



1
2
3
4 **Working Draft**
5 **MEF W106 v1.26**
6
7

8 **LSO Sonata Access E-Line Product
9 Schemas and Developer Guide**
10
11

12 **May 2022**
13
14 **This draft represents MEF work in progress and
15 is subject to change.**
16

Contribution Number

17 Disclaimer

18 © MEF Forum 2022. All Rights Reserved.

19 The information in this publication is freely available for reproduction and use by any recipient and is
20 believed to be accurate as of its publication date. Such information is subject to change without notice and
21 MEF Forum (MEF) is not responsible for any errors. MEF does not assume responsibility to update or
22 correct any information in this publication. No representation or warranty, expressed or implied, is made
23 by MEF concerning the completeness, accuracy, or applicability of any information contained herein and
24 no liability of any kind shall be assumed by MEF as a result of reliance upon such information.

25 The information contained herein is intended to be used without modification by the recipient or user of
26 this document. MEF is not responsible or liable for any modifications to this document made by any other
27 party.

28 The receipt or any use of this document or its contents does not in any way create, by implication or otherwise:

- 30 a) any express or implied license or right to or under any patent, copyright, trademark or trade
31 secret rights held or claimed by any MEF member which are or may be associated with the
32 ideas, techniques, concepts or expressions contained herein; nor
- 33 b) any warranty or representation that any MEF members will announce any product(s) and/or
34 service(s) related thereto, or if such announcements are made, that such announced product(s)
35 and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- 36 c) any form of relationship between any MEF member and the recipient or user of this document.

37 Implementation or use of specific MEF standards, specifications, or recommendations will be voluntary,
38 and no Member shall be obliged to implement them by virtue of participation in MEF Forum. MEF is a
39 non-profit international organization to enable the development and worldwide adoption of agile, assured
40 and orchestrated network services. MEF does not, expressly or otherwise, endorse or promote any specific
41 products or services.

42

43 EXPORT CONTROL: This document contains technical data. The download, export, reexport or disclosure
44 of the technical data contained in this document may be restricted by applicable U.S. or foreign export laws,
45 regulations and rules and/or applicable U.S. or foreign sanctions ("Export Control Laws or Sanctions").
46 You agree that you are solely responsible for determining whether any Export Control Laws or Sanctions
47 may apply to your download, export, reexport or disclosure of this document, and for obtaining (if available)
48 any required U.S. or foreign export or reexport licenses and/or other required authorizations.

49	Table of Contents		
50	1	List of Contributing Members	1
51	2	Abstract.....	2
52	3	Terminology and Abbreviations	3
53	4	Compliance Levels	7
54	5	Numerical Prefix Conventions	8
55	6	Introduction.....	9
56	7	Overview of LSO Sonata	11
57	8	Overview of Access E-Line Services	13
58	8.1	EVCs	13
59	8.2	OVCs	13
60	8.3	Access E-Line.....	14
61	8.4	Roles	14
62	9	Data Model Design Principles and Assumptions.....	16
63	9.1	Mandatory Service Attributes	17
64	9.2	Optional Service Attributes	17
65	9.3	Fixed Service Attributes	17
66	10	Order Milestones.....	19
67	11	Information Model for Access E-Line Product Data Model.....	21
68	11.1	Organization of Service Attributes	22
69	12	Data Models for Access E-Line Product	24
70	12.1	Organization and Structure of the Schemas	24
71	12.2	Example Schema	25
72	12.3	Additional Details.....	28
73	12.3.1	Naming Conventions.....	28
74	12.3.2	OVC End Point Service Attributes.....	28
75	12.3.3	Additional OVC End Point Attributes	28
76	13	Relationship Between Entities.....	30
77	14	Access E-Line Service Attributes.....	32
78	14.1	Access E-Line OVC and OVC End Point Service Attributes	32
79	14.2	Operator UNI Service Attributes.....	41



80	14.3 ENNI Attributes	46
81	14.3.1 ENNI Attributes	46
82	14.3.2 ENNI Common Attributes	46
83	14.3.3 ENNI Service Attributes	47
84	14.3.4 Operator Multilateral Attributes.....	49
85	15 Common Classes and Types.....	51
86	15.1 Bandwidth Profiles	51
87	15.2 Class of Service	55
88	15.3 Color Identifier	60
89	15.4 Egress Maps	62
90	15.5 End Point Maps	67
91	15.6 External Interfaces.....	70
92	15.7 Layer 2 Control Protocols (L2CP).....	72
93	15.8 Link Aggregation.....	73
94	15.9 Service Level Specification.....	76
95	15.10 Carrier Ethernet Standard Types	87
96	15.11 Carrier Ethernet Utility Classes.....	89
97	16 Service Attributes Not Included	94
98	17 References.....	96
99	Appendix A Usage examples (Informative).....	98
100	A.1 High-Level flow	98
101	A.2 Integration of product specifications into the APIs.....	99
102	A.3 action: add	102
103	A.3.1 Use Case 1: Address Validation.....	102
104	A.3.2 Use Case 2: POQ - new Access E-Line, new UNI, low class of service	105
105	A.3.3 Use Case 3: POQ - new Access E-Line, existing UNI, low+high class of service	118
106	A.3.4 Use Case 4: Quote	122
107	A.3.5 Use Case 5: Product Order	123
108	A.4 action: modify.....	125
109	A.4.1 Use Case 6: POQ: Bandwidth change.....	125



110	A.4.2	Use Case 7: Quote: Bandwidth change	126
111	A.4.3	Use Case 8: Product Order: Bandwidth change	126
112	A.4.4	Use Case 9: Product Order – VLAN change at the UNI.....	127
113	A.5	action: delete.....	127
114	A.5.1	Use Case 10: Product Order – delete both Access E-Line and UNI	128
115	A.5.2	Use Case 11: Move Access E-Line to a different Location	129
116	A.6	Example Postman collection	130
117			

118	List of Figures	
119	Figure 1 – LSO Sonata Reference Diagram.....	11
120	Figure 2 – LSO Sonata API Structure.....	12
121	Figure 3 – OVCs and Access E-Line Services.....	14
122	Figure 4 – Service Providers and Network Operators	15
123	Figure 5 – Components of Subscriber and Operator Ethernet Services.....	21
124	Figure 6 – Carrier Ethernet Super classes	22
125	Figure 7 – Access E-Line Information Model	23
126	Figure 8 – Schema File Organization	24
127	Figure 9 – Cantata and Sonata End-to-End Function Flow	98
128	Figure 10 – The Extension Pattern.....	100
129	Figure 11 – UC1: Address Validation request	103
130	Figure 12 – UC1: Address Validation response.....	104
131	Figure 13 – UC2: POQ Request, envelope part	106
132	Figure 14 – UC2: POQ request building steps.....	109
133	Figure 15 – UC2: Setup Diagram	109
134	Figure 16 – UC2: Access E-Line, basic attributes	110
135	Figure 17 – UC2: Access E-Line, Carrier Ethernet SLS	110
136	Figure 18 – UC2: Access E-Line, ENNI Endpoint.....	110
137	Figure 19 – UC2: Access E-Line, UNI Endpoint	111
138	Figure 20 – UC2: UNI	111
139	Figure 21 – UC3: Setup diagram	118
140	Figure 22 - UC3: Access E-Line relationships	119
141	Figure 23 – UC3: Access E-Line, basic attributes	119
142	Figure 24 – UC3: Access E-Line, Carrier Ethernet SLS	120
143	Figure 25 – UC3: Access E-Line, ENNI Endpoint.....	120
144	Figure 26 – UC3: Access E-Line, UNI Endpoint	121
145	Figure 27 - UC4: Quote Request	122
146	Figure 28 - UC5: Product Order request.....	124
147	Figure 29 - UC6: POQ modify request	125



148	Figure 30 - UC6: Access E-Line modified attributes.....	126
149	Figure 31 - UC7: Quote modification request	126
150	Figure 32 - UC8: Order modification request.....	127
151	Figure 33 - UC9: Order modification request.....	127
152	Figure 34 - UC10: Order deletion request	128
153		

154	List of Tables	
155	Table 1 – Terminology and Abbreviations	6
156	Table 2 – Numerical Prefix Conventions.....	8
157	Table 3 – Order Milestones for Access-E-Line	19
158	Table 4 – Product Relationship Roles.....	30
159	Table 5 – Place Relationship Role	31
160	Table 6 – Service Attributes Not Included in Schemas	95
161		

162 **1 List of Contributing Members**

163 The following members of the MEF participated in the development of this document and have requested
164 to be included in this list.

165 *Editor Note 1: This list will be finalized before Letter Ballot. Any member that comments in at least
166 one CfC is eligible to be included by opting in before the Letter Ballot is initiated. Note
167 it is the MEF member that is listed here (typically a company or organization), not their
168 individual representatives.*

- 169 • ABC Networks
170 • XYZ Communications

171

172

2 Abstract

173 The MEF Standard consisting of this schema guide and its associated software artifacts (JSON Schemas)
174 defines and describes the product-specific information used in LSO Sonata APIs for a set of Business Func-
175 tions—specifically, Product Offering Qualification, Quoting, Ordering and Inventory—for an Access E-
176 Line product. The document starts with an overview of LSO Sonata and the Access E-Line Carrier Ethernet
177 service. It then provides a basic information model for the MEF Carrier Ethernet Service Attributes. The
178 final sections describe the Data Model focused on the JSON Schemas associated with this specification.

179 This document can be thought of as a user's guide for the Access E-Line Data Model. MEF Services are
180 described by a set of Service Attributes. Each Service Attribute describes an aspect of the service that is
181 agreed between the provider and the user of the service. The document that describes the Service Attributes
182 for Operator Ethernet Services (of which Access E-Line is one) is [MEF 26.2 \[9\]](#). The service, itself, is
183 specified in MEF 51.1 [11] based on the Service Attributes defined in MEF 26.2. MEF 26.2 specifies 72
184 Service Attributes to describe the various components that compose an Operator Ethernet Service. This
185 document defines a data model that includes 56 of those Service Attributes and also lists the Service At-
186 tributes that are not included in the data model and the reason why each is not included.

187

188 3 Terminology and Abbreviations

189 This section defines the terms used in this document. In many cases, the normative definitions to terms are
190 found in other documents. In these cases, the third column is used to provide the reference that is control-
191 ling, in other MEF or external documents. If the reference includes an asterisk (*), the definition has been
192 adapted from the original.

193

Term	Definition	Reference
Access E-Line	An E-Access Service based on a point-to-point OVC.	MEF 51.1 [11]
Business Applications	The Service Provider functionality supporting Business Management Layer functionality (e.g., product catalog, order management, billing, relationship management, etc.)	MEF 55.1 [12]
BUS	See <i>Business Applications</i>	MEF 55.1 [12]
Business Functions	In the context of this document Business Functions refer to <i>Product Offering Qualification (POQ)</i> , <i>Order Management</i> , <i>Quote Management</i> , and <i>Inventory Management</i> .	This Document
Buyer	For the purposes of this document, a Buyer is the Service Provider who is ordering from an Operator (aka, Seller).	MEF 57.2* [13]
Data Model	A representation of concepts of interest to an environment in a form that is dependent on data repository, data definition language, query language, implementation language, and/or protocol (typically, but not necessarily, all five).	MEF 78.1 [14]
E-Access Service	An Operator Ethernet Service that associates at least one OVC End Point that is at a UNI and at least one OVC End Point that is at an ENNI.	MEF 51.1 [11]
External Network Network Interface	A reference point representing the boundary between two Operator Carrier Ethernet Networks that are operated as separate administrative domains.	MEF 26.2 [9]
ENNI	External Network Network Interface	MEF 26.2 [9]
ENNI Frame	An Ethernet Frame that is exchanged across an ENNI between two Operators.	MEF 26.2* [9]
Ethernet Service Provider	An organization that provides to a Subscriber a connectivity service that carries Ethernet Frames irrespective of the underlying technology and that is specified using Service Attributes as defined in a MEF Standard.	MEF 10.4* [8]
Ethernet Subscriber	The end-user of an Ethernet Service	MEF 10.4 [8]

Term	Definition	Reference
Ethernet UNI	The demarcation point between the responsibility of the Ethernet Service Provider and the Ethernet Service Subscriber.	MEF 10.4 [8]
Ethernet Virtual Connection	An association of EVC End Points	MEF 10.4 [8]
EVC	Ethernet Virtual Connection	MEF 10.4 [8]
EVC End Point	A construct at a (Ethernet) UNI that selects a subset of the Service Frames that pass over the UNI.	MEF 10.4 [8]
External Interface	Either a UNI or an ENNI.	MEF 4 [5]
External Interface Frame	Either a Service Frame or an ENNI Frame	MEF 26.2 [9]
Information Model	A representation of concepts of interest to an environment in a form that is independent of data repository, data definition language, query language, implementation language, and protocol	MEF 78.1 [14]
Inventory	Product Inventory	MEF 81 [16]
Milestone	An event that occurs during the fulfillment process that indicates a significant step in the process has been completed	This document
Operator	The administrative entity of a Carrier Ethernet Network	MEF 26.2 [9]
Operator UNI Service Attributes	The Service Attributes used to describe an agreement between a Service Provider and an Operator for a UNI associated with an OVC.	MEF 26.2 [9]
Order	One or more Product Order Items formulated into a fulfilment request made by a Buyer to a Seller.	This document (derived from MEF 57.2 [13])
Operator Ethernet Service	An Ethernet Service that is provided by an Operator to another Operator or to an Ethernet Service Provider.	MEF 51.1 [11]
Operator Virtual Connection	An association of OVC End Points	MEF 26.2 [9]
OVC	Operator Virtual Connection	MEF 26.2 [9]
OVC End Point	A logical entity at a given External Interface that is associated with a distinct set of frames passing over that External Interface.	MEF 26.2 [9]
Product	One or more goods or services that is or may be sold to a Buyer by a Seller.	MEF 79 [15]

Term	Definition	Reference
Product Offering	The commercial and technical details of a Product sold by a Seller. A Product Offering defines all the commercial terms and, through association with a particular Product Specification, defines all of the technical attributes and behaviors of the Product. A Product Offering may constrain the allowable set of configurable technical attributes and/or behaviors specified in the associated Product Specification.	MEF 79 [15]
Product Order Item	An individual item included in a Product Order that describes the action to be taken on a Product or Product Offering by the Seller. The objective is for the Seller to complete the fulfilment process of this Product or Product Offering at the place defined by the Buyer.	This document
Product Inventory	The inventory managed by the Seller resulting from Order completion.	MEF 81 [16]
Product Specification	A Product Specification defines the template or detailed description from which Product Offerings can be defined.	MEF 79 [15]
POQ	Product Offering Qualification	MEF 79 [15]
Product Offering Qualification	One or more Product Offering Qualification Items formulated into a requirement made by a Buyer to a Seller.	MEF 79 [15]
Product Offering Qualification Item	An individual article included in a POQ that describes a product of a particular type (product offering). The objective is to determine if it is feasible for the Seller to deliver this item as described and for the Seller to inform the Buyer of the estimated time interval to complete this delivery.	MEF 79 [15]
Quote	One or more Quote Items formulated into a request for pricing of a Product or Product Offering made by a Buyer to a Seller.	This document
Quote Item	An individual item included in a Quote that describes the Buyer's interest in a price from the Seller for a Product or Product Offering. The objective is to determine the charges and timeframe for the Seller to deliver this item as described by the Buyer.	This document
Seller	For the purposes of this document, a Seller is the Operator who is providing the product to the Buyer.	MEF 57.2* [13]
Service Attribute	Specific information that is agreed between the provider and the user of the service, that describes some aspect of the service behavior or capability.	MEF 10.4 [8]
Service Frame	An Ethernet Frame that is exchanged across a UNI between a Subscriber and a Service Provider.	MEF 10.4 [8]
Service Provider	In the context of this document, a Service Provider is an Ethernet Service Provider. In this document we use Service Provider to include Super Operator as specified in MEF 26.2 (also referred to as SP/SO).	This Document



Term	Definition	Reference
Sonata	The Management Interface Reference Point supporting the management and operations interactions (e.g., ordering, billing, trouble management, etc.) between two network providers (e.g., Service Provider Domain and Partner Domain).	MEF 55.1 [12]
Subscriber	In the context of this document, a Subscriber is an Ethernet Subscriber.	This Document
UNI	An Ethernet User Network Interface	MEF 10.4 [8]

194

Table 1 – Terminology and Abbreviations

195

196 4 Compliance Levels

197 The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**",
198 "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in
199 this document are to be interpreted as described in BCP 14 (RFC 2119 [3], RFC 8174 [4]) when, and only
200 when, they appear in all capitals, as shown here. All key words must be in bold text.

201 Items that are **REQUIRED** (contain the words **MUST** or **MUST NOT**) are labeled as [**Rx**] for required.
202 Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD NOT**) are labeled as [**Dx**]
203 for desirable. Items that are **OPTIONAL** (contain the words **MAY** or **OPTIONAL**) are labeled as [**Ox**]
204 for optional.

205 5 Numerical Prefix Conventions

206 This Standard uses the prefix notation to indicate multiplier values as shown Table 2.

207

Decimal		Binary	
Symbol	Value	Symbol	Value
k	10^3	Ki	2^{10}
M	10^6	Mi	2^{20}
G	10^9	Gi	2^{30}
T	10^{12}	Ti	2^{40}
P	10^{15}	Pi	2^{50}
E	10^{18}	Ei	2^{60}
Z	10^{21}	Zi	2^{70}
Y	10^{24}	Yi	2^{80}

208

Table 2 – Numerical Prefix Conventions

209 6 Introduction

210 LSO Sonata provides a programmatic interface for establishing (quoting, ordering, etc.) services between
211 Operators and/or Service Providers. This API is hierarchically structured. The outer-most structure includes
212 information relating to the access method (e.g., REST), next is information relating to the function being
213 requested (e.g., Product Order Qualification or Quote, etc.) and the inner-most structure contains infor-
214 mation relating to the specific product, for example Ethernet Private Line or, in this specification, Access
215 E-Line.

216 Access E-Line is an Operator Ethernet Service, i.e., an Ethernet Service that is arranged between a Service
217 Provider and an Operator. The Service Attributes that are agreed to between the parties are defined in MEF
218 26.2 [9]. The Service definition for Access E-Line which is, in effect, a set of constraints on the values of
219 the Service Attributes, is provided in MEF 51.1 [11].

220 This specification is accompanied by a Data Model for the Access E-Line components instantiated as a set
221 of JSON schemas that can be used within the Sonata API to perform Product Order Qualification, Quota-
222 tion, Order, and request an Inventory for the Access E-Line Product consisting of:

- 223 • Access E-Line OVC including two OVC End Points, one at the UNI and one at the ENNI
224 • UNI
225 • ENNI

226 The model assumes an Access E-Line product based on an existing ENNI, so the schemas for ENNI are
227 only included for Inventory. There are a small number of ENNI attributes that might require modification
228 based on the definition of services (like Access E-Line) that traverse the ENNI. To address this issue in the
229 most common situations, a couple of additional OVC End Point Service Attributes have been defined (see
230 section 12.3.3).

231 The document contains the following sections:

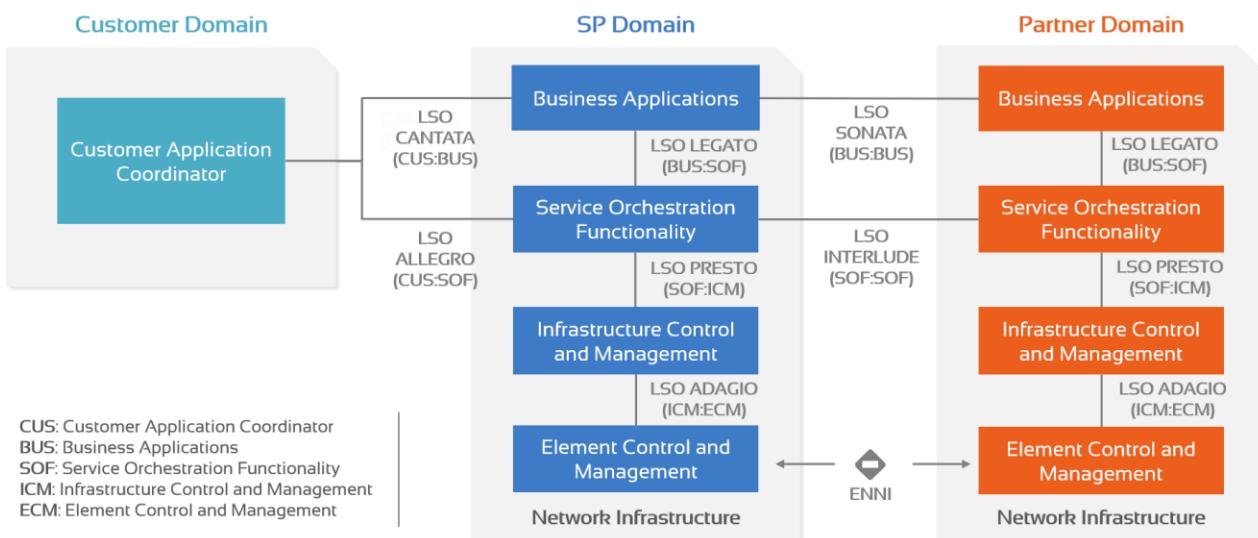
- 232 • An overview of LSO Sonata (section 7)
233 • An overview of the Access E-Line Service (section 8)
234 • Data Model Design Principles (section 9)
235 • Order Milestones (section 10)
236 • An abbreviated Information Model for Access E-Line and explanation of the organization of
237 the Service Attributes in MEF 26.2 (section 11)
238 • Organization of the data model for Access E-Line (section 12)
239 • The relationship between the entities in the service (section 13)

240 These sections are followed by three sections that contain tables that describe the details of the data model.
241 The tables include information about each class and a list of properties in each class. For each property, the
242 JSON Name, description, data type, details about allowed values, and, in some cases, some additional in-
243 formation about relationships between Service Attributes is provided.

- 244 • Section 14 contains the details of the Service Attributes for OVC, OVC End Point, UNI, and
245 ENNI
- 246 • Section 15 contains all of the common classes and types referenced by the Service Attributes
- 247 • Section 16 lists the Service Attributes that are not included in the data models
- 248

249 7 Overview of LSO Sonata

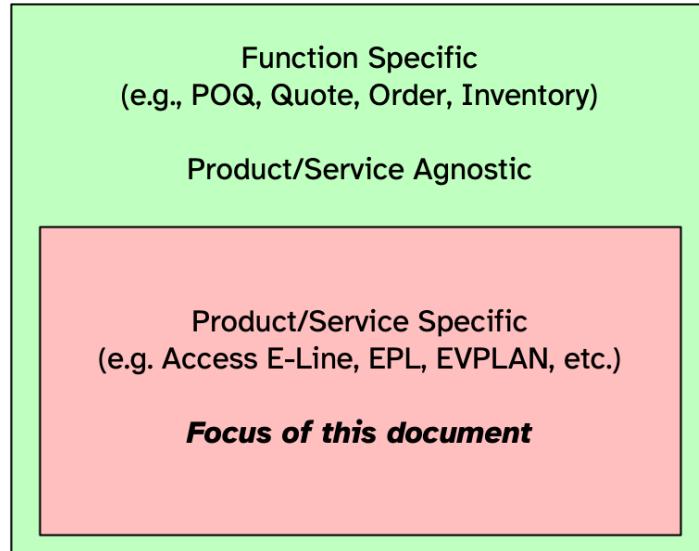
250 MEF 55.1 [12] describes the Reference Architecture for Lifecycle Service Orchestration (LSO) of MEF-defined services. MEF 55.1 defines seven LSO Interface Reference Points (see Figure 1) that are abstract
 251 interconnection points between different entities—either within the Service Provider domain (intra-domain)
 252 or between Service Provider and other business entities (inter-domain). One of these LSO Interface Reference
 253 Points is LSO Sonata which defines the abstract interconnection point between a Service Provider
 254 and an Operator which can provide wholesale (offnet) connectivity services to the Service Provider. It is at
 255 this Interface Reference Point – LSO Sonata – that the Service Provider and the Operator interact to orchestrate
 256 business transactions for the different business functions. Examples of inter-provider Business Functions
 257 include address qualification, product offering qualification, quotes, ordering, trouble ticketing, and
 258 billing/settlement.



260
 261 **Figure 1 – LSO Sonata Reference Diagram**

262 The mutual access to Business Functionalities is automated via APIs at LSO Sonata which are standardized
 263 by MEF as LSO Sonata APIs, and which are made available by MEF in a series of releases of the LSO
 264 Sonata SDK.

265 The LSO Sonata APIs comprise two parts—a product-agnostic API and a set product-specific data models,
 266 as shown in Figure 2.



267

268

Figure 2 – LSO Sonata API Structure

269 This document describes the product-specific data model for a MEF 3.0 Access E-Line service as defined
270 in MEF 51.1 [11].

271

272 8 Overview of Access E-Line Services

273 This specification describes a data model for MEF-defined Access E-Line Services. Access E-Line Services
274 are point-to-point Carrier Ethernet “operator” services. An Operator Ethernet Service is negotiated between
275 an Operator and a Service Provider. An Operator Ethernet Service always includes at least one ENNI (Ex-
276 ternal Network-Network Interface) between two Operators and usually (but not always) includes one or
277 more UNIs (User-Network Interface).

278 The subsequent sub-sections provide background on OVCs and Access E-Line. It starts with a brief expla-
279 nation of EVCs (Ethernet Virtual Connections) which is one of the primary uses of OVCs in MEF services.

280 8.1 EVCs

281 A Subscriber Ethernet Service is built on an Ethernet Virtual Connection (EVC) which is an association of
282 EVC End Points. An EVC End Point is a construct located at a UNI that represents the logical attachment
283 of an EVC to that UNI and selects a subset of Service Frames that pass over the UNI. EVCs and their
284 Service Attributes are described in MEF 10.4 [8]. There are three types of EVCs: point-to-point, multipoint-
285 to-multipoint, and rooted-multipoint. MEF 6.3 [6] defines three types of EVC services based on EVC type:
286 E-Line (Ethernet Private Line and Ethernet Virtual Private Line), E-LAN (Ethernet Private LAN and Ether-
287 net Virtual Private LAN), and E-Tree (Ethernet Private Tree and Ethernet Virtual Private Tree), respec-
288 tively.

289 EVCs are sold to a Subscriber by an Ethernet Service Provider. In most cases, the Ethernet Service Provider
290 is also an Operator that provides some or all of the network infrastructure and resources for the EVC. If all
291 of the UNIs in an EVC are accessible on the Service Provider’s network, then the Service Provider can
292 deliver the EVC without having to coordinate with other Operators.

293 8.2 OVCs

294 If one or more of the UNIs associated by an EVC is not accessible to the Ethernet Service Provider’s net-
295 work, the Service Provider must purchase access to the UNI or UNIs from an Operator that does have
296 access. In this case, the EVC is composed of Operator Virtual Connections (OVCs) where each OVC rep-
297 presents the part of the EVC that is provided by one Operator. OVCs, ENNIs, and their Service Attributes
298 are described in MEF 26.2 [9]. As with EVCs, there are three types of OVCs: point-to-point, multipoint-
299 to-multipoint, and rooted multipoint.

300 Operators connect to each other at ENNIs and each OVC delivers Ethernet Frames between the various
301 External Interfaces, UNIs and ENNIs, within an Operator’s footprint. So, whereas an EVC associates a set
302 of UNIs, an OVC associates one or more ENNIs and zero or more UNIs.

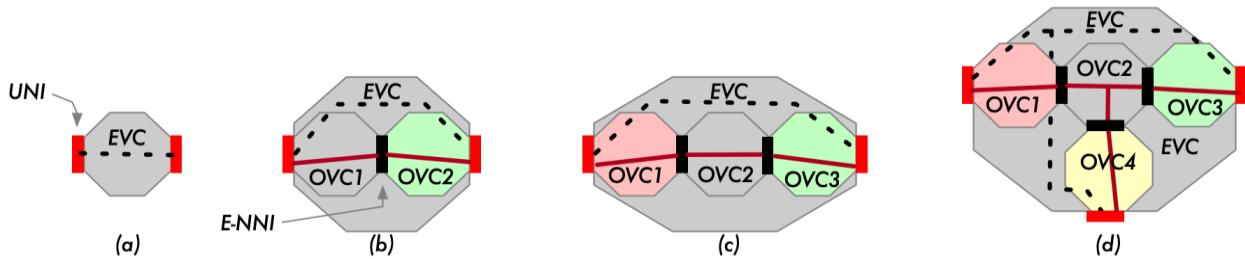
303 Note that although composing an EVC is the primary use of OVCs in MEF specifications, in reality the
304 other side of the ENNI is opaque to the OVC, and it might be providing access to a cloud service or other
305 communications service.

306 MEF 51.1 [11] describes a set of Operator Ethernet Services. It describes three generic service types, O-
307 LINE, O-LAN, and O-TREE, based on the three types of OVCs. In MEF 51.1, an OVC that contains at
308 least one UNI is referred to as an *access* OVC and an OVC that contains only ENNIs is referred to as a
309 *transit* OVC. MEF 51.1 also defines some specific service types. An O-LINE service that includes one UNI

310 is an Access E-Line Service and an O-LINE service that includes only ENNIs is a Transit E-Line Service.
 311 Similarly, there are Access E-LAN and Transit E-LAN Services.

312 **8.3 Access E-Line**

313 As noted above, an Access E-Line Service is a point-to-point Operator Ethernet Service. Since an OVC
 314 must associate at least one ENNI, and an Access OVC must associate at least one UNI, an Access E-Line
 315 service connects exactly one UNI to exactly one ENNI. The Access E-Line service can be used to deliver
 316 E-Line, E-LAN, and E-Tree Services (as well as connection to other communications services). Figure 3
 317 shows some of the possibilities.



319 **Figure 3 – OVCs and Access E-Line Services**

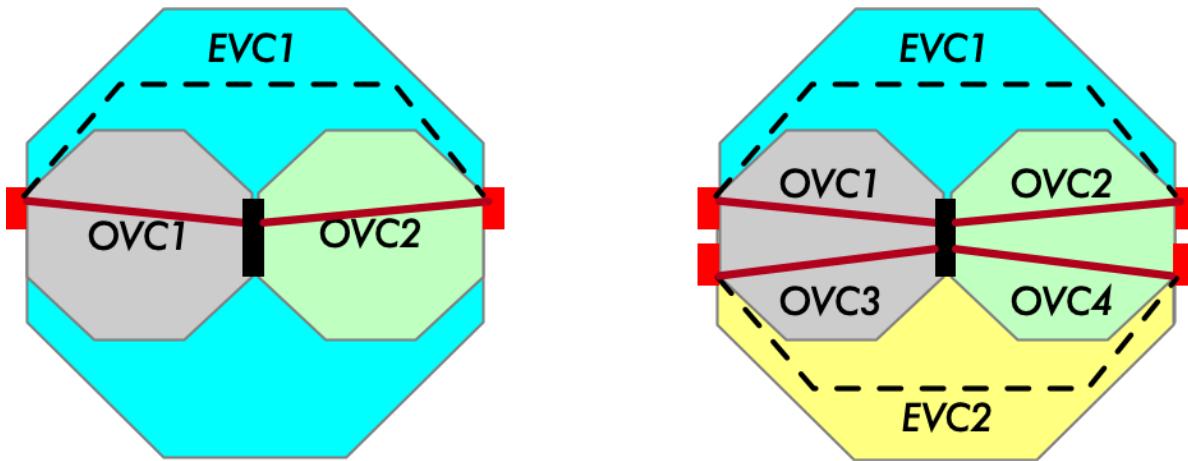
320 In the diagram, in cases (a), (b), and (c), Ethernet Service Provider Gray is delivering an E-Line EVC
 321 (Ethernet Private Line or Ethernet Virtual Private Line) to the Subscriber. In case (d) it is delivering an E-
 322 LAN EVC with three UNIs. In case (a) the E-Line service (EVC) is delivered entirely by Service Provider
 323 Gray on its own network since both UNIs are accessible to it. In case (b) the right-hand UNI is not accessible
 324 on Gray's network, so Gray purchases OVC2 from Network Operator Green in order to reach it. OVC2 is
 325 an access OVC, and it is a point-to-point OVC, so it is an Access E-Line Service. OVC1 is also, technically,
 326 an Access E-Line. Case (c) is a bit more complex. Service Provider Gray does not have access to either
 327 UNI. It purchases an Access E-Line from Operator Red (OVC1) and another one from Operator Green
 328 (OVC3) and provides a transit OVC (Transit E-Line) between them (OVC2). In case (d), Service Provider
 329 Gray (again) does not have access to any of the UNIs, so it purchases three Access E-Line Services, OVC1
 330 from Red, OVC3 from Green, and OVC4 from Yellow, and ties them together with OVC2, a Transit E-
 331 LAN service.

332 In cases (b), (c), and (d), Service Provider Gray is buying Access E-Line Services from other Operators.
 333 The LSO Sonata API with the Access E-Line product configuration as described in this specification allows
 334 the Business Functions associated with purchasing these services (POQ, Quote, Order, and Inventory) to
 335 be performed programmatically.

336 **8.4 Roles**

337 Carrier Ethernet Services are carried over Carrier Ethernet Networks operated by Operators. An Ethernet
 338 Service Provider purchases OVCs from Operators in order to compose the EVC. Frequently, the Ethernet
 339 Service Provider is one of the Operators. This is the case in Figure 3 for Service Provider / Operator Gray.

340 It doesn't have to be that way. There is no reason that a third-party organization—a systems integrator or
 341 unrelated Service Provider—could not purchase OVCs from other Operators and sell an EVC to the Sub-
 342 scriber, for example as shown with Service Provider Blue in the left-hand diagram below.



343

344

Figure 4 – Service Providers and Network Operators

345 But now that Operators Gray and Green have decided to allow a third party to buy OVCs in their networks
346 and, in effect, buy access to the ENNI between them, why just one? Why not allow Service Provider Yellow
347 to do the same thing, as shown in the right-hand picture? (And, of course, Service Provider Yellow might
348 actually be Gray or Green).

349 As a result, different aspects (Service Attributes) of the ENNI are relevant to different parties. Clearly the
350 two Operators care about some aspects of the ENNI (e.g., the physical characteristics). Each of the Service
351 Providers cares about certain aspects of its OVCs that terminate at the ENNI. The ENNI Service Attributes
352 are partitioned into groups based on which organizations need to agree on which attributes. This is discussed
353 further in section 12.1.

354 9 Data Model Design Principles and Assumptions

355 The design for the Access E-Line data model is based on a number of assumptions:

- 356 • None of the Service Attributes included in the schemas are coded as “Required”.
- 357 • Each Seller will divide all Service Attributes included in the schemas into one of three categories for each Business Function:

- 359 ○ Mandatory - attributes that must be provided by the Buyer in a POQ/Quote/Order
360 request(see section 9.1)
- 361 ○ Optional - attributes that may be provided by the Buyer in a POQ/Quote/Order request
362 (see section 9.2)
- 363 ○ Fixed - attributes that are hard coded by the Seller and may not be specified by the
364 Buyer in a POQ/Quote/Order request (see section 9.3)

365 **[R1]** The Seller and Buyer **MUST** agree, for each Service Attribute, whether the Service
366 Attribute is mandatory, optional, or fixed for each Business Function for a given Product
367 Offering.

368 The Service Attribute categorization can be defined and negotiated during the onboarding process or de-
369 fined in a Product Catalog.

370 **[R2]** If a Service Attribute is categorized as optional for a Business Function for a Product
371 Offering, the Seller and Buyer **MUST** agree on the default value for the Service At-
372 tribute.

373 The categorization may depend on:

- 374 • Business Function - a given Service Attribute may, for example, be classified by the Seller as Fixed
375 for the Create POQ request; while it may be considered as Mandatory by the Seller for the Create
376 Product Order request.
- 377 • Product Action - a given Service Attribute may, for example, be classified as Mandatory by the
378 Seller for the Create POQ request for an INSTALL of a new product, while it may be considered
379 as Fixed for the Create POQ request for a MODIFY of an installed Product.
- 380 • Product Offering - a given Service Attribute may, for example, be classified as Mandatory by the
381 Seller for the Create POQ request for a Product Offering (e.g., Premium Service), while it may be
382 considered as Fixed for the Create POQ request for a different Product Offering (e.g., Basic Ser-
383 vice).

384 **[R3]** The Seller **MUST** reject an API request if the value for a Service Attribute requested
385 by the Buyer is not a supported value for a Business Function, Product Action, and
386 Product Offering.

387 The Access E-Line data model supports both INSTALL and CHANGE actions for POQ, Quote, and Order
388 for the OVC and Operator UNI. The data model does not support these functions for ENNI – they are

389 assumed to be created in the inventory as a result of the on-boarding process between two operators. Note
390 that the DISCONNECT action does not require support by the data model.

391 The Access E-Line data model supports the RETRIEVE action for Inventory for the OVC, Operator UNI
392 and ENNI. ENNIs need to be able to be retrieved from the inventory so that they can be referenced in the
393 Access E-Line order (or qualification or quote).¹

394 The location and physical layer of a UNI cannot be changed once it is ordered; instead, this is handled as
395 an installation (UNI at new location) and a disconnect (UNI at previous location), as there is often a re-
396 quirement for a smooth transition with minimum downtime.

397 **9.1 Mandatory Service Attributes**

398 [R4] If a Service Attribute is agreed to be Mandatory for a Business Function, Product Ac-
399 tion, and Product Offering, then the Buyer **MUST** include a value for the Service At-
400 tribute in the corresponding API request.

401 [R5] When the Seller receives a request in which any of the Mandatory Service Attributes
402 are not included, the request **MUST** be rejected by the Seller

403 **9.2 Optional Service Attributes**

404 [O1] If a Service Attribute is agreed to be Optional for a Business Function, Product Action,
405 and Product Offering, then the Buyer **MAY** include a value for the Service Attribute
406 in the corresponding API request.

407 [R6] The Seller **MUST** apply the agreed default value for an Optional Service Attribute if
408 a value is not included by the Buyer in an API request.

409 **9.3 Fixed Service Attributes**

410 A Service Attribute is considered Fixed for a Business Function, Product Action, and Product Offering
411 when only one value is applicable. This can be the case for example if:

- 412 • the Seller supports only a single value, or
- 413 • the value is derived from the value of one or more other Service Attributes or parameters, or
- 414 • the Seller specifies a single value in the Product Catalog for this Product Offering, or
- 415 • the Buyer and the Seller agree on a single value during Onboarding

¹ There are a small number of ENNI attributes that might require modification based on the definition of services (like Access E-Line) that traverse the ENNI. To address this issue in the most common situations, a couple of additional OVC End Point Service Attributes have been defined. See section 12.3.3.

416 The Seller applies the one applicable value for every request for which the Service Attribute is categorized
417 as Fixed.

418 [R7] The Buyer **MUST NOT** submit an API request to the Seller which has a value other
419 than the one applicable value for a Service Attribute that has been categorized as Fixed
420 for the Business Function, Product Action, and Product Offering.

421 [R8] The Seller **MUST** reject any API request from the Buyer if it has a value other than
422 the one applicable value for a Service Attribute that has been categorized as Fixed for
423 the Business Function, Product Action, and Product Offering.

424

425 10 Order Milestones

426 The Service Provider (Seller) can provide Product-Specific Product Order Item Milestone notifications to
427 the Buyer on the status of an Order as a sequence of Milestones for that Order as they are achieved. For
428 ordering an Access E-Line Service (OVC and UNI) the following milestones are commonly used (a Service
429 Provider may support some or all these milestones and not all milestones are applicable for all orders):

430

Milestone Value	Description	Applies To
SITE_SURVEY_SCHEDULED	Site Survey Scheduled	UNI
SITE_SURVEY_COMPLETE	Site Survey Complete	UNI
PLANNING_COMPLETE	Planning Complete	UNI, OVC
FIRM_DELIVERY_DATE_PROVIDED	Firm Delivery Date Provided	UNI, OVC
AWAITING_MUNICIPAL_APPROVAL	Awaiting Municipal Approval	UNI
MUNICIPAL_APPROVAL_GRANTED	Municipal Approval Granted	UNI
AWAITING_LANDLORD_APPROVAL	Awaiting Landlord Approval	UNI
LANDLORD_APPROVAL_GRANTED	Landlord Approval Granted	UNI
CONSTRUCTION_STARTED	Construction Started	UNI
CONSTRUCTION_COMPLETED	Construction Completed	UNI
AWAITING_ACCESS	Awaiting Site Access Permission (for end-to-end test)	UNI, OVC
ACCESS_DENIED	Site Access Denied (for end-to-end test). Issue is to be resolved with the Buyer and access may be re-attempted.	UNI, OVC
AWAITING_WIRING	Awaiting Installation of Inside Wiring by Landlord	UNI
WIRING_COMPLETE	Installation of Inside Wiring by Landlord Complete	UNI
EQUIPMENT_DISPATCHED	Equipment Dispatched	UNI
EQUIPMENT_DELIVERED	Equipment Delivered	UNI
EQUIPMENT_INSTALLED	Equipment Installed	UNI
E2E_TESTING_SCHEDULED	End-to-End Testing Scheduled	OVC
E2E_TESTING_COMPLETED	End-to-End Testing Completed	OVC
E2E_TESTING_FAILED	End-to-End Testing Failed. Issue is to be resolved and testing may be re-attempted.	OVC

431

Table 3 – Order Milestones for Access-E-Line

432 The Milestone Value in the first column of Table 3 is included in
433 *ProductOrderEventPayload.milestoneName* in the Product Order Notification API (see MEF 123 [20]).

434 11 Information Model for Access E-Line Product Data Model

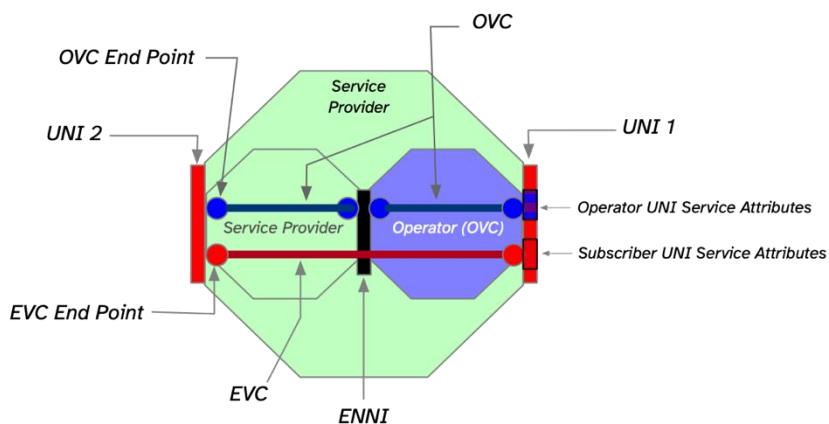
435 Carrier Ethernet Services are composed of three primary classes of objects: The Virtual Connection (EVC
436 and OVC), the External Interfaces or EIs (UNI and ENNI), and the Service End Points.

437 The UNI may be a component of two different business agreements, one between a Service Provider and
438 Subscriber for a Subscriber Ethernet Service, and one between a Service Provider and an Operator for an
439 Operator Ethernet Service. Two different sets of Service Attributes define these agreements, the Subscriber
440 UNI Service Attributes and the Operator UNI Service Attributes, respectively. This product specification
441 provides a data model for the Operator UNI Service Attributes. The data model for the Subscriber UNI
442 Service Attributes is provided as part of a product specification for EVC-based services such as Ethernet
443 Private Line.

444 Similarly, the ENNI is a component of (at least) two different business agreements, one between the two
445 Operators on opposite sides of the ENNI and one between each Operator and each Service Provider to
446 which it is granting access to the ENNI. This affects which Service Attributes are exposed in the Inventory
447 API. This product specification provides a data model for the business agreement between the Operator and
448 the Service Provider (in MEF 26.2 [9] this is referred to as SP/SO, Service Provider/Super Operator) and
449 the relevant file and class names specified in this Standard use this nomenclature).

450 The component that logically connects the EVC or OVC to each External Interface is the Service End Point
451 (EVC End Point or OVC End Point). Although an abstract construct, the End Point can be thought of as the
452 container for the EVC or OVC Service Attributes that can vary independently at each External Interface.
453 For example, the OVC Type (is the OVC point-point or multipoint?) is an OVC Service Attribute, it is
454 invariant at all points in the OVC. However, which VLAN(s) to map to the Service or how much bandwidth
455 to allow are attributes of the OVC that can be different at each EI, and these Service Attributes are encapsulated
456 in the OVC (or EVC) End Point.

457 All of these components are shown in the following diagram.



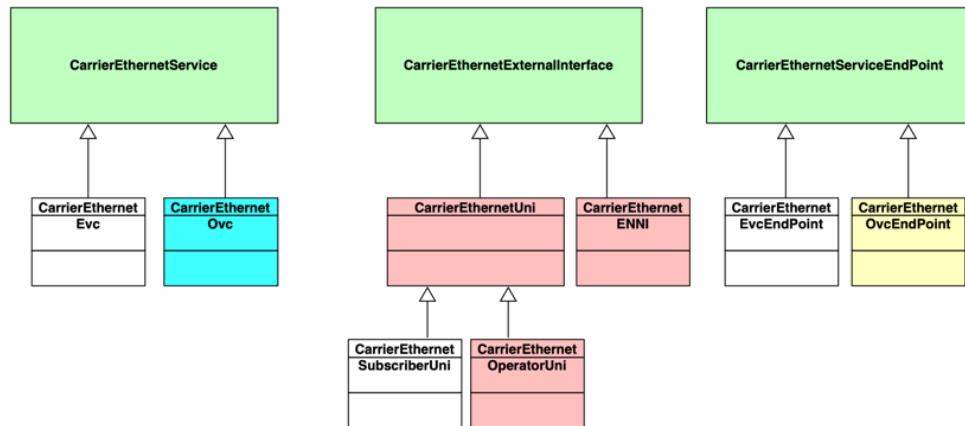
458

459 **Figure 5 – Components of Subscriber and Operator Ethernet Services**

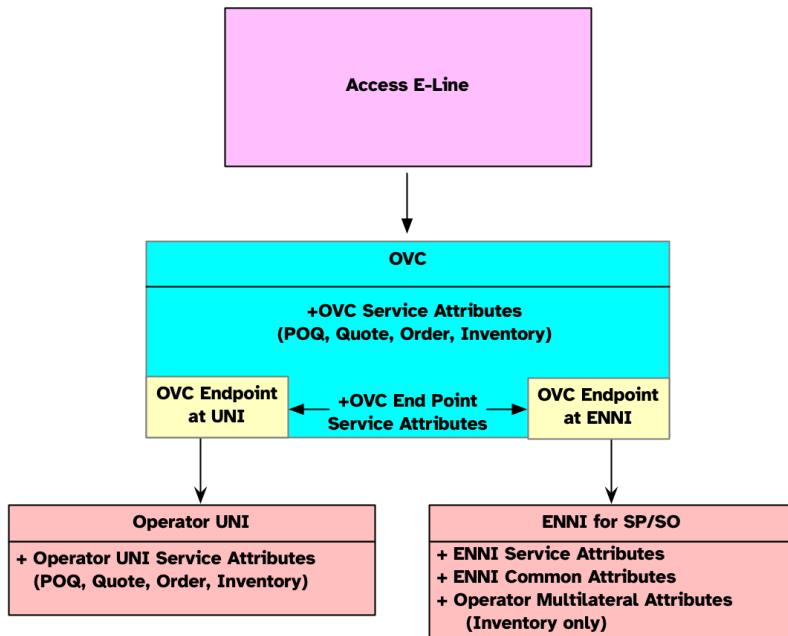
460 In this diagram, the Ethernet Service Provider (large green octagon) is building the EVC on two OVCs, one
 461 that it is providing itself to reach UNI 2 (small green octagon on left) and an Access E-Line Service that it
 462 is buying from another Operator (blue octagon on right). In reality, the OVC across the green octagon is
 463 likely also an Access E-Line Service, but since there is no buyer/seller (the Service Provider is providing it
 464 to itself), there aren't parties to agree on Service Attribute values.

465 **11.1 Organization of Service Attributes**

466 Access E-Line is an OVC Service that provides access to a single UNI from a single ENNI. Therefore, the
 467 primary components of Access E-Line are the OVC, the UNI, the ENNI, and two OVC End Points, one at
 468 the UNI and one at the ENNI (as shown in the blue octagon in Figure 5). Access E-Line configuration is
 469 based on the Service Attributes documented in MEF 26.2. The MEF Services Model (MSM) for Carrier
 470 Ethernet described in MEF 7.4 [7] defines three super classes for all EVC and OVC services, the Service
 471 (Virtual Connection), the External Interface, and the Service End Point. These super classes are shown in
 472 green in the following diagram along with their sub-classes. The sub-classes relevant to Access E-Line are
 473 shown in the color boxes.



474
 475 **Figure 6 – Carrier Ethernet Super classes**
 476 In the context of an Access E-Line Service, and the Service Attributes in MEF 26.2, these components are
 477 organized as follows (the colors in the following diagram are intended to match the colors in the previous
 478 diagram):
 479



480

481

Figure 7 – Access E-Line Information Model

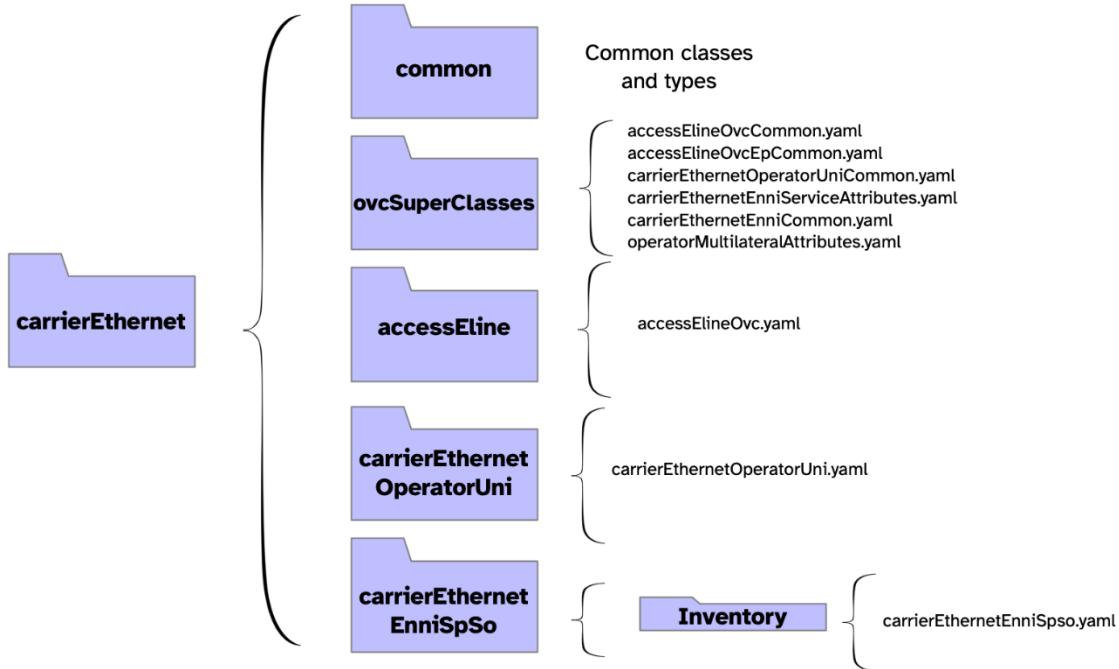
482 The Access E-Line product contains an OVC and two OVC Endpoints, one at the UNI and one at the ENNI.
483 The UNI and the ENNI are independent products that are related to the OVC (see section 13).

484 12 Data Models for Access E-Line Product

485 The data models for the Access E-Line product configuration in the Sonata API are expressed as a set of
 486 JSON schemas based on JSON schema draft 7 and encoded in YAML. These schemas accompany this
 487 document. This section explains the organization and structure of these schemas.

488 12.1 Organization and Structure of the Schemas

489 The schemas are organized into a file structure as shown in Figure 8.



490 491 **Figure 8 – Schema File Organization**

492 For Access E-Line (OVC) and Operator UNI the schemas are in the associated folder. For ENNI, the
 493 schema is down one folder in the *Inventory* folder since Inventory is the only function supported for ENNI
 494 at this time.

495 The Service Attribute definitions are defined in a set of common files in the ovcSuperClasses folder. These
 496 common classes are referenced in the relevant product component schema files. For example, carrierEther-
 497 netOperatorUni.yaml file includes:

```

498
499     allOf:
500         # Operator UNI Service Attributes used in all 4 Business Functions POQ/Quote/Order/Inventory
501         - $ref: "../../ovcSuperClasses/carrierEthernetOperatorUniCommon.yaml#/definitions/OperatorUniCommon"
  
```

502 There are three superclasses for ENNI:

- 503 • *carrierEthernetEnniServiceAttributes.yaml*: Includes attributes that represent the agreement
 504 between the Service Provider and the Operator for the ENNI for one Service Provider.

- 505 • *carrierEthernetEnniCommon.yaml*: Includes the attributes that describe the physical and log-
506 ical characteristics of the ENNI and are agreed to by the Operators on opposite sides of the
507 ENNI. They also need to be agreed to by each Super Operator or Service Provider that agrees
508 to the ENNI Service Attribute S-VLAN ID Control = *Full*.

509 • *operatorMultilateralAttributes.yaml*: Includes attributes that apply to the ENNI but are rele-
510 vant for both Operators, and each Super Operator or Service Provider that agrees to the ENNI
511 Service Attribute S-VLAN ID Control = *Full*.

512 There is a *common* folder that contains several files with classes that are common across several Carrier
513 Ethernet service components as well as a number of utility classes and types. The files are organized in
514 separate files by function, for example, bandwidth profile-related classes, service level specification-related
515 classes, etc.

516 Note that the “Carrier Ethernet” prefix for a schema indicates that it is not specific to a particular service.
517 For example, the *Carrier_Ethernet_Link_Aggregation* schema in the *common* folder is used with the Ac-
518 cess E-Line service schemas, but it can also be used with other Carrier Ethernet Services such as Transit E-
519 Line, Access E-LAN and Subscriber Ethernet services.

520 On a CHANGE request a single Service Attribute cannot be changed. The Buyer must send a full product
521 configuration including all Mandatory Service Attributes (section 9.1) and all Optional Service Attributes
522 (section 9.2) that were previously specified by the Buyer (in an INSTALL request or previous CHANGE
523 request). Any Optional Service Attributes that are not specified in a CHANGE request are reset to their
524 default value.

525 **[R9]** The Product Inventory for a product **MUST** include all Service Attributes that are
526 categorized as Mandatory (see [R1]).

527 **[R10]** The Product Inventory for a product **MUST** include all Service Attributes that are
528 categorized as Optional (see [R1]).

529 **[O2]** The Product Inventory for a product **MAY** contain Service Attributes that are catego-
530 rized as Fixed (see [R1]).

531 Including Service Attributes in the Inventory as specified in the previous requirements facilitates the
532 CHANGE action. The Buyer can RETRIEVE the current values for the Service Attributes and make the
533 desired changes and submit the CHANGE request.

534 12.2 Example Schema

535 Following is an example schema (from *Carrier_Ethernet_Link_Aggregation.yaml* – chosen only because
536 it is one of the shortest files):

537

```
538 ---  
539 "$schema": http://json-schema.org/draft-07/schema#  
540 title:  
541     MEF LSO Sonata - Carrier Ethernet Link Aggregation (Common) Product Specification  
542     # MEF Forum - Date: 2021318 Edit-level: 192  
543     # File: carrierEthernet/common/carrierEthernetLinkAggregation.yaml  
544  
545 definitions:  
546     ConversationIdToAggregationLinkMap:  
547         type: object  
548         description: >-  
549             This is a 2-tuple  $\langle x, y \rangle$  where  $x$  is a list of Port Conversation  
550             IDs or ranges of Port Conversation IDs (a Port Conversation ID  
551             is a VLAN ID or 0 for untagged frames) and  $y$  is a list of Link  
552             Aggregation Links. This is used in the Port Conversation to Aggregation  
553             Link Map for the UNI and ENNI.  
554 properties:  
555     conversationIDs:  
556         description: >-  
557             802.1AX-2014 §6.6.2.1 - A Port Conversation ID is a VLAN ID  
558             (1 to 4094) or 0 to represent untagged and priority-tagged frames.  
559         type: array  
560         minItems: 1  
561         uniqueItems: TRUE  
562         items:  
563             type: object  
564             $ref: "#/definitions/ConversationIdRange"  
565     aggLinkList:  
566         description: 802.1AX-2014 §6.6.2.1 - An ordered list of Aggregation Link Numbers  
567         # See additional validation notes in the Schema Guide  
568         type: array  
569         minItems: 1  
570         uniqueItems: TRUE  
571         items:  
572             type: integer  
573             minimum: 1
```

```
574     required:  
575         - conversationIDs  
576         - aggLinkList  
577     ConversationIdRange:  
578         type: object  
579         description: >-  
580             A range of ConversationID (either a VLAN Id or 0 for untagged  
581             frames) allowing three forms: start, start → start, and start  
582             → end  
583         properties:  
584             start:  
585                 description: >-  
586                     The starting Conversation ID of the range or the only Conversation  
587                     ID if there is no end value  
588                     # start must be less than or equal to end  
589                     # See additional validation notes in the Schema Guide  
590             type: integer  
591             minimum: 0  
592             maximum: 4094  
593             end:  
594                 description: The final Conversation ID in the range  
595                 # end, if specified, must be greater than or equal to start  
596                 # See additional validation notes in the Schema Guide  
597             type: integer  
598             minimum: 0  
599             maximum: 4094  
600         required:  
601             - start  
602 This schema contains two classes related to Link Aggregation, ConversationIdToAggregationLinkMap, and  
603 a class that it uses, ConversationIdRange. ConversationIdToAggregationLinkMap is referenced by the Ex-  
604 ternal Interfaces (UNI and ENNI) as follows (from Carrier_Ethernet_Operator_UNI.yaml):  
605  
606 aggregationLinkMap:  
607             description: >-  
608                 MEF 26.2 §14.6 and 802.1AX-2014 § 6.6.2.1 - A mapping of a
```

609 list of Port Conversation IDs (i.e., VLAN IDs or zero for untagged
610 frames) to an ordered list of physical links if there are multiple
611 physical links in the UNI and the link aggregation is ALL_ACTIVE.
612 # See additional validation notes in the Schema Guide
613 type: array
614 minItems: 0
615 uniqueItems: TRUE
616 items:
617 type: object
618 \$ref: "../../common/carrierEthernetLinkAggregation.yaml#/definitions/ConversationIdToAggregationLinkMap"

619 **12.3 Additional Details**

620 This section includes explanation of some additional conventions for the schema structure as well as some
621 additional attributes that have been added to facilitate product specification for some common edge cases.

622 **12.3.1 Naming Conventions**

623 In the schemas, class and type names are UpperCamelCase and Service Attribute/property names are low-
624 erCamelCase.

625 **12.3.2 OVC End Point Service Attributes**

626 OVC End Points are not separately orderable items. They are part of the OVC. The OVC End Points are
627 the repositories for OVC Service Attributes that can be different at each External Interface (UNI or ENNI)
628 whereas the OVC Service Attributes have the same value at every point in the OVC. The Access E-Line
629 OVC schema includes two properties at the end of the OVC Service Attributes, *uniEP* and *enniEP* each of
630 which refers to the OVC End Point class (which is included in the same schema file).

631 Access E-Line allows this simplified coding since it has exactly two End Points and one of them must be
632 at a UNI and the other must be at an ENNI. In the general case of a service that allows an arbitrary number
633 of End Points (e.g., a multipoint service) or where the external interface types are not predetermined, the
634 OVC properties might include an array of OVC Endpoints rather than the two pre-defined End Points.

635 Note that one of the OVC End Point Service Attributes is OVC End Point Type which can be “UNI” or
636 “ENNI”. Since this information is implicit in the property name (*uniEP* and *enniEP*), this Service Attribute
637 is not included in the schema for Access E-Line, but likely would be included for other Operator Ethernet
638 Services.

639 **12.3.3 Additional OVC End Point Attributes**

640 Two additional attributes are included in the OVC End Point data model, *Ovc End Point Envelopes*, and
641 *Ovc End Point Port Conversation ID To Aggregation Link Map*. These are not in the MEF 26.2 Service
642 Attributes for OVC End Point.

643 As noted in previous sections, the model supported by this specification assumes an existing ENNI. Orders
644 for new or modified ENNIs are not supported. However, there are two items associated with a new OVC
645 that would, normally, require changes to the ENNI, the *ENNI Envelopes Service Attributes* (in the ENNI
646 Service Attributes) and the *ENNI Port Conversation ID to Aggregation Link Map* in the ENNI Common
647 Attributes.

648 Envelopes listed in the *OVC End Point Envelopes Service Attribute* can be specified when all of the Band-
649 width Profile Flows contained in the Envelopes are specific to a given OVC End Point. They must be
650 specified using one of the following OVC End Point Service Attributes:

- 651 • Ingress Bandwidth Profile Per OVC End Point
- 652 • Egress Bandwidth Profile Per OVC End Point
- 653 • Ingress Bandwidth Profile Per Class of Service Name
- 654 • Egress Bandwidth Profile Per Egress Equivalence Class Name

655 Although this specification provides for modifying Operator UNI Service Attributes, the *OVC End Point*
656 *Envelopes Service Attribute* can be used for UNI as well as ENNI (with the same limitations). To create an
657 Envelope containing Bandwidth Profile Flows for more than one OVC End Point, the UNI or ENNI Enve-
658 lopes Service Attribute (as appropriate) must be used. As noted above, modification of the ENNI Attributes,
659 including the ENNI Envelopes Service Attribute, is not supported in this specification, and if a modification
660 is required it must be agreed between the Service Provider and the Operator by other means.

661 If the ENNI uses Link Aggregation in the All-Active mode, a table that maps Port Conversation IDs (S-
662 VLAN IDs) to an ordered set of Aggregation Links is required. In order to avoid the need to modify the
663 ENNI, this can be done in the OVC End Point (in some conditions).

664 Only S-VLAN IDs that are listed in the OVC End Point Map can be listed in the OVC End Point Port
665 Conversation ID To Aggregation Link Map. If the ENNI S-VLAN Control Service Attribute is PARTIAL
666 (rather than FULL) then the Service Provider does not have full visibility of all the S-VLAN IDs mapped
667 to each physical link, and hence cannot pick values that avoid overloading certain links. In this case, the
668 ENNI Port Conversation to Aggregation Link Map Common Attribute must be agreed between the relevant
669 Operators. The means by which this is done is outside the scope of this specification.

670

671 13 Relationship Between Entities

672 This section describes the constraints and relationships between the three primary Order Items (Access E-
673 Line OVC, OperatorUNI, and ENNI).

674 The use case for Access E-Line described above is based on purchasing the Access E-Line OVC and a new
675 or existing UNI. The ENNI used for the OVC in this model is an existing ENNI.

676 The Access E-Line Service is associated with exactly one OVC (with appropriate Service Attribute values),
677 but it depends on two other products, an ENNI and a UNI. The relationship between the Access E-Line
678 OVC and the ENNI and UNI is captured in the product-agnostic part of the POQ, Quote and Order APIs.
679 The values in the Relationship Type column in the table below are used in the *relationshipType* field of the
680 *ProductRelationship*, *QualificationItemRelationship*, *QuoteItemRelationship* and *OrderItemRelationship*
681 types. Specification of the UNI and the ENNI are mandatory at INSTALL and CHANGE of the service.

682 The final column notes that during POQ and Quote, a specific UNI must be specified, but multiple ENNIs
683 might be candidates for use. Another way to think about this is that at POQ and Quote, the Access E-Line
684 service is between *one specific UNI* and one of, *possibly, several ENNIs*.

685

Product	Relationship Type	INSTALL	CHANGE	Product Specification	Multiple Allowed at POQ and Quote?
Access E-Line	CONNECTS_TO_ENNI	Mandatory	Mandatory	ENNI	Yes
Access E-Line	CONNECTS_TO_UNI	Mandatory	Mandatory	UNI	No

686 **Table 4 – Product Relationship Roles**

687 [R11] For an Access E-Line OVC product, the Relationship Type field of the Product Relationship,
688 POQ Item Relationship, Quote Item Relationship and Order Item Relationship types **MUST** contain one of the values shown in the Relationship Type column
689 in Table 4.

691 [R12] For POQ, Quote and Order, the relationship to a UNI **MUST** be specified for every
692 INSTALL of, or CHANGE to, an Access E-Line OVC product.

693 [R13] For an Access E-Line OVC product, the relationship to a UNI **MUST** reference an
694 Operator UNI product or an equivalent POQ Item, Quote Item, or Order Item.

695 [R14] For a CHANGE to an Access-E-Line OVC product the relationship to the UNI **MUST**
696 **NOT** be changed from the value present in the Product Inventory.

697 Note that [R14] indicates that once an Access E-Line OVC is associated with an Operator UNI, it cannot
698 be associated with a different Operator UNI.

699 [R15] For POQ, Quote and Order, the relationship to an ENNI **MUST** be specified for every
700 INSTALL of, or CHANGE to, an Access E-Line OVC product.

701 [R16] For an Access E-Line OVC product, the relationship to an ENNI **MUST** reference an
 702 ENNI product or an equivalent POQ Item, Quote Item, or Order Item.

703 The UNI may be included in the same order as the Access E-Line Service. The UNI is associated with a
 704 specific INSTALL_LOCATION and as noted above, it is required at INSTALL and CHANGE and once a
 705 UNI is associated with a specific location, the INSTALL_LOCATION cannot be changed. The install
 706 location is captured in the product-agnostic part of the POQ, Quote and Order APIs. The value in the Place
 707 Relationship Role column in the table below is used in the *role* field of the *RelatedPlaceRefOrValue* type.

708

Product	Place Relationship Role	INSTALL	CHANGE
Operator UNI	INSTALL_LOCATION	Mandatory	Mandatory

709 **Table 5 – Place Relationship Role**

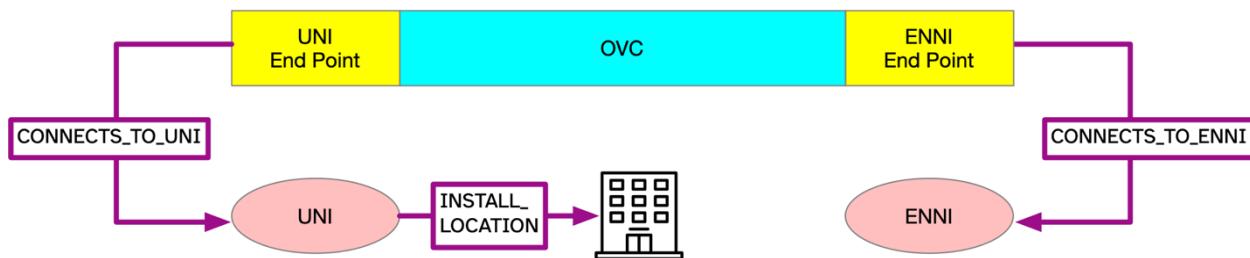
710 [R17] For an Operator UNI product, the Role field (*role*) of the Related Place (*RelatedPlaceRefOrValue*) type **MUST** contain one of the values shown in the Place Rela-
 711 tionship Role column in Table 5.

712 [R18] For POQ, Quote, and Order, the Related Place (*RelatedPlaceRefOrValue*) **MUST** be
 713 specified for every INSTALL of, or CHANGE to, an Operator UNI.

714 [R19] For a CHANGE to an Operator UNI product the Related Place **MUST NOT** be
 715 changed from the value present in the Product Inventory.

716 Changing the UNI Location is not supported for an Operator UNI. The value included in a CHANGE re-
 717 quest must be identical to the value in the Inventory.

718 The relationships applicable to Access E-Line are shown in the following diagram:



719

721 14 Access E-Line Service Attributes

722 The Service Attributes are listed in groups:

- 723 • Access E-Line OVC Service Attributes
- 724 • OVC End Point Service Attributes
- 725 • Operator UNI Service Attributes
- 726 • ENNI Service Attributes
- 727 • Operator Multilateral Attributes
- 728 • ENNI Common Attributes

729 The following sections enumerate the Service Attributes. For each Service Attribute, the first line includes
730 the Service Attribute name, the JSON Name, and the argument type and allowed values. The second line
731 contains a description and reference. Some entries include additional coding and validation notes when the
732 value of the field depends on (or in some cases is depended on) by another field.

733 Not all MEF 26.2 Service Attributes are included in the data models. 56 of the 72 Service Attributes are
734 included. The 16 Service Attributes that are not included are also listed in section 16. Some Service Attrib-
735 utes are not included because they are included in the Product Independent information portion of the API
736 (e.g., many of the Identifiers) and some Service Attributes are not included because they are constants (i.e.,
737 can only have one possible value) either because they are defined that way (e.g., the ENNI Frame Format
738 Service Attribute must be “802.3-2012) or because the value is constrained to a single value for Access E-
739 Line (e.g., the OVC Type Service attribute for an Access E-line Service must be point-to-point).

740 Conversely, there are a few Service Attributes included in the schemas that are not included in MEF 26.2
741 (see section 12.3.3).

742 The following tables are organized by schema file. The first part (in blue) specifies the filename and the list
743 of classes included in the file. Following that, for each class there is a class description (white background)
744 followed by a list of properties in the class (yellow background) and, in some cases, validation notes (light
745 red background).

746 14.1 Access E-Line OVC and OVC End Point Service Attributes

Access Eline OVC

Schema File Name: accessElineOvc
urn:mef:iso:spec:sonata:access-eline:v4.0.0:common

Includes classes:

AccessElineOvc (Access E-Line OVC Service Attributes)
AccessElineOvcEndPoint (Access E-Line OVC End Point Service Attributes)

Class: AccessElineOvc (**Access E-Line OVC Service Attributes**)

OVC Service Attributes control the behavior observable at and between External Interfaces to the Carrier Ethernet Network (CEN). The behaviors are achieved by the Network Operator and the Operator's client (the Service Provider in this case) agreeing on the value for each of the Service Attributes. The Access E-Line Service is defined in MEF 51.1 (Operator Ethernet Service Definitions) based on the Service Attributes defined in MEF 26.2 (External Network Network Interfaces and Operator Service Attributes).



<p>Property Name: Access E-Line OVC Service Attributes Type: object</p> <p>Allowed values: Compose (allOf): Class: AccessElineOvcCommon</p>
<p>Property Name: UNI OVC End Point - Access E-Line Reference: MEF 26.2 sec. 16 JSON Name: uniEp - Type: object</p> <p>Description: The OVC EP object for the OVC EP at the UNI. The UNI OVC End Point must be included in the Access E-Line Product.</p> <p>Allowed values: Class: AccessElineOvcEndPoint</p>
<p>Property Name: ENNI OVC End Point - Access E-Line Reference: MEF 26.2 sec. 16 JSON Name: enniEp - Type: object</p> <p>Description: The OVC EP object for the OVC EP at the ENNI. The ENNI OVC End Point must be included in the Access E-Line Product.</p> <p>Allowed values: Class: AccessElineOvcEndPoint</p>
<p>Class: AccessElineOvcEndPoint (Access E-Line OVC End Point Service Attributes)</p> <p>An OVC End Point is a logical entity at a given External Interface that is associated with a distinct set of frames passing over that External Interface. The OVC End Point schema is included in the definitions section of the OVC schema.</p>
<p>Property Name: Access Eline OVC EP Service Attributes Type: object</p> <p>Allowed values: Compose (allOf): Class: AccessElineOvcEpCommon</p>

747

<p>Access Eline OVC Common</p> <p>Schema File Name: accessElineOvcCommon</p> <p>Includes classes:</p> <p>AccessElineOvcCommon (Access E-Line OVC Common)</p>
<p>Class: AccessElineOvcCommon (Access E-Line OVC Common)</p> <p>This class contains all of the Access E-Line OVC Services Attributes. This class is used to compose the AccessElineOvc class.</p>
<p>Property Name: OVC Maximum Frame Size Service Attribute Reference: MEF 26.2 sec. 12.6 JSON Name: maximumFrameSize - Type: integer</p>

Description: The max length in bytes allowed in a frame mapped to an OVC EP associated by the OVC. The length of the frame includes all bytes from the start of the destination MAC address through the Frame Check Sequence. This must be at least 1526 (1518+C-tag+S-tag).

Allowed values: Minimum: 1526

Validation Notes: This should be validated against maximum frame size at the external interfaces.

- It must be less than or equal to **OperatorMultilateral.maximumFrameSizeMultilateral**, and
- It must less than **OperatorUNI.maximumServiceFrameSize** + 4 (for an OVC with multiple UNIs, this is compared to the UNI with the smallest maximum frame size.)

Property Name: OVC CE-VLAN ID Preservation Service Attribute

Reference: MEF 26.2 sec. 12.7

JSON Name: ceVlanIdPreservation - **Type:** enum

Description: Determines whether the C-tag in the Ingress Frame is Stripped or Preserved in the Egress Frame. Per MEF 51.1, the Operator **must** support *PRESERVE* and **should** support *STRIP*.

Allowed values: Enum: "PRESERVE", "STRIP", "RETAIN"

Property Name: OVC CE-VLAN PCP Preservation Service Attribute

Reference: MEF 26.2 sec. 12.8

JSON Name: cTagPcpPreservation - **Type:** object

Description: If *ENABLED*, then the value of the C-tag PCP field is preserved from Ingress to Egress. Per MEF 51.1, the Operator **must** support *ENABLED* and **should** support *DISABLED*.

Allowed values: Type: EnabledDisabled

Validation Notes: If this Service Attribute has a value of Enabled, then the value of

- AccessElineOvc.ceVlanIdPreservation **must not** be *Strip*.

Property Name: OVC CE-VLAN DEI Preservation Service Attribute

Reference: MEF 26.2 sec. 12.9

JSON Name: cTagDeiPreservation - **Type:** object

Description: If *ENABLED*, then the value of the C-Tag DEI field is preserved from Ingress to Egress. Per MEF 51.1, the Operator **must** support *ENABLED* and **should** support *DISABLED*.

Allowed values: Type: EnabledDisabled

Validation Notes: If this Service Attribute has a value of Enabled, then the value of *AccessElineOvc.ceVlanIdPreservation* **must not** be *Strip*.

Property Name: OVC List of Class of Service Names Service Attribute

Reference: MEF 26.2 sec. 12.11

JSON Name: listOfClassOfServiceNames - **Type:** List of string

Description: The OVC List of Class of Service Names Service Attribute is used to specify all of the Class of Service Names supported by an OVC. The value of the OVC List of Class of Service Names Service Attribute is a non-empty list of Class of Service Names (which may include one or more of the CoS Labels defined in MEF 23.2).

Allowed values: List of [1...*] Unique [Any String]

Property Name: OVC Service Level Specification Service Attribute

Reference: MEF 26.2 sec. 12.13

JSON Name: carrierEthernetSls - **Type:** List of object

Description: Technical details of the service level in terms of Performance Objectives, agreed between the SP/SO and the Operator as part of the Service Level Agreement. For each Performance Metric included in the SLS, a Performance Objective must be included as well as the parameter values needed for the Performance Metric. If the value of this attribute is an empty list, it is equivalent to a value of *None* per MEF 26.2.

Allowed values: List of [0...1] Class: CarrierEthernetSls

Property Name: OVC Frame Delivery Service Attribute

Reference: MEF 26.2 sec. 12.14

JSON Name: frameDisposition - **Type:** object

Description: A 3-tuple that indicates how Unicast, Multicast, and Broadcast Frames are delivered by the service. For each, the value can be *UNCONDITIONAL*, the frames are delivered without condition, *CONDITIONAL*, the frames are delivered based on conditions agreed to between Buyer and Seller (e.g., based on learned addresses), and *DISCARD*. If *CONDITIONAL* is chosen for any of the elements, the conditions must be agreed on by the Service Provider and Operator by some other means.

Allowed values: Class: FrameDisposition

Property Name: OVC Available MEG Level Service Attribute

Reference: MEF 26.2 sec. 12.15

JSON Name: availableMegLevel - **Type:** object

Description: The lowest MEG level for which SOAM Frames are not peered or discarded by the Service Provider. If this attribute is NONE there is no such level (that is, SOAM frames at all MEG levels may be peered or discarded by the Service Provider). MEF 51.1 constrains it to 0-6 for Access E-Line.

Allowed values: Type: AvailableMegList

Property Name: OVC L2CP Address Set Service Attribute

Reference: MEF 45 sec. 8.1 and MEF 26.2 sec. 12.16

JSON Name: ovcL2cpAddressSet - **Type:** object

Description: The subset of the Bridge Reserved Addresses that are Peered or Discarded (i.e. not Passed). The value *CTB* or *CTB2* must be used for port-based (private) services and *CTA* for vlan-based (virtual private) services.

Allowed values: Type: L2cpAddressSet

Validation Notes: If **UniEP.ovcEndPointMap** does not include all C-tags (i.e. 1-4094) this **must** be *CTA* otherwise it must be *CTB* or *CTB2*

**Access Eline OVC EP Common**

Schema File Name: accessElineOvcEpCommon

Includes classes:

AccessElineOvcEpCommon (Access Eline OVC EP Common)

Class: AccessElineOvcEpCommon (**Access Eline OVC EP Common**)

This class contains all of the Access E-Line OVC End Point Services Attributes. This class is used to compose the AccessElineOvcEndPoint class.

Property Name: OVC End Point Identifier Service Attribute**Reference:** MEF 26.2 sec. 16.1**JSON Name:** identifier - **Type:** string

Description: The value of the OVC EP ID Service Attribute is a string that is used to allow the Subscriber and Service Provider to uniquely identify the OVC EP for operations purposes.

Allowed values: String: Min Length: 1 Max Length: 45 Pattern (regex): [\x20-\x7F]+

Property Name: OVC End Point Map Service Attribute**Reference:** MEF 26.2 sec. 16.5**JSON Name:** ovcEndPointMap - **Type:** object

Description: The information that determines which UNI/ENNI Frames are mapped to the OVC End Point. At the UNI and ENNI, the only allowable mapping is a LIST of VLAN IDs (CE-VLAN IDs at the UNI and S-VLAN IDs at the ENNI).

Allowed values: Select (oneOf): Class: EndPointMapFormU, Class: EndPointMapFormE **with Discriminator:** mapType

Validation Notes: For a given UNI, the values of all of the OVC End Point Map Service Attribute **must** be such that each possible CE-VLAN ID value maps to at most one OVC End Point.

For a given ENNI, the values of all of the OVC End Point Map Service Attributes for all Service Providers using the ENNI **must** be such that a Tagged ENNI Frame maps to at most one OVC End Point.

An OVC associating an OVC End Point at a UNI whose value of the OVC End Point Map Service Attribute contains multiple CE-VLAN ID values **must** have the value of the OVC CE-VLAN ID Preservation Service Attribute = Preserve.

Coding hint: This must be FormU at UNI and FormE at ENNI. They can only use LIST (not ALL or UT/PT)

Property Name: OVC End Point Class of Service Identifier Service Attribute**Reference:** MEF 26.2 sec. 16.6**JSON Name:** ingressClassOfServiceMap - **Type:** object

Description: The value of the OVC EP Ingress Class of Service Map Service Attribute is a 3-tuple of the form (F, M, P) where: F is one of the values *ENDPOINT*, *C_TAG_PCP*, *S_TAG_PCP*, or *DSCP*, M is a map that can be used to assign Class of Service Names to External Interface Frames. (The form of M depends on the value of F.), and P is a map with entries of the form (Layer 2 Control Protocol type → Class of Service Name) where the Layer 2 Control Protocol type is determined by the Protocol Identifier (see Section 6.2 of MEF 45.1).

Allowed values: Select (oneOf): Class: CosFromEp, Class: CosFromStagPcp, Class: CosFromCtagPcp, Class:CosFromDscp **with Discriminator:** mapType

Validation Notes: Per MEF 26.2, at UNI End Point this must be *ENDPOINT*, *C_TAG_PCP*, or *DSCP*. At ENNI End Point this must be *S_TAG_PCP*.

The Class of Service Name or Names **must** be included in the *ovc.listOfClassOfServiceNames*.

Property Name: OVC End Point Color Identifier Service Attribute**Reference:** MEF 26.2 sec. 16.7**JSON Name:** colorMap - **Type:** object

Description: The OVC EP Color Map Service Attribute is the mechanism by which the Color for an Ingress External Interface Frame that is mapped to an OVC EP is indicated by the content in the frame header. The value of the OVC EP Color Map Service Attribute is a pair of the form (F, M) where: F is one of the values *ENDPOINT*, *DEI*, *PCP*, or *DSCP* and M is a map that can be used to assign Color to each Ingress Frame. (The form of M depends on the value of F.)

Allowed values: Select (oneOf): Class: ColorFromDei, Class: ColorFromPCP, Class: ColorFromDSCP, Class:ColorFromEp **with Discriminator:** mapType

Validation Notes: MEF 51.1 indicates that the value of this **SHOULD** be DEI (C-tag DEI) at the UNI and at the ENNI (S-tag DEI).

Property Name: OVC End Point Egress Map Service Attribute**Reference:** MEF 26.2 sec. 16.8**JSON Name:** ovcEgressMap - **Type:** List of object

Description: A set of mappings that determine the content of the S-Tag or C-Tag of an egress frame at an ENNI or UNI (respectively) based on the Class of Service Name and Frame Color of the frame. Three forms: (1) CoSName(CN) -> C/S-Tag PCP or Discard, (2) CoSName and Frame Color (CC)--> C/S-Tag DEI or Discard, and (3) CoSName and Frame Color (CC)--> C/S-Tag PCP or Discard. A fourth form is included which combines (1) and (2) because that is a common case.

Allowed values: List of [0...1] Select (oneOf): Class: PcpFromCos, Class: DeiFromCosColor, Class: PcpFromCosColor, Class:PcpFromCosDeiFromCosColor **with Discriminator:** mapType

Validation Notes:

1. At the UNI EP this maps into the C-tag PCP/DEI and at the ENNI EP this maps into the S-tag PCP/DEI.
2. The Egress Map specification can be one of the following depending on other aspects of the OVC:
 - Not specified
 - CN-->PCP only
 - CC-->DEI only
 - CC-->PCP only
 - CN-->PCP and CC-->DEI

Property Name: OVC End Point Egress Equivalence Class Identifier Service Attribute**Reference:** MEF 26.2 sec. 16.9**JSON Name:** egressEquivalenceClassIdentifier - **Type:** object

Description: The mechanism that allows an Egress Equivalence Class Name to be determined for an egress EI Frame. (F, M, P) where F is a protocol field in the egress EI Frame, M is a map that maps each possible value of the field F and the absence of the field F to an Egress Equivalence Class Name and P is a map of L2CP type to Egress Equivalence Class Name.

Allowed values: Select (oneOf): Class: CosFromEp, Class: CosFromStagPcp, Class: CosFromCtagPcp, Class:CosFromDscp **with Discriminator:** mapType

Property Name: Egress Bandwidth Profile per OVC End Point Service Attribute**Reference:** MEF 26.2 sec. 16.11**JSON Name:** egressBandwidthProfilePerEndPoint - **Type:** List of object

Description: Bandwidth Profile Flow parameters for all egress Service Frames mapped to the OVC End Point. If the value of this attribute is an empty list, there is no Egress Bandwidth Profile per OVC at the OVC End Point.



Allowed values: List of [0...1] Class: BwpFlow

Validation Notes: Per MEF 51.1, if included in the request, this **must** be Disabled (i.e., an empty list) at ENNI EP but not at the UNI.

If this attribute is a non-empty list at the UNI, **AccessElineOvcEndPoint.egressBwpPerEgressEquivalenceClassName** must not be included in the request or must be included as an empty list.

Coding hint: Per MEF 51.1 this must be Disabled (an empty list) at ENNI EP

Property Name: Ingress Bandwidth Profile per Class of Service Name Service Attribute

Reference: MEF 26.2 sec. 16.12

JSON Name: ingressBandwidthProfilePerClassOfServiceName - **Type:** List of object

Description: For each CoS Name listed, Bandwidth Profile Flow parameters for all ingress Service Frames mapped to that CoS Name at OVC End Point. This bandwidth profile flow is used to police at Ingress each CoS in an OVC individually. List of 2-tuples {cos, bwpf}.

Allowed values: List of [1...*] Unique Class: BandwidthProfilePerClassOfServiceName

Validation Notes: Per MEF 51.1, the list must contain each CoS Name identified in the **AccessElineOvcEndPoint.ingressClassOfServiceMap** except for the Class of Service name *Discard*. Additionally, at the ENNI, for each Bandwidth Profile specified, the Operator SHOULD support values of 0 and 4 for the *token offset* parameter (F).

Property Name: Egress Bandwidth Profile per Egress Equivalence Class Name Service Attribute

Reference: MEF 26.2 sec. 16.13

JSON Name: egressBwpPerEgressEquivalenceClassName - **Type:** List of object

Description: For each EEC Name listed, Bandwidth Profile Flow parameters, for all egress Service Frames mapped to that EEC Name at the OVC End Point. If the value of this attribute is an empty list, there is no Egress Bandwidth Profile per EEC at the OVC End Point.

Allowed values: List of [0...*] Class: BandwidthProfilePerClassOfServiceName

Validation Notes: If specified, **AccessElineOvcEpCommon.egressEquivalenceClassIdentifier** **MUST** be specified and **AccessElineOvcEndPoint.egressBandwidthProfilePerEndPoint** **MUST NOT** be specified.

Property Name: OVC End Point Aggregation Link Depth Service Attribute

Reference: MEF 26.2 sec. 16.14

JSON Name: aggregationLinkDepth - **Type:** List of object

Description: A list of 2-tuples {SVID, Depth} that indicates the number of ENNI links that can carry ENNI Frames for each S-VLAN ID mapped to the OVC End Point. It indicates the level of resiliency to use for the SVID at the ENNI.

Allowed values: List of [0...*] Unique Class: AggLinkDepth

Validation Notes:

1. Ensure that this is not included in UniOvcEP.
2. The ENNI must be Link Aggregation type ALL ACTIVE
3. The Depth element must not be greater than the number of links in the ENNI
4. **EnniCommon.portconversation** must be configured to enable the depth specified by this attribute.

Not required in Inventory unless the ENNI has ALL ACTIVE Link Aggregation.

Property Name: OVC End Point Source MAC Address Limit Service Attribute**Reference:** MEF 26.2 sec. 16.15**JSON Name:** sourceMacAddressLimit - **Type:** List of object

Description: Specifies a limit on the number of different Source MAC address over a time interval for which ingress Service Frames at this OVC End Point will be delivered. An empty list corresponds to a Service Attribute value of *None*. The value is {N,t} where N is an integer greater than or equal to 1 and t is a time duration.

Allowed values: List of [0...1] Class: SourceMacAddressLimit

Property Name: OVC End Point MIP Service Attribute**Reference:** MEF 26.2 sec. 16.16**JSON Name:** maintenanceIntermediatePoint - **Type:** object

Description: Indicates whether a MIP is instantiated at the OVC End Point and the MEG Level of the MIP. When the value is not *None*, several parameter values need to be determined as described in MEF 30.1.

Allowed values: Type: MegList

Property Name: OVC End Point Maintenance End Point List Service Attribute**Reference:** MEF 26.2 sec. 16.17**JSON Name:** maintenanceEndPointList - **Type:** List of object

Description: A list of the MEPs instantiated at this OVC EP, if any, and the direction of each MEP, List of 2-tuples (l, d) where l is MEG level and d is direction (Up/Down).

Allowed values: List of [0...16] Unique Class: MepLevelAndDirection

Validation Notes: Verify that each MEG level appears at most once in the list for each direction.

Property Name: OVC End Point Envelopes Service Attribute**JSON Name:** ovcEndPointEnvelopes - **Type:** List of object

Description: The Envelopes and Envelope Coupling Flag values to which Bandwidth Profile Flows for this OVC End Point can be mapped. Value is list of {envelopeID, couplingFlagForIndexZero}. Envelopes listed in the OVC End Point Envelopes Service Attribute can be used when all of the Bandwidth Profile Flows contained in the Envelope are specific to a given OVC End Point. They are specified using one of the following OVC End Point Service Attributes:

- Ingress Bandwidth Profile Per OVC End Point
- Egress Bandwidth Profile Per OVC End Point
- Ingress Bandwidth Profile Per Class of Service Name
- Egress Bandwidth Profile Per Egress Equivalence Class Name

To create an Envelope containing Bandwidth Profile Flows for more than one OVC End Point, the UNI or ENNI Envelopes Service Attribute (as appropriate) must be used.

Allowed values: List of [0...*] Unique Class: Envelope

Validation Notes: All Bandwidth Profile Flows in the specified envelopes must be for this OVC End Point. The envelopIDs specified in this Service Attribute must be unique with the envelopIDs specified in the UNI or ENNI Envelopes Service Attribute (as appropriate for the OVC End Point).



Property Name: OVC End Point Port Conversation Map Service Attribute

Reference: MEF 26.2 sec. 13.2

JSON Name: `ovcEndPointPortConversationToAggregationLinkMap` - **Type:** List of object

Description: A mapping of Port Conversation IDs (i.e., VLAN IDs) to an ordered list of physical links if **EnniService.sVlanId-Control** is *FULL* and there are multiple physical links in the ENNI and **EnniCommon.linkAggregation** is *ALL_ACTIVE*. Only S-VLAN IDs that are listed in the **AccessElineOvcEndPoint.ovcEndPointMap** can be listed in this attribute.

If **EnniService.sVlanIdControl** is *PARTIAL* then the Service Provider does not have full visibility of all the S-VLAN IDs mapped to each physical link, and hence cannot pick values that avoid overloading certain links. In this case, the ENNI Port Conversation to Aggregation Link Map Common Attribute must be agreed between the relevant Operators.

Allowed values: List of [0...*] Unique Class: ConversationIdToAggregationLinkMap

Validation Notes: This can only be specified for an OVC End Point at an ENNI. If this is a non-empty list, then **EnniService.sVlanId-Control must be FULL** and **EnniCommon.linkAggregation must be ALL_ACTIVE**.

Coding hint: Not used for OVC End Point at UNI (only ENNI)

751

14.2 Operator UNI Service Attributes

Carrier Ethernet Operator UNI

Schema File Name: carrierEthernetOperatorUni
urn:mef:iso:spec:sonata:carrier-ethernet-operator-uni:v4.0.0:common
Includes classes:
OperatorUNI (Operator UNI Service Attributes)

Class: OperatorUNI (**Operator UNI Service Attributes**)

Operator UNI Service Attributes describe the agreement between the Service Provider and the Operator of the behavior observable at the UNI. These are distinct from the Subscriber UNI Service Attributes that describe the agreement between an Ethernet Service (EVC) Subscriber and the Ethernet Service Provider. All of these Service Attributes can be modified after initial provisioning of the UNI except the components of the List of Physical Links and Link Aggregation-related attributes.

Property Name: Operator UNI Service Attributes **Type:** object

Allowed values: Compose (allOf): Class: OperatorUniCommon

752

Carrier Ethernet Operator UNI Common

Schema File Name: carrierEthernetOperatorUniCommon
Includes classes:
OperatorUniCommon (Operator UNI Common)

Class: OperatorUniCommon (**Operator UNI Common**)

This class contains all of the Operator UNI Services Attributes. This class is used to compose the OperatorUNI class.

Property Name: Operator UNI List of Physical Links Service Attribute

Reference: MEF 26.2 sec. 14.2, 14.3

JSON Name: listOfPhysicalLinks - **Type:** List of object

Description: This Service Attribute combines several of the MEF 26.2 Objects into a single n-tuple that is consistent with the way this information is described in MEF 10.4. The value of the Operator UNI List of Physical Links Service Attribute is a list of 6-tuples of the form (id, physicalLink, synchronousEthernet, precisionTiming, uniConnectorType,uniConnectorGender), with one list item for each physical link. The value of **id** is an identifier for the physical link. The value of **physicalLink** is the specific Ethernet physical layer type. **synchronousEthernet** indicates if synchronous Ethernet is used on the physical link and has the value Enabled or Disabled. The value of **precisionTiming** indicates if the Precision Time Protocol is used on the physical link and has the value Enabled or Disabled. The values of **uniConnectorType** and **uniConnectorGender** specify the type of connector used. Note that the **id** and the **physicalLink** come from the CarrierEthernetPhysicalLink class.

Allowed values: List of [1...*] Unique Type: UniPhysicalLink

Property Name: Operator UNI Link Aggregation Service Attribute

Reference: MEF 26.2 sec. 14.5

JSON Name: linkAggregation - **Type:** object

Description: Service Attribute representing Link Aggregation types. **NONE**-No Link Aggregation is used. **2_LINK_ACTIVE-STANDBY**-The SP network uses Link Aggregation as in either Clause 5.6.1 of IEEE Std 802.1AX-2008 [2] or Clause 6.7.1 of IEEE Std 802.1AX-2014 with one Link Aggregation Group (LAG) across the links supporting the UNI such that all Service Frames are carried on only one of the two links when both links are operational. **ALL_ACTIVE**-The SP Network uses Link Aggregation as specified in Clause 5.3 of IEEE Std 802.1AX-2014, including the use of the version 2 LACPDUs as specified in

Clause 5.3.1h of IEEE Std 802.1AZ-2014, with one Link Aggregation Group (LAG) across the links. **OTHER**-The Operator/Subscriber/Service Provider agree on another other resiliency mechanism.

Allowed values: Type: LinkAggType

Validation Notes: If the number of entries in *OperatorUniCommon.listOfPhysicalLinks* =

- 1: The value must be NONE
- 2: The value must be 2_LINK_ACTIVE-STANDBY, ALL_ACTIVE, or OTHER
- >2: The value must be ALL_ACTIVE or OTHER

Property Name: *Operator UNI Port Conversation ID to Aggregation Link Map Service Attribute*

Reference: MEF 26.2 sec. 14.6 and 802.1AX-2014 sec. 6.6.2.1

JSON Name: *aggregationLinkMap* - **Type:** List of object

Description: A mapping of a list of Port Conversation IDs (i.e., VLAN IDs or zero for untagged frames) to an ordered list of physical links if there are multiple physical links in the UNI and the link aggregation is ALL_ACTIVE.

Allowed values: List of [0...*] Unique Class: ConversationIdToAggregationLinkMap

Validation Notes: This must be empty if the value of *OperatorUniCommon.linkAggregation* is not ALL_ACTIVE or OTHER. If it is not empty, it must map every C-VLAN ID mapped to the OVC.

Property Name: *Operator UNI Maximum Service Frame Size Service Attribute*

Reference: MEF 26.2 sec. 14.8

JSON Name: *maximumServiceFrameSize* - **Type:** integer

Description: Specifies the maximum size of Service Frames that can be transmitted across the UNI.

Allowed values: Minimum: 1522

Validation Notes: *Ovc.maximumFrameSize* cannot be more than 4 bytes greater than this attribute.

Property Name: *Operator UNI Default CE-VLAN ID Service Attribute*

Reference: MEF 26.2 sec. 14.9

JSON Name: *defaultCeVlanId* - **Type:** integer

Description: The value of the Operator UNI Default CE-VLAN ID is an integer that is the CE-VLAN ID value assigned to Untagged Service Frames and Priority Tagged Service Frames.

Allowed values: Minimum: 1 Maximum: 4094

Property Name: *Operator UNI Maximum Number of OVC End Points Service Attribute*

Reference: MEF 26.2 sec. 14.10

JSON Name: *maximumNumberOfEndPoints* - **Type:** integer

Description: The maximum number of OVC End Points that the Operator CEN can support at the UNI.

Allowed values: Minimum: 1 Maximum: 4094

Property Name: Operator UNI Maximum Number of CE-VLAN IDs per End Point Service Attribute**Reference:** MEF 26.2 sec. 14.11**JSON Name:** maximumNumberOfCeVlanIdsPerEndPoint - **Type:** integer

Description: An integer between 1 and 4094 inclusive that limits the number of C-Tag VLAN IDs that can map to each OVC End Point at the UNI.

Allowed values: Minimum: 1 Maximum: 4094

Property Name: Operator UNI Ingress Bandwidth Profile per UNI Service Attribute**Reference:** MEF 26.2 sec. 14.12**JSON Name:** ingressBandwidthProfile - **Type:** List of object

Description: A single set Bandwidth Profile Flow parameters that covers all OVCs at the UNI. The value is either Disabled (empty array) or a BWP.

Allowed values: List of [0...1] Class: BwpFlow

Validation Notes: If this attribute includes a Bandwidth Profile Flow, then the OVC EPs at this UNI cannot have Ingress BWP Per OVC EP or Ingress BWP per CoS Name (consistent with R233 in MEF 26.2). This attribute **MUST** not be used for Access E-Line since that service requires Ingress BWP per CoS Name according to MEF 51.1.

Property Name: Operator UNI Egress Bandwidth Profile per UNI Service Attribute**Reference:** MEF 26.2 sec. 14.13**JSON Name:** egressBandwidthProfile - **Type:** List of object

Description: A single set of Bandwidth Profile Flow parameters that covers all OVCs at the UNI. The value is either Disabled (empty array) or a BWP.

Allowed values: List of [0...1] Class: BwpFlow

Validation Notes: If this is specified then none of the OVC EPs at the UNI can have an Egress BWP per OVC EP or an Egress BWP per EEC Name (as this would break R233 in MEF 26.2).

Property Name: Operator UNI Link OAM Service Attribute**Reference:** MEF 26.2 sec. 14.14**JSON Name:** linkOam - **Type:** object

Description: Controls when and how Link OAM per IEEE Std 802.3-2015 is run on the physical links in the UNI.

Allowed values: Type: EnabledDisabled

Property Name: Operator UNI MEG Service Attribute**Reference:** MEF 26.2 sec. 14.15**JSON Name:** meg - **Type:** object

Description: Indicates if the Service Provider has instantiated a MEG End Point (MEP) at the UNI Maintenance Entity Group (MEG) Level (**ENABLED**) or not (**DISABLED**).

Allowed values: Type: EnabledDisabled

Property Name: Operator UNI LAG Link MEG Service Attribute**Reference:** MEF 26.2 sec. 14.16**JSON Name:** lagLinkMeg - **Type:** object

Description: Indicates if the Service Provider has instantiated a LAG Link MEG End Point (MEP) on each physical link in the UNI if the UNI has more than one physical link (**ENABLED**) or not (**DISABLED**).

Allowed values: Type: EnabledDisabled

Validation Notes: Must not be enabled if **OperatorUNI.linkAggregation = NONE**.

Property Name: Operator UNI Token Share Service Attribute**Reference:** MEF 26.2 sec. 14.18**JSON Name:** tokenShare - **Type:** object

Description: An attribute that indicates whether Bandwidth Profile Envelopes containing more than one Bandwidth Profile Flow are supported by the Service Provider at the Operator UNI.

Allowed values: Type: EnabledDisabled

Validation Notes: If this attribute is **DISABLED**, every Envelope at the UNI and every Envelope at the OVC End Point must contain at most 1 BWP Flow.

Property Name: Operator UNI Envelopes Service Attribute**Reference:** MEF 26.2 sec. 14.19**JSON Name:** envelopes - **Type:** List of object

Description: The Envelopes and Envelope Coupling Flag values to which Bandwidth Profile Flows can be mapped. Value is list of {envelopeID, couplingFlagForIndexZero}. **Note:** Envelopes for the UNI can also be specified at the OVC End Points associated with the UNI.

Allowed values: List of [0...*] Unique Class: Envelope

Validation Notes: If the **couplingFlagForIndexZero** = 1 (TRUE) for this service attribute, then the **couplingFlag** for each Bandwidth Profile Flow in the Envelope must be 0 (FALSE).

Property Name: Operator UNI L2CP Address Set Service Attribute**Reference:** MEF 26.2 sec. 14.20 and MEF 45.1 sec. 8.1**JSON Name:** l2cpAddressSet - **Type:** object

Description: The subset of the Bridge Reserved Addresses that are Peered or Discarded (i.e. not Passed). The value **CTB** or **CTB2** must be used for port-based (private) services and **CTA** for vlan-based (virtual private) services. Per MEF 45.1: If an OVC has an OVC End Point at a UNI, then the value of the Operator UNI L2CP Address Set Service Attribute MUST be the same as the OVC L2CP Address Set Service Attribute.

Allowed values: Type: L2cpAddressSet

Validation Notes: If **UniEP.ovcEndPointMap** does not include all C-tags (i.e. 1-4094) or this **must** be **CTA** otherwise it must be **CTB** or **CTB2**. This value **must** be the same as **AccessElineOvc.l2cpAddressSet**.

Property Name: Operator UNI L2CP Peering Service Attribute**Reference:** MEF 26.2 sec. 14.21 and MEF 45.1 sec. 8.1**JSON Name:** l2cpPeering - **Type:** List of object



Description: L2CP Peering Service Attribute defined in MEF 45.1 specifies which protocols, if any, are peered at the UNI. The L2CP Peering service attribute value must be an empty list, or a list of entries identifying protocols to be Peered where each entry consists of {Destination Address, Protocol Identifier} or {Destination Address, Protocol Identifier, Link Identifier}.

Allowed values: List of [0...*] Class: L2cpPeering

Validation Notes: If **AccessElineOvc.I2cpAddressSet** = **CTB**, this **must not** have a Destination Address that is in Table 6 but not in the CTB subset of Table 6 in MEF 45.1.

754 **14.3 ENNI Attributes**

755 **14.3.1 ENNI Attributes**

Carrier Ethernet ENNI SP SO

Schema File Name: carrierEthernetEnniSpSo
urn:mef:iso:spec:sonata:carrier-ethernet-enni-sp-so:v4.0.0:common
Includes classes:
EnniSpso (ENNI SP/SO Attributes)

Class: EnniSpso (**ENNI SP/SO Attributes**)

This class is the composition of the ENNI Service Attributes, ENNI Common Attributes, and Operator Multilateral Attributes. The ENNI Common Attributes and Operator Multilateral Attributes are all optional since it is only required to share these if the SVLAN Control Service Attribute has the value FULL.

Property Name: ENNI Service **Type:** object

Allowed values: Compose (allOf): Class: EnniService, Class: EnniCommon, Class: OperatorMultilateral

756 **14.3.2 ENNI Common Attributes**

Carrier Ethernet ENNI Common

Schema File Name: carrierEthernetEnniCommon
Includes classes:
EnniCommon (ENNI Common Attributes)

Class: EnniCommon (**ENNI Common Attributes**)

These are the technical details that need to be agreed to by the Network Operators in order for their Carrier Ethernet Networks to be interconnected by the ENNI.

Property Name: ENNI Peering Identifier Common Attribute

Reference: MEF 26.2 sec. 9.1

JSON Name: peeringIdentifier - **Type:** string

Description: An identifier for the ENNI intended for operations purposes by the interconnecting Operators at the ENNI.

Allowed values: String: Min Length: 1 Max Length: 45 Pattern (regex): [\x20-\x7F]+

Property Name: ENNI List of Physical Links

Reference: MEF 26.2 sec. 9.2 and 9.3

JSON Name: listOfPhysicalLinks - **Type:** List of object

Description: This Service Attribute combines several of the MEF 26.2 Objects into a single n-tuple that is consistent with the way this information is described in MEF 10.4. The value of the ENNI List of Physical Links Service Attribute is a list of 2-tuples of the form <(id,pl,>, with one list item for each physical link. The value of **id** is an identifier for the physical link. The value of **pl** specifies a physical layer.

Allowed values: List of [1...*] Unique Type: EnniPhysicalLink

Property Name: ENNI Link Aggregation Common Attribute**Reference:** MEF 26.2 sec. 9.5**JSON Name:** linkAggregation - **Type:** object

Description: If the ENNI is composed of multiple physical links this Service Attribute indicates how they are combined using Link Aggregation.

Allowed values: Type: LinkAggType

Validation Notes: This needs to be validated against the x= cardinality **EnniCommon.listOfPhysicalLinks**. If x=1 this must be "NONE". If x=2 this can be any of the allowed values other than "NONE". If x>2, this must be "ALL_ACTIVE" or "OTHER"

Property Name: ENNI Port Conversation ID to Aggregation Link Map Common Attribute**Reference:** MEF 26.2 sec. 9.6**JSON Name:** portConversation - **Type:** List of object

Description: A mapping of Port Conversation IDs (i.e., VLAN IDs or zero for untagged frames) to an ordered list of physical links if there are multiple physical links in the UNI and the link aggregation is ALL_ACTIVE. **Note:** In some cases the Port Conversation ID to Aggregation Link Map can be specified at the OVC End Point.

Allowed values: List of [0...*] Unique Class: ConversationIdToAggregationLinkMap

Validation Notes: This attribute needs to be verified against **EnniEP.aggregationLinkDepth**. The level of resiliency specified for the S-VLAN ID (Port Conversation ID) has to match the aggregationLinkDepth specified for the OVC (ENNI OVC EP).

Property Name: ENNI MEG Common Attribute**Reference:** MEF 26.2 sec. 9.7**JSON Name:** meg - **Type:** object

Description: Is a Down ENNI MEG MEP Enabled at the ENNI as described in MEF 30.1?

Allowed values: Type: EnabledDisabled

Property Name: ENNI LAG Link MEG Common Attribute**Reference:** MEF 26.2 sec. 9.8**JSON Name:** lagLinkMeg - **Type:** object

Description: If Enabled the Service Provider MUST operate the LAG Link MEG on each link in the ENNI.

Allowed values: Type: EnabledDisabled

Property Name: ENNI Link OAM Common Attribute**Reference:** MEF 26.2 sec. 9.9**JSON Name:** linkOAM - **Type:** object

Description: Indicates whether Link OAM as described in Clause 57 of IEEE Std. 802.3 is supported on the ENNI.

Allowed values: Type: EnabledDisabled

757

758 **14.3.3 ENNI Service Attributes****Carrier Ethernet ENNI Service Attributes**

Schema File Name: carrierEthernetEnniServiceAttributes



Includes classes:
EnniService (ENNI Service Attributes)

Class: EnniService (**ENNI Service Attributes**)

For each instance of an ENNI, there are multiple sets of ENNI Service Attributes. The value for each ENNI Service Attribute in a set for an Operator CEN is specific to a SP/SO that is using the ENNI per Section 8.11 of MEF 26.2. Each such value is agreed to by the SP/SO and the Operator.

Property Name: S-VLAN ID Control Service Attribute

Reference: MEF 26.2 sec. 13.2

JSON Name: sVlanIdControl - **Type:** enum

Description: The value of the S-VLAN ID Control Service Attribute is FULL or PARTIAL. When the value is FULL, the Operator can support only a single SP/SO at the ENNI. In other words, S-VLAN IDs at the ENNI can only be mapped to OVC End Points for a single SP/SO. When the value is PARTIAL, there can be multiple SP/SOs using the ENNI and different S-VLAN IDs can be mapped to OVC End Points for different SP/SOS.

Allowed values: Enum: "FULL", "PARTIAL"

Property Name: Maximum Number of OVCs Service Attribute

Reference: MEF 26.2 sec. 13.3

JSON Name: maximumNumberOfOvcs - **Type:** integer

Description: The value of the Maximum Number of OVCs Service Attribute is a strictly positive integer. It provides an upper bound on the number of OVCs that the Operator will support at the ENNI for the SP/SO.

Allowed values: Minimum: 1

Property Name: Maximum Number of OVC End Points per OVC Service Attribute

Reference: MEF 26.2 sec. 13.4

JSON Name: maximumNumberOfOvcEndPointsPerOvc - **Type:** integer

Description: Usually each OVC has a single End Point at the ENNI but with hairpin switching an OVC can have multiple. This Service Attribute sets the limit on the number of End Points an OVC can have at the ENNI.

Allowed values: Minimum: 1

Property Name: ENNI Token Share Service Attribute

Reference: MEF 26.2 sec. 13.5

JSON Name: tokenShare - **Type:** object

Description: An attribute that indicates whether Bandwidth Profile Envelopes containing more than one Bandwidth Profile Flow are supported by the Service Provider at the ENNI.

Allowed values: Type: EnabledDisabled

Validation Notes: If this attribute is **DISABLED**, every Envelope at the ENNI must contain at most 1 BWP Flow.

Property Name: ENNI Envelopes Service Attribute

Reference: MEF 26.2 sec. 13.6

JSON Name: envelopes - **Type:** List of object

Description: The Envelopes and Envelope Coupling Flag values to which Bandwidth Profile Flows can be mapped. Value is list of {envelopeID, couplingFlagForIndexZero}. **Note:** Envelopes for the ENNI can also be specified at the OVC End Points associated with the ENNI.

Allowed values: List of [0...*] Unique Class: Envelope

759

760 14.3.4 Operator Multilateral Attributes

Operator Multilateral Attributes

Schema File Name: operatorMultilateralAttributes

Includes classes:

OperatorMultilateral (Operator Multilateral Attributes)

Class: OperatorMultilateral (**Operator Multilateral Attributes**)

An Operator Multilateral Attribute is a behavior of a single Operator CEN at an ENNI that is externally visible to a Service Provider or Super Operator using that CEN and ENNI and/or to peer Operators operating CENs on the other side of the ENNI. An Operator Multilateral Attribute has the same value for all Service Providers or Super Operators using that CEN and ENNI to whom the behavior is externally visible.

Property Name: ENNI L2CP Peering Multilateral Attribute

Reference: MEF 26.2 sec. 10.1 and MEF 45 sec. 8.2

JSON Name: l2cpPeering - **Type:** List of object

Description: A list of Layer 2 Control Protocols that will be Peered by a protocol entity at an ENNI: an empty list, or a list of entries identifying protocols to be Peered where each entry consists of {Destination Address, Protocol Identifier} or {Destination Address, Protocol Identifier, Link Identifier}

Allowed values: List of [0...*] Class: L2cpPeering

Validation Notes: Refer to R12 in MEF 45.1 for restrictions on which destination address can/can not be included in the list of peered protocols.

Property Name: ENNI Tagged L2CP Frame Processing Multilateral Attribute

Reference: MEF 26.2 sec. 10.2 and MEF 45 sec. 8.3

JSON Name: taggedL2cpFrameProcessing - **Type:** enum

Description: Reflects the capability of the ENNI to process S-VLAN-tagged L2CP Frames in an 802.1 compliant manner. The allowed values are "802_1_COMPLIANT" or "802_1_NON_COMPLIANT". A value of 802_1_COMPLIANT means the ENNI will apply the special forwarding rules to Tagged L2CP ENNI Frames that map to a VUNI or an OVC End Point supporting a service other than EPL with EPL Option 2 L2CP processing. A value of 802_1_NON_COMPLIANT means the ENNI will pass any Tagged L2CP ENNI Frames. There is an ENNI Tagged L2CP Frame Processing Multilateral Attribute for each ENNI.

Allowed values: Enum: "802_1_COMPLIANT", "802_1_NON_COMPLIANT"



Property Name: ENNI Maximum Frame Size Multilateral Attribute

Reference: MEF 26.2 sec. 10.3

JSON Name: maximumFrameSizeMultilateral - **Type:** integer

Description: The maximum length ENNI Frame in bytes that can be reliably processed.

Allowed values: Minimum: 1526

Validation Notes: This must be at least 1526 (1518+C-tag+S-tag).

761

762 15 Common Classes and Types

763 This section is structure like the previous section but focuses on common classes and types used by the
764 Service Attributes. Most of these are structured to support a variety of Carrier Ethernet Services and hence
765 have filenames that are prefixed with “Carrier Ethernet”. This means that several of them include options
766 and values that aren’t appropriate for some services such as, in this case Access E-Line. Whenever that
767 situation arises the details are included in the description and/or validation notes for the specific Access E-
768 Line Service Attribute and, if appropriate, in the common class.

769 15.1 Bandwidth Profiles

Carrier Ethernet Bandwidth Profile

Schema File Name: carrierEthernetBandwidthProfile

Includes classes:

BwpFlow (Bandwidth Profile Flow)

Envelope (Envelope)

BandwidthProfilePerClassOfServiceName (COS Bandwidth Profile)

Class: BwpFlow (Bandwidth Profile Flow)

The Bandwidth Profile Flow, BwpFlow, object class represents the Bandwidth Profile Flow which includes the bandwidth profile parameter CIR, CIRmax, CBS, EIR, EIRmax, EBS, Coupling Flag, Color Mode, Envelope and Rank, and Token Request Offset. The BwpFlow is used with Service Attributes for CarrierEthernetOperatorUni, CarrierEthernetSubscriberUni, CarrierEthernetVuni, and ENNI as well as OVC and EVC End Points. Reference MEF 10.4 Section 12 and MEF 26.2 Section 17.

Property Name: Committed Information Rate

Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

JSON Name: cir - **Type:** object

Description: cir represents **Committed Information Rate**. When added to unused committed bandwidth provided from higher-ranked Bandwidth Profile Flows (depending on the value of **couplingFlag** for the higher-ranked Bandwidth Profile Flows), limits the average rate in bits per second at which External Interface Frames for this Bandwidth Profile Flow can be declared Green.

Allowed values: Class: InformationRate

Property Name: The Maximum Committed Information Rate

Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

JSON Name: cirMax - **Type:** object

Description: cirMax represents **Maximum Committed Information Rate**. It limits the average rate in bits per second at which External Interface Frames for this Bandwidth Profile Flow can be declared Green (regardless of unused committed bandwidth from higher-ranked Bandwidth Profile Flows).

Allowed values: Class: InformationRate

Property Name: Committed Burst Size

Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

JSON Name: cbs - **Type:** object

Description: **cbs** represents **Committed Burst Size**. Limits by how much, and for how long, the amount of traffic declared Green for this Bandwidth Profile Flow in the short term can exceed the committed bandwidth made available to this Bandwidth Profile Flow over the long term, in bytes.

Allowed values: Class: DataSize

Validation Notes: If cir and/or cirmax are greater than 0, cbs must be greater than 0. MEF 23.2 recommends that cbs should be at least twice the Maximum Frame size (although in practice it is usually quite a bit more than that).

Property Name: *Excess Information Rate*

Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

JSON Name: *eir* - **Type:** object

Description: **eir** represents **Excess Information Rate**. When added to unused excess bandwidth from higher-ranked Bandwidth Profile Flows, and to un-used committed bandwidth (depending on the value of **couplingFlag** for this Bandwidth Profile Flow and **couplingFlagForIndexZero** for the Envelope), limits the average rate in bits per second at which External Interface Frames for this Bandwidth Profile Flow can be declared Yellow.

Allowed values: Class: InformationRate

Property Name: *Maximum Excess Information Rate*

Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

JSON Name: *eirMax* - **Type:** object

Description: **eirMax** represents **Maximum Excess Information Rate**. It limits the average rate in bits per second at which External Interface Frames for this Bandwidth Profile Flow can be declared Yellow (regardless of unused excess bandwidth from higher-ranked Bandwidth Profile Flows or unused committed bandwidth).

Allowed values: Class: InformationRate

Property Name: *Excess Burst Size*

Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

JSON Name: *ebs* - **Type:** object

Description: **ebs** represents **Excess Burst Size**. It limits by how much, and for how long, the amount of traffic declared Yellow for this Bandwidth Profile Flow in the short term can exceed the excess bandwidth made available to this Bandwidth Profile Flow over the long term.

Allowed values: Class: DataSize

Validation Notes: If eir and/or eirmax are greater than 0, ebs must be greater than 0. Similar to cbs, ebs should be at least twice the Maximum Frame size (although in practice it is usually quite a bit more than that).

Property Name: *Coupling Flag*

Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

JSON Name: *couplingFlag* - **Type:** boolean

Description: The **couplingFlag** determines whether unused committed bandwidth for this Bandwidth Profile Flow is made available as excess bandwidth for this Bandwidth Profile Flow or as committed bandwidth for the next lower-ranked Bandwidth Profile Flow. 0/FALSE means overflow green tokens are used as green tokens in the next lowest BWP Flow in the Envelope. 1/TRUE means they are used as yellow tokens for this BWP Flow.

Allowed values:

Validation Notes: This must be 0 (FALSE) if **Envelope.couplingFlagForIndexZero** is 1 (TRUE)

Property Name: *Color Mode***Reference:** MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2**JSON Name:** *colorMode* - **Type:** enum

Description: **colorMode** indicates whether or not External Interface Frames for this Bandwidth Profile Flow that are identified as *Yellow* on input to the Bandwidth Profile Algorithm can be declared *Green*.

Allowed values: Enum: "COLOR_BLIND", "COLOR_AWARE"

Property Name: *Envelope Name***Reference:** MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2**JSON Name:** *envelopeId* - **Type:** string

Description: The name of the envelope that this Bandwidth Profile Flow resides in.

Allowed values: String: Min Length: 1 Max Length: 45 Pattern (regex): [\x20-\x7F]+

Validation Notes: This **must** match an Envelope ID in exactly one of the Envelopes at the External Interface where this Bandwidth Profile Flow is applied (i.e. *OperatorUNI.envelopes*, *SubscriberUNI.envelopes*, *EnniService.envelopes*, or *AccessElineOvcEpCommon.ovcEndPointEnvelopes* or *CarrierEthernetEvcEpNotEpl.evcEndPointEnvelopes*).

All Bandwidth Profile Flows in the same envelope **must** satisfy the same criterion listed in R230 of MEF 26.2 for Operator Ethernet Services and R165 of MEF 10.4 for Subscriber Ethernet Services.

Property Name: *Envelope Rank***Reference:** MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2**JSON Name:** *envelopeRank* - **Type:** integer

Description: The rank in the envelope of this Bandwidth Profile Flow

Allowed values: Minimum: 1

Property Name: *Token Request Offset Field***Reference:** MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2**JSON Name:** *tokenRequestOffset* - **Type:** integer

Description: **tokenRequestOffset** adjusts the bandwidth consumed by each External Interface Frame in the Bandwidth Profile Flow by a fixed number of octets.

Allowed values: [Any Integer]

Class: Envelope (*Envelope*)

Bandwidth Profiles are organized into Envelopes. Multiple Bandwidth Profiles in an envelope can share bandwidth resources. Each Envelope has a name and a Coupling Flag for Index Zero (CF0) that indicates whether unused green tokens are discarded or converted to yellow tokens. All Bandwidth Profile Flows in the same envelope satisfy the same criterion listed in R230 of MEF 26.2.

Property Name: *Envelope Name***Reference:** MEF 10.4 sec. 9.12 and MEF 26.2 sec. 14.19**JSON Name:** *envelopeID* - **Type:** string



Description: The attribute is a string that identifies the Envelope

Allowed values: String: Min Length: 1 Max Length: 45 Pattern (regex): [\x20-\x7F]+

Property Name: Envelope CFI

Reference: MEF 10.4 sec. 9.12 and MEF 26.2 sec. 14.19

JSON Name: couplingFlagForIndexZero - **Type:** boolean

Description: This attribute denotes the coupling flag for index zero. FALSE for 0 (overflow Green tokens are discarded) and TRUE for 1 (overflow Green tokens can be used as Yellow tokens).

Allowed values:

Class: BandwidthProfilePerClassOfServiceName (**COS Bandwidth Profile**)

The value is a list of pairs of the form (x, y) For ingress, x is a Class of Service Name that is in the value of the EVC/OVC List of Class of Service Names Service Attribute (MEF 10.4 sec. 8.7 and MEF 26.2 sec. 12.12) for the EVC/OVC that associates the EVC/OVC End Point. For egress, x is a Class of Service from the EVC List of Class of Service Names Service Attribute or an Egress Equivalence Class Name (for OVC). If specified, y is a reference to a Bandwidth Profile Flow and Disabled otherwise. There is at most one pair in the list for each CoS or EEC Name (Except Discard).

For EVCs it is used by the EVC EP Class of Service Name Ingress Bandwidth Profile Service Attribute (MEF 10.4 sec. 10.9), and EVC EP Class of Service Name Egress Bandwidth Profile Service Attribute (MEF 10.4 sec. 10.11).

For OVCs it is used by the Ingress Bandwidth Profile per Class of Service Name (MEF 26.2 sec. 16.12) Egress Bandwidth Profile per Egress Equivalence Class Name (MEF 26.2 sec. 16.13).

Property Name: CoS or EEC Name

Reference: MEF 26.2 sec. 16.12 and 16.13

JSON Name: classOfServiceName - **Type:** string

Description: For Ingress, this is the Name of the Class of Service that this bandwidth profile flow is applied to. This should be an item from ovc.listOfClassOfServiceNames. For Egress, this is the Class of Service for EVCs and the Egress Equivalence Class (EEC) Name for OVCs. If the specific Class of Service/ Egress Equivalence Class is not used/configured that the End Point, the Bandwidth Profile can be an empty list.

Allowed values: [Any String]

Property Name: CoS Bandwidth Profile

JSON Name: bwpFlow - **Type:** object

Description: The Bandwidth Profile to be applied to all items that are in the specified Class of Service or Egress Equivalence Class.

Allowed values: Class: BwpFlow

770

771

772

15.2 Class of Service

Carrier Ethernet Class of Service

Schema File Name: carrierEthernetClassOfService
Includes classes:
CosFromEp (CoS Name From EP)
CosFromCtagPcp (CoS Name From C-Tag PCP)
CosUniPcpMapEntry (PCP To CoS Map Entry)
CosFromStagPcp (CoS Name From S-Tag PCP)
CosEnniPcpMapEntry (Enni PCP to CoS Name Map Entry)
CosFromDscp (CoS Name From DSCP)
CosDscpMap (DSCP to CoS Name Map)
CosDscpMapEntry (DSCP to CoS Name Map Entry)
DscpValues (List of DSCP Values)
CosL2cp (L2CP to CoS Name)

Class: CosFromEp (**CoS Name From EP**)

Ingress CoS is specified as {F,M,P}. This is when F=OVC_EP or EVC_EP as appropriate.

Property Name: CoS Map Type - EP
JSON Name: mapType - **Type:** const

Description: The field that indicates F = EVC/OVC EP.

Allowed values: "ENDPOINT"

Property Name: CoS Map
JSON Name: map_M - **Type:** string

Description: The single CoS Name at this EVC/OVC EP

Allowed values: [Any String]

Validation Notes: The value must be in the *listOfClassServiceNames* for the EVC/OVC.

Property Name: CoS L2CP
JSON Name: l2cp_P - **Type:** List of object

Description: This specifies the CoS Name for L2CPs carried over the EVC/OVC.

Allowed values: List of [0...*] Unique Class: CosL2cp

Class: CosFromCtagPcp (**CoS Name From C-Tag PCP**)

Ingress CoS is specified as {F,M,P}. This is when F=C_TAG_PCP. This maps each of the 8 PCP values in the C-tag to a Class of Service Name. It also maps UNTAGGED frames to a Class of Service Name.

Property Name: CoS Map Type - CTAG
JSON Name: mapType - **Type:** const

Description: The Field that indicates F = C_TAG_PCP.

Allowed values: "C_TAG_PCP"

Property Name: CoS Map

JSON Name: map_M - **Type:** List of object

Description: This maps each of the possible C-tag PCP values (0-7) and Untagged to a specific CoS Name.

Allowed values: List of [9...9] Unique Class: CosUniPcpMapEntry

Property Name: CoS L2CP

JSON Name: l2cp_P - **Type:** List of object

Description: This specifies the CoS Name for L2CPs carried over the EVC/OVC.

Allowed values: List of [0...*] Unique Class: CosL2cp

Class: CosUniPcpMapEntry (**PCP To CoS Map Entry**)

2-tuple {PCP, CoSName} that maps a PCP value (or Untagged) to a CoS Name.

Property Name: C-Tag PCP Value

JSON Name: pcpVal - **Type:** enum

Description: A PCP value from 0 to 7 or UNTAGGED

Allowed values: Enum: "0", "1", "2", "3", "4", "5", "6", "7", "UNTAGGED"

Property Name: CoS Name

JSON Name: pcpCosName - **Type:** string

Description: A CoS Name from the EVC/OVC List of Class of Service Names Service Attribute.

Allowed values: [Any String]

Validation Notes: The CoS Name must be in *listOfClassServiceNames* for the EVC/OVC.

Class: CosFromStagPcp (**CoS Name From S-Tag PCP**)

Ingress CoS is specified as {F,M,P}. This is when F=S_TAG_PCP. This maps each of the 8 PCP values in the S-tag to a Class of Service Name.

Property Name: CoS Map Type - STAG

JSON Name: mapType - **Type:** const

Description: The Field that indicates F = S_TAG_PCP.

Allowed values: "S_TAG_PCP"

Property Name: CoS Map

JSON Name: map_M - **Type:** List of object

Description: A list of 8 PCP values and their associated CoS Names

Allowed values: List of [8...8] Unique Class: CosEnniPcpMapEntry

Property Name: CoS L2CP

JSON Name: l2cp_P - **Type:** List of object

Description: This specifies the CoS Name for L2CPs carried over the OVC.

Allowed values: List of [0...*] Unique Class: CosL2cp

Validation Notes: This **must** be null at an OVC End Point that is not in a VUNI.

Class: CosEnniPcpMapEntry (**Enni PCP to CoS Name Map Entry**)

2-tuple {PCP, CoSName} that maps a PCP to a CoS Name or DISCARD.

Property Name: ENNI PCP Value

JSON Name: pcpVal - **Type:** object

Allowed values: Type: PcpValue

Property Name: ENNI CoS Name

JSON Name: pcpCosName - **Type:** string

Allowed values: [Any String]

Class: CosFromDscp (**CoS Name From DSCP**)

Ingress CoS is specified as {F,M,P}. This is when F=DSCP. This maps groups of DSCP values (independently for IPv4 and IPv6) to a Class of Service. It also provides a Class of Service mapping for DSCP values that are not specified and for frames that do not contain IP Packets.

Property Name: CoS Map Type - DSCP

JSON Name: mapType - **Type:** const

Description: The Field that indicates F = DSCP.

Allowed values: "DSCP"

Property Name: *CoS Map*
JSON Name: *map_M* - **Type:** *object*

Description: This maps DSCP values to CoS Names for each of IPv4 and IPv6 and also to non-IP packets.

Allowed values: Class: CosDscpMap

Property Name: *CoS L2CP*
JSON Name: *l2cp_P* - **Type:** *List of object*

Description: This specifies the CoS Names for L2CPs carried over the EVC/OVC.

Allowed values: List of [0...*] Unique Class: CosL2cp

Class: CosDscpMap (**DSCP to CoS Name Map**)

This is a list of DSCP to CoS Name Entries followed by items for Not an IP Packet and All Other Values.

Property Name: *DSCP Value to CoS List*
JSON Name: *dscpValueCoSList* - **Type:** *List of object*

Description: This is a list of DSCP to CoS Map Entries: {ipv4DSCPs, ipv6DSCPs, CoSName}

Allowed values: List of [0...*] Unique Class: CosDscpMapEntry

Validation Notes: The IPv4 DSCP values in the different entries in the list must not overlap. The IPv6 DSCP values in the different entries in the list must not overlap.

Property Name: *CoS for Other IPv4*
JSON Name: *otherIPv4* - **Type:** *string*

Description: This is the CoS Name for IPv4 packets with DSCP values other than those listed.

Allowed values: [Any String]

Property Name: *CoS for Other IPv6*
JSON Name: *otherIPv6* - **Type:** *string*

Description: This is the CoS Name for IPv6 packets with DSCP values other than those listed.

Allowed values: [Any String]

Property Name: *CoS for Not IP Packet*
JSON Name: *notIP* - **Type:** *string*

Description: This is the CoS Name for packets that are not IP packets when CoS is determined from DSCP

Allowed values: [Any String]

Class: CosDscpMapEntry (**DSCP to CoS Name Map Entry**)

This is a 3-tuple {*ipv4list*, *ipv6list*, *cosName*}

Property Name: IPv4 List of DSCPs
JSON Name: *ipv4List* - **Type:** object

Description: List of IPv4 DSCP values

Allowed values: Class: DscpValues

Property Name: IPv6 List of DSCPs
JSON Name: *ipv6List* - **Type:** object

Description: List of IPv6 DSCP values

Allowed values: Class: DscpValues

Property Name: Class of Service Name
JSON Name: *cosName* - **Type:** string

Allowed values: [Any String]

Class: DscpValues (**List of DSCP Values**)

A list of 6 bit values

Property Name: List of DSCP Values
JSON Name: *dscpValues* - **Type:** List of integer

Description: A list of unique 6-bit (0-63) values

Allowed values: List of [1...*] Unique Minimum: 0 Maximum: 63

Class: CosL2cp (**L2CP to CoS Name**)

2-tuple {*L2CP Ident*, *CoS Name*}

Property Name: L2CP Identifier
Reference: MEF 45.1 sec. 8.2
JSON Name: *l2cpIdentifier* - **Type:** object

Description: Protocol Identifier

Allowed values: Class: L2cpProtocol

Property Name: L2CP CoS Name
JSON Name: l2cpCosName - **Type:** string

Description: CoS Name String to associate with the L2CP

Allowed values: [Any String]

773 15.3 Color Identifier

Carrier Ethernet Color Identifier

Schema File Name: carrierEthernetColorIdentifier

Includes classes:

- ColorFromDei (Color From DEI)
- ColorFromPCP (Color From PCP)
- ColorFromPCPMapEntry (Color from PCP Map Entry)
- ColorFromDscp (Color From DSCP)
- ColorFromDscpEntry (Color From DSCP Entry)
- ColorFromEp (Color From EP)

Class: ColorFromDei (**Color From DEI**)

This Class is referenced when the color comes from the C-tag DEI field (for EVCs and OVCs) or the S-tag DEI field for OVCs. MEF 10.4 (EVCs) and MEF 26.2 (OVCs) requires that DEI=0 means Green and DEI=1 means Yellow, so no map is needed.

Property Name: Color Map Type - DEI
JSON Name: mapType - **Type:** const

Allowed values: "DEI"

Class: ColorFromPCP (**Color From PCP**)

This Class is referenced when the color comes from the S-tag or C-tag PCP field. Each of the 8 possible PCP values maps to a color. If there is no tag, the frame is Green.

Property Name: Color Map Type - PCP
JSON Name: mapType - **Type:** const

Allowed values: "PCP"

Property Name: Color Map
JSON Name: colorFromPcpMap - **Type:** List of object

Description: This is a list of 8 Color mappings, one for each PCP value.

Allowed values: List of [8...8] Unique Class: ColorFromPCPMapEntry

Class: ColorFromPCPMapEntry (**Color from PCP Map Entry**)

Property Name: PCP Value
JSON Name: pcpValue - **Type:** object

Allowed values: Type: PcpValue

Property Name: PCP Color
JSON Name: pcpColor - **Type:** object

Allowed values: Type: FrameColor

Class: ColorFromDscp (**Color From DSCP**)

This Class is referenced when the color comes from the DSCP field. It is a list of DSCP Entries where each entry has three components: (1) A list of DSCP values, (2) a color for IPv4 packets with one of those DSCP values, and (3) a color for IPv6 packets with one of those DSCP values. Any DSCP value that is not specified and any Frame that does not include an IP packet is Green..

Property Name: Color Map Type - DSCP
JSON Name: mapType - **Type:** const

Allowed values: "DSCP"

Property Name: colorMap
JSON Name: colorFromDscpMap - **Type:** List of object

Allowed values: List of [1...*] Unique Class: ColorFromDscpEntry

Class: ColorFromDscpEntry (**Color From DSCP Entry**)

Property Name: DSCP List
JSON Name: dscpList - **Type:** List of integer

Description: This is a list of DSCP values

Allowed values: List of [1...*] Unique Minimum: 0 Maximum: 63

Property Name: IPv4 Color
JSON Name: ipv4Color - **Type:** object

Allowed values: Type: FrameColor

Property Name: IPv6 Color
JSON Name: ipv6Color - **Type:** object

Allowed values: Type: FrameColor

Class: ColorFromEp (**Color From EP**)

The color of each Ingress Frame is determined solely from the EVC or OVC End Point. In other words, the EP declares all frames GREEN or all frame YELLOW

Property Name: Color Map Type - ENDPOINT**JSON Name:** mapType - **Type:** const**Allowed values:** "ENDPOINT"**Property Name:** Color**JSON Name:** epColor - **Type:** object**Allowed values:** Type: FrameColor774 **15.4 Egress Maps****Carrier Ethernet Egress Maps**

Schema File Name: carrierEthernetEgressMaps

Includes classes:

PcpFromCos (PCP From Class of Service)
CnPcp (CNPCP Entry)
DeiFromCosColor (DEI From CoS and Color)
CcDei (CCDEI Entry)
PcpFromCosColor (PCP From CoS and Color)
CcPcp (CCPCP Entry)
PcpFromCosDeiFromCosColor (PCP From CoS and DEI From CoS and Color)
CnPcpCcDei (CNPCPCCDEI Entry)
EvcEgressMap (EVC Egress Map)
EvcEgressMapEntry (Evc Egress Map Entry)

Class: PcpFromCos (**PCP From Class of Service**)

The PCP Field in the Egress EI Frame is determined from the Class of Service of the corresponding Ingress EI Frame. This is a list that contains an entry for Class of Service Name that includes the CoS Name and the PCP value (or DISCARD) to use for that CoS. For example CoS=Platinum --> PCP = 6. This affects the C-tag at the UNI and the S-tag at the ENNI.

Property Name: Map Type - CN → PCP**JSON Name:** mapType - **Type:** const**Allowed values:** "CN_PCP"**Property Name:** CNPCP Entries**JSON Name:** cnPcpEntries - **Type:** List of object**Description:** A list of mappings of each CoS to an egress PCP value or DISCARD.**Allowed values:** List of [1...*] Unique Class: CnPcp

Class: CnPcp (**CNPcp Entry**)

This is a 2-tuple (x, y) where x is a Class of Service Name and y is a PCP Value or DISCARD.

Property Name: Class of Service Name
JSON Name: cosName - **Type:** string

Description: The class of service name for the corresponding Ingress EI Frame.

Allowed values: [Any String]

Property Name: PCP Value
JSON Name: pcpValue - **Type:** object

Allowed values: Type: PcpValueAndDiscard

Class: DeiFromCosColor (**DEI From CoS and Color**)

The DEI Field in the Egress EI Frame is determined from the Class of Service and the Frame Color. This is a list that contains an entry for Class of Service Name that includes the CoS Name and the DEI value (or DISCARD) to use for Green frames and the DEI value (or DISCARD) to use for Yellow Frames. For example CoS=Platinum and Color=Green --> DEI = 1 and Color=Yellow --> DEI = 0. This affects the C-tag at the UNI and the S-tag at the ENNI.

Property Name: Map Type CC → DEI
JSON Name: mapType - **Type:** const

Allowed values: "CC_DEI"

Property Name: List of CCDEI Entries
JSON Name: ccDeiEntries - **Type:** List of object

Description: A list of entries, one per CoS, that specifies the DEI value for Green frames and Yellow frames.

Allowed values: List of [1...*] Unique Class: CcDei

Class: CcDei (**CCDEI Entry**)

One of the possible values for the OVC Egress Map. This indicates that the Color is mapped to the DEI field of the Egress Frame. Note that the Enums used for the DEI values include DISCARD as a value.

Property Name: Class of Service
JSON Name: cosName - **Type:** string

Description: The class of service name for the corresponding Ingress EI Frame.

Allowed values: [Any String]

Property Name: DEI for Green Frames
JSON Name: deiGreen - **Type:** object

Description: The DEI value (or DISCARD) to use for Green frames.

Allowed values: Type: DeiValueAndDiscard

Property Name: DEI for Yellow Frames
JSON Name: deiYellow - **Type:** object

Description: The DEI value (or DISCARD) to use for Yellow frames.

Allowed values: Type: DeiValueAndDiscard

Class: PcpFromCosColor (**PCP From CoS and Color**)

The PCP Field in the Egress EI Frame is determined from the Class of Service Name and the Frame Color. This is a list that contains an entry for Class of Service Name that includes the CoS Name and two PCP values (or DISCARD), one for Green frames and one for Yellow frames. For example CoS=Platinum and Color=Green --> PCP = 6 and Color=Yellow --> PCP = 5. This affects the C-tag at the UNI and the S-tag at the ENNI.

Property Name: Map Type - CC → PCP
JSON Name: mapType - **Type:** const

Allowed values: "CC_PCP"

Property Name: List of CCPCP Entries
JSON Name: ccPcpEntries - **Type:** List of object

Description: A list of entries, one per CoS that indicates the PCP value for Green Frames and the PCP value for Yellow Frames.

Allowed values: List of [1...*] Unique Class: CcPcp

Class: CcPcp (**CCPCP Entry**)

One of the possible values for the OVC Egress Map. This indicates that the Class of Service and Color are mapped to the PCP field of the Egress Frame. Note that the Enums used for the PCP values include DISCARD as a value.

Property Name: Class of Service
JSON Name: cosName - **Type:** string

Description: The class of service name for the corresponding Ingress EI Frame.

Allowed values: [Any String]

Property Name: PCP For Green Frames
JSON Name: pcpGreen - **Type:** object

Description: A PCP value (0-7) or DISCARD to use for Green frames in the CoS

Allowed values: Type: PcpValueAndDiscard

Property Name: PCP For Yellow Frames

JSON Name: pcpYellow - **Type:** object

Description: A PCP value (0-7) or DISCARD to use for Yellow frames in the CoS

Allowed values: Type: PcpValueAndDiscard

Class: PcpFromCosDeiFromCosColor (**PCP From CoS and DEI From CoS and Color**)

This is a combination of PCP from CoS and DEI from Color. The PCP Field in the Egress EI Frame is determined from the Class of Service and the DEI Field is determined from the Frame Color. This is a list that contains an entry for Class of Service Name that includes the CoS Name, PCP value to use (or DISCARD) and the DEI value to use for each color for that CoS Name. For example CoS=Platinum and Color=Green --> PCP = 6/DEI = 1 and Color=Yellow --> DEI=0. This affects the C-tag at the UNI and the S-tag at the ENNI.

Property Name: Map Type - CN→PCP & CC→DEI

JSON Name: mapType - **Type:** const

Allowed values: "CN_PCP_CC_DEI"

Property Name: List of CNPCP_CCDEI Entries

JSON Name: cnPcpCcDeiEntries - **Type:** List of object

Description: A list of entries, one per CoS, that specifies the PCP value for the CoS and the DEI values for Green and Yellow frames in the CoS.

Allowed values: List of [1...*] Unique Class: PcpFromCosDeiFromCosColor

Class: CnPcpCcDei (**CNCPCCDEI Entry**)

Property Name: Class of Service

JSON Name: cosName - **Type:** string

Description: The class of service name for the corresponding Ingress EI Frame.

Allowed values: [Any String]

Property Name: PCP Value

JSON Name: pcpValue - **Type:** object

Description: A PCP value (0-7) or DISCARD to use frames in the CoS

Allowed values: Type: PcpValueAndDiscard

Property Name: DEI for Green Frames
JSON Name: deiGreen - **Type:** object

Description: A DEI value to use for Green frames in the CoS

Allowed values: Type: DeiValueAndDiscard

Property Name: DEI for Yellow Frames
JSON Name: deiYellow - **Type:** object

Description: A DEI value to use for Yellow frames in the CoS

Allowed values: Type: DeiValueAndDiscard

Class: EvcEgressMap (**EVC Egress Map**)

This map is a table with one entries for each CosName, that includes the CoSName, a PCP value for green frames, a PCP value for yellow frames, a DEI value for green frames and a DEI value for yellow frames. Any of the values can be DISCARD. This table is used at the UNI for EVCs.

Property Name: List of EVC Egress Map Entries
JSON Name: evcEgressMapEntries - **Type:** List of object

Description: This is a table of EVC Egress Map entries. An empty list indicates a value of *None*. Per R140 in MEF 10.4, the value of the EVC EP Egress Map Service Attribute MUST NOT be *None* unless one of the conditions noted in the requirement is met. (See Validation Notes)

Allowed values: List of [0...*] Unique Class: EvcEgressMapEntry

Class: EvcEgressMapEntry (**Evc Egress Map Entry**)

A row in the EVC Egress Map. Contains a CoSName, PCPGreen, PCPYellow, DEIGreen, DEIYellow

Property Name: CoS Name
JSON Name: cosName - **Type:** string

Allowed values: [Any String]

Validation Notes: This must be a CoS Name from **EV C List of Class of Service Names**

Property Name: PCP Green
JSON Name: pcpGreen - **Type:** object

Description: A PCP value (0-7) or DISCARD to use for Green Frames

Allowed values: Type: PcpValueAndDiscard

Property Name: *PCP Yellow*
JSON Name: *pcpYellow* - **Type:** *object*

Description: A PCP value (0-7) or DISCARD to use for Yellow frames

Allowed values: Type: PcpValueAndDiscard

Property Name: *DEI Green*
JSON Name: *deiGreen* - **Type:** *object*

Description: A DEI value to use for Green frames

Allowed values: Type: DeiValueAndDiscard

Validation Notes: This is optional ONLY if *pcpGreen* is set to *DISCARD*

Property Name: *DEI Yellow*
JSON Name: *deiYellow* - **Type:** *object*

Description: A DEI value to use for Yellow frames

Allowed values: Type: DeiValueAndDiscard

Validation Notes: This is optional ONLY if *pcpYellow* is set to *DISCARD*

775

776 15.5 End Point Maps

Carrier Ethernet End Point Maps

Schema File Name: carrierEthernetEndPointMaps

Includes classes:

EndPointMapFormE (OVC End Point Map - E)
EndPointMapFormT (OVC End Point Map - T)
EndPointMapFormV (OVC End Point Map - V)
EndPointMapFormU (OVC End Point Map - U)
EvcEndPointMap (EVC End Point Map)

Class: EndPointMapFormE (**OVC End Point Map - E**)

The value of an End Point Map of Form E (E for ENNI) consists of a list of one or more SVLAN ID values. An S-Tagged ENNI Frame whose S-VLAN ID value matches an entry in the list maps to the OVC End Point.

Property Name: *Map Form - E*
JSON Name: *mapType* - **Type:** *const*

Description: The discriminator for Map Type E

Allowed values: "FORM_E"

Property Name: End Point Map Form E
JSON Name: ovcEndPointMapFormE - **Type:** List of integer

Description: A list of S-VLAN ID values at the ENNI.

Allowed values: List of [1...4094] Unique Minimum: 1 Maximum: 4094

Class: EndPointMapFormT (**OVC End Point Map - T**)

This map type is used when the OVC End Point is a Trunk OVC End Point (O-Tree services only). The value of an OVC End Point Map of Form T (T for Trunk) consists of a pair of S-VLAN ID values, (r,l) is called the Root S-VLAN ID value and l is called the Leaf S-VLAN ID value. An S-Tagged ENNI Frame whose S-VLAN ID value matches one of the two S-VLAN ID values in the OVC End Point Map Form T maps to the OVC End Point. Note that Form T differs from Form E because it distinguishes the role of each S-VLAN ID value.

Property Name: Map Form - T
JSON Name: mapType - **Type:** const

Description: The discriminator for Map Type T

Allowed values: "FORM_T"

Property Name: Root S-VLAN ID
JSON Name: rootSvlan - **Type:** integer

Description: The S-VLAN ID used for Root traffic

Allowed values: Minimum: 1 Maximum: 4094

Property Name: Leaf S-VLAN ID
JSON Name: leafSvlan - **Type:** integer

Description: The S-VLAN ID used for Leaf traffic

Allowed values: Minimum: 1 Maximum: 4094

Class: EndPointMapFormV (**OVC End Point Map - V**)

The value of an OVC End Point Map of Form V (V for VUNI) consists of one S-VLAN ID value and a list of one or more ENNI CE-VLAN ID values. An S-Tagged ENNI Frame whose S-VLAN ID value matches the S-VLAN ID value in the map and whose ENNI CE-VLAN ID value matches one of the ENNI CE-VLAN ID values in the map, maps to the OVC End Point.

Property Name: Map Form - V
JSON Name: mapType - **Type:** const

Description: The discriminator for Map Type V

Allowed values: "FORM_V"

Property Name: S-VLAN ID
JSON Name: svlanId - **Type:** integer

Description: The S-VLAN ID that is carrying the C-VLANs in the List

Allowed values: Minimum: 1 Maximum: 4094

Property Name: C-VLAN ID List
JSON Name: cvlanList - **Type:** List of integer

Description: The List of C-VLAN IDs delivered to the VUNI.

Allowed values: List of [1...4094] Unique Minimum: 1 Maximum: 4094

Class: EndPointMapFormU (**OVC End Point Map - U**)

The EndPointMapFormU object class represents the OVC End Point Map Form U (U for UNI). It is a list of one or more CE-VLAN IDs.

Property Name: Map Form - U
JSON Name: mapType - **Type:** const

Description: The discriminator for Map Type U

Allowed values: "FORM_U"

Property Name: OVC End Point Map Form U
JSON Name: ovcEndPointMapFormU - **Type:** List of integer

Description: A list of C-VLAN ID values at the UNI.

Allowed values: List of [1...4094] Unique Minimum: 1 Maximum: 4094

Class: EvcEndPointMap (**EVC End Point Map**)

This is specifications for which Service Frames are mapped to the EVC End Point. The allowed values in MEF 10.4 are **ALL**, **LIST**, and **UT/PT**. Since Private services must be **ALL**, the map isn't needed for those services, and therefore the type does not include **ALL**. If the type is **LIST** there must be a list of VLAN IDs specified.

Property Name: VLAN Listing Type
JSON Name: vlanType - **Type:** enum

Description: An indication of the type of EVC End Point Map. If the value is UT/PT then no additional information is required. If the value is LIST then a list of C-VLAN ID values must be specified.

Allowed values: Enum: "LIST", "UT/PT"

Property Name: VLAN ID List
JSON Name: vlanId - **Type:** List of integer

Description: A list of VLAN IDs if the value of vlanType is "LIST"

Allowed values: List of [1...4094] Unique Minimum: 1 Maximum: 4094

Validation Notes: This element must be specified if and only if *EndPointMapEvc.vlanType* = "LIST"

777

778 15.6 External Interfaces

Carrier Ethernet External Interfaces

Schema File Name: carrierEthernetExternalInterfaces

Includes classes:

UniPhysicalLink (UNI Physical Link)
EnniPhysicalLink (ENNI Physical Link)
CarrierEthernetPhysicalLink (Carrier Ethernet Physical Link)
UniSpecificAttributes (UNI Specific)
VirtualFrameMap (Virtual Frame Map)

Type: UniPhysicalLink (**UNI Physical Link**)

This class describes the Physical Link attributes at a UNI. It is a 6-tuple (id, phy, frequency sync enable, PTP enable, connector, gender).

Property Name: UNI Physical Link **Type:** object

Allowed values: Compose (allOf): Class: CarrierEthernetPhysicalLink, Class: UniSpecificAttributes

Type: EnniPhysicalLink (**ENNI Physical Link**)

This class describes the Physical Link attributes at an ENNI. It consists of the Carrier Ethernet Physical Links (id and physical link type) with no elements.

Property Name: Carrier Ethernet Physical Link **Type:** object

Allowed values: Compose (allOf): Class: CarrierEthernetPhysicalLink

Class: CarrierEthernetPhysicalLink (**Carrier Ethernet Physical Link**)

A 2-tuple of the form (id, physicalLink). The value of id is an identifier for the physical link. The value of physicalLink is the specific Ethernet physical layer type.

Property Name: Physical Link Identifier

JSON Name: id - **Type:** string

Description: An identifier for the physical link.

Allowed values: String: Min Length: 1 Max Length: 45 Pattern (regex): [\x20-\x7F]+

Property Name: Physical Link PHY Type
JSON Name: physicalLink - **Type:** enum

Description: An Ethernet physical layer

Allowed values: Enum: "1BASE5", "2BASE_TL", "10BASE2", "10BASE5", "10BASE_F", "10BASE_FB", "10BASE_FL", "10BASE_FP", "10BASE_T", "10BASE_T1L", "10BASE_T1S", "10BASE_TE", "10BROAD36", "10PASS_TS", "100BASE_BX10", "100BASE_FX", "100BASE_LX10", "100BASE_T", "100BASE_T1", "100BASE_T2", "100BASE_T4", "100BASE_TX", "100BASE_X", "1000BASE_BX10", "1000BASE_CX", "1000BASE_KX", "1000BASE_LX", "1000BASE_LX10", "1000BASE_PX10", "1000BASE_PX20", "1000BASE_RHA", "1000BASE_RHB", "1000BASE_RHC", "1000BASE_SX", "1000BASE_T", "1000BASE_T1", "1000BASE_X", "2_5GBASE_KX", "2_5GBASE_T", "2_5GBASE_T1", "5GBASE_KR", "5GBASE_T", "5GBASE_T1", "10GBASE_CX4", "10GBASE_E", "10GBASE_ER", "10GBASE_FW", "10GBASE_KR", "10GBASE_KX4", "10GBASE_L", "10GBASE_LR", "10GBASE_LRM", "10GBASE_LW", "10GBASE_LX4", "10GBASE_R", "10GBASE_S", "10GBASE_SR", "10GBASE_SW", "10GBASE_T", "10GBASE_T1", "10GBASE_W", "10GBASE_X", "25GBASE_CR", "25GBASE_CR_S", "25GBASE_ER", "25GBASE_KR", "25GBASE_KR_S", "25GBASE_LR", "25GBASE_SR", "25GBASE_T", "40GBASE_CR4", "40GBASE_ER4", "40GBASE_FR", "40GBASE_KR4", "40GBASE_LR4", "40GBASE_R", "40GBASE_SR4", "40GBASE_T", "50GBASE_CR", "50GBASE_ER", "50GBASE_FR", "50GBASE_KR", "50GBASE_LR", "50GBASE_SR", "100GBASE_CR10", "100GBASE_CR2", "100GBASE_CR4", "100GBASE_DR", "100GBASE_ER4", "100GBASE_KP4", "100GBASE_KR2", "100GBASE_KR4", "100GBASE_LR4", "100GBASE_R", "100GBASE_SR10", "100GBASE_SR2", "100GBASE_SR4", "200GBASE_CR4", "200GBASE_DR4", "200GBASE_ER4", "200GBASE_FR4", "200GBASE_KR4", "200GBASE_LR4", "200GBASE_SR4", "400GBASE_DR4", "400GBASE_ER8", "400GBASE_FR8", "400GBASE_LR8", "400GBASE_SR16", "400GBASE_SR4_2", "400GBASE_SR8", "OTHER"

Property Name: Auto Negotiation
Reference: IEEE Std 802.3-2018 Clause 28 and 37
JSON Name: autoNegotiation - **Type:** object

Description: Indicates whether 802.3 autonegotiation is enabled on the UNI link.

Allowed values: Type: EnabledDisabled

Class: UniSpecificAttributes (**UNI Specific**)

Property Name: Physical Link Synchronous Ethernet
JSON Name: synchronousEthernet - **Type:** object

Description: Indicates if synchronous Ethernet is used on the physical link and has the value either **Enabled** or **Disabled**.

Allowed values: Type: EnabledDisabled

Property Name: Physical Link Precision Timing
JSON Name: precisionTiming - **Type:** object

Description: Indicates if the Precision Time Protocol is used on the physical link and has the value either **Enabled** or **Disabled**

Allowed values: Type: EnabledDisabled

Property Name: UNI Connector Type
JSON Name: uniConnectorType - **Type:** enum

Description: This indicates the type of connector that is presented to the Subscriber.

Allowed values: Enum: "SC", "LC", "RJ45", "OTHER"

Property Name: UNI Connector Gender

JSON Name: uniConnectorGender - **Type:** enum

Description: This indicates whether the Subscriber is presented with a SOCKET (common) or a PLUG (less common).

Allowed values: Enum: "SOCKET", "PLUG"

Class: VirtualFrameMap (*Virtual Frame Map*)

A map for virtual frame mapping. The absence of this attribute corresponds to a Service Attribute value of Not Applicable. Reference MEF 10.4 Section 9.3 Subscriber UNI Virtual Frame Map Service Attribute.

Property Name: Virtual Frame Map

Reference: MEF 10.4 sec. 9.2 and sec. 9.3

JSON Name: vfMap - **Type:** string

Description: When the value of the Subscriber UNI Instantiation Service Attribute = Virtual, there **must** exist a map that maps the set of Virtual Frames that cross the UNI to a sequence of pairs of the form(s,t) where s is a standard Ethernet frame per Clause 3 of IEEE Std 802.3 – 2015 [5] and t is the arrival time at the UNI for all bits in s. Since MEF 10.4 does not specify the contents or format of this map, it is currently defined as an arbitrary length string.

Allowed values: String: Min Length: 0

779

780 15.7 Layer 2 Control Protocols (L2CP)

Carrier Ethernet L2CP

Schema File Name: carrierEthernetL2cp

Includes classes:

L2cpPeering (L2CP Peering Identifier)

L2cpProtocol (L2CP Protocol Identifier)

Class: L2cpPeering (*L2CP Peering Identifier*)

This specifies the L2CP Protocol Identifier and the Destination Address in use by the protocol entity. Reference MEF 45.1 Section 8.2 L2CP Peering Service Attribute

Property Name: Protocol ID

JSON Name: protocolID - **Type:** object

Description: This specifies the Protocol Type for the L2CP. It consists of a field specification (ETHERTYPE or LLC), the EtherType or LLC value, and an optional subtype.

Allowed values: Class: L2cpProtocol

Property Name: Destination MAC Address**JSON Name:** destinationAddress - **Type:** string

Description: The Destination MAC Address of the L2CP to be peered in the standard format of 6 hex bytes separated by hyphen (-).

Allowed values: String: Pattern (regex): [0-9a-fA-F][0-9a-fA-F]([-][0-9a-fA-F][0-9a-fA-F])⁵

Property Name: Link ID List**JSON Name:** linkIdList - **Type:** List of string

Description: Identifiers for the links on which the specified protocol will be peered. If no links are specified the protocol is peered on all links.

Allowed values: List of [0...*] String: Min Length: 1 Max Length: 45 Pattern (regex): [\x20-\x7F]+

Class: L2cpProtocol (*L2CP Protocol Identifier*)

Defines a L2CP protocol LLC address type or EtherType with possible subtype. Reference MEF 45.1 Section 8.2 L2CP Peering Service Attribute.

Property Name: Protocol Type**JSON Name:** l2cpProtocolType - **Type:** enum

Description: The protocol field to inspect to identify the L2CP.

Allowed values: Enum: "ETHERTYPE", "LLC"

Property Name: EtherType or LLC Value**JSON Name:** llcAddressOrEtherType - **Type:** integer

Description: The EtherType value or LLC value

Allowed values: Minimum: 0

Property Name: EtherType SubType**JSON Name:** subType - **Type:** integer

Description: The EtherType can be modified by an optional sub-type.

Allowed values: Minimum: 0

781

782 **15.8 Link Aggregation****Carrier Ethernet Link Aggregation**

Schema File Name: carrierEthernetLinkAggregation

Includes classes:



ConversationIdToAggregationLinkMap (Conversation To Aggregation Link Map)
ConversationIdRange (ConversationID Range)

Class: ConversationIdToAggregationLinkMap (**Conversation To Aggregation Link Map**)

This is a 2-tuple (x, y) where x is a list of Port Conversation IDs or ranges of Port Conversation IDs (a Port Conversation ID is a VLAN ID or 0 for untagged frames) and y is a list of Link Numbers. This is used in the Port Conversation to Aggregation Link Map for the UNI and ENNI.

Property Name: List of Conversation ID Ranges

Reference: 802.1AX-2014 sec. 6.6.2.1

JSON Name: conversationIDs - **Type:** List of object

Description: A Port Conversation ID is a VLAN ID (1 to 4094) or 0 to represent untagged and priority-tagged frames.

Allowed values: List of [1...*] Unique Class: ConversationIdRange

Property Name: Aggregation Link List

Reference: 802.1AX-2014 sec. 6.6.2.1

JSON Name: aggLinkList - **Type:** List of integer

Description: An ordered list of Aggregation Link Numbers

Allowed values: List of [1...*] Unique Minimum: 1

Validation Notes: The values in the list must be in the range 1...n where n is the number of physical links -- i.e. the cardinality of the list of physical links (**SubscriberUNI.listofPhysicalLinks**, **OperatorUNI.listofPhysicalLinks** or **EnniCommon.listOfPhysicalLinks**)

Class: ConversationIdRange (**ConversationID Range**)

A range of ConversationID (either a VLAN Id or 0 for untagged frames) allowing three forms: start, start → start, and start → end

Property Name: Start Conversation ID

JSON Name: start - **Type:** integer

Description: The starting Conversation ID of the range or the only Conversation ID if there is no end value

Allowed values: Minimum: 0 Maximum: 4094

Validation Notes: The **start** value **must** be less than or equal to the **end** value if that is specified.

Coding hint: start must be less than or equal to end

Property Name: End Conversation ID

JSON Name: end - **Type:** integer

Description: The final Conversation ID in the range

Allowed values: Minimum: 0 Maximum: 4094

Validation Notes: The **end** value, if specified, **must** be greater than or equal to the **start** value.

Coding hint: end, if specified, must be greater than or equal to start



783

784

785

15.9 Service Level Specification

Carrier Ethernet Service Level Specification

Schema File Name: carrierEthernetServiceLevelSpecification

Includes classes:

CarrierEthernetSls (Carrier Ethernet SLS)
SlsCosNameEntry (SLS CoS Name Entry)
OneWayFrameDelayPmMetric (One Way Frame Delay)
OneWayMeanFrameDelayPmMetric (One Way Mean Frame Delay)
OneWayFrameDelayRangePmMetric (One Way Frame Delay Range)
OneWayInterFrameDelayVariationPmMetric (One Way InterFrame Delay Variation)
OneWayFrameLostRatioPmMetric (One Way Frame Loss Ratio)
OneWayAvailabilityPmMetric (One Way Availability)
OneWayHighLossIntervalPmMetric (One Way High Loss Intervals)
OneWayConsecutiveHighLossIntervalsPmMetric (One Way Consecutive High Loss Intervals)
OneWayCompositePmMetric (One Way Composite)
OneWayGroupAvailabilityPmMetric (One Way Group Availability)
OrderedPair (Ordered Pair)
OrderedPairList (Sets of Ordered Pairs)

Class: CarrierEthernetSls (**Carrier Ethernet SLS**)

The Service Level Specification Service Attribute (SLS) is, as part of the Service Level Agreement, the technical details of the service level, in terms of Performance Objectives, agreed between the Service Provider and the Subscriber for Subscriber Ethernet Services and between the SP/SO and Operator for Operator Ethernet Services. . A given SLS might contain 0, 1, or more Performance Objectives for each Performance Metric. The SLS is described in section 8.8 (EVC Service Level Specification Service Attribute) of MEF 10.4 for EVCs and section 12.13 (OVC Service Level Specification Service Attribute) for OVCs. MEF 10.4 defines 10 Performance Metrics (all of which are included here). MEF 26.2 defines only 9 Performance Metrics. MEF 26.2 does not include the One Way Composite PM. The other 9 are the same.

Property Name: Start Time

Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13

JSON Name: startTime - Type: string

Description: This represents the date and time for the start of the SLS. It is the beginning of the first *timeInterval*. This is referred to as *ts* in MEF 10.4 and MEF 26.2. The value is a string in ISO 8601 format.

Allowed values: String: Format: date-time

Property Name: Time Duration

Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13

JSON Name: timeDuration - Type: object

Description: The **timeDuration** (referred to as *T* in MEF 10.4 and 26.2) is a time duration, e.g., 1 month, 2 weeks, that is used in conjunction with *startTime* to specify time intervals for determining when Performance Objectives are met. Note that the units for *timeDuration* are not constrained; in particular, 1 month is an allowable value, corresponding to a calendar month, e.g. from midnight on the 10th of one month up to but not including midnight the 10th of the following month.

Allowed values: Class: TimeDuration

Property Name: SLS CoS Entries

Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13

JSON Name: slsCosNameEntry - Type: List of object

Description: This is a list of Performance Metrics. Each entry contains a set of parameters and Performance Metrics for one Class of Service.

Allowed values: List of [1...*] Unique Class: SlsCosNameEntry

Class: SlsCosNameEntry (**SLS CoS Name Entry**)

This is the specification of Performance Metrics (and parameters) for one Class of Service. The first element is the CoS Name. This MUST be unique, i.e. there can not be two instances of this Class with the same CoS Name. The CoS Name is followed by three parameters and then lists of Performance Metrics.

Property Name: CoS Name

JSON Name: cosName - **Type:** string

Description: This is the CoS Name for this PM Entry.

Allowed values: String: Min Length: 1

Validation Notes: The CoS Name should be included in the EVC List of Class of Service Names (*CarrierEthernetEvcCommon.listOfCosNames*) or OVC List of Class of Service Names (*AccessElineOvc.listOfClassOfServiceNames*)

Property Name: Delta T

Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13

JSON Name: deltaT - **Type:** integer

Description: **deltaT** is a time duration in seconds much smaller than **timeInterval**. This is used, primarily, in calculations associated with Availability, High Loss Intervals and Consecutive High Loss Intervals. MEF 10.4 and MEF 26.2 refer to this as Δt and the k-th interval starting at t_s is referred to as Δt_k (k is subscript).

Allowed values: Minimum: 1

Validation Notes: **deltaT** is a time duration in seconds much smaller than **timeInterval** (e.g. 10 seconds).

Property Name: Frame Loss Threshold

Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13

JSON Name: thresholdC - **Type:** number

Description: **thresholdC** is a real number in the range [0,1] used as a threshold to determine whether a given time interval Δt_k has high loss.

Allowed values: Minimum: 0 Maximum: 1

Property Name: HLI Consecutive Intervals

Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13

JSON Name: consecutiveIntervalN - **Type:** integer

Description: **consecutiveIntervalN** is an integer ≥ 1 , used to identify how many consecutive Δt_k intervals must have high loss to trigger a change in Availability.

Allowed values: Minimum: 1

Property Name: One Way Frame Delay

Reference: MEF 10.4 sec. 8.8.2 and MEF 26.2 sec. 12.13.2

JSON Name: oneWayFrameDelayPmMetric - **Type:** List of object

Description: **oneWayFrameDelayPmMetric** is a reference to zero or more instances of One Way Frame Delay Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way Frame Delay. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...*] Unique Class: OneWayFrameDelayPmMetric

Property Name: One Way Mean Frame Delay

Reference: MEF 10.4 sec. 8.8.3 and MEF 26.2 sec. 12.13.3

JSON Name: *oneWayMeanFrameDelayPmMetric* - **Type:** List of object

Description: **oneWayMeanFrameDelayPmMetric** is a reference to zero or more instances of One Way Mean Frame Delay Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way Mean Frame Delay. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...*] Unique Class: OneWayMeanFrameDelayPmMetric

Property Name: One Way Frame Delay Range

Reference: MEF 10.4 sec. 8.8.4 and MEF 26.2 sec. 12.13.4

JSON Name: *oneWayFrameDelayRangePmMetric* - **Type:** List of object

Description: **oneWayFrameDelayRangePmMetric** is a reference to zero or more instances of One Way Frame Delay Range Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way Frame Delay Range. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...*] Unique Class: OneWayFrameDelayRangePmMetric

Property Name: One Way Inter Frame Delay Variation

Reference: MEF 10.4 sec. 8.8.5 and MEF 26.2 sec. 12.13.5

JSON Name: *oneWayInterFrameDelayVariationPmMetric* - **Type:** List of object

Description: **oneWayInterFrameDelayVariationPmMetric** is a reference to zero or more instances of One Way Inter Frame Delay Variation Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way Inter Frame Delay Variation. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...*] Unique Class: OneWayInterFrameDelayVariationPmMetric

Property Name: One Way Frame Loss Ratio

Reference: MEF 10.4 sec. 8.8.6 and MEF 26.2 sec. 12.13.6

JSON Name: *oneWayFrameLossRatioPmMetric* - **Type:** List of object

Description: **oneWayFrameLossRatioPmMetric** is a reference to zero or more instances of One Way Frame Loss Ratio Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way Frame Loss Ratio. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...1] Class: OneWayFrameLostRatioPmMetric

Property Name: One Way Availability**Reference:** MEF 10.4 sec. 8.8.7 and MEF 26.2 sec. 12.13.7**JSON Name:** oneWayAvailabilityPmMetric - **Type:** List of object

Description: **oneWayAvailabilityPmMetric** is a reference to zero or more instances of One Way Availability Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way Availability. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...*] Unique Class: OneWayAvailabilityPmMetric

Property Name: One Way HLI**Reference:** MEF 10.4 sec. 8.8.8 and MEF 26.2 sec. 12.13.8**JSON Name:** oneWayHighLossIntervalsPmMetric - **Type:** List of object

Description: **oneWayHighLossIntervalsPmMetric** is a reference to zero or more instances of One Way High Loss Intervals Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way HLI. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...*] Unique Class: OneWayHighLossIntervalPmMetric

Property Name: One Way CHLI**Reference:** MEF 10.4 sec. 8.8.9 and MEF 26.2 sec. 12.13.9**JSON Name:** oneWayConsecutiveHighLossIntervalsPmMetric - **Type:** List of object

Description: **oneWayConsecutiveHighLossIntervalsPmMetric** is a reference to zero or more instances of One Way Consecutive High Loss Intervals Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way CHLI. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...*] Unique Class: OneWayConsecutiveHighLossIntervalsPmMetric

Property Name: One Way Composite Performance**Reference:** MEF 10.4 sec. 8.8.10**JSON Name:** oneWayCompositePmMetric - **Type:** List of object

Description: **oneWayCompositePmMetric** is a reference to zero or more instances of One Way Composite Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way Composite Performance. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...*] Unique Class: OneWayCompositePmMetric

Property Name: One Way Group Availability**Reference:** MEF 10.4 sec. 8.8.11 and MEF 26.2 sec. 12.13.10**JSON Name:** oneWayGroupAvailabilityPmMetric - **Type:** List of object

Description: **oneWayGroupAvailabilityPmMetric** is a reference to zero or more instances of One Way Group Availability Performance Metric. Each reference contains a set of parameters and performance objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way Group Availability. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

Allowed values: List of [0...*] Unique Class: OneWayAvailabilityPmMetric

**Class:** OneWayFrameDelayPmMetric (**One Way Frame Delay**)

One Way Frame Delay Performance Metric consisting of 2 parameters, a list of Ordered End Point Pairs (referred to as S in MEF 10.4 and MEF 26.2) , a percentile (referred to as Pd in MEF 10.4 and MEF 26.2), and the One Way Frame Delay Objective for this Class of Service Name and these two parameters.

Property Name: Frame Delay Ordered Pairs**Reference:** MEF 10.4 sec. 8.8.1.1 and MEF 26.2 sec. 12.13.1.1**JSON Name:** orderedPairList - **Type:** List of object

Description: The set (referred to as S in MEF 10.4 and MEF 26.2) of ordered pairs of EVC or OVC End Points over which this Performance Metric parameters and objective apply.

Allowed values: List of [1...*] Unique Class: OrderedPair

Validation Notes: Each EVC or OVC End Point in the pair must be one of the End Points in the service, they cannot be the same End Point, and at least one of them must be a *root* End Point.

Property Name: Frame Delay Percentile**Reference:** MEF 10.4 sec. 8.8.2 and MEF 26.2 sec. 12.13.2 (Pd)**JSON Name:** oneWayFdPercentile - **Type:** object

Description: A percentage in (0,100] for Frame Delay metric.

Allowed values: Type: Percentage

Validation Notes: This must be greater than 0.

Property Name: Frame Delay Objective**Reference:** MEF 10.4 sec. 8.8.2 and MEF 26.2 sec. 12.13.2**JSON Name:** oneWayFdObjective - **Type:** object

Description: The Performance Objective for this Performance Metric (this CoS and these 2 parameters). This is a unit of time greater than 0.

Allowed values: Class: TimeDuration

Class: OneWayMeanFrameDelayPmMetric (**One Way Mean Frame Delay**)

One Way Mean Frame Delay Performance Metric consisting of 1 parameter, a list of Ordered End Point Pairs (referred to as S in MEF 10.4 and MEF 26.2), and the One Way Mean Frame Delay Objective for this Class of Service Name and these Ordered End Point Pairs.

Property Name: Mean Frame Delay Ordered Pairs**Reference:** MEF 10.4 sec. 8.8.1.1 and MEF 26.2 sec. 12.13.1.1**JSON Name:** orderedPairList - **Type:** List of object

Description: The set (referred to as S in MEF 10.4 and MEF 26.2) of ordered pairs of EVC or OVC End Points over which this Performance Metric parameters and objective apply.

Allowed values: List of [1...*] Unique Class: OrderedPair

Validation Notes: Each EVC or OVC End Point in the pair must be one of the End Points in the service, they cannot be the same End Point, and at least one of them must be a *root* End Point.

Property Name: Mean Frame Delay Objective

Reference: MEF 10.4 sec. 8.8.3 and MEF 26.2 sec. 12.13.3

JSON Name: oneWayMfdObjective - **Type:** object

Description: The Performance Objective for this Performance Metric (this CoS and these Ordered End Point Pairs). This is a unit of time >0.

Allowed values: Class: TimeDuration

Class: OneWayFrameDelayRangePmMetric (**One Way Frame Delay Range**)

One Way Frame Delay Range Performance Metric consisting of 2 parameters, a list of Ordered End Point Pairs (referred to as S in MEF 10.4 and MEF 26.2) , a percentile (referred to as Pr in MEF 10.4 and MEF 26.2), and the One Way Frame Delay Range Objective for this Class of Service Name and these two parameters.

Property Name: Frame Delay Range Ordered Pairs

Reference: MEF 10.4 sec. 8.8.1.1 and MEF 26.2 sec. 12.13.1.1

JSON Name: orderedPairList - **Type:** List of object

Description: The set (referred to as S in MEF 10.4 and MEF 26.2) of ordered pairs of EVC or OVC End Points over which this Performance Metric parameters and objective apply.

Allowed values: List of [1...*] Class: OrderedPair

Validation Notes: Each EVC or OVC End Point in the pair must be one of the End Points in the service, they cannot be the same End Point, and at least one of them must be a *root* End Point.

Property Name: Frame Delay Range Percentile

Reference: MEF 10.4 sec. 8.8.4 and MEF 26.2 sec.12.13.4

JSON Name: oneWayFdrPercentile - **Type:** object

Description: A percentage in (0,100] for Frame Delay Range metric.

Allowed values: Type: Percentage

Validation Notes: This must be greater than 0.

Property Name: Frame Delay Range Objective

Reference: MEF 10.4 sec. 8.8.4 and MEF 26.2 sec.12.13.4

JSON Name: oneWayFdrObjective - **Type:** object

Description: The Performance Objective for this Performance Metric (this CoS and these 2 parameters). This is a unit of time >0.

Allowed values: Class: TimeDuration

Class: OneWayInterFrameDelayVariationPmMetric (**One Way InterFrame Delay Variation**)



One Way Frame Inter Frame Delay Variation Performance Metric consisting of 3 parameters, a list of Ordered End Point Pairs (referred to as S in MEF 10.4 and MEF 26.2), a time duration (referred to as delta-tau in MEF 10.4 and MEF 26.2), a percentile (referred to as Pv in MEF 10.4 and MEF 26.2), and the Inter Frame Delay Variation Objective for this Class of Service Name and these three parameters.

Property Name: Interframe Delay Variation Ordered Pairs**Reference:** MEF 10.4 sec. 8.8.1.1 and MEF 26.2 sec. 12.13.1.1**JSON Name:** orderedPairList - **Type:** List of object

Description: The set (referred to as S in MEF 10.4 and MEF 26.2) of ordered pairs of EVC or OVC End Points over which this Performance Metric parameters and objective apply.

Allowed values: List of [1...*] Class: OrderedPair

Validation Notes: Each EVC or OVC End Point in the pair must be one of the End Points in the service, they cannot be the same End Point, and at least one of them must be a *root* End Point.

Property Name: Time Duration**Reference:** MEF 10.4 sec. 8.8.5 and MEF 26.2 sec.12.13.5**JSON Name:** timeDuration - **Type:** object

Description: A time duration. This is the time difference between frames for which IFDV is measured. It is called delta-tau in MEF 10.4 and MEF 26.2.

Allowed values: Class: TimeDuration

Property Name: Interframe Delay Variation Percentile**Reference:** MEF 10.4 sec. 8.8.5 and MEF 26.2 sec.12.13.5**JSON Name:** oneWayIfdvPercentile - **Type:** object

Description: A percentage in (0,100] for Frame Delay metric.

Allowed values: Type: Percentage

Validation Notes: This must be greater than 0.

Property Name: Interframe Delay Variation Objective**Reference:** MEF 10.4 sec. 8.8.5 and MEF 26.2 sec.12.13.5**JSON Name:** oneWayIfdvObjective - **Type:** object

Description: The Performance Objective for this Performance Metric (this CoS and these 3 parameters). This is a unit of time >0.

Allowed values: Class: TimeDuration

Class: OneWayFrameLostRatioPmMetric (**One Way Frame Loss Ratio**)

One Way Frame Loss Ratio Performance Metric consisting of 1 parameter, a list of Ordered End Point Pairs (referred to as S in MEF 10.4 and MEF 26.2), and the One Way Frame Loss Ratio Objective for this Class of Service Name and these Ordered End Point Pairs.

**Property Name:** Frame Loss Ratio Ordered Pairs**Reference:** MEF 10.4 sec. 8.8.1.1 and MEF 26.2 sec. 12.13.1.1**JSON Name:** orderedPairList - **Type:** List of object

Description: The set (referred to as S in MEF 10.4 and MEF 26.2) of ordered pairs of EVC or OVC End Points over which this Performance Metric parameters and objective apply.

Allowed values: List of [1...*] Unique Class: OrderedPair

Validation Notes: Each EVC or OVC End Point in the pair must be one of the End Points in the service, they cannot be the same End Point, and at least one of them must be a *root* End Point.

Property Name: Frame Loss Ratio Objective**Reference:** MEF 10.4 sec. 8.8.6 and MEF 26.2 sec. 12.13.6**JSON Name:** oneWayFlrObjective - **Type:** object

Description: The Performance Objective for this Performance Metric (this CoS and these Ordered End Point Pairs).

Allowed values: Type: Percentage

Class: OneWayAvailabilityPmMetric (**One Way Availability**)

One Way Availability Performance Metric consisting of 1 parameter, a list of Ordered End Point Pairs (referred to as S in MEF 10.4 and MEF 26.2), and the One Way Availability Objective for this Class of Service Name and these Ordered End Point Pairs.

Property Name: Availability Ordered Pairs**Reference:** MEF 10.4 sec. 8.8.1.1 and MEF 26.2 sec. 12.13.1.1**JSON Name:** orderedPairList - **Type:** List of object

Description: The set (referred to as S in MEF 10.4 and MEF 26.2) of ordered pairs of EVC or OVC End Points over which this Performance Metric parameters and objective apply.

Allowed values: List of [1...*] Unique Class: OrderedPair

Validation Notes: Each EVC or OVC End Point in the pair must be one of the End Points in the service, they cannot be the same End Point, and at least one of them must be a *root* End Point.

Property Name: Availability Objective**Reference:** MEF 10.4 sec. 8.8.7 and MEF 26.2 sec. 12.13.7**JSON Name:** oneWayAvailabilityObjective - **Type:** object

Description: The PM Objective for this Performance Metric (this CoS and these Ordered End Point Pairs) expressed as a percentage.

Allowed values: Type: Percentage

Class: OneWayHighLossIntervalPmMetric (**One Way High Loss Intervals**)

One Way High Loss Intervals Performance Metric consisting of 1 parameter, a list of Ordered End Point Pairs (referred to as S in MEF 10.4 and MEF 26.2), and the One Way High Loss Intervals Objective for this Class of Service Name and these Ordered End Point Pairs.

Property Name: HLI Ordered Pairs**Reference:** MEF 10.4 sec. 8.8.1.1 and MEF 26.2 sec. 12.13.1.1**JSON Name:** orderedPairList - **Type:** List of object

Description: The set (referred to as S in MEF 10.4 and MEF 26.2) of ordered pairs of EVC or OVC End Points over which this Performance Metric parameters and objective apply.

Allowed values: List of [1...*] Unique Class: OrderedPair

Validation Notes: Each EVC or OVC End Point in the pair must be one of the End Points in the service, they cannot be the same End Point, and at least one of them must be a *root* End Point.

Property Name: HLI Objective**Reference:** MEF 10.4 sec. 8.8.8 and MEF 26.2 sec. 12.13.8**JSON Name:** oneWayHighLossIntervalsObjective - **Type:** integer

Description: The PM Objective for this Performance Metric (this CoS and these Ordered End Point Pairs) expressed as a non-negative number.

Allowed values: Minimum: 0

Class: OneWayConsecutiveHighLossIntervalsPmMetric (**One Way Consecutive High Loss Intervals**)

One Way Frame Delay Range Performance Metric consisting of 2 parameters, a list of Ordered End Point Pairs (referred to as S in MEF 10.4 and MEF 26.2), a number (referred to as p in MEF 10.4 and MEF 26.2), and the One Way Consecutive High Loss Intervals Objective for this Class of Service Name and these two parameters.

Property Name: CHLI Ordered Pairs**Reference:** MEF 10.4 sec. 8.8.1.1 and MEF 26.2 sec. 12.13.1.1**JSON Name:** orderedPairList - **Type:** List of object

Description: The set (referred to as S in MEF 10.4 and MEF 26.2) of ordered pairs of EVC or OVC End Points over which this Performance Metric parameters and objective apply.

Allowed values: List of [1...*] Unique Class: OrderedPair

Validation Notes: Each EVC or OVC End Point in the pair must be one of the End Points in the service, they cannot be the same End Point, and at least one of them must be a *root* End Point.

Property Name: CHLI Consecutive Number**Reference:** MEF 10.4 sec. 8.8.9 and MEF 26.2 sec. 12.13.9**JSON Name:** consecutiveNumberP - **Type:** integer

Description: The number of high loss intervals that constitute a consecutive high loss interval.

Allowed values: Minimum: 1

Property Name: CHLI Objective**Reference:** MEF 10.4 sec. 8.8.9 and MEF 26.2 sec. 12.13.9**JSON Name:** oneWayChliObjective - **Type:** integer

Description: Performance Objective expressed as a non-negative integer.



Allowed values: Minimum: 1

Class: OneWayCompositePmMetric (**One Way Composite**)

One Way Composite Performance Metric consisting of 7 parameters, a list of Ordered End Point Pairs (referred to as S in MEF 10.4), a Composite Performance number (referred to as U in MEF 10.4), indicators about Composite Frame Loss, Composite Frame Delay, and Composite Frame Delay Variation (referred to as Wfl, Wfd, and Wfdv in MEF 10.4), Thresholds for Frame Delay and Delay Variation (referred to as DL and Jt in MEF 10.4) and the One Way Composite Objective for this Class of Service Name and these seven parameters. (Note: This Performance Metric is not defined for OVC services).

Property Name: Composite Ordered Pairs

Reference: MEF 10.4 sec. 8.8.1.1

JSON Name: orderedPairList - **Type:** List of object

Description: The set (referred to as S in MEF 10.4) of ordered pairs of EVC End Points over which this Performance Metric parameters and objective apply.

Allowed values: List of [1...*] Unique Class: OrderedPair

Validation Notes: Each EVC End Point in the pair must be one of the End Points in the service, they cannot be the same End Point, and at least one of them must be a root End Point.

Property Name: Composite Performance Indicator Threshold

Reference: MEF 10.4 sec. 8.8.10

JSON Name: compositePerformanceIndicatorThreshold - **Type:** number

Description: Composite Performance Indicator threshold which if exceeded suggests an unacceptable time interval.

Allowed values: Minimum: 0 Maximum: 1

Property Name: Composite Frame Loss

Reference: MEF 10.4 sec. 8.8.10

JSON Name: compositeFrameLoss - **Type:** integer

Description: Composite indicator for One-way Frame Loss equal to 0 or 1.

Allowed values: Minimum: 0 Maximum: 1

Property Name: Composite Frame Delay

Reference: MEF 10.4 sec. 8.8.10

JSON Name: compositeFrameDelay - **Type:** integer

Description: Composite indicator for One-way Frame Delay equal to 0 or 1.

Allowed values: Minimum: 0 Maximum: 1

Property Name: Composite Frame Delay Variation

Reference: MEF 10.4 sec. 8.8.10

JSON Name: compositeFrameDelayVariation - **Type:** integer

Description: Composite indicator for One-way Frame Delay Variation equal to 0 or 1.



Allowed values: Minimum: 0 Maximum: 1

Property Name: *FD Threshold*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *oneWayFdThreshold* - **Type:** object

Description: Composite One-way Frame Delay threshold in time units.

Allowed values: Class: TimeDuration

Property Name: *IFDV Threshold*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *oneWayIfdvThreshold* - **Type:** object

Description: One-way Frame Delay Variation threshold in time units.

Allowed values: Class: TimeDuration

Property Name: *Composite Objective*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *cpmObjective* - **Type:** object

Description: The PM Objective for this Performance Metric (this CoS and these 7 parameters) expressed as a percentage.

Allowed values: Type: Percentage

Class: OneWayGroupAvailabilityPmMetric (**One Way Group Availability**)

The One Way Group Availability Performance Metric has 2 parameters. The first parameter is a list of sets of Ordered End Point Pairs (referred to as G in MEF 10.4 and MEF 26.2). (Note that this is different than the other Performance Metrics which have a list of Ordered End Point Pairs). The second parameter is the number of sets that must be available (referred to as K in MEF 10.4 and MEF 26.2). The third element is the Performance Objective for the One Way Group Availability.

Property Name: *GA Ordered Pairs*

Reference: MEF 10.4 sec. 8.8.11 and MEF 26.2 sec. 12.13.10

JSON Name: *setOfOrderedPairs* - **Type:** List of object

Description: This is a list of lists of Ordered End Point Pairs.

Allowed values: List of [1...*] Unique Class: orderedPairList

Property Name: *Minimum Number of Sets Available*

Reference: MEF 10.4 sec. 8.8.11 and MEF 26.2 sec. 12.13.10

JSON Name: *minimumNumberOfSetsAvailableK* - **Type:** integer

Description: Specified number of sets of Ordered End Point Pairs available during characterized percentage of time of measurement.

Allowed values: Minimum: 1

Property Name: GA Objective
Reference: MEF 10.4 sec. 8.8.11 and MEF 26.2 sec. 12.13.10
JSON Name: oneWayGroupAvailabilityObjective - **Type:** object

Description: This is the Performance Objective for this Performance Metric expressed as a Percentage.

Allowed values: Type: Percentage

Class: OrderedPair (**Ordered Pair**)

Performance Metrics (except for Group Availability) are specified for a list of Ordered End Point Pairs. In other words a Frame Delay objective (for example) is specified from (A to B) and from (C to D). This class defines one Ordered End Point Pair.

Property Name: From EVC/OVC End Point
JSON Name: fromCarrierEthernetServiceEndPoint - **Type:** string

Description: This is the EVC End Point ID for an EVC End Point in the service or OVC End Point ID for an OVC End Point in the service.

Allowed values: [Any String]

Property Name: To EVC/OVC End Point
JSON Name: toCarrierEthernetServiceEndPoint - **Type:** string

Description: This is the EVC End Point ID for an EVC End Point in the service or OVC End Point ID for an OVC End Point in the service.

Allowed values: [Any String]

Class: OrderedPairList (**Sets of Ordered Pairs**)

The Group Availability is specified for sets of Ordered End Point Pairs, e.g., G1={(A,B), (A,C)} and G2={(A,D), (A,E)}. This class defines one set of Ordered End Point Pairs.

Property Name: List of Ordered Pairs
JSON Name: orderedPairList - **Type:** List of object

Allowed values: List of [1...*] Unique Class: OrderedPair

786

787 **15.10 Carrier Ethernet Standard Types**

Carrier Ethernet Types

Schema File Name: carrierEthernetTypes

Includes classes:

- EnabledDisabled (Enabled Disabled)
- PcpValueAndDiscard (PCP Value and Discard Enum)
- DeiValueAndDiscard (DEI and Discard Enum)
- LinkAggType (LinkAgg Enum)
- FrameColor (FrameColor Enum)

L2cpAddressSet (L2CP Address Set Enum)
MegList (MEG List Enum)
AvailableMegList (Available MEG List Enum)
PcpValue (PCP Value Enum)
TimeDurationUnits (Time Duration Units)

Type: EnabledDisabled (***Enabled Disabled***)

Enumeration to indicate Enabled/Disabled state of an attribute

Property Name: Enabled Disabled **Type:** enum

Allowed values: Enum: "ENABLED", "DISABLED"

Type: PcpValueAndDiscard (***PCP Value and Discard Enum***)

Property Name: PCP Value and Discard **Type:** enum

Allowed values: Enum: "0", "1", "2", "3", "4", "5", "6", "7", "DISCARD"

Type: DeiValueAndDiscard (***DEI and Discard Enum***)

Property Name: DEI Value and Discard **Type:** enum

Allowed values: Enum: "0", "1", "DISCARD"

Type: LinkAggType (***LinkAgg Enum***)

Type that defines the Link Aggregation types.

Property Name: Link Aggregation **Type:** enum

Allowed values: Enum: "NONE", "2_LINK_ACTIVE_STANDBY", "ALL_ACTIVE", "OTHER"

Type: FrameColor (***FrameColor Enum***)

Property Name: Frame Color **Type:** enum

Allowed values: Enum: "GREEN", "YELLOW"

Type: L2cpAddressSet (**L2CP Address Set Enum**)

Property Name: L2CP Address Set Enum **Type:** enum

Allowed values: Enum: "CTA", "CTB", "CTB2"

Type: MegList (**MEG List Enum**)

Property Name: MEG List Enum **Type:** enum

Allowed values: Enum: "0", "1", "2", "3", "4", "5", "6", "7", "NONE"

Type: AvailableMegList (**Available MEG List Enum**)

Property Name: Available Meg List Enum **Type:** enum

Allowed values: Enum: "0", "1", "2", "3", "4", "5", "6"

Type: PcpValue (**PCP Value Enum**)

Property Name: PCP Value Enum **Type:** enum

Allowed values: Enum: "0", "1", "2", "3", "4", "5", "6", "7"

Type: TimeDurationUnits (**Time Duration Units**)

Enum listing the units of measure for a Time Duration

Property Name: Time Duration Units Enum **Type:** enum

Allowed values: Enum: "NS", "US", "MS", "SEC", "MIN", "HOUR", "DAY", "WEEK", "MONTH", "YEAR"

788

789 15.11 Carrier Ethernet Utility Classes

Carrier Ethernet Utility Classes

Schema File Name: carrierEthernetUtilityClasses

Includes classes:

MepLevelAndDirection (MEP Specification)

SourceMacAddressLimit (SA Limit)

AggLinkDepth (Link Depth)

FrameDisposition (Frame Disposition)
FrameDispEnum (Frame Disposition Enumeration)
InformationRate (Information Rate)
DataSize (Data Size)
TimeDuration (Time Duration)
Percentage (Percentage)

Class: MepLevelAndDirection (***MEP Specification***)

A 2-tuple that defines the MEG Level and MEP direction.

Property Name: MEG Level
JSON Name: level - **Type:** integer

Description: The MEG level

Allowed values: Minimum: 0 Maximum: 7

Property Name: MEP Direction
JSON Name: mepDirection - **Type:** enum

Description: Indicates whether this is an Up MEP or Down MEP

Allowed values: Enum: "UP", "DOWN"

Class: SourceMacAddressLimit (***SA Limit***)

This limits the number of source MAC addresses that can be used in ingress frame mapped to the End Point of all types over a time interval. It is a recommendation that each Ingress EI Frame with a source MAC address that would exceed the limit be discarded by the Carrier Ethernet Network.

Property Name: Number of Addresses
JSON Name: limit - **Type:** integer

Description: The number of MAC source addresses that will not subject an Ingress EI Frame to discard during the time interval.

Allowed values: Minimum: 1

Property Name: Time Interval
JSON Name: interval - **Type:** object

Description: The time interval in seconds over which the source address limit is evaluated

Allowed values: Class: TimeDuration

Class: AggLinkDepth (***Link Depth***)

For an Operator Service, this is a pair of indicating that a given VLAN ID maps to a given number of links in the Port Conversation ID to Aggregation Link Map.

Property Name: *vlanId*
JSON Name: *vlanID* - **Type:** integer

Description: The VLAN ID

Allowed values: Minimum: 1 Maximum: 4094

Property Name: *linkDepth*
JSON Name: *lagDepth* - **Type:** integer

Description: The number of Link Aggregation links available to the VLAN

Allowed values: Minimum: 1

Class: FrameDisposition (**Frame Disposition**)

A 3-tuple that indicates how Unicast, Multicast, and Broadcast Frames are delivered by the service. For each, the value can be UNCONDITIONAL, the frames are delivered without condition, CONDITIONAL, the frames are delivered based on conditions agreed to between buyer and seller (e.g., based on learned addresses), and DISCARD.

Property Name: *Unicast Frames*
JSON Name: *unicast* - **Type:** object

Allowed values: Type: FrameDispEnum

Property Name: *Multicast Frames*
JSON Name: *multicast* - **Type:** object

Allowed values: Type: FrameDispEnum

Property Name: *Broadcast Frames*
JSON Name: *broadcast* - **Type:** object

Allowed values: Type: FrameDispEnum

Type: FrameDispEnum (**Frame Disposition Enumeration**)

Property Name: *frameDispEnum* **Type:** enum

Allowed values: Enum: "DISCARD", "DELIVER_UNCONDITIONALLY", "DELIVER_CONDITIONALLY"

Class: InformationRate (***Information Rate***)

A value and a unit of measure that specifies an Information Rate.

Property Name: *IR Value*
JSON Name: *irValue* - **Type:** *number*

Description: The value in the information rate. For example if the information rate is 70 kbps this element is 70.

Allowed values: Minimum: 0

Property Name: *IR Units*
JSON Name: *irUnits* - **Type:** *enum*

Description: The unit of measure for the Information Rate. For example if the Information Rate is 70KBPS this element is KBPS. Note that the values are decimal values. 1 KBPS is 1000 bits per second and 1MBPS is 1,000,000 bits per second.

Allowed values: Enum: "BPS", "KBPS", "MBPS", "GBPS", "TBPS", "PBPS", "EBPS", "ZBPS", "YBPS"

Class: DataSize (***Data Size***)

A value and a unit of measure that specifies a data size for example for a buffer or a burst size.

Property Name: *Data Size Value*
JSON Name: *dataSizeValue* - **Type:** *integer*

Description: The value in the data size. For example, if a burst size is 40 KBYTES, this element is 40.

Allowed values: [Any Integer]

Property Name: *Data Size Units*
JSON Name: *dataSizeUnits* - **Type:** *enum*

Description: The unit of measure in the data size. For example, if a burst size is 40 KBYTES, this element is KBYTES. Note that the units indicate binary values, e.g., KBYTES is 1024 bytes.

Allowed values: Enum: "BYTES", "KBYTES", "MBYTES", "GBYTES", "TBYES", "PBYES", "EBYES", "ZBYTES", "YBYTES"

Class: TimeDuration (***Time Duration***)

This class is used to describe durations expressed as a 2-tuple, (value, units). The units from from nanoseconds to years.

Property Name: *Time Duration Value*
JSON Name: *timeDurationValue* - **Type:** *integer*

Description: The value of the duration. For example, if the duration is 20 ms, this element is 20.

Allowed values: [Any Integer]



Property Name: Time Duration Units
JSON Name: timeDurationUnits - **Type:** object

Description: The unit of measure in the duration. For example, if an interval is 2ms, this element is MS.

Allowed values: Type: TimeDurationUnits

Type: Percentage (**Percentage**)

This is a number of percent - a floating point number between 0 and 100

Property Name: Percentage **Type:** number

Allowed values: Minimum: 0 Maximum: 100

790

791

792 16 Service Attributes Not Included

793 There are 16 Services Attributes from MEF 26.2 that are not included in the API. The two primary reasons
794 for not including them is:

- 795 • They are part of the product agnostic envelope (i.e., many of the Identifiers), and
- 796 • The Service Attribute has a single constant value for Access E-Line. There is no benefit to
797 including these in the API since the only possible use would be to validate that they are correct.
798 There is no information to be gleaned if they are correct.

799 These Service Attributes are listed in the following table.

800

Group	Service Attribute	Discussion
OVC Service Attribute	OVC Identifier	Identifiers for Order Items are not needed since they are in the product-agnostic envelope.
OVC Service Attribute	OVC Type	Not needed for Access E-Line. Only value allowed for Access E-Line is point-to-point
OVC Service Attribute	OVC End Point List	This is a list of OVC End Point IDs. This is not needed since the OVC End Points are included in the OVC definition.
OVC Service Attribute	Maximum Number of UNI OVC End Points	Not needed since it must be "1" for Access E-Line
OVC Service Attribute	Maximum Number of ENNI OVC End Points	Not needed since it must be "1" for Access E-Line
OVC Service Attribute	OVC S-VLAN PCP Preservation	Not needed because there is only one ENNI in the service.
OVC Service Attribute	OVC S-VLAN DEI Preservation	Not needed because there is only one ENNI in the service.
OVC End Point Service Attributes	OVC End Point Type	This indicates whether the End Point is at a UNI or an ENNI. For Access E-Line this information is embedded in the OVC Property containing the End Point (uniEP and enniEP)
OVC End Point Service Attributes	OVC End Point External Interface Identifier	For Access E-Line there is only one UNI and one ENNI and they are identified in the API (as related products), so they don't need to be explicitly called out in the product configuration. For most other Ethernet Services this attribute is needed.
OVC End Point Service Attributes	OVC End Point Role	This is always ROOT for Access E-Line at both the UNI and the ENNI, so not needed.

Group	Service Attribute	Discussion
OVC End Point Service Attributes	OVC End Point Ingress Bandwidth Profile per OVC End Point Service Attribute	Per MEF 51.1 this MUST be Disabled at both the UNI and ENNI for all Operator Services.
Operator UNI Service Attributes	Operator UNI Identifier	Identifiers for Order Items are not needed since they are in the product-agnostic envelope.
Operator UNI Service Attributes	Operator UNI Service Frame Format	Not needed, only a single value allowed: "802.3-2012"
Operator UNI Service Attributes	Operator UNI E-LMI	Not included since MEF 10.4 no longer includes and E-LMI attribute.
Operator UNI Service Attributes	Operator UNI L2CP Address Set	Per MEF 45.1: If an OVC has an OVC End Point at a UNI, then the value of the Operator UNI L2CP Address Set Service Attribute MUST be the same as the OVC L2CP Address Set Service Attribute.
ENNI Service Attributes	Operator ENNI Identifier	Identifiers for Order Items are not needed since they are in the product-agnostic envelope.
ENNI Common Attributes	ENNI Frame Format Common Attribute	Not Needed: Each ENNI-N MUST comply with the Ethernet Media Access Control standards specified in Clauses 3 and 4 of IEEE Std 802.3-2012 [4].

801

Table 6 – Service Attributes Not Included in Schemas

802

17 References

- [1] IEEE Std 802.1AX-2014, Link Aggregation, December 2014
- [2] IEEE Std 802.3-2012, IEEE Standard for Ethernet, August 2012
- [3] IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, by Scott Bradner, March 1997
- [4] IETF RFC 8174, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, by Barry Leiba, May 2017. Copyright © IETF Trust and the persons identified as the document authors (2017). All Rights Reserved.
- [5] MEF 4, Metro Ethernet Network Architecture - Part 1: Generic Framework, May 2004
- [6] MEF 6.3, Subscriber Ethernet Services Definitions, November 2019
- [7] MEF 7.4, Carrier Ethernet Services Information Model, December 2020
- [8] MEF 10.4, Subscriber Ethernet Service Attributes, December 2018
- [9] MEF 26.2, External Network Network Interfaces (ENNI) and Operator Service Attributes, August 2016
- [10] MEF 45.1, Layer 2 Control Protocols in Ethernet Services, December 2018
- [11] MEF 51.1, Operator Ethernet Service Definitions, December 2018
- [12] MEF 55.1, Lifecycle Service Orchestration (LSO): Reference Architecture and Framework, January 2021
- [13] MEF 57.2, Draft Release 3 Product Order Management Requirements and Use Cases, January, 2022
- [14] MEF 78.1, MEF Core Model, July 2020
- [15] MEF 79, Address, Service Site, and Product Offering Qualification Management, Requirements and Use Cases, November 2019
- [16] MEF 81, Product Inventory Management, Requirements and Use Cases, November 2019
- [17] MEF 87, LSO Cantata and LSO Sonata Product Offering Qualification Management API – Developer Guide, January 2022
- [18] MEF 115, LSO Cantata and LSO Sonata Quote Management API – Developer Guide, January 2022
- [19] MEF 121, LSO Cantata and LSO Sonata Address Management API – Developer Guide, January 2022



833 [20] MEF W123, LSO Cantata and LSO Sonata Product Order Management API – Developer
834 Guide, January 2022

835

836 Appendix A Usage examples (Informative)

837 This appendix aims to provide an extensive set of examples to cover:

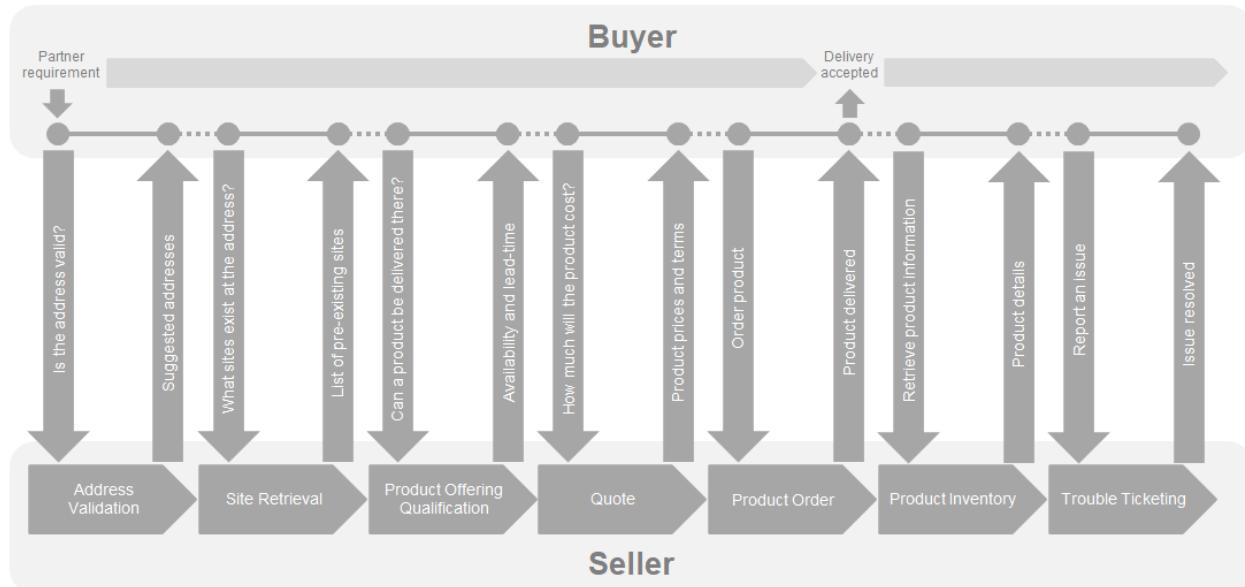
- 838 • different configuration variants (Use Cases 2 and 3)
- 839 • basic all APIs steps walkthrough to order an Access E-Line product (Use Cases 1, 2, 4, 5, 6)
- 840 • common modifications (Use Cases 7-11, 13)
- 841 • deletion of a product (Use Case 12)

842 The examples are delivered in two forms:

- 843 • as part of this document – to allow comments and rich explanation
- 844 • as a Postman collection – for ease of use in testing.

845 A.1 High-Level flow

846 The Cantata and Sonata Interface Reference Points are formed from a set of APIs the serve different functions in the end-to-end flow. Figure 9 shows all of the functions and their sequence.



848
849 **Figure 9 – Cantata and Sonata End-to-End Function Flow**

- 850
- 851 • Address Validation - allows the Buyer to retrieve address information from the Seller, including exact formats, for addresses known to the Seller.
- 852
- 853 • Site Retrieval - allows the Buyer to retrieve Service Site information including exact formats for Service Sites known to the Seller.
- 854

- 855 • Product Offering Qualification (POQ) - allows the Buyer to check whether the Seller can de-
856 liver a product or set of products from among their product offerings at the geographic address
857 or a service site specified by the Buyer; or modify a previously purchased product.
- 858 • Quote - allows the Buyer to submit a request to find out how much the installation of an in-
859 stance of a Product Offering, an update to an existing Product, or a disconnect of an existing
860 Product will cost.
- 861 • Product Order - allows the Buyer to request the Seller to initiate and complete the fulfillment
862 process of an installation of a Product Offering, an update to an existing Product, or a discon-
863 nect of an existing Product at the address defined by the Buyer.
- 864 • Product Inventory - allows the Buyer to retrieve the information about existing Product in-
865 stances from Seller's Product Inventory.
- 866 • Trouble Ticketing - allows the Buyer to create, retrieve, and update Trouble Tickets as well as
867 receive notifications about Incidents' and Trouble Tickets' updates. This allows managing is-
868 sues and situations that are not part of normal operations of the Product provided by the Seller.

869 All of the above-mentioned APIs are provided in the SDK together with accompanying Developer Guides.
870 Please refer to those documents for more details and examples of particular functional APIs.

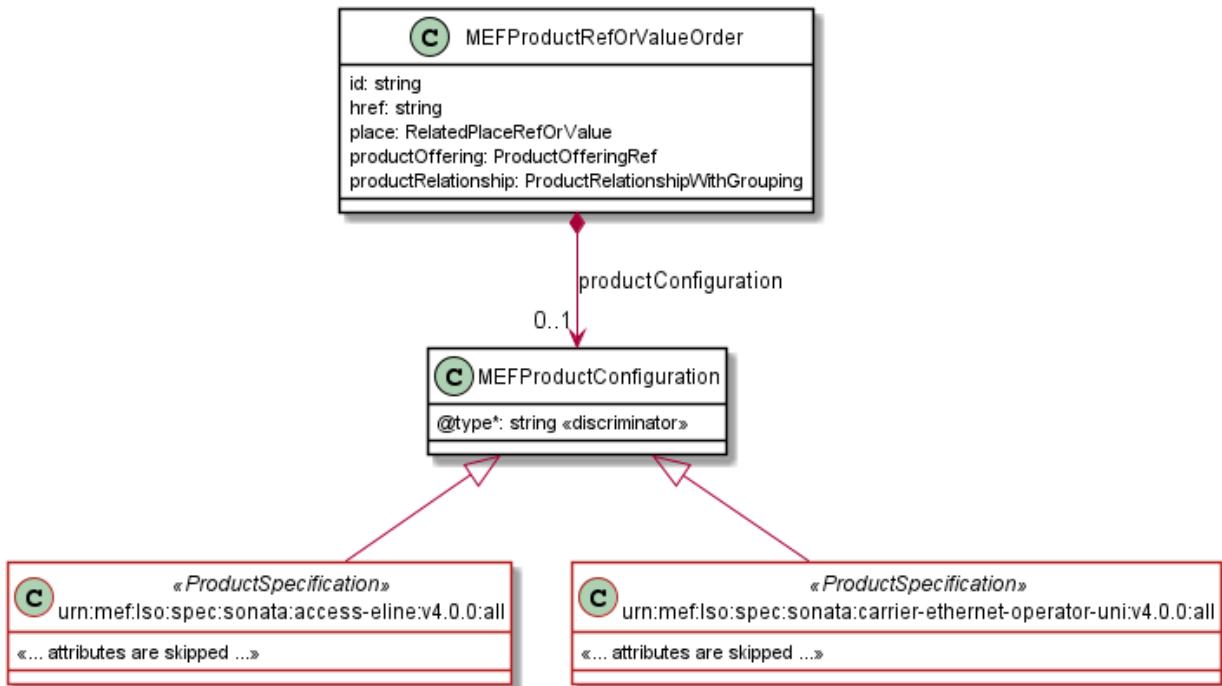
871 **A.2 Integration of product specifications into the APIs.**

872 The above-mentioned APIs are product-agnostic in the meaning that they serve as a business interaction
873 level between the Buyer and the Seller and they do not contain any product-specific information in their
874 specifications. In order to pass the product-specific information, an extension pattern must be used. This
875 applies to four APIs that carry product-specific information: POQ, Quote, Product Order, and Product In-
876 ventory.

877 The extension hosting type in the API data model is “MEFProductConfiguration”. The “@type” attribute
878 of that type must be set to a value that uniquely identifies the product specification (Figure 10). A unique
879 identifier for MEF standard product specifications is in URN format and is assigned by MEF. This identifier
880 is provided as root schema “\$id” and in product specification documentation. In this case, this will be one
881 of:

- 882 • urn:mef:lso:spec:sonata:access-eline:v4.0.0:all
- 883 • urn:mef:lso:spec:sonata:carrier-ethernet-operator-uni:v4.0.0:all
- 884 • urn:mef:lso:spec:sonata:carrier-ethernet-enni-sp-so:v4.0.0:inventory

885 Use of non-MEF standard product definitions is allowed. In such a case the schema identifier must be
886 agreed upon between the Buyer and the Seller.



887

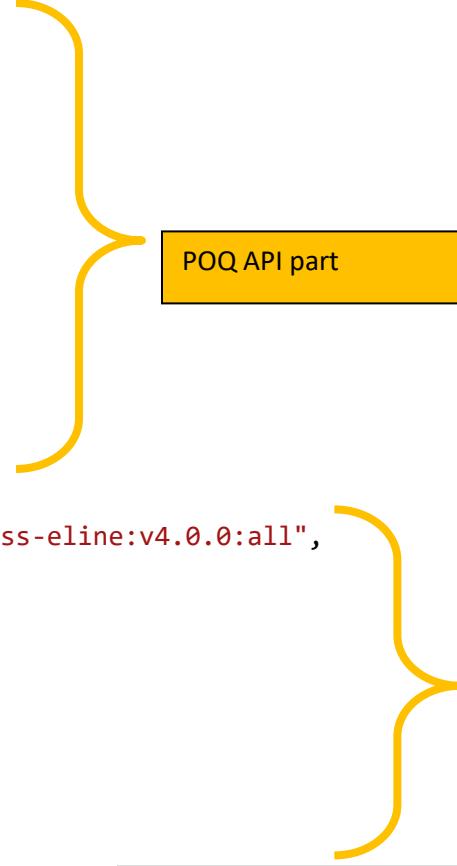
888

Figure 10 – The Extension Pattern

889 Product specifications are provided as Json schemas without the “MEFProductConfiguration” context.
 890 Product-specific attributes are introduced via the “MEFProductRefOrValue” (defined by the Buyer). This
 891 entity has the “productConfiguration” attribute of type “MEFProductConfiguration” which is used as an
 892 extension point for product-specific attributes. The example result of such binding in a request payload may
 893 look like this (for POQ):

894

```
895 {  
896     "instantSyncQualification": true,  
897     "externalId": "BuyerPoq-00001",  
898     "provideAlternative": false,  
899     "projectId": "BuyerProjectX",  
900     "productOfferingQualificationItem": [  
901         {  
902             "id": "item-001",  
903             "action": "add",  
904             "product": {  
905                 "productOffering": {  
906                     "id": "000073"  
907                 },  
908                 "productConfiguration": {  
909                     "@type": "urn:mef:lso:spec:sonata:access-eline:v4.0.0:all",  
910                     "ceVlanIdPreservation": "PRESERVE",  
911                     "cTagPcpPreservation": "ENABLED",  
912                     "cTagDeiPreservation": "ENABLED",  
913                     "maximumFrameSize": 1526,  
914                     "ovcL2cpAddressSet": "CTA",  
915                     <<the rest of the attributes omitted>>  
916                     ...  
917                 }  
918             }  
919         }  
920     ]  
921 }  
922 }
```



POQ API part

Access E-Line Product part

923 **A.3 action: add**

924 This section guides through all the steps of Sonata and Cantata APIs that need to be performed in order to
925 successfully order an Access E-Line product.

926 Note: Sellers are free to mandate some of these steps.

927 Note: As the examples of particular steps in many cases will replicate the product-specific information, in
928 some of the snippets some parts of it will be omitted for better readability.

929 There are common rules for all request items for creation requests (POQ, Quote, Order):

- 930 - “item.action” must be set to “add”
931 - “item.product.id” must not be provided
932 - “product.productConfiguration” must contain all desired configurations.

933 **A.3.1 Use Case 1: Address Validation**

934 For detailed guidance on how to use the Address Validation API, please refer to MEF 121 [19].

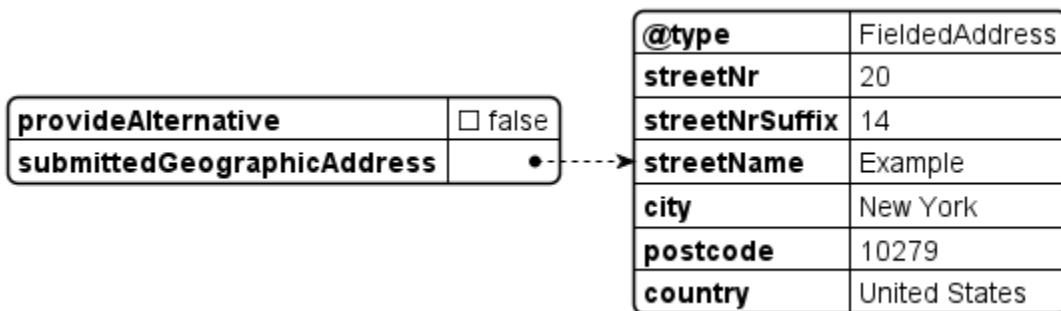
935 The first step of the process is the Address Validation. The aim of this step is to align the address representation
936 between the Buyer and the Seller. This is to overcome the very common problem of different address
937 representation in various countries and systems. The Buyer sends a representation of the address that is
938 intended to be used in further steps (most likely an installation place). The question is “Dear Seller – do
939 you recognize and understand this address?”. Additionally, the Buyer may also ask the Seller to provide
940 alternatives, if there is no clear match. The Seller provides a response where in the “bestMatchGeographicAddress”
941 (if found) a matching address is provided with an id that can be used in further steps to avoid
942 the need for Address resolution.

943 Note: It is not mandatory for the Seller to provide the Id of the returned Address, yet it is recommended.

944 Note: The Seller’s response might come with some enhancements in the Address. It is up to the Seller’s
945 discretion what makes the best match and an alternative.

946 The Buyer in the request places one of 4 possible representations of the Address (FieldedAddress, FormattedAddress,
947 MEG Geographic Point, or GeographicAddressLabel). The following Figure and snippet present an example request:

949



950

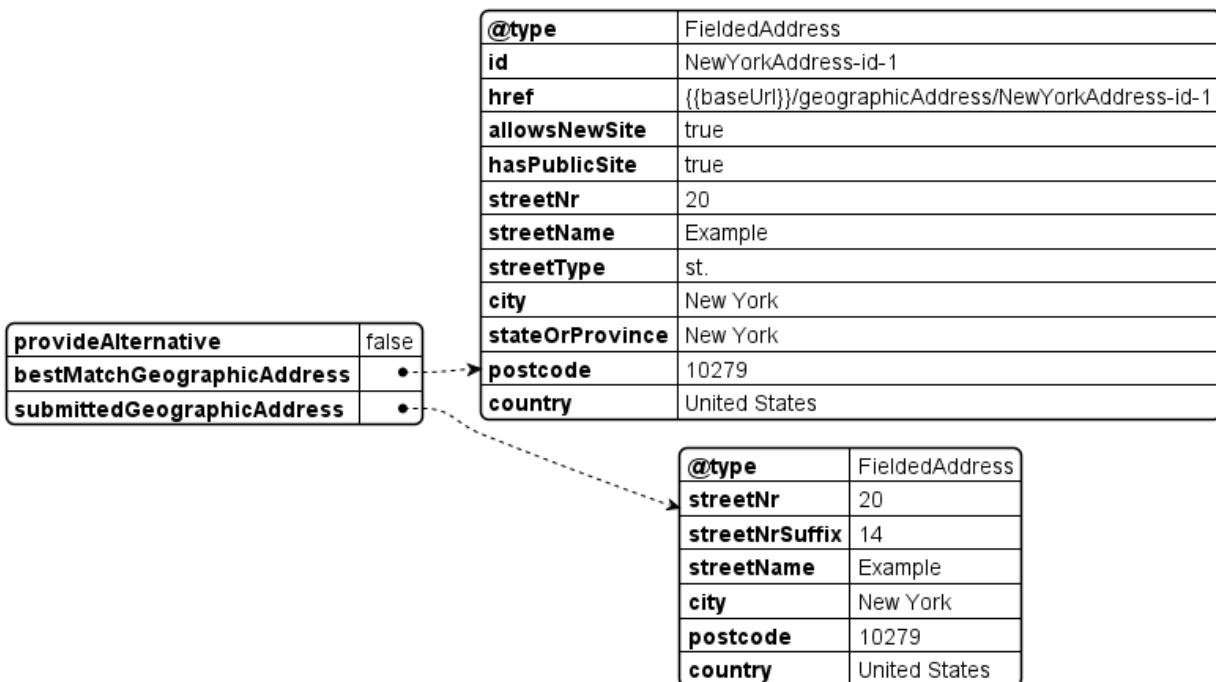
951 **Figure 11 – UC1: Address Validation request**

952 Example Address Validation Request:

```
953 {  
954     "provideAlternative": false,  
955     "submittedGeographicAddress": {  
956         "@type": "FieldedAddress",  
957         "streetNr": "20",  
958         "streetNrSuffix": "14",  
959         "streetName": "Example",  
960         "city": "New York",  
961         "postcode": "10279",  
962         "country": "United States"  
963     }  
964 }
```

965 In the response, the Seller repeats the submitted address for reference and populates the “bestMatchGeographicAddress” and/or the “alternateGeographicAddress”. In the example, the Seller matches the best
966 match address, which has a little more details than the one in the request. The Seller also provides the
967 address id (“NewYorkAddress-id-1”) that the Buyer will refer to in later steps.
968

969 **Note:** The identifiers will most likely be some kind of technical ids to provide uniqueness. In all examples,
970 the identifiers are shortened and made human-readable to make it easier to read and match across the use
971 cases.



972

973

Figure 12 – UC1: Address Validation response

974

975 Seller's response:

```
976 {
977     "provideAlternative": "false",
978     "bestMatchGeographicAddress": {
979         "@type": "FieldedAddress",
980         "id": "NewYorkAddress-id-1",
981         "href": "{baseUrl}/geographicAddress/NewYorkAddress-id-1",
982         "allowsNewSite": "true",
983         "hasPublicSite": "true",
984         "streetNr": "20",
985         "streetName": "Example",
986         "streetType": "st.",
987         "city": "New York",
988         "stateOrProvince": "New York",
989         "postcode": "10279",
990         "country": "United States"
991     },
992     "submittedGeographicAddress": {
993         "@type": "FieldedAddress",
994         "streetNr": "20",
995         "streetNrSuffix": "14",
996         "streetName": "Example",
997         "city": "New York",
998         "postcode": "10279",
999         "country": "United States"
1000    }
1001 }
```

1002 A.3.2 Use Case 2: POQ - new Access E-Line, new UNI, low class of service

1003 For detailed guidance on how to use the Product Offering Qualification API, please refer to MEF 87 [17].

1004 The Product Offering Qualification step is designed for the Buyer to ask the question “Dear Seller, are you
1005 able to provide a certain product (based on “productOffering” and “productConfiguration”) at a given lo-
1006 cation”? The Seller responds with qualification confidence:

- 1007 • green - The Seller has high confidence that this Product can be delivered,
- 1008 • yellow - The Seller believes they can deliver the Product but is not highly confident,
- 1009 • red - The Seller cannot deliver the Product as specified.

1010 In case of yellow or red, additionally, the Seller may return (if requested) an alternative Product Offering,
1011 that might fulfill the Buyer’s needs.

1012 It is very important to understand the pattern of integrating the product configuration (so-called “payload”)
1013 with the functional product-agnostic API (“envelope”). As explained in chapter 11, the Access E-Line prod-
1014 uct model is composed of 3 elements (products):

- 1015 • the Access E-Line itself. It contains the “uniEp” and “enniEp” attributes, which carry some
 1016 endpoint configuration information, yet these are no references to the UNI or ENNI products.
 1017 • the UNI
 1018 • the ENNI.

1019 The information about one single product is carried within the Product Offering Qualification (POQ) API
 1020 by a single “productOfferingQualificationItem” being a subject to qualification. One POQ Request can
 1021 carry more than one POQ Items, that may or may not be related to each other.

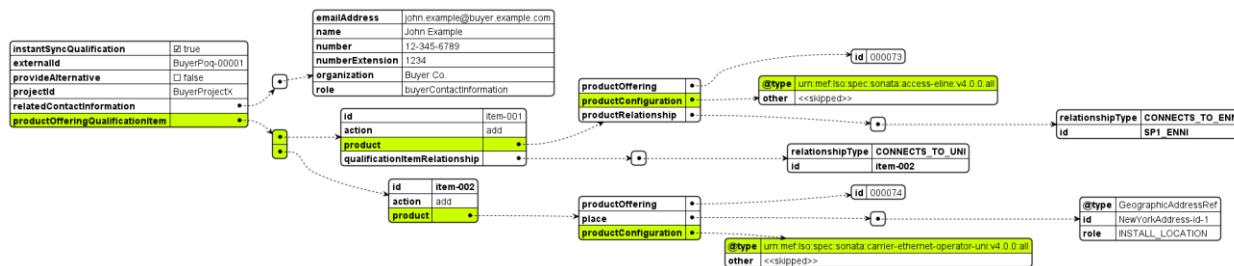
1022 There are 2 ways to reference products:

- 1023 • existing Products – present in the Inventory at the moment of issuing the request, to which the
 1024 Buyer has the “product.id”. These must be referenced by “productOfferingQualifica-
 1025 tionItem.product.productRelationship” with appropriate “product.id” and “relationshipType”.
 1026 Product Specification defines what roles must be used during referencing other products. The
 1027 Access E-Line points to the ENNI product with the “relationshipType” value: “CON-
 1028 NECTS_TO_ENNI” (as specified in Chapter 13).
 1029 • newly created or modified products – ones being created or modified by other POQ Item in
 1030 the same POQ request, so there is a relation between the Items within a POQ. These must be
 1031 referenced using the “productOfferingQualificationItem.qualificationItemRelationship” by
 1032 the target Item “id” and the “relationshipType” (CONNECTS_TO_UNI).

1033 In this use case, both the Access E-Line (OVC) and the UNI products are created or, to be more precise, a
 1034 request to qualify if the creation of both of them is possible. It is assumed, the ENNI is always existing,
 1035 before ordering an Access E-Line (Chapter 9). Since 2 products are being subject to qualification, the POQ
 1036 request contains 2 items with “action=add”. The Access E-Line POQ Item has 2 relations:

- 1037 • to the ENNI, which exists in the Inventory already – by
 1038 “productOfferingQualificationItem.product.productRelationship”.
 1039 • to the UNI, which is being qualified in the same request – by
 1040 “productOfferingQualificationItem.qualificationItemRelationship”

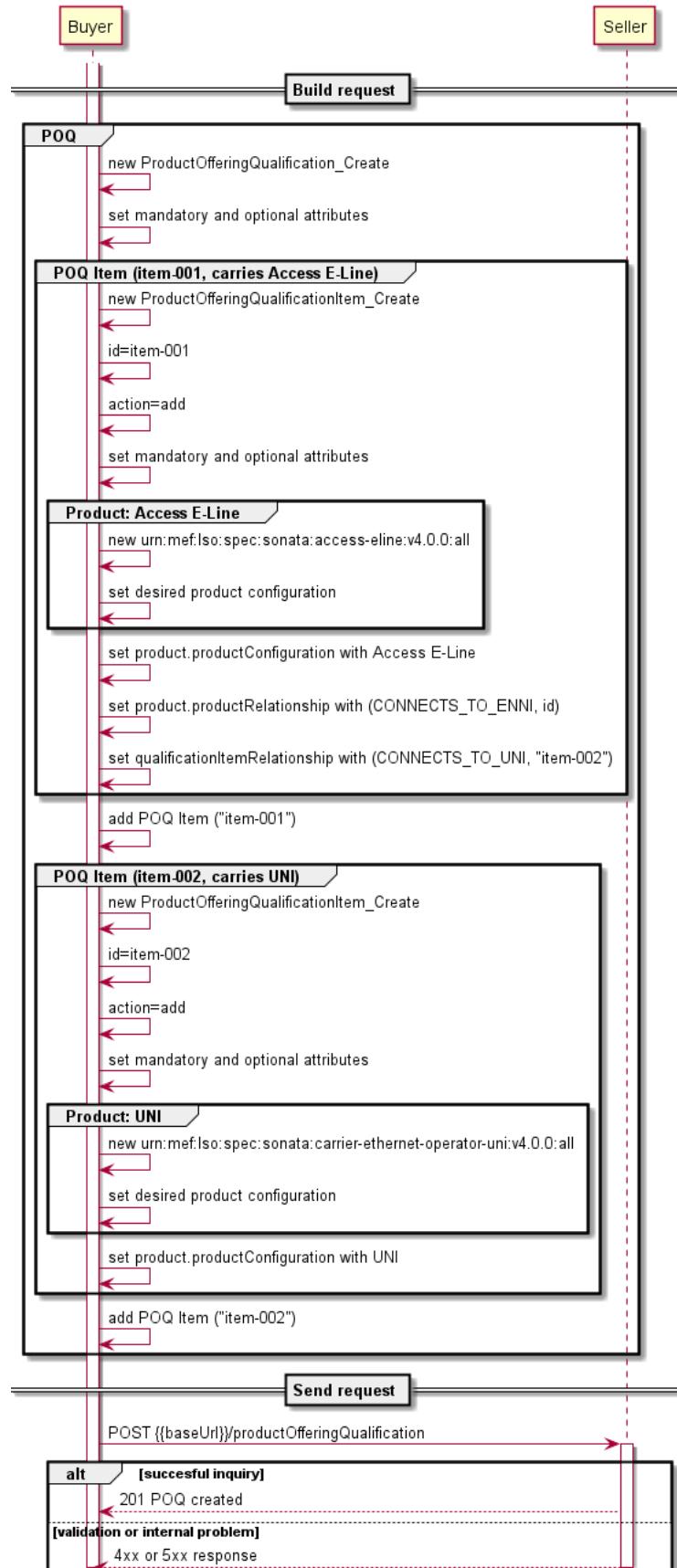
1041 An instance diagram in Figure 13 shows an extracted part from the request, to present the most important
 1042 integration-related attributes. The product configuration attached to a POQ request is highlighted with green
 1043 color, and the product relations are highlighted with a bold font.



1044

1045 **Figure 13 – UC2: POQ Request, envelope part**

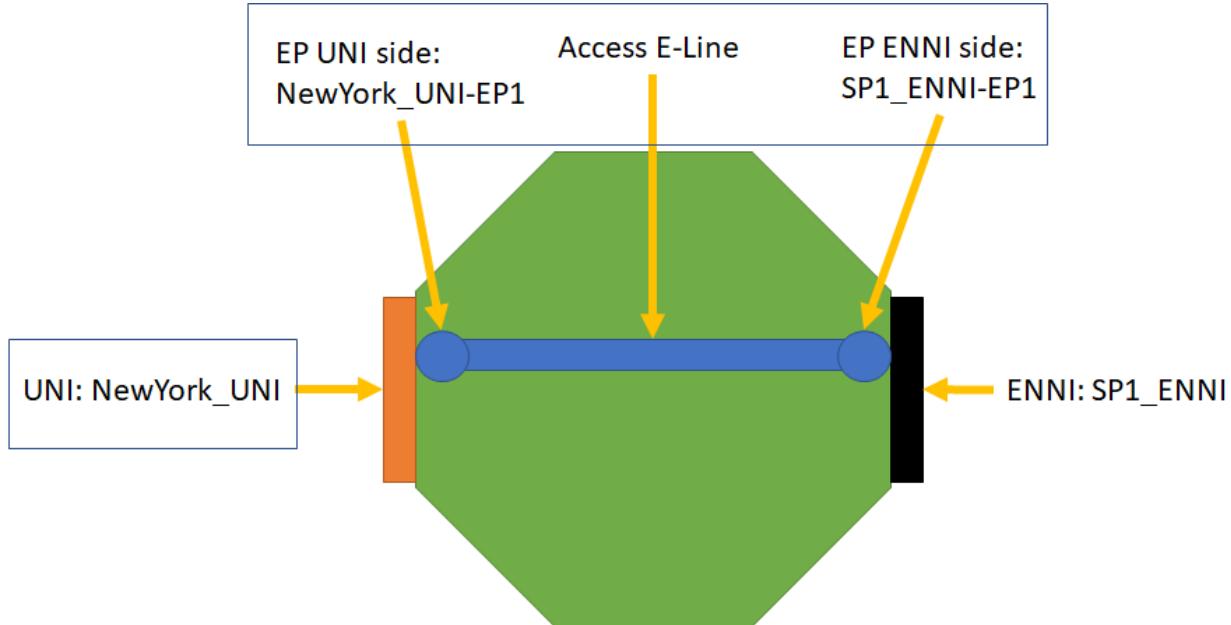
-
- 1046 The sequence diagram below (Figure 14) shows a set of logical steps of building the POQ request:



1048

Figure 14 – UC2: POQ request building steps

1049 The products' integration with the API is covered. Let's go to some details of the products' configuration.
1050 The setup of the Use Case 2 is presented in Figure 15.



1051

1052

Figure 15 – UC2: Setup Diagram

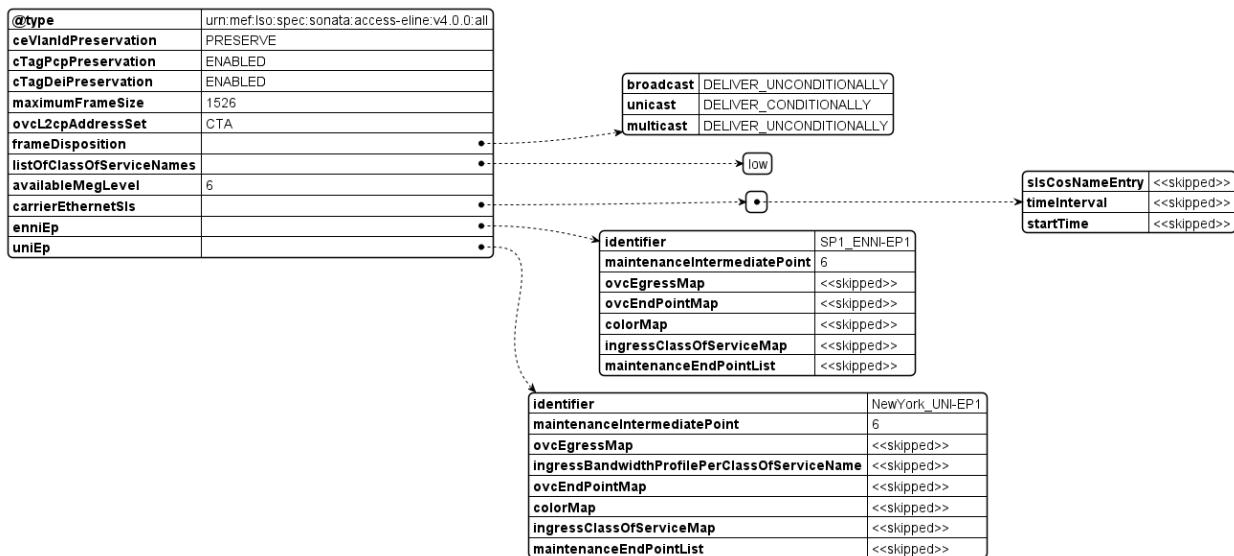
1053 This setup involves:

- 1054 • Creation of the UNI
 - 1055 ○ place: New York (Address id acquired in Use Case 1)
 - 1056 ○ id="NewYork_UNI"
- 1057 • Creation of the Access E-Line, including:
 - 1058 ○ configuration of a new UNI Endpoint with id="NewYork_UNI-EP1", at the UNI with
 - 1059 id="NewYork_UNI", which is also created within the same request.
 - 1060 ○ configuration of a new ENNI Endpoint with id="SP1_ENNI-EP1", at the already ex-
 - 1061 isting ENNI with id="SP1_ENNI"

1062 The diagram aggregates the scope of a particular product configuration into rectangles. This is to stress that
1063 the UNI and ENNI endpoints are parts of the Access E-Line configuration. They are not individual orderable
1064 products (this is the case in point-to-point connections).

1065 The instance diagram for the whole Access E-Line (AEL) configuration is too big to be presented as a
1066 whole so it is split and presented in parts. Figure 16 shows the basic AEL attributes. This diagram as
1067 attached to Figure 13 as the node with "@type=urn:ietf:sonata:access-eline:v4.0.0:all". The attrib-
1068 utes that are skipped on this level are marked with a "<>skipped<>" label and will be presented on the next
1069 diagrams.

1070



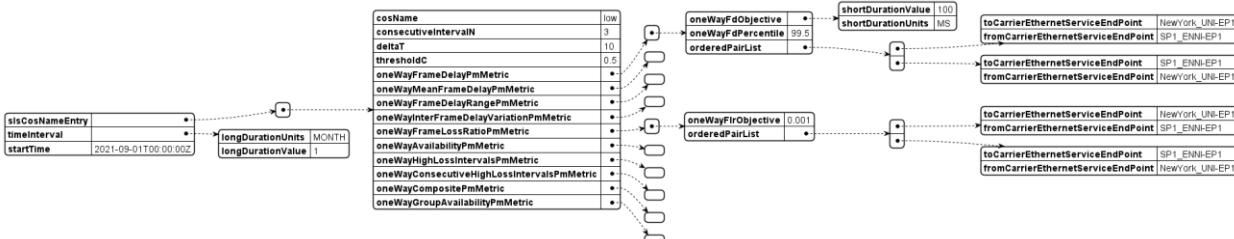
1071

1072

Figure 16 – UC2: Access E-Line, basic attributes

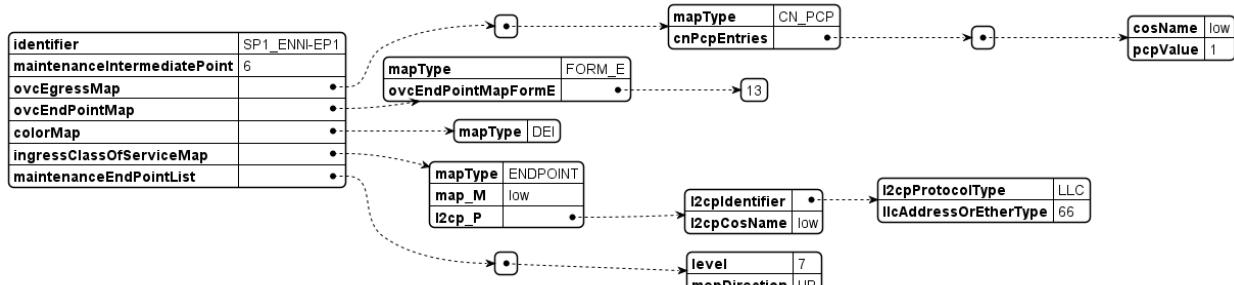
1073 The structures defining the “carrierEthernetSls”, “enniEp”, and the “uniEp” are complex and presented in
1074 the following figures:

1075



1076

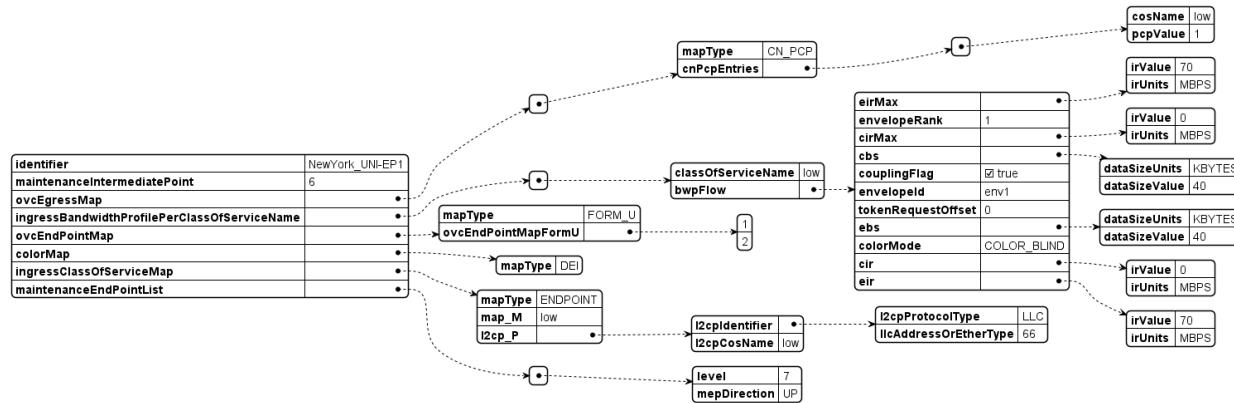
1077

Figure 17 – UC2: Access E-Line, Carrier Ethernet SLS


1078

1079

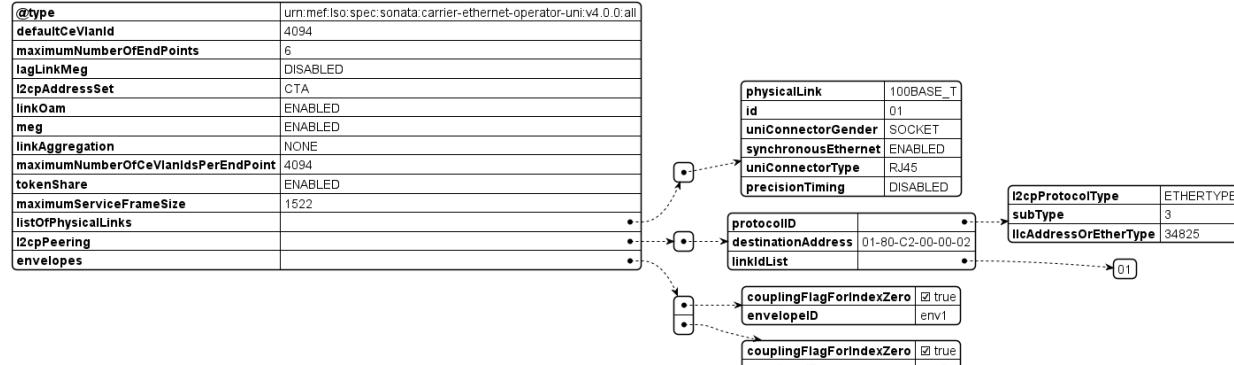
Figure 18 – UC2: Access E-Line, ENNI Endpoint



1080

Figure 19 – UC2: Access E-Line, UNI Endpoint

1082 The last figure in this use case presents the UNI product configuration. There are two envelopes configured
1083 in advance. The second will be used in Use Case 3 so that the UNI does not need to be updated.



1084

1085

Figure 20 – UC2: UNI

1086 POQ Request example:

```
1087 {
1088     "instantSyncQualification": true,
1089     "externalId": "BuyerPoq-00001",
1090     "provideAlternative": false,
1091     "projectId": "BuyerProjectX",
1092     "relatedContactInformation": [
1093         {
1094             "emailAddress": "john.example@buyer.example.com",
1095             "name": "John Example",
1096             "number": "12-345-6789",
1097             "numberExtension": "1234",
1098             "organization": "Buyer Co.",
1099             "role": "buyerContactInformation"
1100         }
1101     ],
1102     "productOfferingQualificationItem": [
1103         {
1104             "id": "item-001",
1105             "action": "add",
1106             "product": {
1107                 "productOffering": {
1108                     "id": "000073"
1109                 },
1110                 "productConfiguration": {
1111                     "@type": "urn:ietf:params:xml:ns:xsd:yang:base#anyType",
1112                     "ceVlanIdPreservation": "PRESERVE",
1113                     "cTagPcpPreservation": "ENABLED",
1114                     "cTagDeiPreservation": "ENABLED",
1115                     "maximumFrameSize": 1526,
1116                     "ovcL2cpAddressSet": "CTA",
1117                     "frameDisposition": {
1118                         "broadcast": "UNCONDITIONAL",
1119                         "unicast": "CONDITIONAL",
1120                         "multicast": "UNCONDITIONAL"
1121                     },
1122                     "listOfClassOfServiceNames": ["low"],
1123                     "availableMegLevel": "6",
1124                     "carrierEthernetSls": [
1125                         {
1126                             "slsCosNameEntry": [
1127                                 {
1128                                     "cosName": "low",
1129                                     "oneWayFrameDelayPmMetric": [
1130                                         {
1131                                             "oneWayFdObjective": {
```

```
1132         "shortDurationValue": 100,
1133         "shortDurationUnits": "MS"
1134     },
1135     "oneWayFdPercentile": 99.5,
1136     "orderedPairList": [
1137         {
1138             "toCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1",
1139             "fromCarrierEthernetServiceEndPoint": "SP1_ENNI-EP1"
1140         },
1141         {
1142             "toCarrierEthernetServiceEndPoint": "SP1_ENNI-EP1",
1143             "fromCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1"
1144         }
1145     ]
1146 },
1147 ],
1148 "oneWayFrameLossRatioPmMetric": [
1149     {
1150         "oneWayFlrObjective": 0.001,
1151         "orderedPairList": [
1152             {
1153                 "toCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1",
1154                 "fromCarrierEthernetServiceEndPoint": "SP1_ENNI-EP1"
1155             },
1156             {
1157                 "toCarrierEthernetServiceEndPoint": "SP1_ENNI-EP1",
1158                 "fromCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1"
1159             }
1160         ]
1161     }
1162 ],
1163 ],
1164 ],
1165 "timeInterval": {
1166     "longDurationUnits": "MONTH",
1167     "longDurationValue": 1
1168 },
1169     "startTime": "2021-09-01T00:00:00Z"
1170 }
1171 ],
1172 "enniEp": {
1173     "identifier": "SP1_ENNI-EP1",
1174     "maintenanceIntermediatePoint": "6",
1175     "ovcEgressMap": [
1176         {
```

```
1177         "mapType": "CN_PCP",
1178         "cnPcpEntries": [
1179             {
1180                 "cosName": "low",
1181                 "pcpValue": "1"
1182             }
1183         ]
1184     },
1185 ],
1186     "ovcEndPointMap": {
1187         "mapType": "FORM_E",
1188         "ovcEndPointMapFormE": [13]
1189     },
1190     "colorMap": {
1191         "mapType": "DEI"
1192     },
1193     "ingressClassOfServiceMap": {
1194         "mapType": "ENDPOINT",
1195         "map_M": "low",
1196         "l2cp_P": {}
1197     },
1198     "maintenanceEndPointList": [
1199         {
1200             "level": 7,
1201             "mepDirection": "UP"
1202         }
1203     ]
1204 },
1205     "uniEp": {
1206         "identifier": "NewYork_UNI-EP1",
1207         "maintenanceIntermediatePoint": "6",
1208         "ovcEgressMap": [
1209             {
1210                 "mapType": "CN_PCP",
1211                 "cnPcpEntries": [
1212                     {
1213                         "cosName": "low",
1214                         "pcpValue": "1"
1215                     }
1216                 ]
1217             }
1218         ],
1219         "ingressBandwidthProfilePerClassName": [
1220             {
1221                 "classOfServiceName": "low",
```

```
1222     "bwpFlow": {
1223         "eirMax": {
1224             "irValue": 70,
1225             "irUnits": "MBPS"
1226         },
1227         "envelopeRank": 1,
1228         "cirMax": {
1229             "irValue": 0,
1230             "irUnits": "MBPS"
1231         },
1232         "cbs": {
1233             "dataSizeUnits": "KBYTES",
1234             "dataSizeValue": 40
1235         },
1236         "couplingFlag": true,
1237         "envelopeId": "env1",
1238         "tokenRequestOffset": 0,
1239         "ebs": {
1240             "dataSizeUnits": "KBYTES",
1241             "dataSizeValue": 40
1242         },
1243         "colorMode": "COLOR_BLIND",
1244         "cir": {
1245             "irValue": 0,
1246             "irUnits": "MBPS"
1247         },
1248         "eir": {
1249             "irValue": 70,
1250             "irUnits": "MBPS"
1251         }
1252     }
1253 ],
1254 "ovcEndPointMap": {
1255     "mapType": "FORM_U",
1256     "ovcEndPointMapFormU": [1, 2]
1257 },
1258 "colorMap": {
1259     "mapType": "DEI"
1260 },
1261 "ingressClassOfServiceMap": {
1262     "mapType": "ENDPOINT",
1263     "map_M": "low",
1264     "l2cp_P": {
1265         "l2cpIdentifier": {
```

```
1267             "l2cpProtocolType": "LLC",
1268             "llcAddressOrEtherType": "66"
1269         },
1270         "l2cpCosName": "low"
1271     }
1272 },
1273 "maintenanceEndPointList": [
1274     {
1275         "level": 7,
1276         "mepDirection": "UP"
1277     }
1278 ],
1279 }
1280 },
1281 "productRelationship": [
1282     {
1283         "relationshipType": "CONNECTS_TO_ENNI",
1284         "id": "SP1_ENNI"
1285     }
1286 ],
1287 },
1288 "qualificationItemRelationship": [
1289     {
1290         "relationshipType": "CONNECTS_TO_UNI",
1291         "id": "item-002"
1292     }
1293 ],
1294 },
1295 {
1296     "id": "item-002",
1297     "action": "add",
1298     "product": {
1299         "productOffering": {
1300             "id": "000074"
1301         },
1302         "place": [
1303             {
1304                 "@type": "GeographicAddressRef",
1305                 "id": "NewYorkAddress-id-1",
1306                 "role": "INSTALL_LOCATION"
1307             }
1308         ],
1309         "productConfiguration": {
1310             "@type": "urn:ietf:params:xml:ns:xsd:yang:empty"
1311             "uni:v4.0.0:all",
```

```
1312     "defaultCeVlanId": 4094,
1313     "maximumNumberOfEndPoints": 6,
1314     "lagLinkMeg": "DISABLED",
1315     "l2cpAddressSet": "CTA",
1316     "linkOam": "ENABLED",
1317     "meg": "ENABLED",
1318     "linkAggregation": "NONE",
1319     "maximumNumberOfCeVlanIdsPerEndPoint": 4094,
1320     "tokenShare": "ENABLED",
1321     "maximumServiceFrameSize": 1522,
1322     "listOfPhysicalLinks": [
1323         {
1324             "physicalLink": "100BASE_T",
1325             "id": "01",
1326             "uniConnectorGender": "SOCKET",
1327             "synchronousEthernet": "ENABLED",
1328             "uniConnectorType": "SC",
1329             "precisionTiming": "DISABLED"
1330         }
1331     ],
1332     "l2cpPeering": [
1333         {
1334             "protocolID": {
1335                 "l2cpProtocolType": "ETHERTYPE",
1336                 "subType": 3,
1337                 "llcAddressOrEtherType": 34825
1338             },
1339             "destinationAddress": "01-80-C2-00-00-02",
1340             "linkIdList": [
1341                 "01"
1342             ]
1343         }
1344     ],
1345     "envelopes": [
1346         {
1347             "couplingFlagForIndexZero": true,
1348             "envelopeID": "env1"
1349         },
1350         {
1351             "couplingFlagForIndexZero": true,
1352             "envelopeID": "env2"
1353         }
1354     ]
1355 }
1356 }
```

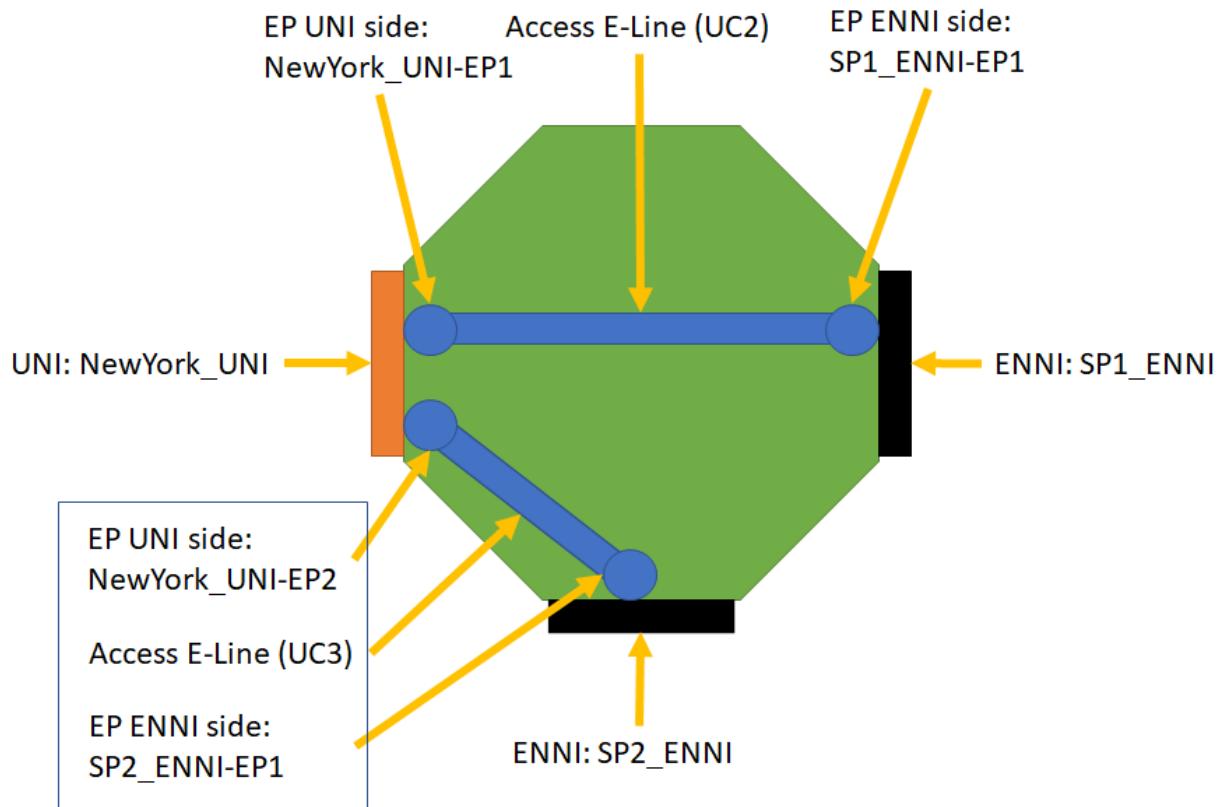
1357 }
1358]
1359 }

1360 A.3.3 Use Case 3: POQ - new Access E-Line, existing UNI, low+high class of service

1361 This use case presents the same step (POQ) as Use Case 2 but with two differences:

- 1362 • an existing UNI is being used, and
1363 • a more complex configuration of an Access E-Line is presented.

1364 The setup of the Use Case 3 is presented in Figure 21:



1366 **Figure 21 – UC3: Setup diagram**

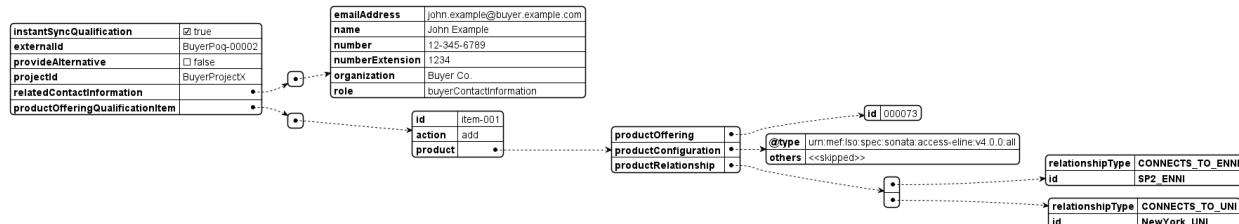
1367

1368 This setup involves:

- 1369 • Creation of the new Access E-Line, including:
1370 ○ configuration of a new UNI Endpoint with id="NewYork_UNI-EP2", at the already existing UNI with id="NewYork_UNI", the one that was created in Use Case 2 (assuming it was successfully ordered)

- 1373 ○ configuration of a new ENNI Endpoint with id="SP2_ENNI-EP1", at the already existing
 1374 ENNI with id="SP2_ENNI" (other than in use case 1)

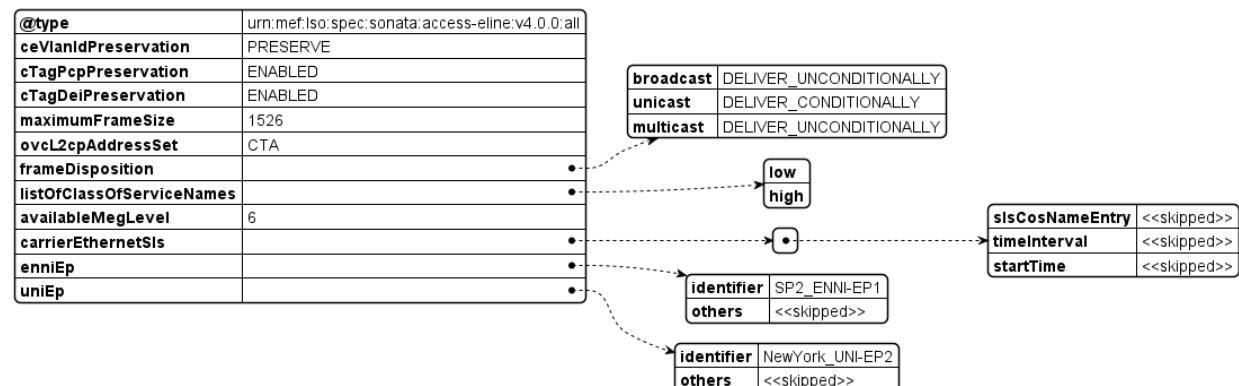
1375 This results in POQ Request having only one POQ Item, as now both ENNI and UNI are existing and
 1376 referenced with the use of "productOfferingQualificationItem.product.productRelationship". This is pre-
 1377 sented on truncated POQ request example in Figure 22:



1378

1379 **Figure 22 - UC3: Access E-Line relationships**

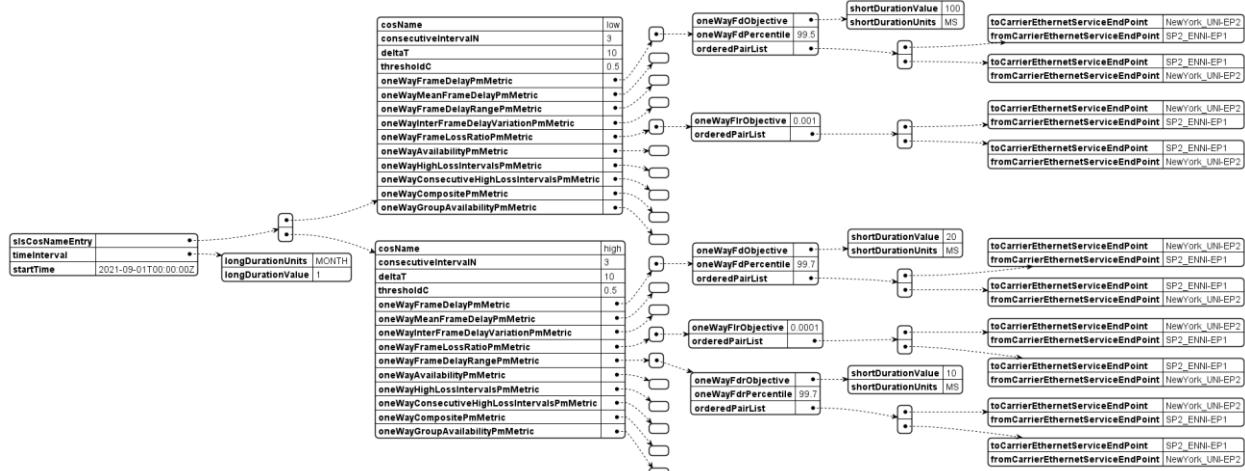
1380 The configuration of the UNI endpoint introduces a configuration with two classes of service: low and high.
 1381 The difference is first noticed at the root Access E-Line type configuration by having a list in the "lis-
 1382 tOfClassOfServiceNames" attribute:



1383

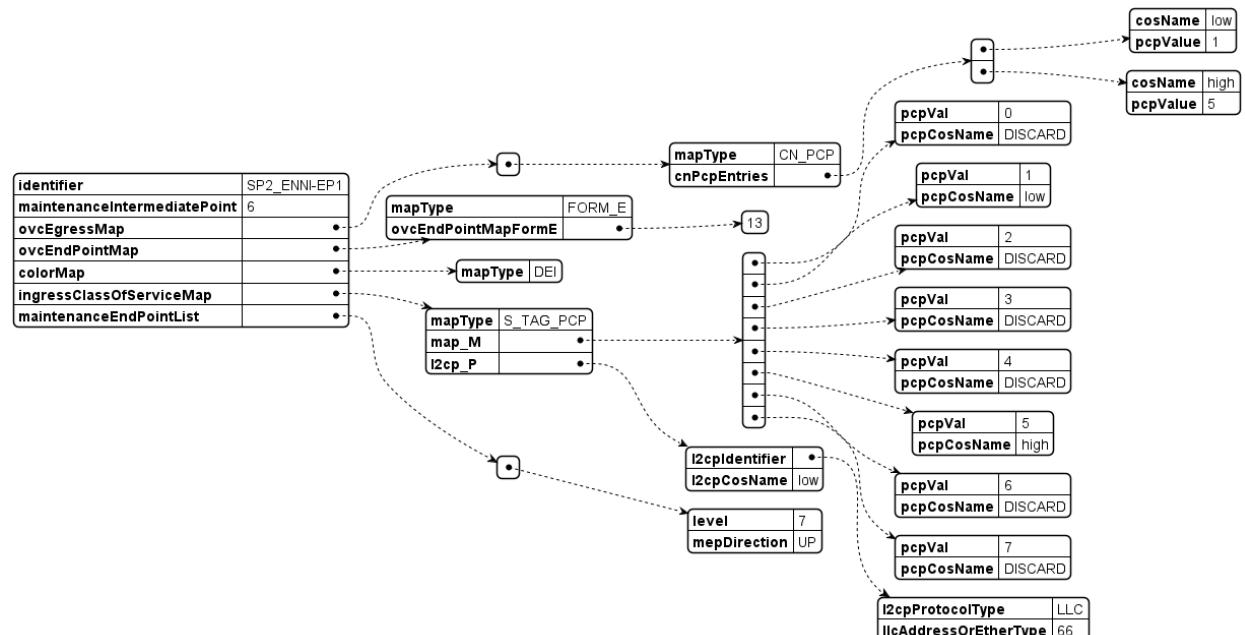
1384 **Figure 23 – UC3: Access E-Line, basic attributes**

1385 Next, the "carrierEthernetSIs" provides the Service Level Specification by defining the performance met-
 1386 rics per class of service. Notice the different metrics and values per low and high ones.



1387

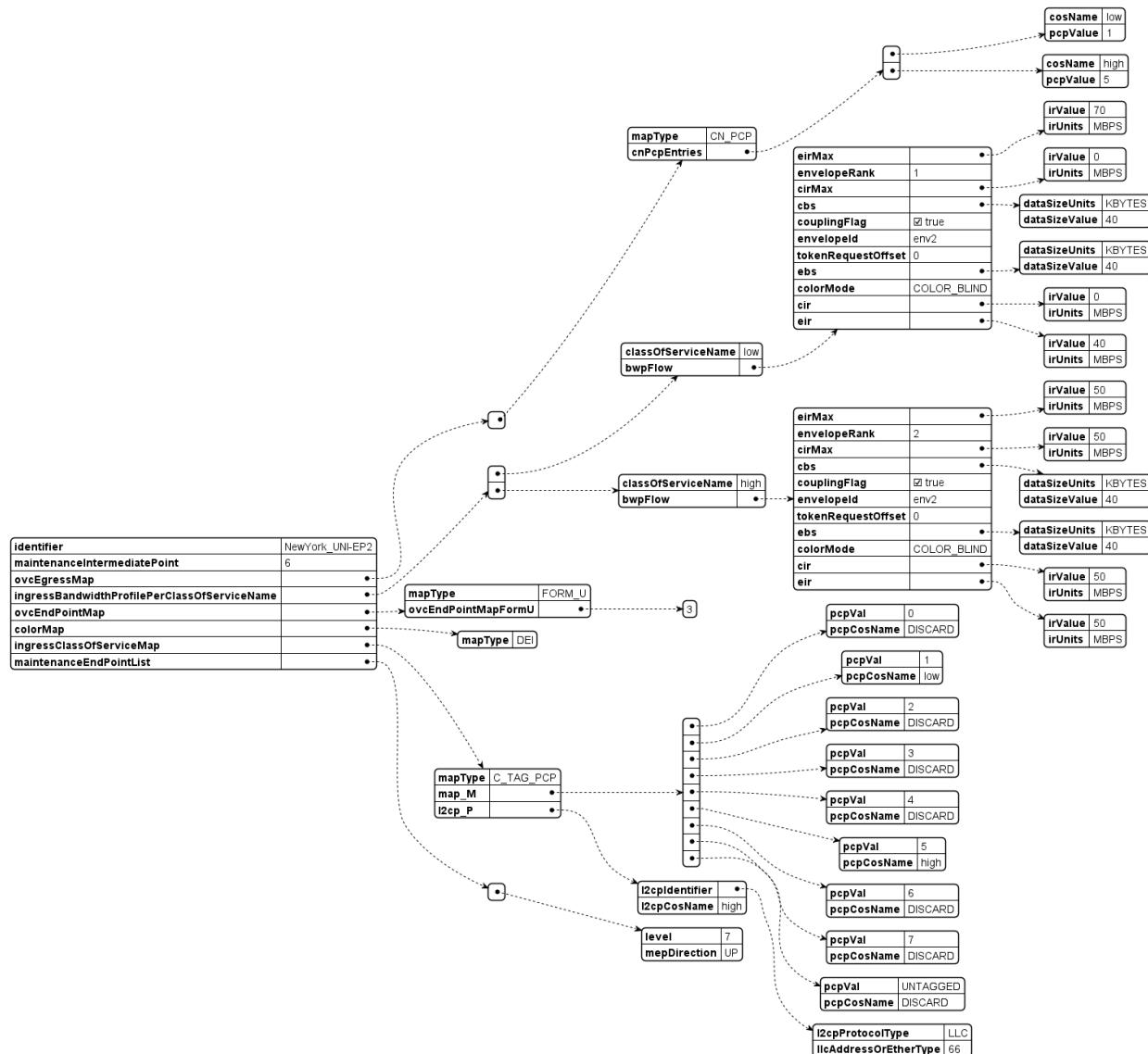
1388

Figure 24 – UC3: Access E-Line, Carrier Ethernet SLS


1389

1390

Figure 25 – UC3: Access E-Line, ENNI Endpoint



1391

1392

Figure 26 – UC3: Access E-Line, UNI Endpoint

1393 The bandwidth profiles defined for both classes of service share the same envelope “env2” which means
 1394 they share the same bandwidth “pipe”. The “high” profile defines the “cirMax” and “eirMax” on the same
 1395 level as corresponding “cir” and “eir” values, which is 50 MBPS. This means that the 50 MBPS bandwidth
 1396 is guaranteed for this profile. The “low” profile defines the “cir” and “cirMax” on the level of 0 MBPS
 1397 which means no bandwidth is guaranteed. The values of and the “eir” of 40 MBPS that a maximum of 40
 1398 MBPS can be used when free and, because “eirMax” is set to 70 MBPS, additional 30 MBPS from the
 1399 “high” CoS reservation can be used, when the traffic is not utilizing the full guaranteed 50 MBPS.

1400 Note that the names of the classes of service – “low” and “high” are used several times across the payload
 1401 and must match respectively in all of the places.

1402 The request example, as huge and repeating can be found in the attached postman collection.

1403 **A.3.4 Use Case 4: Quote**

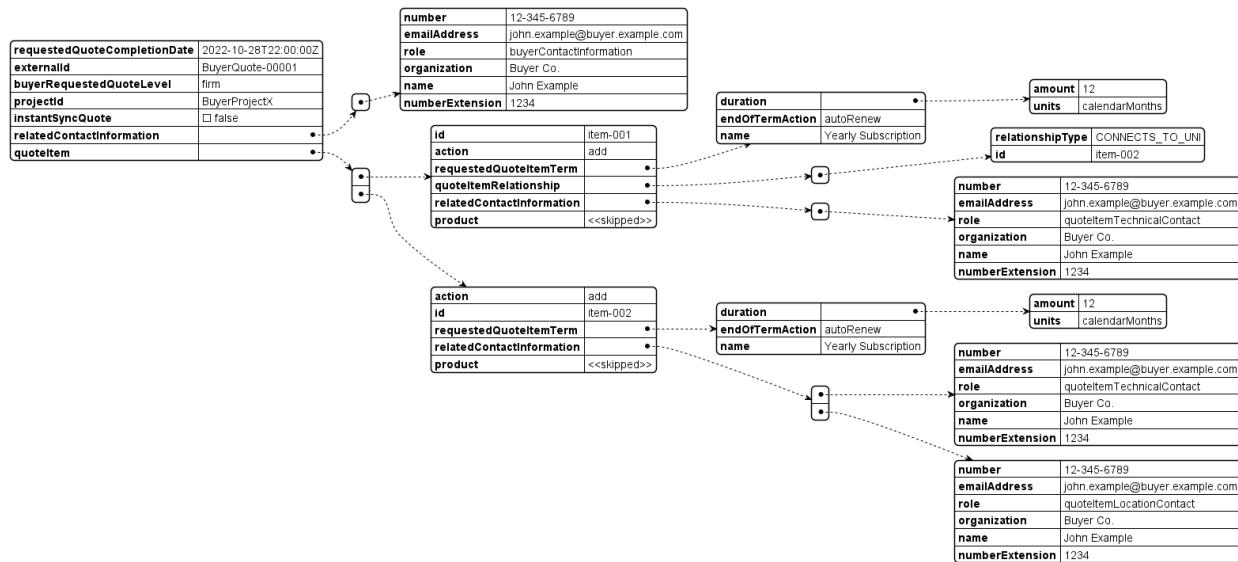
1404 For detailed guidance on how to use the Quote Management API, please refer to MEF 115 [18].

1405 The aim of the Quote step is to allow the Buyer to submit a request to find out how much the installation
 1406 of an instance of a Product Offering, an update to an existing Product, or a disconnect of an existing Product
 1407 will cost.

1408 This use case is the next step after use case 2. It asks for a quotation of the installation of the Access E-Line
 1409 and UNI products, with configuration as described in use case 2 (A.3.2).

1410 The Quote API carries product information exactly the same way as the POQ. The same steps in request
 1411 building (Figure 14) and rules of referencing existing products or ones in the same request, as described in
 1412 section A.3.2, apply.

1413 Figure 27 presents a diagram of a Quote request, with product information skipped.



1414

1415

Figure 27 - UC4: Quote Request

1416 The most important attributes to set in the quote request are:

1417 “instantSyncQuote” – to state the preference of receiving an instant (synchronous) response or a deferred
 1418 (asynchronous) one. In the latter case, the Seller only sends back an acknowledge response and proceeds
 1419 with the quotation. The Buyer may choose to register for notification or perform a periodical poll.

1420 “requestedQuoteCompletionDate” – If an instant response is not required this specifies the requested re-
 1421 sponse time.

1422 “buyerRequestedQuoteLevel” - 3 different types of quotes are managed:

- 1423 - **Budgetary:** A Quote that is provided quickly and with very little analysis such that the Buyer can
1424 get an idea of how much the requested Product Offering could cost. Any charges specified are
1425 subject to change.
- 1426 - **Firm - Subject to Feasibility Check:** A Quote that is provided to the Buyer based on some, but
1427 not a complete, pre-order analysis. At this stage, the Seller may not be willing to perform any
1428 further work on the Quote and requests that the Buyer use the Firm – Subject to Feasibility Check
1429 Quote to proceed to the Order process. Ordering is possible based on the Firm – Subject to Fea-
1430 sibility Check Quote with some stipulations as to how cost identified during delivery is addressed.
1431 The Monthly Recurring Charges specified in the Quote Response are final. Non-Recurring
1432 Charges specified in the Quote Response are subject to change and new Non-Recurring Charges
1433 may be identified during fulfillment.
- 1434 - **Firm:** A Quote provided to the Buyer based on complete pre-order analysis. All Monthly Recur-
1435 ring Charges and Non-Recurring Charges specified on a Firm Quote are committed. A Firm
1436 Quote may expire at some date specified by the Seller.

1437 “requestedQuoteItemTerm” – to specify the term (also known as commitment)

1438 In the response, the Seller confirms (most likely) the “quoteLevel”, “quoteItemTerm” and provides a price
1439 per each quote item. An example of price specification is shown below:

```
1440 "quoteItemPrice": [  
1441   {  
1442     "name": "Monthly Plan 25",  
1443     "priceType": "recurring",  
1444     "recurringChargePeriod": "month",  
1445     "price": {  
1446       "taxRate": 16,  
1447       "dutyFreeAmount": {  
1448         "unit": "EUR",  
1449         "value": 25,  
1450       },  
1451       "taxIncludedAmount": {  
1452         "unit": "EUR",  
1453         "value": 29,  
1454       },  
1455     },  
1456   },  
1457 ]
```

1458 Note: The Seller may require the Buyer to perform POQ prior to sending a Quote request.

1459 A.3.5 Use Case 5: Product Order

1460 Product Order allows the Buyer to request the Seller to initiate and complete the fulfillment process of an
1461 installation of a Product Offering, an update to an existing Product, or a disconnect of an existing Product
1462 at the address defined by the Buyer.

- 1463 This use case is the next step after use case 4. It places an order for the installation of the Access E-Line
 1464 and UNI products, which were qualified and quoted in use cases 2 and 4.
- 1465 The Order API carries product information exactly the same way as the POQ and Quote. The same steps in
 1466 request building (Figure 14) and rules of referencing existing products or ones in the same request, as de-
 1467 scribed in section A.3.2, apply.
- 1468 An example Product Order request can be found in the postman collection. Figure 28 presents it with prod-
 1469 uct information skipped for readability.

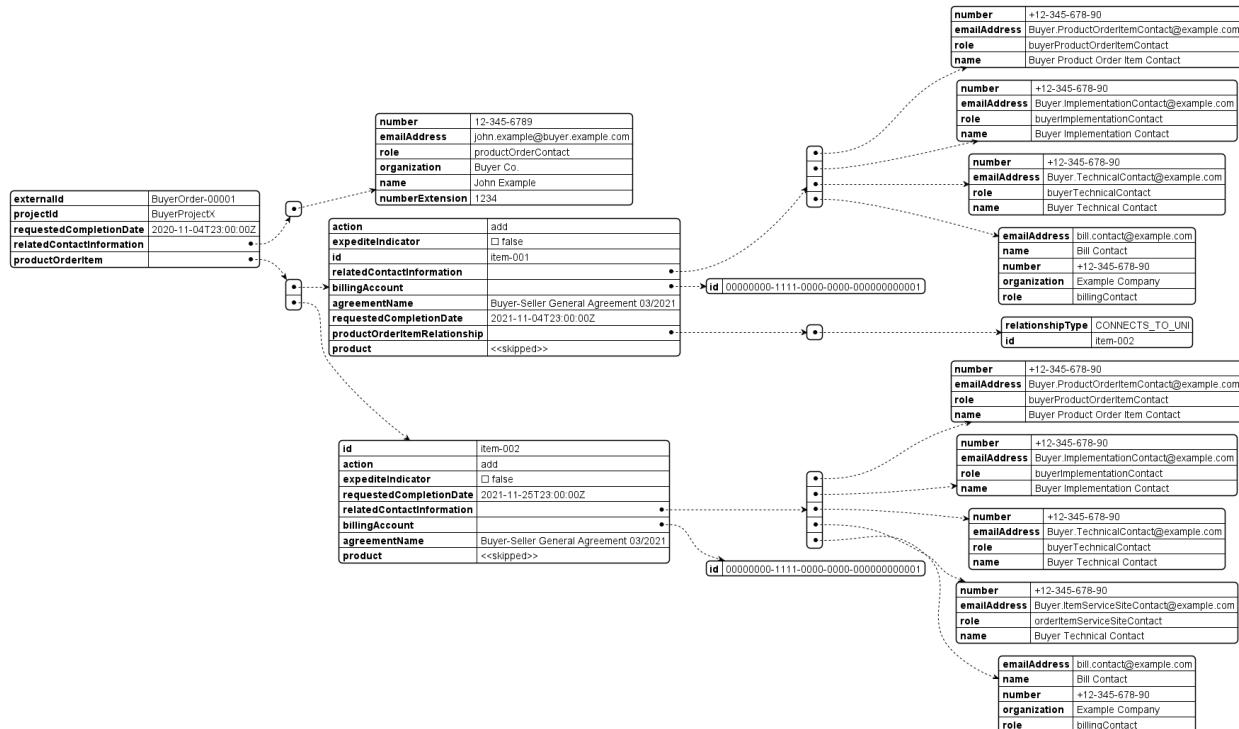


Figure 28 - UC5: Product Order request

- 1470 Again, there are a few attributes to be set by the Seller in the request like “requestedCompletionDate”,
 1471 “expediteIndicator” or “billingAccount” together with required contact information.
- 1472 The Seller responds with an acknowledge confirmation and then starts processing the order. The order
 1473 fulfillment process is longer than a simple request-response one of the previous steps (POQ, Quote) and the
 1474 state machine is more complex. The process may also be more interactive due to charge negotiation, possi-
 1475 ble request updates, etc.
- 1476 Product order API offers much more use cases like updating, expediting, or canceling an order request and
 1477 additional charge negotiation. For detailed guidance on how to use the Product Order Management API,
 1478 please refer to MEF 123 [20].

1481 **A.4 action: modify**

1482 The mechanism of building a modification request for both envelope and payload for all steps are practically
 1483 the same as for the create request.

1484 The differences are in the following common rules (POQ, Quote, Order):

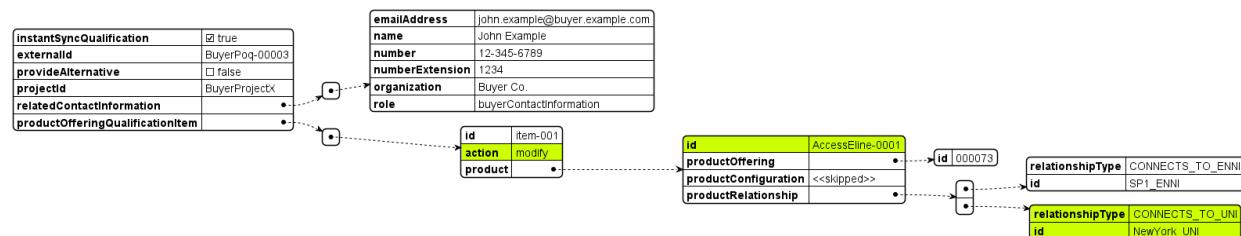
- 1485 - “item.action” must be set to “modify”
- 1486 - “item.product.id” of the product to be updated must be provided
- 1487 - “product.productConfiguration” must contain all desired configuration (not only the updated val-
 1488 ues)
- 1489 - “product.productOffering” must not be changed
- 1490 - The Access E-Line Product does not allow “product.productRelationship”, and “product.place” to
 1491 be changed.

1492 **A.4.1 Use Case 6: POQ: Bandwidth change**

1493 Use cases 6, 7, and 8 present POQ, Quote, and Order for an Access E-Line bandwidth change. The change
 1494 is made only for the attributes of the Access E-Line product, so requests contain only one item (UNI product
 1495 is not modified). The change is made by updating the “eir.irValue” and “eirMax.irValue” from 70 to 100.

1496 Note that since there is no accompanying item with the UNI, the relationship information “CON-
 1497 NECTS_TO_UNI” must be provided with the use of “product.productRelationship” attribute to point to the
 1498 existing UNI instance with “id”=”NewYork_UNI”

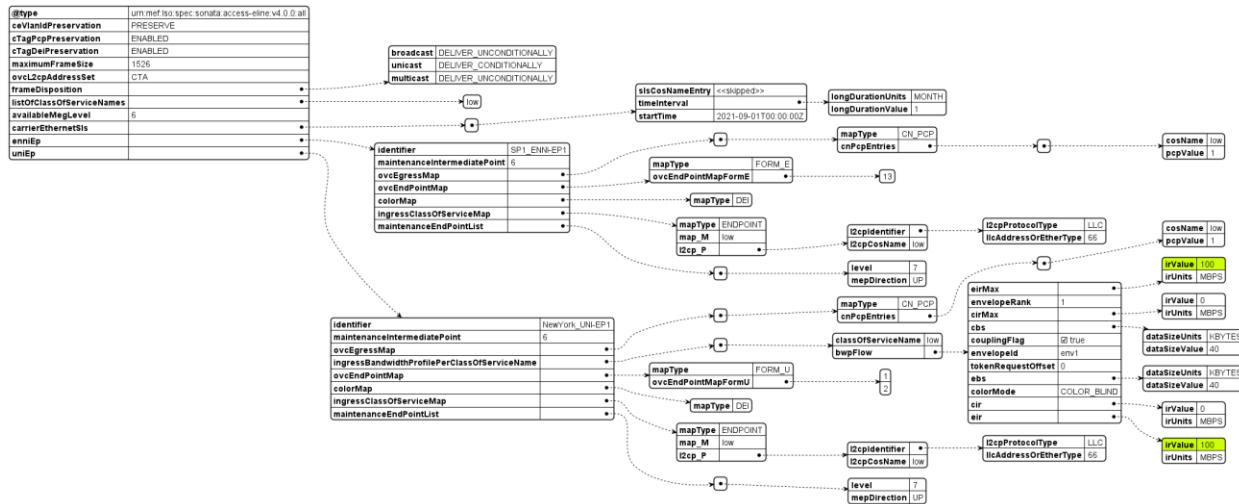
1499 The diagram below shows a POQ request for modification, highlighting the changes compared to the crea-
 1500 tion request.



1501

1502 **Figure 29 - UC6: POQ modify request**

1503 Figure 30 underlines the necessity of providing full product configuration for the modify action. The high-
 1504 lighted attributes are the only ones that are requested to be changed.



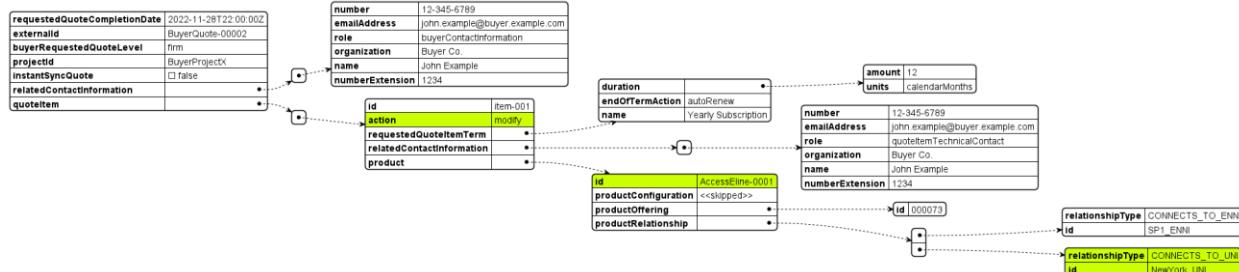
1505

1506

Figure 30 - UC6: Access E-Line modified attributes

1507 A full example request can be found in the attached postman collection.

1508 A.4.2 Use Case 7: Quote: Bandwidth change

1509 As the details of the product modification are already described in the previous chapter, this use case will
1510 only highlight the changes in the quote request, compared to the create request (Figure 31)

1511

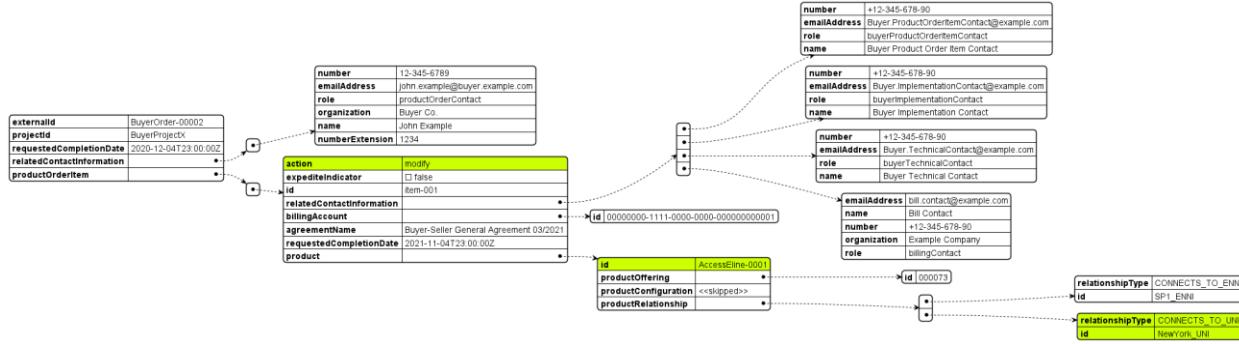
1512

Figure 31 - UC7: Quote modification request

1513 A full example request can be found in the attached postman collection.

1514 A.4.3 Use Case 8: Product Order: Bandwidth change

1515 All rules were described in the two above chapters. Figure 32 presents the Order request with highlighted
1516 changes:



1517

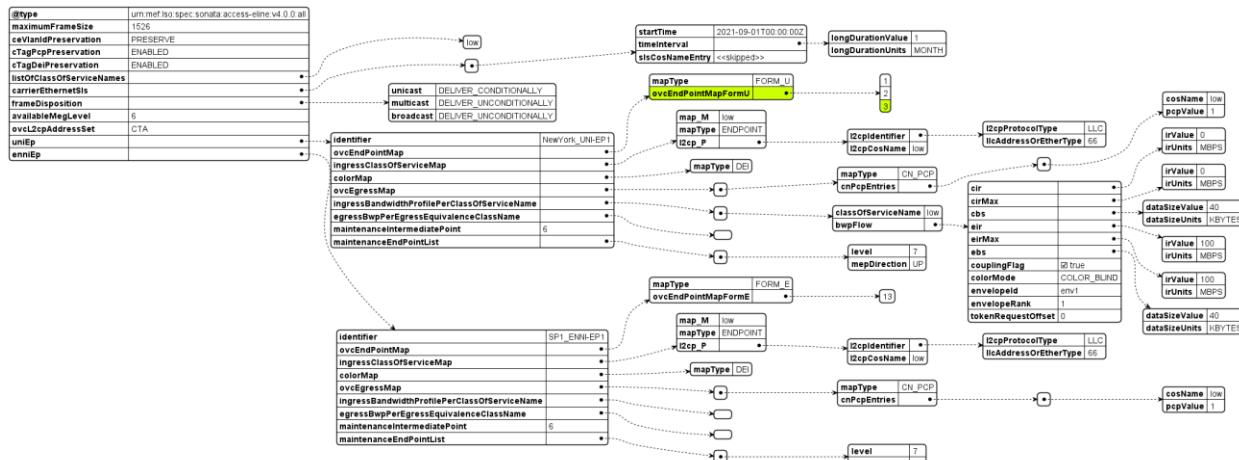
1518

Figure 32 - UC8: Order modification request

1519 A full example request can be found in the attached postman collection.

1520 A.4.4 Use Case 9: Product Order – VLAN change at the UNI

1521 In this case, an order to enhance the list of VLAN IDs are mapped to the UNI End Point is enhanced from
1522 [1, 2] to [1, 2, 3]. This is done with the order request as above, but with a slightly modified product payload,
1523 which is presented in Figure 33:



1524

1525

Figure 33 - UC9: Order modification request

1526 A full example request can be found in the attached postman collection.

1527 A.5 action: delete

1528 Delete requests are for all steps are very straightforward, as they only carry the product “id”.

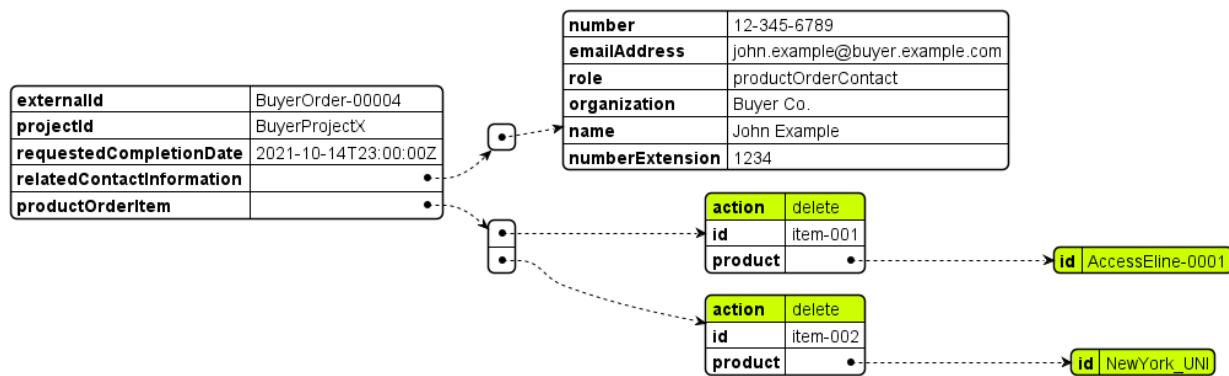
1529 Following common rules apply for disconnect operation:

- 1530 - “item.action” must be set to “delete”
- 1531 - “item.product.id” of the product to be deleted must be provided

- 1532 - “product.productConfiguration” must not be provided
 1533 - no other item attribute may be provided (except for optional “billingAccount” in Order)

1534 **A.5.1 Use Case 10: Product Order – delete both Access E-Line and UNI**

1535 Deletion of both Access E-Line and UNI products can be ordered with a request that is presented in Figure
 1536 34:



1537

1538 **Figure 34 - UC10: Order deletion request**

1539 Note: A disconnect request may result in additional charges (if not quoted earlier).

1540 JSON representation of this request:

```
1541 {
1542     "externalId": "BuyerOrder-00004",
1543     "projectId": "BuyerProjectX",
1544     "requestedCompletionDate": "2021-10-14T23:00:00Z",
1545     "relatedContactInformation": [
1546         {
1547             "number": "12-345-6789",
1548             "emailAddress": "john.example@buyer.example.com",
1549             "role": "productOrderContact",
1550             "organization": "Buyer Co.",
1551             "name": "John Example",
1552             "numberExtension": "1234"
1553         }
1554     ],
1555     "productOrderItem": [
1556         {
1557             "action": "delete",
1558             "id": "item-001",
1559             "product": {
1560                 "id": "AccessEline-0001"
1561             }
1562         },
1563         {
1564             "action": "delete",
1565             "id": "item-002",
1566             "product": {
1567                 "id": "NewYork_UNI"
1568             }
1569         }
1570     ]
1571 }
```

1572 A.5.2 Use Case 11: Move Access E-Line to a different Location

1573 The case of moving the office to another building cannot be realized by a single update of the “place”
1574 attribute of the UNI product. One of the points in Chapter 9 states:

- 1575 • *The location and physical layer of a UNI cannot be changed once it is ordered; instead, this
1576 is handled as an installation (UNI at the new location) and a disconnect (UNI at previous
1577 location), as there is often a requirement for a smooth transition with minimum downtime.*

1578 Nor it can be realized by updating Access E-Line’s product reference to another UNI. Chapter 11:

1579 *Changing the UNI and ENNI Reference or the UNI Location is not supported for an Access E-Line Ser-
1580 vice. The value included in a Change request must be identical to the value in the Inventory.*

1581 So the argument is both business and technical. In order to realize this use case, the following requests must
1582 be performed:

- 1583 1. Creation of new UNI at the new location
- 1584 2. Creation of a new Access E-Line
- 1585 3. Deletion of an old Access E-Line
- 1586 4. Deletion if an old UNI (optionally, if not used by other connections)

1587 Step 1 as potentially requiring physical installation should be performed earlier to prepare for a switchover.
1588 Steps 2 and 3 should be coordinated to assure minimal downtime.

1589 This use case as being built upon already described steps is not part of the attached postman collection.

1590 **A.6 Example Postman collection**

1591 All requests described in this appendix are provided in a form of a postman collection that can be used as a
1592 reference in implementation or for testing purposes.

1593



1594 MEF W106 - Appendix A.postman_collection.json