MEF

Working Draft

MEF W106 v1.12

LSO Sonata Product Specification   Access E-Line

Product Schema Guide

June 2021

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

Disclaimer

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

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

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

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

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

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

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

Table of Contents

List of Figures

List of Tables

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 requirements document and its associated software artifacts (JSON Schemas) defines and describes the product-specific information used in LSO Sonata APIs for a set of Business Functions—specifically, Product Offering Qualification, Quoting, Ordering and Inventory—for an Access E-Line product. The document starts with an overview of LSO Sonata and the Access E-Line Carrier Ethernet service. It then provides a basic information model for the MEF Carrier Ethernet Service Attributes. The final sections describe the Data Model in both a technology-independent way and a technology-specific way focused the JSON Schema associated with this specification.

This document can be thought of as a user's guide for the Access E-Line 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 Operator Ethernet Services (of which Access E-Line is one) is MEF 26.2 [7]. The service, itself, is specified in MEF 51.1 [9] based on the Service Attributes defined in MEF 26.2. MEF 26.2 specifies 72 Service Attributes to describe the various components that compose an Operator Ethernet Service. This document defines a data model that includes 56 of those Service Attributes and also lists the Service Attributes that are not included in the data model and the reason why each is not included.

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. If the reference includes an asterisk (*), the definition has been adapted from the original.

Term	Definition	Reference
Access E-Line	An E-Access Service based on a point-to-point OVC.	MEF 51.1 [9]
Business Applications	The Service Provider functionality supporting Business Management Layer functionality (e.g., product catalog, order management, billing, relationship management, etc.)	MEF 55.1 [10]
BUS	See *Business Applications*	MEF 55.1 [10]
Business Functions	In the context of this document Business Functions refer to *Product Offering Qualification (POQ)*, *Order Management*, *Quote Management*, and *Inventory Management*.	This Document
Buyer	For the purposes of this document, a Buyer is the Service Provider who is ordering from an Operator (aka, Seller).	MEF 57.1* [11]
Data Model	A representation of concepts of interest to an environment in a form that is dependent on data repository, data definition language, query language, implementation language, and/or protocol (typically, but not necessarily, all five).	MEF 78.1 [12]

Term	Definition	Reference
E-Access Service	An Operator Ethernet Service that associates at least one OVC End Point that is at a UNI and at least one OVC End Point that is at an ENNI.	MEF 51.1 [9]
Ethernet 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 [7]
ENNI	*See Ethernet Network Network Interface*	MEF 26.2 [7]
ENNI Frame	An Ethernet Frame that is exchanged across an ENNI between an Operator and a Service Provider or another Operator.	MEF 26.2* [7]
Ethernet Service Provider	An organization that provides to a Subscriber a connectivity service that carries Ethernet Frames irrespective of the underlying technology and that is specified using Service Attributes as defined in a MEF Standard.	MEF 10.4* [6]
Ethernet Subscriber	The end-user of an Ethernet Service	MEF 10.4 [6]
Ethernet UNI	The demarcation point between the responsibility of the Ethernet Service Provider and the Ethernet Service Subscriber.	MEF 10.4 [6]
Ethernet Virtual Connection	An association of EVC End Points	MEF 10.4 [6]
EVC	See *Ethernet Virtual Connection*	MEF 10.4 [6]
EVC End Point	A construct at a (Ethernet) UNI that selects a subset of the Service Frames that pass over the UNI.	MEF 10.4 [6]
External Interface	Either a UNI or an ENNI.	MEF 4 [3]
External Interface Frame	Either a Service Frame or an ENNI Frame	MEF 26.2 [7]
Information Model	A representation of concepts of interest to an environment in a form that is independent of data repository, data definition language, query language, implementation language, and protocol	MEF 78.1 [12]
Inventory	See *Product Inventory*	MEF 81 [14]
Milestone	An event that occurs during the fulfillment process that indicates a significant step in the process has been completed	This document
Operator	The administrative entity of a Carrier Ethernet Network	MEF 26.2 [7]
Operator UNI Service Attributes	The Service Attributes used to describe an agreement between a Service Provider and an Operator for a UNI associated with an OVC.	MEF 26.2 [7]
Order	See *Product Order*	This document
Operator Ethernet Service	An Ethernet Service that is provided by an Operator to another Operator or to an Ethernet Service Provider.	MEF 51.1 [9]
Operator Virtual Connection	An association of OVC End Points	MEF 26.2 [7]
OVC	See *Operator Virtual Connection*	MEF 26.2 [7]
OVC End Point	A logical entity at a given External Interface that is associated with a distinct set of frames passing over that External Interface.	MEF 26.2 [7]
Product	One or more goods or services that is or may be sold to a Buyer by a Seller.	MEF 79 [13]

Term	Definition	Reference
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 [13]
Product Order	One or more Product Order Items formulated into a fulfilment request made by a Buyer to a Seller.	This document
Product Order Item	An individual item included in a Product Order that describes the action to be taken on a Product or Product Offering by the Seller. The objective is for the Seller to complete the fulfilment process of this Product or Product Offering at the place defined by the Buyer.	This document
Product Inventory	The inventory managed by the Seller resulting from Order completion.	MEF 81 [14]
Product Specification	A Product Specification defines the template or detailed description from which Product Offerings can be defined.	MEF 79 [13]
POQ	See *Product Offering Qualification*	MEF 79 [13]
Product Offering Qualification	One or more Product Offering Qualification Items formulated into a requirement made by a Buyer to a Seller.	MEF 79 [13]
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 [13]
Quote	One or more Quote Items formulated into a request for pricing of a Product or Product Offering made by a Buyer to a Seller.	This document
Quote Item	An individual item included in a Quote that describes the Buyer's interest in a price from the Seller for a Product or Product Offering. The objective is to determine the charges and timeframe for the Seller to deliver this item as described by the Buyer.	This document
Seller	For the purposes of this document, a Seller is the Operator who is providing the product to the Buyer.	MEF 57.1* [11]
Service Attribute	Specific information that is agreed between the provider and the user of the service, that describes some aspect of the service behavior or capability.	MEF 10.4 [6]
Service Frame	An Ethernet Frame that is exchanged across a UNI between a Subscriber and a Service Provider.	MEF 10.4 [6]
Service Provider	In the context of this document, a Service Provider is an Ethernet Service Provider. In this document we use Service Provider to include Super Operator as specified in MEF 26.2 (also referred to as SP/SO).	This Document
Sonata	The Management Interface Reference Point supporting the management and operations interactions (e.g., ordering, billing, trouble management, etc.) between two network providers (e.g., Service Provider Domain and Partner Domain).	MEF 55.1 [10]

Term	Definition	Reference
Subscriber	In the context of this document, a Subscriber is an Ethernet Subscriber.	This Document
UNI	An Ethernet User Network Interface	MEF 10.4 [6]

Table 1 – Terminology and Abbreviations

4 Introduction

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

Access E-Line is an Operator Ethernet Service, i.e., an Ethernet Service that is arranged between a Service Provider and an Operator. The Service Attributes that are agreed to between the parties are defined in MEF 26.2 [7]. The Service definition which is, in effect, a set of constraints on the values of the Service Attributes, is provided in MEF 51.1 [9].

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

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

The model assumes an Access E-Line product based on an existing ENNI, so the schemas for ENNI are only included for Inventory. There are a small number of ENNI attributes that might require modification based on the definition of services (like Access E-Line) that traverse the ENNI. To address this issue in the most common situations, a couple of additional OVC End Point Service Attributes have been defined (see section 10.4.3).

The document contains the following sections:

- An overview of LSO Sonata (section 5)
- An overview of the Access E-Line Service (section 6)
- Data Model Design Principles (section 7)
- An abbreviated Information Model for Access E-Line and explanation of the organization of the Service Attributes in MEF 26.2 (section 9)
- Organization of the data model for Access E-Line (section 10)
- The relationship between the entities in the service (section 11)

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

- Section 12 contains the details of the Service Attributes for OVC, OVC End Point, UNI, and ENNI
- Section 13 contains all of the common classes and types referenced by the Service Attributes
- Section 14 lists the Service Attributes that are not included in the data models

5 Overview of LSO Sonata

MEF 55.1 [10] describes the Reference Architecture for Lifecycle Service Orchestration (LSO) of MEF-defined services. MEF 55.1 defines seven LSO Interface Reference Points (see Figure 1) that are abstract interconnection points between different entities—either within the Service Provider domain (intra-domain) or between Service Provider and other business entities (inter-domain). One of these LSO Reference Points is LSO Sonata which defines the abstract boundary point between a Service Provider and an Operator which can provide wholesale (offnet) connectivity services to the Service Provider. It is at this boundary point – LSO Sonata – that the Service Provider and the Operator access each other's Business Functionalities. Examples of inter-provider Business Functionalities include address qualification, product offering qualification, quotes, ordering, trouble ticketing, and billing/settlement.

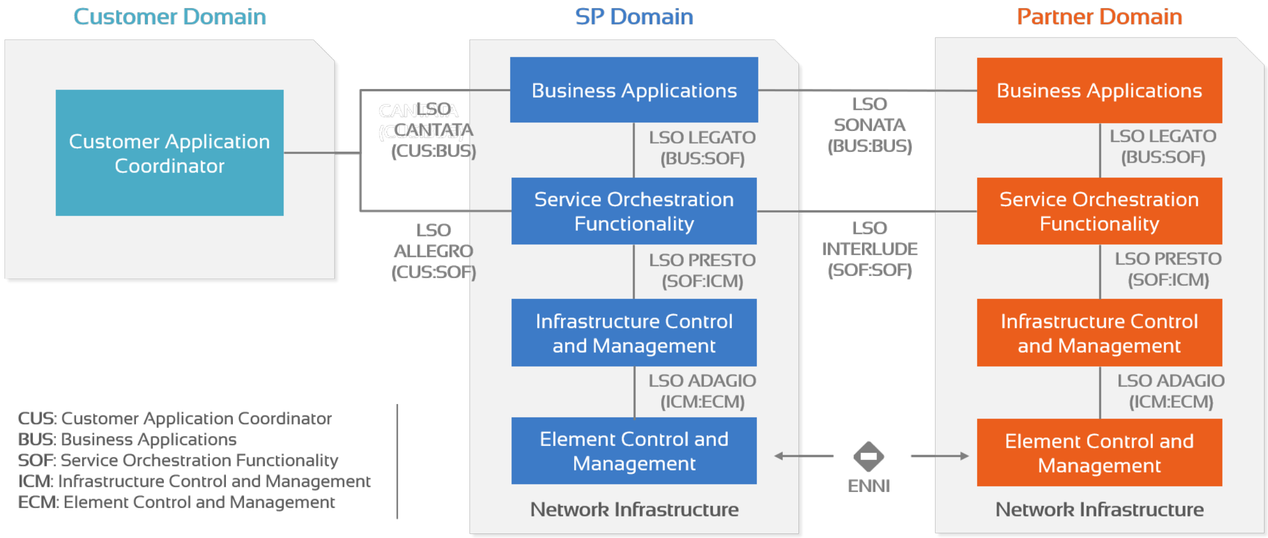
Customer Domain
SP Domain
Partner Domain
Customer Application Coordinator
LSO CANTATA (CUS:BUS)
LSO ALLEGRO (CUS:SOF)
Business Applications
LSO LEGATO (BUS:SOF)
Service Orchestration Functionality
LSO PRESTO (SOF:ICM)
Infrastructure Control and Management
LSO ADAGIO (ICM:ECM)
Element Control and Management
Network Infrastructure
LSO SONATA (BUS:BUS)
LSO INTERLUDE (SOF:SOF)
ENNI
Business Applications
LSO LEGATO (BUS:SOF)
Service Orchestration Functionality
LSO PRESTO (SOF:ICM)
Infrastructure Control and Management
LSO ADAGIO (ICM:ECM)
Element Control and Management
Network Infrastructure
CUS: Customer Application Coordinator
BUS: Business Applications
SOF: Service Orchestration Functionality
ICM: Infrastructure Control and Management
ECM: Element Control and Management

Figure 1 – LSO Sonata Reference Diagram

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

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

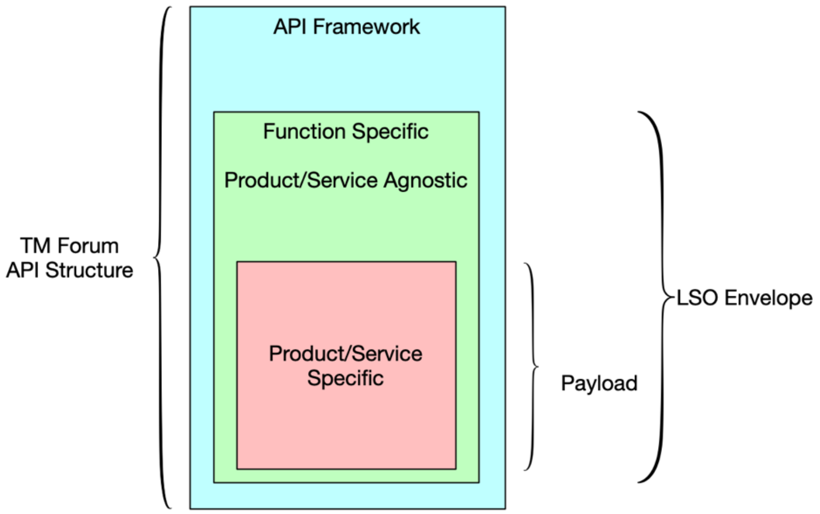

Figure 2 – LSO Sonata API Structure

This document describes the product-specific data model for a MEF 3.0 Access E-Line service as defined in MEF 51.1 [9].

6 Overview of Access E-Line Services

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

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

6.1 EVCs

A Subscriber Ethernet Service is built on an Ethernet Virtual Connection (EVC) which is an association of EVC End Points located at Subscriber Ethernet UNIs. EVCs and their Service Attributes are described in MEF 10.4 [6]. There are three types of EVCs: point-to-point, multipoint-to-multipoint, and rooted-multipoint. MEF 6.3 [4] defines three types of EVC services based on EVC type: E-Line (Ethernet Private Line and Ethernet Virtual Private Line), E-LAN (Ethernet Private LAN and Ethernet Virtual Private LAN), and E-Tree (Ethernet Private Tree and Ethernet Virtual Private Tree), respectively.

EVCs are sold to a Subscriber by an Ethernet Service Provider. In most cases, the Ethernet Service Provider is also an Operator that provides some or all of the network infrastructure and 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.

6.2 OVCs

If one or more of the UNIs associated by an EVC is not accessible to the Ethernet Service Provider’s network, the Service Provider must purchase access to the UNI or UNIs from an Operator that does have access. In this case, the EVC is composed of a set of Operator Virtual Connections (OVCs) where each OVC represents the part of the EVC that is provided by one Operator. OVCs, ENNIs, and their Service Attributes are described in MEF 26.2 [7]. As with EVCs, there are three types of OVCs: point-to-point, multipoint-to-multipoint, and rooted multipoint.

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

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

MEF 51.1 [9] describes a set of Operator Ethernet Services. It describes three generic service types, O-LINE, O-LAN, and O-TREE, based on the three types of OVCs. In MEF 51.1, an OVC that contains at least one UNI is referred to as an *access OVC* and an OVC that contains only ENNIs is referred to as a *transit* OVC. MEF 51.1 also defines some specific service types. An O-LINE service that includes one UNI is an Access E-Line Service and an O-LINE service that includes only ENNIs is a Transit E-Line Service. Similarly, there are Access E-LAN and Transit E-LAN Services.

6.3 Access E-Line

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

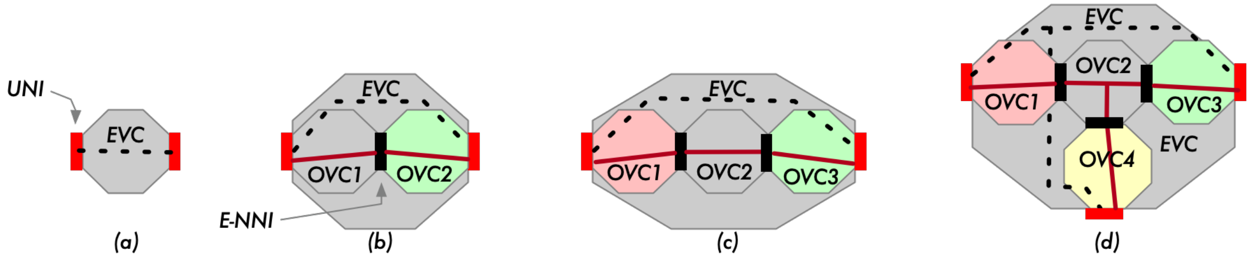

Figure 3 – OVCs and Access E-Line Services

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

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

6.4 Roles

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

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

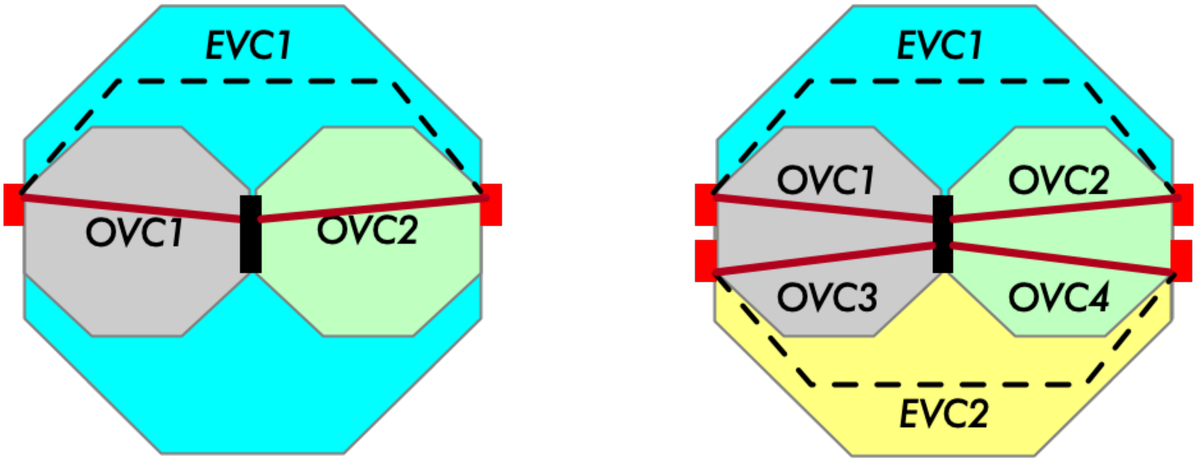

Figure 4 – Service Providers and Network Operators

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

As a result, different aspects (Service Attributes) of the ENNI are relevant to different parties. Clearly the two Operators care about some aspects of the ENNI (e.g., the physical characteristics). Each of the Service Providers cares about certain aspects of its OVCs that terminate at the ENNI. The ENNI Service Attributes are partitioned into groups based on which organizations need to agree on which attributes. This is discussed further in section 10.2.

7 Data Model Design Principles and Assumptions

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

- The requirements for which attributes are necessary at POQ, Quote and Order time differ (each should be a subset, mathematically speaking, of the successive steps). If the same set of attributes, again mathematically speaking, are needed across POQ, Quote and Order, then the number of use-cases goes down significantly.
- The requirements for which Service Attributes can be specified when changing an Access E-Line are different from when "installing" one. (general principle)
- The data model does not support qualifying, quoting for, or ordering an ENNI. ENNIs are assumed to be created in the inventory as a result of the on-boarding process between two operators. ENNIs do need to be able to be retrieved from the inventory so that they can be referenced in the Access E-Line order (or qualification or quote).[1]
- The location and physical layer of a UNI cannot be changed once it is ordered; instead, this is handled as an installation (UNI at new location) and a disconnect (UNI at previous location), as there is often a requirement for a smooth transition with minimum downtime.

These assumptions are shown in Table 2.

Business Function	Operation	UNI	OVC	ENNI
POQ	Install	✓	✓	-
	Change	✓	✓	-
Quote	Install	✓	✓	-
	Change	✓	✓	-
Order	Install	✓	✓	-
	Change	✓	✓	-
Inventory	Retrieve	✓	✓	✓

Table 2 – Allowed operations for each Business Function

[1] There are a small number of ENNI attributes that might require modification based on the definition of services (like Access E-Line) that traverse the ENNI. To address this issue in the most common situations, a couple of additional OVC End Point Service Attributes have been defined. See section 10.4.3.

8 Milestones

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

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

Table 3 – Example Milestones for Order

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

9 Information Model for Access E-Line Product Data Model

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

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

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

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

All of these components are shown in the following diagram.

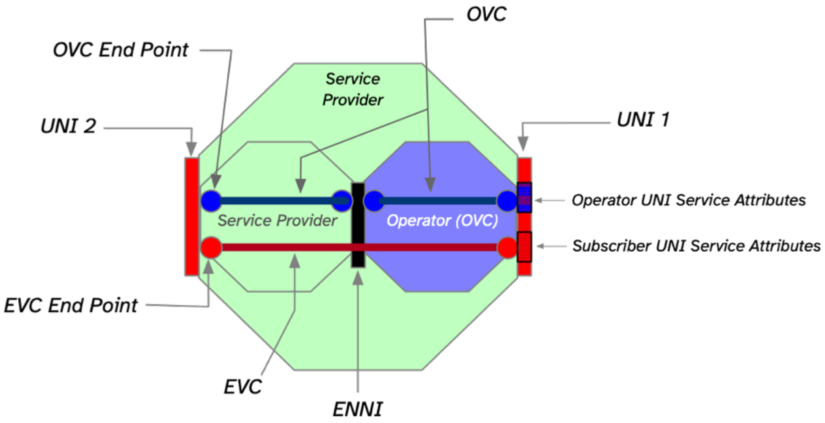

Figure 5 – Components of an EVC and OVC

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

9.1 Organization of Service Attributes

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

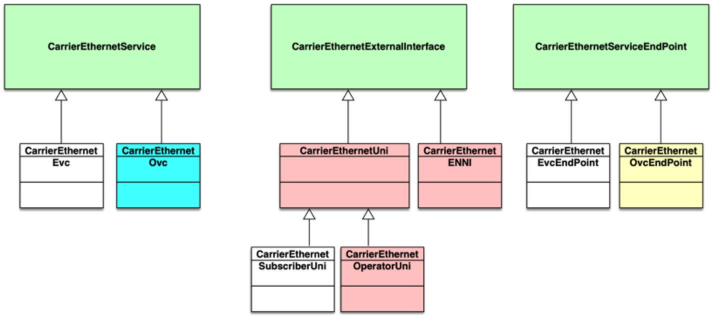
CarrierEthernetService
CarrierEthernetExternalInterface
CarrierEthernetServiceEndPoint
CarrierEthernet Evc
CarrierEthernet Ovc
CarrierEthernetUni
CarrierEthernet ENNI
CarrierEthernet EvcEndPoint
CarrierEthernet OvcEndPoint
CarrierEthernet SubscriberUni
CarrierEthernet OperatorUni

Figure 6 – Carrier Ethernet Super classes

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

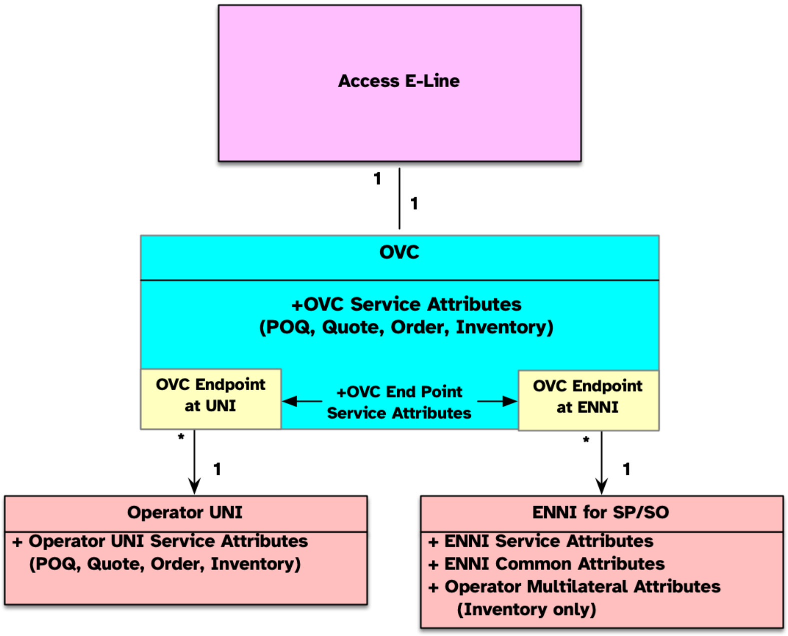
Access E-Line
1
1
OVC
+OVC Service Attributes
(POQ, Quote, Order, Inventory)
OVC Endpoint at UNI
+OVC End Point Service Attributes
OVC Endpoint at ENNI
*
1
*
1
Operator UNI
+ Operator UNI Service Attributes
(POQ, Quote, Order, Inventory)
ENNI for SP/SO
+ ENNI Service Attributes
+ ENNI Common Attributes
+ Operator Multilateral Attributes
(Inventory only)

Figure 7 – Access E-Line Information Model

10 Data Models for Access E-Line Product

The data models for the Access E-Line product configuration in the Sonata API are expressed as a set of JSON schemas that accompany this document. This section explains the organization and structure of the schemas.

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

10.1 Technology Neutral Organization of the Data Models

This standard defines three product specifications that can be used to manage Access E-Line services:

- Access E-Line OVC (including the OVC End Points)
- UNI – This may be a new order or an existing UNI that is adding the new OVC
- ENNI – This is an existing ENNI that is adding the OVC. Creating/modifying ENNIs is not supported in this specification.

10.2 Organization and Structure of the Schemas

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

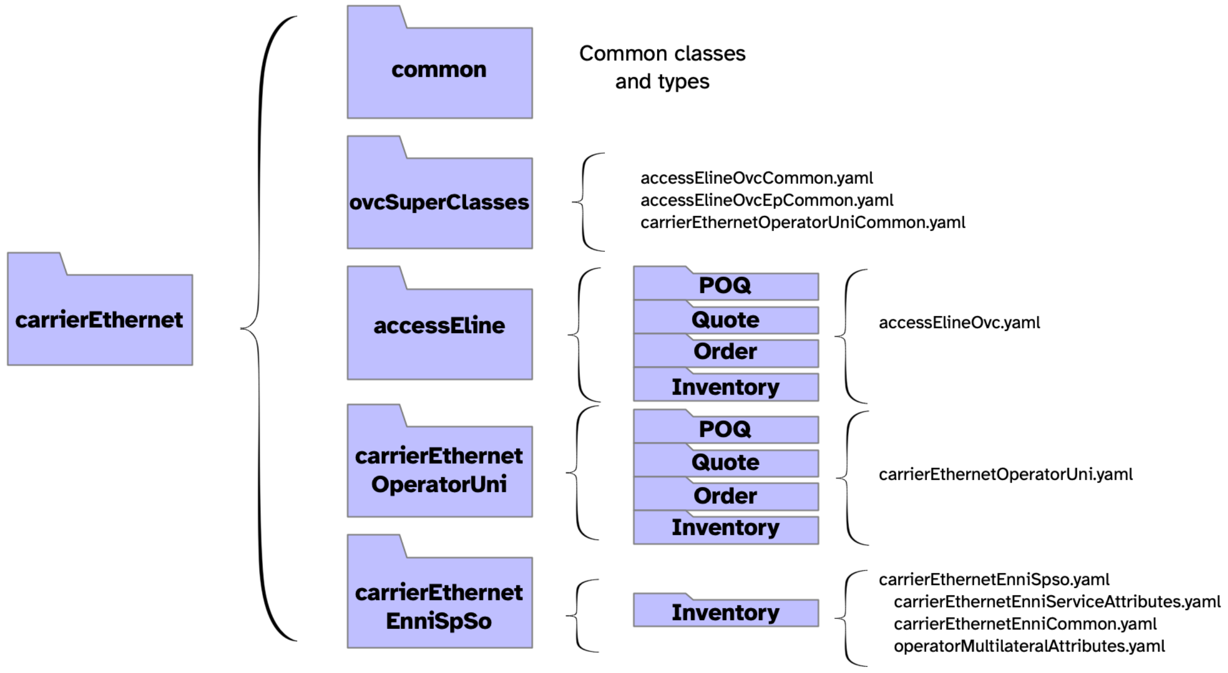
carrierEthernet
common
Common classes and types
ovcSuperClasses
accessElineOvcCommon.yaml
accessElineOvcEpCommon.yaml
carrierEthernetOperatorUniCommon.yaml
accessEline
POQ
Quote
Order
Inventory
accessElineOvc.yaml
carrierEthernet OperatorUni
POQ
Quote
Order
Inventory
carrierEthernetOperatorUni.yaml
carrierEthernet EnniSpSo
Inventory
carrierEthernetEnniSpso.yaml
carrierEthernetEnniServiceAttributes.yaml
carrierEthernetEnniCommon.yaml
operatorMultilateralAttributes.yaml

Figure 8 – Schema File Organization

For each of the major product components (OVC, UNI, and ENNI), there are *POQ*, *Quote*, *Order* and *Inventory* folders which contain the relevant schema files (as noted previously, ENNI just includes *Inventory* since the schemas assume an existing ENNI).

Since the schemas associated with each Business Function include the same Service Attributes (JSON properties) differing only in terms of which are required and which are optional, the Service Attribute definitions are defined in a set of common files in the *ovcSuperClasses* folder. These common classes are referenced in the relevant product component schema files with the list of required Service Attributes appropriate for the Business Function. For example, each *carrierEthernetOperatorUni.yaml* file (for POQ, Quote, Order, Inventory) includes:

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

But the list of "required" Service Attributes that follows this reference is different for each Business Function.

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.

The ENNI *inventory* folder includes four schemas:

- The ENNI Service file includes attributes that represent the agreement between the Service Provider and the Operator for each OVC that terminates at the ENNI for one Service Provider.
- The ENNI Common file includes the attributes that describe the physical and logical characteristics of the ENNI and are agreed to by the Operators on opposite sides of the ENNI. They also need to be agreed to by each Super Operator or Service Provider that agrees to the ENNI Service Attribute S-VLAN ID Control = *Full*.
- The Operator Multilateral Attributes includes attributes that apply to the ENNI but are relevant both Operators, and each Super Operator or Service Provider that agrees to the ENNI Service Attribute S-VLAN ID Control = *Full*.
- The fourth schema is *carrierEthernetEnniForSPSO.yaml* file which is the composition (allOf) of the other three.

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

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

- Inclusion: i.e., some Service Attributes may be included for some Business Functions and have no relevance for other Business Functions and are therefore not included in the data model for those Business Functions. (Note that at this time there is no "inclusion" difference, a Service Attribute is either in all four schemas or none of them.)

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

If a Service Attribute is *optional* for some Business Function, it may or may not be included in the product configuration for operations related to that Business Function. If it is not included, the Service Provider will provide a default value if appropriate for the Service Attribute. If the value of the Service Attribute is important to the Subscriber, it should be included. If it is not included:

- The value is not important to the Subscriber, or
- The Subscriber knows the Service Provider's default value, or
- The controlling MEF Standard specifies a set of constraints on the value and the Subscriber's requirements fall within those constraints.

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

10.3 Example Schema

Following is an example schema (from *Carrier_Ethernet_Link_Aggregation.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 and ENNI.
        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:
            - conversationIDs
            - aggLinkList
    ConversationIdRange:
        type: object
        description:  >-
            A range of ConversationID (either a VLAN Id or 0 for untagged
            frames) allowing three forms: start, start → start, and start
            → end
        properties:
            start:
                description:  >-
                    The starting Conversation ID of the range or the only Conversation
                    ID if there is no end value
                # start must be less than or equal to end
                # See additional validation notes in the Schema Guide
                type: integer
                minimum: 0
                maximum: 4094
            end:
                description: The final Conversation ID in the range
                # end, if specified, must be greater than or equal to start
                # See additional validation notes in the Schema Guide
                type: integer
                minimum: 0
                maximum: 4094
        required:
                - start
```

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

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

10.4 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.

10.4.1 Naming Conventions

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

10.4.2 OVC End Point Service Attributes

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

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

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

10.4.3 Additional OVC End Point Attributes

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

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

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

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

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

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

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

11 Relationship Between Entities

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

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

The Access E-Line Service is an Operator Ethernet Service. The Access E-Line Service is associated with exactly one OVC (with appropriate Service Attribute values), but it depends on two other products, an ENNI and a UNI. This is noted in the following table. Specification of the UNI and the ENNI are mandatory at Install of the service, neither of them can be part of a Change order.

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

	Product Relationship Role	Install	Change	Product Specification	Multiple Allowed at POQ and Quote?
Access E-Line	ENNI_REFERENCE	Mandatory	Not Allowed	ENNI	Yes
Access E-Line	UNI_REFERENCE	Mandatory	Not Allowed	UNI	No

Table 4 – Product Relationship Roles

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

	Place Relationship Role	Install	Change
UNI	INSTALL_LOCATION	Mandatory	Not Allowed

Table 5 – Place Relationship Role

12 Access E-Line Service Attributes

The Service Attributes are listed in groups:

- Access E-Line OVC Service Attributes
- OVC End Point Service Attributes
- Operator UNI Service Attributes
- ENNI Service Attributes
- Operator Multilateral Attributes
- ENNI Common 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. The third line contains usage information, i.e., for which Business Functions (POQ, Quote, Order, Inventory) the Service Attribute is included in the product configuration, and if included, whether it is optional or required. 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 26.2 Service Attributes are included in the data models. 56 of the 72 Service Attributes are included. The 16 Service Attributes that are not included are also listed in section 14. Some Service Attributes are not included because they are included in the Product Independent information portion of the API (e.g., many 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 ENNI Frame Format Service Attribute must be "802.3-2012) or because the value is constrained to a single value for Access E-Line (e.g., the OVC Type Service attribute for an Access E-line Service must be point-to-point).

Conversely, there are a few Service Attributes included in the schemas that are not included in MEF 26.2.

The following 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 background) followed by a list of properties in the class (yellow background).

12.1 Access E-Line OVC and OVC End Point Service Attributes

Access Eline OVC
Schema File Name: accessElineOvc
Includes classes:
AccessElineOvc (Access E-Line OVC Service Attributes)
AccessElineOvcEndPoint (Access E-Line OVC End Point Service Attributes)

[redacted]

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

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

[redacted]

Property Name: *Access E-Line OVC Service Attributes*
JSON Name: *accessElineOvcAttributes* - **Type:** *object*

Allowed values: Compose (allOf): Class: AccessElineOvcCommon

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

[redacted]

Property Name: *UNI OVC End Point - Access E-Line*
Reference: MEF 26.2 sec. 16
JSON Name: *uniEp* - **Type:** *object*

Description: The OVC EP object for the OVC EP at the UNI

Allowed values: Class: AccessElineOvcEndPoint

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

[redacted]

Property Name: *ENNI OVC End Point - Access E-Line*
Reference: MEF 26.2 sec. 16
JSON Name: *enniEp* - **Type:** *object*

Description: The OVC EP object for the OVC EP at the ENNI

Allowed values: Class: AccessElineOvcEndPoint

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

[redacted]

Class: AccessElineOvcEndPoint ***(Access E-Line OVC End Point Service Attributes)***

An OVC End Point is a logical entity at a given External Interface that is associated with a distinct set of frames passing over that External Interface. The OVC End Point schema is included in the definitions section of the OVC schema..

[redacted]

Property Name: *Access Eline OVC EP Service Attributes*
JSON Name: *accessElineOvcEpAttributes* - **Type:** *object*

Allowed values: Compose (allOf): Class: AccessElineOvcEpCommon

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

Access Eline OVC Common
Schema File Name: accessElineOvcCommon
Includes classes:
AccessElineOvcCommon (Access E-Line OVC Common)

Class: AccessElineOvcCommon ***(Access E-Line OVC Common)***

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

Property Name: *OVC Maximum Frame Size Service Attribute*
Reference: MEF 26.2 sec. 12.6
JSON Name: *maximumFrameSize* - **Type:** *integer*

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

Allowed values: Minimum: 1526

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

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

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

Property Name: *OVC CE-VLAN ID Preservation Service Attribute*
Reference: MEF 26.2 sec. 12.7
JSON Name: *ceVlanIdPreservation* - **Type:** *enum*

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

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

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

Property Name: *OVC CE-VLAN PCP Preservation Service Attribute*
Reference: MEF 26.2 sec. 12.8
JSON Name: *cTagPcpPreservation* - **Type:** *object*

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

Allowed values: Class: EnabledDisabled

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

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

Property Name: *OVC CE-VLAN DEI Preservation Service Attribute*
Reference: MEF 26.2 sec. 12.9
JSON Name: *cTagDeiPreservation* - **Type:** *object*

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

Allowed values: Class: EnabledDisabled

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

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

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

Property Name: *OVC List of Class of Service Names Service Attribute*
Reference: MEF 26.2 sec. 12.12
JSON Name: *listOfClassOfServiceNames* - **Type:** *List of string*

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

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

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

Property Name: *OVC Service Level Specification Service Attribute*
Reference: MEF 26.2 sec. 12.13
JSON Name: *carrierEthernetSls* - **Type:** *List of object*

Description: Technical details of the service level in terms of Performance Objectives, agreed between the SP/SO and the Operator as part of the Service Level Agreement. For each Performance Metric included in the SLS, a Performance Objective must be included as well as the parameter values needed for the Performance Metric.

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

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

Property Name: *OVC Frame Delivery Service Attribute*
Reference: MEF 26.2 sec. 12.14
JSON Name: *frameDisposition* - **Type:** *object*

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

Allowed values: Class: FrameDisposition

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

Property Name: *OVC Available MEG Level Service Attribute*
Reference: MEF 26.2 sec. 12.15
JSON Name: *availableMegLevel* - **Type:** *enum*

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

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

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

Property Name: *OVC L2CP Address Set Service Attribute*
Reference: MEF 45 sec. 8.1 and MEF 26.2 sec. 12.16
JSON Name: *ovcL2cpAddressSet* - **Type:** *enum*

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

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

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

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

Access Eline OVC EP Common
Schema File Name: accessElineOvcEpCommon
Includes classes:
AccessElineOvcEpCommon (Access Eline OVC EP Common)

[redacted]

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

This class contains all of the Access E-Line OVC End Point Services Attributes that are used in all four Business Functions (POQ, Quote, Order, Inventory). At this time, this includes all of the Access Eline OVC End Point Service Attributes. This class is used to compose the AccessElineOvcEndPoint class which has one file for each Business Function that differ in terms of the Service Attributes that are "Required" for the Business Function (rather than "Optional").

[redacted]

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

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

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

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

[redacted]

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

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

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

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

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

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

An OVC associating an OVC End Point at a UNI whose value of the OVC End Point Map Service Attribute contains multiple CE-VLAN ID values **must** have the value of the OVC CE-VLAN ID Preservation Service Attribute = Preserve.
Coding hint: This must be FormU at UNI and FormE at ENNI. They can only use LIST (not ALL or UT/PT)

[redacted]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Validation Notes:

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

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

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

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

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

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

Description: Bandwidth Profile Flow parameters for all egress Service Frames mapped to the OVC End Point.

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

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

Validation Notes: Per MEF 51.1 this **must** be Disabled at ENNI but not at the UNI.

If specified (at the UNI), ***AccessElineOvcEndPoint.egressBwpPerEgressEquivalenceClassName*** MUST NOT be specified.
Coding hint: Per MEF 51.1 this must be Disabled at ENNI

Property Name: *Ingress Bandwidth Profile per Class of Service Name Service Attribute*
Reference: MEF 26.2 sec. 16.12
JSON Name: *ingressBandwidthProfilePerClassOfServiceName* - **Type:** *List of object*

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

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

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

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

Property Name: *Egress Bandwidth Profile per Egress Equivalence Class Name Service Attribute*
Reference: MEF 26.2 sec. 16.13
JSON Name: *egressBwpPerEgressEquivalenceClassName* - **Type:** *List of object*

Description: For each EEC Name listed, Bandwidth Profile Flow parameters, for all egress Service Frames mapped to that EEC Name at the OVC End Point.

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

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

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

Property Name: *OVC End Point Aggregation Link Depth Service Attribute*
Reference: MEF 26.2 sec. 16.14
JSON Name: *aggregationLinkDepth* - **Type:** *List of object*

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

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

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

Validation Notes:

1. Ensure that this is not included in UniOvcEP.
2. The ENNI must be Link Aggregation type ALL ACTIVE

3. The Depth element must not be greater than the number of links in the ENNI
4. **EnniCommon.portconversation** must be configured to enable the depth specified by this attribute.

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

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

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

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

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

Validation Notes: Required in Inventory if specified in Order

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Property Name: *OVC End Point Port Conversation Map Service Attribute*
Reference: MEF 26.2 sec. 13.2
JSON Name: *ovcEndPointPortConversationToAggregationLinkMap* - **Type:** *List of object*

Description: A mapping of Port Conversation IDs (i.e., VLAN IDs) to an ordered list of physical links if ***EnniService.sVlanIdControl*** is *FULL* and there are multiple physical links in the ENNI and ***EnniCommon.linkAggregation*** is *ALL_ACTIVE*. Only S-VLAN IDs that are listed in the ***AccessElineOvcEndPoint.ovcEndPointMap*** can be listed in this attribute.
If ***EnniService.sVlanIdControl*** is *PARTIAL* then the Service Provider does not have full visibility of all the S-VLAN IDs mapped to each physical link, and hence cannot pick values that avoid overloading certain links. In this case, the ENNI Port Conversation to Aggregation Link Map Common Attribute must be agreed between the relevant Operators.

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

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

Validation Notes: This can only be specified for an OVC End Point at an ENNI. If this is a non-empty list, then ***EnniService.sVlanIdControl*** **must be** *FULL* and ***EnniCommon.linkAggregation*** **must be** *ALL_ACTIVE*.
Coding hint: Not used for OVC End Point at UNI (only ENNI)

12.2 Operator Uni Service Attributes

Carrier Ethernet Operator UNI
Schema File Name: carrierEthernetOperatorUni
Includes classes:
OperatorUNI (Operator UNI Service Attributes)

[redacted]

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

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

[redacted]

Property Name: *Operator UNI Service Attributes*
JSON Name: *operatorUniAttributes* - **Type:** *object*

Allowed values: Compose (allOf): Class: OperatorUniCommon

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

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

[redacted]

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

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

[redacted]

Property Name: *Operator UNI List of Physical Links Service Attribute*
Reference: MEF 26.2 sec. 14.2, 14.3
JSON Name: *listOfPhysicalLinks* - **Type:** *List of object*

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

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

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

[redacted]

Property Name: *Operator UNI Link Aggregation Service Attribute*
Reference: MEF 26.2 sec. 14.5
JSON Name: *linkAggregation* - **Type:** *enum*

Description: Service Attribute representing Link Aggregation types. **NONE**-No Link Aggregation is used. **2_LINK_ACTIVE_STANDBY**-The SP network uses Link Aggregation as in either Clause 5.6.1 of IEEE Std 802.1AX-2008 [2] or Clause 6.7.1 of IEEE Std 802.1AX-2014 with one Link Aggregation Group (LAG) across the links supporting the UNI such that all Service Frames are carried on only one of the two links when both links are operational. **ALL_ACTIVE**-The SP Network uses Link Aggregation as specified in Clause 5.3 of IEEE Std 802.1AX-2014, including the use of the version 2 LACPDUs as specified in Clause 5.3.1h of IEEE Std 802.1AZ-2014, with one Link Aggregation Group (LAG) across the links. **OTHER**-The Operator/Subscriber/Service Provider agree on another other resiliency mechanism.

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

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

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

Property Name: *Operator UNI Port Conversation ID to Aggregation Link Map Service Attribute*
Reference: MEF 26.2 sec. 14.6 and 802.1AX-2014 sec. 6.6.2.1
JSON Name: *aggregationLinkMap* - **Type:** *List of object*

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

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

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

Validation Notes: Not required in inventory if OperatorUni.linkaggregration≠ALL_ACTIVE.

Property Name: *Operator UNI Maximum Service Frame Size Service Attribute*
Reference: MEF 26.2 sec. 14.8
JSON Name: *maximumServiceFrameSize* - **Type:** *integer*

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

Allowed values: Minimum: 1522

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

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

Property Name: *Operator UNI Default CE-VLAN ID Service Attribute*
Reference: MEF 26.2 sec. 14.9
JSON Name: *defaultCeVlanId* - **Type:** *integer*

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

Allowed values: Minimum: 1 Maximum: 4094

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

Property Name: *Operator UNI Maximum Number of OVC End Points Service Attribute*
Reference: MEF 26.2 sec. 14.10
JSON Name: *maximumNumberOfEndPoints* - **Type:** *integer*

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

Allowed values: Minimum: 1 Maximum: 4094

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

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

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

Allowed values: Minimum: 1 Maximum: 4094

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

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

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

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

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

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

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

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

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

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

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

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

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

Allowed values: Class: EnabledDisabled

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

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

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

Allowed values: Class: EnabledDisabled

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

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

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

Allowed values: Class: EnabledDisabled

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

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

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

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

Allowed values: Class: EnabledDisabled

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

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

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

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

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

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

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

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

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

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Required **Inventory:** Required

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

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

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

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

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

Validation Notes: If **AccessElineOvc.l2cpAddressSet** = *CTB*, this **must not** have a Destination Address that is in Table 6 but not in the CTB subset of Table 6 in MEF 45.1.

12.3 ENNI Attributes

12.3.1 ENNI Service Attributes

Carrier Ethernet ENNI SPSO
Schema File Name: carrierEthernetEnniSpso
Includes classes:
EnniSpso (ENNI SP/SO Attributes)

[redacted]

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

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

[redacted]

Property Name: *ENNI Service*
JSON Name: *enniService* - **Type:** *object*

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

Carrier Ethernet ENNI Service Attributes
Schema File Name: carrierEthernetEnniServiceAttributes
Includes classes:
EnniService (ENNI Service Attributes)

[redacted]

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

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

[redacted]

Property Name: *S-VLAN ID Control Service Attribute*
Reference: MEF 26.2 §13.2
JSON Name: *sVlanIdControl* - **Type:** *enum*

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

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Required

[redacted]

Property Name: *Maximum Number of OVCs Service Attribute*
Reference: MEF 26.2 §13.3
JSON Name: *maximumNumberOfOvcs* - **Type:** *integer*

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

Allowed values: Minimum: 1

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Required

Property Name: *Maximum Number of OVC End Points per OVC Service Attribute*
Reference: MEF 26.2 §13.4
JSON Name: *maximumNumberOfOvcEndPointsPerOvc* - **Type:** *integer*

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

Allowed values: Minimum: 1

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Required

Property Name: *ENNI Token Share Service Attribute*
Reference: MEF 26.2 §13.5
JSON Name: *tokenShare* - **Type:** *object*

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

Allowed values: Class: EnabledDisabled

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Required

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

Property Name: *ENNI Envelopes Service Attribute*
Reference: MEF 26.2 §13.6
JSON Name: *envelopes* - **Type:** *List of object*

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

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Required

12.3.2 ENNI Common Attributes

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

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

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

Property Name: *ENNI Peering Identifier Common Attribute*
Reference: MEF 26.2 §9.1
JSON Name: *peeringIdentifier* - **Type:** *string*

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

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

Property Name: *ENNI List of Physical Links*
Reference: MEF 26.2 §9.2 and 9.3
JSON Name: *listOfPhysicalLinks* - **Type:** *List of object*

Description: This Service Attribute combines several of the MEF 26.2 Objects into a single n-tuple that is consistent with the way this information is described in MEF 10.4. The value of the ENNI List of Physical Links Service Attribute is a list of 5-tuples of the form ⟨⟨id,pl,fs,pt,cn⟩, with one list item for each physical link. The value of **id** is an identifier for the physical link. The value of **pl** specifies a physical layer. **fs** indicates if synchronous Ethernet is used on the physical link corresponding to the 5-tuple and has the value either Enabled or Disabled. The value of **pt** indicates if the Precision Time Protocol is used on the physical link corresponding to the 5-tuple and has the value either Enabled or Disabled. The value of **cn** specifies the type of connector used at the ENNI.

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

Property Name: *ENNI Frame Format Common Attribute*
Reference: MEF 26.2 §9.3
JSON Name: *serviceFrameFormat* - **Type:** *enum*

Description: The ENNI Frame Format Common Attribute has the value Ethernet MAC Frame conforming to Clause 3 of IEEE 802.3-2012.

Allowed values: Enum: "802.3-2012"

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

Property Name: *ENNI Link Aggregation Common Attribute*
Reference: MEF 26.2 §9.5
JSON Name: *linkAggregation* - **Type:** *enum*

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

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

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

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

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

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

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

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

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

Allowed values: Class: EnabledDisabled

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

[redacted]

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

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

Allowed values: Class: EnabledDisabled

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

[redacted]

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

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

Allowed values: Class: EnabledDisabled

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

12.3.3 Operator Multilateral Attributes

Operator Multilateral Attributes
Schema File Name: operatorMultilateralAttributes
Includes classes:
OperatorMultilateral (Operator Multilateral Attributes)

[redacted]

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

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

[redacted]

Property Name: *ENNI L2CP Peering Multilateral Attribute*
Reference: MEF 26.2 §10.1 and MEF 45 §8.2
JSON Name: *l2cpPeering* - **Type:** *List of object*

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

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

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

[redacted]

Property Name: *ENNI Tagged L2CP Frame Processing Multilateral Attribute*
Reference: MEF 26.2 §10.2 and MEF 45 §8.3
JSON Name: *taggedL2cpFrameProcessing* - **Type:** *enum*

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

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

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

[redacted]

Property Name: *ENNI Maximum Frame Size Multilateral Attribute*
Reference: MEF 26.2 §10.3
JSON Name: *maximumFrameSizeMultilateral* - **Type:** *integer*

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

Allowed values: Minimum: 1526

Usage: **POQ:** Not Included **Quote:** Not Included **Order:** Not Included **Inventory:** Optional

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

13 Common Classes and Types

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

13.1 Bandwidth Profiles

Carrier Ethernet Bandwidth Profile
Schema File Name: Carrier_Ethernet_Bandwidth_Profile
Includes classes:
BwpFlow (Bandwidth Profile Flow)
Envelope (Envelope)
BandwidthProfilePerClassOfServiceName (COS Bandwidth Profile)

[redacted]

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.

[redacted]

Element Name: *Committed Information Rate*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *cir* - **Type:** *object*

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

Allowed values: Class: InformationRate

Usage: **Common:** Required

[redacted]

Element Name: *The Maximum Committed Information Rate*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *cirMax* - **Type:** *object*

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

Allowed values: Class: InformationRate

Usage: **Common:** Required

[redacted]

Element Name: *Committed Burst Size*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *cbs* - **Type:** *object*

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

Allowed values: Class: DataSize

Usage: Common: Required

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

Element Name: *Excess Information Rate*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *eir* - **Type:** *object*

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

Allowed values: Class: InformationRate

Usage: Common: Required

Element Name: *Maximum Excess Information Rate*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *eirMax* - **Type:** *object*

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

Allowed values: Class: InformationRate

Usage: Common: Required

Element Name: *Excess Burst Size*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *ebs* - **Type:** *object*

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

Allowed values: Class: DataSize

Usage: Common: Required

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

Element Name: *Coupling Flag*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *couplingFlag* - **Type:** *boolean*

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

Allowed values:

Usage: Common: Required

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

Element Name: *Color Mode*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *colorMode* - **Type:** *enum*

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

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

Usage: **Common:** Required

Element Name: *Envelope Name*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *envelopeId* - **Type:** *string*

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

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

Usage: **Common:** Required

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

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

Element Name: *Envelope Rank*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *envelopeRank* - **Type:** *integer*

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

Allowed values: Minimum: 1

Usage: **Common:** Required

Element Name: *Token Request Offset Field*
Reference: MEF 26.2 §17.1.2 and MEF 10.4 §12.1.2
JSON Name: *tokenRequestOffset* - **Type:** *integer*

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

Allowed values: [Any Integer]

Usage: **Common:** Required

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

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

Element Name: *Envelope Name*
Reference: MEF 10.4 §9.12 and MEF 26.2 §14.19
JSON Name: *envelopeID* - **Type:** *string*

Description: The attribute is a string that identifies the Envelope

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

Usage: **Common:** Required

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

Element Name: *Envelope CF0*
Reference: MEF 10.4 §9.12 and MEF 26.2 §14.19
JSON Name: *couplingFlagForIndexZero* - **Type:** *boolean*

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

Allowed values:

Usage: **Common:** Required

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

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

Element Name: *CoS or EEC Name*
Reference: MEF 26.2 §16.12 and 16.13
JSON Name: *classOfServiceName* - **Type:** *string*

Description: For Ingress, this is the Name of the Class of Service that this bandwidth profile flow is applied to. This should be an item from ovc.listOfClassOfServiceNames. For Egress, this is the Egress Equivalence Class (EEC) Name.

Allowed values: [Any String]

Usage: **Common:** Required

Element Name: *CoS Bandwidth Profile*
JSON Name: *bwpFlow* - **Type:** *object*

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

Allowed values: Class: BwpFlow

Usage: **Common:** Required

13.2 Class of Service

Carrier Ethernet Class of Service

Schema File Name: Carrier_Ethernet_Class_of_Service

Includes classes:

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

[redacted]

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

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

[redacted]

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

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

Allowed values: "ENDPOINT"

Usage: **Common:** Required

[redacted]

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

Description: The single CoS at this EVC/OVC EP

Allowed values: [Any String]

Usage: **Common:** Required

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

[redacted]

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

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

Allowed values: Class: CosL2cp

Usage: **Common:** Required

[redacted]

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

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

[redacted]

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

Description: The Field that indicates CoS is C-Tag PCP.

Allowed values: "C_TAG_PCP"

Usage: **Common:** Required

Element Name: *CoS Map*
JSON Name: *map_M* - **Type:** *List of object*

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

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

Usage: **Common:** Required

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

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

Allowed values: Class: CosL2cp

Usage: **Common:** Required

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

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

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

Description: A value 0 to 7

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

Usage: **Common:** Required

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

Description: A CoS Name or DISCARD

Allowed values: [Any String]

Usage: **Common:** Required

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

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

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

Element Name: *CoS Map Type - STAG*
JSON Name: *mapType* - **Type:** *const*

Description: The Field that indicates CoS is S-Tag PCP.

Allowed values: "S_TAG_PCP"

Usage: **Common:** Required

Element Name: *CoS Map*
JSON Name: *map_M* - **Type:** *List of object*

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

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

Usage: **Common:** Required

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

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

Allowed values: Class: CosL2cp

Usage: **Common:** Required

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

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

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

Element Name: *ENNI PCP Value*
JSON Name: *pcpVal* - **Type:** *enum*

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

Usage: **Common:** Required

Element Name: *ENNI CoS Name*
JSON Name: *pcpCosName* - **Type:** *string*

Allowed values: [Any String]

Usage: **Common:** Required

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

Ingress CoS is specified as {F,M,P}. This is when F=DSCP. This maps groups of DSCP values (independently for ipV4 and ipV6) to a Class of Service. It also provides a Class of Service mapping for DSCP values that are not specified and for frames that do not contain IP Packets.

Element Name: *CoS Map Type - DSCP*
JSON Name: *mapType* - **Type:** *const*

Description: The Field that indicates CoS is DSCP.

Allowed values: "DSCP"

Usage: **Common:** Required

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

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

Allowed values: Class: CosDscpMap

Usage: **Common:** Required

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

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

Allowed values: Class: CosL2cp

Usage: **Common:** Required

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

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

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

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

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

Usage: **Common:** Required

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

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

Allowed values: [Any String]

Usage: **Common:** Required

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

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

Allowed values: [Any String]

Usage: **Common:** Required

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

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

Allowed values: [Any String]

Usage: **Common:** Required

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

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

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

Description: List of IPv4 DSCP values

Allowed values: Class: DscpValues

Usage: **Common:** Optional

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

Allowed values: Class: DscpValues

Usage: **Common:** Optional

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

Allowed values: [Any String]

Usage: **Common:** Required

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

A list of 6 bit values

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

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

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

Usage: **Common:** Optional

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

2-tuple {L2CP Ident, CoS Name}

Element Name: *L2CP Identifier*
Reference: MEF 45.1 §8.2
JSON Name: *l2cpIdentifier* - **Type:** *object*

Description: Protocol Identifier

Allowed values: Class: L2cpProtocol

Usage: **Common:** Required

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

Description: CoS Name String to associate with the L2CP

Allowed values: [Any String]

Usage: **Common:** Required

13.3 Color Identifier

Carrier Ethernet Color Identifier
Schema File Name: Carrier_Ethernet_Color_Identifier
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)

[redacted]

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

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

[redacted]

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

Allowed values: "DEI"

Usage: **Common:** Required

[redacted]

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.

[redacted]

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

Allowed values: "PCP"

Usage: **Common:** Required

[redacted]

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

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

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

Usage: **Common:** Required

[redacted]

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

[redacted]

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

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

Usage: **Common:** Required

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

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

Usage: **Common:** Required

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

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

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

Allowed values: "DSCP"

Usage: **Common:** Required

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

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

Usage: **Common:** Required

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

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

Description: This is a list of DSCP values

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

Usage: **Common:** Required

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

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

Usage: **Common:** Required

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

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

Usage: **Common:** Required

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

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

[redacted]

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

Allowed values: "ENDPOINT"

Usage: **Common:** Required

[redacted]

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

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

Usage: **Common:** Required

13.4 Egress Maps

Carrier Ethernet Egress Maps
Schema File Name: Carrier_Ethernet_Egress_Maps
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)

[redacted]

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. TThis 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.

[redacted]

Element Name: *Map Type - CN → PCP*
JSON Name: *mapType* - **Type:** *const*

Allowed values: "CN_PCP"

Usage: **Common:** Required

[redacted]

Element Name: *CNPCP Entries*
JSON Name: *cnPcpEntries* - **Type:** *List of object*

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

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

Usage: **Common:** Required

[redacted]

Class: CnPcp ***(CNPCP Entry)***

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

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

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

Allowed values: [Any String]

Usage: **Common:** Required

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

Allowed values: Class: PcpValueAndDiscard

Usage: **Common:** Required

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

The DEI Field in the Egress EI Frame is determined from the Class of Service and the Frame Color. This is a list that contains an entry for Class of Service Name that includes the CoS Name and the DEI value (or 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.

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

Allowed values: "CC_DEI"

Usage: **Common:** Required

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

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

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

Usage: **Common:** Required

Class: CcDei ***(CCDEI Entry)***

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

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

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

Allowed values: [Any String]

Usage: **Common:** Required

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

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

Allowed values: Class: DeiValueAndDiscard

Usage: **Common:** Required

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

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

Allowed values: Class: DeiValueAndDiscard

Usage: **Common:** Required

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

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

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

Allowed values: "CC_PCP"

Usage: **Common:** Required

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

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

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

Usage: **Common:** Required

Class: CcPcp ***(CCPCP Entry)***

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

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

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

Allowed values: [Any String]

Usage: **Common:** Required

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

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

Allowed values: Class: PcpValueAndDiscard

Usage: **Common:** Required

Element Name: *PCP For Yellow Frames*
JSON Name: *pcpYellow* - **Type:** *object*

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

Allowed values: Class: PcpValueAndDiscard

Usage: **Common:** Required

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

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

Element Name: *Map Type - CN→PCP & CC→DEI*
JSON Name: *mapType* - **Type:** *const*

Allowed values: "CN_PCP_CC_DEI"

Usage: **Common:** Required

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

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

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

Usage: **Common:** Required

Class: CnPcpCcDei ***(CNPCPCCDEI Entry)***

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

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

Allowed values: [Any String]

Usage: **Common:** Required

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

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

Allowed values: Class: PcpValueAndDiscard

Usage: **Common:** Required

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

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

Allowed values: Class: DeiValueAndDiscard

Usage: **Common:** Required

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

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

Allowed values: Class: DeiValueAndDiscard

Usage: **Common:** Required

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

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

Allowed values: Class: PcpValueAndDiscard

Usage: **Common:** Optional

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

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

Description: A DEI value to use for Green frames

Allowed values: Class: DeiValueAndDiscard

Usage: **Common:** Optional

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

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

Description: A DEI value to use for Yellow frames

Allowed values: Class: DeiValueAndDiscard

Usage: **Common:** Optional

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

693
13.5 End Point Maps

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

[redacted]

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

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

[redacted]

Element Name: *Map Form - E*
JSON Name: *mapType* - **Type:** *const*
Allowed values: "FORM_E"
Usage: **Common:** Required

[redacted]

Element Name: *End Point Map Form E*
JSON Name: *EndPointMapFormE* - **Type:** *List of integer*
Allowed values: List of [1...4094] Unique Minimum: 1 Maximum: 4094
Usage: **Order:** Required **Common:** Required

[redacted]

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

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

[redacted]

[redacted]

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

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

[redacted]

[redacted]

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

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

[redacted]

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

Allowed values: "FORM_U"

Usage: **Common:** Required

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

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

Usage: **Common:** Required

696 ## 13.6 External Interfaces

Carrier Ethernet External Interfaces
Schema File Name: Carrier_Ethernet_External_Interfaces
Includes classes:
UniPhysicalLink (UNI Physical Link)
EnniPhysicalLink (ENNI Physical Link)
CarrierEthernetPhysicalLink (Carrier Ethernet Physical Link)
UniSpecifcAttributes (UNI Specific)

[redacted]

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

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

[redacted]

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

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

Usage: **Common:** Optional

[redacted]

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.

[redacted]

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

Allowed values: Compose (allOf): Class: CarrierEthernetPhysicalLink

Usage: **Common:** Optional

[redacted]

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.*

[redacted]

Element Name: *Physical Link Identifier*
JSON Name: *id* - **Type:** *string*

Description: An identifier for the physical link.

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

Usage: **Common:** Required

[redacted]

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

Description: An Ethernet physical layer

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

Usage: **Common:** Required

Class: UniSpecifcAttributes ***(UNI Specific)***

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

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

Allowed values: Class: EnabledDisabled

Usage: **Common:** Required

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

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

Allowed values: Class: EnabledDisabled

Usage: **Common:** Required

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

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

Allowed values: Enum: "SC", "LC", "RJ45", "FC", "D4", "OTHER"

Usage: **Common:** Required

Element Name: *UNI Connector Gender*
JSON Name: *uniConnectorGender* - **Type:** *enum*

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

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

Usage: **Common:** Required

13.7 Layer 2 Control Protocols (L2CP)

Carrier Ethernet L2CP
Schema File Name: Carrier_Ethernet_L2CP
Includes classes:
L2cpPeering (L2CP Peering Identifier)
L2cpProtocol (L2CP Protocol Identifier)

[redacted]

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

[redacted]

Element Name: *Protocol ID*
JSON Name: *protocolID* - **Type:** *object*

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

Allowed values: Class: L2cpProtocol

Usage: **Common:** Required

[redacted]

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

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

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

Usage: **Common:** Required

[redacted]

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

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

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

Usage: **Common:** Optional

[redacted]

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.

[redacted]

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

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

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

Usage: **Common:** Required

[redacted]

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

Description: The EtherType value or LLC value

Allowed values: Minimum: 0

Usage: **Common:** Required

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

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

Allowed values: Minimum: 0

Usage: **Common:** Optional

13.8 Link Aggregation

Carrier Ethernet Link Aggregation
Schema File Name: Carrier_Ethernet_Link_Aggregation
Includes classes:
ConversationIdToAggregationLinkMap (Conversation To Aggregation Link Map)
ConversationIdRange (ConversationID Range)

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

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

Element Name: *List of Conversation ID Ranges*
Reference: 802.1AX-2014 §6.6.2.1
JSON Name: *conversationIDs* - **Type:** *List of object*

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

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

Usage: **Common:** Required

Element Name: *Aggregation Link List*
Reference: 802.1AX-2014 §6.6.2.1
JSON Name: *aggLinkList* - **Type:** *List of integer*

Description: An ordered list of Aggregation Link Numbers

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

Usage: **Common:** Required

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

Class: ConversationIdRange ***(ConversationID Range)***

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

Element Name: *Start Conversation ID*
JSON Name: *start* - **Type:** *integer*

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

Allowed values: Minimum: 0 Maximum: 4094

Usage: **Common:** Required

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

Element Name: *End Conversation ID*
JSON Name: *end* - **Type:** *integer*

Description: The final Conversation ID in the range

Allowed values: Minimum: 0 Maximum: 4094

Usage: **Common:** Optional

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

13.9 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)

[redacted]

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.

[redacted]

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

Description: This represents the date and time for the start of the SLS. It is the beginning of the first *timeInterval*. This is referred to as *ts* in MEF 10.4 and MEF 26.2.

Allowed values: Class: Time

Usage: **Common:** Required

[redacted]

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

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

Allowed values: Class: LongDuration

Usage: **Common:** Required

[redacted]

Property Name: *SLS CoS Entries*
Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13
JSON Name: *slsCosNameEntry* - **Type:** *List of object*

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

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

Usage: **Common:** Required

[redacted]

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.

[redacted]

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

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

Allowed values: String: Min Length: 1

Usage: **Common:** Required

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

[redacted]

Property Name: *Delta T*
Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13
JSON Name: *deltaT* - **Type:** *integer*

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

Allowed values: [Any Integer]

Usage: **Common:** Required

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

[redacted]

Property Name: *Frame Loss Threshold*
Reference: MEF 10.4 sec. 8.8 and MEF 26.2 sec. 12.13
JSON Name: *thresholdC* - **Type:** *number*

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

Allowed values: Minimum: 0 Maximum: 1

Usage: **Common:** Required

[redacted]

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

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

Allowed values: Minimum: 1

Usage: **Common:** Required

[redacted]

Property Name: *One Way Frame Delay*
Reference: MEF 10.4 sec. 8.8.2 and MEF 26.2 sec. 12.13.2
JSON Name: *oneWayFrameDelayPmMetric* - **Type:** *List of object*

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

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

Usage: Common: Required

Property Name: *One Way Mean Frame Delay*
Reference: MEF 10.4 sec. 8.8.3 and MEF 26.2 sec. 12.13.3
JSON Name: *oneWayMeanFrameDelayPmMetric* - **Type:** *List of object*

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

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

Usage: Common: Required

Property Name: *One Way Frame Delay Range*
Reference: MEF 10.4 sec. 8.8.4 and MEF 26.2 sec. 12.13.4
JSON Name: *oneWayFrameDelayRangePmMetric* - **Type:** *List of object*

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

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

Usage: Common: Required

Property Name: *One Way Inter Frame Delay Variation*
Reference: MEF 10.4 sec. 8.8.5 and MEF 26.2 sec. 12.13.5
JSON Name: *oneWayInterFrameDelayVariationPmMetric* - **Type:** *List of object*

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

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

Usage: Common: Required

Property Name: *One Way Frame Loss Ratio*
Reference: MEF 10.4 sec. 8.8.6 and MEF 26.2 sec. 12.13.6
JSON Name: *oneWayFrameLossRatioPmMetric* - **Type:** *List of object*

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

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

Usage: Common: Required

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

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

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

Usage: **Common:** Required

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

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

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

Usage: **Common:** Required

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

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

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

Usage: **Common:** Required

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

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

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

Usage: **Common:** Required

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

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

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

Usage: **Common:** Required

[redacted]

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

One Way Frame Delay Performance Metric consisting of 2 parameters, a list of Ordered End Point Pairs (referred to as S in the MEF Standards) , a percentile (referred to as Pd in the MEF Standards), and the One Way Frame Delay Objective for this Class of Service Name and these two parameters. See MEF 10.4 section 8.8.2 for EVC and MEF 26.2 section 12.13.2 for OVC.

[redacted]

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

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

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

Usage: **Common:** Required

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

[redacted]

Property Name: *Frame Delay Percentile*
Reference: MEF 10.4 sec. 8.8.1.5 and MEF 26.2 sec. 12.13.2
JSON Name: *oneWayFdPercentile* - **Type:** *object*

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

Allowed values: Class: Percentage

Usage: **Common:** Required

Validation Notes: This must be greater than 0.

[redacted]

Property Name: *Frame Delay Objective*
Reference: MEF 10.4 sec. 8.8.1.5 and MEF 26.2 sec. 12.13.2
JSON Name: *oneWayFdObjective* - **Type:** *object*

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

Allowed values: Class: ShortDuration

Usage: **Common:** Required

[redacted]

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

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

[redacted]

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

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

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

Usage: **Common:** Required

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

Property Name: *Mean Frame Delay Objective*
JSON Name: *oneWayMfdObjective* - **Type:** *object*

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

Allowed values: Class: ShortDuration

Usage: **Common:** Required

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

One Way Frame Delay Range Performance Metric consisting of 2 parameters, a list of Ordered End Point Pairs (referred to as S in the MEF Standards) , a percentile (referred to as Pr in the MEF Standards), and the One Way Frame Delay Range Objective for this Class of Service Name and these two parameters. See MEF 10.4 section 8.8.4 for EVC and MEF 26.2 section 12.13.4 for OVC.

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

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

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

Usage: **Common:** Required

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

Property Name: *Frame Delay Range Percentile*
JSON Name: *oneWayFdrPecentile* - **Type:** *object*

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

Allowed values: Class: Percentage

Usage: **Common:** Required

Validation Notes: This must be greater than 0.

Property Name: *Frame Delay Range Objective*
JSON Name: *oneWayFdrObjective* - **Type:** *object*

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

Allowed values: Class: ShortDuration

Usage: **Common:** Required

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

One Way Frame Inter Frame Delay Variation Performance Metric consisting of 3 parameters, a list of Ordered End Point Pairs (referred to as S in the MEF Standards) , a time duration (referred to as Δt (the greek letter tau, not t) in the MEF Standards), a percentile (referred to as Pv in the MEF Standards), and the Inter

Frame Delay Variation Objective for this Class of Service Name and these three parameters. See MEF 10.4 section 8.8.5 for EVC and MEF 26.2 section 12.13.5 for OVC.

[redacted]

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

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

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

Usage: **Common:** Required

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

[redacted]

Property Name: *Time Duration*
JSON Name: *timeDuration* - **Type:** *object*

Description: A time duration.

Allowed values: Class: ShortDuration

Usage: **Common:** Required

[redacted]

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

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

Allowed values: Class: Percentage

Usage: **Common:** Required

Validation Notes: This must be greater than 0.

[redacted]

Property Name: *Interframe Delay Variation Objective*
JSON Name: *oneWayIfdvObjective* - **Type:** *object*

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

Allowed values: Class: ShortDuration

Usage: **Common:** Required

[redacted]

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

One Way Frame Loss Ratio Performance Metric consisting of 1 parameter, a list of Ordered End Point Pairs (referred to as S in the MEF Standards), and the One Way Frame Loss Ratio Objective for this Class of Service Name and these Ordered End Point Pairs. See MEF 10.4 section 8.8.6 for EVC and MEF 26.2 section 12.13.6 for OVC.

[redacted]

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

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

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

Usage: **Common:** Required

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

[redacted]

Property Name: *Frame Loss Ratio Objective*
JSON Name: *oneWayFlrObjective* - **Type:** *object*

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

Allowed values: Class: Percentage

Usage: **Common:** Required

[redacted]

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

One Way Availability Performance Metric consisting of 1 parameter, a list of Ordered End Point Pairs (referred to as S in the MEF Standards), and the One Way Availability Objective for this Class of Service Name and these Ordered End Point Pairs. See MEF 10.4 section 8.8.7 for EVC and MEF 26.2 section 12.13.7 for OVC.

[redacted]

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

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

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

Usage: **Common:** Required

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

[redacted]

Property Name: *Availability Objective*
JSON Name: *oneWayAvailabilityObjective* - **Type:** *object*

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

Allowed values: Class: Percentage

Usage: **Common:** Required

[redacted]

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

One Way High Loss Intervals Performance Metric consisting of 1 parameter, a list of Ordered End Point Pairs (referred to as S in the MEF Standards), and the One Way High Loss Intervals Objective for this Class of Service Name and these Ordered End Point Pairs. See MEF 10.4 section 8.8.8 for EVC and MEF 26.2 section 12.13.8 for OVC.

[redacted]

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

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

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

Usage: **Common:** Required

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

[redacted]

Property Name: *HLI Objective*
JSON Name: *oneWayHighLossIntervalsObjective* - **Type:** *integer*

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

Allowed values: Minimum: 0

Usage: **Common:** Required

[redacted]

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

One Way Frame Delay Range Performance Metric consisting of 2 parameters, a list of Ordered End Point Pairs (referred to as S in the MEF Standards) , a number (referred to as p in the MEF Standards), and the One Way Consecutive High Loss Intervals Objective for this Class of Service Name and these two parameters. See MEF 10.4 section 8.8.9 for EVC and MEF 26.2 section 12.13.9 for OVC.

[redacted]

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

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

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

Usage: **Common:** Required

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

[redacted]

Property Name: *CHLI Consecutive Number*
JSON Name: *consecutiveNumberP* - **Type:** *integer*

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

Allowed values: Minimum: 1

Usage: **Common:** Required

[redacted]

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

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

Allowed values: Minimum: 1

Usage: **Common:** Required

[redacted]

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

One Way Composite Performance Metric consisting of 7 parameters, a list of Ordered End Point Pairs (referred to as S in the MEF Standards) , a Composite Performance number (referred to as U in the MEF Standards), indicators about Composite Frame Loss, Composite Frame Delay, and Composite Frame Delay Variation (referred to as Wfl, Wfd, and Wfdv in the MEF Standards), Thresholds for Frame Delay and Delay Variation (referred to as DL and Jt in the MEF Standards) and the One Way Composite Objective for this Class of Service Name and these seven parameters. See MEF 10.4 section 8.8.10 for EVC (this Performance Metric is not defined for OVC services).

[redacted]

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

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

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

Usage: **Common:** Required

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

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

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

Allowed values: Minimum: 0 Maximum: 1

Usage: **Common:** Required

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

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

Allowed values: Minimum: 0 Maximum: 1

Usage: **Common:** Required

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

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

Allowed values: Minimum: 0 Maximum: 1

Usage: **Common:** Required

Property Name: *Composite Frame Delay Variation*
JSON Name: *compositeFrameDelayVariation* - **Type:** *integer*

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

Allowed values: Minimum: 0 Maximum: 1

Usage: **Common:** Required

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

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

Allowed values: Class: ShortDuration

Usage: **Common:** Required

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

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

Allowed values: Class: ShortDuration

Usage: **Common:** Required

Property Name: *Composite Objective*
JSON Name: *cpmObjective* - **Type:** *object*

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

Allowed values: Class: Percentage

Usage: **Common:** Required

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

The One Way Group Availability Performance Metric has 2 parameters. The first parameter is a list of sets of Ordered End Point Pairs (referred to as G in the MEF Standards). (Note that this is different than the other Performance Metrics which have a list of Ordered End Point Pairs). The second parameter is the number of sets that must be available (referred to as K in the MEF Standards). The third element is the Performance Objective for the One Way Group Availability. See MEF 10.4 section 8.8.11 for EVC and MEF 26.2 section 12.13.10 for OVC.

Property Name: *GA Ordered Pairs*
JSON Name: *setOfOrderedPairs* - **Type:** *List of object*

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

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

Usage: **Common:** Required

Property Name: *Minimum Number of Sets Available*
JSON Name: *minimumNumberOfSetsAvailableK* - **Type:** *integer*

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

Allowed values: Minimum: 1

Usage: **Common:** Required

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

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

Allowed values: Class: Percentage

Usage: **Common:** Required

Class: OrderedPair ***(Ordered Pair)***

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

Property Name: *From EVC End Point*
JSON Name: *fromCarrierEthernetServiceEndPoint* - **Type:** *string*

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

Allowed values: [Any String]

Usage: **Common:** Required

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

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

Allowed values: [Any String]

Usage: **Common:** Required

[redacted]

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

The Group Availability is specified for sets of Ordered End Point Pairs, e.g., G1=⟨⟨A,B⟩, ⟨A,C⟩⟩ and G2=⟨⟨A,D⟩, ⟨A,E⟩⟩. This class defines one set of Ordered End Point Pairs.

[redacted]

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

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

Usage: **Common:** Required

13.10 Utility Classes and Types

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

[redacted]

Class: MepLevelAndDirection ***(MEP Specification)***

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

[redacted]

Element Name: *MEG Level*
JSON Name: *level* - **Type:** *integer*

Description: The MEG level

Allowed values: Minimum: 0 Maximum: 7

Usage: **Common:** Required

[redacted]

Element Name: *MEP Direction*
JSON Name: *mepDirection* - **Type:** *enum*

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

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

Usage: **Common:** Required

[redacted]

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.

[redacted]

Element Name: *Number of Addresses*
JSON Name: *limit* - **Type:** *integer*

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

Allowed values: Minimum: 1

Usage: **Common:** Required

Element Name: *Time Interval*
JSON Name: *interval* - **Type:** *object*

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

Allowed values: Class: ShortDuration

Usage: **Common:** Required

Class: AggLinkDepth ***(Link Depth)***

This is a pair of indicating that a given VLAN ID maps to a given number of links in the Port Conversation ID to Aggregation Link Map.

Element Name: *vlanId*
JSON Name: *vlanID* - **Type:** *integer*

Description: The VLAN ID

Allowed values: Minimum: 1 Maximum: 4094

Usage: **Common:** Required

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

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

Allowed values: Minimum: 1

Usage: **Common:** Required

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

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

Element Name: *Unicast Frames*
JSON Name: *unicast* - **Type:** *object*

Allowed values: Class: FrameDispEnum

Usage: **Common:** Required

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

Allowed values: Class: FrameDispEnum

Usage: **Common:** Required

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

Allowed values: Class: FrameDispEnum

Usage: **Common:** Required

[redacted]

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

[redacted]

Element Name: *frameDispEnum* **Type:** *enum*

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

Usage: **Common:** Optional

[redacted]

Type: EnabledDisabled ***(Enabled Disabled)***

Enumeration to indicate Enabled/Disabled state of an attribute

[redacted]

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

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

Usage: **Common:** Optional

[redacted]

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

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

[redacted]

Element Name: *IR Value*
JSON Name: *irValue* - **Type:** *number*

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

Allowed values: Minimum: 0

Usage: **Common:** Required

[redacted]

Element Name: *IR Units*
JSON Name: *irUnits* - **Type:** *enum*

Description: The unit of measure for the Information Rate. For example if the Information Rate is 70 kbps this element is kbps.

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

Usage: **Common:** Required

[redacted]

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.

[redacted]

Element Name: *Data Size Value*
JSON Name: *dataSizeValue* - **Type:** *integer*

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

Allowed values: [Any Integer]

Usage: **Common:** Required

Element Name: *Data Size Units*
JSON Name: *dataSizeUnits* - **Type:** *enum*

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

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

Usage: **Common:** Required

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

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

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

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

Allowed values: [Any Integer]

Usage: **Common:** Required

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

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

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

Usage: **Common:** Required

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

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

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

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

Allowed values: [Any Integer]

Usage: **Common:** Required

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

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

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

Usage: **Common:** Required

Class: Time *(Time and Date)*

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

Allowed values: Minimum: 1900

Usage: **Common:** Required

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

Allowed values: Minimum: 1 Maximum: 12

Usage: **Common:** Required

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

Allowed values: Minimum: 1 Maximum: 31

Usage: **Common:** Required

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

Allowed values: Minimum: 0 Maximum: 23

Usage: **Common:** Required

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

Allowed values: Minimum: 0 Maximum: 59

Usage: **Common:** Required

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

Allowed values: Minimum: 0 Maximum: 59

Usage: **Common:** Required

Type: Percentage *(Percentage)*

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

Element Name: *Percentage* **Type:** *number*

Allowed values: Minimum: 0 Maximum: 100

Usage: **Common:** Optional

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

[redacted]

Element Name: *PCP Value and Discard* **Type:** *enum*
Allowed values: Enum: "0", "1", "2", "3", "4", "5", "6", "7", "DISCARD"
Usage: **Common:** Optional

[redacted]

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

[redacted]

Element Name: *DEI Value and Discard* **Type:** *enum*
Allowed values: Enum: "0", "1", "DISCARD"
Usage: **Common:** Optional

14 Service Attributes Not Included

There are 16 Services Attributes from MEF 26.2 that are not included in the API. The two primary reasons for not including them is:

- They are part of the product agnostic envelope (i.e., many of the Identifiers), and
- The Service Attribute has a single constant value for Access E-Line. 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
OVC Service Attribute	OVC Identifier	Identifiers for Order Items are not needed since they are in the product-agnostic envelope.
OVC Service Attribute	OVC Type	Not needed for Access E-Line. Only value allowed for Access E-Line is point-to-point
OVC Service Attribute	OVC End Point List	This is a list of OVC End Point IDs. This is not needed since the OVC End Points are included in the OVC definition.
OVC Service Attribute	Maximum Number of UNI OVC End Points	Not needed since it must be "1" for Access E-Line
OVC Service Attribute	Maximum Number of ENNI OVC End Points	Not needed since it must be "1" for Access E-Line
OVC Service Attribute	OVC S-VLAN PCP Preservation	Not needed because there is only one ENNI in the service.
OVC Service Attribute	OVC S-VLAN DEI Preservation	Not needed because there is only one ENNI in the service.
OVC End Point Service Attributes	OVC End Point Type	This indicates whether the End Point is at a UNI or an ENNI. For Access E-Line this information is embedded in the OVC Property containing the End Point (uniEP and enniEP)
OVC End Point Service Attributes	OVC End Point External Interface Identifier	For Access E-Line there is only one UNI and one ENNI and they are identified in the API (as related products), so they don’t need to be explicitly called out in the product configuration. For most other Ethernet Services this attribute is needed.
OVC End Point Service Attributes	OVC End Point Role	This is always ROOT for Access E-Line at both the UNI and the ENNI, so not needed.
OVC End Point Service Attributes	OVC End Point Ingress Bandwidth Profile per OVC End Point Service Attribute	Per MEF 51.1 this MUST be Disabled at both the UNI and ENNI for all Operator Services.
Operator UNI Service Attributes	Operator UNI Identifier	Identifiers for Order Items are not needed since they are in the product-agnostic envelope.
Operator UNI Service Attributes	Operator UNI Service Frame Format	Not needed, only a single value allowed: "802.3-2012"
Operator UNI Service Attributes	Operator UNI E-LMI	Not included since MEF 10.4 no longer includes and E-LMI attribute.

Group	Service Attribute	Discussion
Operator UNI Service Attributes	Operator UNI L2CP Address Set	Per MEF 45.1: If an OVC has an OVC End Point at a UNI, then the value of the Operator UNI L2CP Address Set Service Attribute MUST be the same as the OVC L2CP Address Set Service Attribute.
ENNI Service Attributes	Operator ENNI Identifier	Identifiers for Order Items are not needed since they are in the product-agnostic envelope.
ENNI Common Attributes	ENNI Frame Format Common Attribute	Not Needed: Each ENNI-N MUST comply with the Ethernet Media Access Control standards specified in Clauses 3 and 4 of IEEE Std 802.3-2012 [4].

717 **Table 6 – Service Attributes Not Included in Schemas**

718

15 References

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

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

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

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

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

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

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

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

[9] MEF 51.1, *Operator Ethernet Service Definitions*, December 2018

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

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

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

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

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