Energy Interoperation Common Transactive Services (CTS) Version 1.0

Committee Specification Draft 01

Working draft 09 Work In Progress

15 August 2021

This stage:
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd07.pdf (Authoritative)
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd07.html
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd07.docx

Previous stage of Version 1.0:
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd06.pdf (Authoritative)
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd06.html
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd01/ei-cts-v1.0-wd06.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

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

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

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) permits energy consumers and producers to interact through energy markets by simplifying 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 the exchange of 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" 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 Mode of the OASIS IPR Policy, 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) 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] and [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:

Table of Contents

Table of Tables

Table of Figures

1 Introduction

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

CTS is an application profile of OASIS Energy Interoperation 1.0 (**[EI]**) specification, with most optionality and complexity stripped away, including specification of communications. While Energy Interoperation defines the messages and services for transactive energy and demand response. CTS defines messages for a transactive energy profile specification and leaving communication details unspecified (in order to permit broad semantic interoperation in multiple environments).

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

TRM is a means to allocate resources including the delivery of commodities including but not limited to electrical energy, electrical power, natural gas, and thermal energy such as steam, hot water, or chilled water. 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.

Transactable energy resources also include the capability to deliver resources, such as transmission line capacity, flow-rate capacity[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 minimal set of messages to enable interoperation through markets of participants irrespective of internal technology. In a similar manner, CTS does not specify the internal organization or operations a market for transactive energy will use, but rather a common set of messages that can be used to operate any particular transactive energy market. The goal of CTS is to enable systems and devices developed today or in the future to participate in markets deployed today or in the future. The reader can find an extended discussion of Transactive Energy (TE) in the EI specification.

CTS is a lightweight profile of the OASIS Energy Interoperation to support an actor model. An essential aspect of the actor model is to use a limited number of simple messages, with each message strongly typed. All CTS messages are simple and make no assumptions about the systems behind the messages.

1.1 Application of the Common Transactive Services

The purpose of this specification is to codify the common interactions and messages required for 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

[1] In North American wholesale electricity markets, transmission rights are bought and sold.

TE demonstrations and deployments to date have not been interoperable—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]**. CTS limits complexity by abstracting market 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 Side attribute (Buy or Sell). 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].**

1.2 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.

1.3 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.

1.4 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]

Examples and Appendices are non-normative.

1.5 Security and Privacy

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.

1.5.1 Security Considerations

Loose integration using the service-oriented architecture (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.

1.5.2 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.

1.6 Semantic Composition

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

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. CTS 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

[2] In ISO and IEC standards, portions that are not normative are *informative*. OASIS uses the term *non-normative*.

Interoperation service references a product defined using **[EMIX]**. EMIX schedules and sequences are defined using [WS-Calendar]. Any additional schedule-related information required by [EI] is expressed using [WS-Calendar].

- Since [EI] 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

1.6.1 Conformance with Energy Interoperation

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

1.6.2 Conformance with EMIX

This specification uses a simplified profile of the models and artifacts defined in 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 Market Context, 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. We extend and clarify those terms, provide an extension mechanism, and discuss the relationship of markets, marketplaces, and products.

1.6.3 Conformance with WS-Calendar Streams

WS-Calendar expresses events and sequences to support machine-to-machine (M2M) negotiation of schedules while being semantically compatible with human schedules, i.e., [iCalendar]. Schemas in [WS-Calendar] support messages that are nearly identical to those used in human schedules, a more abstract Platform Independent Model (PIM) and a compact expression, [Streams], to support remote telemetry and projections. By design and intent, those schemas provide the capability of mapping between human and M2M schedules.

WS-Calendar conveys domain specific information in a per-event payload. An essential concept of WS-Calendar is inheritance, by which a starting time can be applied to an existing message, or by which all events in a sequence can be start with the same payload. Inheritance is used to “complete” a partial message during negotiation. CTS makes use of this to apply common market product across a sequence, or to convey a specific starting time to a market product.

CTS messages conform to **[Streams]** format. See also Section 3.1.

1.6.4 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-Calendar, conforming 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 energy 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 possible future use of Streams in FSGIM-conformant communications.

2 Overview of Common Transactive Services

2.1 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.

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

2.1.1 Applicability to Microgrids (Informative)

As an extended example, using the Common Transactive Services terminology, a microgrid is comprised of interacting nodes each represented by an actor (interacting as CTS parties). Those actors 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]

2.1.2 Specific scope statements

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

- Interaction patterns to support transactive energy, including tenders, transactions, and supporting information
- Information models for price and product communication
- Information models for market characteristics
- 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 1 describes standard bindings, which may be extended by The Energy Mashup Lab or others in the future.

2.2 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.

See Section 3.2 for formal definitions.

We define a Resource as any commodity whose value is determined by time of delivery. Transactable resources include, but are not limited to, energy, heat, natural gas, water, and transport as a support service for these. The ancillary services reactive power, voltage control, and frequency control are also transactable.

A Product names a transactive resource that has been "chunked" for market. These chunks define the market granularity in quantity and in time. For example, the product may be 1 MW of power delivered over an hour. Similarly, another Product may be 1 kW of power over a 5-minute period. Some transactive energy markets in North America today have durations as brief as two seconds. Temporal granularity is equally important as quantity for product definition.

An Instrument is a Product at a specific time. For example, the 1 MW of Power delivered over an hour beginning at 3:00 PM is a different Instrument than the same Product delivered starting at 11:00 PM. We use the semantics from financial markets to name the thing that is bought or sold is 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 3:00pm instrument is traded independently from the 4:00pm instrument.

The Resource definition is extensible using standard UML techniques (subclassing), however CTS 1.0 uses only this base definition.

In future versions of CTS may permit any conforming resource definition to be used to define Products that can be traded using CTS.

These terms are summarized in Table 2-1: .

Table 2-1: 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. A market is defined by its product. 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	A Product instantiated by a particular begin time. Example: the Product beginning at 9:00 AM on April 3. An instrument is tendered to a market with specific quantity and price.
Party	A Party is an Actor that buys or sells Instruments in a CTS Marketplace. A Party may be described by a specific role in a specific interaction, such as Party or Counter Party. For semantic and privacy issues, see Party and Counterparty in Tenders and Transactions below.
Market	Where Products are traded by matching tenders submitted by Parties to buy or sell an Instrument
Marketplace	An actor wherein one or more Markets are conducted

Transactive Entity	Definition
Market Context	In EMIX, the Market Context is 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 might be associated with an EMIX Market Context include: • A list of Products traded in this Marketplace • Specific details of market operation (e.g., rules for registration and qualification, product quality, penalties for non-delivery, etc.) • Currency used for market transactions
Matching Engine	A computing engine to match tenders (offers to sell and to buy) using a particular algorithm. The structure and algorithms of a matching engine are out of scope.

2.3 Common Transactive Services Architecture

The implied CTS architecture is drawn from and is a subset and simplification of the architecture presented in [EI]. 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. The Actor Model uses a small number of simple messages to coordinate behavior among simple agents termed Actors. The Actor Model accomplishes complex behaviors through the fabric that hosts the Actors. This specification makes no assumptions about this fabric. Note that systems represented by Actors need not be actually simple; any modern facility incorporates a number of complex energy systems. This complexity is encapsulated within the Actors and the interactions 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.

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, [EI] 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.

2.3.1 Facets in CTS

Nearly all interactions implied in CTS (and described as payloads) are as defined in **[EI].** That specification defines contracts between systems as services with defined messages and interactions.

This specification describes these roles taken on by actors as *facets* for that Actor, each distinct from other roles the Actor may perform. The facets are named and briefly described in Table 2-2. Each facet includes several messages, as in submitting a Tender, acknowledging a Tender, and cancelling a Tender. Those familiar with **[EI]** will recognize that each facet is mappable to an EI service.

Each facet is discussed in detail starting in Section 5

The message or payload of each facet is similar to the information in an Instrument. *"I would like to buy…?" I would like to sell…?"*

Table 2-2: Transactive Facets Defined in CTS

Facet	Definition
Market Characteristics	A Party to potential transactions needs to know what products are traded in a Marketplace, the granularity (size and time and price), and other Marketplace information. While moving slowly over time, this can generally be viewed as static information about the Marketplace and its Products.
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 record a contract when a tender to buy and a tender to sell are matched. Both parties are notified of contract creation.
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. a Position for an Instrument reflects the algebraic sum of all quantities previously bought or sold.
Measurement and Verification (not part of this specification)	After the Product as represented by an Instrument is sold and delivered, there is typically an asynchronous verification that what was purchased was in fact what was consumed or delivered. It is simplest to think of Delivery as a meter reading, although that meter may be virtual or computed.
Market Information including Quote (not part of this specification)	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. Our base standard **[EnergyInterop]** defines the EiQuote service.

Each of these facets includes multiple messages which are described starting in Section 4. Sometimes one facet precedes the use of another facet, as Tenders may initiate messages for the Transaction Facet.

2.3.2 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.

2.3.3 Party and Counterparty in Tenders and Transactions

Which Party or Parties should be included in a Tender or Transaction payload? Who needs to know and be able to track a reference?

The Party in a Tender is offering to buy or sell.

Delegation may involve a sender (a delegate) that is not the party that is buying or selling. The *PartyID* should always reference the party that is tendering.

The Counterparty for a tender may reference either

1) The Market itself, or
2) A specific Party to which the Tender is made

The former suggests a market tender where the market will match tenders and produce? Transactions. The latter suggests a bilateral interaction not necessarily involving a market. Note that the behavior of the Actor creating a tender is the same, as the process to determine the Counterparty is not in scope.

In market interactions, the Counterparty SHOULD be the PartyID for the Market as determined by the MarketPlace. This value is accessible via the Market Characteristics Facet.

When a Transaction is created, a contract is created between the buyer and the seller.

2.3.4 Responses

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

Table 2-3: 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[3].

[3] As an example of the *Correlation Pattern* for messages

Attribute	Meaning
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,[4] 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*.

[4] See e.g. https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

3 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 **[EI]**.

3.1 Semantic Elements from WS-Calendar

Time and Duration are the essential elements of defining an instrument as well as for interacting with a market. A Stream **[Streams]** is a series of back-to-back intervals each with its own associated information.

Table 3-1: 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.

3.2 Semantic Elements from EMIX

EMIX defines what is sold in a market, when it is sold, how big the units are, and the price at which it is sold. EMIX refers to this as the Item. In CTS, we refactor this into the Resource (what is sold), the Product (how much of a Resource is sold and for how long), and the Instrument (a Product sold at a specific time).

CTS Markets consist of offers (Tenders) to buy and sell these Instruments.

3.2.1 Defining Resource

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

Table 3-2 Defining the Resource

Attribute	Meaning
Resource	Abstract base for describing all Resources. A Resource consists of a Name and a Description.

Attribute	Meaning
Item Description	The Item Description is the common name, same as in EMIX
Item Unit	Item Unit is the unit of measure for the Resource.
Attributes	Optional elements that further describe the Resource, as in `hertz and voltage`

3.2.2 Defining Product

The product completes the re-factoring of the EMIX Item, adding the size and duration to a Resource

Table 3-3 Defining the Product

Attribute	Meaning
Product	Abstract Base for all defining all Products. The core of each Product is the Resource, as described above.
Scale	Mantissa that specifies the size of the Resource Unit. For example, a Product denominated in megawatts has a mantissa of 6.
Size	An integer “chunking” the Product, i.e., the Product could be traded in units of 5 kW, a size of 5 and a scale of 3.
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,

In CTS, Products with differing Warrants are different Products. For example, if an Actor wishes to buy energy with a *Green Warrant* (however defined) then the Actor is responsible for defining its trading strategies to buy the un-warranted Product of the warranted Product is not available.

As a further application example, Actors that wish to buy or sell Neighborhood Solar Power are responsible for submitting Tenders that expire in time to make alternate arrangements, or in cancelling Tenders before fulfillment.

Market implementers should consider carefully whether they wish to support Warrants, as excessive segmentation will lead to markets that are “thinner” or “more congested”. Warrants add additional complexity of definition, i.e. such questions as “Is a Battery which stores power generated by Neighborhood Solar Power considered to be 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? This and similar questions would introduce the type of complexity that violates the design principles of CTS. Such complexity may also reduce interoperability of commodity Actors with specific Markets.

Warrants were defined in EMIX, and are permitted in CTS to support this complexity if desired.

3.2.3 Market-related Elements from EMIX

EMIX defines vocabulary used in market messages.

Table 3-4 Market-related elements from EMIX

Attribute	Meaning
PartyID	The market-based ID of an actor participating in a Market, particularly the actor originating a Tender, Quote, or Contract.
Counter PartyID	The market-based ID of an actor participating in a Market, particularly the actor taking the other side of a contract from the Party. See Section 2.3.3.
Side	An indication of what a Party offers 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 Date-Time of the Instrument.
Market Context	In EMIX, the Market Context is simply a URI to name a Marketplace, and need not be resolvable. See Section 5 for the Market Information Facet.
Standard Terms	Standard Terms are the machine-readable information about a marketplace, and the interactions it supports. In CTS, the Standard Terms include an enumeration of the Products tradable in this Marketplace.

EMIX does not define how Standard Terms are discovered in a Marketplace. The TC welcomes comments during public review as to how an Actor discovers the Standard Terms as it configures itself for a particular marketplace.

CTS Standard Terms are described in Section 5.

Table 3-5 Standard Terms that define market interactions

Attribute	Meaning
Market Context Name	Text providing a descriptive name for a Marketplace. While the Name MAY be displayed in a user interface, but it is not meaningful to the Actors.
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.
Time Offset	A Duration that some markets MAY use to describe trading off of hourly boundaries. A power distribution entity may experience disruption if there is a big price change on the hour. For example, a distribution system operator (DSO) that operates multiple CTS marketplaces could opt to set a different offset on each Marketplace operated out of a given substation. In this model, a Marketplace could use an offset duration of 3 minutes to indicate that all tenders are based on three minutes after the hour.
Time Zone	A Time Zone indicates how all Times and Dates are expressed. The Marketplace Time Zone is a Standard Term.

Attribute	Meaning
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.
Products	The Products traded in this Marketplace. Note that similar products with and without Warrants are different products, each traded in their own Market.

4 Basic Interaction and Terminology

4.1 Structure of Common Transactive Services and Operations

The Common Transactive Services presented in this specification are only

- Transactive Services—for implementing tenders and transactions
- Market Characteristics—to know what products and instruments can be traded

We include UML definitions for the standard payloads for service requests, rather than the service, communication, or other characteristics. In Section 11 we describe standard serialization for the CTS standard payloads; additional bindings may be used by conforming implementations.

4.2 Naming of Services and Operations

The naming of services and operations and service operation payloads follows the pattern defined in [EI]. 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 Tender facet, "Ei" is concatenated with the name fragment (verb) as listed. An operation to cancel an outstanding Tender is called *EiCancelTender*.[5]

Facets describe what would be called services in a full Service-Oriented Architecture implementation, as we do not define SOA services, but only imply and follow a service structure from [EI].

4.3 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.

While the payloads are logically complete with respect to the SOA interactions in [EI], the payloads may be exchanged by any means; such exchanges are below the semantic level of this specification.

4.4 Description of the Facets and Payloads

The sections below provide the following for each service:

- Facet description
- Table of Payloads
- Interaction patterns for payload exchange in graphic form, using Energy Interoperation normative interactions and UML Sequence Diagrams [UML].
- Normative information model using [UML] for key artifacts used by the facet
- Normative operation payloads using [UML] for each interaction

---

[5] This pattern was developed and is used by IEC Technical Committee 57 (Power Systems).

4.5 Responses

Responses may need to be tracked to determine whether an operation succeeds 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 (see Section 5).

It is MANDATORY to return as appropriate both errors and success in responses.[6]

The class diagram in Figure 4-1 shows the generic CTS response.

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

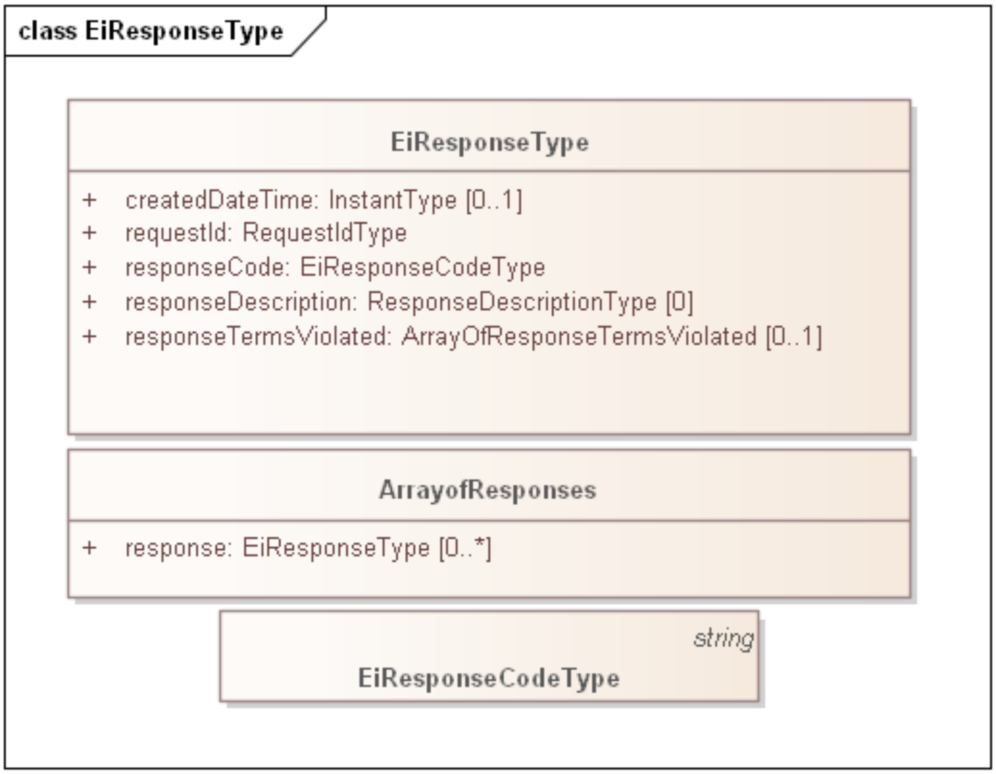


Figure 4-1: Example of generic error response for a service operation

There is no exhaustive list of all possible Response Codes. More detail on Response Codes is in Section 2.3.4.

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

[6] 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).

should extend the pattern above; for example, a response code of 403 should always indicate Requester Error. Response codes not of the form x00 MAY be treated as the parallel x00 response.

5 Market Characteristics Facet

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.

5.1 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.

Note that Market Context identifies a collection of values and behaviors; while an implementation MAY use operations such as POST to a Market Context URI, that behavior is not required.

For 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 and extended for use of standard terms.

5.2 Interaction Pattern for the Market Characteristics Facet

The Market Context Facet enables a Party to request the details of a Marketplace by using its Market Context. Where the Market Characteristics Facet is supported, Parties MAY be able to request and compare Market Contexts to select which markets to participate in.

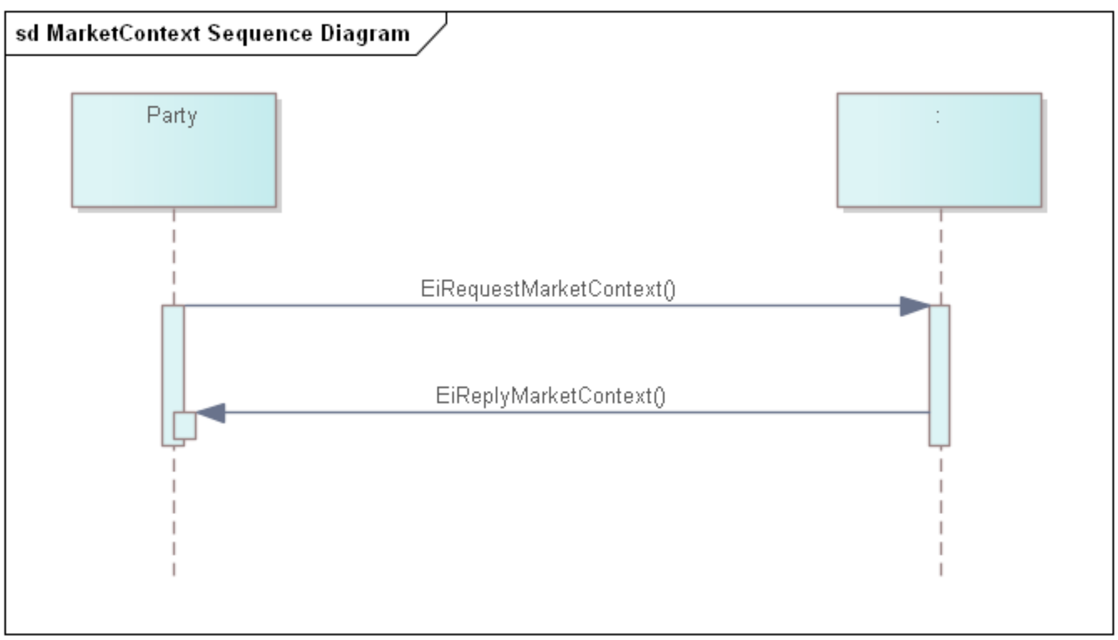


Figure 5-5-1: UML Sequence diagram for Market Context service

The Market Context service can retrieve the full information associated with an EiMarketContext. There two operations with their responding operation.

- EiRequestStandardTerms and EiReplyStandardTerms
    - The reply payload includes all standard terms as name-value pairs per Table 5-1

- EiRequestAllTerms and EiReplyStandardTerms
    - The reply payload includes all standard terms and all extended terms that may be defined for a particular Market

Profiled and simplified market context information is described in the next section.

5.3 Information Model for the Market Characteristics Facet

Simplified profile pending.

A payload referencing a market context can access standard terms as in Table 5-1 Standard Terms that define market interactions below.

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. The CTS Standard Terms MAY be extended to reflect additional capabilities and description.

Table 5-1 Standard Terms that define market interactions

Attribute	Attibute Name	Meaning
Market Context Name	NAME	Text providing a descriptive name for a Marketplace. While the Name MAY be displayed in a user interface, but it is not meaningful to the Actors.
Currency	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.
Time Offset	T_OFFSET	A Duration that some markets MAY use to describe trading off of hourly boundaries. A power distribution entity may experience disruption if there is a big price change on the hour. For example, a distribution system operator (DSO) that operates multiple CTS marketplaces could opt to set a different offset on each Marketplace operated out of a given substation. In this model, a Marketplace could use an offset duration of 3 minutes to indicate that all tenders are based on three minutes after the hour.
Time Zone	TZ	A Time Zone indicates how all Times and Dates are expressed. The Marketplace Time Zone is a Standard Term.
Time Granularity	T_GRAIN	The interval duration in seconds for the specific market
Quantity Granularity	Q_GRAIN	The allowed quantity unit size, e.g. QuantityGranularity == 10 means that a tender for 9 units will be rejected.

Attribute	Attibute Name	Meaning
Price Granularity	PRICE_GRAIN	The allowed price unit, e.g. PriceGranularity == 10 means that a price of 9 will be rejected.
Price Decimal Fraction Digits	PRICE_FRAC	Some market implementation use a market-wide indication of how many decimal fraction digits are used.[7]
Market PartyID	MPARTYID	The PartyID to use in a market tender
Bilateral OK	BILATERALOK	Boolean. True—bilateral transactions with identified parties are permitted. False—bilateral transactions not permitted, only market tenders
Marketplace Products	PRODUCTS	The Products traded in this Marketplace. Note that similar products with and without Warrants are different products, each traded in their own Market.

5.4 Operation Payloads for the Market Characteristics Facet

Payloads including terms pending. Description in Section 5.2.

[7] Integer operations are typically much more efficient than fixed or floating point, so it may be much faster to apply decimal shift on input and output rather than for more frequent comparison operations in the matching engine implementation. Note that the interaction of Price Granularity and Price Decimal Fraction Digits needs to be defined. TBD

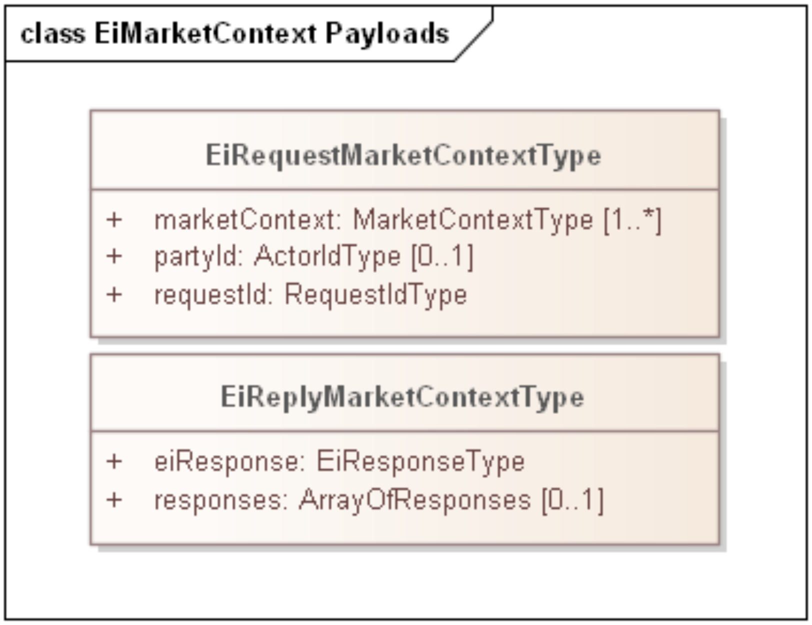


Figure 5-2: UML of Market Context Service payloads

6 Tender Facet

Transactive Services in   define and support the lifecycle of transactions from initial Tender to final settlement. The phases described in [EI] 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 specific actor is identified by its PartyID; see Section 2.3.3.

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.

6.1 Tenders as a Pre-Transaction Payloads

Pre-transaction interactions are those between parties that may prepare for a transaction. The pre-transaction facet in CTS is EiTender (and its close relative, EiDistributeTender) with payloads shown in Table 6-1.

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 6-1: Pre-Transaction Tender Services

Facet	Request Payload	Response Payload	Notes
EiTender	EiCreateTenderType	EiCreatedTenderType	Create and emit Request Payload
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

6.2 Interaction Patterns for the Tender Facet

Figure 6-1 presents the [UML] sequence diagram for the EiTender Service. Note that EiDistributeTender is not part of CTS 1.0 at present, but is being considered for a future release.

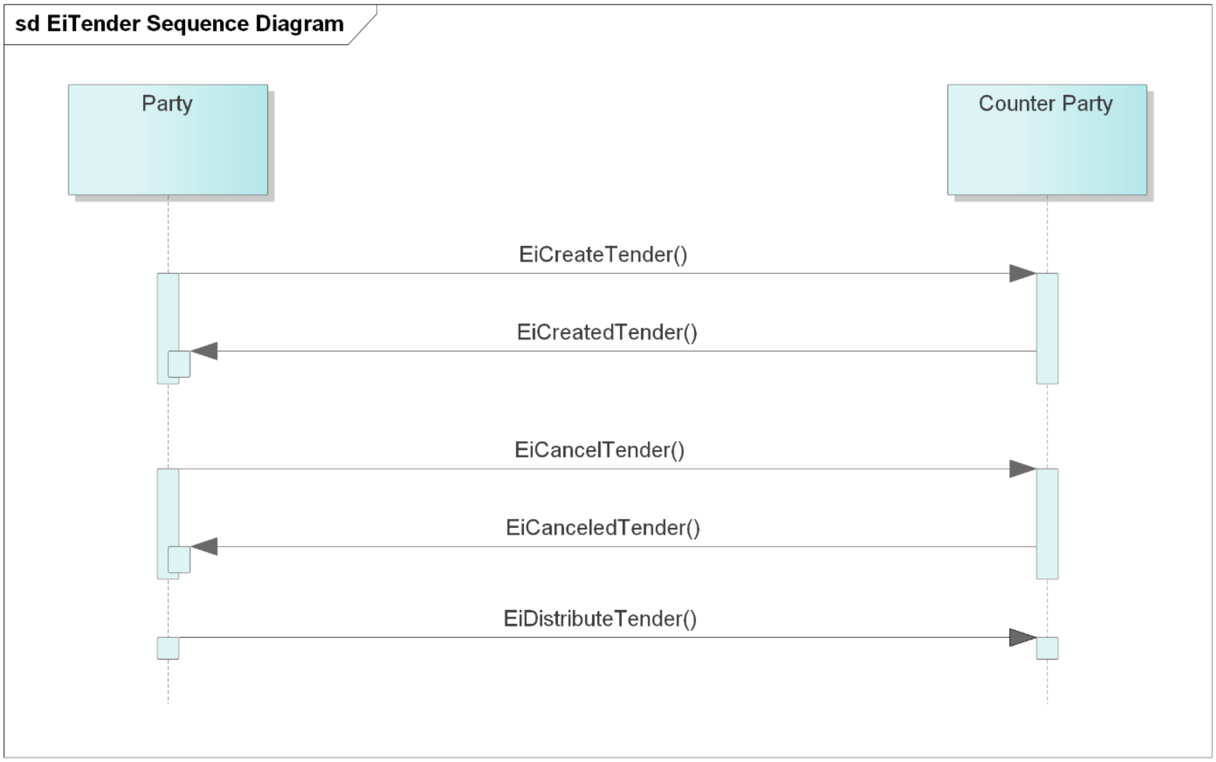


Figure 6-1: UML Sequence Diagram for the EiTender Service

6.3 Information Model for the Tender Facet

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 3.2.

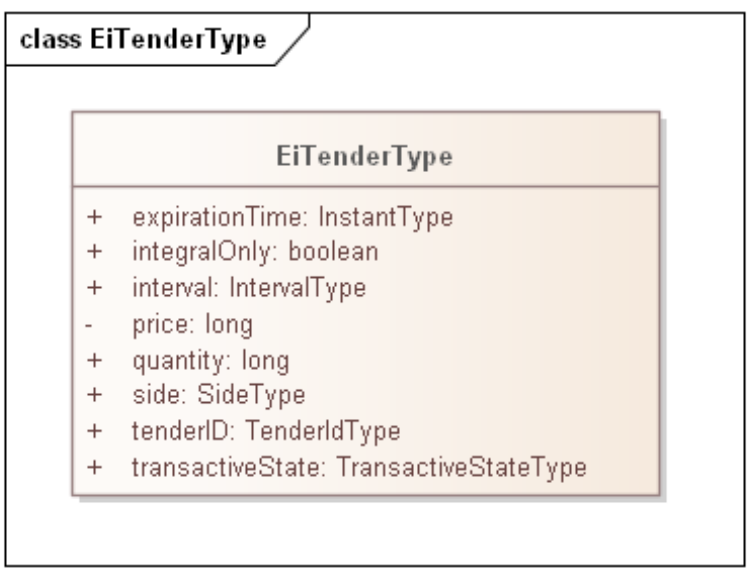


Figure 6-2: Class EiTenderType

The attributes of EiTender are shown in the following table.

Table 6-2: 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 **[EMIX]** describes the state of a transactive artifact. For CTS 1.0, only states *tender* and *transaction* are used.

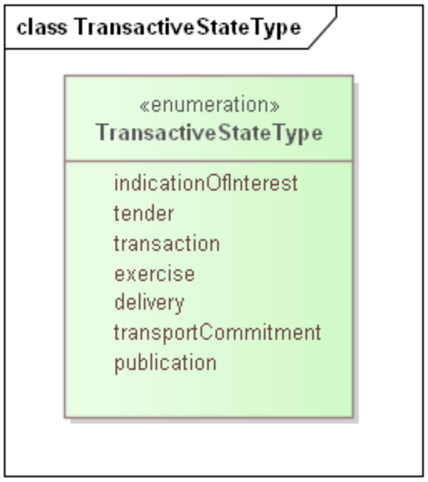


Figure 6-3-3 Enumeration TransactiveStateType

6.4 Payloads for the Tender Facet

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

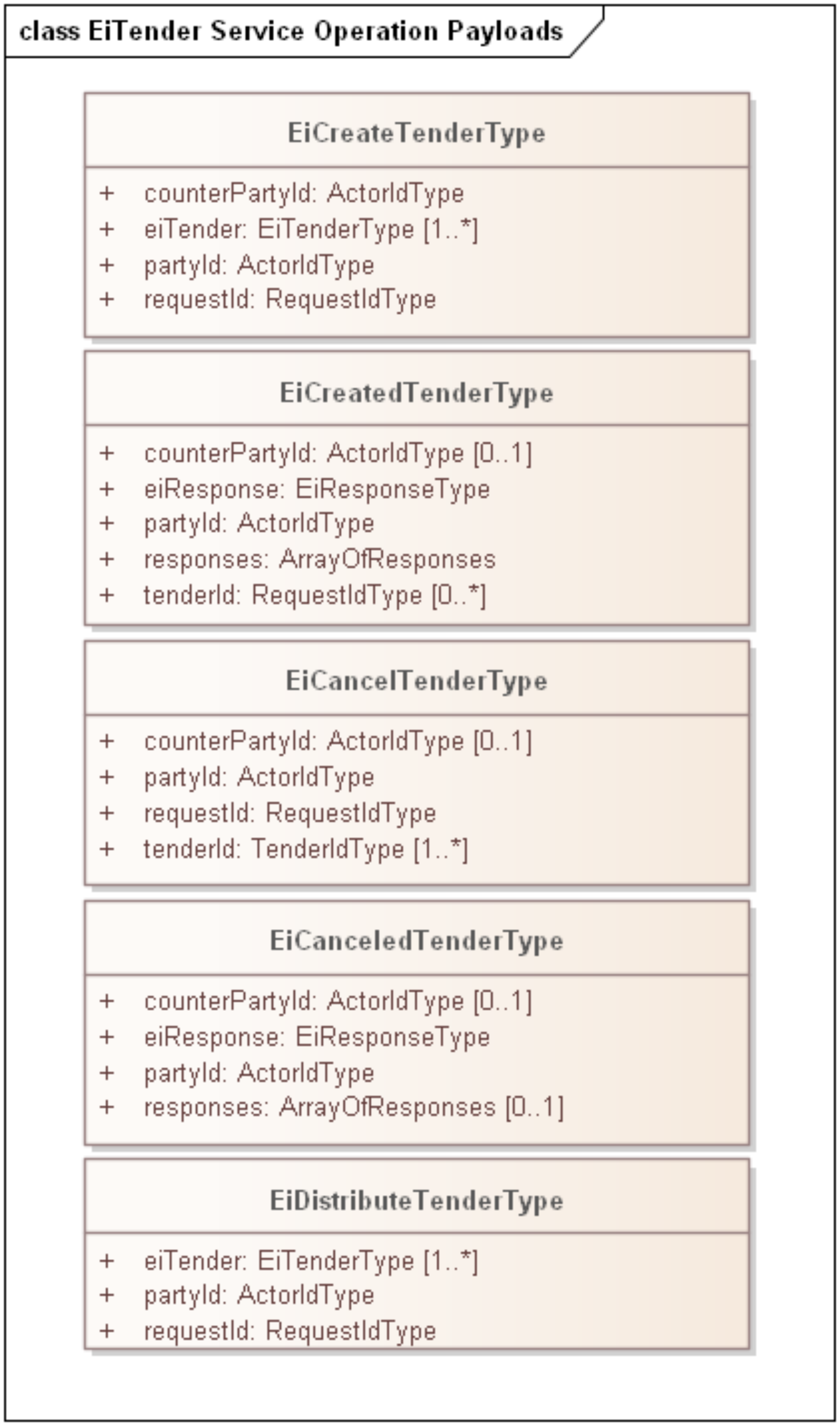


Figure 6-4: UML Class Diagram for the Operation Payloads for the EiTender Service

The following table describes the attributes for EiCreateTenderType

Table 3 EiCreateTenderType Attributes

Attribute	Meaning	Notes
Counter PartyID	The Actor ID for the CounterParty for which the tender is created.	Each market in a MarketPlace has a standard term which is the Counter PartyID to use to indicate the expectation that the market will match and clear the tender if possible. In the alternative, for a bilateral tender/transaction, an Actor's PartyID may be used.
Ei Tender	One or more EiTenders being created.	In its original form, CTS allows one EiTender per EiCreateTender payload. A subsequent Working Draft will use a single-element stream and optionally allow a stream of EiTenders. See Section 12 Creation differs from instantiation in a programming language; in CTS an object describing a Tender is instantiated then sent; the latter is a consequence of processing an EiCreateTender payload.
PartyID	The Actor ID for the Party on whose behalf this Tender is made.	This is the Actor ID showing which Actor proposes the buy or sell side EiCreateTender.
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[8].	

[8] As an example of the *Correlation Pattern* for messages

7 Transaction Facet

7.1 Transaction Services

This section presents the Transaction Facet payloads, used by Actors in the role of creating and responding to Transactions.

In the contributed specification, market context and product are implied

This section makes them explicit, consistent with the definitions in Section 3.

Canceling or modifying transactions is not permitted.[9] Following the approach of distributed agreement protocols[10], compensating tenders and transactions SHOULD be created as needed to compensate for any effects.[11]

Table 7-1: Transaction Management Service

Service	Request	Response	Notes
EiTransaction	EiCreateTransactionType	EiCreatedTransactionType	Create and acknowledge creation of a Transaction

7.2 Interaction Pattern for the Transaction Facet

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

[9] Canceling transaction is not permitted in either CTS or Energy Interoperation

[10] See, e.g., WS-Transaction and WS-BusinessActivity.

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

Figure 7-1: UML Sequence Diagram for the EiTransaction Service

7.3 Information Model for the Transaction Facet

Transactions are a CTS artifact evolved from EMIX including a Stream with time, quantity, and price. Flattening similar to that in the Tender Facet) is used.

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

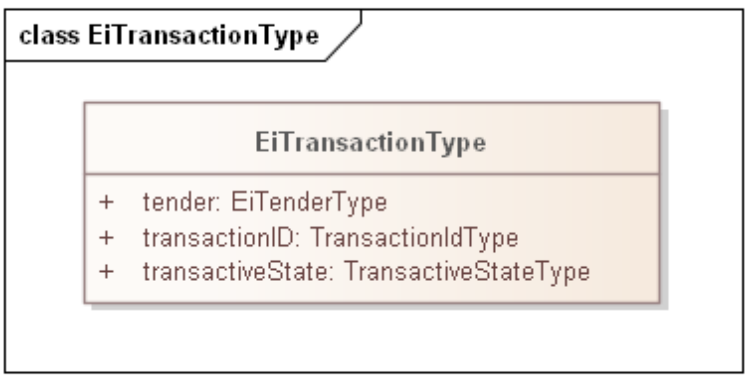


Figure 7-2: UML Class Diagram of EiTransaction

The attributes of EiTransaction are shown in the following table.

Table 7-2: 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 6-3-3 Enumeration TransactiveStateType

7.4 Operation Payloads for the Transaction Facet

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

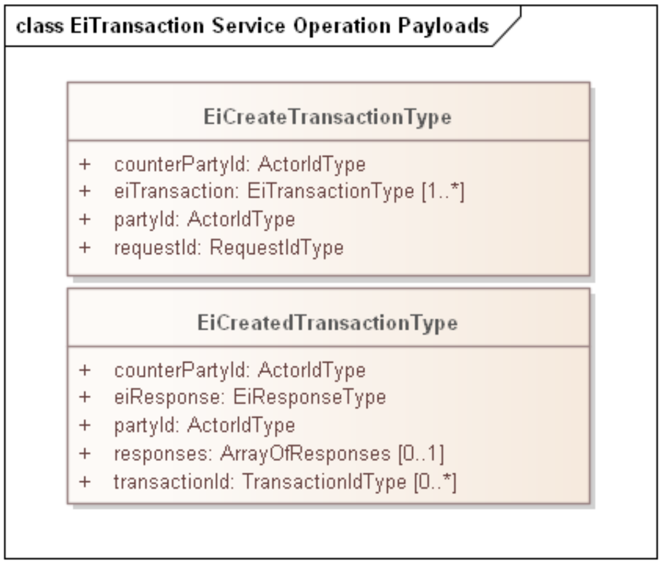
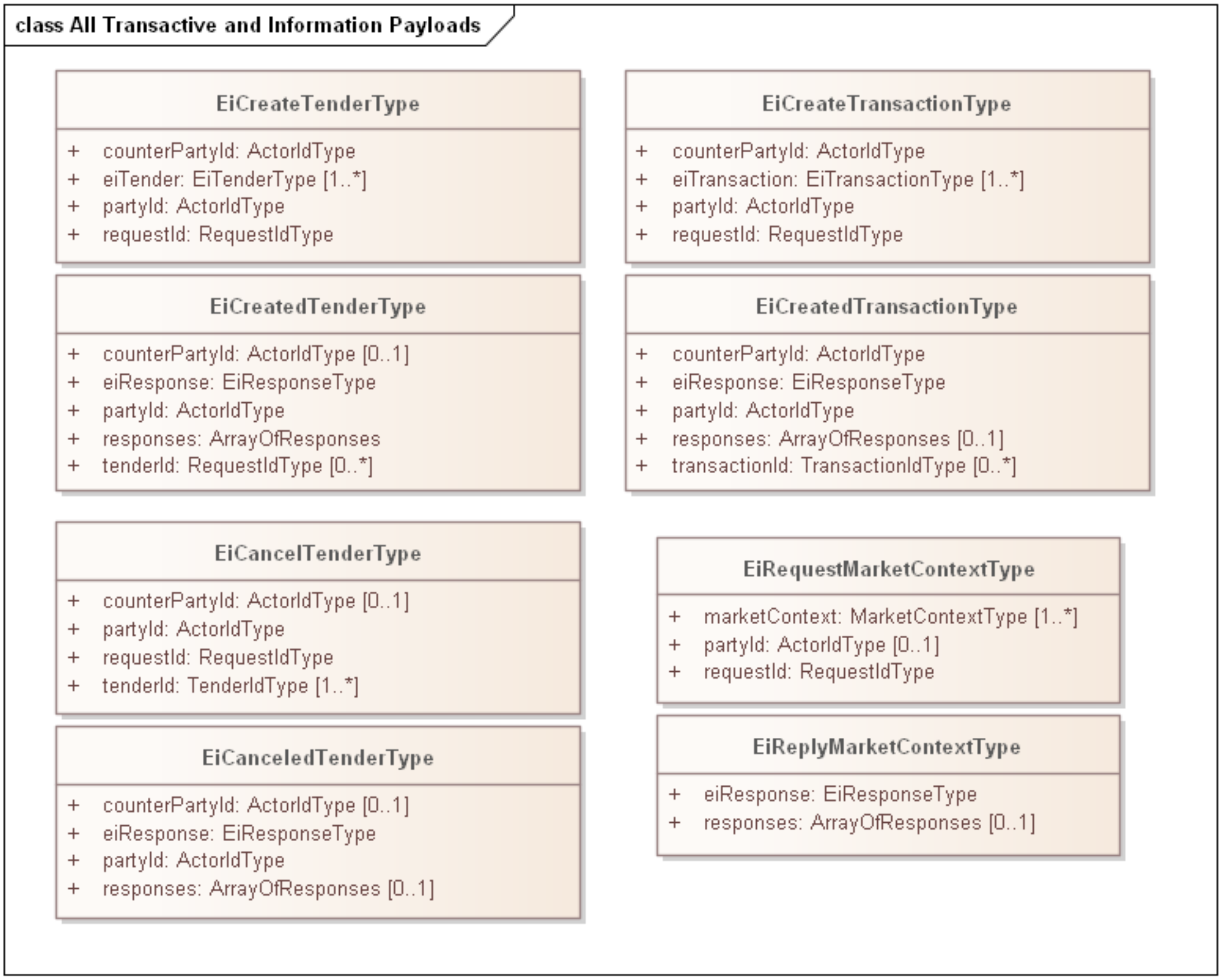


Figure 7-3: UML Class Diagram of EiTransaction Service Operation Payloads

7.5 Comparison of Transactive Payloads

In this section we show the payloads for the Tender and Transactive Facets

Figure 7-4: UML Diagram comparing all Transactive Payloads



8 Position Facet

Pending.

Follows the definition of the EiPosition Service in the Energy Mashup Lab open source implementation of CTS, EML-CTS.[12]

[12] https://github.com/EnergyMashupLab/eml-cts

9 Measurement and Verification Facet

Pending. Following EiDelivery Payloads from [EI]

NOT PLANNED FOR CTS 1.0.

10 Market Information Facet—Quotes and Tickers

Pending.

Show the relationship between a non-actionable quote and market information such as that provided by market tickers.

NOT PLANNED FOR CTS 1.0.

11 Bindings

Payloads and interaction patterns are described in **[UML]** in Section 5 above. This section contains bindings for the payloads in three encoding schemes:

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

11.1 JSON

TODO—JSON Schema available

11.2 XML Schema

TODO—XML Schema available

11.2.1 XML Namespaces

11.3 Simple Binary Encoding

TODO—Work in progress

12 Conformance

(**Note**: The OASIS TC Process 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.

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

Remove this note before submitting for publication.)

Pending update to Facet terminology

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 [EI] 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:
    a. EiQuote
    b. EiEnroll
    c. EiDelivery
3. The following Services from the TeMIX profile are included and simplified as follows.
    a. Attribute names have been made consistent with lowerCamelCase conventions.
    b. The inheritance hierarchy for UIDs and identifier types have been simplified
        i. Only selected identifier types are included
        ii. The identifier types in this draft specification are opaque types rather than strings
    c. The enumeration TransactiveStateType is identical to that in Energy Interoperation, but only the following Transactive States are used:
        i. Tender
        ii. Transaction
        iii. Indication of Interest (pending work in progress)
    d. Market Context and the EMIX Market Context are flattened and simplified as follows:
        i. MarketContextType is a URI.
        ii. 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..

12.1 Claiming Conformance to Common Transactive Services

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

Appendix A. 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.

A.1 Normative References

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

NOTE: INSERT AS FORMATTED REFERENCES. Consider [EI]

- *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

[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>.

[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]
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

A.2 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.

[Actor Model]
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]
W.T. Cox, E. Cazalet, E., A Krstulovic, W Miller, & W.Wijbrandi *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]
B. Desruisseaux, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, https://tools.ietf.org/html/rfc5545. 2009,
See also
C. Daboo & M. Douglas.*Calendar Availability*, https://tools.ietf.org/html/rfc7953, 2016
[GridFaultResilience]
W.T. Cox & T. 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
[Micromarkets]
W.T. Cox & T. Considine, *Energy, Micromarkets, and Microgrids.*
GridInterop 2011, https://www.gridwiseac.org/pdfs/forum_papers11/cox_considine_paper_gi11.pdf
[RFC3552]
E Rescorla & 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]
T. Considine & W.T. 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] (Transactive Resource Management)
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/

Appendix B. 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] 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], for more information.

Appendix C. Glossary of Terms and Abbreviations Used in this document

Throughout this document, abbreviations are used to improve clarity and brevity, especially to reference specifications with long titles.

Table C-1 Abbreviations and Terms used throughout this document for which this document is not normative.

Attribute	Meaning
CTS	Common Transactive Services
EI	Energy Interoperation, an OASIS specification as per the normative references, CTS is a conforming profile of EI. TBD Point to normative reference
EMIX	Energy Market Information Exchange, an OASIS specification used to describe products and markets for resources, particularly those traded in power grids. TBD Point to normative reference

Appendix D. Acknowledgments

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

D.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]

D.2 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)

Appendix E. 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
WD05	5/25/2021	Toby Considine David Holmberg William T Cox	Continues clean-up and condensation of sections 1, 2
WD06	6/7/2021	Toby Considine	Refines Item language into Resource and Products. Explains Message Groups as a conforming descendant of EI Services.
WD07	6/21/2021	Toby Considine William T Cox	Clarified terminology and relationship to implied Service-Oriented Architecture. Structured CTS facets for clearer explanation
WD08	8/5/2021	Toby Considine William T. Cox David Holmberg	Clarify and simplify actor facets descriptions, including Tender, Transaction, and Configuration. Reduce redundant and less relevant content.

Notices