Working Draft
MEF W125 v1.05

LSO Cantata and LSO Sonata - Subscriber Ethernet Schema Guide

June 2021

This draft represents MEF work in progress and is subject to change.

This draft document represents MEF work in progress, has not achieved full MEF standardization and is subject to change. There are known unresolved issues that are likely to result in changes before this becomes a fully endorsed MEF Standard. The reader is strongly encouraged to review the Release Notes when making a decision on adoption. Additionally, because this document has not been adopted as a Final Specification in accordance with MEF's Bylaws, Members are not obligated to license patent claims that are essential to implementation of this document under MEF's Bylaws.

Disclaimer

© MEF Forum 2021. All Rights Reserved.

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

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

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

a) any express or implied license or right to or under any patent, copyright, trademark or trade secret rights held or claimed by any MEF member which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor

b) any warranty or representation that any MEF members will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor

c) any form of relationship between any MEF member and the recipient or user of this document.

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

Table of Contents

107

List of Figures

118

List of Tables

MEF W125
v1.05

© MEF Forum 2021. Any reproduction of this document, or any portion thereof, shall contain the follow-
ing statement: "Reproduced with permission of MEF Forum." No user of this document is authorized to
modify any of the information contained herein.

Page vi

1 List of Contributing Members

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

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

- ABC Networks
- XYZ Communications

139 ## 2 Abstract

140 The MEF Standard consisting of this Requirements document and its associated software artifacts
141 (JSON Schemas) defines and describes the product-specific information used in LSO Cantata and
142 LSO Sonata APIs for Product Offering Qualification, Product Quote, Product Order and Product
143 Inventory, for Subscriber Ethernet Services. The document starts with an overview of LSO Can-
144 tata, LSO Sonata and Subscriber Ethernet Services. It then provides a basic information model for
145 the MEF Subscriber Ethernet Service Attributes. The final sections describe the Data Model in
146 both a technology-independent way and a technology-specific way focused the JSON Schema as-
147 sociated with this specification.

148 This document can be thought of as a user's guide for the Subscriber Ethernet Data Model and the
149 schemas provided that embody the Data Model. MEF Services are described by a set of Service
150 Attributes. Each Service Attribute describes an aspect of the service that is agreed between the
151 provider and the user of the service. The document that describes the Service Attributes for Sub-
152 scriber Ethernet Services is MEF 10.4 [6]. The Services, themselves, are specified in MEF 6.3 [4]
153 based on the Service Attributes defined in MEF 10.4.

154 MEF 10.4 includes 11 EVC Service Attributes, 13 EVC End Point and 17 Subscriber UNI Service
155 Attributes to describe the various components that compose a Subscriber Ethernet Service. This
156 document defines a data model that includes 33 of these Service Attributes and how they are or-
157 ganized across the various schemas to support the different types of Subscriber Ethernet Services.

158

159 # 3   Terminology and Abbreviations

160 This section defines the terms used in this document. In many cases, the normative definitions to
161 terms are found in other documents. In these cases, the third column is used to provide the refer-
162 ence that is controlling, in other MEF or external documents. If the reference includes an asterisk
163 (*), the definition has been adapted from the original.

164

Term	Definition	Reference
Business Applications	The Service Provider functionality supporting Business Manage-ment Layer functionality (e.g., product catalog, order manage-ment, billing, relationship management, etc.)	MEF 55.1 [9]
BUS	See *Business Applications*	MEF 55.1 [9]
Business Functions	In the context of this document Business Functions refer to *Product Offering Qualification (POQ)*, *Order Management*, *Quote Management*, and *Inventory Management*.	This Document
Buyer	In the context of this document, a Buyer is the organization acting as the customer who is ordering from an Operator (aka, Seller).	MEF 57.1 [10]
Cantata	The Management Interface Reference Point supporting the man-agement and operations interactions (e.g., quote, order, billing, etc.) between a Service Provider (Seller) and a Customer Domain (Buyer).	MEF 55.1 [9]
Data Model	A representation of concepts of interest to an environment in a form that is dependent on data repository, data definition lan-guage, query language, implementation language, and/or proto-col (typically, but not necessarily, all five).	MEF 78.1 [11]
Ethernet Network Network Interface	A reference point representing the boundary between two Oper-ator Carrier Ethernet Networks that are operated as separate ad-ministrative domains.	MEF 26.2 [7]
ENNI	*See Ethernet Network Network Interface*	MEF 26.2 [7]
EPL	Ethernet Private Line	MEF 6.3 [4]
EP-LAN	Ethernet Private LAN	MEF 6.3 [4]
EP-Tree	Ethernet Private Tree	MEF 6.3 [4]
Ethernet Private Line	A Port-based Service of E-Line Service Type.	MEF 6.3 [4]
Ethernet Private LAN	A Port-based Service of E-LAN Service Type.	MEF 6.3 [4]
Ethernet Private Tree	A Port-based Service of E-Tree Service Type.	MEF 6.3 [4]
Ethernet Service Provider	An organization that provides to a Subscriber a connectivity ser-vice that carries Ethernet Frames irrespective of the underlying	MEF 10.4 [6]

Term	Definition	Reference
	technology and that is specified using Service Attributes as defined in a MEF Standard.	
Ethernet Subscriber	The end-user of an Ethernet Service	MEF 10.4 [6]
Ethernet UNI	The demarcation point between the responsibility of the Ethernet Service Provider and the Ethernet Service Subscriber.	MEF 10.4 [6]
Ethernet Virtual Connection	An association of EVC End Points	MEF 10.4 [6]
Ethernet Virtual Connection	An association of EVC End Points	MEF 10.4 [6]
Ethernet Virtual Private Line	A VLAN-based Service of E-Line Service Type.	MEF 6.3 [4]
Ethernet Virtual Private LAN	A VLAN-based Service of E-LAN Service Type.	MEF 6.3 [4]
Ethernet Virtual Private Tree	A VLAN-based Service of E-Tree Service Type.	MEF 6.3 [4]
EVC	See *Ethernet Virtual Connection*	MEF 10.4 [6]
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 [6]
EVPL	See *Ethernet Virtual Private Line*	MEF 6.3 [4]
EVP-LAN	See *Ethernet Virtual Private LAN*	MEF 6.3 [4]
EVP-Tree	See *Ethernet Virtual Private Tree*	MEF 6.3 [4]
External Interface	In the context of this document, a UNI.	MEF 4 [3]
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 [11]
Inventory	See *Product Inventory*	MEF 81 [14]
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 [7]
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 [7]
Order	See *Product Order*	This document
Port-based Service	Any service with EVC EP Map Service Attribute value equal to *All* for each EVC EP in the EVC.	MEF 6.3 [4]

Term	Definition	Reference
Product	One or more goods or services that is or may be sold to a Buyer by a Seller.	MEF 79 [12]
Product Offering	The commercial and technical details of a Product sold by a Seller. A Product Offering defines all of 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 [12]
Product Order	One or more Product Order Items formulated into a fulfilment request made by a Buyer to a Seller.	This document
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 [14]
Product Specification	A Product Specification defines the template or detailed description from which Product Offerings can be defined.	MEF 79 [12]
POQ	See *Product Offering Qualification*	MEF 79 [12]
Product Offering Qualification	One or more Product Offering Qualification Items formulated into a requirement made by a Buyer to a Seller.	MEF 79 [12]
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 [12]
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.	MEF 80 [13]
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.	MEF 80 [13]
Seller	In the context of this document, a Seller is the Operator who is providing the product to the Buyer.	MEF 57.1 [10]
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 [6]
Service Frame	An Ethernet Frame that is exchanged across a UNI between a Subscriber and a Service Provider.	MEF 10.4 [6]

Term	Definition	Reference
Service Provider	In the context of this document, a Service Provider is an Ethernet Service Provider.	This Document
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 (Seller) and Partner Domain (Buyer)).	MEF 55.1 [9]
Subscriber	In the context of this document, a Subscriber is an Ethernet Subscriber.	This Document
Subscriber UNI Service Attributes	The Service Attributes used to describe an agreement between a Subscriber and a Service Provider for a UNI associated with an EVC.	MEF 10.4 [6]
UNI	An Ethernet User Network Interface	MEF 10.4 [6]
VLAN-based Service	Any service with the EVC EP Map value equal to *UT/PT* or *List* for each EVC EP in the EVC	MEF 6.3 [4]

165   **Table 1 – Terminology and Abbreviations**

166

4   Introduction

LSO Cantata provides a programmatic interface for establishing an automated exchange of information between business applications (Product Quote, Product Order, etc.) between a Service Provider (Seller) and Customer Domain (Buyer). LSO Sonata provides a similar programmatic interface between two network providers, a Service Provider Domain (Seller) and Partner Domain (Buyer). These APIs are hierarchically structured. The outer-most structure includes information relating to the access method (e.g., REST), next is information relating to the function being requested (e.g., Product Quote, Product Order, etc.) and the inner-most structure contains information relating to the specific product, for example Ethernet Private Line (EPL).

A Subscriber Ethernet Service is an Ethernet Service that is provided by a Service Provider (Seller) to provide connectivity to Subscribers (Buyers), between two or more UNIs. The Service Attributes that are agreed to between the parties are defined in MEF 10.4 [6]. The Service definition which is, in effect, a set of constraints on the values of the Service Attributes is provided MEF 6.3 [4].

This specification is accompanied by a data model for the Subscriber Ethernet components instantiated as a set of JSON schemas that can be used within the Cantata/Sonata API to perform Product Offering Qualification, Product Quote, Product Order, and Product Inventory for Subscriber Ethernet Products consisting of:

- Subscriber Ethernet EVC and EVC End Point Service Attributes

- Subscriber Ethernet UNI Service Attributes

The document contains the following sections:

- An overview of LSO Cantata and LSO Sonata (section 5)

- An overview of Subscriber Ethernet Services (section 6)

- Data Model Design Principles (section 7)

- Example Milestones for the Business Functions (section 8)

- An abbreviated Information Model for Subscriber Ethernet and explanation of the organization of the Service Attributes in MEF 10.4 (section 9)

- Organization of the data model for Subscriber Ethernet (section 10)

- The relationship between the entities in the service (section 11)

196 These sections are followed by three sections that contain tables that describe the details of the
197 data model. These includes the JSON Name, description, data type, details about allowed values,
198 and, in some cases, some additional information about relationships between Service Attributes.

199 • Section 12 contains the details of the Subscriber Ethernet Service Attributes for EVC,
200 EVC End Point, and UNI

201 • Section 13 contains all of the common classes and types referenced by the Service
202 Attributes

203 • Section 14 lists the Service Attributes that are not included in the data models

204 • Section 15 describes the Subscriber Ethernet Envelope and Payload association and
205 provides sequence diagrams and sample code snippet as examples

206

5   Overview of LSO Cantata and LSO Sonata

MEF 55.1 [9] 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 interconnection points between different entities—either within the Service Provider domain (intra-domain) or between Service Provider and other business entities (inter-domain). One of these LSO Reference Points is LSO Sonata which defines the abstract boundary point between a Service Provider and an Operator which can provide wholesale (offnet) connectivity services to the Service Provider. It is at this boundary point – LSO Sonata – that the Service Provider and the Operator access each other's Business Functionalities. Examples of inter-provider Business Functionalities include address qualification, product offering qualification, quote, ordering, trouble ticketing, and billing/settlement management.
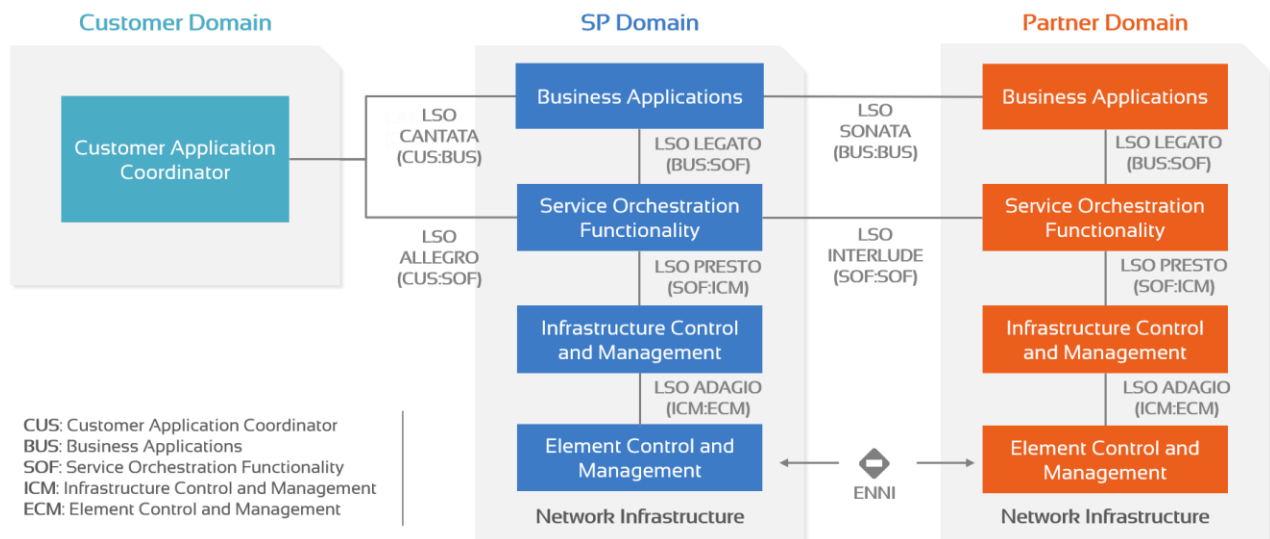
Figure 1 – LSO Cantata and LSO Sonata Reference Diagram

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

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

226
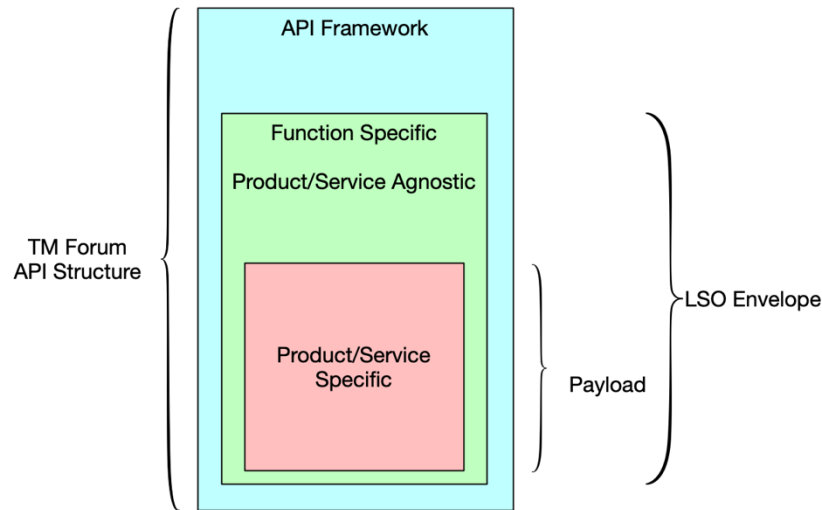
Figure 2 – LSO Cantata and LSO Sonata API Structure

228 This document describes the product specific data model for a MEF Subscriber Ethernet Services
229 as defined in MEF 6.3, along with along with examples showing how to map the product agnostic
230 use cases (e.g., Order, POQ, Quote, Inventory) to the product specific schemas.

231

6   Overview of Subscriber Ethernet Services

This specification describes a data model for MEF-defined Subscriber Ethernet Services. A Subscriber Ethernet Service is an Ethernet Service that is provided by a Service Provider to a Subscriber between two or more UNIs. A UNI is used to interconnect a Subscriber Network to the Service Provider Network and instantiate the Ethernet Services covered in the specification. An Ethernet Service consists of the EVC (Ethernet Virtual Connection) and the corresponding EVC End Points, each located at a UNI, that are defined using the Service Attribute values agreed to between the Service Provider and the Subscriber.

The following sub-sections provide background on EVCs and the different types of Subscriber Ethernet EVC Services (EPL, EVPL, EP-LAN, EVP-LAN, EP-TREE, EVP-TREE). It starts with a brief explanation of EVCs.

6.1   EVCs

A Subscriber Ethernet Service is built on an Ethernet Virtual Connection (EVC) which is an association of two or more EVC End Points located at Subscriber Ethernet UNIs. An EVC End Point is a construct at a UNI that represents the logical attachment of an EVC to a UNI and determines the subset of Service Frames that pass over the UNI. A given UNI can support more than one EVC End Point, but each EVC End Point has to be in a different EVC.

EVCs and their Service Attributes are defined in MEF 10.4 [6]. There are three types of EVC Services based on the type of connectivity provided: point-to-point, multipoint-to-multipoint, and rooted-multipoint. MEF 6.3 [4] defines these three EVC Service Type constructs: E-Line (point-to-point), E-LAN (multipoint-to-multipoint), and E-Tree (rooted-multipoint). In addition, both a Port-based service and VLAN-based service is defined for each of the three Subscriber Ethernet Service Types. The VLAN-based services support Subscribers that desire one type of service while accessing other services from one or more UNIs at the same time.

In Port-based service, the C-Tag VLAN ID is not used in the mapping of Service Frames to the EVC End Point, whereas in a VLAN-based service, the presence and/or value of the C-Tag VLAN ID is used in the mapping of Service Frames to the EVC End Point. The six types of EVC services are shown in Table 2.

261

EVC Service Type	Port-Based EVC Services	VLAN-Based EVC Services
Point-to-Point (E-Line)	EPL (Ethernet Private Line)	EVPL (Ethernet Virtual Private Line)
Multipoint-to-Multipoint (E-LAN)	EP-LAN (Ethernet Private LAN)	EVP-LAN (Ethernet Virtual Private LAN)
Rooted-Multipoint (E-Tree)	EP-Tree (Ethernet Private Tree)	EVP-Tree (Ethernet Virtual Private Tree)

262 **Table 2 – Subscriber Ethernet EVC Service Types**

263

264 EVCs are sold to a Subscriber by an Ethernet Service Provider. In most cases, the Ethernet Service
265 Provider is also an Operator that provides some or all of the network resources for the EVC. If all
266 of the UNIs in an EVC are accessible on the Service Provider's network, then the Service Provider
267 can deliver the EVC without having to coordinate with other Operators.

268 **6.2   EPL, EVPL, EP-LAN, EVP-LAN, EP-TREE, EVP-TREE**

269 The key differentiator between the three types of EVC services is the value of the EVC Type
270 Service Attribute which determines the type of connectivity provided to the Subscriber.

271 Any Ethernet Service that has the EVC Type Service Attribute equal to ***Point-to-Point*** is classified
272 as an E-Line Service Type (EPL, EVPL). E-Line services are intended to be highly transparent, in
273 the sense that Service Frames received at the ingress UNI are delivered at the egress UNI with as
274 few modifications as possible. An E-Line service can only associate exactly two Root EVC End
275 Points, as shown in Figure 3.

276

277
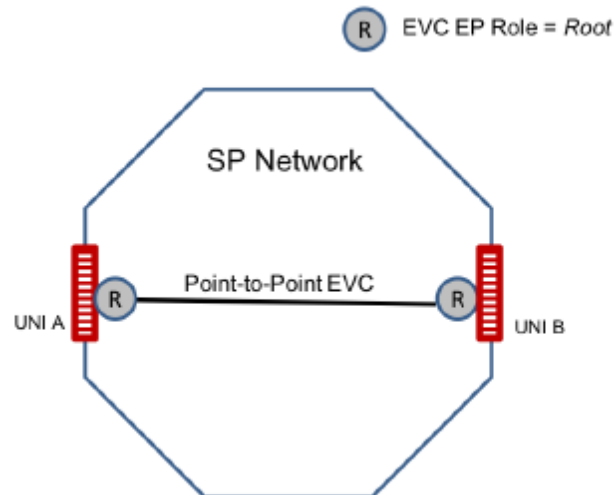
Figure 3 – E-Line Service Type

279 Any Ethernet Service that has the EVC Type Service Attribute equal to ***Multipoint-to-Multipoint***
280 is classified as an E-LAN Service Type (EP-LAN, EVP-LAN). E-LAN services enable Subscrib-
281 ers with multiple sites to interconnect them so that all sites appear to be on the same Local Area
282 Network (LAN) and have the same performance and access to resources such as servers and stor-
283 age. An E-LAN service can associate more than two EVC End Points and can only associate Root
284 EVC End Points, as shown in Figure 4.

285
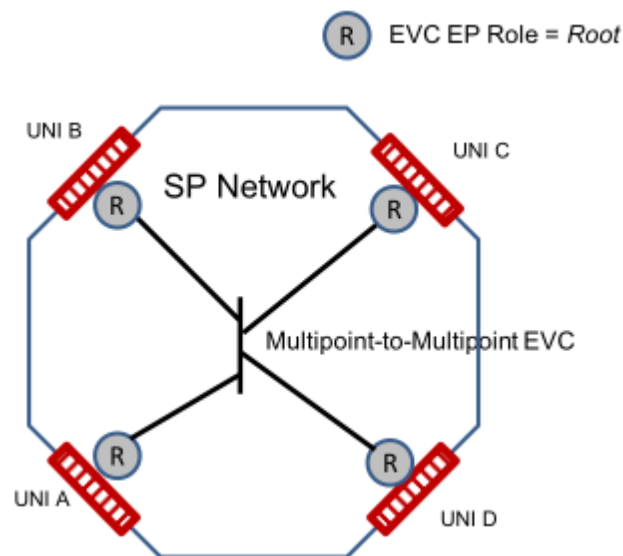
Figure 4 – E-LAN Service Type

287 Any Ethernet Service that has the EVC Type Service Attribute equal to ***Rooted-Multipoint*** is clas-
288 sified as an E-Tree Service Type (EP-TREE, EVP-TREE). An E-Tree service is a highly transpar-
289 ent service that enables Subscribers with multiple sites to interconnect them in a way that does not

290   resemble a LAN. The sites can be connected with a centralized site (or a few core sites), which are
291   designated as a Root and all remaining sites designated as Leaf.

292   An E-Tree service can associate more than two EVC EPs, at least one of which must be a Root
293   EVC End Point and any number of which are Leaf EVC End Points. A restriction on a Leaf EVC
294   End Point is that ingress Service Frames can only be delivered to Root EVC End Points. There is
295   no such restriction on a Root EVC End Point. An E-Tree service with a single Root is illustrated
296   in Figure 5.
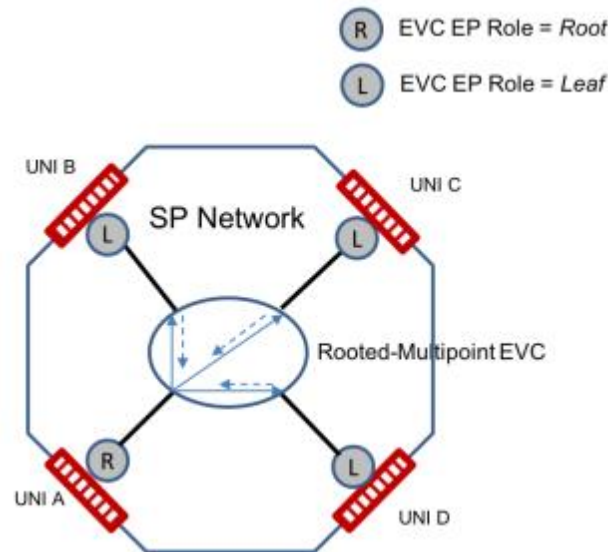
297
298   **Figure 5 – E-Tree Service Type**

299   The LSO Cantata and LSO Sonata API with the Subscriber Ethernet product configuration as de-
300   scribed in this specification allows the Business Functions associated with purchasing these prod-
301   ucts (Product Offering Qualification, Product Quote, Product Order, and Product Inventory) to be
302   performed programmatically.

7 Data Model Design Principles and Assumptions

The design for the Subscriber Ethernet data model is based on a number of assumptions:

- The requirements for which attributes are necessary at Product Offering Qualification(POQ), Product Quote and Product Order time differ (each should be a subset, mathematically speaking, of the successive steps). If the same set of attributes, again mathematically speaking, are needed across POQ, Quote and Order, then the number of use-cases goes down significantly.

- The requirements for which Service Attributes can be specified when changing a Subscriber Ethernet Product are different from when "installing" one (general principle).

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

These assumptions are shown in Table 3.

Business Function	Operation	UNI	EVC
POQ	Install	✔	✔
	Change	✔	✔
Quote	Install	✔	✔
	Change	✔	✔
Order	Install	✔	✔
	Change	✔	✔
Inventory	Retrieve	✔	✔

Table 3 – Allowed operations for each Business Function

320 # 8 Milestones

321 Although the various Business Functions are presented as if they represented a single instant in
322 time, each one consists of a sequence of steps or Milestones. The Service Provider (Seller) can
323 provide notifications to the Buyer on the status of each Business Function as Milestones are
324 achieved. Examples (for Order) are:

325

Milestone	Description	Comments
Order Item Billing Effective Date	Assumed to be Start Billing Date for New Install and Change / End Billing Date for Disconnect (Buyer/Seller to agree)	Seller would determine.
Order Item Way Leave Municipality	The Order Item is awaiting approval from an activity from a municipality	Seller
Order Item Way Leave Landlord	The Order Item has received approval for an activity from a landlord	Seller
Order Item Revised Firm Date Reason	Notes Field (Note to Order Item)	Seller provided if Expected Completion Date has changed.
Order Item Product ID	Product ID provided by Seller when Order Item State=IN_PROGRESS	

326 **Table 4 – Example Milestones for Order**

327 The full list of milestones for each function is provided in the relevant standard for the Business
328 Function.

329 # 9   Information Model for Subscriber Ethernet Product Data Model

330 Subscriber Ethernet Services are composed of three primary classes of objects: The Virtual Con-
331 nection (EVC), the UNI, and the EVC End Points.

332 The UNI is a component of two different business agreements, one between a Service Provider
333 and Subscriber for an EVC, and one between a Service Provider and an Operator for an OVC. Two
334 different sets of Service Attributes define these agreements, the Subscriber UNI Service Attributes
335 and the Operator UNI Service Attributes, respectively. This product specification provides a data
336 model for the Subscriber UNI Service Attributes. The data model for the Operator UNI Service
337 Attributes is provided as part of a product specification for Operator Ethernet OVC-based services
338 such as Access E-Line.

339 The function that logically connects the EVC or OVC to each External Interface is the Service End
340 Point (EVC End Point or OVC End Point). Although an abstract element, the End Point can be
341 thought of as the container for the EVC or OVC Service Attributes that can vary independently at
342 each External Interface. For example, the EVC Type (is the EVC point-point or multipoint?) is an
343 EVC Service Attribute and is invariant at all points for a given EVC. However, which VLAN(s)
344 to map to the Service or how much bandwidth to allow are attributes of the OVC that can be
345 different at each External Interface, and these Service Attributes are encapsulated in the EVC (or
346 OVC) End Points.

347 All of these elements are shown in the following diagram.

348
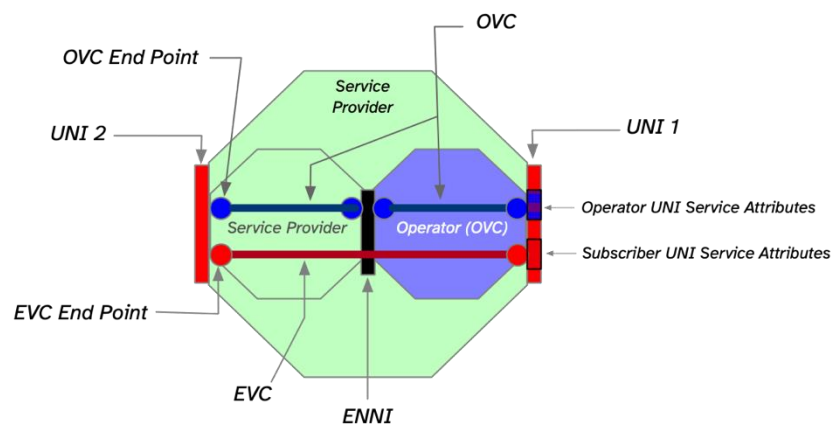
349   **Figure 6 – Components of an EVC**

350 In this diagram, the Ethernet Service Provider (large green octagon) is building the EVC on two
351 OVCs, one that it is providing itself to reach UNI 2 (small green octagon) and an Access E-Line
352 Service that it is buying from another Operator (blue octagon).

353

354 **9.1   Organization of Service Attributes**

355 A Subscriber Ethernet Service is defined as having an EVC and two or more UNIs. The EVC will
356 have an association with two or more EVC End Points. Each EVC End Point is associated with a
357 single Subscriber UNI (as shown in Figure 6). The Subscriber Ethernet configuration is based on
358 the Service Attributes documented in MEF 10.4. The MEF Services Model (MSM) for Subscriber
359 Ethernet Services described in MEF 7.4 defines three superclasses that contain all the service at-
360 tributes for the EVC, EVC End Point and Subscriber UNI. These superclasses (CarrierEther-
361 netService, CarrierEthernetExternalInterface and CarrierEthernetServiceEndPoint) are shown in
362 green in the following Figure 7 along with their sub-classes. The sub-classes relevant to Subscriber
363 Ethernet (CarrierEthernetEvc, CarrierEthernetSubscriberUni and CarrierEthernetEvcEndPoint)
364 are shown in the white boxes.
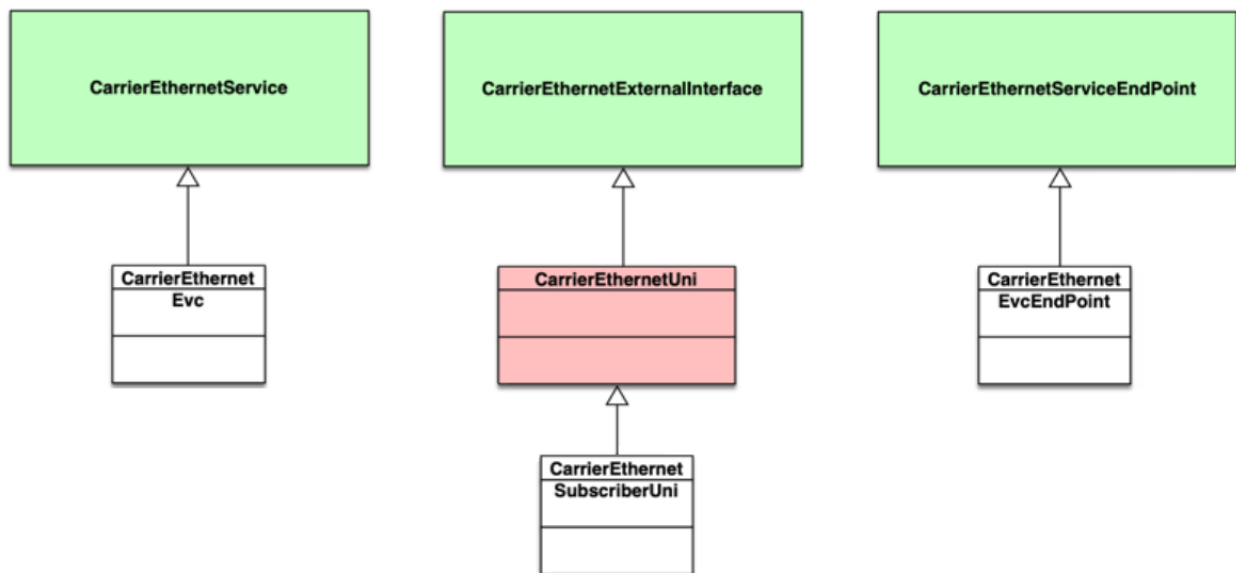
365
366 **Figure 7 – Subscriber Ethernet Superclasses**

367 In the context of an Access E-Line Service, and the Service Attributes in MEF 26.2, these compo-
368 nents are organized as follows (the colors in the following diagram are intended to match the colors
369 in the previous diagram):

370

371
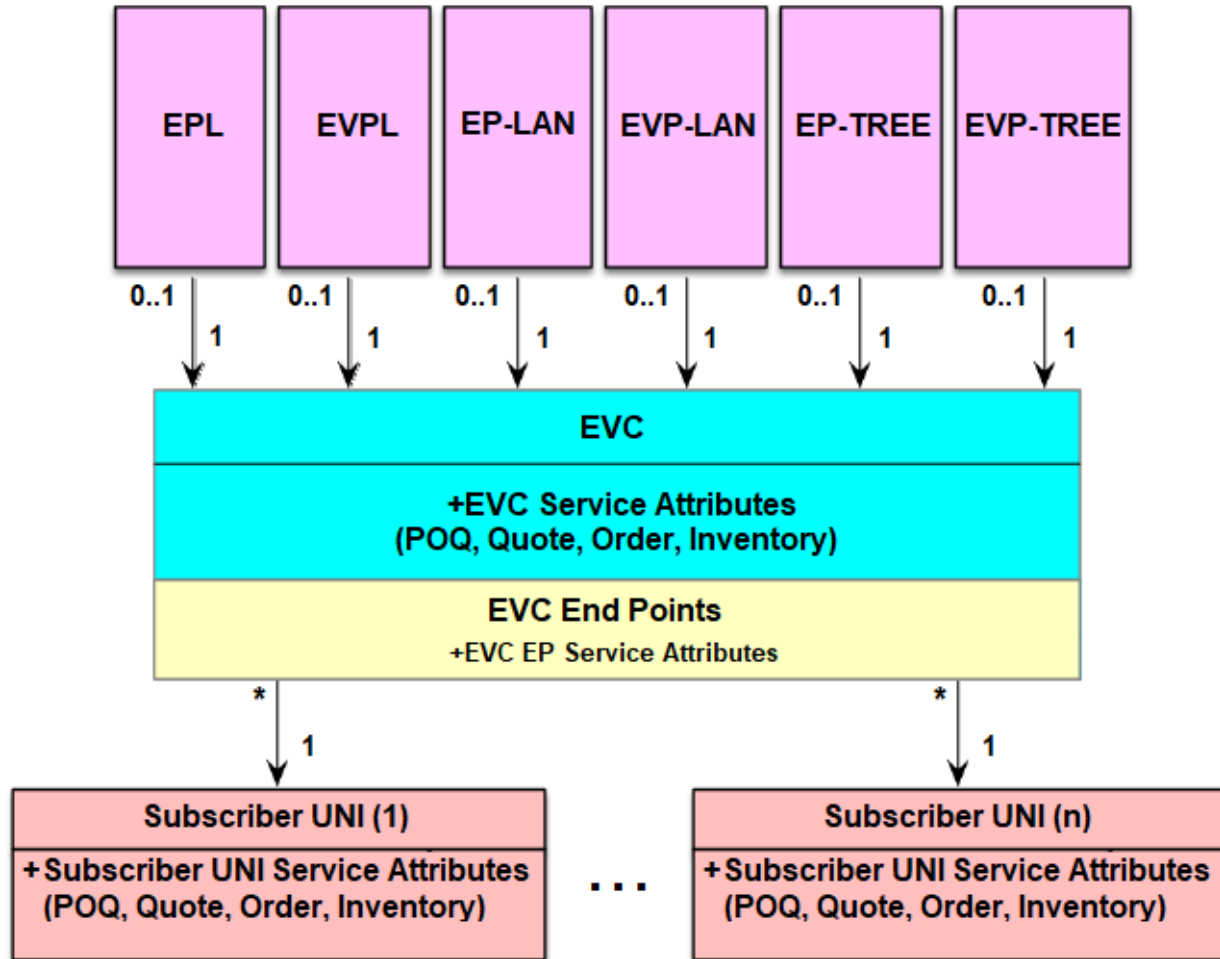
372 **Figure 8 – Subscriber Ethernet Information Model**

10 Data Models for Subscriber Ethernet Products

The data models for the Subscriber Ethernet product configuration in the LSO Cantata and LSO Sonata API are expressed as a set of JSON schemas that accompany this specification. This section explains the organization and structure of the schemas.

First, a brief technology neutral discussion of the data model is presented. This is followed by examples of a schema developed to represent the data model.

10.1 Technology Neutral Organization of the Data Models

This standard defines the following product specifications that can be used to manage Subscriber Ethernet services:

- Subscriber Ethernet EVC and EVC End Points

- UNI – This may be a new order or an existing UNI that is adding the new EVC

10.2 Organization and Structure of the Schemas

The schemas are organized into a file structure as shown in Figure 9.
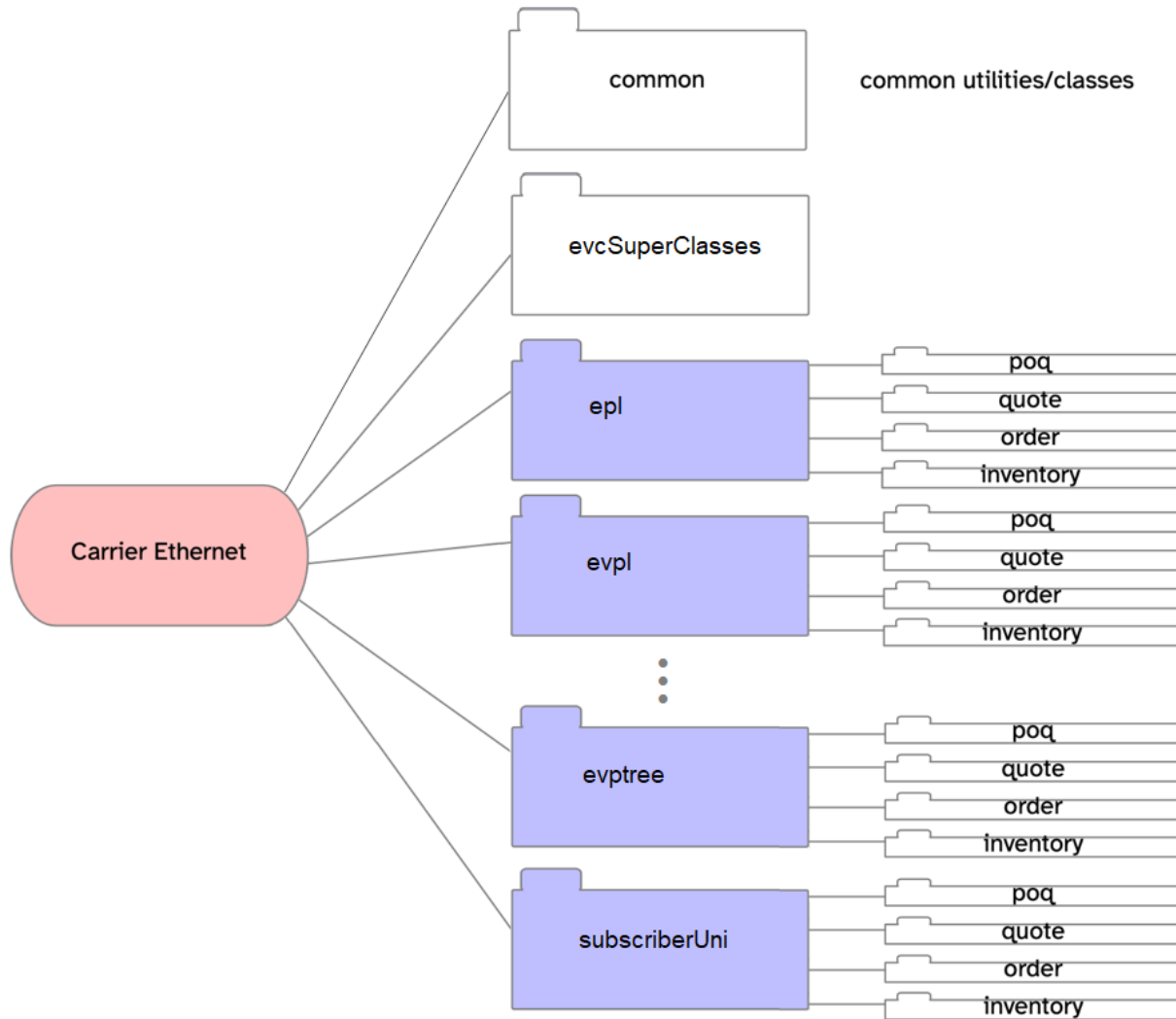
Figure 9 – Schema File Organization

For each of the major product components (EVC, EVC End Points and Subscriber UNI), there are *POQ, Quote, Order* and *Inventory* folders which contain the relevant schema files. In addition, there is a common folder that contains classes that are common across several Carrier Ethernet service components as well as a number of utility classes and types.

Note that the "Carrier Ethernet" prefix for a schema indicates that it is not specific to a particular service. For example, the carrierEthernetSubscriberUni.yaml schema is used with the Subscriber Ethernet service schemas, but it is also used with other Operator Ethernet Services such as Access E-Line or Transit E-LAN (when schemas for those services are developed).

The schema differences between Business Functions (e.g., between Order and Quote) fall into two categories:

398 • Inclusion: i.e., some Service Attributes may be included for some Business
399   Functions and have no relevance for other Business Functions and are therefore
400   not included in the data model for those Business Functions.

401 • Required/optional: Some of the properties for some Service Attributes and for
402   some referenced objects may be *required* for some Business Functions and *op-*
403   *tional* for others and this is reflected in the data model for validation purposes.

404 If a Service Attribute is *optional* for some Business Function, it may or may not be included in the
405 product configuration. If it is not included, the Service Provider will provide a default value if
406 appropriate for the Service Attribute. If the value of the Service Attribute is important to the Sub-
407 scriber, it should be included. If it is not included:

408 • The value is not important to the Subscriber, or

409 • The Subscriber knows the Service Provider's default value, or

410 • The controlling MEF Standard specifies a set of constraints on the value and the Sub-
411   scriber's requirements fall within those constraints.

412 An important thing to note is that there is no difference (other than required or optional) in the
413 definition of an attribute across Business Functions. If an attribute is defined as a 'list of 3 strings'
414 or a reference to *BandwidthProfileFlow* for one Business Function, it is defined that way for all
415 Business Functions (where the attribute is included).

416 **10.3   Example Schema**

417 Following is an example schema (from carrierEthernetLinkAggregation.yaml – chosen only be-
418 cause it is one of the shortest files):

```
419  "$schema": http://json-schema.org/draft-07/schema#
420  title:
421      MEF LSO Sonata - Carrier Ethernet Link Aggregation (Common) Product Specification
422      # MEF Forum - Date: 2021318  Edit-level: 192
423      # File: carrierEthernet/common/carrierEthernetLinkAggregation.yaml
424
425  definitions:
426      ConversationIdToAggregationLinkMap:
427          type: object
428          description:  >-
429              This is a 2-tuple ⟨x,y⟩ where x is a list of Port Conversation
430              IDs or ranges of Port Conversation IDs (a Port Conversation ID
431              is a VLAN ID or 0 for untagged frames) and y is a list of Link
432              Aggregation Links. This is used in the Port Conversation to Aggregation
433              Link Map for the UNI and ENNI.
434          properties:
435              conversationIDs:
436                  description:  >-
437                      802.1AX-2014 §6.6.2.1 - A Port Conversation ID is a VLAN ID
438                      (1 to 4094) or 0 to represent untagged and priority-tagged frames.
439                  type: array
440                  minItems: 1
```

```
441                     uniqueItems: TRUE
442                     items:
443                         type: object
444                         $ref: "#/definitions/ConversationIdRange"
445                 aggLinkList:
446                     description: 802.1AX-2014 §6.6.2.1 - An ordered list of Aggregation Link Numbers
447                     # See additional validation notes in the Schema Guide
448                     type: array
449                     minItems: 1
450                     uniqueItems: TRUE
451                     items:
452                         type: integer
453                         minimum: 1
454             required:
455                 - conversationIDs
456                 - aggLinkList
457         ConversationIdRange:
458             type: object
459             description:  >-
460                 A range of ConversationID (either a VLAN Id or 0 for untagged
461                 frames) allowing three forms: start, start → start, and start
462                 → end
463             properties:
464                 start:
465                     description:  >-
466                         The starting Conversation ID of the range or the only Conversation
467                         ID if there is no end value
468                     # start must be less than or equal to end
469                     # See additional validation notes in the Schema Guide
470                     type: integer
471                     minimum: 0
472                     maximum: 4094
473                 end:
474                     description: The final Conversation ID in the range
475                     # end, if specified, must be greater than or equal to start
476                     # See additional validation notes in the Schema Guide
477                     type: integer
478                     minimum: 0
479                     maximum: 4094
480             required:
481                         - start
```

This schema contains two classes related to Link Aggregation, *ConversationIdToAggregationLinkMap*, and a class that it uses, *ConversationIdRange. ConversationIdToAggregationLinkMap* is referenced by the External Interfaces (UNI) as follows (from carrierEthernetSubscriberUni.yaml) (for any of POQ, Quote, Order, Inventory):

```
486     aggregationLinkMap:
487             description:  >-
488                 MEF 26.2 §14.6 and 802.1AX-2014 § 6.6.2.1 - A mapping of a
489                 list of Port Conversation IDs (i.e., VLAN IDs or zero for untagged
490                 frames) to an ordered list of physical links if there are multiple
491                 physical links in the UNI and the link aggregation is ALL_ACTIVE.
492             # See additional validation notes in the Schema Guide
493             type: array
494             minItems: 0
495             uniqueItems: TRUE
496             items:
497                 type: object
498                 $ref:  "../../common/carrierEthernetLinkAggregation.yaml#/definitions/ConversationIdToAg-
499     gregationLinkMap"
```

500   ## 10.4   Additional Details

501   This section includes explanation of some additional conventions for the schema structure as well
502   as some additional attributes that have been added to facilitate product specification for some com-
503   mon edge cases.

504   ### 10.4.1   Naming Conventions

505   In the schemas, class and type names are UpperCamelCase and attribute/property names are low-
506   erCamelCase.

507   ### 10.4.2   EVC End Point Service Attributes

508   EVC End Points are not separately orderable items. They are part of the EVC. The EVC End Points
509   are the repositories for EVC Service Attributes that can be different at each UNI whereas the EVC
510   Service Attributes have the same value at every point in the EVC. The Subscriber Ethernet EVC
511   schema includes an array of properties at the end of the EVC Service Attributes, **EVC End Points**
512   which refers to the EVC End Points class (which is included in the same schema file).

513

11 Relationship Between Entities

This section describes the constraints and relationships between the three primary Order Items (EVC, Subscriber UNI, and EVC End Points).

The use case for Subscriber Ethernet described above is based on purchasing the Subscriber Ethernet EVC and a set of new or existing UNIs.

The Subscriber Ethernet Service is associated with exactly one EVC (with appropriate Service Attribute values), but it depends on two or more UNIs. This is noted in the following table. Specification of the UNIs are mandatory at Install of the service, and may not be part of a Change order.

The final column notes that during POQ and Quote, a UNI must be specified, but multiple UNIs might be candidates for use. Another way to think about this is that at POQ and Quote, the Subscriber Ethernet service is between *two UNI* and *possibly, several UNIs*.

	Product Relationship Role	Install	Change	Product Specification	Multiple Allowed at POQ and Quote?
Subscriber Ethernet	UNI_REFERENCE	Mandatory	Not Allowed	UNI	Yes

Table 5 – Product Relationship Roles

The UNI may be included in the same order as the Subscriber Ethernet Service. The UNI is associated with a specific Install Location and as noted above, it is required at Install and once a UNI is associated with the Service it cannot be changed. This is captured in the following table.

	Place Relationship Role	Install	Change
UNI	INSTALL_LOCATION	Mandatory	Not Allowed

Table 6 – Place Relationship Role

533 # 12 Subscriber Ethernet Service Attributes

534 The Service Attributes are listed in groups:

535 • EVC Service Attributes

536 • EVC End Point Service Attributes

537 • Subscriber UNI Service Attributes

538 The EVC and EVC End Point Service Attributes are organized into 7 groups (Classes), 3 for EVC
539 and 4 for EVC End Point, with the schemas for each of the different types of Subscriber Ethernet
540 Services (EPL, EVPL, EP-LAN, EVP-LAN, EP-TREE, EVP-TREE) that are defined as composi-
541 tions of the appropriate groups, along with the appropriate EVC End Points Service Attribute for
542 the corresponding EVC type (the composite tables are listed in section 12.4).

543 **12.1   Service Attribute Groupings**

544 Table 7 below shows how the EVC and EVC EP Service Attributes for each of the Subscriber
545 Ethernet EVC types are constructed from the set of 7 Service Attribute groups.

546

	Service Attributes						
	EVC Groups			**EVC EP Groups**			
EVC Type	Common	Not EPL	Virtual	Common	Not EPL	Virtual	ETREE
EPL	x			x			
EVPL	x	x	x	x	x	x	
EP-LAN	x	x		x	x		
EVP-LAN	x	x	x	x	x	x	
EP-TREE	x	x		x	x		x
EVP-TREE	x	x	x	x	x	x	x

547 **Table 7 – Service Attribute Groupings**

548 **Examples:**

549 For EPL:

550 • EVC Service Attributes is the Common EVC group

551 • EVC EP Service Attributes is the Common EVC EP group

552

553 For EVP-LAN:

554 • EVC Service Attributes is the concatenation of the Common, Not EPL and Virtual EVC
555 groups

556 • EVC EP Service Attributes is the concatenation of the Common, Not EPL and Virtual EVC
557 EP groups

558 For EP-TREE:

559 • EVC Service Attributes is the concatenation of the Common and Not EPL EVC groups

560 • EVC EP Service Attributes is the concatenation of the Common, Not EPL and ETREE
561 EVC EP groups

562 The following sections enumerate the Service Attributes. For each Service Attribute, the first line
563 includes the Service Attribute name, the JSON Name, and the argument type and allowed values.
564 The second line contains a description and reference. The third line contains usage information,
565 i.e., for which Business Functions (POQ, Quote, Order, Inventory) the Service Attribute is in-
566 cluded in the product configuration, and if included, whether it is optional or required. Some en-
567 tries include additional coding and validation notes when the value of the field depends on (or in
568 some cases is depended on) by another field.

569 Not all MEF 10.4 Service Attributes are included in the data models. The Service Attributes that
570 are not included are also listed in section 14. Some Service Attributes are not included because
571 they are included in the Product Independent information portion of the API (e.g., some of the
572 Identifiers) and some Service Attributes are not included because they are constants (i.e., can only
573 have one possible value) either because they are defined that way (e.g., the Subscriber UNI Service
574 Frame Format Service Attribute must be "802.3-2012) or because the value is implicit for Sub-
575 scriber Ethernet (e.g., the EVC Type Service attribute for an EPL Service must be point-to-point).

576 Conversely, there are a few Service Attributes included in the schemas that are not included in
577 MEF 10.4.

578 **12.2   Subscriber Ethernet EVC Service Attributes**

579 The following tables are organized by schema file. The first part (in blue) specifies the filename
580 and the list of classes included in the file. Following that, for each class there is a class description
581 (white background) followed by a list of properties in the class (yellow background).

582 **12.2.1   EVC Common Service Attributes**

583 This Service Attribute group is common to all Subscriber Ethernet Services:

Carrier Ethernet EVC Common
  Schema File Name: carrierEthernetEvcCommon
  Includes classes:
    CarrierEthernetEvcCommon (Carrier Ethernet EVC Common)

Class: CarrierEthernetEvcCommon *(Carrier Ethernet EVC Common)*

These are the MEF 10.4 EVC Service Attributes that are common to all Subscriber (EVC-based) Ethernet Services defined in MEF 6.3.

Property Name: *EVC List of Class of Service Names Service Attribute*
 Reference: MEF 10.4 sec. 8.7
  JSON Name: *listOfCosNames* - **Type:** *List of string*

Description: The EVC List of Class of Service Names Service Attribute is used to specify all of the Class of Service Names supported by an EVC. The value of the EVC 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]

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

Property Name: *EVC Service Level Specification Service Attribute*
 Reference: MEF 10.4 sec. 8.8
  JSON Name: *carrierEthernetSls* - **Type:** *List of object*

Description: The SLS includes technical details of the service level in terms of Performance Objectives, agreed between the Service Provider and the Subscriber or between Service Provider and the Operator as part of the Service Level Agreement. A given SLS might contain 0,1 or more Performance Objectives for each Performance Metric.

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

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

Property Name: *EVC Maximum Service Frame Size Service Attribute*
 Reference: MEF 10.4 sec. 8.10
  JSON Name: *maximumFrameSize* - **Type:** *integer*

Description: Maximum size of Service frames that can be carried over the EVC. This must be at least 1522 (1518+C-tag).

Allowed values: Minimum: 1522

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

Property Name: *EVC Available MEG Level Service Attribute*
 Reference: MEF 10.4 sec. 8.11
  JSON Name: *availableMegLevel* - **Type:** *enum*

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 6.3 constrains it to 0-6 for all Subscriber Ethernet Services.

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

Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required

585

586 **12.2.2 EVC Common Service Attributes - Not EPL**

587 This Service Attribute group is only used for all Subscriber Ethernet Services other than EPL:

Carrier Ethernet EVC EP Not EPL
Schema File Name: carrierEthernetEvcEpNotEpl
Includes classes:
CarrierEthernetEvcNotEpl (Carrier Ethernet EVC Not EPL)

Class: CarrierEthernetEvcNotEpl *(Carrier Ethernet EVC Not EPL)*
These are the MEF 10.4 EVC Service Attributes that are common to all Subscriber (EVC-based) Ethernet Services defined in MEF 6.3 based on the Service Attributes in MEF 10.4 - except EPL.

Property Name: *EVC Data Service Frame Disposition Service Attribute* **Reference:** MEF 10.4 sec. 8.4   **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 Subscriber by some other means.
Allowed values: Class: FrameDisposition
Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required
Validation Notes: For EPL this must be unconditional for all 3 entries so not included

590

591   **12.2.3   EVC Common Service Attributes - Virtual Private**

592   This Service Attribute group is common to all Virtual Private Services (EVPL, EVP-LAN, EVP-
593   TREE):

Carrier Ethernet VP EVC Common
Schema File Name: carrierEthernetVpEvcCommon
Includes classes:
CarrierEthernetVpEvcCommon (Carrier Ethernet Virtual Private EVC Common)
Class: CarrierEthernetVpEvcCommon *(Carrier Ethernet Virtual Private EVC Common)*
These are the MEF 10.4 EVC Service Attributes that are common to all Virtual Private Subscriber (EVC-based) Ethernet Services (EVPL, EVP-LAN, EVP_TREE) defined in MEF 6.3.
Property Name: *EVC C-Tag PCP Preservation Service Attribute*
Reference: MEF 10.4 sec. 8.5
JSON Name: *cTagPcpPreservation* - **Type:** *object*
Description: The EVC C-Tag PCP Preservation Service Attribute can be used to preserve the value of the PCP field in C-Tagged Service Frames across an EVC. The value of the EVC C-Tag PCP Preservation Service Attribute is either Enabled or Disabled.
Allowed values: Class: EnabledDisabled
Usage: POQ: Optional **Quote:** Optional **Order:** Required **Inventory:** Required
Validation Notes: For Private Services this **must** be *Enabled* so it is only included for Virtual Private Services.
Property Name: *EVC C-Tag DEI Preservation Service Attribute*
Reference: MEF 10.4 sec. 8.6
JSON Name: *cTagDeiPreservation* - **Type:** *object*
Description: If *ENABLED*, then the value of the C-Tag DEI field is preserved from Ingress to Egress.
Allowed values: Class: EnabledDisabled
Usage: POQ: Optional **Quote:** Optional **Order:** Required **Inventory:** Required
Validation Notes: For Private Services this **must** be *Enabled* so it is only included for Virtual Private Services.

595

596 **12.3 Subscriber Ethernet EVC End Point Service Attributes**

597 **12.3.1 EVC EP Common Service Attributes**

598 This End Point Service Attribute group is common to all Subscriber Ethernet Services:

Carrier Ethernet EVC Common
  Schema File Name: carrierEthernetEvcCommon
  Includes classes:
    CarrierEthernetEvcEpCommon (Carrier Ethernet EVC EP Common)

Class: CarrierEthernetEvcEpCommon *(Carrier Ethernet EVC EP Common)*

This set of EVC End Point Service Attributes are applicable to all Subscriber (EVC-based) Ethernet Services defined in MEF 6.3 based on the Service Attributes in MEF 10.4.

Property Name: *EVC EP ID Service Attribute*
 Reference: MEF 10.4 sec. 10.1
  JSON Name: *identifier* - **Type:** *string*

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

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

Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required

Property Name: *EVC EP UNI Service Attribute*
 Reference: MEF 10.4 sec. 10.2
  JSON Name: *uni* - **Type:** *object*

Description: The value of the EVC EP UNI Service Attribute is a Subscriber UNI ID Service Attribute value per Section 9.1. The value of the EVC EP UNI Service Attribute serves to specify the UNI where the EVC EP is located. The EVC EP is said to be at this UNI.

Allowed values: Class: CarrierEthernetEiReference

Usage: **POQ:** Required **Quote:** Required **Order:** Required **Inventory:** Required

Property Name: *EVC EP Ingress Class of Service Map Service Attribute*
 Reference: MEF 10.4 sec. 10.5 and MEF 45.1 sec. 6.2
  JSON Name: *ingressClassOfServiceMap* - **Type:** *object*

Description: The value of the EVC 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*, or *DSCP*, M is a map that can be used to assign Class of Service Names to Service 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: CosFromCtagPcp, Class: CosFromDscp **with Discriminator:** mapType

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

Validation Notes: The Class of Service Name or Names **must** be included in the ***ovc.listOfClassOfServiceNames***. Per MEF 6.3 if there is only one Class of Service in *EplEvc.listOfCosNames* this **should** be *ENDPOINT*.

Property Name: *EVC EP Color Map Service Attribute*
 Reference: MEF 10.4 sec. 10.6
  JSON Name: *colorMap* - **Type:** *object*

Description: The value of the EVC 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 Service Frame. (The form of M depends on the value of F.). The EVC EP Color Map Service Attribute is the mechanism by which the Color for an Ingress Service Frame that is mapped to an EVC EP is indicated by the content in the Service Frame header.

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

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

Property Name: *EVC EP Egress Map Service Attribute*
 Reference: MEF 10.4 sec. 10.7
  JSON Name: *egressMap* - **Type:** *object*

Description: The value of the EVC EP Egress Map Service Attribute is None or a map of the form {Corresponding Ingress Service Frame Class of Service Name, Corresponding Ingress Service Frame Color} to {Egress Service Frame C-Tag PCP value, Egress Service Frame C-Tag DEI value} or Discard.

Allowed values: Class: EvcEgressMap

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

Validation Notes: This is required for all EVCs unless the EVC Map is UT/PT.

Property Name: *EVC EP Class of Service Name Ingress Bandwidth Profile Service Attribute*
 Reference: MEF 10.4 sec. 10.9
  JSON Name: *ingressBandwidthProfilePerClassofServiceName* - **Type:** *object*

Description: The EVC EP Class of Service Name Ingress Bandwidth Profile Service Attribute is used to limit the rate of Ingress Service Frames with a given Class of Service Name at a UNI. The value is non-empty list of pairs of the form ⟨x,y⟩, where x is a Class of Service Name that is in the value of the EVC List of Class of Service Names Service Attribute and y = BWP Flow Parameters.

Allowed values: Class: BandwidthProfilePerClassOfServiceName

Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required

Property Name: *EVC EP Subscriber MEG MIP Service Attribute*
 Reference: MEF 10.4 sec. 10.13
  JSON Name: *subscriberMegMip* - **Type:** *enum*

Description: The value of the EVC EP Subscriber MEG MIP Service Attribute is either None or an integer in the range 0 – 7 that indicates the MEG Level of a Subscriber MEG MIP.

Allowed values: Enum: "0", "1", "2", "3", "4", "5", "6", "7", "NONE"

Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required

600

601 **12.3.2   EVC EP Common Service Attributes - Not EPL**

602 This End Point Service Attribute group is common to all Subscriber Ethernet Services other than
603 EPL:

Carrier Ethernet EVC EP Not EPL
  Schema File Name: carrierEthernetEvcEpNotEpl
  Includes classes:
    CarrierEthernetEvcEpNotEpl (Carrier Ethernet EVC EP Not EPL)

Class: CarrierEthernetEvcEpNotEpl *(Carrier Ethernet EVC EP Not EPL)*

This set of EVC End Point Service Attributes are applicable to all Subscriber (EVC-based) Ethernet Services defined in MEF 6.3 based on the Service Attributes in MEF 10.4 - except EPL.

Property Name: *EVC EP Class of Service Name Egress Bandwidth Profile Service Attribute*
 Reference: MEF 10.4 sec. 10.11
  JSON Name: *egressBandwidthProfilePerClassofServiceName* - **Type:** *object*

Description: The EVC EP Class of Service Name Egress Bandwidth Profile Service Attribute is used to limit the rate of all Egress Service Frames with a given Class of Service Name, as determined at the ingress UNI for each frame per the EVC EP Ingress Class of Service Map Service Attribute (MEF 10.4 Section 10.5). The value of the EVC EP Class of Service Name Egress Bandwidth Profile Service Attribute is either None or a non-empty list of pairs of the form ⟨x, y⟩ where x is a Class of Service Name contained in the value of the EVC List of Class of Service Names Service Attribute (MEF 10.4 Section 8.7) but not equal to Discard and y= the 3-tuple ⟨CIR, CIRmax, ER⟩ where the elements of the 3-tuple are defined in MEF 10.4 Section 12.1.2.

Allowed values: Class: BandwidthProfilePerClassOfServiceName

Usage: **POQ:** Optional   **Quote:** Optional   **Order:** Optional   **Inventory:** Optional

Property Name: *EVC EP Source MAC Address Limit Service Attribute*
 Reference: MEF 10.4 sec. 10.12
  JSON Name: *sourceAddressMacLimit* - **Type:** *object*

Description: The value of the EVC EP Source MAC Address Limit Service Attribute is either None or the pair ⟨N, t ⟩ where N is an integer ≥1 and t is a time duration. When the value of the EVC EP Source MAC Address Limit Service Attribute = ⟨N, t⟩ the number of source MAC Addresses that can be used in Ingress Service Frames is limited.

Allowed values: Class: SourceMacAddressLimit

Usage: **POQ:** Optional   **Quote:** Optional   **Order:** Optional   **Inventory:** Optional

Validation Notes: Per MEF 6.3, the value of this must be NONE (absent).

605

12.3.3 EVC EP Service Attributes - Virtual Private

This End Point Service Attribute group is common to all Virtual Private Services (EVPL, EVP-LAN, EVP-TREE):

Carrier Ethernet EVC EP Virtual Private
Schema File Name: carrierEthernetEvcEpVirtualPrivate Includes classes: CarrierEthernetEvcEpVirtualPrivate (Carrier Ethernet EVC EP Virtual Private)
Class: CarrierEthernetEvcEpVirtualPrivate *(Carrier Ethernet EVC EP Virtual Private)*
Property Name: *EVC EP Map Service Attribute* **Reference:** MEF 10.4 sec. 10.4   **JSON Name:** *evcEndPointMap* - **Type:** *object*
Description: The EVC EP Map Service Attribute is used to map Service Frames to the EVC EP. The value of the EVC EP Map Service Attribute (in MEF 10.4) is one of List, All, or UT/PT. This attribute is only available for Virtual Private services since Private services always require *All*. For Virtual Private Services it can be *UT/PT* (Untagged/Priority Tagged) or *UT/PT*. When the value of the EVC EP Map Service Attribute = List, a non-empty list of unique integers, each in the range 1,2,…,4094 is specified.
Allowed values: Class: EndPointMapEvc
Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required
Validation Notes: For Virtual Private services, this **must** be *LIST* or *UT/PT*. For a given UNI, the values of all of the EVC End Point Map Service Attribute **must** be such that each possible C-VLAN ID value maps to at most one EVC End Point. **Coding hint:** This must be ALL for EPL, EP-LAN, EP-TREE Services, LIST or UT/PT for EVPxxx Services

611   **12.3.4   EVC EP Service Attributes - E-Tree**

612   This End Point Service Attribute group is only used for EP-TREE and EVP-TREE:

Carrier Ethernet Etree Evc Specific
Schema File Name: carrierEthernetEtreeEvcSpecific
Includes classes:
CarrierEthernetEvcEpEtree (Etree Specific EVC EP Service Attributes)
Class: CarrierEthernetEvcEpEtree *(Etree Specific EVC EP Service Attributes)*
These EVC End Point Service Attributes (only one, actually) are applicable to E-TREE Services, i.e., EP-TREE and EVP-TREE
Property Name: *EVC EP Role Service Attribute*   **Reference:** MEF 10.4 sec. 10.3   **JSON Name:** *role* - **Type:** *enum*
Description: Indicates how EI Frames mapped to the EVC End Point can be forwarded
Allowed values: Enum: "ROOT", "LEAF", "TRUNK"
Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required
Validation Notes: Not needed for EPL since the two EPs are always Roots

614

615 **12.4 Subscriber Ethernet EVC and EVC End Point Service Attributes**

616 This section lists the EVC and EVC End Point Service Attribute tables for each of the Subscriber
617 Ethernet EVC types that are constructed as a composition from the set of 7 Service Attribute groups
618 described in sections 12.2 and 12.3 above, along with the appropriate EVC End Points Service
619 Attribute for the corresponding EVC type.

620 **12.4.1 Subscriber Ethernet EPL Service Attributes**

Ethernet Private Line EVC
    Schema File Name: ethernetPrivateLineEvc
    Includes classes:
        EplEvc   (EPL EVC Service Attributes)
        EplEvcEndPoint   (EPL EVC End Point Service Attributes)

Class: EplEvc   *(EPL EVC Service Attributes)* *The EVC Service Attributes for Ethernet Private Line (EPL) Service as defined in MEF 6.3 based on the Service Attributes defined in MEF 10.4. The EVC Service Attributes are applicable at every UNI in the EVC.*

Property Name: *EPL EVC Service Attributes*   **JSON Name:** *eplEvcAttributes* - **Type:** *object*
Allowed values: Compose (allOf): Class: CarrierEthernetEvcCommon
Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

Property Name: *EVC End Points - EPL* **Reference:** MEF 10.4 sec. 10   **JSON Name:** *evcEndPoints* - **Type:** *List of object*
Description: An array of two EVC End Points.
Allowed values: List of [2...2] Unique Class: EplEvcEndPoint
Usage: **POQ:** Required **Quote:** Required **Order:** Required **Inventory:** Required

Type: EplEvcEndPoint   *(EPL EVC End Point Service Attributes)* *An EVC End Point is a logical entity at a given UNI that is associated with a distinct set of frames passing over that UNI. EVC End Point Service Attributes are the attributes of an EVC that can be different at each UNI in the EVC. These are the EVC End Point Service Attributes applicable to Ethernet Private Line (EPL) Services defined in MEF 6.3 based on the Service Attributes in MEF 10.4.*

Property Name: *EPL EVC End Point* **Type:** *object*
Allowed values: Compose (allOf): Class: CarrierEthernetEvcEpCommon
Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

623

624 **12.4.2 Subscriber Ethernet EVPL Service Attributes**

Ethernet Virtual Private Line EVC
Schema File Name: ethernetVirtualPrivateLineEvc
Includes classes:
EvplEvc (EVPL EVC Service Attributes)
EvplEvcEndPoint (EVPL EVC End Point Service Attributes)

Class: EvplEvc *(EVPL EVC Service Attributes)*

Property Name: *EVPL EVC Service Attributes*
  JSON Name: *evplEvcAttributes* - **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcCommon, Class: CarrierEthernetVpEvcCommon, Class: CarrierEthernetEvcNotEpl

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

Property Name: *EVC End Points - EVPL*
 Reference: MEF 10.4 sec. 10
  JSON Name: *evcEndPoints* - **Type:** *List of object*

Description: An array of two EVC End Points.

Allowed values: List of [2...2] Unique Class: EvplEvcEndPoint

Usage: **POQ:** Required **Quote:** Required **Order:** Required **Inventory:** Required

Type: EvplEvcEndPoint *(EVPL EVC End Point Service Attributes)*

Property Name: *EVPL EVC End Point* **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcEpCommon, Class: CarrierEthernetEvcEpVirtualPrivate, Class: CarrierEthernetEvcEpNotEpl

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

628

629   ### 12.4.3   Subscriber Ethernet EP-LAN Service Attributes

Ethernet Private LAN EVC
  Schema File Name: ethernetPrivateLanEvc
  Includes classes:
    EplanEvc (EP-LAN EVC Service Attributes)
    EplanEvcEndPoint (EP-LAN EVC End Point Service Attributes)

Class: EplanEvc *(EP-LAN EVC Service Attributes)*

Property Name: *EP-LAN EVC Service Attributes*
  JSON Name: *eplanEvcAttributes* - **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcCommon, Class: CarrierEthernetEvcNotEpl

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

Property Name: *EVC End Points - EPLAN*
 Reference: MEF 10.4 sec. 10
  JSON Name: *evcEndPoints* - **Type:** *List of object*

Description: An array of two or more EVC End Points.

Allowed values: List of [2...*] Unique Class: EplanEvcEndPoint

Usage: **POQ:** Required **Quote:** Required **Order:** Required **Inventory:** Required

Type: EplanEvcEndPoint *(EP-LAN EVC End Point Service Attributes)*

Property Name: *EP-LAN EVC End Point* **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcEpCommon, Class: CarrierEthernetEvcEp-NotEpl

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

633

634 **12.4.4** **Subscriber Ethernet EVP-LAN Service Attributes**

Ethernet Virtual Private LAN EVC
Schema File Name: ethernetVirtualPrivateLanEvc
Includes classes:
EvplanEvc (EVP-LAN EVC Service Attributes)
EvplanEvcEndPoint (EVP-LAN EVC End Point Service Attributes)

Class: EvplanEvc *(EVP-LAN EVC Service Attributes)*

Property Name: *EVP-LAN EVC Service Attributes*
 JSON Name: *evplanEvcAttributes* - **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcCommon, Class: CarrierEthernetVpEvcCommon, Class: CarrierEthernetEvcNotEpl

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

Property Name: *EVC End Points - EVPLAN*
 Reference: MEF 10.4 sec. 10
 JSON Name: *evcEndPoints* - **Type:** *List of object*

Description: An array of two or more EVC End Points.

Allowed values: List of [2...*] Unique Class: EvplanEvcEndPoint

Usage: **POQ:** Required **Quote:** Required **Order:** Required **Inventory:** Required

Type: EvplanEvcEndPoint *(EVP-LAN EVC End Point Service Attributes)*

Property Name: *EVP-LAN EVC End Point* **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcEpCommon, Class: CarrierEthernetEvcEpVirtualPrivate, Class: CarrierEthernetEvcEpNotEpl

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

638

639   **12.4.5   Subscriber Ethernet EP-TREE Service Attributes**

Ethernet Private Tree EVC
    Schema File Name: ethernetPrivateTreeEvc
    Includes classes:
        EptreeEvc (EP-TREE EVC Service Attributes)
        EptreeEvcEndPoint (EP-TREE EVC End Point Service Attributes)

Class: EptreeEvc *(EP-TREE EVC Service Attributes)*

Property Name: *EP-TREE EVC Service Attributes*
  JSON Name: *eptreeEvcAttributes* - **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcCommon, Class: CarrierEthernetEvcNotEpl

Usage: **POQ:** Optional   **Quote:** Optional   **Order:** Optional   **Inventory:** Optional

Property Name: *EVC End Points - EPTREE*
 Reference: MEF 10.4 sec. 10
  JSON Name: *evcEndPoints* - **Type:** *List of object*

Description: An array of two or more EVC End Points.

Allowed values: List of [2...*] Unique Class: EptreeEvcEndPoint

Usage: **POQ:** Required   **Quote:** Required   **Order:** Required   **Inventory:** Required

Type: EptreeEvcEndPoint *(EP-TREE EVC End Point Service Attributes)*

Property Name: *EP-TREE EVC End Point*   **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcEpCommon, Class: CarrierEthernetEvcEpE-tree, Class: CarrierEthernetEvcEpNotEpl

 Usage: **POQ:** Optional   **Quote:** Optional   **Order:** Optional   **Inventory:** Optional

643

644 **12.4.6   Subscriber Ethernet EVP-TREE Service Attributes**

Ethernet Virtual Private Tree EVC
    Schema File Name: ethernetVirtualPrivateTreeEvc
    Includes classes:
        EvptreeEvc (EVP-TREE EVC Service Attributes)
        EvptreeEvcEndPoint (EVP-TREE EVC End Point Service Attributes)

Class: EvptreeEvc *(EVP-TREE EVC Service Attributes)*

Property Name: *EVP-TREE EVC Service Attributes*
  JSON Name: *evptreeEvcAttributes* - **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcCommon, Class: CarrierEthernetVpEvcCommon, Class: CarrierEthernetEvcNotEpl

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

Property Name: *EVC End Points - EVPTREE*
 Reference: MEF 10.4 sec. 10
  JSON Name: *evcEndPoints* - **Type:** *List of object*

Description: An array of two or more EVC End Points.

Allowed values: List of [2...*] Unique Class: EvptreeEvcEndPoint

Usage: **POQ:** Required **Quote:** Required **Order:** Required **Inventory:** Required

Type: EvptreeEvcEndPoint *(EVP-TREE EVC End Point Service Attributes)*

Property Name: *EVP-TREE EVC End Point* **Type:** *object*

Allowed values: Compose (allOf): Class: CarrierEthernetEvcEpCommon, Class: CarrierEthernetEvcEpVirtualPrivate, Class: CarrierEthernetEvcEpEtree, Class:CarrierEthernetEvcEpNotEpl

 Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

647

648 **12.5   Subscriber Ethernet UNI Service Attributes**

Carrier Ethernet Subscriber UNI
   Schema File Name: carrierEthernetSubscriberUni
   Includes classes:
      SubscriberUni (Subscriber UNI Service Attributes)

Class: SubscriberUni *(Subscriber UNI Service Attributes)*

The Subscriber UNI Service Attributes values are agreed to by the Subscriber and the Service Provider to describe the behavior of the UNI. 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: *Subscriber UNI Service Attributes*
  JSON Name: *subscriberUniAttributes* - **Type:** *object*

Allowed values: Compose (allOf): Class: SubscriberUniCommon

Usage: POQ: Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

---

650

Carrier Ethernet Subscriber UNI Common
   Schema File Name: carrierEthernetSubscriberUniCommon
   Includes classes:
      SubscriberUniCommon (Subscriber UNI Common)

Class: SubscriberUniCommon *(Subscriber UNI Common)*

This class contains all of the Subscriber UNI Services Attributes that are used in all four Business Functions (POQ, Quote, Order, Inventory). At this time, this includes all of the Subscriber UNI Service Attributes. This class is used to compose the SubscriberUNI class which has one file for each Business Function that differ in terms of the Service Attributes that are "Required" for the Business Function (rather than "Optional").

Property Name: *Subscriber UNI Instantiation Service Attribute*
 Reference: MEF 10.4 sec. 9.2
  JSON Name: *instantiation* - **Type:** *enum*

Description: When the value is *Physical*, the UNI is implemented using one or more instances of a standard Physical Layer. When the value is Virtual, the physical layer is not specified.

Allowed values: Enum: "PHYSICAL", "VIRTUAL"

Usage: POQ: Optional **Quote:** Optional **Order:** Required **Inventory:** Required

Property Name: *Subscriber UNI Virtual Frame Map Service Attribute*
 Reference: MEF 10.4 sec. 9.2 and sec. 9.3
  JSON Name: *virtualFrameMap* - **Type:** *List of object*

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: List of [0...1] Class: VirtualFrameMap

Usage: POQ: Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

---

Property Name: *Subscriber UNI List of Physical Links Service Attribute*
 Reference: MEF 10.4 sec. 9.4
  JSON Name: *listOfPhyLinks* - **Type:** *List of object*

Description: The value of the Subscriber UNI List of Physical Links Service Attribute is a list of 5-tuples of the form , 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 value of **demarcConnectorType** specifies the type of connector used. Note that the **id** and the **physicalLink** come from the CarrierEthernetPhysicalLink class.

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

Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required

Validation Notes: If **Subscriber UNI Instantiation Service Attribute** is *PHYSICAL*, there must be at least one Physical Link. If **Subscriber UNI Instantiation Service Attribute** is *VIRTUAL*, there must be 0 Physical Links.

Property Name: *Subscriber UNI Link Aggregation Service Attribute*
 Reference: MEF 10.4 sec. 9.5
  JSON Name: *linkAggregation* - **Type:** *enum*

Description: Data type 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/ENNI 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: Enum: "NONE", "2_LINK_ACTIVE_STANDBY", "ALL_ACTIVE", "OTHER"

Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required

Property Name: *Subscriber UNI Port Conversation ID to Aggregation Link Map Service Attribute*
 Reference: MEF 10.4 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.

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

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

Property Name: *Subscriber UNI Maximum Service Frame Size Service Attribute*
 Reference: MEF 10.4 sec. 9.8
  JSON Name: *maximumServiceFrameSize* - **Type:** *integer*

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

Allowed values: Minimum: 1522

Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required

Property Name: *Subscriber UNI Maximum Number of EVC EPs Service Attribute*
 Reference: MEF 10.4 sec. 9.9
  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

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

Property Name: *Subscriber UNI Maximum Number of C-Tag VLAN IDs per EVC EP Service Attribute*
 Reference: MEF 10.4 sec. 9.10
  JSON Name: *maximumNumberOfCtagVlanIdsPerEndPoint* - **Type:** *integer*

Description: An integer greater than or equal to 1 that limits the number of C-Tag VLAN IDs that can map to each EVC End Point.

Allowed values: Minimum: 1 Maximum: 4094

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

Property Name: *Subscriber UNI Token Share Service Attribute*
 Reference: MEF 10.4 sec. 9.11
  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 Subscriber UNI.

Allowed values: Class: EnabledDisabled

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

Property Name: *Subscriber UNI Envelopes Service Attribute*
 Reference: MEF 10.4 sec. 9.12
  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}.

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

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Required

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: *Subscriber UNI Link OAM Service Attribute*
 Reference: MEF 10.4 sec. 9.13
  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: Class: EnabledDisabled

Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required

Property Name: *Subscriber UNI MEG Service Attribute*
 Reference: MEF 10.4 sec. 9.14
  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: Class: EnabledDisabled

Usage: **POQ:** Optional **Quote:** Optional **Order:** Required **Inventory:** Required

Property Name: *Subscriber UNI LAG Link MEG Service Attribute*
 Reference: MEF 10.4 sec. 9.15
  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: Class: EnabledDisabled

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional

Property Name: *Subscriber UNI L2CP Address Set Service Attribute*
 Reference: MEF 10.4 sec. 9.16 and MEF 45.1 sec. 8.1
  JSON Name: *l2cpAddressSet* - **Type:** *enum*

Description: L2CP Address Set Service Attribute is defined in MEF 45.1. **CTA** CEVLAN Tag Aware for VLAN-based services where the CE-VLAN ID is used to map a frame to a service. **CTB** CVLAN Tag Blind for Port-based services where the CE-VLAN ID not used to map a frame to a service. **CTB2** CVLAN Tag Blind Option 2 for point-to-point Port-based services that support the EPL Option 2 L2CP processing.

Allowed values: Enum: "CTA", "CTB", "CTB2"

Usage:　　**POQ:** Optional　**Quote:** Optional　**Order:** Required　**Inventory:** Required

Property Name: *Subscriber UNI L2CP Peering Service Attribute*
 Reference: MEF 10.4 sec. 9.17 and MEF 45.1 sec. 8.2
  JSON Name: *l2cpPeering* - **Type:** *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: Class: L2cpPeering

Usage:　　**POQ:** Optional　**Quote:** Optional　**Order:** Required　**Inventory:** Required

652

13 Common Classes and Types

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

13.1   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

Usage: Common: Required

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

Usage: Common: Required

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

Usage: Common: Required

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

Usage: Common: Required

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

Usage: Common: Required

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

Usage: Common: Required

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:

Usage: Common: Required

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"

Usage: Common: Required

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]+

Usage: Common: Required

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**, or **EnniService.envelopes**.

All Bandwidth Profile Flows in the same envelope **must** satisfy the same criterion listed in R230 of MEF 26.2.

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

Usage: Common: Required

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]

Usage: Common: Required

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]+

Usage: Common: Required

Validation Notes: For an Operator service the *Envelope Name* must be in either **OperatorUNI.envelopes** or **AccessElineOvcEndPoint.ovcEndPointEnvelopes**. For a Subscriber service the *Envelope Name* must be in **SubscriberUNI.envelopes**.

Property Name: *Envelope CF0*
 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:

Usage: **Common:** Required

Class: BandwidthProfilePerClassOfServiceName *(COS Bandwidth Profile)*

This Class is used for both the Ingress Bandwidth Profile per Class of Service Name (MEF 26.2 §16.12) and Egress Bandwidth Profile per Egress Equivalence Class Name (MEF 26.2 §16.13) . The value is a list of pairs of the form⟨x, y⟩ where x is a Class of Service Name that is in the value of the OVC List of Class of Service Names Service Attribute (Section 12.12) for the OVC that associates the OVC End Point for Ingress and an Egress Equvalence Class Name for Egress. And y, if specified, is a reference to a Bandwidth Profile Flow. There is at most one pair in the list for each CoS or EEC Name (Except Discard).

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 Egress Equivalence Class (EEC) Name.

Allowed values: [Any String]

Usage: **Common:** Required

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

Usage: **Common:** Required

665

666   ## 13.2   Class of Service

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

Class: CosFromEp *(CoS 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 CoS = EVC/OVC EP.

Allowed values: "ENDPOINT"

Usage: Common: Required

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

Description: The single CoS at this EVC/OVC EP

Allowed values: [Any String]

Usage: Common: Required

Validation Notes: The value must be in the AccessElineOvc.listOfClassOfServiceNames

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

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

Allowed values: Class: CosL2cp

Usage: Common: Required

Class: CosFromCtagPcp *(CoS 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 CoS is C-Tag PCP.

Allowed values: "C_TAG_PCP"

Usage: Common: Required

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 Class of Service.

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

Usage: Common: Required

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

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

Allowed values: Class: CosL2cp

Usage: Common: Required

Class: CosUniPcpMapEntry *(CoS to UNI PCP Map Entry)*

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

Property Name: *UNI PCP Value*
  JSON Name: *pcpVal* - **Type:** *enum*

Description: A value 0 to 7

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

Usage: **Common:** Required

Property Name: *UNI CoS Name*
  JSON Name: *pcpCosName* - **Type:** *string*

Description: A CoS Name or DISCARD

Allowed values: [Any String]

Usage: **Common:** Required

Validation Notes: If not DISCARD, the CoS Name must be in AccessElineOvc.listOfClassOfServiceNames

Class: CosFromStagPcp *(CoS 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 CoS is S-Tag PCP.

Allowed values: "S_TAG_PCP"

Usage: **Common:** Required

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

Usage: **Common:** Required

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

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

Allowed values: Class: CosL2cp

Usage: **Common:** Required

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

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

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

Property Name: *ENNI PCP Value*
  JSON Name: *pcpVal* - **Type:** *enum*
Allowed values: Enum: "0", "1", "2", "3", "4", "5", "6", "7"
Usage: Common: Required

Property Name: *ENNI CoS Name*
  JSON Name: *pcpCosName* - **Type:** *string*
Allowed values: [Any String]
Usage: Common: Required

Class: CosFromDscp *(CoS 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 CoS is DSCP.
Allowed values: "DSCP"
Usage: Common: Required

Property Name: *CoS Map*
  JSON Name: *map_M* - **Type:** *object*
Description: This maps CoS values to a list of DSCP values for each of IPv4 and IPv6 and also to non-IP packets.
Allowed values: Class: CosDscpMap
Usage: Common: Required

Property Name: *CoS L2CP*
  JSON Name: *l2cp_P* - **Type:** *object*
Description: This specifies the CoS for L2CPs carried over the OVC.
Allowed values: Class: CosL2cp
Usage: Common: Required

Class: CosDscpMap *(CoS to DSCP Map)*

This is a list of CoS to DSCP Map 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

Usage: **Common:** Required

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

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

Allowed values: [Any String]

Usage: **Common:** Required

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

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

Allowed values: [Any String]

Usage: **Common:** Required

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

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

Allowed values: [Any String]

Usage: **Common:** Required

Class: CosDscpMapEntry ***(CoS to DSCP 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

Usage: **Common:** Optional

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

Allowed values: Class: DscpValues

Usage: **Common:** Optional

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

Allowed values: [Any String]

Usage: **Common:** Required

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

Usage: Common: Optional

Class: CosL2cp *(CoS to L2CP)*

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

Usage: Common: Required

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

Description: CoS Name String to associate with the L2CP

Allowed values: [Any String]

Usage: Common: Required

677

13.3   Color Identifier

678

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 S-tag or C-tag DEI field. MEF 26.2 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"

Usage: Common: Required

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"

Usage: Common: Required

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

Usage: Common: Required

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

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

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

Usage: Common: Required

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

Allowed values: Enum: "GREEN", "YELLOW"

Usage: Common: Required

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"

Usage: Common: Required

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

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

Usage: Common: Required

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

Usage: Common: Required

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

Allowed values: Enum: "GREEN", "YELLOW"

Usage: Common: Required

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

Allowed values: Enum: "GREEN", "YELLOW"

Usage: Common: Required

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

Allowed values: "ENDPOINT"

Usage: Common: Required

Property Name: *Color*
  JSON Name: *epColor* - **Type:** *enum*

Allowed values: Enum: "GREEN", "YELLOW"

Usage: Common: Required

685

686 ## 13.4 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"

Usage: **Common:** Required

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

Usage: **Common:** Required

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]

Usage: **Common:** Required

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

Allowed values: Class: PcpValueAndDiscard

Usage: **Common:** Required

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 DIS-CARD) 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"

Usage: **Common:** Required

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

Usage: **Common:** Required

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]

Usage: **Common:** Required

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

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

Allowed values: Class: DeiValueAndDiscard

Usage: **Common:** Required

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

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

Allowed values: Class: DeiValueAndDiscard

Usage: **Common:** Required

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"

Usage: **Common:** Required

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

Usage: **Common:** Required

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]

Usage: **Common:** Required

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: Class: PcpValueAndDiscard

Usage: **Common:** Required

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: Class: PcpValueAndDiscard

Usage: **Common:** Required

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"

Usage: Common: Required

Property Name: *List of CNPCP_CCDEI Entries*
  JSON Name: *cnPcpCcDeiEntries* - **Type:** *List of object*

Description: A list of entries, one per CoS, the 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

Usage: Common: Required

Class: CnPcpCcDei *(CNPCPCCDEI 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]

Usage: Common: Required

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: Class: PcpValueAndDiscard

Usage: Common: Required

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: Class: DeiValueAndDiscard

Usage: Common: Required

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: Class: DeiValueAndDiscard

Usage: Common: Required

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 (in which case they are all treated as 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*

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

Usage: Common: Required

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]

Usage: Common: Required

Validation Notes: This must be a CoS Name from ***EVC 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: Class: PcpValueAndDiscard

Usage: Common: Required

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

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

Allowed values: Class: PcpValueAndDiscard

Usage: Common: Optional

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

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

Description: A DEI value to use for Green frames

Allowed values: Class: DeiValueAndDiscard

Usage: Common: Optional

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: Class: DeiValueAndDiscard

Usage: Common: Optional

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

697

698   **13.5   End Point Maps**

Carrier Ethernet End Point Maps
   Schema File Name: carrierEthernetEndPointMaps
   Includes classes:
      EndPointMapFormE (End Point Map - E)
      EndPointMapFormT (End Point Map - T)
      EndPointMapFormV (End Point Map - V)
      EndPointMapFormU (End Point Map - U)
      EndPointMapEvc (End Point Map - EVC)

Class: EndPointMapFormE *(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*

Allowed values: "FORM_E"

Usage: Common: Required

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

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

Usage: Order: Required **Common:** Required

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

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.

Class: EndPointMapFormV *(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.

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

The EndPointMapFormU object class represents the EVC/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*

Allowed values: "FORM_U"

Usage: Common: Required

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

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

Usage: Common: Required

Class: EndPointMapEvc *(End Point Map - EVC)*

*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 about how the Vlan List argument is interpreted

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

Usage: Common: Required

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

Description: A list of VLAN IDs that is interpreted based on the value of vlanType.

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

Usage: Common: Optional

704

705 **13.6   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)
      UniSpecifcAttributes (UNI Specific)
      CarrierEthernetEiReference (Carrier Ethernet EI Reference)
      VirtualFrameMap (Virtual Frame Map)

Type: UniPhysicalLink *(UNI Physical Link)*

This class describes the Physical Link attributes at a UNI. It is a 4-tuple {carrierEthernet, frequency sync enable, PTP enable, connector}. carrierEthernet is a 2-tuple containing an id and physical link type.

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

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

Usage: **Common:** Optional

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

Usage: **Common:** Optional

Class: CarrierEthernetPhysicalLink *(Carrier Ethernet Physical Link)*

*A 2-tuple of the form . 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]+

Usage: **Common:** Required

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_EW", "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"

Usage: **Common:** Required

Class: UniSpecifcAttributes *(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: Class: EnabledDisabled

Usage: **Common:** Required

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: Class: EnabledDisabled

Usage: **Common:** Required

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", "FC", "D4", "OTHER"

Usage: **Common:** Required

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"

Usage: **Common:** Required

Class: CarrierEthernetEiReference *(Carrier Ethernet EI Reference)*

Property Name: *EI Reference href*
  JSON Name: *href* - **Type:** *string*

Description: A reference to an External Interface (UNI, ENNI).

Allowed values: String: Format: uri

Usage: **Common:** Required

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

Usage: **POQ:** Optional **Quote:** Optional **Order:** Optional **Inventory:** Optional **Common:** Optional

712

13.7   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

Usage: **Common:** Required

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 either by colon (:) or hyphen (-).

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

Usage: **Common:** Required

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]+

Usage: **Common:** Optional

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"

Usage: **Common:** Required

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

Description: The EtherType value or LLC value

Allowed values: Minimum: 0

Usage: **Common:** Required

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

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

Allowed values: Minimum: 0

Usage: **Common:** Optional

716

717 ## 13.8   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 Aggregation Links. 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

Usage: **Common:** Required

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

Usage: Common: Required

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 (***OperatorUNI.listofPhysicalLinks*** and ***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

Usage: Common: Required

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

Usage: Common: Optional

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

720

13.9   Service Level Specification

721

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 the technical details of the service level, in terms of Performance Objectives, agreed between the Service Provider and the Subscriber as part of the Service Level Agreement. 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:** *object*

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.

Allowed values: Class: Time

Usage: **Common:** Required

Property Name: *Time Interval*
 Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13
  JSON Name: *timeInterval* - **Type:** *object*

Description: This is the time interval over which to evaluate the performance for the SLS. The **timeInterval** (referred to as *T* in MEF 10.4 and 26.2) is a time interval, 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 **timeInterval** 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: LongDuration

Usage: **Common:** Required

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 one one Class of Service.

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

Usage: **Common:** Required

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

Usage: **Common:** Required

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 *ts* is referred to as *Δtk* (k is subscript).

Allowed values: [Any Integer]

Usage: **Common:** Required

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 *Δtk* has high loss.

Allowed values: Minimum: 0 Maximum: 1

Usage: **Common:** Required

Property Name: *HLLI Consecutive Intervals*
 Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13
  JSON Name: *consecutiveIntervalN* - **Type:** *integer*

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

Allowed values: Minimum: 1

Usage: **Common:** Required

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 this 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

Usage: **Common:** Required

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 this 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

Usage: **Common:** Required

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 this 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

Usage: **Common:** Required

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 this 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: OneWayInterFrameDelayVariationPmMetric

Usage: **Common:** Required

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 this 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

Usage: **Common:** Required

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 this 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

Usage: **Common:** Required

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 this 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

Usage: **Common:** Required

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 this 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

Usage: **Common:** Required

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 this 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

Usage: **Common:** Required

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 this 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

Usage: **Common:** Required

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 the MEF Standards) , a percentile (referred to as Pd in the MEF Standards), and the One Way Frame Delay Objective for this Class of Service Name and these two parameters. See MEF 10.4 section 8.8.2 for EVC and MEF 26.2 section 12.13.2 for OVC.

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 the MEF Standards) 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

Usage: **Common:** Required

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. 8.8.1.5 and MEF 26.2 sec. 12.13.2
  JSON Name: *oneWayFdPercentile* - **Type:** *object*

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

Allowed values: Class: Percentage

Usage: **Common:** Required

Validation Notes: This must be greater than 0.

Property Name: *Frame Delay Objective*
 Reference: MEF 10.4 sec. 8.8.1.5 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: ShortDuration

Usage: **Common:** Required

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

One Way Frame Mean Delay Performance Metric consisting of 1 parameter, a list of Ordered End Point Pairs (referred to as S in the MEF Standards), and the One Way Mean Frame Delay Objective for this Class of Service Name and these Ordered End Point Pairs. See MEF 10.4 section 8.8.3 for EVC and MEF 26.2 section 12.13.3 for OVC.

Property Name: *Mean Frame Delay Ordered Pairs*
  JSON Name: *orderedPairList* - **Type:** *List of object*

Description: The set (referred to as *S* in the MEF Standards) 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

Usage: **Common:** Required

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*
  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: ShortDuration

Usage: **Common:** Required

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 the MEF Standards) , a percentile (referred to as Pr in the MEF Standards), and the One Way Frame Delay Range Objective for this Class of Service Name and these two parameters. See MEF 10.4 section 8.8.4 for EVC and MEF 26.2 section 12.13.4 for OVC.

Property Name: *Frame Delay Range Ordered Pairs*
  JSON Name: *orderedPairList* - **Type:** *List of object*

Description: The set (referred to as *S* in the MEF Standards) 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

Usage: **Common:** Required

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*
  JSON Name: *oneWayFdrPecentile* - **Type:** *object*

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

Allowed values: Class: Percentage

Usage: **Common:** Required

Validation Notes: This must be greater than 0.

Property Name: *Frame Delay Range Objective*
  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: ShortDuration

Usage: **Common:** Required

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 the MEF Standards) , a time duration (referred to as Δt (the greek letter tau, not t) in the MEF Standards), a percentile (referred to as Pv in the MEF Standards), and the Inter Frame Delay Variation Objective for this Class of Service Name and these three parameters. See MEF 10.4 section 8.8.5 for EVC and MEF 26.2 section 12.13.5 for OVC.

Property Name: *Interframe Delay Variation Ordered Pairs*
  JSON Name: *orderedPairList* - **Type:** *List of object*

Description: The set (referred to as *S* in the MEF Standards) 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

Usage: **Common:** Required

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*
  JSON Name: *timeDuration* - **Type:** *object*

Description: A time duration.

Allowed values: Class: ShortDuration

Usage: **Common:** Required

Property Name: *Interframe Delay Variation Percentile*
  JSON Name: *oneWayIfdvPencentile* - **Type:** *object*

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

Allowed values: Class: Percentage

Usage: **Common:** Required

Validation Notes: This must be greater than 0.

Property Name: *Interframe Delay Variation Objective*
  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: ShortDuration

Usage: **Common:** Required

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 the MEF Standards), and the One Way Frame Loss Ratio Objective for this Class of Service Name and these Ordered End Point Pairs. See MEF 10.4 section 8.8.6 for EVC and MEF 26.2 section 12.13.6 for OVC.

Property Name: *Frame Loss Ratio Ordered Pairs*
  JSON Name: *orderedPairList* - **Type:** *List of object*

Description: The set (referred to as *S* in the MEF Standards) 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

Usage: **Common:** Required

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*
  JSON Name: *oneWayFlrObjective* - **Type:** *object*

Description: The Performance Objective for this Performance Metric (this CoS and these Ordered End Point Pairs).

Allowed values: Class: Percentage

Usage: **Common:** Required

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 the MEF Standards), and the One Way Availability Objective for this Class of Service Name and these Ordered End Point Pairs. See MEF 10.4 section 8.8.7 for EVC and MEF 26.2 section 12.13.7 for OVC.

Property Name: *Availability Ordered Pairs*
  JSON Name: *orderedPairList* - **Type:** *List of object*

Description: The set (referred to as *S* in the MEF Standards) 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

Usage: **Common:** Required

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*
  JSON Name: *oneWayAvailabilityObjective* - **Type:** *object*

Description: The PM Objective for this Performance Metric (this CoS and these Ordered End Point Pairs) expressed as a oercentage.

Allowed values: Class: Percentage

Usage: Common: Required

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 the MEF Standards), and the One Way High Loss Intervals Objective for this Class of Service Name and these Ordered End Point Pairs. See MEF 10.4 section 8.8.8 for EVC and MEF 26.2 section 12.13.8 for OVC.

Property Name: *HLI Ordered Pairs*
  JSON Name: *orderedPairList* - **Type:** *List of object*

Description: The set (referred to as *S* in the MEF Standards) 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

Usage: Common: Required

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*
  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

Usage: Common: Required

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 the MEF Standards) , a number (referred to as p in the MEF Standards), and the One Way Consecutive High Loss Intervals Objective for this Class of Service Name and these two parameters. See MEF 10.4 section 8.8.9 for EVC and MEF 26.2 section 12.13.9 for OVC.

Property Name: *CHLI Ordered Pairs*
  JSON Name: *orderedPairList* - **Type:** *List of object*

Description: The set (referred to as *S* in the MEF Standards) 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

Usage: Common: Required

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*
  JSON Name: *consecutiveNumberP* - **Type:** *integer*

Description: The number of high loss intervals that constitute a consecutive high loss interval.

Allowed values: Minimum: 1

Usage: **Common:** Required

Property Name: *CHLI Objective*
  JSON Name: *oneWayChliObjective* - **Type:** *integer*

Description: Performance Objective expressed as a non-negative integer.

Allowed values: Minimum: 1

Usage: **Common:** Required

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 the MEF Standards) , a Composite Performance number (referred to as U in the MEF Standards), indicators about Composite Frame Loss, Composite Frame Delay, and Composite Frame Delay Variation (referred to as Wfl, Wfd, and Wfdv in the MEF Standards), Thresholds for Frame Delay and Delay Variation (referred to as DL and Jt in the MEF Standards) and the One Way Composite Objective for this Class of Service Name and these seven parameters. See MEF 10.4 section 8.8.10 for EVC (this Performance Metric is not defined for OVC services).

Property Name: *Composite Ordered Pairs*
  JSON Name: *orderedPairList* - **Type:** *List of object*

Description: The set (referred to as *S* in the MEF Standards) 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

Usage: **Common:** Required

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: *Composite Performance Indicator Threshold*
  JSON Name: *compositePerformanceIndicatorThreshold* - **Type:** *number*

Description: Composite Performance Indicator threshold which if exceeded suggests an unacceptable time interval.

Allowed values: Minimum: 0 Maximum: 1

Usage: **Common:** Required

Property Name: *Composite Frame Loss*
  JSON Name: *compositeFrameLoss* - **Type:** *integer*

Description: Composite indicator for One-way Frame Loss equal to 0 or 1.

Allowed values: Minimum: 0 Maximum: 1

Usage: **Common:** Required

Property Name: *Composite Frame Delay*
  JSON Name: *compositeFrameDelay* - **Type:** *integer*

Description: Composite indicator for One-way Frame Delay equal to 0 or 1.

Allowed values: Minimum: 0 Maximum: 1

Usage: **Common:** Required

Property Name: *Composite Frame Delay Variation*
  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

Usage: **Common:** Required

Property Name: *FD Threshold*
  JSON Name: *oneWayFdThreshold* - **Type:** *object*

Description: Composite One-way Frame Delay threshold in time units.

Allowed values: Class: ShortDuration

Usage: **Common:** Required

Property Name: *IFDV Threshold*
  JSON Name: *oneWayIfdvThreshold* - **Type:** *object*

Description: One-way Frame Delay Variation threshold in time units.

Allowed values: Class: ShortDuration

Usage: **Common:** Required

Property Name: *Composite Objective*
  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: Class: Percentage

Usage: **Common:** Required

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 the MEF Standards). (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 the MEF Standards). The third element is the Performance Objective for the One Way Group Availability. See MEF 10.4 section 8.8.11 for EVC and MEF 26.2 section 12.13.10 for OVC.

Property Name: *GA Ordered Pairs*
  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

Usage: **Common:** Required

Property Name: *Minimum Number of Sets Available*
  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

Usage: **Common:** Required

Property Name: *GA Objective*
  JSON Name: *oneWayGroupAvailabilityObjective* - **Type:** *object*

Description: This is the Performance Objective for this Performance Metric expressed as a Percentage.

Allowed values: Class: Percentage

Usage: Common: Required

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 End Point*
  JSON Name: *fromCarrierEthernetServiceEndPoint* - **Type:** *string*

Description: This is the EVC End Point ID for and EVC End Point in the service or OVC End Point ID for an OVC End Point in the service.

Allowed values: [Any String]

Usage: Common: Required

Property Name: *To EVC End Point*
  JSON Name: *toCarrierEthernetServiceEndPoint* - **Type:** *string*

Description: This is the EVC End Point ID for and EVC End Point in the service or OVC End Point ID for an OVC End Point in the service.

Allowed values: [Any String]

Usage: Common: Required

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

Usage: Common: Required

736

---

737 ## 13.10   Utility Classes and Types

Utility Classes and Types
   Schema File Name: utilityClassesAndTypes
   Includes classes:
      MepLevelAndDirection (MEP Specification)
      SourceMacAddressLimit (SA Limit)
      AggLinkDepth (Link Depth)
      FrameDisposition (Frame Disposition)
      FrameDispEnum (Frame Disposition Enumeration)
      EnabledDisabled (Enabled Disabled)
      InformationRate (Information Rate)
      DataSize (Data Size)
      ShortDuration (Short Duration)
      LongDuration (Long Duration)
      Time (Time and Date)
      Percentage (Percentage)
      PcpValueAndDiscard (PCP and Discard Enum)
      DeiValueAndDiscard (DEI and Discard Enum)

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

Usage: **Common:** Required

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"

Usage: **Common:** Required

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

Usage: **Common:** Required

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: ShortDuration

Usage: Common: Required

Class: AggLinkDepth **(Link Depth)**

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

Usage: Common: Required

Property Name: *linkDepth*
  JSON Name: *lagDepth* - **Type:** *integer*

Description: The number of Link Aggregation links available to the VLAN

Allowed values: Minimum: 1

Usage: Common: Required

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: Class: FrameDispEnum

Usage: Common: Required

Property Name: *Multicast Frames*
  JSON Name: *multicast* - **Type:** *object*

Allowed values: Class: FrameDispEnum

Usage: Common: Required

Property Name: *Broadcast Frames*
  JSON Name: *broadcast* - **Type:** *object*

Allowed values: Class: FrameDispEnum

Usage: Common: Required

Type: FrameDispEnum *(Frame Disposition Enumeration)*

Property Name: *frameDispEnum* **Type:** *enum*
Allowed values: Enum: "DISCARD", "DELIVER_UNCONDITIONALLY", "DELIVER_CONDITIONALLY"
Usage: Common: Optional

Type: EnabledDisabled *(Enabled Disabled)*

Enumeration to indicate Enabled/Disabled state of an attribute

Property Name: *Enabled Disabled Enum* **Type:** *enum*
Allowed values: Enum: "ENABLED", "DISABLED"
Usage: Common: Optional

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
Usage: Common: Required

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 70 kbps this element is kbps.
Allowed values: Enum: "BPS", "KBPS", "MBPS", "GBPS", "TBPS", "PBPS", "EBPS", "ZBPS", "YBPS"
Usage: Common: Required

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]
Usage: Common: Required

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.

Allowed values: Enum: "BYTES", "KBYTES", "MBYTES", "GBYTES", "TBYTES", "PBYTES", "EBYTES", "ZBYTES", "YBYTES"

Usage: Common: Required

Class: ShortDuration ***(Short Duration)***

This class is used to describe durations expressed in minutes or smaller time units

Property Name: *Short Duration Value*
  JSON Name: *shortDurationValue* - **Type:** *integer*

Description: The value of the duration. For example, if the duration is 20 ms, this element is 20.

Allowed values: [Any Integer]

Usage: Common: Required

Property Name: *Short Duration Units*
  JSON Name: *shortDurationUnits* - **Type:** *enum*

Description: The unit of measure in the duration. For example, if an interval is 2ms, this element is MS.

Allowed values: Enum: "NS", "US", "MS", "SEC", "MIN"

Usage: Common: Required

Class: LongDuration ***(Long Duration)***

This class is used to describe durations expressed in days or larger time units

Property Name: *Long Duration Value*
  JSON Name: *longDurationValue* - **Type:** *integer*

Description: The value of the duration. For example, if the duration is 20 days, this element is 20.

Allowed values: [Any Integer]

Usage: Common: Required

Property Name: *Long Duration Units*
  JSON Name: *longDurationUnits* - **Type:** *enum*

Description: The unit of measure in the duration. For example, if an interval is 2 weeks, this element is WEEKS.

Allowed values: Enum: "DAY", "WEEK", "MONTH", "YEAR"

Usage: Common: Required

Class: Time *(Time and Date)*

Property Name: *Year*
  JSON Name: *year* - **Type:** *integer*

Allowed values: Minimum: 1900

Usage:　　**Common:** Required

Property Name: *Month*
  JSON Name: *month* - **Type:** *integer*

Allowed values: Minimum: 1 Maximum: 12

Usage:　　**Common:** Required

Property Name: *Day*
  JSON Name: *day* - **Type:** *integer*

Allowed values: Minimum: 1 Maximum: 31

Usage:　　**Common:** Required

Property Name: *Hour*
  JSON Name: *hour* - **Type:** *integer*

Allowed values: Minimum: 0 Maximum: 23

Usage:　　**Common:** Required

Property Name: *Minute*
  JSON Name: *minute* - **Type:** *integer*

Allowed values: Minimum: 0 Maximum: 59

Usage:　　**Common:** Required

Property Name: *Second*
  JSON Name: *second* - **Type:** *integer*

Allowed values: Minimum: 0 Maximum: 59

Usage:　　**Common:** Required

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

Usage:　　**Common:** Optional

Type: PcpValueAndDiscard *(PCP and Discard Enum)*

Property Name: *PCP Value and Discard* **Type:** *enum*

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

Usage: **Common:** Optional

Type: DeiValueAndDiscard *(DEI and Discard Enum)*

Property Name: *DEI Value and Discard* **Type:** *enum*

Allowed values: Enum: "0", "1", "DISCARD"

Usage: **Common:** Optional

752

753 # 14 Service Attributes Not Included

754 There are 8 Services Attributes from MEF 10.4 that are not included in the API. The two primary
755 reasons for not including them are:

756 - They are part of the product agnostic envelope (i.e., several Identifiers), and
757 - The Service Attribute has a single constant value for Subscriber Ethernet. There is no
758   benefit to including these in the API since the only possible use would be to validate
759   that they are correct. There is no information to be gleaned if they are correct.

760 These Service Attributes are listed in the following table.

761

Group	Service Attribute	Discussion
EVC Service Attribute	EVC Identifier	Identifiers for Order Items are not needed since they are in the product-agnostic envelope.
EVC Service Attribute	EVC Type	EVC Type is not included, because it is implicit in the service type. For example, when ordering an E-TREE service, the EVC Type is known to be rooted multipoint and as such there is no value in specifying it.
EVC Service Attribute	List of EVC End Points	This is a list of EVC End Point IDs. This is not needed since the EVC End Points are included in the EVC definition.
EVC Service Attribute	EVC Group Membership	Out of scope for this release. May be included in a future release of the schema.
EVC End Point Service Attributes	EVC End Point Ingress Bandwidth Profile	Bandwidth Profiles must be none as specified in MEF 6.3 R4.
EVC End Point Service Attributes	EVC End Point Egress Bandwidth Profile	Bandwidth Profiles must be none as specified in MEF 6.3 R5.
Subscriber UNI Service Attributes	Subscriber UNI Identifier	Identifiers for Order Items are not needed since they are in the product-agnostic envelope.
Subscriber UNI Service Attributes	Subscriber UNI Service Frame Format	Not needed, only a single value allowed: "802.3-2012"

762 **Table 8 – Service Attributes Not Included in Schemas**

763

764   **15   Subscriber Ethernet Envelope and MEF Payload Association**

765   The following section provides an overview of how the two components of the Cantata/Sonata
766   API are associated from an implementation perspective. Specifically covered is the binding be-
767   tween the envelope and payload.

768   **15.1   Product Order Sequence Diagram**

769   This section provides developers with guidance on how to associate the envelope part of the Can-
770   tata/Sonata API with the Subscriber Ethernet payload. A sequence diagram is used to illustrate the
771   pseudo steps needed to create the payload, associate to the envelope and POST the request.

772   The envelope and payload are associated with the Order and specifically the OrderItem. The en-
773   velope OrderItem is associated with a payload resource such as carrierEthernetSubscriberUni.

774   The Figure 10 below illustrates the sequencing of coding operations used by a Subscriber in build-
775   ing the Cantata/Sonata Envelope and associated set of Subscriber Ethernet specific Order Items.
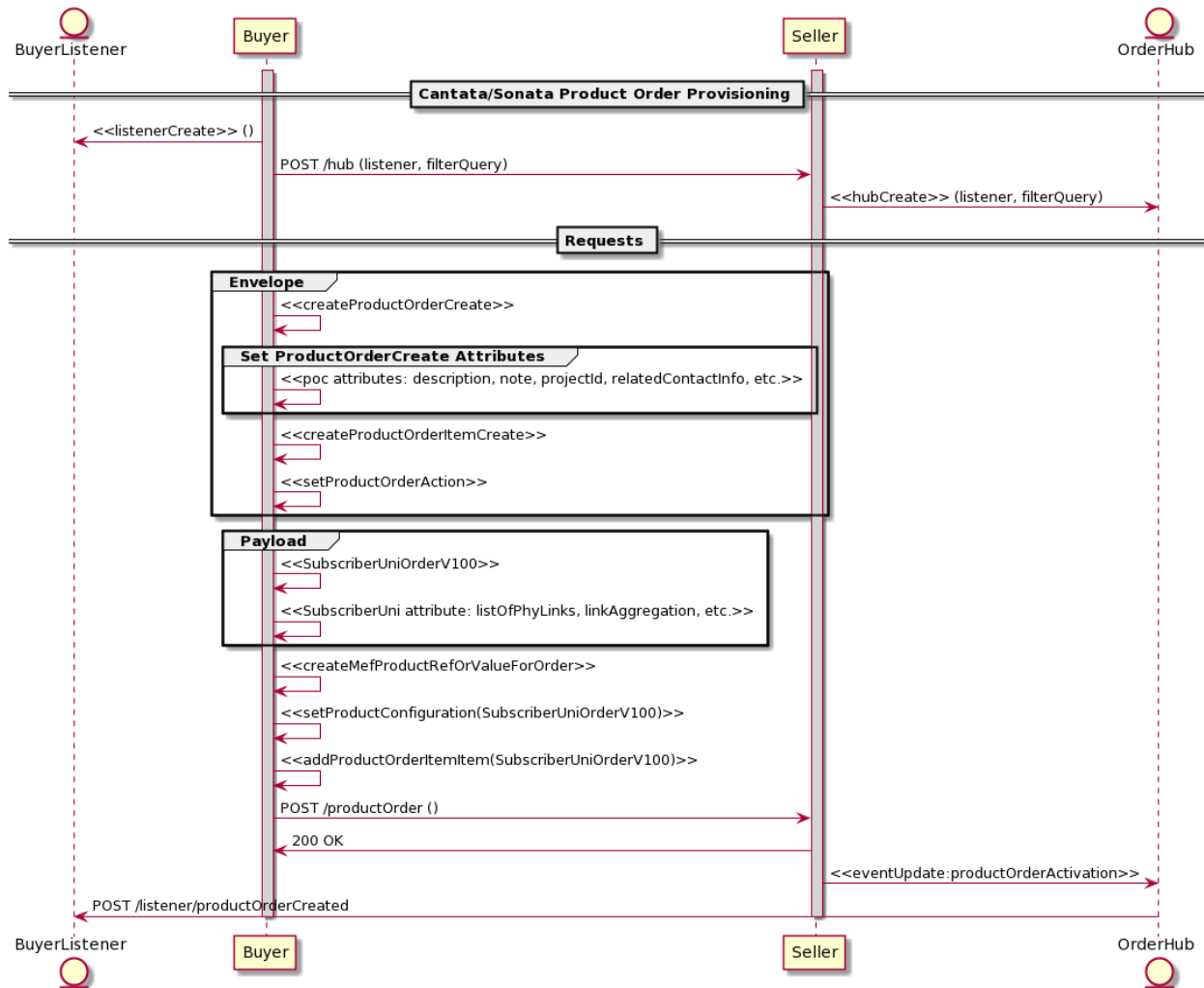776   Once the Order is complete it is sent to the Service Provider (Seller) with a POST message.

777

Figure 10 – Subscriber Ethernet Product Order Flow

779

16 References

[1] IEEE Std 802.1AX-2014, *Link Aggregation*, December 2014

[2] IEEE Std 802.3-2012, *IEEE Standard for Ethernet*, August 2012

[3] MEF 4, *Metro Ethernet Network Architecture - Part 1: Generic Framework*, May 2004

[4] MEF 6.3, *Subscriber Ethernet Services Definitions*, November 2019

[5] MEF 7.4, *Carrier Ethernet Services Information Model*, December 2020

[6] MEF 10.4, *Subscriber Ethernet Service Attributes*, December 2018

[7] MEF 26.2, *External Network Network Interfaces (ENNI) and Operator Service Attributes*, August 2016

[8] MEF 45.1, *Layer 2 Control Protocols in Ethernet Services*, December 2018

[9] MEF 55.1, *Lifecycle Service Orchestration (LSO): Reference Architecture and Framework*, January 2021

[10] MEF 57.1, *Ethernet Ordering Technical Specification, Business Requirements and Use Cases*, December 2018

[11] MEF 78.1, *MEF Core Model*, July 2020

[12] MEF 79, *Address, Service Site, and Product Offering Qualification Management, Requirements and Use Cases*, November 2019

[13] MEF 80, *Quote Management, Requirements and Use Cases - Draft Standard*, December 2020

[14] MEF 81, *Product Inventory* Management*, Requirements and Use Cases*, November 2019