Energy Interoperation Common Transactive Services (CTS) Version 1.0

Working Draft 14.5

February 2022

This 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

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:

This document is related to:
• Schedule Signals and Streams Version 1.0. Edited by Toby Considine and William T. Cox. Latest version: http://docs.oasis-open.org/ws-calendar/signals/v1.0/signals-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/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/).

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]


Notices:
Copyright © OASIS Open 2021. All Rights Reserved.
Distributed under the terms of the OASIS IPR Policy, [https://www.oasis-open.org/policies-guidelines/ipr].
For complete copyright information please see the Notices section in the Appendix.
# Table of Contents

1 Introduction .................................................................................................................. 9
    1.1 Application of the Common Transactive Services .................................................. 10
    1.2 Support for Developers ......................................................................................... 10
    1.3 Naming Conventions ............................................................................................ 11
    1.4 Editing Conventions ............................................................................................... 11
    1.5 Use of terms Actors and Facets in this specification .............................................. 11
    1.6 Security and Privacy ............................................................................................. 12
        1.6.1 Security Considerations ................................................................................. 12
        1.6.2 Privacy Considerations ................................................................................. 12
    1.7 Semantic Composition ............................................................................................ 13
        1.7.1 Conformance with Energy Interoperation ...................................................... 13
        1.7.2 Conformance with EMIX .............................................................................. 13
        1.7.3 Conformance with WS-Calendar Streams ...................................................... 14
    1.8 Conformance with Facilities Smart Grid Information Model .................................. 14
2 Overview of Common Transactive Services ................................................................. 15
    2.1 Scope of Common Transactive Services ................................................................ 15
        2.1.1 Applicability to Microgrids (Informative) ...................................................... 15
        2.1.2 Specific scope statements .............................................................................. 15
    2.2 Common Transactive Services Roles ..................................................................... 17
        2.2.1 Parties as Market Participants ....................................................................... 17
        2.2.2 Facets in CTS ................................................................................................. 18
        2.2.3 Party and Counterparty in Tenders and Transactions ...................................... 19
    2.3 Responses ............................................................................................................... 19
3 Common Semantic Elements of CTS ............................................................................ 21
    3.1 Semantic Elements from WS-Calendar .................................................................. 21
    3.2 Semantic Elements from EMIX .............................................................................. 21
        3.2.1 Defining Resource ......................................................................................... 22
        3.2.2 Defining Product ......................................................................................... 22
        3.2.3 Market Semantics from EMIX ....................................................................... 22
4 Basic Interaction and Terminology .............................................................................. 24
    4.1 Structure of Common Transactive Services and Operations ................................... 24
    4.2 Naming of Services and Operations ...................................................................... 24
    4.3 Payloads and Messages ......................................................................................... 24
    4.4 Description of the Facets and Payloads ................................................................ 24
    4.5 Responses ............................................................................................................... 25
5 CTS Streams ................................................................................................................ 26
    5.1 CTS Streams Introduction ..................................................................................... 26
    5.2 Information Model for CTS Streams ..................................................................... 26
6 Marketplace Facet ......................................................................................................... 29
    6.2 Registering in a Marketplace ................................................................................. 29
    6.3 Discovery of Marketplace Terms ......................................................................... 29
Table of Tables

Table 2-1: Definitions used in CTS Markets ................................................................. 16
Table 2-2: Facets Defined in CTS .................................................................................. 18
Table 2-3: Responses .................................................................................................... 20
Table 3-1: CTS Elements from WS-Calendar .............................................................. 21
Table 3-2: Defining the Resource – TODO – Not sure if this is still correct ................. 22
Table 3-3: Defining the Product .................................................................................... 22
Table 3-4: Market-related elements from EMIX .......................................................... 23
Table 5-1: CTS Stream Attributes .............................................................................. 27
Table 5-2: Stream Interval Attributes ......................................................................... 28
Table 8-1: Standard Terms returned by Market Facet .................................................. 33
Table 8-2: Elements that define Products in a Market or Marketplace ....................... 35
Table 9-1: Pre-Transaction Tender Facets ................................................................. 36
Table 9-2: EiTender Attributes ..................................................................................... 38
Table 9-3 EiCreateTenderPayload Attributes .............................................................. 40
Table 10-1: Transaction Management Service ............................................................ 41
Table 10-2: EiTransaction Attributes ......................................................................... 42
Table 11-1: Position Facet ........................................................................................... 45
Table 11-2: Attributes of Position Facet Payloads ....................................................... 47
Table 12-1: Delivery Facet ............................................................................................ 49
Table 12-2: Attributes of Delivery Facet Payloads ....................................................... 51
Table 13-1: Quotation Facet ........................................................................................ 54
Table 13-2: Market Information Facet Ticker .............................................................. 55
Table C-1 Abbreviations and Terms used throughout this document ......................... 65
### Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1</td>
<td>Example of generic error response for a service operation</td>
<td>25</td>
</tr>
<tr>
<td>5-1</td>
<td>CTS Stream Definition</td>
<td>26</td>
</tr>
<tr>
<td>6-1</td>
<td>UML Sequence Diagram for the Market Facet</td>
<td>30</td>
</tr>
<tr>
<td>8-1</td>
<td>UML Sequence Diagram for the Market Facet. TODO: Update for Market Information Facet</td>
<td>32</td>
</tr>
<tr>
<td>8-2</td>
<td>UML of Market Facet payloads</td>
<td>33</td>
</tr>
<tr>
<td>9-1</td>
<td>UML Sequence Diagram for the Tender Facet</td>
<td>37</td>
</tr>
<tr>
<td>9-2</td>
<td>Class EiTender TODO – revisit Model and update references</td>
<td>37</td>
</tr>
<tr>
<td>9-3</td>
<td>UML Class Diagram for the Operation Payloads for the EiTender Facet</td>
<td>39</td>
</tr>
<tr>
<td>10-1</td>
<td>UML Sequence Diagram for the EiTransaction Facet</td>
<td>41</td>
</tr>
<tr>
<td>10-2</td>
<td>UML Class Diagram of EiTransaction</td>
<td>42</td>
</tr>
<tr>
<td>10-3</td>
<td>UML Class Diagram of EiTransaction Facet Operation Payloads</td>
<td>43</td>
</tr>
<tr>
<td>10-4</td>
<td>UML Diagram comparing Tender and Transaction Facet Payloads</td>
<td>44</td>
</tr>
<tr>
<td>11-1</td>
<td>UML Sequence Diagram for the Position Facet</td>
<td>46</td>
</tr>
<tr>
<td>11-2</td>
<td>UML Class Diagram of Payloads for the Position Facet</td>
<td>47</td>
</tr>
<tr>
<td>12-1</td>
<td>UML Sequence Diagram for the Delivery Facet</td>
<td>50</td>
</tr>
<tr>
<td>12-2</td>
<td>UML Class Diagram of Payloads for the Position Facet</td>
<td>51</td>
</tr>
<tr>
<td>13-1</td>
<td>UML Sequence Diagram for the Quotation Facet</td>
<td>54</td>
</tr>
<tr>
<td>13-2</td>
<td>Market Information Facet Ticker Sequence Diagram</td>
<td>56</td>
</tr>
</tbody>
</table>
1 Introduction

The Common Transactive Services (CTS) is an application profile of the OASIS Energy Interopera-
tion 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
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 the use of remote or cloud-based 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.

Commented [TC1]: Rewritten for clarity to address DGH comments

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,

```xml
[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,

```xml
<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.

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. SOA focuses on the requested results rather than the specific process used. Service orientation complements loose integration and organizes distributed capabilities that may be in different ownership domains. [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. 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 it makes no assumptions of the mechanisms or even motives internal to an Actor. An Actor is simply a thing that acts. The Actor implementation may

---

2 In ISO and IEC standards, portions that are not normative are informative. OASIS uses the term non-normative.
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 buy or sell power. 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 set 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.

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

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

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

1.7.3 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].

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 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] 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 interoperating node must translate and enforce constraints to the microgrid by means of CTS.

[StructuredEnergy][SmartGridBusiness]

2.1.2 Specific scope statements

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

- Interaction patterns to support transactive energy, including tenders, transactions, and supporting information
- Information models for price and product communication
- Information models for market characteristics
- Payload definitions for Common Transactive Services

The following are out of scope for Common Transactive Services:

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

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

Commented [TC2]: Simplified in language and in examples in response to DGH criticism.
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 granularity in quantity and in time. For example, the Product may be 1 MW of power delivered over an hour. Similarly, another Product may be 1 kW of power over a 5-minute period. Some transactive energy markets in North America today have durations as brief as two seconds. Temporal granularity is equally important as quantity for Product definition.

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

A market considers all the tenders it has received offering to buy or sell an Instrument, using a Matching Engine to decide which can be cleared (satisfied) in full or in part. The 3:00pm Instrument is traded independently from the 4:00pm Instrument. This specification does not assume or require an Order Book, a Double Auction, or another 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>
<thead>
<tr>
<th>Transactive Entity</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>A measurable commodity, substance, service, or force, whose value is determined by time of delivery.</td>
</tr>
<tr>
<td>Product</td>
<td>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.</td>
</tr>
<tr>
<td>Instrument</td>
<td>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.</td>
</tr>
</tbody>
</table>

Commented [TC3]: Some material brought down from introduction to avoid pre-definition, merged here into normative definition.
## 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 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

### Transactive Entity

<table>
<thead>
<tr>
<th>Party</th>
<th>A Party is an Actor that buys or sells Instruments in a CTS Marketplace. A Party may be described by a specific role in a specific interaction, such as Party or Counter Party. For semantic and privacy issues, see Section 2.2.3 below.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market</td>
<td>A Facet where Parties trade for a Product using tenders submitted to buy or sell an Instrument.</td>
</tr>
<tr>
<td>Marketplace</td>
<td>A Marketplace, identified in [EMIX] by a URL, is the set of Markets accessible to a Party. The Marketplace Information Facet supplies information common to all Markets in the Marketplace. The Facet also enumerates all Resources and Products available in the Marketplace, as well as an enumeration of the Market for each Product.</td>
</tr>
<tr>
<td>Market Context</td>
<td>In EMIX, the Market Context is a URI identifying a Marketplace.</td>
</tr>
<tr>
<td>Matching Engine</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 describes roles taken on by Actors as *Facets* for that Actor, each distinct from other roles the Actor may perform. The Facets are named and briefly described in Table 2-2. Each Facet includes several messages, as in submitting a Tender, acknowledging a Tender, and cancelling a Tender. Each facet is discussed in detail starting in Section 5.

<table>
<thead>
<tr>
<th>Facet</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketplace</td>
<td>A Marketplace is the sum of all Product markets available to a Party. A Party may query the Marketplace to discover what Products are traded in a Marketplace, including Resources and Products. 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 static information about the Marketplace and its Products and Markets.</td>
</tr>
<tr>
<td>Market Information</td>
<td>A Market is a facet for trading a particular Product. 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 algorithm for this matching, whether one-for-one as in order-book markets, or group-by-group as in double auction markets, is out of scope. Some Markets MAY register transactions privately agreed to among Parties. See Section 8 Market Information Facet</td>
</tr>
<tr>
<td>Registration</td>
<td>A Party must Register with a Market to participate in that Market. See Section 7, Party Registration Facet</td>
</tr>
<tr>
<td>Tender</td>
<td>A Tender is an actionable offer to buy or to sell an Instrument at a given price. Tenders go to the market and are generally private. It is possible to request that a Tender be advertised to all Parties in the Market. <em>Note: a Tender for one side MAY match more than one Tender on the other side, which would generate multiple Transactions.</em> See Section 9 Tender Facet</td>
</tr>
<tr>
<td>Transaction</td>
<td>A Transaction is created by the Market to record a contract when a Tender to buy and a Tender to sell are matched. Each party is notified of the Transaction Creation. <em>Note: a Tender for one side MAY match more than one Tender on the other side, which would generate multiple Transactions.</em></td>
</tr>
</tbody>
</table>

Commented [HDG(5)]: Does this apply to a double auction? In that case the market engine is finding the matching point of load and supply bid curves.

Commented [TC6R5]: Original language merely said tenders matched, was non-specific as to mechanism. Added language calling out DA specifically.

Commented [TC7]: Issue 724.
Facet | Definition
---|---
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, completed Transactions, and other market information. See Section 13 Market Information Facet—Quote and Ticker

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.

Some Parties and some Markets may use a delegate, that is, a party that is neither the Party that is buying nor the Party that is selling. A Party acting as a delegate should have its own PartyId that can be used in Tenders, Quotes, and Transactions. For clarity of exposition, this PartyID is named the DelegateID. A DelegateID may be used in place of any PartyID when needed.

In some Market designs, such as double auction, there may be no identifiable Counterparty. Such a market should use a Delegate ID as the CounterpartyId in all Transactions. Transactions are announcements to the Parties involved, so the PartyID is always that of the Party receiving the notification.

Where privacy requirements require shielding of a PartyID, a DelegateID should be used as the PartyID in distributed Tickers and Quotes. The PartyID in a Tender should always reference the party that is tendering.


### 2.3 Responses

This section re-iterates terms and simplifies models from [EI]. That specification is normative. The response types are common across all message categories.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request ID</td>
<td>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.</td>
</tr>
<tr>
<td>Response Code</td>
<td>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, 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</td>
</tr>
</tbody>
</table>

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

---

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 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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>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?”</td>
</tr>
<tr>
<td>Offset</td>
<td>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</td>
</tr>
<tr>
<td>Begin Date-Time</td>
<td>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.</td>
</tr>
<tr>
<td>Expiration Date-Time</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 what is sold in a market, when it is sold, what the units are, what the standard trade size is, EMIX refers to this as the Item. 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 virtual packaging of the Resource into a size and Duration for sale), and the Instrument (a Product sold at a specific time).

Instruments are what is bought and sold in CTS markets.

7 Including Resource Designator, Stream Start, and Decimal Fraction
3.2.1 Defining Resource

A Resource is the commodity that is bought or sold in a CTS marketplace. A Party can query a marketplace to discover a list of the Resources that can be bought and sold therein.

Table 3-2: Defining the Resource – TODO – Not sure if this is still correct

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>A Resource consists of a Designator, Name and a Description.</td>
</tr>
<tr>
<td>Item Description</td>
<td>The Item Description is a common name, as defined in EMIX</td>
</tr>
<tr>
<td>Item Unit</td>
<td>Item Unit is the unit of measure for the Resource.</td>
</tr>
<tr>
<td>Attributes</td>
<td>Optional elements that further describe the Resource, as in hertz and voltage</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Abstract Base for all defining all Products. The core of each Product is the Resource, as referenced by the Resource Designator.</td>
</tr>
<tr>
<td>Scale</td>
<td>Exponent that specifies the size of the Resource Unit. For example, a Product denominated in Megawatts has a Scale of 6.</td>
</tr>
<tr>
<td>Size</td>
<td>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.</td>
</tr>
<tr>
<td>Warrant</td>
<td>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 so long as the Product clears, that is, delivery is taken in the same interval as it is bought. Products with differing Warrants are different Products.</td>
</tr>
</tbody>
</table>

Products with differing Warrants are different Products.

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartyID</td>
<td>The market-based ID of an actor participating in a Market, particularly the actor originating a Tender, Quote, or Contract.</td>
</tr>
<tr>
<td>Counter PartyID</td>
<td>The market-based ID of an actor participating in a Market, particularly the actor taking the other side of a contract from the Party. See Section 2.2.3.</td>
</tr>
<tr>
<td>Side</td>
<td>An indication of what a Party offers in a Tender or other message, i.e., “Buy” or “Sell”.</td>
</tr>
<tr>
<td>Expiration Date-Time</td>
<td>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.</td>
</tr>
<tr>
<td>Market Context</td>
<td>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, “Market Facet”.</td>
</tr>
<tr>
<td>Standard Terms</td>
<td>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”.</td>
</tr>
</tbody>
</table>

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

- Market 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 Facet—Quotes and market Tickers

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.

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. 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

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

Responses may need to be tracked to determine whether an operation succeeds or not. This may be complicated by the fact that any given Transaction may involve the transmission of one or more information objects.

An EiResponse returns the success or failure of the entire operation, with possible detail included in responseTermsViolated (see Section 5).

It is MANDATORY to return responses. Indicating partial or complete success or failure.

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

CTS uses EiResponseType is from Ei, changing only the cardinality of responseDescription (to zero, that is, not passed).

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

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

The Response Codes are intended to enable even the smallest device to interpret Response. This specification uses a pattern consisting of a 3-digit code, with the most significant digit sufficient to interpret success or failure. This pattern is intended to support that smallest device, while still supporting more nuanced messages that may be developed.

While the only value after the leading digit the Response Code defined in Ei 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.

---

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).
5 CTS Streams

Most 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.

5.1 CTS Streams Introduction

CTS Streams are a conformant specialization of WS-Calendar Streams. Conformance can be established by mapping the elements to the [Streams] standard. For CTS, the simplification contains all defining metadata within a single object, rather than that metadata potentially being found or queried in multiple places. In addition, CTS Streams include elements that are outside the Streams standard but may be determined by examining referring type instances. CTS Streams have neither interaction patterns nor payloads, as it is used in defining Facet Payloads.

5.2 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 Payload simplified into a streamPayloadValue and the internal local UID for the stream element.

![Figure 5-1: CTS Stream Definition](image)

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.
### Table 5-1: CTS Stream Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Designator</td>
<td>A Long Integer that indicates the Resource for the Product and Market</td>
<td>The Resource Designator in a Market should match Resource Designator as enumerated in the Marketplace</td>
</tr>
<tr>
<td>Stream Scale</td>
<td>The Scale is the exponent that determines the size of the Resource.</td>
<td>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.</td>
</tr>
<tr>
<td>Stream Duration</td>
<td>The duration for each of the contiguous Stream Intervals</td>
<td>This completes the Product definition of a Resource at a Scale and Size delivered over a Duration.</td>
</tr>
<tr>
<td>Stream Price Granularity</td>
<td>Price granularity expressed as an exponent. Applies to all Intervals in the Stream. Not required for all Facets.</td>
<td>For example, if the price granularity is -3, and the value is 1500, the price is 1.500 currency units.</td>
</tr>
<tr>
<td>Stream Start</td>
<td>The Start Date and Time for a bound CTS Stream</td>
<td>See WS-Calendar Date-Time in Section 3.1.</td>
</tr>
<tr>
<td>Stream Intervals</td>
<td>The ordered set of Stream Intervals</td>
<td>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]</td>
</tr>
</tbody>
</table>
### Table 5-2: Stream Interval Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream Payload Value</td>
<td>The value for this specific Stream Interval</td>
<td>In CTS Streams, this is either Price, or Quantity, or Both, depending upon the Facet.</td>
</tr>
<tr>
<td>UID</td>
<td>A “Local UID” used to order the Interval within the Stream</td>
<td>As conformant CTS implementations need not be owned by the same implementer, intermarket gateways (however defined) may deserialize and re-serialize to different specifications</td>
</tr>
</tbody>
</table>
6 Marketplace Facet

The Marketplace is a new facet in CTS. The Marketplace includes all the Markets wherein a Party can trade and all the Products a Party can trade for. Where all trading is in a single microgrid, the marketplace is available 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 a utility from setting up multiple Marketplaces, such as a Wholesale or a Retail, or a sourced Marketplace such as Solar.

Using the language of Resource / Product / Instrument, 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 market rules common to all the local Markets, and catalogs how to participate in each Market.

6.1 The [EI] Market Context and the Marketplace

Market Contexts in [EMIX] and [EI] are URIs and are used to express Market Information 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 market information were to be transmitted in every information exchange, messages would be overly repetitive.

The Marketplace Facet for CTS is simplified from Market Context EI and extended for access to standard terms.

6.2 Registering in a Marketplace

- Getting PartyID
- Establishing billing terms, if any
- Location Required

TODO: Market Registration interaction sequence

6.3 Discovery of Marketplace Terms

6.4 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 6-1: UML Sequence Diagram for the Market Facet

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

Currency
Time Zone
List of Resources
List of Products/Resources
...
7 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 messages necessary for an actor (Party) to join a market and to leave or be removed from a market.

- Create Party associates an actor with an 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,
- Register Party 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.

ToDo:
- Party Location

Commented [TC14]: Comment here to link to ToDo
8 Market Information Facet

All interactions in a Market are subject to common rules of engagement which can be discovered through the Market Information 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. Instruments in a Market differ from other Instruments in that Market only in their Starting Date and Time. The matching engine is within the Market and different matching engines have no visibility past the Market Facet. 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 Market Information that rarely changes, so it is not necessary to communicate it with each message.

In CTS, this is refined to the Marketplace (Section 6) and to Market Information.

8.2 Interaction Pattern for the Market Information 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 Information Facet

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

Delivering an EiRequestMarketCharacteristics payload requests the standard terms for a Market; the reply payload EiReplyMarketCharacteristics returns those terms as name-value pairs.
### 8.3 Information Model for the Market Facet

Sending an EiRequestMarketCharacteristics payload referencing a Market (by containing a market context) 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.

#### Table 8-1: Standard Terms returned by Market Facet

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute Name</th>
<th>Attribute Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market Name</td>
<td>NAME</td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Attribute Name</td>
<td>Attribute Type</td>
<td>Meaning</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Currency</td>
<td>CURRENCY</td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>Time Offset</td>
<td>T_OFFSET</td>
<td>Long</td>
<td>A Duration that some marketplaces MAY use to describe trading where a first interval is not on an hourly boundary.</td>
</tr>
<tr>
<td>Time Zone</td>
<td>TZ</td>
<td>String</td>
<td>A Time Zone indicates how all Times and Dates are expressed.</td>
</tr>
<tr>
<td>Price Decimal Fraction Digs</td>
<td>PRICE_FRAC</td>
<td>Long</td>
<td>Some market implementations use a market-wide indication of how many decimal fraction digits are used.</td>
</tr>
<tr>
<td>Market Party ID</td>
<td>MPARTYID</td>
<td>String</td>
<td>The PartyID to use in a Tender (reference 2.2.3)</td>
</tr>
</tbody>
</table>
| Bilateral OK               | BILATERAL OK   | Long           | **Boolean expressed as integer**
1. True—bilateral Transactions with identified parties are permitted.
2. False—bilateral Transactions not permitted, only market Tenders.                     |
| Resource Designator        | R_ID           | Long           | 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 2-1: Definitions used in CTS Markets. It SHALL match the Product Definition indicated in the Marketplace for this Market. |

---

10 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.

11 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.
Each Product in a Marketplace is defined using attributes as below

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Designator</td>
<td>R_ID</td>
<td>Reference to the required Resource</td>
</tr>
<tr>
<td>Time Granularity</td>
<td>T_GRAIN</td>
<td>The interval duration in seconds for the specific Product definition</td>
</tr>
<tr>
<td>Quantity Scale</td>
<td>Q_SCALE</td>
<td>The exponent of the Quantity. For example, a Product denominated in kilowatts has a Q_SCALE of 3.</td>
</tr>
<tr>
<td>Quantity Granularity</td>
<td>Q_GRAIN</td>
<td>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.</td>
</tr>
<tr>
<td>Price Granularity</td>
<td>PRICE_GRAIN</td>
<td>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.</td>
</tr>
<tr>
<td>Market</td>
<td>MARKET</td>
<td>The message endpoint to access the market where this Product is traded.</td>
</tr>
<tr>
<td>Warrants</td>
<td>WARRANT</td>
<td>Optional further specificity of Product</td>
</tr>
</tbody>
</table>

Transactive Services in E1 define and support the lifecycle of transactions from initial Tender to final settlement. The phases described in E1 are

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

Commented [TC17]: Compare carefully to the Price Granularity used in CTS-Stream to make sure we do not create confusion.

Commented [TC18]: Review to see if this language from initial submission still makes sense.
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

<table>
<thead>
<tr>
<th>Facet</th>
<th>Request Payload</th>
<th>Response Payload</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EiTender</td>
<td>EiCreateTender</td>
<td>EiCreatedTender</td>
<td>Send a CTS-Stream of one or more Tenders. Create and emit Request Payload</td>
</tr>
<tr>
<td>EiTender</td>
<td>EiCancelTender</td>
<td>EiCanceledTender</td>
<td>Cancel one or more Tenders</td>
</tr>
<tr>
<td>EiTender</td>
<td>EiDistributeTender</td>
<td>None</td>
<td>Describe a list of Tenders to be notified to a set of parties</td>
</tr>
</tbody>
</table>

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.
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.

The EiTenders is in the form of a CTS Stream containing Price and Quantity in the Payload.
### Table 9-2: EiTender Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expiration Time</td>
<td>The date and time after which this Tender is no longer valid.</td>
<td></td>
</tr>
<tr>
<td>Integral Only</td>
<td>All of the Tender must be bought or sold at once; no partial sale or purchase</td>
<td>In CTS set to False. Partial sale or purchase is always allowed. The attribute is present for possible future evolution.</td>
</tr>
<tr>
<td>Interval</td>
<td>The time interval for the Product being offered</td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>The unit price for the Product being Tendered</td>
<td>Total price is the product of price and quantity. Note that price is subject to the Price Decimal Fraction value.</td>
</tr>
<tr>
<td>Quantity</td>
<td>The quantity of the Product being Tendered</td>
<td>Total price is the product of price and quantity</td>
</tr>
<tr>
<td>Side</td>
<td>Whether The Tender is to buy or to sell the product</td>
<td></td>
</tr>
<tr>
<td>Tender ID</td>
<td>An ID for this Tender</td>
<td>Commented [WTC19]: Eliminated Transactive State. See e.g. ENERGYINTEROP-730</td>
</tr>
</tbody>
</table>
9.4 Payloads for the Tender Facet

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

Figure 9-3: UML Class Diagram for the Operation Payloads for the EiTender Facet
The following table describes the attributes for EiCreateTenderPayload

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter Party ID</td>
<td>The Actor ID for the CounterParty for which the Tender is created.</td>
<td>Among the Standard Terms for a Market is PartyID of that Market. This MarketID indicates the expectation that the market will match and clear the tender if possible. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of the other Actor may be used.</td>
</tr>
<tr>
<td>Party ID</td>
<td>The Actor ID for the Party on whose behalf this Tender is made.</td>
<td>Indicates which Actor proposes the buy or sell side EiCreateTender.</td>
</tr>
<tr>
<td>EiTender</td>
<td>One or more EiTenders to be created.</td>
<td>CTS uses CTS stream of EiTenders.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In CTS an object describing a Tender is instantiated then sent; the latter is a consequence of processing an EiCreateTender payload.</td>
</tr>
<tr>
<td>Resource Designator</td>
<td>The Resource being tendered</td>
<td>Must match the Market Resource Designator on receipt at the Market</td>
</tr>
<tr>
<td>Request ID</td>
<td>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(^\text{12}).</td>
<td></td>
</tr>
<tr>
<td>Responses</td>
<td>Responses for each attempted EiTender creation</td>
<td>Array Of Responses [EI]</td>
</tr>
</tbody>
</table>

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

Note that if more than one EiTender is requested to be created, 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.

\(^\text{12}\) 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.

Canceling or modifying Transactions is not permitted.

#### 10.2 Interaction Pattern for the Transaction Facet

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

![UML Sequence Diagram for the EiTransaction Facet](image)

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.

---

13 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.
The attributes of EiTransaction are shown in the following table.

**Table 10-2: EiTransaction Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tender</td>
<td>The Tender (Fig. 4-2) that led to this Transaction.</td>
<td>The ID, quantity and price may differ from that originally tendered due to market actions.</td>
</tr>
<tr>
<td>Transaction ID</td>
<td>An ID for this Transaction</td>
<td>The contained Tender has its own Tender Id</td>
</tr>
</tbody>
</table>
10.4 Payloads for the Transaction Facet

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

```
+ counterPartyId: ActorIdType
+ partyId: ActorIdType
+ requestId: ReIdType
+ transaction: EiTransaction

+ counterPartyId: ActorIdType
+ partyId: ActorIdType
+ refId: RefIdType
+ response: EiResponse
+ transactionId: TransactionIdType
```

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

class Tender and Transaction Facet Payloads

**EICreateTenderPayload**
- counterPartyId: ActorIdType
- partyId: ActorIdType
- requestId: RollIdType
- resourceDesignator: long
- tender: EITender [1..n]

**EIRejectedTenderPayload**
- counterPartyId: ActorIdType
- inResponseTo: RollIdType
- partyId: ActorIdType
- response: EIResponse
- responses: ArrayOfResponses [0..n]
- tenderId: TenderIdType [0..n]

**EICanceledTenderPayload**
- canceledResponse: EICanceledResponse
- counterPartyId: ActorIdType
- inResponseTo: RollIdType
- partyId: ActorIdType
- response: EIResponse
- responses: ArrayOfResponses [0..n]

**EICancelTransactionPayload**
- canceledResponse: EIDeletedResponse
- counterPartyId: ActorIdType
- partyId: ActorIdType
- requests: RollIdType
- transaction: EITransaction

**EIRejectedTransactionPayload**
- counterPartyId: ActorIdType
- requestId: RollIdType
- response: EIResponse
- transactionId: TransactionIdType
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.\(^\text{14}\)

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

<table>
<thead>
<tr>
<th>Facet</th>
<th>Request Payload</th>
<th>Response Payload</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>EiRequestPosition</td>
<td>EiReplyPosition</td>
<td>Request an Actor’s Position(s) for a specific time interval, and reply with those Position(s) if access is authorized.</td>
</tr>
</tbody>
</table>

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

---

\(^{14}\) One may say that a Party’s position for an instrument is evolved from an accumulation of trades for that instrument.
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.

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

![Class Diagram of Payloads for the Position Facet]

#### Table 11-2: Attributes of Position Facet Payloads

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounding Interval</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Position Party</td>
<td>The Party whose position is being requested. Allows a request for another Party’s position, with appropriate privacy and security constraints</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Meaning</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Market Context</td>
<td>The market context of interest</td>
<td>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</td>
</tr>
<tr>
<td>Request ID</td>
<td>A reference to this payload</td>
<td>May be used as a correlation ID</td>
</tr>
<tr>
<td>Requestor</td>
<td>The Party requesting the position.</td>
<td>A failure indication will be returned if the Requestor is not authorized to access position information for Position Party. Addresses the auditor use case.</td>
</tr>
<tr>
<td>Positions</td>
<td>CTS Streams containing the positions for Position Party for each Resource. Positions are signed or zero.</td>
<td>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</td>
</tr>
<tr>
<td>Response</td>
<td>An EiResponse. Will indicate failure if Requestor is not authorized to access position information for Position Party for any of the requested intervals.</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Facet</th>
<th>Request Payload</th>
<th>Response Payload</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivery</td>
<td>EiRequestDelivery</td>
<td>EiReplyDelivery</td>
<td>Request Delivery through a specific Measurement Point</td>
</tr>
</tbody>
</table>

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.

![UML Class Diagram of Payloads for the Position Facet](image)

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

**Table 12-2: Attributes of Delivery Facet Payloads**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounding Interval</td>
<td>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.</td>
<td>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</td>
</tr>
<tr>
<td>Measurement Point</td>
<td>The Point for which telemetry is provided about the flow of the resources.</td>
<td></td>
</tr>
<tr>
<td>Request ID</td>
<td>A reference to this payload</td>
<td>May be used as a correlation ID</td>
</tr>
<tr>
<td>Attribute</td>
<td>Meaning</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>Requestor</td>
<td>The Party requesting the position.</td>
<td>A failure indication will be returned if the Requestor is not authorized to access position information for Position Party. Addresses the auditor use case.</td>
</tr>
<tr>
<td>Delivered</td>
<td>A CTS Stream containing the Delivery information for the Resource. Delivery value is signed or zero.</td>
<td>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</td>
</tr>
<tr>
<td>Response</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
13 Market Information Facet—Quote and Ticker

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, actually the price of the last transaction. Parties use this public information to plan whether to submit new Tenders, or perhaps to cancel old Tenders.

This information, revealing historic price information as well as current advertising, is the subject of the Market Information Facet.

13.1 Quotes

[EI] defines a quotation as a market price or possible price, which needs a Tender and acceptance to reach a Transaction. An advertisement of an attractive price for limited amount of power might only be available to the first to respond. That said, a Quote looks very much like a Tender.

Different CTS-based systems may want to distribute Quotes in different ways. Some may permit an Actor to broadcast Quotes to all other Actors. Others will require that a Quote be submitted to the Market which will then distribute the Quote to all subscribers. A market MAY choose to protect privacy by indicating its own Party Id as the originator of all quotes.
13.1.1 Interaction Pattern for Market Information Facet Quote

This is the [UML] sequence diagram for the Market Information Facet Quote:

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

13.1.2 Information Model for the Quote

An Actor may submit quotes for a number of 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.

Table 13-1: Quotation Facet

<table>
<thead>
<tr>
<th>Facet</th>
<th>Request</th>
<th>Response</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market Information</td>
<td>EiCreateQuote</td>
<td>EiCreatedQuote</td>
<td>Creates a Quote</td>
</tr>
<tr>
<td>Market Information</td>
<td>EiDistributeQuote</td>
<td></td>
<td>Used for a broadcast of a Quote. Depending on system business rules, MAY be only to subscribers</td>
</tr>
<tr>
<td>Market Information</td>
<td>EiCancelQuote</td>
<td>EiCanceledQuote</td>
<td>This can be point to point or broadcast per system design</td>
</tr>
</tbody>
</table>

13.2 Tickers

Ticker names the stream of information representing the price of each Instrument in a Market over time based on the actual prices as gleaned from the Transactions in that market. The service is named by the analogy to financial markets, and to tickers and ticker-tape.
While Tenders may be private, the existence of contracts are expected to be public (although typically without party identification). Subscribing Actors are continuously informed of executed contracts by means of Tickers. This facet is named by analogy to the earliest financial communications medium, which transmitted stock price information to a machine called a stock ticker, which printed the information on a continuous paper strip. The term "ticker" came from the sound made by the machine as it printed.

**EDITOR'S NOTE** Ticker needs a Product; there is some confusion between functions of a messaging

### 13.2.1 Information Model for the Market Information Facet Ticker

The information model for the Ticker is the same as that for a Transaction. Depending on specific system and privacy requirements, the ticker may replace one or both of the Party ID and Counter-Party ID may be absent from the Ticker. If a requirement for strong message typing requires their inclusion, the Party ID of the market can be substituted for either or both.

**Table 13-2: Market Information Facet Ticker**

<table>
<thead>
<tr>
<th>Facet</th>
<th>Request</th>
<th>Response</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market Information</td>
<td>EiCreateTicker</td>
<td>EiCreatedTicker</td>
<td>Create a ticker for a specific Market and reply with a reference.</td>
</tr>
<tr>
<td>Market Information</td>
<td>EiCancelTicker</td>
<td>EiCanceledTicker</td>
<td>This can be point to point or broadcast as per system design</td>
</tr>
<tr>
<td>Market Information</td>
<td>EiDistributeTicker</td>
<td></td>
<td>Distribute a reference to a Ticker. Similar conditions apply as for EiDistributeTender</td>
</tr>
</tbody>
</table>

### 13.2.2 Interaction Pattern for the Market Information Facet Ticker

This is the [UML] sequence diagram for the Market Information Facet Ticker:
Figure 13-2 Market Information Facet Ticker Sequence Diagram
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 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]


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]

[Framework]

[CTS2016]

[EML-CTS]

[FSGIM]

[iCalendar]

See also

[GridFaultResilience]

[IEC62746-10-3]

[Micromarkets]


[RFC3552]

[SmartGridBusiness]

[StructuredEnergy]

OASIS Committee Specification 01.


[TRM] (Transactive Resource Management)


[UML]


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/xmlext11-1/, April 2012,


[ZeroTrust]

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.

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 same 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.

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 system design form 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--1 Abbreviations and Terms used throughout this document

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTS</td>
<td>Common Transactive Services</td>
</tr>
<tr>
<td>EI</td>
<td>Energy Interoperation, an OASIS specification as per the normative references, CTS is a conforming profile of EI.</td>
</tr>
<tr>
<td>EMIX</td>
<td>Energy Market Information Exchange, an OASIS specification used to describe products and markets for resources, particularly those traded in power grids.</td>
</tr>
</tbody>
</table>
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.\(^{15}\)

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)

## Appendix F. Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Editor</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>WD01</td>
<td>2/15/2021</td>
<td>Toby Considine</td>
<td>Initial reformatting and conversion of the specification contributed by The Energy Mashup Lab to create a document for committee work.</td>
</tr>
<tr>
<td>WD02</td>
<td>3/3/2021</td>
<td>Toby Considine</td>
<td>Added prose definitions of Resource, Product, and Instrument</td>
</tr>
<tr>
<td>WD03</td>
<td>4/5/2021</td>
<td>Toby Considine</td>
<td>Simplified introductory material, raised message type to earlier in document. Removed some repetitive material. Revised UML required.</td>
</tr>
<tr>
<td>WD04</td>
<td>5/7/2021</td>
<td>Toby Considine, David Holmberg, William T Cox</td>
<td>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</td>
</tr>
<tr>
<td>WD05</td>
<td>5/25/2021</td>
<td>Toby Considine, David Holmberg, William T Cox</td>
<td>Continues clean-up and condensation of sections 1, 2</td>
</tr>
<tr>
<td>WD06</td>
<td>6/7/2021</td>
<td>Toby Considine</td>
<td>Refines Item language into Resource and Products. Explains Message Groups as a conforming descendant of EI Services.</td>
</tr>
<tr>
<td>WD07</td>
<td>6/21/2021</td>
<td>Toby Considine, William T Cox</td>
<td>Clarified terminology and relationship to implied Service-Oriented Architecture. Structured CTS facets for clearer explanation</td>
</tr>
<tr>
<td>WD08</td>
<td>8/5/2021</td>
<td>Toby Considine, William T Cox, David Holmberg</td>
<td>Clarify and simplify actor facets descriptions, including Tender, Transaction, and Configuration. Reduce redundant and less relevant content.</td>
</tr>
<tr>
<td>WD09</td>
<td>9/14/2021</td>
<td>William T Cox, Toby Considine, David Holmberg</td>
<td>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.</td>
</tr>
<tr>
<td>WD11</td>
<td>10/22/2021</td>
<td>David Holmberg, William T Cox, Toby Considine</td>
<td>Corrections for clarity. Improved UML diagrams. Flagged requests for comments in Public Review</td>
</tr>
<tr>
<td>CSD01</td>
<td>10/29/2021</td>
<td>OASIS TC Administration</td>
<td>Content as in WD11, formatted to include OASIS metadata and references to the published specification</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>--------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WD12</td>
<td>1/10/2022</td>
<td>William T Cox Toby Considine</td>
<td>Simpler edits in response to comments from PR</td>
</tr>
<tr>
<td>WD13</td>
<td></td>
<td>William T Cox Toby Considine</td>
<td>Clarification of Resource/Product/Instrument Removal of references to “Architecture” Responses to “Clarity” tagged issues</td>
</tr>
<tr>
<td>WD14</td>
<td>2/22/2022</td>
<td>William T Cox Toby Considine</td>
<td>Clarification of front material Section 1/-2 compared to eliminate duplicative definitions Numerous issues applied as per Jira</td>
</tr>
</tbody>
</table>
Notices

Copyright © OASIS Open 2021. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website:

[https://www.oasis-open.org/policies-guidelines/ipr].

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

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

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

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

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

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

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

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