Energy Interoperation Common
Transactive Services (CTS) Version 1.0

Committee Specification Draft 01
Working draft 07 draft

21 June 2021

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

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

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

Technical Committee:
OASIS Energy Interoperation TC

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

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

Additional artifacts:
This document is one component of a Work Product that also includes:
This prose specification is one component of a Work Product that also includes:
- UML models
- JSON schemas
- FIX Simple Binary Encoding binding (SBE)
- XML schemas

Related work:
This document replaces or supersedes:

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

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

TC members should send comments on this document to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at https://www.oasis-open.org/committees/energyinterop/.

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

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

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

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

[Energyinterop-CTS-v1.0]

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

For complete copyright information please see the Notices section in the Appendix.
Table of Contents

1 Introduction .......................................................................................................................... 7
  1.1 Application of the Common Transactive Services .......................................................... 7
  1.2 Support for Developers .................................................................................................. 8
  1.3 Naming Conventions ....................................................................................................... 8
  1.4 Editing Conventions ......................................................................................................... 9
  1.5 Architecture .................................................................................................................... 9
    1.5.1 Security Considerations ......................................................................................... 9
    1.5.2 Privacy Considerations .......................................................................................... 9
  1.6 Semantic Composition ..................................................................................................... 9
    1.6.1 Conformance with Energy Interoperation .............................................................. 10
    1.6.2 Conformance with EMIX ....................................................................................... 10
    1.6.3 Conformance with WS-Calendar Streams .............................................................. 10
    1.6.3.1 Schedule Negotiation with WS-Calendar ......................................................... 10
    1.6.3.2 Streams and Inheritance .................................................................................. 12
  1.6.4 Compatibility with Facilities Smart Grid Information Model ..................................... 12
  2 Overview of Common Transactive Services ................................................................. 14
    2.1 Scope of Common Transactive Services .................................................................... 14
      2.1.1 Applicability to Microgrids ............................................................................... 14
      2.1.2 Specific scope statements .................................................................................. 14
    2.2 Resources, Products and Instruments ......................................................................... 14
    2.3 Common Transactive Services Architecture ............................................................. 16
      2.3.1 Facets of CTS .................................................................................................... 16
      2.3.2 Sides in Tenders and Transactions ...................................................................... 17
      2.3.3 Responses ......................................................................................................... 18
    3 Common Semantic Elements of CTS .............................................................................. 19
      3.1 Semantic Elements from WS-Calendar ................................................................. 19
      3.2 Semantic Elements from EMIX ............................................................................. 19
        3.2.1 Defining Resource .......................................................................................... 19
        3.2.2 Defining the Product ....................................................................................... 20
        3.2.3 Market-related Elements from EMIX .............................................................. 20
      4 Basic Interaction and Terminology .............................................................................. 23
        4.1 Structure of Common Transactive Services and Operations ................................ 23
        4.2 Naming of Services and Operations .................................................................... 23
        4.3 Payloads and Messages ....................................................................................... 23
        4.4 Description of the Facets and Payloads ............................................................... 23
        4.5 Responses .......................................................................................................... 24
      5 Configuration and Market Characteristics Facet ......................................................... 26
        5.1 The Market Context .............................................................................................. 26
        5.2 Interaction Pattern for the Market Context Facet .................................................. 26
        5.3 Information Model for the EiMarketContext Facet ............................................. 27
        5.4 Operation Payloads for the EiMarketContext Facet ............................................ 27
      6 Tender Facet .................................................................................................................. 28
        6.1 Tenders as a Pre-Transaction Payloads ............................................................... 28

 ei-cts-v1.0-wd07
Standards Track Work Product Copyright © OASIS Open 2021. All Rights Reserved. 21 June 2021
Page 4 of 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.1</td>
<td>Interaction Pattern for the Tender Facet</td>
<td>28</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Information Model for the Tender Facet</td>
<td>29</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Payloads for the EiTender Facet</td>
<td>31</td>
</tr>
<tr>
<td>7</td>
<td>Transaction Facet</td>
<td>33</td>
</tr>
<tr>
<td>7.1</td>
<td>Transaction Services</td>
<td>33</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Interaction Pattern for the EiTransaction Service</td>
<td>33</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Information Model for the EiTransaction Service</td>
<td>33</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Operation Payloads for the EiTransaction Facet</td>
<td>35</td>
</tr>
<tr>
<td>7.2</td>
<td>Comparison of Transactive Payloads</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>Position Facet</td>
<td>37</td>
</tr>
<tr>
<td>9</td>
<td>Measurement and Verification Facet</td>
<td>38</td>
</tr>
<tr>
<td>10</td>
<td>Market Information Facet—Quotes and Tickers</td>
<td>39</td>
</tr>
<tr>
<td>11</td>
<td>Bindings</td>
<td>40</td>
</tr>
<tr>
<td>11.1</td>
<td>JSON</td>
<td>40</td>
</tr>
<tr>
<td>11.2</td>
<td>XML Schema</td>
<td>40</td>
</tr>
<tr>
<td>11.2.1</td>
<td>XML Namespaces</td>
<td>40</td>
</tr>
<tr>
<td>11.3</td>
<td>Simple Binary Encoding</td>
<td>40</td>
</tr>
<tr>
<td>12</td>
<td>Conformance</td>
<td>41</td>
</tr>
<tr>
<td>12.1</td>
<td>Claiming Conformance to Common Transactive Services</td>
<td>41</td>
</tr>
<tr>
<td>Appendix A.</td>
<td>References</td>
<td>42</td>
</tr>
<tr>
<td>A.1</td>
<td>Normative References</td>
<td>42</td>
</tr>
<tr>
<td>A.2</td>
<td>Informative References</td>
<td>42</td>
</tr>
<tr>
<td>Appendix B.</td>
<td>Security and Privacy Considerations</td>
<td>44</td>
</tr>
<tr>
<td>Appendix C.</td>
<td>Glossary of Terms and Abbreviations Used in this document</td>
<td>45</td>
</tr>
<tr>
<td>Appendix D.</td>
<td>Acknowledgments</td>
<td>46</td>
</tr>
<tr>
<td>D.1</td>
<td>Special Thanks</td>
<td>46</td>
</tr>
<tr>
<td>D.2</td>
<td>Participants</td>
<td>46</td>
</tr>
<tr>
<td>Appendix E.</td>
<td>Revision History</td>
<td>47</td>
</tr>
<tr>
<td>Notices</td>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>
## Table of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Core Semantics from WS-Calendar</td>
<td>11</td>
</tr>
<tr>
<td>2-1</td>
<td>Abstract Definitions used in CTS Markets</td>
<td>15</td>
</tr>
<tr>
<td>2-2</td>
<td>Transactive Service Message Groups</td>
<td>16</td>
</tr>
<tr>
<td>2-3</td>
<td>Responses</td>
<td>18</td>
</tr>
<tr>
<td>3-1</td>
<td>CTS Elements from WS-Calendar</td>
<td>19</td>
</tr>
<tr>
<td>3-2</td>
<td>Defining the Resource</td>
<td>19</td>
</tr>
<tr>
<td>3-3</td>
<td>Defining the Product</td>
<td>20</td>
</tr>
<tr>
<td>3-4</td>
<td>Market-related elements from EMIX</td>
<td>20</td>
</tr>
<tr>
<td>3-5</td>
<td>Standard Terms that define market interactions</td>
<td>21</td>
</tr>
<tr>
<td>6-1</td>
<td>Pre-Transaction Tender Services</td>
<td>28</td>
</tr>
<tr>
<td>6-2</td>
<td>EiResponse Attributes</td>
<td>30</td>
</tr>
<tr>
<td>7-1</td>
<td>Transaction Management Service</td>
<td>33</td>
</tr>
<tr>
<td>7-2</td>
<td>EiTransaction Attributes</td>
<td>34</td>
</tr>
<tr>
<td>C-1</td>
<td>Abbreviations and Terms used throughout this document for which this document is not normative</td>
<td>45</td>
</tr>
</tbody>
</table>

## Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Basic Power Object from EMIX</td>
<td>12</td>
</tr>
<tr>
<td>1-2</td>
<td>Applying Basic Power to a Sequence</td>
<td>12</td>
</tr>
<tr>
<td>4-1</td>
<td>Example of generic error response for a service operation</td>
<td>24</td>
</tr>
<tr>
<td>4-5-1</td>
<td>UML Sequence diagram for Market Context service</td>
<td>26</td>
</tr>
<tr>
<td>6-1</td>
<td>UML Sequence Diagram for the EiTender Service</td>
<td>29</td>
</tr>
<tr>
<td>6-2</td>
<td>Class EiTenderType</td>
<td>29</td>
</tr>
<tr>
<td>6-3-3</td>
<td>Enumeration TransactiveStateType</td>
<td>31</td>
</tr>
<tr>
<td>6-4</td>
<td>UML Class Diagram for the Operation Payloads for the EiTender Service</td>
<td>32</td>
</tr>
<tr>
<td>7-1</td>
<td>UML Sequence Diagram for the EiTransaction Service</td>
<td>33</td>
</tr>
<tr>
<td>7-2</td>
<td>UML Class Diagram of EiTransaction</td>
<td>34</td>
</tr>
<tr>
<td>7-3</td>
<td>UML Class Diagram of EiTransaction Service Operation Payloads</td>
<td>35</td>
</tr>
<tr>
<td>7-4</td>
<td>UML Diagram comparing all Transactive Payloads</td>
<td>36</td>
</tr>
</tbody>
</table>
1 Introduction

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

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

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

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

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

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

Neither EI nor CTS specifies which technologies participants will use; rather CTS defines a technology-agnostic minimal set of messages to enable interoperation through markets of participants irrespective of internal technology. In a similar manner, CTS does not specify the internal organization or operations a market for transactive energy will use, but rather a common set of messages that can be used to operate any particular transactive energy market. The goal of CTS is to enable systems and devices developed today or in the future to participate in markets deployed today or in the future. The reader can find an extended discussion of Transactive Energy (TE) in the EI specification.

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

1.1 Application of the Common Transactive Services

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

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

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

TE demonstrations and deployments to date have been unique systems—each uses its own message model and its own market dynamics. Many early implementations required the use of central or cloud-

---

1 In North American wholesale electricity markets, transmission rights are bought and sold.
based markets. Central markets discount local decision making while introducing new barriers to resilience. Others rely on a single price-setting supplier. None are interoperable either at the system level or for the actors involved.

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

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

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

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

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

1.2 Support for Developers

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

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

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

1.3 Naming Conventions

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

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

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

JSON and where possible SBE names follow the same conventions.
1.4 Editing Conventions

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

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

Information in the **Meaning** column of the tables is normative. Information appearing in the **Notes** column is explanatory and non-normative.²

Examples and Appendices are non-normative.

1.5 Architecture

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

1.5.1 Security Considerations

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

1.5.2 Privacy Considerations

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

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

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

Both security and privacy considerations are addressed in Appendix B.

1.6 Semantic Composition

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

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

- EMIX describes price and product for electricity markets.
- WS-Calendar communicates schedules and sequences of operations. This specification uses the [Streams] optimization which is a standalone specification, rather than part of Energy Interoperation 1.0.
- Energy Interoperation uses the vocabulary and information models defined by those specifications to describe the services that it provides. The payload for each Energy Interoperation service references a product defined using [EMIX]. EMIX schedules and

² In ISO and IEC terminology, portions that are not normative are *informative*. OASIS uses the term *non-normative* instead.
sequences are defined using [WS-Calendar]. Any additional schedule-related information
required by [EI] is expressed using [WS-Calendar].

• Since [EI] was published, a semantically equivalent but simpler [Streams] specification was
developed in the OASIS WS-Calendar Technical Committee. CTS uses that simpler
[Streams] specification.

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

Assumptions

1.6.1 Conformance with Energy Interoperation

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

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

1.6.2 Conformance with EMIX

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

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

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

1.6.3 Conformance with WS-Calendar Streams

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

1.6.3.1 Schedule Negotiation with WS-Calendar

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

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

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

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

See Section A.1 Normative References
WS-Calendar specifies rules for composing a schedule, perhaps with successive service calls. CTS uses this pattern to define instruments and schedule resource delivery.

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

Table 1-1: Core Semantics from WS-Calendar

<table>
<thead>
<tr>
<th>WS-Calendar Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>Duration is the length of time for an event scheduled using iCalendar or any of its derivatives. <em>It is unfortunate but true that the Duration “objects” defined in many programming languages are not identical. The [XCAL] Duration is a data type using the string representation defined in the iCalendar ([RFC5545]) Duration.</em></td>
</tr>
<tr>
<td>Interval</td>
<td>An Interval has as attributes a single Duration. An Interval may be part of a Sequence. An entire Sequence can be scheduled by scheduling a single Interval in that sequence. For this reason, Intervals are defined through Duration rather than through Start or End.</td>
</tr>
<tr>
<td>Sequence</td>
<td>A set of Intervals with defined temporal relationships. In Streams, Sequences have no gaps between intervals. A Sequence is re-locatable, i.e., it does not have a specific date and time. A Sequence may consist of a single Interval. An entire Sequence can be scheduled by scheduling a single Interval.</td>
</tr>
<tr>
<td>Gluon</td>
<td>A Gluon influences the serialization of Intervals in a Sequence, through inheritance and through schedule setting. The Gluon is similar to the Interval, but has no service or schedule effects until applied to an Interval or Sequence.</td>
</tr>
<tr>
<td>Payload</td>
<td>The placeholder in an Interval Component that holds that thing that occurs during an Interval. In Streams, this specification refers to the Payload conveyed by an Interval. In CTS 1.0, every Interval in a Stream inherits the same Product with price and quantity varying by Interval.</td>
</tr>
<tr>
<td>Lineage</td>
<td>The ordered set of Parents that results in a given inheritance or execution context for a Sequence.</td>
</tr>
<tr>
<td>Inheritance</td>
<td>A pattern by which information in Sequence is completed or modified by information from a Gluon. Information specified in one informational object is considered present in another that is itself lacking expression of that information.</td>
</tr>
<tr>
<td>Bequeath</td>
<td>A Parent Bequeaths attributes (Inheritance) to its Children.</td>
</tr>
</tbody>
</table>

Normative descriptions of the terms in the table above are in [WS-Calendar].
1.6.3.2 Streams and Inheritance

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

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

Figure 1-1: Basic Power Object from EMIX

A Stream conveys repeating intervals over time, with something that changes over the course of the schedule. The information that is true for every interval is expressed once only. The information that changes during each interval, is expressed as part of each interval.

Figure 1-2: Applying Basic Power to a Sequence

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

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

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

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

1.6.4 Compatibility with Facilities Smart Grid Information Model

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

The Architecture of EML-CTS is premised on distinct physical systems being able to interoperate by coordinating their production and consumption of energy irrespective of their ownership, motivations, or internal mechanisms. This specification defines messages and interactions of that interoperation.
CTS tenders and transactions can be used to express FSGIM load requests. CTS 1.0 uses single-interval Streams to express single-interval tenders in anticipation of the possible use of Streams in FSGIM-conformant communications.
2 Overview of Common Transactive Services

2.1 Scope of Common Transactive Services

CTS engages Transactive Resources, e.g. Distributed Energy Resources (DER), as well as any provider or consumer of energy, while making no assumptions as to their internal processes or technology. This specification supports agreements and transactional obligations, while offering flexibility of implementation to support specific approaches and goals of the various participants.

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

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

2.1.1 Applicability to Microgrids

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

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

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

[StructuredEnergy][SmartGridBusiness]

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.
- Information models for price and product communication.
- Payload definitions for Common Transactive Services

The following are out of scope for Common Transactive Services:

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

Section 1 describes standard bindings, which may be extended by The Energy Mashup Lab or others in the future.

2.2 Resources, Products and Instruments

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

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

A Product names a transactive resource that has been “chunked” for market. These chunks define the market granularity in quantity and in time. For example, the product may be 1 kW or 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 delivered beginning at 3:00 PM is a different Instrument than is the same Product at 11:00 PM. We use the semantics from financial markets to name the thing that is bought or sold is an Instrument.

A market considers all the tenders it has received offering to buy or sell an Instrument, using a Matching Engine to decide which can be cleared (satisfied) in full or in part. The 3:00pm instrument is traded independently from the 4:00pm instrument.

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

These level 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 half hour.</td>
</tr>
<tr>
<td>Instrument</td>
<td>The thing that tendered in a market, i.e., a Product instantiated by a particular begin time. Example: the Product beginning at 9:00 AM on April 3.</td>
</tr>
<tr>
<td>Party</td>
<td>A Party is an Actor that buys or sells Instruments in a CTS Marketplace. A Party may be referenced by a specific role in a specific interaction, such as Party or Counter Party.</td>
</tr>
<tr>
<td>Market</td>
<td>Where Products are traded by matching tenders submitted by Parties to buy or sell an Instrument</td>
</tr>
<tr>
<td>Marketplace</td>
<td>An actor wherein one or more Markets are conducted</td>
</tr>
<tr>
<td>Market Context</td>
<td>In EMIX, the Market Context is a URI identifying a Marketplace. In CTS, the Market Context MAY be resolvable and available so an Actor can retrieve machine-readable information describing a Marketplace. Examples of information that may be found in a Market Context include:</td>
</tr>
<tr>
<td></td>
<td>- A list of Products traded in this Marketplace</td>
</tr>
<tr>
<td></td>
<td>- Specific details of market operation (e.g., rules for registration and qualification, product quality, penalties for non-delivery, etc.)</td>
</tr>
<tr>
<td></td>
<td>- Currency used for market transactions</td>
</tr>
</tbody>
</table>
### 2.3 Common Transactive Services Architecture

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

The **Actor Model** names a style of system integration used for high scalability and resilience. The Actor Model uses a small number of simple messages to coordinate behavior among simple agents termed Actors. The Actor Model accomplishes complex behaviors through the fabric that hoists the Actors. This specification makes no assumptions about this fabric. Note that systems represented by Actors need not be actually simple; any modern facility incorporates a number of complex energy systems. This complexity is encapsulated within the Actors and the interactions are reduced to simple messages.

It is important to understand that an Actor may take on roles for its TE-related messages. In a Tender or Transaction, one Actor is the Party, the other is the Counterparty.

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

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

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

#### 2.3.1 Facets of CTS

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

This specification considers these contracts as facets, each separate from all other roles the Actor may perform. The facets are categorized in Table 2-2. Each facet may support several messages, as in submitting a Tender, acknowledging a Tender, and cancelling a Tender. The message or payload of each facet is similar to the information in an Instrument. “**Would you like to buy…?**” “**Would you like to sell…?**”

While each facet is discussed later in detail, Table 2-2 names these facets so we can describe the distinctions between them later. Those familiar with EI will recognize that each facet is mappable to an EI service.

<table>
<thead>
<tr>
<th>Facet</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration and Market Characteristics</td>
<td>A Party to potential transactions needs to know what products are traded in a Marketplace, the granularity (size and time and price), and other Marketplace information. While moving slowly over time, this can generally be viewed as static information about the Marketplace and its Products.</td>
</tr>
<tr>
<td>Facet</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</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 Actors in the Marketplace.</td>
</tr>
<tr>
<td>Transaction</td>
<td>A Transaction is created by the Market to memorialize a contract when a tender to buy and a tender to sell are matched. Both parties are notified of contract creation.</td>
</tr>
<tr>
<td>Position</td>
<td>At any moment, a Party has a position which represents the cumulative amount of an Instrument that an actor has previously transacted for that time interval. For example, a Position for an Instrument reflects the algebraic sum of all quantities previously bought or sold.</td>
</tr>
<tr>
<td>Measurement and Verification</td>
<td>After the Product as represented by an Instrument is bought sold and delivered, there is typically an asynchronous verification that what was purchased was in fact what was consumed or delivered. It is simplest to think of Delivery as a meter reading, although that meter may be virtual or computed.</td>
</tr>
<tr>
<td>Market Information including Quote</td>
<td>A Quote is a non-actionable indication of a potential price or availability of an instrument. Different Markets may restrict which actors may issue Quotes, say from only Market Agents or from External Actors.</td>
</tr>
</tbody>
</table>

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 Contracts (Transaction Facet).

### 2.3.2 Sides in Tenders and Transactions

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

- **Buy**, or
- **Sell**

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

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

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

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

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

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

Table 2-3: Responses

<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(^4).</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,(^5) 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>

\(^4\) As an example of the Correlation Pattern for messages

\(^5\) See e.g. [https://en.wikipedia.org/wiki/List_of_HTTP_status_codes](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.

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>A Duration that some markets MAY use to transfer trading off of hourly boundaries. A power distribution entity may experience disruption if there is a big price change on the hour. Offset enables a market rule to trade, for example, 3 minutes after the hour.</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 of the Instrument.</td>
</tr>
</tbody>
</table>

3.2 Semantic Elements from EMIX

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

3.2.1 Defining Resource

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

Table 3-2 Defining the Resource

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>Abstract base for describing all Resources. A Resource consists of a Name and a Description.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Item Description</td>
<td>The Item Description is the common name, same as 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 the Product

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

#### Table 3-3 Defining the Product

<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 described above.</td>
</tr>
<tr>
<td>Scale</td>
<td>Mantissa that specifies the size of the Resource Unit. For example, a Product denominated in megawatts has a mantissa 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 is beyond the product definition. For example, it is possible to trade only in Neighborhood Solar Power so long as the product clears, that is sold in the same interval it is bought,</td>
</tr>
</tbody>
</table>

In CTS, a Product "with" and "without" a Warrant are different Products. If an Actor wishes to buy energy with a “green Warrant”, however defined, then the Actor is responsible trading strategies to buy the un-warranted Product of the warranted Product is not available. Actors that wish to buy or sell Neighborhood Solar Power are responsible for submitting Tenders that expire in time to make alternate arrangements, or in cancelling Tenders before fulfillment. Market implementers should consider carefully whether they wish to support Warrants, as excessive segmentation will lead to markets that are “thinner” or “more congested”. Warrants add additional complexity of definition, i.e. such questions as “Is a Battery which stores power generated by Neighborhood Solar Power considered to be selling Neighborhood Solar Power when it discharges?” Alternately, if a market rule requires a Solar Panel to purchase a policy from other sources to insure its capability of Delivery, is that power considered Neighborhood Solar Power? The and similar questions would introduce the type of complexity that violates the design principles of CTS. Such complexity may also reduce interoperability of commodity Actors with specific Markets. Warrants were defined in EMX, and are permitted in CTS to support this complexity if desired.

### 3.2.3 Market-related Elements from EMIX

EMIX defines vocabulary used in market messages.

#### Table 3-4 Market-related elements from EMIX

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Party Id</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>Attribute</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Counter Party ID</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.</td>
</tr>
<tr>
<td>Side</td>
<td>An indication of what a Party intends to do 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 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 Marketplace, and need not be resolvable.</td>
</tr>
<tr>
<td>Standard Terms</td>
<td>Standard Terms are the machine-readable information about a marketplace, and the interactions it supports. In CTS, the Standard Terms include an enumeration of the Products tradable in this Marketplace.</td>
</tr>
</tbody>
</table>

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

Table 3-5 Standard Terms that define market interactions

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market Context Name</td>
<td>Text providing a descriptive name for a Marketplace. While the Name MAY be displayed in a user interface, but it is not meaningful to the Actors.</td>
</tr>
<tr>
<td>Currency</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>Offset</td>
<td>A Duration that some markets MAY use to transfer trading off of hourly boundaries. A power distribution entity may experience disruption if there is a big price change on the hour. For example, a distribution system operator (DSO) that operates multiple CTS marketplaces could opt to set a different offset on each Marketplace operated out of a given substation. In this model, a Marketplace could use an offset duration of 3 minutes to indicate that all tenders are based on three minutes after the hour.</td>
</tr>
<tr>
<td>Time Zone</td>
<td>A Time Zone indicates how all Times and Dates are expressed. The Marketplace Time Zone is a Standard Term.</td>
</tr>
<tr>
<td>Terms</td>
<td>EMIX Terms are extrinsic to the product delivery but effect how each party interacts with others. Terms may be tied to basic operational needs, or state schedules of availability, or suggest limits on bids and prices acceptable.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Products</td>
<td>An enumeration of the Products traded in this Marketplace. Note that similar products with and without Warrants are different products, each traded in their own Market.</td>
</tr>
</tbody>
</table>
4 Basic Interaction and Terminology

4.1 Structure of Common Transactive Services and Operations

The Common Transactive Services presented in this specification are only
- Transactive Services—for implementing tenders and transactions

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

4.2 Naming of Services and Operations

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

- Create—Created An object is created and sent to the other Party
- Cancel—Canceled A previously created request is canceled

For example, to construct an operation name for the EiTender facet, "Ei" is concatenated with the name
fragment (verb) as listed. An operation to cancel an outstanding Tender is called EiCancelTender.6

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

4.3 Payloads and Messages

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

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

4.4 Description of the Facets and Payloads

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

---

6 This pattern was developed and is used by current work in the IEC Technical Committee 57 (Power Systems).
### 4.5 Responses

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

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

It is **MANDATORY** to return as appropriate both errors and success in responses.\(^7\)

The class diagram in Figure 4-1 shows the generic response in CTS 1.0.

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

\(^7\) This contrasts with Energy Interoperation, where it is not mandatory to return any responses if the entire EICancelTender service operation was completed successfully. The pattern in Energy Interoperation is to return those that have failed (required) and those that succeeded (optional).
EML-CTS uses response code 200 for success.
5 Configuration and Market Characteristics Facet

Each Event and Service in Energy Interoperation takes place within a Marketplace. All interactions in a Marketplace are subject to common rules of engagement which are termed a Market Context. The Market Context defines the behaviors that each Party can expect from the other. This concept with some simplification is part of the Common Transactive Services.

5.1 The Market Context

Market Contexts are resolvable URIs and are used to express market information that rarely changes, so it is not necessary to communicate it with each message. For any market context, there are standing terms and expectations about product offerings. If these standing terms and expectations are not known, many exchanges may need to occur before finding products that meet those expectations. If these expectations are only known through local knowledge, then national and international products need to be re-configured for each local market that they enter. If all market information were to be transmitted in every information exchange, messages based on EMIX would be overly repetitive.

The Market Context for CTS is simplified from that in Energy Interoperation and extended for use of standard terms.

5.2 Interaction Pattern for the Market Context Facet

The Market Context Facet enables a Party to request the details of a Marketplace by using its Market Context. Parties MAY be able to request and compare Market Contexts to select which markets to participate in. Such Interactions are out of scope for this specification.

![MarketContext Sequence Diagram](image)

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

The Market Context service can retrieve the full information associated with an EiMarketContext. There is one operation and a responding operation. Profiled and simplified market context information is planned for a future release.
5.3 Information Model for the EiMarketContext Facet

Simplified profile pending.

5.4 Operation Payloads for the EiMarketContext Facet

Payloads including terms pending.

Figure 5-2: UML of Market Context Service payloads
6 Tender Facet

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

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

For transactive services, the roles are Parties and Counterparties.

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

6.1 Tenders as a Pre-Transaction Payloads

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

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

Table 6-1: Pre-Transaction Tender Services

<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>EiCreateTenderType</td>
<td>EiCreatedTenderType</td>
<td>Create and send Tender</td>
</tr>
<tr>
<td>EiTender</td>
<td>EiCancelTenderType</td>
<td>EiCanceledTenderType</td>
<td>Cancel one or more Tenders</td>
</tr>
<tr>
<td>EiTender</td>
<td>EIDistributeTenderType</td>
<td>None</td>
<td>Distribute a list of Tenders to a transport or messaging system defined list of parties</td>
</tr>
</tbody>
</table>

6.1.1 Interaction Pattern for the Tender Facet

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

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

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

Figure 6-2: Class EiTenderType

The attributes of EiTender are shown in the following table.
<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 offered</td>
<td>Total price is the product of price and quantity</td>
</tr>
<tr>
<td>Quantity</td>
<td>The quantity of the product being offered</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></td>
</tr>
<tr>
<td>Transactive State</td>
<td>The transactive state of this payload (tender)</td>
<td>See below</td>
</tr>
</tbody>
</table>

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

The [UML] class diagram describes the payloads for the EiTender service operations.
Figure 6-4: UML Class Diagram for the Operation Payloads for the EiTender Service
7 Transaction Facet

7.1 Transaction Services

This section presents the Transaction Facet payloads.

In the contributed specification, market context and product are implied. This section makes them explicit, consistent with the definitions in Section 3.

Canceling or modifying transactions is not permitted in either CTS or Energy Interoperation. Following the approach in distributed agreement protocols\(^8\), compensating tenders and transactions SHOULD be created as needed to compensate for any effects.\(^9\)

<table>
<thead>
<tr>
<th>Service</th>
<th>Request</th>
<th>Response</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EiTransaction</td>
<td>EiCreateTransactionType</td>
<td>EiCreatedTransactionType</td>
<td>Create and acknowledge creation of a Transaction</td>
</tr>
</tbody>
</table>

7.1.1 Interaction Pattern for the EiTransaction Service

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

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

7.1.2 Information Model for the EiTransaction Service

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

---

\(^8\) See, e.g., WS-Transaction and WS-BusinessActivity.

\(^9\) This is consistent with the way that distributed agreement protocols such as [WS-BusinessActivity] manage compensation rather than cancelation.
Although an EiT Tender object includes the original EiT Tender, the EiT Transaction carries its own Transactive State.

```
class EiTransactionType

<table>
<thead>
<tr>
<th>EiTransactionType</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ tender: EiT TenderType</td>
</tr>
<tr>
<td>+ transactionID: TransactionIdType</td>
</tr>
<tr>
<td>+ transactiveState: TransactiveStateType</td>
</tr>
</tbody>
</table>
```

Figure 7-2: UML Class Diagram of EiT Transaction

The attributes of EiT Transaction are shown in the following table.

Table 7-2: EiT Transaction 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 TenderId</td>
</tr>
<tr>
<td>Transactive State</td>
<td>The transactive state of this payload is transaction</td>
<td>See Figure 6-3-3 Enumeration TransactiveStateType</td>
</tr>
</tbody>
</table>
7.1.3 Operation Payloads for the EiTransaction Facet

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

![UML Class Diagram of EiTransaction Service Operation Payloads](image)

7.2 Comparison of Transactive Payloads

In this section we show the payloads for the Transactive Facet with those for the Tender Facet.
Figure 7-4: UML Diagram comparing all Transactive Payloads

```plaintext
class All Transactive and Information Payloads

EiCreateTenderType
+ counterPartyId: ActorIdType
+ eITender: EITenderType[1..*]
+ partyId: ActorIdType
+ requestId: RequestIdType

EiCreatedTenderType
+ counterPartyId: ActorIdType[0..1]
+ eIResponse: EIResponseType
+ partyId: ActorIdType
+ responses: ArrayOfResponses
+ tenderId: RequestIdType[0..*]

EiCancelTenderType
+ counterPartyId: ActorIdType[0..1]
+ partyId: ActorIdType
+ requestId: RequestIdType
+ tenderId: RequestIdType[1..*]

EiCanceledTenderType
+ counterPartyId: ActorIdType[0..1]
+ eIResponse: EIResponseType
+ partyId: ActorIdType
+ responses: ArrayOfResponses[0..1]

EiCreateTransactionType
+ counterPartyId: ActorIdType
+ eITransaction: EITransactionType[1..*]
+ partyId: ActorIdType
+ requestId: RequestIdType

EiCreatedTransactionType
+ counterPartyId: ActorIdType
+ eIResponse: EIResponseType
+ partyId: ActorIdType
+ responses: ArrayOfResponses[0..1]
+ transactionId: TransactionIdType[0..*]

EiRequestMarketContextType
+ marketContext: MarketContextType[1..*]
+ partyId: ActorIdType[0..1]
+ requestId: RequestIdType

EiReplyMarketContextType
+ eIResponse: EIResponseType
+ responses: ArrayOfResponses[0..1]
```
8 Position Facet

Pending. Follows the definition of the EiPosition Service in the Energy Mashup Lab open source implementation of CTS.
9 Measurement and Verification Facet

Pending. Following EiDelivery Payloads from [Eli]
10 Market Information Facet—Quotes and Tickers

Pending.

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

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

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

11.1 JSON

TODO—JSON Schema available

11.2 XML Schema

TODO—XML Schema available

11.2.1 XML Namespaces

11.3 Simple Binary Encoding

TODO—Work in progress
12 Conformance

(Note: The OASIS TC Process requires that a specification approved by the TC for public review, or for publication at the Committee Specification or OASIS Standard level must include a separate section, listing a set of numbered conformance clauses, to which any implementation of the specification must adhere in order to claim conformance to the specification (or any optional portion thereof). This is done by listing the conformance clauses here.

For the definition of "conformance clause," see OASIS Defined Terms.


Remove this note before submitting for publication.)

Pending update to Facet terminology

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

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

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

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

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

12.1 Claiming Conformance to Common Transactive Services

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

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

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

A.1 Normative References

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

[RFC8174]

[JSON]

[MIN]

[RFC2119]

[RFC2246]

[SBE]

[Streams]

[WS-Calendar-PIM]

[XSD]

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]
National Institute of Standards and Technology, NIST Framework and Roadmap for Smart Grid
Interoperability Standards, Release 1.0, January 2010,

[CTS2016]

[EML-CTS]

[FSGIM]

[iCalendar]
B. Desruisseaux, Internet Calendaring and Scheduling Core Object Specification (iCalendar),
See also

[GridFaultResilience]

[Micromarkets]

[RFC3552]

[SmartGridBusiness]

[StructuredEnergy]
Structured Energy: Microgrids and Autonomous Transactive Operation,

[TRM] (Transactive Resource Management)

[UML]
Appendix B. Security and Privacy Considerations

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

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

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

We encourage editors and TC members concerned with this subject to read Guidelines for Writing RFC Text on Security Considerations, IETF [RFC3552], for more information.
Appendix C. Glossary of Terms and Abbreviations Used in this document

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

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

<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 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 D. Acknowledgments

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

D.1 Special Thanks

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

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

[Participant Name, Affiliation | Individual Member]

D.2 Participants

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

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

<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>
</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.