



Working Draft
MEF W125 v1.12

LSO Cantata and LSO Sonata - Subscriber Ethernet
Product Schemas and Developer Guide

June 2022

**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.

22 Disclaimer

23 © MEF Forum 2022. All Rights Reserved.

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

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

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

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

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

Table of Contents

1	List of Contributing Members	1
2	Abstract	2
3	Terminology and Abbreviations	3
4	Compliance Levels	7
5	Numerical Prefixes	8
6	Introduction	9
7	Overview of LSO Cantata and LSO Sonata	11
8	Overview of Subscriber Ethernet Services	13
8.1	EVCs	13
8.2	EPL, EVPL, EP-LAN, EVP-LAN, EP-TREE, EVP-TREE	14
9	Data Model Design Principles and Assumptions	17
9.1	Mandatory Service Attributes	18
9.2	Optional Service Attributes	18
9.3	Fixed Service Attributes	18
10	Order Milestones	20
11	Information Model for Subscriber Ethernet Product Data Model	21
11.1	Organization of Service Attributes	21
12	Data Models for Subscriber Ethernet Products	26
12.1	Organization and Structure of the Schemas	26
12.2	Service Attribute Groupings	27
12.3	Example Schema	28
12.4	Additional Details	29
12.4.1	Naming Conventions	29
12.4.2	EVC End Point Service Attributes	30
12.4.3	Additional EVC End Point Attributes	30
13	Relationship Between Entities	31
14	Subscriber Ethernet Service Attributes	34
14.1	EVC Groups	34
14.1.1	EVC and EVC EP Group – Common	34
14.1.2	EVC EP Group – EPL	37
14.1.3	EVC and EVC EP Group – Not EPL	37
14.1.4	EVC and EVC EP Group – Virtual	39
14.2	Subscriber Ethernet EVC and EVC End Point Service Attributes	41
14.2.1	EPL Service Attributes	41
14.2.2	EVPL Service Attributes	42
14.2.3	EP-LAN Service Attributes	43
14.2.4	EVP-LAN Service Attributes	44
14.2.5	EP-TREE Service Attributes	45
14.2.6	EVP-TREE Service Attributes	46



91	14.3	Subscriber Ethernet UNI Service Attributes	47
92	15	Common Classes and Types.....	51
93	15.1	Bandwidth Profiles	51
94	15.2	Class of Service	55
95	15.3	Color Identifier	60
96	15.4	Egress Maps.....	62
97	15.5	End Point Maps	67
98	15.6	External Interfaces	70
99	15.7	Layer 2 Control Protocols (L2CP).....	73
100	15.8	Link Aggregation.....	74
101	15.9	Service Level Specification	75
102	15.10	Carrier Ethernet Standard Types	86
103	15.11	Carrier Ethernet Utility Classes	88
104	16	Service Attributes Not Included.....	91
105	17	References	92
106			

List of Figures

107	
108	Figure 1 – LSO Cantata and LSO Sonata Reference Diagram 11
109	Figure 2 – LSO Cantata and LSO Sonata API Structure 12
110	Figure 3 – E-Line Service Type..... 14
111	Figure 4 – E-LAN Service Type..... 15
112	Figure 5 – E-Tree Service Type..... 15
113	Figure 6 – Components of a Subscriber Ethernet Service 21
114	Figure 7 – Subscriber Ethernet Superclasses 22
115	Figure 8 – EPL and EVPL 23
116	Figure 9 – EP-LAN, EVP-LAN, EP-TREE and EVP-TREE 24
117	Figure 10 – Schema File Organization 26
118	Figure 11 – Relationships for EPL and EVPL..... 33
119	Figure 12 – Relationships for EP-LAN, EVP-LAN, EP-TREE and EVP-TREE..... 33
120	

List of Tables

121	
122	Table 1 – Terminology and Abbreviations 6
123	Table 2 – Numerical Prefix Conventions..... 8
124	Table 3 – MEF 6.3 Compliant Subscriber Ethernet Services 13
125	Table 4 – Order Milestones for Subscriber Ethernet 20
126	Table 5 – Service Attribute Groupings 28
127	Table 6 – Product Relationship Roles..... 31
128	Table 7 – Place Relationship Role 32
129	Table 8 – Service Attributes Not Included in Schemas 91

130

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](#)

2 Abstract

The MEF Standard consisting of this schema guide and its associated software artifacts (JSON Schemas) defines and describes the product-specific information used in LSO Cantata and LSO Sonata APIs for a set of Business Functions - specifically, Product Offering Qualification, Quote, Order and Inventory, for Subscriber Ethernet Services conforming to MEF 6.3 [7]. The document starts with an overview of LSO Cantata, LSO Sonata and Subscriber Ethernet Services. It then provides a basic information model for the MEF Subscriber Ethernet Service Attributes. The final sections describe the Data Model focused on the JSON Schemas associated with this specification.

This document can be thought of as a user's guide for the Subscriber Ethernet Data Model and the schemas provided that embody the Data Model. MEF Services are described by a set of Service Attributes. Each Service Attribute describes an aspect of the service that is agreed between the provider and the user of the service. The document that describes the Service Attributes for Subscriber Ethernet Services is MEF 10.4 [9]. The Services, themselves, are specified in MEF 6.3 based on the Service Attributes defined in MEF 10.4.

MEF 10.4 includes 11 EVC Service Attributes, 13 EVC End Point and 17 Subscriber UNI Service Attributes to describe the various components that compose a Subscriber Ethernet Service. This document defines a data model that includes 33 of these Service Attributes and also lists the 8 Service Attributes that are not included in the data model and the reason why each is not included. It also describes how they are organized across the various schemas to support the different types of Subscriber Ethernet Services.

3 Terminology and Abbreviations

This section defines the terms used in this document. In many cases, the normative definitions to terms are found in other documents. In these cases, the third column is used to provide the reference that is controlling, in other MEF or external documents.

Term	Definition	Reference
Business Functions	In the context of this document Business Functions refer to <i>Product Offering Qualification (POQ)</i> , <i>Order Management</i> , <i>Quote Management</i> , and <i>Inventory Management</i> .	This Document
Buyer	In the context of this document, a Buyer is the organization acting as the Subscriber who is ordering from a Service Provider (aka, Seller).	MEF 57.2 [13]
Cantata	The Management Interface Reference Point supporting the management and operations interactions (e.g., quote, order, billing, etc.) between a Service Provider (Seller) and a Subscriber(Buyer).	MEF 55.1 [12]
Data Model	A representation of concepts of interest to an environment in a form that is dependent on data repository, data definition language, query language, implementation language, and/or protocol (typically, but not necessarily, all five).	IETF RFC 3444 [4]
External Network Network Interface	A reference point representing the boundary between two Operator Carrier Ethernet Networks that are operated as separate administrative domains.	MEF 26.2 [10]
ENNI	External Network Network Interface	MEF 26.2 [10]
EPL	Ethernet Private Line	MEF 6.3 [7]
EP-LAN	Ethernet Private LAN	MEF 6.3 [7]
EP-Tree	Ethernet Private Tree	MEF 6.3 [7]
Ethernet Private Line	A Port-based Service of E-Line Service Type.	MEF 6.3 [7]
Ethernet Private LAN	A Port-based Service of E-LAN Service Type.	MEF 6.3 [7]
Ethernet Private Tree	A Port-based Service of E-Tree Service Type.	MEF 6.3 [7]
Ethernet Service	A connectivity service that carries Ethernet Frames irrespective of the underlying technology and that is specified using Service Attributes as defined in an MEF Specification.	MEF 10.4 [9]
Ethernet Subscriber	The end-user of an Ethernet Service.	MEF 10.4 [9]
Ethernet UNI	The demarcation point between the responsibility of the Service Provider and the Subscriber for an Ethernet Service.	MEF 10.4 [9]
Ethernet Virtual Connection	An association of EVC End Points.	MEF 10.4 [9]

Term	Definition	Reference
Ethernet Virtual Private Line	A VLAN-based Service of E-Line Service Type.	MEF 6.3 [7]
Ethernet Virtual Private LAN	A VLAN-based Service of E-LAN Service Type.	MEF 6.3 [7]
Ethernet Virtual Private Tree	A VLAN-based Service of E-Tree Service Type.	MEF 6.3 [7]
EVC	Ethernet Virtual Connection	MEF 10.4 [9]
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 [9]
EVPL	Ethernet Virtual Private Line	MEF 6.3 [7]
EVP-LAN	Ethernet Virtual Private LAN	MEF 6.3 [7]
EVP-Tree	Ethernet Virtual Private Tree	MEF 6.3 [7]
External Interface	In the context of this document, a UNI.	MEF 4 [6]
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.	IETF RFC 3444 [4]
Inventory	The inventory managed by the Seller resulting from Order completion.	MEF 81 [17]
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 [10]
Order	One or more Product Order Items formulated into a fulfillment request made by a Buyer to a Seller.	MEF 57.2 [13]
Port-based Service	Any service with EVC EP Map Service Attribute value equal to <i>All</i> for each EVC EP in the EVC.	MEF 6.3 [7]
Product	One or more goods or services that is or may be sold to a Buyer by a Seller.	MEF 79 [15]
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 [15]
Product Order Item	An individual item included in a Product Order that describes the action to be taken on a Product or Product Offering by the Seller. The	This Document

Term	Definition	Reference
	objective is for the Seller to complete the fulfillment process of this Product or Product Offering at the place defined by the Buyer.	
Product Specification	A Product Specification defines the template or detailed description from which Product Offerings can be defined.	MEF 79 [15]
POQ	Product Offering Qualification	MEF 79 [15]
Product Offering Qualification	One or more Product Offering Qualification Items formulated into a requirement made by a Buyer to a Seller.	MEF 79 [15]
Product Offering Qualification Item	An individual article included in a POQ that describes a product of a particular type (product offering). The objective is to determine if it is feasible for the Seller to deliver this item as described and for the Seller to inform the Buyer of the estimated time interval to complete this delivery.	MEF 79 [15]
Quote	One or more Quote Items formulated into a request for pricing of a Product or Product Offering made by a Buyer to a Seller.	MEF 80 [16]
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 [16]
Seller	In the context of this document, a Seller is the Service Provider who is providing the product to the Buyer.	MEF 57.2 [13]
Service Attribute	Specific information that is agreed between the provider and the user of the service, that describes some aspect of the service behavior or capability.	MEF 10.4 [9]
Service Frame	An Ethernet Frame that is exchanged across a UNI between a Subscriber and a Service Provider.	MEF 10.4 [9]
Service Provider	In the context of this document, a Service Provider is an organization that provides a Subscriber Ethernet Service to Subscribers.	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 [12]

Term	Definition	Reference
Subscriber	In the context of this document, a Subscriber is an Ethernet Subscriber.	This Document
Subscriber Ethernet Service	In the context of this document, an Ethernet Service conforming to one of the six Subscriber Ethernet Services defined in MEF 6.3.	This Document
Subscriber UNI Service Attributes	The Service Attributes used to describe an agreement between a Subscriber and a Service Provider for a UNI as part of a Subscriber Ethernet Service.	MEF 10.4 [9]
UNI	In the context of this document, a UNI means an Ethernet UNI.	MEF 10.4 [9]
VLAN-based Service	Any service with the EVC EP Map value equal to <i>UT/PT</i> or <i>List</i> for each EVC EP in the EVC.	MEF 6.3 [7]

Table 1 – Terminology and Abbreviations

4 Compliance Levels

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this specification are to be interpreted as described in BCP 14 (RFC 2119 [3], RFC 8174 [5]) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

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

5 Numerical Prefixes

This document uses the prefix notation to indicate multiplier values as shown in Table 2.

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

Table 2 – Numerical Prefix Conventions

6 Introduction

LSO Cantata provides a programmatic interface for establishing an automated exchange of information between business applications (Quote, Order, etc.) between a Service Provider (Seller) and a Subscriber (Buyer). LSO Sonata provides a similar programmatic interface between two network providers, a Service Provider (Seller) and Partner (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., Quote, 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 [9]. The Service definition which is, in effect, a set of constraints on the values of the Service Attributes is provided MEF 6.3 [7].

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, Quote, Order, and 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 7)
- An overview of Subscriber Ethernet Services (section 8)
- Data Model Design Principles (section 9)
- Order Milestones (section 10)
- An abbreviated Information Model for Subscriber Ethernet and explanation of the organization of the Service Attributes in MEF 10.4 (section 11)
- Organization of the data model for Subscriber Ethernet (section 12)
- The relationship between the entities in the service (section 13)

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

- Section 14 contains the details of the Subscriber Ethernet Service Attributes for EVC, EVC End Point, and UNI

- 215 • Section 15 contains all of the common classes and types referenced by the Service
216 Attributes
- 217 • Section 16 lists the Service Attributes that are not included in the data models

7 Overview of LSO Cantata and LSO Sonata

MEF 55.1 [12] describes the Reference Architecture for Lifecycle Service Orchestration (LSO) of MEF-defined services. MEF 55.1 defines seven LSO Interface Reference Points (see Figure 1) that are abstract 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 Cantata which defines the abstract interconnection point between a Subscriber (Buyer) and a Service Provider (Seller) and another is LSO Sonata which defines the abstract interconnection point between a Service Provider (Buyer) and an Operator (Seller). It is at these Interface Reference Points – LSO Cantata and LSO Sonata – that the Buyer and the Seller interact to orchestrate business transactions for the different Business Functions. Examples of inter-provider Business Functions include address qualification, product offering qualification, quote, ordering, trouble ticketing, and billing/settlement management.

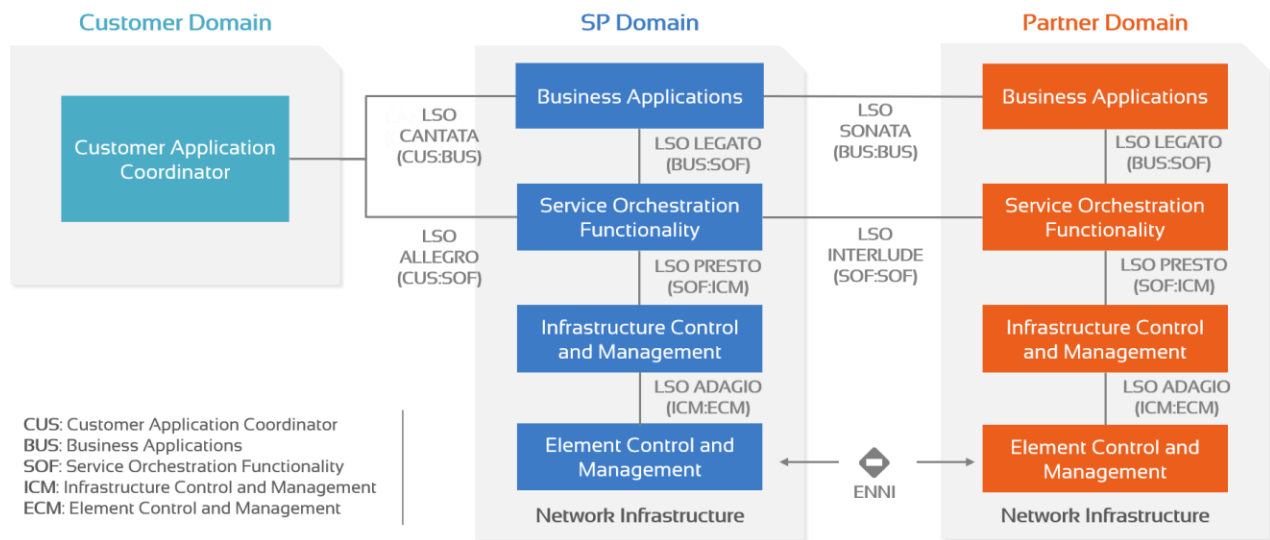


Figure 1 – LSO Cantata and LSO Sonata Reference Diagram

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 of product-specific data models, as shown in Figure 2.

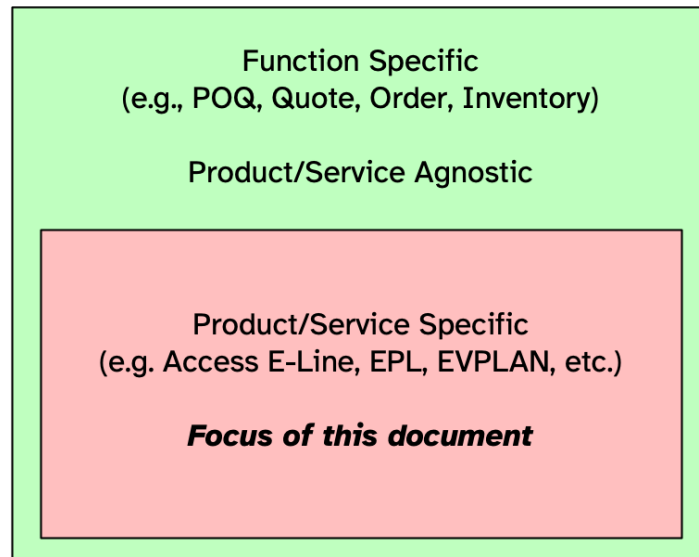


Figure 2 – LSO Cantata and LSO Sonata API Structure

This document describes the product specific data model for a MEF Subscriber Ethernet Services as defined in MEF 6.3 [7].

8 Overview of Subscriber Ethernet Services

This specification describes a data model for MEF-defined Subscriber Ethernet Services. 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) defined in MEF 6.3 [7]. It starts with a brief explanation of EVCs.

8.1 EVCs

A Subscriber Ethernet Service is built on an Ethernet Virtual Connection (EVC) which is an association of two or more EVC End Points. An EVC End Point is a construct located at a UNI that represents the logical attachment of an EVC to a UNI and selects a 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 [9]. 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 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 EVC Service Type constructs, resulting in a total of six Subscriber Ethernet Services that are compliant with MEF 6.3. 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 Subscriber Ethernet Services defined in MEF 6.3 are shown in Table 3.

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)

Table 3 – MEF 6.3 Compliant Subscriber Ethernet Services

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

8.2 EPL, EVPL, EP-LAN, EVP-LAN, EP-TREE, EVP-TREE

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

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

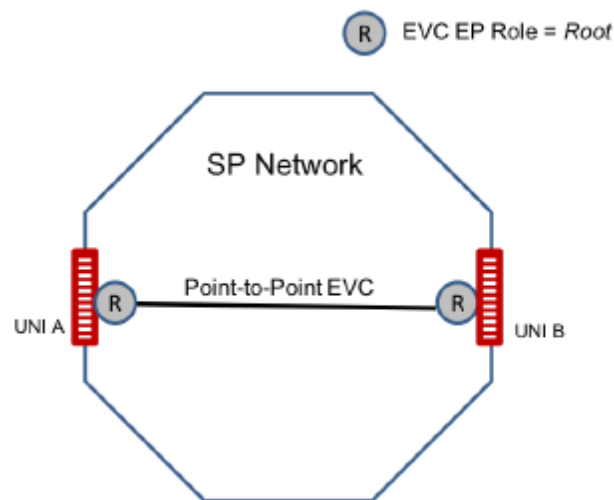


Figure 3 – E-Line Service Type

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

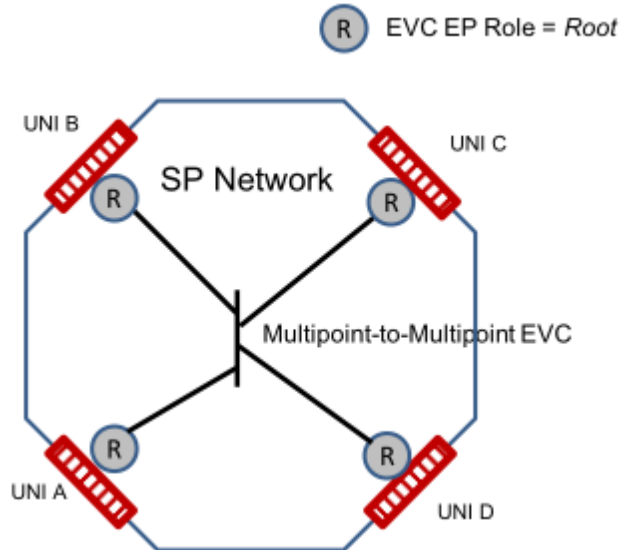


Figure 4 – E-LAN Service Type

Any Ethernet Service that has the EVC Type Service Attribute equal to **Rooted-Multipoint** is classified as an E-Tree Service Type (EP-TREE, EVP-TREE). An E-Tree service is a highly transparent service that enables Subscribers with multiple sites to interconnect them in a way that does not resemble a LAN. The sites can be connected with a centralized site (or a few core sites), which are designated as a Root and all remaining sites designated as Leaf.

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

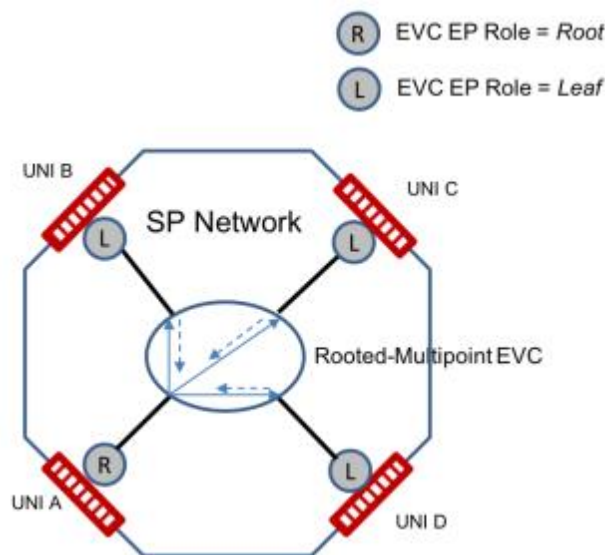


Figure 5 – E-Tree Service Type

305 The LSO Cantata and LSO Sonata API with the Subscriber Ethernet product configuration as de-
306 scribed in this specification allows the Business Functions associated with purchasing these prod-
307 ucts (Product Offering Qualification, Quote, Order, and Inventory) to be performed programmat-
308 ically.

9 Data Model Design Principles and Assumptions

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

- None of the Service Attributes included in the schemas are coded as “Required”.
- Each Seller will divide all Service Attributes included in the schemas into one of three categories for each Business Function:
 - Mandatory - attributes that must be provided by the Buyer in a POQ/Quote/Order request (see section 9.1)
 - Optional - attributes that may be provided by the Buyer in a POQ/Quote/Order request (see section 9.2)
 - Fixed - attributes that are hard coded by the Seller and may not be specified by the Buyer in a POQ/Quote/Order request (see section 9.3)

[R1] The Seller and Buyer **MUST** agree, for each Service Attribute, whether the Service Attribute is Mandatory, Optional or Fixed for each Business Function for a given Product Offering.

The Service Attribute categorization can be defined and negotiated during the onboarding process or defined in a Product Catalog.

[R2] If a Service Attribute is categorized as Optional for a Business Function for a Product Offering, the Seller and Buyer **MUST** agree on a default value for the Service Attribute.

The categorization may depend on:

- Business Function - a given Service Attribute may, for example, be classified by the Seller as Fixed for the Create POQ request; while it may be considered as Mandatory by the Seller for the Create Order request.
- Product Action - a given Service Attribute may, for example, be classified as Mandatory by the Seller for the Create POQ request for an INSTALL of a new Product, while it may be considered as Fixed for the Create POQ request for a MODIFY of an installed Product.
- Product Offering - a given Service Attribute may, for example, be classified as Mandatory by the Seller for the Create POQ request for a Product Offering (e.g., Premium Service), while it may be considered as Fixed for the Create POQ request for a different Product Offering (e.g., Basic Service).

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

The Subscriber Ethernet data model supports both INSTALL and CHANGE actions for POQ, Quote, and Order for the EVC and Subscriber UNI. Note that the DISCONNECT action does not require support by the data model.

The Subscriber Ethernet data model supports the RETRIEVE action for Inventory for the EVC and Subscriber UNI.

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.

9.1 Mandatory Service Attributes

[R4] If a Service Attribute is agreed to be Mandatory for an Business Function, Product Action, and Product Offering, then the Buyer **MUST** include a value for the Service Attribute in the corresponding API request.

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

9.2 Optional Service Attributes

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

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

9.3 Fixed Service Attributes

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

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

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

- 373 **[R7]** The Buyer **MUST NOT** submit an API request to the Seller which has a value
374 other than the one applicable value for a Service Attribute that has been cate-
375 gorized as Fixed for the Business Function, Product Action, and Product Of-
376 fering.
- 377 **[R8]** The Seller **MUST** reject any API request from the Buyer if it has a value other
378 than the one applicable value for a Service Attribute that has been categorized
379 as Fixed for the Business Function, Product Action, and Product Offering.

10 Order Milestones

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

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

Table 4 – Order Milestones for Subscriber Ethernet

The Milestone Value in the first column of Table 4 is included in *ProductOrderEventPayload.milestoneName* in the Product Order Notification API (see MEF 123 [21]).

11 Information Model for Subscriber Ethernet Product Data Model

Subscriber Ethernet Services are composed of three primary classes of objects: The Virtual Connection (EVC), the UNI, and the EVC End Points. This product specification provides a data model for the Service Attributes for these three components.

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

All of these elements are shown in the following diagram.

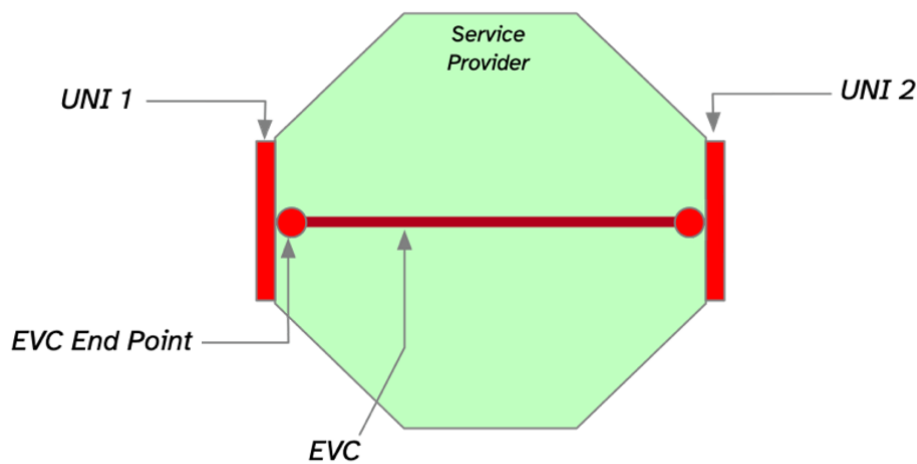


Figure 6 – Components of a Subscriber Ethernet Service

In this diagram, the Service Provider (large green octagon) is providing an EVC between UNI 1 and UNI 2. The EVC End Points (small red circles) are each located at the corresponding UNI.

11.1 Organization of Service Attributes

A Subscriber Ethernet Service is defined as having an EVC and two or more UNIs. The EVC will have an association with two or more EVC End Points. Each EVC End Point is associated with a single UNI (as shown in Figure 6). The Subscriber Ethernet configuration is based on the Service Attributes documented in MEF 10.4[9]. The MEF Services Model (MSM) for Subscriber Ethernet Services described in MEF 7.4 [8] defines three superclasses that contain all the common service attributes for the EVC, EVC End Point and Subscriber UNI. These superclasses (CarrierEthernetService, CarrierEthernetExternalInterface and CarrierEthernetServiceEndPoint) are shown in green in the following Figure 7 along with their sub-classes. The sub-classes relevant to Subscriber

Ethernet (CarrierEthernetSvc, CarrierEthernetSubscriberUni and CarrierEthernetSvcEndPoint) are shown in the colored boxes. Note: the EVC Group Membership Service Attribute is out of scope for this release, because the Multiple EVC Service Level Specification Service Attribute is not currently part of MEF 7.4 [8].

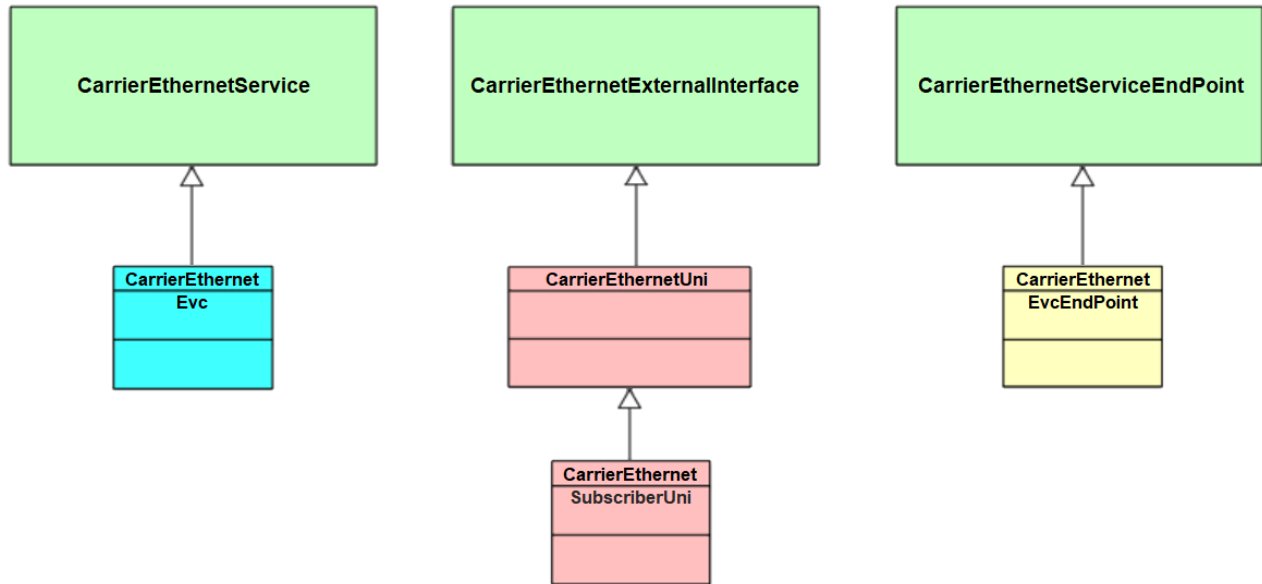


Figure 7 – Subscriber Ethernet Superclasses

In the context of Subscriber Ethernet Service, and the Service Attributes in MEF 6.3 [7] and MEF 10.4 [9], these components are organized as shown in Figure 8 and Figure 9 below (the items in the blue, yellow and red boxes match the colors of the corresponding sub-classes in Figure 7):

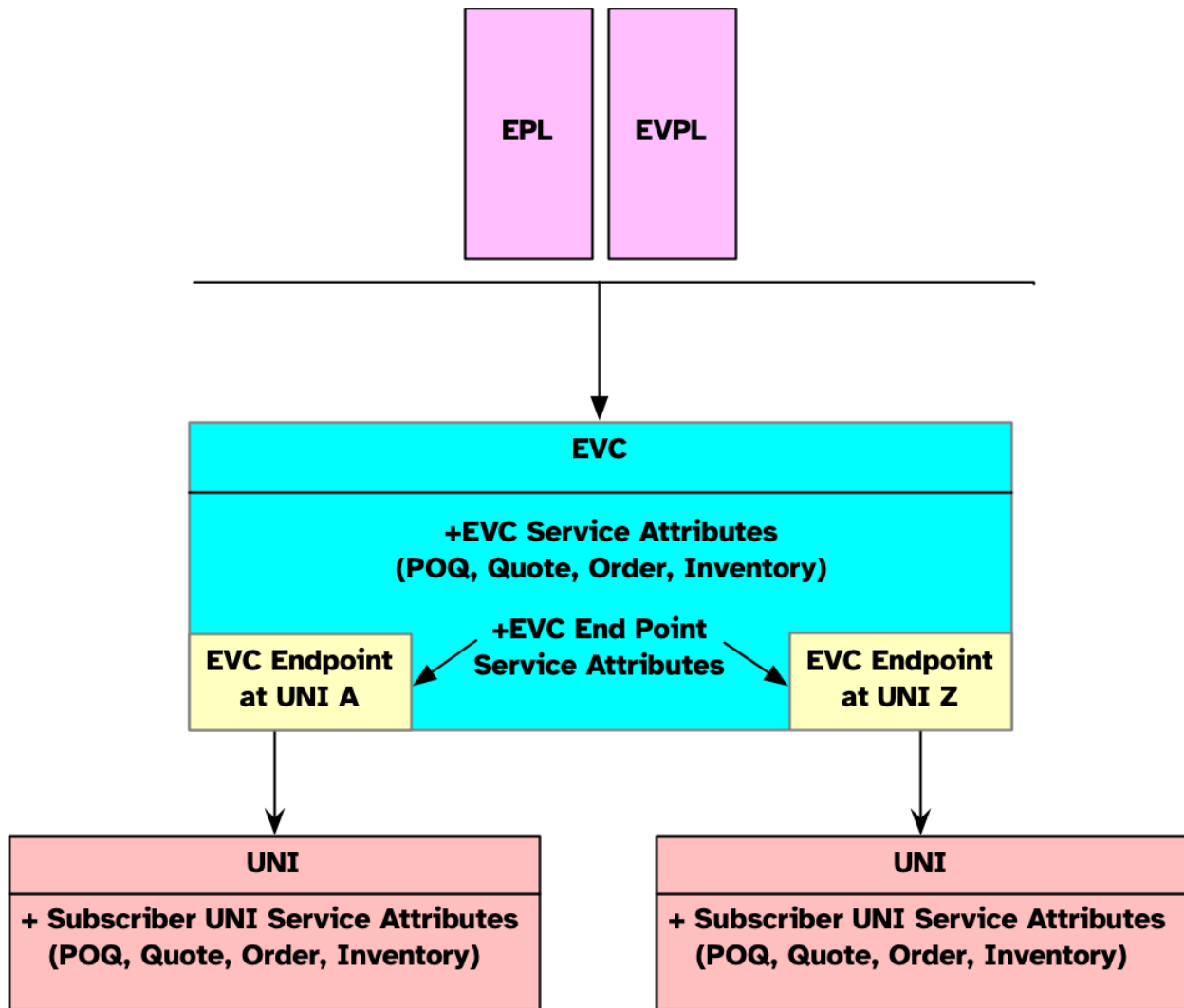


Figure 8 – EPL and EVPL

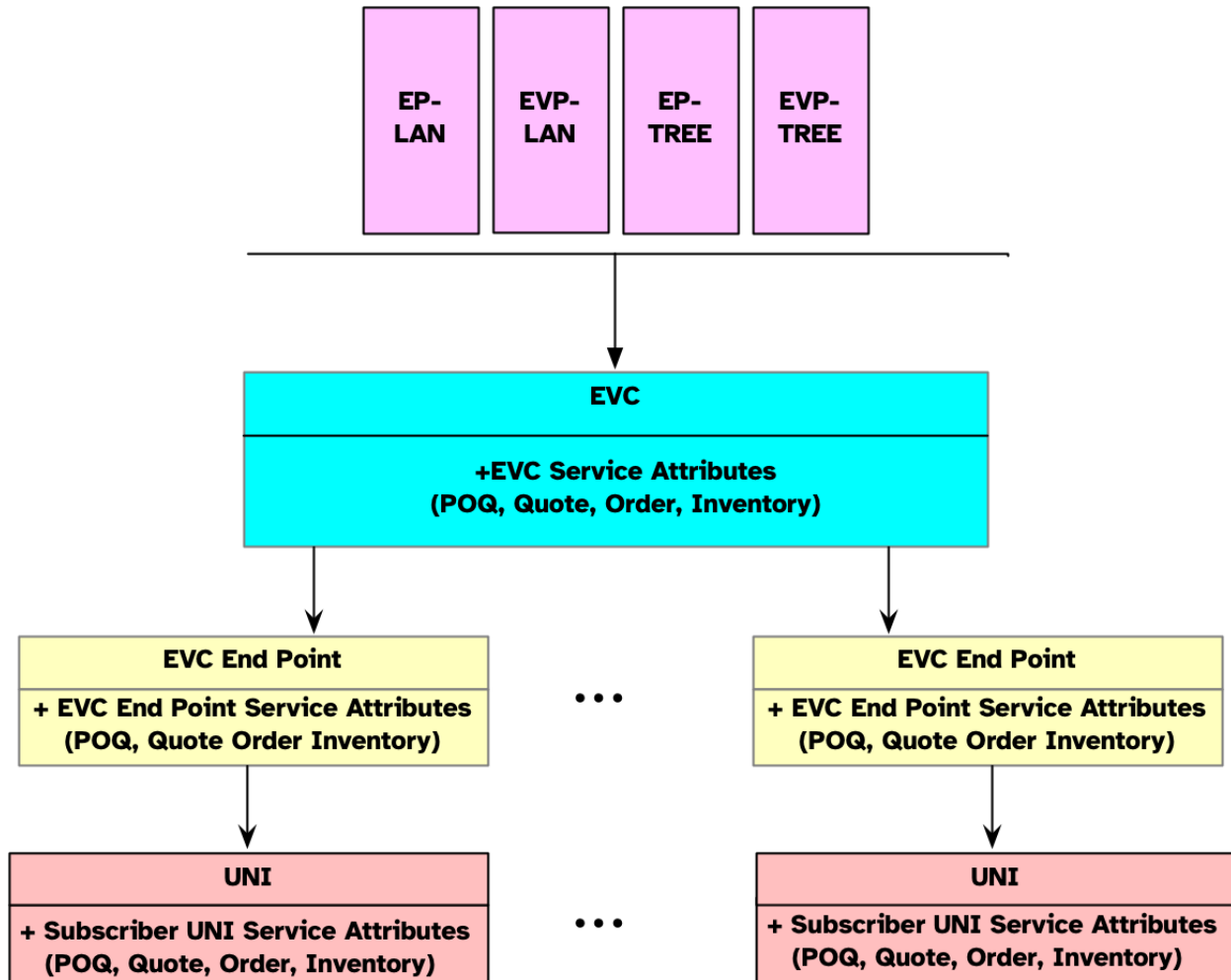


Figure 9 – EP-LAN, EVP-LAN, EP-TREE and EVP-TREE

The Subscriber Ethernet products defined in this specification are:

- EPL EVC (section 14.2.1)
- EVPL EVC (section 14.2.2)
- EP-LAN EVC (section 14.2.3)
- EP-LAN EVC EP (section 14.2.3)
- EVP-LAN EVC (section 14.2.4)
- EVP-LAN EVC EP (section 14.2.4)
- EP-TREE EVC (section 14.2.5)
- EP-TREE EVC EP (section 14.2.5)

- 437 • EVP-TREE EVC (section 14.2.6)
- 438 • EVP-TREE EVC EP (section 14.2.6)
- 439 • Subscriber UNI (section 14.3)

12 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.

12.1 Organization and Structure of the Schemas

The schemas are organized into a file structure as shown in Figure 10.

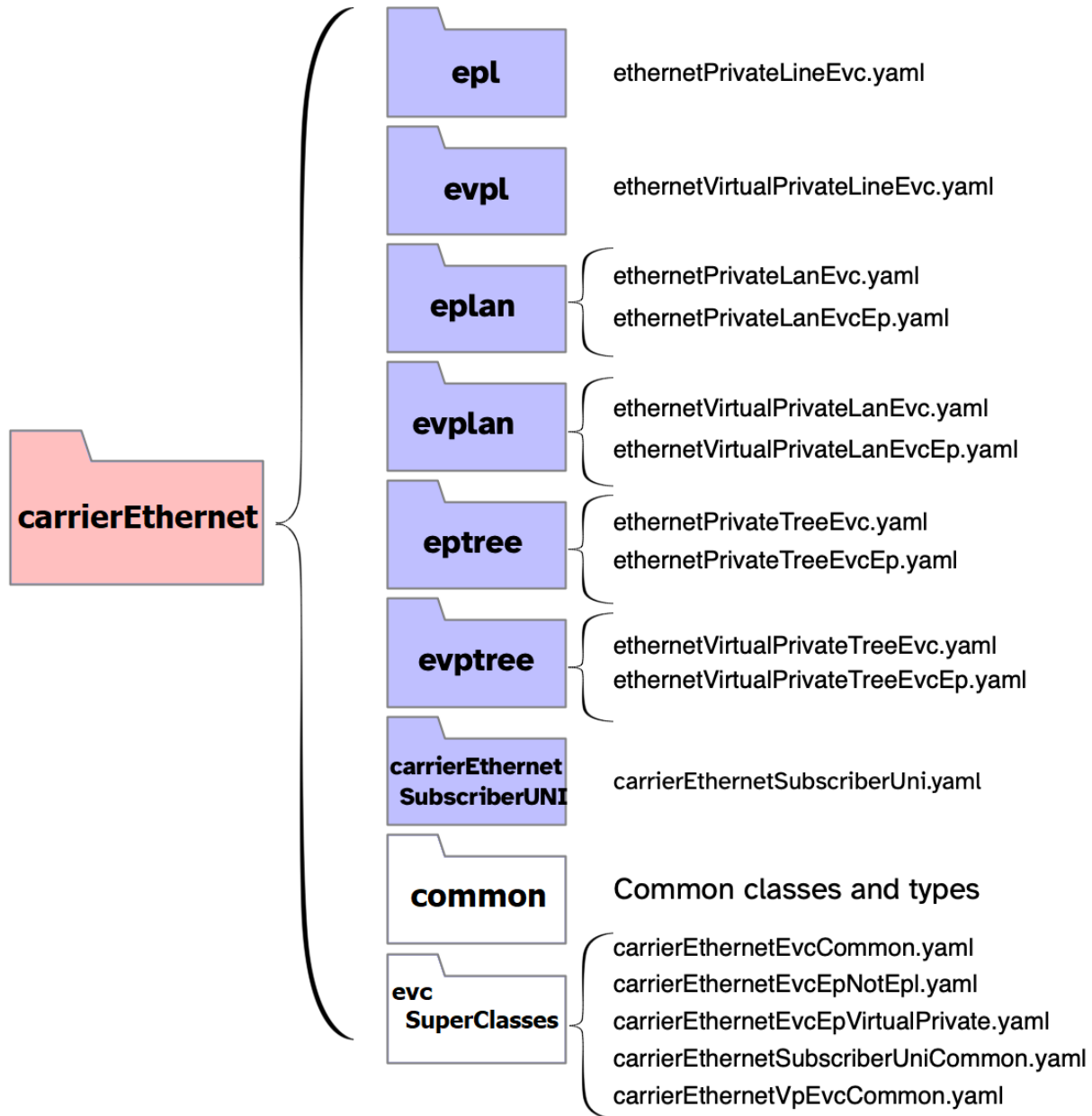


Figure 10 – Schema File Organization

For each of the major product components, Subscriber UNI, EPL, EVPL, etc., there is a folder which contain the relevant schema files.

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

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).

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

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

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

[O2] The Product Inventory for a product **MAY** contain Service Attributes that are categorized as Fixed (see [R1]).

Including Service Attributes in the Product Inventory as [R9] specified in the previous requirements facilitates the CHANGE action. The Buyer can RETRIEVE the current values for the Service Attributes and make the desired changes and submit the CHANGE request.

12.2 Service Attribute Groupings

The EVC and EVC End Point Service Attributes are organized into 6 groups, 3 EVC and 3 EVC End Point (EVC EP), with the schemas for each of the different types of Subscriber Ethernet Services (EPL, EVPL, EP-LAN, EVP-LAN, EP-TREE, EVP-TREE) defined as compositions of the appropriate EVC and EVC EP groups for the corresponding EVC type.

Table 5 below shows how the EVC and EVC End Point Service Attributes schemas for each of the Subscriber Ethernet EVC types are constructed from these 6 Service Attributes groups (the composite schema tables are listed in section 14.2).

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

Table 5 – Service Attribute Groupings

Examples:

For EPL:

- EVC Service Attributes consists of the EVC group *Common*

For EP-LAN:

- EVC EP Service Attributes consists of the EVC EP groups *Common* and *Not EPL*

12.3 Example Schema

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

```
"$schema": http://json-schema.org/draft-07/schema#
title:
  MEF LSO Sonata - Carrier Ethernet Link Aggregation (Common) Product Specification
  # MEF Forum - Date: 2021318 Edit-level: 192
  # File: carrierEthernet/common/carrierEthernetLinkAggregation.yaml
definitions:
  ConversationIdToAggregationLinkMap:
    type: object
    description: >-
      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.
    properties:
      conversationIDs:
        description: >-
          802.1AX-2014 §6.6.2.1 - A Port Conversation ID is a VLAN ID
          (1 to 4094) or 0 to represent untagged and priority-tagged frames.
        type: array
        minItems: 1
        uniqueItems: TRUE
        items:
          type: object
          $ref: "#/definitions/ConversationIdRange"
      aggLinkList:
        description: 802.1AX-2014 §6.6.2.1 - An ordered list of Aggregation Link Numbers
        # See additional validation notes in the Schema Guide
        type: array
        minItems: 1
        uniqueItems: TRUE
        items:
          type: integer
          minimum: 1
    required:
```

```

525     - conversationIDs
526     - agglinkList
527 ConversationIdRange:
528   type: object
529   description: >-
530     A range of ConversationID (either a VLAN Id or 0 for untagged
531     frames) allowing three forms: start, start → start, and start
532     → end
533   properties:
534     start:
535       description: >-
536         The starting Conversation ID of the range or the only Conversation
537         ID if there is no end value
538       # start must be less than or equal to end
539       # See additional validation notes in the Schema Guide
540       type: integer
541       minimum: 0
542       maximum: 4094
543     end:
544       description: The final Conversation ID in the range
545       # end, if specified, must be greater than or equal to start
546       # See additional validation notes in the Schema Guide
547       type: integer
548       minimum: 0
549       maximum: 4094
550   required:
551     - start

```

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

```

556 aggregationLinkMap:
557   description: >-
558     MEF 26.2 §14.6 and 802.1AX-2014 § 6.6.2.1 - A mapping of a
559     list of Port Conversation IDs (i.e., VLAN IDs or zero for untagged
560     frames) to an ordered list of physical links if there are multiple
561     physical links in the UNI and the link aggregation is ALL_ACTIVE.
562   # See additional validation notes in the Schema Guide
563   type: array
564   minItems: 0
565   uniqueItems: TRUE
566   items:
567     type: object
568     $ref: "../../common/carrierEthernetLinkAggregation.yaml#/definitions/ConversationIdToAg-
569     gregationLinkMap"

```

12.4 Additional Details

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

12.4.1 Naming Conventions

In the schemas, class and type names are UpperCamelCase and attribute/property names are lowerCamelCase.

12.4.2 EVC End Point Service Attributes

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

12.4.3 Additional EVC End Point Attributes

Conversely, there is one Service Attribute, the EVC End Point Envelopes that is included in the schemas, but not part of MEF 10.4.

13 Relationship Between Entities

This section describes the constraints and relationships between the three primary Order Items (EVC, 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. The relationship between the EVC and the UNI is captured in the product-agnostic part of the POQ, Quote and Order APIs. The values in the Relationship Type column in the table below are used in the *relationshipType* field of the *ProductRelationship*, *QualificationItemRelationship*, *QuoteItemRelationship* and *OrderItemRelationship* types. This relationship is also shown in Figure 11 and Figure 12 below.

#	Product	Relationship Type	Cardinality	Source Product Specification	Target Product Specification
1	EPL/EVPL	CONNECTS_TO_UNI_A	1	EVC	UNI
2	EPL/EVPL	CONNECTS_TO_UNI_Z	1	EVC	UNI
3	EP-LAN/EVP-LAN	ENDPOINT_OF_EVC	1	EVC EP	EVC
4	EP-TREE/EVP-TREE	ROOT_ENDPOINT_OF_EVC	0..1	EVC EP	EVC
5	EP-TREE/EVP-TREE	LEAF_ENDPOINT_OF_EVC	0..1	EVC EP	EVC
6	EVC End Point for EP-LAN/EVP-LAN, EP-TREE/EVP-TREE	CONNECTS_TO_UNI	1	EVC EP	UNI

Table 6 – Product Relationship Roles

[R11] For a Subscriber Ethernet EVC product, the Relationship Type field of the Product Relationship, POQ Item Relationship, Quote Item Relationship and Order Item Relationship types **MUST** contain one of the values shown in the Relationship Type column in Table 6.

[R12] For POQ, Quote and Order, the relationship to a UNI **MUST** be specified for every INSTALL of, or CHANGE to, a Subscriber Ethernet EVC product.

[R13] For a Subscriber Ethernet EVC product, the relationship to a UNI **MUST** reference a Subscriber Ethernet UNI or an equivalent POQ Item, Quote Item or Order Item.

[R14] For a CHANGE to a Subscriber Ethernet EVC product the relationship to the UNI **MUST NOT** be changed from the value present in the Product Inventory.

Note that [R14] indicates that once a Subscriber Ethernet EVC is associated with a UNI, it cannot be associated with a different EVC.

The relationships listed in Table 6 for all rows, except #4 and #5 are mandatory for both Install and for Change Orders. For EP-TREE and EVP-TREE it is mandatory to specify one of the two End Point relationships, either #4 or #5.

A UNI and an EVC EP may be included in the same order as the EVC. The UNI is associated with a specific `INSTALL_LOCATION`, which is required at `INSTALL` and `CHANGE`. Once a UNI is associated with a specific location, the `INSTALL_LOCATION` cannot be changed and as such, the same `INSTALL_LOCATION` value must be specified for every `CHANGE`. The install location is captured in the product-agnostic part of the POQ, Quote and Order APIs. The value in the Place Relationship Role column in the table below is used in the *role* field of the *RelatedPlaceRefOrValue* type.

Product	Place Relationship Role	Cardinality	Install	Change
UNI	INSTALL_LOCATION	1	Mandatory	Mandatory (Must be same value as Product Inventory)

Table 7 – Place Relationship Role

[R15] For a Subscriber Ethernet UNI product, the Role field (role) of the Related Place (RelatedPlaceRefOrValue) type **MUST** contain one of the values shown in the Place Relationship Role in Table 7.

[R16] For POQ, Quote and Order, the Related Place (*RelatedPlaceRefOrValue*) **MUST** be specified for every `INSTALL` of, or `CHANGE` to, a Subscriber Ethernet UNI.

[R17] For a `CHANGE` to a Subscriber Ethernet UNI product, the Related Place **MUST NOT** be changed from the value present in the Product Inventory.

Changing the UNI Location is not supported for a Subscriber Ethernet Service. The value included in a `CHANGE` request must be identical to the value in the Inventory.

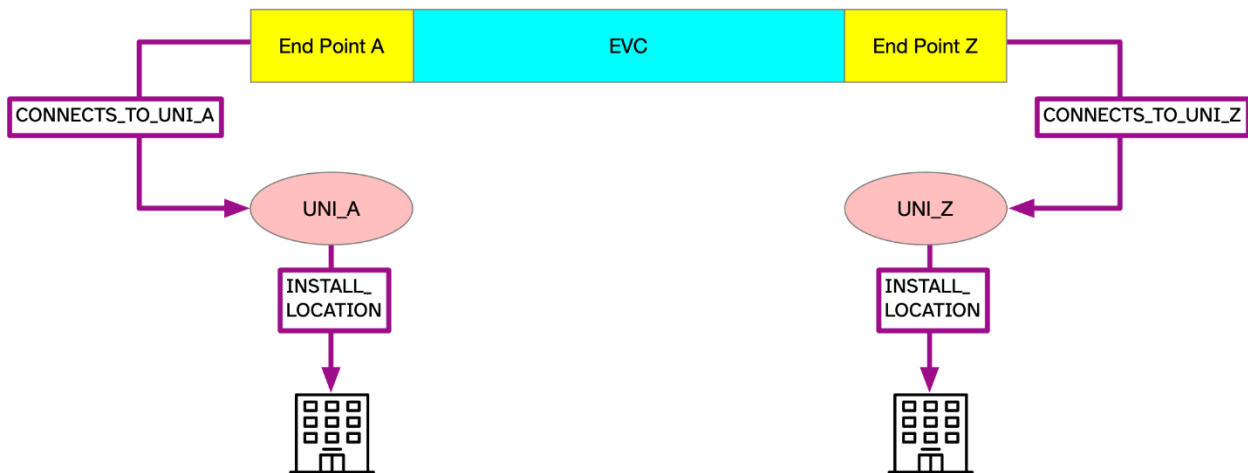
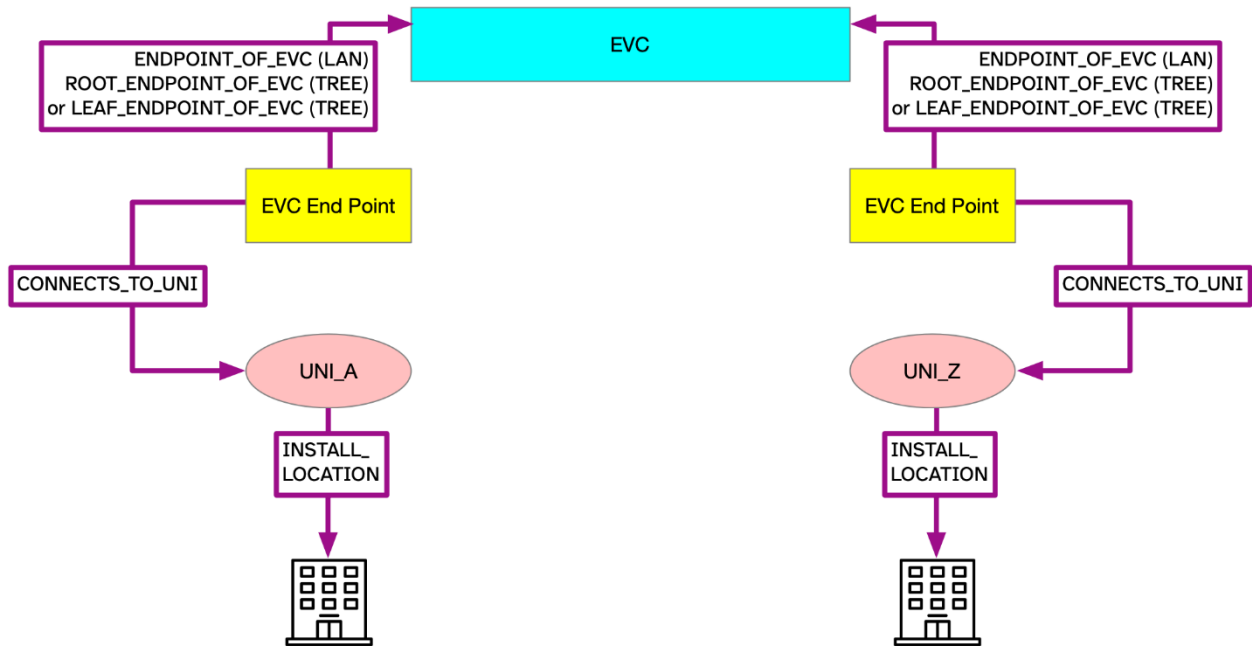


Figure 11 – Relationships for EPL and EVPL

Figure 12 – Relationships for EP-LAN, EVP-LAN, EP-TREE and EVP-TREE

14 Subscriber Ethernet Service Attributes

The following Subscriber Ethernet Service Attributes Schemas are described in the tables in this section:

- EVC Groups
- EVC EP Groups
- EVC Service Attributes (constructed from the appropriate EVC Groups)
- EVC End Point Service Attributes (constructed from the appropriate EVC EP Groups)
- Subscriber Ethernet UNI Service Attributes

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

Not all MEF 10.4 Service Attributes are included in the data models. The Service Attributes that are not included are also listed in section 16. Some Service Attributes are not included because they are included in the Product Independent information portion of the API (e.g., some of the Identifiers) and some Service Attributes are not included because they are constants (i.e., can only have one possible value) either because they are defined that way (e.g., the Subscriber UNI Service Frame Format Service Attribute must be “802.3-2012”) or because the value is constrained to a single value by MEF 6.3 [7] (e.g., the EVC Type Service attribute for an EPL Service must be point-to-point).

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

14.1 EVC Groups

This section enumerates the Subscriber Ethernet Service Attributes Schemas for the three EVC Groups (Common, Not EPL, Virtual) and the four EVC Endpoint Groups (Common, EPL, Not EPL, Virtual).

14.1.1 EVC and EVC EP Group – Common

This EVC and EVC End Point Service Attributes group is common to all Subscriber Ethernet Services.

Carrier Ethernet EVC Common

Schema File Name: carrierEthernetEvcCommon

Includes classes:

CarrierEthernetEvcCommon (Carrier Ethernet EVC Common)

CarrierEthernetEvcEpCommon (Carrier Ethernet EVC EP 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]

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. If the value of this attribute is an empty list, it is equivalent to a value of None per MEF 10.4.

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

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

Validation Notes: The value of the EVC Maximum Service Frame Size Service Attribute **MUST** be less than or equal to the value of the Subscriber UNI Maximum Service Frame Size Service Attribute, *SubscriberUniCommon.maximumServiceFrameSize*, for all UNIs in the EVC.

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). For all Subscriber Ethernet Services defined in MEF 6.3 the value is constrained to the range 0...6.

Allowed values: Type: AvailableMegList

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]+

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*

Validation Notes: The Class of Service Name or Names **must** be included in the *CarrierEthernetEvcCommon.listOfCosNames*. Per MEF 6.3 if there is only one Class of Service in *CarrierEthernetEvcCommon.listOfCosNames* the *mapType* element of this property **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*

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. If this property is or an empty list, the value is assumed to have the value *None*.

Allowed values: List of [0...1] Class: *EvcEgressMap*

Validation Notes: Per R140 in MEF 10.4, the value of the EVC EP Egress Map Service Attribute MUST NOT be *None* (an empty list) unless one of the following conditions :

- The value of the EVC EP Map Service Attribute (MEF 10.4, section 10.4) is UT/PT.
- The value of the EVC C-Tag PCP Preservation Service Attribute (MEF 10.4, section 8.5) = *Enabled*, the value of the EVC C-Tag DEI Preservation Service Attribute (MEF 10.4, section 8.6) = *Enabled*, and the value of the EVC EP Map Service Attribute (MEF 10.4, section 10.4) = *List* for all EVC EPs in the EVC.
- The value of the EVC C-Tag PCP Preservation Service Attribute (MEF 10.4, section 8.5) = *Enabled*, the value of the EVC C-Tag DEI Preservation Service Attribute (MEF 10.4, section 8.6) = *Enabled*, and the value of the EVC EP Map Service Attribute (MEF 10.4, section 10.4) = *All*.

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 *None* or a non-empty list of pairs of the form $\langle x, y \rangle$, 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. If this property is an empty list, the value is assumed to have the value *None*.

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

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: Type: MegList

14.1.2 EVC EP Group – EPL

This EVC End Point Service Attribute group is used for EPL and EVPL.

Carrier Ethernet EVC EP Eline

Schema File Name: carrierEthernetEvcEpEline

Includes classes:

CarrierEthernetEvcEpEline (Carrier Ethernet EVC EP Eline)

Class: CarrierEthernetEvcEpEline (**Carrier Ethernet EVC EP Eline**)

Property Name: *EVC End Point Identifier 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): $[\backslash x20-\backslash x7F]^+$

14.1.3 EVC and EVC EP Group – Not EPL

This EVC and EVC End Point Service Attributes group is 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)

CarrierEthernetEvcEpNotEpl (Carrier Ethernet EVC EP 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: *FramedDisposition*

Validation Notes: For EPL this must be unconditional for all 3 entries so not included

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 $\langle x, y \rangle$ 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 $\langle CIR, CIR_{max}, ER \rangle$ where the elements of the 3-tuple are defined in MEF 10.4 Section 12.1.2. If this property an empty list, the value is assumed to have the value *None*.

Allowed values: List of $[0...1]$ Class: *BandwidthProfilePerClassOfServiceName*

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 $\langle N, t \rangle$ 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 = $\langle N, t \rangle$ the number of source MAC Addresses that can be used in Ingress Service Frames is limited. If this property is an empty list, the value is assumed to have the value *None*.

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

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

14.1.4 EVC and EVC EP Group – Virtual

This EVC Service Attributes group is common to all Virtual Private Services (EVPL, EVP-LAN, EVP-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: Type: EnabledDisabled

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: Type: EnabledDisabled

Validation Notes: For Private Services this **must** be *Enabled* so it is only included for Virtual Private Services.

This EVC 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 *ALL*, *LIST*, 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 *LIST*. 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

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

Property Name: *EVC End Point Envelopes Service Attribute*

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

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

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

To create an Envelope containing Bandwidth Profile Flows for more than one EVC End Point, the UNI Envelopes Service Attribute must be used.

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

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

14.2 Subscriber Ethernet EVC and EVC End Point Service Attributes

This section lists the Subscriber Ethernet EVC and EVC End Point Service Attribute Schemas for each of the Subscriber Ethernet EVC types that are constructed from the appropriate set of 7 Service Attribute groups described in sections 12.2 above.

14.2.1 EPL Service Attributes

Ethernet Private Line EVC

Schema File Name: ethernetPrivateLineEvc
urn:mef:iso:spec:cantata-sonata:epl:v0.3.0:common
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.

Property Name: EPL EVC Service Attributes **Type:** object

Allowed values: Compose (allof): Class: CarrierEthernetEvcCommon

Property Name: EPL End Point A

Reference: MEF 10.4 sec. 10

JSON Name: evcEndPointA – **Type:** object

Description: EVC End Point at UNI A. UNI EVC End Point A and UNI EVC End Point Z must be included in the EPL Product.

Allowed values: Type: EplEvcEndPoint

Property Name: EPL End Point Z

Reference: MEF 10.4 sec. 10

JSON Name: evcEndPointZ – **Type:** object

Description: EVC End Point at UNI Z. UNI EVC End Point A and UNI EVC End Point Z must be included in the EPL Product.

Allowed values: Type: EplEvcEndPoint

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: CarrierEthernetEvcEpEline, Class: CarrierEthernetEvcEpCommon

698

14.2.2 EVPL Service Attributes

Ethernet Virtual Private Line EVC

Schema File Name: ethernetVirtualPrivateLineEvc
urn:mef:iso:spec:cantata-sonata:evpl:v0.3.0:common
Includes classes:
EvplEvc (EVPL EVC Service Attributes)
EvplEvcEndPoint (EVPL EVC End Point Service Attributes)

Class: EvplEvc (**EVPL EVC Service Attributes**)

The EVC Service Attributes for Ethernet Virtual Private Line (EVPL) Service as defined in MEF 6.3 based on the Service Attributes defined in MEF 10.4.

Property Name: EVPL EVC Service Attributes **Type:** object

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

Property Name: EVPL EVC Endpoint A

Reference: MEF 10.4 sec. 10

JSON Name: evcEndPointA – **Type:** object

Description: EVC End Point at UNI A. UNI EVC End Point A and UNI EVC End Point Z must be included in the EVPL Product.

Allowed values: Type: EvplEvcEndPoint

Property Name: EVPL EVC Endpoint Z

Reference: MEF 10.4 sec. 10

JSON Name: evcEndPointZ – **Type:** object

Description: EVC End Point at UNI Z. UNI EVC End Point A and UNI EVC End Point Z must be included in the EVPL Product.

Allowed values: Type: EvplEvcEndPoint

Type: EvplEvcEndPoint (**EVPL 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 Virtual Private Line (EVPL) Services defined in MEF 6.3 based on the Service Attributes in MEF 10.4.

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

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

702

703 14.2.3 EP-LAN Service Attributes

Ethernet Private LAN EVC

Schema File Name: ethernetPrivateLanEvc
urn:mef:iso:spec:cantata-sonata:eplan:v0.3.0:common
Includes classes:
EplanEvc (EP-LAN EVC Service Attributes)
EplanEvcEndPoint (EP-LAN EVC End Point Service Attributes)

Class: EplanEvc (EP-LAN EVC Service Attributes)

The EVC Service Attributes for Ethernet Private LAN (EP-LAN) Service as defined in MEF 6.3 based on the Service Attributes defined in MEF 10.4.

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

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

Type: EplanEvcEndPoint (EP-LAN 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 LAN (EP-LAN) Services defined in MEF 6.3 based on the Service Attributes in MEF 10.4.

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

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

706

707 **14.2.4 EVP-LAN Service Attributes**
Ethernet Virtual Private LAN EVC

Schema File Name: ethernetVirtualPrivateLanEvc
urn:mef:iso:spec:cantata-sonata:evplan:v0.3.0:common
Includes classes:
EvplanEvc (EVP-LAN EVC Service Attributes)
EvplanEvcEndPoint (EVP-LAN EVC End Point Service Attributes)

Class: EvplanEvc (EVP-LAN EVC Service Attributes)

The EVC Service Attributes for Ethernet Virtual Private LAN (EVP-LAN) Service as defined in MEF 6.3 based on the Service Attributes defined in MEF 10.4.

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

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

Type: EvplanEvcEndPoint (EVP-LAN 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 Virtual Private LAN (EVP-LAN) Services defined in MEF 6.3 based on the Service Attributes in MEF 10.4.

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

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

710

711 14.2.5 EP-TREE Service Attributes

Ethernet Private Tree EVC

Schema File Name: ethernetPrivateTreeEvc
urn:mef:iso:spec:cantata-sonata:eptree:v0.3.0:common
Includes classes:
EptreeEvc (EP-TREE EVC Service Attributes)
EptreeEvcEndPoint (EP-TREE EVC End Point Service Attributes)

Class: EptreeEvc (EP-TREE EVC Service Attributes)

The EVC Service Attributes for Ethernet Private TREE (EP-TREE) Service as defined in MEF 6.3 based on the Service Attributes defined in MEF 10.4.

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

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

Type: EptreeEvcEndPoint (EP-TREE 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 Tree (EP-TREE) Services defined in MEF 6.3 based on the Service Attributes in MEF 10.4.

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

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

714

715

14.2.6 EVP-TREE Service Attributes

Ethernet Virtual Private Tree EVC

Schema File Name: ethernetVirtualPrivateTreeEvc
urn:mef:iso:spec:cantata-sonata:evptree:v0.3.0:common
Includes classes:
EvptreeEvc (EVP-TREE EVC Service Attributes)
EvptreeEvcEndPoint (EVP-TREE EVC End Point Service Attributes)

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

The EVC Service Attributes for Ethernet Virtual Private TREE (EVP-TREE) Service as defined in MEF 6.3 based on the Service Attributes defined in MEF 10.4.

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

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

Type: EvptreeEvcEndPoint (**EVP-TREE 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 Virtual Private Tree (EVP-TREE) Services defined in MEF 6.3 based on the Service Attributes in MEF 10.4.

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

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

717

14.3 Subscriber Ethernet UNI Service Attributes

This section enumerates the Subscriber Ethernet UNI Service Attributes.

Carrier Ethernet Subscriber UNI

Schema File Name: carrierEthernetSubscriberUni
urn:mef:iso:spec:cantata-sonata:carrier-ethernet-subscriber-uni:v0.3.0:common
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.

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

Allowed values: Compose (allof): Class: SubscriberUniCommon

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. This class is used to compose the SubscriberUNI class.

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"

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

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 $(id, physicalLink, synchronousEthernet, precisionTiming, demarcConnectorType, gender)$, 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...*] Type: UniPhysicalLink

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: Type: LinkAggType

Validation Notes: If the value of *SubscriberUniCommon.listOfPhyLinks* =

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

Property Name: *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

Validation Notes: This must be empty if the value of *SubscriberUniCommon.linkAggregation* is not ALL_ACTIVE or OTHER. If it is not empty, it must map every C-VLAN ID mapped to the EVC. (See MEF 10.4 R77, R78, R80.)

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

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 EVC End Points that can be supported at the UNI.

Allowed values: Minimum: 1 Maximum: 4094

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

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: Type: EnabledDisabled

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 None or non-empty list of {envelopeID, couplingFlagForIndexZero}. If this property an empty list, the value is assumed to have the value None.

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

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

Property Name: *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: Type: EnabledDisabled

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: Type: EnabledDisabled

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: Type: EnabledDisabled

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: Type: L2cpAddressSet

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

15 Common Classes and Types

This section is structured 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.

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

15.1 Bandwidth Profiles

Carrier Ethernet Bandwidth Profile

Schema File Name: carrierEthernetBandwidthProfile

Includes classes:

BwpFlow (Bandwidth Profile Flow)

Envelope (Envelope)

BandwidthProfilePerClassOfServiceName (COS Bandwidth Profile)

Class: BwpFlow (**Bandwidth Profile Flow**)

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

Property Name: Committed Information Rate

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

JSON Name: cir – **Type:** object

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

Allowed values: Class: InformationRate

Property Name: The Maximum Committed Information Rate

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

JSON Name: cirMax – **Type:** object

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

Allowed values: Class: InformationRate

Property Name: Committed Burst Size

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

JSON Name: cbs – **Type:** object

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

Allowed values: Class: DataSize

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

Property Name: *Excess Information Rate*

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

JSON Name: *eir* – **Type:** *object*

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

Allowed values: Class: InformationRate

Property Name: *Maximum Excess Information Rate*

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

JSON Name: *eirMax* – **Type:** *object*

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

Allowed values: Class: InformationRate

Property Name: *Excess Burst Size*

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

JSON Name: *ebs* – **Type:** *object*

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

Allowed values: Class: DataSize

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

Property Name: *Coupling Flag*

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

JSON Name: *couplingFlag* – **Type:** *boolean*

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

Allowed values:

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

Property Name: *Color Mode*

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

JSON Name: *colorMode* – **Type:** *enum*

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

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

Property Name: *Envelope Name*
Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

JSON Name: *envelopeId* – **Type:** *string*
Description: The name of the envelope that this Bandwidth Profile Flow resides in.

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

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

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

Property Name: *Envelope Rank*
Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

JSON Name: *envelopeRank* – **Type:** *integer*
Description: The rank in the envelope of this Bandwidth Profile Flow

Allowed values: Minimum: 1

Property Name: *Token Request Offset Field*
Reference: MEF 26.2 sec. 17.1.2 and MEF 10.4 sec. 12.1.2

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

Allowed values: [Any Integer]

Class: Envelope (***Envelope***)

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

Property Name: *Envelope Name*
Reference: MEF 10.4 sec. 9.12 and MEF 26.2 sec. 14.19

JSON Name: *envelopeID* – **Type:** *string*
Description: The attribute is a string that identifies the Envelope

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

Property Name: *Envelope 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:
Class: BandwidthProfilePerClassOfServiceName (***COS Bandwidth Profile***)

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

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

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

Property Name: *CoS or EEC Name*

Reference: MEF 26.2 sec. 16.12 and 16.13

JSON Name: *classOfServiceName* – **Type:** *string*

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

Allowed values: [Any String]

Property Name: *CoS Bandwidth Profile*

JSON Name: *bwpFlow* – **Type:** *object*

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

Allowed values: Class: BwpFlow

15.2 Class of Service

Carrier Ethernet Class of Service

Schema File Name: carrierEthernetClassOfService

Includes classes:

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

Class: CosFromEp (CoS Name From EP)

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

Property Name: CoS Map Type – EP

JSON Name: mapType – **Type:** const

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

Allowed values: "ENDPOINT"

Property Name: CoS Map

JSON Name: map_M – **Type:** string

Description: The single CoS at this EVC/OVC EP

Allowed values: [Any String]

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

Property Name: CoS L2CP

JSON Name: l2cp_P – **Type:** object

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

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

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

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

Property Name: CoS Map Type – CTAG

JSON Name: mapType – **Type:** const

Description: The Field that indicates F = C_TAG_PCP.

Allowed values: "C_TAG_PCP"

Property Name: CoS Map

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

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

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

Property Name: *CoS L2CP*

JSON Name: *l2cp_P* – **Type:** *object*

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

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

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

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

Property Name: *C-Tag PCP Value*

JSON Name: *pcpVal* – **Type:** *enum*

Description: A value 0 to 7 or UNTAGGED

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

Property Name: *CoS Name*

JSON Name: *pcpCosName* – **Type:** *string*

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

Allowed values: [Any String]

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

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

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

Property Name: *CoS Map Type – STAG*

JSON Name: *mapType* – **Type:** *const*

Description: The Field that indicates F = S_Tag_PCP.

Allowed values: "S_TAG_PCP"

Property Name: *CoS Map*

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

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

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

Property Name: *CoS L2CP*

JSON Name: *l2cp_P* – **Type:** *object*

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

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

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

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

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

Property Name: ENNI PCP Value
JSON Name: pcpVal – **Type:** enum
Allowed values: Type: PcpValue

Property Name: ENNI CoS Name
JSON Name: pcpCosName – **Type:** string
Allowed values: [Any String]

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

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

Property Name: CoS Map Type – DSCP
JSON Name: mapType – **Type:** const
Description: The Field that indicates F = DSCP.
Allowed values: "DSCP"

Property Name: CoS Map
JSON Name: map_M – **Type:** object
Description: This maps DSCP values to CoS Names for each of Ipv4 and Ipv6 and also to non-IP packets.
Allowed values: Class: CosDscpMap

Property Name: CoS L2CP
JSON Name: l2cp_P – **Type:** object
Description: This specifies the CoS Names for L2CPs carried over the EVC/OVC.
Allowed values: List of [0...*] Unique Class: CosL2cp

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

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

Property Name: DSCP Value to CoS List
JSON Name: dscpValueCoSList – **Type:** List of object
Description: This is a list of DSCP to CoS Map Entries: {ipv4DSCPs, ipv6DSCPs, CoSName}
Allowed values: List of [0...*] Unique Class: CosDscpMapEntry

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

Property Name: CoS for Other Ipv4
JSON Name: otherIPv4 – **Type:** string
Description: This is the CoS Name for Ipv4 packets with DSCP values other than those listed.
Allowed values: [Any String]

Property Name: CoS for Other Ipv6
JSON Name: otherIPv6 – **Type:** string
Description: This is the CoS Name for Ipv6 packets with DSCP values other than those listed.

Allowed values: [Any String]

Property Name: *CoS for Not IP Packet*

JSON Name: *notIP* – **Type:** *string*

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

Allowed values: [Any String]

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

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

Property Name: *Ipv4 List of DSCPs*

JSON Name: *ipv4List* – **Type:** *object*

Description: List of Ipv4 DSCP values

Allowed values: Class: DscpValues

Property Name: *Ipv6 List of DSCPs*

JSON Name: *ipv6List* – **Type:** *object*

Allowed values: Class: DscpValues

Property Name: *Class of Service Name*

JSON Name: *cosName* – **Type:** *string*

Allowed values: [Any String]

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

A list of 6 bit values

Property Name: *List of DSCP Values*

JSON Name: *dscpValues* – **Type:** *List of integer*

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

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

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

2-tuple {L2CP Ident, CoS Name}

Property Name: *L2CP Identifier*

Reference: MEF 45.1 sec. 8.2

JSON Name: *l2cpIdentifier* – **Type:** *object*

Description: Protocol Identifier

Allowed values: Class: L2cpProtocol

Property Name: *L2CP CoS Name*

JSON Name: *l2cpCosName* – **Type:** *string*

Description: CoS Name String to associate with the L2CP

Allowed values: [Any String]

751

15.3 Color Identifier

Carrier Ethernet Color Identifier

Schema File Name: carrierEthernetColorIdentifier

Includes classes:

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

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

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

Property Name: Color Map Type – DEI

JSON Name: mapType – **Type:** const

Allowed values: "DEI"

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

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

Property Name: Color Map Type – PCP

JSON Name: mapType – **Type:** const

Allowed values: "PCP"

Property Name: Color Map

JSON Name: colorFromPcpMap – **Type:** List of object

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

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

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

Property Name: PCP Value

JSON Name: pcpValue – **Type:** enum

Allowed values: Type: PcpValue

Property Name: PCP Color

JSON Name: pcpColor – **Type:** enum

Allowed values: Type: FrameColor

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

This Class is referenced when the color comes from the DSCP field. It is a list of DSCP Entries where each entry has three components: (1) A list of DSCP values, (2) a color for Ipv4 packets with one of those DSCP values, and (3) a color for Ipv6 packets with one of those DSCP values. Any DSCP value that is not specified and any Frame that does not include an IP packet is Green.

Property Name: *Color Map Type – DSCP*

JSON Name: *mapType – Type: const*

Allowed values: "DSCP"

Property Name: *colorMap*

JSON Name: *colorFromDscpMap – Type: List of object*

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

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

Property Name: *DSCP List*

JSON Name: *dscpList – Type: List of integer*

Description: This is a list of DSCP values

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

Property Name: *Ipv4 Color*

JSON Name: *ipv4Color – Type: enum*

Allowed values: Type: FrameColor

Property Name: *Ipv6 Color*

JSON Name: *ipv6Color – Type: enum*

Allowed values: Type: FrameColor

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

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

Property Name: *Map Type – ENDPOINT*

JSON Name: *mapType – Type: const*

Allowed values: "ENDPOINT"

Property Name: *Color*

JSON Name: *epColor – Type: enum*

Allowed values: Type: FrameColor

15.4 Egress Maps

Carrier Ethernet Egress Maps

Schema File Name: carrierEthernetEgressMaps

Includes classes:

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

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

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

Property Name: Map Type - CN → PCP

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

Allowed values: "CN_PCP"

Property Name: CNPCP Entries

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

Description: A list of mappings of each CoS to an egress PCP value or DISCARD.

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

Class: CnPcp (**CNPCP Entry**)

This is a 2-tuple $\langle x, y \rangle$ where x is a Class of Service Name and y is a PCP Value or DISCARD.

Property Name: Class of Service Name

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

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

Allowed values: [Any String]

Property Name: PCP Value

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

Allowed values: Type: PcpValueAndDiscard

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

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

Property Name: *Map Type CC → DEI*
JSON Name: *mapType* - **Type:** *const*
Allowed values: "CC_DEI"

Property Name: *List of CCDEI Entries*
JSON Name: *ccDeiEntries* - **Type:** *List of object*
Description: A list of entries, one per CoS, that specifies the DEI value for Green frames and Yellow frames.
Allowed values: List of [1...*] Unique Class: CcDei

Class: CcDei (**CCDEI Entry**)

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

Property Name: *Class of Service*
JSON Name: *cosName* - **Type:** *string*
Description: The class of service name for the corresponding Ingress EI Frame.
Allowed values: [Any String]

Property Name: *DEI for Green Frames*
JSON Name: *deiGreen* - **Type:** *object*
Description: The DEI value (or DISCARD) to use for Green frames.
Allowed values: Type: DeiValueAndDiscard

Property Name: *DEI for Yellow Frames*
JSON Name: *deiYellow* - **Type:** *object*
Description: The DEI value (or DISCARD) to use for Yellow frames.
Allowed values: Type: DeiValueAndDiscard

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

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

Property Name: *Map Type - CC → PCP*
JSON Name: *mapType* - **Type:** *const*
Allowed values: "CC_PCP"

Property Name: *List of CCPCP Entries*
JSON Name: *ccPcpEntries* - **Type:** *List of object*
Description: A list of entries, one per CoS that indicates the PCP value for Green Frames and the PCP value for Yellow Frames.
Allowed values: List of [1...*] Unique Class: CcPcp

Class: CcPcp (CCPCP Entry)

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

Property Name: Class of Service

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

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

Allowed values: [Any String]

Property Name: PCP For Green Frames

JSON Name: pcpGreen - **Type:** object

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

Allowed values: Type: PcpValueAndDiscard

Property Name: PCP For Yellow Frames

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

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

Allowed values: Type: PcpValueAndDiscard

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

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

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

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

Allowed values: "CN_PCP_CC_DEI"

Property Name: List of CNPCP_CCDEI Entries

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

Description: A list of entries, one per CoS, 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

Class: CnPcpCcDei (CNPCPCDEI Entry)

Property Name: Class of Service

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

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

Allowed values: [Any String]

Property Name: *PCP Value*

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

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

Allowed values: Type: PcpValueAndDiscard

Property Name: *DEI for Green Frames*

JSON Name: *deiGreen* - **Type:** *object*

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

Allowed values: Type: DeiValueAndDiscard

Property Name: *DEI for Yellow Frames*

JSON Name: *deiYellow* - **Type:** *object*

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

Allowed values: Type: DeiValueAndDiscard

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

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

Property Name: *List of EVC Egress Map Entries*

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

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

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

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

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

Property Name: *CoS Name*

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

Allowed values: [Any String]

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

Property Name: *PCP Yellow*

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

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

Allowed values: Type: PcpValueAndDiscard

Property Name: *DEI Green***JSON Name:** *deiGreen* - **Type:** *object***Description:** A DEI value to use for Green frames**Allowed values:** Type: *DeiValueAndDiscard***Validation Notes:** This is optional ONLY if *pcpGreen* is set to *DISCARD***Property Name:** *DEI Yellow***JSON Name:** *deiYellow* - **Type:** *object***Description:** A DEI value to use for Yellow frames**Allowed values:** Type: *DeiValueAndDiscard***Validation Notes:** This is optional ONLY if *pcpYellow* is set to *DISCARD*

771

15.5 End Point Maps

Carrier Ethernet End Point Maps

Schema File Name: carrierEthernetEndPointMaps

Includes classes:

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

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

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

Property Name: Map Form - E

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

Description: The discriminator for Map Type E

Allowed values: "FORM_E"

Property Name: End Point Map Form E

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

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

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

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

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

Property Name: Map Form - T

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

Description: The discriminator for Map Type T

Allowed values: "FORM_T"

Property Name: Root S-VLAN ID

JSON Name: rootSvlan - **Type:** integer

Description: The S-VLAN ID used for Root traffic

Allowed values: Minimum: 1 Maximum: 4094

Property Name: Leaf S-VLAN ID

JSON Name: leafSvlan - **Type:** integer

Description: The S-VLAN ID used for Leaf traffic

Allowed values: Minimum: 1 Maximum: 4094

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

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

Property Name: Map Form - V

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

Description: The discriminator for Map Type V

Allowed values: "FORM_V"

Property Name: S-VLAN ID

JSON Name: svlanId - **Type:** integer

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

Allowed values: Minimum: 1 Maximum: 4094

Property Name: C-VLAN ID

JSON Name: cvlanId - **Type:** integer

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

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

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

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

Property Name: Map Form - U

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

Description: The discriminator for Map Type U

Allowed values: "FORM_U"

Property Name: OVC End Point Map Form U

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

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

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

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

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

Property Name: VLAN Listing Type

JSON Name: vlanType - **Type:** enum

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

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

Property Name: *VLAN ID List*

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

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

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

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

778

15.6 External Interfaces

Carrier Ethernet External Interfaces

Schema File Name: carrierEthernetExternalInterfaces

Includes classes:

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

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

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

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

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

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

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

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

Allowed values: Compose (allOf): Class: CarrierEthernetPhysicalLink

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

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

Property Name: Physical Link Identifier

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

Description: An identifier for the physical link.

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

Property Name: Physical Link PHY Type

JSON Name: physicalLink - **Type:** enum

Description: An Ethernet physical layer

Allowed values: Enum: "1BASE5", "2BASE_TL", "10BASE2", "10BASE5", "10BASE_F", "10BASE_FB", "10BASE_FL", "10BASE_FP", "10BASE_T", "10BASE_T1L", "10BASE_T1S", "10BASE_TE", "10BROAD36", "10PASS_TS", "100BASE_BX10", "100BASE_FX", "100BASE_LX10", "100BASE_T", "100BASE_T1", "100BASE_T2", "100BASE_T4", "100BASE_TX", "100BASE_X", "1000BASE_BX10", "1000BASE_CX", "1000BASE_KX", "1000BASE_LX", "1000BASE_LX10", "1000BASE_PX10", "1000BASE_PX20", "1000BASE_RHA", "1000BASE_RHB", "1000BASE_RHC", "1000BASE_SX", "1000BASE_T", "1000BASE_T1", "1000BASE_X", "2_5GBASE_KX", "2_5GBASE_T", "2_5GBASE_T1", "5GBASE_KR", "5GBASE_T", "5GBASE_T1", "10GBASE_CX4", "10GBASE_E", "10GBASE_ER", "10GBASE_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"

Property Name: *Auto Negotiation*

Reference: IEEE Std 802.3-2018 Clause 28 and 37

JSON Name: *autoNegotiation* - **Type:** *object*

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

Allowed values: Type: EnabledDisabled

Class: UniSpecificAttributes (**UNI Specific**)

Property Name: *Physical Link Synchronous Ethernet*

JSON Name: *synchronousEthernet* - **Type:** *object*

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

Allowed values: Type: EnabledDisabled

Property Name: *Physical Link Precision Timing*

JSON Name: *precisionTiming* - **Type:** *object*

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

Allowed values: Type: EnabledDisabled

Property Name: *UNI Connector Type*

JSON Name: *uniConnectorType* - **Type:** *enum*

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

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

Property Name: *UNI Connector Gender*

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

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

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

Class: VirtualFrameMap (**Virtual Frame Map**)

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

Property Name: *Virtual Frame Map*

Reference: MEF 10.4 sec. 9.2 and sec. 9.3

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

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

Allowed values: String: Min Length: 0

786

15.7 Layer 2 Control Protocols (L2CP)

Carrier Ethernet L2CP

Schema File Name: carrierEthernetL2cp

Includes classes:

L2cpPeering (L2CP Peering Identifier)

L2cpProtocol (L2CP Protocol Identifier)

Class: L2cpPeering (**L2CP Peering Identifier**)

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

Property Name: Protocol ID

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

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

Allowed values: Class: L2cpProtocol

Property Name: Destination MAC Address

JSON Name: destinationAddress - **Type:** string

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

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

Property Name: Link ID List

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

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

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

Class: L2cpProtocol (**L2CP Protocol Identifier**)

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

Property Name: Protocol Type

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

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

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

Property Name: EtherType or LLC Value

JSON Name: llcAddressOrEtherType - **Type:** integer

Description: The EtherType value or LLC value

Allowed values: Minimum: 0

Property Name: EtherType SubType

JSON Name: subType - **Type:** integer

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

Allowed values: Minimum: 0

15.8 Link Aggregation

Carrier Ethernet Link Aggregation

Schema File Name: carrierEthernetLinkAggregation

Includes classes:

ConversationIdToAggregationLinkMap (Conversation To Aggregation Link Map)

ConversationIdRange (ConversationID Range)

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

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

Property Name: List of Conversation ID Ranges

Reference: 802.1AX-2014 sec. 6.6.2.1

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

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

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

Property Name: Aggregation Link List

Reference: 802.1AX-2014 sec. 6.6.2.1

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

Description: An ordered list of Aggregation Link Numbers

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

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

Class: ConversationIdRange (**ConversationID Range**)

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

Property Name: Start Conversation ID

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

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

Allowed values: Minimum: 0 Maximum: 4094

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

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

Property Name: End Conversation ID

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

Description: The final Conversation ID in the range

Allowed values: Minimum: 0 Maximum: 4094

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

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

15.9 Service Level Specification

Carrier Ethernet Service Level Specification

Schema File Name: carrierEthernetServiceLevelSpecification

Includes classes:

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

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

The Service Level Specification Service Attribute (SLS) is 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. The value is a string in ISO 8601 format.

Allowed values: String; Format: date-time

Property Name: *Time Duration*

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

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

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

Allowed values: Class: TimeDuration

Property Name: *SLS CoS Entries*

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

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

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

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

Class: SlsCosNameEntry (SLS CoS Name Entry)

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

Property Name: CoS Name

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

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

Allowed values: String; Min Length: 1

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

Property Name: Delta T

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

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

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

Allowed values: Minimum: 1

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

Property Name: Frame Loss Threshold

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

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

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

Allowed values: Minimum: 0 Maximum: 1

Property Name: HLI Consecutive Intervals

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

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

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

Allowed values: Minimum: 1

Property Name: One Way Frame Delay

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

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

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

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

Property Name: One Way Mean Frame Delay

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

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

Description: **oneWayMeanFrameDelayPmMetric** is a reference to zero or more instances of One Way Mean Frame Delay Performance Metric. Each reference contains a set of parameters and performance

objective for the Performance Metric. If there are 0 instances then this Class of Service does not have an objective for One Way Mean Frame Delay. If there are multiple instances for this Class of Service then at least one of the parameters must be different between the instances.

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

Property Name: *One Way Frame Delay Range*

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

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

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

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

Property Name: *One Way Inter Frame Delay Variation*

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

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

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

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

Property Name: *One Way Frame Loss Ratio*

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

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

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

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

Property Name: *One Way Availability*

Reference: MEF 10.4 sec. 8.8.7 and MEF 26.2 sec. 12.13.7

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

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

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

Property Name: *One Way HLI*

Reference: MEF 10.4 sec. 8.8.8 and MEF 26.2 sec. 12.13.8

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

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

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

Property Name: *One Way CHLI*

Reference: MEF 10.4 sec. 8.8.9 and MEF 26.2 sec. 12.13.9

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

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

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

Property Name: *One Way Composite Performance*

Reference: MEF 10.4 sec. 8.8.10

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

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

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

Property Name: *One Way Group Availability*

Reference: MEF 10.4 sec. 8.8.11 and MEF 26.2 sec. 12.13.10

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

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

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

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

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

Property Name: *Frame Delay Ordered Pairs*

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

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

Description: The set (referred to as S in 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

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

Property Name: *Frame Delay Percentile*

Reference: MEF 10.4 sec. 8.8.2 and MEF 26.2 sec. 12.13.2 (Pd)

JSON Name: *oneWayFdPercentile* - **Type:** *object*

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

Allowed values: Type: Percentage

Validation Notes: This must be greater than 0.

Property Name: *Frame Delay Objective*

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

JSON Name: *oneWayFdObjective* - **Type:** *object*

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

Allowed values: Class: TimeDuration

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

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

Property Name: *Mean Frame Delay Ordered Pairs*

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

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

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

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

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

Property Name: *Mean Frame Delay Objective*

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

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

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

Allowed values: Class: TimeDuration

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

One Way Frame Delay Range Performance Metric consisting of 2 parameters, a list of Ordered End Point Pairs (referred to as S in MEF 10.4 and MEF 26.2) , a percentile (referred to as Pr in MEF 10.4 and MEF 26.), and the One Way Frame Delay Range Objective for this Class of Service Name and these two parameters.

Property Name: *Frame Delay Range Ordered Pairs*

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

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

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

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

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

Property Name: *Frame Delay Range Percentile*

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

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

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

Allowed values: Type: Percentage

Validation Notes: This must be greater than 0.

Property Name: *Frame Delay Range Objective*

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

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

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

Allowed values: Type: TimeDuration

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

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

Property Name: *Interframe Delay Variation Ordered Pairs*

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

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

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

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

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

Property Name: *Time Duration*

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

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

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

Allowed values: Class: TimeDuration

Property Name: *Interframe Delay Variation Percentile*

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

JSON Name: *oneWayIfdvPencentile* - **Type:** *object*

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

Allowed values: Type: Percentage

Validation Notes: This must be greater than 0.

Property Name: *Interframe Delay Variation Objective*

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

JSON Name: *oneWayIfdvObjective* - **Type:** *object*

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

Allowed values: Class: TimeDuration

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

One Way Frame Loss Ratio Performance Metric consisting of 1 parameter, a list of Ordered End Point Pairs

(referred to as *S* in MEF 10.4 and MEF 26.2), and the One Way Frame Loss Ratio Objective for this Class of Service Name and these Ordered End Point Pairs.

Property Name: *Frame Loss Ratio Ordered Pairs*

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

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

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

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

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

Property Name: *Frame Loss Ratio Objective*

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

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

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

Allowed values: Type: Percentage

Class: *OneWayAvailabilityPmMetric* (**One Way Availability**)

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

Property Name: *Availability Ordered Pairs*

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

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

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

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

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

Property Name: *Availability Objective*

Reference: MEF 10.4 sec. 8.8.7 and MEF 26.2 sec. 12.13.7

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

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

Allowed values: Type: Percentage

Class: *OneWayHighLossIntervalPmMetric* (**One Way High Loss Intervals**)

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

Property Name: *HLI Ordered Pairs*

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

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

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

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

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

Property Name: *HLI Objective*

Reference: MEF 10.4 sec. 8.8.8 and MEF 26.2 sec. 12.13.8

JSON Name: *oneWayHighLossIntervalsObjective* - **Type:** *integer*

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

Allowed values: Minimum: 0

Class: OneWayConsecutiveHighLossIntervalsPmMetric (**One Way Consecutive High Loss Intervals**)

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

Property Name: *CHLI Ordered Pairs*

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

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

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

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

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

Property Name: *CHLI Consecutive Number*

Reference: MEF 10.4 sec. 8.8.8 and MEF 26.2 sec. 12.13.8

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

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

Allowed values: Minimum: 1

Property Name: *CHLI Objective*

Reference: MEF 10.4 sec. 8.8.9 and MEF 26.2 sec. 12.13.9

JSON Name: *oneWayChliObjective* - **Type:** *integer*

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

Allowed values: Minimum: 1

Class: OneWayCompositePmMetric (**One Way Composite**)

One Way Composite Performance Metric consisting of 7 parameters, a list of Ordered End Point Pairs (referred to as S in MEF 10.4) , a Composite Performance number (referred to as U in MEF 10.4), indicators about Composite Frame Loss, Composite Frame Delay, and Composite Frame Delay Variation (referred to as Wfl, Wfd, and Wfdv in MEF 10.4), Thresholds for Frame Delay and Delay Variation (referred to as DL and Jt in MEF 10.4) and the One Way Composite Objective for this Class of Service Name and these seven parameters. (Note: This Performance Metric is not defined for OVC services).

Property Name: *Composite Ordered Pairs*

Reference: MEF 10.4 sec. 8.8.1.1

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

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

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

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

Property Name: *Composite Performance Indicator Threshold*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *compositePerformanceIndicatorThreshold* - **Type:** *number*

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

Allowed values: Minimum: 0 Maximum: 1

Property Name: *Composite Frame Loss*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *compositeFrameLoss* - **Type:** *integer*

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

Allowed values: Minimum: 0 Maximum: 1

Property Name: *Composite Frame Delay*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *compositeFrameDelay* - **Type:** *integer*

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

Allowed values: Minimum: 0 Maximum: 1

Property Name: *Composite Frame Delay Variation*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *compositeFrameDelayVariation* - **Type:** *integer*

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

Allowed values: Minimum: 0 Maximum: 1

Property Name: *FD Threshold*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *oneWayFdThreshold* - **Type:** *object*

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

Allowed values: Class: TimeDuration

Property Name: *IFDV Threshold*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *oneWayIfdvThreshold* - **Type:** *object*

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

Allowed values: Class: TimeDuration

Property Name: *Composite Objective*

Reference: MEF 10.4 sec. 8.8.10

JSON Name: *cpmObjective* - **Type:** *object*

Description: The PM Objective for this Performance Metric (this CoS and these 7 parameters) expressed as a percentage.

Allowed values: Type: Percentage

Class: OneWayGroupAvailabilityPmMetric (**One Way Group Availability**)

The One Way Group Availability Performance Metric has 2 parameters. The first parameter is a list of sets of Ordered End Point Pairs (referred to as G in MEF 10.4 and MEF 26.2). (Note that this is different than the other Performance Metrics which have a list of Ordered End Point Pairs). The second parameter is the number of sets that must be available (referred to as K in MEF 10.4 and MEF 26.2). The third element is the Performance Objective for the One Way Group Availability.

Property Name: GA Ordered Pairs

Reference: MEF 10.4 sec. 8.8.11 and MEF 26.2 sec. 12.13.10

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

Description: This is a list of lists of Ordered End Point Pairs.

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

Property Name: Minimum Number of Sets Available

Reference: MEF 10.4 sec. 8.8.11 and MEF 26.2 sec. 12.13.10

JSON Name: minimumNumberOfSetsAvailableK - **Type:** integer

Description: Specified number of sets of Ordered End Point Pairs available during characterized percentage of time of measurement.

Allowed values: Minimum: 1

Property Name: GA Objective

Reference: MEF 10.4 sec. 8.8.11 and MEF 26.2 sec. 12.13.10

JSON Name: oneWayGroupAvailabilityObjective - **Type:** object

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

Allowed values: Type: Percentage

Class: OrderedPair (**Ordered Pair**)

Performance Metrics (except for Group Availability) are specified for a list of Ordered End Point Pairs. In other words a Frame Delay objective (for example) is specified from (A to B) and from (C to D). This class defines one Ordered End Point Pair.

Property Name: From EVC/OVC End Point

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

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

Allowed values: [Any String]

Property Name: To EVC/OVC End Point

JSON Name: toCarrierEthernetServiceEndPoint - **Type:** string

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

Allowed values: [Any String]

Class: OrderedPairList (**Sets of Ordered Pairs**)

The Group Availability is specified for sets of Ordered End Point Pairs, e.g., $G1 = \langle \langle A, B \rangle, \langle A, C \rangle \rangle$ and $G2 = \langle \langle A, D \rangle, \langle A, E \rangle \rangle$. 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

808

15.10 Carrier Ethernet Standard Types

Carrier Ethernet Types

Schema File Name: carrierEthernetTypes

Includes classes:

EnabledDisabled (Enabled Disabled)
PcpValueAndDiscard (PCP Value and Discard Enum)
DeiValueAndDiscard (DEI and Discard Enum)
LinkAggType (LinkAgg Enum)
FrameColor (FrameColor Enum)
L2cpAddressSet (L2CP Address Set Enum)
MegList (MEG List Enum)
AvailableMegList (Available MEG List Enum)
PcpValue (PCP Value Enum)
TimeDurationUnits (Time Duration Units)

Type: EnabledDisabled (**Enabled Disabled**)

Enumeration to indicate Enabled/Disabled state of an attribute

Property Name: Enabled Disabled Enum **Type:** enum

Allowed values: Enum: "ENABLED", "DISABLED"

Type: PcpValueAndDiscard (**PCP Value and Discard Enum**)

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

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

Type: DeiValueAndDiscard (**DEI and Discard Enum**)

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

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

Type: LinkAggType (**LinkAgg Enum**)

Type that defines the Link Aggregation types.

Property Name: Link Aggregation Enum **Type:** enum

Allowed values: Enum: "NONE", "2_LINK_ACTIVE_STANDBY", "ALL_ACTIVE", "OTHER"

Type: FrameColor (**FrameColor Enum**)

Property Name: Frame Color Enum **Type:** enum

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

Type: L2cpAddressSet (**L2CP Address Set Enum**)

Property Name: L2CP Address Set Enum **Type:** enum

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

Type: MegList (**Meg List Enum**)

Property Name: Meg List Enum **Type:** enum

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

Type: AvailableMegList (**Available Meg List Enum**)

Property Name: Available Meg List Enum **Type:** enum

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

Type: PcpValue (**PCP Value Enum**)

Property Name: PCP Value Enum **Type:** enum

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

Type: TimeDurationUnits (**Time Duration Units**)

Enum listing the units of measure for a Time Duration

Property Name: Time Duration Units Enum **Type:** enum

Allowed values: Enum: "NS", "US", "MS", "SEC", "MIN", "HOUR", "DAY", "WEEK", "MONTH", "YEAR"

15.11 Carrier Ethernet Utility Classes

Utility Classes and Types

Schema File Name: carrierEthernetUtilityClasses

Includes classes:

MepLevelAndDirection (MEP Specification)
SourceMacAddressLimit (SA Limit)
AggLinkDepth (Link Depth)
FrameDisposition (Frame Disposition)
FrameDispEnum (Frame Disposition Enumeration)
InformationRate (Information Rate)
DataSize (Data Size)
TimeDuration (Time Duration)
Percentage (Percentage)

Class: MepLevelAndDirection (**MEP Specification**)

A 2-tuple that defines the MEG Level and MEP direction.

Property Name: MEG Level

JSON Name: level - **Type:** integer

Description: The MEG level

Allowed values: Minimum: 0 Maximum: 7

Property Name: MEP Direction

JSON Name: mepDirection - **Type:** enum

Description: Indicates whether this is an Up MEP or Down MEP

Allowed values: Enum: "UP", "DOWN"

Class: SourceMacAddressLimit (**SA Limit**)

This limits the number of source MAC addresses that can be used in ingress frame mapped to the End Point of all types over a time interval. It is a recommendation that each Ingress EI Frame with a source MAC address that would exceed the limit be discarded by the Carrier Ethernet Network.

Property Name: Number of Addresses

JSON Name: limit - **Type:** integer

Description: The number of MAC source addresses that will not subject an Ingress EI Frame to discard during the time interval.

Allowed values: Minimum: 1

Property Name: Time Interval

JSON Name: interval - **Type:** object

Description: The time interval in seconds over which the source address limit is evaluated

Allowed values: Class: ShortDuration

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

Property Name: *linkDepth*

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

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

Allowed values: Minimum: 1

Class: *FrameDisposition* (***Frame Disposition***)

A 3-tuple that indicates how Unicast, Multicast, and Broadcast Frames are delivered by the service. For each, the value can be UNCONDITIONAL, the frames are delivered without condition, CONDITIONAL, the frames are delivered based on conditions agreed to between buyer and seller (e.g., based on learned addresses), and DISCARD.

Property Name: *Unicast Frames*

JSON Name: *unicast* - **Type:** *object*

Allowed values: Class: *FrameDispEnum*

Property Name: *Multicast Frames*

JSON Name: *multicast* - **Type:** *object*

Allowed values: Class: *FrameDispEnum*

Property Name: *Broadcast Frames*

JSON Name: *broadcast* - **Type:** *object*

Allowed values: Class: *FrameDispEnum*

Type: *FrameDispEnum* (***Frame Disposition Enumeration***)

Property Name: *frameDispEnum* **Type:** *enum*

Allowed values: Enum: "DISCARD", "DELIVER_UNCONDITIONALLY", "DELIVER_CONDITIONALLY"

Class: *InformationRate* (***Information Rate***)

A value and a unit of measure that specifies an Information Rate.

Property Name: *IR Value*

JSON Name: *irValue* - **Type:** *number*

Description: The value in the information rate. For example if the information rate is 70 kbps this element is 70.

Allowed values: Minimum: 0

Property Name: *IR Units*

JSON Name: *irUnits* - **Type:** *enum*

Description: The unit of measure for the Information Rate. For example if the Information Rate is 70KBPS this element is KBPS. Note that the values are decimal values. 1 KBPS is 1000 bits per second and 1MBPS is 1,000,000 bits per second.

Allowed values: Enum: "BPS", "KBPS", "MBPS", "GBPS", "TBPS", "PBPS", "EBPS", "ZBPS", "YBPS"

Class: *DataSize* (**Data Size**)

A value and a unit of measure that specifies a data size for example for a buffer or a burst size.

Property Name: *Data Size Value*

JSON Name: *dataSizeValue* - **Type:** *integer*

Description: The value in the data size. For example, if a burst size is 40 KBYTES, this element is 40.

Allowed values: [Any Integer]

Property Name: *Data Size Units*

JSON Name: *dataSizeUnits* - **Type:** *enum*

Description: The unit of measure in the data size. For example, if a burst size is 40 KBYTES, this element is KBYTES. Note that the units indicate binary values, e.g., KBYTES is 1024 bytes.

Allowed values: Enum: "BYTES", "KBYES", "MBYES", "GBYES", "TBYTES", "PBYTES", "EBYES", "ZBYTES", "YBYTES"

Class: *TimeDuration* (**Time Duration**)

This class is used to describe durations expressed as a 2-tuple, (value, units). The units are from nanoseconds to years.

Property Name: *Time Duration Value*

JSON Name: *TimeDurationValue* - **Type:** *integer*

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

Allowed values: [Any Integer]

Property Name: *Time Duration Units*

JSON Name: *TimeDurationUnits* - **Type:** *object*

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

Allowed values: Type: *TimeDurationUnits*

Type: *Percentage* (**Percentage**)

This is a number of percent - a floating point number between 0 and 100

Property Name: *Percentage* **Type:** *number*

Allowed values: Minimum: 0 Maximum: 100

16 Service Attributes Not Included

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

- They are part of the product agnostic envelope (i.e., several Identifiers), and
- The Service Attribute has a single constant value for Subscriber Ethernet services defined in MEF 6.3 [7]. There is no benefit to including these in the API since the only possible use would be to validate that they are correct. There is no information to be gleaned if they are correct.

These Service Attributes are listed in the following table.

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, because the Multiple EVC Service Level Specification Service Attribute is not currently part of MEF 7.4 [8].
EVC End Point Service Attributes	EVC End Point Ingress Bandwidth Profile	Bandwidth Profiles must be none as specified in MEF 6.3 [R3]
EVC End Point Service Attributes	EVC End Point Egress Bandwidth Profile	Bandwidth Profiles must be none as specified in MEF 6.3 [R4].
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"

Table 8 – Service Attributes Not Included in Schemas

17 References

- [1] IEEE Std 802.1AX-2014, *Link Aggregation*, December 2014
- [2] IEEE Std 802.3-2012, *IEEE Standard for Ethernet*, August 2012
- [3] IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, by Scott Bradner, March 1997
- [4] IETF RFC 3444, *On the Difference between Information Models and Data Models*, January 2003
- [5] IETF RFC 8174, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, by Barry Leiba, May 2017. Copyright © IETF Trust and the persons identified as the document authors (2017). All Rights Reserved
- [6] MEF 4, *Metro Ethernet Network Architecture - Part 1: Generic Framework*, May 2004
- [7] MEF 6.3, *Subscriber Ethernet Services Definitions*, November 2019
- [8] MEF 7.4, *Carrier Ethernet Services Information Model*, December 2020
- [9] MEF 10.4, *Subscriber Ethernet Service Attributes*, December 2018
- [10] MEF 26.2, *External Network Network Interfaces (ENNI) and Operator Service Attributes*, August 2016
- [11] MEF 45.1, *Layer 2 Control Protocols in Ethernet Services*, December 2018
- [12] MEF 55.1, *Lifecycle Service Orchestration (LSO): Reference Architecture and Framework*, January 2021
- [13] MEF 57.2, Draft Release 3 *Product Order Management Requirements and Use Cases*, January 2022
- [14] MEF 78.1, *MEF Core Model*, July 2020
- [15] MEF 79, *Address, Service Site, and Product Offering Qualification Management, Requirements and Use Cases*, November 2019
- [16] MEF 80, *Quote Management Requirements and Use Cases*, May 2021
- [17] MEF 81, *Product Inventory Management, Requirements and Use Cases*, November 2019
- [18] MEF 87, *LSO Cantata and LSO Sonata Product Offering Qualification Management API – Developer Guide*, January 2022

- 862 [19] MEF 115, LSO Cantata and LSO Sonata Quote Management API – Developer Guide,
863 January 2022
- 864 [20] MEF 121, LSO Cantata and LSO Sonata Address Management API – Developer
865 Guide, January 2022
- 866 [21] MEF W123, LSO Cantata and LSO Sonata Product Order Management API – Devel-
867 oper Guide, January 2022
- 868