



Energy Interoperation Common Transactive Services (CTS) Version 1.0

Working Draft ~~2229~~

~~25 March~~09 September 2024

This stage:

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd03/ei-cts-v1.0-csd03.pdf> (Authoritative)

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd03/ei-cts-v1.0-csd03.html>

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd03/ei-cts-v1.0-csd03.docx>

Previous stage:

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd02/ei-cts-v1.0-csd02.pdf> (Authoritative)

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd02/ei-cts-v1.0-csd02.html>

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd02/ei-cts-v1.0-csd02.docx>

Latest stage:

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.pdf> (Authoritative)

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.html>

<https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.docx>

Technical Committee:

OASIS Energy Interoperation TC

Chairs:

William T. Cox ([wtcox@coxsoftwarearchitects.com](mailto:wtcx@coxsoftwarearchitects.com)), Individual

David Holmberg (david.holmberg@nist.gov), NIST

Editor:

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

Related work:

This document replaces or supersedes:

- Common Transactive Services 1.0. The Energy Mashup Lab Specification. Edited by William T. Cox, Toby Considine 30 November 2020. <https://www.theenergymashuplab.org/s/cts-1-0-draft-20201130.pdf>.

This document is related to:

- *Energy Interoperation Version 1.0*. Edited by Toby Considine, 11 June 2014. OASIS Standard. <http://docs.oasis-open.org/energyinterop/ei/v1.0/os/energyinterop-v1.0-os.html>. Latest version: <http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html>. and its TeMIX Profile
- *Energy Market Information Exchange (EMIX) Version 1.0*. Edited by Toby Considine. Latest version: <http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html>.
- *WS-Calendar Platform Independent Model (PIM) Version 1.0*. Edited by William Cox and Toby Considine. Latest version: <http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html>.

- *Schedule Signals and Streams Version 1.0*. Edited by Toby Considine and William T. Cox. Latest version: <http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html>.

Abstract:

Common Transactive Services (CTS) permits energy consumers and producers to interact through energy markets by simplifying actor interaction with any market. CTS is a streamlined and simplified profile of the OASIS Energy Interoperation (EI) specification, which describes an information and communication model to coordinate the exchange of energy between any two Parties that consume or supply energy, such as energy suppliers and customers, markets and service providers.

Status

This document was last revised or approved by the OASIS Energy Interoperation TC on the above date. The level of approval is also listed above. Check the "Latest stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.groups.oasis-open.org/committeescommunities/tc_home.php?wg_abbrev=energyinterop-community-home2?CommunityKey=d3a22f3f-887d-465e-bb0f-018dc7d3f405#technical.

~~Any individual may submit comments to the TC at Technical-Committee-Comments@oasis-open.org. 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 "" button on the TC's web page at-~~

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, use the following citation format:

[Energyinterop-CTS-v1.0]

Energy Interoperation Common Transactive Services (CTS) Version 1.0. Edited by Toby Considine. 28 April 2022March 2024. OASIS Committee Specification Draft 0203. <https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/csd03/ei-cts-v1.0-csd03.html>. Latest stage: <https://docs.oasis-open.org/energyinterop/ei-cts/v1.0/ei-cts-v1.0.html>.

Notices:

Copyright © OASIS Open 2022-20232024. All Rights Reserved.

Distributed under the terms of the OASIS IPR Policy, [<https://www.oasis-open.org/policies-guidelines/ipr/>]. For complete copyright information please see the Notices section in the Appendix.

Table of Contents

1	Introduction.....	11
1.1	Application of the Common Transactive Services	12
1.2	Support for Developers.....	12
1.3	Naming Conventions	13
1.4	Editing Conventions.....	13
1.5	FIX and the Language of Trading	13
1.6	Use of terms Actors and Facets in this specification	14
1.7	Security and Privacy	15
1.7.1	Security Considerations	15
1.7.2	Privacy Considerations.....	15
1.8	Semantic Composition.....	15
1.9	Applicability to Microgrids (Informative)	16
1.10	Specific scope statements	16
1.11	Naming of Messages and Operations	17
2	Overview of Common Transactive Services	18
2.1	Parties.....	18
2.2	Trading semantics from FIX Protocol	18
2.2.1	Parties and Orders	19
2.2.2	Instruments.....	19
2.2.3	Market Crossing	19
2.2.4	Markets and Market Segments	20
2.3	Common Transactive Services Roles	20
2.3.1	Parties as Market Participants.....	20
2.3.2	Party and Counterparty and Transactions	20
2.3.3	Facets in the CTS Specification	21
2.4	Responses	22
2.5	Identities.....	24
3	Market Semantics: Resource, Product, Instrument, and Streams	26
3.1	Resource, Product, & Instrument	26
3.1.1	Defining Resource	28
3.1.2	Defining Product.....	29
3.1.3	Defining Instrument	31
3.1.4	Summary of Instrument Specification	32
3.2	CTS Streams: Expressing Time Series	33
3.3	The Bounding Interval Pattern in CTS.....	35
4	Party Registration Facet.....	37
5	The Tender Facet (Order messages).....	38
5.1	Messages for the Tender Facet.....	38
5.1.1	Illustrative Narrative on Tenders [Non-Normative]	38
5.2	Interaction Patterns for the Tender Facet.....	39
5.3	Information Model for the Tender Facet	41
5.3.1	Interval Tenders and Stream Tenders	49
5.3.2	Execution Instructions	50

5.3.3 Use of Warrants in Tenders	51
5.4 Contingent Tenders	52
5.4.1 Illustrative Narrative on Contingent Tenders [Non-Normative]	52
5.5 Rejecting a Tender	52
5.6 Message Payloads for the Tender Facet.....	52
6 The Transaction Facet (Execution)	59
6.1 Messages for the Transaction Facet	59
6.2 Interaction Pattern for the Transaction Facet	59
6.3 Information Model for the Transaction Facet.....	61
6.4 Payloads for the Transaction Facet.....	64
6.5 Comparison of Tender and Transaction Payloads	67
6.6 Off-Market Transactions	68
7 The Position Facet.....	70
7.1 Introduction	70
7.2 Information Model for the Position Facet.....	71
7.3 Payloads for the Position Facet.....	71
8 The Delivery Facet	75
8.1 Interaction Pattern for the Delivery Facet.....	75
8.2 Information Model for the Delivery Facet.....	76
8.3 Payloads for the Delivery Facet.....	76
9 The Negotiation Facet	79
9.1 Negotiation Vocabulary.....	80
9.2 Narrative on Negotiation (non-normative)	81
9.3 Messages for the Negotiation Facet.....	82
9.4 Interaction Pattern for the Negotiation Facet.....	84
9.4.1 Interaction Patterns for RFQ and Quote	85
9.4.2 Creating Transactions from Quotes	87
9.4.3 Interaction Pattern for Market-Mediated Negotiation	87
9.4.4 Interaction Patterns Restricted by Market Mechanism	89
9.4.4.1 Quote-Driven Markets.....	90
9.4.4.2 Request for Quotations Market.....	91
9.5 Information Model for the Negotiation Facet	91
9.5.1 A Note on Stream Quotes	91
9.5.2 The Request for Quotation	92
9.5.3 Quotes	94
9.6 Messages for the Negotiation Facet.....	96
9.6.1 RFQ Messages.....	96
9.6.2 Quote Messages	101
9.6.2.1 Cancelling a Quote	107
9.6.2.2 Accepting a Quote	108
9.6.2.3 Rejecting a Quote	108
10 Subscription Facet.....	111
10.1 Messages for the Subscription Facet	111
10.2 Interaction Pattern for the Subscription Facet	112
10.3 Information Model for Subscription Requests.....	112
11 Tickers	116

11.1	Messages for Tickers.....	118
11.2	Interaction Pattern for Tickers	119
11.3	Exceptions to Ticker Subscription Interactions.....	120
11.4	Interaction Patterns for Ticker Data.....	120
11.5	Information Model for Ticker Payloads	121
11.5.1	Tender Tickers	123
11.5.2	Quote Tickers	123
11.5.3	RFQ Tickers	123
11.5.4	Transaction Tickers	123
11.6	Message Payloads for Managing Ticker Subscriptions.....	124
12	Instrument Data Subscriptions	126
12.1	Messages for Instrument Reference Data Subscriptions	126
12.2	Interaction Pattern for Instrument Reference Data Subscriptions	128
12.3	Information Model for Manage Instrument Reference Data Subscription Payloads	129
12.4	The Instrument Session Reports	130
12.4.1	Information Model for the Instrument Session Report Type	130
12.4.2	The Instrument Summary Types	133
12.4.2.1	The Instrument Session Summary.....	133
12.4.2.2	The Instrument Book Summary	134
13	Market Structure Reference Data: Market, Segment, and Session Subscriptions	136
13.1	Market Mechanisms.....	136
13.2	Market Reference Data.....	140
13.2.1	Messages for Market Structure Reference Data.....	140
13.2.2	Interaction Pattern for Market Reference Data	140
13.2.3	Information Model for Market Reference Data	141
13.2.4	Payloads for Market Reference Data	143
13.3	Segment Reference Data	144
13.3.1	Messages for Segment Reference Data	145
13.3.2	Interaction Pattern for Segment Reference Data.....	148
13.3.3	Information Model for Segment Reference Data.....	149
13.3.4	Payloads for Segment Reference Data.....	155
13.4	Trading Session Data	156
13.4.1	Messages for Trading Session Data	156
13.4.2	Interaction Pattern for Trading Session Data.....	156
13.4.3	Information Model for Trading Session Data.....	156
13.4.4	Payloads for Trading Session Data.....	158
14	Conformance	161
14.1	Introduction to Conformance	161
14.2	Claiming Conformance to Common Transactive Services.....	161
14.3	Annex: Conformance statements from Spec not yet incorporated into this section.....	161
14.3.1	Conforming with Use of Warrants in Tenders	161
Appendix A.	References	162
A.1	Normative References.....	162
A.2	Informative References	163
Appendix B.	Choosing a Market Mechanism.....	165

B.1 Central Limit Order Book (LB): Simple Bids & Offers.....	165
B.2 Periodic Auctions (PA)	165
B.3 Quote-Driven Markets (QB).....	165
B.4 Request for Quote Markets (RQ)—Negotiating	166
B.5 Market mechanisms not defined in FIX MMT.....	167
B.5.1 Off-Book segment (OB)	167
B.5.2 Real Time Pricing (RT)	168
B.5.3 Spot Market (SP)	168
B.5.4 Self Executing (SX).....	168
Appendix C. Security and Privacy Considerations	169
C.1 CTS and Security Considerations	169
C.2 CTS and Privacy Considerations	169
Appendix D. Semantic Composition from Energy Interoperation, EMIX, and WS-Calendar.....	172
D.1 Conformance with Energy Interoperation	172
D.2 Conformance with EMIX	172
D.3 Conformance with WS-Calendar Streams	173
D.4 CTS and WS-Calendar Streams	173
Appendix E. Glossary of Terms and Abbreviations Used in this document	176
Appendix F. Acknowledgments.....	177
F.1 Participants	177
F.2 Special Thanks	177
Appendix G. Revision History	178
Notices	181

Table of Tables

Table 2-1: Facets Defined in CTS	21
Table 2-2: Attributes of EiResponse.....	23
Table 3-1: Defining the Resource	28
Table 3-2: Defining the Product	30
Table 3-3: Specifying the Instrument	31
Table 3-4: Specifying the Stream	34
Table 5-1: Tender Facet Payloads.....	38
Table 5-2: EiTender Attributes	44
Table 5-3 Tender Base Attributes	45
Table 5-4: Trading Instructions	51
Table 5-5 EiCreateTenderPayload Attributes	56
Table 5-6 EiCreatedTenderPayload Attributes	56
Table 5-7 EiCancelTender Payload Attributes	57
Table 5-8 EiCanceledTenderPayload Attributes	58
Table 6-1: Transaction Facet	59
Table 6-2: EiTransaction Attributes.....	64
Table 6-3 EiCreateTransactionPayload Attributes	66
Table 6-4 EiCreatedTransactionPayload Attributes	66
Table 7-1: Position Facet	70
Table 7-2: Attributes of Position Facet Payloads	72
Table 8-1: Delivery Facet	75
Table 8-2: Attributes of Delivery Facet Payloads.....	77
Table 9-1: Negotiation Terminology	80
Table 9-2 Messages for the Negotiation Facet	84
Table 9-3 Attributes of EiRfqType	93
Table 9-4 Attributes of EiQuoteType	95
Table 9-5: EiCreateRFQ Payload Attributes	98
Table 9-6 EiCancelRfq and EiCanceledRfq Payload Attributes	100
Table 9-7 EiCreateQuotePayload	103
Table 9-8 EiCreatedQuotePayload	105
Table 9-9 EiCancelQuote Payload Attributes.....	107

Table 9-10 EiCanceledQuote Payload Attributes	107
Table 9-11 EiReject and EiRejected Quote Payload Attributes	109
Table 10-1 Messages for the Subscription Facet	112
Table 10-2 EiSubscriptionRequest Attributes	114
Table 10-3 EiSubscriptionResponse Attributes	115
Table 11-1: Types of Tickers in CTS Facet.....	117
Table 11-2 Ticker Facet Messages	118
Table 11-3 Attributes for the Ticker Payload Base and Ticker Types	122
Table 11-4 Attributes for the EiManage and EiManagedTickerSubscription Payloadfs	124
Table 12-1 Messages for Instrument Reference Data.....	127
Table 12-2: Attributes for Manage Instrument Data Payload.....	129
Table 12-3: Attributes for the Instrument Session Report Type.....	131
Table 12-4: Instrument Session Summary Type attributes.....	134
Table 12-5: Instrument Book Summary Attributes	134
Table 12-6 Book Entry Attributes.....	135
Table 13-1 Market Mechanism Types in CTS.....	138
Table 13-2 Messages for Market Reference Data.....	140
Table 13-3 Attributes for Market Reference Data	142
Table 13-4 Messages for Segment Reference Data	145
Table 13-5 Segment Reference Data	151
Table 13-6 Messages for Trading Session Data	156
Table 13-7 Session Data	157
Table C--14-1 Abbreviations and Terms used throughout this document.....	176

Table of Figures

Figure 2-1 UML Class Diagram of EiResponseType and MarketAttributeViolationType.....	23
Figure 2-2 UML Class Diagram of ID Types in CTS	25
Figure 3-1 Informal sketch showing relationship between Resource, Product, and Instrument ..	27
Figure 3-2 UML Class Diagram for Resource, Product, and Instrument	28
Figure 3-3 UML Class Diagram for Product showing Inheritance from Resource.....	30
Figure 3-4 UML Class Diagram for Instrument showing Inheritance from Resource & Product	33
Figure 3-5: UML Class Model for CtsStream and the Stream Intervals	34
Figure 5-1: UML Sequence Diagram for the Tender Facet.....	40
Figure 5-2 UML Class Diagram Showing Commonality between Tender, Quote, and RFQ	42
Figure 5-3 UML Class Diagram showing EiTenderType.....	43
Figure 5-4 UML Class Diagram for Tender Facet Payloads	54
Figure 6-1: UML Sequence Diagram for the EiTransaction Facet.....	60
Figure 6-2: UML Class Diagram of EiTransactionType	63
Figure 6-3: UML Class Diagram of EiTransaction Facet Payloads	65
Figure 6-4: UML Diagram comparing Tender and Transaction Facet Payloads.....	68
Figure 7-1: UML Sequence Diagram for the Position Facet	71
Figure 7-2: UML Class Diagram of Payloads for the Position Facet.....	72
Figure 8-1: UML Sequence Diagram for the Delivery Facet	76
Figure 8-2: UML Class Diagram of Payloads for the Delivery Facet	77
Figure 9-1 UML Sequence Diagram for Negotiation Facet RFQ (Request for Quote).....	85
Figure 9-2 UML Sequence Diagram for Negotiation Facet Quote.....	87
Figure 9-3 Market Mediated Quote and Responses Sequence Diagram	89
Figure 9-4 Quote-Driven Market (MMT_QUOTE_DRIVEN) Responses to EiCreateQuote—the only possible response is EiAcceptQuote.	90
Figure 9-5 Request for Quotations Market (RQ) MMT_RFQ Responses to EiCreateQuote—responses to EiAccept, EiReject, EiCreateQuote and EiCreateRfq are required but not shown..	91
Figure 9-6 UML Class Diagram for EiRfqType.....	93
Figure 9-7 UML Class Diagram of EiQuoteType showing inherited attributes.....	95
Figure 9-8 UML Class Diagram Showing Negotiation Facet RFQ Payloads	96
Figure 9-9 Negotiation Facet Quote Payloads	102
Figure 9-10 Negotiation Facet Quote Payloads	108
Figure 9-11 EiReject and EiRejectedQuote Payloads	109

Figure 10-1 UML Class Diagram for Subscription Request and Response Types.....	114
Figure 11-1 TickerType Enumeration	117
Figure 11-2: UML Sequence Diagram for the Ticker Facet.....	119
Figure 11-3 Ticker Payloads and Ticker Type showing inheritance	121
Figure 11-4 Ticker Manage Subscription Payloads showing inherited attributes	124
Figure 12-1 Manage Instrument Reference Data Subscription	128
Figure 12-2 UML Class Diagram for Manage Instrument Reference Data Messages	129
Figure 12-3 UML Class Diagram for Instrument Session Report Type.....	131
Figure 12-4 Instrument Summary Type UML Class Diagram	133
Figure 13-1 Market Mechanism Type Enumeration.....	138
Figure 13-2: UML Sequence Diagram for the Market Reference Data.....	141
Figure 13-3: UML Class Diagram for Market Reference Data	141
Figure 13-4 UML Class Diagram for Market Reference Data Subscription Payloads.....	144
Figure 13-5 UML Sequence Diagram for Segment Reference Data	149
Figure 13-6 Segment Reference Data	150
Figure 13-7 UML Class Diagram of Payloads for Segment Reference Data Subscriptions	155
Figure 13-8: UML Sequence Diagram for Manage Trading Session Data	156
Figure 13-9 UML Class Diagram for Session Data.....	157
Figure 13-10 UML Class Diagram of the Payloads for Manage Session Data	159
Figure C-14-1: CtsStreamDefinition.....	174

1 Introduction

~~The Common Transactive Services (CTS) is an application profile of the OASIS Energy Interoperation 1.0 (EI) specification, with most optionality and complexity stripped away. CTS defines the messages for transactive energy, leaving communication details unspecified.~~ Transactive energy names the collaboration techniques to balance energy supply and energy demand at every moment even as power generation becomes decentralized and as the ownership of energy assets becomes more diverse. The OASIS Energy Interoperation 1.0 (EI) specification defined the interactions and communication required for transactive energy.

The Common Transactive Services (CTS) is an application profile of [EI] with most optionality and complexity stripped away. CTS is strongly influenced by both the TEMIX profile of [EI] and by the philosophy behind TEMIX. CTS defines the messages for transactive energy, leaving communication details unspecified. CTS extends the TEMIX approach using lessons learned in the world's largest financial markets. CTS is both a simplification and extension of [EI] and not part of EI.

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

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

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

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

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

A goal of CTS is to enable systems and devices developed today or in the future to address the challenges of increasing distributed energy resources. CTS enables distributed actors to participate in markets deployed today or in the future.

CTS defines interactions between actors in energy markets. We do not identify whether an actor is a single integrated system, or a distributed collection of systems and devices working together. See Section 1.6 for a discussion of the term Actor in this specification. Autonomous market actors must be able to recognize patterns and make choices to best support their own needs.

CTS assumes the perspective of a trader, that is of a market participant. [EI] was developed with significant input from Economists and energy market regulators, and it relies on language from economics and regulation. The Committee deliberately chose to seek guidance from financial traders and to use their language. Many data elements and message types have been renamed to align with FIX-based markets.

CTS messages are simple and strongly -typed and make no assumptions about the systems or technologies behind the actors. Rather, CTS defines a technology-agnostic minimal set of messages to enable interoperation through markets of participants irrespective of internal technology. In a similar

manner, CTS does not specify the internal organization of a market, but rather a common set of messages that can be used to communicate with any transactive energy market.

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

In a Service-Oriented Architecture [SOA] environment, the semantic payloads are those sent and returned by the *services* described. CTS enables any SOA or other framework to exchange equivalent semantic information without presuming the specific messaging system(s) or architecture used, thus allowing straightforward semantic interoperation.¹ See Section 2.3. [Th](#)

1.1 Application of the Common Transactive Services

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

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

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

TE demonstrations and deployments have seldom been interoperable—each uses its own message model and its own market dynamics. ~~Such markets discount local decision making while introducing new barriers to resilience such as network failure. Others rely on a single price-setting supplier.~~ Systems built to participate in these demonstrations and deployments ~~have been unable~~ [are not able](#) to interoperate with other implementations. The intent of this specification is to enable systems and markets developed for future deployments to interoperate even as the software ~~continues~~ [and markets continue](#) to evolve.

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

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

1.2 Support for Developers

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

¹ SOA is occasionally mis-described as a *client-server* approach. In distinction, services are requested by an Actor, and fulfilled by another Actor. In SOA the services offered are key, and the actors take different roles in different interactions.

~~(1) This specification provides [JSON] schemas compatible with JSON Abstract Data Notation [JADN] format.~~

~~(2) This specification provides [SBE] schemas. The Committee plans to release artifacts defining the commonly used XML and JSON schemas.~~

The FIX Simple Binary Encoding ~~specification~~(SBE) is used in financial markets and for general high-performance messaging—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. [The TC plans to release a SBE schema as well.](#)

[All Schemas will be in a separate release after this specification is complete.](#)

1.3 Naming Conventions

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

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

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

For the names of types within XSD files, the names follow the UpperCamelCase convention with all names starting with an upper-case letter suffixed by “type-”. For example,

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

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

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

JSON and where possible SBE names follow the same conventions.

1.4 Editing Conventions

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

All elements in the tables not marked as “optional” are mandatory. This is the opposite of the convention used in the specification of FIX [Protocol](#).

Information in the **FIX Field** column is non-normative, and includes in parentheses zero or more FIX Tags that are related to the field. This provides guidance for those integrating CTS markets to interoperate with markets supporting the FIX Protocol.

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

Examples and Appendices are non-normative. In particular, architectural and functional examples are presented only to support narrative description. The specific processes, structures, and algorithms are out of scope.

1.5 FIX and the Language of Trading

As noted above, this specification strives to apply the language of financial trading to resource markets. FIX is the language of trading.

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

We thank members of the FIX Trading Community (<https://www.fixtrading.org/>) for their extensive input and close reading. FIX was formed in 1991 to connect the global ecosystem of venues, asset managers, banks/brokers, vendors and regulators by standardizing the communication among participants. FIX relies on 4 key principles:

- Creating and maintaining robust open standards across the across the trade life-cycle with its pre-trade, trade, and post-trade environments.
- Providing advice and counsel to regulatory bodies in a transparent and unbiased way.
- Seeking ways to improve the trading process front to back for the global financial services industry.
- Providing FIX members with a neutral, collaborative environment to come together through member-driven [committees, working groups and](#) conferences ~~and other critical forums~~ to promote, support and educate.

This specification relied strongly on their assistance.

1.6 Use of terms Actors and Facets in this specification

This specification defines message content and interaction patterns.

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

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

At the time of this specification, typical architectures decompose applications into smaller, independent building blocks that are easier to develop, deploy and maintain. A single market participant in energy may be embodied as several of these independent blocks (actors).

For the Internet of Things (IoT), the term Actor begins and ends at the interfaces to things. The “actor model” makes no assumptions of the mechanisms or even motives internal to an Actor. An Actor is simply a thing that acts. The Actor ~~implementation~~ may be [byinstantiated by software in](#) a traditional computer, a cloud node, [by](#) a human behind a user interface, or [anyby a](#) device on the Internet of things.

In transactive energy, the actor model supports the diversity of IoT and of markets. An energy seller may be a generator or a solar panel or a virtual power plant or a demand responsive facility or a financial entity. An energy buyer or seller may be a home or commercial facility or an embedded device or a microgrid or an energy district. A Market acts to match Tenders. An Actor may take a market-maker role, buying and/or selling power for itself. An energy storage system may act as a buyer or as a seller at any time.

We use the term “Facet” to name a ~~coherent~~[cohesive](#) set of messages that an Actor may use to communicate with other Actors. An Actor submits tenders to buy or to sell. An Actor may operate a Market. If the Architecture includes a telemetry Actor, measuring Resource flow (metering), then that Actor MAY represent the Market or the market participant or even a third party. This specification makes no requirement as to how to distribute or make use of these facets.

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

1.7 Security and Privacy

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

1.7.1 Security Considerations

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

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

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

1.7.2 Privacy Considerations

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

The essence of any transaction is the agreement of a Party to sell, and of another [a](#)Party to buy. The identity of the buyer and the identity of the seller are each part of the transaction. Some transaction notifications may hide the identity of the buyer from the seller. Some transaction notifications may hide the identity of the seller from the buyer. Some transactions, such as [those arising from what the energy world calls](#) a double auction³, may be between the market participants as a whole, and not with any particular counterparty. Where ~~this is~~ required, the Market itself may be designated as the counterparty in a notification.

Both security and privacy considerations are addressed in Appendix C.

1.8 Semantic Composition

The semantics and interactions of CTS are selected from and derived from OASIS Energy Interoperation [EI]. EI references two other standards, [EMIX] and [WS-Calendar], and uses an earlier Streams definition. We adapt, update, and simplify the use of the referenced standards, while maintaining conformance.

Appendix D ~~See~~, Semantic Composition from Energy Interoperation, EMIX, and WS-Calendar.

- ~~EMIX~~ describes price and product for electricity markets. WS-Calendar communicates schedules and sequences of operations. ~~CTS uses the [Streams] optimization, which is a standalone specification, rather than part of EI 1.0.~~

³ In a double auction, there are tenders to buy and tenders to sell, and all participants clear at the same price. FIX simply uses the term "Auction".

- EI uses the vocabulary and information models defined by those specifications to describe the services that it provides. The payload for each EI service references a product defined using [EMIX]. EMIX schedules and sequences are defined using [WS-Calendar]. Any additional schedule-related information required by [EI] is expressed using [WS-Calendar].
- Since OASIS published [EI], a semantically equivalent but simpler [Streams] specification was developed in the OASIS WS-Calendar Technical Committee. CTS uses that simpler [Streams] specification.

See Appendix D, Semantic Composition from Energy Interoperation, EMIX, and WS-Calendar.

In [EI], the fundamental resource definition was the [EMIX] Item, composed of a resource name, a unit of measure, a scale factor, and a quantity. For example, a specific EMIX Item may define a Market denominated in 25 MWh bids. In CTS, we group and name these elements as a Resource, Product, and Instrument. These terms are defined in Section 2.2.4, “Markets and [Market Segments](#)”

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

1.9 Applicability to Microgrids (Informative)

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

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

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

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

1.10 Specific scope statements

This specification interprets Energy Interoperation from the perspective of a Trader interacting with a Market. CTS defines Pre-Trade, Trade, and Post-Trade information exchanges. Trading refers to the specific interactions that buy or sell a resource. A Trader uses pre-trade information to discern the operation of the Market and the actions of the other Traders. Post-Trade information informs the participants of the Trade, tracks whether the resource is delivered, and any resulting changes to the Trader’s ability to participate in the Market.

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

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

The following are out of scope for Common Transactive Services:

- Requirements specifying the type of agreement, contract, Product definition, or tariff used by a particular market.

- 272 • Computations or agreements that describe how power is sold into or sold out of a market.
273 • Communication protocols, although semantic interaction patterns are in scope.
274 This specification describes standard messages, the set of which may be extended.

275 1.11 Naming of Messages and Operations

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

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

283 *Cancel—Canceled* A previously created request is canceled.

284 For example, to construct an operation name for the Tender Facet, "Ei" is concatenated with the name
285 fragment (verb) as listed. An operation to cancel an outstanding Tender is called *EiCancelTender*.⁴

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

⁴ This pattern was developed and is used by IEC Technical Committee 57 (Power Systems).

2 Overview of Common Transactive Services

CTS provides for the exchange of resources among [actors, in the role of](#) parties, which represent any provider or consumer of energy ~~(e.g., a distributed energy system)~~. [Systems use CTS to interoperate in 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.](#)

~~Systems use the common transactive services to interoperate in 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.~~

Although the Common Transactive Services are a profile [and extension](#) of Energy Interoperation, the CTS focus is markets and trading. The language used in the Energy Interoperation specification was developed with extensive input from economists, regulators, and participants in highly regulated markets. [This profile](#)

[CTS](#) strives ~~instead for to use~~ the language of [financial](#) markets and traders.

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

2.1 Parties

~~The Common Transactive Services (This CTS) specification~~ defines [defines](#) interactions [between participants](#) in a resource market. This Resource Market is a means to make collaborative decisions that allocate power or other resources over time. We follow [EI] and financial markets by calling market participants “Parties”.

~~A Party may represent a single actor, or the roles (see Facets, below) of a single Party may be distributed across multiple Actors.~~ When the market recognizes tenders that match each other, however decided, the market generates a transaction that represents a contract (“Trade”) between the buyer and the seller. This transaction includes a party and a counterparty.

2.2 Trading semantics from FIX Protocol

The FIX Community [Protocol](#) divides messages into Pre-Trade, Trade, and Post-Trade Messages.

Pre-Trade messages convey information that traders need to discover how to use the market and to develop a strategy to buy and sell successfully. Pre-trade messages include advertisements and announcements (“Tickers”) of offers and contracts in the market, and negotiations between parties (“Quotes”). Other pre-Trade messages describe how the market itself works and what a Party can expect when interacting with the market.

Trade messages include submitting and cancelling orders (“Tenders”) to the market and executing contracts (“Transactions”) when orders to sell match (however defined) orders to buy.

Post-trade messages [in FIX](#) include [allocation, confirmation, settlement and, position and collateral management](#). [CTS does not include allocation or collateral management.](#)

For narrative purposes, this specification begins with the Trade facets: ~~Tenders and Contracts~~ [Transactions](#). It then discusses the post-trade facets of Delivery and Position. This covers all the functions in some transactive resource markets. This specification then describes Negotiation, an optional pre-Trade facet. It next describes the pre-trade market data reports (“Tickers”) that inform an Actor about the activities of other participants. The pre-trade [Market Instrument Report](#) [Market Data](#) facet provides summary information about Tenders currently held in the market. Finally, the pre-trade Market Structure facet conveys how a Trader may interact with the market, which includes how to find each Facet and which messages this market supports.

An Actor interacting with the market would first discover the market structure, subscribe to Tickers relevant to its interest, and then use the facets and messages that are permitted in this market to Trade.

A Party MAY not understand negotiation, or MAY skip subscribing to Tickers, but any party MUST be able to Trade.

When available, this specification references matching field names and field numbers from the FIX Protocol. FIX Protocol fieldnames are generally upper camel case and we follow their convention that the field name is followed by the field number in parentheses, as in FieldName(0).

2.2.1 Parties and Orders

In Energy Interop as in FIX, a trade is executed between two parties. While Energy Interoperation acknowledges only a Party and a Counterparty, FIX is more semantically rich.

What Energy Interoperation (and this specification) terms Tenders, FIX terms orders. An order that is on the book in the market is a Resting or Passive order. An order that enters the market to match a Resting order is the Initiating or Aggressive order. Passive orders increase market liquidity. Aggressive orders decrease market liquidity. when they match to existing orders. Regulators of financial markets are often interested in liquidity and in the ratios of Aggressive to Passive orders.

When it makes the discussion clearer, this specification uses the terms Resting, Passive, Initiating, and Aggressive as they are used in financial markets.

2.2.2 Instruments

Financial Markets trade financial instruments. CTS borrows this language from FIX. See Section 3, Market Semantics: Resource, Product, Instrument, for a discussion.

2.2.3 Market Crossing

Market Crossing refers to either the opening or to the closing of a market or market segment. A traditional exchange opens in the morning and closes in the afternoon. Tenders are not matched prior to market opening or after the market close.

In many markets, parties wishing to trade pay close attention to prices and volumes in the period around closing. Many traders prefer not to trade close to a crossing because it is a period of high price volatility on a market. Many markets announce aan indicative “closing price” and an anticipated indicative “opening price”, even asthough no transaction may have occurred at either of those prices.

As transactive resource markets are in essence markets in time of delivery, individual instruments can be considered to open and close as well. In a continuously open market segment, a rule might prevent trading more than 24 hours in advance. In that same market, an instrument for delivery of a resource between 10:00 AM and 11:00 AM may no longer be traded at noon in the previous day.

Transactive resource markets may have regulatory time limits on trading. Some electricity markets have banned transactions more than a day prior to delivery. CTS traders must be able to understand the local rules and adjust their trading tactics without human intervention. A Market MAY accept Tenders prior to the opening of the Market Segment or Instrument. Transactive market researchers have used tenders submitted prior to opening to generate opening prices in black-start scenarios. Others have used trade residue, which is the tenders left in the market after closing to seed real-time prices for unplanned energy use.

~~Consider a utility that provides day-ahead hourly pricing from the local bulk power market to retail customers. The prices for tomorrow may be posted at 9:00 AM each day. These prices may be good until 3:00 PM each day. The market may begin accepting tenders an hour before opening. This describes a market segment with two crosses each day, opening at 9:00 AM and closing at 3:00 PM. After 3:00 PM orders may be processed as a normal forward market using an order book to process tenders into transactions.~~

~~As transactive resource markets are in essence markets in time of delivery, individual instruments can be considered to open and close as well. In a continuously open market segment, a rule might prevent trading more than 24 hours in advance. In that same market, an instrument for delivery of a resource between 10:00 AM and 11:00 AM can no longer be traded at noon.~~

2.2.4 Markets and VenuesMarket Segments

~~Systems use the common transactive services to interoperate in 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.~~

~~Systems use the common transactive services to interoperate in 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.~~

A Market ~~MAY be divided up into~~ is composed of different ~~venues~~segments wherein different products are traded, perhaps with different rules. Following the FIX Protocol, we term these Market Segments, and we use the FIX classification ~~Venue~~Market Mechanism Type (MMT⁵) to name the market activities of each Segment. A Market may have one or many Market Segments.

2.3 Common Transactive Services Roles

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

2.3.1 Parties as Market Participants

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

A Party can take one of two Sides in Transaction:

- Buy, or
- Sell

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

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

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

2.3.2 Party and Counterparty and Transactions

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

⁵ The MMT standard originated from an initiative of the Federation of European Securities Exchanges aiming at improving the consistency and comparability of data from different data sources. In order for the MMT standard to become more widely recognized and adopted, MMT has been placed under the FIX Protocol Limited Trust. Since then, MMT has been developed together with a broader range of industry participants, including trading venues, data vendors and data users, and has expanded into new asset classes.

When the Market recognizes tenders that *match* each other (however defined), the market generates a transaction that represents an agreement between the buyer and the seller. This transaction includes [asending a EiTransaction message to both the](#) Party and a Counterparty. [If the match was composed from multiple Tenders, each Party receives an EiTransaction for each Tender matched.](#)

2.3.3 Facets in the CTS Specification

This specification refers to a ~~coherent~~[cohesive](#) set of interactions, that is, closely related requests and responses, as Facets. ~~An Actor~~[A Party](#) sends and receives defined messages through [one or more Facets](#). [A Party may be composed of one or more Actors, each with one or more Facets. A Party may communicate with its Facet](#)~~composite Actors through the same Facets or through other Facets not defined in this specification.~~

[Actors use Facets](#) to interact with other Actors that expose a complementary Facet. An Actor in a CTS-based system of systems may expose all Facets, a single Facet, or any collection of Facets. A particular Market may use some or all named Facets. A participant in a Market must include Actors supporting each Facet required in that Market; there is no requirement that each Actor supports all these Facets.

Detailed descriptions of each facet begin in Section 4.

Table 2-1: Facets Defined in CTS

Facet	Description
Registration	A Party must Register with a Market to participate in the Market Segments in that Market. See Section 4, “Party Registration Facet”.
Tender	Tenders are actionable offers to buy or to sell an Instrument at a given price. Tenders are may be sent to a specific counterparty or sent to the whole Market Segment and are generally private. It is possible to request that a Tender be advertised, published via a Ticker to all Parties in the Market Segment . See Section 5, “The Tender Facet”.
Transaction	A Transaction records the trade when a Tender to buy and a Tender to sell are matched. Each Party is notified of the creation of the Transaction. Note: a Tender for one side MAY match more than one Tender on the other side, and could generate multiple Transactions, potentially at different prices. See Section 6, “The Transaction Facet”.
Position	At any moment, a Party has a position which represents the cumulative amount quantity for each of the Instruments that the Party has previously transacted for within a bounding time interval across all Segments in the Market. A Position for an Instrument reflects the algebraic sum of all quantities previously bought or sold. Note that parties that can store or generate power or that can buy from another market MAY be able to sell more than their market position. See Section 7, “The Position Facet”.
Delivery	It is simplest to think of Delivery as a meter reading, although that meter may be virtual or computed. Some implementations may compare what was purchased or sold with what was delivered. What a system does after this comparison is out of scope. See Section 8, “The Delivery Facet”.
Negotiation	Negotiation covers uses messages that may lead to a Tender that will be accepted. Negotiation includes Requests for Quotes (RFQs), Indications of Interest (IOI), and Quotes, and Quote Responses. See Section 9, “The Negotiation Facet”.

Facet	Description
Tickers	A Ticker is a continuous live view of market interactions—consider the historical ticker tape. A Ticker is one form of Market Subscriptions as defined by FIX. See Section 1, “ Tickers ”
Market Instrument Summaries	A Market Instrument Summary is a compressed or summarized variant of Market Data as defined by FIX. See Section 12, “Instrument Data Subscriptions ”.
Market Structure Reference and Dynamic Data	The Reference Data Facet communicates Market Facet exchanges information about Reference Data that describes the Market and its Products and each Market Segments. Segment; Session data is more dynamic. An Actor may query the Market to discover the Resource and Products traded in a Market. While a Market trades a single Resource, it may consist of multiple Market Segments trading multiple Products. See Section 13 “Market Structure Reference Data: Market, Segment, and Session Subscriptions .”

Each of these facets includes multiple messages which are described starting in Section 3.2 below. Sometimes the use of one facet precedes the use of another facet, as Tenders may initiate messages that result in messages for the Transaction Facet.

2.4 Responses

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

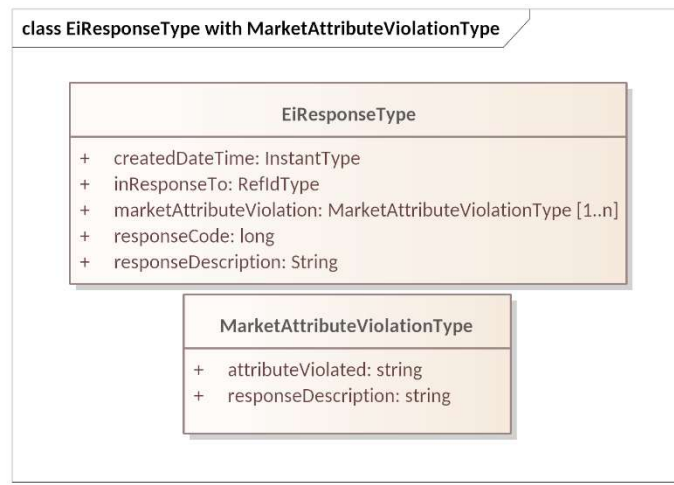
~~The Error! Reference source not found.~~Table 1: Responses

Attribute	Meaning
Created Date Time	Timestamp for creation of this response
Market Attribute Violation	Market and Segment attributes violated in the referenced request. See Section 13 Market Structure Subscription and Table 13-2 and Table 13-3.
In Response To	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 ID⁶.
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,⁷ 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
Response Description	A string describing the response, e.g. “Duration doesn’t match Segment configured Duration”

⁶ As an example of the *Correlation Pattern* for messages

⁷ See e.g. https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

shows UML class diagram for responses.



Most messages elicit a response. Information-only messages, as in Tickers, do not.

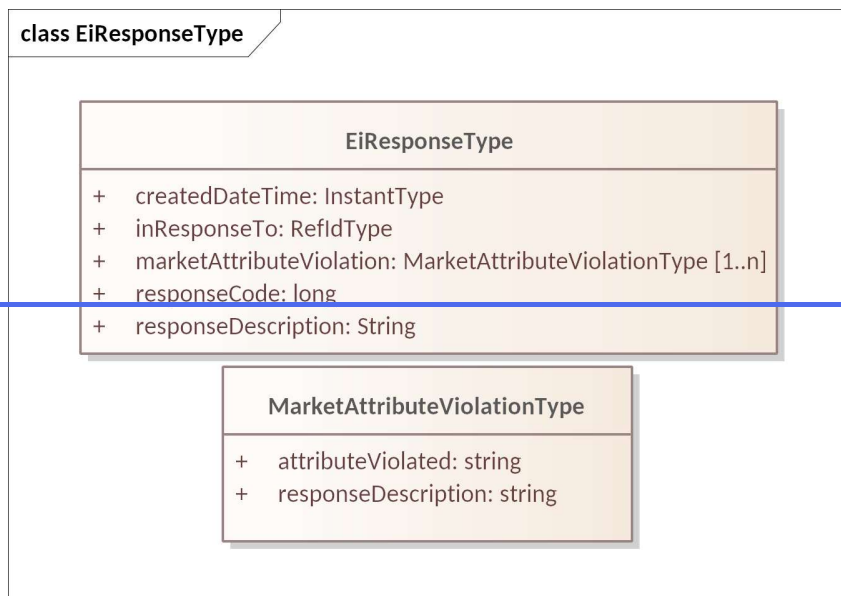


Figure 2-1 UML Class Diagram of EiResponseType and MarketAttributeViolationType

Attributes for responses are shown in Table 2-2. The various attribute types are not in FIX.

Table 2-2: Attributes of EiResponse

Attribute	Type	Meaning
Created Date Time	Instant Type	Timestamp for creation of this response
Market Attribute Violation	Pairs of strings	Market and Segment attributes violated in the referenced request. See Section 13 Market Structure Reference Data: Market, Segment, and Session Subscriptions.

Attribute	Type	Meaning
In Response To	Ref ID Type	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 ID⁸.
Response Code	Long	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,⁹ 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
Response Description	String	A string describing the response, e.g. “Duration doesn’t match Segment configured Duration”

Most messages elicit a response. Information-only messages, as in Tickers, do not.

2.5 Identities

In general, CTS uses specific types that inherit from UID Type, with a string as the inherited attribute. This allows representation of unique identifiers variously called UIDs, GUIDs, and other names, while maintaining type safety.

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

⁹ See e.g. https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

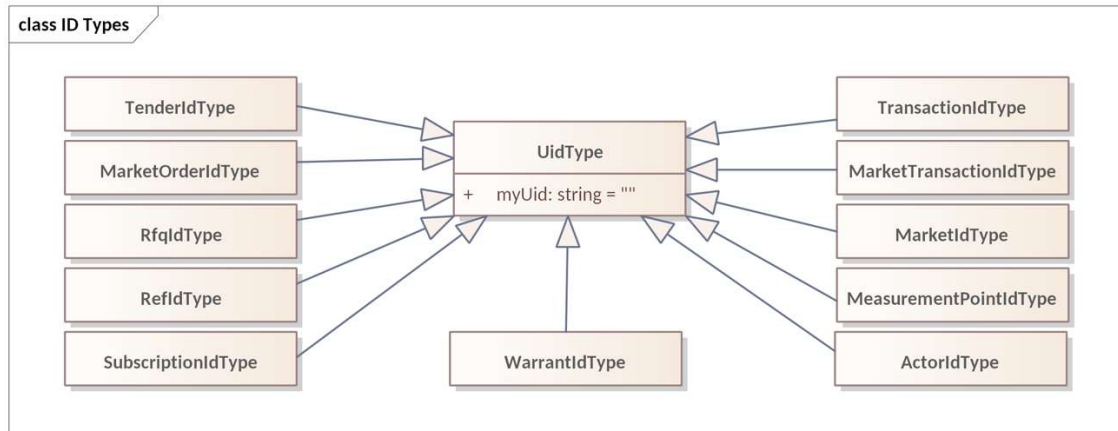
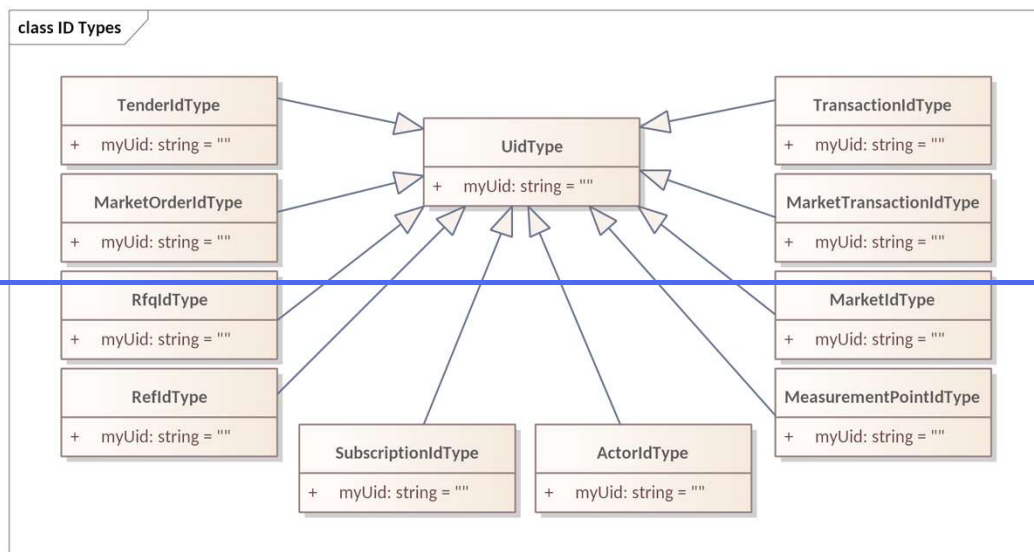


Figure 2-2 UML Class Diagram of ID Types in CTS



3 Market Semantics: Resource, Product, Instrument, and Streams

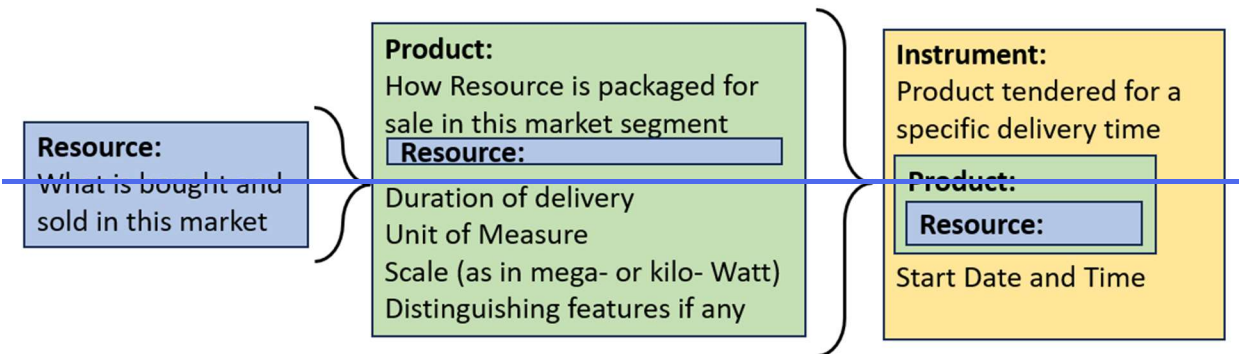
The messages of CTS use a few common elements. These elements derive from and are compatible with definitions in [WS-Calendar], [EMIX], and in [EI].

Every CTS-based market offers the exchange of a specific resource. Each CTS market segment is a venue for trading a single product, which is a resource packaged for sale. All tenders and transactions are for instruments, which are products scheduled for delivery at a specific time.

3.1 Resource, Product, & Instrument

We define a Resource as a commodity whose value depends on time of delivery. A Party subscribes (see Section 10) to a Market to discover the Resource that is traded in the market, and the Products available in different Market Segments. (See Section 13, “Market Structure [Reference Data: Market, Segment, and Session Subscriptions](#)”) A Party can then trade Instruments, a Product at a specific time, in a Market Segment. This specification leaves Market Definition until the end of the specification, as the meaning and import of the terms used to define each Segment are first described in the trading process.

Figure 3-1 illustrates the relationship between Resource, Product and Instrument. [This is expressed formally as UML in Figure 3-2](#)



[The relationship is illustrated twice, with an informal sketch and with formal UML below.](#)

[Understanding these three terms is essential to understanding CTS.](#)

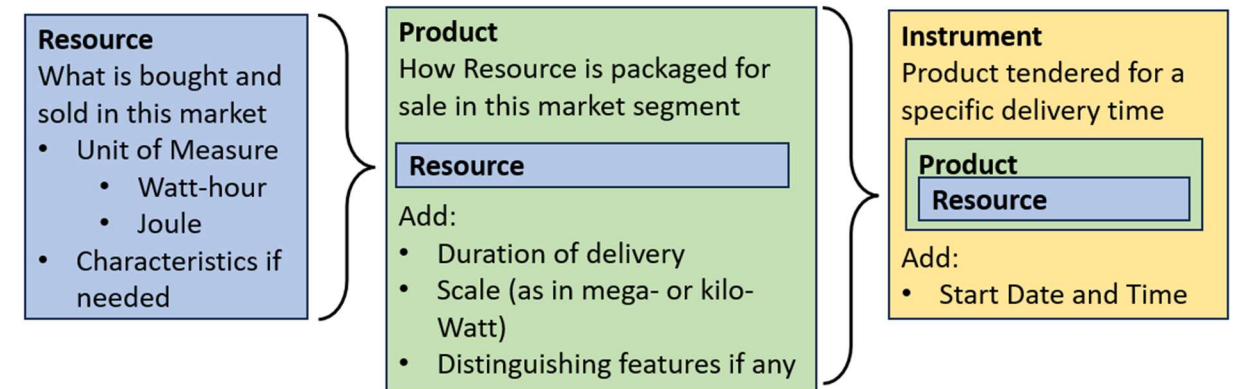
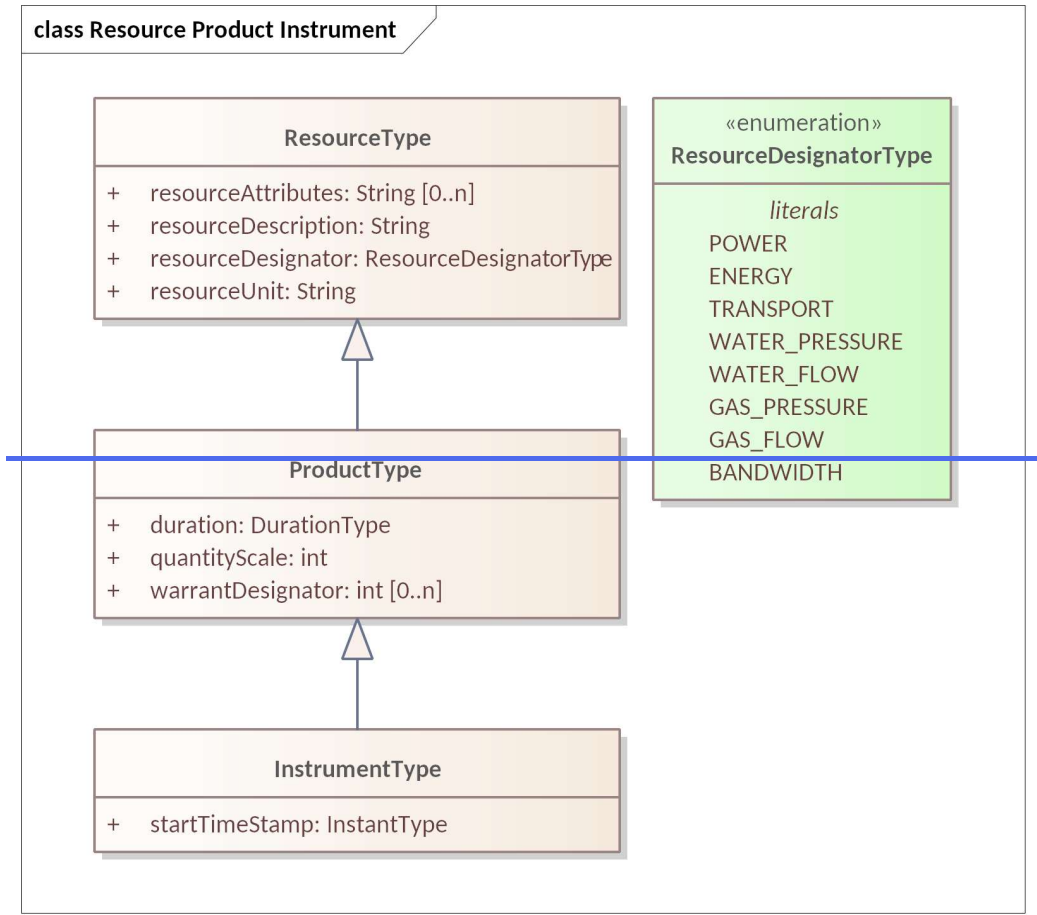


Figure 3-1 [Informal sketch](#) showing the relationship between Resource, Product, and Instrument

The Product incorporates the Resource, defining how the Resource is “packaged” for market. Adding a start date-time to a Product defines an Instrument.

A Market Segment trades Instruments, as a financial market trades financial instruments. CTS trades Instruments to deliver Product at a specific time.

The UML in Figure 3-2 shows the relationship between Resource, Product, and Instrument.



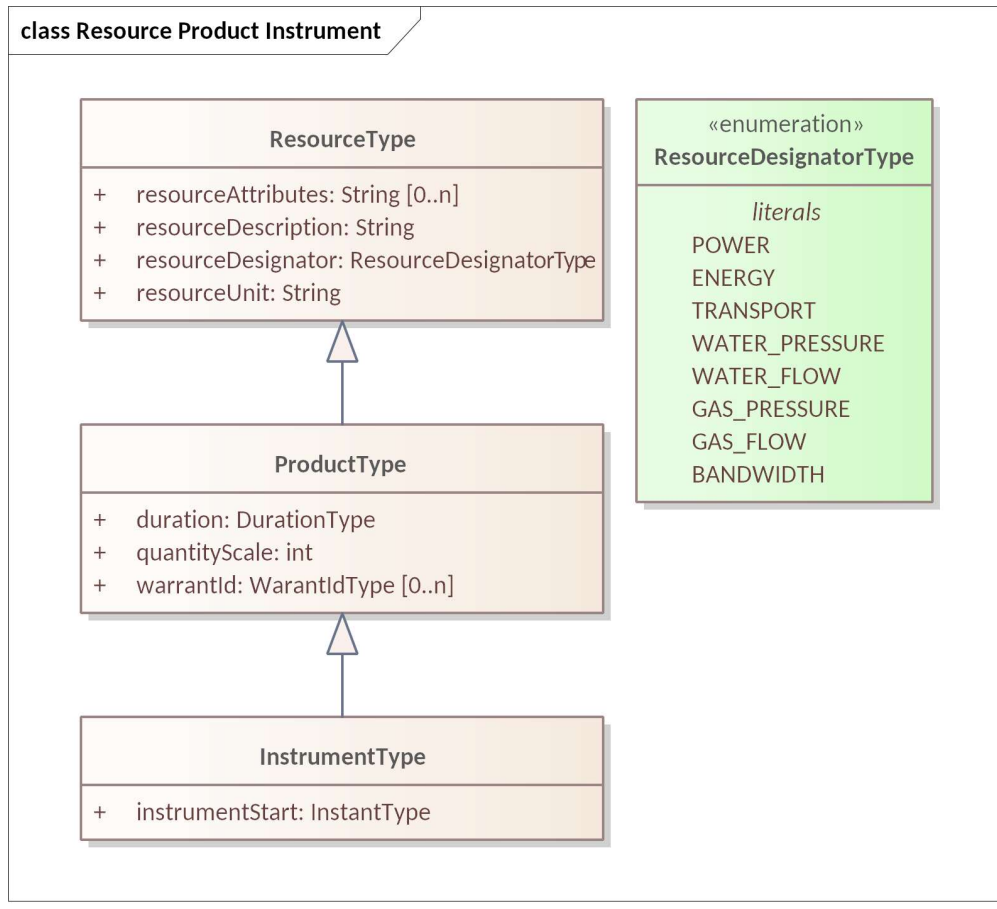


Figure 3-2 UML Class Diagram for Resource, Product, and Instrument

3.1.1 Defining Resource

We define a Resource as a commodity whose value depends on time of delivery. A developer may extend the Resource enumeration using standard UML techniques (subclassing); however, CTS 1.0 uses only the limited list in the Resource Designator Type (Figure 3-2).

A Market typically includes some information that further specifies the Resource, for example voltage and frequency for Power.

Table 3-1: Defining the Resource

Attribute	Type	FIX Field Name	Meaning	Notes
Resource Attributes	String	Not in FIX	Optional elements that further describe the Resource	e.g. Hertz and Voltage. Different Commodities will require different attributes to be specific.
Resource Description	String	Not in FIX	Text description of the Resource	

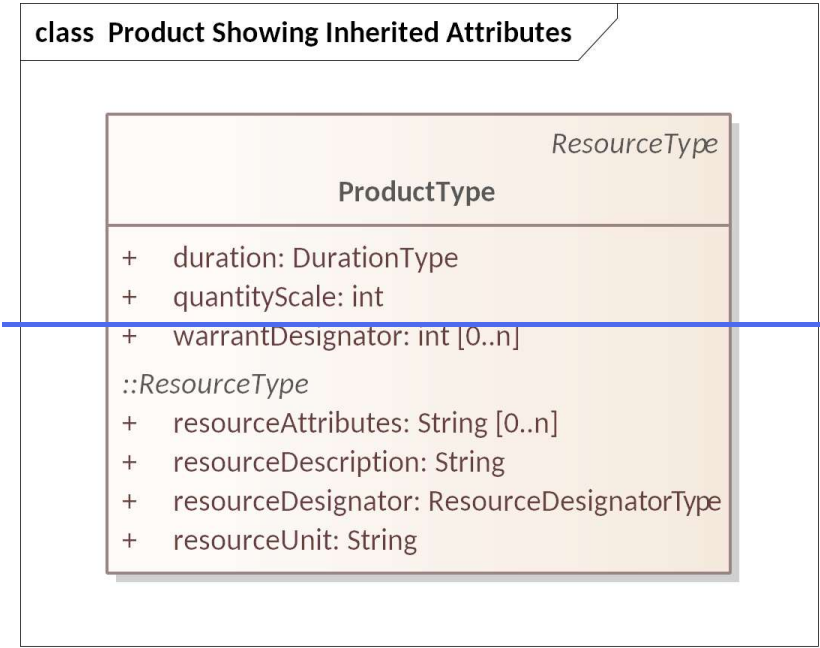
Attribute	Type	FIX Field Name	Meaning	Notes
Resource Designator	String	Not in FIX	POWER ENERGY TRANSPORT WATER_PRESSURE WATER_FLOW GAS_PRESSURE GAS_FLOW BANDWIDTH	The Resource Designator serves a purpose similar to that of the FIX AssetSubClass(1939) with AssetClass(1938)=5 (Commodity) The list is extensible
Resource Unit	String	Not in FIX	The unit of measure for the Resource	Item Unit in [EMIX] The Resource Unit serves a purpose similar to that of the FIX UnitOfMeasure (996)
Resource Attributes	String	Not in FIX	Optional elements that further describe the Resource	e.g. Hertz and Voltage

The Resource is named in the Market. Each Market deals in a single Resource. Segments of a Market restrict trading into profiles of the Resource. Position and Delivery (see Sections 7, 8 below) itemize Resource quantities.

3.1.2 Defining Product

The Product is a Resource packaged for Market. The size and duration of the Product define what is, in effect, the “package size” for the commodity. A Market may offer multiple Products for the same Resource in different Market Segments.

Note that the Product is derived from the [EMIX] ItemBase.



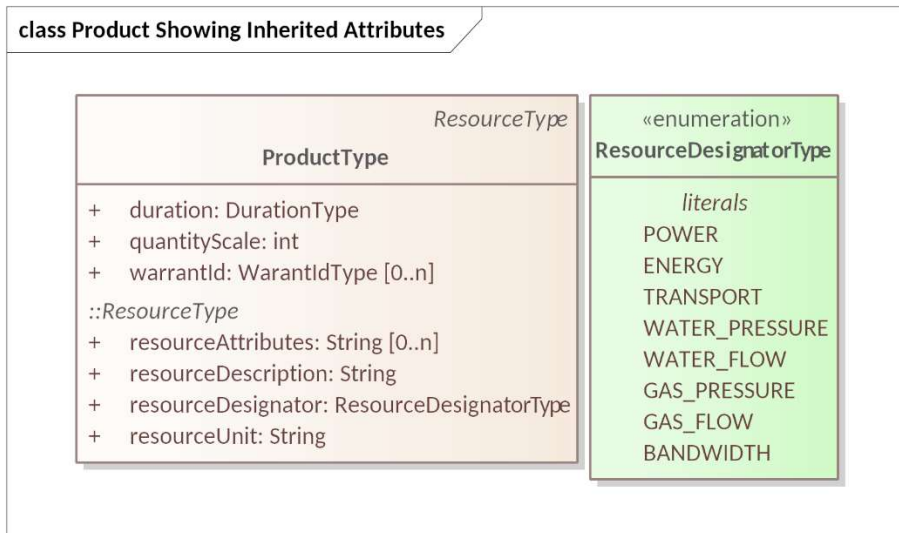


Figure 3-3 UML Class Diagram for Product showing Inheritance from Resource

Table 3-2, below, defines each of the fields in the Product.

Table 3-2: Defining the Product

Attribute	Type	FIX Field Name	Meaning	Notes
Duration	DurationType String	Not in FIX	The interval Duration for the specific Product definition.	As defined in [WS-Calendar]
Quantity Scale	Integer Int	Not in FIX	The exponent of the Quantity	For example, a Product denominated in kilowatts has a QuantityScale of 3.
Resource Designator	String	Not in FIX	Reference to the Resource as defined in this Market.	Used to support Tender validation and auditing
Resource Unit	String	Not in FIX	The unit of measure for the Resource	Item Unit in [EMIX]
Warrant Designator (Optional)	Int Warrant ID Type	Not in FIX	Optional further specificity of Product.	Warrants are itemized in the Market. This specification does not define Warrants.
Other attributes are inherited from Resource Type (Table 3-1)				

Products with differing Warrants are different Products and therefore traded in different Market Segments.

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

Warrants are defined in [EMIX], and CTS permits Warrants to support this complexity if desired, but not described in this specification. A Market MAY define a list of Warrants and Warrant Designators. Warrants

were defined in [EI]-1 as additional non-essential characteristics of a Resource such as how it was produced, or an attribute of regulatory interest. Warrants are defined in the Market but are offered per Market Segment.

3.1.3 Defining Instrument

A Market Segment trades Instruments for a single Product. In CTS, an Instrument is a Product delivered for a specific duration beginning at a certain time. CTS includes Duration explicitly in both the Tender and the Quote. The Instrument follows the pattern defined in WS-Calendar—a Resource bound to a Duration (“forming a Product”) and the Product bound to a Starting DateTime.

[The Instrument Start time added to a Product creates an Instrument. See Figure 3-2.](#)

Table 3-3: Specifying the Instrument

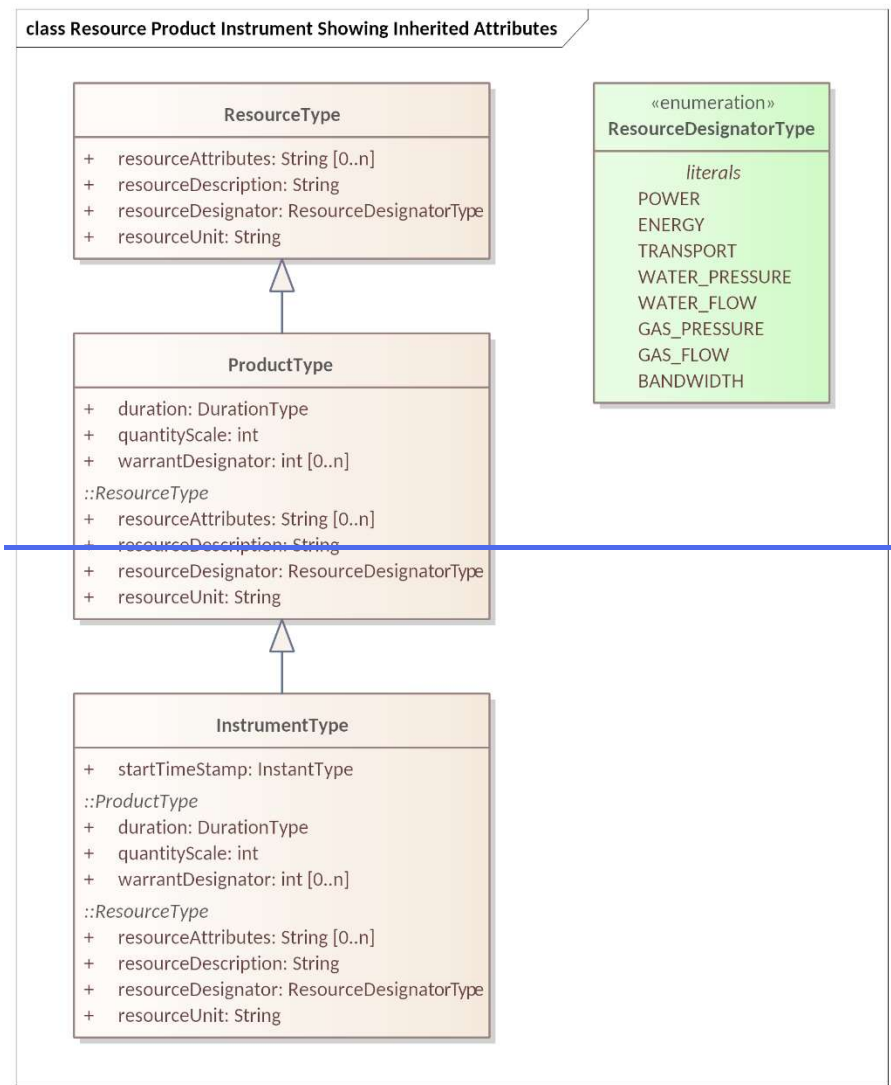
Attribute	Type	FIX Field Name	Meaning	Notes
The fields below as defined in the Product, Table 3-2				
Duration	String	Not in FIX	The interval Duration for the specific Product definition.	As defined in [WS-Calendar]
Quantity Scale	Int	Not in FIX	The exponent of the Quantity	For example, a Product denominated in kilowatts has a QuantityScale of 3.
Resource Designator	String	Not in FIX	Reference to the Resource as defined in this Market.	Used to support Tender validation and auditing
Resource Unit	String	Not in FIX	The unit of measure for the Resource	Item Unit in [EMIX]
Warrant Designator (Optional)	Int	Not in FIX	Optional further specificity of Product.	This specification does not define Warrants.
A start time completes the Product (above) into a tradable Instrument				
Instrument Start Time	DateTime (UTC)Instant Type	Not in FIX	Starting Date & Time	A start time completes the specification of Product into a tradable Instrument. The Start Time serves a purpose similar to that of the FIX repeating group EvntGrp with EventType(865)=21 (Delivery start time).
The fields are inherited from Product Type. Table 3-2				

Every Tender, Transaction, and Quote is to buy or sell a quantity of an Instrument.

Within a market Segment, the Start Date and Time uniquely identifies an Instrument. Because an off-market Segment, sometimes known as over-the-an Over The Counter (OTC) Segment can transact products of any Duration, Tenders, Quotes, and Transactions all use the Segment identifier, the Start Time, and the Duration to identify the Instrument and Product.

3.1.4 Summary of Instrument Specification

A UML model for the Instrument showing all inheritance is in Figure 3-4 below:



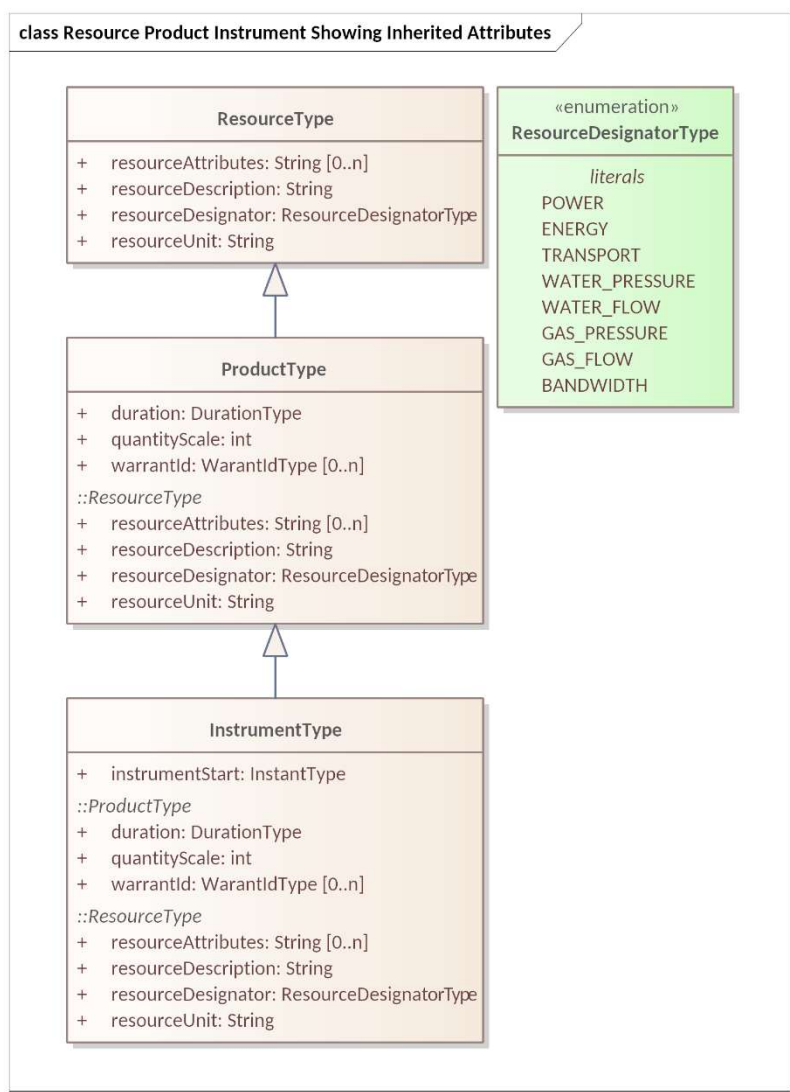


Figure 3-4 UML Class Diagram for Instrument showing Inheritance from Resource & Product

3.2 CTS Streams: Expressing Time Series

Resource Markets are based on time-of-delivery. It is often useful to convey requests and information about consecutive durations. This specification uses the simplified pattern described in WS-Calendar [Streams], that is, common information followed by a repeating set of information for each consecutive Interval. Each Interval uses a common Duration. All Intervals in a Stream are consecutive.

~~Streams are a response to a request for a Stream. A Request for Quotes may specify Intervals for an entire day. A Request for a Position or for Delivery may also include several Intervals. In either case, the Request includes the common information (as in what is requested, a Duration, and a bounding interval, expressed as a Start DateTime, and an End DateTime).~~

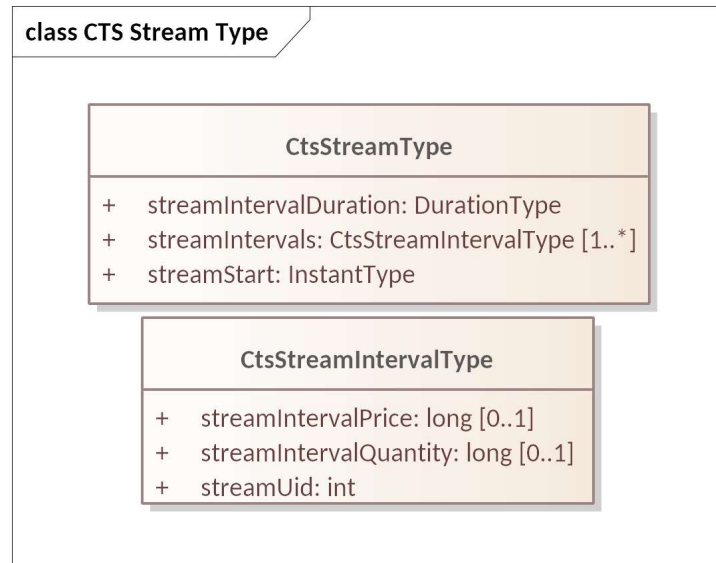


Figure 3-5: UML Class Model for CtsStream and the Stream Intervals

The response to a request for a stream is a stream.

For example, the common information in [a TenderStream, derived from the CTS StreamsStream](#), is the Product and the Start DateTime [for the first element of the Stream](#). The Product specifies Resource and Duration. The consecutive intervals in the CtsStream begin with the Start DateTime for the specified Duration. The second Interval has an implied start of the end of the first Interval. The third Interval has an implied start of the end of the second Interval...and so on.

Each interval carries what can be considered a *local UID*.¹⁰

Several Facets request a CtsStream in the response. They are:

- Position Facet
- Delivery Facet

Certain payloads may include a CtsStream, including:

- Tender Facet (see “Interval Tenders and Stream Tenders”, Section 5.3.1)
- Quote and Negotiation Facet (see *Stream Quote*)

shows payloads for the Position and Delivery Facets as an example of the pattern for requesting and responding with streams.

Table 3-4: Specifying the Stream

Attribute	Type	FIX Field Name	Meaning	Notes
Stream Interval Duration	StringDuration Type	<i>Not in FIX</i>	The interval Duration for each Stream element.	As defined in [WS-Calendar] Optional if inherited from message containing Stream

¹⁰ Certain serializations for payloads do not guarantee order, so a small integer serves as a unique identifier for each interval.

Attribute	Type	FIX Field Name	Meaning	Notes
Stream Start	Date Time Instant Type	<i>Not in FIX</i>	Starting Date & Time for the first element in the series of Intervals.	After the first Interval, each Interval starts when the proceeding preceding Interval finishes
Stream Interval Price Value	Long	Price (44)	Price per Unit during Interval	Optional depending upon purpose of message including Stream
Stream Interval Quantity Value	Long	OrderQty (38)	The Quantity of the Product during the Interval	Optional depending upon purpose of message including Stream
StreamUID	Integer Int	<i>Not in FIX</i>	Unique identifier for each interval; local to the Stream instance .	Certain serializations for deserializations do not guarantee order- --- the UID enables it to be reconstructedreconstructing the order . A simple integer suffices as a sortable UID for streams .

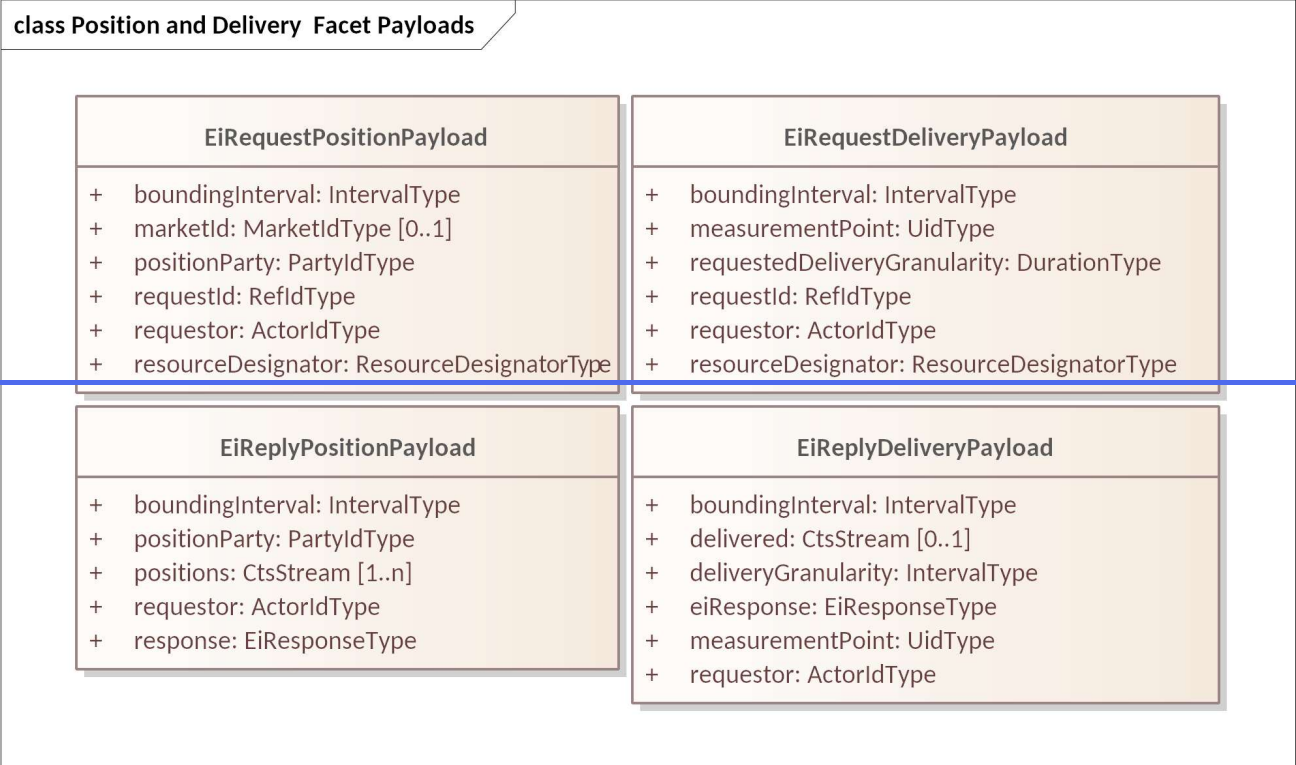
3.3 The Bounding Interval Pattern in CTS ~~pattern includes~~

~~The CTS requests may include~~ a Bounding Interval, ~~and~~. The response ~~requests~~ is ~~typically~~ all ~~CtsStream~~
Intervals ([CTS Stream Intervals, or Instruments](#)) that are contained within the Bounding Interval including
those which align with the ends of the Bounding Interval.

~~More formally, given a request including a Bounding Interval the request will return information on all~~
~~Instruments or Stream Intervals within the Bounding Interval whose start is at or later than the Bounding~~
~~Interval start and whose end point is at or before the end of the Bounding Interval.~~

573

[See Section 7](#) “The Position Facet



574

575

[Figure - : UML Examples of Payloads incorporating Streams](#)

576

”

577

The information within each Interval varies per message type. For example, a StreamQuote ~~or a~~ [StreamTender](#) will put the Price and Quantity in each interval. A Delivery (metering) payload will put only the Quantity in each Interval.

578

579

4 Party Registration Facet

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

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

EiCreateParty associates an actor with a Party ID and informs the Market of that ID. CTS makes no representation on whether that ID is an immutable characteristic, such as a MAC address, a stable network address, such as an IP, or assigned during registration.

EiRegisterParty names the exchange of information about an actor that enables full participation in a CTS Market. It may exchange information needed for financial transfers including, perhaps, reference to an existing customer or vendor ID, or proof of financial bond for large participants, or issuance of crypto-tokens, or any other local market requirements. A Registered Party is ready to be a full participant in the local Market.

Cancel Party Registration removes a Party from the Market. It may include final settlement, cancellation of outstanding Tenders, backing out of future contracts, or other activities as defined in a particular CTS Market.

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

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

Some Markets MAY wish to associate one or more measurement points with a Party. Such measurement points could be used to audit Transaction completion, to assess charges for using uncontracted-for-energy, etc. Measurement points are referenced in *Section 8 "The Delivery Facet"*, Markets that require this functionality may want to include an enumeration of Measurement Points in Party Registration.

An implementation is not required to use the Party Registration Facet. For example, if uniqueness and universality are satisfied, any assignment of Party IDs should work.

5 The Tender Facet (~~TradeOrder~~ messages)

A party wishing to buy or sell submits an order (“Tender”) ~~to using~~ the Tender Facet. The Service descriptions and payloads ~~in [Ei]~~ are simplified and updated ~~from those defined in EICTS~~. The FIX Protocol classifies Tenders as one of the ~~TradeOrder~~ messages. ~~Simple Tenders are handled as what the FIX Protocol would describe as Simple General Orders and are handled as in FIX SimpleGeneralOrderHandling.~~

5.1 Messages for the Tender Facet

~~TradeParties exchange Order~~ messages ~~are exchanged between parties~~ to find or create a Transaction. The Tender Facet payloads are shown in Table 5-1.

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 5-1: Tender Facet Payloads

Facet	CTS Initial Message	CTS Response Message	Meaning
EiTender	EiCreateTender	EiCreatedTender	Create A Party sends a Create message containing one or more Tenders- Created to requesting that the [Market] ¹¹ create a Tender. The [Market] returns the Created acknowledgement or returns errors, and when successful returns the Market-assigned ID for the submitted Tender
EiTender	EiCancelTender	EiCanceledTender	Cancel one or more Tenders

In the FIX specification, a Tender is “completed” when it is ~~satisfied~~~~fully filled~~, when it is cancelled, or when it is replaced. CTS does not permit replacing tenders, instead requiring that a Party cancel a tender and submit a new one. If a Tender is already partially filled, cancellation cancels only the unfilled portion.¹²

5.1.1 Illustrative Narrative on Tenders [Non-Normative]

For example, Party A submits a Tender 1 to buy 100 kWh over an hour. A Tender from Party B for 45 kWh matches Party A’s Tender and the Market creates a Transaction (see Section 6, “*The Transaction Facet*” for a discussion of Transactions). A Tender from Party C for 35 kWh matches Party A’s Tender and the Market creates a Transaction. Party A’s Tender 1 remains on the market with 20 kWh remaining. If Party A wishes to increase the price offered to get the 20 kWh for a critical operation, Party A must cancel Tender 1, with 20 kWh remaining, and submit a Tender 2 offering a new price. Cancelling Tender 1 does not invalidate either of the two completed Transactions.

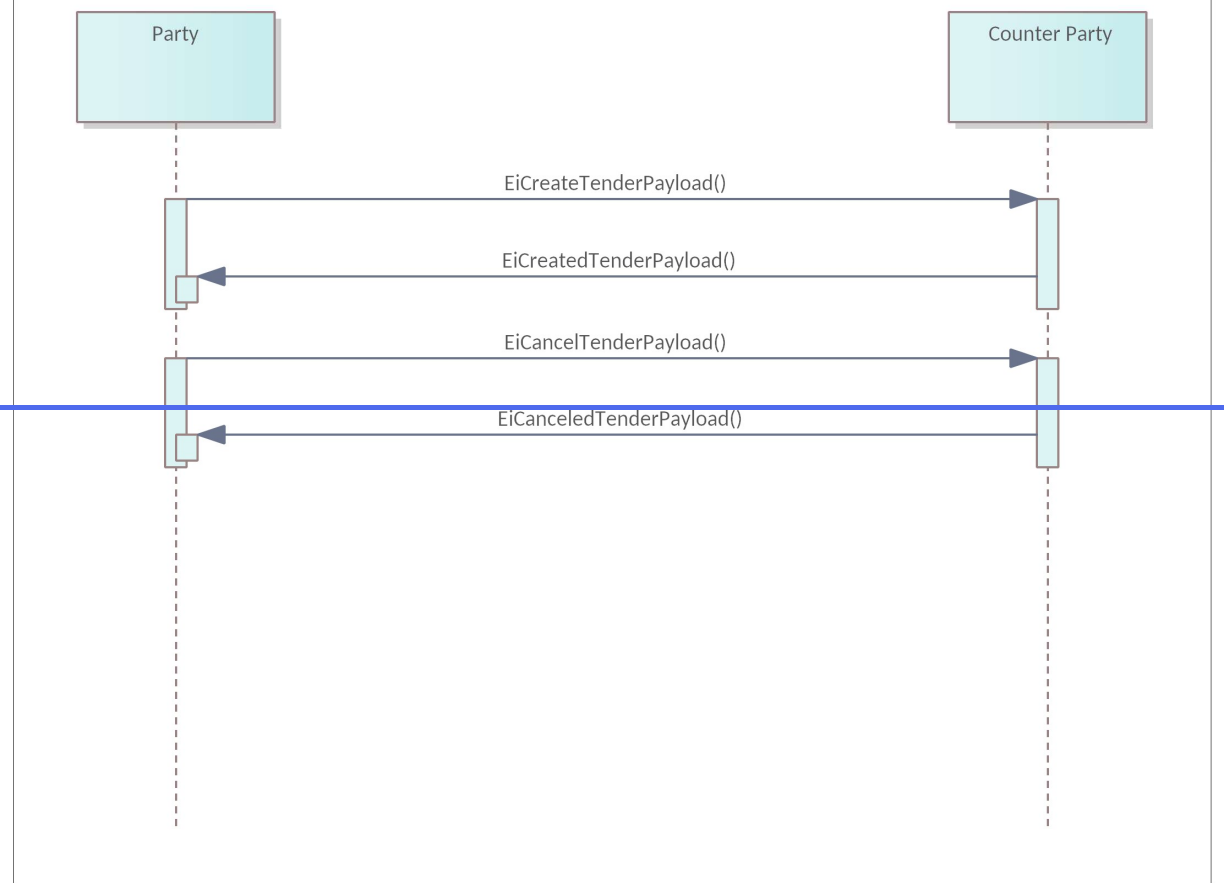
¹¹ See Section 9 “*The Negotiation Facet*” and Section 13.1, “*Market Mechanisms*” for discussions where the message target may not be the Market.

¹² This avoids a potential race condition in variable latency distributed systems.

630 5.2 Interaction Patterns for the Tender Facet

631 Figure 5-1 presents the UML sequence diagram for the EiTender Facet. Note that while [EI] defines a
632 message EIDistributeTender, CTS uses the Negotiation Facet (Section 9, “*The Negotiation Facet*”) and
633 Ticker Subscriptions (Section 1, “[Tickers](#)”) to accomplish similar purposes.

sd Tender Facet Sequence Diagram



sd Tender Facet Sequence Diagram

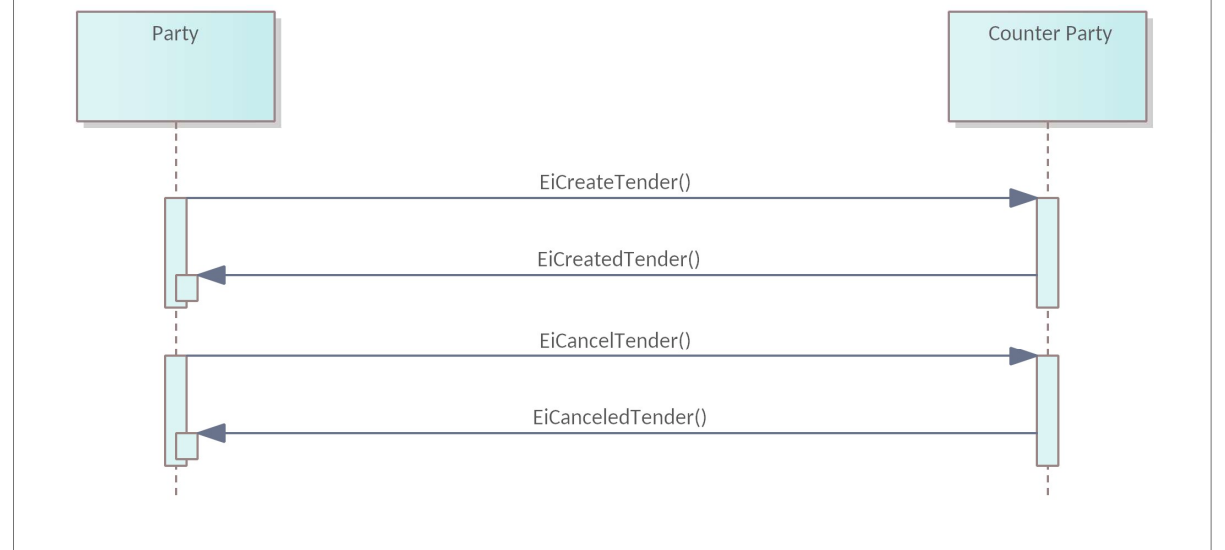


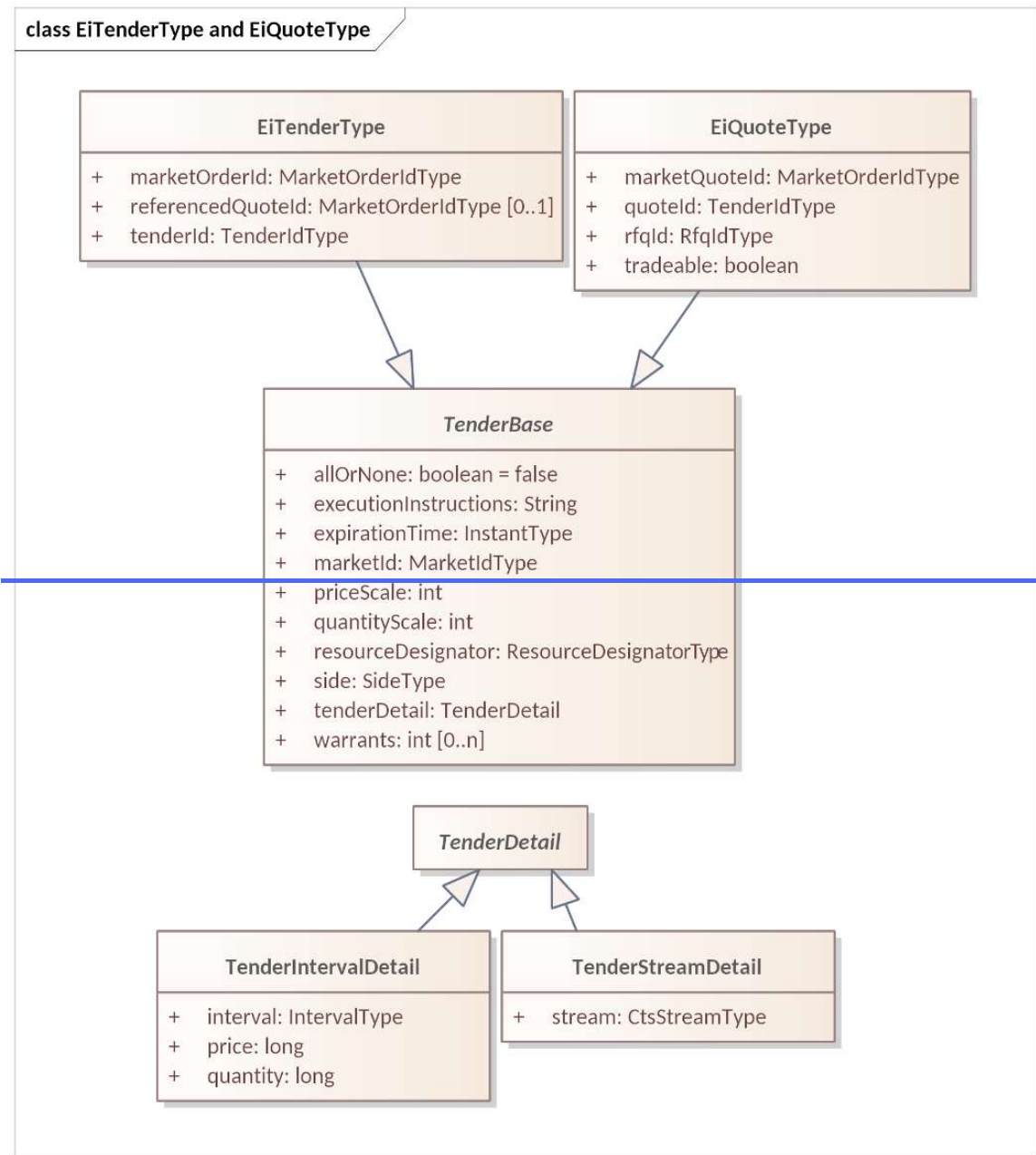
Figure 5-1: UML Sequence Diagram for the Tender Facet

5.3 Information Model for the Tender Facet

The information model for the Tender Facet artifacts follows that of [EMIX] but flattened and with Product definition implied by the implementation. See Section 5.6 Message Payloads for the Tender Facet below.

The [EiTenderTender](#) and [EiQuoteQuote and RFQ](#) classes share most attributes in common. Accordingly, a superclass [TenderBaseTender Base](#) holds those common attributes as shown in Figure 5-2.

TenderBase is an abstract class, so no object can be of that class.



class EiTenderType and EiQuoteType and EiRfqType

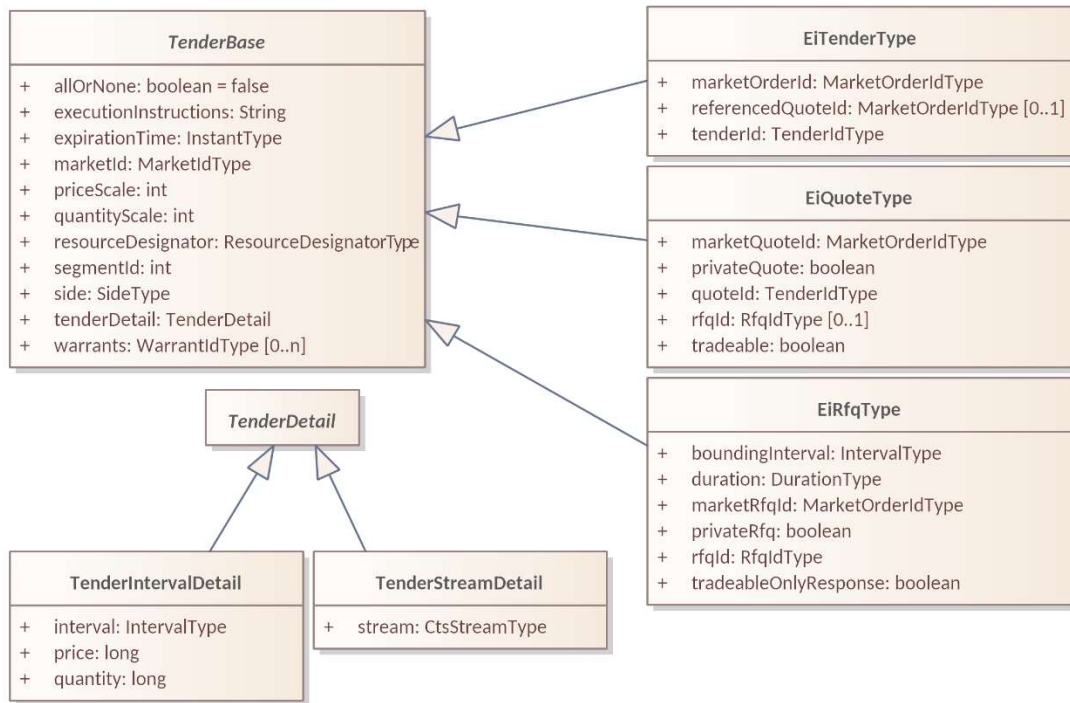


Figure 5-2 UML Class Diagram Showing [Commonality between Tender, Quote, and RFQ](#)

[Figure 5-3 shows all attributes for EiTenderType and their sources.](#)

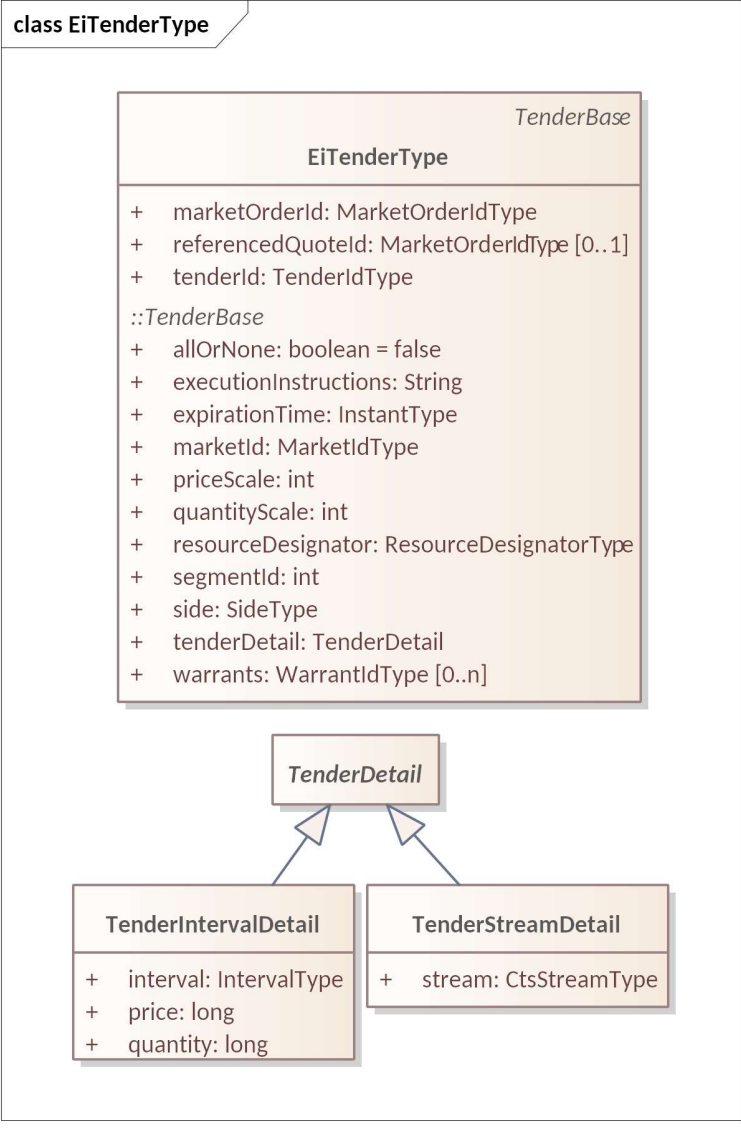


Figure 5-3.UML Class Diagram showing EiTenderType

[EiTenderType](#) inherits from [TenderBase](#), which holds the common attributes between [EiTenderType](#), [Tender](#), [Quote](#), and [EiQuoteTypeREQ](#). Attributes used in [Tenders](#), [and TenderBase](#) are shown in [Table 5-2](#) and [Table 5-3](#). Of the attributes in [Table 5-2](#) [Tender ID](#) and [Referenced Quote ID](#) ([Referenced Quote Id](#)) are unique to [EiTenderType](#); the others are inherited from [Tender Base](#) and shared with [EiQuoteType](#) and [EiRfqType](#). See [Section 9, “The Negotiation Facet”](#), for a discussion of [Quotes](#) and [Requests For Quotes](#).

Table 5-2 Tender, EiTender Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
All or None	Boolean	In Execution Instructions	All or none of the tendered or quoted amount must be traded.	In Energy Interoperation 1.0 this was called <i>IntegralOnly</i> .
Execution Instructions	String	ExecInst (18)	FIX Supports many instructions for how to execute a tender (or Tradable Quote)	See Table 5-4 below. Modeled as a String in CTS.
Expiration Time	Instant Type	ExpireTime (126)	The Tender or Quote expires at the specific time.	Always expressed in UTC
Interval	Start Time and Duration	Not in FIX	Start Date and Time for Product delivery together with Duration of delivery. Part of Instrument	While a Market Segment only accepts Tenders and Quotes of a single configured duration, the complete description is required to ensure validity and for off market interactions.
Market ID	UID	MarketID (1301)	Identifies the Market	
Market Segment ID	UID	MarketSegmentID (1300)	Identifies the Segment processing the Tender, Transaction, or Quote	This should be a unique combination paired with the Market Order ID
Order ID	UID	ClOrdID (11)	ID assigned by originating Party	
Market Order ID	UID <u>Market Order ID Type</u>	OrderId (37)	ID A market-assigned by the Segment or Market unique identifier for an Order (Tender in CTS)	Used in acknowledgment and in all future market messages
Price	Long	Price (44)	The unit price for the Product being Tendered	Amount is the product of Price and Quantity. Note that Price is subject to the Price Decimal Fraction value.
Price Scale	Int	Not Defined in FIX	A multiplier for the Price	A Market Segment may be denominated in, for example dollars or 10ths of a cent.
Quantity	Long	OrderQty (38)	The quantity of the Product being Tendered	Must meet the SCALE and Round Lot requirements of the Segment
Quantity Scale	Int		A scale factor on the Resource unit for this Market	For example, “mega” vs “kilo” vs “femto”

Attribute	Type	FIX Field Name	Meaning	Notes
Referenced Quote ID	UID	QuoteMsgID (1166)	<u>Tradable Quote reference in TenderID of the Tradeable Quote to which this is a response.</u>	<u>Reference to Market Quote ID assigned by Market. See Negotiation Facet (Section) for use.Optional. If Quote ID is not known to the Market Segment, or if the referenced Quote has expired, then the Tender is rejected.</u>
<u>Tender ID</u>	<u>Tender ID Type</u>	<u>ClOrdId(11)</u>	<u>An ID for this Tender generated by the submitting Party</u>	
<i>Other attributes are inherited from TenderBase—See Table 5-3</i>				

The complete description of the Interval for a Tender is in the TenderDetail—either an Interval with a price and quantity, or a Cts Stream with that information for each Stream Interval.

While a Market Segment only accepts Tenders and Quotes of a single configured duration, the complete description is required to ensure validity and for off-market interactions.

Table 5-3 Tender Base Attributes

Attribute	Type	FIX Field	Meaning	Notes
<u>All or None</u>	<u>Boolean</u>	<u>In FIX, this one among many Execution Instructions</u>	<u>All or none of the tendered or quoted amount must be traded.</u>	<u>In Energy Interoperation 1.0 this was called <i>IntegralOnly</i>. In CTS, this is promoted from Execution Instruction to top-level attribute.</u>
<u>Execution Instructions</u>	<u>String</u>	<u>ExecInst (18)</u>	<u>FIX Supports many instructions for how to execute a tender.</u>	<u>See Table 5-4 below. Modeled as a String in CTS.</u>
<u>Expiration Time</u>	<u>Instant Type</u>	<u>ExpireTime (126)</u>	<u>The Tender or Quote expires at the specific time.</u>	<u>Always expressed in UTC</u>
<u>Market ID</u>	<u>Market ID Type</u>	<u>MarketID (1301)</u>	<u>Identifies the Market</u>	<u>Note that in FIX, this is generally a formal identifier (e.g., “NYSE”). If the market is a house, there is no place to look this up. There is always a UID for a Market.</u>
<u>Price Scale</u>	<u>Integer</u>	<u>Not Defined in FIX</u>	<u>A multiplier for the Price</u>	<u>Must match the price Scale of the Segment.</u>
<u>Quantity Scale</u>	<u>Integer</u>		<u>A scale factor on the Resource unit for this Market</u>	<u>For example, “mega” vs “kilo” vs “femto”</u>

Attribute	Type	FIX Field	Meaning	Notes
Resource Designator	Resource Designator	Not defined in FIX	Identifier of the Resource being offered (Optional in many markets)	While a Market only accepts Tenders and Quotes for a single Resource, the complete description is required to ensure validity and for off-market interactions.
Tender Detail	Tender Detail	Not defined in FIX	Unit price and quantity for this tender	May be Interval or Stream as permitted
TenderSegment ID	UID	MarketSegmentID (1300)	Identifies the Segment processing the Tender, Transaction, or Quote as submitted to Market	Identifies Tender until This should be a unique combination paired with the Market Order ID is assigned by Market
Tender Interval	Tender Interval Detail	Not defined in FIX	Interval, price and quantity for this tender	Used in Interval Tender
Tender Stream	Tender Stream Detail	Not defined in FIX	Stream of consecutive Intervals with Prices and Quantities	Sometime referred to as a Load Curve in Power Markets.
Side	Side Type	Side (54)	Whether the Tender is to buy or to sell the Product	Buy or Sell side
Warrant Ref	Int	Not in Fix	Reference to Warrants as defined in the Market	If used, see comments Warrants in Tenders, Section 5.3.3.

663

The following diagram shows EiTenderType showing inherited and included attributes.

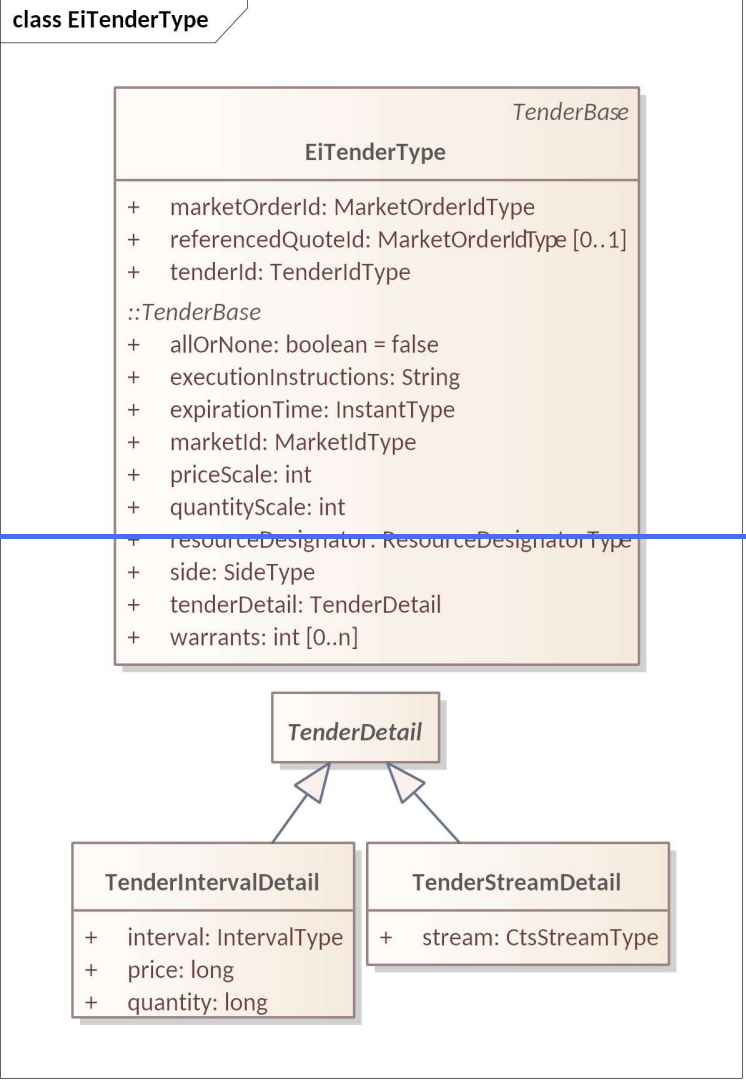


Figure -: UML Class Diagram for EiTenderType showing attributes inherited from Tender Base

Of the attributes in Tender ID and Referenced Quote ID (Referenced Quote Id) are unique to EiTender Type; the others are inherited from Tender Base and shared with EiQuote Type. See Section , “”, for a discussion of Quotes.

Table -: EiTender Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
Tender ID	UID		An ID for this Tender generated by the submitting Party	
Referenced Quote ID	UID	QuoteMsgID (1166)	ID of the Tradable Quote to which this is a response.	Optional. If Quote ID is not known to the Market Segment, or if the referenced Quote has expired, then the Tender is rejected.

Attribute	Type	FIX Field Name	Meaning	Notes
The fields below are as defined in the Tender, Table 5-2				
All or None	Boolean	In-Execution Instructions	All or none of the tendered or quoted amount must be traded.	In Energy Interoperation 1.0 this was called <i>IntegralOnly</i>
Execution Instructions	String	ExecInst (18)	FIX Supports many instructions for how to execute a tender (or Tradable Quote)	See Table 5-4 below. Modeled as a String in CTS.
Expiration Time	Instant Type	ExpireTime (126)	The Tender or Quote expires at the specific time.	Always expressed in UTC
Interval	Start Time and Duration	<i>Not in FIX</i>	Start Date and Time for Product delivery together with Duration of delivery. Part of Instrument	While a Market Segment only accepts Tenders and Quotes of a single configured duration, the complete description is required to ensure validity and for off-market interactions.
Market ID	UID	MarketID (1301)	Identifies the Market	
Market Segment ID	Int	MarketSegmentID (1300)	Identifies the Segment processing the Tender, Transaction, or Quote	This should be a unique combination paired with the Market Order ID
Order ID	UID	ClOrdID (11)	ID assigned by originating Party	
Market Order ID	UID	ORDERID (37)	ID assigned by the Segment or Market.	Used in acknowledgment and in all future market messages
Price	Long	Price (44)	The unit price for the Product being Tendered	Amount is the product of Price and Quantity. Note that Price is subject to the Price Decimal Fraction value.
Price Scale	Int	Not Defined in FIX	A multiplier for the Price	A Market Segment may be denominated in, for example dollars or 10ths of a cent.
Quantity	Long	OrderQty (38)	The quantity of the Product being Tendered	Must meet the SCALE and Round Lot requirements of the Segment
Quantity Scale	Int		A scale factor on the Resource unit for this Market	For example, “mega” vs “kilo” vs “femto”
Resource Designator	Text	Not defined in FIX	Identifier of the Resource being offered (Optional in many markets)	While a Market only accepts Tenders and Quotes for a single Resource, the complete description is required to ensure validity and for off-market interactions.

Attribute	Type	FIX-Field Name	Meaning	Notes
Tender Detail	Tender Detail	Not defined in FIX	Unit price and quantity for this tender	May be Interval or Stream as permitted
Tender ID	UID	ClOrdId(11)	ID as submitted to Market	Identifies Tender until Market Order ID is assigned by Market
Tender Interval Detail	Tender Interval Detail	Not defined in FIX	Interval, price and quantity for this tender	Used in Interval Tender
Tender Stream Detail	Tender Stream Detail	Not defined in FIX	Stream of consecutive Intervals with Prices and Quantities	Sometime referred to as a Load Curve in Power Markets.
Side	Side-Type	Side (54)	Whether the Tender is to buy or to sell the Product	Buy or Sell
Warrant-Ref	IntInteger	Not defined in FIX	Reference to Warrants as defined in the Market	If used, see comments -Warrants in Tenders, Section 5.3.3.
<i>The following attributes are in Tender Interval Detail or Tender Stream Detail—See Figure 5-3</i>				
Interval	Interval Type	Not defined in FIX	Start Instant for Product delivery together with Duration of delivery. Part of Instrument	While a Market Segment only accepts Tenders and Quotes of a single configured duration, the complete description is required to ensure validity and for off-market interactions.
Price	Long	Price (44)	The unit price for the Product being Tendered	Amount is the product of Price and Quantity. Note that Price is subject to the Price Scale, and Quantity to the Quantity Scale.
Quantity	Long	OrderQty (38)	The quantity of the Product being Tendered	Must meet the Quantity Scale and Round Lot requirements of the Segment. (see Table 13-5)
Stream	CTS Stream Type	Not defined in FIX		Attribute of TenderStreamDetail—see Figure 5-3.

5.3.1 Interval Tenders and Stream Tenders

The most common Tender is the simple ~~or~~-Interval Tender, that is, an offer for a Product [in a single interval](#) beginning at a specific date and time.

In financial markets, a *multi-leg order* ~~submits simultaneous offers to buy or sell~~ [is submitted for securities that are made up of multiple securities, known as a single order all of which must be accepted or rejected together](#) ~~legs. The legs are not traded individually~~. This specification describes a specialized type of multi-leg order for use in in some Market Segments which we term a Stream Tender. A Stream Tender defines a consecutive series of Intervals of identical Duration. The price and quantity tendered must be specified for each Interval.

For example, an industrial customer in a power market may intend to buy power to support a long running

process. In power markets, such a sequence of power use is sometimes referred to as a *load curve*. Such multi-leg orders are expressed using a CtsStream (see 3.2, “CTS Streams: Expressing Time Series”). While the information contained in a Stream Tender can be mapped precisely to a group of Interval Tenders, multi-leg semantics and processing of the related tenders leads to a Stream Tender. ~~For example, All or None would refer to the entire set of Intervals in the Stream Tender.~~ Not all Market Segments permit Stream Tenders; some may require them. A Party submits a Stream Tender, when permitted or required, just as a Party submits an Interval Tender. A Market responds to the submission of a Stream Tender, when permitted or required, just as it responds to an Interval Tender. ~~The submission of a Party's may submit Stream Tender is restricted~~ Tenders only to Market Segments that specifically permit or require them ~~and forbidden in; submission to~~ all other Segments. ~~are forbidden.~~ See Section 13, “Market Structure [Reference Data: Market, Segment, and Session](#) Subscriptions”. Market Segments that support Stream Tenders SHALL also support Stream Quotes (if they support Quotes) and Stream Transactions. See Section 9, “The Negotiation Facet”, for a discussion of Quotes.

5.3.2 Execution Instructions

FIX supports many Execution Instructions, ~~only a profile of which are addressed in this specification while CTS restricts them to a reduced set.~~ ~~Additional Future versions of CTS may incorporate additional Execution Instructions as defined by FIX are conforming but may not be implemented in all Venues. Since more than one may be present (albeit with semantic constraints) into future versions of CTS.~~¹³. These are modeled as a string using single letters for each FIX Execution Instruction Code [separated by a space](#).

For example, the string [HKAHKA](#) indicates the following:

- Cross is forbidden.
- Reinstate on system failure.
- Cancel on trading halt.

Table 5-4 presents a subset of the FIX Execution Instructions permitted for use in [version 1.0 of](#) CTS.

¹³ Segment Reference Data includes which Execution Instructions are supported.

Table 5-4: Trading Instructions

Instruction	FIX Code	Abbreviation	Notes
No cross	A	[NoCross]	Tender is cancelled after any market transition (See 13.4, “Trading Session DataCross-is Forbidden”)
OK to cross	B	[OKToCross]	Cross is Permitted. (See 13.4, “Trading Session DataIn Segments that have Successors, this MAY permit conversion.”)
All or none – AON	G	[AllOrNone]	Ignored in deference to the AllOrNone attribute.
Reinstate on system failure	H	[ReinstateOnSystemFailure]	Mutually exclusive with Q and l (lower case L).
Reinstate on trading halt	J	[ReinstateOnTradingHalt]	Mutually exclusive with K and m.
Cancel on trading halt	K	[CancelOnTradingHalt]	Mutually exclusive with J and m.
Cancel on system failure	Q	[CancelOnSystemFailure]	Mutually exclusive with H and l (lower case L).
Cancel if not best	Z	[CancelIfNotBest]	Cancel if order is not immediately matchable
Ignore price validity checks	c	[IgnorePriceValidityChecks]	
Suspend on system failure	l	[SuspendOnSystemFailure]	Mutually exclusive with H and Q.
Suspend on trading halt	m	[SuspendOnTradingHalt]	Mutually exclusive with J and K.

708 **5.3.3 Use of Warrants in Tenders**

709
710
711
712
713
714
715
716
717
718
719

Warrants increase the specificity of Product (and Instrument). A Buyer who does not specify a Warrant will be satisfied [by](#) Delivery of a Product whether or not it has a Warrant. A Buyer who requests Product with a Warrant will only be satisfied by Delivery of a Product that has that Warrant.

Consider a buyer who wishes to buy a package of coffee beans and a buyer who wishes to buy a package of organic coffee beans. The word “Organic” on the label serves as a Warrant. The first buyer will buy solely on price, and is indifferent to seeing the word “Organic” on the label. The second buyer will choose only from among those packages with the warrant “Organic” on the label.

When a Tender on the Buy side specifies a Warrant, it must be rejected by any Market Segment that does not include that Warrant. A Tender on the Sell side that specifies a Warrant may be accepted by any Segment where the same Resource and Duration are traded. Conversely, a Tender on the Sell side without a Warrant must be rejected by any Segment that specifies a Warrant.

5.4 Contingent Tenders

Contingent Tenders are [FIX permits](#) multiple Tenders submitted in a single message. The FIX List Order bundles multiple Tenders (Orders) with a common instruction ([Contingency Type, 1385](#)) that influences how fulfilling each Tender affects the other Tenders. A Market Segment either forbids or requires the use of Contingent Tenders. Tender Contingency Types are defined [in](#) the [FIX Order List Contingency codeset \(FIX values of the ContingencyType\(1385\)\)](#).

5.4.1 Illustrative Narrative on Contingent Tenders [Non-Normative]

The Contingency Type describes how the other Tenders in the List are affected by the acceptance of any one Tender in the Market. A [common usage for Contingent Tenders is to submit a Party submitting a List of Tenders or Stream Tenders \(Load Curves\), to support a long-running industrial process. The submitter wishes a contract for only one of with atMostOne = True is willing to accept whatever Tender matches the Tenders-Transaction that returns from the Market.](#) In CTS Version 1, the [only in FIX-defined](#) Contingency is OCO or “One Cancels the Other” and it is expressed as a Boolean *atMostOne* in [EiCreateTenderPayload and EiCreateStreamTenderPayload](#).

~~A Party submitting a List with the *atMostOne* = True is willing to accept whatever schedule matches the Transaction that returns from the Market.~~

~~Stream Tenders are a special case. Stream Tenders (Load Curves) support business needs such as acquiring power for a long-running industrial process. The sub-Tenders that compose a Stream Tender are always treated as “All or None”.~~

A Party MAY wish to probe the market to make a more nuanced decision. This may include choosing one of several options. A decision to schedule a long-running process may depend upon being able to acquire a specific load curve over the entire schedule. A party that requires such complex contingent behavior should use the Negotiation Facet (section 9) to obtain [TradableTradeable](#) Quotes, and then make its own choices based on those Quotes.

5.5 Rejecting a Tender

A Market may reject a Tender that violates market rules or which if transacted would violate the market's integrity and other constraints (e.g. liquidity goals). Rejection Reasons include but are not limited to:

- Tender exceeds price limits on the potential transaction.
- Tender exceeds total value limits on the potential transaction.
- Tender violates total quantity limits for this Market Segment.
- Party is not in good standing with the Market.
- Tender violates lot size requirements of the Market Segment.
- Tender violates starting time requirements for instruments in the Market Segment.
- Market Segment is not open.
- Instrument is prior to temporal trading limits for this Market Segment.
- Instrument is past ~~to~~ temporal trading limits for this Market Segment.
- Tender is incomplete or corrupt.
- Referenced Quote not found.
- Referenced Quote has expired.

Details for rejection MAY be included in the *EiResponse* included in the *EiCreatedTenderPayload*.

5.6 Message Payloads for the Tender Facet

Figure 5-4 is a [UML] class diagram for the payloads for the Tender Facet operations. Note that each operation supports a Tender Set, and any set may consist of any number of Tenders, Interval or Stream.

class Tender Facet Payloads

EiCreateTenderPayload

- + atMostOne: boolean
- + counterPartyId: ActorIdType
- + executionInstructions: String
- + marketId: intMarketIdType
- + partyId: ActorIdType
- + requestId: RefIdType
- + segmentId: int
- + tender: EiTenderType [1..n]

EiCreatedTenderPayload

- + counterPartyId: ActorIdType
- + inResponseTo: RefIdType
- + marketOrderId: MarketOrderIdType [0..n]
- + partyId: ActorIdType [1..n]
- + response: EiResponseType [1..n]
- + tenderId: TenderIdType [0..n]

EiCancelTenderPayload

- + counterPartyId: ActorIdType
- + marketOrderIds: MarketOrderIdType [1..n]
- + partyId: ActorIdType
- + requestId: RefId

EiCanceledTenderPayload

- + counterPartyId: ActorIdType
- + eiCanceledResponse: EiCanceledResponseType [0..n]
- + eiResponse: EiResponseType [1..n]
- + inResponseTo: RefIdType
- + partyId: ActorIdType

EiCanceledResponseType

- + cancelReason: CancelReasonType
- + marketOrderId: MarketOrderIdType
- + remainingQuantity: int
- + success: boolean

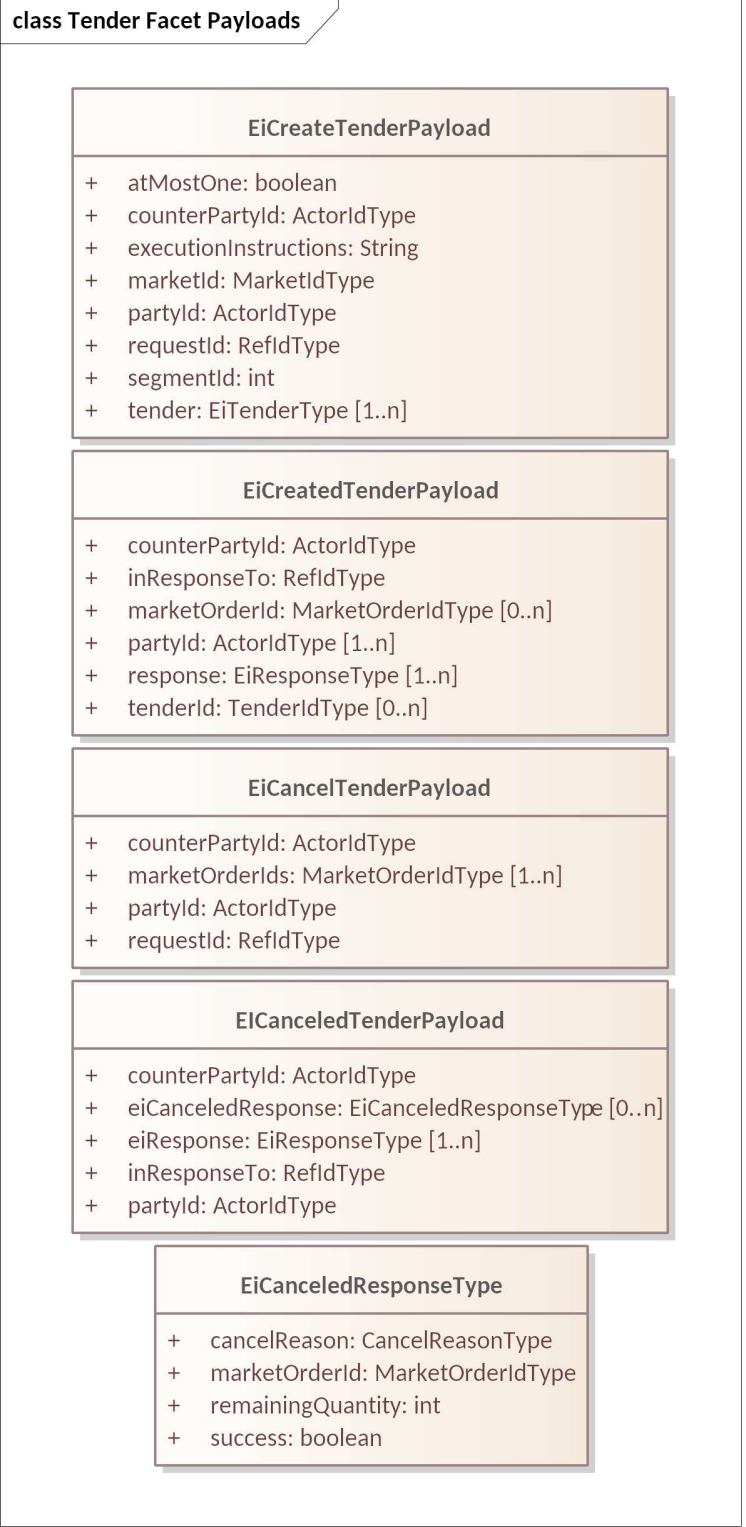


Figure 5-4 UML Class Diagram for Tender Facet Payloads

The Market Order ID is assigned by the Market on receipt of a Tender. The Market makes no assumption that the [Tender ID \(Client Order ID\) \(CLOrderID\(11\)\)](#) submitted as part of the Tender is unique across all Parties in the Market. The Market responds with a Market Order ID for each Tender ID submitted. The

769 submitting Party should record this Market Order ID, as it will be used in any Transactions awarded by the
770 Market, and is required to cancel any Tender.

771 Specific Market Segments may limit all Tender submissions to either Interval Tenders or to Stream
772 Tenders or may accept both. Specific Market Segments may restrict each Tender Set to all Interval
773 Tenders or all Stream Tenders. Specific Market Segments may limit the cardinality of a Tender Set to any
774 count. In the absence of such Segment specification, to support minimal interoperability, Interval Tenders
775 are permitted, Stream Tenders, and the cardinality of each Tender Set is limited to one.

776 See Section 13⁽⁴⁾ for details.

777 The following tables describe the attributes for the Tender Facet Payloads.

Table 5-5 EiCreateTenderPayload Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
At Most One	Boolean	See ExecInst (18ContingencyType (1385))	Used to express alternatives, only one of which is to be effective	See Trading Instructions in Table 5-4. First match cancels other Tenders.
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the Counterparty for which the Tender is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
Execution Instructions	String	ExecInst (18)	Execution Instruction.	Used only for multi leg, and applies to all tenders in multi leg. Execution instructions apply to each Tender in the Listpayload . Multi-leg (Stream) Tenders are always All-or-None.
Market ID	UID Market ID Type	MarketID (1301)	Identifies the Market	Note that in FIX, this is generally a formal identifier (e.g. "NYSE"). If the market is a house, there is no place to look this up. There is always a UID for a Market.
Market Segment ID	Int Integer	MarketSegmentID (1300)	Identifies the Segment processing the Tender, Transaction, or Quote	This should be a unique combination paired with the Market Order ID
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party on whose behalf this Tender is made.	Indicates which Actor proposes the buy or sell side EiCreateTender.
TenderRequest ID	Ref ID Ei Tender Type	OrderID (37)CIOrder ID (11)	An identifier for this Create Tender PayloadTenders requested to be created	One or more Tenders per-. The FIX Protocol makes no assumption that IDs submitted by market participants will actually be complete.
TenderRequest ID	Ei Tender Type Ref ID		Tenders requested to be createdAn identifier for this Create Tender Payload	One or more Tenders per Table 5-2: EiTender Attributes.

Table 5-6 EiCreatedTenderPayload Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between

Attribute	Type	FIX Field Name	Meaning	Notes
				two specific parties, the PartyID of a specific counterparty is used.
In Response To	Ref ID		An identifier for Create Tender Payload to which this is a response	
<u>Market Order ID</u>	<u>UID</u>	<u>OrderID (37)</u>	<u>ID assigned by the Segment or Market.</u>	<u>Used in acknowledgment and in all future market messages.</u>
<u>Party ID</u>	<u>Actor ID</u>		<u>The Actor ID for the Party on whose behalf this Tender is made.</u>	<u>Indicates which Actor proposes the buy or sell side EiCreateTender.</u>
Response	EiResponse Type		Specific error responses	See Section 2.4
The fields below are as defined in the Tender, Table 5-2				
<u>Market Order ID</u>	<u>UID</u>	<u>ORDERID (37)</u>	<u>ID assigned by the Segment or Market.</u>	<u>Used in acknowledgment and in all future market messages</u>
<u>Party ID</u>	<u>Actor ID</u>		<u>The Actor ID for the Party on whose behalf this Tender is made.</u>	<u>Indicates which Actor proposes the buy or sell side EiCreateTender.</u>
<u>Market Order ID</u>	<u>Market Order ID Type</u>	<u>OrderID (37)</u>	<u>The market assigned identifier for the order in the Create Tender Payload to which this is a response</u>	
Tender ID	Tender ID Type	<u>CIOrderID (37)</u>	<u>Identifies the tenders in the Create Tender Payload that was used to submit the Tender to which this is a response</u>	<u>While UIDs should be truly unique, with a mix of technologies and possible faulty implementations in low-end devices, CTS follows the FIX Protocol in assuming that Customer Order is only unique for this Customer today.</u>

Table 5-7 EiCancelTender Payload Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
Counter Party ID	Actor ID	<u>PartyID (448)</u>	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
Party ID	Actor ID	<u>PartyID (448)</u>	Actor ID for the Party that created the Tender	

Request ID	Ref ID	<u>Not in FIX</u>	An identifier for this Cancel Tender Payload	
The fields below are as defined in the Tender, Table 5-2				
Market Order ID	UID	OrderID (37)	ID assigned by the Segment or Market.	Used in acknowledgment and in all future market messages. As defined in Section 5.

Table 5-8 EiCanceledTenderPayload Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party on whose behalf this Tender was made.	Indicates which Actor proposes the buy or sell side EiCreateTender.
In Response To	Ref ID	<u>Not in FIX</u>	An identifier for the Cancel Tender Payload to which this is a response	
Ei Canceled Response	Canceled Response Type	<u>Not in FIX</u>	Detailed response for each tender that was included in the EiCancelTender Payload	See Section 5.5.
EiResponse	EiResponse Type	<u>Not in FIX</u>	Specific error responses	See Section 2.4

6 The Transaction Facet (Execution)

This section presents the Transaction Facet, used by the Market to notify of the creation of Transactions. FIX terms the matching of a Buyer and a Seller as a “Trade” or “Execution”. CTS follows EI (and the term transactive energy) in naming it a Transaction.

In the general case, the Market notifies each Party of the creation of a Transaction when two Tenders match as discovered by the Market’s internal [executionmatching](#) engine. To protect participant privacy, the market MAY use the MarketID as the counterparty to each Party receiving the Notification.

Unlike in financial markets, the market operator must [stillcooperate with relevant system operators to enforce flow](#) limits imposed by physical infrastructure limits. For example, a substation or distribution cable will have physical limits for power transferred during a given Interval. The reasons and mechanisms for such an enforcement are out of scope for CTS.

See Section 9, “*The Negotiation Facet*” for a discussion of Transactions based upon a [TradableTradeable](#) Quote.

All Transactions are committed, that is, they cannot be cancelled or modified under normal market operations. Transactions in aggregate make up the Position. (See Section 7, “*The Position Facet*” for a discussion of Position.) A Party may thereafter choose to sell any [portion](#) or all of its Position in any instrument.

6.1 Messages for the Transaction Facet

A Transaction is created by a Market or Segment (See Section 13) based on some Mechanism internal to the Market.¹⁴ ([See Section 13.1 for what a Party can know of the Mechanism.](#)) When a Market recognizes a potential Transaction, it creates a Transaction ID, and notifies the participating Parties.

Table 6-1: Transaction [Management ServiceFacet](#)

Facet	CTS Initial Message	CTS Response Message	Meaning
TransactionEiTransaction	EiCreateTransaction	EiCreatedTransaction	Create and acknowledge creation of a Transaction; typically initiated by the matching engine of the Market Segment engine.
EiTransaction	EiCreateStreamTransaction	EiCreatedStreamTransaction	Create and acknowledge creation of a Stream Transaction; typically initiated by the Market Segment engine

6.2 Interaction Pattern for the Transaction Facet

~~In Off-Market venues (see Section ,”), the Parties match Tenders themselves, and inform the Market of their agreement. Even in Off-Market venues, the market operator must still enforce limits that affect physical integrity.~~

Figure 6-1 shows the UML sequence diagram [orfor](#) the EiTransaction Facet:.

¹⁴ Some aspects of the market’s mechanism(s) are visible to actors who are trading, generally where the mechanism affects rational bidding strategies. For example, bidding very low in an auction market is reasonable (as you get the clearing price), but bidding very low in an order book market is not (as you may get something like what you offered). See Section 13 and Market Mechanisms.

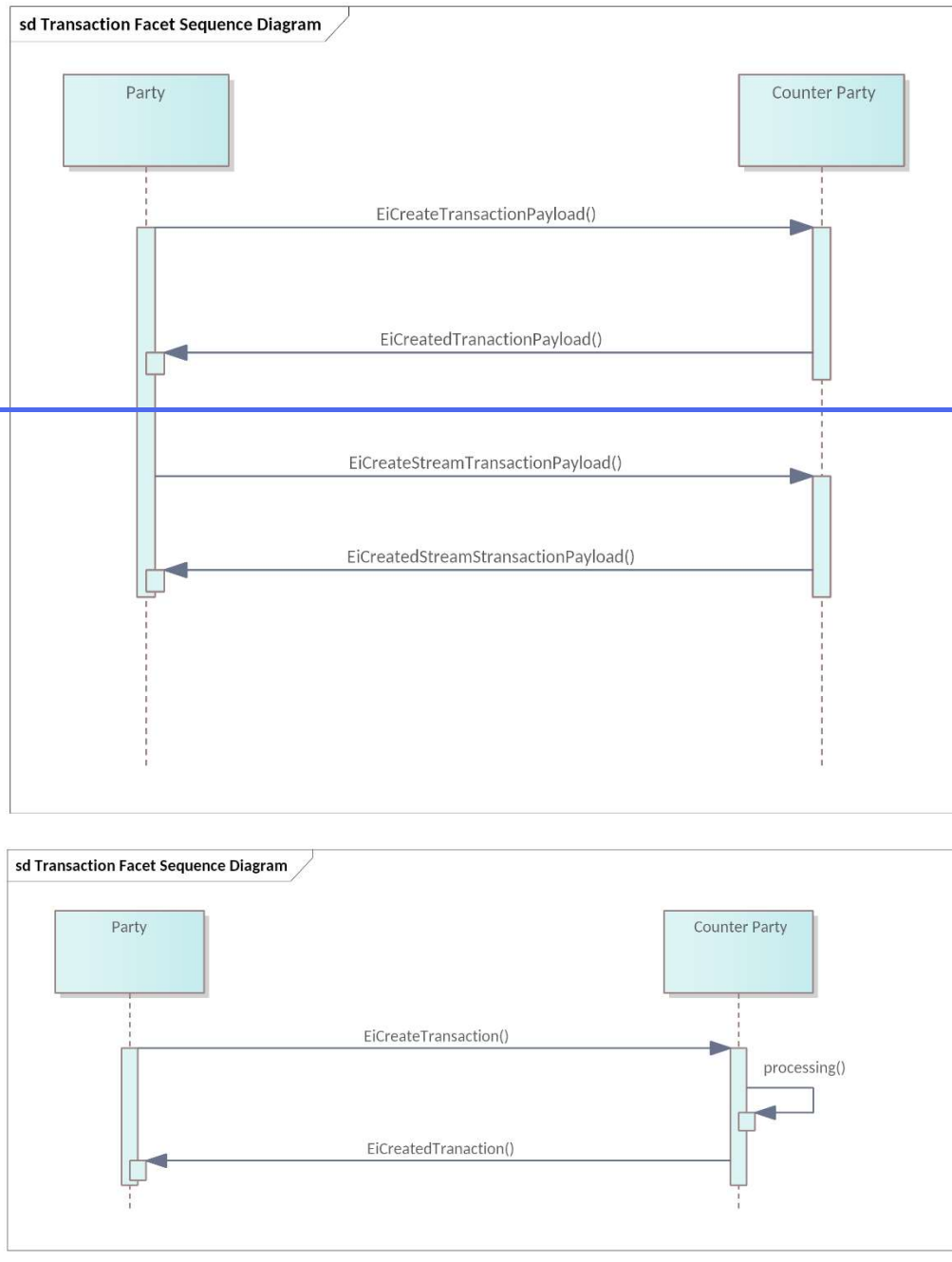


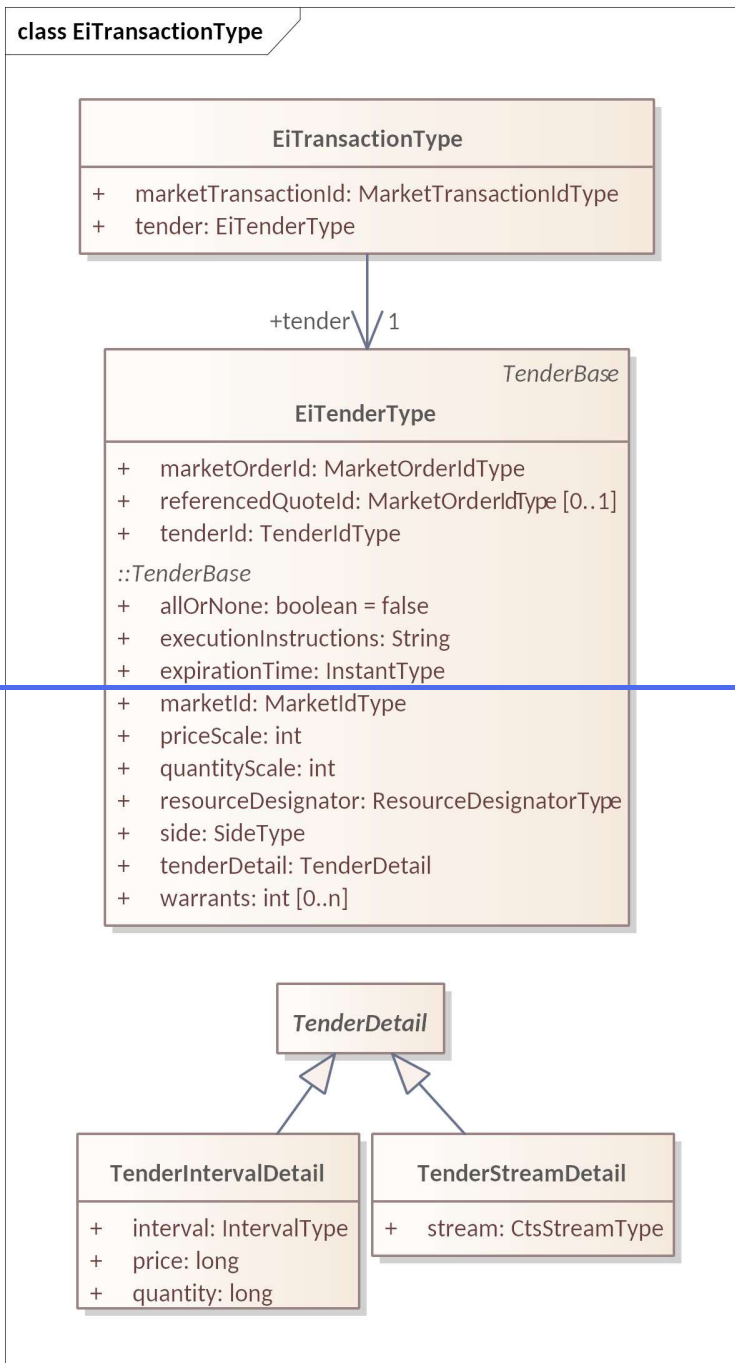
Figure 6-1: UML Sequence Diagram for the EiTransaction Facet

Most Transactions are mediated by a market. The Market matches Tenders, creates a Transaction, and notifies the submitting Parties.

In Off-Market [venues \(see and quote-based Segments \(See Section 13, “”\),](#) the Parties match Quote and Tender, and inform the Market [to create the Transaction](#). Even in Off-Market [venues and quotation-based Segments](#), the market operator must still enforce physical or other limitations. [Interaction patterns for such Segments are defined in Section 9, “The Negotiation Facet”](#).

822 **6.3 Information Model for the Transaction Facet**

823 The EiTransaction object includes the information in the original EiTender, possibly updated to reflect the
824 actual price and quantity rather than the requested price and quantity.



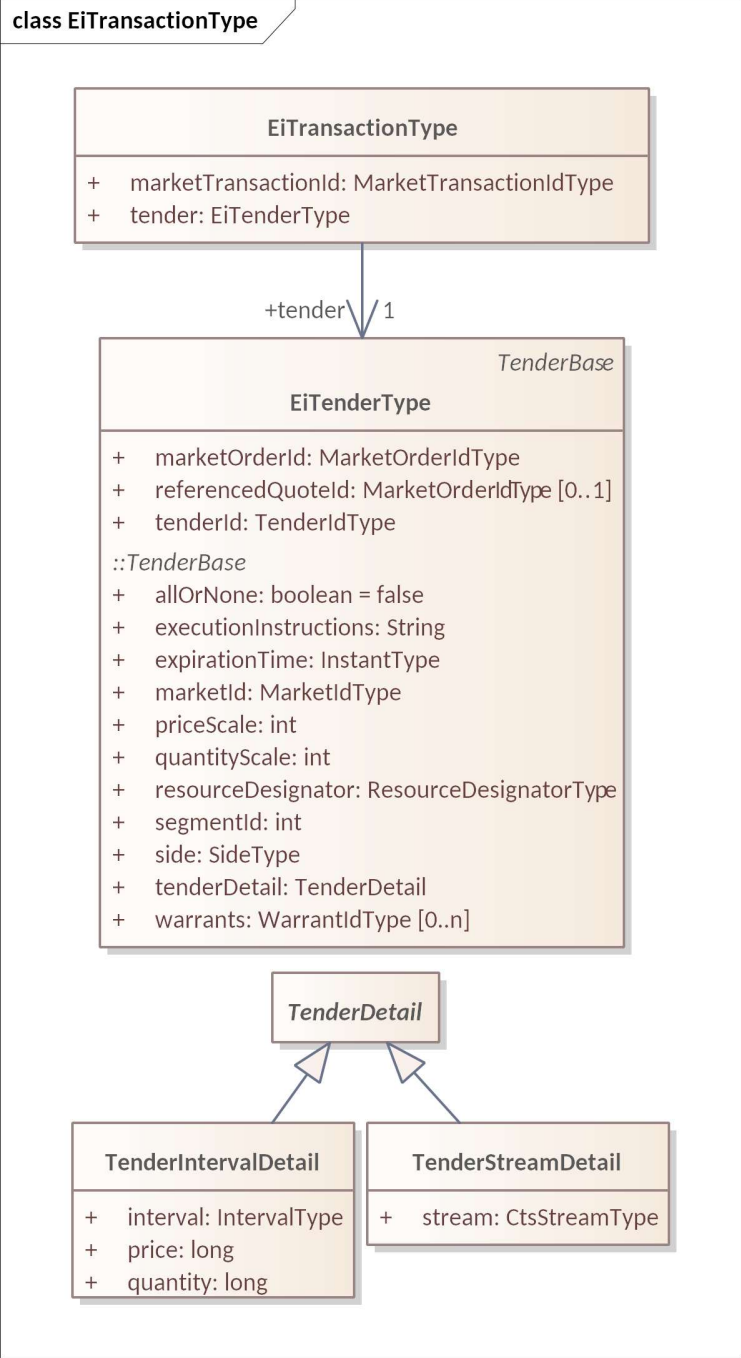


Figure 6-2: UML Class Diagram of EiTransactionType

The attributes of EiTransactionType are shown in Table 6-2.

830

Table 6-2: EiTransaction Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
Market Transaction ID	Market Transaction ID Type	TradeID (1003)	ID Assigned this Transaction (Trade) by the Market (Segment)	Note that Energy Interoperation defines Transaction ID differently than does FIX. This is assigned by the actor that performed the match, typically a market segment.
All other fields attributes are as defined in the Tender Tenderbase. see Figure 5-2				

831

832 **6.4 Payloads for the Transaction Facet**

833 The [UML] class diagram in Figure 6-3 describes the payloads for the EiTransaction facet operations.

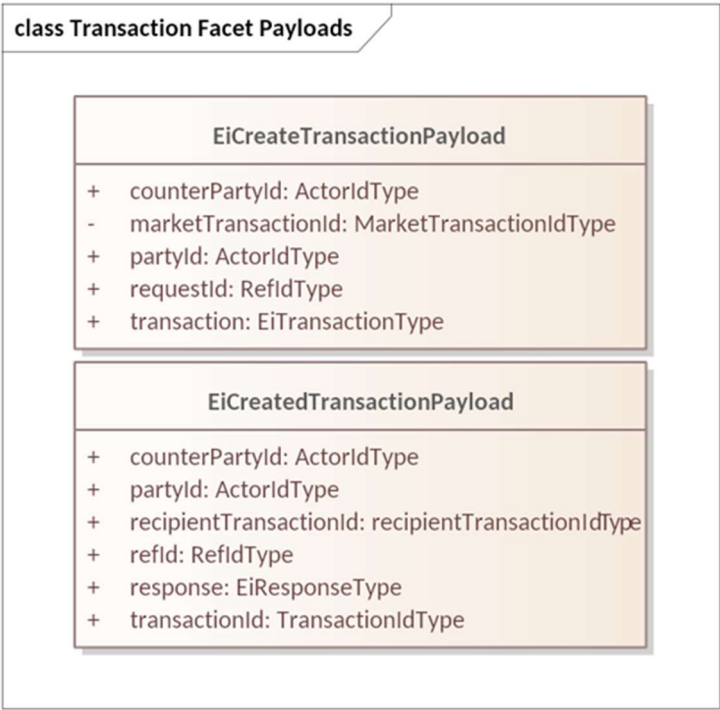
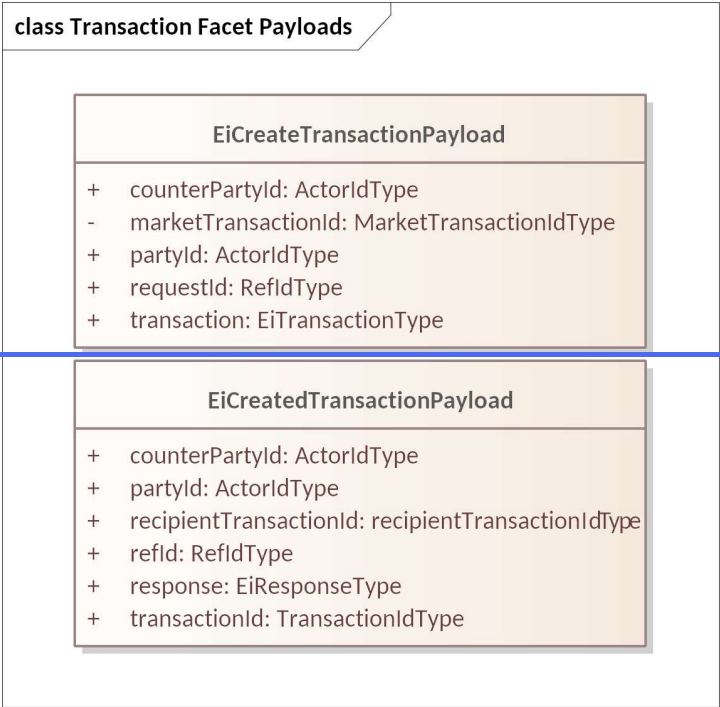


Figure 6-3: UML Class Diagram of EiTransaction Facet Payloads

The following tables list the attributes of the Transaction Facet Payloads.

Transactions are produced by a market or actor that performs matches; the resulting Transaction information is sent to the Parties whose Tender(s) are matched. Note that there is not a one-to-one relationship of Tender to Tender, or Tender to Contract. A Tender to buy one- hundred might match multiple Tenders to sell ten; this results in ~~one Tender resulting in~~ multiple Transactions ~~for one Tender~~.

Each Transaction is created by an interaction between a Tender to buy and a Tender to Sell. The Transaction payloads “echo” to each Tender to the Party that submitted it to become part of the Transaction.

The Tender included as part of a Transaction payload indicates a buy side or a sell side. When the Transaction indicates “buy”, then the PartyID is that of the Buyer. When the Transaction indicates “sell”, then the PartyID is that of the Seller. The CounterpartyID is the other participant in the Transaction.

As in Financial markets often designate a “clearing” or “central” counterparty. Privacy concerns, particularly for transactions involving homes, are one reason for using the PartyID of the central counterparty. Under Some rules, may require revealing the identity of certain Parties must be revealed. For example, the PartyID of a dominant participant such as a distribution serving operator MAY be deemed public information; transactions involving such a designated participant would use the participant’s PartyID in the payload.

When use of a PartyID for the clearing counterparty is required, CTS uses the PartyID of the Market.

Table 6-3 EiCreateTransactionPayload Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	PartyID of the Party on the other “side” from the Tender in the payload.	May be the PartyID of the clearing counterparty.
Market Order Transaction ID	UID	TradeID (1003)OrderID (37)	ID assigned by the Market when processing Tender generating a Trade	Assigned by the Market
Party ID	Actor ID	PartyID (448)	Party ID of the Party on the same “side” of the Tender in the Payload.	Side of the included transaction determines the Party.
Trade ID	String	TrdId (1003)	ID assigned to the trade entity once it is received or matched	Assigned by the Market
Reference ID	String	ExecId (17)	An identifier for this message	
Tender	TenderBase		Price and Quantity for Interval[s] in Transaction	

Table 6-4 EiCreatedTransactionPayload Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	PartyID of the Party on the other “side” from the Tender in the payload.	May be the PartyID of the clearing counterparty.
Market Transaction ID	UID	TradeID (1003)	ID assigned by the Market when generating a Trade	Assigned by the Market

Attribute	Type	FIX Field Name	Meaning	Notes
Party ID	Actor ID	PartyID (448)	Party ID of the Party on the same “side” of the Tender in the Payload.	Side of the included transaction determines the Party.
Trade ID	String	TrdId (1003)	Identifier for the Market’s ID for the received Transaction	
Recipient Transaction ID	Recipient Transaction ID Type	XID	The ID assigned to the received Transaction by the recipient of the associated EiCreateTransaction	
Reference ID	String	ExecId (17)	The Ref ID for the message payload indicating the cleared Transaction	
Response	EiResponse Type		Specific error responses	See Section 2.4

6.5 Comparison of ~~Transactive~~Tender and Transaction Payloads

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

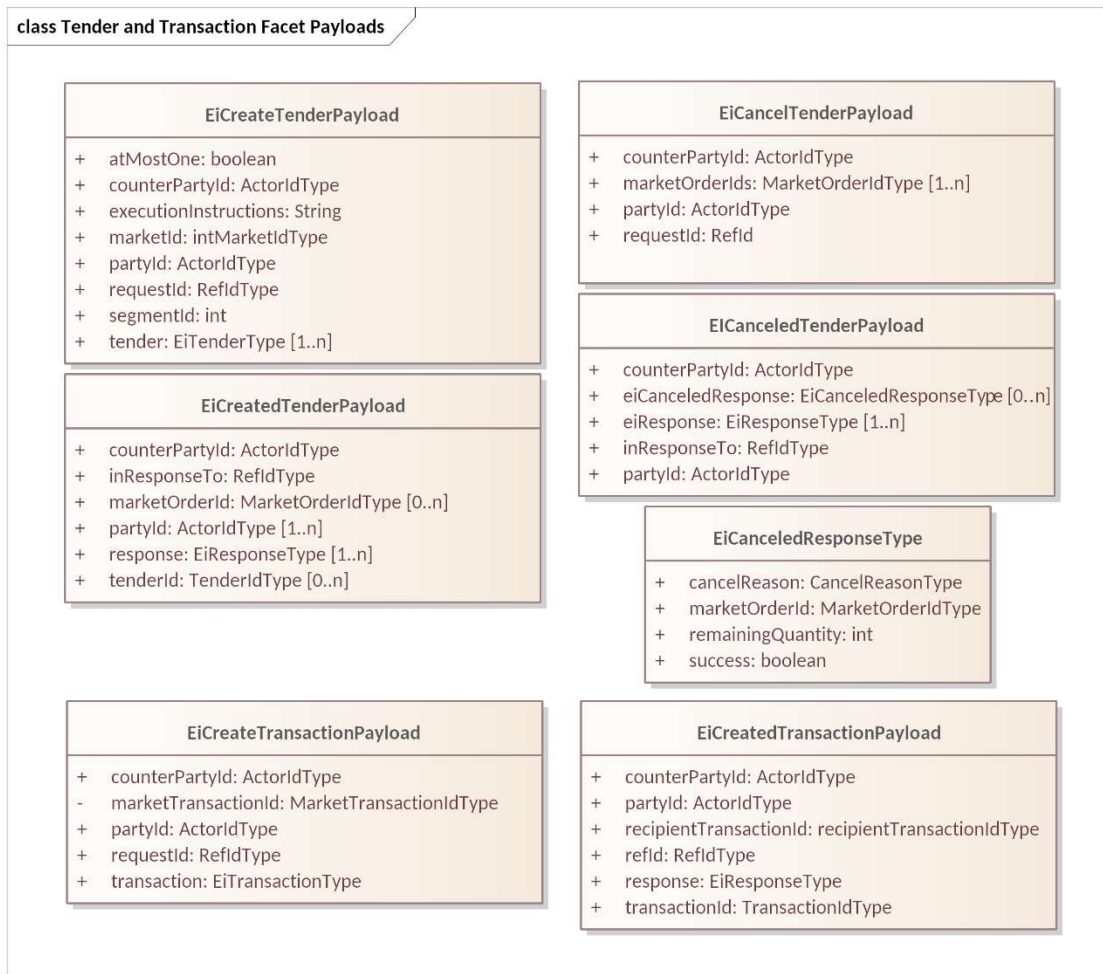


Figure 6-4: UML Diagram comparing Tender and Transaction Facet Payloads

6.6 Off-Market Transactions

While most transactions originate as Tenders submitted to the Market, ~~are then matched by which some mechanism inside~~ the Market ~~matches~~, and result in a Transaction created by the Market, there are use cases for bilateral actions that generate a Transaction that did not come through the market.

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

Off-Market agreements require both parties to report to the Market. The originating Party sends a ~~TradableTradeable~~ Quote to the Market, including the ID of the counterparty. The simplest means is for one Party to publish a targeted Quote (see Section 9, "The Negotiation Facet", below) naming the Counterparty in the Quote. The Counterparty then ~~submits a Tender accepts the Quote by submitting a message~~ referencing ~~that QuoteId the Quote Id~~ and ~~the terms on including a Tender matching the Tender in~~ the Quote.

Some Markets will have specific Market Segments for Off-Market Transactions with specific message patterns. An OTC Market is notable for permitting violations of the Lot Size constraint and of the start time and duration constraints of other market segments. For example, in a Market with a Market Segment with

881 a product of Lot Size 20 kWh and a Duration of one hour, an Off-Market execution could register a
882 transaction of 23 kWh delivered over 27 minutes beginning at 2:48.
883 See Sections 13.1, ~~“”~~ and ~~“”~~ [Market Mechanisms](#).

7 The Position Facet

The Position Facet provides the sum of a Resource transacted for by a Party, positive and negative, for each interval ~~and for each Segment~~, within a possibly larger bounding Interval. For example, a Position ~~may~~ sum ~~up~~ all transactions over the course of a ~~[day]~~. It is typically requested by an auditor or settlement agent (See Section 8 The Delivery Facet) or by a Party to get information about its own position.

~~For example~~, A Party may buy and sell from several Market Segments, perhaps with different Durations. A Party may also transact with specific counterparties in an Over-The-Counter (OTC) market. All of these are part of the Party's position.

In most Resource markets, a Party may also take delivery (see Section 8, The Delivery Facet) which is measured by a meter. But what is the Quantity for this "self-executed" Transaction? This amount ~~can be~~ is calculated by the difference between Position and Delivery and thereby creates Transactions for the used-but-never-bought Resource.

There may be other reasons to track Position. A market rule may require a Party designated as a Market Maker to maintain a Position of a certain quantity. A Party representing a Storage System may have specific rules for Position before a weather event. This specification does not catalog all the uses for Position that a Market or Party may require.

7.1 Introduction

The purpose of the Position Facet is to allow access to the accumulated position for actors supporting specific Roles. A Party's **Position** for a time period is the algebraic sum of committed supply or sales for instruments overlapping that time period. A Party's position for an Instrument is ~~evolved~~computed from ~~an accumulation of~~ trades for that Instrument. In CTS, purchasing a Resource increases the Position, and Selling a Resource reduces the Position.

An Actor may, with appropriate authorization, request positions for other parties. This permits the specification and implementation of an auditor Actor. Roles using the Position Facet include:

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

Position Interactions follow the Streams pattern. A request for position includes a bounding interval. The response reports, at least, the Position for each Interval included within the bounded Interval of the Request.

Table 7-1: Position Facet

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

This is the UML sequence diagram for the Position Facet:

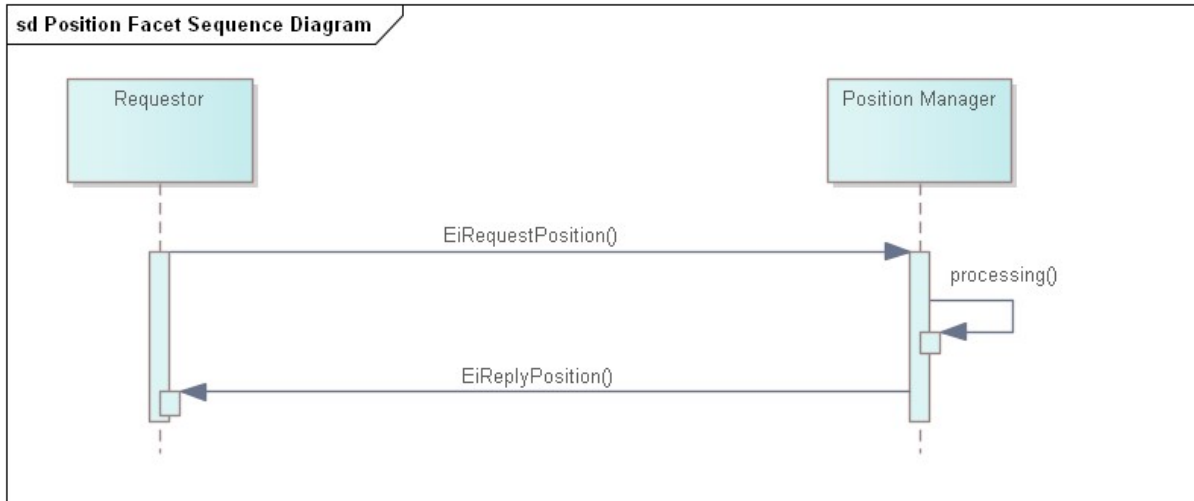


Figure 7-1: UML Sequence Diagram for the Position Facet

7.2 Information Model for the Position Facet

This Facet applies Section 3.3 *"The Bounding Interval Pattern in CTS"*.

For Position, a bounding interval is specified and the position in each interval contained in the closed bounding interval is returned. A Request for Position specifies ~~either a Product or~~ a Resource.

When the Position Request is for a Resource, then the Position is assembled from all Transactions for that Resource. ~~For example, a Transaction When the Position Request is for a Product, then the Position is assembled from all Transactions for that Product. Consider, for example, a Position Request~~ for Green Power, however defined, may only exist between 1:00 PM and 4:00 PM. The Position for Power for the rest of the day may be assembled from several sources, perhaps with different Warrants.

A Position is concerned with the total amount under contract, not the prices. If an Actor has positions in more than one Product, say, in a one-hour Product and in a one-minute Product, then the returned Position SHALL use the shorter Duration.

The attributes are shown in the following section.

7.3 Payloads for the Position Facet

The Position payload is in the format of a CTS Stream, with only a Quantity in the Interval Payload. Position stated against the sum of Transactions in all Segments.

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

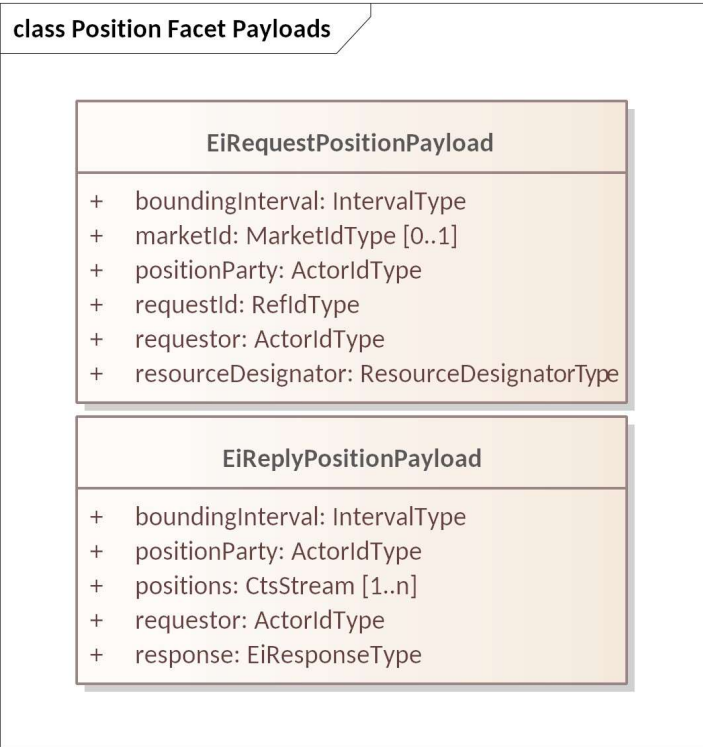


Figure 7-2: UML Class Diagram of Payloads for the Position Facet

Table 7-2: Attributes of Position Facet Payloads

Attribute	Attribute Type	FIX Field	Meaning	Notes
Bounding Interval	Interval Type	Not in FIX	The [closed] time interval for which position information is requested. The first Positions Stream element starts at or after the start of the Bounding Interval. The last Stream element ends at or before the start of the Bounding Interval.	
Position PartyMarket ID	The Party whose position is being requested. Market ID Type	MarketID (1301) Allows a request for another Party's position, with appropriate privacy and security constraints	Identifier of the market of interest. An actor MAY be able to participate in more than one Market See Section 13.	

Attribute	Attribute Type	FIX Field	Meaning	Notes
<u>Resource Designator Position Party</u>	<u>Actor ID</u>	<u>PartyID (448)</u>	The Resource for which <u>Party whose</u> position is being requested. <u>Allows a request for another Party's position, with appropriate privacy and security constraints</u>	Should match the identified Market's Resource Designator
<u>Market IDResource Designator</u>	<u>Identifier of the market of interestResource Designator Type</u>	<u>A Party MAY be able to participate in more than one Market See Section--Not in FIX</u>	<u>The Resource for which Position is being requested. Should match the identified Market's Resource Designator</u>	
<u>Request ID</u>	<u>A reference to this payloadRef ID Type</u>	<u>Not in FIX</u>	<u>A reference to this payload.</u> May be used as a correlation ID	
<u>Requestor</u>	<u>Actor IDThe Party requesting the position.</u>	<u>PartyID (448)</u>	<u>The Party requesting the position.</u> A failure indication will be returned if the Requestor is not authorized to access position information for Position Party.	
<u>Positions</u>	<u>Cts Stream TypeCTS Streams containing the positions for Position Party for each Resource. Positions are signed and may be zero.</u>	<u>Not in FIX</u>	<u>CTS Streams containing the positions for Position Party for each Resource. Positions are signed and may be zero. In CTS, purchasing a Resource increases the Position, and Selling a Resource reduces the Position.</u> Each CtsStream interval that is contained within the Bounding Interval will have a value associated (signed integer). Note that a CtsStream contains a Resource Designator	
<u>Response</u>	<u>EI Response Type</u>	<u>Not in FIX</u>	An EiResponse will indicate failure if Requestor is not authorized to access position information for Position Party for any of the requested intervals.	

The following purposes for requesting Position are system-specific requirements are and out of scope for this specification. Potential uses include:

- 942
- 943
- 944
- 945
- ~~Different systems may support Position requests for different purposes.~~ An Actor may request its own position(s) to recover from failure.
 - A supplier of last resort may compare Positions to Delivery to impute transactions for unpurchased power delivered. (See [Section 8 The Delivery Facet](#).)

8 The Delivery Facet

The CTS Delivery Facet can be considered as the meter telemetry facet. We name it “Delivery” to align with the market focus of this specification, that is, a building takes delivery of power, or a distributed energy Resource (DER) delivers power. A CTS Delivery payload contains a CtsStream that conveys the measured or computed flow of a specific Resource through a particular point on the Resource’s delivery network during a specific Interval.

[This Facet applies Section 3.3 “The Bounding Interval Pattern in CTS”.](#)

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

In most cases, a node that takes delivery of more power or other Resource during an Interval than contracted for must eventually pay for that delivery. For example, An *auditor* (however defined) could sum all positions (See section 7, *The Position Facet*) and compare the result to total Delivery. The Auditor can then impute a transaction for the over-delivery. This may not be a simple “spot price”; if multiple Actors are taking over-delivery, then the last transaction is likely underpriced. Systems that track “actor reputation” may lower the reputation score. These examples explain the potential use of the information delivered by this facet and are not meant to suggest or dictate any particular business process or system model.

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

A CTS Market MAY ~~have~~ assess penalties for Delivery outside certain bounds from the Position—as do many of today’s tariffed markets. Such bounds and penalties are out of scope for CTS. Computation and notification of Penalties is outside of scope.

A request for delivery specifies a Resource, unit of measure, [bounding interval \(Section 3.3\)](#), and a temporal granularity [Duration]. While the unit of measure and temporal granularity need to be within the capabilities of the telemetry node, they need not match any particular Product.

8.1 Interaction Pattern for the Delivery Facet

Table 8-1: Delivery Facet

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

Figure 8-1 is the UML sequence diagram for the Delivery Facet.

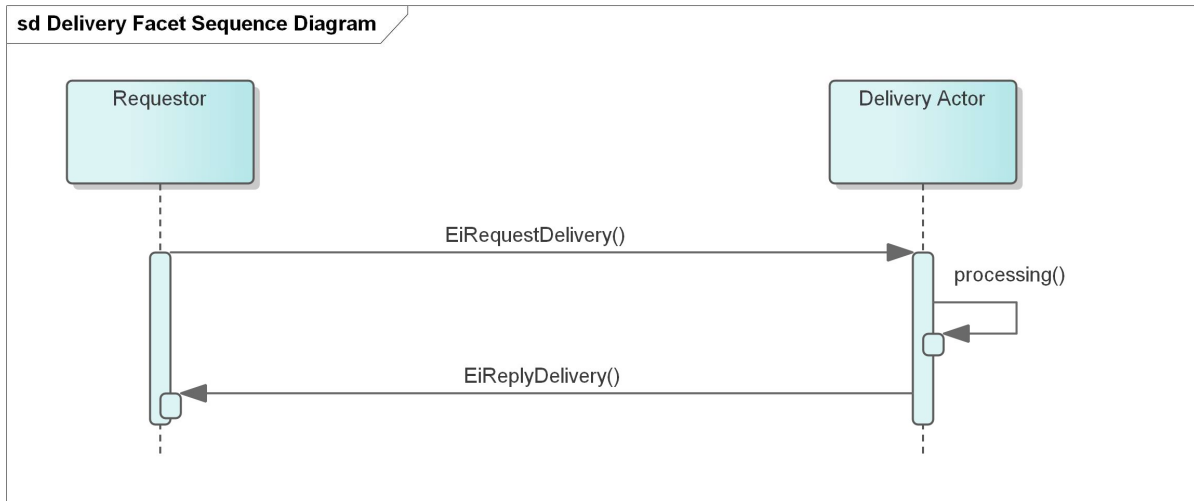


Figure 8-1: UML Sequence Diagram for the Delivery Facet

8.2 Information Model for the Delivery Facet

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

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

The attributes are shown in the following section.

8.3 Payloads for the Delivery Facet

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

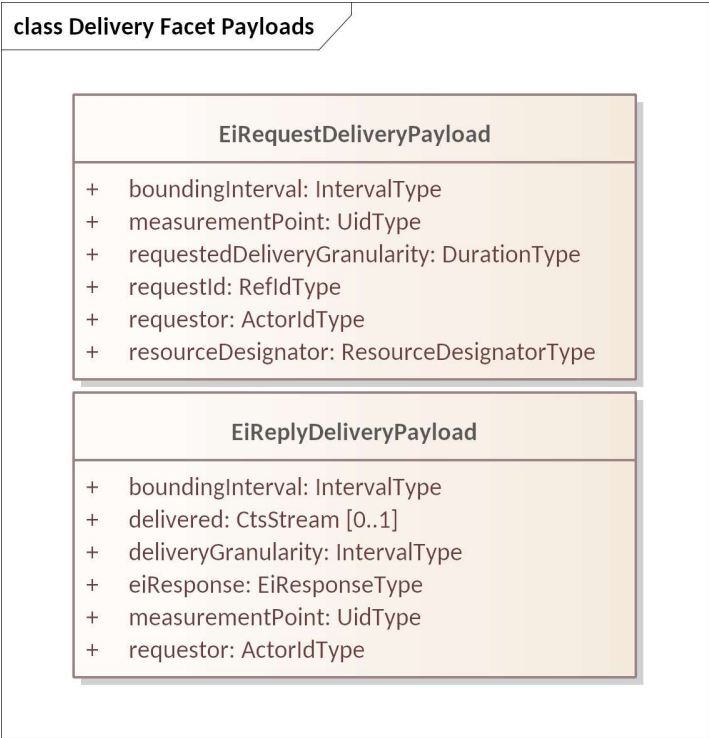


Figure 8-2: UML Class Diagram of Payloads for the Delivery Facet

Table 8-2: Attributes of Delivery Facet Payloads

Attribute	Type	Meaning	Notes
Bounding Interval	Interval Type	The [closed] time interval for which position information is requested.	The first PositionsDelivered Stream element starts at or after the start of the Bounding Interval. The last Stream element ends at or before the startend of the Bounding Interval. See Section.3.3 “The Bounding Interval Pattern in CTS”
Measurement Point	ID	An identification of the Point where measurements are made of the flow of the Resource is measured .	Information should be secured secure in conformance with appropriate privacy and security constraints
Requested Delivery Granularity	Duration Type	The granularity requested for delivery information	Temporal Granularity in reply, as in 1 hour. If empty, determined by capabilities of Measurement Point.
Request ID	Ref ID Type	A reference to this payload	May be used as a correlation ID
Requestor	Actor ID PartyID	The Party requesting the position.	A failure indication will be returned if the Requestor is not must be authorized to access position delivery information for Position this point. May be Party. Addresses the auditor use case or other .

Attribute	Type	Meaning	Notes
Delivered	CtsStream	A CtsStream containing the Quantity delivered in each Interval.	
Response		An EiResponse. Will indicate failure if Requestor is not authorized to access position information for .	If the Requested Delivery Granularity cannot be used, the Response MAY indicate what granularity can be used.

9 The Negotiation Facet

Negotiations are part of what FIX terms Pre-Trade Information. So far, this specification has discussed interactions between a Party and a Market in which some internal process decides how a described an order book market of simple Tender, Transaction is created and Delivery. This section describes instead how discusses more advanced interactions. A Segment-based matching engine, however defined, matches Tenders to Buy and Tenders to Sell and creates Transactions.

With this Section, we introduce the messages used in Segments wherein the Buyer and the Seller find matching Tenders. Negotiations rely on what FIX terms Pre-Trade Information. This section describes Parties come to an agreement to create a Transaction through direct communication. The Parties conduct this conversation is conducted using various types of quotations: requests for quotes, quotes, and quote responses. The Market facilitates the quote process but does not intervene—it acts as a neutral party.

Any Segment may support Negotiation as indicated in the Market Structure (see Section , “”).

9.1 The Negotiation Process (non-normative)

The Negotiation process is inherently flexible. A Transaction may come after many rounds of negotiation, or directly from a response to the first. In essence, a Quote contains a Tender. The matching message accepting the Quote contains a matching Tender. The Parties must inform the Segment of the agreement, and the Segment creates the Transaction memorializing that agreement. The Market may still reject the agreement because of credit limits, or because the Tenders are incompatible, or because a third Party has already lifted the quote, or because the Transaction would exceed operating limits of the system, or for some other reason.

The messages and interactions are determined by the mechanism used in the market Segment. See Section 13 tradable quote. This section describes some potential interactions to clarify the concepts before we define the message types in the following sections.

A Party that wishes to transact some amount of a resource, to find a potential counterparty, or to arrive at an agreement with a specific known counterparty begins a Negotiation by sending a Request for Quotation (RFQ), an Indication of Interest (IOI), or perhaps a unsolicited Quote. In CTS, the distinction between an IOI and Quote is Boolean variable “tradable” which is false for an IOI and True for a Quote.

The initial message may be general, advertised to all participants in a Segment, or targeted, sent to one or more chosen Parties. The IOI is non-tradable and sent to elicit a counteroffer which could be a tradable Quote, another IOI, or an RFQ. A tradable Quote, whether solicited or unsolicited, invites a Tender which will result in a Transaction.

for a discussion of Market Mechanisms and how to select a Segment to trade in. Note: not all Markets must support all Market Mechanisms.

Requests for Quotes and Indicative Quotes (see below for definitions) may be public and if they are, they appear in a Quotes Ticker.

Financial markets assume that the same party, called the Issuer, initiates all quotes in a specific negotiation. The recipient of a quote can accept the quote, if it is tradabletradeable and the terms are agreeable, or reject the quote, i.e., end the negotiation. When a Party accepts (“hits” or “lifts”) a tradabletradeable quote, the Market executes the Transaction—the issuer of the quote cannot back out. The market facilitates the quote process but does not intervene—it acts asA recipient MAY abandon the negotiation, choosing to initiate a neutral party-new negotiation with a new Quote.

CTS negotiations differ from financial practice in that in financial negotiationnegotiations, the instrument never changes. Over the course of a CTS negotiation, the time of delivery may change, which is a change of Instrument.

Negotiation may be used to enable large buyers to plan significant Resource use over time, for example, scheduling a long running industrial process which may also require off-market mechanisms such as labor planning. Such a buyer could submit multiple Requests for Quotes with different schedules, and then select from among the Quotes received in response.

This specification does not require that a Market include any of the scenarios described above. We include them to illustrate how the essential components of Negotiation might fit together in a specific market.

9.1 Negotiation Vocabulary

Negotiations use information elements defined above in TenderBase (5.3), also used in Tenders and Transactions. Note that the term Quote by itself includes both indicative and tradeable Quotes.

Table 9-1: Negotiation Terminology

<u>Term</u>	<u>Purpose</u>	<u>Comment</u>
<u>Request for Quote (RFQ)</u>	<u>A Party submits a Request for Quote to try to find a market in an Instrument or Instruments.</u> <u>A Request for a Quote may be for a time range of Instruments.</u>	<u>May be used pre-opening to elicit tenders, both buy and sell, to determine market opening prices.</u>
<u>Quote</u>	<u>Indicates the price and quantity at which an instrument can be bought or sold. A Quote may be issued in response to an RFQ or it may initiate a negotiation.</u>	<u>The CTS Quote may be either a Bid Quote (buy) or an Ask Quote (sell).</u> <u>The initiator may choose to advertise any Quote to attract potential counterparties by requesting Publication.</u>
<u>Indicative Quote</u>	<u>A Quote that cannot be used to create a commitment leading to a Transaction.</u>	<u>As part of a Negotiation, a Party may submit a counter Quote to ask for a better Quote. Indicative quote(s) may also be issued in response to an RFQ.</u>
<u>Interval Quote</u>	<u>A Quote provided for only a Specific Interval.</u>	<u>Some Segments MAY limit negotiations to Intervals (in TenderBase) only by disallowing Streams. See Section 13.3, “Segment Reference Data”</u>
<u>Stream Quote</u>	<u>Prices and Quantities for a Product in a series of consecutive Instruments submitted as a single Quote.</u>	<u>In energy markets, a stream quote is often referred to as a “Load Curve.”</u>
<u>Tradeable Quote</u>	<u>An offer to buy or sell up to a specific quantity of an Instrument for a specific price.</u>	<u>A Tradeable Quote is registered by the Segment and can be referenced (“lifted”) to initiate a Trade as if it were a Tender.</u>
<u>Quote Response</u>	<u>A response to a an RFQ or Quote. The response may accept the Quote, or counter with another Quote or announce an end to a Negotiation.</u>	<u>Only a Tradeable Quote can be accepted to create a Transaction.</u>
<u>Private Quote</u> <u>Private RFQ</u>	<u>A quote or RFQ sent only to selected Counterparties during a Negotiation.</u>	<u>An implementation may use the Segment to distribute Quotes to Counterparties or it may expect Parties to message Counterparties directly.</u>
<u>Public Quote</u> <u>Public RFQ</u>	<u>A Quote or RFQ published to all subscribers to a Segment’s Quotes Ticker. (See Section 11.5.2)</u>	<u>RFQs, Indicative Quotes, and Tradeable Quotes may be Published.</u>

Term	Purpose	Comment
Issuer	The Issuer is the Party that originates a Quote, whether in response to an RFQ, or unsolicited.	The Issuer must accept a Transaction created in response to a Tradeable Quote.

9.2 Narrative on Negotiation (non-normative)

An extended discussion of use cases and negotiation in markets is in Appendix B.

The Negotiation process is inherently flexible. A Transaction may come after many rounds of negotiation, or directly from a response to the first tradeable quote. This section describes some potential interactions to clarify the concepts before defining message types in the following sections.

A Party that wishes to transact some amount of a resource, to find a potential counterparty, or to arrive at an agreement with a specific known counterparty may start a Negotiation by sending either a Request for Quotation (RFQ), or perhaps a Quote.

Message semantics and sequencing are in this section in the relevant diagrams and tables.

This Facet applies Section 3.3 “The Bounding Interval Pattern in CTSAnd”.

An RFQ uses a Bounded an optional Bounding Interval to indicate what an acceptable IOI or Quote response would be. The possible situations are.

- Consider Party A that wishes to buy 15 kW of power over a two-hour period, sometime within an 8-hour window. This would take the form of an RFQ with an eight-hour Bounding Interval with a specific start time, but with a Stream of two Intervals with a Duration of 1 hour but with no starting time specified.
- Consider instead that Party A further wishes to buy 10 kW of energy over an hour at \$0.05/kWh sometime during the work day. Party A create an RFQ, with a bounding duration of the workday, containing a single unscheduled Interval of one hour containing the Price and the Quantity. Party A and B can send these (1) A Bounding Interval is included.¹⁵ This indicates that a Stream Quote that matches the Bounding Interval is likely to be acceptable. The responder has the option of submitting a counter-quote, that is, initiating a new Quote/Response interaction, perhaps with a different Interval proposing a different Interval¹⁶. (2) A Bounding Interval is not included in the Quote or RFQ—any response must match the included interval or stream.

RFQs and Quotes may be addressed directly to one or more potential counterparties or advertise #published to the entire market. Because it is not tradable, Segment by means of a Ticker. The Market does not need to know about or register the RFQ or Indicative Quote because it cannot lead directly to a trade or Transaction. The recipient may issue a Quote Response to counter or reject the Quote. The recipient may also drop the negotiation and start a new one by issuing a Quote or RFQ. See Section 13.1 “Market Mechanisms. The response is a quote, either indicative (IOI) or tradable (Quote).”, as well as Appendix B for a discussion of interaction patterns in different markets.

Party A may receive one or more offers in response. These become more specific, perhaps two Quotes issued by the same counterparty with different prices at different times. A quote issuer may make a counteroffer by sending a quote proposing different quantities and/or prices. Perhaps the responding Party considers that it will turn on a generator, but only if it can operate the generator at an economic rate for an economic duration. A quote MAY be for only one of the two hours indicated in the original RFQ, leaving the requesting Party to find an acceptable match from among all the offers (quotes) it receives.

¹⁵ This is the same pattern used in Sections 3.3 “The Bounding Interval Pattern in CTS”, 7 (Position Facet) and 8. (Delivery Facet)

¹⁶ Consider a Buyer seeking a Seller willing to run a generator for three hours. The Seller, for economic or operational reasons is unwilling to run the generator for less than 6 hours, and returns a stream quote indicating this longer Interval.

~~The prices may be higher or lower than requested in the original RFQ. Until one party issues a tradable Quote, than all responses are RFQs or IOIs, issued to continue the negotiation.~~

When the ~~respondent~~ Party that has received a Tradeable Quote decides that there is an essential meeting of requirements, ~~that Party submits a tradable Quote, that is a quote that a matching Tender will turn into a Transaction. For a CTS quote to be tradable, the Party informs the Segment, even if it is a private quote and not generally advertised~~ a recipient accepts ("lifts") the Quote; in CTS, the recipient must inform the Market to create the Transaction.

~~To accept a tradable Quote, a Party submits a Tender to the Segment, referencing the Quote ID, and matching the details of the quote. The Segment compares the Tender to the quote, and, if they match, it awards the Transaction without going through the Segment's matching engine. All tradable quotes are treated as if they are marked All-or-None (AON).~~

~~The issuer MUST accept a Tender received in response to a tradable Quote.~~

Negotiations may include Interval Quotes or Stream Quotes, a pattern that matches that of Tenders (See Section 5.3.1, *"Interval Tenders and Stream Tenders."*) ~~A Stream Quote must be matched to a Stream Tender to create a Transaction. A requester that wishes to get a Stream Quote back indicates it in the RFQ or IOI.~~ The stream in an RFQ need not fill the Bounding Interval; an overnight bounding interval of fifteen hours may be seeking any ~~proposal~~ proposed three-hour stream during that interval.

~~There are three scenarios for negotiation below to illustrate the flexibility of Negotiation: (1) Single-provider, (2) over-the-counter, and (3) system recovery.~~

~~1) In certain Market Segments, a single Provider may operate entirely by general advertisement, available to the entire Market. Consider a local distribution electric utility that provides hourly prices for the next day. The providing Party announces day-ahead prices at 9:00 AM each day through 24 indications of interest (IOI) (or a single Stream Indication) to the Segment. Each indication expires at 11:30 AM. Each IOI includes the maximum Power that the Party is willing to sell during each hour of the next day. Each Consumer that wishes to buy on these terms submits RFQs to the issuer for the power they want. The Issuer prepares an actionable quote for each Consumer; each Consumer accepts that Quote by means of a Tender. The Issuer MAY repeat the process, perhaps at different prices, until Transactions for all the power that the Issuer intends to sell are executed.~~

~~2) Parties may choose to use an over-the-counter (OTC) trade because they wish to bypass certain market restrictions. A Segment with an Off-market Venue Type would permit, for example, a Party to buy 87 kWh of Power for a period over 1 hour and 5 minutes beginning 15 minutes after the hour. Parties negotiate as above, come to terms, as above, and the Segment records the Transaction.~~

~~Order Book Segments impose restrictions on Round Lots and Intervals to improve Market Liquidity, that is, the likelihood of a match between a Tender to Buy and a Tender to Sell. If OTC Parties already have made an agreement, then there is no need to improve liquidity. This makes the Durations and Round Lots in Off-market venues indicative rather than prescriptive. 87 kWh is a rough match for Off-Market Segment with a 20 kWh round lot—a gWh is not. In a comparable way, the quote's Duration is a rough match for Segment with an Hour Duration while 3 minutes is not.~~

~~3) Markets commonly project opening prices for Instruments before they open. If a system recovery requires a market re-start, there may be no good information to set opening prices. Prior to a re-start, a Segment may advertise RFQs to buy and to sell. The Operator may use a Segment to probe the potential market in this way several times, perhaps with different prices and quantities, to discover an opening price in which the Resource will be in rough balance. When the market has enough information, then the market opens a Segment for trading, announcing the opening prices.~~

~~This specification does not require that a Market include any of the scenarios described above. We include them to illustrate how the essential components of Negotiation might fit together in a specific system.~~

9.21.1 Negotiation Vocabulary

The Messages use advertisement and negotiation are essentially identical to Tenders. Note that the lowercase quote includes both the IOI and the actionable Quote.

Table --: Negotiation Message Types

Term	Purpose	Comment
Request for Quote (RFQ)	A Party submits a Request for Quote to try to find a market in an Instrument or Instruments. A Request for a Quote may be for a time range of Instruments	May be used pre-opening to elicit tenders, both buy and sell, to determine market opening prices.
Quote	Indicates the price and quantity at which an instrument can be bought or sold. A quote may be issued in response to an RFQ or to a prior quote, or it may start a negotiation.	The CTS Quote may be either a Bid Quote or an Ask Quote. Any Quote or IOI may be either advertised. <i>Note CTS does not support two-sided quotes</i>
Interval Quote	A Quote provided for only a Specific Interval.	Some Segments MAY limit negotiations to Intervals only.
Stream Quote	Prices and Quantities for a Product in a series of consecutive Instruments submitted as a single Quote	In energy markets, a stream curve may be referred to as a "Load Curve."
Tradable Quote	An offer to buy or sell up to a specific quantity of an Instrument for a specific price.	A Tradable Quote is registered by the Segment and can be referenced ("lifted") to initiate a Trade as if it were a Tender.
Indication of Interest (IOI)	A type of quote that does not create a commitment to accept a Transaction.	As part of a Negotiation, an IOI may encourage further Negotiation. May be issued in response to an RFQ.
Private Quote Private RFQ	A quote sent only to potential Counterparties during a Negotiation	An implementation may use the Segment to distribute Quotes to Counterparties or it may expect Parties to message Counterparties directly.
Advertised Quote Advertised RFQ	Setting the Boolean requestPublication when creating a quote requests that the quote be advertised to all Segment participants.	A Segment advertises quotes on the Indication Ticker. (See 11.3) It is undefined what a private Party (other than the Segment) does after receiving a requestPublication
Issuer	The Issuer is the Party that originates a Quote, whether in response to an RFQ or unsolicited.	The Issuer must accept a matching Tender sent in response to a Tradable Quote.

9.3 Messages for the Negotiation Facet

A Request for Quotes (RFQ) is a message describing what is to be quoted, and may be sent to the Segment or to one or more intended counterparties.

A Quote is either spontaneous or in response to an RFQ ~~or prior IOI. Quotes may be Tradable, in which case a Counterparty may respond with a Tender acknowledged by a Transaction, avoiding the market clearing process. A recipient (CounterParty) of a Tradeable Quote may respond with a Quote response.~~

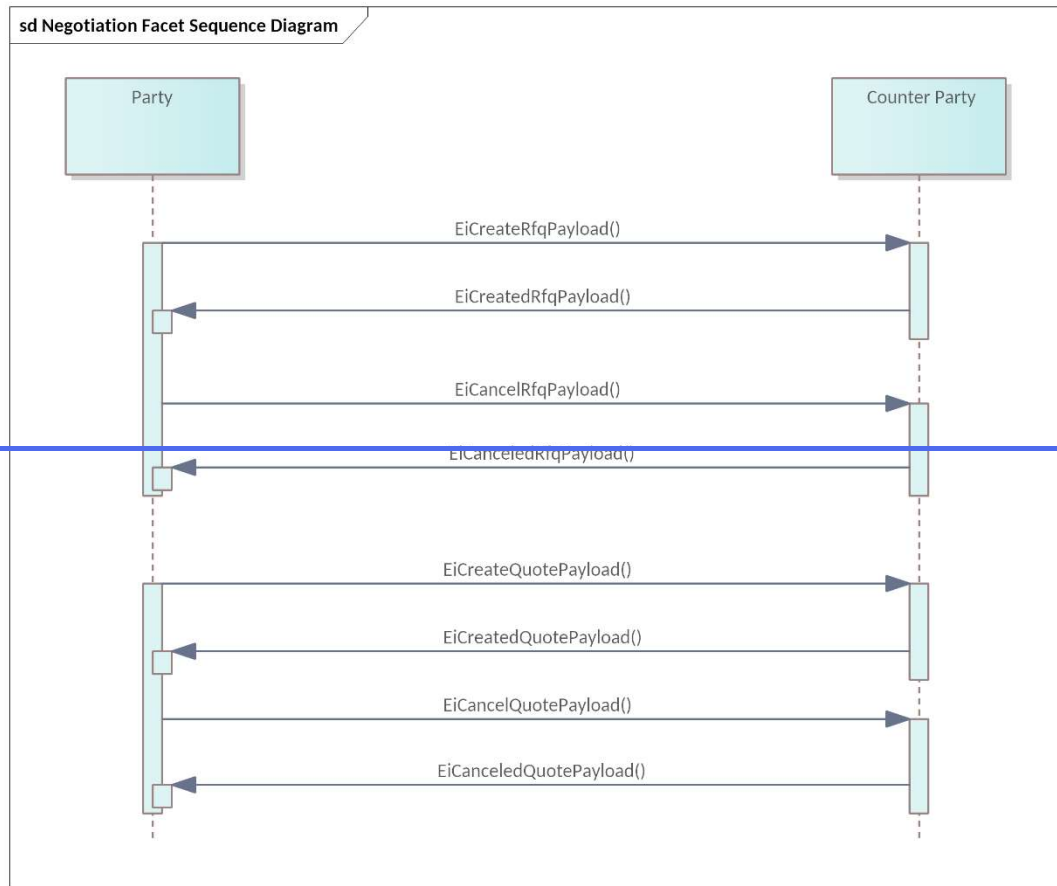
Table 9-2 Messages for the Negotiation Facet

Facet	Request Payload	Response Payload	Notes
Negotiation	EiCreateRfq	EiCreatedRfq	Create and send an RFQ. If The RFQ is directed to be advertised, the Counterparty is the ID of the Market. Otherwise, it goes to the intended CounterPartyPartys or published to the Segment. The sender of EiCreateRfq may request Publication , but has no guarantee that Publication is performedoccurs.
EiCancelRfq		EiCanceledRfq	Indicates that the RFQ Issuer no longer wishes to receive Quotes.
Negotiation	EiCreateQuote	EiCreatedQuote	Create and send a Quote. If the RFQQuote is to be advertisedpublished , the Counterparty is the ID of the Market. Otherwise, it goes to the intended Counterparty. The sender of EiCreateQuote may request publication, but has no guarantee that publication is performed;the Market publishes the Quote.
EiAcceptQuote		EiAcceptedQuote	EiAcceptedQuote returns any errors, and when successful returns a CreatedTransaction payload..
Negotiation	EiCancelQuote	EiCancelledQuote EiCanceledQuote	Cancel RFQ-or a Quote -or . This may be rejected if the Quote was tradeable and had already been lifted by the Counterparty.
EiRejectQuote		EiRejectedQuote	Recipient explicitly rejects referenced Quote.

~~NOTE TO REVIEWERS: Quotes and RFQs could be merged into a single Create an Indication of Interest. Is that an improvement? This Draft goes part way there by including a RequestPublication flag in the respective Create message payloads. This would be subsumed by the implied IOI convergence, and may lead to elimination of the undefined EiDistributeIOI interactions. We request specific comments on this issue.~~

9.4 Interaction Pattern for the Negotiation Facet

~~This is~~These are the UML sequence ~~diagramdiagrams~~ for the Negotiation Facet: Different Market Mechanism Types (MMT) may have shortened Interaction patterns. Due to the complexity of the Quote diagram, we show the RFQ and Quote aspects in separate diagrams.



9.4.1 Interaction Patterns for RFQ and Quote

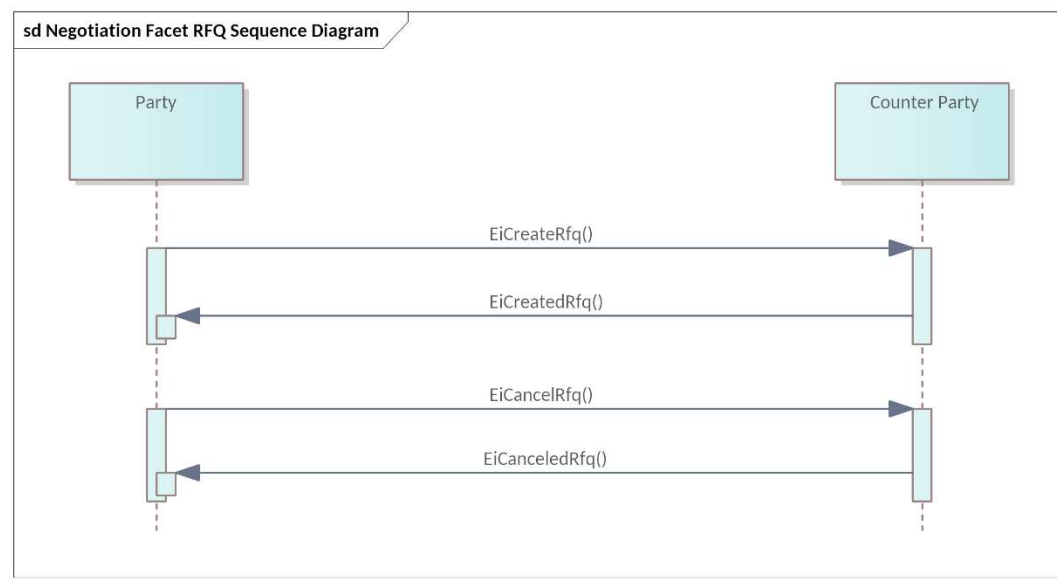
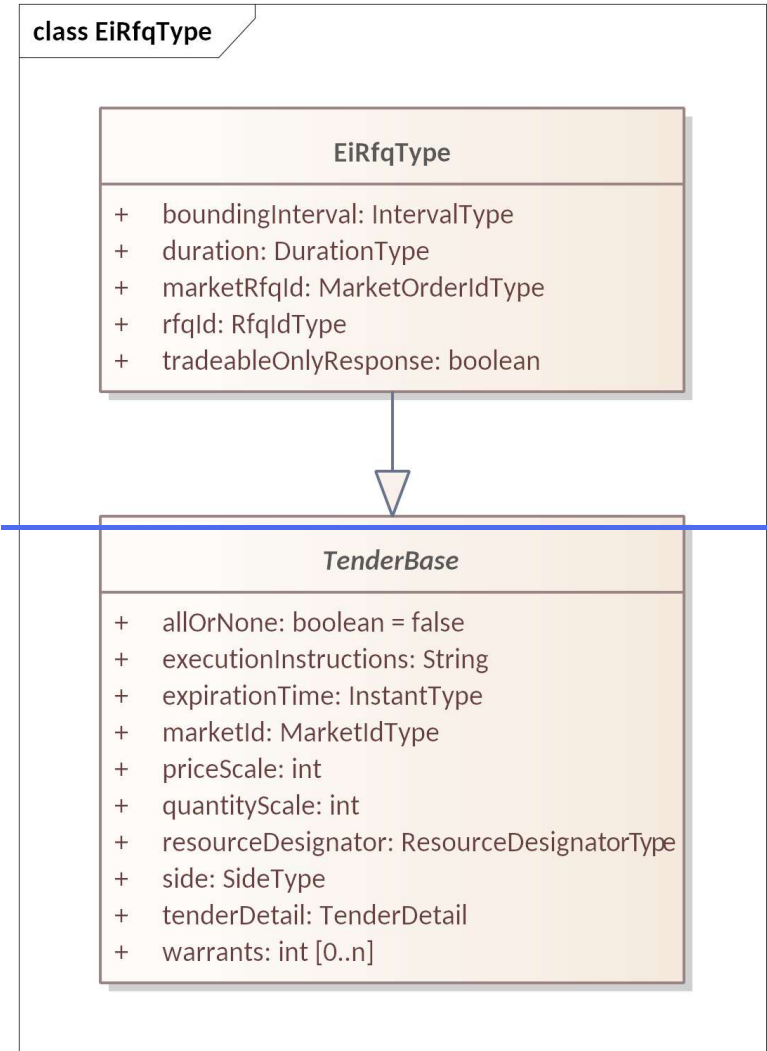


Figure 9-1 UML Sequence Diagram for the Negotiation Facet RFQ (Request for Quote)

9.5 Information Model for RFQ and Quote

1168 The UML Class Diagram for the EiRfqType are shown in.



1169
1170

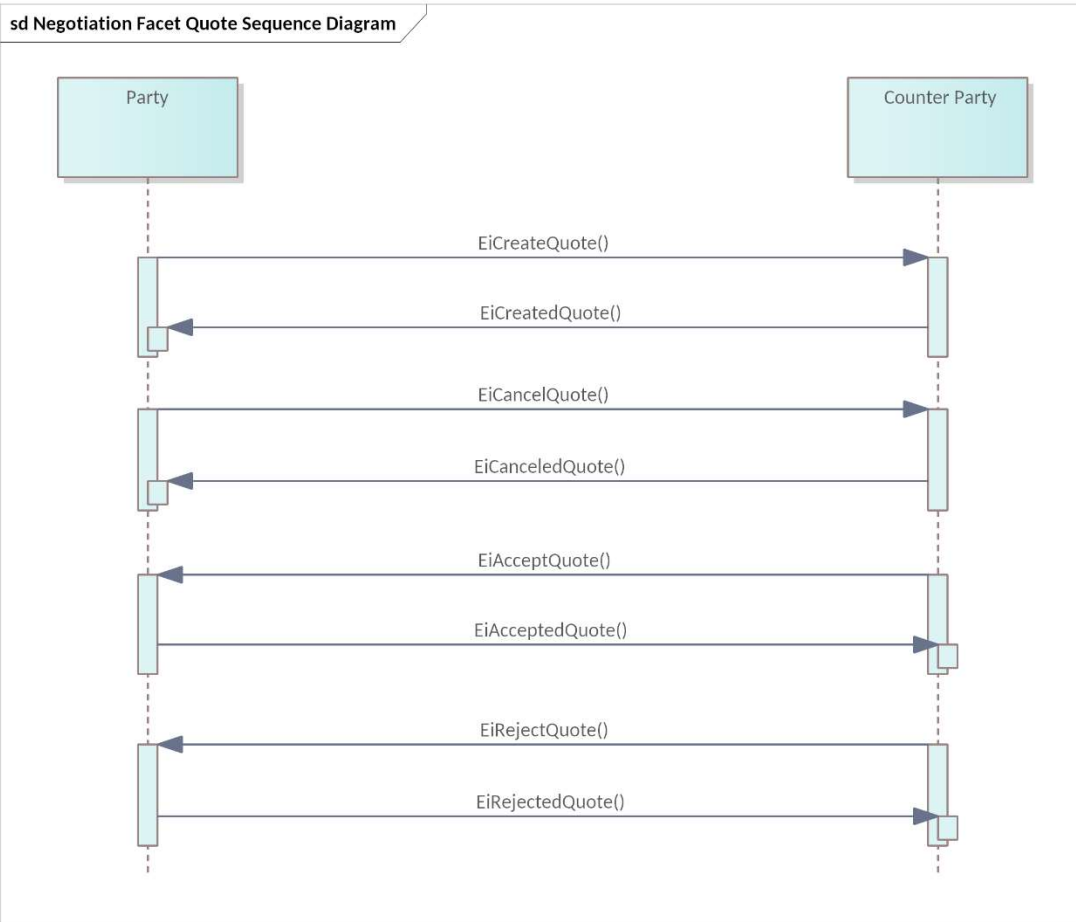


Figure 9-2 UML [ClassSequence](#) Diagram for [EiRfqTypeNegotiation Facet Quote](#)

9.4.2 Creating Transactions from Quotes

A Party receiving a Tradeable Quote MAY respond by submitting an AcceptQuote that references that Quote. The Market registers a Tradeable Quote it receives AS IF it were a Tender, and retains this information until it expires or is cancelled.

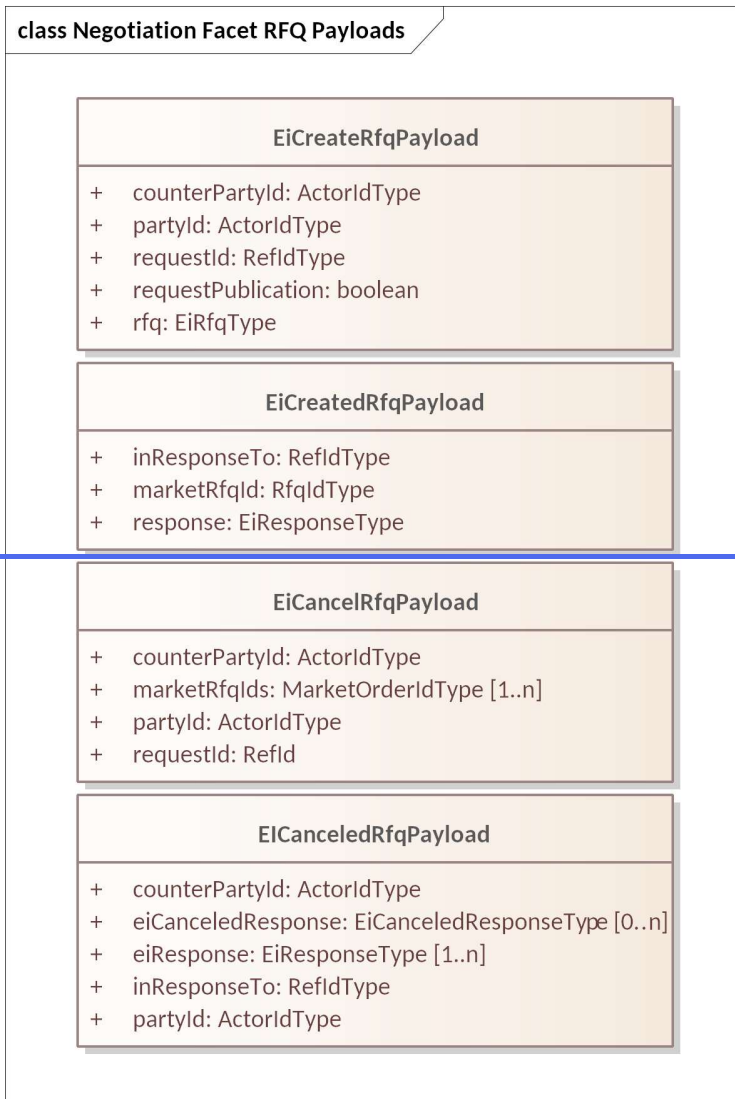
[EiAcceptQuotePayload](#) is a subclass¹⁷ of an [EiCreateTenderPayload](#) that references the ID of the Tradeable Quote being accepted; see Section 6.4 for attributes. Figure 9-10 shows this relationship.

9.4.3 Interaction Pattern for Market-Mediated Negotiation

Certain Quotes and RFQs may have attributes which require Market Segment knowledge. These include Private Quotes, Published Quotes, and Tradeable Quotes

As a partial example of this agency of a Market or Segment see Figure 9-3 “Market Mediated Quote and Responses Sequence Diagram” The UML Class Diagram for the RFQ payloads is shown below. Attributes are in—

¹⁷ In UML formal terminology, [EiAcceptQuotePayload](#)/[EiAcceptedQuotePayload](#) are generalizations respectively of [EiCreateTenderPayload](#) and [EiCreatedTenderPayload](#). Informally, one would say “[EiAcceptQuotePayload](#) is an [EiCreateTenderPayload](#).” All attributes are inherited from the base classes.



1185
1186 [". Similar interaction patterns take place for other market-mediated interactions, e.g., for Tradeable](#)
1187 [Quotes.](#)

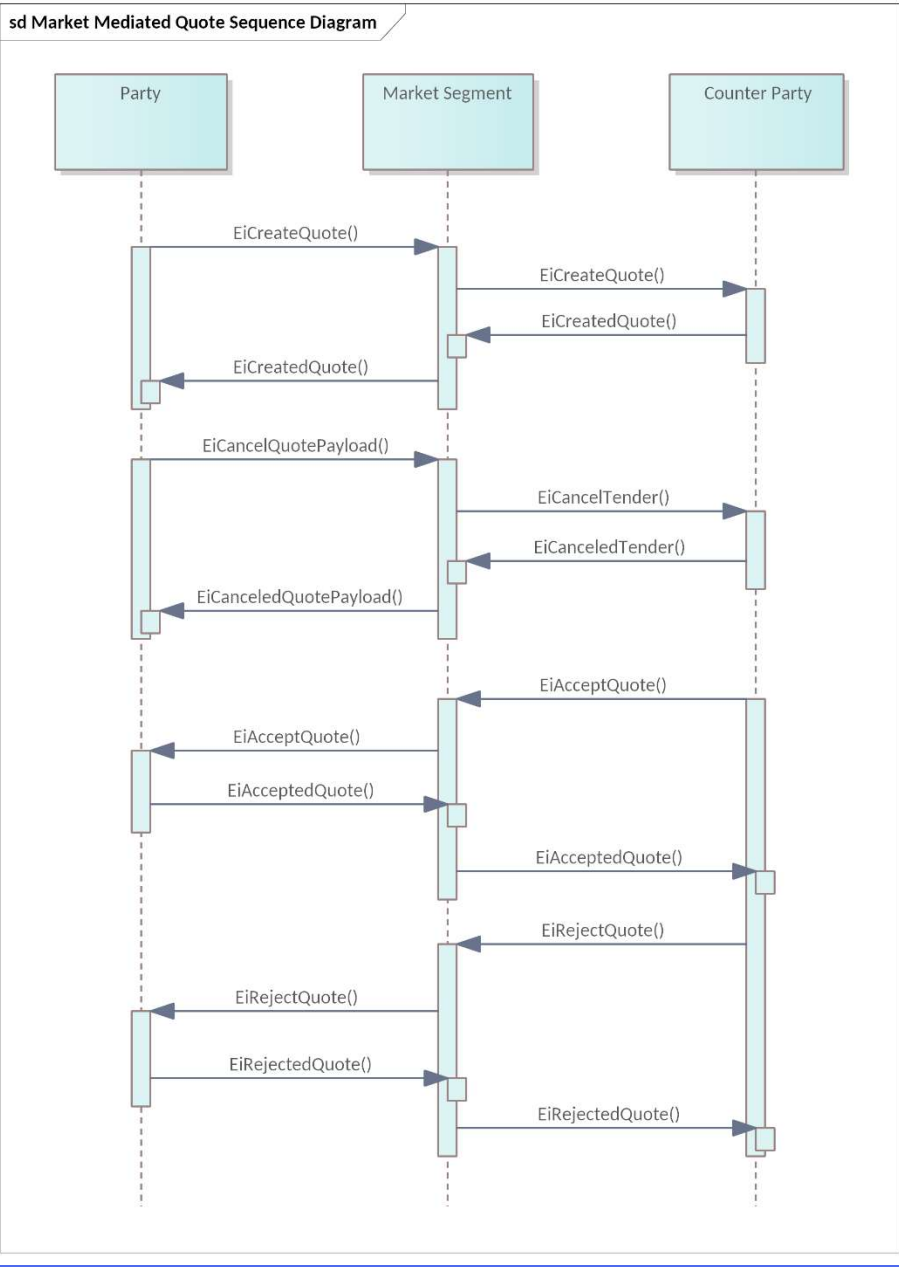


Figure 9-3 [Market Mediated Quote and Responses Sequence Diagram](#)

9.4.4 Interaction Patterns Restricted by Market Mechanism

[Certain Market Mechanisms \(See Section 13.1 "*Market Mechanisms*"\) restrict possible responses to an EiCreateQuote payload.](#)

9.4.4.1 Quote-Driven Markets

Quote-Driven markets are typified by one or more dominant players who provide Quotes and Parties can lift some or all of each Quote. In a Quote-Driven Market Mechanism (MMT_QUOTE_DRIVEN) after receiving and responding to an EiCreateQuote the allowable responses after EiAcceptQuote are shown in the Sequence Diagram in Figure 9-4.

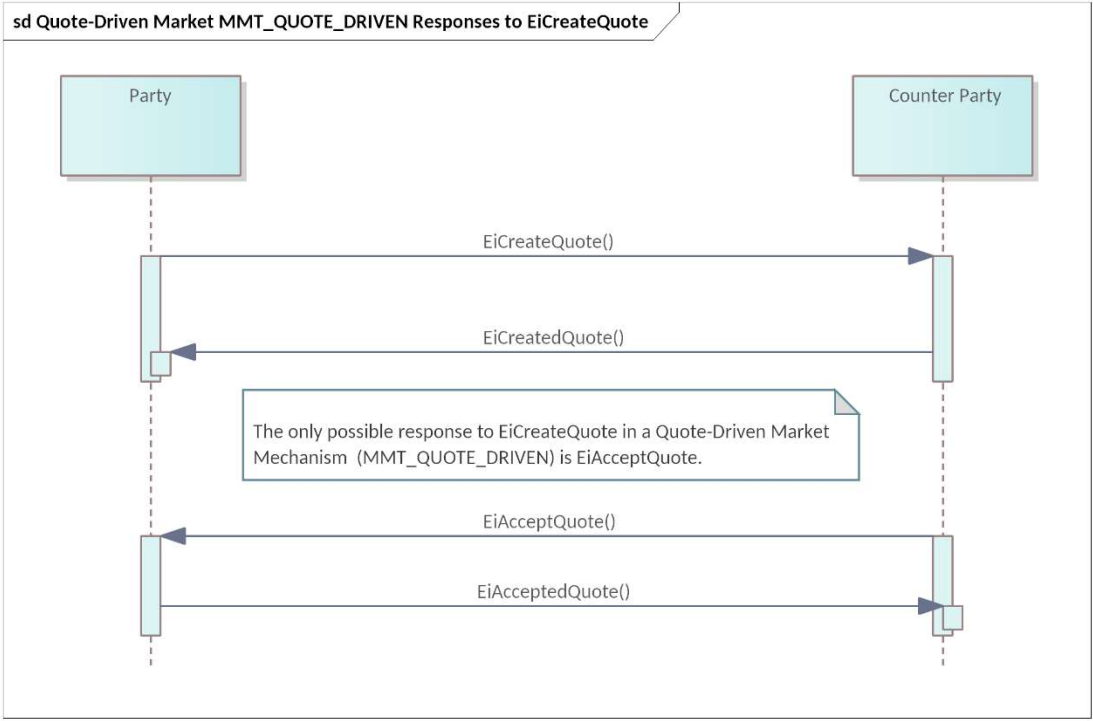


Figure 9-4 Quote-Driven Market (MMT_QUOTE_DRIVEN) Responses to EiCreateQuote—the only possible response is EiAcceptQuote.

9.4.4.2 Request for Quotations Market

In a Request for Quotations Market (RQ) (MMT_RFQ) after receiving an EiCreateQuote and responding with EiCreatedQuote, the allowable responses are below and include EiCreateRfq as shown in the Sequence Diagram in Figure 9-5 “Request for Quotations Market (RQ) MMT_RFQ Responses to EiCreateQuote”.

- EiAcceptQuote
- EiRejectQuote
- EiCreateQuote
- EiCreateRfq

Note that either EiCreateQuote or EiCreateRfq initiates a new negotiation.

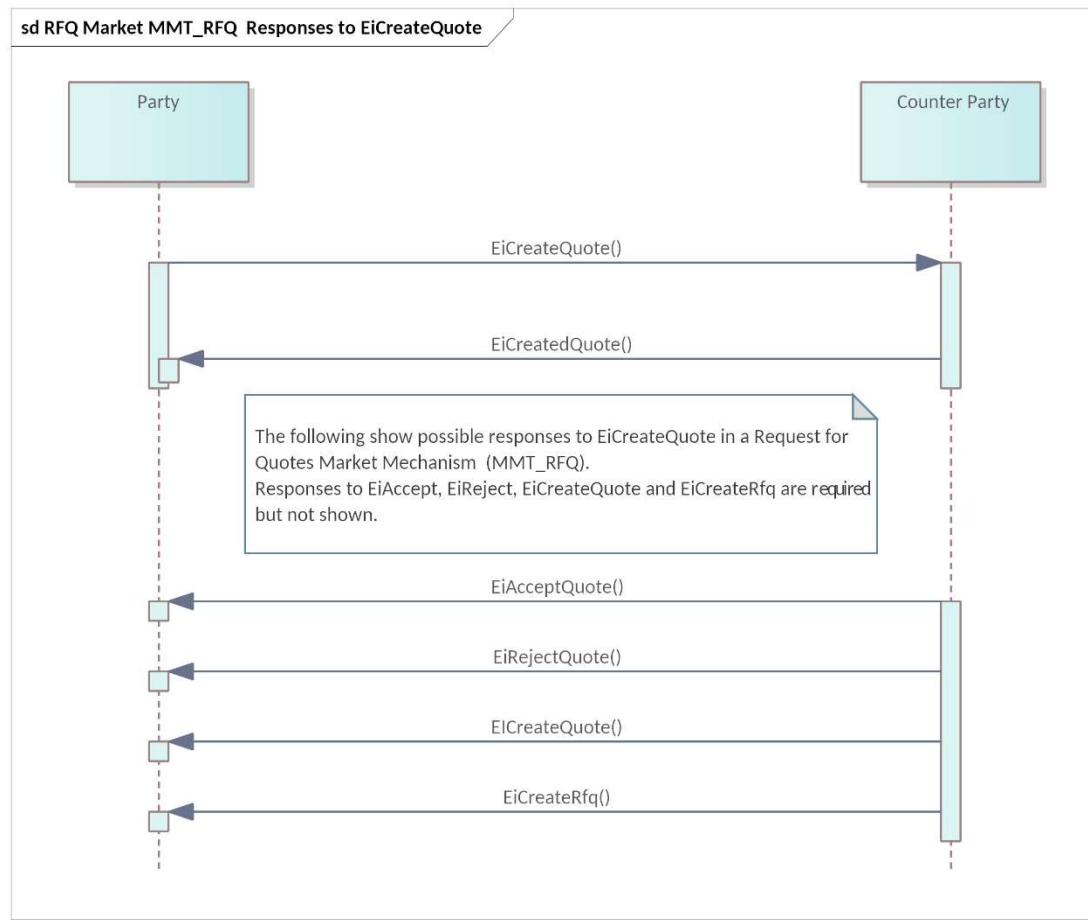


Figure 9-5 Request for Quotations Market (RQ) MMT_RFQ Responses to EiCreateQuote—responses to EiAccept, EiReject, EiCreateQuote and EiCreateRfq are required but not shown.

9.5 Information Model for the Negotiation Facet

The RFQ can be considered as preliminary to a Quote, and so has more optionality.

9.5.1 A Note on Stream Quotes

Some Segments may permit or require Stream Quotes, that is, a single Quote for multiple consecutive Instruments..

Stream Quotes are treated as multi-legged tenders, that is, a Party that wishes to lift a Stream Quote must lift ALL of the Stream Quote. In Power markets, Stream Quotes are used to buy or to sell load

curves, that is, Power in each Interval over a longer time.¹⁸. Stream Quotes are generally used solely in Segments using a Request for Quotation mechanism. See Section 13.1 “Market Mechanisms” and Appendix B for discussions of Market Mechanisms.

Stream quotes and quotes are the same UML class and are related just as Interval Tenders and Stream Tenders are related.

9.5.2 The Request for Quotation

The RFQ can be considered as preliminary to a Quote, and so has more optionality. An RFQ could solicit a quote for 15 minutes of power sometime in an 8-hour window. It could be precise, as in a request for a specific amount of power for a specific duration at a specific time.

The UML Class Diagram for the `EiRfqType` are shown in Figure 9-6.

¹⁸ A large generator may have a ramp up period to reach full power followed by a ramp down period. A long-running industrial process may issue RFQs to find the best time to run a process, and then lift a Quotation to select an operating schedule.

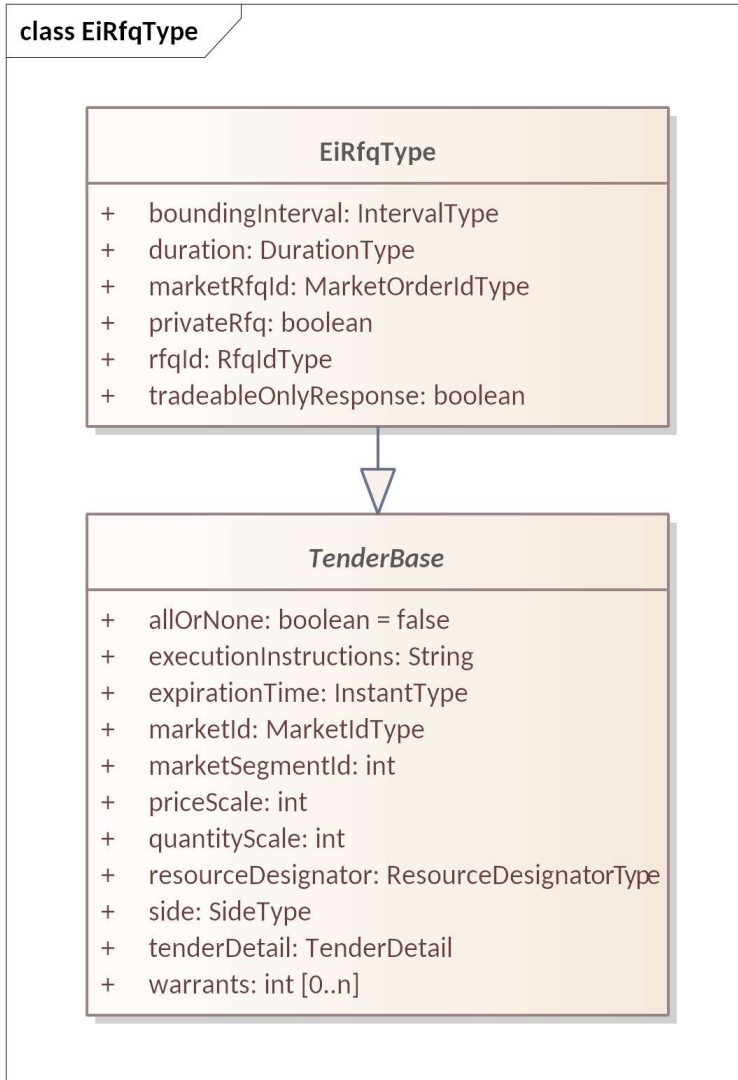


Figure 9-6 UML Class Diagram for EiRfqType

Attributes of EiRfqType are shown in Table 9-3. Attributes inherited from TenderBase are defined in Table 5-3.

For example, consider a Party requesting a Quote for three hours of Power this evening between 4 PM and midnight. The RFQ would have a Bounding Interval 4 PM to 12M and a Duration of 3 hours of the amounts specified in the TenderBase. If the TenderBase references a Segment of 0, the Request goes to all Segments.

Table 9-3 Attributes of EiRfqType

Attribute	Type	FIX Field	Meaning	Notes
Bounding Interval	Interval Type	Not in FIX	The [closed] time interval for which information is requested.	A Quote outside the Interval is permitted. In the example above, this is the “4pm to Midnight”. See Section 3.3 “The Bounding Interval Pattern in CTS”

<u>Attribute</u>	<u>Type</u>	<u>FIX Field</u>	<u>Meaning</u>	<u>Notes</u>
<u>Duration</u>	<u>Duration Type</u>	<u>Not in FIX</u>	<u>The desired duration in the responding responsive Quote.</u>	<u>This is the “3 hours” in the example above. Zero means not specified.</u>
<u>Market RFQ ID</u>	<u>Market Order Id Type</u>	<u>OrderID (37)</u>	<u>ID assigned by the Segment or Market.</u>	<u>Used in acknowledgment and in all future market messages</u>
<u>Private RFQ</u>	<u>Boolean</u>	<u>Related to PrivateQuote (1171)</u>	<u>The RFQ is specific to a single Party.</u>	
<u>RFQ ID</u>	<u>RFQ ID Type</u>	<u>RefOrderID (1080)</u>	<u>ID assigned by originating Party.</u>	<u>See also FIX ClOrdID (11)</u>
<u>Tradeable - Only Response</u>	<u>Boolean</u>	<u>Not in FIX</u>	<u>Indicates whether the initiator wants only Tradeable Quotes in response.</u>	
<i>Attributes for TenderBase are defined in Table 5-3 Tender Base Attributes</i>				

9.5.3 Quotes

As described in Section 5.3 “Information Model for the Tender Facet” and in this Section, EiRfq and EiQuote are subclasses of and inherit from abstract class TenderBase. In Table 9-4, only the first five attributes are part of EiQuoteType; the rest are inherited as shown.

Figure 9-7 is a UML Class Diagram of EiQuoteType showing inherited and included attributes.

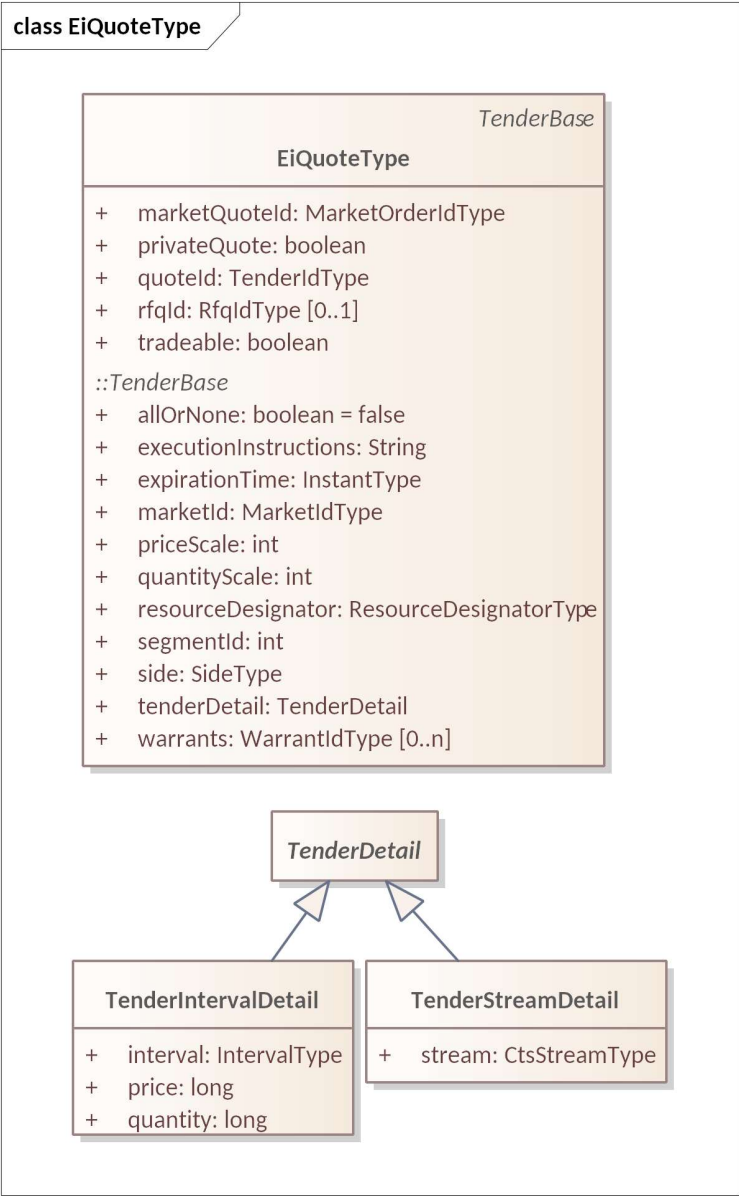


Figure 9-7 UML Class Diagram of EiQuoteType showing inherited attributes.

Table 9-4 Attributes of EiQuoteType

Attribute	Type	FIX Field	Meaning	Notes
Market Quote ID	Market Order ID Type	OrderID (37)	ID assigned by the Segment or Market.	Used in acknowledgment and in all future market messages.
Private Quote	Boolean	Private Quote (1171)	Quote is available specified counterparty only.	Quote is not available to the Segment.

Attribute	Type	FIX Field	Meaning	Notes
Quote ID	Tender ID Type	QuoteID (117)	ID as submitted by Quote originator/issuer	Used in off-market negotiation
RFQ ID	RFQ ID Type	QuoteReqID (131)	Market assigned ID of the RFQ to which this quote is responding	Referenced by a Quote responding to RFQ. Optional.
Tradeable	Boolean	QuoteType (537)	Indicates whether the Quote is tradeable or not	If true, the quote is tradeable. If false the quote is not tradeable, which is by definition an Indicative Quote, consistent with FIX terminology.
All Other Attributes are as defined in TenderBase Table 5-3				

The Quote, RFQ, and Tender share common information using TenderBase. See Figure 5-2 “UML Class Diagram Showing Commonality between Tender, Quote, and RFQ”.

9.6 Messages for the Negotiation Facet

9.6.1 RFQ Messages

The UML Class Diagram for the RFQ payloads is shown in Figure 9-8 below.

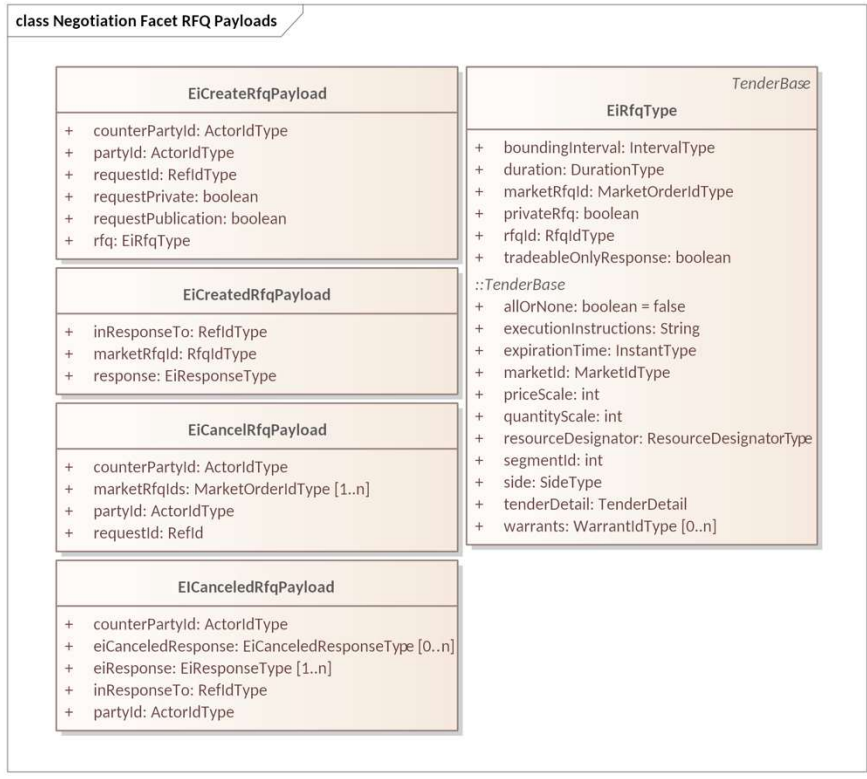


Figure 9-8 UML Class Diagram Showing Negotiation Facet RFQ Payloads

1266 [The attributes of EiCreatedRFQ response payloads are in Table 9-5.](#)

Table 9-5: EiCreateRFQ Payload Attributes

Attribute	Type	Fix Field Name	Meaning	Notes
Counter Party IDs	Actor ID		The Party IDs for the CounterParties for which the RFQ is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counter party is used.
Party ID	Actor ID		The Actor ID for the Party requesting the Quote.	Indicates which Actor proposes the buy or sell side EiCreateTender .
Attribute	Type	FIX Field	Meaning	Notes
<u>Counter Party ID</u>	<u>Actor ID Type</u>	<u>PartyID (448)</u>	The Party IDs for the CounterParties for which the RFQ is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counter-party is used.
<u>Party ID</u>	<u>Actor ID Type</u>	<u>PartyID (448)</u>	The Actor ID for the Party requesting the Quote.	Indicates which Actor proposes the buy or sell side <u>EiCreateTender</u> .
Request ID	RefID Type		Reference to this message payload	
<u>RFQ Request Private</u>	<u>BooleanEiRfqType</u>	<u>Private Quote (1171)</u>	The sender requests that RFQ to be Private only to specified Counter Party or Parties. The RFQ transmitted.	<u>Unlike EiCreateTender, only one RFQ is created per message payload. FIX has Public as an antonym of Private; due to privacy related market rules, CTS separates the concepts and clarifies that it is a request.</u>
Request Publication	Boolean	<u>Private Quote (1171)</u>	<u>Publication of the RFQ is requested.</u>	The sender of <u>an EiCreateRfq</u> may request Payload requests publication by setting on the Quotes Ticker if available. <u>This is a request and may not take place.</u> <u>See also Request Publication to true, but has no guarantee that publication is performed Private.</u>
The fields below are as defined in EiRfqType				

RFQ Id	QuoteIdType EiRfqType		The RFQ transmitted. Assigned by the sender of EiCreateRfqPayload	Creator's ID for the RFQ. An RFQ may use a Stream to indicate what sort of Stream Quote it is looking for or even multiple Streams to indicate an interest in transactions over time.
Bounding Interval	IntervalType		A closed interval which encloses all Quotes requested.	Fields of EiRfqType are shown in Figure 9-6. See Sections (Position and Delivery Facets). Alignment of instruments is a characteristic of the Segment, and Table 9-3
Duration	DurationType		Resource and Duration determine product.	An explicit duration rather than a database lookup simplifies consumption and generation planning.
All Other fields are as defined in the Tender, Table 5-2				

The attributes for EiCancelRfq and EiCanceledRfq are in Table 9-6.

Table 9-6 EiCreateRfq, EiCancelRfq and EiCanceledRfq Payload Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the Counterparty for which the Tender is created.	Unlike Tenders, Negotiations are typically directed to a specific Party. If the Quote or RFQ is published, the Counterparty is the Party ID of the Segment.
Market Request for Quote ID	RfqIdType Market Order Id	OrderID (37) RFQId	Market-assigned ID for this Request for Quote	Market Assigned as it in for parallel with Tenders. Used for Cancel request.
Response Party ID	EiResponseType Actor ID	PartyID (448)	Standard response object. The Actor ID for the Party on whose behalf this RFQ is made.	Indicates which Actor proposes the buy or sell side RFQ.
Request ID	Ref ID		An identifier for this Cancel RFQ Payload	
In Ei Canceled Response To	RefIdType EiCanceledResponse Type		ID for the corresponding EiCreateRfq Payload. Optional Detailed response for each RFQ for which cancellation was requested	

1273

Table-- EiCancelRfq and EiCanceledRfq Payload Attributes

Attribute	Type	FIX Field Name	Meaning	Notes
Counter Party ID	Actor ID		The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
Party ID	Actor ID		The Actor ID for the Party on whose behalf this Tender is made.	Indicates which Actor proposes the buy or sell side EiCreateTender.
Request ID	Ref ID		An identifier for this Cancel RFQ Payload	
Market Request for Quote ID	RfqIdType	OrderID (37)	Market assigned ID for the subject Request for Quote	Market Assigned in parallel with Tenders.
Quote Id	Quote ID Type		The Quote ID included in the CreateQuoteTender Payload or the Tenders within an CreateQuotePayload	
Canceled Response	EiCanceled Response Type		Optional Detailed response for each RFQ for which cancellation was requested	
In Response To	RefIdType		The EiCancelRfqPayload that is responded to in the Canceled Payload	

1274

1275 **9.6.2 Information Model Quote Messages**

1276 The UML Class Diagram for the Quote payloads is shown below. Attributes are in tables starting with
1277 Table 9-7.

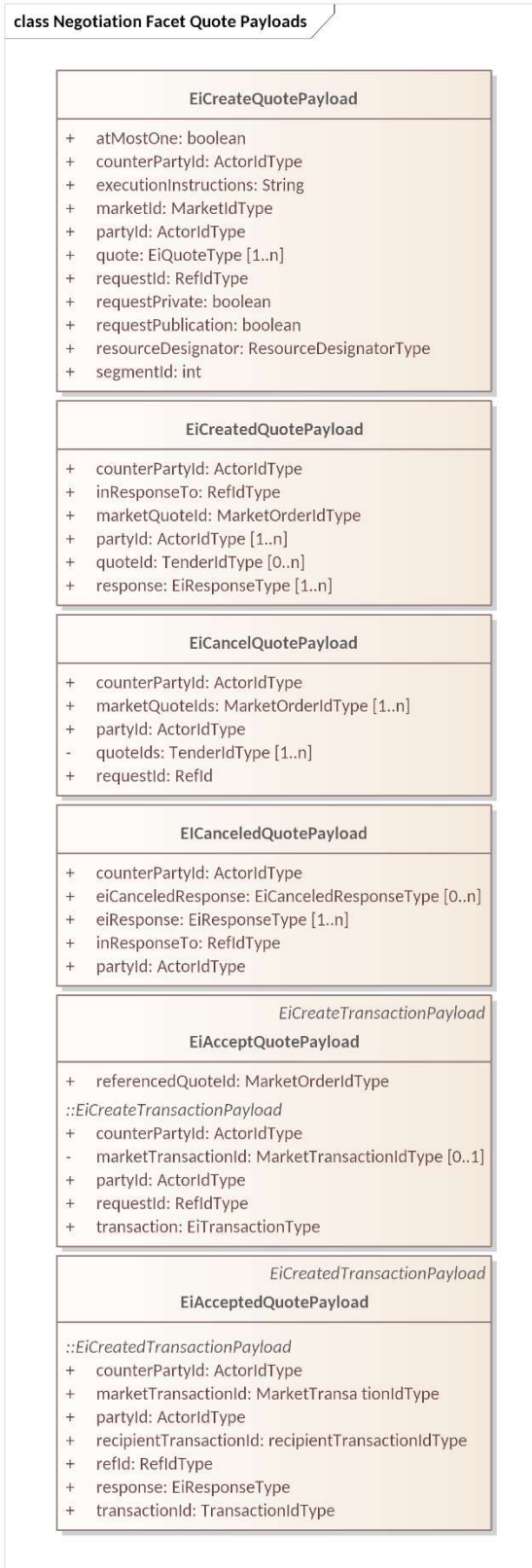


Figure 9-9 Negotiation Facet Quote Payloads

As described in Section 9.1, EiQuote and EiTender are subclasses of abstract class TenderBase. In the following table, only the first three attributes are part of EiQuoteType; the rest are inherited. See 9.2 for details. See 9.3 for the inheritance and relationship between EiTenderType and EiQuoteType.

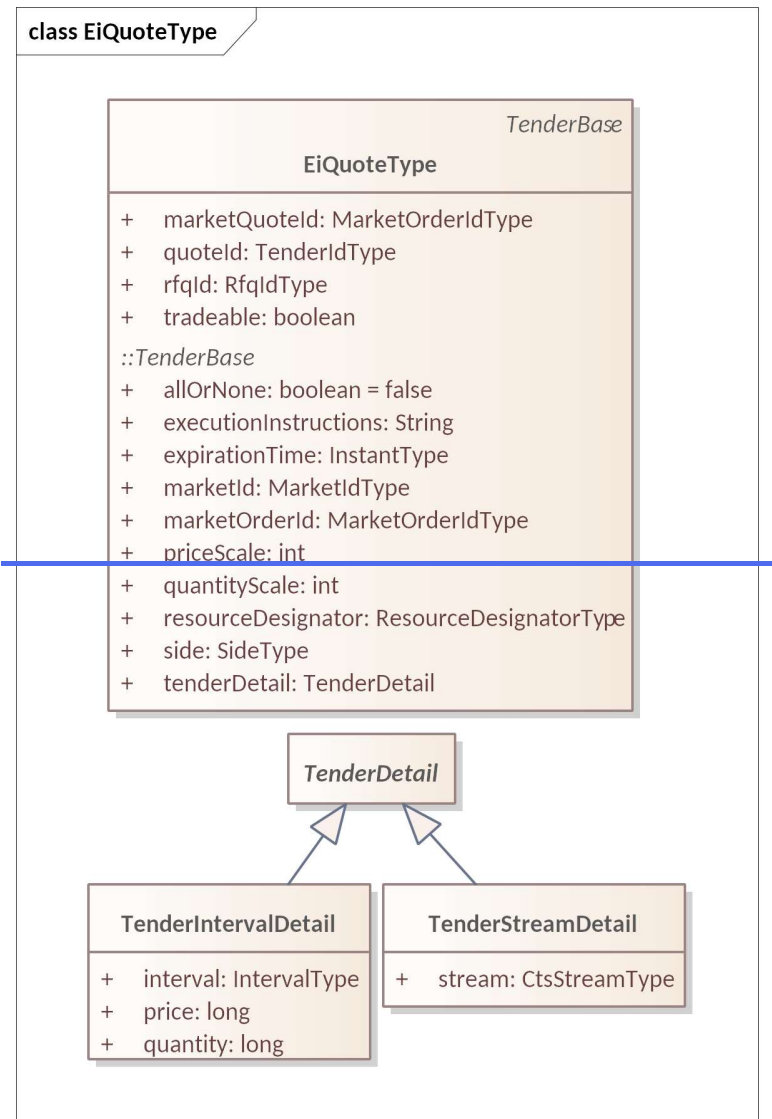


Figure 9-7: UML Diagram of EiQuoteType showing inherited attributes.

The following tables show attributes for the Quote Messages.

Table 9-7: Attributes of EiCreateQuotePayloadEiQuoteType

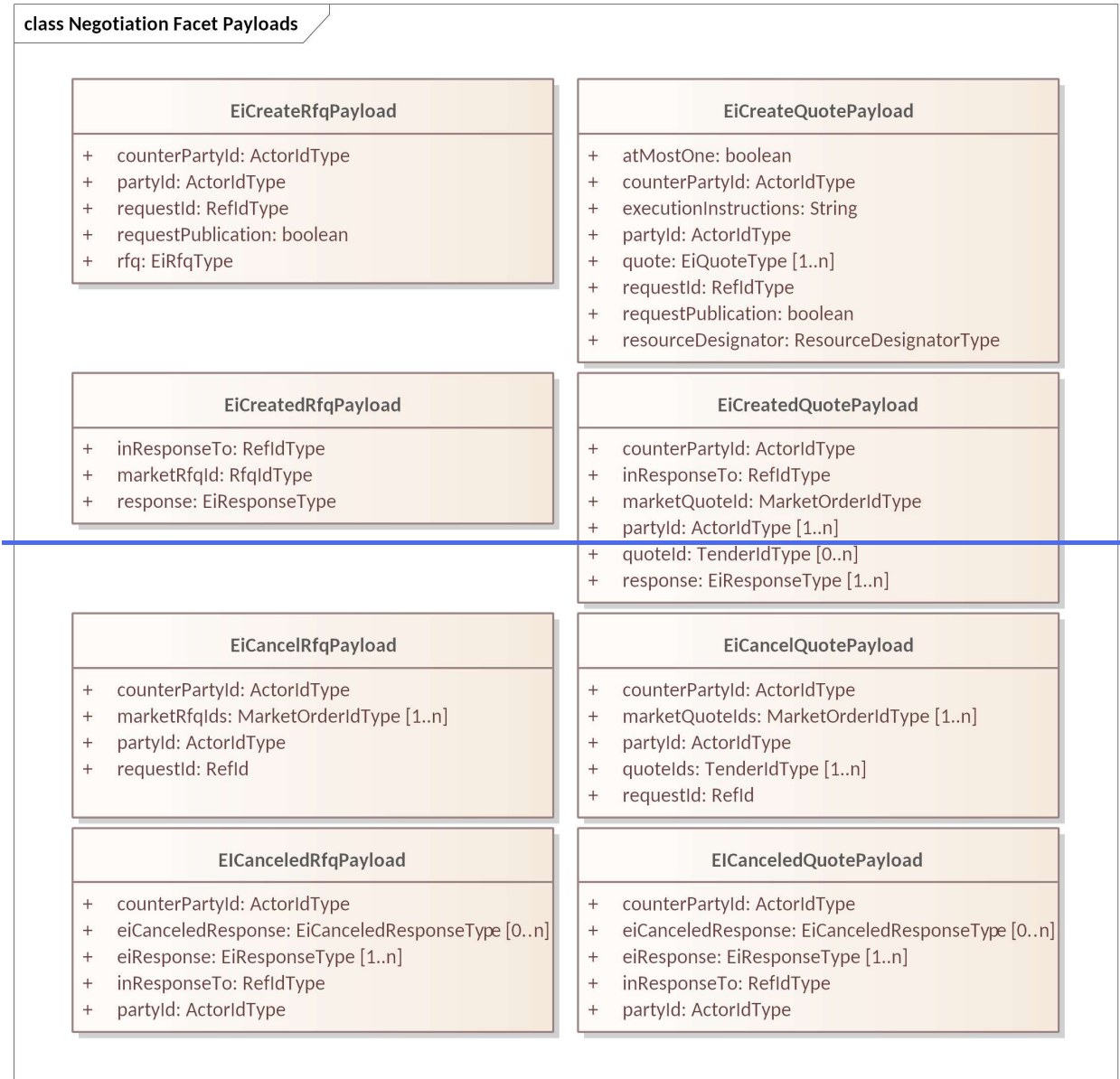
Attribute	Type	FIX Field Name	Meaning	Notes
At Most One	Boolean	See ContingencyType (1385)	Used to express alternatives, only one of which is to be effective	See Trading Instructions in Table 5-4. First match cancels other Tenders.

Attribute	Type	FIX Field Name	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Quote is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
Execution Instructions	String	ExecInst (18)	Execution Instruction.	Used only for multi-leg, and applies to all tenders in multi-leg. Execution instructions apply to each Tender in the List.
Market Quote-ID	Market OrderID Type	MarketID (1301)	ID assigned by the Segment or Market ID	Used in acknowledgment of tradable quote in future market messages. Identifier of the market of interest. An actor MAY be able to participate in more than one Market See Section 13
Private Quote Segment ID	Boolean	PrivateQuote (1171)MarketSegmentID (1300)	Specifies Quote is available to a specified counterparty or only identifies the Segment processing the Tender, Transaction, or Quote	This should be a unique combination paired with the Market Order ID
Quote Party ID	Quote ID Type Actor ID	QuoteID (117)PartyID (448)	The Actor ID as submitted by for the Party requesting the Quote originator.	Used in off market negotiation Also used in Tender to reference a Tradable Quote Indicates which Actor proposes the buy or sell side
RFQ ID Quote	RFQ ID Quote Type	QuoteReqID (131)	ID as submitted by RFQ originator The quote transmitted by this message payload	Referenced by Quote responding to RFQ One or more quotes
Request ID	RefID Type		Reference to this message payload	
Request Publication Quote Type	Boolean	QuoteType (537)	IOI or Tradable	Tradable is a Boolean; if true, the quote is tradable. If false the quote is not tradable, or an Indication of Interest (IOI) consistent with FIX terminology. The sender of EiCreateQuote (the initiator) requests publication by setting Request Publication to true. This is a request—there is no guarantee that publication is performed.

Attribute	Type	FIX Field Name	Meaning	Notes
All Other fields are as defined in the Tender, Table 5-2				

9.6 Negotiation Facet Payloads

shows UML Class Diagrams for the Negotiation Facet Payloads.



Figure—UML Class Diagram for Negotiation Facet Payloads

The following tables show attributes for the Quote Payloads.

Table 9-8 EiCreateQuotePayload EiCreatedQuotePayload				
Attribute	Type	FIX Field Name	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Quote is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counter-party is used.
EiQuoteInResponseTo	Ref ID	Not in FIX	As above An identifier for the payload to which this is a response	
Market Quote ID	Market Order ID Type	OrderID (37)	ID for this quote assigned by the Segment or Market	Used in acknowledgement and in future market messages
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party requesting the Quote.	Indicates which Actor proposes the buy or sell side
TradableQuote ID	Tender ID Type Boolean	(part of Codeset in FIX) OrderID	If false, this is an indicative quote. The quote transmitted by the EiCreateQuote message payload	Contains Zero or more quotes Boolean for whether the quote is Tradable.
Request Publication Response	Boolean EiResponse Type		Specific error responses	The sender of EiCreateQuote may request publication by setting See Section 2.4 Request Publication to true, but has no guarantee that publication is performed.

9.7 Information Model for Stream Quotes

~~Some Market Segments may permit Stream Quotes, that is, a single Quote for multiple consecutive Instruments. When responding to a Request for Quote, the range of quoted intervals must be within the bounding interval of the Request.~~

~~Stream quotes and quotes are the same class.~~

9.8 Rejecting a Quote

~~Quotes may be rejected if ill formed, violate granularity rules, or other requirements of the [The](#) Segment or Market.~~

~~9.8.11.1.1 Creating Transactions from Quotes~~

~~A Party receiving a Tradable Quote MAY respond by submitting a Tender that references that Quote. The Market registers a Tradable Quote it receives AS IF it were a Tender, that is, places them in a book until it expires. A Quote that is marked as Advertised by using an EiDistributeQuotePayload is included in the Indications Ticker (see Section).~~

~~The Market [normally](#) does not [Advertise](#)[publish](#) a Quote that is [private or](#) directed to a specific Party or Parties, [though it MAY follow](#). [If the Quote Issuer requests Publication, then the Segment MAY do so](#)~~

following its anonymization and advertising publication practices. A Segment Publishes a Quote by distributing it using the Quotes Ticker. See Section 11.5.2, "Quote Ticker"

While the Quote Issuer can request Publication, the decision to Publish a Quote is made by the Segment. The Segment MAY be required to Publish Quotes from Parties identified as significant in the Market. Another Segment may decline to publish any Quotes to comply with privacy regulations.

9.6.2.1 Cancelling a Quote

A Party May cancel a Quote at any time so long as it has not previously been lifted by a Counterparty. The attributes of the Ei Cancel Quote payloads are in Table 9-9 and Table 9-10.

Table 9-9 EiCancelQuote Payload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market. To indicate a bilateral exchange, i.e., a Tender between two specific parties, the PartyID of a specific counterparty is used.
Market Quote IDs	Market Order ID Type	OrderID (37)	ID assigned by the Segment or Market.	One or more Market Quote IDs to request cancellation.
Party ID	Actor ID	PartyID (448)	Actor ID for the Party that created the Tender	
Request ID	Ref ID	Not in FIX	An identifier for this Cancel Tender Payload	

Table 9-10 EiCanceledQuote Payload Attributes

Attribute	Type	FIX Field	Meaning	Notes
Counter Party ID	Actor ID	PartyID (448)	The Actor ID for the CounterParty for which the Tender is created.	In CTS, generally the PartyID for the Market
Ei Canceled Response	Canceled Response Type	Not in FIX	Detailed response for each quote that was included in the EiCancelQuote Payload	
EiResponse	EiResponse Type	Not in FIX	Specific error responses	See Section 2.4
In Response To	Ref ID	Not in FIX	An identifier for the Cancel Tender Payload to which this is a response	
Party ID	Actor ID	PartyID (448)	The Actor ID for the Party on whose behalf this Tender was made.	Indicates which Actor proposes the buy or sell side EiCreateTender.

9.6.2.2 Accepting a Quote

To accept a [TradableTradeable](#) Quote, whether on first notice or after negotiation, a Party submits a [Tender](#) [EiAcceptQuote Payload](#) matching the Price and Quantity of the Quote and referencing the [Market Quote ID](#). [FIX and financial markets call this lifting a quote](#).

The [EiAcceptQuote](#) payload is exactly an [EiCreateTransaction](#) payload with the addition of a reference to the quote accepted because the match has been performed by the Quote/Accept cycles.

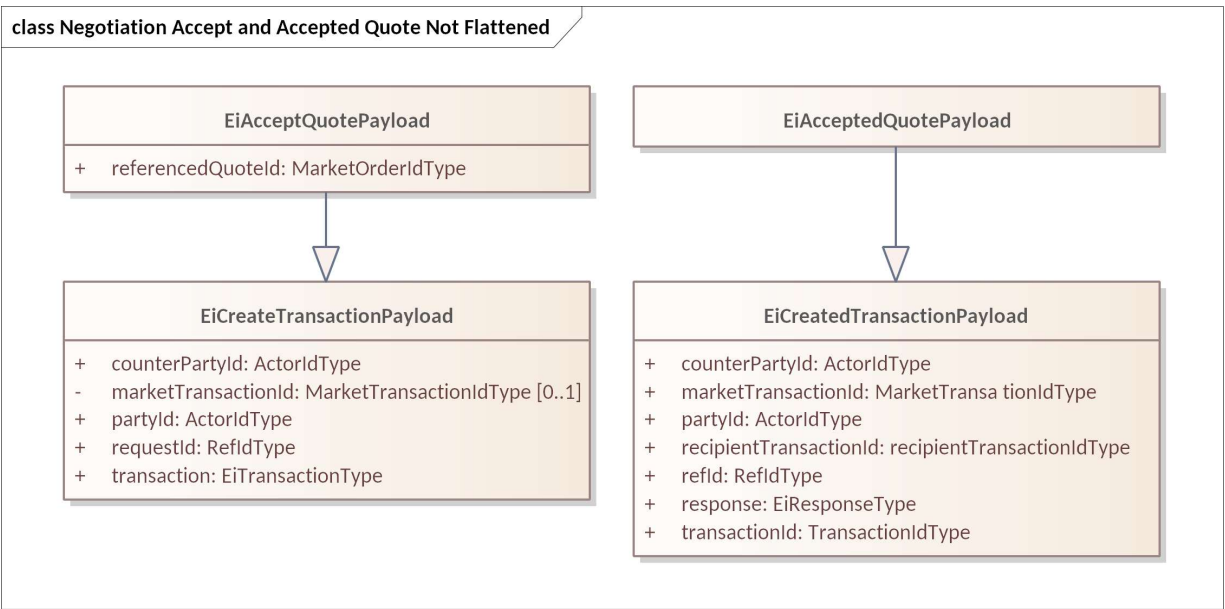


Figure 9.10QuoteID- Negotiation Facet Quote Payloads

The Market will then validate the match and create a Transaction if it fits. [The Tender must match price, be within the acceptable quantity by sending an EiCreateTransaction Payload. The TenderBase in the EiAcceptQuote must match Instrument, Price, and Quantity in the Quote, except in a Quote-Driven Market, wherein EiAcceptQuote can lift a part of the Quote, and so on. An. Quotes in Segments with Market Mechanisms other than Quote Driven must have an execution instruction of All-or-None Execution Instruction on either Quote or Tender MUST be honored. If the Quote accepts partial fulfillment, the. The Segment typically maintains the balance remaining balance on the Quote is decremented as it is for a Tender in a Quote.](#)

Notwithstanding any negotiation, the Market may reject the [TenderAccept Quote](#) if accepting it would interfere with resource operations or violate financial requirements on participants.

If a [TradableTradeable](#) Quote is open when the Instrument closes, it is the responsibility of the Party that submitted the Quote to cancel it. [If the Negotiator, and/or to have an appropriate expiration for the Quote. If the issuer](#) still wishes to accept an instrument scheduled for 11:00 at 11:30, that is up to the Parties; how that is accomplished is out of scope. [The Market will enforce its own rules for accepting the Transaction.](#)

9.9 Negotiation Messages

9.6.2.3 The Rejecting a Quote

[When a Party wants to end further negotiation messages are variations on a common theme, but are distinguished by, it replies with a Reject Quote message type.](#)

The [Quote Service does not waitUML class diagram](#) for [or expect acknowledgements of advertised Quotes.](#)

If permitted in the venue (Segment), a Party may submit quotes for several consecutive Intervals, a set of Instruments for an identical Product, as a single Quote. In Power, this is called a “Load Curve”. Stream Quotes are constructed `EiRejectQuote` and `EiRejectedQuote` is in Figure 9-11 interpreted Attributes are described in Table 9-11 the same manner as are Stream Tenders. All elements of the stream share the duration and the stream has the explicitly stated start time.

9.10 Discussion of Negotiation (Non-Normative)

Any Market Segment may support Negotiation as indicated in the Market Structure information. A Quote-Driven Market is one of the venue types (see) for a GTS Market Segment. In financial markets, a quote-driven market is a secondary market trading structure wherein buyers and sellers interact with dealers. It is not as transparent as an order-driven market in that the market orders and prices that traders are willing to buy or sell at are not available to the counterparty. A Quote-Driven Market MAY be chosen to negotiate with a dominant supplier that represents many third parties, i.e., a Distribution System Operator acting as an intermediary to a bulk power market. Such a Quote-Driven Market can permit large buyers to plan significant resource use over time, for example, scheduling a long running industrial process which also requires labor planning. Such a buyer could submit multiple Requests for Quotes with different schedules, and then select from among the Quotes received in response.

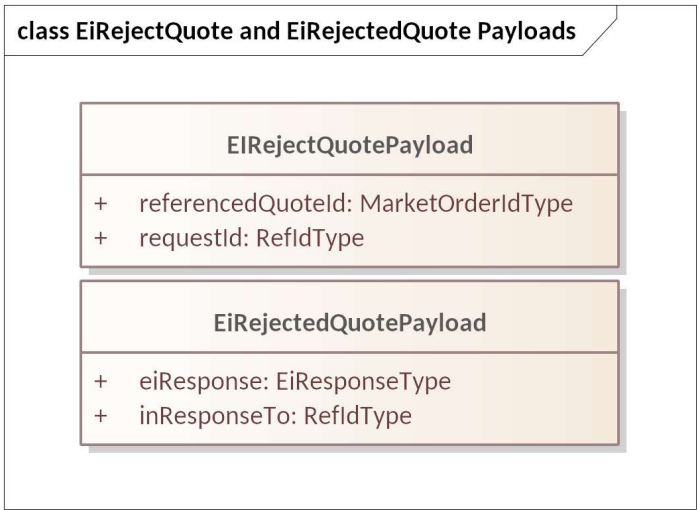


Figure 9-11 `EiReject` and `EiRejectedQuote` Payloads

Table 9-11 The overarching Market tracks all Transactions and monitors all Delivery.

10 Subscriptions

EiReject and EiRejected Quote Payload Attributes

<u>Attribute</u>	<u>Type</u>	<u>FIX Field</u>	<u>Meaning</u>	<u>Notes</u>
<u>Referenced Quote ID</u>	<u>Market Order ID Type</u>	<u>PartyID (448)</u>	<u>The Actor ID for the CounterParty for which the Tender is created.</u>	<u>In CTS, generally the PartyID for the Market</u>
<u>Request ID</u>	<u>RefIDType</u>	<u>Not in FIX</u>	<u>Reference to this message payload</u>	
<u>EiResponse</u>	<u>EiResponse Type</u>	<u>Not in FIX</u>	<u>Specific error responses</u>	<u>See Section 2.4</u>
<u>In Response To</u>	<u>Ref ID</u>	<u>Not in FIX</u>	<u>An identifier for the Cancel Tender Payload to which this is a response</u>	

10 Subscription Facet

A Party wishing to trade in a market naturally wants to be kept apprised of changing information about the market. This can be roughly divided into granular information about what other Parties are doing in the Market, and information about the Market ~~as~~, [Segment, Trading Session, or instruments \(high price, low price, quantity sold, etc.\)](#).

~~In this section we describe the common aspects of subscriptions, including starting and stopping, or a whole, one-time information message.~~

The FIX Protocol specification ~~terms~~[describes](#) these as Market Data, that is, granular or aggregate information about activities in a Market, and Market Structure ~~Reports~~[Reference Data](#), that is, information about how each Market Segment is operating.

~~FIX distinguishes between~~

- ~~• [Reference Data which changes very slowly if at all—think the name of a market, or that a market segment trades one hour energy, and](#)~~
- ~~• [Dynamic Data which changes more frequently—think orders to buy or sell an instrument, session trading status, session intraday unscheduled auctions, and the like.](#)~~

In the FIX Protocol, a Party gets this information by means of Subscriptions. A Party subscribes to the information it needs and thereafter receives periodic updates relating to that subscription. The FIX interaction model defines a *subscription* as how an Actor requests one or more market reports.

A Market consists of multiple Market Segments, each trading a single Product based on the Resource traded in that market. Multiple Market Segments in a Market MAY trade the same Product, perhaps with different trading rules, or different schedules of operation. The ~~Market~~ Segments in a Market may support different Market [Structure Reference Data](#) reports. Information about a Market and its ~~Market~~ Segments is conveyed in the Market Structure [Subscription Reference Data Subscriptions](#).

Subscriptions are how a Party requests specific Pre-Trade information. Not all Markets and Segments will support all Subscription types. The Subscriptions supported by each Segment are described in Section 13, “Market Structure [Reference Data: Market, Segment, and Session](#) Subscriptions”.

~~The following sections each use the Subscription pattern:~~

- ~~• [Section 1, “Tickers”—Tenders, Quotes, RFQs, and Transactions](#)~~
- ~~• [Section 12, “Instrument Data Subscriptions”—outstanding tenders to buy or sell, high and low prices](#)~~
- ~~• [Section 13, “Market Structure Reference Data: Market, Segment, and Session Subscriptions”—slowly or unchanging reference data, more changeable dynamic data](#)~~

Some markets ~~will or segments may~~ not support ~~granular~~[fine-grained](#) subscriptions. ~~The~~[In such cases, the Managed Subscription payload and/or Market Structure will instead Data Report MAY](#) indicate a multi-cast point or other source ~~that a Party can to which an actor may~~ choose to listen ~~to~~.

10.1 Subscriptions

~~Several varieties~~[In CTS, the message transport is layered and out](#) of market and trade-related information are [scope](#).

10.1 Messages for the Subscription Facet

~~All subscriptions follow a common pattern for creation, management, and cancelation. This facet includes messages for Tickers, Instrument Data, and Market, Segment, and Session Data, as described in this specification. To request and acknowledge the following sections. Those messages inherit from the core subscription messages, which are of abstract type as no actual messages use only this base.~~

[Table 10-1](#) changes objects of the EiSubscriptionRequest and Response Types Messages for the Subscription Facet

Facet	Request Payload	Response Payload	Notes
Subscription	EiManageSubscription	EiManagedSubscription	Create, manage, and cancel subscriptions

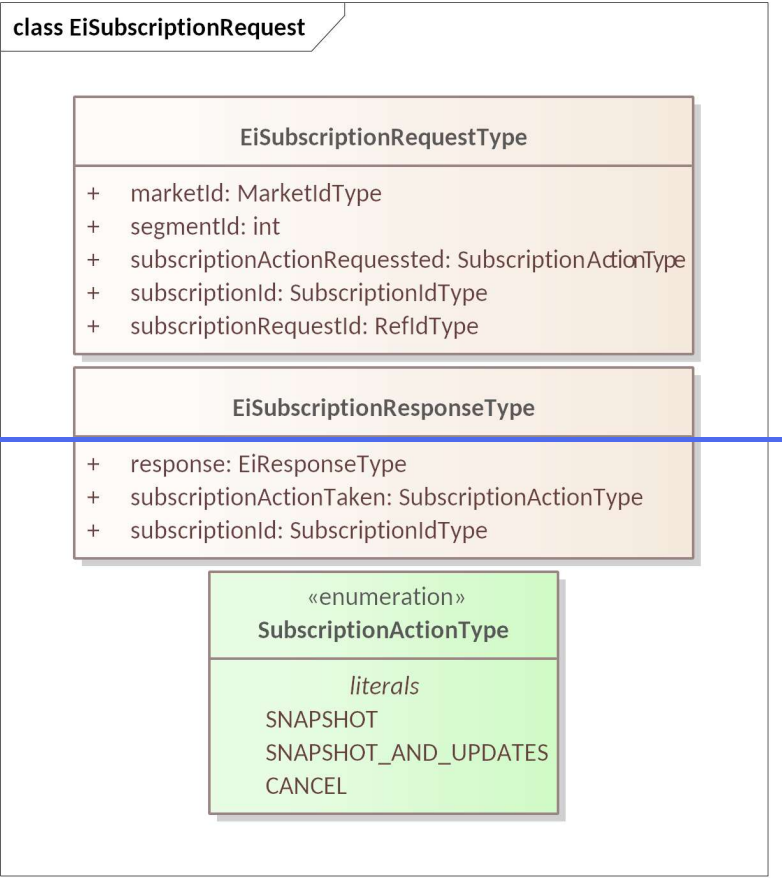
10.2 Interaction Pattern for the Subscription Facet

There is no UML sequence diagram for the Subscription Facet because the payload is abstract. The [manage interactions](#) are included defined in [Sections 1, 12, and 13](#) the appropriate subscription management messages. Specific subscriptions inherit from this pattern.

~~10.2~~10.3 Information Model for Subscription Requests and Responses

The UML Class Diagram for the Subscription Request and Response is shown in Figure 10-1. Specific requests, [for example](#) for tickers ~~or, instrument, and~~ market information, are defined in [Section ,the following sections.](#)

1430



1431

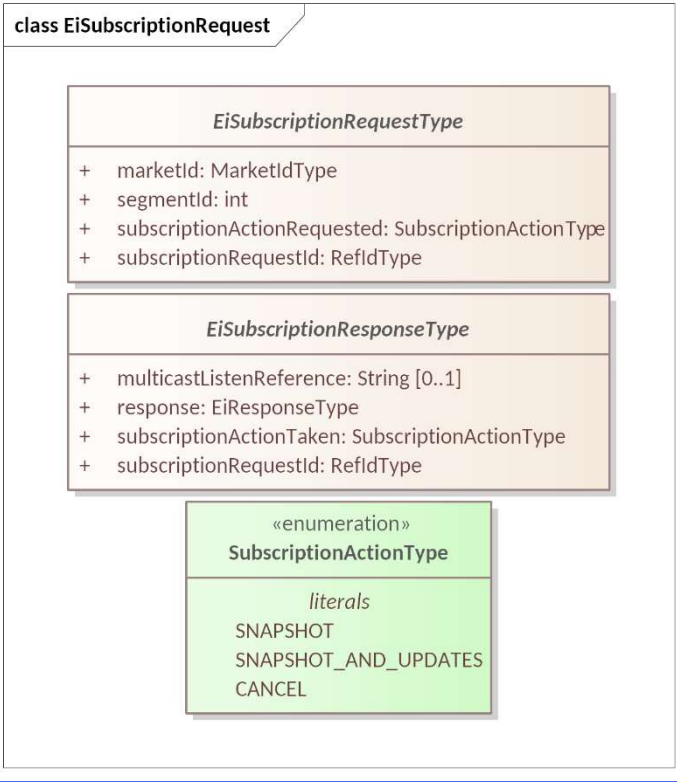


Figure 10-1 UML Class Diagram for Subscription Request and Response Types

Attributes for [EiSubscriptionRequest](#)the Subscription Request are shown in Table 10-2. We follow FIX's approach in using the message ID (RefID) as the subscription identifier.

Table 10-2 EiSubscriptionRequest Attributes

Attribute	Attribute Type	FIX Field Name	Meaning
Market ID	Market ID Type	MarketID (1301)	Identifier of the market of interest. An actor MAY be able to participate in more than one Market. See Section 13 “Market Structure Reference Data: Market, Segment, and Session SubscriptionsA UID referencing a Market”
Segment ID	Natural NumberInteger	MarketSegmentID (1300)	The FIX MarketSegmentID is a UID represented by a string; CTS uses an integer for Segment ID. If Segment ID is non-zero, the request is limited to reporting on the indicated single Segment. If zero, the subscription requests reporting on all Segments of the indicated Market.
Subscription Action Requested	EnumerationSubscription Action Type (enumeration)	SubscriptionRequestType (263)	The Subscription response type of subscription being requested. CTS uses an enumeration carrying that matches the same information as the pattern of FIX numeric codes: 0 – SNAPSHOT 1 – SNAPSHOT + Updates (update frequency is determined by AND UPDATES 2 – CANCEL See the supplier) 2—End the indicated Subscription discussion following this Table.
Subscription ID	Subscription ID Type		A UID indicating the created subscription, so that it may be canceled or modified. Should be null for initial request.
Subscription Request ID	Ref ID Type	MDReqID (262)	Used to identify this request for managing a subscription. This is an identifier for the subscription and must be used to cancel. See ALSO FIX MarketDataRequest (35=DR).

Attributes for [EiSubscriptionResponse](#)Subscription Response are shown in Table 10-3.

Subscriptions are inherently asynchronous. A Snapshot subscription request asks for a full report when the provider responds. A Snapshot and Updates returns a full report and will return updates at some future times. Cancel stops all future Updates and ends the Subscription.

There is no expectation that each market participant can or should be able to get perfect knowledge about all other participants; and creating a capability of doing so would likely prevent the development of the emergent knowledge which is the purpose of transactive resource markets.

Table 10-3 EiSubscriptionResponse Attributes

Attribute	Attribute Type	FIX Field Name	Meaning
MultiCast Listen Reference	String		If present and non-null the Subscription Manager provisions the subscription by sending a reference to (e.g.) a multicast in lieu of direct messages. Optional.
Response	EiResponse Type		A standard CTS response type; see Section 2.4, “Responses”.
Subscription Action Taken	EnumerationSubscription Action Type (enumeration)	SubscriptionRequestType (263)	The action taken on the referenced or newly created Subscription ID .
Subscription Request ID	SubscriptionRef ID Type		A UID indicating the newly created, modified, or canceled subscription.

11 Ticker Facet

A Party wishing to trade in a market naturally wants to be kept apprised of changing information about the market. FIX divides this information into two categories: Market Data and Market Structure. Market Data reports activity in the Market, Market Structure describes how the Market (or Segment in CTS) is organized.

11 Tickers

This section [applies the subscription pattern of Section 10](#) and describes mechanisms to access continuous Market Data on the activities of market participants. CTS calls these *Tickers*. Tickers update continuously, on a schedule determined by the provider, as Parties interact with a Segment.

~~A Party wishing to trade in a market naturally wants to be kept apprised of changing information about the market. There are~~The FIX Protocol divides this information into three categories: Orders, Trades, and Bids/Offers. CTS defines Tickers for Tenders [Orders], Transactions [Trades], Quotes [Bids/Offers], and Requests for Quotation [RFQs].

The four types of Tickers, [are](#) represented as an enumeration. See Table 11-1 and Figure 11-1 below.

Table 11-1: Types of Tickers in CTS Facet

Ticker Type	Request Payload
BidsQuotes	Anonymized Tenders to Buy Published Indicative (non-Tradeable) Quotes
RFQsIndications	Advertised Indications of Interest (IOI) and Published RFQs RFQs
OffersTenders	Anonymized Tenders offering to Buy or to Sell
Transactions	Anonymized Completed Transactions Trades, whether from market matches or from Negotiation

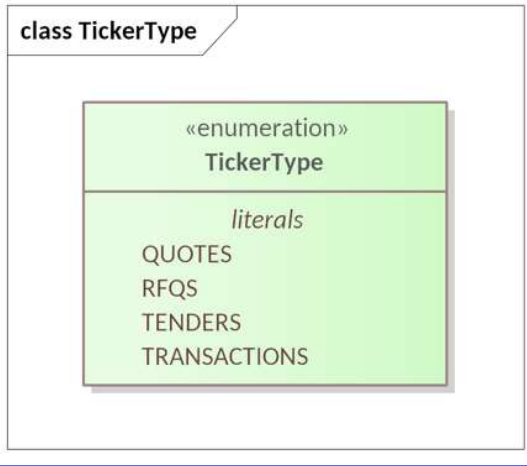
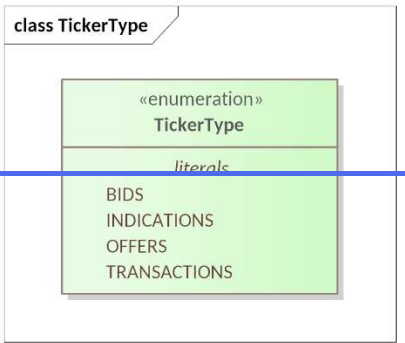


Figure 11-1 TickerType Enumeration

Not all Markets or Market Segments support Ticker subscriptions or all Ticker types. Actors can discover what Tickers a Segment supports and how to interact with them through the Market [Structure reports](#).

~~Market Structure is~~Reports as discussed in Section 13, “Market Structure [Reference Data: Market Segment, and Session Subscriptions](#)”.

Private Quotes do not appear in Tickers.

~~In many markets,~~ It is ~~required for common that, following market or segment rules,~~ most ~~participants to be parties in tickers are~~ anonymized, that is, the identity of the ~~submitter kept private; party is not disclosed.~~ In such ~~marketssituations,~~ the Market [Party ID](#) is used as the Party ID ~~and/or Counterparty ID~~ in the Ticker.

In Resource markets as in financial markets, ~~Partys in~~Parties with specific and/or influential roles are not anonymized. For example, a Market may ~~optchoose~~ not to anonymize the Party ID of the distribution system operator (DSO). ~~CTS~~

[This specification](#) makes no statement about what anonymization rules a resource market must use. This specification offers ~~the~~ general guidance that most participants be anonymized to preserve privacy, but that Ticker messages for significant participants ~~canmay~~ be ~~submitteddistributed~~ under their own identity.

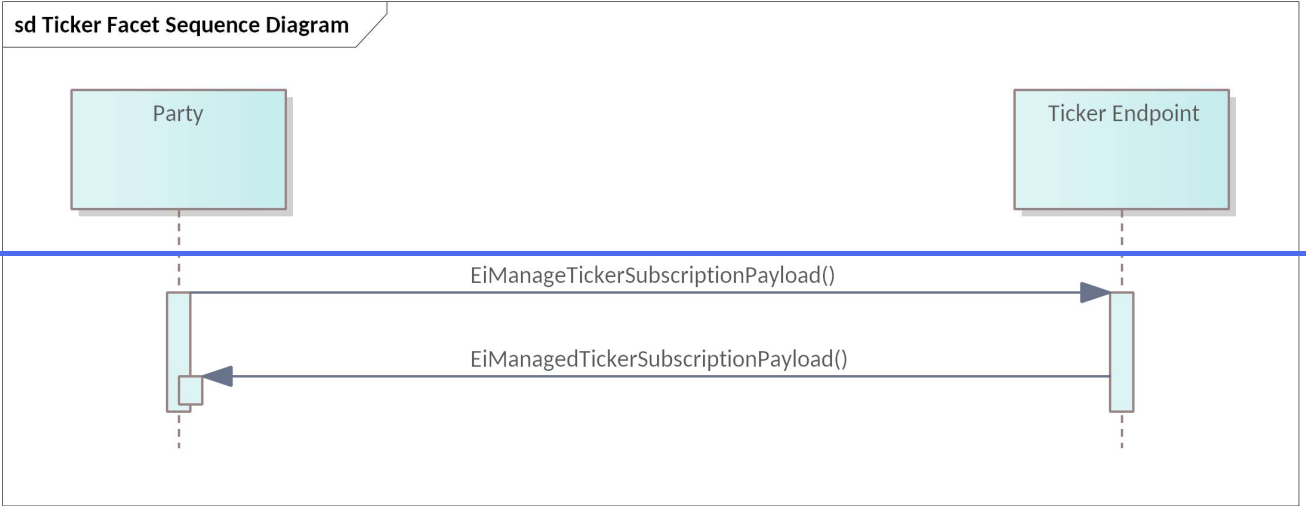
11.1 Ticker Facet Subscriptions

11.1 A PartyMessages for Tickers

[An Actor](#) subscribes to a Ticker ~~common FIX market~~[based on the](#) subscription model (Section 10, “[Subscription Facet](#)”). ~~A Party~~). [An Actor](#) can subscribe to a single Market Segment or [any or](#) all Market Segments [in a Market](#). Each Ticker Type, if available, requires a separate Subscription.

Table 1121-2:2 Ticker Facet [Messages](#)

Facet	Request Payload	Response Payload	Notes
Ticker	EiManage Ticker Subscription- Payload	EiManaged Ticker Subscription- Payload	As multiple Markets may use same Ticker service, must allow multiple subscriptions.
Ticker	EiDistributeTicker	None	Publish Indication to the Ticker



11.2 Interaction Pattern for Tickers

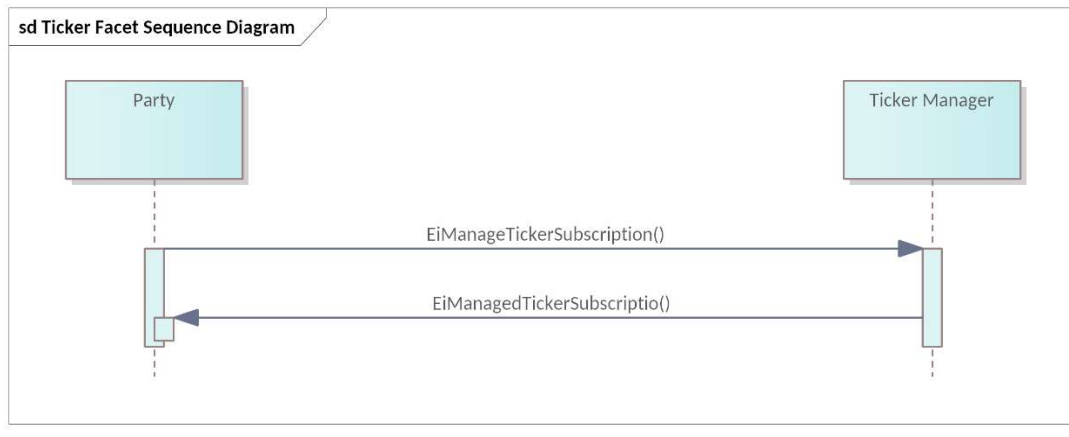


Figure 11-2: UML Sequence Diagram for the Ticker Facet

11.1.1 Information Model for Ticker Facet Subscriptions

The messages for adding, changing, or deleting a Ticker subscription contain only the ticker type and a subscription request or response. The UML Class Diagrams for the payloads are shown in .

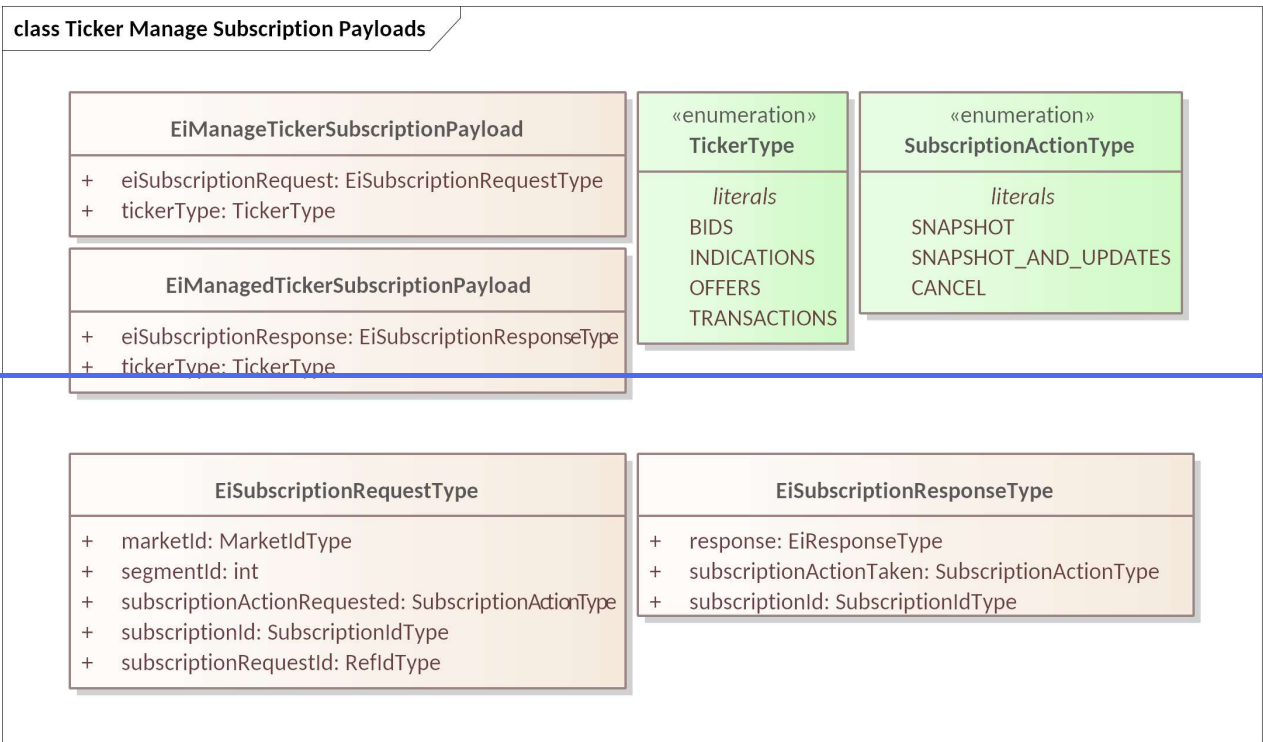


Figure 11-3 Ticker Manage Subscription Payloads

11.211.3 Exceptions to Ticker Subscription Interactions

A given Segment may ~~support the same~~ provide a single Ticker ~~for data stream combining~~ any or all of the Ticker ~~types. In that case, Payloads.~~ An Actor that subscribes ~~one of these Tickers to any Ticker implicitly~~ subscribes to all the Types included ~~in~~ with that Ticker.

In larger markets, there may be a ~~common~~ broadcast ~~or multicast~~ channel for a Ticker. In such markets, there is no subscription; the Actor simply ~~listed~~ listens to that broadcast channel. The Subscription Id is not part of the multicast and an Actor unsubscribes as per the transport used.

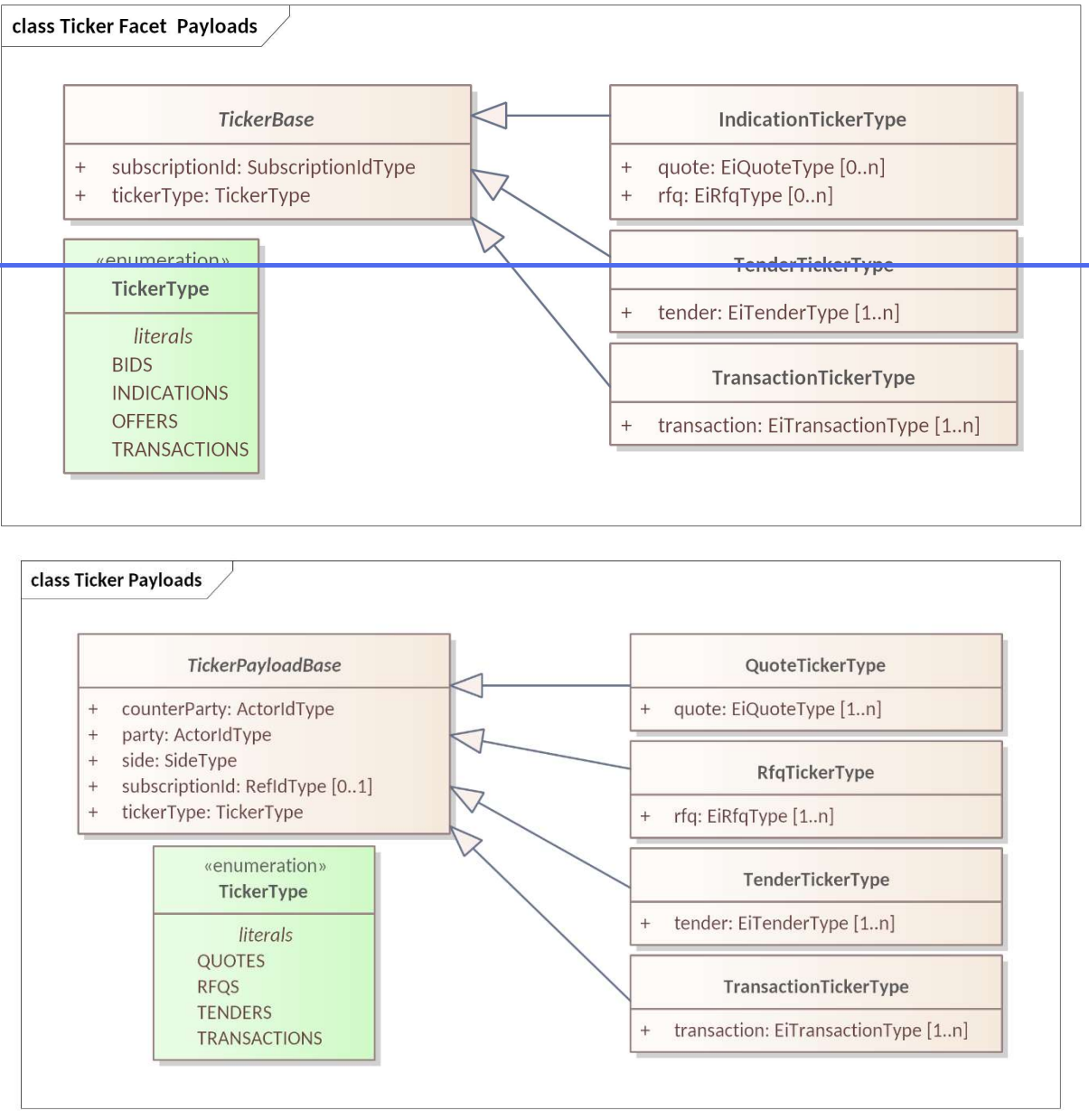
11.311.4 Ticker Interaction Patterns for Ticker Data

The various types of tickers share a common approach:

- A subscription is created using EiManageTickerSubscription, passing the requested change and which ticker is being managed.
- The ticker payloads contain the subscription ID and the relevant object for the ticker type:
 - TenderTickerType is EiTenderType for Bid and Offer tickers.
 - TransactionTickerType is EiTransactionType
 - IndicationTickerTypeQuoteTickerType is ~~either an~~ EiQuoteType (not ~~tradable~~) or tradeable)
 - RfqTickerType is EiRfqType.
- The Ticker Payloads are described below in Figure 11-3 delivery. Delivery of Ticker Payloads is ~~defined by the provider out of scope.~~
 - Large or complex markets might use a multicast for delivery, ~~with using~~ the relevant ticker payloads: (out of scope)
 - Small or less complex markets might use a market-defined delivery mechanism (out of scope)

1519 **11.5 Information Model for Ticker Payloads**

1520 Ticker payloads are sent asynchronously when subscribed. The UML Class Diagram Diagrams for Ticker
1521 types is shown Payloads and Ticker Type are in Figure 11-3.



1523
1524 Figure 11-3 Ticker Facet Payloads and Ticker Type showing inheritance

1525 The attributes for the Ticker Types Payloads are shown in Table 11-3. Ticker Payloads will be delivered
1526 pursuant to ticker subscriptions on a Segment.¹⁹

¹⁹ Just as for message payloads, how these are delivered is out of scope. Some Markets and Segments may use multicast or direct delivery.

Table 11-3 Attributes for the Ticker [TypePayload Base](#) and [Ticker Types](#)

Attribute	Attribute Type	FIX Field Name	Meaning
Counter Party	Actor ID Type	PartyID (448)	The counterparty in the ticker payload by type; may be anonymized per market rules.
Party	Actor ID Type	PartyID (448)	The party in the referenced ticker; may be anonymized per market rules.
Side	Side Type	Side (54)	The side for the referenced ticker; note that an EiTender, etc., have side in the inherited TenderBase. The Side in the Ticker Payload Base MUST match that in any referenced object.
Subscription Request ID	SubscriptionRef ID Type		<p>AAn optional UID indicating the related subscription.</p> <p>Present only for individual subscriptions but MAY be absent even then.</p> <p>NOTE that if delivered via (e.g.) multicast or broadcast, customization of Subscription IDs cannot be done, so this attribute MAY be absent.</p> <p>Cancellation of a multicast Snapshot-and-Updates subscriptions is accomplished by sending an EiManageTicker Payload with the original Subscription Request ID.</p>
Ticker Type	TickerType Ticker Type enumeration		See Figure 11-3Enumeration consisting of literals BIDS, INDICATIONS, OFFERS, or for class diagram. The values are QUOTES, RFQS, TENDERS, and TRANSACTIONS.
Quote	EiQuoteType		For QuoteTickerType; the Side attribute in TenderBase MUST be the same as Side in Ticker Payload Base.
RFQ	EiRfqType		For RfqTickerType; the Side attribute in TenderBase MUST be the same as Side in Ticker Payload Base.
Tender	EiTenderType		For TenderTickerType; the Side attribute in TenderBase MUST be the same as Side in Ticker Payload Base.
Transaction	EiTransactionType		For TransactionTickerType; the Side attribute in TenderBase MUST be the same as Side in Ticker Payload Base.
Quote	EiQuoteType		Alternate for IndicationTickerType; either all Quotes or all RFQs.
RFQ	EiRfqType		Alternate for IndicationTickerType; either all Quotes or all RFQs.

~~11.3.111.5.1~~ **The Bids and OffersTender Tickers**

Bids and Offers are simply Buy or Sell side Tenders. When a Tender is submitted, the Segment announces the Tender on the Ticker. ~~A Segment may use a single Ticker for both Bids and Offers as described in its Market Structure reports. Tenders are submitted to the entire market; There is no guarantee that an Offer or a Bid will still be there when a Party submits a matching Tender. subject to the Segment rules and requests for publication and privacy.~~

~~Tradeable Quotes MAY be published in a ticker only, if it is delivered by means of an EiDistributeQuote payload. See Section Negotiation Facet.~~

~~The payload for Bids and OffersTenders are submitted to the entire market segment; there is no guarantee that a Tender will still be available when a Party submits a matching Tender.~~

~~The Market and/or Segment may Publish Quotes subject to Issuer request for Publication, subordinate to Market and/or Segment rules.~~

~~The payload for Tender~~ Tickers includes one or more EiTenderType objects with attributes anonymized following market or segment rules. Attributes are shown in Table 5-2: [EiTender](#) Attributes.

~~A Party that wishes to receive Bids or OffersTenders from a Segment must subscribe to the Bid or to the Offerthat Segment's Tender Ticker for that.~~

11.5.2 Quote Tickers

~~If a Segment. If a single Ticker only is provided Bids and Offers, then subscribing to either one subscribes to both.~~

11.4 The Indications Ticker

~~If a Segmentits Market Mechanism supports Negotiations, then it supports an Indication Ticker. While the messages in Bids and Offers are essentially identical, except for BUY or SELL side, a Quotes Ticker. There is more diversity in the messages for IndicationsQuotes than in Tenders.~~

~~The payloadQuote attribute of the IndicationsQuotes Ticker Type is just as defined for the IOI in Section 9 "Negotiations." Because the purpose of a public offer ("publishing a Quote") is to initiate a Negotiation between PartysParties, the IndicationsQuotes Ticker is not anonymized.~~

~~The objects in the Indication Ticker payload are either all Quotes or all RFQs.~~

11.5 The Transactions Ticker

11.5.3 RFQ Tickers

~~While the type and semantics of RFQs and Quotes are closely related, the separation simplifies the data model. There may be reasons for a negotiation market to not support an RFQ Ticker.~~

11.5.4 Transaction Tickers

~~The Transactions Ticker is the continuous publication of information about Transactionsadvertisement of Trades executed in a Market Segment. Both Parties are listed on a Transaction, although either or both may be anonymized as specified in market rules.~~

~~In some Market Mechanisms (see 13.1, "Market MechanismsIn Negotiation-based Markets, the Transaction announcement is the first public appearance of the Negotiation. includes the payload") the contract may be negotiated privately. Note: even a Transaction that was negotiated privately will be published in the Transaction ticker based on market rules.~~

11.6 Message Payloads for the TransactionTicker—an EiTransactionType object withManaging Ticker Subscriptions

The messages for adding, changing, or deleting a Ticker subscription contain only the ticker type and a subscription request or response as defined in Table 10-2 and Table 10-3. The UML Class Diagrams for the message payloads are shown in Figure 11-4.

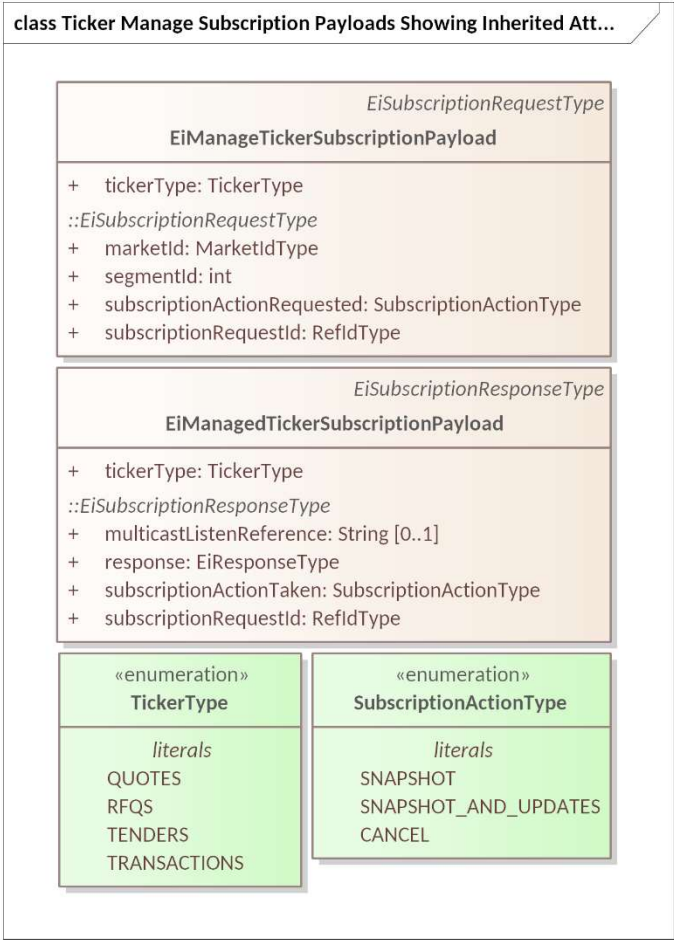


Table 11-4 Ticker Manage Subscription Payloads showing inherited attributes

Table 11-4 shows the attributes for the EiManage and EiManaged Ticker Subscription Payloads.

Table 11-4: EiCompletedTransaction-4 Attributes for the EiManage and EiManagedTickerSubscription Payloads

Attribute	Attribute Type	FIX Field Name	Meaning	Notes
BuyerIdTicker Type	UidTicker Type enumeration		ID of the Buyer in this TransactionThe Type of Ticker for subscription.	May be ExchangeID if anonymized
SellerId	Uid		ID of the Seller in this Transaction	May be ExchangeID if anonymized
All other attributes are as in EiSubscriptionRequest and EiSubscriptionResponse in Table 10-2The remaining fields are as defined in the Tender, and Table 10-3				

12 Instrument Market Data Subscriptions

Instrument Summaries are [Subscriptions obtained by Subscription](#) (described in Section 10) ~~that and~~ provide [marketdynamic](#) data about ~~the~~ specific Instruments traded in the Segment. Like other Subscriptions, Instrument Summary Subscriptions provide [partan aspect](#) of what FIX [termcalls](#) Pre-Trade Data.

12.1 The information in the Instrument Summary Subscription

An Instrument Summary Subscription requests periodic data on a bounded range of Instruments. Within a Market Segment, trading is for a single Product, both Reference and Instruments are distinguished by Start Time and Date. The Subscription returns market data for [Dynamic](#)—the Reference Data includes all Instruments whose Start Time falls within the Bounding Interval of the Subscription.

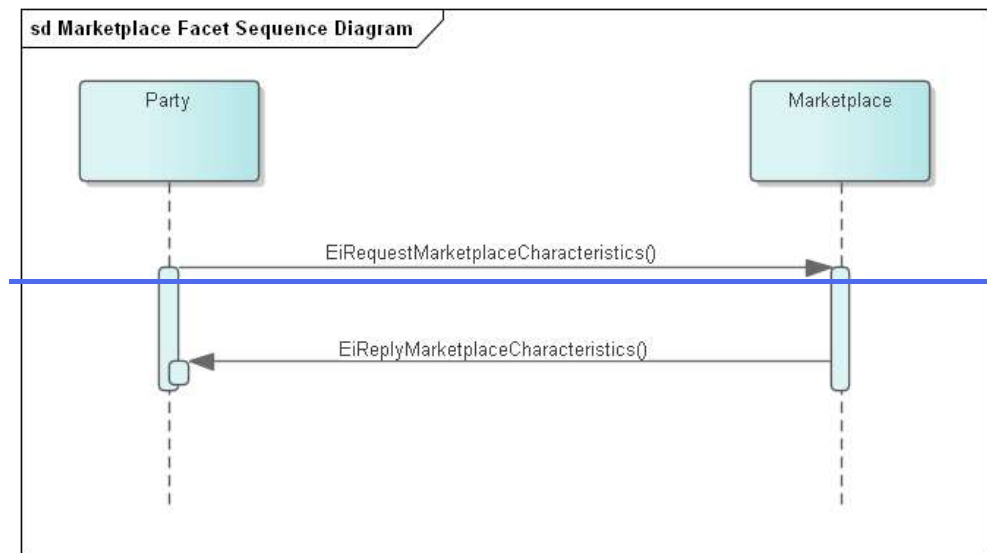


Figure — [Need new UML](#) the attributes of Instrument Session Report Type except for the Instrument Summary [Subscription Request](#)

: the Reference data in effect describes the market, segment, resource, and similar reference data. The combination is dynamic, with static identifying information.

The request for a subscription includes the usual requests for snapshot, snapshot and updates, and unsubscribe (see Section 10) with the addition for Instrument Reference data of how to update.

The Subscription Manager may restrict the frequency and the content. Certain requests, e.g., multiple levels of the order book, or many instruments, involve a lot of data. Some restrictions are described in Section 13.3.3 "Information Model for Segment Reference Data Instrument Summary Subscription Request specifies the type of summary and the" (Max Summary Instruments and Market Depth.

12.1 Messages for Instrument Reference Data Subscriptions

Subscription requests need additional information beyond the EiManageSubscription payloads:

- [The Bounding Interval](#) for instruments requested.
- [A limit on how many instruments to supply reference data](#)
- [How to update the requested reference data if the subscription requests updates—incremental or a full update](#)

Table 12-1: Information Model Messages for Instrument Summary Subscriptions Reference Data

Attribute	Attribute Type	FIX Field Name	Meaning	Notes		
<u>Facet</u>	<u>Market Data Request ID Payload</u>		<u>String Response Payload</u>	<u>MDReqId (262) Notes</u>	<u>ID of this Market Data Request</u>	<u>Use the ID of previous Market Data Request to modify or cancel a previous request</u>
<u>Request Type</u>	Char	<u>Subscription Request Type (263)</u>	Indicates to the other party what type of response is expected.	0 = A snapshot request only asks for current information. 1 = A snapshot and updates. 2 = Unsubscribe, will cancel any future update messages from the counter party		
<u>Update Type</u>	Char	<u>MDUpdateType (265)</u>	0 = Full Refresh 1 = Incremental Refresh			
<u>Market Segment ID</u>	String	<u>MarketSegmentId (1300)</u>	Unique Identifier for Segment	Market Data Reports are for a single Segment		
<u>Bounding Interval</u>	<u>Start Date</u> <u>End Date</u>	<u>Bounding Interval</u>	<u>Bounding Interval?</u>	Subscription covers all Instruments within Bounding Interval		
<u>Subscription = Instrument Summary Type Reference Data</u>	<u>IntEiManage Instrument Ref Data</u>		<u>Type of EiManaged Instrument Summary to subscribe to:</u> 0 = Session Summary 1 = Top Of Book Ref Data	<u>Create, manage, and cancel subscriptions</u>		

Attribute	Attribute Type	FIX Field Name	Meaning	Notes		
Facet	Market-Data Request IDPayload		StringResponsePayload	MDReqId (262)Notes	ID of this Market-Data Request	Use the ID of previous Market-Data Request to modify or cancel a previous request
Market Depth	Int	MktDepth (264)	Depth of market for Book Snapshot / Incremental updates 0 = full book depth 1 = top of book 2 or greater = book depth (number of levels)			

The

12.2 [Interaction Pattern for Instrument Summary ReportReference Data Subscriptions](#)

[An Instrument Reference Data Subscription requests data on contiguous temporal range of Instruments. Within a Market Segment, trading is for a single Product, and Instruments are distinguished by the resource delivery Interval. The Subscription returns market data for all Instruments whose interval falls within the Bounding Interval of the Subscription request.](#)

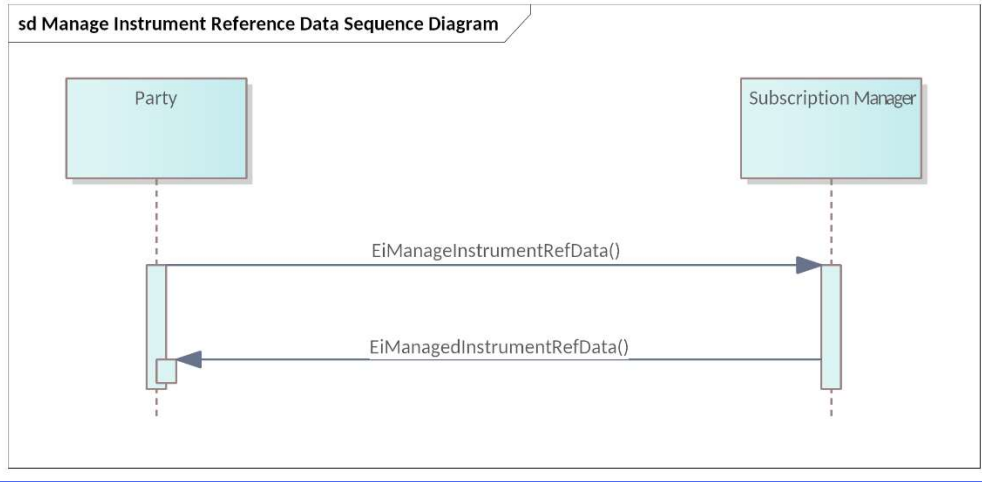
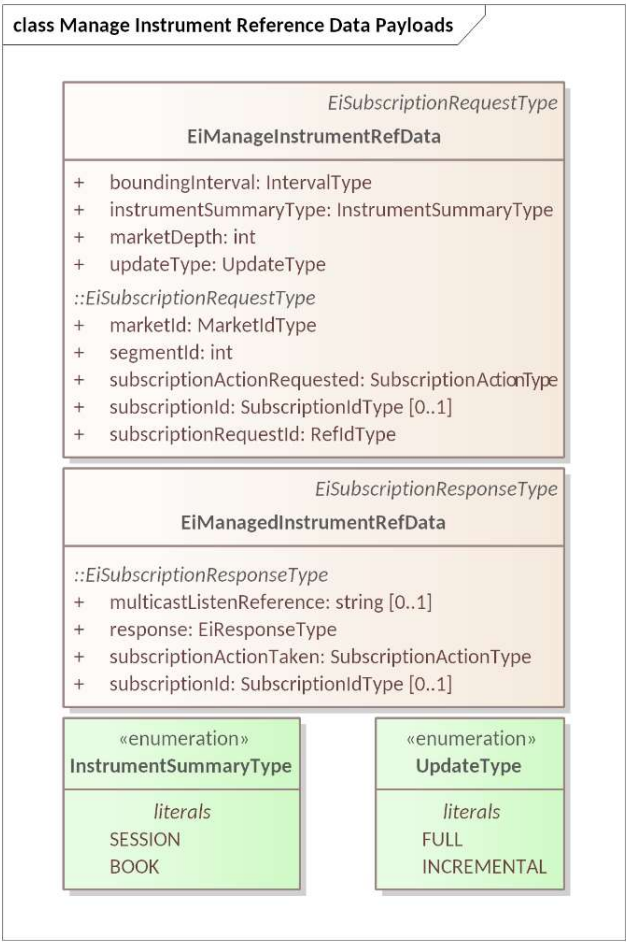


Figure 12-1 [Manage Instrument Reference Data Subscription](#)

1616 **12.3 Information Model for Manage Instrument Reference Data**
1617 **Subscription Payloads**

1618 The UML class diagram for the Manage Instrument Reference Data messages is in Figure 12-2.



1619
1620 Figure 12-2 UML Class Diagram for Manage Instrument Reference Data Messages

1621
1622 The Manage Instrument Reference Data payload specifies the type of summary and instruments
1623 requested. Its attributes are in Table 12-2.

1624 Table 12-2: Attributes for Manage Instrument Data Payload

Attribute	Attribute Type	FIX Field	Notes
Bounding Interval	Interval Type	Not in FIX	Subscription request is for all Instruments within the Bounding Interval. What is returned is at the discretion of the Segment. The request will return information on all instruments within the [closed] time interval whose start is at or later than the Bounding Interval start and whose end point is at or before the end of the Bounding Interval. See Section 3.3 “The Bounding Interval Pattern in CTS”

<u>Attribute</u>	<u>Attribute Type</u>	<u>FIX Field</u>	<u>Notes</u>
<u>Instrument Summary Type</u>	<u>Instrument Summary Type Enumeration</u>	<u>Not in FIX</u>	<u>Type of Instrument Summary for subscription. FIX integer values are</u> <u>0 = Session Summary (CTS: SESSION)</u> <u>1 = Book (see Market Depth attribute) (CTS: BOOK)</u>
<u>Market Depth</u>	<u>Integer</u>	<u>MarketDepth (264)</u>	<u>Depth of market requested for Book Snapshot and/or Incremental updates</u> <u>0 = full book depth</u> <u>1 = top of book</u> <u>2 or greater = book depth (number of levels).</u> <u>The Segment may limit the response the depth indicated by Market Depth attribute of Segment Reference Data.</u>
<u>Update Type</u>	<u>Update Type Enumeration</u>	<u>MDUpdateType (265)</u>	<u>Fix values and Enumeration</u> <u>0 = FULL</u> <u>1 = INCREMENTAL</u> <u>The nature and frequency of Incremental Updates is at the discretion of the Subscription Manager.</u>
<u>The remaining attributes are inherited from EiSubscriptionRequestType (Table 10-2)</u>			

The attributes for the *Managed Instrument Reference Data Payload* are all inherited from EiSubscriptionResponseType (Table 10-3).

12.4 The Instrument Session Reports

As with Tickers (See Section 1 "Tickers Instrument Summary") the actual requested information may be delivered by various means, including but not limited to multicast, point-to-point delivery, and publication to be downloaded by the actor, some of which may not support subscription request identifiers.

The Instrument Session Reports provide summary information about one or more instruments. Common information about the report is presented in a Market Dataclass Instrument Header. Information about each instrument is presented in Report (FIX calls this the Market Data Instrument Summary-Header).

The Information about each instrument is included as attributes in what FIX calls the Market Data Instrument Summary-Header.

In CTS, the messages are modelled modeled by Instrument Session Report Type which has zero or more Instrument Summary Type instances.

12.4.1 Information Model for the Instrument Session Report Type

The UML class model for Instrument Session Report Type is shown in Figure 12-2.

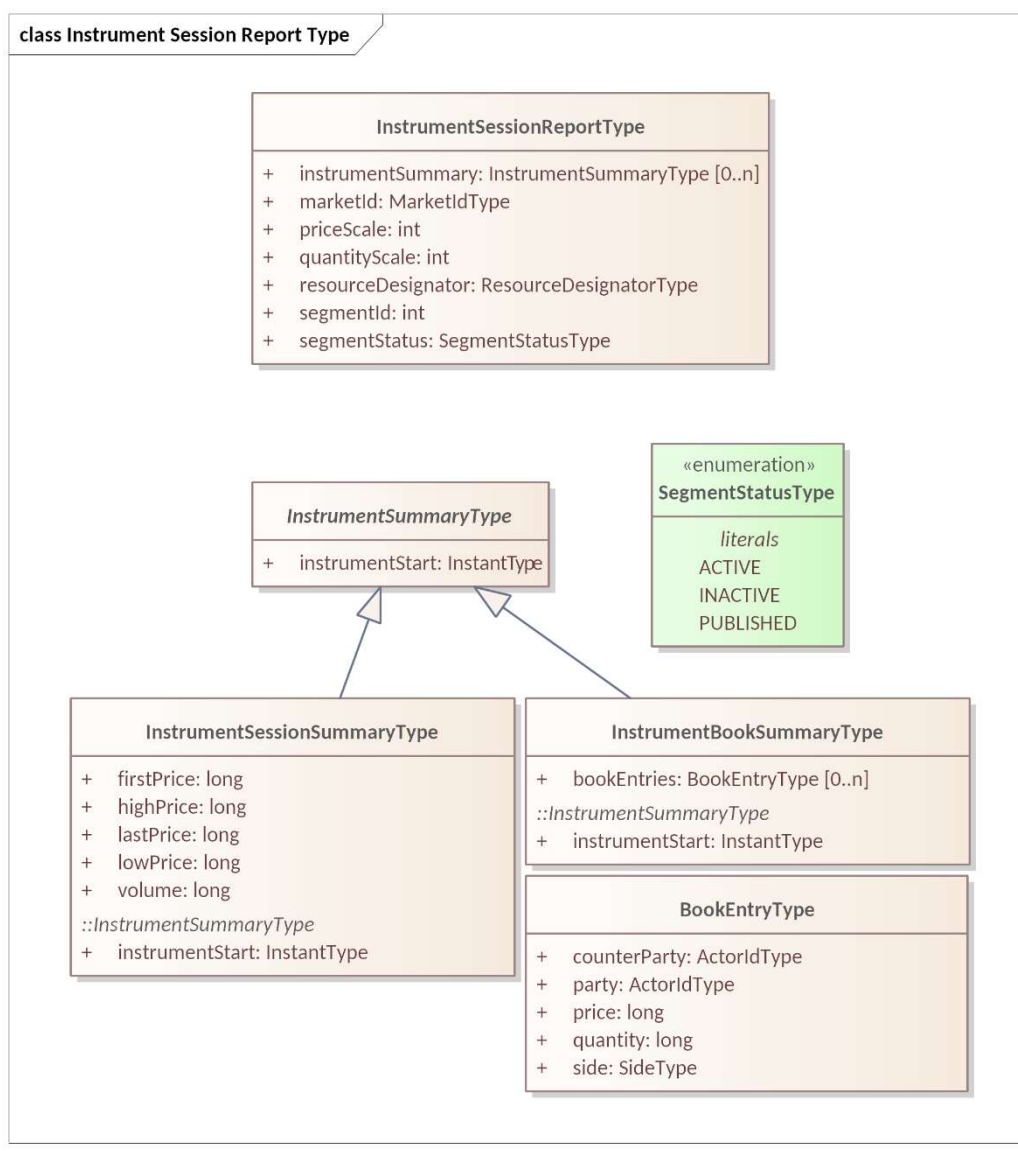


Figure 12-3 UML Class Diagram for Instrument Session Report Type

The attributes for the classes in Figure 12-3 are shown below.

Table 12-3: Information Model: Attributes for the Market-Instrument SummarySession Report Type

Attribute	Attribute Type	FIX Field Name	Meaning	Notes
Instrument Summary	Instrument Summary Type	Series	In FIX a repeating series for each Instrument in the Report. The information varies by the Summary Type requested.	Zero or more Instrument Summaries; type is in response to that requested in the Instrument Summary Type included in the Manage Instrument Reference Data payload .

Attribute	Attribute Type	FIX Field Name	Meaning	Notes
Market Data Request ID	StringMarket Id Type	MDReqId (262)Market ID (1301)	ID of thisMarket Data Request	Use the ID of previous Market Data Request to modify or cancel a previous requestIdentifier of the market of interest. An actor MAY be able to participate in more than one Market See Section 13.
Price Scale	Integer	Not defined in FIX	A multiplier for the Price used in this segment	A market segment might be denominated in e.g. dollars or 10ths of a cent.
Quantity Scale	Integer	Not defined in FIX	A scale factor for the Resource unit for this Segment	A multiplicative factor, e.g. 100, to convert from market quantity units to the base unit size traded in this Segment.
Request TypeResource Designator	CharResource Designator Enumeration	Subscription Request Type (263)FIX Instrument Component	Indicates to the other party what type of response is expected. Identifier of the Resource being offered	While a Market only accepts Tenders and Quotes for a single Resource, the complete description is required to ensure validity and for off-market interactions. 0 = A snapshot request only asks for current information. 1 = A snapshot and updates. 2 = Unsubscribe, will cancel any future update messages from the counter party
Market Segment ID	StringInteger	Segment (1300)	Unique Identifier for Segment	FIX Segment is a string to allow a UID. CTS Segment is an integer intended to be used with a MarketID UID.
Market Segment Status	IntSegment Status Type Enumeration	MarketSegStat (2542)	1 = Active: Market segment is active, i.e. trading is possible. 2 = Inactive: Market segment has previously been active and is now inactive. 3 = Published: Market segment information is provided prior to its first activation. Segment status as of time of report	1 = Active (CTS: ACTIVE): Market segment is active, i.e. trading is possible. 2 = Inactive (CTS: INACTIVE): Market segment has previously been active and is now inactive. 3 = Published (CTS: PUBLISHED): Market segment information is provided prior to its first activation.
Instrument Summary		Series	Repeating series for each Instrument in the Report. The information varies by the Summary Type requested.	

12.2.112.4.2 The Instrument Summary Types

The Instrument Summary is the information in an Instrument Summary Report that is repeated for each Instrument in the range.

The Instrument Summary are a repeating group of summaries, one for each instrument in the Market Data Instrument Summary Subscription.

The information conveyed varies with the Instrument Subscription Type. The UML class model for Instrument Summary Type is shown in Figure 12-4. The attributes are shown in Table 12-4, Table 12-5, and Table 12-6.

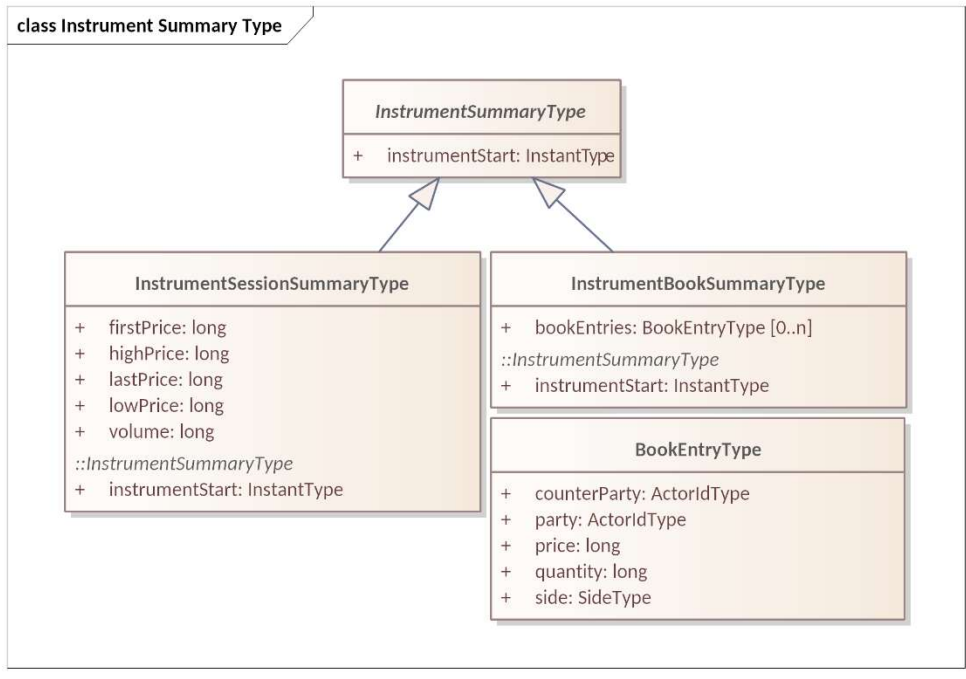


Figure 12-4 Instrument Summary Type UML Class Diagram

12.2.112.4.2.1 The Instrument Session Summary

A common change reported in an Instrument a Session Summary is market crossing, announcing when a Segment opens and announces when a Segment closes. Session changes its state. In transactive resources, each Instrument crosses as well closes on its own schedule. A Segment may not permit trading in an Instrument more than forty-eight hours in the future. A Segment might not permit trading an Instrument with a Start DateTime in the past can no longer be traded. We term the union of Segment crossingschedule and Instrument crossingtradability the Instrument Session.

Instrument Session report summaries include opening prices and closing prices, and volume traded. Note that all prices are scaled using Price Scale in the Session Report. Volume is not scaled.

The UML class diagram is in Figure 12-4

Attribute	Attribute Type	FIX Field Name	FIX Attribute Type	Meaning
Instrument Start Begin DateTime	Begin DateTime Instant Type	Date Time Not in FIX	DATETIME	Start Time Start time that identifies this instrument, and thereby this Instrument Session Summary Detail
High First Price	Integer High Px (332)	First Px (1025) Price	PRICE	The high end of Indicates the first price range prior to the open of a Session; can be a bid, offer, or reopen trade price.
Low High Price	Integer Low Px (333)	High Px (332) Price	PRICE	The low high end of the price range prior to the open or reopen
First Last Price	Integer First Px (1025)	Last Px (31) Price	PRICE	Indicates the first last price of a trading Session; can be a bid, ask offer, or trade price.
Last Low Price	Integer Last Px (31)	Low Px (333) Price	PRICE	Indicates The low end of the last price of a trading session; can be a bid, ask, range prior to the open or trade price. reopen
Volume	Integer	Total Volume Traded (387)	QUANTITY	Total volume traded of an instrument, including negotiated and market trades.

12.2.1.212.4.2.2 The Instrument Book Summary

The Book is the set of all Tenders, including Tradable Tradeable Quotes, in the Market Segment. In an active market, unless there are restrictions on matching, all Tenders to sell are priced higher than all Tenders to buy; if there were an overlap, they would already be matched, have generated Transactions generated, and the Tenders already would be removed from the book.

All Tenders are sorted by price. Tenders to sell are sorted by ascending price—the top of the book is the tender offering the cheapest price. Tenders to buy are sorted by descending price—the top of the book is the tender offering the highest price.

The depth of the Book is the number of Tenders in a list of the volume bid or offered at each list price. The Book sorts Bids by descending price. The Book sorts Offers by ascending price. A Top of the Book request, subscription depth of 1, provides just the top entry in each list, anonymized. A subscription depth of 0 provides both entire sorted lists, anonymized. Any other subscription level (n) provides the first (n) entries in each level.

The UML class diagram is in Figure 12-4.

Attribute	Attribute Type	FIX Field Name	Meaning
Begin Date Time Instru ment Start	Date Time Instant Type	Begin Date Time Begin Date eTime	Start Time Time stamp (inherited from Instrument, and thereby this Instrument Session Summary Detail Type)

Attribute	Attribute Type	FIX Field Name	Meaning
Book <u>Entry</u> <u>Entries</u>	<u>CompoundBook</u> <u>Entry Type</u>		Repeating element for each side and level of the Book

The Book Entry is the repeating information for each Side in the Book. The Book Entry is the same message format as a Quote, anonymized as required by market rules. Book Entry attributes are in Table 12-6. The UML class diagram is in Figure 12-4.

Table 12-6~~The Book entry is always summarized. That is, if there are five Tenders at the top of the offer side with the same price, the quantity returned in the Book Entry is the sum of the Quantities in each of those~~ Book Entry Attributes

<u>Attribute</u>	<u>Attribute Type</u>	<u>FIX Field</u>	<u>FIX Attribute Type</u>	<u>Meaning</u>
<u>Counter Party</u>	<u>Actor ID Type</u>	<u>PartyID (488)</u>	<u>String</u>	<u>Party for the specific Side Type. MAY be anonymized following Market Rules.</u>
<u>Party</u>	<u>Actor ID Type</u>	<u>PartyID (488)</u>	<u>String</u>	<u>Party for the specific Side Type. MAY be anonymized following Market Rules.</u>
<u>Price</u>	<u>Long</u>		<u>Price</u>	<u>Price in the book. Subject to Price Scale.</u>
<u>Quantity</u>	<u>Long</u>		<u>Qty</u>	<u>Quantity in the book. Subject to Quantity Scale.</u>
<u>Side</u>	<u>Side Type</u>	<u>Side (54)</u>	<u>Char</u>	<u>On which side is the Party?</u>

Tenders. The same rule applies to bid side of the Book.

13 Market Structure Reference Data: Market, Segment, and Session Subscriptions

For any Market, 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 what Products and Tenders meet those expectations. a Party may have to use many interactions to discover where to trade for the Products that meet that Party's needs.~~ ~~The Market Structure Report tells the Party how to trade.~~ Following the classification used by FIX, Market Structure is just one part of Pre-Trade Information.

For the Trader, the questions include

- "What products are traded in this Market, and where are they traded?" (Market Structure changes are limited)
- "How and when can I trade in each venue in this market? (Segment description)
- "When can I trade and what instruments can I trade now (Session information)

~~CTS uses the standard mechanism of the CTS Subscription to Opening, Closing, Settlement and to the currently Active query the Market structure including enumerating the Segments, to describe each Segment, and the status of the current trading session in each Segment. A Trading Session is a period for trading in a Segment between the opening and the closing of the Segment.~~

~~Segment Structure includes Opening, Closing, as well as Crossing information for specific Instruments. In CTS Markets, the Instruments tradeable in a Trading Session may change regularly as Instruments enter or exit the trading window of the Market.~~

~~A Party must interact with a specific Segment to trade a specific Product. A Market MAY contain two or more Market Segments trading the same Product; such segments may differ in the Market Mechanism, or in trading window. For example, a regulated provider may offer a day-ahead hourly market based on an Auction between 9:00 AM and 3:00 PM. It also includes detailed information to guide Training, detailed rules and negotiations and trading.~~

13.1.1 Venue Types

~~The same actor may trade the same Product by order book in another Segment. The Auction and the Order Book are different mechanisms for matching buyer and seller.~~

~~A Party chooses to trade in the Segment that it anticipates will be to its greatest advantage. The Party will make this choice based on anticipated price, or on block size, or even on Warrants. Because Transactions are committed when created, a Party may buy on one Segment, and thereafter sell part of it on another. Segments may be available for trading on different schedules, and the Instruments available in each Segment change over time. The Segment Structure provides detailed information to guide trading, negotiation, and settlement. The Segment Structure defines when Sessions open and when Sessions close.~~

~~All trades occur in Trading Sessions. Trading Session Data provides information over time on trading in a Segment. Trading Session Status informs whether a Session is available for trading, and when that status will change. A Trading Session's Tradeable Instrument Trading Range enables a Party to compute whether an Instrument is currently tradeable.~~

~~A Party discovers a Market, including changes over time, by subscribing to Market Structure Reference Data. Market Structure Reference Data includes a description of all Segments in the Market. A Party discovers and monitors a Segment by subscribing to the Segment Reference Data. A Party monitors the changing constraints on a Segment by subscribing to Trading Session Data.~~

~~This Section describes the interactions to subscribe to Market Reference Data, to Segment Reference Data, and to Trading Session Data.~~

13.1 Market Mechanisms

One of the most important distinctions between ~~Market~~ Segments is the ~~Venue Type~~.

[Market Mechanism. The FIX Trading Community defines standard Market Model Types \[MMT\]. MMT classifies the Venue Type to describe the mechanisms and general mechanism of trading algorithms that operate a Market.](#)

A Party participating in trading may change its behavior based on the [Venue Type mechanism the Segment uses to settle trades](#). The optimum trading strategy for a Party will change between an order book and an auction. If there is only a single seller, the Buyer will want to attend closely to the quotes from that buyer.

[CTS characterizes each Segment in part by its mechanism. The Venue Type categorizes how a Transaction is generated by the Venue. A Party wishing to buy or to sell will change strategies based on the rules of the Venue. FIX MMT defines many Venue Types \(FIX tag 1430\), only some of which mechanisms that are supported by CTS-conforming Markets not included by CTS. CTS also supports mechanisms not included in FIX, such as a self-executing mechanism to settle the difference between consumption as measured at the Meter \(Delivery\) and the Position as known \(see Section 7, "The Position Facet" and Section 8 "The Delivery Facet"\).](#)

[Figure 13-1 lists shows the UML Class Diagram for the Market Mechanism Enumeration. Detailed description is in Table 13-1 Venue showing the Market Mechanism Types supported by CTS and the FIX MMT information.](#)

[A Market Mechanism is an attribute of a Segment; Sessions related to each Segment share the ordinary trading mode from the Segment MMT, but a Session MAY have a trading mode, e.g. "scheduled opening auction" or "unscheduled auction", which is described in Section 13.4 "Trading Session Data".](#)

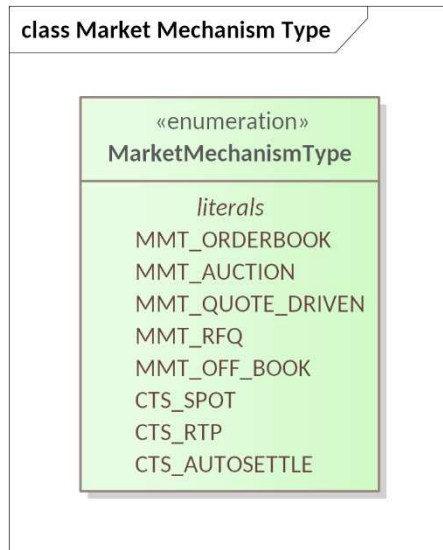


Figure 13-1 Market Mechanism Type Enumeration

Table 13-1: ~~Venue~~ Market Mechanism Types in CTS

FIXM MT Code	MMT Name	CTS Enumeration	Meaning
BLB	Centralized Limit Order Book OrderBook	MMT_ORDERBOOK	Participants submit their buy and sell orders, which are matched based on specific rules and executed accordingly. Order Books provide transparency, liquidity, and support for different types of orders, but they also have limitations and rely on the integrity and fairness of the exchange. Also referred to as a Centralized Limit Order Book. CTS specifies a “hard” order book, which executes orders immediately and automatically.
OPA	OffMarketPeriodic Auction	MMT_AUCTION	An Auction Driven Market matches Tenders only in scheduled auctions wherein all participants clear at the same price. Trades are conducted outside the market, but registered in the automated exchange. Sometimes referred to as an Over the counter (OTC) or Bilateral market. Off-Market trades may be for odd lots, for custom durations, and span the temporal boundaries of Products. The exchange may reject a trade that would cause violation of the physical limits of the Resource distribution. In existing power markets, also referred to as a “Double Auction”, that is, an auction in which both sellers and buyers submit bids.

FIXMMT Code	MMT Name	CTS Enumeration	Meaning
QOB	QuoteDrivenMarketQuoteDriven Market	MMT_QUOTE_DRIVEN	Quote Driven Markets are used for Markets with a single one or more dominant suppliers suppliers . Parties are notified of the Quoted price for each Instrument and submit Tenders to match those Quotes . Example: A System Operator provides 24 quotes for the next day's hourly pricing in Quote Responses.
NRQ	QuoteNegotiationRequest for Quotes	MMT_RFQ	A Quote Negotiation A Request for Quotes Market is used for bilateral negotiations around price. Sellers may advertise round lots that they would like to buy or to sell, and to indicate an interest in buying or selling. Trades in a Request for Quotes Market may be for odd lots, for custom durations, and span the temporal boundaries of Products
AOB	AuctionDrivenMarketOff Book	MMT_OFF_BOOK	CTS reserves Off Book mechanisms for direct allocations of Resources from one Party to another. The Segment notifies the Parties executing the Transaction . An Auction Driven Market matches Tenders only in scheduled auctions wherein all participants clear at the same price. In Resource Markets, also referred to as a "Double Auction".
SSM	SpotMarket (Spot Market (CTS only; not in FIXMMT))	CTS_SPOT	A Ticker in a Spot Market indicates the "instant" special price in the Market Segment that the Segment will be used for "instant" purchases or sales, e.g. due to a transient or emergency situation related to the resource.
RT	Real Time Pricing (CTS only; not in MMT)	CTS_RTP	A Ticker broadcasts Indicative Quotes. Parties make no Tenders but consume a resource (as needed). Later, an Automatic Settlement Segment will generate Transactions based on Delivery.
XAS	Automatic Settlement (CTS only; not in FixMMT)	CTS_AUTOSETTLE	A Automatic Settlement venue creates Trades to align with consumption as measured at the meter (Delivery). Automatic Settlement self-executes Transactions for ResourceResources consumed without previously being bought. Automated Settlement occurs in any Market in which Delivery (consumption) is not limited to prior Position.

[A non-normative discussion about trading in Segments with each mechanism can be found in Appendix B Choosing a Market Mechanism.](#)

1764 **13.2 Market Reference Data**

1765 **13.2.1 Messages for Market Structure Reference Data**

1766 The payloads for Market Reference Data are shown below.

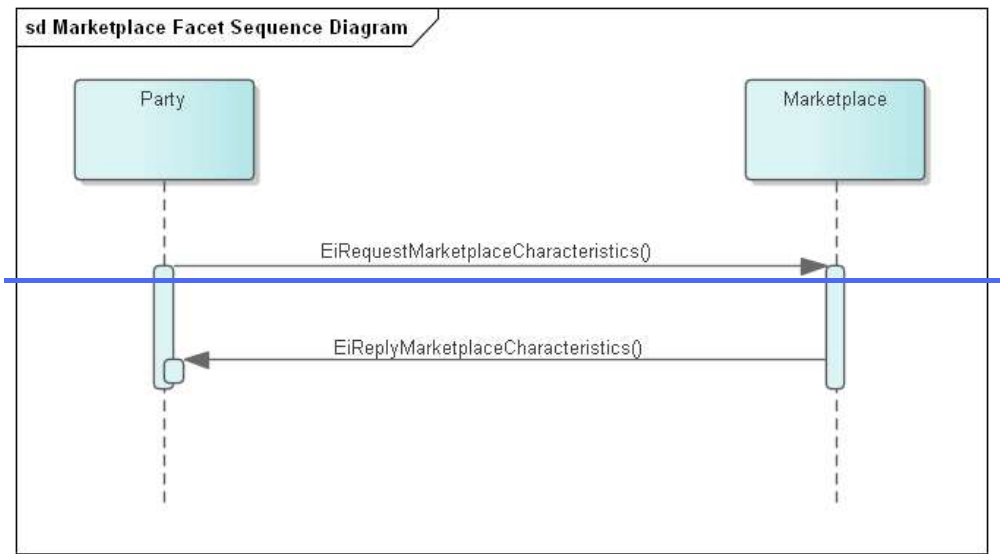
1767 Table 13-2 Messages for Market Reference Data

<u>Facet</u>	<u>CTS Initial Message</u>	<u>CTS Response Message</u>	<u>Meaning</u>
<u>Market Reference Data</u>	<u>EiManage Market Reference Data</u>	<u>EiManaged Market Reference Data</u>	<u>Request reference data for a Market (FIX Term is <i>Exchange</i>).</u>

1768 **~~13.1.2~~13.2.2 Interaction Pattern for Market StructureReference Data**

1769 The Market ~~Definition Facet~~Reference Data subscription enables a ~~Party~~an Actor to request the details of
1770 a Market and its ~~Market~~ Segments. The initial request returns the Market and all Segments.
1771 ~~SubsequentUpdate~~ reports occur when there is a change to a ~~Market~~ Segment or to Market Reference
1772 Data, and include the Market Reference Data plus only the changed Market Segment(s). ~~A request to~~
1773 ~~refresh a subscription returns the entire Market Structure.~~ A request to cancel the Subscription suspends
1774 all further updates.

1775 See Section 10 "Subscription Facet"



1776 "

1777 "

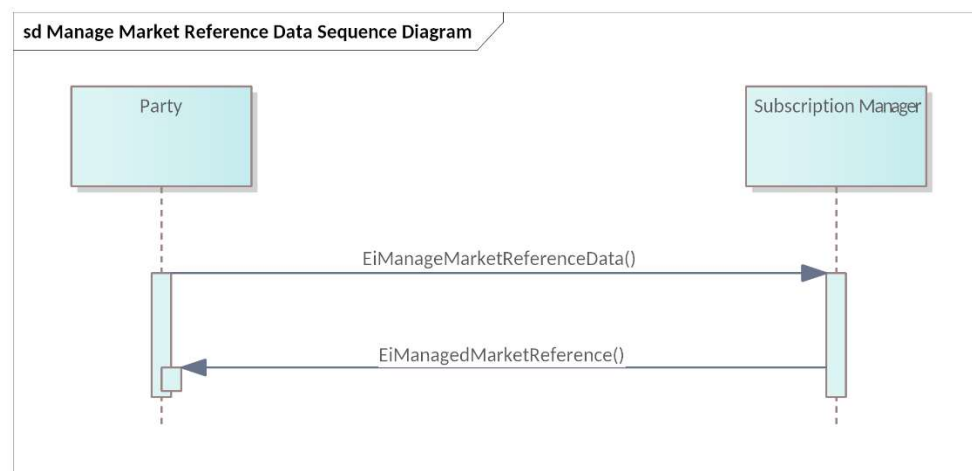
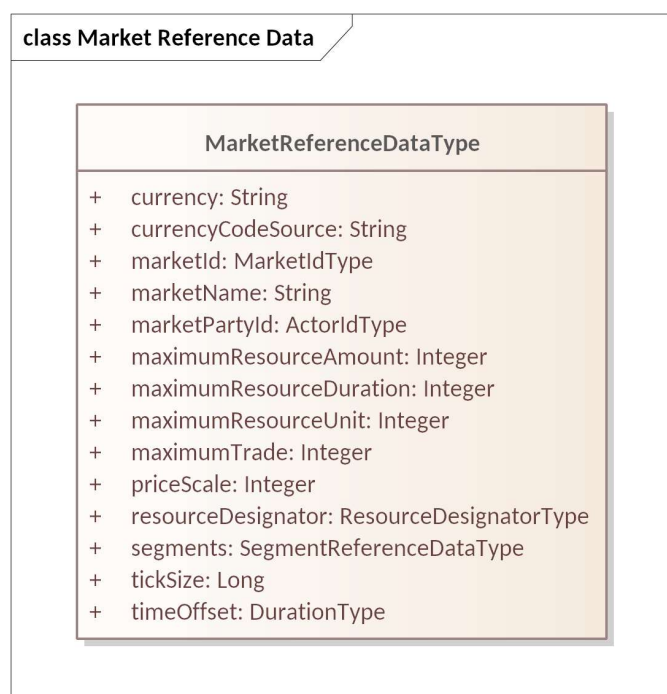


Figure 13-2: UML Sequence Diagram for the Market [Definition Facet Reference Data](#)

A Party may watch changes to a single [MarketSegment](#) by naming that Segment in the subscription request. This will return only that Segment and updates to that Segment.

[13.1.3](#) [13.2.3](#) [Information Model for Market Definition Reference Data](#)



[Table](#)

Figure 13-3 [Information Model: UML Class Diagram](#) for the Market [Definition Reference Data](#)

[Attributes for Market Reference Data are described in Table 13-3](#)

Table 13-3 [Attributes for Market Reference Data](#)

Attribute	Attribute Type	FIX Field Name	Meaning
Market Name	String	NAME	Text providing a descriptive name for a Market. While the Name MAY be displayed in a user interface; it is not meaningful to the Actors.
Currency	String	Currency (537)	String indicating how value is denominated in a market.
Currency Code Source	String	Currency Code Source (2897)	ISO – Fiat Currency per ISO 4217 DTI – Digital Token Identifier LOC – Locally defined Currency
Time OffsetMarket ID	DurationMarket ID Type	MarketID (1301) T-OFFSET	A Duration that some Markets MAY use to describe trading where a first interval is not on an hourly boundary. Note that in FIX, this is generally a formal identifier (e.g.) NYSE. If the market is a house, there is no place to look this up. There is always a UID for a Market.
Market Name	String	Not in FIX	Text providing a descriptive name for a Market. While the Name MAY be displayed in a user interface; it is not meaningful to the Actors.
Market Party Id Tick Size	PriceActor ID Type	Party Id Tick Increment (1208)	The PartyID used in Tenders to the Market and in Transactions with the Market. May also be used for anonymization of Parties. Specifies the valid price increments at which a Party may quote or trade an Instrument.²⁰ Use if a common Tick Increment required for all Market Segments. Tick Increments can increase market liquidity.
Maximum Resource Amount	Integer		Maximum Quantity of Resource Units per Maximum Resource Duration that the Market will permit.
Maximum Resource Duration	Duration		Duration for Maximum Resource Quantity
Maximum Resource Unit	Integer	Maximum Resource Unit	Units for Maximum Resource Flow per Duration.
Maximum Trade	Integer	Maximum Trade	The value of the largest trade that the Market permits.
Price Scale	Integer		Used to avoid floating point numbers in prices.

²⁰ Integer operations are typically much more efficient than fixed or floating point, so it is likely to be much faster to apply decimal shift on input and output rather than for more frequent comparison operations in the Execution Engine implementation.

Attribute	Attribute Type	FIX Field Name	Meaning
Resource Designator	Resource Designator Type	Resource	The Resource traded in this Market and Segment
Party Id		Party Id	The PartyID used in Tenders to the Market and in Transactions with the Market.
Market Segments	Segment Reference Data Type	Market Segment	A list of Market Segment descriptions for each Market Segment contained in the Market. See Section 13.3 “Segment Reference Data.”
Tick Size Maximum	Integer	Tick Increment (1208) MAX	Specifies the valid price increments at which a Party may quote or trade an Instrument.²¹ Use if a common Tick Increment required for all Market Segments. Tick Increments can increase market liquidity. Maximum Transaction size the Market will accept. Tick Size is a price and is scaled using PriceScale.
Time Offset	Duration Type	T_OFFSET	A Duration that some Markets MAY use to describe trading where a first interval is not on an “natural” boundary.²² For example, a market in one hour Power MAY start at 7 minutes after the hour.

13.2.4 Payloads for Market Reference Data

The following Figure 13-4 shows the UML Class Diagram for the Market Reference Data [Subscription] payloads.

The attributes are inherited from Subscription Request and Response (Figure 10-1 and Table 10-2 and Table 10-3) so are not repeated here.

²¹ Integer operations are typically much more efficient than fixed or floating point, so it is likely to be much faster to apply decimal shift on input and output rather than for more frequent comparison operations in the Execution Engine implementation.

²² 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 Markets could opt to set a different offset on each Market Segment operated out of a given substation. In this model, a Market could use an offset duration of 3 minutes to indicate that all tenders are based on three minutes after the hour.

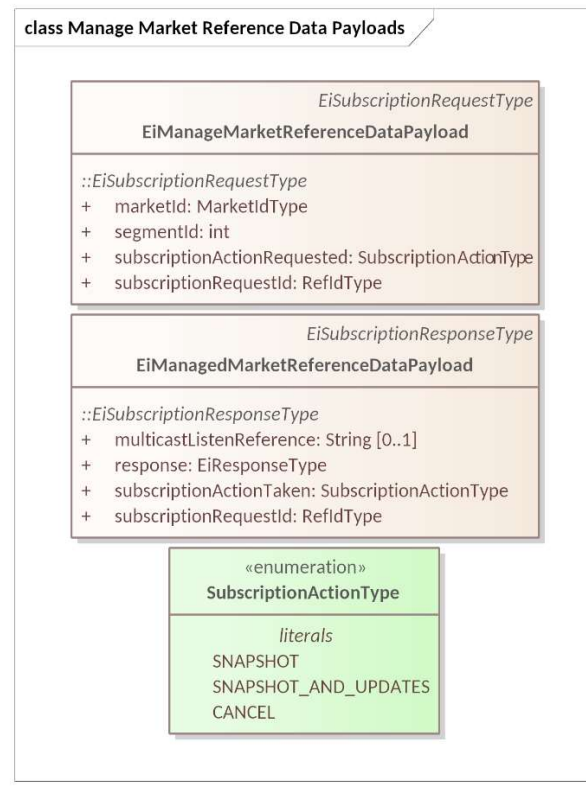


Figure 13-4 UML Class Diagram for Market Reference Data Subscription Payloads

13.213.3 Segment DefinitionReference Data

A Party must interact with a specific [Market SegmentTrading Session](#) to trade a specific Product. A Market MAY contain two or more [Market SegmentSegments](#) trading the same Product; such segments may differ in [venue typeMarket Mechanism](#), or in [trading window.schedule](#). ~~For example, a regulated provider may offer a day ahead hourly market based on an Auction between 9:00 AM and 3:00 PM. Thereafter, a forward Market in the same Product may move to another Market Segment using an Order Book. The Auction and the Order Book are different venues, matching buyer, and seller by different mechanisms.~~

A Party chooses the [Market](#) Segment that it anticipates will be to its greatest advantage. The Party will make this choice based on anticipated price, or on block size, or even on Warrant. Because Transactions are committed when created, a Party may buy on one Market Segment, and thereafter sell part of it on another.

A Party discovers Market Structure, including changes over time, by subscribing to that Market. ~~and/or its Segments.~~ Even without market activity, the information ~~inprovided by~~ a Subscription may change. For example, a Segment may open or close and the biddable Instruments change regularly.

1809 **13.3.1 Messages for Segment Reference Data**

1810 Table 13-4 Market Messages for Segment Description Reference Data

AttributeFacet	AttributeTypeRequestPayload	FIX-Field NameResponsePayload	MeaningNotes	Comments
MarketSegmentID	String	SEGMENT (1300)	Unique Identifier for Segment	
MarketSegmentName	String	SEGNAME (1396)	Optional text providing a descriptive name for a Market Segment.	While the Name MAY be displayed in a user interface; it is not meaningful to the Actors.
Product	Compound		Product transactable this Segment. See Defining Product (Section 3.1.2) for details.	Each Product shares a Resource with the Market
EndPoint	String	EndPoint	End point to access the Market Segment.	May be a transport end point, mailbox address or other.
SegmentStatus		MktSegStat	Current trading status of the Market Segment.	1 = Active: Market segment is active, i.e. trading is possible. 2 = Inactive: Market segment has previously been active and is not currently Open. 3 = Published: Market segment information is provided prior to its first activation.
TradableInterval	Interval		Instruments whose start is within the bounding of the Interval can be traded	
SessionStartTime	DateTime	TradSesStartTime	Date and Time when Tenders may first be submitted for the current or next Session	

AttributeFacet	AttributeTypeRequestPayload		FIX-Field NameResponsePayload	MeaningNotes	Comments
Session Open Time	DateTime	TradSessionTime	Date and Time Market Segment next opens (or when current or last session Opened)		
Session Close Time	DateTime	TradSessionCloseTime	Date and Time current Market Segment next Closes (or when last session Closed)		
Venue Type	String	VenueType	Description of Venue used to match and execute trades.	See Table 13-1, below.	
Residue Disposition	String	RESIDUE (Optional)	How is Residue processed. Allowable values are CANCEL / CONVERT / TRANSFER	Residue is the unfulfilled Tenders on the books when the Segment or Instrument closes. While Resource markets are typically continuously open, Individual Instruments close. i.e., 11:00 AM power closes at 11:00 AM.	
Reference Data	SuccessorEIManage Segment Reference Data Payload		StringEIManaged Segment Reference Data Payload	SUCCESSOR (Optional) The Successor is the Segment ID of the Market Successor that is the follow-on after the closing of this Segment. Messages are subclasses of the Subscription Management Messages	If Residue is "TRANSFERred" then this is the segment it is transferred to.
Maximum	Integer	MaxTradeVol	The maximum order quantity (as round lots) that can be submitted.		
Execution Instructions	String		A list of FIX Execution Instructions that are accepted in this Segment (see Table 5-4);		

AttributeFacet	AttributeTypeRequestPayload		FIX-Field NameResponsePayload	MeaningNotes	Comments
Stream Trading	Integer	MLegOK ²³ (Optional)	Applies to both Tenders and Quotes	0—Prohibited (default if missing) 1—Permitted 2—Required	
Negotiations Permitted		(Optional except Mandatory for Venue Type “Q”)	Segment supports Negotiation	0—Prohibited (default if missing) 1—Permitted 2—Private Quotes Only 3—Advertised Quotes Only	
Private Negotiations through Market	Integer	(Optional; Prohibited if missing)	Private Quotes are sent to the Segment which then forwards them to Counterparties	0—Prohibited—Private Quotes not forwarded by Segment 1—Permitted—Segment forwards Private Quotes to listed CounterParties	
Market Segment ID	Int	Market Segment ID (1300)	Identifies the Segment processing the Tender, Transaction, or Quote	This should be a unique combination paired with the Market Order ID	
Market Instrument Summary	String	(Optional)	If blank or absent, no Market Instrument Summary is available		
Max Summary Instruments	Int		0—Unlimited Instruments 1—N—Maximum Instruments in a Subscription		
Top of Book Depth	Int		0—Unlimited 1—N—Up to N levels of Book can be requested		

²³ Fix refers to a trade of multiple instruments together as a Multi-Legged Trade

AttributeFacet	AttributeTypeRequestPayload		FIX-Field NameResponsePayload	MeaningNotes	Comments
Ticker Advertisement	Int		Advertisement is publication of Tenders and Advertised Quotes as they are entered into the Book	0—Unavailable 1—Available for this segment	
Ticker Announcement	Int		Announcement of Transactions, that is, Matched Tenders and completed Negotiations in this Segment.	0—Unavailable 1—Available for this segment	
The fields below are as defined in the Tender, Table 5-2					
Duration	Duration	Not in FIX	Duration of delivery.	Part of all Instruments	
Price Scale	Int	Not Defined in FIX	A multiplier for the Price	A Market Segment may be denominated in, for example dollars or 10ths of a cent.	
Quantity Scale	Int		A scale factor on the Resource unit for this Market	For example, “mega” vs “kilo” vs “femto”	
Resource Designator	Resource Designator Enumeration	FIX Instrument component	Identifier of the Resource being offered (Optional in many markets)	While a Market only accepts Tenders and Quotes for a single Resource, the complete description is required to ensure validity and for off-market interactions.	
Warrant Ref	Int	Not in Fix	Reference to Warrants as defined in the Market	If used, see comments Warrants in Tenders, Section 5.3.3.	

13.3.2 Interaction Pattern for Segment Reference Data

Figure 13-5 shows the UML Sequence Diagram for Segment Reference Data.

See Section 10 “Subscription Facet.”

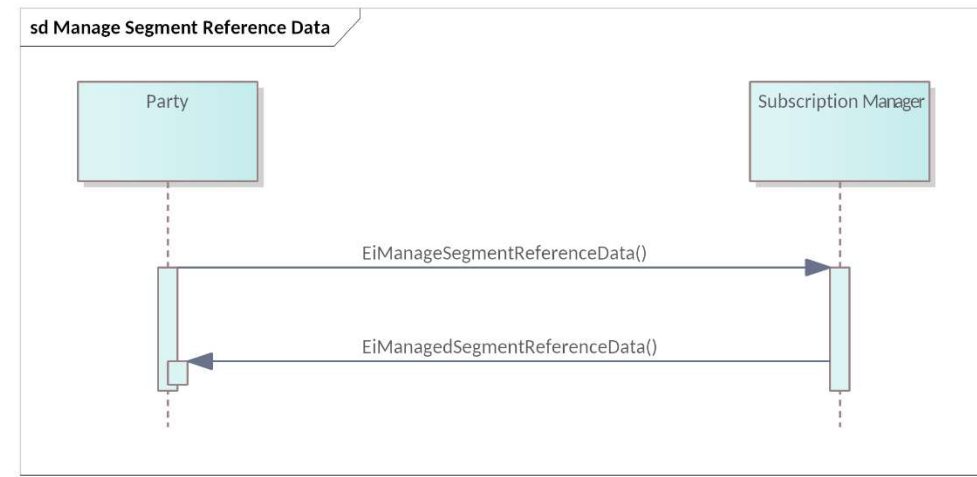


Figure 13-5 UML Sequence Diagram for Segment Reference Data

13.3.3 Information Model for Segment Reference Data

Segment Reference Data is relatively static, as Segments in typical use are long-lived.

The UML Class Diagram for Segment Reference Data is in Figure 13-6; the attribute definitions follow in Table 13-5.

class Segment Reference Data



Figure 13-6 Segment Reference Data

The following table lists the attributes in the Segment Reference Data class shown above; certain attributes are present in both the Segment Reference Data and in the Session Data (See Section 13.4 "Trading Session Data" below.

Table 13-5 Segment Reference Data

<u>Attribute</u>	<u>Attribute Type</u>	<u>FIX Field</u>	<u>Meaning</u>	<u>Comments</u>
<u>Auction at Close</u>	<u>Boolean</u>		<u>Scheduled behavior for related Sessions.</u>	<u>Related to Session second level MMT. Current session mode is in Session Data.</u>
<u>Auction at Open</u>	<u>Boolean</u>		<u>Scheduled behavior for related Sessions.</u>	<u>Related to Session second level MMT Current session mode is in Session Data.</u>
<u>Execution Instructions Accepted</u>	<u>String</u>	<u>ExecInst (18)</u>	<u>A list of FIX Execution Instructions that are accepted in this Segment (see Table 5-4).</u>	
<u>Market Depth</u>	<u>Integer</u>	<u>Market Depth (264)</u>	<u>Levels of Book that can be requested</u>	<u>0 – Unlimited 1-N – 0 – Unlimited 1-N – Up to N</u>
<u>Market ID</u>	<u>Market ID Type</u>	<u>MarketID (1301)</u>	<u>Identifies the containing market</u>	
<u>Market Instrument Summary Available</u>	<u>Boolean</u>	<u>(Optional)</u>	<u>If FALSE, no Market Instrument Summary is available</u>	
<u>Market Mechanism</u>	<u>Market Mechanism Type Enumeration</u>	<u>Analogous to Venue Type in the FIX Protocol, the Mechanism is from a separate activity of FIX.</u>	<u>Description of mechanism used to match and execute trades.</u>	<u>This is the default Mechanism Type during a Session for this Segment (for (continuous) trading).</u> <u>Sessions may use various MMT Level 2 Market Mechanisms from time to time.</u> ²⁴ <u>See Section 13.1 “Market Mechanisms” and Section 13.4 “Trading Session Data”.</u>
<u>Max Summary Instruments</u>	<u>Integer</u>		<u>0 – U Unlimited Instruments 1-N – Maximum Instruments in a Subscription</u>	<u>If Market Instrument Summary Available is False, this value is ignored.</u>

²⁴ For example, a Session might pre-open with an Auction, followed by Order Book for the bulk of the session, and end with a closing Auction.

<u>Attribute</u>	<u>Attribute Type</u>	<u>FIX Field</u>	<u>Meaning</u>	<u>Comments</u>
<u>Max Tender Quantity</u>	<u>Integer</u>	<u>MaxTradeVol (1140)</u>	<u>The maximum order quantity in units of Instruments that can be submitted.</u>	<u>FIX TradeVolType (1786) allows round lots or units n (of an instrument). The default is Units; CTS uses units of the instrument.</u> <u>This is the maximum quantity that can be tendered or a quote lifted. Some Segments MAY set different limits for different Parties.</u>
<u>Min Tender Quantity</u>	<u>Integer</u>	<u>MinQty (110)</u>	<u>The minimum number of units that can be ordered.</u>	<u>This is the minimum quantity that can be tendered or a quote lifted.</u>
<u>Negotiations Permitted</u>	<u>Boolean</u>	<u>Not in FIX</u>	<u>Segment supports Negotiation</u>	<u>(Optional except Mandatory for MMT “RQ”)</u>
<u>Negotiations Type</u>	<u>Negotiation Type Enumeration</u>	<u>(Optional except Mandatory for MMT “RQ”)</u>	<u>Segment supports the indicated style of negotiation</u>	<u>Private Quotes Only (CTS: PRIVATE ONLY)</u> <u>Public Quotes Only (CTS: PUBLIC ONLY)</u> <u>As Requested (CTS: AS_REQUESTED)</u>
<u>Price Scale</u>	<u>Integer</u>	<u>Not defined in FIX</u>	<u>A multiplier for the Price used in this segment</u>	<u>A market segment might be denominated in e.g. dollars or 10ths of a cent.²⁵</u>
<u>Private Negotiation via Market</u>	<u>Boolean</u>	<u>Not in FIX</u>	<u>Private Quotes are sent to the Segment which then forwards them to Counterparties</u>	<u>False – Prohibited – Private Quotes not forwarded by Segment</u> <u>True – Permitted – Segment forwards Private Quotes to listed CounterParties</u> <u>(FIX uses 0 to represent False, 1 to represent True)</u>
<u>Product</u>	<u>Product Type</u>	<u>Not in FIX</u>	<u>Product transactable this Segment. See Defining Product (Section 3.1.2) for details.</u>	<u>Each Product shares a Resource with the Market</u>

²⁵ In a Segment with Price Scale of 100, a trade of one unit is one one-hundredth of the intrinsic unit—trading is in tenths of a cent if the currency is USD.

<u>Attribute</u>	<u>Attribute Type</u>	<u>FIX Field</u>	<u>Meaning</u>	<u>Comments</u>
<u>Quantity Scale</u>	<u>Integer</u>	<u>Not defined in FIX</u>	<u>A scale factor for the Resource unit for this Segment</u>	<u>A multiplicative factor, e.g. 100, to convert from market quantity units to the base unit size.²⁶</u>
<u>Round Lot</u>	<u>Integer</u>	<u>RoundLot (561)</u>	<u>The trading lot size for an instrument. Chunking quantity for which a Tender may be submitted</u>	<u>For example, for Round Lot of 10, Tenders of 10 and 20 are accepted, and Tenders of 17 are rejected. This is an attribute of the Segment.</u>
<u>Scheduled Session Close Time</u>	<u>Instant Type</u>	<u>TradSes CloseTime (344)</u>	<u>Closing Time of the trading session.</u> <u>Date and Time current Session next Closes (or when last session Closed)</u>	<u>Session times may vary for different Market Mechanisms (See Section 13.1)</u> <u>FIX uses UTC Time Stamps; Instant Type is consistent with ISO 8601</u> <u>Session Data includes the Session's actual Start, Open, and Close Times; if present these are the default or typical session times.</u>
<u>Scheduled Session Open Time</u>	<u>Instant Type</u>	<u>TradSes OpenTime (342)</u>	<u>Opening Time of the trading session.</u> <u>Date and Time current Session next opens (or when current or last session Opened)</u>	<u>Session Data includes the Session's Start, Open, and Close Times; if present these are the default or typical session times.</u>
<u>Scheduled Session Start Time</u>	<u>Instant Type</u>	<u>TradSes StartTime (341)</u>	<u>Starting Time of the trading session.</u> <u>Date and Time when Tenders may first be submitted for the current or next Session</u>	<u>Session Data includes the Session's Start, Open, and Close Times; if present these are the default or typical session times.</u>
<u>Segment Desc</u>	<u>String</u>	<u>Market Segment Desc (1396)</u>	<u>Optional text providing a description for the Market Segment.</u>	<u>While the Name MAY be displayed in a user interface; it is not meaningful to the Actors.</u>

²⁶ In a Segment with a Quantity Scale of 1000, a trade of one unit is actually a trade of one one-thousandth of the intrinsic unit—if the intrinsic unit is kilowatts, a trade of 1 unit is a trade of one watt.

<u>Attribute</u>	<u>Attribute Type</u>	<u>FIX Field</u>	<u>Meaning</u>	<u>Comments</u>
<u>Segment ID</u>	<u>Integer</u>	<u>SEGMENT (1300)</u>	<u>Unique Identifier for Segment</u>	<u>This is a unique when considered with the Market ID.</u>
<u>Segment Status</u>	<u>Segment Status Type Enumeration</u>	<u>Market Segment Status (2542)</u>	<u>Current trading status of the Market Segment.</u>	<u>1 = Active: (CTS: ACTIVE) Market segment is active, i.e. trading is possible.</u> <u>2 = Inactive: (CTS: INACTIVE) Market segment has previously been active and is not currently Open.</u> <u>3 = Published: (CTS: PUBLISHED) Market segment information is provided prior to its first activation.</u>
<u>Stream Trading OK</u>	<u>Stream Trading OK Enumeration</u>	<u>Stream Trading is analogous to what FIX terms multi-leg orders, in which all instruments are [bought] or none.</u>	<u>Applies to both Tenders and Quotes</u>	<u>0 – Prohibited (default if missing) (CTS: ST_PROHIBITED)</u> <u>1 – Permitted (CTS: ST_PERMITTED)</u> <u>2 – Required (CTS: ST_REQUIRED)</u>
<u>Subscription Endpoint</u>	<u>String</u>	<u>Endpoint</u>	<u>Endpoint for subscriptions to Segment</u>	<u>May be the same as the Trade Endpoint, Segment-specific, the same across a Market, or specific to an Actor.</u>
<u>Ticker Quotes</u>	<u>Boolean</u>		<u>A Ticker is available for Quotes in this Segment</u>	<u>True – Available for this segment</u>
<u>Ticker RFQs</u>	<u>Boolean</u>		<u>A Ticker is available for RFQs in this Segment.</u>	<u>True – Available for this segment</u>
<u>Ticker Tenders</u>	<u>Boolean</u>		<u>A Ticker is available for Tenders in this Segment.</u>	<u>True – Available for this segment</u>
<u>Ticker Transactions</u>	<u>Boolean</u>		<u>A Ticker is available for Transactions in this Segment.</u>	<u>True – Available for this segment. Transactions Ticker shows Matched Tenders and completed Negotiations in this Segment.</u>

<u>Attribute</u>	<u>Attribute Type</u>	<u>FIX Field</u>	<u>Meaning</u>	<u>Comments</u>
<u>Tradeable Instrument Range</u>	<u>Interval Type</u>	<u>Not in FIX</u>	<u>Instruments whose Interval is contained in the Tradeable Instrument Range may be traded.</u>	<u>Uses the Bounding Interval pattern (See Section 3.3 “The Bounding Interval Pattern in CTS”)</u> <u>Sessions have their own Tradeable Instrument Range which may be more dynamic.</u>
<u>Trade Endpoint</u>	<u>String</u>	<u>Endpoint</u>	<u>Endpoint to access trade facets of the Segment.</u>	<u>May be Segment-specific, the same across a Market, or specific to an Actor.</u>

13.3.4 Payloads for Segment Reference Data

The payloads for managing Segment Reference Data are subclasses of the Subscription Management Messages. See Figure 13-7.

The attributes are inherited from Subscription Request and Response (Figure 10-1 and Table 10-2 and Table 10-3) so are not repeated here.

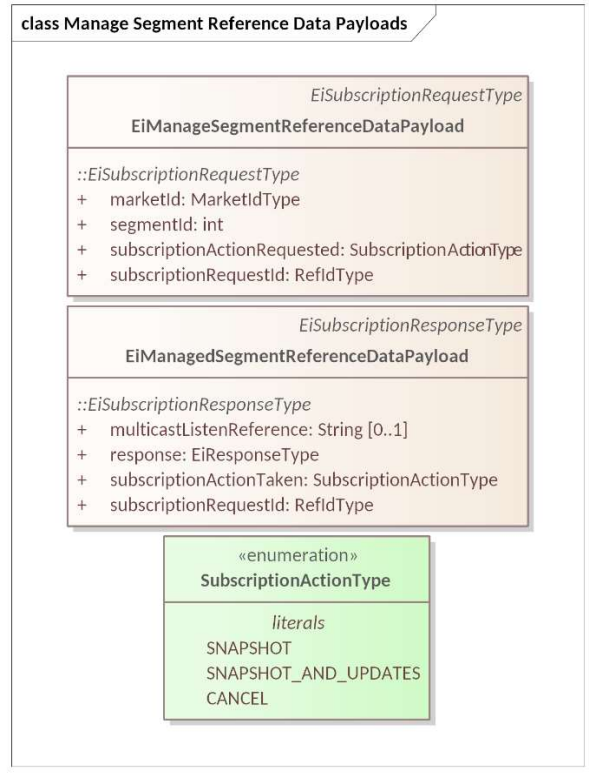


Figure 13-7 UML Class Diagram of Payloads for Segment Reference Data Subscriptions

The subscription payloads for delivery of Session Reference Data is a single Segment Reference Data Type object (Figure 13-6 and Table 13-5 Segment Reference Data).

13.4 Trading Session Data

The Market Structure Report tells the Party how to trade. Following the classification used by FIX, Market Structure Reference Data is just one part of Pre-Trade Information.

Segment Structure includes Opening, Closing, as well as Crossing information for specific Instruments. It also includes detailed information to guide trading, negotiation, and settlement (wherein the difference between market Position and measured Delivery is settled).

13.4.1 Messages for Trading Session Data

Table 13-6 Messages for Trading Session Data

Facet	Request Payload	Response Payload	Notes
Reference Data	EiManage Session Data Payload	EiManaged Session Data Payload	Messages are subclasses of the Subscription Management Messages

13.4.2 Interaction Pattern for Trading Session Data

Trading Session Data is very dynamic, and includes (e.g.) information on planned and unplanned closures, auctions, and more. It follows the Subscription pattern—see Section 10 “*Subscription Facet*.”

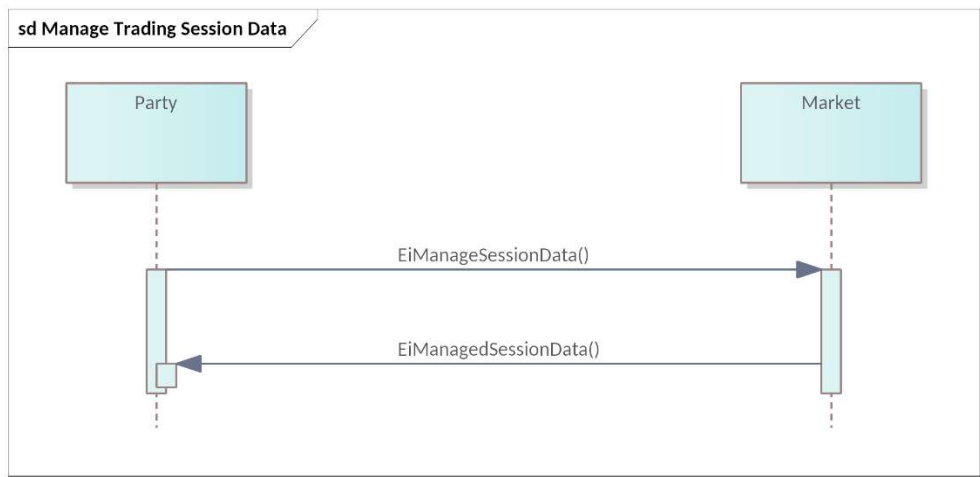


Figure 13-8: UML Sequence Diagram for Manage Trading Session Data

A Party may watch changes to a single Session by naming that Session in the subscription request. This will return only that Session and updates to that Session.

13.4.3 Information Model for Trading Session Data

Figure 13-9 shows the UML Class Diagram for Session Data and the enumerations used.

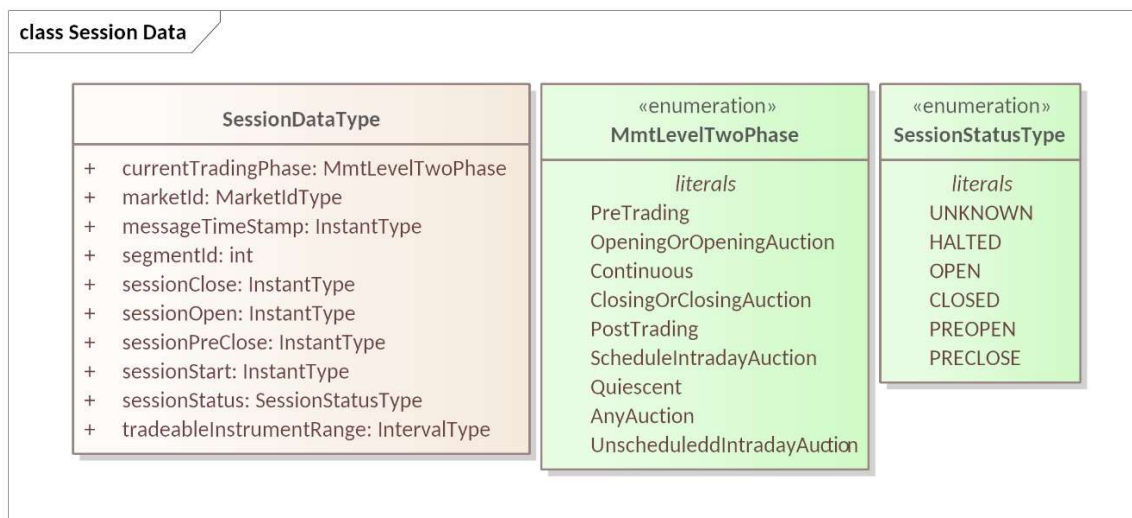


Figure 13:9 UML Class Diagram for Session Data

The attributes for Session Data are in Table 13-7.

Table 13-7 Session Data

Attribute	Attribute Type	FIX Field	Meaning	Comments
Current Trading Phase	MMT Level Two Phase Enumeration	TradingSessionSubID (625)	Active trading phase for the session. The Enumeration is based on the description of FIX Codes 1 through 9 and describes the same trading phases.	The values used in CTS are <ul style="list-style-type: none"> PreTrading Opening Or Opening Auction Continuous [trading] Closing Or Closing Auction PostTrading Scheduled IntraDay Auction Quiescent Any Auction Unscheduled Intraday Auction
Market ID	Market ID Type	MarketID (1301)	Identifies the containing market	
Message Time Stamp	Instant Type	Not in FIX	The timestamp for when the Session Data was produced.	
Segment ID	Integer	MarketSegmentID (1300)	Identifies the containing Segment	This is unique when paired with the Market ID

Attribute	Attribute Type	FIX Field	Meaning	Comments
Session Close	Instant Type	TradSesCloseTime (344)	Closing time of the trading session	Session times may vary for different Market Mechanisms (Section 0)
Session Open	Instant Type	TradSesOpenTime (342)	Time of the opening of the trading session	
Session PreClose	Instant Type	TradSesPreCloseTime (343)	Time of the pre-close of the trading session	
Session Start	Instant Type	TradSesStartTime (341)	Starting time of the trading session	
Session Status	Session Status Type Enumeration	TradSesStatus (340)	The status of this session (from FIX code set)	Values are <ul style="list-style-type: none">UnknownHaltedOpenClosedPreOpenPreClose
Tradeable Instrument Range	Interval Type	Not in FIX	Instruments whose Interval is contained in the Tradeable Instrument Range may be traded.	See Section 3.3 “The Bounding Interval Pattern in CTS” Segments have their own Tradeable Instrument Range which may be less dynamic.

13.4.4 Payloads for Trading Session Data

The payloads for Session Data are those of the Subscription Management Messages. See Figure 13-10. The attributes are inherited from Subscription Request and Response (Figure 10-1 and Table 10-2 and Table 10-3) so are not repeated here.

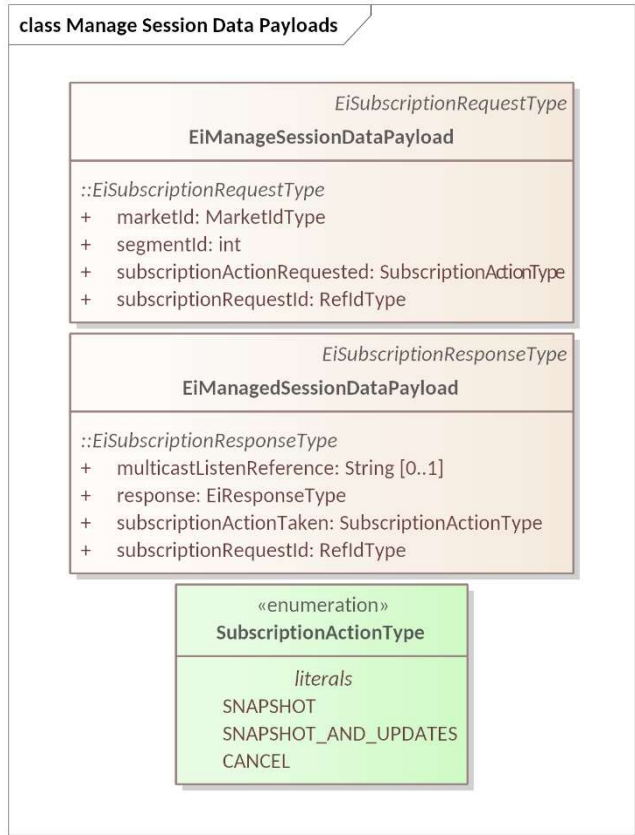


Figure 13-10

Bindings

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

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

13.3 JSON

PENDING — JSON Schema awaiting stable payload definitions

13.4 XML Schema

PENDING — XML Schema awaiting stable payload definitions

13.4.1 XML Namespaces

PENDING — XML Namespaces awaiting XML Schema

13.5 Simple Binary Encoding

TODO — SBE Schema awaiting stable payload definitions; [UML Class Diagram of the Payloads for Manage Session Data](#)

14 Conformance

14.1 Introduction to Conformance

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

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

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

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

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

This specification simplifies WS-Calendar Schedule Streams and Signals [Streams] as CTS Streams (see Section 3.2) and refactors the TEMIX profile of **[EI]**.

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

14.2 Claiming Conformance to Common Transactive Services

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

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

14.3 Annex: Conformance statements from Spec not yet incorporated into this section

14.3.1 Conforming with Use of Warrants in Tenders

Warrants increase the specificity of Product (and Instrument). A Buyer who does not specify a Warrant will be satisfied Delivery of a Product whether or not it has a Warrant. A Buyer who requests Product with a Warrant will only be satisfied by Delivery of a Product that has that Warrant.

Consider a buyer who wishes to buy a package of coffee beans and a buyer who wishes to buy a package of organic coffee beans. The word "Organic" on the label serves as a Warrant. The first buyer will buy solely on price, and is indifferent to seeing the word "Organic" on the label. The second buyer will choose only from among those packages with the warrant "Organic" on the label.

When a Tender on the Buy side specifies a Warrant, it must be rejected by any Market Segment that does not include that Warrant. A Tender on the Sell side that specifies a Warrant may be accepted by any Segment where the same Resource and Duration are traded. Conversely, a Tender on the Sell side without a Warrant must be rejected by any Segment that specifies a Warrant.

Appendix A. References

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

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

A.1 Normative References

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

~~NOTE: INSERT AS FORMATTED REFERENCES. Consider [EI]~~

[CAL-MIN]

WS-Calendar Minimal PIM-Conformant Schema Version 1.0. Edited by William Cox and Toby Considine. 26 August 2016. OASIS Committee Specification. <http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.html>

[CAL-PIM]

OASIS WS-Calendar Platform-Independent Model version 1.0, Committee Specification 02 Edited by William T. Cox and Toby Considine, 21 August 2015. <http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/cs02/ws-calendar-pim-v1.0-cs02.html> Latest version: <http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html>

[EI]

Energy Interoperation Version 1.0. Edited by Toby Considine, 11 June 2014. OASIS Standard. <http://docs.oasis-open.org/energyinterop/ei/v1.0/os/energyinterop-v1.0-os.html> Latest version: <http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html>. and its TeMIX Profile

[EMIX]

OASIS Energy Market Information Exchange (EMIX) Version 1.0 Committee Specification 02 Edited by Toby Considine, 11 January 2012. <http://docs.oasis-open.org/emix/emix/v1.0/cs02/emix-v1.0-cs02.html> Latest version: <http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html>

[JSON]

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

[MMT]

FIX Trading Community Market Model Typology v4.2, retrieved July 2, 2024. <https://www.fixtrading.org/mmt/>

[RFC8174]

Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<http://www.rfc-editor.org/info/rfc8174>>.

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

[RFC2246]

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

[SBE]

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

1970 **[Streams]**
1971 *Schedule Signals and Streams Version 1.0*. Edited by Toby Considine and William T. Cox. 18 September
1972 2016. OASIS Committee Specification. <http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams->
1973 [v1.0.html](http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html).

1974 **A.2 Informative References**

1975 The following referenced documents are not required for the application of this document but may assist
1976 the reader with regard to a particular subject area.

1977 **[Actor Model]**

1978 C. Hewitt, "Actor Model of Computation: Scalable Robust Information Systems," arxiv.org, 2010.

1979 **[Fractal Microgrids]**

1980 Art Villanueva et al, *Camp Pendleton Fractal Microgrid Demonstration*, California Energy Commission
1981 Report CEC-500-2016-013,j available at
1982 [http://400.sydneyplus.com/CaliforniaEnergy_SydneyEnterprise/Download.aspx?template=Books&field=PublicURL&record=57483797-a40e-49e7-b675-2858a3ad0d91&showSave=False&repeat=d4e63b56-](http://400.sydneyplus.com/CaliforniaEnergy_SydneyEnterprise/Download.aspx?template=Books&field=PublicURL&record=57483797-a40e-49e7-b675-2858a3ad0d91&showSave=False&repeat=d4e63b56-27d1-4476-9300-7ede86a533ca)
1983 [27d1-4476-9300-7ede86a533ca](http://400.sydneyplus.com/CaliforniaEnergy_SydneyEnterprise/Download.aspx?template=Books&field=PublicURL&record=57483797-a40e-49e7-b675-2858a3ad0d91&showSave=False&repeat=d4e63b56-27d1-4476-9300-7ede86a533ca)
1984

1985 **[Framework]**

1986 National Institute of Standards and Technology, *NIST Framework and Roadmap for Smart Grid*
1987 *Interoperability Standards, Release 1.0*, January 2010,
1988 http://nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf

1989 **[CTS2016]**

1990 W.T. Cox, E. Cazalet, E., A Krstulovic, W Miller, & W.Wijbrandi *Common Transactive Services*. TESC
1991 2016. Available at
1992 [http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/Common%20Transactive%20](http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/Common%20Transactive%20Services%20Paper%2020160516.pdf)
1993 [Services%20Paper%2020160516.pdf](http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/Common%20Transactive%20Services%20Paper%2020160516.pdf)

1994 **[EML-CTS]**

1995 Energy Mashup Lab Common Transactive Services (open-source software)
1996 <https://github.com/EnergyMashupLab/eml-cts>)

1997 **[FIXIMATE]**

1998 [FIXimate FIX Interactive Message And Tag Explorer](http://fiximate.fixtrading.org/)
1999 <http://fiximate.fixtrading.org/>

2000 **[FSGIM]**

2001 *Facility smart grid information model*. ISO 17800. <https://www.iso.org/standard/71547.html> 2017

2002 **[iCalendar]**

2003 B. Desruisseaux, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*,
2004 <https://tools.ietf.org/html/rfc5545>. 2009,
2005 See also
2006 C. Daboo & M. Douglas. *Calendar Availability*, <https://tools.ietf.org/html/rfc7953>, 2016

2007 **[GridFaultResilience]**

2008 W.T. Cox & T. Considine. *Grid Fault Recovery and Resilience: Applying Structured Energy and*
2009 *Microgrids*. IEEE Innovative Smart Grid Technologies 2014. Available at
2010 [http://coxsoftwarearchitects.com/Resources/ISGT_2014/ISGT2014_GridFaultRecoveryResilienceStructur](http://coxsoftwarearchitects.com/Resources/ISGT_2014/ISGT2014_GridFaultRecoveryResilienceStructuredMicrogrids_Paper.pdf)
2011 [edMicrogrids_Paper.pdf](http://coxsoftwarearchitects.com/Resources/ISGT_2014/ISGT2014_GridFaultRecoveryResilienceStructuredMicrogrids_Paper.pdf)

2012 **[IEC62746-10-3] International Standard.**

2013 Systems interface between customer energy management system and the power management system -
2014 Part 10-3: Open automated demand response - Adapting smart grid user interfaces to the IEC common
2015 information model, <https://webstore.iec.ch/publication/59771> 2018.

2016 **[Micromarkets]**
 2017 W.T. Cox & T. Considine, *Energy, Micromarkets, and Microgrids*.
 2018 GridInterop 2011, https://www.gridwiseac.org/pdfs/forum_papers11/cox_considine_paper_gi11.pdf
 2019 [RFC3552]
 2020 E Rescorla & B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC
 2021 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.

2022 **[SmartGridBusiness]**
 2023 T. Considine & W.T. Cox, *Smart Loads and Smart Grids—Creating the Smart Grid Business Case*. Grid-
 2024 Interop 2009. Available at [http://coxsoftwarearchitects.com/Resources/Grid-](http://coxsoftwarearchitects.com/Resources/Grid-Interop2009/Smart%20Loads%20and%20Smart%20Grids.pdf)
 2025 [Interop2009/Smart%20Loads%20and%20Smart%20Grids.pdf](http://coxsoftwarearchitects.com/Resources/Grid-Interop2009/Smart%20Loads%20and%20Smart%20Grids.pdf)

2026 **[StructuredEnergy]**
 2027 *Structured Energy: Microgrids and Autonomous Transactive Operation*,
 2028 http://coxsoftwarearchitects.com/Resources/ISGT_2013/ISGT-Cox_StructuredEnergyPaper518.pdf.
 2029 Innovative Smart Grid Technologies 2013 (IEEE).

2030 **[TPB-EI]**
 2031 Transport Protocol Bindings for OASIS Energy Interoperation 1.0 Version 1.0. 01 October 2012. OASIS
 2032 Committee Specification 01.
 2033 <https://docs.oasis-open.org/energyinterop/tpb-ei/v1.0/tpb-ei-v1.0.pdf> .

2034 **[TeMIX]**
 2035 TeMIX Transactive Energy Market Information Exchange [TeMIX] an approved Note of the EMIX TC. Ed
 2036 Cazalet et al., 23 May 2010. [http://www.oasis-open.org/committees/download.php/37954/TeMIX-](http://www.oasis-open.org/committees/download.php/37954/TeMIX-20100523.pdf)
 2037 [20100523.pdf](http://www.oasis-open.org/committees/download.php/37954/TeMIX-20100523.pdf)

2038 **[TransactiveMicrogrids]**
 2039 Jennifer M. Worrall, Edward G. Cazalet, PhD, William T. Cox, PhD, Narayanan Rajagopal, Thomas
 2040 R. Nudell, PhD, and Paul D. Heitman, *Energy Management in Microgrid Systems*, TESC 2016. Available
 2041 at
 2042 [http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/tes2016_microgrids_paper_Fi](http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/tes2016_microgrids_paper_Final.pdf)
 2043 [nal.pdf](http://coxsoftwarearchitects.com/Resources/TransactiveSystemsConf2016/tes2016_microgrids_paper_Final.pdf)

2044 **[TRM] (Transactive Resource Management)**
 2045 B. Huberman and S. H. Clearwater, *Thermal markets for controlling building environments*, Energy
 2046 Engineering, vol. 91, no. 3, pp. 26- 56, January 1994.

2047 **[UML]**
 2048 Object Management Group, *Unified Modeling Language (UML)*, V2.4.1, August 2011.
 2049 <http://www.omg.org/spec/UML/2.4.1/>

2050 **[XSD]**
 2051 *W3C XML Schema Definition Language (XSD) 1.1*. Part 1: Structures, S Gao, C. M. Sperberg-McQueen,
 2052 H Thompson, N Mendelsohn, D Beech, M Maloney <http://www.w3.org/TR/xmlschema11-1/>, April 2012,
 2053 Part 2: Datatypes, D Peterson, S Gao, A Malhotra, C. M. Sperberg-McQueen, H Thompson, P Biron,
 2054 <http://www.w3.org/TR/xmlschema11-2/> April 2012

2055 **[ZeroTrust]**
 2056 Zero Trust Architecture, S Rose, O Borchert, S Mitchell, S Connelly. NIST Special Publication 800-207,
 2057 <https://doi.org/10.6028/NIST.SP.800-20> August 2020
 2058

Appendix B. Choosing a Market Mechanism

A Market may consist of several segments. Segments differ chiefly in the Products traded and in the Mechanism they use for trading. Market Participants will select different Segments based in part on how well the Market Mechanism in that Segment supports its goals.

The authors of CTS cannot specify which market mechanism to use in every situation. This non-normative section discusses each named Market Mechanism and describes some scenarios in which it might be used.

See Section 13.1 “Market Mechanisms” for the normative description of market mechanisms.

B.1 Central Limit Order Book (LB): Simple Bids & Offers

The central limit order book, also known colloquially simply as an order book, is the simplest market for many to understand. Would-be buyers submit bids which the market immediately executes or writes in the book. Would-be sellers submit offers which the market writes in the book. The order book represents the collective actions of buyers and sellers who place orders to buy or sell an asset at a specific price.

The order book continuously updates as new orders are added or existing ones are matched or canceled. Transactions are created when the Segment matches compatible buy orders and sell orders. Matches are made based on price-time priority, where orders are filled based on the highest bid and the lowest ask, and when these prices overlap, the order that was placed first gets priority.

When a submitted order is matched against one or more orders on the book, trades are executed, and the order book is updated to reflect the new supply and demand levels.

In summary, the order book market matching process is a continuous cycle of order formation, order matching, and price discovery, driven by the interactions of buyers and sellers in the marketplace.

B.2 Periodic Auctions (PA)

Participants in periodic auctions submit bids and offers up until a published deadline. After the deadline, all tenders are evaluated and a common price is determined, e.g. at which the greatest volume can be executed. All bids (tenders to buy) above that common price are accepted, and all offers (tenders to sell) below that price are accepted. All transactions clear at the common price. Any remaining Tenders are referred to as the Residual.

The North American bulk power markets are run largely through periodic auctions (e.g., with transactions in day-ahead markets announced the day before) to enable large generators to schedule their operations.

Periodic auctions are also referred to “Double Auctions”, that is, auctions in which both sellers and buyers submit bids.

B.3 Quote-Driven Markets (QB)

Quote-driven markets typify markets with dominant suppliers or market makers providing additional liquidity by offering to buy and/or sell at any time. The price of the resource is determined and announced by the dominant suppliers. The dominant suppliers MAY represent many third parties, i.e., a distribution system operator (DSO) acting as an intermediary to a bulk power market.

Quote-Driven Markets permit partial lifting, that is a Party may issue a Tender for 7kWh in response to a Quote for 150 kWh. Quote Driven Markets do not normally process Rejections or accept counter-Quotes as a Quote Response; Quotes are issued on a take-it-or-leave-it basis.

In typical FIX Protocol-based markets, quotes are non-negotiable and are valid for a short time, perhaps three seconds. In existing power markets using quotes for day-ahead markets, they may be available for hours. In either case, if buyers take all of the quantity in a quote, a dominant supplier has the option of submitting a new quote, potentially at a different price.

A common example of a quote-driven market is one in which an electric utility announces 24 hourly prices for the next day. These are good until a certain time and indicate the maximum quantity that the issuer is willing to sell at that price. In CTS, these will be tradeable quotes. A buyer or seller submits a Quote Response containing a Tender which can automatically match against a tradeable quote.

This pattern is useful because at the limit, all resource markets are limited by the maximum potential rate of resource delivery. Once that limit is reached, the market maker may avoid all further transactions by not entering additional quotes.

In some regulated electricity markets, the price received for selling power is less than the price paid for buying power, establishing a spread. A market maker may opt to publish a range of quotes to buy as well and benefit from the spread by buying and selling a resource at different times as the prices fluctuate.

Quote driven CTS markets are typically highly regulated markets.

B.4 Request for Quote Markets (RQ)—Negotiating

Request for quote markets support bilateral negotiations around price and quantity. Interactions begin with a request for quote (RFQ), which may be vague as to prices, quantities, or even the price schedule of the Instruments. The recipient of an RFQ may reply with one or more indicative or tradeable quotes, perhaps for different delivery times or quantities. When an acceptable tradeable Quote is received, the party wishing to lift that Quote responds with a Quote Response that notifies the Market of the acceptance. The Market then generates a Transaction.

The Negotiation process is inherently flexible. A Transaction may come after many rounds of negotiation, or directly from a response to the first tradeable quote. This section describes some potential interactions to clarify the concepts.

An RFQ Market can permit large buyers to plan significant resource use over time, for example, scheduling a long running industrial process which also requires labor planning. Such a buyer could submit multiple Requests for Quotes with different schedules, and then select from among the Quotes received in response.

An RFQ uses a Bounded Interval to indicate what an acceptable Quote would be.

- Consider Party A that wishes to buy 15 kW of power over a two-hour period, sometime within an 8-hour window. This would take the form of submitting an RFQ to Party B with an eight-hour Bounding Interval with a specific start time, but with a Stream of two Intervals with a Duration of 1 hour but with no starting time specified.
- Consider instead that Party A further wishes to buy 10 kW of energy over an hour at \$0.05/kWh sometime during the work day. Party A can issue an RFQ, with a bounding duration of the workday, containing a single unscheduled Interval of one hour containing the Price and the Quantity.

Party A and B can send these RFQs directly to one or more potential counterparties or published to the entire market. Because it is not tradeable, a RFQ does not need to be submitted to the Market and the Segment does not need to register the RFQ. The response is a quote, either indicative or tradeable.

Party A may receive one or more offers in response. These become more specific, perhaps two Quotes issued by the same counterparty with different prices at different times. A quote issuer may make a counteroffer by sending a quote proposing different quantities and/or prices. Perhaps the responding Party considers that it will turn on a generator, but only if it can operate the generator at an economic rate for an economic duration. A quote MAY be for only one of the two hours indicated in the original RFQ, leaving the requesting Party to find an acceptable match from among all the offers (quotes) it receives. The prices may be higher or lower than requested in the original RFQ. Until one party issues a tradeable Quote, all responses are RFQs or indicative Quotes, issued to continue the negotiation.

When either Party thinks that there is an essential meeting of requirements, that Party submits a tradeable Quote, that is a quote that a matching Tender will turn into a Transaction. For a CTS quote to be tradeable, the Party informs the Segment, even if it is a private quote and not published.

To accept a tradeable Quote, a Party submits a Tender to the Segment, referencing the Quote ID, and matching the details of the quote. The market mechanism compares the Tender to the quote, and, if they

match, it executes the Transaction. All tradeable quotes are treated as if they are marked All-or-None (AON).

The issuer normally accepts a Tender received in response to a tradeable Quote, with exceptions for expiration or for another Tender having gotten to the Segment first.

The issuer MUST accept a Tender received in response to a tradeable Quote.

Negotiations may include Interval Quotes or Stream Quotes, a pattern that matches that of Tenders (See Section 5.3.1, "Interval Tenders and Stream Tenders.") A Stream Quote must be matched to a Stream Tender in the Quote Response to create a Transaction. A requester that wishes to acquire a power curve indicates this with a Stream Quote back indicates it in the RFQ or indicative Quote. Note: the response does not need to match the request; the Indicative or Tradeable Quote received in response may propose a different Stream.

Below are three non-normative scenarios for negotiation to illustrate the flexibility of Negotiation: (1) Single-provider, (2) over-the-counter, and (3) system recovery.

- 1) Parties may choose to use an negotiated trade because they wish to bypass certain market restrictions. For example, consider a Party wishing to buy 87 kWh of Power for a period over 1 hour and 5 minutes beginning 15 minutes after the hour. Parties negotiate as above, come to terms, as above, and the Segment records the Transaction. Order Book Segments impose restrictions on Round Lots and Intervals to improve Market Liquidity, that is, the likelihood of a match between a Tender to Buy and a Tender to Sell. If Parties already have made an agreement, then there is no need to improve liquidity. This makes the Durations and Round-Lots in negotiated markets indicative rather than prescriptive. 87 kWh is a rough match for Off-Market Segment with a 20 kWh round lot—a gWh is not. In a comparable way, the quote's Duration is a rough match for Segment with an Hour Duration while 3 minutes is not.
- 2) Markets commonly project opening prices for Instruments before they open. If a system recovery requires a market re-start, there may be no good information to set opening prices. Prior to a re-start, a Segment may publish RFQs to buy and to sell. The Operator may use a Segment to probe the potential market in this way several times, perhaps with different prices and quantities, to discover an indicative opening price at which the Resource will be in rough balance. (This rough balancing MAY be by an implied auction.) When the market has enough information, then the market opens a Segment for trading, announcing the indicative opening prices for each Instrument.

This specification does not require that a Market segment include any of the scenarios described above. We include them to illustrate how the essential components of Negotiation might fit together in a specific system.

B.5 Market mechanisms not defined in FIX MMT

As traditional regulated and centrally-managed markets migrate to TE, CTS supports some mechanisms not defined in the FIX Market Model Typology (MMT).

B.5.1 Off-Book segment (OB)

Off Book mechanisms are reserved in CTS for direct allocation of Resource from one Party to another by a process external to the Market. Parties are notified through the receipt of trade notices (Transactions).

A transactive resource market may be used to balance resource flows within a microgrid or other local distribution system. Markets solve the knowledge problem of balancing supply and demand over time even when all parties or systems have a common owner.

A common scenario, say on a campus or base, is to handle scarce resources through direct assignment to one of the parties. Consider a campus of 10 buildings and a hospital. The owner may wish to create a Transaction in which each of the buildings transfers 10 kWh to the hospital which receives 100 kWh into its position. The donor buildings must then trade within their own accounts to rebalance supply and demand, or re-balance operations to stay within their new position.

2201 [On military bases, this can be referred to as power following command intent.](#)

2202 **[B.5.2 Real Time Pricing \(RT\)](#)**

2203 [Price quotes are broadcast for each Interval, but the Segment has no mechanism for negotiations or](#)
2204 [Tenders. Transactions are generated later by reading Delivery and generating transactions in a self-](#)
2205 [executing Segments.](#)

2206 **[B.5.3 Spot Market \(SP\)](#)**

2207 [A Ticker in a spot market indicates the "instant" price in the Segment indicating the Price for purchases or](#)
2208 [sales. A spot market Segment may limit active trading to a small window of time.](#)

2209 [A spot market segment MAY accept Tenders to sell as the market maker tries to pull a resource back into](#)
2210 [the market to address a looming shortfall.](#)

2211 [A spot market may support an asymmetry of self-execution, perhaps creating transactions for un-planned](#)
2212 [consumption but not for un-planned sales.](#)

2213 **[B.5.4 Self Executing \(SX\)](#)**

2214 [A self-executing Segment creates Trades to align with consumption as measured at the meter and](#)
2215 [reported by the Delivery facet. Self-execution generates Transactions for Resource consumed without](#)
2216 [previously being bought. Self Execution aligns the Position known to the Market with the amount](#)
2217 [consumed as indicated by Delivery.](#)

2218 [A self-executing Segment is needed to augment any other market mechanism so long as the Resource](#)
2219 [delivered to a customer is not limited to the amounts transacted in advance; in today's power markets, the](#)
2220 [power delivered will not be limited to the customer's market position.](#)

Appendix B. ~~Appendix C.~~ Security and Privacy Considerations

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

~~B.1C.1~~ CTS and Security Considerations

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

- A price that is falsely low may cause the buyer to operate a system when there is inadequate power, potentially harming systems within a facility, or harming other facilities on the same circuit.
- A price that is falsely low may cause the seller to leave the market.
- A price that is falsely high may cause the buyer to shut down operation of systems or equipment.
- A price that is falsely high may cause the seller to increase operations when there is neither a ready consumer nor perhaps even grid capacity to take delivery.

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

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

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

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

~~B.2C.2~~ CTS and Privacy Considerations

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

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

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

linked to increased security risks criminals may be able to access the data and use the information to enable inferences about what people are doing in their home or if they are away from home.

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

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

- 1) The Collection Limitation Principle. There should be limits to the collection of personal data and any such data should be obtained by lawful and fair means and, where appropriate, with the knowledge or consent of the data subject.
- 2) The Data Quality Principle. Personal data should be relevant to the purposes for which they are to be used and, to the extent necessary for those purposes, should be accurate, complete and kept up to date.
- 3) The Purpose Specification Principle. The purposes for which personal data are collected should be specified not later than at the time of data collection and the subsequent use limited to the fulfillment of those purposes or such others as are not incompatible with those purposes and as are specified on each occasion of change of purpose.
- 4) The Use Limitation Principle. Personal data should not be disclosed, made available or otherwise used for purposes other than those specified, except a) with the consent of the data subject, or b) by the authority of law.
- 5) The Security Safeguards Principle. Personal data should be protected by reasonable security safeguards against such risks as loss or unauthorized access, destruction, use, modification or disclosure of data.
- 6) The Openness Principle. There should be a general policy of openness about developments, practices and policies with respect to personal data. Means should be readily available of establishing the existence and nature of personal data and the main purposes of their use, as well as the identity and usual residence of the data controller.
- 7) The Individual Participation Principle. An individual should have the right:

to obtain from a data controller, or otherwise, confirmation of whether or not the data controller has data relating to him.

to have data relating to him communicated to him, within a reasonable time, at a charge, if any, that is not excessive; in a reasonable manner, and in a form that is readily intelligible to him.

to be given reasons if a request made under subparagraphs (a) and (b) is denied and to be able to challenge such denial; and

to challenge data relating to him and, if the challenge is successful, to have the data erased, rectified, completed or amended.

- 8) The Accountability Principle. A data controller should be accountable for complying with measures which give effect to the principles stated above.

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

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

The system developer should keep the privacy principals in mind when making specific technology choices. For example, messages between an actor and the market MAY be encrypted to protect the privacy of people represented by individual actors. While the transactive energy market must know both

2318 buyers and sellers to support transactions and settlements, the developer should take steps to guard that
2319 information. A developer may opt that each notice of contract sent to an actor always has a counterparty
2320 of the market, so as to protect the sources and uses of electricity.

2321 It is beyond the scope of this specification to specify security practices and privacy design for markets
2322 built using this specification.

2323

~~C.1D.1~~ Appendix D. Semantic Composition from Energy Interoperation, EMIX, and WS-Calendar

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

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

- EMIX describes price and product for electricity markets.
- WS-Calendar communicates schedules and sequences of operations. CTS uses the [Streams] optimization, which is a standalone specification, rather than part of EI 1.0.
- EI uses the vocabulary and information models defined by those specifications to describe the services that it provides. The payload for each EI service references a product defined using [EMIX]. EMIX schedules and sequences are defined using [WS-Calendar]. Any additional schedule-related information required by [EI] is expressed using [WS-Calendar].
- Since [EI] was published, a semantically equivalent but simpler [Streams] specification was developed in the OASIS WS-Calendar Technical Committee. CTS uses that simpler [Streams] specification.

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

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

In CTS, we group and name these elements as a Resource, Product, and Instrument. These terms are defined in Section 2.2.4, “Markets and [Market Segments](#)”

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

~~C.1D.1~~ C.2D.1 Conformance with Energy Interoperation

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

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

~~C.2D.2~~ C.2D.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 defines Market Context, a URI used as the identifier of the Market. EMIX further defines Standard Terms as retrievable information about the Market that an actor can use to configure itself for interoperation with a given Market. We extend and clarify those terms, provide an extension mechanism, and discuss the relationship of markets, market segments, and products.

C.3D.3 Conformance with WS-Calendar Streams

WS-Calendar expresses events and sequences to support machine-to-machine (M2M) negotiation of schedules while being semantically compatible with human schedules as standardized in [iCalendar]. Schemas in [WS-Calendar] support messages that are nearly identical to those used in human schedules. We use a conformant but simpler and more abstract Platform Independent Model [CAL-PIM] and the [Streams] compact expression²⁷, to support telemetry (Delivery Facet) and series of Tenders while not extending the semantics of [Streams].²⁸

WS-Calendar conveys domain specific information in a per-event payload within a schedule-centric message; in CTS, the domain is the price, product, and quantity. An essential concept of WS-Calendar is inheritance, by which a starting time can be applied to an existing message, or by which all events in a sequence share common information such as duration. Inheritance is used to “complete” a partial message during negotiation. CTS makes use of this to apply a common product across a sequence, or to convey a specific starting time to a market product.

C.4D.4 CTS and WS-Calendar Streams

The [Streams] specification describes how to handle repeating time series of similar data, applying repeating information to a series of schedulable intervals, expressing common information once for the series, overriding the common information only if needed within a specific interval, and potentially scheduling (“binding”) the entire series by adding a starting date and time to one of the Intervals.

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

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

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

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

The CtsStream follows this pattern. The elements from [Streams] have been flattened into the CTS Stream, and the Stream Interval and payload flattened into a streamPayloadValue and the internal local UID for the stream element.

²⁷ Simplified as CTS Streams in this specification.

²⁸ Some specifications (e.g. [FSGIM]) have extended the basic [Streams] capabilities, but this brings additional complexity which does not benefit our use cases.

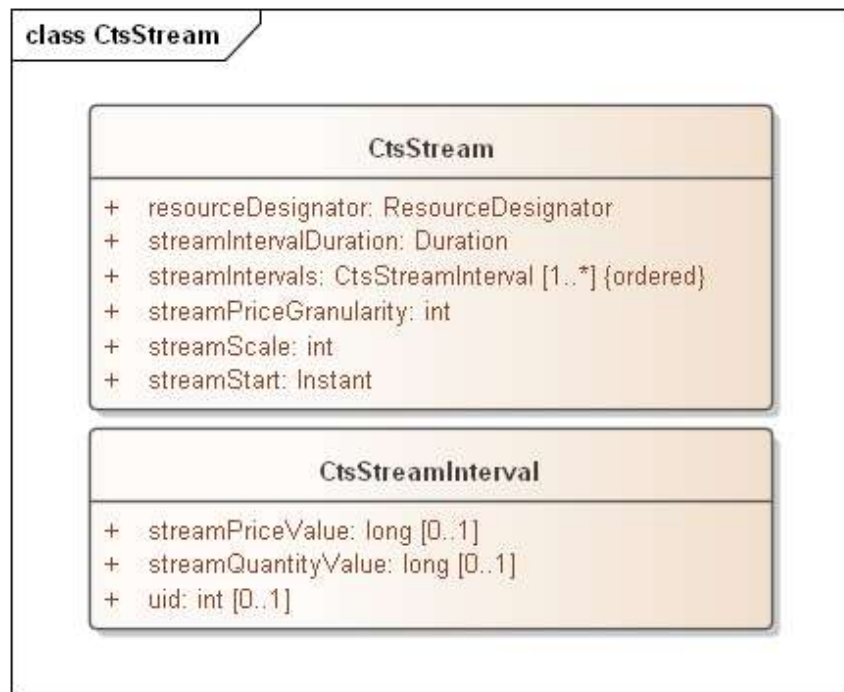


Figure C-14-1: CtsStreamDefinition

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

Appendix D. Conformance to the TEMIX Profile of Energy Interoperation

TBD

Appendix E. 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--14-1 Abbreviations and Terms used throughout this document

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

Appendix F. Acknowledgments

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

Portions of models and text is derived from The Energy Mashup Lab open-source project, EML-CTS and is used under terms of the Apache 2.0 License for that project.²⁹

F.1 Participants

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

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

F.2 Special Thanks

The Technical Committee extends a special thanks to Hanno Klein, co-chair of the FIX Global Technical Committee, and Senior Standards advisor at FIXdom. Hanno's patient explanations of trading semantics and suggestions for approaches that would increase the commonality of financial markets and CTS markets were invaluable. His knowledge of global, open and free standards for trading is unparalleled. Where we missed our mark, it is where we misunderstood his advice.

²⁹ <https://github.com/EnergyMashupLab/eml-cts>

Appendix G. Revision History

Revision	Date	Editor	Changes Made
WD01	2/15/2021	Toby Considine	Initial reformatting and conversion of the specification contributed by The Energy Mashup Lab to create a document for committee work.
WD02	3/3/2021	Toby Considine	Added prose definitions of Resource, Product, and Instrument
WD03	4/5/2021	Toby Considine	Simplified introductory material, raised message type to earlier in document. Removed some repetitive material. Revised UML required.
WD04	5/7/2021	Toby Considine David Holmberg William T Cox	Reordered intro material to reduce repetition, Reference Actor Model more consistently, Revise and re-factor Resource/Product/Instrument Add Section 3 to elevate common semantic elements
WD05	5/25/2021	Toby Considine David Holmberg William T Cox	Continues clean-up and condensation of sections 1, 2
WD06	6/7/2021	Toby Considine	Refines Item language into Resource and Products. Explains Message Groups as a conforming descendant of EI Services.
WD07	6/21/2021	Toby Considine William T Cox	Clarified terminology and relationship to implied Service-Oriented Architecture. Structured CTS facets for clearer explanation
WD08	8/5/2021	Toby Considine William T Cox David Holmberg	Clarify and simplify actor facets descriptions, including Tender, Transaction, and Configuration. Reduce redundant and less relevant content.
WD09	9/14/2021	William T Cox Toby Considine David Holmberg	Added Facet descriptions for Position, Market Characteristics, CTS Streams, and drafts of Privacy Consideration, Delivery and Party Registration Facets. Numerous edits for clarity and conciseness.
WD10	10/4/2021	Toby Considine William T Cox David Holmberg	Extended Market Facets. Defined Position and Delivery facets. Made references more consistent. Updated UML model and diagrams.
WD11	10/22/2021	David Holmberg William T Cox Toby Considine	Corrections for clarity. Improved UML diagrams. Flagged requests for comments in Public Review

Revision	Date	Editor	Changes Made
CSD01	10/29/2021	OASIS TC Administration	Content as in WD11, formatted to include OASIS metadata and references to the published specification
WD12	1/10/2022	William T Cox Toby Considine	Simpler edits in response to comments from PR
WD13		William T Cox Toby Considine	Clarification of Resource/Product/Instrument Removal of references to "Architecture" Responses to "Clarity" tagged issues
WD14	2/22/2022	William T Cox Toby Considine	Clarification of front material Section 1/-2 compared to eliminate duplicative definitions Numerous issues resolutions applied as per Jira
WD15	3/20/2020	William T Cox Toby Considine	Clarity, responses to issues from Review
WD16	4/12/2022	William T Cox Toby Considine	Marketplace and Market characteristics responses to issues Expanded Quotes and Tickers Focus on capitalization
WD17	4/25/2022	William T Cox Toby Considine	Updated UML Market Information added OTC Transactions Edits for Clarity
WD18	9/19/2023	Toby Considine	First response to FIX meetings Changed to Market/Market Segment language Reference FIX Tags when known Closings and Crossings added First pass at FIX-conformant Market Data Reports
WD19	10/2024	Toby Considine	Response to Second PR Preparations to work with
WD20-22		Toby Considine	Re-writes while discussing with FIX. Added Negotiations, Tickers, Instrument Data, Market Structure
WD23	6/23/2024	Toby Considine	Post PR03, showing all comments received
WD24	7/7/2024	Toby Considine William T. Cox	Working through public review comments Simplification of Tickers and Market Structure Re-work and simplification of Negotiations
WD25	7/7/2024	Toby Considine	Accepted simpler comments to focus attention to larger issues

Revision	Date	Editor	Changes Made
WD26	7/17/2024	Toby Considine	Moved material on selecting a Market Mechanism and on non-normative illustrations of business interactions to an appendix
WD27	8/12/2024	William T. Cox Toby Considine	Rework of models and exposition for all negotiations & subscriptions (9-12)
WD28	8/27/2024	William T Cox	UML updates and revisions for the entire technical content of the specification.
WD29		William T Cox Toby Considine David Holmberg	Subscriptions, and more consistent delineations of Structure (non-volatile) and Session (volatile) data throughout. Many smaller edits to align earlier parts of document with the specification as it has emerged in later details.

2434

Notices

Copyright © OASIS Open 20222024. 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: [\[https://www.oasis-open.org/policies-guidelines/ipr/\]](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.