Availability, the VEN may choose to change it for a limited period. This decision is communicated with an Opt (as in "Opt In" and "Opt Out").

The primary information payload for an Opt is a collection of Vavailability artifacts. An optional element inside each Availability artifact determines whether the particular repeating schedule within indicates availability or unavailability.

Business rules require that someone Opting declare their reason, using one of the specific enumerated reasons or an extension as allowed by the local Market.

Table 5-7: Opt

Opt Element	Description
Opt	Opts are used by the VEN to temporarily modify availability in 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.

Opt Element	Description
Opt ID	A reference ID for a particular Opt notification. This identifier may be used by other entities to refer to this instance of an Opt.
Opt	Opts are used by the VEN to temporarily modify availability in 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.
Opt Type	Either Opt-In or Opt-Out. This element determines the processing of the Vavailability. If Opt In, then any available time is added to the pre-existing schedule. If Opt-Out, then for the period bracketed by the Availability, the schedule replaces the pre-existing schedule.
Opt Reason	Reason for the Opt. Enumerated reasons include: Economic, Emergency, Must Run, Not Participating, Outage Run Status, Override Status, Participating

The Opt Type controls specific differences in how an Opt is processed against the pre-existing availability.

Opt-In: After processing, the new schedule and availability is added to the existing availability for the period bounded by the Opt Availabilities.

Opt-Out: After processing, the new schedule and availability replace the existing availability for the period bounded by the Opt Availabilities.

In either case, when the bounding period is over, Availability reverts to the previous schedule.

5.4 Monitoring, Reporting, and Projection

A Party may request that another Party measure something and report back. The thing measured may include Power, Voltage, Peak, or any other attribute associated with the products exchanged. These measurements may or may not be in relation to an Event. An EiReport is the record of a measurement or series of measurements made by one Party and delivered to another.

A Party requests that another Party prepare a Report by means of a Report Request. Report Requests can be delivered using the Report service, or can accompany an Event. The Historian and Projection services also make use of the Report Request.

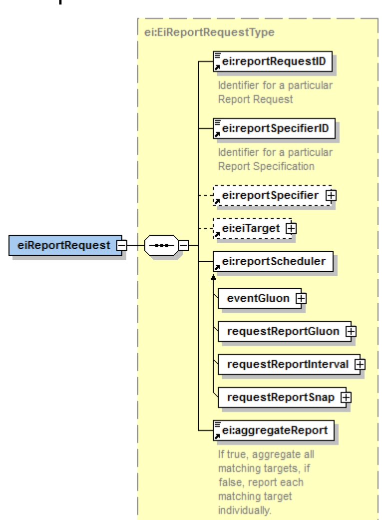


Figure 5-7: The Report Request

Table 5-8: Elements of the Report Request

Report Specifier	Description
Report Request ID	Identifies this request
Report Specifier ID	References the Report Specifier for this Request. The Specifier may be known from a previous request, or may be a standard Specifier within this Market Context.
Report Specifier	Request MAY optionally include the Report Specifier lest it is not otherwise known to the Party receiving the Request.
Target	Standard group of Parties, Resources, Groups, et al. that the Report concerns.
Report Scheduler	Indication of when the report is to be run, for how long, etc.
Aggregate Report	As the Target of a Report Request may indicate multiple Parties or Resources, this Boolean indicates whether a single report or one for each entity matching the Target is requested,

5.4.1 The Report Specifier

A Party species what reports it wants by means of a Report Specifier. Report Specifiers may be delivered in the Report Request are be known from the Market Context.

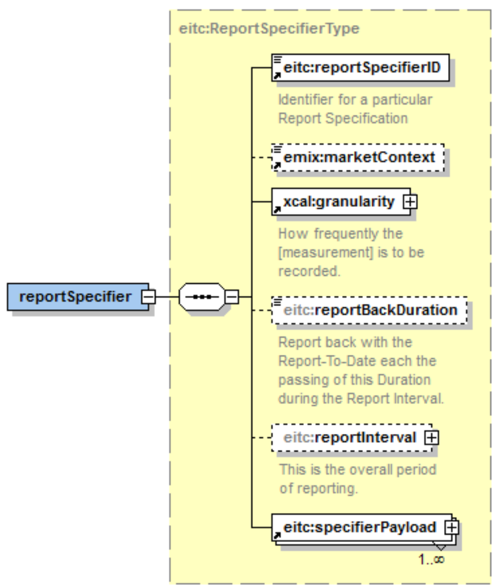


Figure 5-8: The Report Specifier

A single Report Specifier may generate quite different Reports based upon which service it is delivered by and how it is scheduled. The elements of a Report Specifier are as follows:

Table 5-9: Elements of the Report Specifier

Report Specifier	Description
Specifier ID	Identifies this Report Specifier
Market Context	The Optional Market Context MAY provide information about the Product that is being reported, or about where this Specifier came from.
Granularity	Duration defining temporal detail, i.e., “read the meter every 5 minutes”

Report Specifier	Description
Report-Back Duration	Report Back to requestor, with the report-to-date at each passing of this Duration during the Report Interval. If Optional, no Report-Back is expected.
Report Interval	Interval indicating the total span of the report. Parallel to Active Interval. May be influenced by a Gluon in the Report Scheduler. If the Interval contains a Start Date and no Duration, then the Report is to begin at the Start date and continue indefinitely.
Specifier Payload	The Specifier Payload indicates exactly what is to be in the report.

5.4.1.1 The Report Specifier Payload

The Specifier Payload indicates exactly what is in the Report. It consists of an **[EMIX]** ItemBase and a Report Type.

Table 5-10: Report Specifier Payload

Report Specifier	Description
rID	Identifies this Payload. If only one Payload is requested, the rID should be omitted; if multiple Payloads are requested in the same Report, each should have an rID.
Item Base	The Item Base is the core of an EMIX Product Description. Examples of an Item Base denominated value include Real Power, Real Energy, Voltage, et al.
Report Type	Defines what is being measured and reported. Measurements are in units of Item Base unless the Report Type indicates otherwise.

The Report Type specifies what is measured and, sometimes, how it is measured.

5.4.1.2 The Report Types

Report Types are an enumeration that indicates what how it is to be measured. These enumerations parallel the Signal Types used in Events.

Table 5-11: Report Types

Report Types	Description
Reading	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
Usage	Report indicates an amount of units (denominated in Item Base or in the EMIX Product) over a period. Payload Type is Quantity. A typical Item Base is Real Energy.
Demand	Report indicates an amount of units (denominated in Item Base or in the EMIX Product). Payload Type is Quantity. A typical Item Base is Real Power.
Set Point	Report indicates the amount (denominated in Item Base or in the EMIX Product) currently set. May be a confirmation/return of the set point control value sent from the VTN. Payload Type is Quantity. A typical ItemBase is Real Power.
Delta Usage	Change in Usage as compared to the Baseline

Report Types	Description
Delta Set point	Changes in Set point from previous schedule
Delta Demand	Change in Demand as compared to the Baseline
Baseline	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.
Deviation	Difference between some instruction and actual state.
Average Usage	Average usage over the duration indicated by the Granularity
Average Demand	Average usage over the duration indicated by the Granularity
Operating State	Generalized state of a resource such as on/off, occupancy of building, etc. No ItemBase is relevant. Requires an Application Specific Payload Extension.
Up Regulation Capacity Available	Up Regulation capacity available for dispatch, expressed in EMIX Real Power. Payload is always expressed as positive Quantity.
Down Regulation Capacity Available	Down Regulation capacity available for dispatch, expressed in EMIX Real Power. Payload is always expressed as positive Quantity.
Regulation Set point	Regulation set point as instructed as part of regulation services
Current Storage	Item Base is expressed as Real Energy and Payload is expressed as a Quantity.
Target Storage	Item Base is expressed as Real Energy and Payload is expressed as a Quantity.
Available Storage Capacity	Capacity available for further energy storage, presumably to get to Target Storage.
Price	Report Prices per ItemBase at each interval
Level	Report Simple Level at each interval. ItemBase is not meaningful.

Report Type is implemented as an enumerated string with extensibility. Parties wishing to extend the enumeration MUST defined the report payload requirements.

5.4.2 Report Scheduler

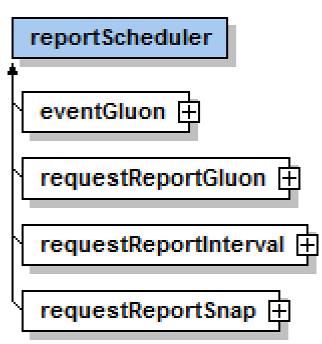


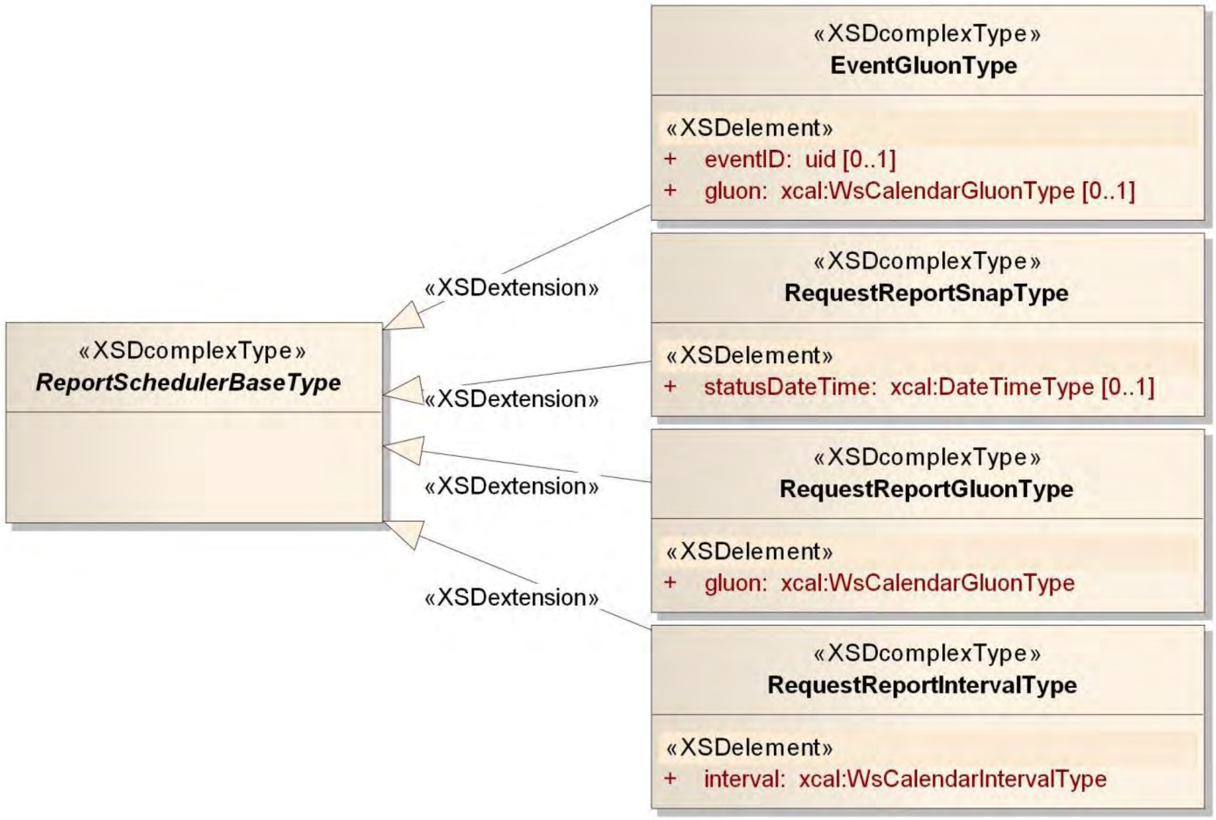
Figure 5-9: Report Scheduler

The report scheduler is an abstract type that specifies how often and for how long a report will be prepared. The Report Scheduler adds flexibility and consistency by enabling a single Report Specifier to be used in multiple scenarios. One option for Report Scheduler enables a Report Request to be associated with an Event.

Table 5-12: Types of Report Scheduler

Report Scheduler	Description
Event Gluon	Associates a Report Request with a particular event. This type consists of a Gluon and a reference to the Event ID. The Gluon sets the Report Interval relative to the Active Interval of the Event. For example: **SS –T20M.** The Report interval starts 20 minutes before (-T20M) the Active Interval starts (Start to Start). **FF T1H**. The Report interval Finishes 1 hour after (T1H) the Active Interval Finishes (Finish to Finish). If absent, the Report Interval is the same as the Active Interval, i.e., the Report runs during Active Interval. The Event ID indicates the Event this report is related to. If absent, the Report Request must be delivered as part of a an EiEvent
Request Report Gluon	Used if the Report Specifier includes a Report Interval to influence the expression of that Interval. Information in the Gluon is inherited by the Report Interval in conformance with WS-Calendar.
Request Report Interval	The Interval in Scheduler is the Report Interval for the Report. If the Specifier included an Interval, it is replaced by the one in the Schedule.
Request Report Snap	Indicates that the readings indicated by the Specifier are to be made once at the Status Date and Time and then returned to the Requester. If the Status Date and Time are omitted, then the Snap is to be made at the time of receipt.

1182 ### 5.4.2.1 UML Diagram of Report Scheduler



1183

1184 *Figure 5-10: UML Diagram of Report Scheduler*

5.4.3 UML Diagram of Report Request

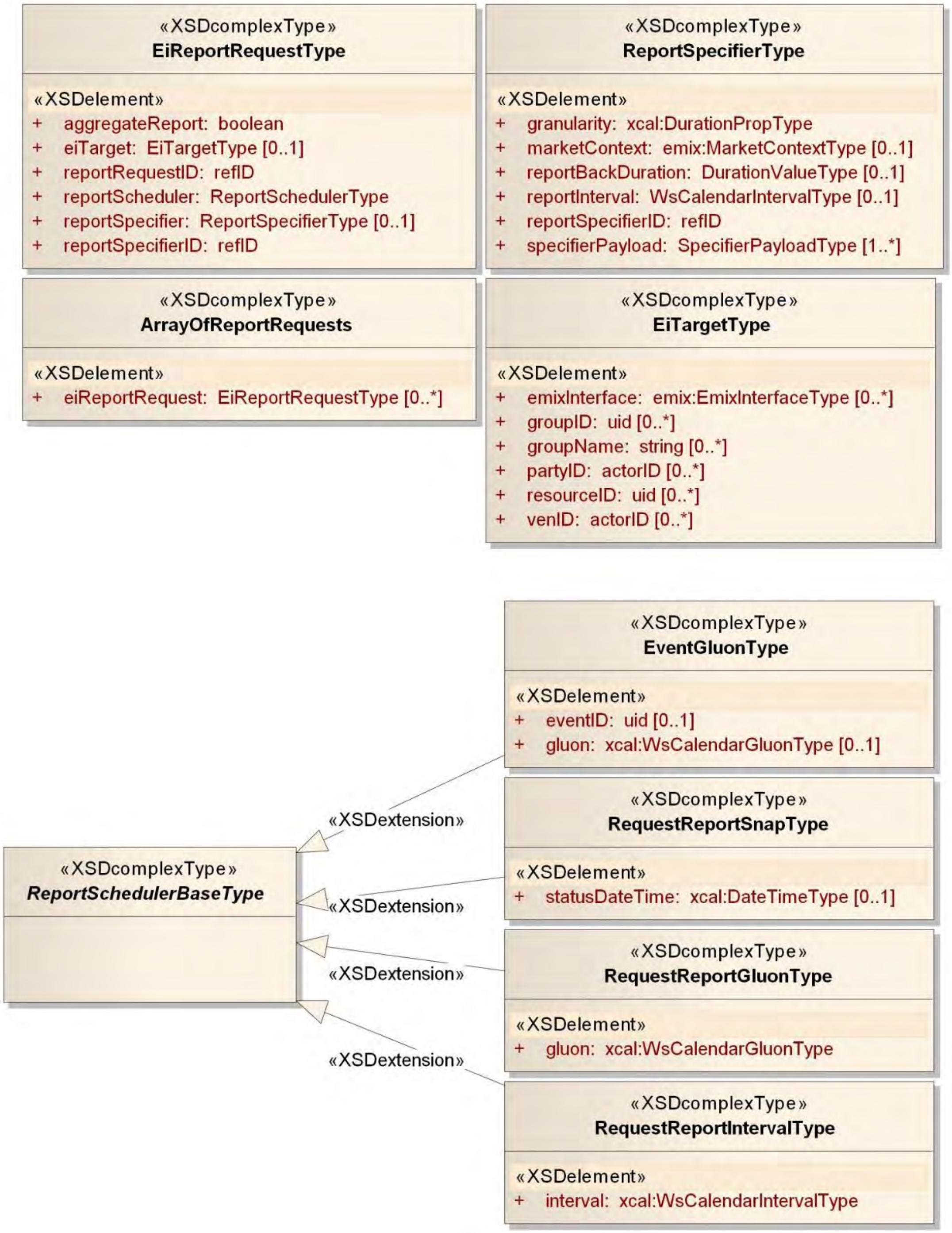


Figure 5-11: UML Class Diagram of Report Request

5.5 Reports, Snaps, and Projections

Reports are simple Streams with some metadata identifying the report and a collection of Intervals containing the Payloads for each [measurement]. Reports can be of the past, the present, or the future. A Report appears as a series of [measurements] in the past. A Snap is a Report made as of a single moment. A Projection is in the same form as a report, but it includes projections of what will be in the future, including a confidence level in the payload.

Table 5-13: Reports

Report Metadata	Description
Report ID	Unique identifier for this Report. The Report ID persists over multiple Report-Backs.
Report Request ID	Identifies the Request that resulted in this Report.
Report Specifier ID	Identifies the Report Specifier that resulted in this Report.

The above information is sufficient to uniquely identify each Report, why it was made, and to what specifications. The full form of a report is as follows in Figure 5-12.

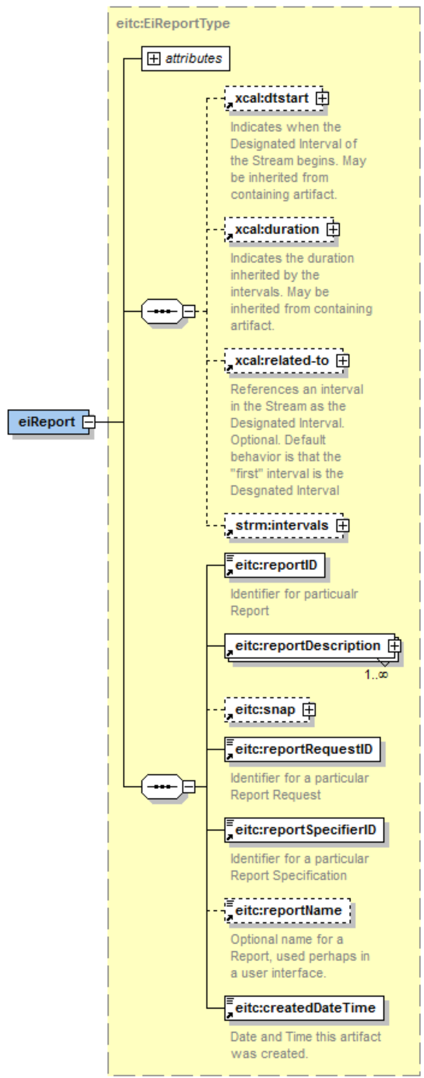


Figure 5-12: The Report

5.5.1 Elements of the Report

Table 5-14: Elements of Reports

Report Elements	Description
Start Date and Time	Indicates the beginning of the Report
Duration	Indicates the Duration of each Interval in the Report
Related To	Inherited from Stream Base but not used in Reports. Must be Ignored.
Report Name	Optional human-friendly name for the report
Report Description	Type describing the make-up of the report which MAY not be entirely determinable from the Specifier. Also, explains the interpretation of each Value.
Created Date and Time	Indicates when the Report was prepared for delivery to the requestor.

5.5.2 Report Description

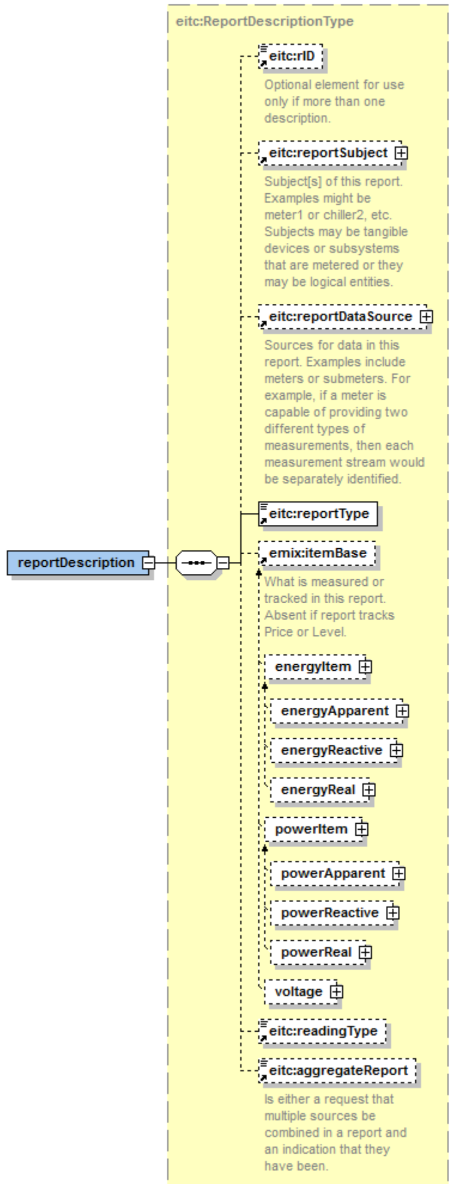


Figure 5-13: The Report Description

The Report Description indicates what is in the Report, which may be different from what was specified, particularly if multiple elements were in the Target. A Report may include multiple Report Descriptions if multiple payloads are delivered in each interval. Conversely, if the Recipient is able to rely completely on the Report Specifier, the Report Description MAY be omitted.

The Elements of the Report Description are as follows:

Table 5-15: Elements of the Report Description

Report Elements	Description
rID	Optional report identifier required only if multiple payloads are delivered in each Interval.

Report Elements	Description
Subject	Identifies the specific thing or things being measured in this report. Subject is in the form of a Target, which means is can include one or more Parties, Resources, Assets, Groups, etc.
Data Source	Identifies the Source of the information or measurement provided. A common use is to identify the MRIDs of the meter[s] that apply to the Subject. Data Source is in the form of a Target.
Report Type	Identifies what is the meaning of each measurement, as defined in Section 5.4.1.2.
Item Base	Identifies the Units being measured, unless the Report Type indicates this element is meaningless.
Reading Type	If present, indicates metadata about the Readings, i.e., direct measurement or computation. Conforming profiles MAY ignore Reading Type.
Aggregate Report	Identifies whether each payload represents an individual subject, or the sum of multiple subjects.

5.5.3 Report Payloads

The details in each Interval in a Report bear a lot of similarity to those in the Signals. In many cases, a Signal requests that a system provide something similar to its Signal Value. Reporting back in the same format enables ready comparisons. These values are conveyed in the Payload.

Signals, though, are ideal. Reports describe real world effects, and therefore messy. For this reason, Report Payloads include some additional information.

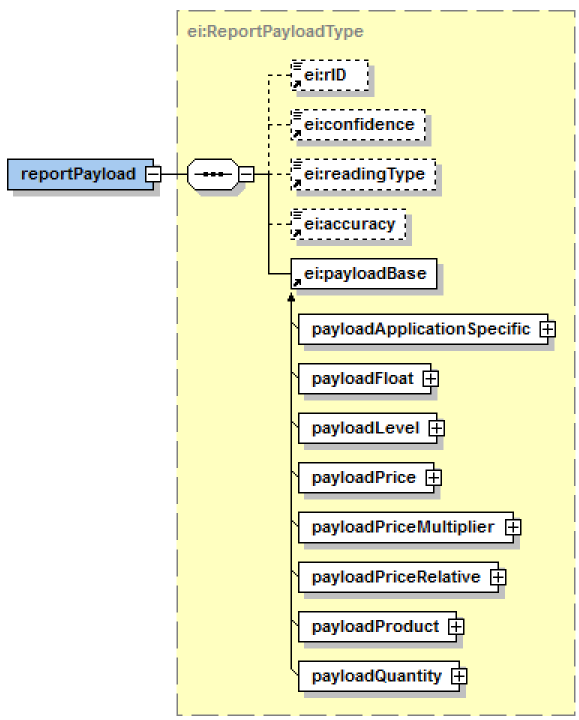


Figure 5-14: the Report Payload

Figure 5-14 shows the information qualifications alongside the Payload. If an Application within a VEN has specific reporting requirements, a new Payload Type can be derived from the abstract Payload Application Specific type; a type so derived can be delivered by a conforming report service.

Table 5-16: Report Payload Qualifiers

Report Metadata	Description
Confidence	An optional information structure that indicates in each interval how likely the information is to be precise.
Reading Type	An enumerated indication different ways to derive a reading
Accuracy	An indicator of Payload accuracy

5.5.3.1

The Reading Type describes the information returned in a report. Specifically, the Reading Type describes how the number in the payload was arrived at. The Reading Type MAY be in the stream Gluon, and be inherited by each Interval in the Sequence (or by the Snap, if present). The Reading Type MAY also appear in any Interval where the reporting system is indicating that one payload differs from others in the Sequence. Reading Types are described in Table 5-17.

Table 5-17: Reading Types

Reading Type	Description
Direct Read	Reading is read from a device that increases monotonically, and usage must be computed from pairs of start and stop readings.
Net	Meter or [resource] prepares its own calculation of total use over time
Allocated	Meter covers several [resources] and usage is inferred through some sort of pro rata computation.
Estimated	Used when a reading is absent in a series in which most readings are present.
Summed	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.
Derived	sage is inferred through knowledge of run-time, normal operation, etc.
Mean	Reading is the mean value over the period indicated in Granularity
Peak	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.
Hybrid	If aggregated, refers to different reading types in the aggregate number.
Contract	Indicates reading is pro forma, i.e., is reported at agreed upon rates
Projected	Indicates reading is in the future, and has not yet been measured.

5.5.3.2 Contrasting semantics of Summary and Aggregate in Reports

Consider the following industrial facility with a single ESI acting as a VEN. This facility chose to offer four Resources to its VTN: one industrial Resource and three office Resources, one for each floor. Two of the office Resources, Floor 2 and Floor 3, have their own zones and meters. Floor 1 has two zones, 1A and 1B, that are metered separately. The three office Resources are all in a single Group, Office. The single industrial Resource is in its own Group, Factory.

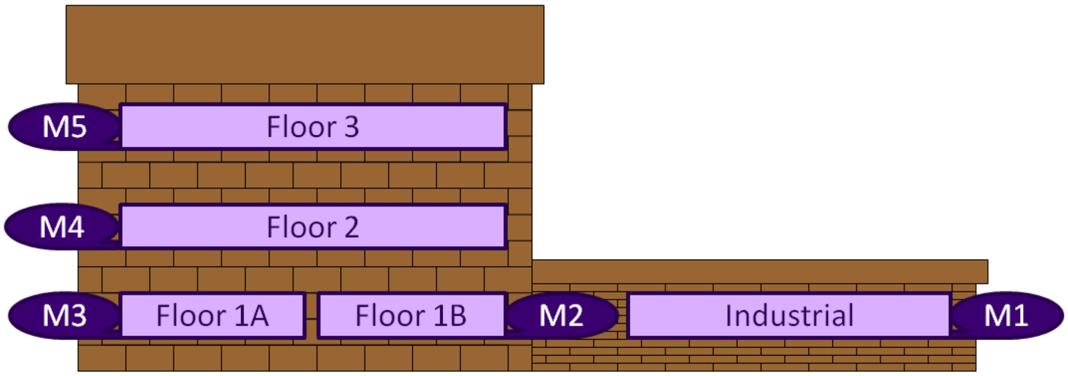


Figure 5-15: Illustrating Aggregate vs. Summary

A Usage report with a Target of Office applies to three Resources, Floor 1, Floor 2, and Floor 3. If the Aggregate flag is True, the VEN prepares a single report that aggregates the information from all three Resources. If a report Target indicates Industrial or Factory, Group or Resource, there is no distinction between an Aggregate or non-Aggregate request.

The Data Sources for the Usage Reports are the Meters, M1-M5. The Report for Floor 3 has a Data Source of M5. The Report for Floor 2 has a Data Source of M4. The Report for Floor 1 has two data sources, M2 and M3, and the single Reading for Floor 1 is of the Type “Summary”

Aggregate refers to the combining of multiple Subjects (things named in Target) into a single report; Summary refers to the combination of multiple Data Sources [meters] into a single value.

5.5.4 UML Diagram of Report

Figure 5-16: UML Class Diagram of Reports

5.6 Reponses and Error Reporting

All Services share a common Response. The Response shares a common extensible code, a readable description, and a reference to the Message that this is in response to.

Table 5-18: Responses

Response Elements	Description

Response Elements	Description
EI Response	Response is the generic model for responding to any Servicer Request
Response Code	Code consisting of 3 digits for automated processing. The simplest devices need understand only the first digit, others are for extension as needed within the higher order error indicated by the first digit. • 1xx: Informational - Request received, continuing process • 2xx: Success - The Request 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. Where possible, the HTTP errors should be used.
Response Description	Optional String describing the response or the reason for the response
Message UID	Reference to the Message that elicited this response
Response Terms Violated	Optional Array of EMIX Terms and Response Descriptions to provide a machine interpretable Response. For example, if the Request fails because it violated the "Minimum Notification Duration" of one hour, the responder could send back the Term (with value) and an Response Description

5.6.1 Event Responses

Responses to events are not stateless, so they require further information. All Responses regarding Events have the elements in Table 5-19 in addition to the elements listed in Table 5-18.

Table 5-19: Event Response

Event Responses	Description
Event ID	ID of the Event which caused this Response
Modification Number	Modification Number of the Message about an Event that caused this Response
Opt Type	Indicates whether this Response results in a VEN Opting In or Opting Out of the Event.

Some services communicate multiple messages, and the different messages may warrant different responses. In these cases, there is a single EiResponse (or EiEventResponse) which reports on the conveys an overall response. If this overall response is Success (2xx), then there is no need for the recipient to examine the message further. If the overall Response is anything other than success, then the response for each Element in the original Request can be found by examining the array of responses (type responses) or the array of Event Responses (type eventResponses) for detailed information.

5.6.2 References in Responses

Response is a general Type that must reference any number of messages, reports, requests, etc. These critical cross interaction types are each identified by a Reference ID. The Reference ID for each is derived from a common refID type that enables type-safe substitution in Response and in other payloads.

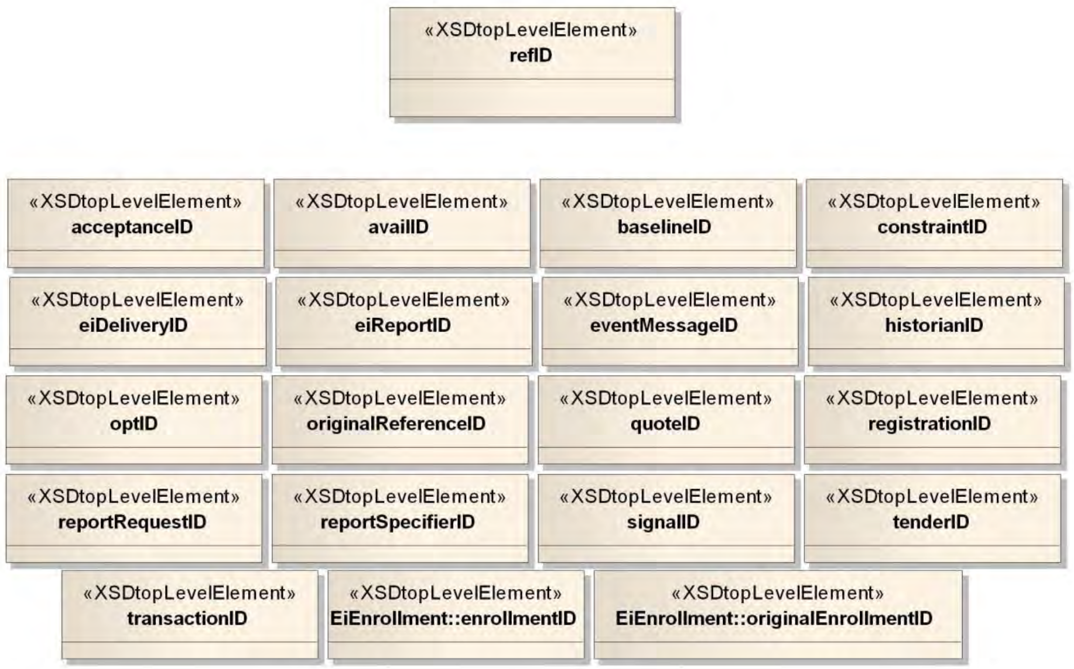


Figure 5-17: UML Diagram showing refID and its derived types

5.7 Availability Behavior

In different Market Contexts, Availability is interpreted differently by the VTN. This availability behavior is published as part of the EI Market Context as it is in effect a meta-term for the market.

Table 5-20: Availability Behavior

Availability Behavior	Description
Behavior	When an Event is issued by the VTN, it is validated against the parameters and constraints that were established when the Market Context was set up, i.e., the market Rules support Events between 12:00 and 16:00. If the Event is not within 12:00 and 16:00 then VEN must take some action to resolve the conflict.
Accept	Simply accept the issued DR event regardless of any conflicts
Reject	Reject any DR events that conflict with configured Availability
Restrict	Modify the DR event parameters so that they legally fall within the bounds of the configured parameters.

6 Introduction to Services and Operations

In the following sections services and operations consistent with **[SOA-RM]** are described. For each service operation there is an actor that *invokes* the service operation and one that *provides* the service. These roles are indicated by the table headings *Service Consumer* for the actor or role that consumes or invokes the service operation named in the *Operation* column and *Service Provider* for the actor or role that provides or implements the service operation as named in the *Operation* column.

This terminology is used through all service definitions presented in this specification.

The column labeled *Response Operation* lists the name of the service operation invoked as a response. Most operations have a response, excepting primarily those operations that broadcast messages. The roles of *Service Consumer* and *Service Provider* are reversed for the *Response Operation*.

All communication between customer devices and energy service providers is through the ESI.

For transactive services any party may receive tenders (priced offers) of service and possibly make tenders (priced offers) of service.

Any party using Transactive Energy services may own generation or distributed generation or reduce or increase energy from previously transacted energy amounts. These activities are not identified in transactive services. The dispatch of these resources and the use of energy by a party are influenced by tenders between Parties that may result in new Transactions and changes in operations.

The VEN/VTN services provide a characterization of the aggregate resources of a VEN that may be communicated to the VTN; that relationship depends also on the EiMarketContext in which the interactions take place.

The next section describes the role of Resources, Curtailment and Generation. In a transactive approach tendering and prices are used by parties to discover and negotiate transactions that respect the preferences of each party and energy usage, generation, storage and controllability directly available to each party. There is no formal communication of resource characteristics in the transactive approach.

6.1 Resources, Curtailment, and Generation

If the VEN participates in a demand response program or provides distributed energy resources, its ESI is the interface to at least one dispatchable resource (Resource), that is, to a single logical entity. A Resource may or may not expose any fine structure.[6] The Resource terminology and the duality of generation and curtailment are from **[EMIX]**.

Under a demand response program, a Resource is capable of shedding load in response to Demand Response Events, Electricity Price Signals or other system events (e.g. detection of under-frequency). The VTN can query the actual state of a Resource with the EiReport service and request ongoing information. The VEN can query the status of the VTN-VEN relationship using the EiRequestEvent operation.

Alternatively, a Resource may provide generation in response to similar information. The net effect is the same.

[6] A finer level of granularity is sometimes called an *asset*. Assets are not in scope for this specification.

6.2 Structure of Energy Interoperation Services and Operations

Energy Interoperation defines a web services implementation to formally describe the services and interactions although fully compliant services and operations may be implemented using other technologies.

The services presented in this specification are divided into five broad categories:

- Transactive Services—for implementing energy transactions, registration, and tenders
- Event Services—for implementing events and linked Reports
- Report Services—for exchanging remote sensing and feedback.
- Enrollment Services—for identifying and qualifying service providers, resources, and more
- Support Services—for additional capabilities

The structure of each section is a table with the service name, operations, service provider and consumer, and notes in columns.

The services are grouped so that profiles can be defined for purposes such as price distribution, and Demand Response (with the functionality of **[OpenADR]**). This specification defines three profiles, the OpenADR Profile, the TeMIX (Transactive EMIX) Profile, and the Price Distribution Profile.

The normative XML schemas are in separate files, accessible through the **[namespace]** on the cover page.

6.3 Naming of Services and Operations

The naming of services and operations follows a pattern. Services are named starting with the letters *Ei* capitalization which follows the Upper Camel Case convention. Operations in each service use one or more of the following patterns. The first listed is a fragment of the name of the initial service operation; the second is a fragment of the name of the response message which acknowledges receipt, describes errors, and may pass information back to the invoker of the first operation.

Create—Created	An object is created and sent to the other Party
Cancel—Canceled	A previously created request is canceled
Request—Reply	A request is made for all objects of the specified type previously created and relevant to this VTN-VEN relationship
Distribute	An object (such as a price quote, a curtailment or generation request) is created and sent without expectation of response.

For example, to construct an operation name for the EiEvent service, "Ei" is concatenated the name fragment (verb) as listed. For example, an operation to cancel an outstanding operation or event is called *EiCancelEvent*.

The pattern of naming is consistent with current work in the IEC Technical Committee 57 groups responsible for the **[TC57CIM]**.

6.4 Push and Pull Patterns

The Service Operation naming includes application-level acknowledgements, which in nearly every case carry application-level information, and allow for both push and pull of messages. This description applies to both transactive and VTN/VEN interactions as both are performed by Parties taking on various roles.

Push and Pull are with respect to the invoker of the operation. So if a Party produces information that describes a price quote, it can invoke (in the case of Push) an operation to send it to one or more other Parties. In the alternative, each Party (in the case of Pull) can invoke a request for information by polling, or pulling it, another Party respect to a particular relationship or Market Context.

The Pull operation is performed by the Party invoking the Request service operation pattern and fulfilled with a Reply service operation pattern invoked by the receiving Party.

So a series of Push operations from one Party to a counter-Party is analogous to a series of Pull operations from the counter-Party to the Party.

In the VTN-VEN context, a series of Push operations from a VTN to its VENs is analogous to a series of Pull operations from the VEN to its VTN; by examining (e.g.) the absence of an Event that was visible on a previous Pull the VEN can infer that that Event was canceled. The VEN could then send a Canceled service operation as if it had received a Cancel service operation.

One special case is the *Distribute* pattern, which expects no response to the invoker.

The service quality of the Pull operations (and in particular the load on the VTN from repeated polling) is not in scope for this specification.

6.5 WSDL Integration

A WSDL represents a contract between two systems that are being integrated. As such additional attributes may need to be passed in addition to the attributes that are specific to a message payload (representing the core set off information being passed). At a high level, any given integration may need to include a header, request, and/or reply in addition to the message payload as shown in the figure below.

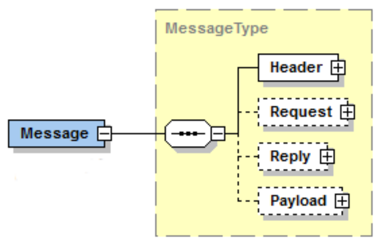


Figure 6-1: Generalized view of the high-level message structure

For example, for WSDL-based integration in which information regarding a demand response event, details regarding the specifics of the event are contained in the message payload. However, additional details that work to ensure the successful integration may be included in the header, request, or reply.

A message header contains information about the sender and receiver of the message or other information used to correlate the service request, to guarantee delivery, or to support non-repudiation as seen in the [non-normative] figure below.

Message headers are out of scope for this specification.

6.6 Description of the Services and Operations

Each service is described as follows. In the sections that follow, we will:

- Describe the service
- Show the table of operations
- Show the interaction patterns for the service operations in graphic form
- Describe the information model using **[UML]** for key artifacts used by the service
- Describe the operation payloads using **[UML]** for each operation

6.7 Responses

In a service interaction responses may need to be track to determine if the transaction is successful or not. This may be complicated by the fact that any given transaction may involve the transmission of one or more information objects.

The class diagram below reflects the generic response.

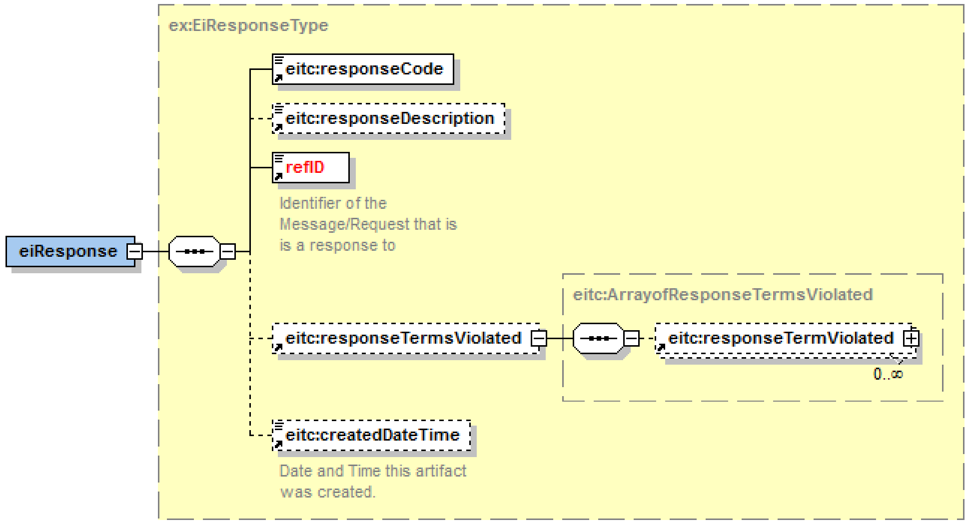


Figure 6-2: Example of generic error response for a service operation

The Reference ID (refID) identifies the artifact or message element that this response is to. The response code indicates success or failure of the operation requested. The Response Description is unconstrained text, perhaps for use in a user interface.

There is no exhaustive list of all possible Response Codes. The Response Codes are intended to enable even the smallest device to interpret Response. This specification uses a pattern consisting of a 3 digit code, with the most significant digit sufficient to interpret success or failure. This pattern is intended to support that smallest device, while still supporting more nuanced messages that may be developed.

- 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

While the only value of xx that is defined as of this version is 00, conforming specifications may extend these errors to defining more fine grained errors. These errors should extend the pattern above, though. A response code such as 403 should always be within the realm of Requester Error.

6.7.1 Terms Violated

Terms Violated is an optional element of a Response. Terms communicate business expectations. It may be that a Service Request fails not because it is improperly formed, but because it violates one or more of these business rules. For example, a Market Term may indicate a 20 minute notification duration. A Service Request that asks for a performance with only a 5 minute notification violates that Term. By passing that Term back in the Response, that service provider can make known what its requirements are.

It is outside the scope of this specification whether a provider MAY present terms while still accepting a Service.

6.7.2 Response Derivations

Because some responses provide require additional context relative to the Service requested.

6.7.2.1 Event Responses

Event Responses are derived from the Response Type and add elements useful for Event-based interactions. Event Responses include Event ID and Modification Number to indicate exactly which Event they are responding do. Event Responses also include the Opt Type (Opt In or Opt Out) to describe what response is being made to an event.

6.7.2.2 Enrollment Responses

Enrollment Responses are derived from the Response Type and add elements useful for Event-based interactions. The Enrollment response includes an Enrollment ID to indicate which Enrollment is being referenced.

Enrollment establishes a business relationship between a Party and a particular Market Context. A Party may be enrolled in several Market Contexts. Enrollment Responses include the Market Context that is affected by the Response.

A single request to Enroll may create many Enrollment IDs. For example, a Party offering several Resources may get an Enrollment ID for each. Similarly, a single Resource may become enrolled in both a power and a regulation Market Context. An Enrollment Response includes a Market Context to indicate which Market Context was affected.

As stated above, a single request to Enroll may create many Enrollment IDs. It can be helpful to know the original request's reference ID to understand the Response. An Enrollment Response MAY include an Original Reference ID.

6.7.3 Compound Responses

Many service interactions may affect a number of messages. For examples, a single service interaction may include multiple Tenders, or Events. A single Enrollment request may result in multiple Enrollments. All such Responses have the pattern of a single Response (or Event Response, or Enrollment Response) accompanied by a collection of Responses. This specification defines the collections Responses, Event Responses, and Enrollment Responses.

The end-point receiving a compound Service Payload, including both Responses and Responses can follow the following rules.

- If the Response indicates success, there is no need to examine each element in the Responses.
- If some elements fail and other succeed, the Response will indicate the error, and the recipient should evaluate each element in the Responses to discover which components of the operation failed.

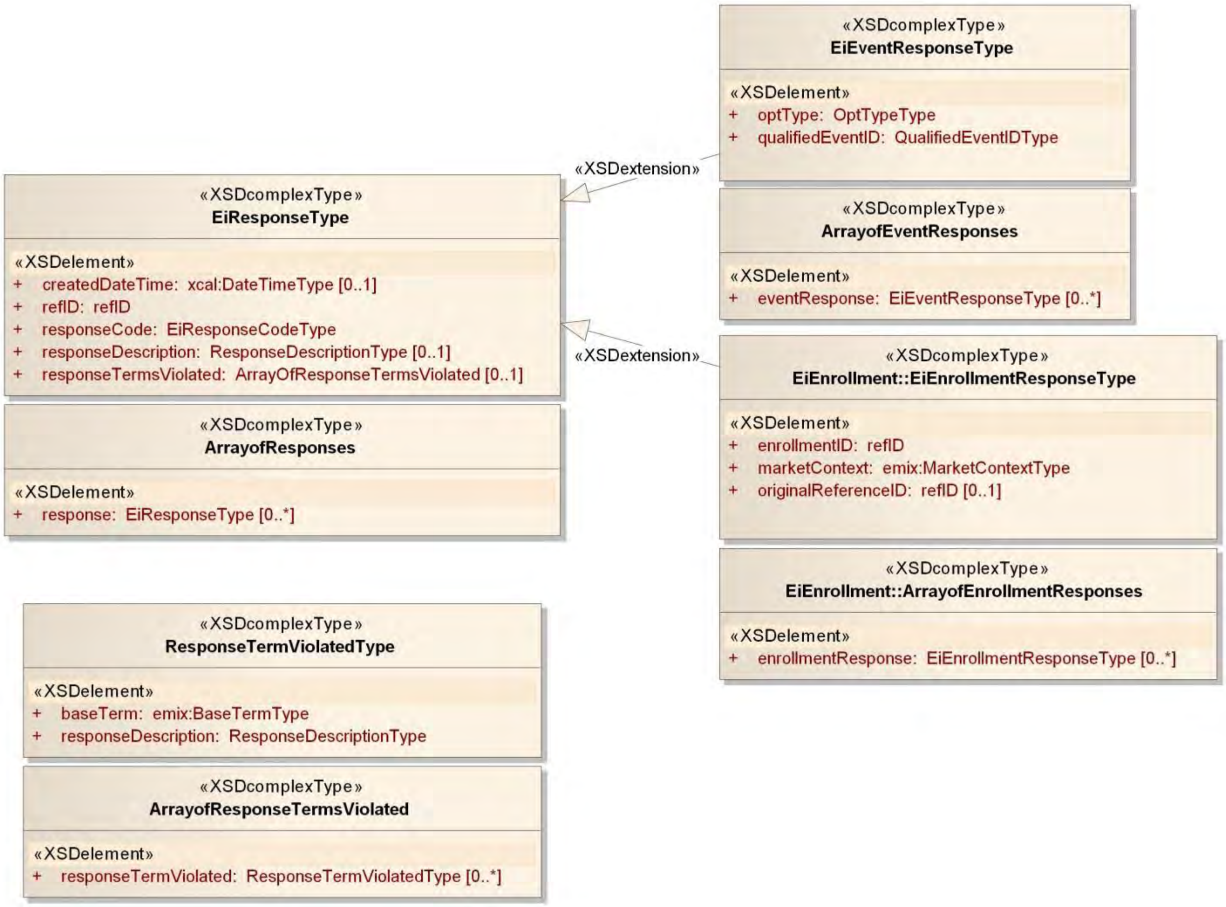


Figure 6-3: UML for Response

6.7.3.1 Summary of Response and Responses

An Response returns the success or failure of the entire operation. The responses returns an ID. and a response for each.

It is MANDATORY to return errors in responses. It is OPTIONAL to return successes in responses. For Cancel, in particular, it is not mandatory to return any responses if the entire operation was completed successfully. The pattern is to return those that have failed (required) and those that succeeded (optional).

6.7.4 Requests

Each of the Services includes a Request, which is essentially a status update. For Service Foo, tell me all the Foos that we have outstanding. The meaning of Outstanding varies from Service to Service. In general, either party may make invoke the Request operations on the other. Tell me all the Quotes you have given me is the mirror of Tell me all the Quotes you have received from Me; both Requests share the same semantics.

Each optional element in a Request refines or narrows the scope of the Request by requesting only those Foos for which the named elements match. If there are more than one instances of the same named element, then this restriction element is treated as if a logical OR were applied, i.e., where element = A OR element = B. Where more than one type of element is named, than the restriction is treated as an AND, i.e., element A = “foo” AND element B = “fie”.

A special element that is included in most Requests is the Interval. The Interval is treated as a temporal restriction. For example, an Interval that encompasses a business day can request all Foo for delivery on that day. Intervals MAY be open-ended. An Interval conveying only a Start Date matches all Foo that is current from that date and time forward. An Interval conveying only an End Date matches all Foo that is current that date and time/ If there is any ambiguity about what “matches” means, it is defined within the Service section below, c.f., the definition of pending Events in Section 9.2 *“Special Semantics of the Event Request Operations”*.

7 Transactive Services

Transactive Services define and support the lifecycle of transactions inside an overarching agreement, from initial quotations and indications of interest to final settlement. The phases are

- Registration—to enable further phases.
- Pre-Transaction —non-binding quotes and binding tenders for transactions.
- Transaction Services—execution and management of transactions including transaction with optionality.
- Post-Transaction—settlement, energy used or demanded, payment, position.

For transactive services, the roles are **Parties** and **Counterparties**. For event and resource services, the Parties adopt a VTN or VEN role for interactions. The terminology of this section is that of business agreements: tenders, quotes, and transaction execution and (possibly delayed) performance under an option or DR transaction.

The register services identify the parties for future interactions. This is not the same as (e.g.) a program registration in a demand response context—here, registration can lead to exchange of tenders and quotes, which in turn may lead to a transaction which will determine the VTN and VEN roles of the respective parties.

7.1 EiRegisterParty Service

The EiRegisterParty service operations create a registration for potential Parties in interactions. This is necessary in advance of an actor interacting with other parties in various roles such as VEN, VTN, tenderer, and so forth

Table 7-1: Register Services

Service	*Operation*	*Response*	*Service Consumer*	*Service Provider*	*Notes*
EiRegisterParty	EiCreateParty Registration	EiCreatedParty Registration	Party	Party	Create and send a Party Registration request
EiRegisterParty	EiRequestParty Registration	EiReplyParty Registration	Party	Party	Request semantics with optional Interval
EiRegisterParty	EiCancelParty Registration	EiCanceledPart yRegistration	Party	Party	Cancel one or more Party Registrations

7.1.1 Interaction Pattern for the EiRegisterParty Service

This is the **[UML]** interaction diagram for the EiRegisterParty Service

Figure 7-1: Interaction Diagram for EiRegisterParty Service

7.1.2 Information Model for the EiRegisterParty Service

The details of a Party are outside the scope of this specification. The application implementation needs to identify additional information beyond that in the class EiParty.

Figure 7-2: EiParty UML Class Diagram

7.1.3 Operation Payloads for the EiRegisterParty Service

The **[UML]** class diagram describes the payloads for the EiRegisterParty service operations.

Figure 7-3: UML Class Diagram for EiRegisterParty Service Operation Payloads

7.2 Pre-Transaction Services

Pre-transaction services are those between parties that may or may not prepare for a transaction. The services are EiTender and EiQuote. A quotation is not a tender, but rather a market price or possible price, which needs a tender and acceptance to reach a transaction.

Price distribution, which is sometimes referred to as *price signals,* is accomplished using the EiQuote and EiTender services. Quotes are indications of a possible tender price; they are not actionable. A Tender offers prices at which Transactions may be made; they are actionable.

As with other services, a Party MAY inquire from a counterparty what offers the counterparty acknowledges as open by invoking the EiSendTender service to receive the outstanding tenders.

There is no operation to “delete” a quote; when a quote has been canceled the counterparty MAY delete it at any time. To protect against recycled or dangling references, the counterparty SHOULD invalidate any identifier it maintains for the cancelled quote.

Tenders, quotes, and transactions are [EMIX] artifacts, which contain terms such as schedules and prices in varying degrees of specificity or concreteness.

Table 7-2: Pre-Transaction Tender Services

Service	Operation	Response	Service Consumer	Service Provider	Notes
EiTender	EiCreateTender	EiCreatedTender	Party	Party	Create and send Tender
EiTender	EiRequestTender	EiReplyTender	Party	Party	Request outstanding Tenders; request semantics with optional time Interval
EiTender	EiCancelTender	EiCanceledTender	Party	Party	Cancel one or more Tenders
EiTender	EiDistributeTender	—	Party	Party	For broadcast or distribution of Tenders

Table 7-3: Pre-Transaction Quote Services

Service	Operation	Response	Service Consumer	Service Provider	Notes
EiQuote	EiCreateQuote	EiCreatedQuote	Party	Party	Create and send a quote
EiQuote	EiRequestQuote	EiReplyQuote	Party	Party	Request outstanding Tenders; request semantics with optional time Interval
EiQuote	EiCancelQuote	EiCanceledQuote	Party	Party	Cancel one or more quotes

Service	*Operation*	*Response*	*Service Consumer*	*Service Provider*	*Notes*
EiQuote	EiDistributeQuote	--	Party	EiTarget	For broadcast or distribution of quotes

7.2.1 Interaction Pattern for the EiTender and EiQuote Services

This is the **[UML]** interaction diagram for the EiTender Service.

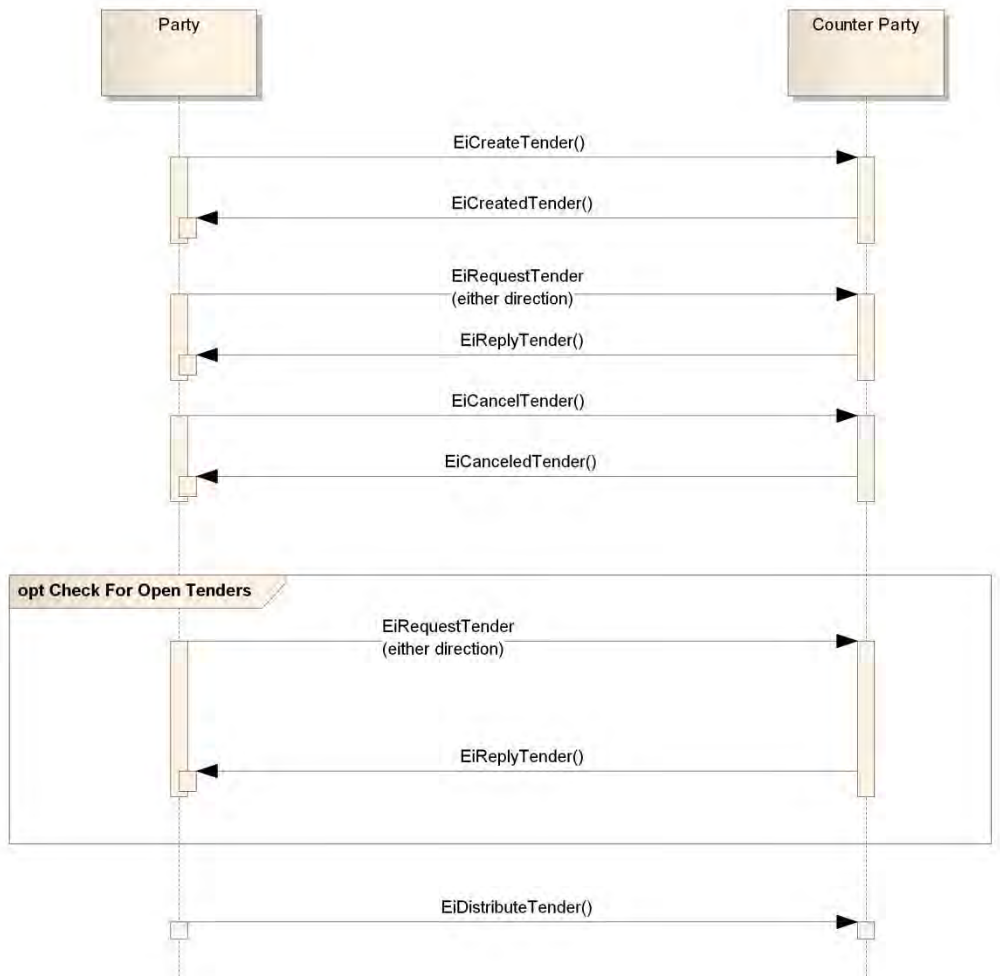


Figure 7-4: Interaction Diagram for the EiTender Service

This is the **[UML]** interaction diagram for the EiQuote Service

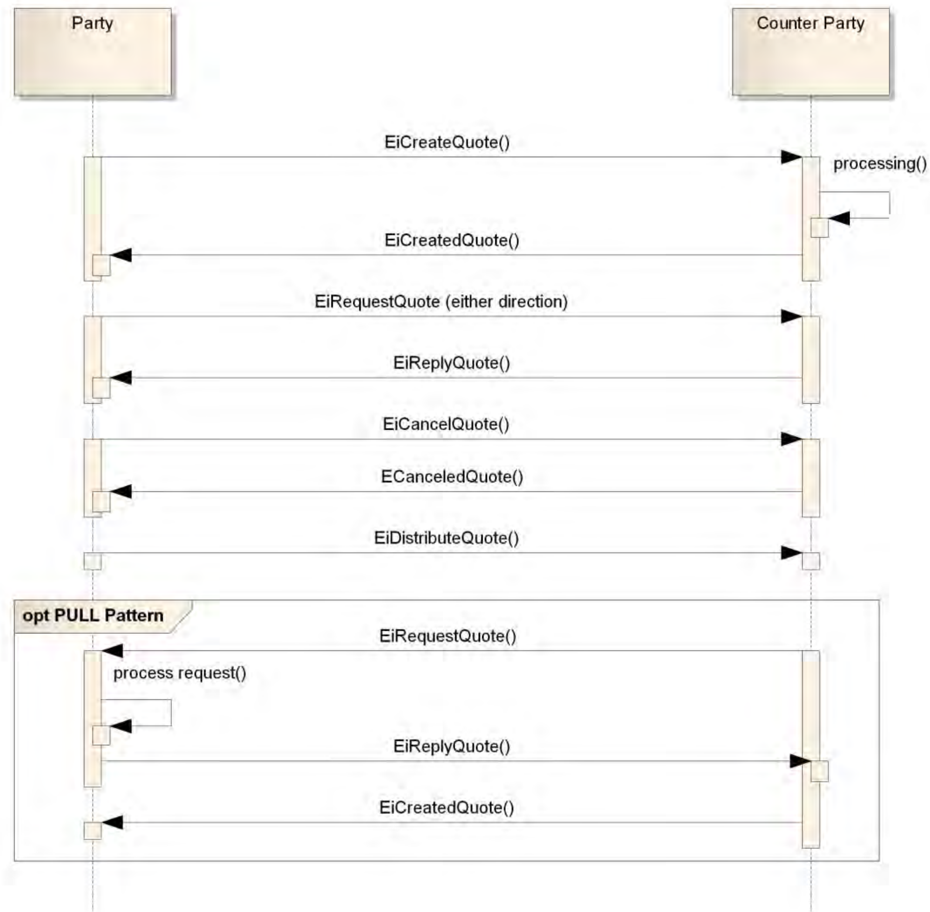


Figure 7-5: Interaction Diagram for the EiQuote Service

7.2.2 Information Model for the EiTender and EiQuote Services

The information model for the EiTender and the EiQuote Service artifacts is that of **[EMIX]**. EMIX provides a product description as well as a schedule over time of prices and quantities.

7.2.3 Operation Payloads for the EiTender Service

The **[UML]** class diagram describes the payloads for the EiTender and EiQuote service operations.

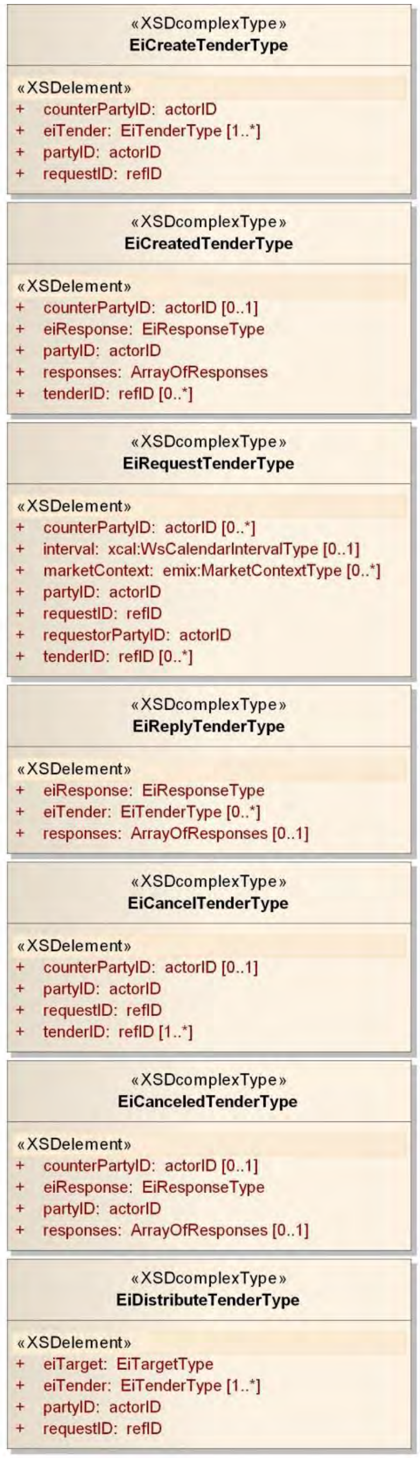


Figure 7-6: UML Class Diagram for the Operation Payloads for the EiTender Service

7.2.4 Operation Payloads for the EiQuote Service

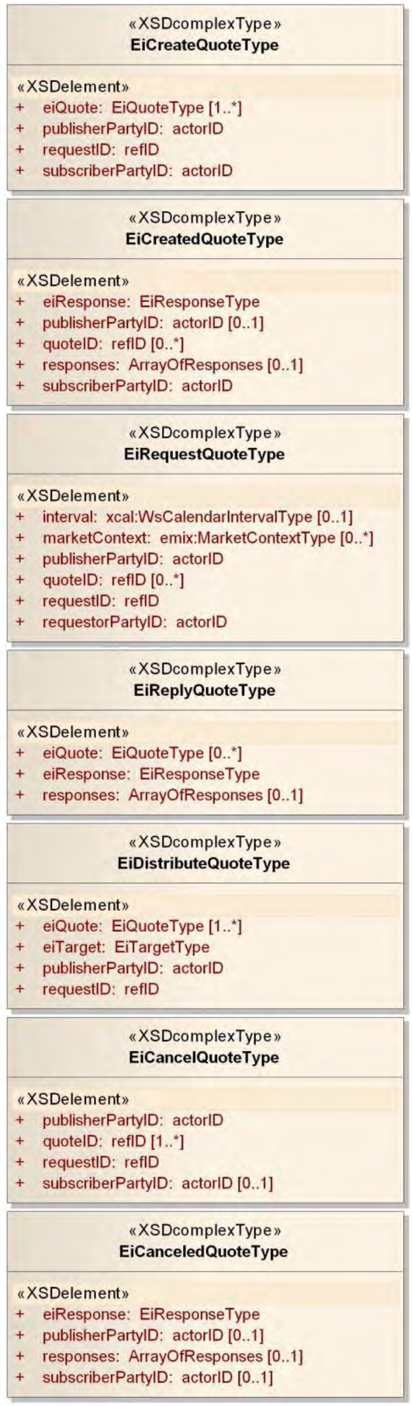


Figure 7-7: UML Class Diagram for the EiQuote Service Operation Payloads

7.3 Transaction Management Services

The service operations in this section manage the exchange of transactions. For example, in demand response, the [overarching] agreement is the context in which events and response take place—what is often called a *program.* This agreement is identified by the information element Market Context here and elsewhere.

There are no EiCancelTransaction or EiChangeTransaction operations. A compensating transaction SHOULD be created to clarify the economic effect of the reversal.[7]

Table 7-4: Transaction Management Service

Service	*Operation*	*Response*	*Service Consumer*	*Service Provider*	*Notes*
EiTransaction	EiCreateTransaction	EiCreatedTransaction	Party	Party	Create and send Transaction
EiTransaction	EiRequestTransaction	EiReplyTransaction	Party	Party	Request extant Transactions

7.3.1 Interaction Patterns for the EiTransaction Service

This is the **[UML]** interaction diagram for the EiTransaction Service:

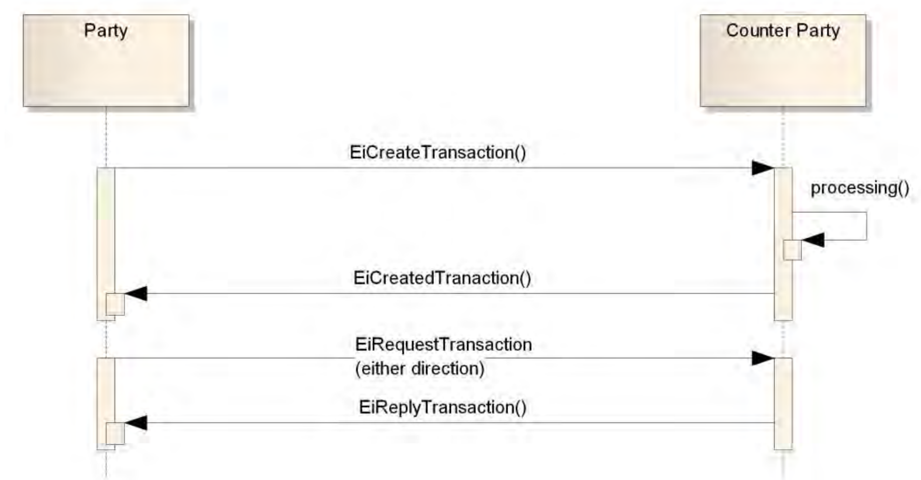


Figure 7-8: Interaction Diagram for the EiTransaction Service

7.3.2 Information Model for the EiTransaction Service

Transactions are **[EMIX]** artifacts with the identification of the Parties.

[7] This is consistent with the way that distributed agreement protocols such as [WS-BusinessActivity] manage compensation rather than cancelation.

7.3.3 Operation Payloads for the EiTransaction Service

The **[UML]** class diagram describes the payloads for the EiTransaction service operations.

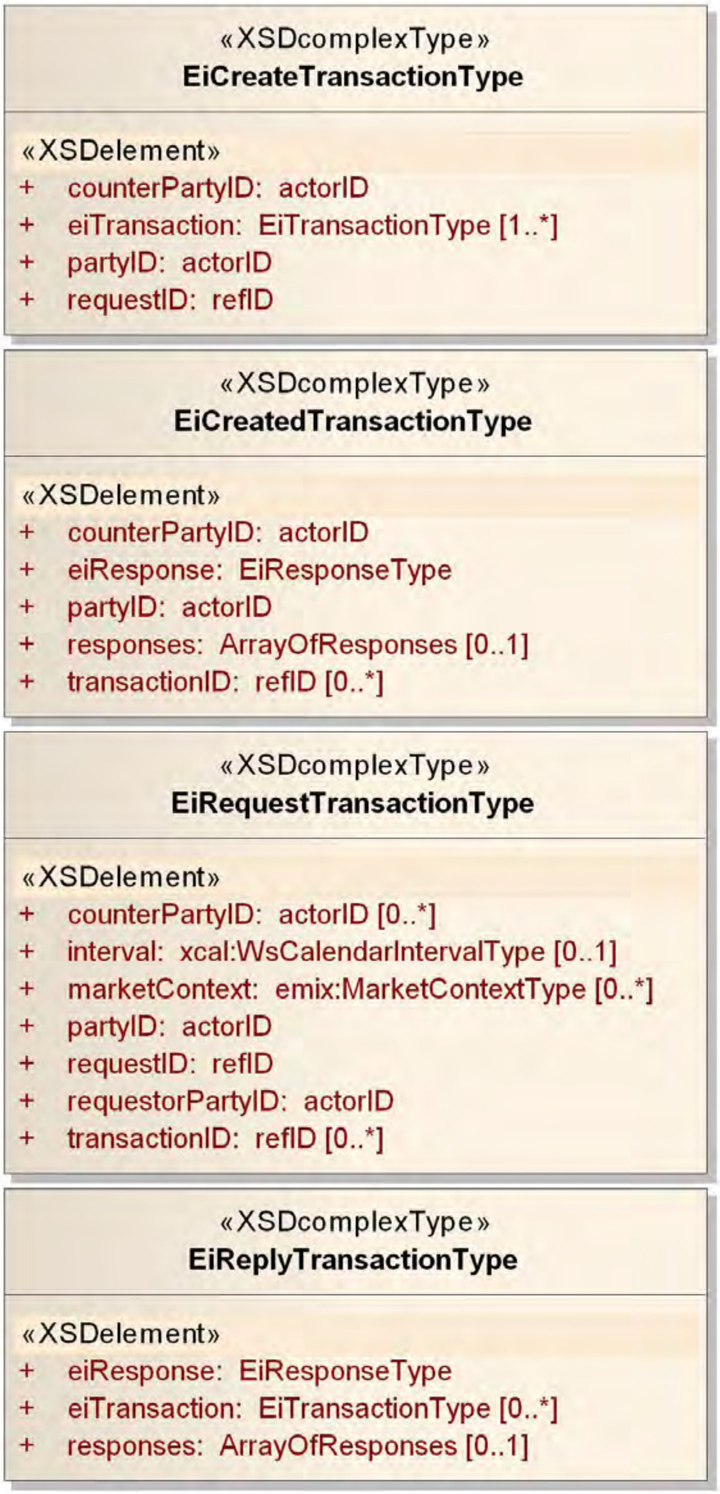


Figure 7-9: UML Class Diagram of EiTransaction Service Operation Payloads

7.4 Comparison of Transactive Payloads

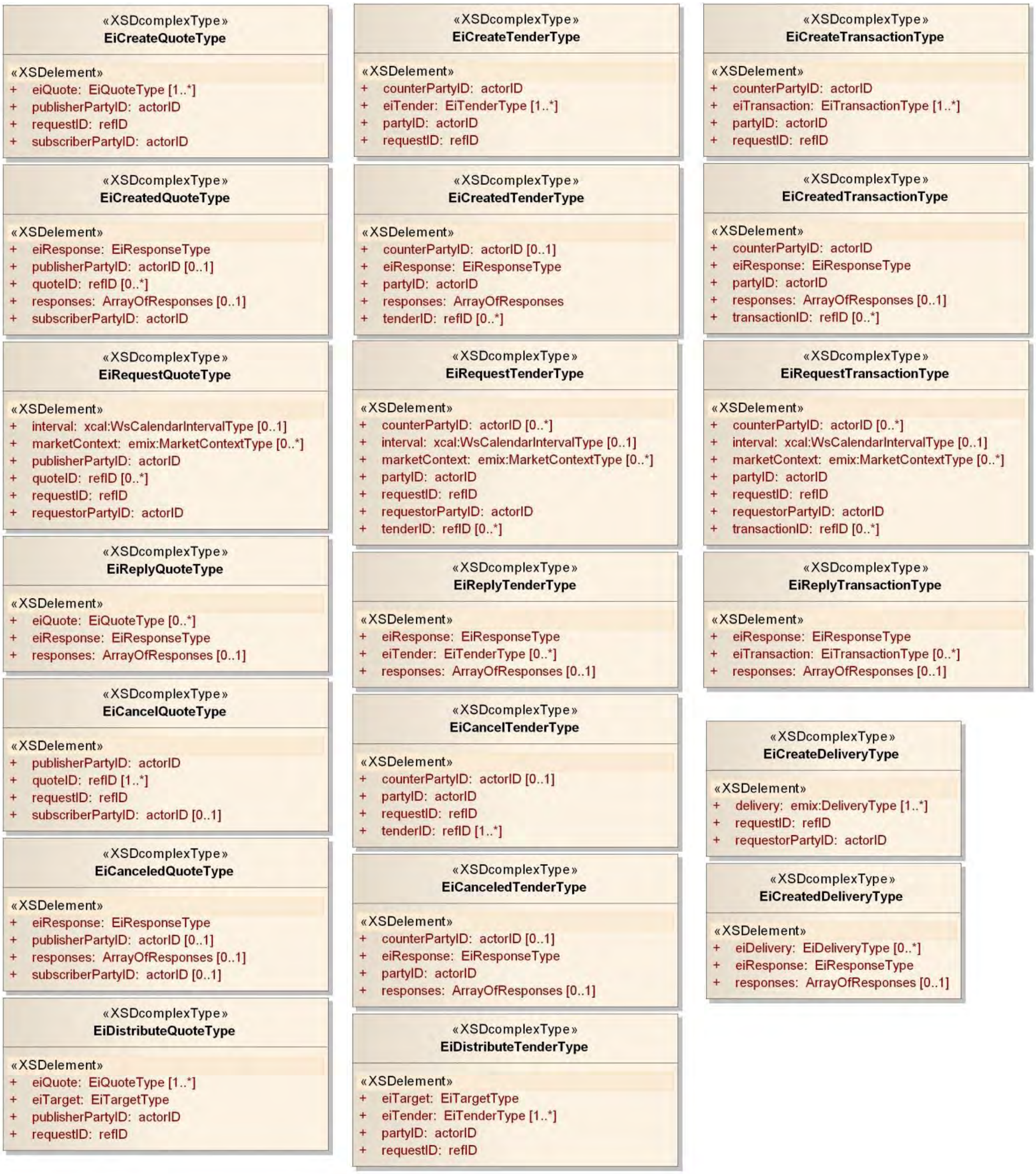


Figure 7-10: UML Diagram comparing all Transactive Payloads

7.5 Post-Transaction Services

In a market of pure transactive energy, verification would be solely a function of meter readings. The seed standard for smart grid meter readings is the NAESB Energy Usage Information **[NAESB EUI]** specification.

In today's markets, with most customers on Full Requirements tariffs, the situation is necessarily more complex. Full Requirements describes the situation where purchases are not committed in advance. The seller is generally obligated to provide all that the buyer requires. Full requirements tariffs create much of the variance in today's DR markets.

These sections will apply a measurement model consistent with the **[NAESB EUI]** as in the EiReport Services.

7.5.1 Energy Delivery Information

These operations respond with Energy Usage Information or any other single item of interest to the caller. This is very simple, requesting one thing measured for one interval, and waiting to return a value until the information is available. For anything more complex the Report Services should be used.

Table 7-5: Energy Delivery

Service	*Operation*	*Response*	*Service Consumer*	*Service Provider*	*Notes*
EiDelivery	EiCreateDelivery	EiCreatedDelivery	Party	Party	Party-to-Party, specifying interval, what is to be measured, and the direction for the measurement

7.5.1.1 Interaction Pattern for the EiDelivery Service

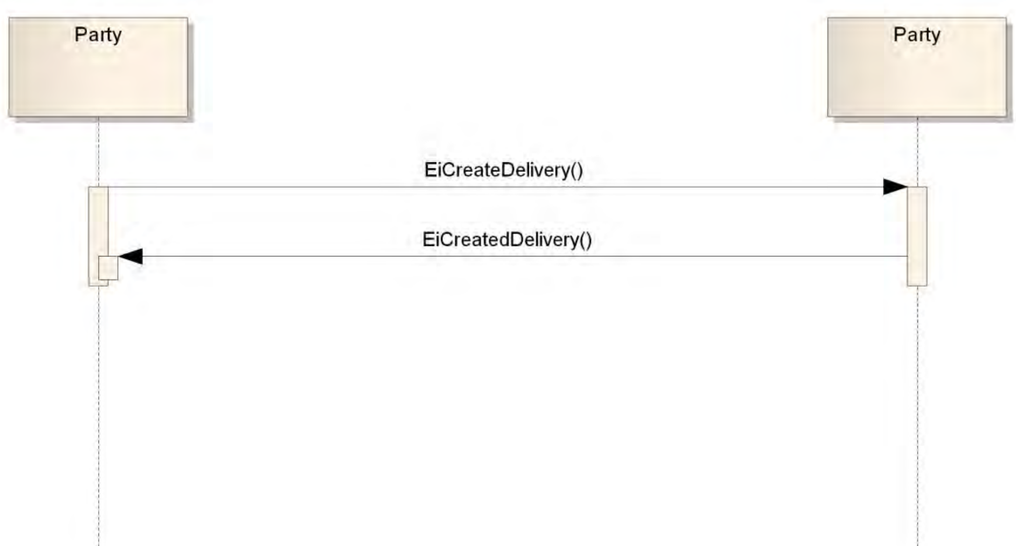


Figure 7-11: Interaction Diagram for Delivery Service

7.5.1.2 Information Model for the EiDelivery Service

The EiDelivery Type is a simplified EiReport.

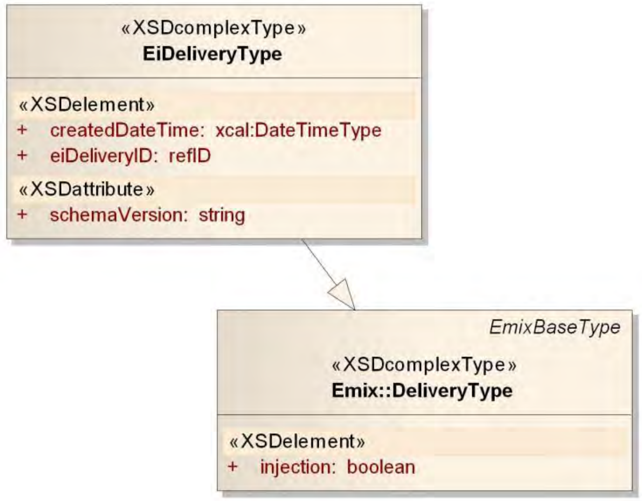


Figure 7-12: UML of EiDelivery Type

7.5.1.3 UML of EiDelivery Payloads

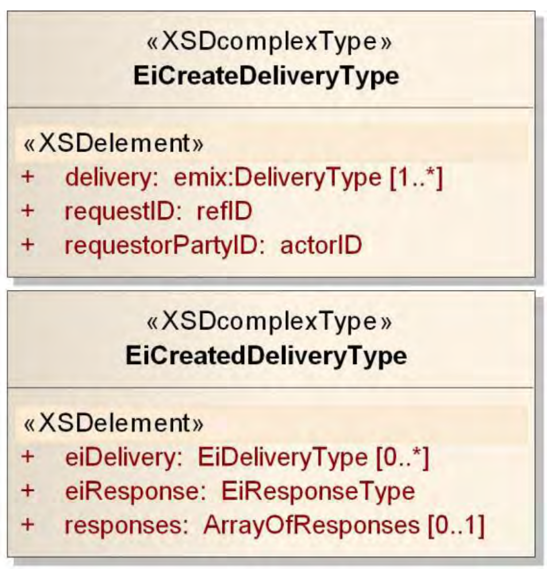


Figure 7-13: UML Class Diagram of Delivery and Delivery Payload

8 Enroll Service

Enrollment is distinct from Registration in Energy Interoperation. Registration establishes an identity for an actor (a party or a device such as a generator or a meter on a premise). Enrollment establishes a relationship between two actors as a basis for further interactions. Energy Interop supports two classes of interactions; Transactive and VTN/VEN interactions.

In the case of enrollment in Transactive Interactions, the Enrollment Service identifies the two parties and the Enabling Agreement, Market, Tariff, Purchasing Selling, etc. that the parties agree to use for their interactions.

In the case of enrollment in a VTN/VEN relationship the enrollment service identifies the two actors, generally a registered Resource and a Service Provider acting as a Designated Dispatch Entity (DDE). Registration of a Resource may sometimes be automatic with enrollment of the Resource.

The entities described in the following table can be enrolled. These are described in the **[UML]** diagrams as concrete classes that inherit from the Enrollee type. The strings are used to describe the entity; the standard approach to extensibility where a prefix of “x-“ indicates an extension SHALL be used.

The types of entity used may depend on the implementation. All implementations SHALL support Resources.

Table 8-1 Enrollee Descriptions

Entity	String	Description	Comment
Aggregator	aggregator	An entity that combines or aggregates generation or consumption	
Consumer	customer	An entity that is generally a net consumer of electricity	
Distribution	distribution	An entity that distributes electricity	E.g. a distribution utility
Enrolling Authority	enrollingAuthority	An entity that can perform enrolling services	
Generator	generator	An entity that is generally a net producer of electricity	
Load Serving Entity	lse	An entity which supports loads rather than generation	
Market	market	A Market that enrolls in another Market Context	
Meter Authority	meterAuthority	An entity that provides metering services	
Resource	resource	An EMIX Resource with additional information	A Resource including performance envelope and additional information including Resource Name

Entity	String	Description	Comment
Scheduling Entity	schedulingEntity	An entity that provides scheduling services	
Service Provider	serviceProvider	An entity that provides services	A potential provider of services to the VTN in support of VTN business processes
Supplier	supplier	An entity that is generally a net supplier of electricity	
System Operator	systemOperator	An entity that operates a grid	
TDSP	tdsp	An entity which supports transmission and distribution of electricity	
Transmission	transmission	An entity which supports transmission of electricity	

Table 8-2: EiEnroll Service Operations

Service	*Operation*	*Response*	*Service Consumer*	*Service Provider*	*Notes*
EiEnroll	EiCreateEnroll	EiCreatedEnroll	Party	Party	Create and send Enrollment
EiEnroll	EiRequestEnroll	EiReplyEnroll	Party	Party	Requests outstanding Enrollment information; request semantics with no time Interval.
EiEnroll	EiCancelEnroll	EiCanceledEnroll	Party	Party	Cancel one or more Enrollments

8.1 Interaction Patterns for the EiEnroll Service

This is the **[UML]** interaction diagram for the EiEnroll Service.

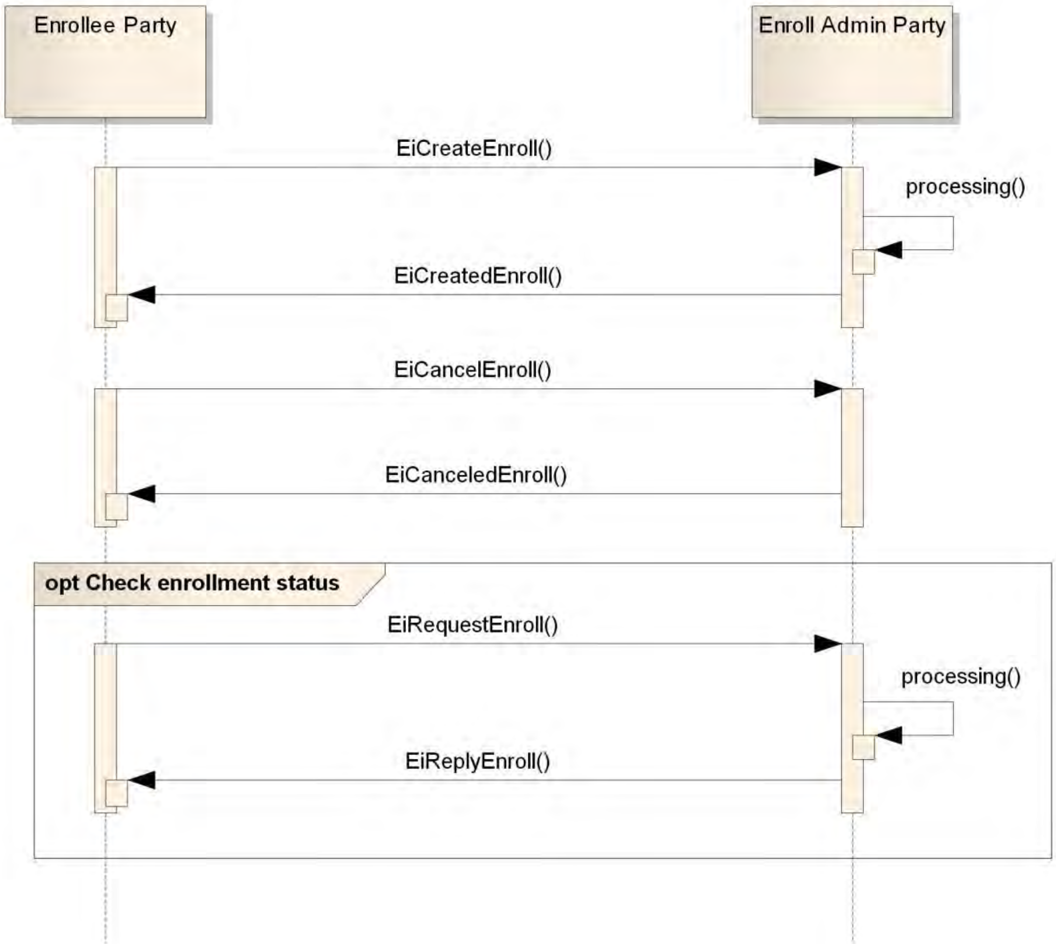


Figure 8-1: Interaction Diagram for the EiEnroll Service

8.2 Information Model for the EiEnroll Service

The EiEnroll service has an abstract class for the respective types. The abstract class also has the entity identifier, type (as a string), and name. The standard values for the type are listed in Table 8-1 Enrollee Descriptions. Other values MAY be used but MUST be prefixed by "x-" as described in Appendix C

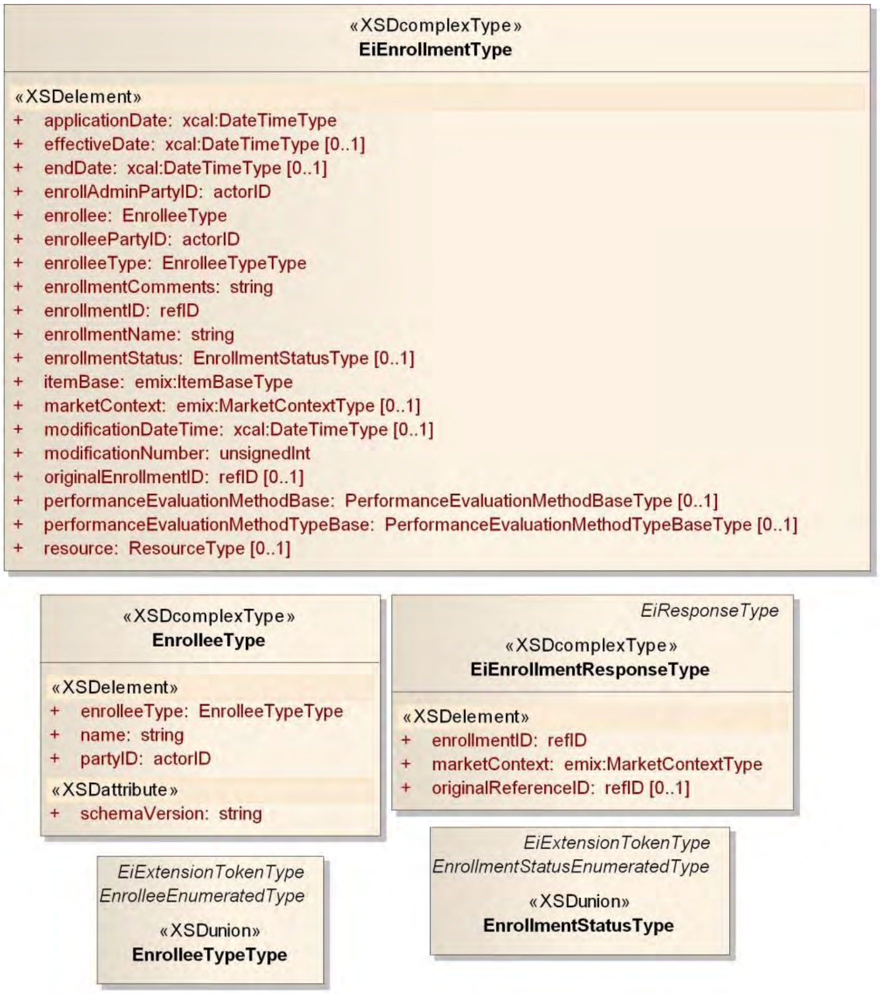


Figure 8-2: UML Model for EiEnrollment Classes

8.3 Enrollee Types

The **[UML]** class diagram describes the Enrollee Types.

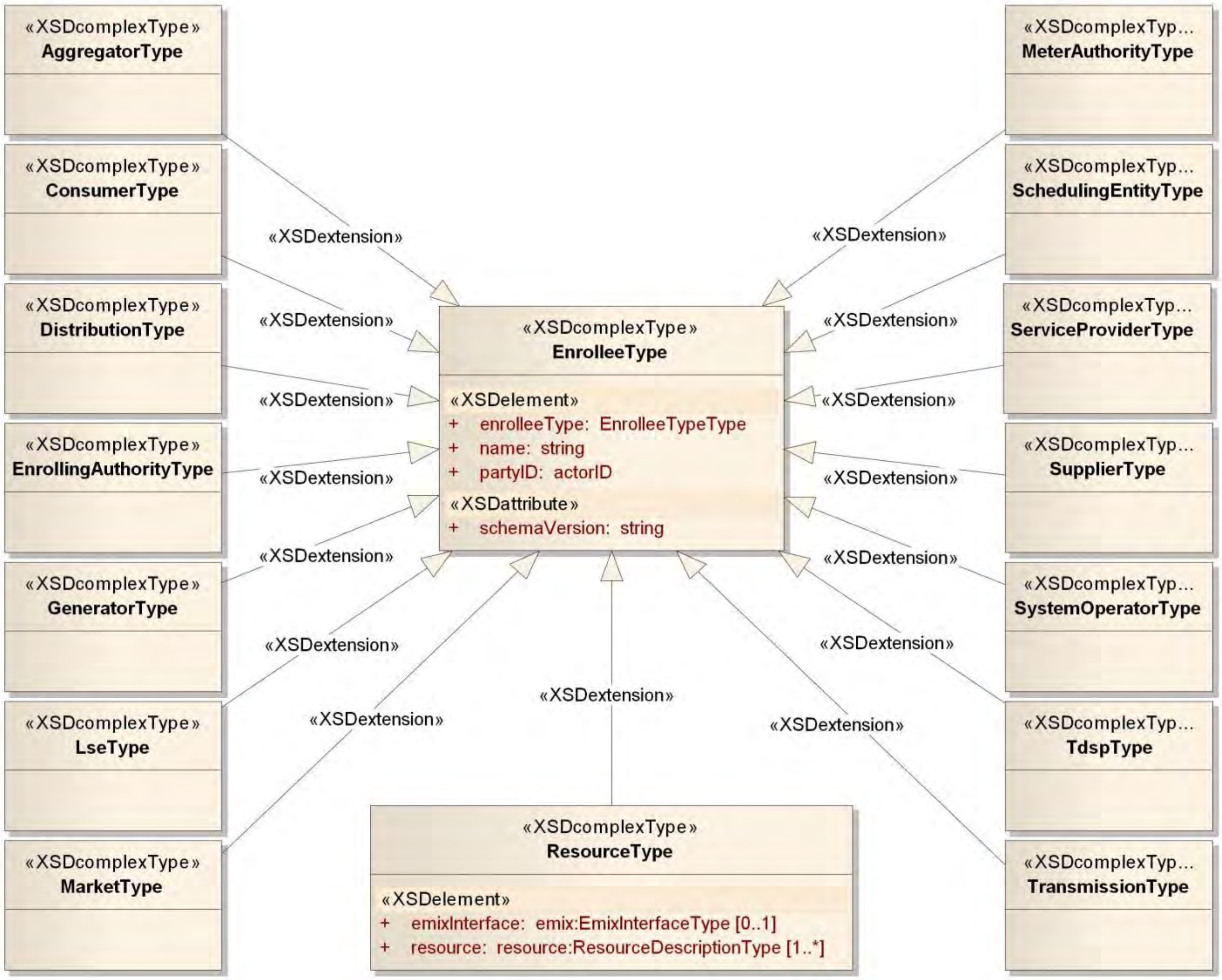


Figure 8-3: UML Class Diagram showing Enrollee Types

8.4 Operation Payloads for the EiEnroll Service

The **[UML]** class diagram describes the payloads for the EiEnroll service operations.

Figure 8-4: UML Class Diagram for Enrollment Payloads

9 Event Services

The Event Service is used to call for performance under a transaction. The service parameters and event information distinguish different types of events. Event types include reliability events, emergency events, and more—and events MAY be defined for other actions under a transaction. For transactive services, two parties may enter into a call option. Invocation of the call option by the Promissee on the Promissor can be thought of as raising an event. But typically the Promissee may raise the event at its discretion as long as the call is within the terms of the call option transaction.

For example, an ISO that has awarded an ancillary services transaction to a Party may issue dispatch orders, which can also be viewed as Events. In this specification, what is sometimes called a *price event* would typically be communicated using the EiSendQuote operation (see 7.2 *"Pre-Transaction Services")*.

Table 9-1: Event Services

Service	*Operation*	*Response Operation*	*Service Consumer*	*Service Provider*	*Notes*
EiEvent	EiCreateEvent	EiCreatedEvent	VTN	VEN	Create and send a new Event
EiEvent	EiChangeEvent	EiChangedEvent	VTN	VEN	Modify an existing Event
EiEvent	EiRequestEvent	EiReplyEvent	Either	Either	Request outstanding Events; request semantics with optional time Interval
EiEvent	EiRequestPending Event	EiReplyPending Event	Either	Either	Similar to Request Events except that Reply returns Event IDs and Modification Numbers only.
EiEvent	EiCancelEvent	EiCanceledEvent	VTN	VEN	Cancel one or more Events
EiEvent	EiDistributeEvent	—	VTN	VEN	Broadcast of Event.

9.1 Information Model for the EiEvent Service

The event is the core Demand Response information structure, and the most complex of the payloads. Understanding the information model of the Event is critical to understanding the operations of the Event Services. This section reviews the Event semantics as defined in Section 5.3 *"Event-based Interactions"*.

The sub-sections below provide a reprise of the Event structure (9.1.1) and a UML description of the event (9.1.2)

9.1.1 Structure of the Event

The semantics of the Event are defined Section 5.3 *“Event-based Interactions”*.

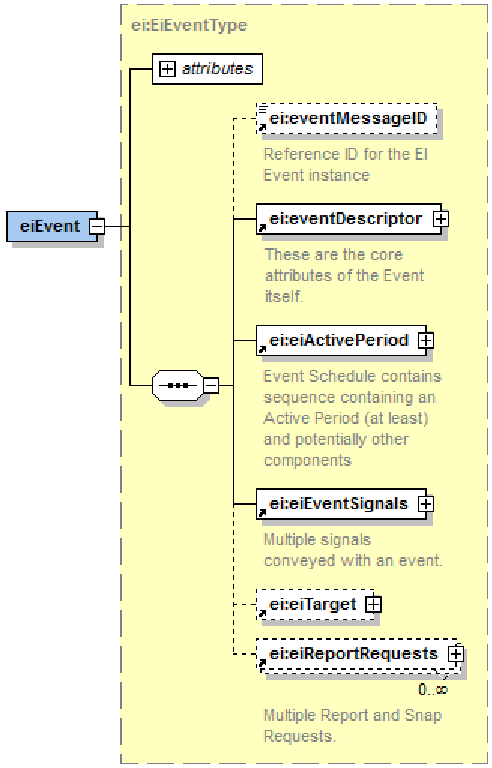


Figure 9-1: EiEvent summarized

The type EiEvent MAY be identified by an Event Message ID and which has associations with the classes Active Period, Event Descriptor, and Event Signals, a collection of Signals and Baselines.

Since the event is the core Demand Response information structure, we begin with Unified Modeling Language **[UML]** diagrams for the EiEvent class and for each of the operation payloads. Core semantics for the Event are defined in Section 5.3 *“Event-based Interactions”.*

9.1.2 UML Model of an Event and its Signals

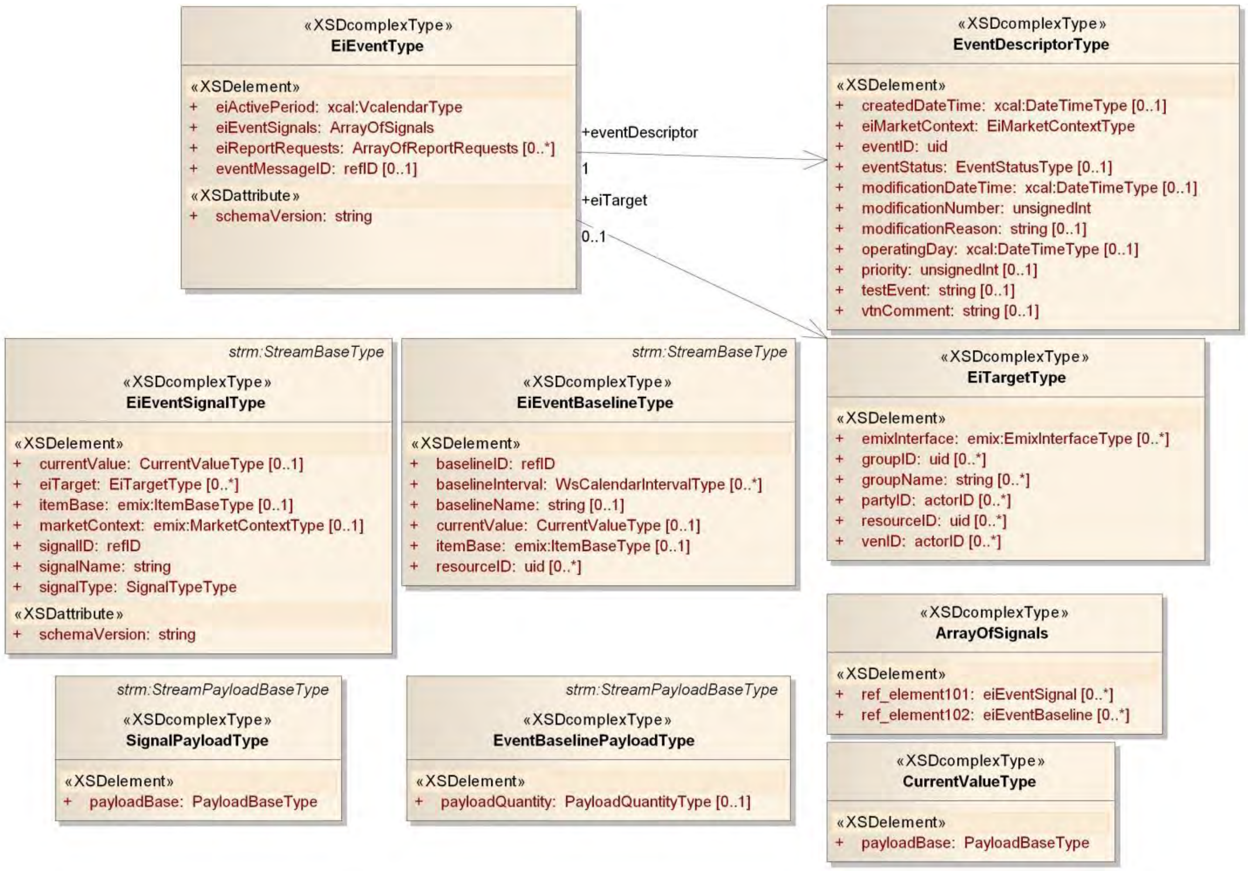


Figure 9-2: UML Class Diagram for EiEventType and Related Classes (w/o Signals detail)

An Event may include a number of Schedules, which are expressed as Streams. These schedules are the Signals, the Baselines, and they may return Baselines, Reports, and Delivery. The EI Event Signal derives from the Streams element and conveys elements of the Type Signal Payload in its Schedule.

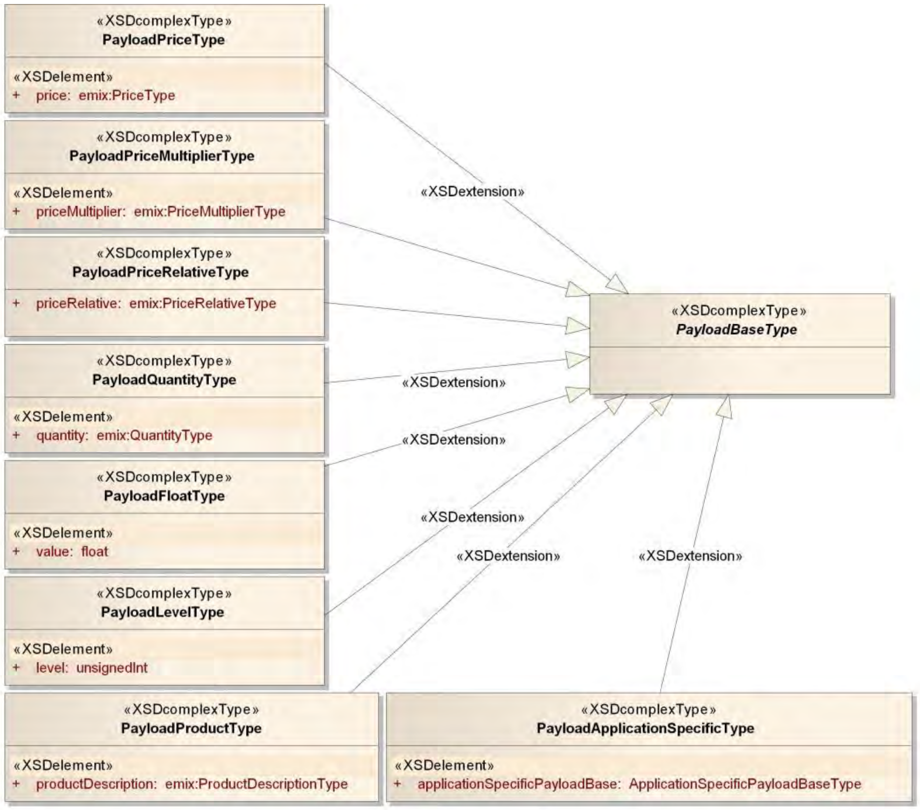


Figure 9-3 UML Class Diagram Showing Details of the Signal Payloads or EiEventSignals

9.2 Special Semantics of the Event Request Operations

The Events are the largest messages exchanged in Energy Interoperation. They exist in two forms, the EiEventRequest, and EiEventRequestPending. EiEventReply returns entire Events in response to a Request, following the general pattern of all Energy Interoperation Services. EiEventRequestPending returns the Event IDs and Modification Numbers only. EiEventRequestPending is useful for black-start and other situations in which the VEN and VTN need to assess the information shared with its partner.

The Modification Number returned in the Replies is for assessment only. The recipient MAY use it to determine that the sender is using out-of-date information, but any replacement or update SHALL convey the current Modification.

9.2.1 Event Ordering

The Event Requests include an option to restrict the number of Events returned in in Reply to any Request. For consistency, this requires that a VTN or VEN be able to order Events. The rules for ordering Events are applied sequentially as follows:

1. Active events have priority over pending events
2. Within Active Events, 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 events the one with the earlier start time has the higher priority

5. After processing rules 1-4, if Priority is still indeterminate within a set of Intervals, then the order is indeterminate within that set. A Reply containing Events with indeterminate Order MUST maintain that order in response to successive Requests while they remain indeterminate.

The definitions of Active and Pending are consistent with those described for the Event Filter in Table 9-2.

9.2.2 Event Filter described

Both the Event Request operations MAY use of the Event Filter to restrict the Events exchanged during Request and Reply.

Table 9-2: Event Filter described

Event Filter	Description
Active	An event qualifies if the Active Interval coincides with the Request. If specified with an accompanying Interval, an Event qualifies if any part of the Active Interval occurs within the specifying Interval; without accompanying Interval, "now" is treated as an infinitesimal Interval with a current starting date and time.
Pending	An event qualifies if the Active Interval starting date and time is in the future. If specified with an accompanying Interval, the Event qualifies if the Active Interval has not started (is not Active) at the Start of the Interval, and the Active Interval start is within the bounds of the specifying Interval.
All	An event qualifies if it would qualify as either Active or Pending.
Completed	An Event qualifies if the Active Interval is completed before the Request. If specified with an accompanying Interval, and Event qualifies if the end of the Active Interval before the start of the Requesting Interval. Conforming profiles MAY return a NULL set in response to a Request for Completed Intervals, as there is no requirement to store or be able to retrieve Completed Events.
Cancelled	An Event qualifies if it has been Cancelled. If specified with an accompanying Interval, and Event qualifies if the Event would have qualified as Active during the Interval. Conforming profiles MAY return a NULL set in response to a request for Completed Intervals as there is no requirement to store or be able to retrieve Cancelled Events.

9.2.3 Using EiRequestEvent EiRequestEventPending together

The two Request operations in the Event Service are essentially the same. Each enables a VEN or VTN to query its partner about what Events it knows. The difference is in the Replies. EIReplyEvent returns a collection of Events, EIReplyEventPending returns a collection of Qualified Event IDs. i.e., an Event ID and the Modification Number.

Figure 9-4: Qualified Event ID

With a list of Qualified Event IDs either one knows about the other. Events that are missing can be requested or sent. A VEN can infer cancellation when its VTN removes an Event ID. Using the

Modification Number, a VTN can know to re-send the latest version, or a VEN can know to request an update.

While the Event Requests follow the pattern common to all EI Requests, because of the extra options, they are summarized in table [reference] below. All query elements are optional.

Table 9-3: Event Requests summarized

Request Element	Description
VEN ID	Names the VEN that is Requesting or currently knows of these Events
Event ID	A list of Event IDs to be returned. If present, all other filters are ignored.
Market Context	Request is to return Events that are in a Market Context. For example, in a given Program, a VEN could request all Electric Vehicle (EV) related Events.
Filter	As described above (Table 9-2). Can be combined with Interval
Interval	Requests Events “within” an Interval. Interval may contain only a Start Date to request all Events from that date forward, or may include only an End Date to include events before that Date. If no Filter is present, this is interpreted as if the Filter were “all”.
Reply Limit	Return only the first N matching events, where N is the Reply Limit. “First is defined according to the Order as described above.

A common pattern for either a VEN or a VTN to request Event IDs with the EiRequestPending, and to then request information about events that it is missing or that need updates using EiRequestEvent. A VTN after a similar query might use EiCreateEvent to pass the missing or updated Events to the VEN

9.3 Interaction Patterns for the EiEvent Service

This is the **[UML]** interaction diagram for the EiEvent Service.

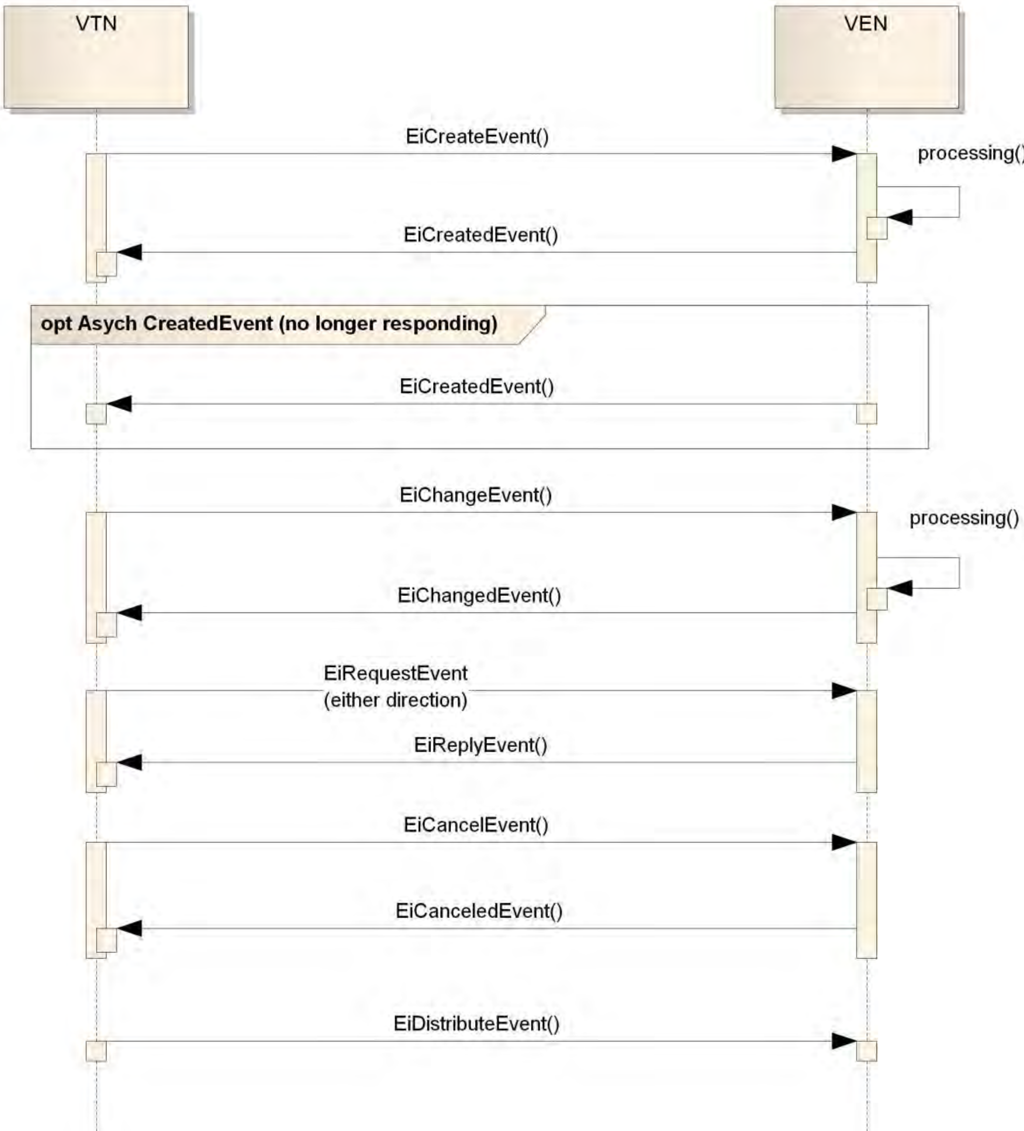


Figure 9-5: UML Interaction Diagram for the EiEvent Service Operations

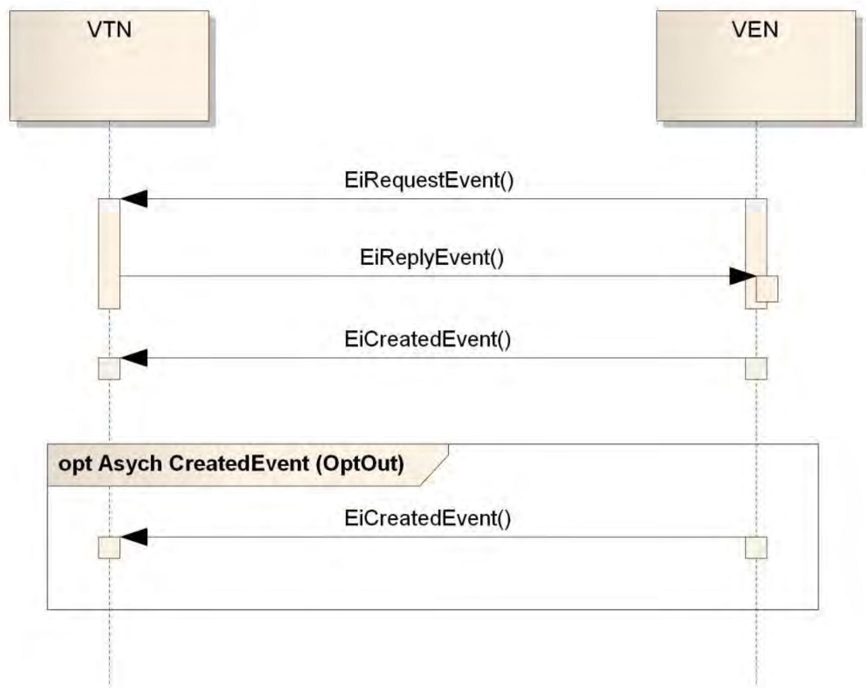


Figure 9-6: UML for example PULL pattern for EiEvent

Figure 9-7: Interaction Diagram for Pending Event operation

9.4 Operation Payloads for the EiEvent Service

The **[UML]** class diagram describes the payloads for the EiEvent service operations.

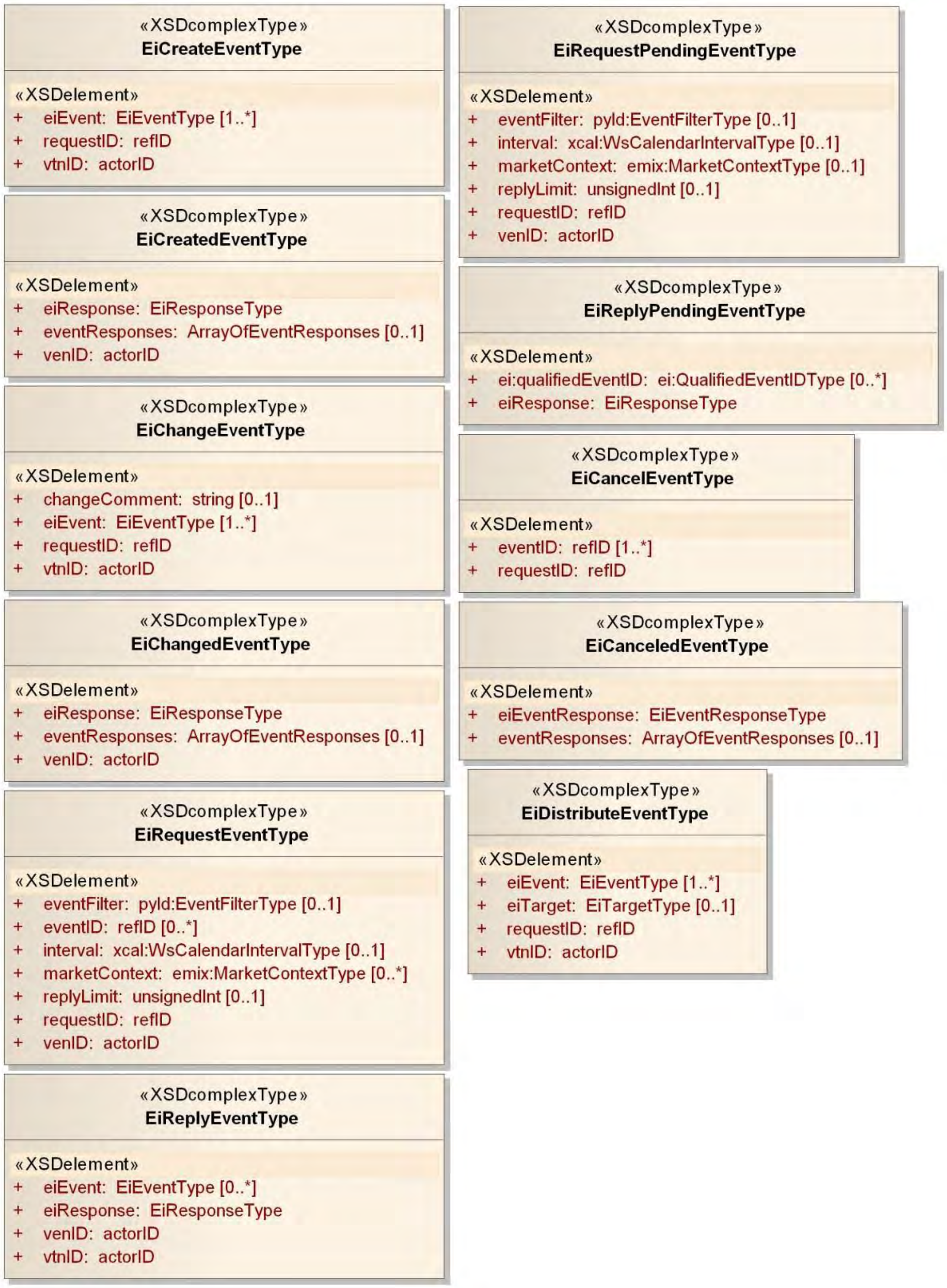


Figure 9-8: UML Class Diagram for EiEvent Service Operation Payloads

10 Report Service

Energy Interoperation Reports convey information from remote sensing or about remote state back to the requester. The Historian operations support the collection of data for Reports. Reports can be associated with an Event or can be requested through the Report Services described in this section.

The general pattern of the Report service is to request that the Historian gather data, and for the Report Service to deliver the Report when it is Ready. A single history may generate only a final Report, or it may report-back periodically. The report requester MAY ask for the report-to-date, or for a time-constrained portion of the Report at any time while it is running.

One interaction pattern for the Report service is what one may call “Set and Forget”. Under this pattern, the Requester asks that information be logged, but specifies no Report delivery. Under this pattern, the Requester can, at any time, request delivery of a Report for a specified Interval.

Projections are a special class of Reports, i.e., Reports about the future. Projections follow the general form of Reports and include additional metadata about the reliability of the future information in each window.

The semantics of Reports are described in sections 5.4 *“Monitoring, Reporting”* and 5.5 *“Reports, Snaps, and Projections”.*

The range of Payloads that can be delivered by means of a Report can be extended by deriving new types from the Payload Base Type, and defining a new Report Type not in Enumerated Report Types, and requesting such a Report.

10.1 Overview of Report Services

Event-based reports are requested as part of the EiEvent service. EI Report operations request Reports independently of any Event. Whether created as part of an Event or independently, all Reports support the same post-creation operations

EiReport operations are independent of EiEvent operations in that they can be requested at any time independent of the status or history of EiEvents.

Table 10-1: Report Service

Service	Operation	Response	Service Con-sumer	Service Provi-der	Notes
EiReport	EiCreateHistorian	EiCreatedHistorian	any	any	Create a new Historian and start it recording indicated information
EiReport	EiRequestHistorian	EiReplyHistorian	any	any	Reply with HistorianIDs that meet the criteria
EiReport	eiCancelHistorian	eiCanceledHistorian	any	any	Cancel Historian recording, optionally requesting a final report
EiReport	eiCreateProjection	eiCreatedProjection	any	any	Creates a projection, returned as a report stream

Service	Operation	Response	Service Con-sumer	Service Provi-der	Notes
EiReport	eiCreateReport	eiCreatedReport	any	any	One time and/or periodic response
EiReport	EiUpdateReport	EiUpdatedReport	any	any	Used to update the Report, e.g. periodic responses
EiReport	EiRequestReport	EiReplyReport	any	any	The carrier for periodic response
EiReport	eiCancelReport	eiCanceledReport	any	any	Cancel pending reports, optionally requesting a final report

10.1.1 Interaction Pattern for Historian Operations

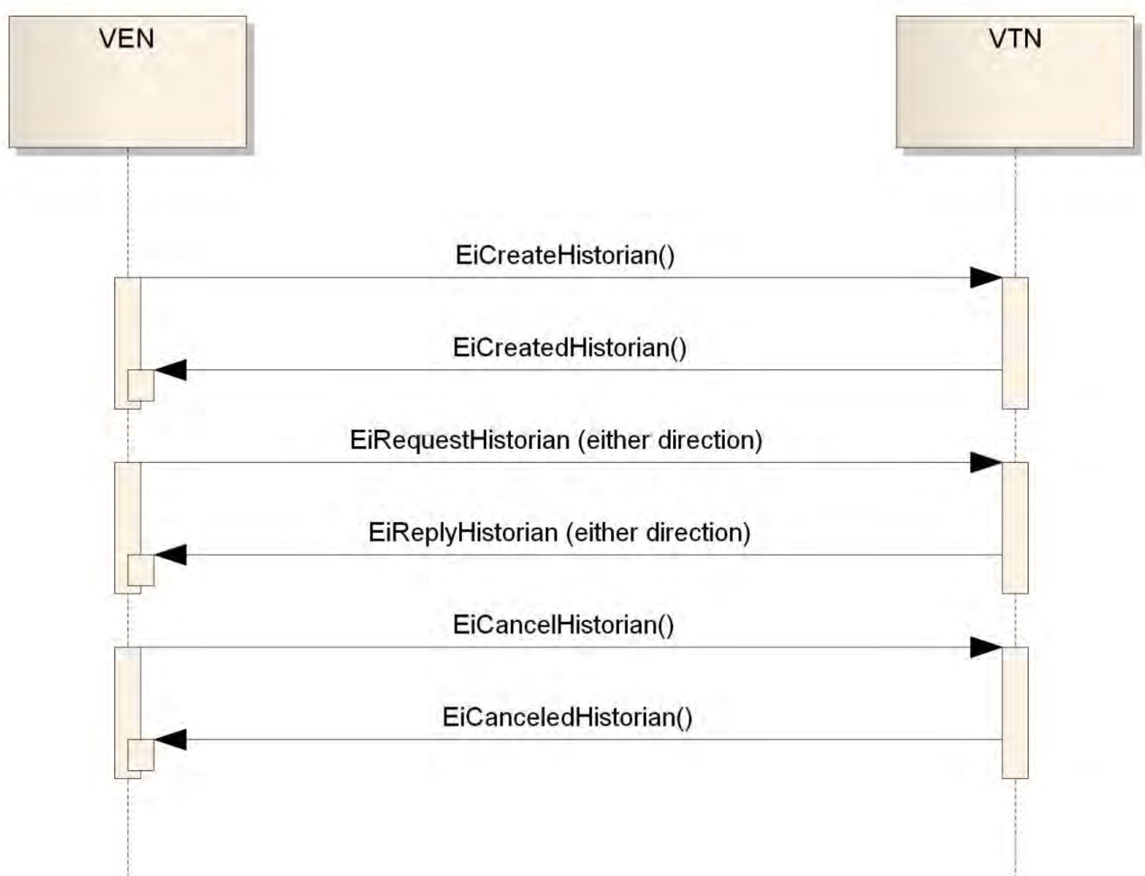


Figure 10-1: Interaction Pattern for Historian Operations (Report Service)

10.1.2 UML Diagram of Historian Operations Payloads

Figure 10-2: UML Diagram of Historian Payloads

10.1.3 Interaction Pattern for the Report Operations

This is the **[UML]** interaction diagram for the EiReport Service.

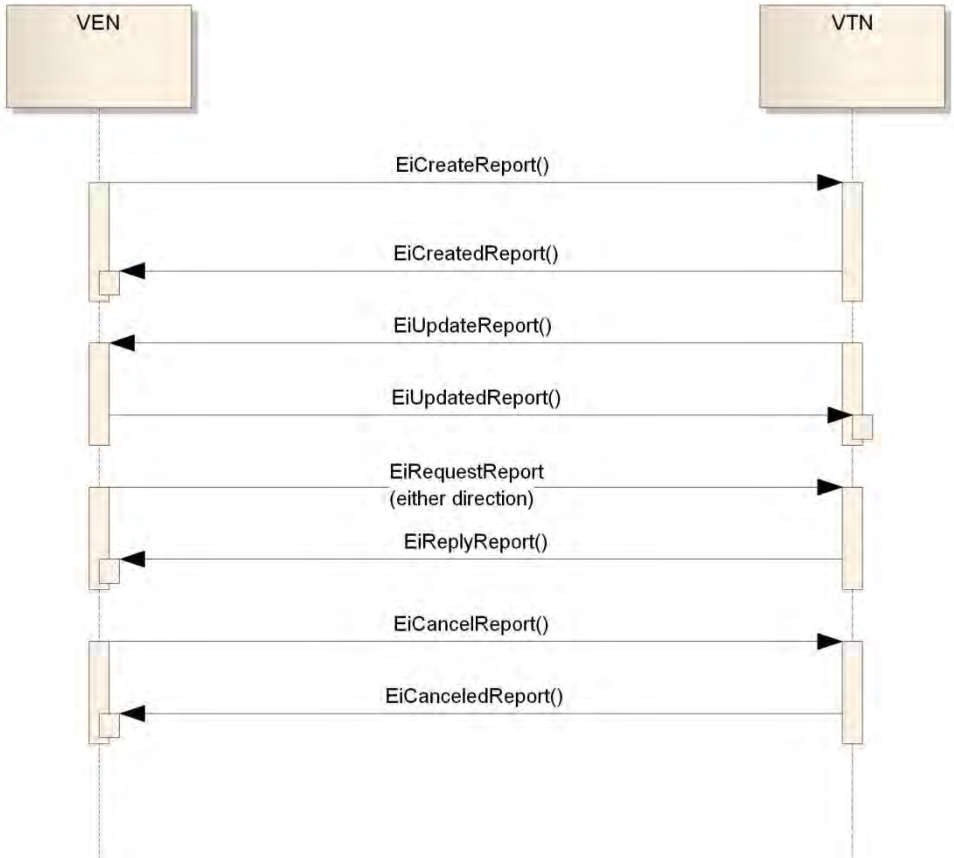


Figure 10-3: UML Interaction Diagram for the EiReport Operations (Report Service)

10.1.4 UML Diagram of Report Operations

Figure 10-4: UML Diagram of Report Payloads

10.1.5 Interaction Pattern for Projection Operations

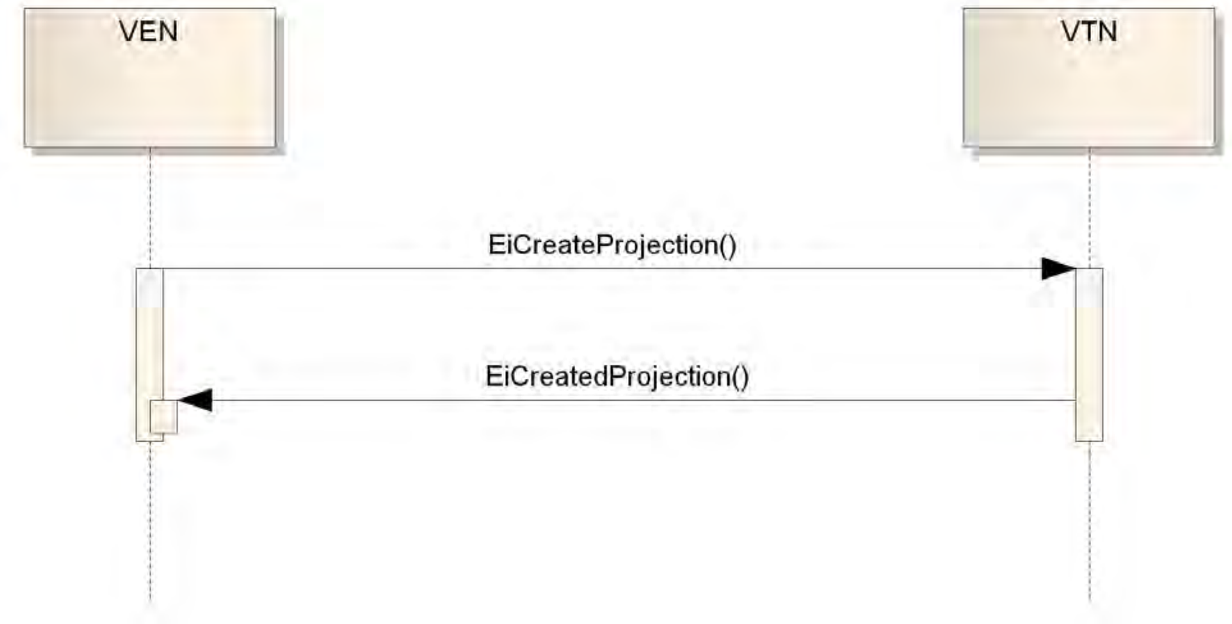


Figure 10-5: Interaction Pattern for Projection Operations (Report Service)

10.1.6 UML Diagram of Projection Operations

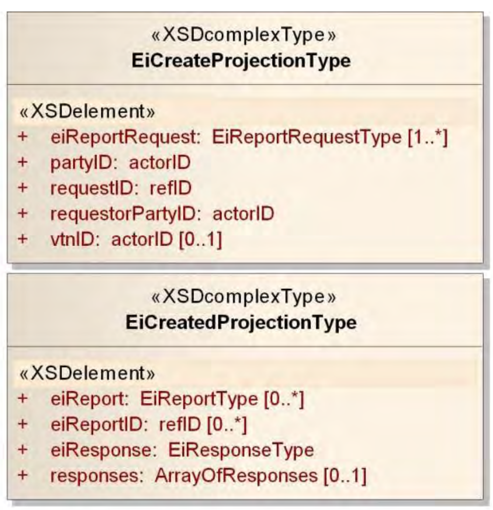


Figure 10-6: UML Diagram of Projection Payloads

10.1.7 Information Model for the EiReport

EiReport is prepared by a Party upon request and supplied to the requesting party. It may also be defined in the expectations of the Market Context.

Figure 10-7: UML Class Diagram for the EiReport Class

10.2 All Operation Payloads for the EiReport Service

The **[UML]** class diagram below recaps the payloads for all operations of the EiReportService.

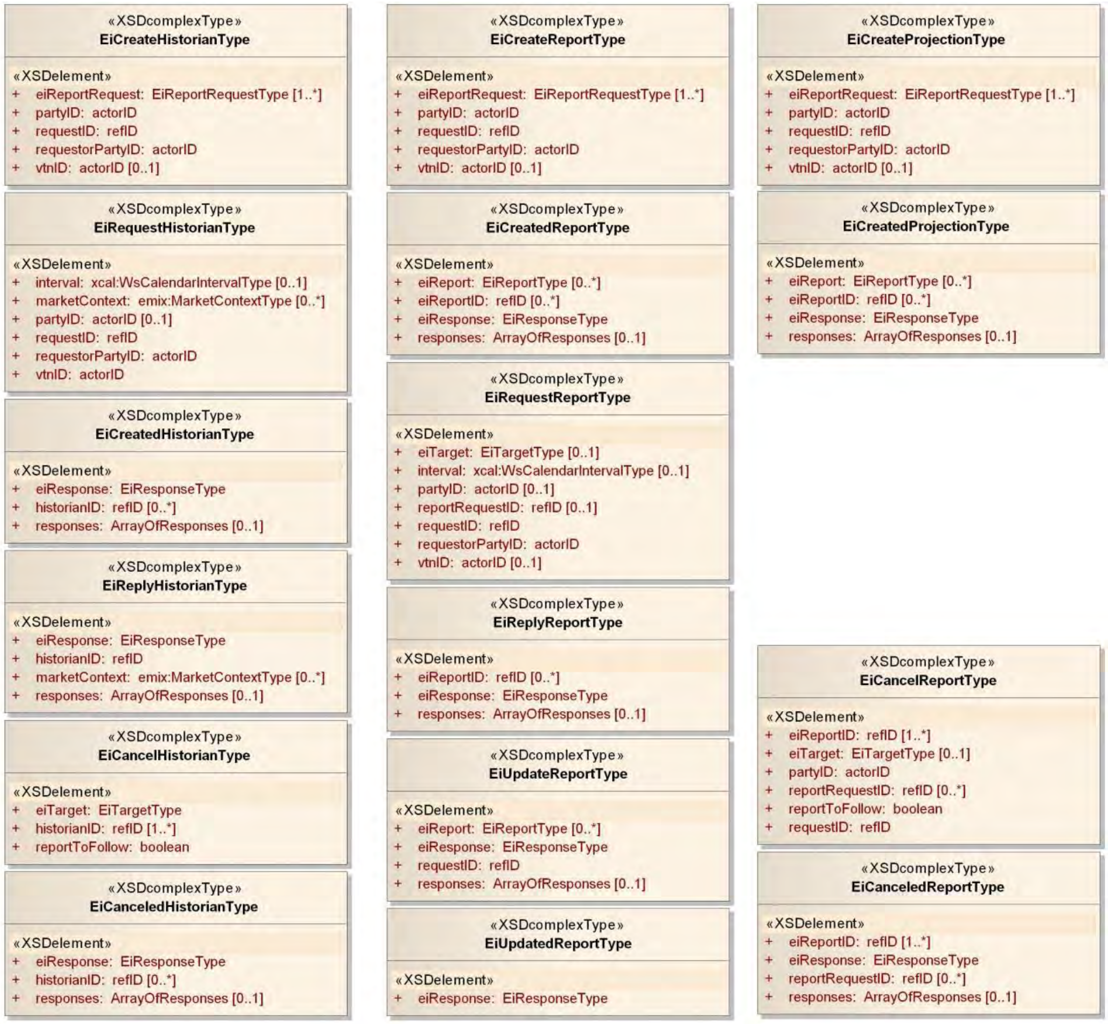


Figure 10-8: UML Class Diagram for all EiReportService Operation Payloads

11 Event Support Services

Users of **[OpenADR]** found that they needed to be able to constrain the application of remote DR services. For The DR Operator, advanced knowledge of these constraints improved the ability to predict results. The services in this section are based on the services used to tailor expectations in **[OpenADR]**.

Availability and Opt are similar in that they communicate when a Party is willing to receive an Event. Availability is a long-term schedule for when a Party will consider a response. Availability could be set in the Market Context or at program enrollment. Opt (as in *opt in* or *opt out*) encompasses short-term additions to or replacement of the schedule in Availability.

The combination of Availability and Opt states together define the times during which a committed response from the VEN is possible or likely.

11.1 Relationship of Availability and Opt Information

Availability and Opt apply to interactions where an action is requested (e.g. curtailment and DER actions), and only indirectly to (e.g.) price distribution interactions.

Availability is a long-term description and may be complex. Opt is a short-term description that replaces or is combined into the long-term availability description.

Availability and Opt-In and Opt-Out, as well as Market Rules, use the *VavailabilityType* defined in **[WS-Calendar]** which in turn is an XML serialization of **[Vavailability]**. The semantics are defined in **[Vavailability]**.

The behavior of the Availability schedule is defined as follows. We call the parameter passed for Opt-In and Opt-Out the *Opt Vavailability*.

- The EiAvailability class describes when the VEN expects/commits/plans to be available to respond to a request for performance, generally an EiEvent.
- Exactly one *Vavailability* is included in the *EiAvailability* and the *EiOpt* objects.
- An *EiOpt* that is used in a message MUST have a bounded interval (the *Opt Interval*) in the *Opt Vavailability*[8]
- An **Opt-In** while in effect adds the available times of the *Opt Vavailability* to the available times in the bounded interval for the VEN with respect to a MarketContext, effectively performing a logical OR operation on the available times but only within the opt Interval
- An **Opt-Out** while in affect replaces the entire portion of the EiAvailability within the opt interval
- Exactly zero or one Opt functions MAY be in effect at any time

In short, Opt-In adds the *Opt Vavailability* available times to the overall VEN vavailability; Opt-Out replaces the entirety of its *opt Intervals* with the contents of the Opt-Out Vavailability.

[8] By defining an end time for the *Vavailability*

11.2 EiAvail Service

The Availability[9] is set by the VEN and indicates when an event may or may not be accepted and executed by the VEN with respect to a Market Context. Knowing the Availability and Opt information for its VENs improves the ability of the VTN to estimate response to an event or request.

When Availability is set, opt-in or opt-out does not affect the Availability except for the specific interval(s) described by the Opt—opting out is temporary unavailability, which may have transaction and business consequences if an event is created during the opt-out period.

The modeling for Availability includes behavior indications for the situation where an EiEvent overlaps a constrained time interval.

EiAvailability describes only the available times, using the patterns defined in **[WS-Calendar]** and **[Vavailability]**.

Table 11-1: Avail Service

Service	Operation	Response	Service Consumer	Service Provider	Notes
EiAvail	EiCreateAvail	EiCreatedAvail	VEN	VTN	Create an Avail for this VEN; return the AvailID
EiAvail	EiRequestAvail	EiReplyAvail	VEN	VTN	Request Avail information for this VEN; request semantics with no time Interval
EiAvail	EiCancelAvail	EiCanceledAvail	VEN	VTN	Cancel the Avail referenced by the AvailID

The element EiAvailBehavior defines how an issued EiEvent that conflicts with the current EiAvail is performed:

- ACCEPT – accept the issued EiEvent regardless of conflicts with the EiAvail
- REJECT – reject any EiEvent whose schedule conflicts with the EiAvail
- RESTRICT – modify the EiEvent parameters so that they fall within the bounds of the EiAvail

[9] Called *Constraints* in **[OpenADR1]**

11.2.1 Interaction Patterns for the EiAvailability Service

This is the **[UML]** interaction diagram for the EiAvail Service.

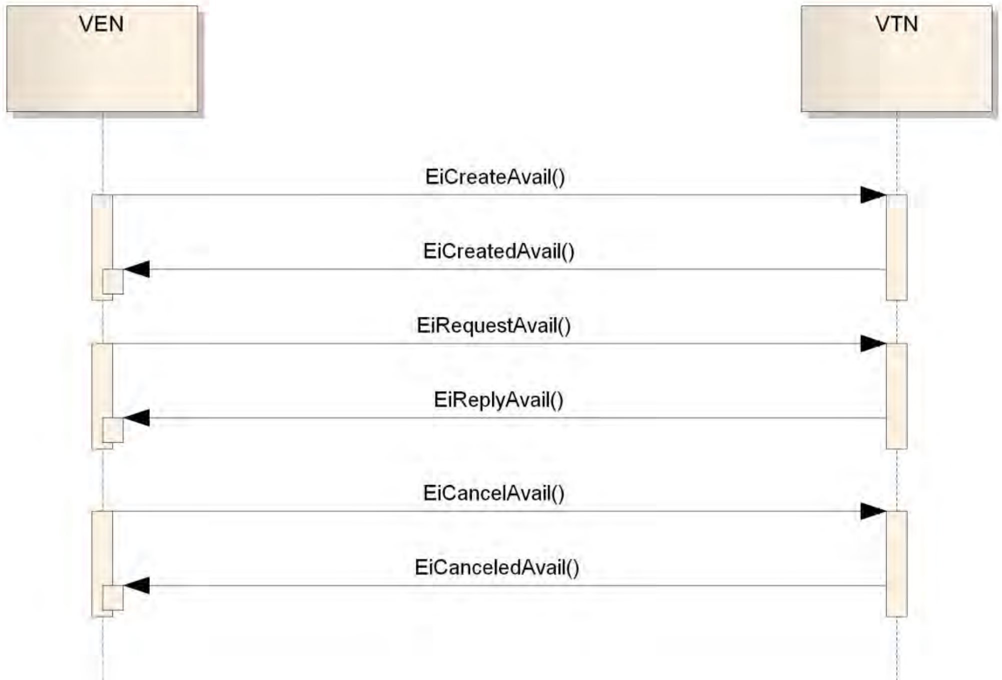


Figure 11-1: Interaction Pattern for the EiAvailability Service.

11.2.2 Information Model for the EiAvail Type

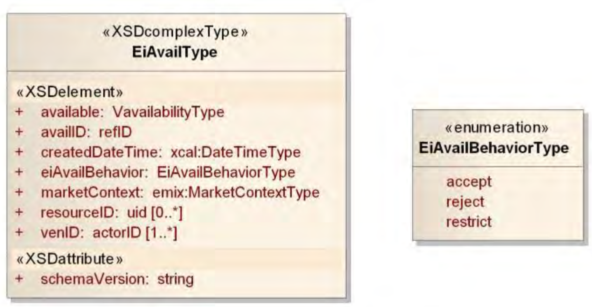


Figure 11-2: UML Class Diagram for the EiAvail Type

11.2.3 Operation Payloads for the EiAvail Service

The **[UML]** class diagram describes the payloads for the EiAvail service operations.

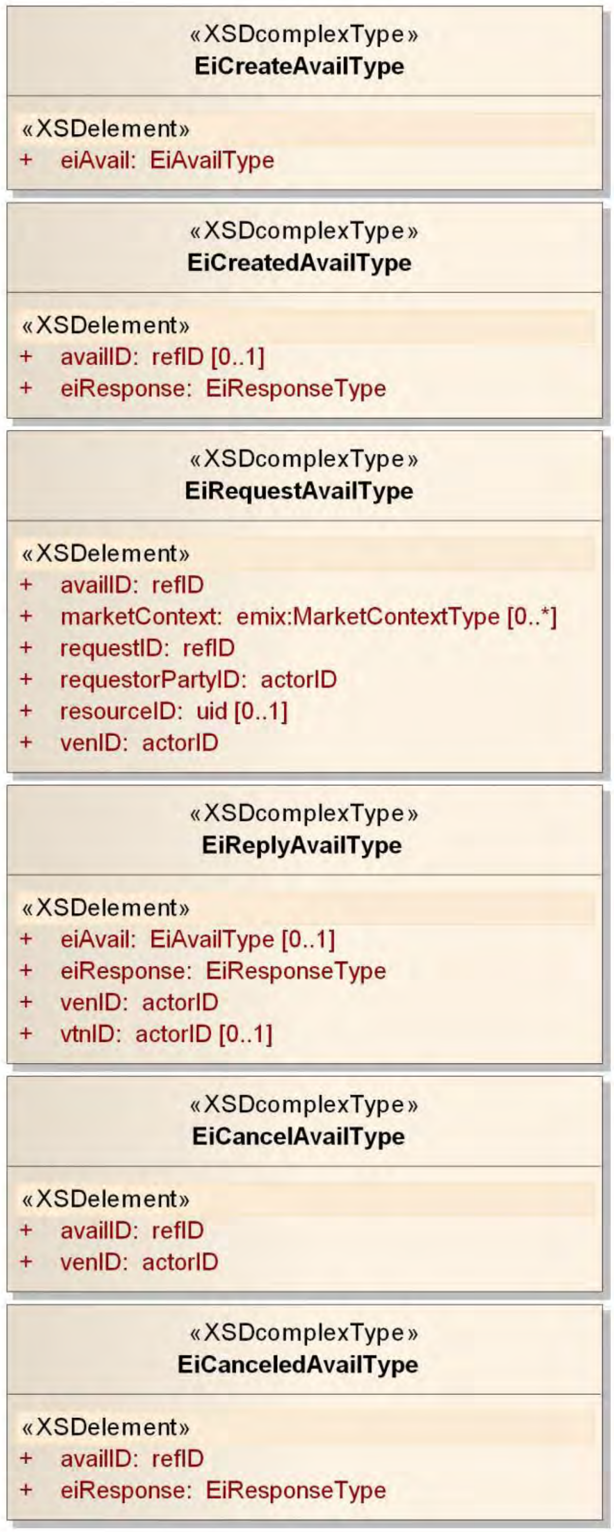


Figure 11-3: UML Class Diagram for EiAvail Service Operation Payloads

11.3 Opt Service

The Opt service creates and communicates Opt-In and Opt-Out schedules from the VEN to the VTN. Schedules are combined with EiAvailability and the Market Context requirements to give a complete picture of the willingness of the VEN to respond to EiEvents received by the VEN.

- Exactly one Vavailability MUST be provided in EiCreateOptIn and EiCreateOptOut.
- Opt schedules SHALL override any Availability in place while there is an Opt in effect. See Section 11.1

Applying EiCreateOptIn or EiCreateOptOut if an Opt is currently in effect replaces the current Opt in effect with that in the Opt Vavailability, which effectively cancels the current Opt state and Creates a new one.

Table 11-2: Opt Service

Service	Operation	Response	Service Consumer	Service Provider	Notes
EiOpt	EiCreateOpt	EiCreatedOpt	VEN	VTN	Create and send an Opt, receiving an Opt ID
EiOpt	EiRequestOpt	EiReplyOpt	VEN	VTN	Request the Opts from the VTN that are currently in effect, at most one per Market Context.
EiOpt	EiCancelOpt	EiCanceledOpt	VEN	VTN	Cancel the identified Opt

11.3.1 Interaction Patterns for the EiOpt Service

This is the **[UML]** interaction diagram for the EiOpt Service.

Figure 11-4: Interaction Diagram for the EiOpt Service NEEDS UPDATE

11.3.2 Information Model for the EiOpt Class

Opting in or out is a temporary situation indicating that the VEN will or will not respond to a particular event or in a specific time period, without changing the potentially complex Availability. The *EiOpt* schedule is a **[WS-Calendar]** VavailabilityType.

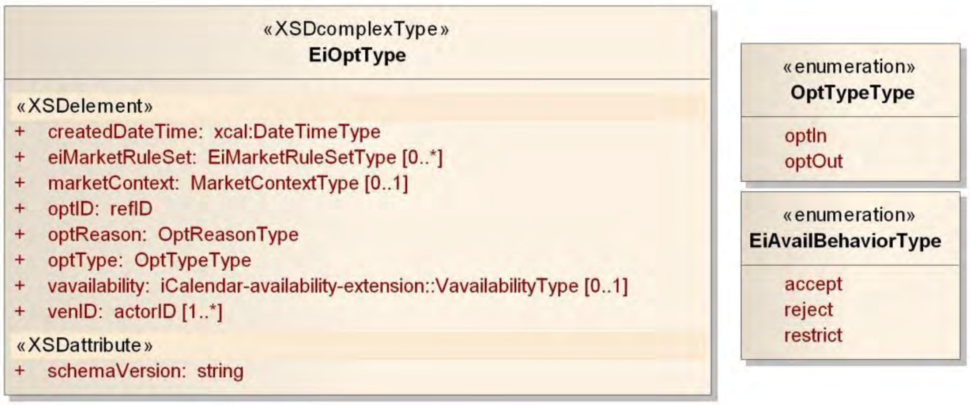


Figure 11-5: UML Class Diagram for EiOpt Type

11.3.3 Operation Payloads for the EiOpt Service

The **[UML]** class diagram describes the payloads for the EiOpt service operations.

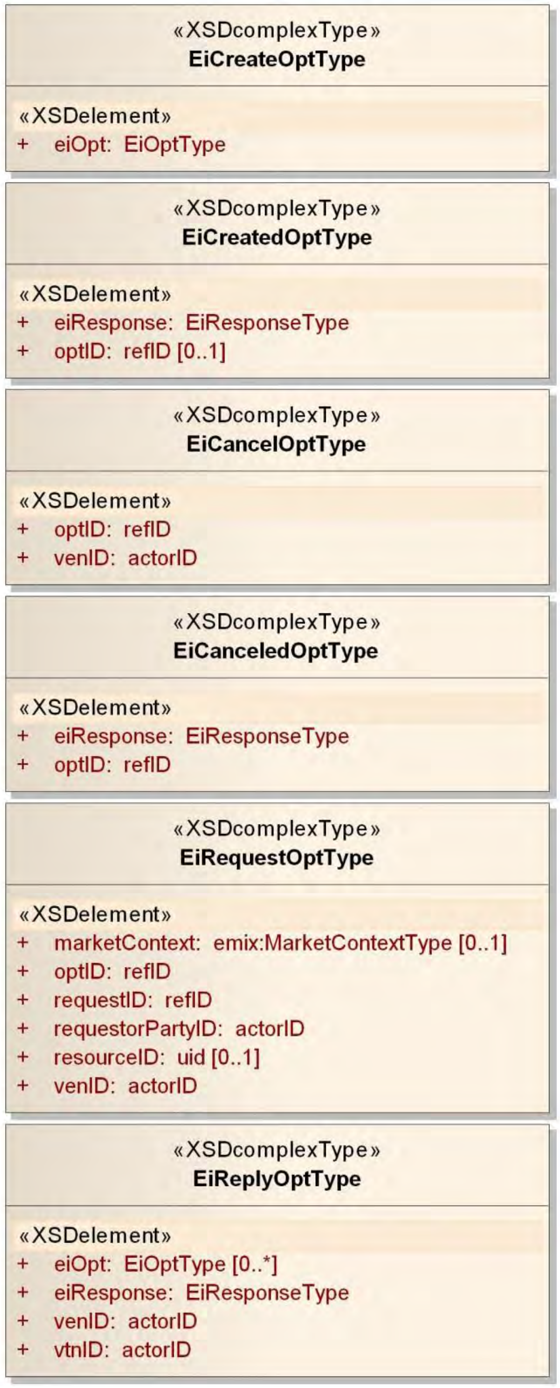


Figure 11-6: UML Class Diagram for EiOpt Service Operation Payloads

12 Market Information

Each Event and Service in Energy Interoperation takes place within a Market Context. This Context defines the behaviors that that each Party can expect from the other.

12.1 The Market Context

Market Contexts are used to express market information that rarely changes once, and thereafter not need to communicate it with each message.

In any market context, there are standing terms and expectations about product offerings. If these standing terms and expectations are not known, many exchanges may need to occur that offer products that do not meet those expectations. If those expectations are only known through local knowledge, then then national and international products need to be re-configured for each local market that they enter. If all market information were to be transmitted in every information exchange, messages based on EMIX would be overly repetitious.

As described in Section 5.2 “Market Context”, The EI Market Contexts is a super-set of the **[EMIX]** Standard Terms, and they can be referenced using the EMIX Market Context as an identifier. The EMIX Market Context is expressed as an URI.

12.2 Market Context Service

The Market Context Service enables a Party to request the details of a Market Context. These MAY be mandatory in many of today’s interactions. Parties MAY be able to request and compare Market Contexts to select which markets to participate in. Such Interactions are out of scope for this specification.

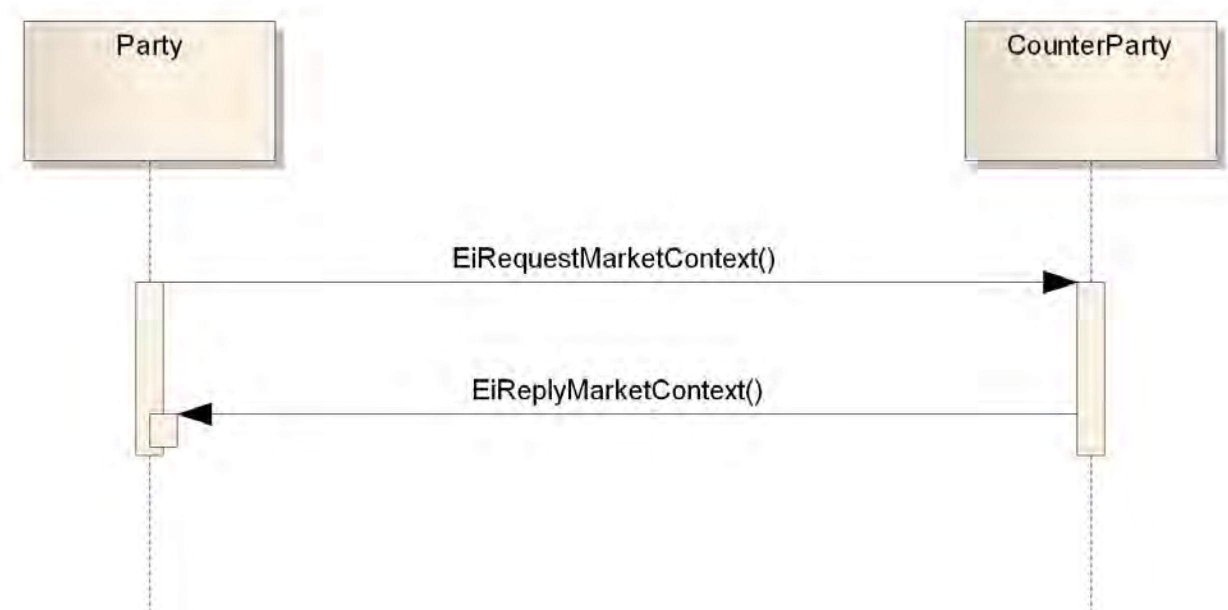


Figure 12-1: Sequence diagram for Market Context service

The Market Context service can retrieve the full information in an EiMarketContext given the identifier, an EMIX Market Context. There is one operation and a responding operation.

Table 12-1: Market Context Service

Service	Operation	Response	Service Consumer	Service Provider	Notes
EiMarketContext	EiRequest MarketContext	EiReply MarketContext	Party	Party	Respond with the full EiMarketContext for each EMIX Market Context sent; request semantics with no time Interval

12.3 UML Overview of Market Context

Figure 12-2: UML Class Diagram for Market Context

12.4 Operation Payloads for Market Context Service

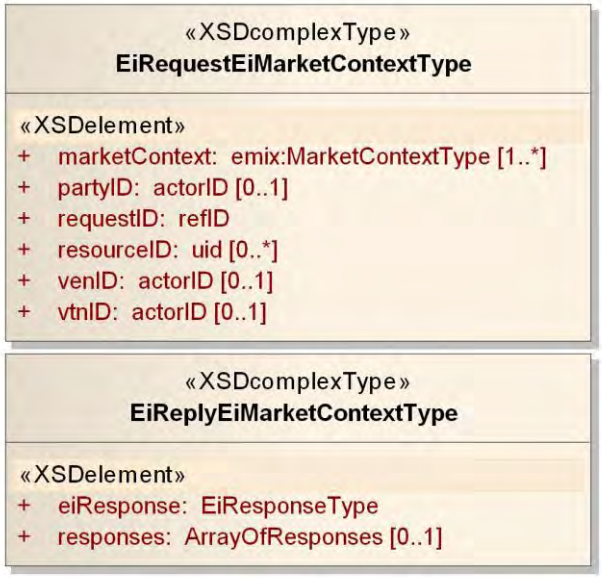


Figure 12-3: UML of Market Context Service payloads

13 Security and Composition [Non-Normative]

In this section, we describe the enterprise software approach to security and composition as applied to this Energy Interoperation specification.

Service orientation has driven a great simplification of interoperation, wherein software is no longer based on Application Programming Interfaces (APIs) but is based on exchange of information in a defined pattern of services and service operations **[SOA-RM]**.

The approach for enterprise software has evolved to defining key services and information to be exchanged, without definitively specifying how to communicate with services and how to exchange information—there are many requirements for distributed applications in many environments that cannot be taken into account in a service and information standard. To make such choices is the realm of other standards for specific areas of practice, and even there due care must be taken to avoid creating a monoculture of security.[10]

13.1 Security and Reliability Example

Different interactions require different choices for security, privacy, and reliability. Consider the following set of specifics. (We repeat the figure and re-label it.).

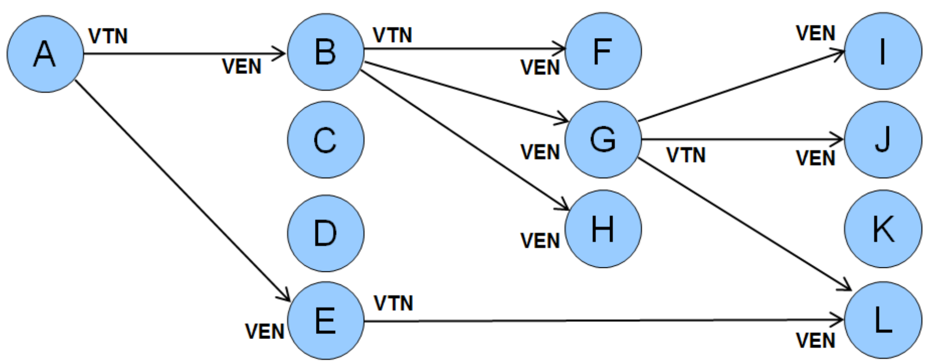


Figure 13-1: Web of Example DR Interactions

We specifically model a Reliability DR Event initiated by the Independent System Operator[11] A, who sends a reliability event to its first-level aggregators B through E. Aggregator B, in turn invokes the same service on its customers (say real estate landlords) F, G, and H.

Those customers might be industrial parks with multiple facilities, real estate developments with multiple tenants, or a company headquarters with facilities in many different geographical areas, which would invoke the same operation on their VENs.

---

[10] See e.g. the STUXNET worm effects on a monoculture of software SCADA systems, 2010. See http://en.wikipedia.org/wiki/Stuxnet

[11] Using North American Terminology.

For our example, say that G is a big-box store regional headquarters and I, J, and L are their stores in the affected area.

Each interaction will have its own security and reliability composed as needed—the requirements vary for specific interactions. For example

- For service operations between A to B, typical implementations include secure private frame-relay networks with guaranteed high reliability and known latency. In addition, rather than relying on the highly reliable network, in this case A requires an acknowledgment message from B back to A proving that the message was received.
- From the perspective of the ISO, the communication security and reliability between B and its customers F, G, and H may be purely the responsibility of B, who in order to carry out B's transaction commitments to A will arrange its business and interactions to meet B's business needs.
- G receives the signal from aggregator B. In the transaction between G and B, there are service, response, and likely security and other requirements. To meet its transactional requirements, the service operations between B and G will be implemented to satisfy the business needs of both B and G. For our example, they will use the public Internet with VPN technology and explicit acknowledgement, with a backup of pagers and phone calls in the unlikely event that the primary communication fails. And each message gets an explicit application level acknowledgement.
- Security between B and G depends on the respective security models and infrastructure supported by B and G—no one size will fit all. So that security will be used for that interaction
- The big box store chain has its own corporate security architecture and implementation, as well as reliability that meets its business needs—again, no one size will fit all, and there is tremendous variation; there is no monoculture of corporate security infrastructures.
- Store L has security, reliability, and other system design and deployment needs and implementations within the store. These may or may not be the same as the WAN connection from regional headquarters G, in fact are typically not the same (although some security aspects such as federated identity management and key distribution might be the same).
- Store L also has a relationship with aggregator E, which we will say for this example is Store L's local utility; the Public Utility Commission for the state in which L is located has mandated (in this example) that all commercial customers will use Energy Interoperation to receive certain mandated signals and price communications from the local utility. The PUC, the utility, and the owner of the store L have determined the security and reliability constraints. Once again, one size cannot fit all—and if there were one "normal" way to accommodate security and reliability, there will be a different "normal" way in different jurisdictions.

So for a simple Demand Response event distribution, we have potentially four different security profiles

The following table has sample functional names for selected nodes.

Table 13-1: Interactions and Actors for Security and Reliability Example

Label	Structure Role	Possible Actor Names
A	VTN	***System Operator***
B	VEN (wrt A), VTN (wrt F, G, H)	***Aggregator***
G	VEN (wrt B), VTN (wrt I, J, L)	***Regional Office***
L	VEN (wrt G and wrt E)	***Store***
E	VEN (wrt A, VTN wrt L)	***Local Utility***

(Note: wrt means "with respect to")

13.2 Composition

In state-of-the art software architecture, we have moved away from monolithic implementations and standards to ones that are composed of smaller parts. This allows the substitution of a functionally similar technology where needed, innovation in place, and innovation across possible solutions.

In the rich ecosystem of service and applications in use today, we *compose* or (loosely) *assemble* applications rather than craft them as one large thing. See for example OASIS Service Component Architecture **[OASIS SCA]**, which addresses the assembly, substitution, and independent evolution of components.

A typical web browser or email system uses many standards from many sources, and has evolved rapidly to accommodate new requirements by being structured to allow substitution. The set of standards (information, service, or messaging) is said to be *composed* to perform the task of delivery of email. Rather than creating a single application that does everything, perhaps in its own specific way, we can use components of code, of standards, and of protocols to achieve our goal. This is much more efficient to produce and evolve than large integrated applications such as older customized email systems.

In a similar manner, we say we *compose* the required security into the applications—say an aspect of OASIS **[WS-Security]** and OASIS Security Access Markup Language **[SAML]**—and further *compose* the required reliability, say by using OASIS **[WS-ReliableMessaging]** or perhaps the reliable messaging supported in an Enterprise Service Bus that we have deployed.

A service specification, with specific information to be exchanged, can take advantage of and be used in many different business environments without locking some in and locking some out, a great benefit to flexibility, adoption, and re-use.

13.3 Energy Interoperation and Security

In this section we describe some specific technologies and standards in our palette for building a secure and reliable implementation of Energy Interoperation. Since Energy Interoperation defines only the core information exchanges and services, and other technologies are composed in, there is no optionality related to security or reliability required or present in Energy Interoperation.

The information model in Energy Interoperation 1.0 is just that—an information model without security requirements. Each implementation must determine the security needs (outside the scope of this standard) broadly defined, including privacy (see e.g. OASIS Privacy Management Reference Model [ref]), identity (see e.g. OASIS Identity in the Cloud, OAISIS Key Management Inteoperability, OASIS Enterprise Key Management Infrastructure, OASIS Provisioning Services, OASIS Web Services Federation TC, OASIS Web Services Secure Exchange and more)

Energy Interoperation defines services together with service operations, as is now best practice in enterprise software. The message payloads are defined as information models, and include such artifacts as Energy Market Information Exchange **[EMIX]** price and product definition, tenders, and transactions, the EiEvent artifacts defined in this specification, and all information required to be exchanged for price distribution, program event distribution, demand response, and distributed energy resources.

This allows the composition and use of required interoperation standards without restriction, drawing from a palette of available standards, best practices, and technologies. The requirements to be addressed for a deployment are system issues and out of scope for this specification.

As in other software areas, if a particular approach is commonly used a separate standard (or standardized profile) may be created. In this way, WS-SecureConversation composes WS-Reliability and WS-Security.

So Energy Inteoperation defines the exchanged information, the services and operations, and as a matter of scope and broad use does not address any specific application as the security, privacy, performance, and reliability needs cannot be encompassed in one specification. Many of the TCs named above have produced OASIS Standards,

(SEE http://www.oasis-open.org/committees/tc_cat.php?cat=security)

14 Profiles [Normative]

These sections define the three normative profiles that are part of Energy Interoperation 1.0.

A profile includes a selection of interfaces, services, and options for a particular purpose.

14.1 OpenADR [Normative]

The OpenADR Profile defines the services required to implement functionality similar to that in **[OpenADR]**. The inclusion of the Energy Interoperation structure of VTNs and VENs, as well as use of the Energy Market Information Exchange **[EMIX]** cross-cutting price and product definition standard and WS-Calendar **[WS-Calendar]** based on the IETF **[iCalendar]** RFC updates and gives a broader range of applicability in what has been described as the *OpenADR 2 Profile*.

We present in simplified tabular form the Energy Interoperation services required as part of the OpenADR Profile. When a service is included, all of the listed operations are included, so we list only the service name and the section of this document.

Table 14-1: Services used in OpenADR Profile

Service	*Section*	*Notes*
EiRegisterParty	7.1	Register to identify and receive information
EiQuote	7.2	EiDistributeQuote for distributing dynamic prices (push), other operations for pull including block and tier tariff communication
EiEvent	9	The core event functions and information models
EiReport	10	The ability to set periodic or one-time information on the state of a Resource
EiAvail	11.2	Constraints on the possible time a Resources is available or not
EiOpt	11.3	Overrides the EiAvail; addresses short-term changes in availability
EiEnroll	8	Used to enroll a Resource for participation in Events.
EiMarketContext	12.2	Used to discover program rules, standard reports, etc.

14.2 TEMIX [Normative]

The Transactive EMIX (TEMIX) Profile defines the services required to implement functionality for energy market interactions.

We present in simplified tabular form the Energy Interoperation services required as part of the TEMIX Profile. When a service is included, all of the listed operations are required, so we list only the service name and the section of this document.

Table 14-2: Services used in TEMIX Profile

Service	*Section*	*Notes*
EiRegisterParty	7.1	Register to identify and receive information

Service	Section	Notes
EiQuote	7.2	EiDistributeQuote for distributing dynamic prices (push), other components for pull
EiTender	7.2	The basic offer of agreement is called a tender
EiTransaction	7.3	The core services to reach agreement
EiEnroll	8	Used to enroll a Resource for participation in Events.
EiMarketContext	12.2	Used to discover program rules, standard reports, etc.
EiDelivery	7.5.1	Post-Transaction delivery information

14.3 Price Distribution [Normative]

Many current initiatives envision Price Distribution as a separate Profile requiring neither transactive energy nor event-based interactions. The Price Distribution profile defines the minimal set of services required to interact with a pure Price Distribution context.

We present in simplified tabular form the Energy Interoperation services required as part of the Price Distribution Profile. When a service is included, all of the listed operations are required, so we list only the service name and the section of this document.

Table 14-3: Services used in Price Distribution Profile

Service	Section	Notes
EiRegisterParty	7.1	Register to interact with other Parties
EiQuote	7.2	EiDistributeQuote for distributing dynamic prices (push), other components for pull
EiEnroll	8	Used to enroll in a Market to receive Price Distribution.
EiMarketContext	12.2	Used to discover program rules, standard terms, etc.

15 Conformance and Processing Rules for Energy Interoperation

15.1 Conformance for Energy Interoperation

We define four conformance points for Energy Interoperation 1.0, modified by the networking technology used

- Full Conformance
- Conformance

And further define

- Conformance to a Named Profile
- Conformance with Alternate Interoperation

In this section Named Profile is one of the profiles defined in Section 14 ***"Profiles [Normative]"***.

15.1.1 General Conformance Requirements

The version of Energy Interoperation to which conformance is claimed MUST be specified in the implementation's conformance statement.

Any extension(s) used by the implementation, whether of information structures, services, service operations, or payloads MUST be described in the Implementation's conformance statement including the service operations, payloads, and information artifacts.

The phrase "support all XML artifacts" includes the support of XML artifacts as extended; similarly, message headers (SOAP Headers for Web services) MAY be extended as needed to compose other technologies including but not limited to reliability and security. The payloads defined in this specification are for required information exchanges, and a Conforming implementation MAY extend either the data types, payloads, or message headers appropriate to their transport/networking as necessary. It is required that those extensions, restrictions, and so forth be documented in the conformance statement.

15.1.2 Full Conformance to Energy Interoperation

An implementation claiming **Full Conformance to Energy Interoperation 1.0** MUST do all of the following as defined in this Work Product including specification, schemas, and WSDL files:

- Implement all services and service operations ("Services and Operations")
- Support all XML artifacts as defined in the schemas ("XML")
- Interoperate using Web services and the **[WSDL]** files ("Web Services Interoperation")
- Conform to all of Sections 13.2 through the end of Section 13 ("Additional Conformance")
- Describe how any relevant XML artifacts are derived from the Work Product

It is RECOMMENDED that interoperation be achieved using the WSI Basic Profile **[WSI-Basic]**

15.1.3 Conformance to Energy Interoperation

An implementation claiming **Conformance to Energy Interoperation 1.0** MUST do all of the following as defined in this Work Product including specification, schemas, and WSDL files:

- Interoperate using Web services and the **[WSDL]** files ("Web Services Interoperation")
- Conform to all of Sections 13.2 through the end of Section 13 ("Additional Conformance")
- Describe how any relevant XML artifacts are derived from the Work Product

In addition, if the application claiming conformance does not support one or more Services or Operations as defined in this specification, then the conformance statement for the implementation must:

- List all Services and Operations that are supported in the implementation.
- List all Services and Operations that are not supported in the implementation.
- For each Operation that is not supported, define the error response that will be returned if invoked.

For those operations that are supported by an implementation, but whose use or semantics are restricted, a conforming implementation SHALL

- List the subset of XML artifacts as defined by the schemas used in the implementation
- List the subset of XML artifacts as defined by the schemas that are not used in the specification
- State any restrictions, i.e., in cardinality or optionality, that is applied to artifacts defined herein

15.1.4 Full Conformance with Alternate Interoperation to Energy Interoperation

An implementation claiming **Full Conformance with Alternate Interoperation to Energy Interoperation 1.0** MUST be able to claim **Full Conformance to Energy Interoperation**, except that networking technologies other than Web services MAY be used by the implementation. A description of networking technologies used MUST be included in the implementation's conformance statement.

An implementation MAY claim Full Conformance as well as Full Conformance with Alternate Interoperation. The Conformance statement MUST describe the extensions or departures from Full Conformance.

15.1.5 Conformance with Alternate Interoperation to Energy Interoperation

An implementation claiming **Conformance with Alternate Interoperation to Energy Interoperation 1.0** MUST be able to claim **Conformance to Energy Interoperation**, except that networking technologies other than Web services MAY be used by the implementation. A description of networking technologies used MUST be included in the implementation's conformance statement.

An implementation MAY claim Conformance as well as Conformance with Alternate Interoperation. The Conformance statement MUST describe the extensions or departures from Full Conformance.

15.1.6 Conformance to Named Profiles of Energy Interoperation

In this section Named Profile refers to one of the profiles defined in Section 14 *"**Profiles [Normative]**"*.

15.1.6.1 Full Conformance to a Named Profile of Energy interoperation

An implementation claiming **Full Conformance to a Named Profile of Energy Interoperation** MUST be able to claim **Full Conformance to Energy Interoperation** excepting only the following:

- Services and Operations in sections not included in the named Profile as defined in Section 14 [wd35 – should be link]

It is RECOMMENDED that Web services interoperation be achieved using the WSI Basic Profile **[WSI-Basic]**

15.1.6.2 Conformance to a Named Profile of Energy interoperation

An implementation claiming **Conformance to a Named Profile of Energy Interoperation** MUST be able to claim **Conformance to Energy Interoperation** excepting only the following:

- Services and Operations in sections not included in the named Profile

It is RECOMMENDED that Web services interoperation be achieved using the WSI Basic Profile **[WSI-Basic]**

15.1.6.3 Full Conformance or Conformance with Alternate Interoperation to a Named Profile

An implementation claiming **Conformance with Alternate Interoperation** or **Full Conformance with Alternate Interoperation to a Named Profile of Energy Interoperation** MUST be able to claim the respective **Full Conformance with Alternate Interoperation** or **Conformance with Alternate Interoperation to Energy Interoperation** excepting only the following:

- Services and Operations in sections not included in the Named Profile

In addition, interoperation payloads MUST be used as defined or extended; in the event that payloads are extended a description of the extension(s) SHALL be included in the Implementation's conformance statement.

15.2 Conformance with the Semantic Models of EMIX and WS-Calendar

This section specifies conformance with the semantic models of **[EMIX]** and **[WS-Calendar]**. Energy Interoperation is strongly dependent on each of these information models.

[WS-Calendar] is a general specification and makes no assumptions about how its information model is used. **[WS-Calendar]** has specific rules which define Inheritance as a means to reduce the conveyancen of repetitive information. As this specification constrains schedule communications to specific business interactions, these inheritance rules are extended to embrace rules of interaction and rules of process that further reduce the information that must be expressed in each interval.

Implementations of Energy Interoperation SHALL conform to the rules of **[WS-Calendar]** and **[EMIX]**. These rules include the following conformance types:

- Conformance to the ***inheritance rules*** in **[WS-Calendar]**, including the direction of inheritance
- ***Specific attributes*** for each type that MUST or MUST NOT be inherited.
- ***Conformance rules*** that Referencing Specifications MUST follow
- Description of ***Covarying attributes*** with respect to the Reference Specification
- ***Semantic Conformance*** for the information within the Artifacts exchanged.
- Conformance to the ***inheritance rules*** in **[EMIX]**, including inheritance of Product Definitions and Standard Terms.

Energy Interoperation implementations also use the EMIX Products and Resources also extend the Inheritance patterns of **[WS-Calendar]** as specified in the EMIX information model. We address each of these in the following sections.

15.2.1 Recapitulation of Requirements from WS-Calendar and EMIX

[WS-Calendar] uses the term Sequence to refer to one or more Intervals with Temporal Relations defined between them that may inherit from zero or more Gluons. **[EMIX]** introduced the term Schedule to refer to Product Descriptions applied to a Sequence. Streams recapitulate these rules with specific addenda as they include both Gluon and Sequence.

15.2.1.1 Specific Attribute Inheritance within Schedules

The rules that define inheritance, including direction in **[WS-Calendar]**, are recapitulated.

I1: Proximity Rule Within a given lineage, inheritance is evaluated though each Parent to the Child before what the Child bequeaths is evaluated.

I2: Direction Rule Intervals MAY inherit attributes from the nearest Gluon subject to the Proximity Rule and Override Rule, provided those attributes are defined as Inheritable.

I3: Override Rule If and only if there is no value for a given attribute of a Gluon or Interval, that Gluon or Interval SHALL inherit the value for that attribute from its nearest Ancestor in conformance to the Proximity Rule.

I4: Comparison Rule Two Sequences are equivalent if a comparison of the respective Intervals succeeds as if each Sequence were fully Bound and redundant Gluons are removed.

I5: Designated Interval Inheritance [To facilitate composition of Sequences] the Designated Interval in the ultimate Ancestor of a Gluon is the Designated Interval of the composed Sequence. Special conformance rules for Designated Intervals apply only to the Interval linked from the Designator Gluon.

I6: Start Time Inheritance When a start time is specified through inheritance, that start time is inherited only by the Designated Interval; the start time of all other Intervals are computed through the durations and temporal; relationships within the Sequence. The Designated Interval is the Interval whose parent is at the end of the lineage. In Events, the Active Interval is the Designated Interval

15.2.1.2 Time Zone Specification

The time zone MUST be explicitly known in any conforming Energy Interoperation artifact.

This may be accomplished in two ways:

- The time, date, or date and time MUST be specified using **[ISO8601]** utc-time (also called *zulu time*)
- The **[WS-Calendar]** Time Zone Identifier, TZID, MUST be in the Lineage of the artifact, as extended by the Market Context. Generally, the Market Context acts as a Gluon bequeathing the TZID. See Section 15.3 below.

If neither expression is included, the Artifact does not conform to this specification and its attempted use in information exchanges MUST result in an error condition.

15.2.1.3 Specific Rules for Optimizing Inheritance

If the Designated Interval in a Series has a Price only, all Intervals in the Sequence have a Price only and there is no Price in the Product.

1. If the Designated Interval in a Series has a Quantity only, all Intervals in the Sequence have a Quantity only and there is no quantity in the Product.
2. If the Designated Interval in a Series has a Price & Quantity, all Intervals in the Sequence MUST have a Price and Quantity and there is neither Price not Quantity in the Product.

15.3 TEMIX Conformance

The TeMIX Profile MUST apply the conformance rules for TeMIX described in **[EMIX]**.

15.4 Inheritance within Events

For purposes of processing, inheritance, and conformance, Signal Information is treated as an **[EMIX]** Product Description, applied to a Sequence, and the Active Period is considered as an **[WS-Calendar]** Schedule. The Streams in Signals and Event-linked Reports inherit from the Active Interval as if it were a Gluon.

Signals within an Event arrive in a setting established by a Market Context. Within an event, there may be multiple Signal types. For purposes of inheritance, An Event may include multiple Stream-derived information elements each with an associated Sequence. For purposes of processing, the body of the Stream is treated as a [WS-Calendar] Gluon, and the Signal Information in each Interval in the Sequence inherits from that Gluon.

Each Specifies a Market Context. If that Market Context is associated with Standard Terms, then those Terms enter the Lineage of the Schedule and are inherited by each Interval. Standard Terms associated with a Market Context enter the Lineage of the Schedule as if the Market Context were a Gluon. Product Description, TZID, Level Definition, Terms, et al. can be inherited in this way.

15.4.1 Sequence Optimization within Events

As described in 4.3.2 “Conformance of Streams to WS-Calendar”, Signals, Reports, and Baselines MUST conform to WS-Calendar.

15.5 Version Conformance

Implementations that use the Schema Version attribute, and that claim full conformance to this specification, MAY use the use the value “1.0.2011.11” for that attribute.

Appendix A. Background and Development history

There is a significant disconnect between customer load and the value of energy. The demand is not sensitive to supply constraints; the load is not elastic; and the market fails to govern consumer behavior. In particular, poor communications concerning high costs at times of peak use cause economic loss to energy suppliers and consumers. There are today a limited number of high demand periods (roughly ten days a year, and only a portion of those days) when the failure to manage peak demand causes immense costs to the provider of energy; and, if the demand cannot be met, expensive degradations of service to the consumer of energy.

As the proportion of alternative energies on the grid rises, and more energy comes from intermittent sources, the frequency and scale of these problems will increase and there will be an increasing need for 24/7 coordination of supply and demand. In addition, new electric loads such as electric vehicles will increase the need for electricity and with new load characteristics and timing.

Energy consumers can use a variety of technologies and strategies to shift energy use to times of lower demand as well as to reduce use during peak periods. This shifting and reduction can reduce the need for new power plants, and transmission and distribution systems. These changes will reduce the overall costs of energy through greater economic efficiency. This process is known by various names, including load shaping, demand shaping, and demand response (DR). Consistent interfaces and messages for DR is a high priority cross-cutting issue identified in the NIST Smart Grid Interoperability Roadmap.

Distributed energy resources, including generation and storage, now challenge the traditional hierarchical relationship of supplier and consumer. Alternative and renewable energy sources may be located closer to the end nodes of the grid than traditional bulk generation, or even within the end nodes. Wind and solar generation, as well as industrial co-generation, allow end nodes to sometimes supply. Energy storage, including mobile storage in plug-in hybrid vehicles, means that even a device may be sometimes a supplier, sometime a customer. As these sources are all intermittent, they increase the challenge of coordinating supply and demand to maintain the reliability of the electric grid. These resource, with their associated issues, are generally named distributed energy resources (DER). The NIST Smart Grid Interoperability Roadmap, this specification, and **[EMIX]** see a continuum between DR and DER.

Better communication of energy prices addresses growing needs for lower-carbon, lower-energy buildings, net zero-energy systems, and supply-demand integration that take advantage of dynamic pricing. Local generation and local storage require that the consumer (in today's situation) make investments in technology and infrastructure including electric charging and thermal storage systems. People, buildings, businesses and the power grid will benefit from automated and timely communication of energy prices, capacity information, and other grid information.

Consistency of interface for interoperation and standardization of data communication will allow essentially the same model to work for homes, small businesses, commercial buildings, office parks, neighborhood grids, and industrial facilities, simplifying interoperation across the broad range of energy providers, distributors, and consumers, and reducing costs for implementation.

These communications will involve energy consumers, producers, transmission systems, and distribution systems. They must enable aggregation of production, consumption, and curtailment resources. These communications must support market makers, such as Independent System Operators (ISOs), utilities, and other evolving mechanisms while maintaining interoperation as the Smart Grid evolves. On the consumer side of these interfaces, building and facility agents will be able to make decisions on energy sale, purchase, and use that fit the goals and requirements of their home, business, or industrial facility.

The new symmetry of energy interactions demands symmetry of interaction. A net consumer of energy may be a producer when the sun is shining, the wind is blowing, or an industrial facility is cogenerating[12]. Each interface must support symmetry as well, with energy and economic transactions able to flow each way.

Energy Interoperation defines the market interactions between smart grids and their end nodes (Customers), including Smart Buildings and Facilities, Enterprises, Industry, Homes, and Vehicles. Market interactions are defined here to include all informational communications and to exclude direct process control communications. This document defines signals to communicate interoperable dynamic price, reliability, and emergency signals to meet business and energy needs, and scale, using a variety of communication technologies.

[12] Cogeneration refers the combined generation of multiple energy resources, i.e., a boiler that both spins a turbine to generate electricity and produces team to run an industrial process. Cogeneration can include any number of energy distributions, including heat, cold, pressure, et al.

Appendix B. Glossary

No definition in this glossary supplants normative definitions in this or other specifications. They are here merely to provide a guidepost for readers at to terms and their special uses. Implementers will want to be familiar with all referenced standards.

Agreement is broad context that incorporates market context and programs. Agreement definitions are out of scope in Energy Interoperation.

DR Resource: see Resource.

EMIX: As used in this document, EMIX objects are descriptions applied to a WS-Calendar Sequence. EMIX defines Resource capabilities, used in tenders to match capabilities to need, and in Products, used in tenders and in specific performance and execution calls.

Feedback: Information about the state of a Resource; typically in relation to planning or executing a response to an Event

Resource (as used in Energy Interoperation): a Resource is a logical entity that is dispatchable. The Resource is solely responsible for its own response. A resource description specifies the performance envelope for a Resource. If a Resource can participate in multiple markets, it may have multiple desriptions.

Resource (as defined in EMIX): A Resource is something that can describe its capabilities in a Tender into a market. How those Capabilities vary over time is defined by application of the Capability Description to a WS-Calendar Sequence. See **[EMIX]**.

Status: Information about an Event, perhaps in relation to a specific Resource.

Sequence: A set of temporally related intervals with a common relation to some informational artifact as defined in WS-Calendar. Time invariant elements are in the artifact (known as a gluon) and time-varying elements are in each interval.

Tender: A tender is an offering for a Transaction. See Transaction.

Transaction: A binding commitment between parties entered into under an agreement.

VEN – see Virtual End Node

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

Virtual Top Node (VTN): a Party that is in the role of aggregating information and capabilities of distributed energy resources. The VTN is able to communicate with both the Grid and the VEN devices or systems in its domain. A VTN may take the role of a VEN interacting with another VTN.

VTN – see Virtual Top Node

Appendix C. Extensibility in Energy Interoperation

Extensibility was a critical design constraint for Energy Interoperation. Extensibility allows the Energy Interoperation specification to be used in markets and in interactions that were not represented on the Technical Committee. Formal extensibility rules also create a set of complaint extensions for incorporation into later versions that are already compliant.

C.1 Extensibility in Enumerated values

EI defines a number of enumerations. Some of these, such as measurements of power, are predictably stable. Others, such as market contracts or energy sources, may well have new elements added. In general, these accept any string beginning with “x-“ as a legal extension. In particular, these are defined using the following mechanism in the formal schemas (XSD’s).

In ei.xsd, the extensibility pattern is defined. This pattern look like:

```
   <xs:simpleType name="EiExtensionType">
          <xs:annotation>
                 <xs:documentation>Pattern used for extending string
enumeration, where allowed</xs:documentation>
          </xs:annotation>
          <xs:restriction base="xs:string">
                 <xs:pattern value="x-\S.*"/>
          </xs:restriction>
   </xs:simpleType>
```

Non-extensible enumerated types look like this:

```
   <xs:simpleType name="VoltageUnitsType">
          <xs:restriction base="xs:string">
                 <xs:enumeration value="MV"/>
                 <xs:enumeration value="KV"/>
                 <xs:enumeration value="V"/>
          </xs:restriction>
   </xs:simpleType>
```

In this case, we use the suffix “EnumeratedType” to allow for the possibility of other Measurement Protocols that are not enumerated. Actual compliance, though, is based upon the type:

```
   <xs:simpleType name="MeasurementProtocolType">
          <xs:union memberTypes="power:MeasurementProtocolEnumeratedType
emix:EmixExtensionType"/>
   </xs:simpleType>
```

That is, valid values for the measurement protocol are the enumerated values, and any that match the extension pattern “x-*”

C.2 Extension of Structured Information Collective Items

EI anticipates adding some information structures that are more complex than simple strings can be extended as well. A challenge for these items is that they are more complicated and so require formal definition. Formal definitions, expressed as additions to schema, could require changes to the specification. Without formal definition, it is difficult for trading partners to agree on valid messages.

EI uses abstract classes for many information exchanges. For example, trading partners could agree on the exchange of additional Payloads. The existing list of Payloads are derived from the empty, abstract Payload Base Type. Parties that wish to exchange other Payloads can derive new Types from Payload Base and use them in Signals, Baselines, Reports, and Delivery.

The resulting schema, which references the approved EI schemas, but does not change them, can then be distributed to business partners to validate the resulting message exchanges.

Appendix D. Mapping NAESB Definitions to Terminology of Energy Interoperation

Energy Interoperation can be used in today's markets and business interactions. Generally accepted business terms for these markets were defined for both the retail and wholesale electrical quadrants in the ***NAESB PAP09 Requirements Phase 2*** **[NAESB PAP09].**

Because Energy Interoperation describes a general-purpose mechanism that can be used by parties for today's market interactions at several levels of today's markets as well as for new and extended future interactions, the terms do not determinatively map to the NAESB semantics. Symmetric use of the interfaces in this specification can make some mappings ambiguous.

There are several kinds of definitions used in Energy Interoperation and in EMIX.

Abstraction over a class of similar information (for example, the EMIX Interface, the *EmixInterfaceType* abstract type, addresses all locational information including geospatial, P-Node, AP-Node, and more.)

Simplification (for example, Party addresses all Business Entities as the focus is on the service interaction; a Business Entity presents and assumes various roles and interfaces)

Algebraic combination (for example, a Resource summarizes characteristics from both curtailment and generation/battery draw-down as equivalent, though the market values and markets may vary)

Some terms are outside the scope of Energy Interoperation, hence neither used nor defined (for example, Asset, Resource Object, Regulator).

With these caveats, most of the terms defined by NAESB can be mapped to those in this specification.

NOTE: Market Participant is not defined explicitly; Party is the generalization of business entities. A Party enrolls and some of the "things enrolled and is qualified in" are roles such as LSE, MA, etc…so the answer for those is "Party enrolled as …"

NAESB Term	Definition from NAESB	Energy Interoperation Term
Asset	A logical entity with measurable and reportable consumption, e.g. an Asset may be a physical device with its own meter, or the main meter at the Service Delivery Point of a Service Location.	Not used in 1.0
Asset Group	A logical entity that has a reportable interval level consumption, e.g. an Asset Group may be a physical entity with its own meter, a neighborhood of homes that has a net meter, or an estimate of consumption of an aggregation of retail customers.	Not used in 1.0
Business Entity	The wholesale or retail entity that interacts with other entities in its market.	**Party**
Communication Method	The method by which an object communicates with another object to instruct, measure, report or control.	Out of scope. Energy Interoperation defines SOA Web Services
Control	The role associated with the control of an end device.	Out of scope
Designated Dispatch Entity (DDE)	A role which carries the responsibility of receiving and processing demand resource dispatch instructions or market information and (optionally) providing response information.	Party enrolled as DDE
Distributed Energy Resources (DER)	DERs are small, modular, energy generation and storage technologies that provide electric capacity or energy where it is needed. Definition of DER provided by the Department of Energy, http://www1.eere.energy.gov/femp/pdfs/31570.pdf	**Resource**
Environmental Authority (EA)	A regulatory authority responsible for the development, reporting and enforcement of environmental activities.	Out of scope
Federal Regulator (FR)	A federal regulatory authority.	Out of scope

NAESB Term	Definition from NAESB	Energy Interoperation Term
Load-Serving Entity (LSE)	The responsible entity that secures energy and Transmission Service (and related Interconnected Operations Services) to serve the electrical demand and energy requirements of its end-use customers.	**Party** enrolled as LSE
Local Authority (LA)	A regulatory authority responsible for the oversight and administration of utility service-related functions within its jurisdiction.	Out of scope
Market Enrollment	The collection of enrollment or tariff data for a Resource Object to provide a specific market product or service.	Enrollment of a **Resource** combined with **Market Standard Terms**
Market Participant (MP)	An organization registered with the System Operator that may take on roles such as SP, LSE, TDSP, DDE, SE, and/or MA in accordance with the SO's market rules.	**Party** enrolled as an MP
Measurement	The role associated with the device or algorithm that measures the consumption or supply of an end device.	**Measurement**
Meter Authority (MA)	A role which carries the responsibility of providing data necessary to determine the performance of a Resource.	**Party** enrolled as an MA
P-Node	The price location of the Premise in the transmission and/or distribution network.	**EMIX Interface** is superclass
Participant	The entity that represents resources to a market or distribution operator.	Party
Regulator	A rule-making and enforcement entity.	Out of scope
Resource	A market-dependent group of Response Method Aggregations that represents a dispatchable entity.[13]	**EMIX Resource**

[13] This presumably is a DDE earlier in the table, as Dispatch Entity is not defined here.

NAESB Term	Definition from NAESB	Energy Interoperation Term
Resource Object	Physical and logical types of demand response resource objects.	Out of scope
Scheduling Entity(SE)	A role which carries the responsibility of submitting bids/offers and receives schedules and awards.	**Party** enrolled an SE
Service Delivery Point	The identifier of the location where electric service is delivered to the Service Location.	**EMIX Interface** is superclass
Service Location	The physical location at which connection to the transmission or distribution system is made.	**EMIX Interface** is superclass
Service Provider (SP)	A role which carries the responsibility of coordinating resources to deliver electricity products and services to a market or distribution operator.	**Party** enrolled as an SP. All roles offer services.
State Regulator (SR)	A regulatory authority responsible for the oversight and administration of electric utilities.	Out of scope
Supporting Objects	Objects that support the interaction of Business Entities and Resource Objects.	Out of scope
Transmission/Distribution Service Provider (TDSP)	A role which carries the responsibility of operating a local electricity transmission and/or distribution system.	**Party** enrolled as a TDSP
Utility Customer (UC)	An end-use customer of the Utility Distribution Operator that takes on roles such as Premise or Resource.	Not defined explicitly. **Party** may take role
Utility Distribution Operator (UDO)	An entity which carries the responsibility of operating an electricity distribution system.	Not defined explicitly. **Party** that provides transport products
Zone	A physical or electrical region.	**EMIX Interface** is the superclass

2388

Appendix E. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Hans Aanesen, Individual
Bruce Bartell, Southern California Edison
Timothy Bennett, Drummond Group Inc.
Carl Besaw, Southern California Edison
Anto Budiardjo, Clasma Events, Inc.
Edward Cazalet, Individual
Joon-Young Choi, Jeonju University
Kevin Colmer, California Independent System Operator
Toby Considine, University of North Carolina
William Cox, Individual
Sean Crimmins, California Independent System Operator
Phil Davis, Schneider Electric
Sharon Dinges, Trane
Robert Dolin, Echelon Corporation
Rik Drummond, Drummond Group Inc.
Ernst Eder, LonMark International
Thomas Ferrentino, Individual
Craig Gemmill, Tridium, Inc.
Girish Ghatikar, Lawrence Berkeley National Laboratory
Gerald Gray, Southern California Edison / Electric Power Research Institute (EPRI)
Anne Hendry, Individual
Thomas Herbst, Cisco Systems, Inc.
David Holmberg, NIST
Gale Horst, Electric Power Research Institute (EPRI)
Ali Ipakchi, Open Access Technology International Inc. (OATi)
Oliver Johnson, Tendril Networks, Inc.
Sila Kiliccote, Lawrence Berkeley National Laboratory
Ed Koch, Akuacom Inc
Michel Kohanim, Universal Devices, Inc.
Larry Lackey, TIBCO Software Inc.
Derek Lasalle, JPMorganChase
Jeremy Laundergan, Southern California Edison
Benoit Lepeuple, LonMark International
Edgardo Luzcando, Midwest ISO and ISO/RTO Council (IRC)
Carl Mattocks, Individual
Dirk Mahling, CPower
Kyle Meadors, Drummond Group Inc.
Scott Neumann, Utility Integration Solutions Inc.
Robert Old, Siemens AG
Mary Ann Piette, Lawrence Berkeley National Laboratory
Joshua Phillips, Southwest Power Pool and ISO/RTO Council (IRC)
Donna Pratt, New York ISO and ISO/RTO Council (IRC)
Ruchi Rajasekhar, Midwest Independent System Operator
Jeremy Roberts, LonMark International
Anno Scholten, Individual
Pornsak Songkakul, Siemens AG
Jane Snowdon, IBM
Aaron Snyder, NIST

William Stocker, New York ISO and ISO/RTO Council (IRC)
Pornsak Songkakul, Siemens AG
Robert Stayton, Individual
Jake Thompson, EnerNOC
Matt Wakefield, Electric Power Research Institute (EPRI)
Douglas Walker, California Independent System Operator
Evan Wallace, NIST
Dave Watson, Lawrence Berkeley National Laboratory
David Wilson, Trane
Leighton Wolffe, Individual
Brian Zink, New York Independent System Operator

The Technical Committee also acknowledges the work of the contributing groups who did so much to bring requirements and use cases to the attention of the Committee. In particular, the ISO/RTO Council task force on Demand Response, the UCAIug OpenSG Task Force on OpenADR, and the NAESB Smart Grid Task Force provided invaluable guidance and frequent feedback.

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

UCAIug OpenSG OpenADR Task Force:

Albert Chiu, Pacific Gas & Electric
Bruce Bartell, Southern California Edison
Gerald Gray, Southern California Edison

The ISO / RTO Council Smart Grid Standards Project:

We want to thank the IRC team, in particular those who directly participated in this Technical Committee:

Edgardo Luzcando, Midwest ISO and ISO/RTO Council (IRC)
Donna Pratt, New York ISO and ISO/RTO Council (IRC)
William Stocker, New York ISO and ISO/RTO Council (IRC)

The IRC team consisted of a large group of participants from ISOs and RTOs. See the IRC Smart Grid Standards web site for additional details about the project and team members - http://www.isorto.org/site/c.jhKQIZPBImE/b.6368657/k.CCDF/Smart_Grid_Project_Standards.htm

NAESB Smart Grid Standards Development Subcommittee Co-chairs:

Brent Hodges, Reliant
Robert Burke, ISO New England
Wayne Longcore, Consumers Energy
Joe Zhou, Xtensible Solutions

Appendix F. Revision History

Revision	Date	Editor	Changes Made
1.0 WD 01		Toby Considine	Initial document, largely derived from OpenADR
1.0 WD 02		Toby Considine	
1.0 WD 03		Toby Considine	
1.0 WD 04		Toby Considine	
1.0 WD 05		Toby Considine	
1.0 WD 06		Toby Considine	
1.0 WD 07		Toby Considine	
1.0 WD 08	2010-03-09	Toby Considine	Reduced core functions to two service groups, transactive energy and eliminated references to managed energy
1.0 WD 09	2010-03-23	Toby Considine	
1.0 WD 10	2010-05-11	William Cox	Updated interaction model per analysis and drawings in TC meetings in April and early May
1.0 WD 11	2010-05-18	William Cox and David Holmberg	Improved model; editorial and clarity changes. Addressed comments on interaction and service model from TC meetings in May 2010.
1.0 WD 12	2010-05-21	William Cox	Editorial and content corrections and updates. Consistency of tone; flagged portions that are more closely related to EMIX.
1.0 WD 13	2010-08-31	Toby Considine Ed Cazalet	Recast to meet new outline, Removed much of the "marketing" content or moved, for now, to appendices. Re-wrote Sections 2, 3. Created placeholders in 4, 5,6 for services definitions.
1.0 WD 14	2010-10-31	William Cox	Completed service descriptions and restructured the middle of the document. Completed the EiEvent service and included UML diagrams. Deleted no longer relevant sections.
1.0 WD 15	2010-11-15	William Cox Toby Considine	Re-wrote sections 5, 7. Re-cast and combined to divergent sections 3. Misc Jira responses

1.0 WD 16	2010-11-18	William Cox	Added missing Section 6
1.0 WD 17	2010-11-22	Toby Considine, William Cox	Responded to many comments, added Program Services, added description of Resources and EMIX and WS-Calendar (4). Added Glossary
1.0 WD 18	2010-11-24	Toby Considine	Responded to formal comments Added additional language on WS-Calendar Incorporated missing Program Call Added Simple Market Model to Interactions
1.0 WD 19	2011-02-06	Toby Considine	"Clearing the Underbrush" – numerous trivial edits from PR process
1.0 WD20	2011-03-03	Ed Cazalet, Toby Considine	Reorganization of material into new document structure
1.0 WD21	2011-03-06	Ed Cazalet, Toby Considine	Completion of reorganization (transitional material) and repair of all (I hope) links and cross-references
1.0 WD22	2011-03-07	William Cox Toby Considine	Update of UML and Services Repaired documents (links & numbering broken again)
1.0 WD23	2011-05-10	David Holmberg William Cox Toby Considine	Update to add interaction diagrams, improve text, and add sections on service operation naming, push, and pull.
1.0 WD24	2011-06-28	William Cox Toby Considine	Updates to EiEvent, EiOpt, EiAvail, EiFeedbak, EiStatus. Deleted EiProgram. Updated model, schemas, and diagrams.
1.0 WD25	2011-07-04	Toby Considine William Cox	Numerous Jira issues, new schemas, new UML,
1.0 WD26	2011-07-08	Toby Considine	No changes to Spec, updated schemas to refer to EMIX PR03
1.0 WD27	2011-08-21	Gerald Gray Ed Cazalet David Holmberg	Updated to include Interaction work by Gerald Gray, Ed Cazalet, Appendix mapping to NAESB terms by Holmberg, Cazalet, Cox. Note that the Cazalet and Gray interaction models for Enrollment are different in approach. I have included them both for Committee discussion (Tables 7.1, 7.2).
1.0 WD28	2011-08025	Gerald Gray	Service Interactions re-written, re-titled to meet CIM expectations. All new interaction diagrams from Gray.

WD29	2011-10-10	Toby Considine	Expanded section on Composition, WS-Calendar, EMIX (4) Added section on Semantics of EI (5) Fixed broken references
WD30	2011-10-15	Toby Considine	Edits of first 5 sections for clarity, update of pictures
WD31	2011-10-17	Toby Considine William T Cox	New Section 10 Revised Reports discussion
WD32	2011-10-22	Toby Considine William T Cox Ed Koch Ed Cazalet	Re-wrote Streams and Reports for more clarity, to eliminate snaps, and to allow multiplicity. Refined Event description Defined Report Types New introduction to section 3
WD33	2011-10-28	Toby Considine William T. Cox Gerry Gray	Many niggling edits. Jira Issues as per log New Service Operation tables Updated namespaces Clean up of References Added general discussion of Requests and Responses to the intro to Services Split Reports into their own section (10) New UML, Interaction diagrams
WD34	2011-11-04	Toby Considine	Reordered section on Event Services, incorporating event Filter and Order New Figures throughout Section 3.3 added to discuss Roles and Resources Numerous small edits in response to Jira
WD35	2011-11-08	Toby Considine	Misc, Small Edits Added conformance section 1
WD36	2011-11-08	Toby Considine	Changes to Conformance Section 1 Misc formatting errors Figures 7-8, 7-10 updated