Energy Interoperation Common Transactive Services (CTS) Version 1.0

Committee Specification Draft 02

28 April 2022

This stage:
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd02/ei-cts-v1.0-csd02.pdf (Authoritative)
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd02/ei-cts-v1.0-csd02.html
https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd02/ei-cts-v1.0-csd02.docx

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

Latest stage:
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

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
- *Energy Market Information Exchange (EMIX) Version 1.0.* Edited by Toby Considine. Latest version: http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html.
- *WS-Calendar Platform Independent Model (PIM) Version 1.0*. Edited by William Cox and Toby Considine. Latest version: http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html.
- *Schedule Signals and Streams Version 1.0.* Edited by Toby Considine and William T. Cox. Latest version: http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html.

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. 28 April 2022. OASIS Committee Specification Draft 02. https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd02/ei-cts-v1.0-csd02.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) is an application profile of the OASIS Energy Interoperation 1.0 (**[EI]**) specification, with most optionality and complexity stripped away. CTS defines the messages for transactive energy, leaving communication details unspecified. Transactive energy names the collaboration techniques to balance energy supply and energy demand at every moment even as power generation becomes decentralized and as the ownership of energy assets becomes more diverse.

The purpose of CTS is to enable broad semantic interoperation between systems in transactive energy-based markets, or in any markets whose products are commodities distinguished chiefly by time of delivery. These time-volatile commodities are termed resources, and the interactions defined in CTS are common to any market used to manage resources over time.

To encourage broad adoption, CTS uses terms from financial markets in preference to the relatively obscure terms used in specialized energy markets. Among these is the use of the term instrument for a tradable asset, or a negotiable item. In CTS, the term instrument encompasses a quantity of a Resource delivered at a particular time for a particular duration. A transaction is created when a buyer and seller agree on the price for an instrument.

Transactive resource markets coordinate Resource supply and Resource use through markets that trade instruments. The initial research into transactive resource markets used a market to allocate heat from a single furnace within a commercial building. Transactive resource markets balance supply and demand over time using automated voluntary transactions between market participants.

Examples of transactable resources include, but are not limited to, electrical energy, electrical power, natural gas, and thermal energy such as steam, hot water, or chilled water. The capability to transmit such time-dependent resources is also a transactable resource, as instruments can be defined for transmission rights as well as for the services that maintain grid frequency or voltage.

When we apply transactive resource markets to the distribution of power or energy, we refer to it as transactive energy. A significant driver of transactive energy is the desire to smooth supply and demand variability, or alternatively, to match demand to variable supply. We anticipate this variability to increase as additional variable and distributed generation sources are connected to the power grid. The reader can find an extended discussion of Transactive Energy (TE) in the EI specification **[EI]**

A goal of CTS is to enable systems and devices developed today or in the future to address the challenges of increasing distributed energy resources. CTS enables distributed actors 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 defines interactions between actors in energy markets. We do not identify whether an Actor is a single integrated system, or a distributed collection of systems and devices working together. See Section 1.5 for a discussion of the term Actor in this specification. Autonomous market actors must be able to recognize patterns and make choices to best support their own needs.

CTS messages are simple and strongly-typed, and make no assumptions about the systems or technologies behind the actors. 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 of a market, but rather a common set of messages that can be used to communicate with any transactive energy market.

The Common Transactive Services, strictly speaking, are a definition of the payloads and exchange patterns necessary for a full-service environment for interaction with markets. In other words, CTS describes the message payloads to be exchanged, defining the semantic content and ordering of messages. Any message exchange mechanism may be used, including but not limited to message queues and Service-Oriented mechanisms.

In a Service-Oriented Architecture [SOA] environment, the semantic payloads are those sent and returned by the *services* described. CTS enables any SOA or other framework to exchange equivalent semantic information without presuming the specific messaging system(s) or architecture used, thus allowing straightforward semantic interoperation.[1] See Section 2.2.

1.1 Application of the Common Transactive Services

The purpose of this specification is to codify the common interactions and messages required for energy markets. Any system able to use CTS should be able to interoperate with any CTS-conforming market with minimal or no change to system logic. The full protocol stack and cybersecurity requirements for message exchange between systems using CTS are out of scope.

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

- Smart Buildings/Homes/Industrial Facilities
- Building systems/devices
- Business Enterprises
- Electric Vehicles
- Microgrids
- Collections of IoT (Internet of Things) devices

TE demonstrations and deployments have seldom been interoperable—each uses its own message model and its own market dynamics. Many early implementations required transmitting information far beyond that needed for transactions to remote or cloud-based decision aggregators termed markets. Such markets discount local decision making while introducing new barriers to resilience such as network failure. Others rely on a single price-setting supplier. Systems built to participate in these demonstrations and deployments have been unable to interoperate with other implementations. The intent of this specification is to enable systems and markets developed for future deployments to interoperate even as the software continues to evolve.

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 **[SmartGridBusiness].**

1.2 Support for Developers

Specific coding, message, and protocol recommendations are beyond the scope of this specification which specifies information content and interactions between systems. The Common Transactive Services payloads are using the Universal Modelling Language **[UML] and** defined in XML schemas **[XSD]**. Many software development tools can accept artifacts in UML or in XSD to enforce proper message formation. To further support message interoperability, two additional common serializations are defined:

This specification 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] SOA is occasionally mis-described as a *client-server* approach. In distinction, services are requested by an Actor, and fulfilled by another Actor. In SOA the services offered are key, and the actors take different roles in different interactions.

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 an upper-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 and for attributes in the UML models, 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. In particular, architectural and functional examples are presented only to support narrative description. The specific processes, structures, and algorithms are out of scope.

1.5 Use of terms Actors and Facets in this specification

This specification defines message content and interaction patterns.

The EI 1.0 specification in 2011, presumed web services for interactions. That specification described a Service-Oriented Architecture (SOA) approach. Service orientation complements loose integration and organizes distributed capabilities that may be in different ownership domains by focusing solely on requested results rather than on mechanisms. [EI] uses the language of web services to describe all interactions.

There is a growing use of the descriptive term “cloud-native computing” for extending the architecture and technologies developed for use in clouds not only in data centers but to edge computing, where IoT devices reside. A discussion of the rapidly-evolving topics of cloud-native computing and edge computing is beyond the scope of this specification.

At the time of this specification, typical architectures decompose applications into smaller, independent building blocks that are easier to develop, deploy and maintain. A single market participant in energy may be embodied as several of these independent blocks (actors). Message queues provide loosely-coupled communication and coordination within and among these distributed applications. Message queues enable asynchronous communication, i.e., the endpoints that are producing and consuming messages interact with the queue, not each other. Publishers can add requests to the queue without waiting for their processing. Subscribers process messages only when they are available.

In the Internet of Things (IoT), the term Actor is preferred as the “actor model” makes no assumptions of the mechanisms or even motives internal to an Actor. An Actor is simply a thing that acts. The Actor

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

implementation may be by a traditional computer, a cloud node, a human behind a user interface, or any device in the Internet of things.

In transactive energy, we see the diversity supported by the term Actor in the IoT. An energy seller may be a generator or a solar panel or a virtual power plant or a demand responsive facility or a financial entity. An energy buyer may be a home or commercial facility or an embedded device or a microgrid or an energy district. A Marketplace acts to match Tenders, but may also participate to buy or sell power for itself. An energy storage system may act as a buyer or as a seller at any time.

Architectures MAY decompose applications into smaller independent Actors. We use the term “Facet” to name a coherent se of interactions that such an Actor may use to communicate with other Actors. An Actor submits tenders to buy or to sell. An Actor may operate a Market. If the architecture requires telemetry for Resource flow (metering), one of many facets supported by a Resource-consuming Actor MAY provide it, or separate Actor MAY present only the telemetry, logically and physically separated from the Resource-consuming Actor. This specification makes no requirement as to how to distribute these facets.

While this specification discusses messages between Actors, it establishes no requirement or expectation of specific implementation. While this specification uses the language of Actor and Facet, there is no architectural expectation linked to this language. One could apply the terms Actor and Facet throughout the [EI] specification. A traditional [EI] application consisting of a several unitary systems each presenting all facets as web services described by WSDL can be conformant so long as it uses a compatible set of information payloads.

A discussion of the rapidly-evolving topic of cloud-native computing is beyond the scope of this specification. This specification does not require that implementations conform to any specific implementation of cloud-native computing. Cloud-native and edge computing have informed the language of this specification, just as web services and SOA informed [EI].

1.6 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 Facet is used without needing to know all the details of its implementation. Consumers of services generally pay for results, not for effort.

1.6.1 Security Considerations

Size of transactions, costs of failure to perform, confidentiality agreements, information stewardship, and even changing regulatory requirements can require that similar transactions be expressed within quite different security contexts. Loose integration using the service-oriented architecture (SOA) style assumes careful definition of security requirements between partners. 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.

The best practice in cloud-native computing is to use Zero-Trust security **[ZeroTrust]**. Zero Trust security requires authentication and authorization of every device, person, and application. The best practice is to encrypt all messages, even those between the separate components of an application within the cloud.

This specification makes no attempt to describe methods or technologies to enable Zero Trust interactions between Actors.

1.6.2 Privacy Considerations

Detailed knowledge of offers to buy or sell or knowledge of energy inputs and outputs for an Actor may reveal information on actions and operations. For example, transactions or tenders may indicate whether a production line is starting or stopping, or anticipated energy needs, or who has been buying or selling power. Making such information public may be damaging to actors. Similarly, an adverse party may be able to determine the likelihood that a dwelling is presently occupied.

The essence of any transaction is the agreement of a Party to sell, and a Party to buy. The identity of the buyer and the identity of the seller are each part of the transaction. Some transaction notifications may hide the identity of the buyer from the seller. Some transaction notifications may hide the identity of the seller from the buyer. Some transactions, such as double auction, may be between the market participants as a whole, and not with any particular counterparty. Where this is required, the Market itself may be designated as the counterparty in a notification.

Both security and privacy considerations are addressed in Appendix B.

1.7 Semantic Composition

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

EI references two other standards, [EMIX] and [WS-Calendar], and uses an earlier Streams definition. We adapt, update, and simplify the use of the referenced standards, while maintaining conformance.

- 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 EI 1.0.
- EI uses the vocabulary and information models defined by those specifications to describe the services that it provides. The payload for each EI 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.

In [EI], the fundamental resource definition was the [EMIX] Item, composed of: a resource name, a unit of measure, a scale factor, and a quantity. For example, a specific EMIX item may define a Market denominated in 25 MW-hour bids. [EI] defined how to buy and sell items during specific intervals defined by a duration and a start time. The Quotes, Tenders, and Transactions that are the subject of [EI] added specific prices and quantities to the item and interval. EMIX optionally included a location, i.e., a point of delivery for each [EI] service.

In CTS, we group and name these elements as a Resource, Product, and Instrument. These terms are defined in Section 2.1.3, “Resources, Products, and Instruments”

Note that the informational elements in a fully defined tender or transaction are identical to those described in EMIX. The conceptual regrouping enables common behaviors including Market discovery and interoperation between Actors built on different code bases.

1.7.1 Conformance with Energy Interoperation

EI defines an end-to-end interaction model for transactive services and for demand response. CTS uses the EI transactive services, and draws definitions of parties and transactive interactions primarily from the [EI] TEMIX profile.

This specification can be viewed as a minimal transactive profile of [EI]

1.7.2 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 Marketplace that an actor can use to configure itself for interoperation with a given Marketplace. We extend and clarify those terms, provide an extension mechanism, and discuss the relationship of markets, Marketplaces, and products.

1.7.3 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 as standardized in [**iCalendar**]. Schemas in **[WS-Calendar]** support messages that are nearly identical to those used in human schedules. We use a conformant but simpler and more abstract Platform Independent Model **[CAL-PIM]** and the [**Streams**] compact expression[3], to support telemetry (Delivery Facet) and series of Tenders while not extending the semantics of [**Streams**].[4]

By design and intent, the **[WS-Calendar]** 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 share common information such as duration. Inheritance is used to “complete” a partial message during negotiation. CTS makes use of this to apply a 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.

1.7.4 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 or other Resource 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 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.

FSGIM load requests can be expressed using CTS tenders. CTS 1.0 uses single-interval **[Streams]** to express single-interval tenders in anticipation of possible future use of Streams in FSGIM-conformant communications.

[3] Simplified as CTS Streams in this specification.

[4] Some specifications (e.g. [FSGIM]) have extended the basic [Streams] capabilities, but this brings additional complexity which does not benefit our use cases.

2 Overview of Common Transactive Services

2.1 Scope of Common Transactive Services

CTS provides for the exchange of resources among parties which represent any provider or consumer of energy (e.g., a distributed energy resource). CTS makes 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.

An actor can represent a microgrid within a larger micromarket; the actor would in effect aggregate the resources in the microgrid. As above, such an actor would not reveal any internal mechanisms, but only its interest in buying and selling power. There is no explicit bound on repeating this interoperation pattern.

An actor representing a microgrid may interoperate with markets in a regional grid, which may or may not be using CTS. The regional grid may use transactive energy expressed in non-CTS messages, or infrastructure capacity may limit delivery to the microgrid. In either case, the An actor representing a microgrid must translate and enforce constraints and share information with the other nodes in the microgrid solely by means of CTS. Any translations or calculations performed are out of scope.

See informative references **[StructuredEnergy]** and **[SmartGridBusiness]** for a discussion. **[Fractal Microgrids]** is an early reference that describes hierarchies of microgrids. **[Transactive Microgrids]** describes transactive energy in microgrids.

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 Marketplace and 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.

This specification describes standard messages, the set of which may be extended.

2.1.3 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.

In [EI], the fundamental resource definition was the [EMIX] Item, composed of: a resource name, a unit of measure, a scale factor, and a quantity. For example, a specific EMIX item may define a Market denominated in 25 MW-hour bids. [EI] defined how to buy and sell items during specific intervals defined by a duration and a start time. The Quotes, Tenders, and Transactions that are the subject of [EI] added specific prices and quantities to the item and interval. EMIX optionally included a location, i.e., a point of delivery for each [EI] service.

In CTS, we group and name these elements as a Resource, Product, and Instrument. A Resource is the name and the unit of measure, as in the EMIX Item. A Product, i.e., what can be bought and sold in a Market, in addition specifies "how much" and "for how long" as well as optional elements such as location and Warrants. The term Instrument, as in financial markets, adds a specific start time to a Product.

We define a Resource as any commodity whose value is determined by a fine-grained 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's 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, following common usage in financial markets where an instrument names the thing that is bought or sold. 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.

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. This specification does not assume or require an Order Book, a Double Auction, or any other mechanism in the Matching Engine.

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

These terms are summarized in Table 2-1: .

Table 2-1: Definitions of CTS Market terms

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 quarter 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.

Transactive Entity	Definition
Party	An Actor or a set of coordinating Actors 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 Section 2.2.3 below.
Market	A Facet where Parties trade a Product using tenders submitted to buy or sell an Instrument.
Marketplace	A Marketplace names a set of Markets accessible to a Party. The Marketplace Facet supplies limited information common to all Markets in the Marketplace. The Facet also enumerates all Resources and Products available in the Marketplace, as well as a directory of the Markets for each Resource. CTS differs from EMIX in adding a distinction between Market and Marketplace, while making no assumption about how the distinction is implimented or even whether Markets and Marketplaces share common ownership. The [EMIX] Market Context, identified by a URI, is akin to the CTS Marketplace.
Market Context	A URI identifying an individual Market, as defined in EMIX.
Marketplace Context	A URI identifying a Marketplace, as defined in the EMIX Market Context.
Matching Engine	There are many Market processes to exchange offers and reach agreements on transactions. Different parts of the same Marketplace MAY employ different Market processes. We term each of these processes a Matching Engine. This specification uses the term Matching Engine only to support narrative description. The specific processes, structures, and algorithms of Matching Engines are out of scope.

2.2 Common Transactive Services Roles

Actors interact through Facets. The specification makes no assertions about the behaviors, processes, or motives within each Actor. A particular Actor may use all Facets, a subset of Facets, or even a single Facet. This specification defines Facet messages and interactions.

[EI] defines contracts between Actors as services with defined messages and interactions. All **[EI]** services map to CTS Facets. Nearly all Facets defined in CTS are services as defined in **[EI].** CTS defines two additional Facets for market operations not derived from the Services in [EI], namely Position and Delivery, as well as two facets for Market discovery, the Marketplace and Market Facets. CTS does not require a conforming transactive energy market to use every Facet.

2.2.1 Parties as Market Participants

The Common Transactive Services (CTS) defines interactions in a Resource Market. This Resource Market is a means to make collaborative decisions that allocate power or other Resource over time. We follow [EI] and financial markets by terming market participants as Parties.

A Party can take one of two Sides in 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 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.

A Party may represent a single actor, or the roles (see Facets, below) of a single Party may distributed across multiple Actors.

We do not specify how to manage delivery of the Resource.

2.2.2 Facets in CTS

This specification refers to coherent se of interactions, that is, closely related requests, responses, as Facets. A Facet sends and receives defined messages to interact with other Actors that expose the same Facet. An Actor in a CTS-based system of systems may expose all Facets, a single Facet, or any collection of Facets. A particular Market may use some or all named Facets. A participant in a Market must include Actors supporting each Facet required in that Market; there is no requirement that each Actor supports all these Facets.

Each Facet named below groups a mandatory set of related messages and interactions. Detailed descriptions of each facet begin in Section 6.

Table 2-2: Facets Defined in CTS

Facet	Definition
Marketplace	The Marketplace Facet exchanges information about the Marketplace and its Products and Markets. A Party registers with a Marketplace through this Facet. A Party may query the Marketplace to discover the Resources and Products traded in a Marketplace. When a Marketplace includes multiple Products, the Party needs to know where to find the Market for each Product. While a Marketplace may change slowly over time, the Marketplace facet can generally be viewed AS conveying static information.
Market	A Market Facet exchanges information for trading a particular Product in a particular Market. Parties submit Tenders to a Market, and the Market notifies the Parties of Transactions. A Market Facet contains a Matching Engine that matches Tenders to buy and Tenders to sell. The Market Facet conveys information as to how the Market matches orders, which may change the strategies used by a Market participant. Some Markets MAY register transactions privately agreed to among Parties. See Section 8 Market Facet

Facet	Definition
Registration	A Party must Register with a Marketplace to participate in the Markets in that Marketplace. See Section 6, Party Registration Facet.
Tender	Tenders are actionable offers 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 Parties in the Market. Note: a Tender for one side MAY match more than one Tender on the other side, which could generate multiple Transactions. See Section 9, Tender Facet.
Transaction	A Transaction records a contract when a Tender to buy and a Tender to sell are matched. Each Party is notified of the creation of the Transaction. Note: a Tender for one side MAY match more than one Tender on the other side, which would generate multiple Transactions. See Section 10, Transaction Facet.
Position	At any moment, a Party has a position which represents the cumulative amount of an Instrument that an actor has previously transacted for within a bounding time interval. A Position for an Instrument reflects the algebraic sum of all quantities previously bought or sold. See Section 11, Position Facet.
Delivery	It is simplest to think of Delivery as a meter reading, although that meter may be virtual or computed. Some implementations may compare what was purchased or sold with what was delivered. What a system does after this comparison is out of scope. See Section 12, Delivery Facet.
Quote	A Quote is a non-actionable indication of a potential price or availability of an Instrument. **[EI]** defines the EiQuote service. This specification extends the Quote to include forecasts, information about completed Transactions, and other Market information. See Section 13 Market Information—the Quote and Ticker Facets
Ticker	Named for the stock ticker, best known for its printed output the ticker tape. A ticker provides public information about transactions over time. See Section 13 Market Information—the Quote and Ticker Facets

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.

2.2.3 Party and Counterparty in Tenders and Transactions

The Party in a Tender is offering to buy or sell. The *PartyID* in a Tender should always reference the Party that is tendering.

When the Market recognizes Tenders that match each other, however defined, the Market generates a Transaction that represents a contract between the buyer and the seller. This Transaction includes a Party and a Counterparty.

See Section 13, "*Market Information—the Quote and Ticker* Facets" for a discussion of Market Information.

2.3 Responses

This section re-iterates terms and simplifies models from [EI]. That specification is normative. The form of the Response is common across all Facets.

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[5].
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,[6] 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 Most operations have a response.

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

[6] 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 and compatible with 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. Section 5 defines the *CTS Stream* as a conformant specialization of **[Streams]**, integrating information that is outside of a Stream data structure but associated with a Stream.[7]

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 Date-Time?"
Offset	An offset (expressed as a WS-Calendar Duration) that some markets MAY use to transfer trading away from hourly boundaries. A power distribution entity may experience disruption if there is a big price change on the hour. Offset enables a Market to trade, for example, 3 minutes after the hour. See also Market Facet
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 Date-Time of the Instrument.
Interval	An Interval in CTS is a Duration with a Begin Date-Time. This maps to what WS-Calendar names a "Scheduled Interval".

3.2 Semantic Elements from EMIX

EMIX defines the specification of commodity goods and services whose value is determined by time and location of delivery. EMIX defines an "Item" by what is sold in a Market, when it is sold, what the units are, and what the standard trade size is. EMIX further defines how to communicate the date and time of delivery for that commodity to define a unique Product that can be bought and sold in a Market.

In CTS, we maintain the semantics of EMIX while giving name to each refinement of the information. These names are the Resource (what is sold), the Product (how the Resource is packaged into a size

[7] Including Resource Designator, Stream Start, and Decimal Fraction

and Duration for sale), and the Instrument (a Product sold at a specific time). Instruments are what are bought and sold in CTS markets.

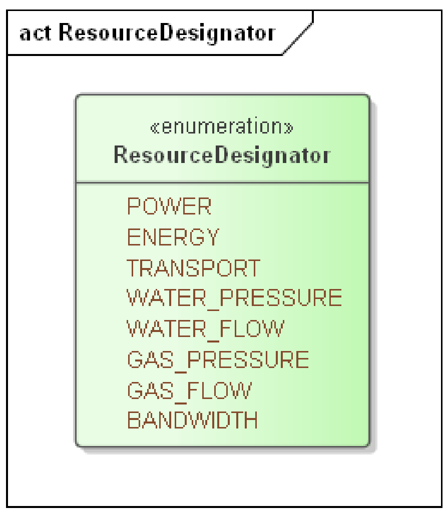
3.2.1 Defining Resource

Here we define a Resource as a commodity that is bought or sold in a CTS Marketplace. A Party can query a Marketplace to discover the Resources that can traded in each of the Markets in the Marketplace.

Table 3-2: Defining the Resource

Attribute	Meaning
Resource	A Resource consists of a Resource Designator, a Resource Name and a Resource [Item] Description.
Item Description	The Item Description is a common name, as defined 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

A Resource Designator is an extensible enumeration. The standard enumeration is defined in Figure 3-1 Resource Designator Extensible Enumeration:



j

Figure 3-1 Resource Designator Extensible Enumeration

3.2.2 Defining Product

The Product is a Resource packaged for Market. The size and duration of the Resource define what is, in effect, the "package size" for the commodity. A Marketplace may offer multiple Products for the same 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 referenced by the Resource Designator.

Attribute	Meaning
Scale	Exponent that specifies the size of the Resource Unit. For example, a Product denominated in Megawatts has a Scale 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 restricts the Product beyond the Resource definition. For example, it is possible to trade in power designated to be Neighborhood Solar Power. In CTS, Products that are identical other than the Warrant are traded in different Markets within the same Marketplace.

Products with differing Warrants are different Products and therefore traded in different Markets.

As non-normative examples, if an Actor wishes to buy energy with a *Green Warrant* (however defined) then the Actor, not the Market, is responsible for defining its trading strategies if the warranted Product is not available. Similarly, an Actor that wishes to buy or sell Neighborhood Solar Power is responsible for submitting Tenders that expire in time to make alternate arrangements, or in time to cancel Tenders before fulfillment. This specification establishes no expectation that the Market engine address these issues automatically.

Warrants are defined in **[EMIX]**, and are permitted in CTS to support this complexity if desired, but not described in this specification.

3.2.3 Market Semantics from EMIX

EMIX defines vocabulary used in market messages and interactions.

Table 3-4: Market-related elements from EMIX

Attribute	Meaning
PartyID	The Marketplace-based ID of an actor participating in Markets, particularly the actor originating a Tender, Quote, or Contract.
Counter PartyID	The Marketplace-based ID of an actor participating in Markets, particularly the actor taking the other side of a contract from the Party. See Section 2.2.3.
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 or subsequent to the Begin Date-Time of the Instrument.
Market Context	In EMIX, the Market Context is simply a URI to name a Market, and need not be resolvable. CTS distinguishes between a Marketplace, where many Products may be sold and the Market, where a specific Product is sold. See Section 6. "Marketplace Facet".

Standard Terms	Standard Terms are the machine-readable information about a Marketplace or Market, and the interactions it supports. In CTS, the Standard Terms include an enumeration of the Products and their respective Markets tradable in this Marketplace. See Section 6, “Market Facet”.

EMIX does not define how an Actor discovers the Standard Terms in a Marketplace. CTS defines the Marketplace Facet and the Market Facet to discover and expose Products and Standard Terms.

4 Basic Interaction and Terminology

4.1 Structure of Common Transactive Services and Operations

The Common Transactive Services presented in this specification are described in the following sections, and are

- Marketplace Facet—characteristics and to know what Products and Instruments can be traded
- Party Registration Facet—identification of actors within a Market or Marketplace
- Tender Facet—make offers to buy and sell Instruments
- Transaction Facet—for expressing transactions.
- Position Facet—Describe what has been previously bought or sold
- Delivery Facet—Request data on actual deliveries
- Market Information—the Quote and Ticker Facets

We include UML definitions for the standard payloads for service requests, rather than the service, communication, or other characteristics. In Section 14 we describe standard serialization for the CTS standard payloads; additional bindings may be used by conforming implementations.

Transactive Services in EI define and support the lifecycle of transactions from preparation (registration) to initial Tender to final settlement. The phases described in EI are in the following table with the CTS Facets in Column 2.

Table 4-1: Mapping CTS Facets to EI Phases

EI Phase	CTS Facet(s)
Registration (and Market discovery)	Party Registration Facet Marketplace Facet Market Facet
Pre-Transaction	Quote Facet Tender Facet
Transaction	Transaction Facet
Post-Transaction	Position Facet Delivery Facet Ticker Facet

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*.[8]

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 EI 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

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 responses[9] indicating partial or complete success or failure.

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

CTS uses a simplified version of EiResponseType from EI, deleting ArrayOfResponseTermsViolated and responseDescription (to zero, that is, not passed). Response Terms Violated is renamed *Market Attribute Violation.*

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

[9] This contrasts with EI, where it is not mandatory to return any responses if the entire EiCancelTender service operation was completed successfully. The pattern in EI is to return those that have failed (required) and those that succeeded (optional).

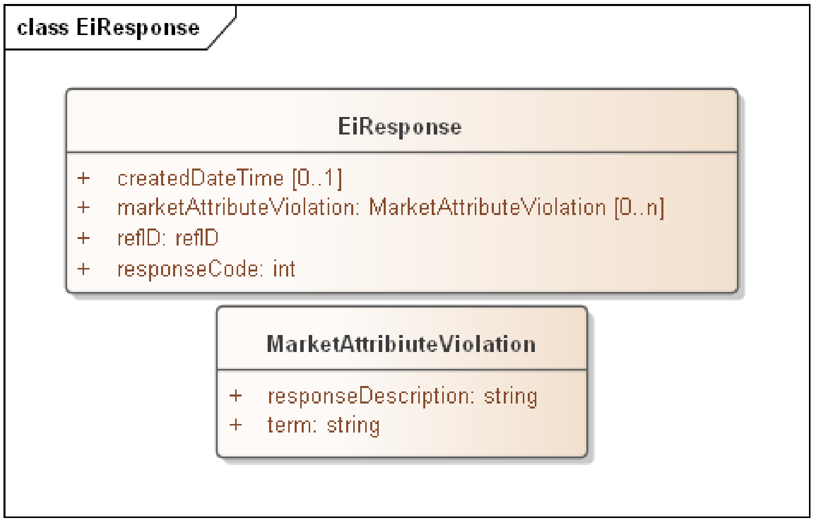


Figure 4-1: Example of generic response object

There is no exhaustive list of all possible Response Codes. More detail on Response Codes is in Section 2.3.

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.[10]

The only defined value in EI after the leading digit of the Response Code 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 indicate Requester Error. Response codes not of the form x00 MAY be treated as the parallel x00 response.

As an example, consider a request to quote 13.5 kW at three minutes offset for 17 minutes where the market characteristics and its product include 10kW granularity, zero offset, and five minute duration. The terms in the Market Attribute Violation therefore include at least these violations:

- T_GRAIN, 5m
- Q_GRAIN, 10kW
- OFFSET, 0

The definition of the respective terms is in *Section 8, Market Facet*.

[10] This is parallel to HTTP response codes.

5 CTS Streams

Aside from registration and market information, Payloads in CTS are derived from and conformant with WS-Calendar Streams. The essence of Streams is that for a series of consecutive Durations over time, called Intervals, invariant information is in the header or preface to the stream, and only the varying information is expressed in each Interval.

For CTS, this means that a Product is fully described in the header, and only the elements that vary, such as the Price or the Quantity, are expressed in the intervals.

CTS Streams use this same format even when the Intervals contain only a single Interval.

In addition, CTS Streams include energy-market elements that are outside the Streams standard but follow the pattern of referrals as defined in [Steams] conformance.

CTS Streams have neither interaction patterns nor payloads, as they are a common abstract information model used to define the messages used in Facet messages.

5.1 Information Model for CTS Streams

The CTS Stream is defined as follows. The elements from **[Streams]** have been flattened into the CTS Stream, and the Stream Interval and payload flattened into a streamPayloadValue and the internal local UID for the stream element.

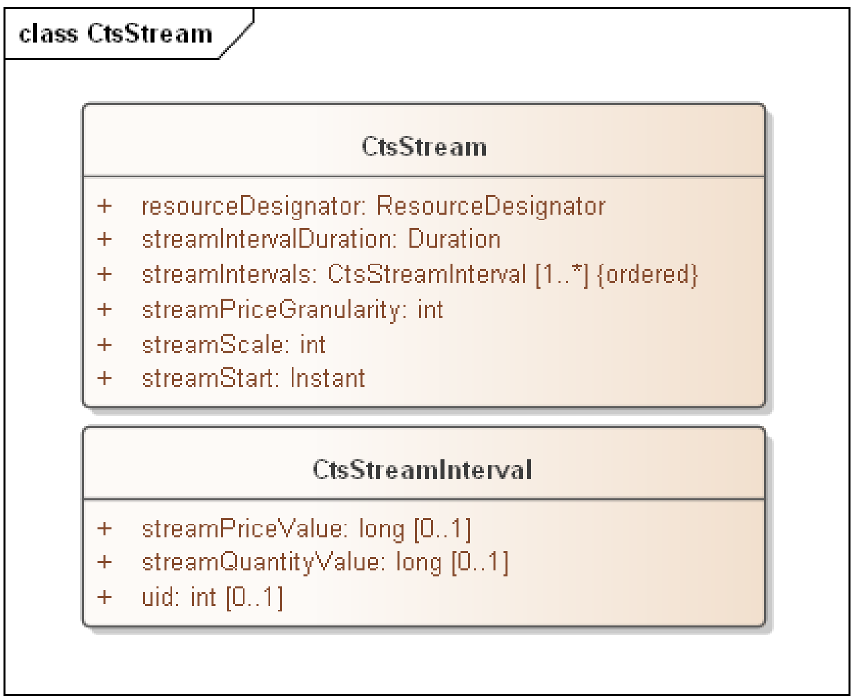


Figure 5-1: CTS Stream Definition

As with [Streams], CTS Stream Intervals are ordered, that is the sequence of intervals is essential. Some serialization specifications, notably XML, do not require that order be preserved when deserializing a list. The UID enables proper ordering of the Stream Intervals if order is not preserved. Since conformant CTS implementations need not be owned by the same implementer, and may pass through multiple translations, the UID property is required.

The following tables describe the attributes for CTS Streams and Stream Intervals.

542 *Table 5-1: CTS Stream Attributes*

Attribute	Meaning	Notes
Resource Designator	An item from an enumeration that indicates the Resource for the Product and Market	The Resource Designator in a Market should match the Resource Designator indicated in the Marketplace
Stream Scale	The Scale is the exponent that determines the size of the Resource.	For example, if Scale is 3 and the Resource is Watts, then the value is in kW. If the Scale is 6, then the value is in MW.
Stream Interval Duration	The duration for each of the contiguous Stream Intervals	This completes the Product definition of a Resource at a Scale and Size delivered over a Duration.
Stream Price Granularity	Price granularity expressed as an exponent. Applies to all Intervals in the Stream. Not required for all Facets.	For example, if the price granularity is -3, and the value is 1500, the price is 1.500 currency units.
Stream Start	The Start Date and Time for a bound CTS Stream	See WS-Calendar Date-Time in Section 3.1.
Stream Intervals	The ordered set of Stream Intervals	The set of Intervals is ordered by means of a local UID which is concatenated with the Stream UID as described in [Streams] and in [EI]

543

Table 5-2: Stream Interval Attributes

Attribute	Meaning	Notes
Stream Price Value	The Price value for this specific Stream Interval, subject to indicated Scale/Granularity	At least one of (Stream Price Value, Stream Quantity Value) MUST be present.
Stream Quantity Value	The Quantity value for this specific Stream Interval, subject to indicated Scale/Granularity	At least one of (Stream Price Value, Stream Quantity Value) MUST be present.
UID	A "Local UID" used to order the Interval within the Stream	As conformant CTS implementations need not be owned by the same implementer, intermarket gateways (however defined) may deserialize and re-serialize to different specifications

6 Party Registration Facet

Background (adapted from **[EI]**)

A valid Party ID is required to interact with a Market and is included in most payloads.

Party Registration is described in EI. This facet describes the messages necessary for an actor to register and obtain a Party ID in order to participate in a Market.

- EiCreateParty associates an actor with a Party ID and informs the Marketplace of that ID. CTS makes no representation on whether that ID is an immutable characteristic, such as a MAC address, a stable network address, such as an IP, or assigned during registration,
- EiRegisterParty names the exchange of information about an actor that enables full participation in a CTS Marketplace. It may exchange information needed for financial transfers including, perhaps, reference to an existing customer or vendor ID, or proof of financial bond for large participants, or issuance of crypto-tokens, or any other local market requirements. A Registered Party is ready to be a full participant in the local Market.
- Cancel Party Registration removes a Party from the Market. It may include final settlement, cancellation of outstanding Tenders, backing out of future contracts, or other activities as defined in a particular CTS Marketplace.

Aside from the business services as described, Party Registration may have additional low-level requirements tied to the protocol itself used in a particular implementation based on CTS.

This specification does not attempt to standardize these interactions and messages beyond naming the Register Party facet. A more complete discussion can be found in the [EI] specification.

Some Marketplaces MAY wish to associate one or more measurement points with a Party. Such measurement points could be used to audit Transaction completion, to assess charges for using uncontracted for energy, etc. Measurement points are referenced in *Section 12 "Delivery Facet"*, Markets that require this functionality may want to include an enumeration of Measurement Points in Party Registration.

7 Marketplace Facet

The Marketplace facet is an extension of [EI]. In CTS the Marketplace includes all the Markets wherein a Party can trade, and are associated with all the Products a Party can trade for.

For example, where all trading is in a single microgrid, the Marketplace is implicitly tied to that microgrid. Where trading is across a city or across a traditional utility or across a region the Marketplace hosts all Market interactions for that utility or region. Nothing in this specification prohibits multiple Marketplaces, such as a Wholesale or a Retail, or a sourced Marketplace such as Solar.

Using the Resource / Product / Instrument terminology, each Product has its own market, and these different markets may have different rules, or different matching engines. All are in the same Marketplace.

The Marketplace Facet defines characteristics common to all the local Markets, and catalogs how to participate in each Market.

7.1 The [EI] Market Context and the Marketplace

Market Contexts in [EMIX] and [EI] are URIs and are used to request information about the Market or Marketplace that rarely changes, so it is not necessary to communicate it with each message.

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

For any Marketplace, 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 and Tenders that meet those expectations. If all information about the Marketplace were to be transmitted in every information exchange, messages would be overly repetitive.

7.2 Registering in a Marketplace

The scope of a Party ID is a Marketplace. The Party ID MUST be unique within a Marketplace.

Only the acquiring of Party ID is in scope in the following list:

- Obtain Party ID
- Establish billing terms, if any
- Exchange Location, if required

See Party Registration Facet for more information.

7.3 Interaction Pattern for the Marketplace Facet

An Actor MUST interact with a specific Market to trade a specific Product. A Market matches Tenders for all Instruments based on a given Product. The matching engine is contained within the Market and different matching engines have no visibility past the Market Facet.

The Marketplace Facet enables a Party to request the details of a Marketplace and the Markets contained in the Marketplace. Using the Marketplace Facet for discovery, Parties MAY request and compare Market Characteristics to select which markets to participate in.

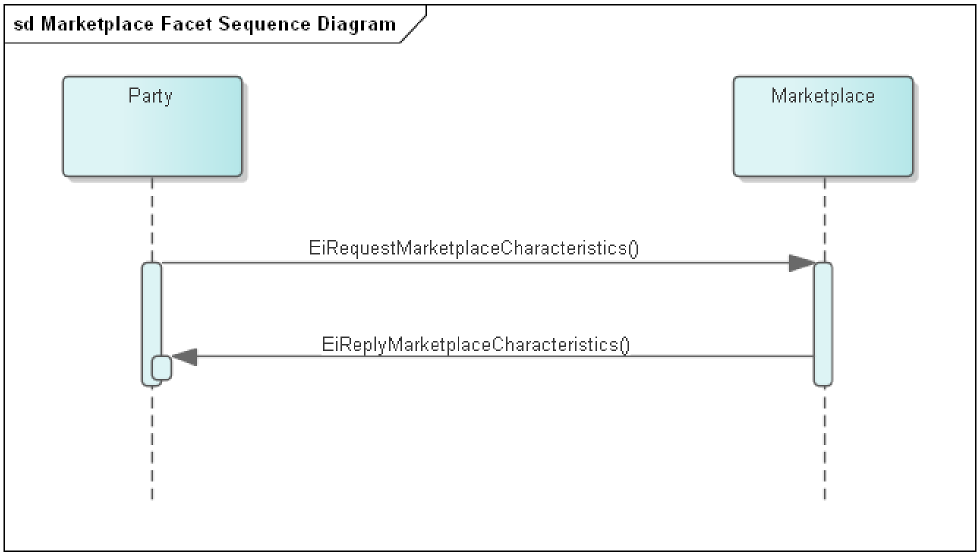


Figure 7-1: UML Sequence Diagram for the Markeplacet Facet

Once Markets are identified as a candidate, the Market Facet can retrieve the standard terms associated with those Markets. See Market Facet.

7.4 Information Model for the Marketplace Facet

Table 7-1 Information Model for the Marketplace Facet

Attribute	Attribute Name	Attribute Type	Meaning
Marketplace Name	NAME	String	Text providing a descriptive name for a Marketplace. While the Name MAY be displayed in a user interface; it is not meaningful to the Actors.
Currency	CURRENCY	String	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	Long	A Duration that some Marketplaces MAY use to describe trading where a first interval is not on an hourly boundary.[11]

[11] 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 Market could use an offset duration of 3 minutes to indicate that all tenders are based on three minutes after the hour.

Attribute	Attribute Name	Attribute Type	Meaning
Time Zone	TZ	String	A Time Zone indicates how all Times and Dates are expressed.
Price Decimal Fraction Digits	PRICE_FRAC	Long	Some market implementations use a Marketplace-wide indication of how many decimal fraction digits are used.
Resource Descriptor	RESOURCE	String	The Resource traded in this Market.
Markets	MARKETS	Market Description	A list of Market Descriptions for each Market contained in the Marketplace.

Market Descriptions in a Marketplace

A marketplace itemizes each of the Markets in the marketplace. This is indicated by a set of Market Descriptors with the following attributes, one for each contained Market:

Table 7-2 Market Description

Attribute	Attribute Name	Attribute Type	Meaning
Market Name	MARKET	String	Optional text providing a descriptive name for a Market. While the Name MAY be displayed in a user interface; it is not meaningful to the Actors.
Resource	Resource	Resource Designator	[Extensible] enumeration indicating what is sold in each Market
URI	MARKET-URI	String	URI to access the Market.

Descriptions of the Product found in each Market are found in each Market and are not replicated into the Marketplace.

8 Market Facet

All interactions in a Market are subject to Standard Terms which are discovered through the Market Facet.

1. A Party interacts with the Marketplace to discover all Markets in which the Resources the Party is interested in are traded.
2. A Party queries each of the Markets trading that Resource discover the Products in each Market, and the Standard Terms for each.
3. Resources with Warrants are in their own Markets, which may have their own Standard Terms. The Warrant is a Market Term.
4. The Party uses the Party ID determined during Marketplace Registration for all Tenders.
5. The Party determines which Products it wants by submitting Tenders to the Market is chooses.
6. Each Tender is for a specific Instrument, which is the Market Product plus a Starting Time.

A Market matches Tenders to create Transactions using the Tender and Transaction Facets.

While there is no standard matching algorithm defined in CTS, the Standard Terms include indicators of how the Market matches Tenders. For example, different bidding strategies may be used when submitting to a double auction market than for an order book market.

Interactions with the Market are through the Tender (see Section 9) and Transaction (see Section 10) Facets.

8.1 Market Context History

Market Contexts in [EMIX] and [EI] are URIs and express Standard Terms that rarely changes, so it is not necessary to communicate it with each message.

In CTS, this is refined to the Marketplace Facet (Section 7) and the Market Facet (Section 8).

8.2 Payloads for the Market Facet

Facet	Request Payload	Response Payload	Notes
Market	EiRequestMarketCharacteristcs	EiReplyMarketCharacteristcs	Request specific Market Characteristics

8.3 Interaction Pattern for the Market Facet

An Actor interacts with a specific Market to trade a specific Product. A Market matches Tenders for all Instruments based on a given Product. The matching engine is contained within the Market and different matching engines have no visibility past the Market Facet.

The Market Facet enables a Party to request the details of a Marketplace. Using the Market Facet, Parties MAY be able to request and compare Market Contexts to select which markets to participate in.

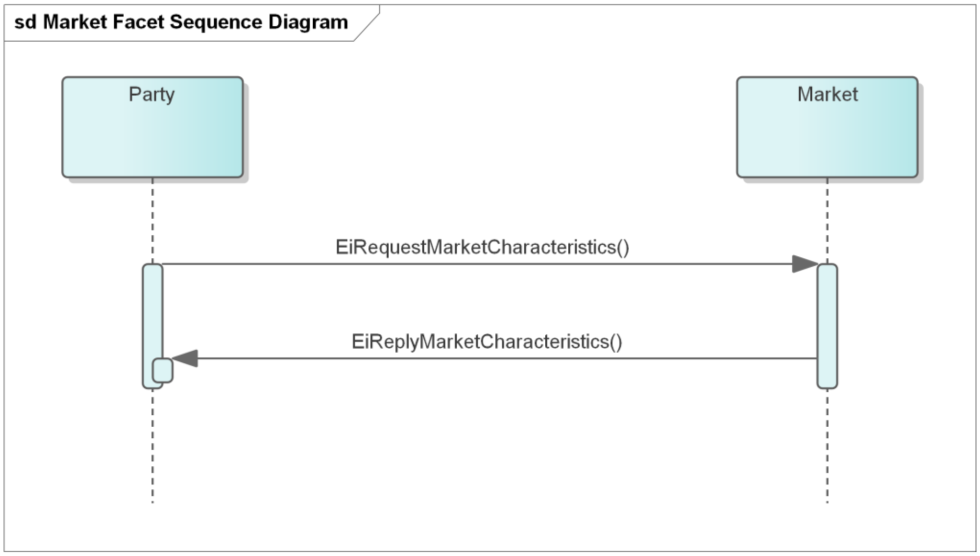


Figure 8-1: UML Sequence Diagram for the Market Facet. TODO: Update for Market Facet

The Market Facet can be used to retrieve the standard terms associated with a Market.

An EiRequestMarketCharacteristics payload requests the standard terms for a Market; the reply payload EiReplyMarketCharacteristics returns those terms as name-value pairs.

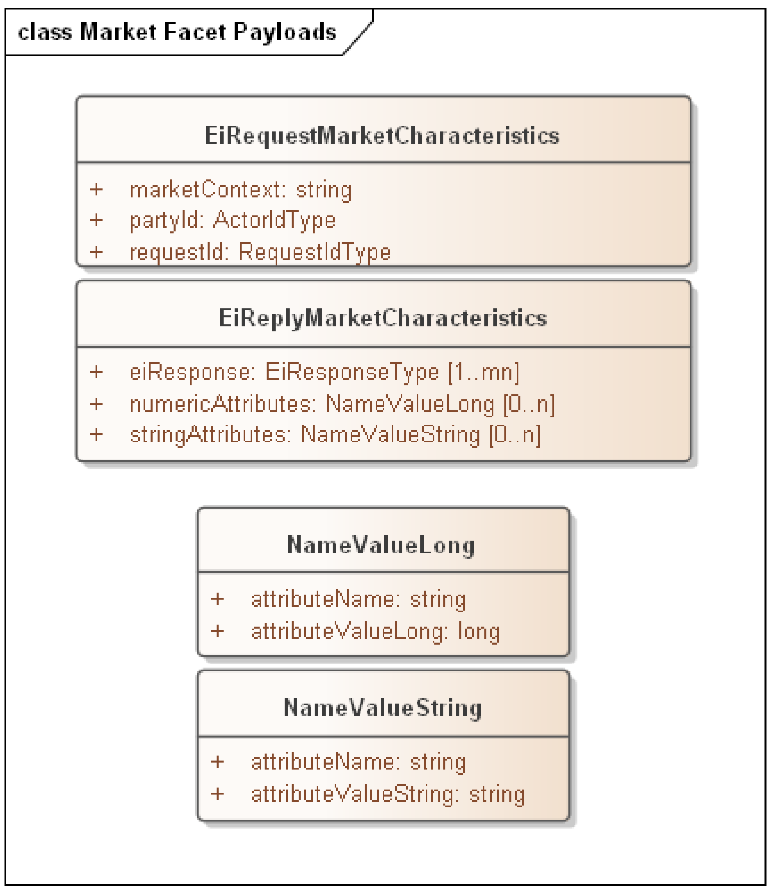


Figure 8-2: UML of Market Facet payloads

8.4 Information Model for the Market Facet

Sending an EiRequestMarketCharacteristics payload referencing a Market requests standard terms as given in Table 8-1: Standard Terms .

These are derived and extended from EMIX Terms; those are extrinsic to the Product delivery but effect how each Party interacts with others. Terms may be tied to basic operational needs, or schedules of availability, or limits on bids and prices acceptability.

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

Strings returned for attribute values MUST be no longer than 256 bytes.

TODO Consider requiring resolvable URIs for contexts, or pairing with potential transport endpoints.

Table 8-1: Standard Terms returned by Market Facet

Attribute	Attribute Name	Attribute Type	Meaning
Market Name	NAME	String	Text providing a descriptive name for a Market. While the Name MAY be displayed in a user interface; it is not meaningful to the Actors.
Currency	CURRENCY	String	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	Long	A Duration that some Marketplaces MAY use to describe trading where a first interval is not on an hourly boundary.[12]
Time Zone	TZ	String	A Time Zone indicates how all Times and Dates are expressed.
Price Decimal Fraction Digits	PRICE_FRAC	Long	Some market implementations use a Marketplace-wide indication of how many decimal fraction digits are used.[13]
Market Party ID	MPARTYID	String	The PartyID to use in a Tender (reference 2.2.3)

[12] 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 Market could use an offset duration of 3 minutes to indicate that all tenders are based on three minutes after the hour.

[13] Integer operations are typically much more efficient than fixed or floating point, so it is likely to be much faster to apply decimal shift on input and output rather than for more frequent comparison operations in the matching engine implementation

Attribute	Attribute Name	Attribute Type	Meaning
Bilateral OK	BILATERAL OK	Long	Boolean expressed as an integer: 0 - False—bilateral Tenders or Transactions not permitted, only Market Tenders 1 - True—bilateral Tenders or Transactions with identified parties are permitted.
Resource Designator	R_ID	Resource Designator	[Extensible] enumeration indicating the Resource traded in this Market. This establishes the Resource Designator used in Product definitions and in messages
Containing Marketplace	MPLACE	String	URI for Marketplace Context
Product	PRODUCT	Array of Ordered Pairs	See Product Definition, Table 8-2: Elements that define Products in a Market. It SHALL match the Product Definition indicated in the Marketplace for this Market.
Tender Grouping	TGROUP	Long	Enumeration expressed as an integer for treatment of multiple tenders either in a single EiCreateTender payload or across all Tenders for the same Instrument: 0 – Tenders are independent (*JBOT*) 1 – All Tenders within a single EiCreateTenderPayload SHALL BE treated by the market as points on a supply or demand curve as indicated by the Side of the Tenders (ALLINPAYLOAD)
Clearing Approach	CLEAR	Long	Enumeration expressed as an integer to describe market clearing approach: 0 – CONTINUOUS – continuous clearing 1 – PERIODIC – not continuous, typically with periodicity related to Product T_GRAIN.
Clearing Duration	CLEAR-DURATION	String	Duration before Instrument start time used for market matches. Only valid if Clearing Approach is 1 – PERIODIC Not valid if Periodic Time is specified. *"All matches on the hourly market are made 30 minutes before the hour"*

Attribute	Attribute Name	Attribute Type	Meaning
Clearing Time	CLEAR-TIME	String	Time for market matches. Only valid if Clearing Approach is 1 – PERIODIC Not valid if Periodic Duration is specified. *"All matches on the Day-Ahead hourly market are made at 3:00 PM".Table 9-1*
Clearing Days Ahead	CLEAR-DA	Long	Number of days prior to the Instruments for the CLEAR-TIME. For example if a Two-day ahead market clears at 3pm, CLEAR-DA = 2 and CLEAR-TIME = "3:00PM"
All transactions for an Instrument at the same clearing price	ALL_AT_CLEAR	Long	Boolean expressed as integer 0 - False—Tenders for a specific Instrument MAY clear at different prices.[14] 1 - True—As in Double Auction, all participants clear at the same price.
Maximum	MAX	Integer	Maximum Transaction size the Market will accept.
Quote Reference	QUOTE-REF	String	A string indicating the Quote Reference for this Market to which an actor may subscribe or unsubscribe.
Ticker Reference	TICKER-REF	String	A string indicating the Ticker Reference for this Market to which an actor may subscribe or unsubscribe.

Each Product in a Marketplace is defined using attributes as below

Table 8-2: Elements that define Products in a Market

Attribute	Attribute Name	Meaning
Resource Designator	R_ID	[Extensible] enumeration indicating the required Resource
Time Granularity	T_GRAIN	The interval duration in seconds for the specific Product definition

[14] A sophisticated Party may change its Bidding strategy based on this Market Characteristic. For example, in a Double Auction all negative tenders clear at a market price that may be positive, so a negative bid in a market with positive prices ensures that the bidder is "in the money", so negative bids are a realistic strategy.

Attribute	Attribute Name	Meaning
Quantity Scale	Q_SCALE	The exponent of the Quantity. For example, a Product denominated in kilowatts has a Q_SCALE of 3.
Quantity Granularity	Q_GRAIN	The allowed quantity unit size, e.g. Q_GRAIN == 10 means that a Tender for 9 units will be rejected but any multiple of 10 will be accepted.
Price Granularity	PRICE_GRAIN	The allowed price unit, e.g. Price Granularity == 10 means that that any multiple of 10 CURRENCY units is acceptable, but any price not matching, say a price of 9 CURRENCY units, is rejected. May be negative as in -3, Prices are multiples of .001.
Market	MARKET	The message endpoint to access the market where this Product is traded.
Warrants	WARRANT	Optional further specificity of Product

9 Tender Facet

The terminology of this section is that of business agreements: Tender and Transaction. The Service descriptions and payloads are simplified and updated from those defined in EI.

9.1 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 the Tender Facet (and including EiDistributeTender), with payloads shown in Table 9-1.

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 9-1: Pre-Transaction Tender Facets

Facet	Request Payload	Response Payload	Notes
EiTender	EiCreateTender	EiCreatedTender	Send a CTS-Stream of one or more Tenders. Create and emit Request Payload
EiTender	EiCancelTender	EiCanceledTender	Cancel one or more Tenders
EiTender	EiDistributeTender	None	Describe a list of Tenders to be notified to a set of parties

9.2 Interaction Patterns for the Tender Facet

Figure 9-1 presents the [UML] sequence diagram for the EiTender Facet. Note that EiDistributeTender is not part of CTS 1.0 at present, but is being considered for a future release.

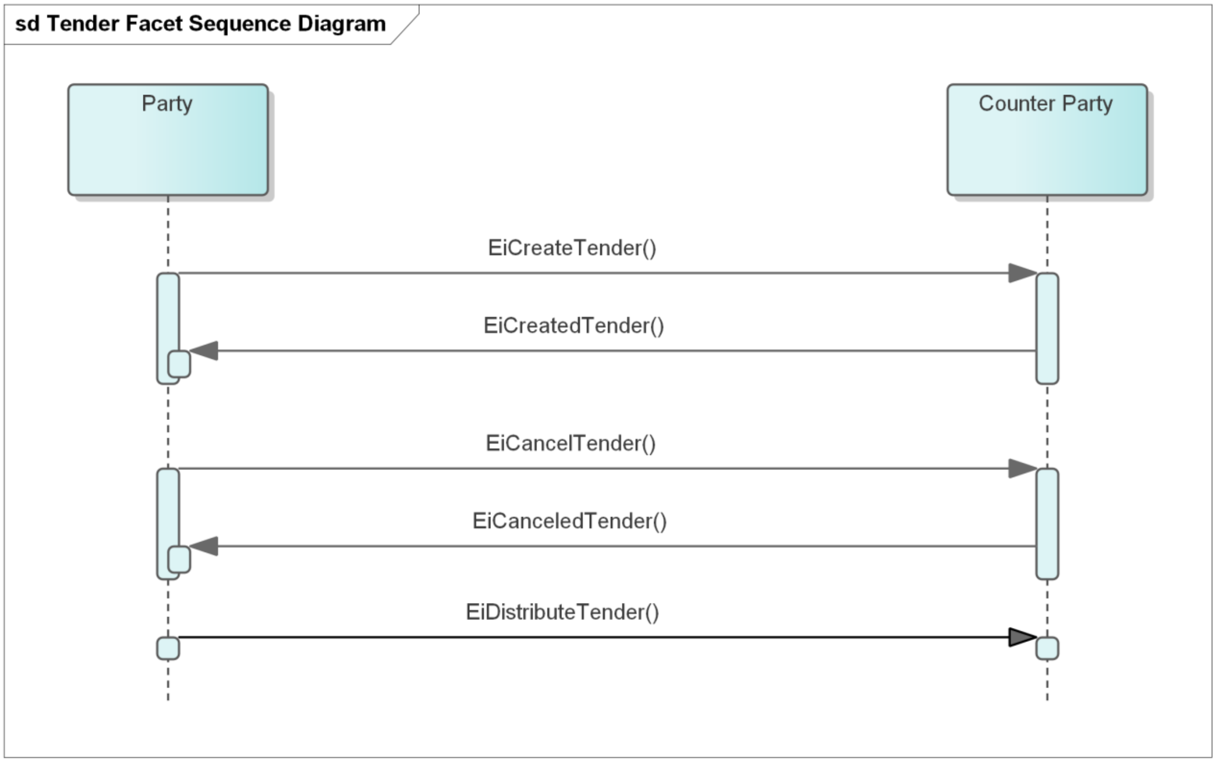


Figure 9-1: UML Sequence Diagram for the Tender Facet

9.3 Information Model for the Tender Facet

The information model for the EiTender Facet artifacts follows that of **[EMIX]**, but flattened and with Product definition implied by the implementation. See Section Payloads for the Tender Facet below.

Start time, price, and quantity are key elements for an Instrument offering to buy or sell a Product. The other aspects of Product definition (e.g. Resource, units, and duration) are described in Section 3.2.

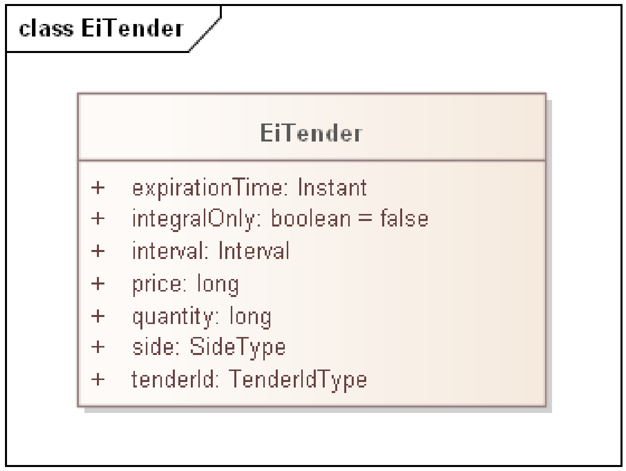


Figure 9-2: Class EiTender

Table 9-2: EiTender 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 Tendered	Total price is the product of price and quantity. Note that price is subject to the Price Decimal Fraction value. See Scale and Granularity constraints in Section 8, “Market Facet”
Quantity	The quantity of the Product being Tendered	Total price is the product of price and quantity suitably scaled
Side	Whether The Tender is to buy or to sell the Product	
Tender ID	An ID for this Tender	

9.4 Payloads for the Tender Facet

The **[UML]** class diagram describes the payloads for the Tender Facet operations.

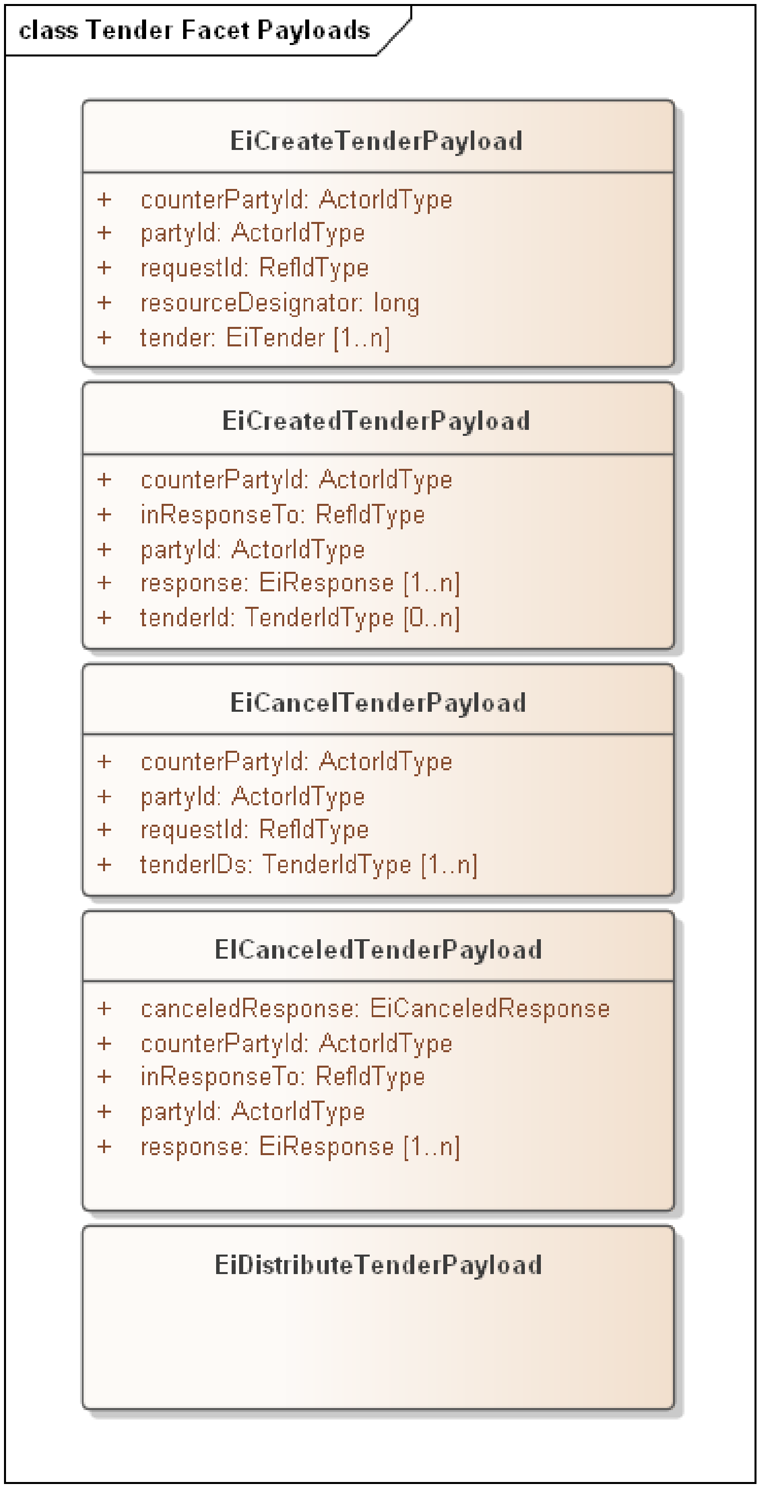


Figure 9-3: UML Class Diagram for the Operation Payloads for the Tender Facet

The following table describes the attributes for EiCreateTenderPayload

Table 9-3 EiCreateTenderPayload Attributes

Attribute	Meaning	Notes
Counter Party ID	The Actor ID for the CounterParty for which the Tender is created.	This is most frequently the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific Party is used.
Party ID	The Actor ID for the Party on whose behalf this Tender is made.	Indicates which Actor proposes the buy or sell side EiCreateTender.
EiTender	One or more EiTenders to be created.	In CTS an object describing a Tender is instantiated then sent; the latter is a consequence of processing an EiCreateTender payload.
Resource Designator	The Resource being tendered	Must match the Market Resource Designator on receipt at the Market
Request ID	A reference ID which identifies the artifact or message element. The Request ID uniquely identifies this request, and can serve as a messaging correlation ID[15].	
Responses	Responses for each attempted EiTender creation	Array Of Responses [EI]

EiCreateTenderPayload with more than one EiTender SHALL be treated as a shorthand for sending each EiTender in a separate payload.

See Information Model for the Market Facet. Note that if more than one EiTender is included in a single EiCreateTenderPayload, and the Market Characteristics include TGROUP = JBOT , there is no implication that there be an all or none meaning. This avoids the complexity of database-style transaction processing consistency, and simplifies implementations.

If TGROUP is ALLINPAYLOAD then all Tenders in the EiCreateTenderPayload SHALL be taken by the market to represent a supply or demand curve

TODO: Add example of how different components in a building can create their own bid curves independently.

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

10 Transaction Facet

10.1 Transaction Services

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

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

here is no CTS payload that permits Canceling or modifying Transactions; given disparate ownership techniques from Distributed Agreement Protocols SHOULD be applied is not permitted.[16]

Table 10-1: Transaction Management Service

Service	Request Payload	Response Payload	Notes
EiTransaction	EiCreateTransaction	EiCreatedTransaction	Create and acknowledge creation of a Transaction

10.2 Interaction Pattern for the Transaction Facet

This is the [UML] sequence diagram or the EiTransaction Facet:

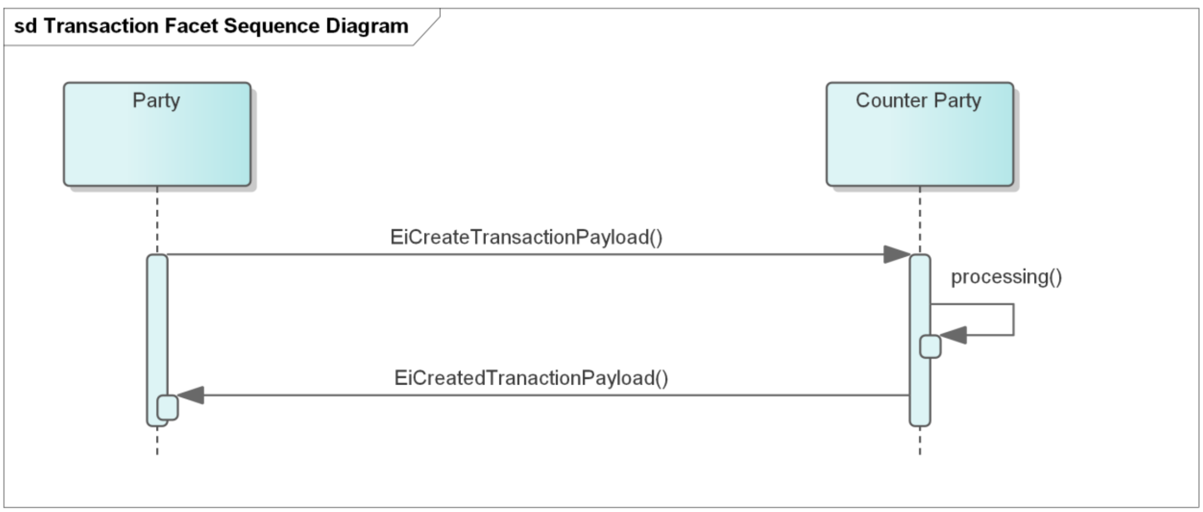


Figure 10-1: UML Sequence Diagram for the EiTransaction Facet

A transaction may be mediated by a market, in which case an EiCreateTransactionPayload is sent to each of the matched Parties.

10.3 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.

The EiTransaction object includes the original EiTender, possibly rewritten to reflect the clearing price and quantity.

---

[16] Following the approach of distributed agreement protocols, compensating tenders and transactions SHOULD be created as needed to compensate for any effects. This is consistent with the way that distributed agreement protocols such as [WS-BusinessActivity] manage compensation rather than cancelation.

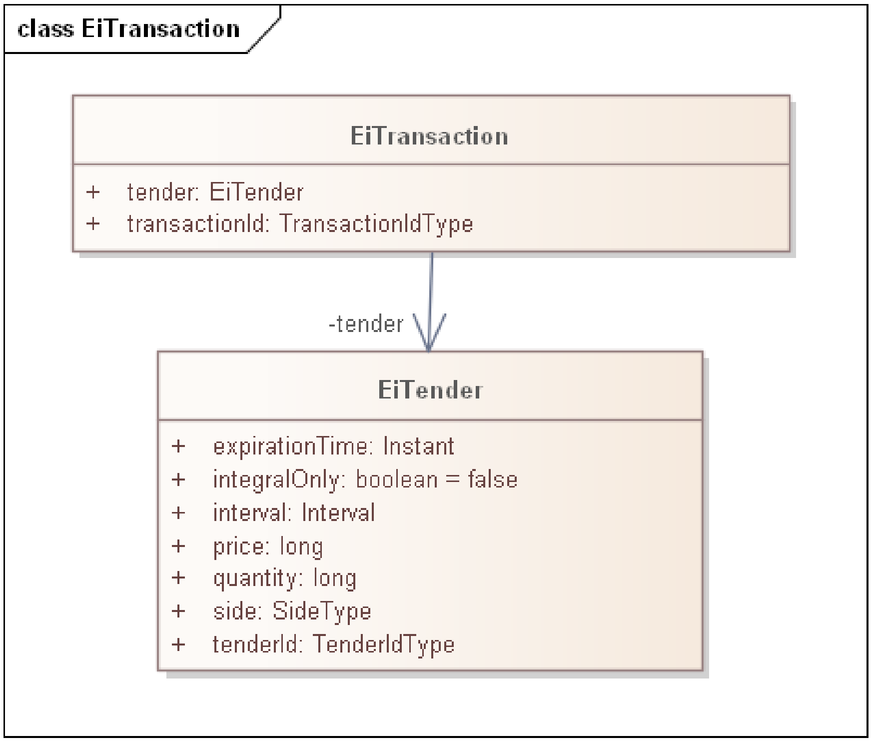


Figure 10-2: UML Class Diagram of EiTransaction

The attributes of EiTransaction are shown in the following table.

Table 10-2: 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 Tender Id

10.4 Payloads for the Transaction Facet

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

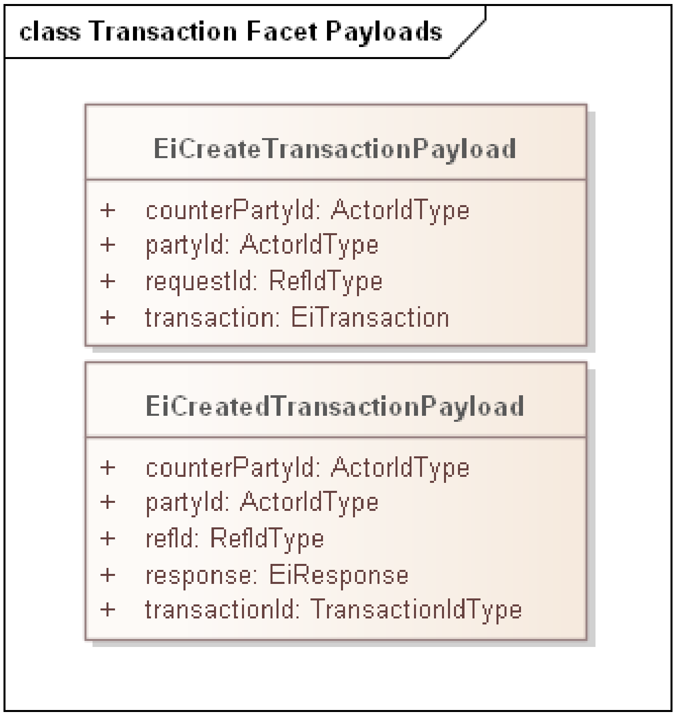
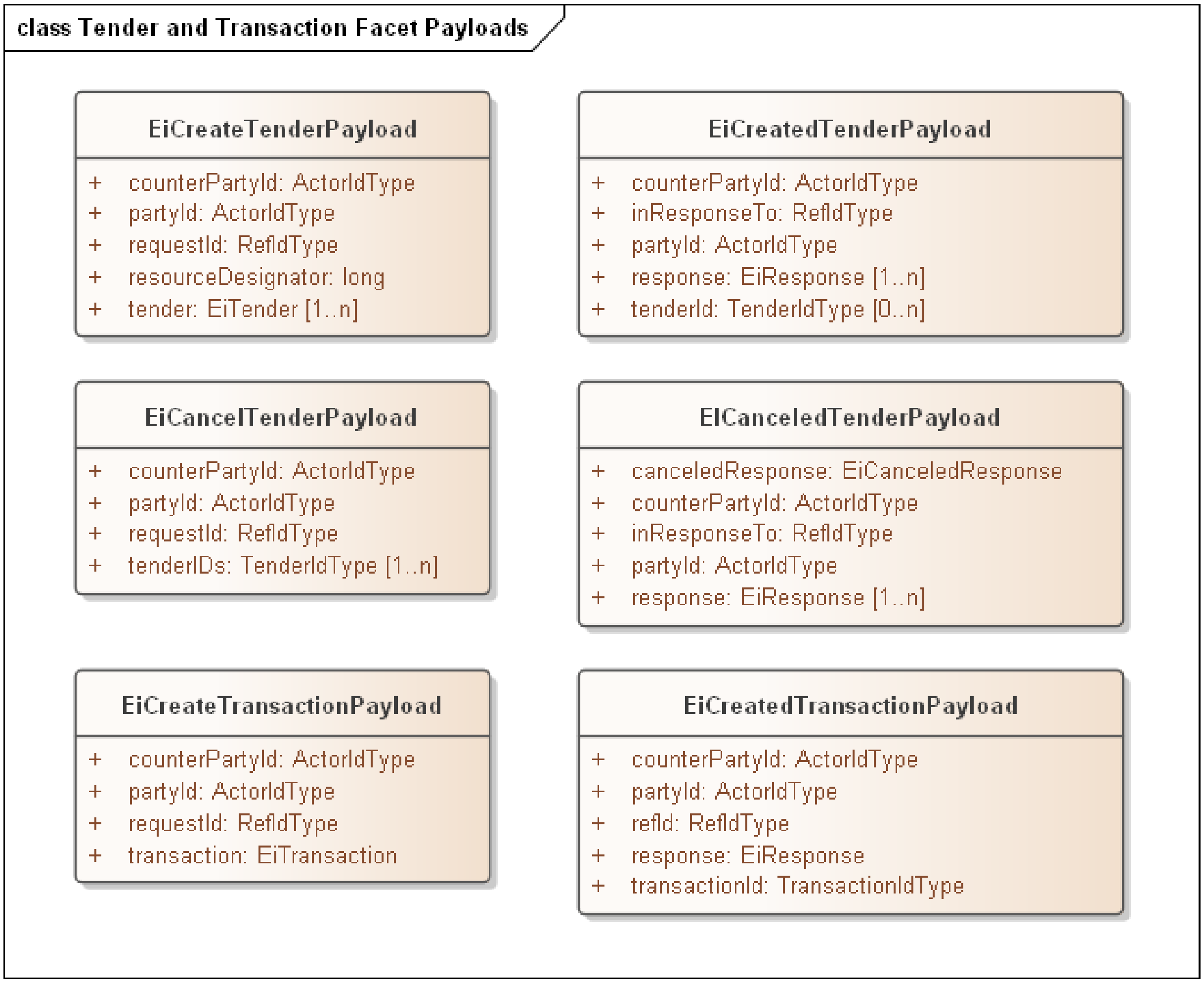


Figure 10-3: UML Class Diagram of EiTransaction Facet Operation Payloads

10.5 Comparison of Transactive Payloads

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

Figure 10-4: UML Diagram comparing Tender and Transaction Facet Payloads



While most transactions originate as Tenders submitted to the Market, matched by the Market, and the resule in a Transaction created by the Market, there are use cases for bilateral actions that generate a Transaction that did not come through the market.

For example, two parties within a market may choose to transact directly. A party may opt to buy directly from his neighbor's solar power. Another market may permit Charity, that is an anonymous donation to the Position of a neighbor. In either case, the Transaction is to the Market so that each Party's Position is maintained and so that the Buyer does not get double billed. These transactions are referred to as over-the-counter (OTC) agreements.

OTC Agreements can span multiple instruments.

The parties must wait for the Market's acknowledgment and approval before they proceed with the delivery. This acknowledgment is by an EiTransactionCreated message to each Party.

[The Committee is most interested in comments on OTC Transactions, and the potential errors that would prevent them being registered. It is possible that EiRegisterTransaction and EiTransactionAcknowledged is a correct interaction.]

11 Position Facet

11.1 Introduction

The purpose of the Position Facet is to allow access to the accumulated position for actors.

Roles in using the Position Facet include

- The Actor whose position is being requested—the *position Party*
- An Actor who is authorized to request position information for other actors—including but not limited to an auditor—the *requestor*
- The Market and Product for which the Position is being requested.

11.2 Position Definition

A Party's ***Position*** for a time period is the algebraic sum of committed supply or sale typically represented as purchases and sales expressed by means of EiCreateTransaction payloads for that instrument and Party.[17]

The time period for position intervals SHOULD be the same as for the underlying market used to buy and sell, but need not be; conversion of differing time granularity is programmatic and not required by this specification.

A Party needs to know both

- The Party's projected needs for a time interval (not in scope)
- The Party's committed net inflow and outflow for the interval

Note that committed inflow and outflow may be outside a market, e.g. local generation or battery interaction.

An Actor may, with appropriate authorization, request positions for other parties. This permits the specification and implementation of an auditor Actor.

An Actor sees its own Tenders and Transactions, and can maintain its own position. This facet allows the offloading of that data management, but could in fact be a request to a local Position manager.

11.3 Interaction Pattern for the Position Facet

Table 11-1: Position Facet

Facet	Request Payload	Response Payload	Notes
Position	EiRequestPosition	EiReplyPosition	Request an Actor's Position(s) for a specific time interval, and reply with those Position(s) if access is authorized.

This is the [UML] sequence diagram for the Position Facet:

[17] One may say that a Party's position for an Instrument is evolved from an accumulation of trades for that Instrument.

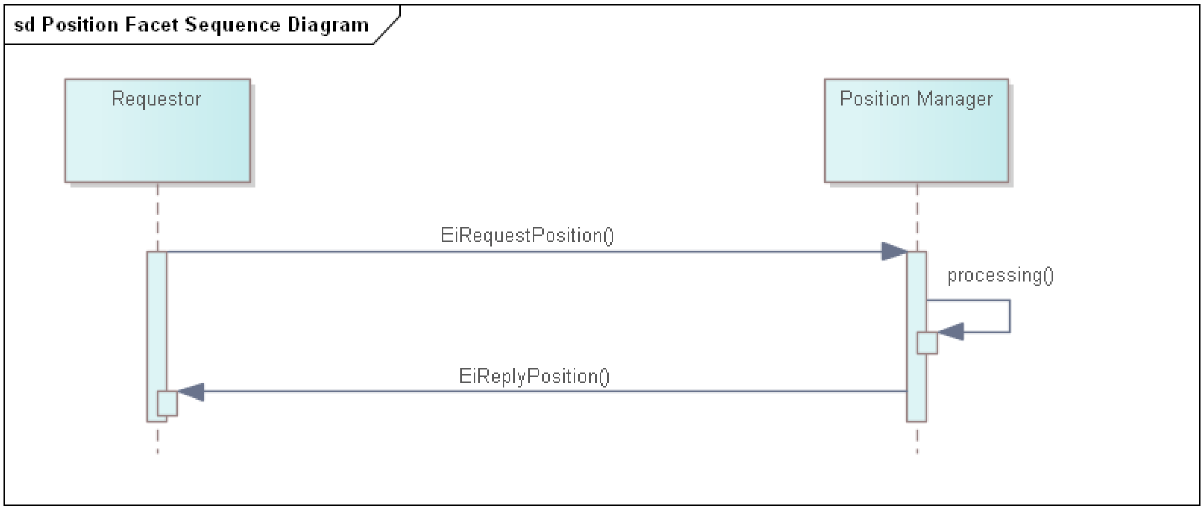


Figure 11-1: UML Sequence Diagram for the Position Facet

11.4 Information Model for the Position Facet

For Position, a bounding interval is specified and the position in each interval contained in the closed bounding interval is returned. An Actor has a position in a Product, and a Product specifies a temporal granularity or Interval duration. This Product duration defines the Interval duration for the returned CTS Stream. All elements of the stream share the duration and the stream has an explicitly stated start time.

A position is concerned with the total amount under contract, not the prices. If an Actor has positions in more than one Product, say, in a one-hour Product and in a one-minute Product, then that requires two requests for position, and the two replies have different interval durations. The integration of these two Positions into a single combined Position is the responsibility of the Requestor.

The attributes are shown in the following section.

11.5 Payloads for the Position Facet

The Position payload is in the format of a CTS Stream, with only a Quantity in the Interval Payload.
TODO: discuss overlapping positions, as in 1 Hour position overlaid with a 5 minute position.

The **[UML]** class diagram describes the payloads for the Position facet.

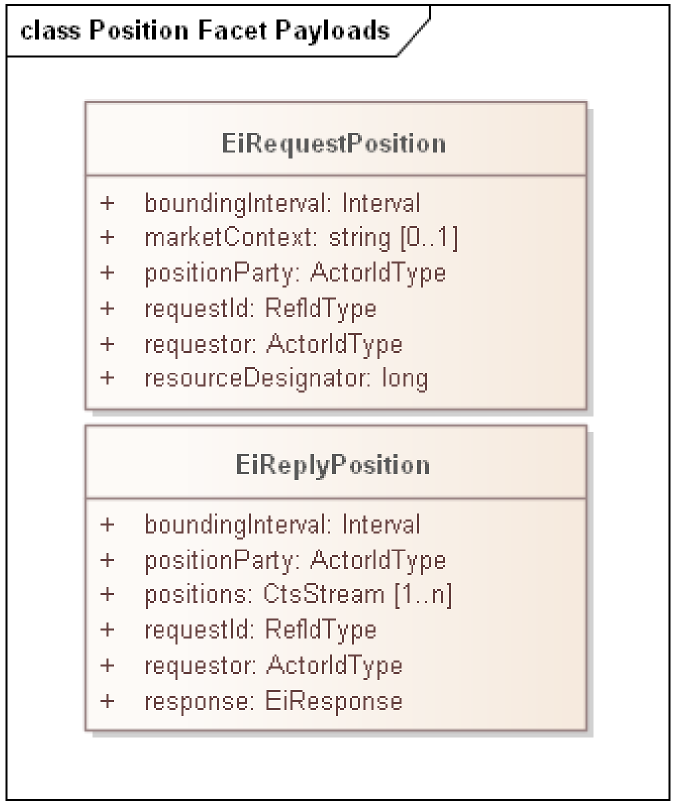


Figure 11-2: UML Class Diagram of Payloads for the Position Facet

Table 11-2: Attributes of Position Facet Payloads

Attribute	Meaning	Notes
Bounding Interval	The [closed] time interval for which position information is requested. The first Positions Stream element starts at or after the start of the Bounding Interval. The last Stream element ends at or before the start of the Bounding Interval.	
Position Party	The Party whose position is being requested.	Allows a request for another Party's position, with appropriate privacy and security constraints

Attribute	Meaning	Notes
Market Context	The market context of interest	Used to determine the Resource for position. If not present, any Resource of which the responder is aware, with no claim to completeness, will be used
Request ID	A reference to this payload	May be used as a correlation ID
Requestor	The Party requesting the position.	A failure indication will be returned if the Requestor is not authorized to access position information for Position Party. Addresses the auditor use case.
Positions	CTS Streams containing the positions for Position Party for each Resource. Positions are signed or zero.	Each CTS Stream interval that is contained within the Bounding Interval will have a value associated (signed integer, zero permitted). Note that a CTS Stream contains a Resource Designator
Response	An EiResponse. Will indicate failure if Requestor is not authorized to access position information for Position Party for any of the requested intervals.	

The following system-specific requirements are out of scope:

- Different systems may support Position requests for different purposes. An Actor MAY request its own position(s) to recover from failure.
- Positions MAY be used to compute Actor reliability.
- A supplier of last resort MAY compare Positions to Delivery to impute transactions for unpurchased power delivered. (See 12 Delivery Facet)

12 Delivery Facet

The CTS Delivery Facet can be considered as the telemetry facet. We term it “Delivery” to align with the market focus of this specification, i.e., a building takes delivery of power or a distributed energy Resource (DER) delivers power. A CTS Delivery payload contains a CTS Stream that conveys the flow of a specific Resource through a particular point on the Product’s delivery network between particular times.

CTS Delivery is typically derived from reading one or more meters, but it may be computed, implied or derived from some other method. Every contract is between a Party that promises to buy and a Party that promises to sell. Consider an actor that performs temporal arbitrage, i.e., buys one-hour Products and sells one-minute Products during the same hour. The Actor MAY report that it took delivery in each minute of that Interval, and the sales to other Actors MAY be visible only as reductions as recorded in Delivery.

In most TE markets, a node that takes delivery of more power or other Resource during an Interval than contracted for must eventually pay for that delivery. For example, An auditor, however defined, could sum all positions (See section 11, *Position* Facet) and compare the result to Delivery. The Auditor can then impute a transaction for the over-delivery. This may not be a simple “spot price”; if multiple Actors are taking over-delivery, then the last small transaction is likely underpriced. Systems that track “actor reputation” may lower the reputation score. These examples explain the potential use of the information delivered by this facet, and are not meant not to dictate any particular business process or system model.

A CTS Delivery payload reports on the flow of a Resource because the temporal granularity MAY not match that of any particular Product. The payload may (e.g.) report the sum of a one-hour market and of a one-minute market for the same Resource.

A CTS Marketplace MAY have expectations about levelized load—as do many of today’s tariffed markets. Exceeding the limiting bounds for Delivery may result in a market penalty. It is outside the scope of this specification to define the bounds or the nature of the penalty.

A request for delivery specifies a Resource, physical granularity, and temporal granularity. While the physical granularity and temporal granularity need to be within the capabilities of the telemetry node, they need not match any particular Product.]

12.1 Interaction Pattern for the Delivery Facet

Table 12-1: Delivery Facet

Facet	Request Payload	Response Payload	Notes
Delivery	EiRequestDelivery	EiReplyDelivery	Request Delivery through a specific Measurement Point

Figure 4-1 is the [UML] sequence diagram for the Delivery Facet:

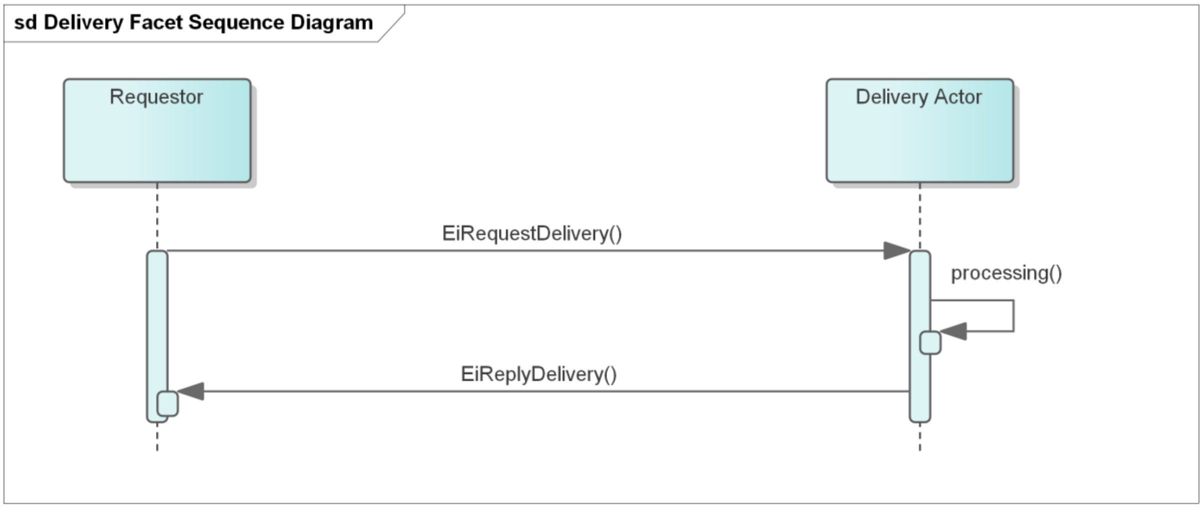


Figure 12-1: UML Sequence Diagram for the Delivery Facet

12.2 Information Model for the Delivery Facet

A Delivery response returns a single CTS Stream of intervals of the requested Duration, with a quantity in each.

As with the Position Facet a bounding interval is specified and the delivery in each interval contained in the closed bounding interval is returned. The granularity as requested MAY not be available, or the Delivery Actor may convert and combine—for example a request for one hour delivery intervals could be responded to using information from 1 minute or 5-minute measurement cycles.

The attributes are shown in the following section.

12.3 Payloads for the Delivery Facet

The **[UML]** class diagram describes the payloads for the Delivery facet.

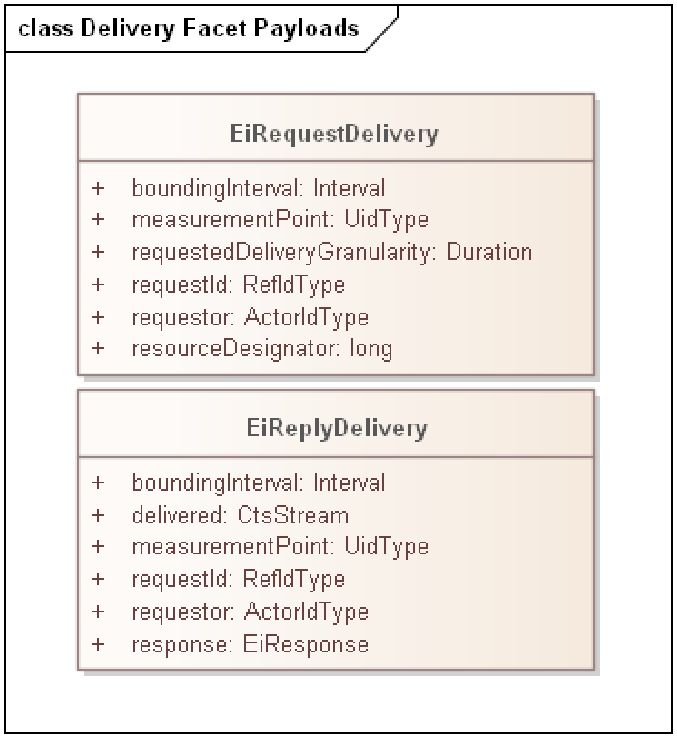


Figure 12-2: UML Class Diagram of Payloads for the Position Facet

Table 12-2: Attributes of Delivery Facet Payloads

Attribute	Meaning	Notes
Bounding Interval	The [closed] time interval for which position information is requested. The first Positions Stream element starts at or after the start of the Bounding Interval. The last Stream element ends at or before the start of the Bounding Interval.	
Measurement Point	The Point for which telemetry is provided about the flow of the resources..	Allows a request to any Measurement Point for information on Resource flow at that point over time. Information should be secured in conformance with appropriate privacy and security constraints
Request ID	A reference to this payload	May be used as a correlation ID

Attribute	Meaning	Notes
Requestor	The Party requesting the position.	A failure indication will be returned if the Requestor is not authorized to access position information for Position Party. Addresses the auditor use case.
Delivered	A CTS Stream containing the Delivery information for the Resource. Delivery value is signed or zero.	Each CTS Stream interval that is contained within the Bounding Interval will have a value associated (signed integer, zero permitted). Note that a CTS Stream contains a Resource Designator which SHOULD match that in the requested Resource Designator
Response	An EiResponse. Will indicate failure if Requestor is not authorized to access position information for Position Party for any of the requested intervals. If the Requested Delivery Granularity cannot be used, MAY indicate what granularity can be used.	

13 Market Information—the Quote and Ticker Facets

Tenders are typically private in a market, whether the market matches tenders using an order book, a double auction, or some other means to match buyer and seller to award contracts. Markets generate order by enabling price knowledge to emerge from the tenders of independent actors. If all tenders are public, then this price cannot emerge. No seller would ever offer a price less than the highest outstanding tender to buy; no buyer would ever offer a price higher than the lowest outstanding tender to sell. Moreover, analysis of tenders can reveal detailed information about the market participant beyond that necessary to balance supply and demand. (See *Appendix B.2*, *CTS and Privacy Considerations.*)

Even so, some Actors may wish to advertise specific Tenders. In a transitional environment, a utility may wish to publish day ahead prices for each hour of the day. An Actor may wish to draw others into the market quickly in response to a system failure or unplanned-for need—and may offer an unusually high or low price to attract sellers or buyers. Others may wish to quickly dispose of a previous position. A distribution operator in TE markets may wish to advertise short term deals temporal price boundaries to protect grid components by smoothly ramping power delivery requirements. Whatever the reason, [EI] specifies the EiQuote service for advertising Tenders.

Transaction prices are public information. Consider a financial market, which lists the current stock price, or more precisely, the price of the last transaction. Parties use this public information to plan whether to submit new Tenders, or perhaps to cancel old Tenders. An old technology for broadcasting financial transactions was the ticker producing ticker tape, so named for the sound it made with each transmission. CTS uses the term “Ticker” for the completed transaction information.

The information payloads for Quotes and Tickers are nearly identical.

For each Market, there MAY be URI for the Ticker service and for the Quote service. The Marketplace Characteristics MAY include a URI for the Ticker service and a URI for the Quote service.

This specification has no position on the whether a common Ticker for all Markets for a given Resource is better than a Ticker for each Market. Similarly, a Marketplace MAY choose to put all “Green” (however defined) Products in one Ticker stream and “conventional” in another. Any number of Markets within a Marketplace MAY use the same URI for the ticker service, and other Markets share another. Such decisions are left up to the developers of CTS-based systems.

Quotes use the same mechanisms, and like Tickers may have many Markets or a single Market in a single Quote service.

13.1 Quotes

[EI] defines a quotation as a market price or possible price, which does not replace the Tender and acceptance to reach a Transaction. The Quote message looks very much like a Tender.

As noted above in Section 9, the Tender Facet, a Party may wish to advertise certain of its Tenders to the market. An advertisement of an attractive price for limited amount of power might only be available to the first to respond Such a public Tender is distributed to other Parties by the Quote Facet by including a Tender ID for the parallel Tender.

Publish-Subscribe semantics are a likely communication paradigm.

13.1.1 Interaction Pattern for the Quote Facet

Table 13-1: Quote Facet

Facet	Request Payload	Response Payload	Notes
Quote	EiSubscribeQuote	EiSubscribedQuote	As multiple Markets may use same Quote service, must tolerate multiple subscriptions.

Facet	Request Payload	Response Payload	Notes
Quote	EiUnsubscribeQuote	EiUnsubscribedQuote	Unsubscribe for all Markets on this facet.
Quote	EiDistributeQuote	None	Post to a Quote endpoint.

See Figure 13-1 for the UML sequence diagram for the Quote Facet:

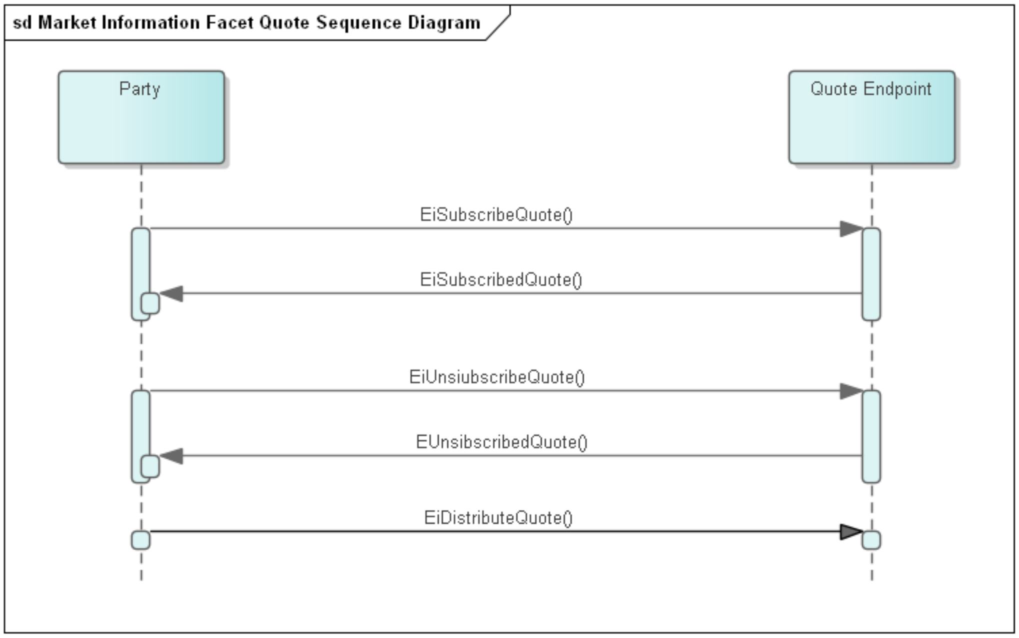


Figure 13-1: UML Sequence Diagram for the Quote Facet

The Quote Service does not wait for or expect acknowledgements of distributed Quotes.

13.1.2 Information Model for the Quote Facet

The **[UML]** class diagram describes the information model for EiQuote for the Quote Facet. The diagram includes an informative class diagram of EiTender for comparison.

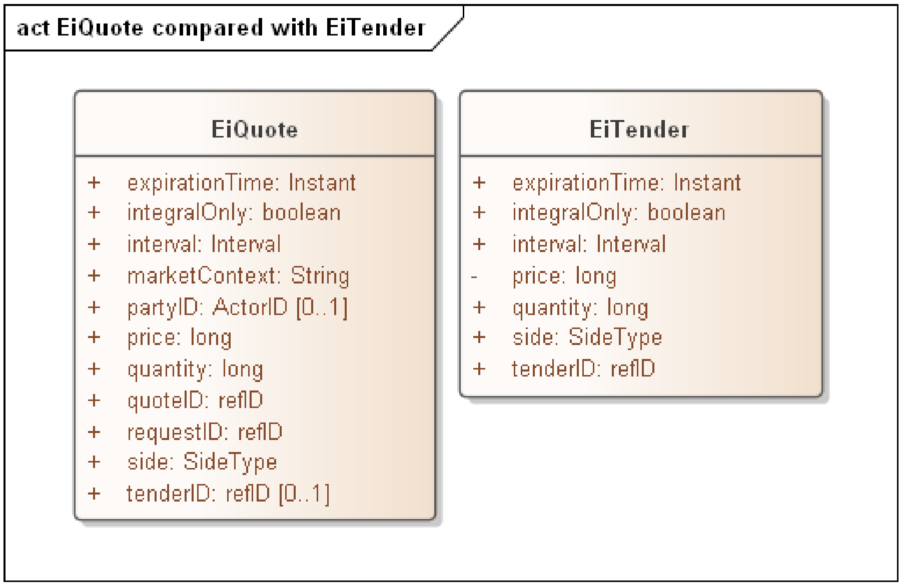


Figure 13-2: UML Class Diagram of EiQuote

The following table details the attributes of the EiQuote class.

Table 13-2:EiQuote Attributes

Attribute	Meaning	Notes
Expiration Time	The date and time after which this Quote is no longer valid.	If an advertised Tender, the expiration time for the Tender.
Integral Only	All of the Quote must be bought or sold at once; no partial sale or purchase	Useful for advertisement of Tenders. In CTS Integral Only is conformed to False.
Interval	The time interval for the Product being offered	The Resource Designator is that from the Market.
Party ID	Identifies the Party making the Quote	See Appendix B.2, CTS and Privacy Considerations. Optional.
Price	The unit price for the Product being Quoted	Total price is the product of price and quantity. Note that price is subject to the Price Decimal Fraction value for the Market. See Scale and Granularity Constraints in Section 8, “Market Facet”
Quantity	The quantity of the Product being Tendered	Total price is the product of price and quantity suitably scaled
Side	Whether the Quote is to buy or to sell the Product	
Quote ID	An ID for this Quote	
Tender ID	ID for the Tender being advertised, if any.	Optional. If present MAY claim that a Tender has been submitted to the Market, in effect advertising that Tender.
Market Context	The market context for which this is a quote	The Quote Reference is a Market Characteristic.

An Actor may submit quotes for several consecutive Intervals, a set of Instruments for an identical Product. An example is a load serving entity quoting 24 prices for the next day. All elements of the stream share the duration and the stream has the explicitly stated start time.[18]

[18] Integration of CtsStreams is pending. TODO

13.1.3 Payloads for the Quote Facet

Conceptually the Quote streams may be considered implemented as Publish-Subscribe streams (Pub-Sub).

The Market Characteristics provide a Quote Reference for a Market by requesting the QUOTE-REF characteristic.

The mechanism and setup for subscribing is out of scope, as is the mechanism for publishing. The payloads are as follows:

- EiSubscribeQuote – PartyID for the sender and the Quote Reference for the market as found in Market Characteristics
- EiSubscribedQuote – EiResponse indicating success or failure. Implementations MAY send information related to the Subscribed stream.
- EiUnsubscribeQuote – Removes the subscription to the indicated Quote Reference
- EiUnsubscribedQuote – EiResponse indicating success or failure. Implementations MAY send information related to the now un-Subscribed stream.
- EiDistributeQuote – Publish an EiQuote to the Quote Reference.

13.2 Tickers

Ticker interactions and payloads are nearly identical to those for Quotes. Tickers are anonymized public information about Transactions submitted to the Ticker service by the Markets. The mechanism and interactions of this submission are out of scope.

13.2.1 Interaction Pattern for the Ticker Facet

Table 13-3: Ticker Facet

Facet	Request Payload	Response Payload	Notes
Ticker	EiSubscribeTicker	EiSubscribedTicker	As multiple Markets may use same Ticker service, must tolerate multiple subscriptions.
Ticker	EiUnsubscribeTicker	EiUnsubscribedTicker	Unsubscribe for all Markets on this facet.
Ticker	EiDistributeTicker	None	Publish to the Ticker Reference

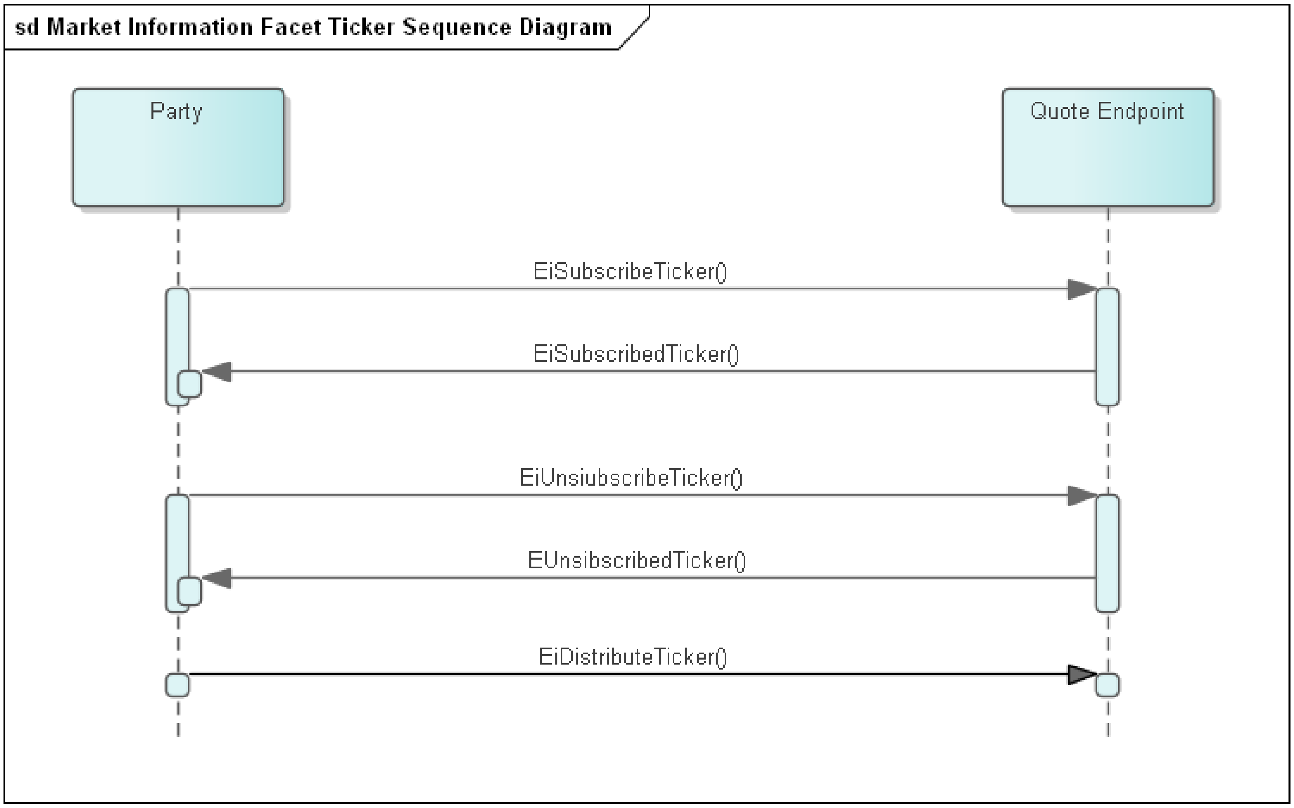


Figure 13-3: UML Sequence Diagram for the Ticker Facet

13.2.2 Information Model for the Ticker Facet

The **[UML]** class diagram describes the information model for EiTicker for the Ticker Facet. The diagram includes an informative class diagram of EiQuote for comparison.

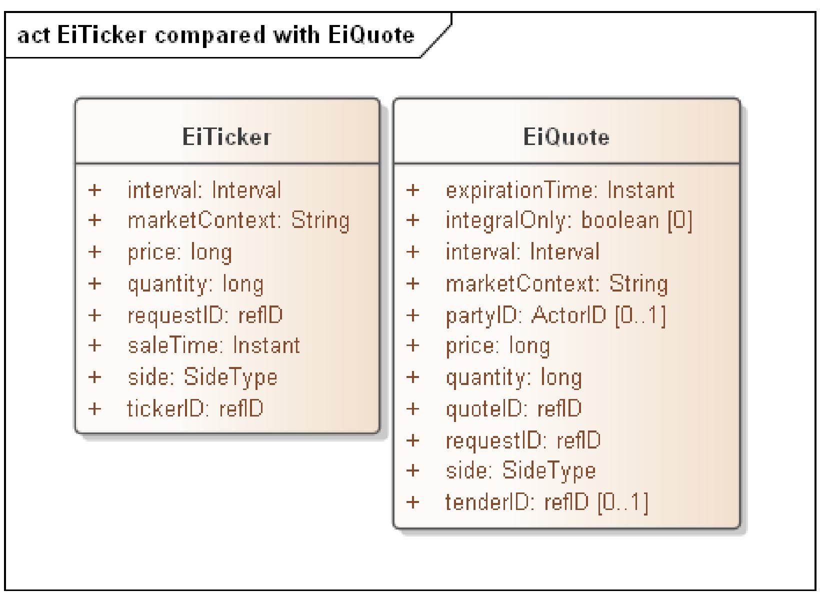


Figure 13-4: UML Class Diagram of EiTicker compared with EiQiuote

The following table details the attributes of the EiTicker class.

Table 13-4:EiTicker Attributes

Attribute	Meaning	Notes
Interval	The time interval for the Product sold	The Resource Designator is that from the Market.
Market Context	The market context for which this is a Ticker	The Ticker Reference is a Market Characteristic.
Price	The unit price for the Product sold	Total price is the product of price and quantity. Note that price is subject to the Price Decimal Fraction value for the Market. See Scale and Granularity Constraints in Section 8, “Market Facet”
Quantity	The quantity of the sold	Total price is the product of price and quantity suitably scaled
Sale Time	Timestamp indicating when the sale took place.	
Side	Whether the sale was to buy or to sell the Product.	An implementation MAY deliver only Buy side Ticker elements
Ticker ID	An ID for this Ticker	

An Actor may submit ticker instances for several consecutive Intervals, a set of Instruments for an identical Product. An example is a load serving entity quoting 24 prices for the next day. All elements of the stream share the duration and the stream has the explicitly stated start time.[19]

13.2.3 Payloads for the Ticker Facet

The **[UML]** class diagram describes the payloads for the Delivery facet.

Conceptually the Ticker streams may be considered implemented as Publish-Subscribe streams (Pub-Sub).

The Market Characteristics provide a Ticker Reference for a Market by requesting the TICKER-REF characteristic.

The mechanism and setup for subscribing is out of scope, as is the mechanism for publishing. The payloads are as follows:

- EiSubscribeTicker – PartyID for the sender and the Ticker Reference for the market as found in Market Characteristics
- EiSubscribedTicker – EiResponse indicating success or failure. Implementations MAY send information related to the Subscribed stream.

[19] Integration of CtsStreams is pending. TODO

- EiUnsubscribeTicker – Removes the subscription to the indicated Ticker Reference
- EiUnsubscribedTicker – EiResponse indicating success or failure. Implementations MAY send information related to the now un-Subscribed stream.
- EiDistributeTicker – Publish an EiTicker to the Ticker Reference.

14 Bindings

Payloads and interaction patterns are described in **[UML]** in Sections 6 through 12 above. This section contains bindings for the payloads in three encoding schemes:

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

14.1 JSON

PENDING—JSON Schema awaiting stable payload definitions

14.2 XML Schema

PENDING—XML Schema awaiting stable payload definitions

14.2.1 XML Namespaces

PENDING—XML Namespaces awaiting XML Schema

14.3 Simple Binary Encoding

TODO—SBE Schema awaiting stable payload definitions

15 Conformance

15.1 Introduction to Conformance

By design, CTS is a simplified and restricted subset profile of TeMIX. See Appendix

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

- OASIS WS-Calendar **[WS-Calendar]**
- A definition of Streams contained in **[EI]**

We normatively reference and apply the evolution of these specifications, in particular

- OASIS WS-Calendar Schedule Streams and signals **[Streams]**, simplified as CTS Streams (see CTS Streams
- The WS-Calendar **[CAL-MIN]** interval is used directly (as IntervalType).

This specification simplifies WS-Calendar Schedule Streams and Signals [Streams] as CTS Streams, and refactors the TEMIX profile of **[EI].**

Conformance of the CTS evolved specification can be shown with the techniques of **[IEC62746-10-3]** is described in informative Appendix C.

15.2 Claiming Conformance to Common Transactive Services

Implementations claim conformance to Common Transactive Services 1.0 by asserting conformance statements on the numbered items below.

1. The conformance statement MUST list all Facets which it supports in full or and in part.
2. The conformance statement MUST describe all extensions to payloads described in this specification.
3. The conformance statement MUST describe the Binding(s) which it supports along with any extensions. If the implementation does not use a standard binding as defined in Section 13, the conformance statement MUST define the binding used, at a similar level to detail to Section 13.
4. The conformance statement MUST describe how each payload definition conforms to the UML and/or profiled definitions for each payload unless it uses only standard Bindings in Section 13.
5. The conformance statement MUST indicate cardinality for message payload attributes where there is flexibility in this specification.
6. The conformance statement MUST describe any facets it defines to extend this specification.

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]

[CAL-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

[CAL-PIM]
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

[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

[EMIX] 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

[JSON]
JavaScript Object Notation and JSON Schema. https://cswr.github.io/JsonSchema/

[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>.

[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.

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.

[Fractal Microgrids]
Art Villanueva et al, *Camp Pendleton Fractal Microgrid Demonstration,* California Energy Commission Report CEC-500-2016-013,j available at http://400.sydneyplus.com/CaliforniaEnergy_SydneyEnterprise/Download.aspx?template=Books&field=PublicURL&record=57483797-a40e-49e7-b675-2858a3ad0d91&showSave=False&repeat=d4e63b56-27d1-4476-9300-7ede86a533ca

[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

[IEC62746-10-3] International Standard. Systems interface between customer energy management system and the power management system - Part 10-3: Open automated demand response - Adapting smart grid user interfaces to the IEC common information model, https://webstore.iec.ch/publication/59771 2018

[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).

[TPB-EI] Transport Protocol Bindings for OASIS Energy Interoperation 1.0 Version 1.0. 01 October 2012. OASIS Committee Specification 01.
.

[TeMIX] TeMIX Transactive Energy Market Information Exchange [TeMIX] an approved Note of the EMIX TC. Ed Cazalet et al., 23 May 2010. http://www.oasis-open.org/committees/download.php/37954/TeMIX-20100523.pdf

[TransactiveMicrogrids]
Jennifer M. Worrall, Edward G. Cazalet, PhD, William T. Cox, PhD, Narayanan Rajagopal, Thomas R. Nudell, PhD, and Paul D. Heitman, *Energy Management in Microgrid Systems,*TESC 2016. Available at http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/tes2016_microgrids_paper_Final.pdf

[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/

[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

[ZeroTrust]
Zero Trust Architecture, S Rose, O Borcher, S Mitchell, S Connelly. NIST Special Publication 800-207, https://doi.org/10.6028/NIST.SP.800-20 August 2020

Appendix B. Security and Privacy Considerations

This specification defines message payloads only. Security must be composed in. Privacy considerations must be decided when implementing specific systems for specific purposes.

B.1 CTS and Security Considerations

Procuring energy for local use and selling energy for remote use are each at the cusp of finance and operations.

- A price that is falsely low may cause the buyer to operate a system when there is inadequate power, potentially harming systems within a facility, or harming other facilities on the same circuit.
- A price that is falsely low may cause the seller to leave the market.
- A price that is falsely high may cause the buyer to shut down operation of systems or equipment.
- A price that is falsely high may cause the seller to increase operations when there is neither a ready consumer or perhaps even grid capacity to take delivery.

For these reasons, it is important that each system guard the integrity of each message, assure the identities of the sender and of the receiver, and prove whether a message was received or not.

Messages should be encrypted to prevent eavesdropping. Any node should be able to detect replay, message insertion, deletion, and modification. A system must guard against and detect man-in-the-middle" attacks wherein an intermediary node passes of messages as originating from a known and trusted source.

The Technical Committee generally recommends that production implementations use Zero-Trust security **[ZeroTrust]**, especially because of the wide distribution and potentially diverse ownership of TRM Actors. Zero Trust security requires authentication and authorization of every device, person, and application. The best practice is to encrypt all messages, even those between the separate components of an application within the cloud.

This specification makes no attempt to describe methods or technologies to enable Zero Trust interactions between Actors.

B.2 CTS and Privacy Considerations

The United Nations has defined privacy as “the presumption that individuals should have an area of autonomous development, interaction and liberty, a ‘private sphere’ with or without interaction with others, free from state intervention and excessive unsolicited intervention by other uninvited individuals. The right to privacy is also the ability of individuals to determine who holds information about them and how that information is used” (UN General Assembly 2013:15).

Electrical usage data inherently creates a privacy risk. Published work has demonstrated that simple usage data can be used to reveal the inner operations and decisions in a home. Other research has demonstrated that anonymous electrical usage data can be “de-anonymized” to identify an individual electricity user. The more fine-grained the data, the more intimate the details that can be garnered from meter telemetry.

In an amicus brief in a case on smart metering, the Electronic Freedom Foundation testified that that aggregate smart meter data collected from someone’s home in 15-minute intervals could be used to infer, for example, whether they tend to cook meals in the microwave or on the stove; whether they make breakfast; whether and how often they use exercise equipment, such as a treadmill; whether they have an in-home alarm system; when they typically take a shower; if they have a washer and dryer, and how often they use them; and whether they switch on the lights at odd hours, such as in the middle of the night. And these inferences, in turn, can permit intimate deductions about a person's lifestyle, including their occupation, health, religion, sexuality, and financial circumstances. These privacy concerns are linked to increased security risks criminals may be able to access the data and use the information to enable inferences about what people are doing in their home or if they are away from home.

This specification describes how to share communications beyond mere electrical usage telemetry. Communications reveal what the user would like to buy, how much they would be willing to spend, and future intents and plans.

System developers using this specification should consider legal requirements under the Fair Practice Principles and the European Union's General Data Protection Regulation. These include:

1) The Collection Limitation Principle. There should be limits to the collection of personal data and any such data should be obtained by lawful and fair means and, where appropriate, with the knowledge or consent of the data subject.
2) The Data Quality Principle. Personal data should be relevant to the purposes for which they are to be used and, to the extent necessary for those purposes, should be accurate, complete and kept up-to-date.
3) The Purpose Specification Principle. The purposes for which personal data are collected should be specified not later than at the time of data collection and the subsequent use limited to the fulfillment of those purposes or such others as are not incompatible with those purposes and as are specified on each occasion of change of purpose.
4) The Use Limitation Principle. Personal data should not be disclosed, made available or otherwise used for purposes other than those specified, except a) with the consent of the data subject, or b) by the authority of law.
5) The Security Safeguards Principle. Personal data should be protected by reasonable security safeguards against such risks as loss or unauthorized access, destruction, use, modification or disclosure of data.
6) The Openness Principle. There should be a general policy of openness about developments, practices and policies with respect to personal data. Means should be readily available of establishing the existence and nature of personal data and the main purposes of their use, as well as the identity and usual residence of the data controller.
7) The Individual Participation Principle. An individual should have the right:
    a. to obtain from a data controller, or otherwise, confirmation of whether or not the data controller has data relating to him;
    b. to have data relating to him communicated to him, within a reasonable time, at a charge, if any, that is not excessive; in a reasonable manner, and in a form that is readily intelligible to him;
    c. to be given reasons if a request made under subparagraphs (a) and (b) is denied and to be able to challenge such denial; and
    d. to challenge data relating to him and, if the challenge is successful, to have the data erased, rectified, completed or amended;
8) The Accountability Principle. A data controller should be accountable for complying with measures which give effect to the principles stated above.

In developing this specification, the Technical Committee has kept in mind the need to support a developer wishing to support privacy. Actors representing an up-stream electrical serving entity, say a distribution system operator or traditional utility, use the tame messages as anyone else—no actor is inherently privileged. Messages to provide market information or “ticker-tape” functions do not include Party IDs. General advertising of Tenders, while necessary to draw matching Tenders quickly to market, may be anonymous.

In some messages and some markets, it is necessary to use a proxy ID to protect privacy or to simply conveyance of a transaction from a complex matching mechanism. To protect privacy, a market may transmit such a proxy ID in place of a Party Id in Quotes, Tenders, Transactions, and Tickers. Markets that use cumulative matching algorithms such as double auction cannot identify a specific Counter Party to a transaction.

The system developer should keep the privacy principals in mind when making specific technology choices. For example, messages between an actor and the market MAY be encrypted to protect the privacy of people represented by individual actors. While the transactive energy market must know both buyers and sellers to support transaction contracts and settlements, the developer should take steps to guard that information. A developer may opt that each notice of contract sent to an actor always has a counterparty of the market, so as to protect the sources and uses of electricity.

It is beyond the scope of this specification to specify security practices and privacy design for markets built using this specification.

Appendix C. Conformance to the TEMIX Profile of Energy Interoperation

TBD

Appendix D. 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--15-1 Abbreviations and Terms used throughout this document

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

Appendix E. Acknowledgments

This work is derived from the specification Common Transactive Services 1.0 , contributed by The Energy Mashup Lab, written by William T. Cox and Toby Considine.

Portions of models and text is derived from The Energy Mashup Lab open source project, EML-CTS and is used under terms of the Apache 2.0 License for that project.[20]

E.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)

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

Appendix F. 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.
WD09	9/14/2021	William T Cox Toby Considine David Holmberg	Added Facet descriptions for Position, Market Characteristics, CTS Streams, and drafts of Privacy Consideration, Delivery and Party Registration Facets. Numerous edits for clarity and conciseness.
WD10	10/4/2021	Toby Considine William T Cox David Holmberg	Extended Market Facets. Defined Position and Delivery facets. Made references more consistent. Updated UML model and diagrams.
WD11	10/22/2021	David Holmberg William T Cox Toby Considine	Corrections for clarity. Improved UML diagrams. Flagged requests for comments in Public Review

CSD01	10/29/2021	OASIS TC Administration	Content as in WD11, formatted to include OASIS metadata and references to the published specification
WD12	1/10/2022	William T Cox Toby Considine	Simpler edits in response to comments from PR
WD13		William T Cox Toby Considine	Clarification of Resource/Product/Instrument Removal of references to “Architecture” Responses to “Clarity” tagged issues
WD14	2/22/2022	William T Cox Toby Considine	Clarification of front material Section 1/-2 compared to eliminate duplicative definitions Numerous issues resolutions applied as per Jira
WD15	3/20/2020	William T Cox Toby Considine	Clarity, responses to issues from Review
WD16	4/12/2022	William T Cox Toby Considine	Marketplace and Market characteristics responses to issues Expanded Quotes and Tickers Focus on capitalization
WD17	4/25/2022	William T Cox Toby Considine	Updated UML Market Information added OTC Transactions Edits for Clarity

Notices