

Energy Interoperation Common Transactive Services (CTS) Version 1.0

Committee Specification Draft 01  
Working draft 04

7 May 2021

This stage:

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd04.pdf> (Authoritative)

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd04.html>

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd04.docx>

Previous stage of Version 1.0:

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd03.pdf> (Authoritative)

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd03.html>

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd03.docx>

Latest stage of Version 1.0:

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.pdf> (Authoritative)

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.html>

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.docx>

Technical Committee:

[OASIS Energy Interoperation TC](https://www.oasis-open.org/committees/energyinterop/)

Chairs:

David Holmberg ([david.holmberg@nist.gov](mailto:david.holmberg@nist.gov)), [NIST](http://www.nist.gov)

William T. Cox ([wtcox@coxsoftwarearchitects.com](mailto:wtcox@coxsoftwarearchitects.com)), Individual

Editor:

Toby Considine ([toby.considine@unc.edu](mailto:toby.considine@unc.edu)), [University of North Carolina at Chapel Hill](http://www.unc.edu/)

Additional artifacts:

This document is one component of a Work Product that also includes:

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

* UML models
* JSON schemas
* FIX Simple Binary Encoding binding (SBE)
* XML schemas

Related work:

This document replaces or supersedes:

* Common Transactive Services 1.0. The Energy Mashup Lab Specification. Edited by William T. Cox, Toby Considine 30 November 2020. <https://www.theenergymashuplab.org/s/cts-1-0-draft-20201130.pdf>

This document is related to:

* *Energy Interoperation Version 1.0*. Edited by Toby Considine, 11 June 2014. OASIS Standard. <http://docs.oasis-open.org/energyinterop/ei/v1.0/os/energyinterop-v1.0-os.html> Latest version: <http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html>. and its TeMIX Profile
* OASIS Energy Market Information Exchange (EMIX) Version 1.0 Committee Specification 02 Edited by Toby Considine, 11 January 2012. <http://docs.oasis-open.org/emix/emix/v1.0/cs02/emix-v1.0-cs02.html> Latest version: <http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html>
* OASIS WS-Calendar Platform-Independent Model version 1.0, Committee Specification 02 Edited by William T. Cox and Toby Considine, 21 August 2015. <http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/cs02/ws-calendar-pim-v1.0-cs02.html> Latest version: http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html
* OASIS WS-Calendar Schedule Signals and Streams Version 1.0 Committee Specification 01. Edited by Toby Considine and William T. Cox, 18 September 2016. <http://docs.oasis-open.org/ws-calendar/streams/v1.0/cs01/streams-v1.0-cs01.html> Latest version: <http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html>

Declared XML namespaces:

* list namespaces declared within this document (hyperlink if HTTP-based)

Abstract:

Common Transactive Services (CTS) allows actor interaction with any market. CTS is a streamlined and simplified profile of the OASIS Energy Interoperation (EI) specification, which describes an information and communication model to coordinate energy between any two Parties that consume or supply energy, such as energy suppliers and customers, markets and service providers.

Status:

This document was last revised or approved by the OASIS Energy Interoperation TC on the above date. The level of approval is also listed above. Check the "Latest stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at <https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=energyinterop#technical>.

TC members should send comments on this document to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "[Send A Comment](https://www.oasis-open.org/committees/comments/index.php?wg_abbrev=energyinterop)" button on the TC's web page at <https://www.oasis-open.org/committees/energyinterop/>.

This document is provided under the [RF on Limited Terms](https://www.oasis-open.org/policies-guidelines/ipr#RF-on-Limited-Mode) Mode of the [OASIS IPR Policy](https://www.oasis-open.org/policies-guidelines/ipr), the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this document, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (<https://www.oasis-open.org/committees/energyinterop/ipr.php>).

Note that any machine-readable content ([Computer Language Definitions](https://www.oasis-open.org/policies-guidelines/tc-process#wpComponentsCompLang)) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Key words:

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [[RFC2119](#RFC2119)] and [[RFC8174](#RFC8174)] when, and only when, they appear in all capitals, as shown here.

Citation format:

When referencing this document, the following citation format should be used:

[Energyinterop-CTS-v1.0]

*Energy Interoperation Common Transactive Services (CTS) Version 1.0*. Edited by Toby Considine. 12 February 2021. OASIS Committee Specification Draft 01. <https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-csd01.html>. Latest stage: <https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.html>.

Notices:

Copyright © OASIS Open 2021. All Rights Reserved.

Distributed under the terms of the OASIS IPR Policy, [<https://www.oasis-open.org/policies-guidelines/ipr>]. For complete copyright information please see the Notices section in the Appendix.

Table of Contents

[1 Introduction 7](#_Toc71244131)

[1.1 Application of the Common Transactive Services 7](#_Toc71244132)

[1.2 Support for Developers 8](#_Toc71244133)

[1.3 Naming Conventions 8](#_Toc71244134)

[1.4 Editing Conventions 9](#_Toc71244135)

[1.5 Architecture 9](#_Toc71244136)

[1.5.1 Security Considerations 9](#_Toc71244137)

[1.6 Semantic Composition 9](#_Toc71244138)

[1.6.1 Conformance with Energy Interoperation 10](#_Toc71244139)

[1.6.2 Conformance with EMIX 10](#_Toc71244140)

[1.6.3 Conformance with WS-Calendar Streams 10](#_Toc71244141)

[1.6.3.1 Schedule Negotiation with WS-Calendar 10](#_Toc71244142)

[1.6.3.2 Streams and Inheritance 11](#_Toc71244143)

[1.6.4 Compatibility with Facilities Smart Grid Information Model 12](#_Toc71244144)

[2 Overview of Common Transactive Services 13](#_Toc71244145)

[2.1 Scope of Common Transactive Services 13](#_Toc71244146)

[2.1.1 Applicability to Microgrids 13](#_Toc71244147)

[2.1.2 Specific scope statements 13](#_Toc71244148)

[2.2 Resources, Products and Instruments 13](#_Toc71244149)

[2.3 Common Transactive Services Architecture 15](#_Toc71244150)

[2.3.1 Message Categories 15](#_Toc71244151)

[2.3.2 Sides in Tenders and Transactions 16](#_Toc71244152)

[2.3.3 Responses 17](#_Toc71244153)

[3 Common Semantic Elements of CTS 18](#_Toc71244154)

[3.1 Semantic Elements from WS-Calendar 18](#_Toc71244155)

[3.2 Semantic Elements from EMIX 18](#_Toc71244156)

[3.2.1 Defining Resource 18](#_Toc71244157)

[3.2.2 Product Definition in CTS 19](#_Toc71244158)

[3.2.3 Market-related Elements from EMIX 20](#_Toc71244159)

[4 Services and Operations 23](#_Toc71244160)

[4.1 Structure of Common Transactive Services and Operations 23](#_Toc71244161)

[4.2 Naming of Services and Operations 23](#_Toc71244162)

[4.3 Payloads and Messages 23](#_Toc71244163)

[4.4 Description of the Services and Operations 23](#_Toc71244164)

[4.5 Responses 23](#_Toc71244165)

[5 Transactive Messages 25](#_Toc71244166)

[5.1 Pre-Transaction Services 25](#_Toc71244167)

[5.1.1 Interaction Pattern for the EiTender Service 25](#_Toc71244168)

[5.1.2 Information Model for the EiTender Services 26](#_Toc71244169)

[5.1.3 Operation Payloads for the EiTender Service 28](#_Toc71244170)

[5.2 Transaction Management Services 29](#_Toc71244171)

[ Interaction Pattern for the EiTransaction Service 30](#_Toc71244172)

[ Information Model for the EiTransaction Service 30](#_Toc71244173)

[5.2.1 Operation Payloads for the EiTransaction Service 32](#_Toc71244174)

[5.3 Comparison of Transactive Payloads 32](#_Toc71244175)

[6 Market Information 34](#_Toc71244176)

[6.1 The Market Context 34](#_Toc71244177)

[6.2 Interaction Pattern for the Market Context Service 34](#_Toc71244178)

[6.3 Information Model for the EiMarketContext Service 34](#_Toc71244179)

[6.4 Operation Payloads for the EiMarketContext Service 35](#_Toc71244180)

[7 Bindings 37](#_Toc71244181)

[7.1 JSON 37](#_Toc71244182)

[7.2 XML Schema 37](#_Toc71244183)

[7.2.1 XML Namespaces 37](#_Toc71244184)

[7.3 Simple Binary Encoding 37](#_Toc71244185)

[8 Conformance 38](#_Toc71244186)

[8.1 Claiming Conformance to Common Transactive Services 38](#_Toc71244187)

[Appendix A. References 39](#_Toc71244188)

[A.1 Normative References 39](#_Toc71244189)

[A.2 Informative References 40](#_Toc71244190)

[Appendix B. Security and Privacy Considerations 42](#_Toc71244191)

[Appendix C. Acknowledgments 43](#_Toc71244192)

[C.1 Special Thanks 43](#_Toc71244193)

[C.2 Participants 43](#_Toc71244194)

[Appendix D. Revision History 44](#_Toc71244195)

[Notices 45](#_Toc71244196)

Table of Tables

[Table 1‑1: Core Semantics from WS-Calendar 13](#_Toc71292319)

[Table 2‑1: Abstract Definitions used in CTS Markets 16](#_Toc71292320)

[Table 2‑2: Transactive Message Categories 18](#_Toc71292321)

[Table 2‑3: Responses 19](#_Toc71292322)

[Table 3‑1: CTS Elements from WS-Calendar 20](#_Toc71292323)

[Table 3‑2 Defining the Resource 21](#_Toc71292324)

[Table 3‑3 Defining the Product 21](#_Toc71292325)

[Table 3‑4 Market-related elements from EMIX 22](#_Toc71292326)

[Table 3‑5 Standard Terms that define market interactions 22](#_Toc71292327)

[Table 5‑1: Pre-Transaction Tender Services 27](#_Toc71292328)

[Table 5‑2: EiResponse Attributes 29](#_Toc71292329)

[Table 5‑3: Transaction Management Service 32](#_Toc71292330)

[Table 5‑4: EiTransaction Attributes 33](#_Toc71292331)

Table of Figures

[Figure 1‑1: Basic Power Object from EMIX 14](#_Toc71292332)

[Figure 1‑2: Applying Basic Power to a Sequence 14](#_Toc71292333)

[Figure 4‑1: Example of generic error response for a service operation 26](#_Toc71292334)

[Figure 5‑1: UML Sequence Diagram for the EiTender Service 28](#_Toc71292335)

[Figure 5‑2: Class EiTenderType 28](#_Toc71292336)

[Figure 5‑3-3 Enumeration TransactiveStateType 30](#_Toc71292337)

[Figure 5‑4: UML Class Diagram for the Operation Payloads for the EiTender Service 31](#_Toc71292338)

[Figure 5‑5: UML Sequence Diagram for the EiTransaction Service 32](#_Toc71292339)

[Figure 5‑6: UML Class Diagram of EiTransaction 33](#_Toc71292340)

[Figure 5‑7: UML Class Diagram of EiTransaction Service Operation Payloads 34](#_Toc71292341)

[Figure 5‑8: UML Diagram comparing all Transactive Payloads 35](#_Toc71292342)

[Figure 5‑6‑1: UML Sequence diagram for Market Context service 36](#_Toc71292343)

# Introduction

The Common Transactive Services (CTS) enable actor interaction with any resource market.

CTS is an application profile of OASIS Energy Interoperation 1.0 (**[EnergyInterop]**) specification, with most optionality and complexity stripped away. Energy Interoperation (abbreviated *EI*) defines the messages and services needed for transactive energy and for demand response. CTS defines a transactive energy profile specification, simplifying definitions and stripping away communication details.

Transactive energy coordinates resource supply and use between any two Parties using markets trading instruments based on time. Transactive energy applies Transactive Resource Management [TRM] to energy markets.

The initial research in TRM used a market to allocate heat from a single furnace within a commercial building. A resource is defined as a tradable commodity whose value depends on price, location, and time of delivery [EMIX]. TRM balances supply and demand over time using automated voluntary transactions between market participants.

TRM is a means to allocate resources including the delivery of commodities such as electrical energy, electrical power, natural gas, and thermal energy such as steam, hot water, or chilled water. Transactable energy resources also include the capability to deliver resources, such as transmission line capacity, flow-rate capacity.[[1]](#footnote-1), and network bandwidth.

TRM applied to energy is commonly referred to as Transactive Energy (TE), although the resource managed might be energy, power, frequency, voltage, or other characteristic. We use “Energy” and “Power” interchangeably in this specification.

Neither EI nor CTS specifies which technologies participants will use; rather CTS defines a technology-agnostic means to enable accelerated market development of such technologies. The reader can find an extended discussion of Transactive Energy in the EI specification.

CTS usesan Actor model. An Actor is an addressable computer process that can store private data, receive messages from other actors, and pass messages to other actors. An essential aspect of the Actor Model is to use a limited number of simple messages, with each message strongly typed. Actor interactions make no assumptions about the systems behind the messages. **[Actors]** are used in highly scalable architectures and highly resilient systems of systems. The Actor Model is noted for its ability to support complex emergent behaviors, including market order.

CTS is a lightweight profile of the OASIS Energy Interoperation specification that uses an Actor Model. All CTS messages are simple and make no assumptions about the systems behind the messages.

## Application of the Common Transactive Services

The purpose of this specification is to codify the common interactions and messages required for simple markets, hence for simple transactive energy markets. Any system able to use CTS should be able to interoperate with any CTS-conforming market with minimal or no change.

Systems that can be represented by CTS Actors include but are not limited to

* Smart Buildings/Homes/Industrial Facility
* Building systems/devices
* Business Enterprises
* Vehicles
* Microgrids
* Collections of IoT (Internet of Things) devices

TE demonstrations and deployments to date have been unique systems—each uses its own message model and its own market dynamics. Many early implementations required the use of central or cloud-based markets. Central markets discount local decision making while introducing new barriers to resilience. Others rely on a single price-setting supplier. None are interoperable either at the system level or for the actors involved.

CTS defines communications between market actors and does not define the market or the device controls. Autonomous market actors must be able to recognize patterns and make choices to best support their own needs. Actors need not share details of their internal operations with others.

CTS is valuable for creating micromarkets [Micromarkets] to manage power within microgrids. Micromarkets support the capability for dynamic restructuring of grids for fault resilience and efficiency **[GridFaultResilience]**. Micromarkets contain complexity by abstracting interactions to the few common messages of CTS within a bounded scope.

A device, building, market, or microgrid implementing CTS can exchange information with any other market or system using CTS, meaning that an application need not be reimplemented or tailored to different CTS-enabled markets.

CTS does not presume a market with a single seller (e.g., a utility). CTS recognizes two parties to a transaction, and the role of any Party can switch from buyer to seller from one transaction to the next. Each Resource Offer (Tender) has a Buy or Sell side attribute. We assume that when each transaction is committed (once the product has been purchased) it is owned by the purchaser, and it can be re-sold as desired or needed.

A CTS-operated micromarket may balance power over time in a traditional distribution system attached to a larger power grid or it may bind to and operate a stand-alone autonomous microgrid **[BusinessCase].**

## Support for Developers

The Common Transactive Services are defined in XML schemas **[XSD]** and described using Universal Modelling Language **[UML]**. Many software development tools can accept artifacts in UML or in XSD to enforce proper message formation.

This specification also provides **[JSON]** schemas compatible with JSON Abstract Data Notation **[JADN]** format.

The FIX Simple Binary Encoding **[SBE]** specification is used in financial markets. SBE is designed to encode and decode messages using fewer CPU instructions than standard encodings and without forcing memory management delays. SBE-based messaging is used when very high rates of message throughput are required. This specification will deliver schemas for generating SBE messages based on the common message content.

## 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 and UML models, 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 clarity in UML models the suffix “type” is not always used.

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.

JSON and where possible SBE names follow the same conventions.

## 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 UML models, and in the XML and JSON schemas.

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

Information in the **Meaning** column of the tables is normative. Information appearing in the **Notes** column is explanatory and non-normative.[[2]](#footnote-2)

Examples and Appendices are non-normative.

## Architecture

Service requests and responses are generally considered public actions of each interoperating system, with limitations to address privacy and security considerations (see Appendix B). Service actions are independent from private actions behind the interface (i.e., device control actions). A service is used without needing to know all the details of its implementation. Services are generally paid for results, not effort.

### Security Considerations

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 is free to evolve over time and to support different needs. The Common Transactive Services allow for this composition, without prescribing any particular security implementation.

### Privacy Considerations

Detailed knowledge of offers to buy or sell or of energy inputs and outputs for an actor may reveal information on actions and operations.

For example, indicating whether a production line is starting or stopping, or anticipated energy needs, or who has been buying or selling power may imply business information damaging to actors.

Similarly, an adverse party may be able to determine the likelihood that a dwelling is presently occupied.

Both security and privacy considerations are addressed in Appendix B.

## Semantic Composition

The semantics and interactions of CTS are selected from and derived from [EnergyInterop].

Energy Interoperation references two other standards, [EMIX] and [WS-Calendar], and uses an early Streams definition.

* EMIX describes price and product for electricity markets.
* WS-Calendar communicates schedules and sequences of operations. This specification uses the [Streams] optimization which is a standalone specification, rather than part of Energy Interoperation 1.0.
* Energy Interoperation uses the vocabulary and information models defined by those specifications to describe the services that it provides. The payload for each Energy Interoperation service references a product defined using [EMIX]. EMIX schedules and sequences are defined using [WS-Calendar]. Any additional schedule-related information required by [EnergyInterop] is expressed using [WS-Calendar].
* Since [EnergyInterop] was published, a semantically equivalent but simpler [Streams] specification was developed in the OASIS WS-Calendar Technical Committee . CTS uses that simpler [Streams] specification.

All terms used in this specification are as defined in their respective specifications.  
Assumptions

### Conformance with Energy Interoperation

OASIS Energy Interoperation [**EnergyInterop]** Transactive Services is the basis for CTS, which draws definitions of parties and transactive interactions from the Energy Interoperation TEMIX profile.

Energy Interop assumes an Energy Services Interface (ESI) as the external face of the energy-consuming or supplying node. Energy Interop defines an end-to-end interaction model; as does CTS.

### Conformance with EMIX

This specification uses models and artifacts simplified from 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.

The EMIX product definition is the Transactive Resource in CTS 1.0.

EMIX also defines a Market Context, often no more than a URI used as the identifier of the Market. EMIX further defines Standard Terms as retrievable information about the market that an Actor can use to configure itself for interoperation with a given market.

### Conformance with WS-Calendar Streams

The WS-Calendar specifications[[3]](#footnote-3) express sequences and enable negotiation of schedules in a manner that is semantically compatible with human schedules, i.e., [iCalendar]. A goal of the initial WS-Calendar was to create messages that were nearly identical to those used in human schedules. Later work defined an abstract Platform Independent Model (PIM) to which the initial specification conforms to. EnergyInterop defined a compact expression of WS-Calendar for remote telemetry and projections. This work was then accepted by the WS-Calendar Technical Committee as the basis for Schedule Signals and Streams [Streams], a general-purpose compact schedule expression that conforms with WS-Calendar-PIM, and thereby with WS-Calendar.

#### Schedule Negotiation with WS-Calendar

WS-Calendar considers information model for services to negotiate a schedule. Any scheduled event can be fully described by any two of three elements, when the event begins, the duration of the event, and when the event ends. With any two, the third can be computed.

Because WS-Calendar models physical processes, or services derived from physical processes. It generally constructs a schedule around Duration. “When is the best time to perform this activity for an Hour?” Schedule negotiation is the process of fully specifying when this Duration occurs in time.

A Schedule can be specified by adding either the Starting Date and Time or the Ending Date and Time to the Duration. CTS 1.0 uses the Starting date and time.

For some schedule communications, either the Date or the Time may be initially known. Consider a process that can run any day at 9:00 AM, but a date must be specified. Alternately consider a process that can run at any time on Tuesday, but requires a starting time to be scheduled.

WS-Calendar specifies rules for composing a schedule, perhaps with successive service calls. CTS uses this pattern to define instruments and schedule resource delivery.

WS-Calendar uses the terms in Table 1‑1 to describe the composition of a schedule. This specification does not redefine these terms; they are listed here solely as a convenience to the reader.

Table ‑: Core Semantics from WS-Calendar

| WS-Calendar Term | Description | |
| --- | --- | --- |
| Duration | | Duration is the length of time for an event scheduled using iCalendar or any of its derivatives.  *It is unfortunate but true that the Duration “objects” defined in many programming languages are not identical. The [XCAL] Duration is a data type using the string representation defined in the iCalendar ([RFC5545]) Duration.* |
| Interval | | An Interval has as attributes a single Duration. An Interval may be part of a Sequence. An entire Sequence can be scheduled by scheduling a single Interval in that sequence. For this reason, Intervals are defined through Duration rather than through Start or End. |
| Sequence | | A set of Intervals with defined temporal relationships. In Streams, Sequences have no gaps between intervals. A Sequence is re-locatable, i.e., it does not have a specific date and time. A Sequence may consist of a single Interval. An entire Sequence can be scheduled by scheduling a single Interval. |
| 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. |
| Payload | | The placeholder in an Interval Component that holds that thing that occurs during an Interval. In Streams, this specification refers to the Payload conveyed by an Interval. In CTS 1.0, every Interval in a Stream inherits the same Product with price and quantity varying by Interval. |
| Lineage | | The ordered set of Parents that results in a given inheritance or execution context for a Sequence. |
| 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. |

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

#### Streams and Inheritance

Streams convey sets of similar payloads with values that vary over time, i.e., it is described using a sequence of intervals. Many communications, involve information about a single interval of time. For simplicity and parsimony of expression, single Intervals are expressed as a stream with a cardinality of one.

Consider a simple Power payload as defined in **[EMIX].**



Figure ‑: Basic Power Object from EMIX

A Stream conveys repeating intervals over time, with something that changes over the course of 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.

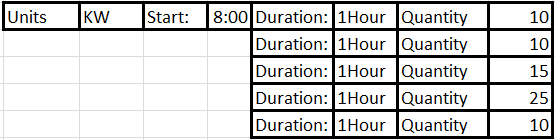
\*

Figure ‑: Applying Basic Power to a Sequence

WS-Calendar calls this pattern Inheritance and specifies a number of rules that govern Inheritance. Repeated reference to a Stream may add more information, for example a Duration to a Stream, and another reference add a Date. Elements of a Payload MAY also be inherited. A Stream is Fully Bound when all information it is payload is complete, and it has all the elements necessary for a schedule. i.e., a Duration and a Starting Date and Starting Time. This specification does not redefine these terms; they are listed here solely as a convenience to the reader.

The Stream specification extends the use of Inheritance as defined in WS-Calendar. Messages convey a Schedule, whether for Tender or g for a Contract. Each Interval in the Schedule contains an information payload. Each payload is completed through inheriting information from the Stream. The Stream itself inherits information from the context of the interaction, especially from the Market Context, as if from a Gluon.

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

For most messages, there is a cardinality of one (1), that is, only a single Interval is described in a message payload. A Market may permit messages to have a cardinality greater than one, for example, a Tender for 24 durations of one hour to express day-ahead prices. Where permitted, CTS considers these to be identical to [24] consecutive messages.

### Compatibility with Facilities Smart Grid Information Model

The Facilities Smart Grid Information Model [FSGIM] was developed to define the power capabilities and requirements of building systems over time. FSGIM addresses the so-called *built environment* and uses the semantics of WS-Calendar and EMIX to construct its information models for [power] use over time. These sequences of [power] requirements are referred to as load curves. Load curves can potentially be relocated in time, perhaps delaying or accelerating the start time to get a more advantageous price for [power].

Because FSGIM load curves use the information models of EMIX and WS-Calendarthese load curves submitted by a facility could be the basis upon which a TE Agent would base its market decisions.

The Architecture of EML-CTS is premised on distinct physical systems being able to interoperate by coordinating their production and consumption of **[power]** irrespective of their ownership, motivations, or internal mechanisms. This specification defines messages and interactions of that interoperation.

CTS tenders and transactions can be used to express FSGIM load requests. CTS 1.0 uses single-interval Streams to express single-interval tenders in anticipation of the possible use of Streams in FSGIM-conformant communications.

# Overview of Common Transactive Services

## Scope of Common Transactive Services

CTS engages Transactive Resources, e.g. Distributed Energy Resources (DER), as well as any provider or consumer of energy, while making no assumptions as to their internal processes or technology.

This specification supports agreements and transactional obligations, while offering flexibility of implementation to support specific approaches and goals of the various participants.

No particular agreements are endorsed, proposed or required in order to implement this specification. Energy market operations are beyond the scope of this specification although interactions that enable management of the actual delivery and acceptance are within scope but not included in CTS 1.0.[[4]](#footnote-4)

As shown in [CTS2016] the Common Transactive Services with suitable product definitions can be used to communicate with essentially any market.

### Applicability to Microgrids

As an extended example, using the Common Transactive Services terminology, a microgrid is comprised of a number of interacting nodes (parties). Those parties interact in a micromarket co-extensive in scope with the microgrid. No actor reveals any internal mechanisms, but only its interest in buying and selling power.

CTS can also be used for the fractal integration of microgrids. Any micromarket can be bound to or co-extensive with a node in a larger microgrid. A micromarket participating in this way exposes only its aggregate market position. Any participant in CTS effectively aggregates resources it logically contains.

Any participant in the original micromarket MAY itself represent a contained autonomous microgrid or any autonomous entity whether or not it is managed in turn by a market. **[StructuredEnergy][SmartGridBusiness]**

### Specific scope statements

Interaction patterns and service definitions to support the following are in scope for Common Transactive Services:

* Interaction patterns to support transactive energy.
* Information models for price and product communication.
* Payload definitions for Common Transactive Services

The following are out of scope for Common Transactive Services:

* Requirements specifying the type of agreement, contract, product definition, or tariff used by a particular market.
* Computations or agreements that describe how power is sold into or sold out of a marketplace.
* Communication protocols, although semantic interaction patterns are in scope.

Section 6 describes standard bindings, which may be extended by The Energy Mashup Lab or others in the future.

## Resources, Products and Instruments

Systems use the common transactive services to operate transactive resource markets. A transactive resource market balances the supply of a resource over time and the demand for that resource by using a market specifying the time of delivery.

We define a Resource as any commodity whose value is determined by time of delivery. Transactable resources include, but are not limited to, heat, natural gas, water, and sewer as well as energy. Transactive energy considers energy or power as resources. The services that support energy delivery such as transport and congestion fees are also considered transactable energy resources. The ancillary services reactive power, voltage control, and frequency control are transactable.

A Product names a transactive resource that has been “chunked” for market. These chunks define the market granularity in quantity and in time. Transactable power, for example, could be in Watts. Those Watts can be “chunked” by scale, say in kilowatts or megawatts and in multiples of a base quantity, e.g., a Product could be described as energy in 5 kiloWatt chunks. A Party intending to buy or sell needs to know what is bought or sold in this market.

Similarly, an energy market might describe the product in Watt Hours (Wh), which are similarly “chunked” in kilowatt hours.

Temporal granularity is equally important as quantity for product definition. Power delivered over an hour is a different product than the same quantity delivered over 5 minutes. One-hour markets are traded in many markets, where the energy is delivered over an hour. Some transactive energy markets in North America today have durations as brief as two seconds.

An Instrument is a Product at a specific time, for example, several units of the 5 kW for an hour product offered for delivery beginning at 3:00 PM. following keeping with financial market the thing that is bought or sold an Instrument.

A market considers all the tenders it has received offering to buy or sell an instrument, using a Matching Engine to decide which can be cleared (*satisfied*) in full or in part. The Market for the instrument for this product at 3:00 PM is logically separate Market from the trading for the similar instrument for the same product at 4:00 pm. Both Markets may be in the same Marketplace.

Just as in EMIX, the Resource is extensible; any conforming resource definition can be used to define Products that can be traded in Marketplaces based on this specification.

These level terms are summarized in Table 2‑1: .

Table ‑: Abstract Definitions used in CTS Markets

| Transactive Entity | Definition |
| --- | --- |
| Resource | A measurable commodity, substance, service, or force, whose value is determined by time of delivery |
| Product | A Resource defined by size/granularity of the Resource and by the granularity of time. Example 1: Electric Power in 10 kW units delivered over an hour of time. Example 2: Electric Energy in 1 kWh units delivered over a half hour. |
| Instrument | What is tendered in a market, i.e., a Product instantiated by a particular begin time. Example: the Product beginning at 9:00 AM on April 3. |
| Party | A Party is an Actor that buys or sells Instruments in a CTS Marketplace. A Party may be referenced by a specific role in a specific interaction, such as Party or Counter Party. |
| Market | Where Products are traded by matching tenders submitted by Actors to buy or sell an Instrument |
| Marketplace | An Actor wherein one or more Markets are conducted |
| Market Context | A URI identifying a Marketplace. In CTS, the Market Context MAY be resolvable and available so an Actor can retrieve machine-readable information describing a Marketplace. Examples of information that may be found in a Market Context include:   * A list of Products traded in this Market * Currency used for market transactions |
| Matching Engine | A computing engine to match tenders (offers to sell and to buy) using a particular algorithm. |

## Common Transactive Services Architecture

The implied CTS architecture is drawn from and is a subset and simplification of the architecture presented in [EnergyInterop]. Specifically, the Energy Interoperation architecture uses the Service-Oriented Architecture (SOA) model which has become the consensus view for energy-related interoperation. CTS refines and simplifies this to an Actor model.

The Actor Model names a style of system integration used for high scalability and resilience.[[5]](#footnote-5) The Actor Model uses a small number of simple messages to coordinate behavior among simple agents termed Actors. Note that Actors need not be actually simple; any complexity in the Actors are reduced to simple messages.

It is important to understand that an Actor may take on roles for its TE-related messages. In a Tender or Transaction, one Actor is the Party, the other is the CounterParty.[[6]](#footnote-6)

The Common Transactive Services are a lightweight profile of the OASIS Energy Interoperation specification, simplified into Actor-to-Actor messages. Each CTS message is simple and makes no assumptions about the systems behind the messages. The market receives tenders and announces contracts. Only the simple messages of CTS are used.

CTS is agnostic about how CTS messages are transported. In distinction, [EnergyInterop] specifies transport (e.g. XML-based SOAP message exchanges). CTS messages may be thought of as the information exchange in a Service-Oriented Architecture environment, with the same implied message patterns.

Just as the market participants present simple messages, so too, does the market. The internals of a market contain a Matching Engine to match tenders and to declare contracts. The rules used to match tenders could be a continuously clearing order book, or a periodic double auction, or some other model. This complexity is hidden from the Actors.

### Message Categories

All interactions described in CTS are as defined in [EnergyInterop]. That specification describes interactions between actors, and, in a deployment, relationships are established among actors. Actors may perform in pairwise chains of actors.

This specification defines interactions as messages rather than as services. The interactions are categorized in Table 2‑2. Each Category may include several messages, as in submitting a Tender, receiving a Tender, and cancelling a Tender. The payload of each Category is similar to the information in an Instrument. *“Would you like to buy …?” Would you like to sell…?*” While each message category is discussed later in detail, describes the Categories and distinguishing characteristics are discussed later in detail. We define the Categories in Table 2‑2 so the distinctions between them will be clear in those later discussions.

Table ‑: Transactive Message Types

| Types | Definition |
| --- | --- |
| Tender | A Tender is an actionable offer to buy or to sell an instrument at a given price. Tenders go to the market and are generally private. It is possible to request that a Tender be advertised to all Actors in the marketplace. |
| Transaction | A Transaction is created by the Market to memorialize a contract when a tender to buy and a tender to sell are matched. Both parties are notified of contract creation. |
| Quote | A Quote is a non-actionable indication of a potential price or availability of an instrument. Different Markets may restrict which actors may issue Quotes, say from only Market Agents or from External Actors. |
| Delivery | It is simplest to think of Delivery as a meter reading, although that meter may be virtual or computed. |
| Position | At any moment, a Party has a position which represents the cumulative amount of an Instrument that an actor has previously transacted for that time interval. For example, a Position for an Instrument reflects the algebraic sum of all quantities previously bought or sold. |

Each of these message categories has multiple messages which are described below. As sometimes one message causes another message category to originate, they ae described here to make the descriptions coherent.

### Sides in Tenders and Transactions

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

* Buy, or
* Sell

A Party selling [an Instrument] takes the Sell Side of the Transaction. A Party buying [an Instrument] takes the Buy Side of the Transaction. The offering Party is called the Party in a Transaction; the other Party is called the CounterParty

From the perspective of the market, there is no distinction between a Party selling additional power and party selling from its previously acquired position. An Actor representing a generator would generally take the Sell side of a transaction. An Actor representing a consumer generally takes the Buy side of a transaction.

However, a generator may take the Buy Side of a Transaction in order to reduce its own generation, in response either to changes in physical or market conditions or to reflect other commitments made by the actor.

A consumer may choose to sell from its current position if its plans change, or if it receives an attractive price. A power storage system actor may choose to buy or sell from interval to interval, consistent with its operating and financial goals.

We do not specify how the [Product related to the Instrument] is delivered. For example, a long-distance transfer might be implemented with the seller selling power to its local grid and the buyer buying power from its local grid, with financial reconciliation producing the same result as a direct sale and delivery.

### Responses

This section re-iterates terms and simplifies models from [EnergyInterop]. That specification is normative. The response types are common across all message categories.

Table ‑: Responses

| Attribute | Meaning |
| --- | --- |
| Request ID | A reference ID which identifies the artifact or message element to which this is a response. The Request ID uniquely identifies this request, and can serve as a messaging correlation ID[[7]](#footnote-7). |
| Response Code | The Response Code indicates success or failure of the operation requested. The Response Description is unconstrained text, perhaps for use in a user interface.  The code ranges are those used for HTTP response codes,[[8]](#footnote-8) specifically  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 |

The column labeled *Response* lists the name of the service operation payload (in Energy Interoperation and its TEMIX profile, this includes the service operation as well) invoked in response. Most operations have a response. The roles of *Service Consumer* and *Service Provider* are reversed for the *Response*.

# Common Semantic Elements of CTS

The messages of CTS use a few common elements. These elements are derived from definitions in **[WS-Calendar]**, **[EMIX]**, and in **[EnergyInterop]**.

## Semantic Elements from WS-Calendar

Time and Duration are the essential elements of defining an instrument as well as for interacting with a market.

Table ‑: CTS Elements from WS-Calendar

| **Attribute** | **Meaning** |
| --- | --- |
| Duration | Duration is used to define Products, as in “Power can be purchased and there is a one-hour (duration) market for Power”*.*  Duration is also used in Delivery to specify the period over which Delivery is measured, as in “How much Power was delivered in the 4 hours beginning with the Begin DateTime. |
| Offset | A Duration that some markets MAY use to transfer trading off of hourly boundaries. A power distribution entity may experience disruption if there is a big price change on the hour. Offset enables a market rule to trade, for example, 3 minutes after the hour. |
| Begin Date-Time | Begin Date-Time fully binds a Duration into an Interval. When applied to a Product, the Begin Date-Time defines an Instrument., i.e., something that is directly traded in the Market. |
| Expiration Date-Time | Expiration is used to limit the time a Tender is on the Market. There is an implicit expiration for every Tender equal to the Begin Date-Time of the instrument. Expiration Date-Time is needed only if the requested Expiration is prior to the Begin of the Instrument. |

## Semantic Elements from EMIX

EMIX defines what is sold in a market, when it is sold, and the price at which it is sold. In CTS, we re-factor EMIX to the Product Base, that is, what is sold, and the Product Duration, and then transact around the time and price.

### Defining Resource

Each Resource in a market must be defined in that market. A given market MAY have multiple products defined on the same resource.

Table ‑ Defining the Resource

| **Attribute** | **Meaning** |
| --- | --- |
| Resource Base | Abstract Base for all describing all Resources. This replaces what in EMIX is termed the Item, but the granularity of Resource and of Time are moved to the Product. In EMIX, the Item is built from a description of the Resource, the scale of the Resource, and the Units of the resource. The Product is similar to the Item, but it includes the Duration the item is traded for as well. |
| Item Description | The Item Description is the common name of a Product |
| Item Unit | Item Unit is the unit of measure for the Item. |
| Attributes | A list of elements that further refine the product, as in hertz and voltage |

### Product Definition in CTS

EMIX defines what is sold in a market, when it is sold, and the price at which it is sold. In CTS, we re-factor EMIX to the Product, and transact around the time and price.

Table ‑ Defining the Product

| **Attribute** | **Meaning** |
| --- | --- |
| Product Base | Abstract Base for all Describing all Products. In EMIX, the Item is built from a description of the Resource, the scale of the Resource, and the Units of the resource. The Product is similar to the Item, but it includes the Duration the item is traded for as well. |
| SI Scale | At the time EMIX was created, the Committee expected that the Open-Science or other initiative would soon create a standard code table for orders of magnitude for measurement. EMIX defines an SI Scale Code Table of its own which is used here. I recommend we replace with an integer indicating the mantissa of the Power unit. |
| Size | An integer further defining the Product Scale, i.e., the Product could be traded in units of 5 kW. |
| Attributes | A list of elements that further refine the product, as in hertz and voltage |
| Warrant | Undefined element of a product that is beyond the product definition. For example, it is possible to trade only in Neighborhood Solar Power so long as the product clears, that is sold in the same interval it is bought, |

Cautionary Note on Warrants. Market implementers should consider carefully whether they wish to support Warrants, as excessive segmentation will lead to markets that are “thinner” or “more congested”. Actors that wish to buy or sell Neighborhood Solar Power are responsible for Tender expiration or Tender Cancellation as the complexities of meeting the desires of each Actor would add considerable complexity to the market. There are additional complexities such as: Is a Battery which stores power generated by Neighborhood Solar Power selling Neighborhood Solar Power when it discharges? Alternately, if a market rule requires a Solar Panel to purchase a policy from other sources to insure its capability of Delivery, is that power considered Neighborhood Solar Power? While these issues can be resolved, each of them introduces complexity into the market and reduces the ability of the Market to generate optimum decisions. This complexity may also reduce interoperability of commodity Actors with specific Markets. Warrants are carried into CTS from EMIX to permit this complexity if desired.

### Market-related Elements from EMIX

EMIX defines vocabulary used in market messages.

Table ‑ Market-related elements from EMIX

| **Attribute** | **Meaning** |
| --- | --- |
| Party Id | The market-based ID of an actor participating in a Market, particularly the actor originating a Tender, Quote, or Contract. |
| Counter Party ID | The market-based ID of an actor participating in a Market, particularly the actor taking the other side of a contract the Party. |
| Side | An indication of what a Party intends to do in a tender or other message, i.e., “Buy” or “Sell”/ |
| Expiration Date-Time | Expiration is used to limit the time a Tender is on the Market. There is an implicit expiration for every Tender equal to the Begin Date-Time of the instrument. Expiration Date-Time is needed only if the requested Expiration is prior to the Begin of the Instrument. |
| Market Context | A URI identifying a market place. A Market Context specifies a Marketplace, the Products traded therein, and standard terms used therein. |
| Standard Terms | In EMIX, Standard Terms are the machine-readable information about a marketplace and what interactions it supports. |

EMIX does not define how Standard Terms are discovered in a Marketplace. EMIX does define how Standard Terms are expressed.

Table ‑ Standard Terms that define market interactions

| **Attribute** | **Meaning** |
| --- | --- |
| Market Context Name | Optional text providing a descriptive name for a Marketplace. |
| Currency | String indicating how value is denominated in a market. If fiat currency, should be selected from current codes maintained by UN CEFACT. May also be cryptocurrencies or local currency. Currency is one of the Standard Terms |
| Offset | A Duration that some markets MAY use to transfer trading off of hourly boundaries. A power distribution entity may experience disruption if there is a big price change on the hour. Offset could indicate, that all Intervals in a Marketplace are traded based on, for example, 3 minutes after the hour. The Offset is a Standard Term |
| Time Zone | A Time Zone indicates how all Times and Dates are expressed. The Marketplace Time Zone is a Standard Term. |
| Terms | EMIX Terms are extrinsic to the product delivery but effect how each party interacts with others. Terms may be tied to basic operational needs, or state schedules of availability, or suggest limits on bids and prices acceptable. Terms may originate with the buyer, the seller, or the market context. Simple EMIX does not define any Terms other than the Abstract base, permitting Marketplaces to use any EMIX Terms or conforming extensions. |
| Products | Standard Terms include an itemization of all Products traded in this Marketplace. |

# Services and Operations

## Structure of Common Transactive Services and Operations

The Common Transactive Services presented in this specification are only

* Transactive Services—for implementing transactions and tenders

We include UML definitions for the standard payloads for service requests, rather than the service, communication, or other characteristics. In [Section 7] we describe standard serialization for the CTS standard payloads; additional bindings may be used by conforming implementations.

## Naming of Services and Operations

The naming of services and operations follows a pattern defined in [EnergyInterop]. Services are named starting with the letters ***Ei*** following 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

For example, to construct an operation name for the EiTender service, "Ei" is concatenated with the name fragment (verb) as listed. An operation to cancel an outstanding Tender is called *EiCancelTender*.[[9]](#footnote-9)

## Payloads and Messages

We define only the payloads; the particular networking technique and message structure is determined by the applications sending and receiving CTS payloads.

## Description of the Services and Operations

The sections below provide the following for each service:

* Service description
* Table of operations
* Interaction patterns for the service operations in graphic form, using Energy Interoperation normative interactions
* Normative information model using [UML] for key artifacts used by the service
* Normative operation payloads using [UML] for each operation

## Responses

In a service interaction, responses may need to be tracked 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.

An EiResponse returns the success or failure of the entire operation, with possible detail included in responseTermsViolated (not in this release)..

It is MANDATORY to return as appropriate both errors and success in responses.[[10]](#footnote-10)

The class diagram in Figure 3-1 reflects the generic response in CTS 1.0.

The description of EiResponseType is from Energy Interoperation, changing only the cardinality of responseDescription (to zero, that is, not passed).

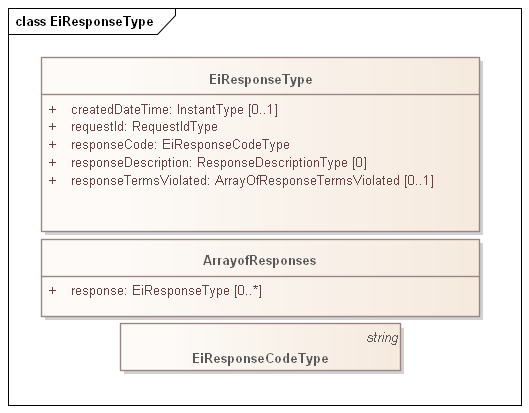


Figure ‑: Example of generic error response for a service operation

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.

While the only value after the leading digit the Response Code defined in Energy Interoperation is 00, conforming specifications may extend these codes to define more fine-grained response codes. These should extend the pattern above; for example, a response code of 403 should always be within the realm of Requester Error.

EML-CTS uses response code 200 for success.

# 

# Transactive Messages

Transactive Services in [EnergyInterop] define and support the lifecycle of transactions from initial Tender to final settlement. The phases described in [EnergyInterop] are

* Registration—to enable further phases. (Not part of CTS)
* Pre-Transaction —binding tenders for transactions. (Part of CTS)
* Transaction Services—execution and management of transactions. (Part of CTS)
* Post-Transaction—settlement, energy used or demanded, payment, position. (Not part of CTS)

For transactive services, the roles are **Parties** and **Counterparties**.

The terminology of this section is that of business agreements: tenders and transaction. The Service descriptions and payloads are simplified and updated from those defined in Energy Interoperation.

## Pre-Transaction Services

Pre-transaction services are those between parties that may prepare for a transaction. The pre-transaction services in CTS is EiTender with payloads shown in Table 5‑1.

Tenders and transactions are artifacts based on **[EMIX]** artifacts suitably flattened and simplified, and which contain schedules and prices in varying degrees of specificity or concreteness.

Table ‑: Pre-Transaction Tender Services

|  |  |  |  |
| --- | --- | --- | --- |
| Service | Request Payload | Response Payload | Notes |
| EiTender | EiCreateTenderType | EiCreatedTenderType | Create and send Tender |
| EiTender | EiCancelTenderType | EiCanceledTenderType | Cancel one or more Tenders |
| EiTender | EiDistributeTenderType | None | Distribute a list of Tenders to a transport or messaging system defined list of parties |

### Interaction Pattern for the EiTender Service

Figure 5‑1 presents the [UML] sequence diagram for the EiTender Service. Note that EiDistributeTender is not part of CTS 1.0, but is being considered for a future release.

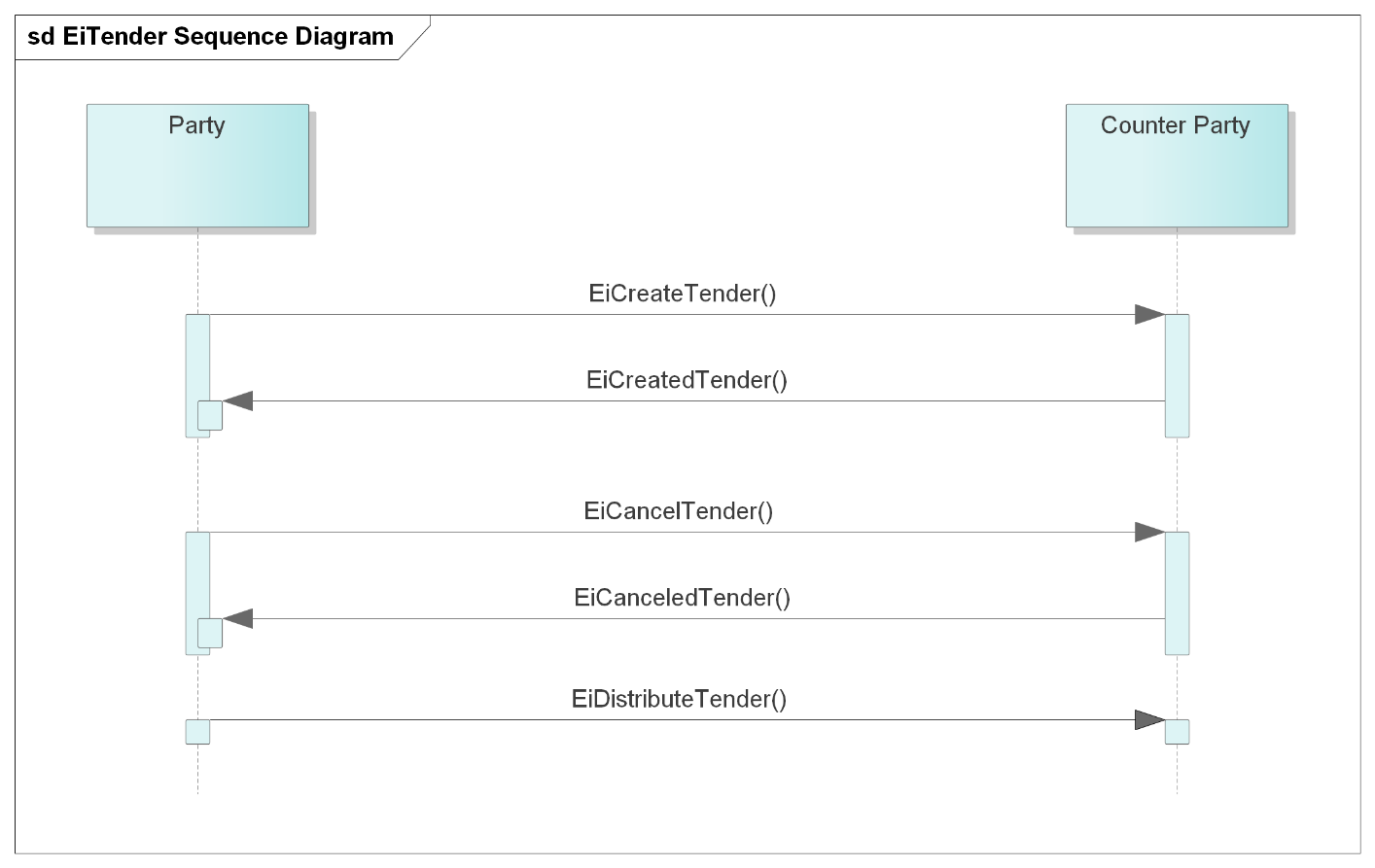


Figure ‑: UML Sequence Diagram for the EiTender Service

### Information Model for the EiTender Services

The information model for the EiTender Service artifacts follows that of **[EMIX]**, but flattened and with product definition implied by the implementation.

Time interval, price, and quantity are key elements for a product; the other aspects of product definition (e.g. energy and units) are implicit as described in Section **Error! Reference source not found.**.

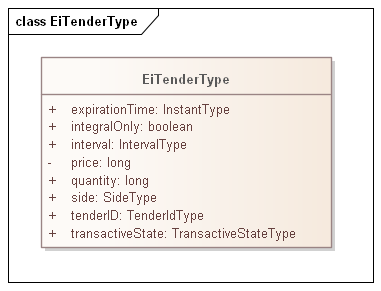


Figure ‑: Class EiTenderType

The attributes of EiTender are shown in the following table.

Table ‑: EiResponse Attributes

|  |  |  |
| --- | --- | --- |
| Attribute | Meaning | Notes |
| Expiration Time | The date and time after which this Tender is no longer valid. |  |
| Integral Only | All of the Tender must be bought or sold at once; no partial sale or purchase | In CTS set to False. Partial sale or purchase is always allowed. The attribute is present for possible future evolution. |
| Interval | The time interval for the product being offered |  |
| Price | The unit price for the product being offered | Total price is the product of price and quantity |
| Quantity | The quantity of the product being offered | Total price is the product of price and quantity |
| Side | Whether the tender is to buy or to sell the product |  |
| Tender ID | An ID for this tender |  |
| Transactive State | The transactive state of this payload (tender) | See below |

Transactive State is a concept from EMIX; it describes the state of an object. For CTS 1.0, only states *tender* and *transaction* are used.

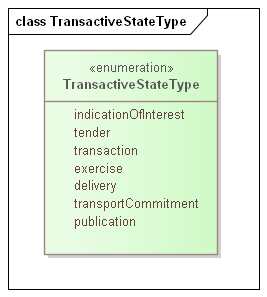


Figure ‑-3 Enumeration TransactiveStateType

### Operation Payloads for the EiTender Service

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

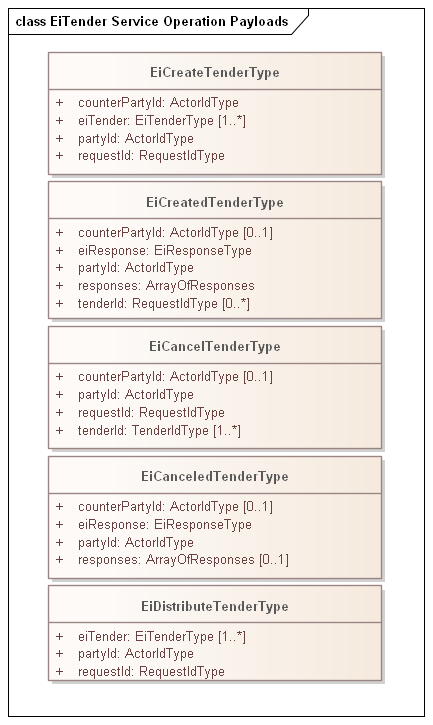


Figure ‑: UML Class Diagram for the Operation Payloads for the EiTender Service

## Transaction Management Services

This section presents the Transaction Service payloads.

market context and product are implied and may in the future be made explicit with a Market Context reference (see Section **Error! Reference source not found.**). Canceling or modifying transactions is not permitted in either CTS or Energy Interoperation. Following the approach in distributed agreement protocols[[11]](#footnote-11), compensating tenders and transactions SHOULD be created as needed to compensate for any effects.[[12]](#footnote-12)

Table ‑: Transaction Management Service

| Service | Request | Response | Notes |
| --- | --- | --- | --- |
| EiTransaction | EiCreateTransactionType | EiCreatedTransactionType | Create and send Transaction |

### Interaction Pattern for the EiTransaction Service

This is the [UML] sequence diagram for the EiTransaction Service:

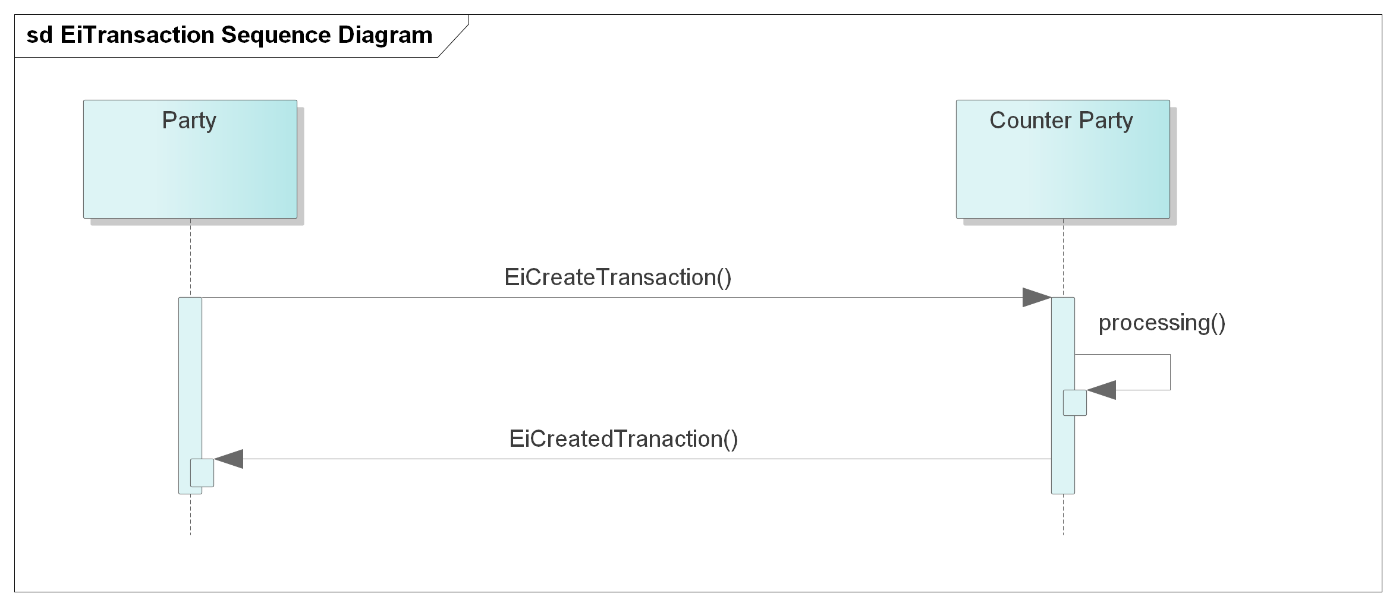


Figure ‑: UML Sequence Diagram for the EiTransaction Service

### Information Model for the EiTransaction Service

Transactions are derived from **[EMIX]** artifacts including a Stream with time, quantity, and price. Flattening similar to that in EiTender is used.

Although an EiTransaction object includes the original EiTender, the EiTransaction carries its own Transactive State.

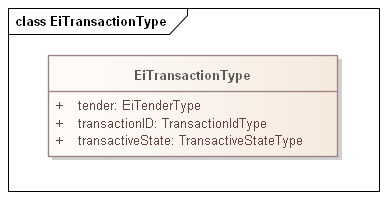


Figure ‑: UML Class Diagram of EiTransaction

The attributes of EiTransaction are shown in the following table.

Table ‑: EiTransaction Attributes

|  |  |  |
| --- | --- | --- |
| Attribute | Meaning | Notes |
| Tender | The tender (Fig. 4-2) that led to this Transaction. | The ID, quantity and price may differ from that originally tendered due to market actions. |
| Transaction ID | An ID for this Transaction | The contained Tender has its own TenderID |
| Transactive State | The transactive state of this payload is *transaction* | See Figure 5‑3-3 Enumeration TransactiveStateType |

### Operation Payloads for the EiTransaction Service

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

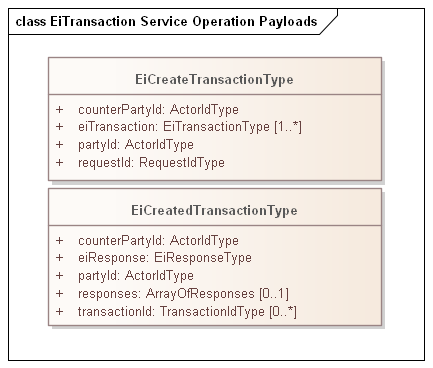
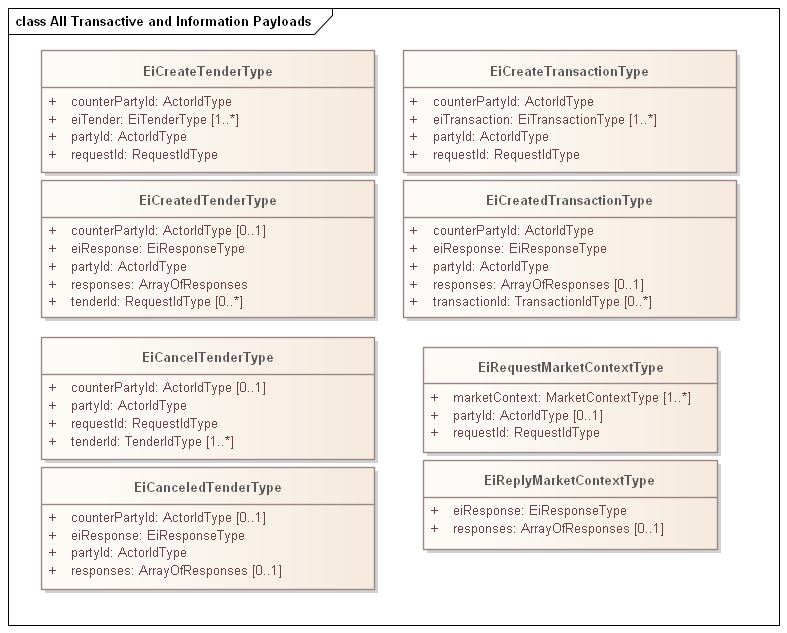


Figure ‑: UML Class Diagram of EiTransaction Service Operation Payloads

## Comparison of Transactive Payloads

Figure ‑: UML Diagram comparing all Transactive Payloads



# Market Information

Each Event and Service in Energy Interoperation takes place within a Marketplace. All interactions in a Marketplace are subject to common rules of engagement which are termed a Market Context. The Market Context defines the behaviors that that each Party can expect from the other.

This concept with some simplification is part of the Common Transactive Services.

## The Market Context

Market Contexts are resolvable URIs and are used to express market information that rarely changes, so it is not necessary 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 before finding products that meet those expectations. If these expectations are only known through local knowledge, 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 repetitive.

The Market Context for CTS is simplified from that in Energy Interoperation.

## Interaction Pattern for the Market Context Service

The Market Context Service enables a Party to request the details of a Market Context. 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.

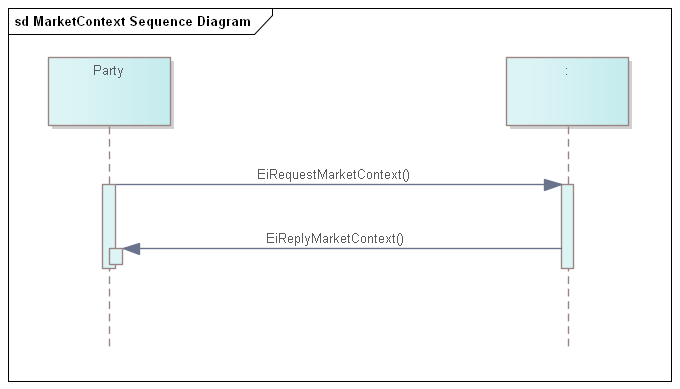


Figure 5‑‑: UML Sequence diagram for Market Context service

The Market Context service can retrieve the full information associated with an EiMarketContext. There is one operation and a responding operation.

Profiled and simplified market context information is planned for a future release.

## Information Model for the EiMarketContext Service

Simplified profile pending.

## Operation Payloads for the EiMarketContext Service

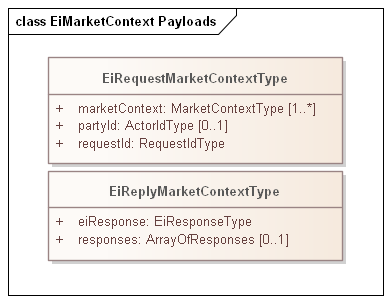


Figure 5‑2: UML of Market Context Service payloads

# Bindings

Payloads and interaction patterns are described in **[UML]** in Section  above. This section contains bindings for the payloads in three encoding schemes:

* JSON **[JSON]**
* XML Schema **[XSD]**
* FIX Simple Binary Encoding **[SBE]**

## JSON

TODO—JSON Schema available

## XML Schema

TODO—XML Schema available

### XML Namespaces

## Simple Binary Encoding

TODO—Work in progress

# Conformance

(**Note**: The [OASIS TC Process](https://www.oasis-open.org/policies-guidelines/tc-process#wpComponentsConfClause) requires that a specification approved by the TC for public review, or for publication at the Committee Specification or OASIS Standard level must include a separate section, listing a set of numbered conformance clauses, to which any implementation of the specification must adhere in order to claim conformance to the specification (or any optional portion thereof). This is done by listing the conformance clauses here.

For the definition of "conformance clause," see [OASIS Defined Terms](https://www.oasis-open.org/policies-guidelines/oasis-defined-terms-2018-05-22#dConformanceClause).

See "Guidelines to Writing Conformance Clauses":   
<https://docs.oasis-open.org/templates/TCHandbook/ConformanceGuidelines.html>.

Remove this note before submitting for publication.)

By design, CTS is a simplified and restricted subset profile of TeMIX. CTS simplifies aspects of OASIS Energy Interoperation, and omits other aspects. This section informally describes how CTS relates to the TeMIX profile. CTS is a profile of the TeMIX Profile of Energy Interoperation 1.0, described in Section 14.2 of [EnergyInterop] with the following changes:

1. Only the Payloads for Service Operation and the interaction patterns are defined.
2. The following Services from the TeMIX profile are omitted:
   1. EiQuote
   2. EiEnroll
   3. EiDelivery
3. The following Services from the TeMIX profile are included and simplified as follows.
   1. Attribute names have been made consistent with lowerCamelCase conventions.
   2. The inheritance hierarchy for UIDs and identifier types have been simplified
      1. Only selected identifier types are included
      2. The identifier types in this draft specification are opaque types rather than strings
   3. The enumeration TransactiveStateType is identical to that in Energy Interoperation, but only the following Transactive States are used:
      1. Tender
      2. Transaction
      3. Indication of Interest (pending work in progress)
   4. Market Context and the EMIX Market Context are flattened and simplified as follows:
      1. MarketContextType is a URI.
      2. Standard Terms are not profiled in this draft, but are planned to be a flattened and simplified subset of the EMIX Standard Terms.

Portions of CTS conform to and use updated and simplified versions of the specifications consumed by Energy Interoperation, specifically

* OASIS WS-Calendar [MIN]
* OASIS WS-Calendar Schedule Streams and signals [Streams]

This draft specification uses the WS-Calendar [MIN] interval directly (as IntervalType). An update in progress will instead use WS-Calendar Schedule Streams and Signals [Streams] with single interval streams. This will permit future implementations to use streams of values where appropriate..

## Claiming Conformance to Common Transactive Services

This section will describe conformance clauses for implementations claiming conformance to Common Transactive Services.

1. References

This appendix contains the normative and informative references that are used in this document. Normative references are specific (identified by date of publication and/or edition number or Version number) and Informative references may be either specific or non-specific.

While any hyperlinks included in this appendix were valid at the time of publication, OASIS cannot guarantee their long-term validity.

Note: Any normative work cited in the body of the text as needed to implement the work product must be listed in the Normative References section below. Each reference to a separate document or artifact in this work must be listed here and must be identified as either a Normative or an Informative Reference.

For all References – Normative and Informative:

Recommended approach: Set up **[Reference]** label elements as "Bookmarks", then create hyperlinks to them within the document at locations from which the references are cited. Citations in the body of the text should be hyperlinked to the appropriate Reference entry, not directly to targets which are not a part of this Work Product.

The proper format for citation of technical work produced by an OASIS TC (whether Standards Track or Non-Standards Track) is:

**[Citation Label]**

Work Product title (italicized). Edited by Albert Alston, Bob Ballston, and Calvin Carlson. Approval date (DD Month YYYY). OASIS Stage Identifier and Revision Number (e.g., OASIS Committee Specification Draft 01). Principal URI (stage-specific URI, e.g., with stage component: somespec-v1.0-csd01.html). Latest stage: (static URI, without stage identifiers, used as a symbolic link to most recently published stage of this Version).

For example:

**[OpenDoc-1.2]**

Open Document Format for Office Applications (OpenDocument) Version 1.2. Edited by Patrick Durusau and Michael Brauer. 19 January 2011. OASIS Committee Specification Draft 07. https://docs.oasis-open.org/office/v1.2/csd07/OpenDocument-v1.2-csd07.html. Latest stage: https://docs.oasis-open.org/office/v1.2/OpenDocument-v1.2.html.

Reference sources:

For references to IETF RFCs, use the approved citation formats at:

<https://docs.oasis-open.org/templates/ietf-rfc-list/ietf-rfc-list.html>.

For references to W3C Recommendations, use the approved citation formats at:

<https://docs.oasis-open.org/templates/w3c-recommendations-list/w3c-recommendations-list.html>.

Remove this note before submitting for publication.

* 1. Normative References

The following documents are referenced in such a way that some or all of their content constitutes requirements of this document.

[RFC8174]

Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<http://www.rfc-editor.org/info/rfc8174>>.

[ISO/IEC Directives] (Only used if ISO/IEC Key words are elected in front pages)

ISO/IEC Directives, Part 2 (Eighth edition) *Principles and rules for the structure and drafting of ISO and IEC documents*, International Organization for Standardization and International Electrotechnical Commission, 2018. <https://www.iso.org/sites/directives/current/part2/index.xhtml>.

[JSON]

JavaScript Object Notation and JSON Schema. <https://cswr.github.io/JsonSchema/>

[MIN]

*WS-Calendar Minimal PIM-Conformant Schema* Version 1.0. Edited by William Cox and Toby Considine. 26 August 2016. OASIS Committee Specification. <http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.html>

[RFC2119] (Only used if RFC Key words are elected in front pages)

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

**[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.

**[SBE]**

Simple Binary Encoding Technical Specification 1.0. FIX Trading Community, June 16, 2016. <https://www.fixtrading.org/standards/sbe/>

[Streams]

*Schedule Signals and Streams Version 1.0.* Edited by Toby Considine and William T. Cox. 18 September 2016. OASIS Committee Specification. <http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html>.

[WS-Calendar-PIM]

*WS-Calendar Platform Independent Model (PIM)* Version 1.0. Edited by William Cox and Toby Considine. 21 August 2015. OASIS Committee Specification. **Error! Hyperlink reference not valid.**<http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html>.

[XSD]

*W3C XML Schema Definition Language (XSD) 1.1*. Part 1: Structures, S Gao, C. M. Sperberg-McQueen, H Thompson, N Mendelsohn, D Beech, M Maloney <http://www.w3.org/TR/xmlschema11-1/>, April 2012, Part 2: Datatypes, D Peterson, S Gao, A Malhotra, C. M. Sperberg-McQueen, H Thompson, P Biron, <http://www.w3.org/TR/xmlschema11-2/> April 2012

* 1. Informative References

The following referenced documents are not required for the application of this document but may assist the reader with regard to a particular subject area.

**[Actors]**

C. Hewitt, "Actor Model of Computation: Scalable Robust Information Systems," arxiv.org, 2010.

**[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>

**[CTS2016]**

Cox, W. T., Cazalet, E., Krstulovic, A., Miller, W., & Wijbrandi, W. *Common Transactive Services*. TESC 2016. Available at <http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/Common%20Transactive%20Services%20Paper%2020160516.pdf>

**[EML-CTS]**

Energy Mashup Lab Common Transactive Services (open-source software) <https://github.com/EnergyMashupLab/eml-cts>)

**[FSGIM]**

*Facility smart grid information model*. ISO 17800. <https://www.iso.org/standard/71547.html> 2017

**[iCalendar]**

*Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, <https://tools.ietf.org/html/rfc5545>. 2009, B. Desruisseaux, See also *Calendar Availability*, <https://tools.ietf.org/html/rfc7953>, C. Daboo, M. Douglas. 2016

**[GridFaultResilience]**

William Cox and Toby Considine, *Grid Fault Recovery and Resilience: Applying Structured Energy and Microgrids*. IEEE Innovative Smart Grid Technologies 2014. Available at <http://coxsoftwarearchitects.com/Resources/ISGT_2014/ISGT2014_GridFaultRecoveryResilienceStructuredMicrogrids_Paper.pdf>

[RFC3552]

Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

**[SmartGridBusiness]**

Toby Considine and William Cox, *Smart Loads and Smart Grids—Creating the Smart Grid Business Case*. Grid-Interop 2009. Available at <http://coxsoftwarearchitects.com/Resources/Grid-Interop2009/Smart%20Loads%20and%20Smart%20Grids.pdf>

**[StructuredEnergy]**

*Structured Energy: Microgrids and Autonomous Transactive Operation*, <http://coxsoftwarearchitects.com/Resources/ISGT_2013/ISGT-Cox_StructuredEnergyPaper518.pdf> . Innovative Smart Grid Technologies 2013 (IEEE).

**[TRM]**

B. Huberman and S. H. Clearwater, *Thermal markets for controlling building environments*, Energy Engineering*,* vol. 91, no. 3, pp. 26- 56, January 1994.

**[UML]**

Object Management Group, *Unified Modeling Language (UML), V2.4.1*, August 2011. http://www.omg.org/spec/UML/2.4.1/

1. Security and Privacy Considerations

Note: OASIS strongly recommends that Technical Committees consider issues that might affect safety, security, privacy, and/or data protection in implementations of their work products and document these for implementers and adopters. For some purposes, you may find it required, e.g. if you apply for IANA registration.

While it may not be immediately obvious how your work product might make systems vulnerable to attack, most work products, because they involve communications between systems, message formats, or system settings, open potential channels for exploit. For example, IETF [[RFC3552](#RFC3552)] lists “eavesdropping, replay, message insertion, deletion, modification, and man-in-the-middle” as well as potential denial of service attacks as threats that must be considered and, if appropriate, addressed in IETF RFCs.

In addition to considering and describing foreseeable risks, this section should include guidance on how implementers and adopters can protect against these risks.

We encourage editors and TC members concerned with this subject to read Guidelines for Writing RFC Text on Security Considerations, IETF [[RFC3552](#RFC3552)], for more information.

1. Acknowledgments

This work is derived from the specification EML-CTS, contributed by The Energy Mashup Lab, written by William T. Cox and Toby Considine.

* 1. Special Thanks

Note: This is an optional subsection to call out contributions from TC members. If a TC wants to thank non-TC members then they should avoid using the term "contribution" and instead thank them for their "expertise" or "assistance".

Substantial contributions to this document from the following individuals are gratefully acknowledged:

[Participant Name, Affiliation | Individual Member]

* 1. Participants

The following individuals were members of this Technical Committee during the creation of this document and their contributions are gratefully acknowledged:

Rolf Bienert, OpenADR Alliance  
Toby Considine, University of North Carolina at Chapel Hill  
William T. Cox, Individual Member  
Pim van der Eijk, Sonnenglanz Consulting  
David Holmberg, National Institute for Standards & Technology (NIST)  
Elysa Jones, Individual   
Chuck Thomas, Electric Power Research Institute (EPRI)

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Editor** | **Changes Made** |
| WD01 | 2/15/2021 | Toby Considine | Initial reformatting and conversion of the specification contributed by The Energy Mashup Lab to create a document for committee work. |
| WD02 | 3/3/2021 | Toby Considine | Added prose definitions of Resource, Product, and Instrument |
| WD03 | 4/5/2021 | Toby Considine | Simplified introductory material, raised message type to earlier in document. Removed some repetitive material. Revised UML required. |
| WD04 | 5/7/2021 | Toby Considine David Holmberg William T Cox | Reordered intro material to reduce repetition, Reference Actor Model more consistently, Revise and re-factor Resource/Product/Instrument Add Section 3 to elevate common semantic elements |

Notices

Copyright © OASIS Open 2021. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](https://www.oasis-open.org/policies-guidelines/ipr) may be found at the OASIS website: [<https://www.oasis-open.org/policies-guidelines/ipr>].

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS DOCUMENT OR ANY PART THEREOF.

As stated in the OASIS IPR Policy, the following three paragraphs in brackets apply to OASIS Standards Final Deliverable documents (Committee Specifications, OASIS Standards, or Approved Errata).

[OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this deliverable.]

[OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this OASIS Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this OASIS Standards Final Deliverable. OASIS may include such claims on its website, but disclaims any obligation to do so.]

[OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this OASIS Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.]

The name "OASIS" is a trademark of [OASIS](https://www.oasis-open.org/), the owner and developer of this document, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, documents, while reserving the right to enforce its marks against misleading uses. Please see <https://www.oasis-open.org/policies-guidelines/trademark> for above guidance.

1. In North American wholesale electricity markets, transmission rights are bought and sold. [↑](#footnote-ref-1)
2. In ISO and IEC terminology, portions that are not normative are *informative*. OASIS uses the term *non-normative* instead. [↑](#footnote-ref-2)
3. See Section A.1 Normative References [↑](#footnote-ref-3)
4. See e.g. Energy Interoperation EiDelivery Service <https://docs.oasis-open.org/energyinterop/ei/v1.0/os/energyinterop-v1.0-os.html#_Toc388604056> [↑](#footnote-ref-4)
5. See e.g. C. Hewitt, "Actor Model of Computation: Scalable Robust Information Systems," arxiv.org, 2010, or C. Hewitt, "A Universal Modular Actor Formalism for Artificial Intelligence," ICJA, 1973. [↑](#footnote-ref-5)
6. This follows standard financial terminology. [↑](#footnote-ref-6)
7. As an example of the *Correlation Pattern* for messages [↑](#footnote-ref-7)
8. See e.g. <https://en.wikipedia.org/wiki/List_of_HTTP_status_codes> [↑](#footnote-ref-8)
9. This pattern was developed and is used by current work in the IEC Technical Committee 57 (Power Systems). [↑](#footnote-ref-9)
10. This contrasts with Energy Interoperation, where it is not mandatory to return any responses if the entire EiCancelTender service operation was completed successfully. The pattern in Energy Interoperation is to return those that have failed (required) and those that succeeded (optional). [↑](#footnote-ref-10)
11. See, e.g., WS-Transaction and WS-BusinessActivity. [↑](#footnote-ref-11)
12. This is consistent with the way that distributed agreement protocols such as [WS-BusinessActivity] manage compensation rather than cancelation. [↑](#footnote-ref-12)