



The Energy Mashup Lab

Common Transactive Services 1.0

The Energy Mashup Lab Specification

30 November 2020

Editor:

William T. Cox (cto@theenergymashuplab.org [wtcox@coxsoftwarearchitects.com](mailto:wtcx@coxsoftwarearchitects.com))

Toby Considine (director@theenergymashuplab.org toby.considine@gmail.com)

Additional artifacts:

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

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

Related work:

This specification is related to:

- OASIS Energy Interoperation v1.0 (OASIS Standard) and its TeMIX Profile
- OASIS WS-Calendar Platform-Independent Model v1.0
- OASIS WS-Calendar Streams v1.0

Abstract:

TBD

Status:

This document is produced by The Energy Mashup Lab

Copyright © The Energy Mashup Lab 2020. All Rights Reserved.

Portions taken from *Energy Interoperation Version 1.0*. Edited by Toby Considine. 11 June 2014. OASIS Standard Copyright © OASIS Open 2014

See full copyright notices below.

36 Notices

37 Copyright © The Energy Mashup Lab 2020. All Rights Reserved.

38 Portions Copyright © OASIS Open 2014. All Rights Reserved.

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

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

51 The limited permissions granted above are perpetual and will not be revoked by The Energy
52 Mashup Lab or its successors or assigns.

53 This document and the information contained herein is provided on an "AS IS" basis and OASIS
54 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT
55 LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN
56 WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF
57 MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

58 The Energy Mashup Lab requests that any party that believes it has patent claims that would
59 necessarily be infringed by implementations of this Specification or OASIS Standard, to notify
60 The Energy Mashup Lab Director and provide an indication of its willingness to grant patent
61 licenses to such patent.

62 The Energy Mashup Lab invites any party to contact The Energy Mashup Lab Director if it is
63 aware of a claim of ownership of any patent claims that would necessarily be infringed by
64 implementations of this specification by a patent holder that is not willing to provide a license to
65 such patent. The Energy Mashup Lab may include such claims on its website, but disclaims any
66 obligation to do so.

67 The Energy Mashup Lab takes no position regarding the validity or scope of any intellectual
68 property or other rights that might be claimed to pertain to the implementation or use of the
69 technology described in this document or the extent to which any license under such rights might
70 or might not be available; neither does it represent that it has made any effort to identify any such
71 rights.

72 Table of Contents

73	1	Introduction.....	7
74	1.1	Generality of the Common Transactive Services.....	8
75	1.2	Application of the Common Transactive Services.....	8
76	1.3	The EML-CTS System.....	9
77	1.4	Terminology.....	9
78	1.5	Normative References.....	9
79	1.6	Non-Normative References.....	10
80	1.7	Naming Conventions.....	11
81	1.8	Editing Conventions.....	11
82	1.9	Architecture.....	12
83	1.9.1	Security Considerations.....	12
84	1.9.2	CTS Extended Example.....	12
85	2	Overview of Common Transactive Services.....	13
86	2.1	Scope of Common Transactive Services.....	13
87	2.2	Specific scope statements.....	13
88	2.3	Assumptions.....	13
89	2.3.1	Conformance with Energy Interoperation.....	13
90	2.3.2	Conformance with EMIX.....	14
91	2.3.3	Conformance with WS-Calendar Streams.....	14
92	2.3.4	Compatibility with Facilities Smart Grid Information Model.....	14
93	2.4	Common Transactive Services Architecture.....	15
94	2.4.1	Sides in Tenders and Transactions.....	16
95	2.4.2	Semantic Composition.....	16
96	2.5	Products and Instruments.....	17
97	3	Services and Operations.....	19
98	3.1	Structure of Common Transactive Services and Operations.....	19
99	3.2	Naming of Services and Operations.....	19
100	3.3	Payloads and Messages.....	20
101	3.4	Description of the Services and Operations.....	20
102	3.5	Responses.....	20
103	4	Transactive Services.....	23
104	4.1	Pre-Transaction Services.....	23
105	4.1.1	Interaction Pattern for the EiTender Service.....	23
106	4.1.2	Information Model for the EiTender Services.....	24
107	4.1.3	Operation Payloads for the EiTender Service.....	26
108	4.2	Transaction Management Services.....	27
109	4.2.1	Interaction Pattern for the EiTransaction Service.....	27

110	4.2.2 Information Model for the EiTransaction Service	27
111	4.2.3 Operation Payloads for the EiTransaction Service.....	28
112	4.3 Comparison of Transactive Payloads	29
113	5 Market Information	31
114	5.1 The Market Context.....	31
115	5.2 Interaction Pattern for the Market Context Service	31
116	5.3 Information Model for the EiMarketContext Service	32
117	5.4 Operation Payloads for the EiMarketContext Service	32
118	6 Bindings.....	33
119	6.1 JSON.....	33
120	6.2 XML Schema	33
121	6.2.1 XML Namespaces	33
122	6.3 Simple Binary Encoding.....	33
123	7 Conformance of Common Transactive Services	35
124	8 Claiming Conformance to Common Transactive Services	37
125	Appendix A. Acknowledgments.....	39
126	Appendix B. Background and Development history	40
127	Appendix C. Glossary	41
128	Appendix D. Revision History	42
129		

130	Index to Figures	
131	Figure 3-1: Example of generic error response for a service operation.....	21
132	Figure 4-1: UML Sequence Diagram for the EiTender Service.....	24
133	Figure 4-2: Class EiTenderType.....	24
134	Figure 3-3 Enumeration TransactiveStateType.....	25
135	Figure 4-4: UML Class Diagram for the Operation Payloads for the EiTender Service.....	26
136	Figure 4-5: UML Sequence Diagram for the EiTransaction Service	27
137	Figure 4-6: UML Class Diagram of EiTransaction	28
138	Figure 4-7: UML Class Diagram of EiTransaction Service Operation Payloads.....	28
139	Figure 4-8: UML Diagram comparing all Transactive Payloads	29
140	Figure 5-1: UML Sequence diagram for Market Context service.....	31

141		
142	Index to Tables	
143	Table 3-1: EiResponse Attributes.....	21
144	Table 4-1: Pre-Transaction Tender Services.....	23
145	Table 4-2: EiResponse Attributes.....	25
146	Table 4-3: Transaction Management Service.....	27
147	Table 4-4: EiTransaction Attributes.....	28

149 1 Introduction

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

154 Transactive Resource Management (TRM) has been used for many non-energy resources, such
155 as water delivery, network bandwidth, and even internet advertising. The initial research in TRM
156 used a market to allocate heat within a commercial building [TRM]. In TRM, a resource is
157 defined as a tradable commodity whose value depends on price, location, and time of delivery
158 [EMIX]¹. TRM balances supply and demand over time using automated voluntary transactions
159 between market participants.

160 TRM applied to energy is commonly referred to as Transactive Energy (TE). Neither EI nor CTS
161 specifies which technologies participants will use; rather they define a technology-agnostic
162 interface to enable accelerated market development of such technologies.

163 TRM is a means to allocate transactable energy resources including the delivery of commodities
164 such as electrical energy, electrical power, natural gas, and thermal energy such as steam, hot
165 water, or chilled water. Transactable energy resources also include the capability to deliver
166 resources, such as transmission line capacity and flow-rate capacity.²

167 The Common Transactive Services are a lightweight profile of the OASIS Energy Interoperation
168 specification. All CTS messages are simple and make no assumptions about the systems behind
169 the messages.

170 The target actors for CTS include but are not limited to

- 171 • Smart Buildings/Homes/Industrial Facility
- 172 • Building systems/devices
- 173 • Business Enterprises
- 174 • Vehicles
- 175 • Microgrids
- 176 • IoT (Internet of Things) devices

177 Transactive Energy has the potential to make our electrical system more efficient, by better
178 matching supply and demand in real time. TE enable actors to use energy when it is less
179 expensive and produce energy when it's more valuable, thus reducing reliance on distant
180 suppliers while maximizing use of local power sources. TE relies on markets and consumer
181 choice to make better decisions about power supply and use.

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

¹ See http://docs.oasis-open.org/emix/emix/v1.0/cs02/emix-v1.0-cs02.html#_Toc319594576

² In North American wholesale electricity markets, transmission rights are bought and sold.

184 central or cloud-based markets. Central markets discount local decision making while
185 introducing new barriers to resilience. Others rely on a single price-setting supplier. None are
186 interoperable either at the system level or for the actors involved.

187 Turning back to the more general Transactive Resource Management, nothing in CTS restricts its
188 use to electricity-based markets. Natural gas markets share many characteristics with electricity
189 markets. Local thermal energy distribution systems can balance electricity markets while having
190 their own surpluses and shortages.

191 Progress toward TE can be accelerated if a common interaction model is used across systems.
192 Looked at from another perspective, a client written for a participant in one such system should
193 be able to interoperate with another TE system. The Common Transactive Services from The
194 Energy Mashup Lab fulfil that promise.

195 **1.1 Generality of the Common Transactive Services**

196 CTS can be used to trade (Tender, Transact on) any [Transactive Resource]. While our focus is
197 generally on electrical energy or power, in the rest of this document we use **[power]** to mean
198 *electrical energy or power or any other Transactive Resource*.

199 The actual product in EML-CTS (next section) is implicit in the market with which one
200 communicates. This limits complexity of product definition to a useful level, so market and
201 product definition may be considered configuration rather than data.

202 **1.2 Application of the Common Transactive Services**

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

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

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

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

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

222 1.3 The EML-CTS System

223 In 2015, the US National Institute for Standards and Technology (NIST) began the Transactive
224 Energy Modeling and Simulation Challenge (TE Challenge). A report delivered to TE Challenge
225 in 2016 [CTS2016] defined a small subset of Energy Interoperation, known as the Common
226 Transactive Services.

227 In 2019, The Energy Mashup Lab, under contract to NIST, began developing an open source
228 software system (Apache 2.0 license) that uses a robust financial or “order book” market for
229 peer-to-peer transactions. The system architecture separates market interactions from the actors
230 buying and selling power. The architecture also permits changing the market engine itself. This
231 implementation is called EML-CTS and is available today.³

232 In creating EML-CTS members of The Energy Mashup Lab further simplified CTS as a smaller
233 subset profile of the Energy Interoperation TeMIX profile [TEMIX].

234 TE demonstrations have used different market engines, including double auction markets. EML-
235 CTS was designed to be able to use any (e.g. either, both, or some other market engine) while
236 keeping interactions between systems and the market unchanged.

237 The EML-CTS 1.0 implementation uses Java class definitions similar to those in the UML in this
238 specification. Messages are sent using REST POST operations, and JSON serialization uses the
239 Java classes.

240 1.4 Terminology

241 The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”,
242 “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this
243 document are to be interpreted as described in [RFC2119]

244 1.5 Normative References

245 [EMIX] *Energy Market Information Exchange (EMIX) Version 1.0*, January 2012,
246 OASIS Committee Specification 02, Latest version: [http://docs.oasis-](http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html)
247 [open.org/emix/emix/v1.0/emix-v1.0.html](http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html).

248 [EnergyInterop] *Energy Interoperation Version 1.0*. Edited by Toby Considine. 11 June
249 2014. OASIS Standard. [http://docs.oasis-](http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html)
250 [open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html](http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html).

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

253 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
254 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

255 [RFC2246] T. Dierks, C. Allen *Transport Layer Security (TLS) Protocol Version 1.0*,
256 <http://www.ietf.org/rfc/rfc2246.txt>, IETF RFC 2246, January 1999.

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

³ <https://github.com/EnergyMashupLab/eml-cts>

- 259 **[WS-Calendar-PIM]** *WS-Calendar Platform Independent Model (PIM) Version 1.0.*
 260 Edited by William Cox and Toby Considine. 21 August 2015. OASIS
 261 Committee Specification. [http://docs.oasis-open.org/ws-calendar/ws-](http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html)
 262 [calendar-pim/v1.0/ws-calendar-pim-v1.0.html](http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html).
- 263 **[MIN]** *WS-Calendar Minimal PIM-Conformant Schema Version 1.0.* Edited by
 264 William Cox and Toby Considine. 26 August 2016. OASIS Committee
 265 Specification. [http://docs.oasis-open.org/ws-calendar/ws-calendar-](http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.html)
 266 [min/v1.0/ws-calendar-min-v1.0.html](http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.html).
- 267 **[Streams]** *Schedule Signals and Streams Version 1.0.* Edited by Toby Considine and
 268 William T. Cox. 18 September 2016. OASIS Committee Specification.
 269 <http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html>.
- 270 **[XSD]** *W3C XML Schema Definition Language (XSD) 1.1.* Part 1: Structures, S
 271 Gao, C. M. Sperberg-McQueen, H Thompson, N Mendelsohn, D Beech,
 272 M Maloney <http://www.w3.org/TR/xmlschema11-1/>, April 2012, Part 2:
 273 Datatypes, D Peterson, S Gao, A Malhotra, C. M. Sperberg-McQueen, H
 274 Thompson, P Biron, <http://www.w3.org/TR/xmlschema11-2/> April 2012

275 1.6 Non-Normative References

- 276 **[Actors]** C. Hewitt, "Actor Model of Computation: Scalable Robust Information
 277 Systems," arxiv.org, 2010.
- 278 **[Framework]** National Institute of Standards and Technology, *NIST Framework and*
 279 *Roadmap for Smart Grid Interoperability Standards, Release 1.0*, January
 280 2010,
 281 [http://nist.gov/public_affairs/releases/upload/smartgrid_interoperability_fi-](http://nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf)
 282 [nal.pdf](http://nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf)
- 283 **[CTS2016]** Cox, W. T., Cazalet, E., Krstulovic, A., Miller, W., & Wijbrandi, W.
 284 *Common Transactive Services*. TESC 2016. Available at
 285 [http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016](http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/Common%20Transactive%20Services%20Paper%2020160516.pdf)
 286 [/Common%20Transactive%20Services%20Paper%2020160516.pdf](http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/Common%20Transactive%20Services%20Paper%2020160516.pdf)
- 287 **[EML-CTS]** Energy Mashup Lab Common Transactive Services (open source
 288 software) <https://github.com/EnergyMashupLab/eml-cts>
- 289 **[FSGIM]** *Facility smart grid information model*. ISO 17800.
 290 <https://www.iso.org/standard/71547.html> 2017
- 291 **[iCalendar]** *Internet Calendaring and Scheduling Core Object Specification*
 292 *(iCalendar)*, <https://tools.ietf.org/html/rfc5545>. 2009, B. Desruisseaux,
 293 See also *Calendar Availability*, <https://tools.ietf.org/html/rfc7953>, C.
 294 Daboo, M. Douglas. 2016
- 295 **[SmartGridBusiness]** Toby Considine and William Cox, *Smart Loads and Smart Grids—*
 296 *Creating the Smart Grid Business Case*. Grid-Interop 2009. Available at
 297 [http://coxsoftwarearchitects.com/Resources/Grid-](http://coxsoftwarearchitects.com/Resources/Grid-Interop2009/Smart%20Loads%20and%20Smart%20Grids.pdf)
 298 [Interop2009/Smart%20Loads%20and%20Smart%20Grids.pdf](http://coxsoftwarearchitects.com/Resources/Grid-Interop2009/Smart%20Loads%20and%20Smart%20Grids.pdf)
- 299 **[StructuredEnergy]** *Structured Energy: Microgrids and Autonomous Transactive*
 300 *Operation*, [http://coxsoftwarearchitects.com/Resources/ISGT_2013/ISGT-](http://coxsoftwarearchitects.com/Resources/ISGT_2013/ISGT-Cox_StructuredEnergyPaper518.pdf)
 301 [Cox_StructuredEnergyPaper518.pdf](http://coxsoftwarearchitects.com/Resources/ISGT_2013/ISGT-Cox_StructuredEnergyPaper518.pdf). Innovative Smart Grid Technologies
 302 2013 (IEEE).

- 303 **[GridFaultResilience]** William Cox and Toby Considine, *Grid Fault Recovery and*
 304 *Resilience: Applying Structured Energy and Microgrids*. IEEE Innovative
 305 Smart Grid Technologies 2014. Available at
 306 [http://coxsoftwarearchitects.com/Resources/ISGT_2014/ISGT2014_GridF](http://coxsoftwarearchitects.com/Resources/ISGT_2014/ISGT2014_GridFaultRecoveryResilienceStructuredMicrogrids_Paper.pdf)
 307 [aultRecoveryResilienceStructuredMicrogrids_Paper.pdf](http://coxsoftwarearchitects.com/Resources/ISGT_2014/ISGT2014_GridFaultRecoveryResilienceStructuredMicrogrids_Paper.pdf)
 308 **[TRM]** B. Huberman and S. H. Clearwater, *Thermal markets for controlling*
 309 *building environments*, Energy Engineering, vol. 91, no. 3, pp. 26- 56,
 310 January 1994.
 311 **[UML]** Object Management Group, *Unified Modeling Language (UML), V2.4.1*,
 312 August 2011. <http://www.omg.org/spec/UML/2.4.1/>

313 1.7 Naming Conventions

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

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

318 For example,

```
319   <element name="componentType" type="ei:ComponentType"/>
```

320 For the names of types within XSD files, the names follow the UpperCamelCase convention
 321 with all names starting with a lower-case letter prefixed by “type-“. For example,

```
322   <complexType name="ComponentServiceType">
```

323 For clarity in UML models the suffix “type” is not always used.

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

327 JSON and where possible SBE names follow the same conventions.

328 1.8 Editing Conventions

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

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

333 Information in the **Meaning** column of the tables is normative. Information appearing in the
 334 **Notes** column is explanatory and non-normative.⁴

335 Examples and Appendices are non-normative.

⁴ In ISO and IEC terminology, portions that are not normative are *informative*. OASIS uses the term *non-normative* instead.

336 **1.9 Architecture**

337 Services requests and responses are public actions of each interoperating system. Service actions
338 are independent from private actions behind the interface (i.e., device control actions). A service
339 is used without needing to know all the details of its implementation. Services are generally paid
340 for results, not effort.

341 **1.9.1 Security Considerations**

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

350 **1.9.2 CTS Extended Example**

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

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

359 In a similar way, in considering a topology of microgrids, any participant in the original
360 micromarket MAY itself represent a contained autonomous microgrid or, in fact, any
361 autonomous entity whether or not it is managed in turn by a market.
362 [StructuredEnergy][SmartGridBusiness]

363 2 Overview of Common Transactive Services

364 2.1 Scope of Common Transactive Services

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

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

369 No particular agreements are endorsed, proposed or required in order to implement this
370 specification. Energy market operations are beyond the scope of this specification although
371 interactions that enable management of the actual delivery and acceptance are within scope but
372 not included in CTS 1.0.⁵

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

375 2.2 Specific scope statements

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

- 378 • Interaction patterns to support transactive energy.
- 379 • Information models for price and product communication.
- 380 • Payload definitions for Common Transactive Services

381 The following are out of scope for Common Transactive Services:

- 382 • Requirements specifying the type of agreement, contract, product definition, or tariff used
383 by a particular market.
- 384 • Computations or agreements that describe how power is sold into or sold out of a market.

385 Section 0 describes standard bindings, which may be extended by The Energy Mashup Lab or
386 others in the future.

387 2.3 Assumptions

388 2.3.1 Conformance with Energy Interoperation

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

⁵ See e.g. Energy Interoperation EiDelivery Service https://docs.oasis-open.org/energyinterop/ei/v1.0/os/energyinterop-v1.0-os.html#_Toc388604056

392 Energy Interop assumes an Energy Services Interface (ESI) as the external face of the energy-
393 consuming or supplying node. Energy Interop defines an end-to-end interaction model; this
394 characteristic is shared by CTS.

395 **2.3.2 Conformance with EMIX**

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

399 The EMIX product definition, as included in the Transactive Resource, is implied in CTS 1.0.
400 Future CTS specifications may include market context from EMIX and Energy Interop, as well
401 as other information on products and markets including market terms.

402 **2.3.3 Conformance with WS-Calendar Streams**

403 The WS-Calendar specifications⁶ express sequences and enable negotiation of schedules in a
404 manner that is semantically compatible with human schedules, i.e., [iCalendar]. The WS-
405 Calendar Platform Independent Model (PIM) [WsCalendar-PIM] defines common semantics for
406 the specifications. WS-Calendar is the standard under the NIST Smart Grid Roadmap for all such
407 communication.

408 WS-Calendar is used to describe products whose value changes with time of delivery, and again
409 into Energy Interoperation, which uses Transactive Resources.

410 This specification bases its representation of single intervals on Schedule Signals and Streams
411 [Streams], a WS-Calendar-PIM conforming specification for expressing consecutive occurrences
412 of schedules or products.

413 A current implementation, EML-CTS, transacts a single interval at a time expressed as a single-
414 interval Stream. Energy systems supported by CTS-based markets may express their
415 requirements and capabilities over time using multi-interval Streams or in separate single-
416 interval Streams.

417 **2.3.4 Compatibility with Facilities Smart Grid Information Model**

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

425 The Architecture of EML-CTS is premised on distinct physical systems being able to
426 interoperate by coordinating their production and consumption of [power]⁷ irrespective of their

⁶ See Section 1.5 Normative References

⁷ See Section 1.1.

427 ownership, motivations, or internal mechanisms. This specification defines messages and
428 interactions of that interoperation.

429 CTS tenders and transactions can be used to express FSGIM load requests. CTS 1.0 uses single-
430 interval Streams to express single-interval tenders in anticipation of the possible use of Streams
431 in FSGIM-conformant communications.⁸

432 **2.4 Common Transactive Services Architecture**

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

437 The Energy Mashup Lab uses the Actor Model, which can be implemented in SOA with a few
438 lightweight Service Operations. The Lab adapted the SOA model of Energy Interoperation into
439 an Actor-to-Actor model that requires fewer and lighter weight messages.

440 The Actor Model names a style of system integration used for high scalability and resilience.⁹
441 The Actor Model uses a small number of simple messages to coordinate behavior among simple
442 agents termed Actors. Note that Actors need not be actually simple; any complexity in the Actors
443 are reduced to simple messages.

444 Simple messages are an essential aspect of actor architectures. The Common Transactive
445 Services are a lightweight profile of the OASIS Energy Interoperation specification, and in fact
446 of the TEMIX profile of Energy Interoperation. All CTS messages are simple and make no
447 assumptions about the systems behind the messages.

448 Just as the market participants present simple messages, so too, does the market. The internals of
449 a market contain a market engine to match tenders and to declare contracts. The rules used to
450 match tenders could be nearly instant order book, or periodic double auction, or some other
451 model. This complexity is hidden. The market receives tenders and announces contracts. Only
452 the simple messages of CTS are used.

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

456 All interactions and actors below are described as if for Actors in electrical energy markets. For
457 use in other transactive energy markets, or even transactive resource markets, only the product or
458 resources would be changed.

459 An actor takes on a role, for example a business role as a Party. In the UML model, *PartyId* and
460 *CounterPartyId* inherit from *ActorId* which in turn inherits from class *UidType*.

⁸ Conformance with the Energy Interoperation TEMIX Profile may require single intervals.

⁹ See C. Hewitt, "Actor Model of Computation: Scalable Robust Information Systems,"
arxiv.org, 2010, or C. Hewitt, "A Universal Modular Actor Formalism for Artificial
Intelligence," ICJA, 1973, or many other references

461 2.4.1 Sides in Tenders and Transactions

462 At any moment, a Party has a *position* which represents the cumulative amount of power (or
463 other product) that an actor has previously transacted for that time interval.

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

- 465 • Buy, or
- 466 • Sell

467 A Party selling [power] relative to its current position takes the Sell Side of the Transaction. A
468 Party buying [power] relative to its current position takes the Buy Side of the Transaction.

469 From the perspective of the market, there is no distinction between a party selling additional
470 power and party selling from its previously acquired position. An Actor representing a generator
471 generally takes the Sell side of a transaction. An Actor representing a consumer generally takes
472 the Buy side of a transaction. A generator may take the Buy Side of a Transaction in order to
473 reduce its own generation, in response either to changes in physical or market conditions or to
474 reflect other commitments made by the actor. A consumer may choose to sell from its current
475 position if its plans change, or if it receives an attractive price. A power storage system actor
476 may choose to buy or sell from interval to interval, consistent with its operating and financial
477 goals.

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

481 2.4.2 Semantic Composition

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

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

- 485 • EMIX describes price and product for electricity markets.
- 486 • WS-Calendar communicates schedules and sequences of operations. This specification
487 uses the [Streams] optimization which is a standalone specification, rather than part of
488 Energy Interoperation 1.0.
- 489 • Energy Interoperation uses the vocabulary and information models defined by those
490 specifications to describe the services that it provides. The payload for each Energy
491 Interoperation service references a product defined using [EMIX]. EMIX schedules and
492 sequences are defined using [WS-Calendar]. Any additional schedule-related information
493 required by [EnergyInterop] is expressed using [WS-Calendar].
- 494 • Since [EnergyInterop] was published, a semantically equivalent but simpler [Streams]
495 specification was developed in the OASIS WS-Calendar Technical Committee¹⁰. CTS
496 uses that simpler [Streams] specification.

¹⁰ https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-calendar

497 In effect, CTS is a profile of Energy Interoperation but with simplified information models and
498 defines only payloads, not the messaging.

499 CTS 1.0 supports a market for a single product (say energy) in multiple time intervals.

500 Product definition in CTS 1.0 are implicit but characteristics can be discerned using the market
501 context requests and responses.

502 Future development of CTS is planned to include discoverable market and product description
503 information through the EMIX and Energy Interoperation Market Context. The EMIX Market
504 Context associates market rules and catalogs the products tradeable.

505 Future versions of CTS will support multiple markets and multiple products.

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

507 **2.5 Products and Instruments**

508 An EMIX product is a specific resource. To transact that product, it is packaged in a tender with
509 a specific quantity for a specific duration.

510 CTS transacts power products at specific times. The thing traded (often called an *instrument* in
511 financial markets) includes the product together with quantity and the time interval. Tenders
512 become contracts when a tender to buy some quantity of a product are matched with a tender to
513 sell the same quantity of that product.¹¹

¹¹ Most underlying matching engines and markets trade instruments, which are associated with a product definition—which may not be fully expressed in an EMIX product definition—plus quantity and time.

514 3 Services and Operations

515 This section re-iterates terms and simpler models from [EnergyInterop]. That specification is
516 normative.

517 This terminology is used through all payload definitions presented in this specification.

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

522 For transactive services any party may receive tenders (priced offers) of service and possibly
523 make tenders (priced offers) of service.

524 Any party using transactive energy services may own generation or distributed generation or
525 reduce or increase energy from previously transacted energy amounts. The dispatch of these
526 resources and the use of energy by a party are influenced by tenders between Parties that may
527 result in new Transactions and changes in operations.

528 3.1 Structure of Common Transactive Services and Operations

529 The Common Transactive Services presented in this specification are only

- 530 • Transactive Services—for implementing transactions and tenders

531 We include UML definitions for the standard payloads for service requests, rather than the
532 service, communication, or other characteristics. In Section 6 we describe standard serialization
533 for the CTS standard payloads; additional bindings may be used by conforming implementations.

534 3.2 Naming of Services and Operations

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

541 *Create—Created* An object is created and sent to the other Party

542 *Cancel—Canceled* A previously created request is canceled

543 For example, to construct an operation name for the *EiTender* service, "Ei" is concatenated with
544 the name fragment (verb) as listed. An operation to cancel an outstanding Tender is called
545 *EiCancelTender*.¹²

¹² This pattern was developed and is used by current work in the IEC Technical Committee 57 (Power Systems).

546 **3.3 Payloads and Messages**

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

549 **3.4 Description of the Services and Operations**

550 The sections below provide the following for each service:

- 551 • Service description
- 552 • Table of operations
- 553 • Interaction patterns for the service operations in graphic form, using Energy
554 Interoperation normative interactions
- 555 • Normative information model using [UML] for key artifacts used by the service
- 556 • Normative operation payloads using [UML] for each operation

557 **3.5 Responses**

558 In a service interaction, responses may need to be tracked to determine if the transaction is
559 successful or not. This may be complicated by the fact that any given transaction may involve
560 the transmission of one or more information objects.

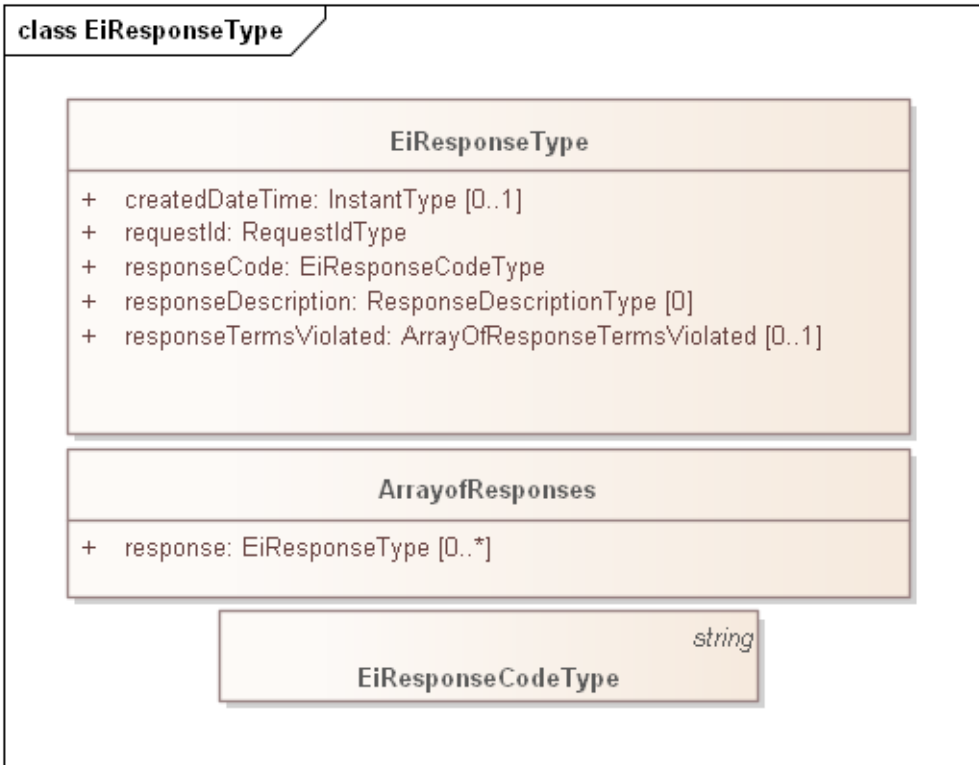
561 An EiResponse returns the success or failure of the entire operation, with possible detail included
562 in responseTermsViolated (not in this release)..

563 It is MANDATORY to return as appropriate both errors and success in responses.¹³

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

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

¹³ 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).



567
568 *Figure 3-1: Example of generic error response for a service operation*

569 The attributes of EiResponseType are in the following table.

570 *Table 3-1: EiResponse Attributes*

Attribute	Meaning	Notes
Created DateTime	Optional timestamp indicating the date and time when this EiResponse was created	
Request ID	A reference ID which identifies the artifact or message element to which this is a response. The Request ID uniquely identifies this request, and can serve as a messaging correlation ID ¹⁴ .	Provided by the payload generation and/or messaging system.
Response Code	The Response Code indicates success or failure of the operation requested. The Response Description is unconstrained text, perhaps for use in a user interface. The code ranges are those used for HTTP response codes, ¹⁵ specifically 1xx: Informational - Request received, continuing process	EML-CTS uses response code 200 for success

¹⁴ As an example of the *Correlation Pattern* for messages

¹⁵ See e.g. https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

	<p>2xx: Success - The action was successfully received, understood, and accepted</p> <p>3xx: Pending - Further action must be taken in order to complete the request</p> <p>4xx: Requester Error - The request contains bad syntax or cannot be fulfilled</p> <p>5xx: Responder Error - The responder failed to fulfill an apparently valid request</p>	
Response Description	The Response Description is in the model but profiled to be cardinality 0..0.	Not present in CTS 1.0 payloads
Response Terms Violated	The Terms Violated by the request to which this is a response. Conforming CTS 1.0 implementations SHALL omit this attribute.	Market Terms and Market Context may be implemented in a future release. Work is in progress to profile and simplify the terms.

571

572 There is no exhaustive list of all possible Response Codes. The Response Codes are intended to
573 enable even the smallest device to interpret Response. This specification uses a pattern consisting
574 of a 3-digit code, with the most significant digit sufficient to interpret success or failure. This
575 pattern is intended to support that smallest device, while still supporting more nuanced messages
576 that may be developed.

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

581 EML-CTS uses response code 200 for success.

582 4 Transactive Services

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

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

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

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

594 4.1 Pre-Transaction Services

595 Pre-transaction services are those between parties that may prepare for a transaction. The pre-
596 transaction services in CTS is EiTender with payloads shown in Table 4-1.

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

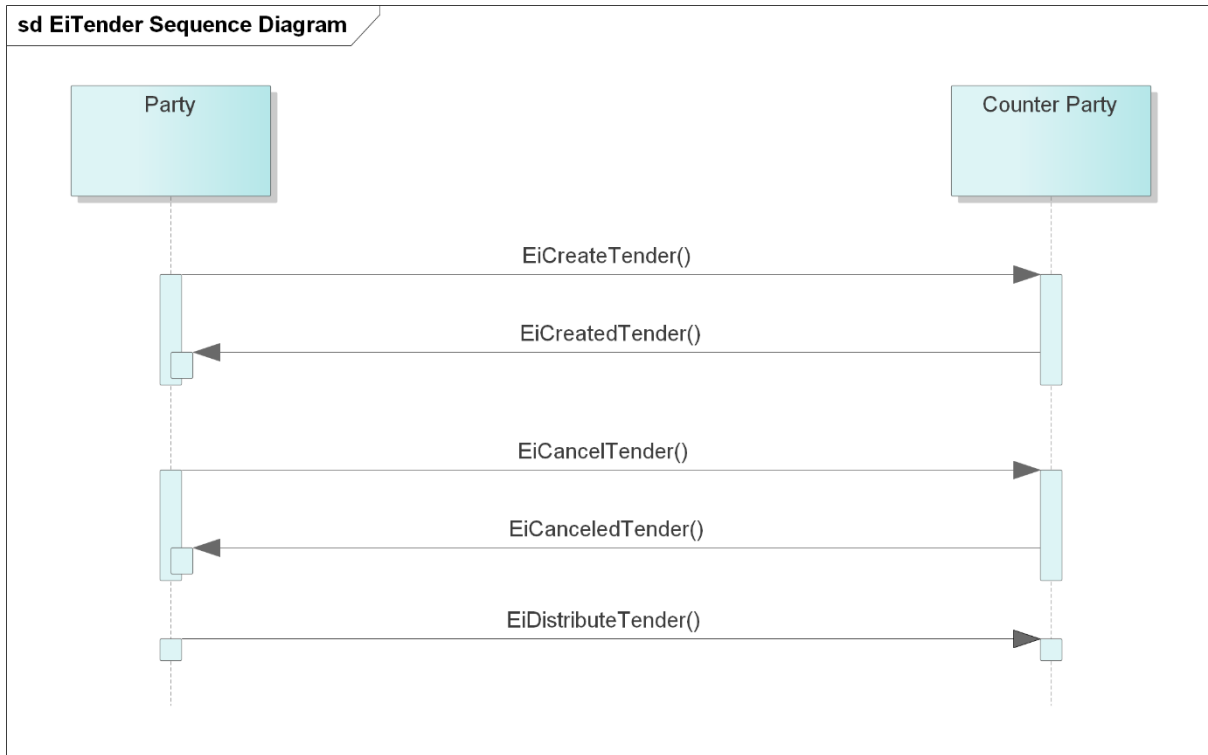
600 *Table 4-1: Pre-Transaction Tender Services*

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

601

602 4.1.1 Interaction Pattern for the EiTender Service

603 Figure 4-1 presents the [UML] sequence diagram for the EiTender Service. Note that
604 EiDistributeTender is not part of CTS 1.0, but is being considered for a future release.

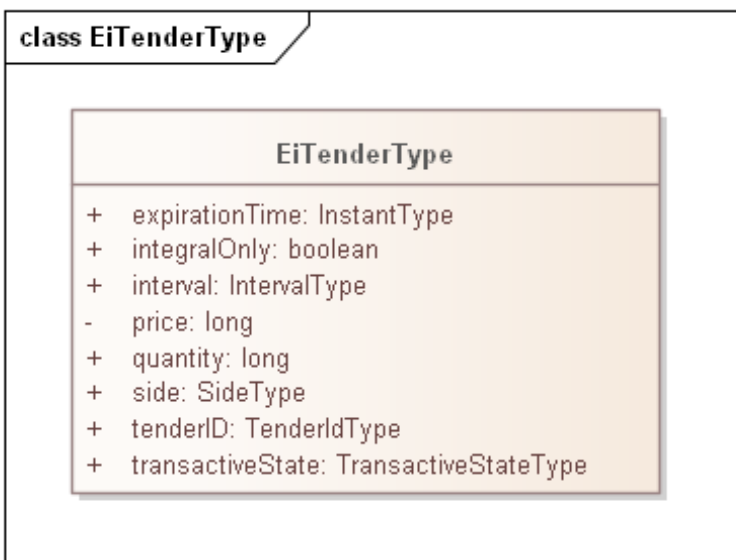


605
606 *Figure 4-1: UML Sequence Diagram for the EiTender Service*

607 **4.1.2 Information Model for the EiTender Services**

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

610 Time interval, price, and quantity are key elements for a product; the other aspects of product
611 definition (e.g. energy and units) are implicit as described in Section 2.4.2.



612
613 *Figure 4-2: Class EiTenderType*

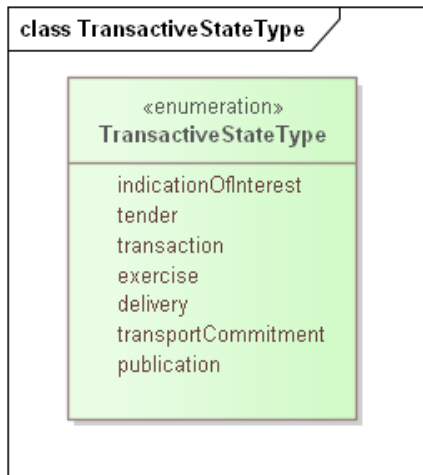
614 The attributes of EiTender are shown in the following table.

615 Table 4-2: EiResponse Attributes

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

616

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

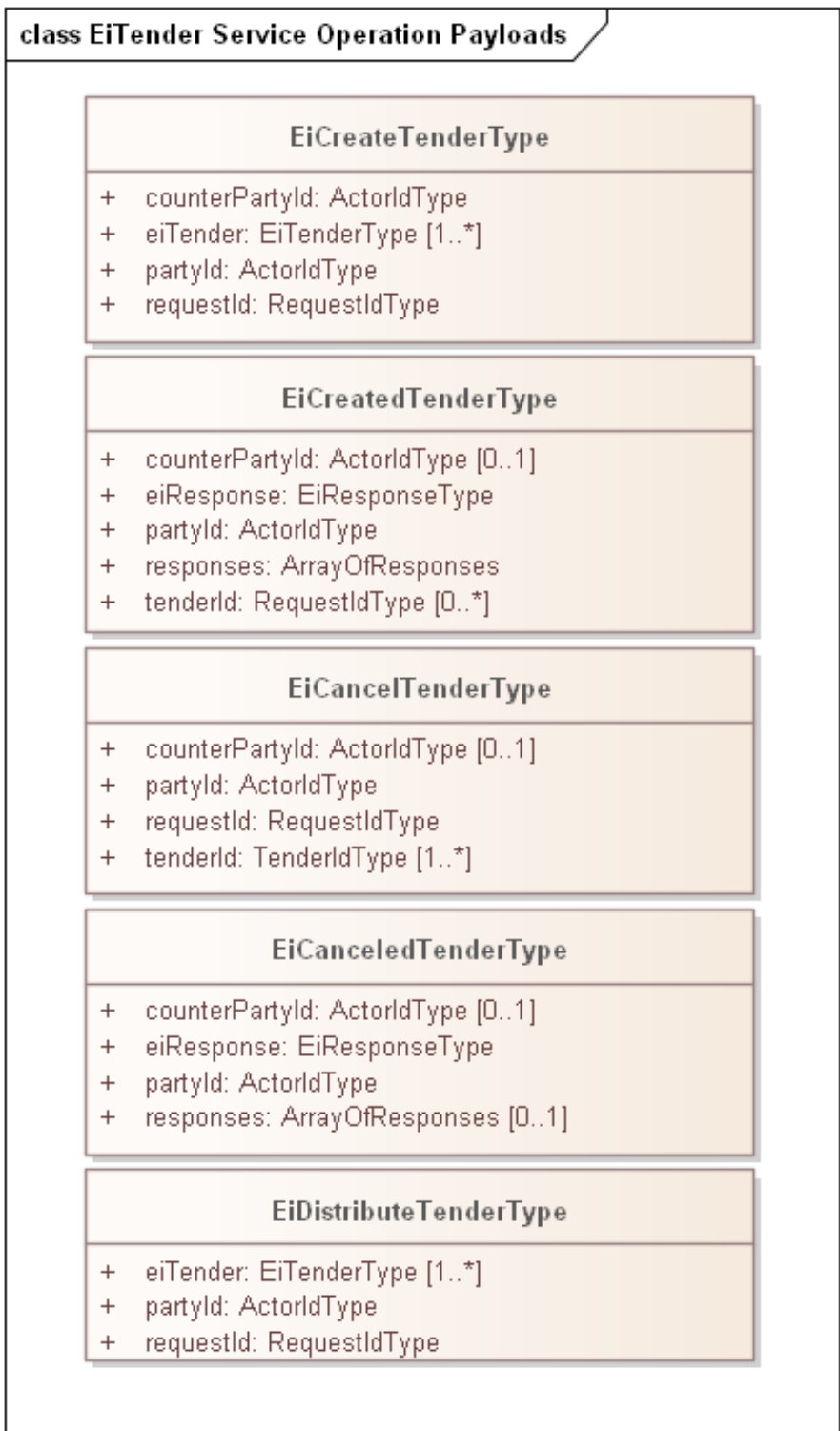


619

620 Figure 3-3 Enumeration TransactiveStateType

621 **4.1.3 Operation Payloads for the EiTender Service**

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



623

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

625 **4.2 Transaction Management Services**

626 This section presents the Transaction Service payloads.

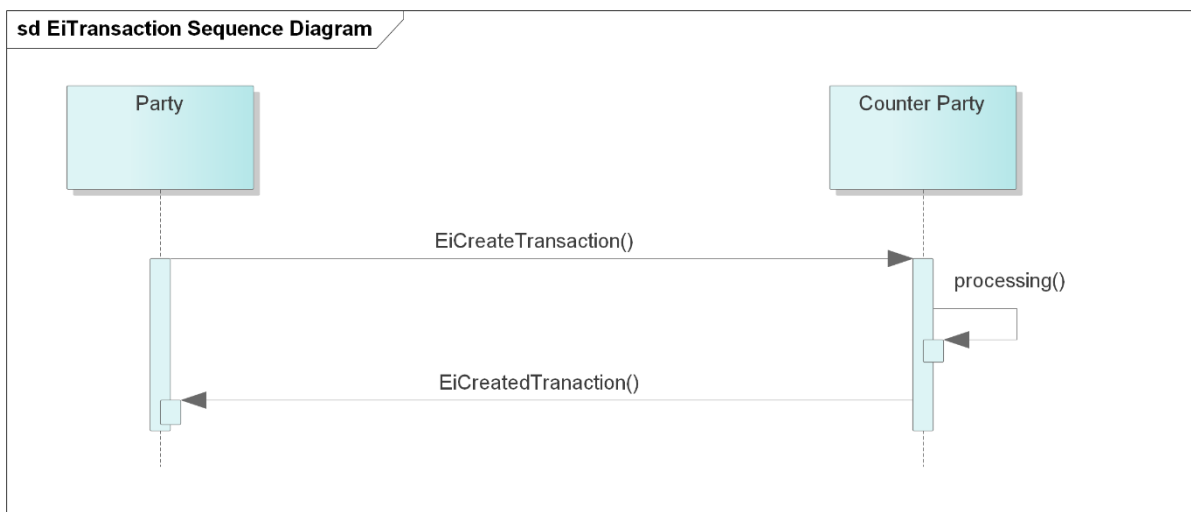
627 market context and product are implied and may in the future be made explicit with a Market
 628 Context reference (see Section 2.4.2). Canceling or modifying transactions is not permitted in
 629 either CTS or Energy Interoperation. Following the approach in distributed agreement
 630 protocols¹⁶, compensating tenders and transactions SHOULD be created as needed to
 631 compensate for any effects.¹⁷

632 *Table 4-3: Transaction Management Service*

Service	Request	Response	Notes
EiTransaction	EiCreateTransactionType	EiCreatedTransactionType	Create and send Transaction

633 **4.2.1 Interaction Pattern for the EiTransaction Service**

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



635
 636 *Figure 4-5: UML Sequence Diagram for the EiTransaction Service*

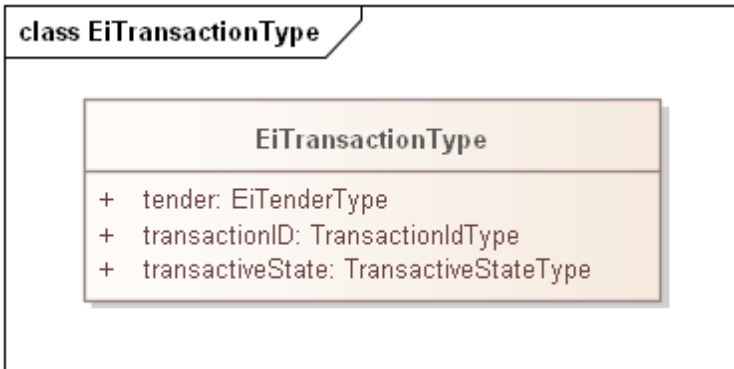
637 **4.2.2 Information Model for the EiTransaction Service**

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

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

¹⁶ See, e.g., WS-Transaction and WS-BusinessActivity.

¹⁷ This is consistent with the way that distributed agreement protocols such as [WS-BusinessActivity] manage compensation rather than cancelation.



642

643 *Figure 4-6: UML Class Diagram of EiTransaction*

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

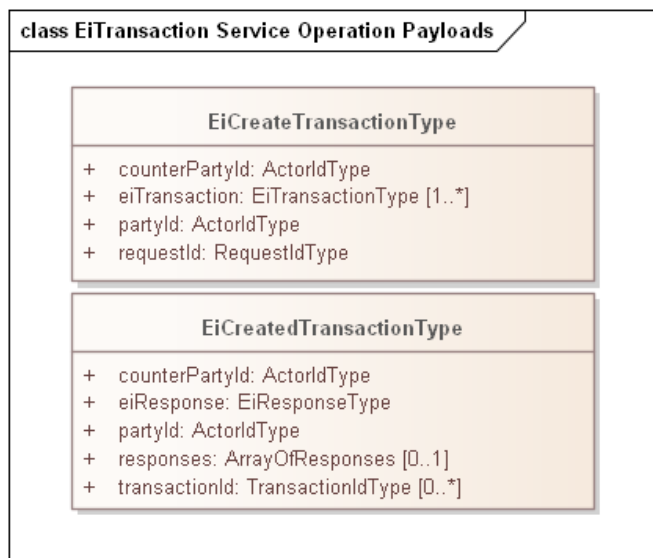
645 *Table 4-4: EiTransaction Attributes*

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

646

647 4.2.3 Operation Payloads for the EiTransaction Service

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



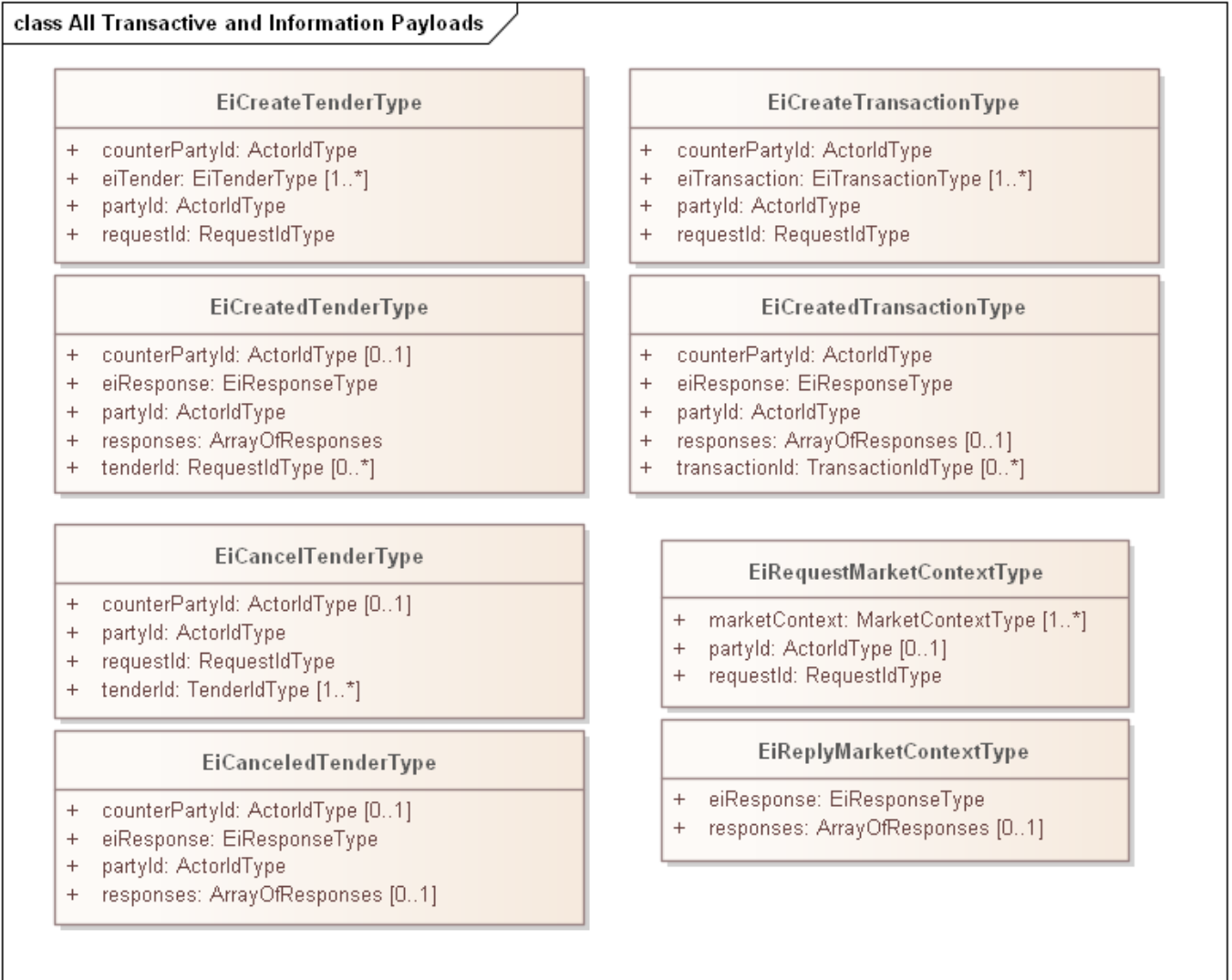
649

650 *Figure 4-7: UML Class Diagram of EiTransaction Service Operation Payloads*

651 **4.3 Comparison of Transactive Payloads**

652

653 *Figure 4-8: UML Diagram comparing all Transactive Payloads*



654

656 5 Market Information

657 Each Event and Service in Energy Interoperation takes place within a Market Context. This
658 Context defines the behaviors that that each Party can expect from the other.

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

660 This is work in progress.

661 5.1 The Market Context

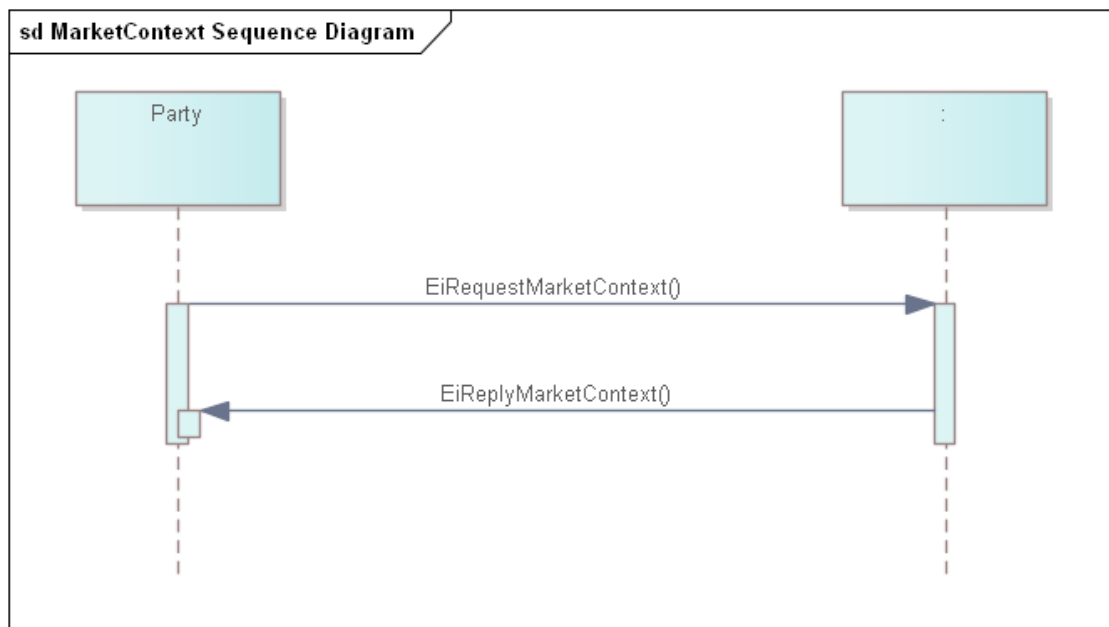
662 Market Contexts are resolvable URIs and are used to express market information that rarely
663 changes, so it is not necessary to communicate it with each message.

664 In any market context, there are standing terms and expectations about product offerings. If these
665 standing terms and expectations are not known, many exchanges may need to occur before
666 finding products that meet those expectations. If these expectations are only known through local
667 knowledge, then national and international products need to be re-configured for each local
668 market that they enter. If all market information were to be transmitted in every information
669 exchange, messages based on EMIX would be overly repetitive.

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

671 5.2 Interaction Pattern for the Market Context Service

672 The Market Context Service enables a Party to request the details of a Market Context. Parties
673 MAY be able to request and compare Market Contexts to select which markets to participate in.
674 Such Interactions are out of scope for this specification.



675
676 *Figure 5-1: UML Sequence diagram for Market Context service*

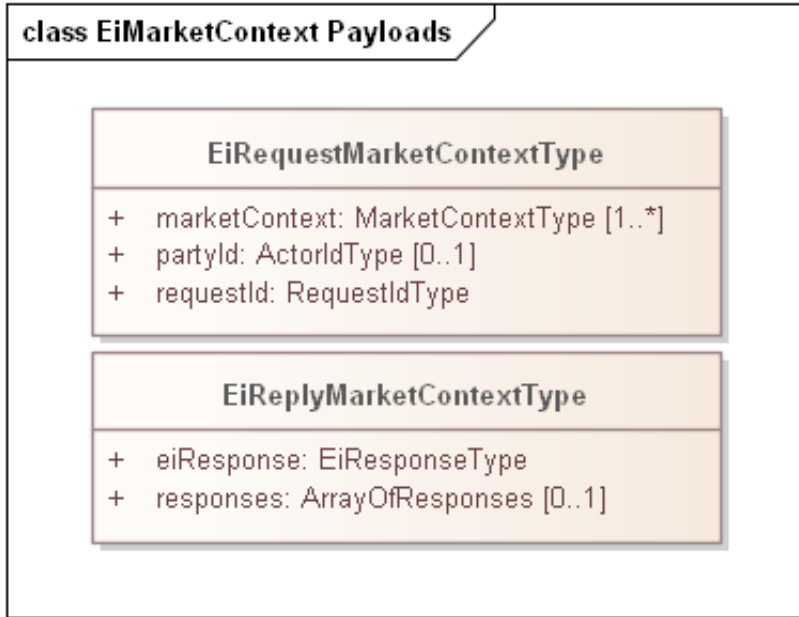
677 The Market Context service can retrieve the full information associated with an
678 EiMarketContext. There is one operation and a responding operation.

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

680 **5.3 Information Model for the EiMarketContext Service**

681 Simplified profile pending.

682 **5.4 Operation Payloads for the EiMarketContext Service**



683

684 *Figure 5-2: UML of Market Context Service payloads*

685 **6 Bindings**

686 Payloads and interaction patterns are described in [UML] in Section 1 above. This section
687 contains bindings for the payloads in three encoding schemes:

- 688 • JSON [JSON]
- 689 • XML Schema [XSD]
- 690 • FIX Simple Binary Encoding [SBE]

691 **6.1 JSON**

692 TODO—JSON Schema available

693 **6.2 XML Schema**

694 TODO—XML Schema available

695 **6.2.1 XML Namespaces**

696 **6.3 Simple Binary Encoding**

697 TODO—Work in progress

698 7 Conformance of Common Transactive Services

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

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

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

- 727 • OASIS WS-Calendar [MIN]
- 728 • OASIS WS-Calendar Schedule Streams and signals [Streams]

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

733 **8 Claiming Conformance to Common Transactive Services**

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

736 Appendix A. Acknowledgments

737 This specification began with William Cox leading the Common Transactive Services team in
738 the 2015-2016 NIST Transactive Energy Challenge to define the initial structure of the CTS
739 [CTS2016].

740 Others picked up and used that work, culminating in a contract from NIST with TC9, Inc and
741 Cox Software Architects LLC to develop agents to support co-simulation of bilateral markets
742 with GridLAB-D™ input models in the NIST Cyberphysical systems modelling platform. That
743 contract required all work to be open source from day one, and all work to be done in the open.
744 TC9 opted to perform the work in the open repositories of The Energy Mashup Lab. NIST has
745 incorporated that code into their TE Market simulation model.

746 The initial draft of CTS 1.0 (this specification) was based on clarifications and simplifications
747 discovered building the internal services and APIs of that project. The Lab has continued to
748 refine that work through and with the NJIT Capstone Projects.

749 All work continues in the open GitHub repositories, and all code is licensed under an Apache 2.0
750 license.

751 The following individuals have participated in the creation, refining, and implementation of this
752 specification and are gratefully acknowledged:

- 753 • NIST, the National Institute of Standards and Technology, including
 - 754 ○ David Holmberg
 - 755 ○ Thomas Roth
- 756 • Members of the WS-Calendar, Energy Market Information Exchange, and Energy
757 Interoperation TCs (see acknowledgement in the respective specifications)
- 758 • Members of the NIST Transactive Energy Challenge Common Transactive Services work
759 group (see acknowledgement in the respective specification and paper)
- 760 • New Jersey Institute of Technology and the NJIT Capstone program where The Energy
761 Mashup Lab worked with faculty, staff, and teams of Seniors and Masters students,
762 specifically
 - 763 ○ Professor Osama Eljabiri
 - 764 ○ Capstone Executive Team members for each term listed below
 - 765 ○ Team Members Fall 2020: Omair Abdul, Omar Janouk, Matthew Molinari
 - 766 ○ Team members Summer 2020: Indira A. Akkiraju, Josiah Nieves, Alex Shepherd
 - 767 ○ Team members Spring 2020: Matt Amato, Dhruvinkumar Desai, Anupam Saini,
768 Justin Schuster
 - 769 ○ Team members Fall 2019: Rajeev Chanchlan, Jasper Sam David, Mounica Gona,
770 Dhrumil Shah, Karan Shah
- 771 • The Energy Mashup Lab, its officers and associates
 - 772 ○ Toby Considine
 - 773 ○ William Cox
 - 774 ○ David A Cohen

775 **Appendix B. Background and Development history**

776 The Common Transactive Services (CTS) are a lightweight profile of the OASIS Energy
777 Interoperation Standard [EnergyInterop].

778 The Energy Interoperation Technical Committee was formed to define the necessary interactions
779 between Smart Grids and their end nodes, including Smart Buildings, Enterprises, Industry,
780 Homes, and Vehicles. The specification defines data and communication models that enable
781 standard exchange of signals for dynamic pricing, reliability, and emergencies. Energy
782 Interoperation supports market-based balancing of energy supply and demand while increasing
783 fluidity of contracts.

784 In 2015, the US National Institute for Standards and Technology (NIST) began the Transactive
785 Energy Modeling and Simulation Challenge (TE Challenge). A report delivered to the TE
786 Challenge and a paper delivered to the Transactive Energy Systems Conference [**TESC2016**]
787 defined a minimal subset of Energy Interoperation, which became known as the common
788 transactive services. The report further showed commonality between the messages of existing
789 TE systems, including several not based on Energy Interoperation.

790 The Energy Mashup Lab has created an open source implementation using the Common
791 Transactive Services called EML-CTS¹⁸, which has in turn helped us to further simplify the
792 original description of CTS and led to this evolved specification.

793 The EML-CTS v1.0 system uses CTS message payloads expressed in JSON for all market
794 communications. The Lab plans to contribute this specification to the OASIS Energy
795 Interoperation Technical Committee as the basis for work on a standard lightweight specification
796 for The Common Transactive Services.

¹⁸ <https://github.com/EnergyMashupLab/eml-cts>

797 Appendix C. Glossary

- 798 No definition in this glossary supplants normative definitions in this or referenced specifications.
799 They are here merely to provide a guidepost for readers at to terms and their special uses.
800 Implementers will want to be familiar with all referenced standards.
- 801 Actor is an architectural component that interacts with other actors. Actors may take on roles,
802 e.g. as a Party in a transaction.
- 803 Agreement is broad context that incorporates market context. Agreement definitions are out of
804 scope in the Common Transactive Services.
- 805 EMIX: As used in this document, EMIX objects are descriptions applied to a WS-Calendar
806 Sequence. EMIX defines Resource capabilities, used in tenders to match capabilities to
807 need, and in Products, used in tenders and in specific performance and execution calls.
808 Please note that CTS uses more recent WS-Calendar specifications than that used in
809 EMIX, and that the product definition in CTS 1.0 is implicit.
- 810 Party or Transactive Party is a role that an actor may take. In the EML-CTS implementation, the
811 Local Market Agent (LMA) is not a party, but the Transactive Energy User Agent
812 (TEUA) is a party and represents its Energy Manager.
- 813 Resource (as defined in EMIX¹⁹): A Resource is something that can describe its capabilities in a
814 Tender into a market. How those Capabilities vary over time is defined by application of
815 the Capability Description to a WS-Calendar Sequence. See [EMIX].
- 816 Stream: A set of contiguous intervals of the same size. See [Streams]
- 817 Tender: A tender is an offering for a Transaction. See Transaction.
- 818 Transaction: A binding commitment between parties entered into under an agreement.

¹⁹ See http://docs.oasis-open.org/emix/emix/v1.0/cs02/emix-v1.0-cs02.html#_Toc319594576

819 **Appendix D. Revision History**

820

Revision	Date	Editor	Changes Made
CTS 1.0 Draft of 2020-10-28	2020-10-28	William Cox	First published document. Evolved from OASIS Energy Interoperation Standard, CTS reports and papers, and the EML-CTS project
CTS 1.0 Contribution Version	2020-11-30	William Cox	Updated text and references for planned contribution to OASIS Energy Interoperation Technical Committee

821