MEF Standard
MEF 106

LSO Sonata Access E-Line Product Schemas and Developer Guide

February 2023

Disclaimer

© MEF Forum 2023. All Rights Reserved.

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

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

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

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

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

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

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

Table of Contents

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.

- Amartus
- Cisco Systems
- Colt Technology Services
- Lumen Technologies
- Proximus

2 Abstract

The MEF Standard consisting of this schema guide and its associated software artifacts (JSON Schemas) defines and describes the product-specific information used in LSO 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 focused on the JSON Schemas 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 [9]. The service, itself, is specified in MEF 51.1 [11] 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.

This Standard normatively incorporates the following files by reference as if they were part of this document, from GitHub repository https://github.com/MEF-GIT/MEF-LSO-Sonata-SDK at commit ID: 41592db42d1ff54d0f33a393fa506217947a5567.

Schemas for Operator Ethernet Classes (Access-Eline and Operator Ethernet only)
- productSchema/carrierEthernet/operatorEthernet/
  - accessEline/accessElineOvc.yaml
  - carrierEthernetEnniSpSo/inventory/carrierEthernetEnniSpSo.yaml
  - carrierEthernetOperatorUni/carrierEthernetOperatorUni.yaml
  - ovcProductComponents/accessElineOvcEpCommon.yaml
  - ovcProductComponents/carrierEthernetEnniServiceAttributes.yaml
  - ovcProductComponents/carrierEthernetEnniCommon.yaml
  - ovcProductComponents/operatorMultilateralAttributes.yaml
  - operatorEthernetCommon/ovcEndPointMaps.yaml
  - operatorEthernetCommon/ovcEgressMaps.yaml
  - operatorEthernetCommon/ovcExternalInterfaces.yaml
  - operatorEthernetCommon/ovcClassOfService.yaml

Also included in the GitHub repository is a Postman file that contains informative examples illustrating use of the Access E-Line payloads. This file is not part of this standard but is referred to in Appendix A.
- documentation/productSchema/carrierEthernet/operatorEthernet/MEF 106 - Appendix A.postman_collection.json

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 [11]
Business Applications	The Service Provider functionality supporting Business Management Layer functionality (e.g., product catalog, order management, billing, relationship management, etc.)	MEF 55.1 [12]
BUS	See *Business Applications*	MEF 55.1 [12]
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 57.2* [13]
Data Model	A representation of concepts of interest to an environment in a form that is dependent on data repository, data definition language, query language, implementation language, and/or protocol (typically, but not necessarily, all five).	MEF 78.1 [15]
E-Access Service	An Operator Ethernet Service that associates at least one OVC End Point that is at a UNI and at least one OVC End Point that is at an ENNI.	MEF 51.1 [11]
External Network Network Interface	A reference point representing the boundary between two Operator Carrier Ethernet Networks that are operated as separate administrative domains.	MEF 26.2 [9]
ENNI	External Network Network Interface	MEF 26.2 [9]
ENNI Frame	An Ethernet Frame that is exchanged across an ENNI between two Operators.	MEF 26.2* [9]
Ethernet Service Provider	An organization that provides to a Subscriber a connectivity service that carries Ethernet Frames irrespective of the underlying technology and that is specified using Service Attributes as defined in a MEF Standard.	MEF 10.4* [8]
Ethernet Subscriber	The end-user of an Ethernet Service	MEF 10.4 [8]
Ethernet UNI	The demarcation point between the responsibility of the Ethernet Service Provider and the Ethernet Service Subscriber.	MEF 10.4 [8]
Ethernet Virtual Connection	An association of EVC End Points	MEF 10.4 [8]
EVC	Ethernet Virtual Connection	MEF 10.4 [8]
EVC End Point	A construct at a (Ethernet) UNI that selects a subset of the Service Frames that pass over the UNI.	MEF 10.4 [8]
External Interface	Either a UNI or an ENNI.	MEF 4 [5]
External Interface Frame	Either a Service Frame or an ENNI Frame	MEF 26.2 [9]
Information Model	A representation of concepts of interest to an environment in a form that is independent of data repository, data definition language, query language, implementation language, and protocol	MEF 78.1 [15]

Term	Definition	Reference
Inventory	Product Inventory	MEF 81 [17]
Milestone	An event that occurs during the fulfillment process that indicates a significant step in the process has been completed	This document
Operator	The administrative entity of a Carrier Ethernet Network	MEF 26.2 [9]
Operator UNI Service Attributes	The Service Attributes used to describe an agreement between a Service Provider and an Operator for a UNI associated with an OVC.	MEF 26.2 [9]
Order	One or more Product Order Items formulated into a fulfilment request made by a Buyer to a Seller.	This document (derived from MEF 57.2 [13]
Operator Ethernet Service	An Ethernet Service that is provided by an Operator to another Operator or to an Ethernet Service Provider.	MEF 51.1 [11]
Operator Virtual Connection	An association of OVC End Points	MEF 26.2 [9]
OVC	Operator Virtual Connection	MEF 26.2 [9]
OVC End Point	A logical entity at a given External Interface that is associated with a distinct set of frames passing over that External Interface.	MEF 26.2 [9]
Product	One or more goods or services that is or may be sold to a Buyer by a Seller.	MEF 79 [16]
Product Catalog	A Product Catalog lists and describes the Product Offerings made available by a Seller to potential Buyers.	This document
Product Offering	The commercial and technical details of a Product sold by a Seller. A Product Offering defines all the commercial terms and, through association with a particular Product Specification, defines all of the technical attributes and behaviors of the Product. A Product Offering may constrain the allowable set of configurable technical attributes and/or behaviors specified in the associated Product Specification.	MEF 79 [16]
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 [17]
Product-Specific Attributes	The Service Attributes and their parameters for a Product that would be delivered to fulfill a Business Function.	This document
Product Specification	A Product Specification defines the template or detailed description from which Product Offerings can be defined.	MEF 79 [16]
POQ	Product Offering Qualification	MEF 79 [16]
Product Offering Qualification	One or more Product Offering Qualification Items formulated into a requirement made by a Buyer to a Seller.	MEF 79 [16]
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 [16]
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	This document

Term	Definition	Reference
	objective is to determine the charges and timeframe for the Seller to deliver this item as described by the Buyer.	
Seller	For the purposes of this document, a Seller is the Operator who is providing the product to the Buyer.	MEF 57.2* [13]
Service Attribute	Specific information that is agreed between the provider and the user of the service, as described in a MEF Standard, that describes some aspect of the service behavior or capability.	MEF 61.1 [14]
Service Frame	An Ethernet Frame that is exchanged across a UNI between a Subscriber and a Service Provider.	MEF 10.4 [8]
Service Provider	In the context of this document, a Service Provider is an Ethernet Service Provider. In this document we use Service Provider to include Super Operator as specified in MEF 26.2 (also referred to as SP/SO).	This Document
Sonata	The Management Interface Reference Point supporting the management and operations interactions (e.g., ordering, billing, trouble management, etc.) between two network providers (e.g., Service Provider Domain and Partner Domain).	MEF 55.1 [12]
Subscriber	In the context of this document, a Subscriber is an Ethernet Subscriber.	This Document
UNI	An Ethernet User Network Interface	MEF 10.4 [8]

Table 1 – Terminology and Abbreviations

4 Compliance Levels

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

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

5 Numerical Prefix Conventions

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

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

Table 2 – Numerical Prefix Conventions

6 Introduction

LSO 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 [9]. The Service definition for Access E-Line which is, in effect, a set of constraints on the values of the Service Attributes, is provided in MEF 51.1 [11].

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 one OVC End Point 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 12.3.3).

The document contains the following sections:
- An overview of LSO Sonata (section 7)
- An overview of the Access E-Line Service (section 8)
- Data Model Design Principles (section 9)
- Order Milestones (section 10)
- An abbreviated Information Model for Access E-Line and explanation of the organization of the Service Attributes in MEF 26.2 (section 11)
- Organization of the data model for Access E-Line (section 12)
- The relationship between the entities in the service (section 13)

These sections are followed by three sections that contain tables that describe the details of the data model. 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 14 contains the details of the Service Attributes for OVC, OVC End Point, UNI, and ENNI
- Section 15 contains all of the common classes and types referenced by the Service Attributes

- Section 16 lists the Service Attributes that are not included in the data models

7 Overview of LSO Sonata

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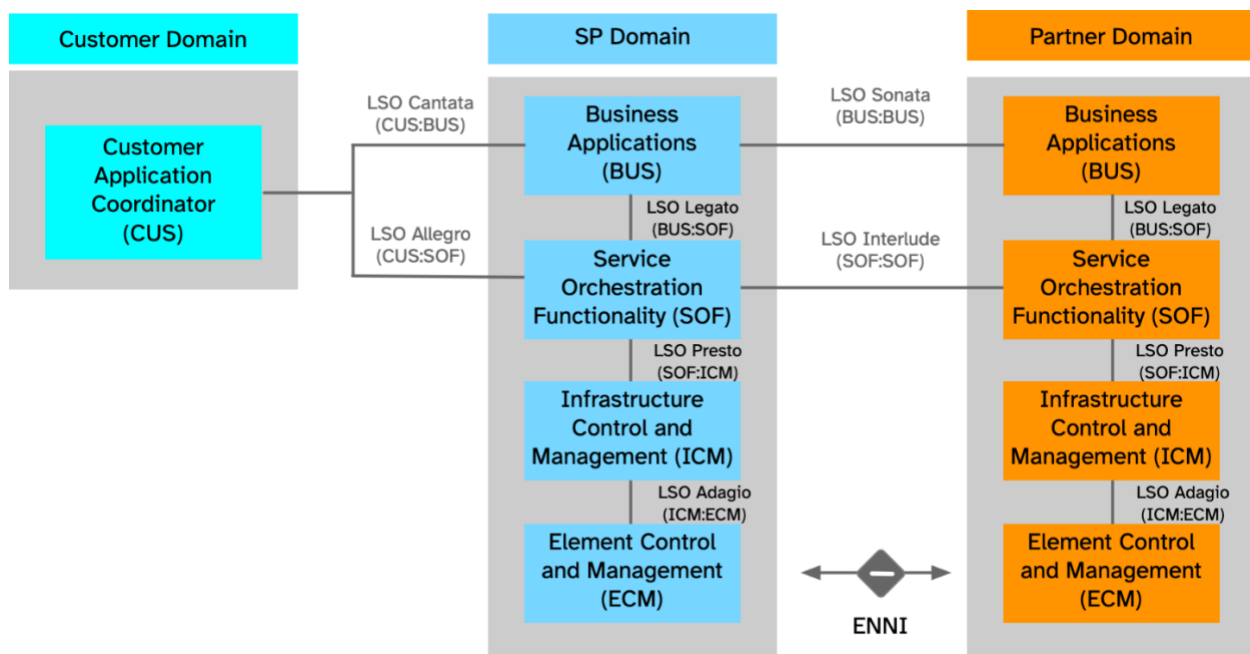
Figure 1 – LSO Sonata Reference Diagram

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.
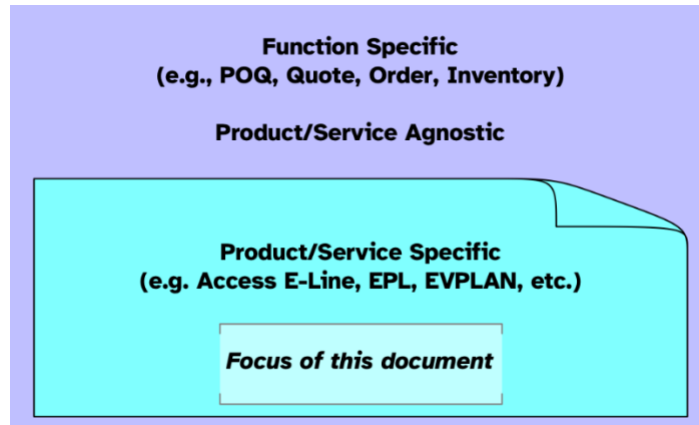
Figure 2 – LSO Sonata API Structure

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

8   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 one of the primary uses of OVCs in MEF services.

8.1   EVCs

A Subscriber Ethernet Service is built on an Ethernet Virtual Connection (EVC) which is an association of EVC End Points. An EVC End Point is a construct located at a UNI that represents the logical attachment of an EVC to that UNI and selects a subset of Service Frames that pass over the UNI. EVCs and their Service Attributes are described in MEF 10.4 [8]. There are three types of EVCs: point-to-point, multipoint-to-multipoint, and rooted-multipoint. MEF 6.3 [6] 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.

8.2   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 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 [9]. 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 [11] 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.

8.3   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.
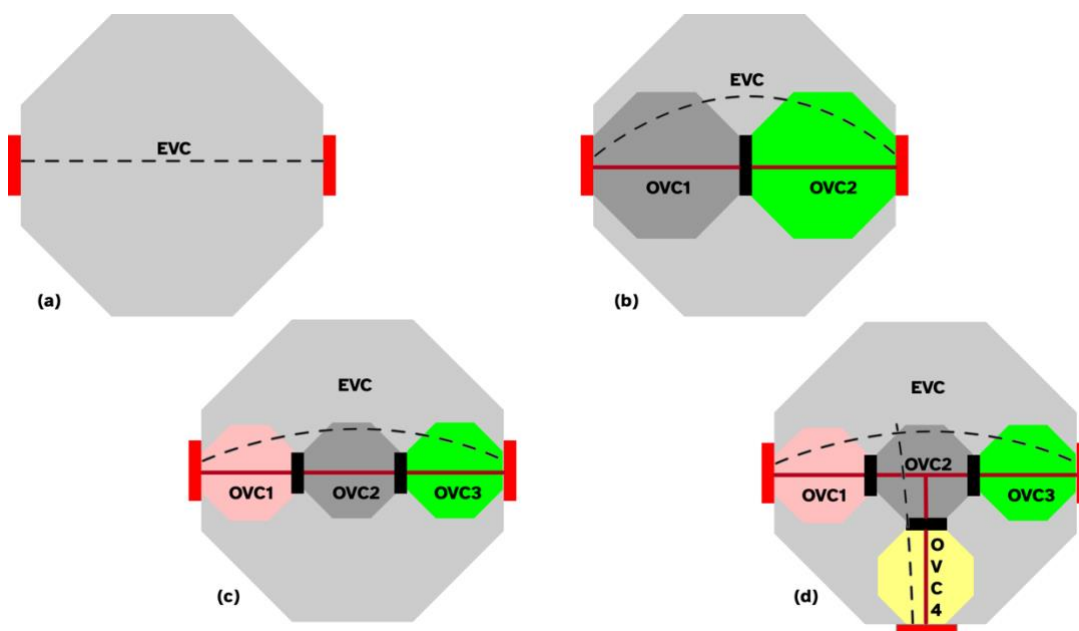
Figure 3 – OVCs and Access E-Line Services

In the diagram, in cases (a), (b), and (c), Ethernet Service Provider Gray is delivering an E-Line EVC (Ethernet Private Line or Ethernet Virtual Private Line) 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.

8.4   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.
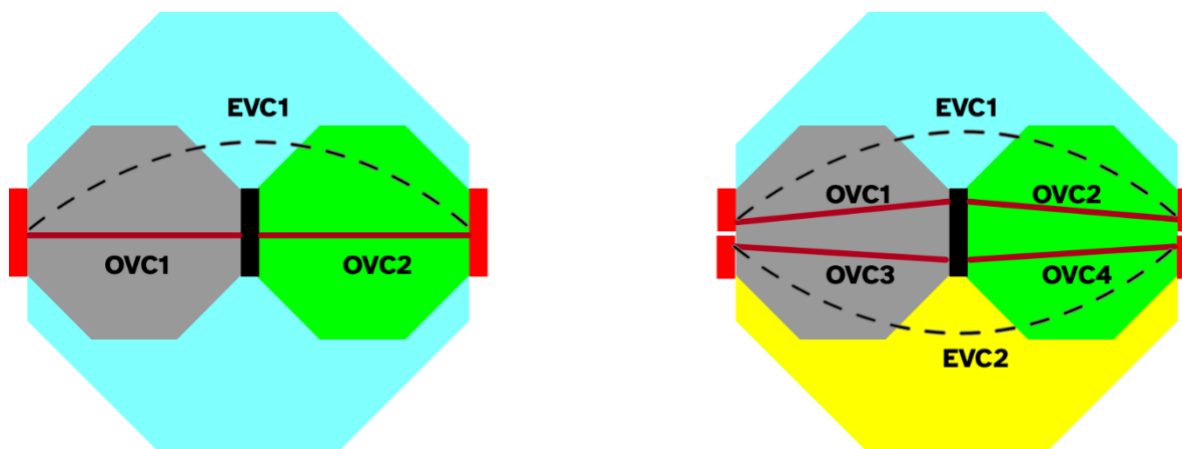
Figure 4 – Service Providers and Network Operators

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 12.1.

9 Data Model Design Principles and Assumptions

A Service Attribute for a Product can have a value that is a simple datatype such as an integer or string (or list of simple datatypes) or a value that is an object with multiple properties such as a Bandwidth Profile or a composition of objects such as an End Point Map. Within this document, each simple value (integer, string, boolean, etc.) is referred to as a Product-Specific Attribute. A Product-Specific Attribute could be a Service Attribute (in the case where the Service Attribute itself has a simple type) or it could be a parameter within a Service Attribute (if the Service Attribute is a structured object or a composition of such objects). There are no Product-Specific Attributes that are tagged as "Required" in the Access E-Line data model and, as such, each must be assigned by each Seller into one of three classifications as defined below.

Note: The one exception to the previous paragraph are the two OVC End Points defined for the Access E-Line OVC. These must be included in the Access E-Line data model since the OVC is incomplete without them and are therefore tagged as "Required".

The design for the Access E-Line data model is based on a number of assumptions:
- None of the Product-Specific Attributes included in the schemas are coded as "Required".
- The Buyer and Seller agree to assign each Product-Specific Attribute included in the schemas into one of three classifications. The classification for each Product-Specific Attribute may be different across Business Function, Product Action, and Product Offering.
    - o Mandatory - attributes that must be provided by the Buyer in a POQ/Quote/Order request or must be returned by the Seller for an Inventory request as specified in section 9.1.
    - o Optional - attributes that may be provided by the Buyer in a POQ/Quote/Order request and may be returned by the Seller for an Inventory request as specified in section 9.2.
    - o Fixed - attributes that are hard coded and may be specified by the Buyer in a POQ/Quote/Order request (subject to agreement between the Buyer and Seller) and may be returned by the Seller for an Inventory request (subject to agreement between the Buyer and Seller) as specified in section 9.3.

As noted above, the classification may depend on:

- Business Function - a given Product-Specific Attribute may, for example, be classified as Fixed for the Create POQ request; while it may be classified as Mandatory for the Create Product Order request.

- Product Action - a given Product-Specific Attribute may, for example, be classified as Mandatory for the Create POQ request for an INSTALL of a new product, while it may be classified as Fixed for the Create POQ request for a CHANGE of an installed Product.

- Product Offering - a given Product-Specific Attribute may, for example, be classified as Mandatory for the Create POQ request for a Product Offering (e.g., Premium Service),

while it may be classified as Fixed for the Create POQ request for a different Product Offering (e.g., Basic Service).

The Product-Specific Attribute classification can be defined and negotiated during the onboarding process or defined in a Product Catalog.

[R1] The Seller and Buyer **MUST** agree, for each Product-Specific Attribute, whether the attribute is Mandatory, Optional, or Fixed for each Business Function (POQ, Quote, Order) and Product Action (INSTALL, CHANGE) for a Product Offering.

[R2] The Seller and Buyer **MUST** agree, for each Product-Specific Attribute, whether the attribute is Mandatory, Optional, or Fixed for Inventory for a Product Offering.

[R3] If, for a Product Offering, a Product-Specific Attribute is classified as Optional for any Business Function and, if applicable, Product Action, the Seller and Buyer **MUST** agree on the default value for the attribute.

[R4] The Seller **MUST** reject an API request if the value for a Product-Specific Attribute requested by the Buyer is not a supported value for the applicable Product Offering.

The Access E-Line data model supports both INSTALL and CHANGE actions for POQ, Quote, and Order for the OVC and Operator UNI. The data model does not support these functions for ENNI – they are assumed to be created in the inventory as a result of the on-boarding process between two operators. Note that the DISCONNECT action does not require support by the data model.

The Access E-Line data model supports the RETRIEVE action for Inventory for the OVC, Operator UNI and ENNI. ENNIs 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.

9.1 Mandatory Product-Specific Attributes

[R5] If a Product-Specific Attribute is agreed to be Mandatory for a Business Function (POQ, Quote, Order), Product Action (INSTALL, CHANGE), and Product Offering, then the Buyer **MUST** include a value for the attribute in the corresponding API request.

---

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

[R6] If a Product-Specific Attribute is agreed to be Mandatory for Inventory for a Product Offering, then the Seller **MUST** include a value for the attribute in the corresponding API response.

[R7] When the Seller receives a POQ, Quote, or Order request in which any of the Mandatory Product-Specific Attributes are not included, the request **MUST** be rejected by the Seller.

9.2 Optional Product-Specific Attributes

[O1] If a Product-Specific Attribute is agreed to be Optional for a Business Function (POQ, Quote, Order), Product Action (INSTALL, CHANGE), and Product Offering, then the Buyer **MAY** include a value for the attribute in the corresponding API request.

[R8] The Seller **MUST** apply the agreed default value for an Optional Product-Specific Attribute if a value is not included by the Buyer in the corresponding API request.

[R9] If a Product-Specific Attribute is agreed to be Optional for Inventory for a Product Offering, then the Seller **MUST** include a value for the attribute in the corresponding API response if the value is not the agreed default value.

[O2] If a Product-Specific Attribute is agreed to be Optional for Inventory for a Product Offering, then the Seller **MAY** include a value for the attribute in the corresponding API response if the value is the agreed default value.

9.3 Fixed Product-Specific Attributes

A Product-Specific Attribute may be classified as Fixed for a Business Function, Product Action, and Product Offering when only one value is applicable for the Seller. This can be the case for example if:
* the Seller supports only a single value, or
* the value is derived by the Seller from the value of one or more other Product-Specific Attributes, or
* the Seller specifies a single value in the Product Catalog for a specific Product Offering, or
* the Buyer and the Seller agree on a single value during onboarding

Since these are Product-Specific Attributes, each value must still be agreed in some way between the Buyer and the Seller, which implies that even in the first two cases, the Seller must make the Buyer aware of what the value is or how it is derived, before the Buyer places an order. How this is done is outside the scope of this document.

The Seller applies the one applicable value for every request for which the Product-Specific Attribute is classified as Fixed.

[R10]   The Buyer and Seller **MUST** agree on whether the Buyer can include Product-Specific Attributes that have been classified as Fixed in API requests for POQ, Quote, and Order.

[R11]   If the Buyer and Seller agree that Product-Specific Attributes classified as Fixed cannot be included in API requests (see [R10]), the Buyer and Seller **MUST** agree on whether the Seller includes Product-Specific Attributes classified as Fixed in the corresponding API responses.

[R12]   If the Buyer and Seller agree that Product-Specific Attributes classified as Fixed cannot be included in API requests (see [R10]), the Seller **MUST** reject an API request from the Buyer if it includes a Product-Specific Attribute that has been classified as Fixed for the Business Function (POQ, Quote, Order), Product Action (INSTALL, CHANGE), and Product Offering.

[R13]   If the Buyer and Seller agree that Product-Specific Attributes classified as Fixed cannot be included in API requests (see [R10]), and if a Product-Specific Attribute is classified to be Fixed for Inventory for a Product Offering, then the Seller **MUST NOT** include a value for the Product-Specific Attribute in the Inventory API responses.

[R14]   If the Buyer and Seller agree that Product-Specific Attributes classified as Fixed can be included in API requests (see [R10]), the Seller **MUST** reject an API request from the Buyer if it includes a Product-Specific Attribute that has been classified as Fixed for the Business Function (POQ, Quote, Order), Product Action (INSTALL, CHANGE), and Product Offering and includes a value that is different than the agreed-on fixed value.

[R15]   If the Buyer and Seller agree that Product-Specific Attributes classified as Fixed can be included in API requests (see [R10]), and if a Product-Specific Attribute is classified to be Fixed for Inventory for a Product Offering, then the Seller **MUST** include a value for the Product-Specific Attribute in the Inventory API responses.

10 Order Milestones

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

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

Table 3 – Order Milestones for Access-E-Line

The Milestone Value in the first column of Table 3 is included in *ProductOrderEventPayload.milestoneName* in the Product Order Notification API.

11 Information Model for Access E-Line Product Data Model

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

The UNI may be a component of two different business agreements, one between a Service Provider and Subscriber for a Subscriber Ethernet Service, and one between a Service Provider and an Operator for an Operator Ethernet Service. 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 is 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. This product specification provides a data model for the business agreement between the Operator and the Service Provider (in MEF 26.2 [9] 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.
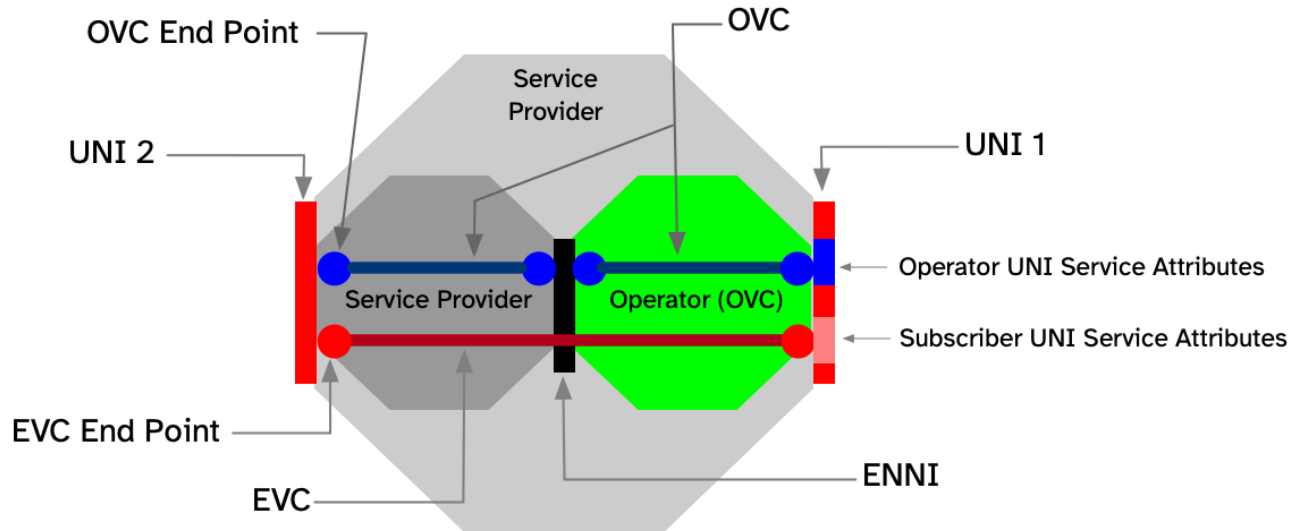
Figure 5 – Components of Subscriber and Operator Ethernet Services

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

11.1   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 [7] defines three superclasses for all EVC and OVC services, the Service (Virtual Connection), the External Interface, and the Service End Point. These superclasses are shown in green in the following diagram along with their sub-classes. The sub-classes relevant to Access E-Line are shown in the color boxes.
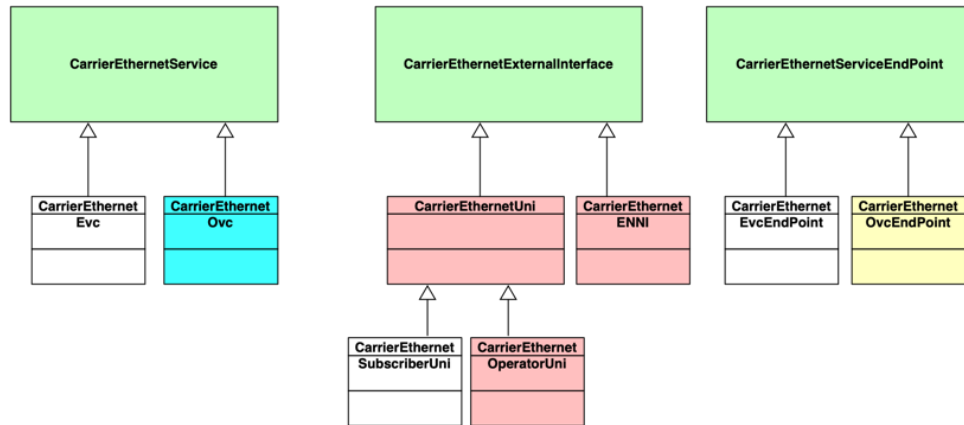
Figure 6 – Carrier Ethernet Superclasses

Although this provides an abstract model for understanding how the services are defined and how the various components of the Carrier Ethernet Services are organized, the model for the Carrier Ethernet Products is somewhat more complicated.

For example, while the EVC and OVC classes do share a few Service Attributes in common, in some cases the allowable values are different (e.g., the *maximumFrameSize* Service Attribute for EVC must be 1522 or greater, while *maximumFrameSize* for OVC must be 1526 or greater). Since these constraints must be reflected in the schemas, the two Service Attributes appear to be the same, but are actually different. While there are a few Service Attributes that could be included in a common superclass – for example, the *carrierEthernetSls* – a more flexible approach is taken whereby both the EVC and OVC classes (and product-specific variants of each) incorporate a common utility class (e.g. *CarrierEthernetSls*) by composition.

As a result, the actual organization of the Carrier Ethernet schemas differs somewhat from the service model shown in Figure 6 and is shown in Figure 7.

In the context of an Access E-Line Service, and the Service Attributes in MEF 26.2, the OVC, UNI, ENNI, and OVC End Points are organized as follows (the colors in the following diagram are intended to match the colors in the previous diagram):
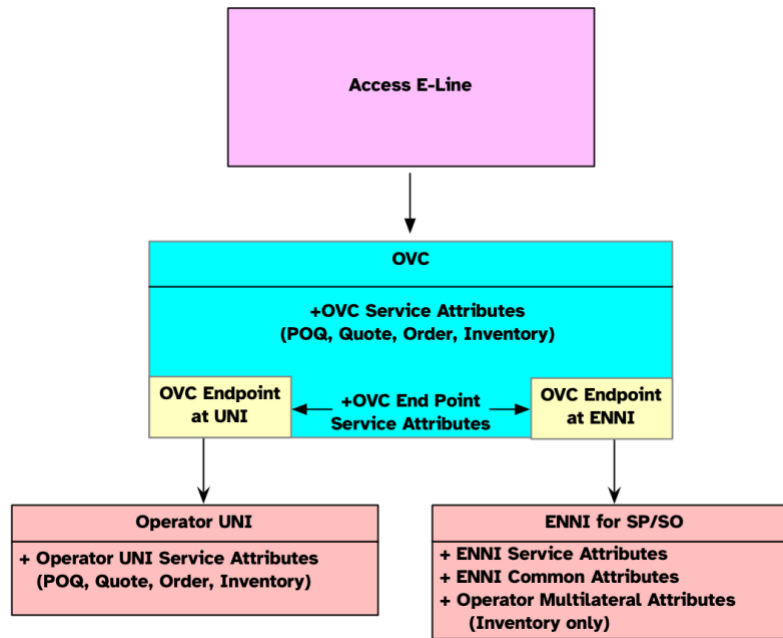
Figure 7 – Access E-Line Information Model

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

12 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 based on JSON schema draft 7 and encoded in YAML. These schemas accompany this document. This section explains the organization and structure of these schemas.

12.1 Organization and Structure of the Schemas

The schemas are organized into a file structure as shown in Figure 8.
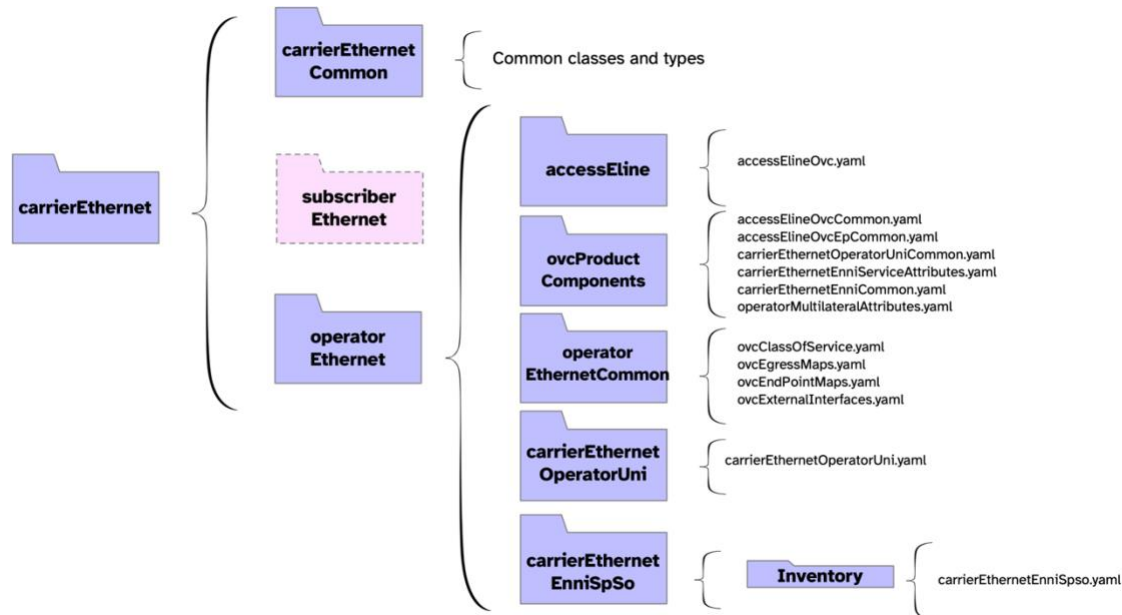
Figure 8 – Schema File Organization

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

The Service Attribute definitions are defined in a set of common files in the ovcProductComponents folder. These common classes are referenced in the relevant product component schema files. For example, accessElineOvc.yaml file includes:

```
allOf:
 - $ref: "../ovcProductComponents/accessElineOvcCommon.yaml#/definitions/AccessElineOvcCommon"
```

The ENNI schema is composed from three component classes:
- *carrierEthernetEnniServiceAttributes.yaml*: Includes attributes that represent the agreement between the Service Provider and the Operator for the ENNI for one Service Provider.
- *carrierEthernetEnniCommon.yaml*: Includes the attributes that describe the physical and logical characteristics of the ENNI and are agreed to by the Operators on each side 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*.

- *operatorMultilateralAttributes.yaml*: Includes attributes that describe ENNI parameters that apply to each side of the ENNI and are agreed to by the Operators on each side 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*.

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

Note that the "Carrier Ethernet" prefix for a schema indicates that it is not specific to a particular service. For example, the *Carrier_Ethernet_Link_Aggregation* schema in the *carrierEthernetCommon* 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, and Subscriber Ethernet services.

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

[R16] The Product Inventory for an ENNI **MUST** include all the ENNI Service Attributes.

[R17] The Product Inventory for an ENNI **MUST** include all the ENNI Common Attributes and Operator Multilateral Attributes if the value of the ENNI Service Attribute *SVLAN Control* is *Full*.

12.2   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*):

```
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"
```

12.3   Additional Details

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

12.3.1   Naming Conventions

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

12.3.2   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 might 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.

12.3.3 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 also be used instead of the *Operator UNI Envelopes Service Attribute* (with the same limitations noted above for the ENNI). 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.

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

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*.

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

Table 4 – Product Relationship Roles

[R18] For an Access E-Line OVC product, the Relationship Type field of the Product Relationship, POQ Item Relationship, Quote Item Relationship and Order Item Relationship types **MUST** contain one of the values shown in the Relationship Type column in Table 4.

[R19] For POQ, Quote and Order, the relationship to an Operator UNI **MUST** be specified for every INSTALL of, or CHANGE to, an Access E-Line OVC product.

[R20] For an Access E-Line OVC product, the relationship to an Operator UNI **MUST** reference an Operator UNI product or an equivalent POQ Item, Quote Item, or Order Item.

[R21] For a CHANGE to an Access-E-Line OVC product the relationship to the Operator UNI **MUST NOT** be changed from the value present in the Product Inventory.

Note that [R21] indicates that once an Access E-Line OVC is associated with an Operator UNI, it cannot be associated with a different Operator UNI.

[R22] For POQ, Quote and Order, the relationship to an ENNI **MUST** be specified for every INSTALL of, or CHANGE to, an Access E-Line OVC product.

[R23]   For an Access E-Line OVC product, the relationship to an ENNI **MUST** reference an ENNI product.

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 CHANGE and once a UNI is associated with a specific location, the INSTALL_LOCATION cannot be changed. The install location is captured in the product-agnostic part of the POQ, Quote and Order APIs. The value in the Place Relationship Role column in the table below is used in the *role* field of the *RelatedPlaceRefOrValue* type.

Product	Place Relationship Role	INSTALL	CHANGE
Operator UNI	INSTALL_LOCATION	Mandatory	Mandatory

Table 5 – Place Relationship Role

[R24]   For an Operator UNI product, the Role field (*role*) of the Related Place (*RelatedPlaceRefOrValue)* type **MUST** contain one of the values shown in the Place Relationship Role column in Table 5.

[R25]   For POQ, Quote, and Order, the Related Place (*RelatedPlaceRefOrValue)* **MUST** be specified for every INSTALL of, or CHANGE to, an Operator UNI.

[R26]   For a CHANGE to an Operator UNI product the Related Place **MUST NOT** be changed from the value present in the Product Inventory.

Changing the UNI Location is not supported for an Operator UNI. The value included in a CHANGE request must be identical to the value in the Inventory.

The relationships applicable to Access E-Line are shown in the following diagram:
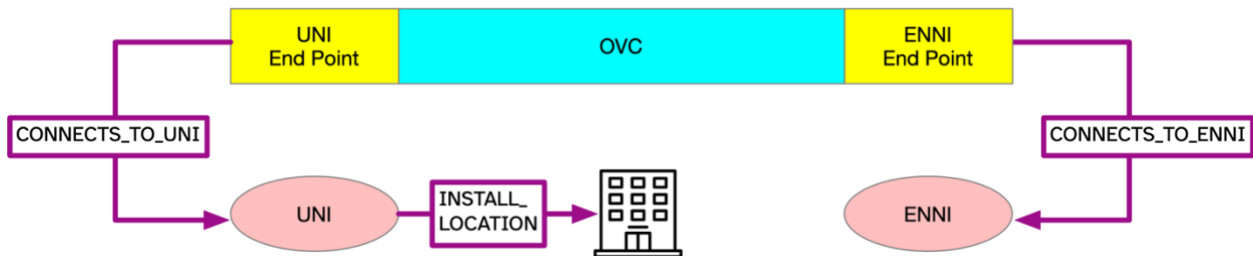
14 Access E-Line Service Attributes

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. 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 16. 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 (see section 12.3.3).

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) and, in some cases, validation notes (light red background).

Note that several of the validation notes specify a relationship between properties, for example, the validation note for the *OVC Maximum Frame Size* Service Attribute includes that it must be less than the *Operator UNI Maximum Service Frame Size* + 4. This is not a statement about whether either of these properties must appear in the API request—one might be required and the other fixed or optional, or they might both be fixed or optional. Rather, it is an indication of the required relationship between the properties regardless of whether values are derived directly from the API request or from the Product Catalog or from another source.

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

Access Eline OVC
  Schema File Name: accessElineOvc
    urn:mef:lso:spec:sonata:access-eline-ovc:v5.0.0:common
  Includes classes:
    AccessElineOvc (Access E-Line OVC Service Attributes)
    AccessElineOvcEndPoint (Access E-Line OVC End Point Service Attributes)

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

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

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

  Allowed values: Compose (allOf): Class: AccessElineOvcCommon

---

  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. The UNI OVC End Point must be included in the Access E-Line Product.

  Allowed values: Class: AccessElineOvcEndPoint

  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. The ENNI OVC End Point must be included in the Access E-Line Product.

  Allowed values: Class: AccessElineOvcEndPoint

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.

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

  Allowed values: Compose (allOf): Class: AccessElineOvcEpCommon

---

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. This class is used to compose the AccessElineOvc class.

  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

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

  • It must be less than or equal to **OperatorMultilateral.maximumFrameSizeMultilateral**, and

---

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

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

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

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

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

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

Allowed values: Type: EnabledDisabled

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

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

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

Allowed values: Type: EnabledDisabled

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

Property Name: *OVC List of Class of Service Names Service Attribute*
  Reference: MEF 26.2 sec. 12.12
    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]

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. An empty list indicates the value *None* per MEF 26.2.

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

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

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

Allowed values: Class: FrameDisposition

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

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

Allowed values: Type: AvailableMegLevel

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

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

Allowed values: Type: L2cpAddressSet

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Allowed values: Select (oneOf): Class: CosNameFromEp, Class: CosNameFromStagPcp, Class: CosNameFromCtagPcp, Class:CosNameFromDscp   **with Discriminator:** mapType

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

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

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

Description: The OVC EP Color Map Service Attribute is the mechanism by which the Color for an Ingress External Interface Frame that is mapped to an OVC EP is indicated by the content in the frame header. The value of the OVC EP Color Map

---

Service Attribute is a pair of the form ⟨F, M⟩ where: F is one of the values *ENDPOINT*, *DEI*, *PCP*, or *DSCP* and M is a map that can be used to assign Color to each Ingress Frame. (The form of M depends on the value of F.)

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

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

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

Description: A set of mappings that determine the content of the S-Tag or C-Tag of an egress frame at an ENNI or UNI (respectively) based on the Class of Service Name and Frame Color of the frame. Three forms: (1) CoSName(CN) -> C/S-Tag PCP or Discard, (2) CoSName and Frame Color (CC)--> C/S-Tag DEI or Discard, and (3) CoSName and Frame Color (CC)--> C/S-Tag PCP or Discard. A fourth form is included which combines (1) and (2) because that is a common case. An empty list indicates that the value is not specified per section 16.8.7 of MEF 26.2.

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

Validation Notes:

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

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

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

Allowed values: Select (oneOf): Class: CosNameFromEp, Class: CosNameFromStagPcp, Class: CosNameFromCtagPcp, Class:CosNameFromDscp **with Discriminator:** mapType

---

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

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

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

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

If this attribute is a non-empty list at the UNI, ***AccessElineOvcEndPoint.egressBwpPerEgressEquivalenceClassName*** must not be included in the request or must be included as an empty list.
 Coding hint: Per MEF 51.1 this must be Disabled (an empty list) at ENNI EP

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

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

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

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

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

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

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

Validation Notes: If this Service Attribute has a value other than *Disabled*, there **must** be a value included for *AccessElineOvcEpCommon.egressEquivalenceClassIdentifier* and *AccessElineOvcEndPoint.egressBandwidthProfilePerEndPoint* **must not** have a value..

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. The value can be a non-empty list if and only if the ENNI uses Link Aggregation with type ALL_ACTIVE.

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

Validation Notes:

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

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

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

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

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

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

Allowed values: Type: MegLevel

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

Description: A list of the MEPs instantiated at this OVC EP, if any, and the direction of each MEP, List of 2-tuples ⟨l, d⟩ where l is MEG level and d is direction (Up/Down). An empty list indicates that no MEPs are instantiated at the OVC EP.

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

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

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

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

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

To create an Envelope containing Bandwidth Profile Flows for more than one OVC End Point, the UNI or ENNI Envelopes

Service Attribute (as appropriate) must be used. If this is an empty list, then there are no Envelopes defined at this End Point or the Envelopes are defined at the external interface (UNI or ENNI).

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

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

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

Description: A mapping of Port Conversation IDs (i.e., VLAN IDs) to an ordered list of physical links if ***EnniService.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

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)

14.2   Operator UNI Service Attributes

Carrier Ethernet Operator UNI
  Schema File Name: carrierEthernetOperatorUni
  urn:mef:lso:spec:sonata:carrier-ethernet-operator-uni:v5.0.0:common
  Includes classes:
    OperatorUNI (Operator UNI Service Attributes)

Class: OperatorUNI *(Operator UNI Service Attributes)*

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

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

Description: The value of the Operator UNI List of Physical Links Service Attribute is a list of *UNIPhysicalLinks*, one entry for each physical link that composes the UNI. UNIPhysicalLinks specifies several properties of the physical link such as the PHY type and the connector type.

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

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

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

Allowed values: Type: LinkAggType

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

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

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

Description: A mapping of a list of Port Conversation IDs (i.e., VLAN IDs or zero for untagged frames) to an ordered list of physical links if there are multiple physical links in the UNI and the link aggregation is ALL_ACTIVE. This must be an empty list if and only if the value of *OperatorUniCommon.linkAggregation* is not ALL_ACTIVE or OTHER.

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

Validation Notes: If this is not an empty list, it must map every C-VLAN ID mapped to the OVC.

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

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

Allowed values: Minimum: 1522

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

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

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

Allowed values: Minimum: 1 Maximum: 4094

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

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

Allowed values: Minimum: 1 Maximum: 4094

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

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

Allowed values: Minimum: 1 Maximum: 4094

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

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

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

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

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

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

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

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

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

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

Allowed values: Type: EnabledDisabled

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

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

Allowed values: Type: EnabledDisabled

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

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

Allowed values: Type: EnabledDisabled

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

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

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

Allowed values: Type: EnabledDisabled

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

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

Description: The Envelopes and Envelope Coupling Flag values to which Bandwidth Profile Flows can be mapped. Value is list of {envelopeID, couplingFlagForIndexZero}. **Note:** Envelopes for the UNI can also be specified at the OVC End Points associated with the UNI.An empty list indicates that no Envelopes are defined at the UNI.

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

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

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

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

Allowed values: Type: L2cpAddressSet

Validation Notes: If **UniEP.ovcEndPointMap** does not include all C-tags (i.e. 1-4094) 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 (if there are no peered Protocols), or a list of entries identifying protocols to be Peered where each entry consists of {Destination Address, Protocol Identifier} or {Destination Address, Protocol Identifier, Link Identifier}.

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

Validation Notes: If **AccessElineOvc.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.

14.3 ENNI Attributes

Carrier Ethernet ENNI SP SO
  Schema File Name: carrierEthernetEnniSpSo
    urn:mef:lso:spec:sonata:carrier-ethernet-enni-sp-so:v5.0.0:common
  Includes classes:
    EnniSpso (ENNI SP/SO Attributes)

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

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

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

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

---

14.3.1 ENNI Service Attributes

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

Class: EnniService *(ENNI Service Attributes)*

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

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

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

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

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

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

    Allowed values: Minimum: 1

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

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

    Allowed values: Minimum: 1

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

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

    Allowed values: Type: EnabledDisabled

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

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

Description: The Envelopes and Envelope Coupling Flag values to which Bandwidth Profile Flows can be mapped. Value is list of {envelopeID, couplingFlagForIndexZero}. **Note:** Envelopes for the ENNI can also be specified at the OVC End Points associated with the ENNI. An empty list indicates that no Envelopes are defined at the ENNI.

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

14.3.2   ENNI Common Attributes

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

Class: EnniCommon   *(ENNI Common Attributes)*

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

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

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

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

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

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

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

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

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

Allowed values: Type: LinkAggType

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

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

Description: A mapping of Port Conversation IDs (i.e., VLAN IDs or zero for untagged frames) to an ordered list of physical links if there are multiple physical links in the UNI and the link aggregation is ALL_ACTIVE. **Note:** In some cases the Port Conversation ID to Aggregation Link Map can be specified at the OVC End Point. Any empty list (implying no map) is allowed only when the ENNI Link Aggregation Common Attribute is not ALL_ACTIVE.

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

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

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

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

Allowed values: Type: EnabledDisabled

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

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

Allowed values: Type: EnabledDisabled

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

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

Allowed values: Type: EnabledDisabled

14.3.3   Operator Multilateral Attributes

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

Class: OperatorMultilateral *(Operator Multilateral Attributes)*

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

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

Description: A list of Layer 2 Control Protocols that will be Peered by a protocol entity at an ENNI: an empty list (if there are no peered Protocols), or a list of entries identifying protocols to be Peered where each entry consists of {Destination Address, Protocol Identifier} or {Destination Address, Protocol Identifier, Link Identifier}

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

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

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

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

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

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

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

Allowed values: Minimum: 1526

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

15 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.

Classes that are common between Operator Ethernet and Subscriber Ethernet are described in MEF 125 [21], the Subscriber Ethernet equivalent to this document, and this document just includes a list of the classes. Each of the following sections contains either: (1) a list of common classes that are described in MEF 125 or (2) a list of Operator Ethernet-specific classes along with their descriptions or (3) both (i.e., a list of common classes and a list of Operator Ethernet-specific classes with descriptions).

15.1 Bandwidth Profiles

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

See MEF 125 section 15.1 for a description of the common Bandwidth Profile classes listed above.

15.2 Class of Service

> **Carrier Ethernet Class of Service**
>   Schema File Name: carrierEthernetClassOfService
>   Includes classes:
>   CosNameFromEp (CoS Name From EP)
>   CosNameFromCtagPcp (CoS Name From C-Tag PCP)
>   CosUniPcpMapEntry (PCP To CoS Map Entry)
>   CosNameFromStagPcp (CoS Name From S-Tag PCP)
>   CosStagPcpMapEntry (S-Tag PCP to CoS Name Map Entry)
>   CosNameFromDscp (CoS Name From DSCP)
>   CosDscpMap (DSCP to CoS Name Map)
>   CosDscpMapEntry (DSCP to CoS Name Map Entry)
>   DscpValues (List of DSCP Values)
>   CosL2cp (L2CP to CoS Name)

See MEF 125 15.2 for a description of the common Class of Service classes listed above. The following table includes the Operator Ethernet-specific Class of Service classes.

> **OVC Class of Service**
>   Schema File Name: ovcClassOfService
>   Includes classes:
>   CosNameFromStagPcp (CoS Name From S-Tag PCP)
>   CosStagPcpMapEntry (S-Tag PCP to CoS Name Map Entry)
>
> ---
>
> **Class:** CosNameFromStagPcp *(CoS Name From S-Tag PCP)*
>
> *Ingress CoS Name is derived from {F,M,P} where F=S_TAG_PCP. This maps each of the 8 PCP values in the S-tag to a Class of Service Name.*
>
> ---
>
>   **Property Name:** *CoS Map Type - STAG*
>   **JSON Name:** *mapType* - **Type:** *const*

Description: The Field that indicates F = S_TAG_PCP.

Allowed values: "S_TAG_PCP"

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

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

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

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

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

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

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

Class: CosStagPcpMapEntry *(S-Tag PCP to CoS Name Map Entry)*

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

Property Name: *S-Tag PCP Value*
   JSON Name: *pcpVal* - **Type:** *object*

Allowed values: Type: PcpValue

Property Name: *S-Tag CoS Name*
   JSON Name: *pcpCosName* - **Type:** *string*

Allowed values: [Any String]

15.3   Color Identifier

Carrier Ethernet Color Identifier
   Schema File Name: carrierEthernetColorIdentifier
   Includes classes:
      ColorFromDei (Color From DEI)
      ColorFromPCP (Color From PCP)
      ColorFromPCPMapEntry (Color from PCP Map Entry)
      ColorFromDscp (Color From DSCP)
      ColorFromDscpEntry (Color From DSCP Entry)
      ColorFromEp (Color From EP)

See MEF 125 section 15.3 for a description of the common Color Identifier classes listed above.

15.4   Egress Maps

OVC Egress Maps
   Schema File Name: ovcEgressMaps
   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)

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

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

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

Allowed values: "CN_PCP"

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

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

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

Class: CnPcp *(CNPCP Entry)*

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

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

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

Allowed values: [Any String]

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

Allowed values: Type: PcpValueAndDiscard

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

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

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

Allowed values: "CC_DEI"

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

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

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

Class: CcDei *(CCDEI Entry)*

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

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

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

Allowed values: [Any String]

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

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

Allowed values: Type: DeiValueAndDiscard

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

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

Allowed values: Type: DeiValueAndDiscard

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

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

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

Allowed values: "CC_PCP"

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

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

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

Class: CcPcp *(CCPCP Entry)*

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

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

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

Allowed values: [Any String]

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

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

Allowed values: Type: PcpValueAndDiscard

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

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

Allowed values: Type: PcpValueAndDiscard

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

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

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

Allowed values: "CN_PCP_CC_DEI"

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

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

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

Class: CnPcpCcDei *(CNPCPCCDEI Entry)*

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

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

Allowed values: [Any String]

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

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

Allowed values: Type: PcpValueAndDiscard

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

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

Allowed values: Type: DeiValueAndDiscard

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

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

Allowed values: Type: DeiValueAndDiscard

15.5   End Point Maps

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

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

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

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

Description: The discriminator for Map Type E

Allowed values: "FORM_E"

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

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

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

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

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

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

Description: The discriminator for Map Type T

Allowed values: "FORM_T"

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

Description: The S-VLAN ID used for Root traffic

Allowed values: Minimum: 1 Maximum: 4094

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

Description: The S-VLAN ID used for Leaf traffic

Allowed values: Minimum: 1 Maximum: 4094

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

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

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

Description: The discriminator for Map Type V

Allowed values: "FORM_V"

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

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

Allowed values: Minimum: 1 Maximum: 4094

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

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

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

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

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

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

Description: The discriminator for Map Type U

Allowed values: "FORM_U"

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

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

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

15.6   External Interfaces

Carrier Ethernet External Interfaces
   Schema File Name: carrierEthernetExternalInterfaces
   Includes classes:
      UniPhysicalLink (UNI Physical Link)
      CarrierEthernetPhysicalLink (Carrier Ethernet Physical Link)
      UniSpecificAttributes (UNI Specific)

See MEF 125 section 15.7 for a description of the common External Interfaces classes listed above. The following table contains the Operator Ethernet-specific External Interface classes.

OVC External Interfaces
   Schema File Name: ovcExternalInterfaces
   Includes classes:
      EnniPhysicalLink (ENNI Physical Link)

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).*
>
> **Property Name:** *Carrier Ethernet Physical Link* **Type:** *object*
> **Allowed values:** Compose (allOf): Class: CarrierEthernetPhysicalLink

15.7   Layer 2 Control Protocols (L2CP)

Carrier Ethernet L2CP
    Schema File Name: carrierEthernetL2cp
    Includes classes:
        L2cpPeering (L2CP Peering Identifier)
        L2cpProtocol (L2CP Protocol Identifier)

See MEF 125 section 15.8 for a description of the common L2CP classes listed.

15.8   Link Aggregation

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

See MEF 125 section 15.9 for a description of the common L2CP classes listed.

15.9   Service Level Specification

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

See MEF 125 section 15.10 for a description of the common Service Level Specification classes listed above.

15.10   Standard Enums

```
Carrier Ethernet Enums
   Schema File Name: carrierEthernetEnums
   Includes classes:
      EnabledDisabled  (Enabled Disabled)
      PcpValueAndDiscard  (PCP Value and Discard Enum)
      DeiValueAndDiscard  (DEI and Discard Enum)
      LinkAggType  (LinkAgg Enum)
      FrameColor  (FrameColor Enum)
      L2cpAddressSet  (L2CP Address Set Enum)
      MegLevel  (MEG Level Enum)
      AvailableMegLevel  (Available MEG Level Enum)
      PcpValue  (PCP Value Enum)
      TimeDurationUnits  (Time Duration Units)
      InformationRateUnits  (Information Rate Units)
      DataSizeUnits  (Data Size Units)
```

See MEF 125 section 15.11 for a description of the common enums.

15.11   Utility Classes

```
Carrier Ethernet Utility Classes
   Schema File Name: carrierEthernetUtilityClasses
   Includes classes:
      MepLevelAndDirection  (MEP Specification)
      SourceMacAddressLimit  (SA Limit)
      AggLinkDepth  (Link Depth)
      FrameDisposition  (Frame Disposition)
      FrameDispEnum  (Frame Disposition Enumeration)
      InformationRate  (Information Rate)
      DataSize  (Data Size)
      TimeDuration  (Time Duration)
      Percentage  (Percentage)
```

See MEF 125 section 15.12 for a description of the common Utility classes listed above.

16 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].

Table 6 – Service Attributes Not Included in Schemas

17 References

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

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

[3] IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, by Scott Bradner, March 1997

[4] IETF RFC 8174, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, by Barry Leiba, May 2017. Copyright © IETF Trust and the persons identified as the document authors (2017). All Rights Reserved.

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

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

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

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

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

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

[11] MEF 51.1, *Operator Ethernet Service Definitions*, December 2018

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

[13] MEF 57.2, *Product Order Management Requirements and Use Cases*, August, 2022

[14] MEF 61.1, *IP Service Attributes*, May 2019

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

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

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

[18] MEF 87, *LSO Cantata and LSO Sonata Product Offering Qualification API – Developer Guide*, May 2022

[19] MEF 115, *LSO Cantata and LSO Sonata Quote Management API – Developer Guide*, May 2022

[20] MEF 121, *LSO Cantata and LSO Sonata Address Management API – Developer Guide*, May 2022

[21] MEF 125, *LSO Cantata and LSO Sonata Subscriber Ethernet Product Schemas and Developer Guide*, February 2023

Appendix A   Usage examples (Informative)

This appendix aims to provide an extensive set of examples to cover:
- different configuration variants (Use Cases 2 and 3)
- basic all APIs steps walkthrough to order an Access E-Line product (Use Cases 1, 2, 4, 5, 6)
- common modifications (Use Cases 7-11, 13)
- deletion of a product (Use Case 12)

The examples are delivered in two forms:
- as part of this document – to allow comments and rich explanation
- as a Postman collection – for ease of use in testing.

A.1   High-Level flow

The Cantata and Sonata Interface Reference Points are formed from a set of APIs the serve different functions in the end-to-end flow. Figure 9 shows all of the functions and their sequence.
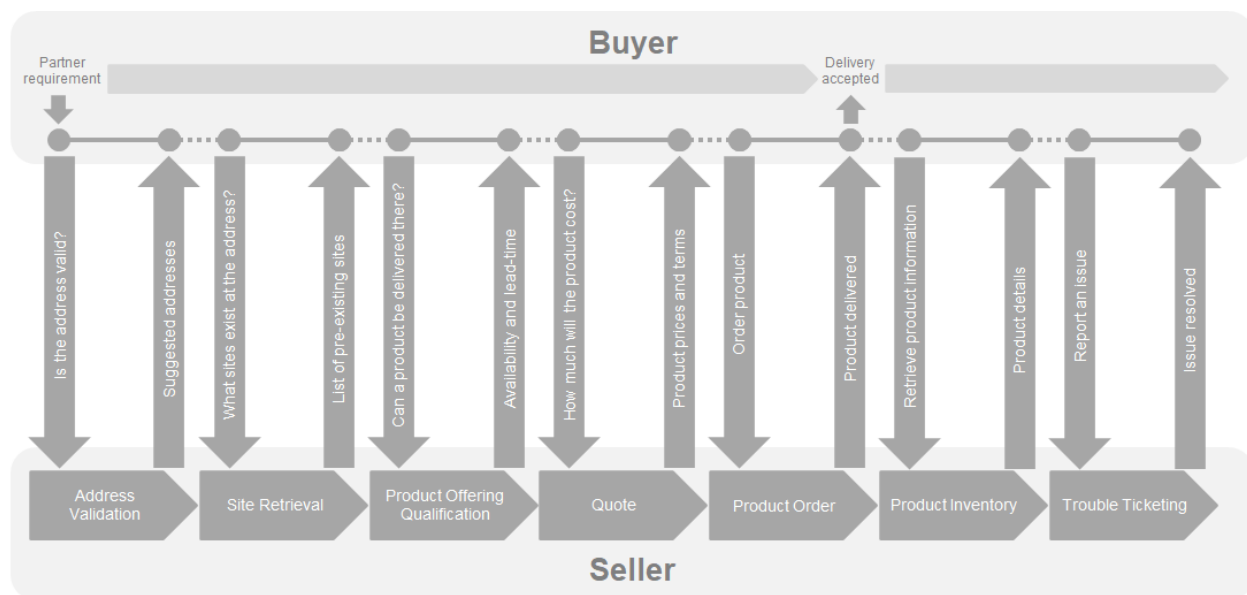
Figure 9 – Cantata and Sonata End-to-End Function Flow

- Address Validation - allows the Buyer to retrieve address information from the Seller, including exact formats, for addresses known to the Seller.
- Site Retrieval - allows the Buyer to retrieve Service Site information including exact formats for Service Sites known to the Seller.
- Product Offering Qualification (POQ) - allows the Buyer to check whether the Seller can deliver a product or set of products from among their product offerings at the geographic address or a service site specified by the Buyer; or modify a previously purchased product.

- Quote - allows the Buyer to submit a request to find out how much the installation of an instance of a Product Offering, an update to an existing Product, or a disconnect of an existing Product will cost.
- Product Order - allows the Buyer to request the Seller to initiate and complete the fulfillment process of an installation of a Product Offering, an update to an existing Product, or a disconnect of an existing Product at the address defined by the Buyer.
- Product Inventory - allows the Buyer to retrieve the information about existing Product instances from Seller's Product Inventory.
- Trouble Ticketing - allows the Buyer to create, retrieve, and update Trouble Tickets as well as receive notifications about Incidents' and Trouble Tickets' updates. This allows managing issues and situations that are not part of normal operations of the Product provided by the Seller.

All of the above-mentioned APIs are provided in the SDK together with accompanying Developer Guides. Please refer to those documents for more details and examples of particular functional APIs.

A.2   Integration of product specifications into the APIs.

The above-mentioned APIs are product-agnostic in the meaning that they serve as a business interaction level between the Buyer and the Seller and they do not contain any product-specific information in their specifications. In order to pass the product-specific information, an extension pattern must be used. This applies to four APIs that carry product-specific information: POQ, Quote, Product Order, and Product Inventory.

The extension hosting type in the API data model is "MEFProductConfiguration". The "@type" attribute of that type must be set to a value that uniquely identifies the product specification (Figure 10). A unique identifier for MEF standard product specifications is in URN format and is assigned by MEF. This identifier is provided as root schema "$id" and in product specification documentation. In this case, this will be one of:
- urn:mef:lso:spec:sonata:access-eline-ovc:v5.0.0:all
- urn:mef:lso:spec:sonata:carrier-ethernet-operator-uni:v5.0.0:all
- urn:mef:lso:spec:sonata:carrier-ethernet-enni-sp-so:v5.0.0:inventory

Use of non-MEF standard product definitions is allowed. In such a case the schema identifier must be agreed upon between the Buyer and the Seller.
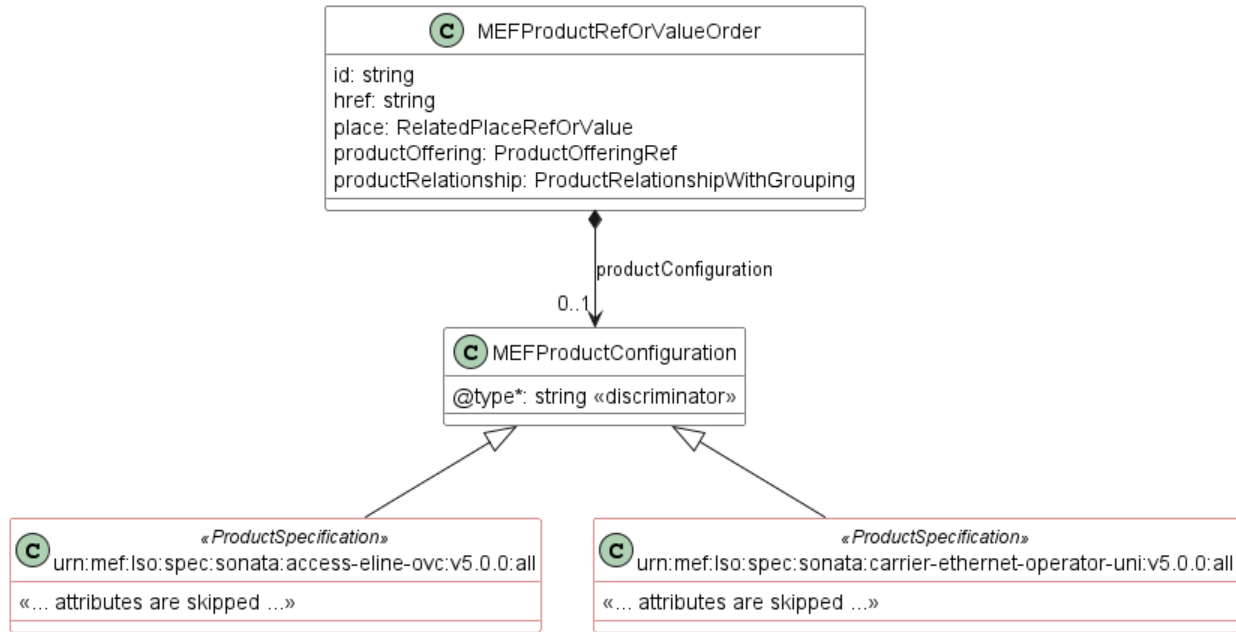
Figure 10 – The Extension Pattern

Product specifications are provided as Json schemas without the "MEFProductConfiguration" context. Product-specific attributes are introduced via the "MEFProductRefOrValue" (defined by the Buyer). This entity has the "productConfiguration" attribute of type "MEFProductConfiguration" which is used as an extension point for product-specific attributes. The example result of such binding in a request payload may look like this (for POQ):

```json
{
  "instantSyncQualification": true,
  "externalId": "BuyerPoq-00001",
  "provideAlternative": false,
  "projectId": "BuyerProjectX",
  "productOfferingQualificationItem": [
    {
      "id": "item-001",
      "action": "add",
      "product": {
        "productOffering": {
          "id": "000073"
        },
        "productConfiguration": {
          "@type": "urn:mef:lso:spec:sonata:access-eline-ovc:v5.0.0:all",
          "ceVlanIdPreservation": "PRESERVE",
          "cTagPcpPreservation": "ENABLED",
          "cTagDeiPreservation": "ENABLED",
          "maximumFrameSize": 1526,
          "ovcL2cpAddressSet": "CTA",
          <<the rest of the attributes omitted>>
```

POQ API part

```
        …
      }
    }
  }
  ]
}
```


Access E-Line Product part

A.3   action: add

This section guides through all the steps of Sonata and Cantata APIs that need to be performed in order to successfully order an Access E-Line product.

Note: Sellers are free to mandate some of these steps.

Note: As the examples of particular steps in many cases will replicate the product-specific information, in some of the snippets some parts of it will be omitted for better readability.

There are common rules for all request items for creation requests (POQ, Quote, Order):

- "item.action" must be set to "add"

- "item.product.id" must not be provided

- "product.productConfiguration" must contain all desired configurations.

A.3.1   Use Case 1: Address Validation

For detailed guidance on how to use the Address Validation API, please refer to MEF 121 [20].

The first step of the process is the Address Validation. The aim of this step is to align the address representation between the Buyer and the Seller. This is to overcome the very common problem of different address representation in various countries and systems. The Buyer sends a representation of the address that is intended to be used in further steps (most likely an installation place). The question is "Dear Seller – do you recognize and understand this address?". Additionally, the Buyer may also ask the Seller to provide alternatives if there is no clear match. The Seller provides a response where in the "bestMatchGeographicAddress" (if found) a matching address is provided with an id that can be used in further steps to avoid the need for Address resolution.

Note: It is not mandatory for the Seller to provide the Id of the returned Address, yet it is recommended.

Note: The Seller's response might come with some enhancements in the Address. It is up to the Seller's discretion what makes the best match and an alternative.

The Buyer in the request places one of 4 possible representations of the Address (FieldedAddress, FormattedAddress, MEFGeographicPoint, or GeographicAddressLabel). The following Figure and snippet present an example request:
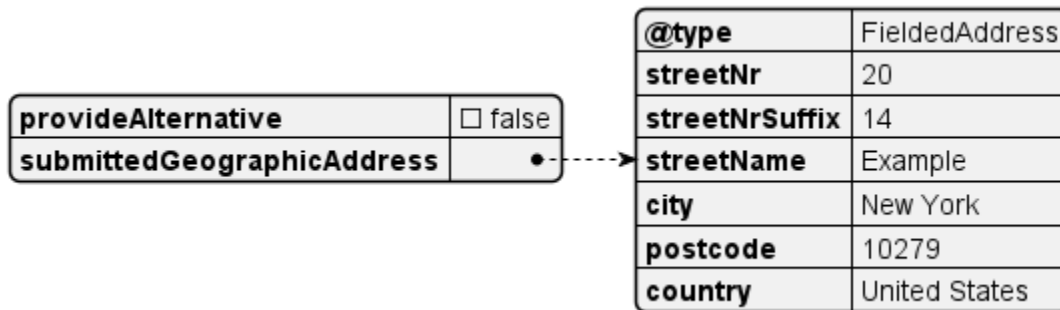
Figure 11 – UC1: Address Validation request

Example Address Validation Request:
```
{
  "provideAlternative": false,
  "submittedGeographicAddress": {
    "@type": "FieldedAddress",
    "streetNr": "20",
    "streetNrSuffix": "14",
    "streetName": "Example",
    "city": "New York",
    "postcode": "10279",
    "country": "United States"
  }
}
```

In the response, the Seller repeats the submitted address for reference and populates the "bestMatchGeographicAddress" and/or the "alternateGeographicAddress". In the example, the Seller matches the best match address, which has a little more details than the one in the request. The Seller also provides the address id ("NewYorkAddress-id-1") that the Buyer will refer to in later steps.

Note: The identifiers will most likely be some kind of technical ids to provide uniqueness. In all examples, the identifiers are shortened and made human-readable to make it easier to read and match across the use cases.
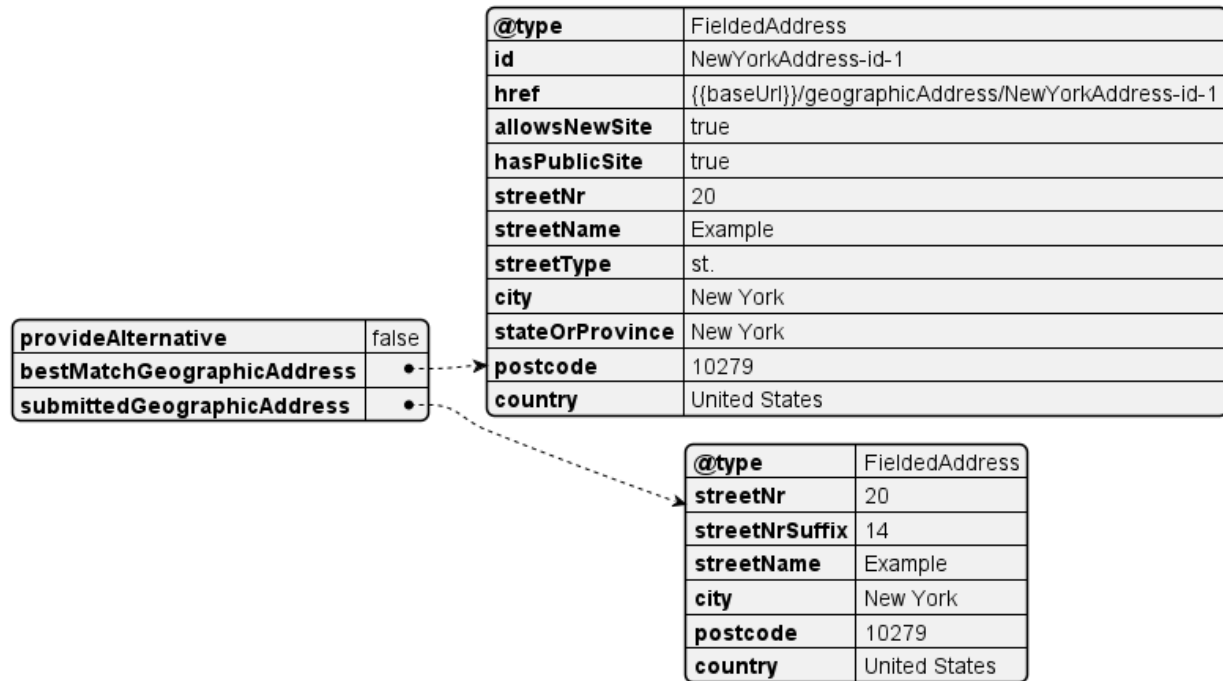
Figure 12 – UC1: Address Validation response

Seller's response:
```
{
  "provideAlternative": "false",
  "bestMatchGeographicAddress": {
    "@type": "FieldedAddress",
    "id": "NewYorkAddress-id-1",
    "href": "{{baseUrl}}/geographicAddress/NewYorkAddress-id-1",
    "allowsNewSite": "true",
    "hasPublicSite": "true",
    "streetNr": "20",
    "streetName": "Example",
    "streetType": "st.",
    "city": "New York",
    "stateOrProvince": "New York",
    "postcode": "10279",
    "country": "United States"
  },
  "submittedGeographicAddress": {
    "@type": "FieldedAddress",
    "streetNr": "20",
    "streetNrSuffix": "14",
    "streetName": "Example",
```

```
    "city": "New York",
    "postcode": "10279",
    "country": "United States"
  }
}
```

A.3.2 Use Case 2: POQ - new Access E-Line, new UNI, low class of service

For detailed guidance on how to use the Product Offering Qualification API, please refer to MEF 87 [18].

The Product Offering Qualification step is designed for the Buyer to ask the question "Dear Seller, are you able to provide a certain product (based on "productOffering" and "productConfiguration") at a given location"? The Seller responds with qualification confidence:
- green - The Seller has high confidence that this Product can be delivered,
- yellow - The Seller believes they can deliver the Product but is not highly confident,
- red - The Seller cannot deliver the Product as specified.

In case of yellow or red, additionally, the Seller may return (if requested) an alternative Product Offering, that might fulfill the Buyer's needs.

It is very important to understand the pattern of integrating the product configuration (so-called "payload") with the functional product-agnostic API ("envelope"). As explained in chapter 11, the Access E-Line product model is composed of 3 elements (products):
- the Access E-Line itself. It contains the "uniEp" and "enniEp" attributes, which carry some endpoint configuration information, yet these are no references to the UNI or ENNI products.
- the UNI
- the ENNI.

The information about one single product is carried within the Product Offering Qualification (POQ) API by a single "productOfferingQualificationItem" being a subject to qualification. One POQ Request can carry more than one POQ Items, that may or may not be related to each other.

There are 2 ways to reference products:
- existing Products – present in the Inventory at the moment of issuing the request, to which the Buyer has the "product.id". These must be referenced by "productOfferingQualificationItem.product.productRelationship" with appropriate "product.id" and "relationshipType". Product Specification defines what roles must be used during referencing other products. The Access E-Line points to the ENNI product with the "relationshipType" value: "CONNECTS_TO_ENNI" (as specified in Chapter 13).
- newly created or modified products – ones being created or modified by other POQ Item in the same POQ request, so there is a relation between the Items within a POQ. These must be referenced using the "productOfferingQualificationItem.qualificationItemRelationship" by the target Item "id" and the "relationshipType" (CONNECTS_TO_UNI).

In this use case, both the Access E-Line (OVC) and the UNI products are created or, to be more precise, a request to qualify if the creation of both of them is possible. It is assumed, the ENNI is always existing, before ordering an Access E-Line (Chapter 9). Since 2 products are being subject to qualification, the POQ request contains 2 items with "action=add". The Access E-Line POQ Item has 2 relations:

- to the ENNI, which exists in the Inventory already – by "productOfferingQualificationItem.product.productRelationship".
- to the UNI, which is being qualified in the same request – by "productOfferingQualificationItem.qualificationItemRelationship"

An instance diagram in Figure 13 shows an extracted part from the request, to present the most important integration-related attributes. The product configuration attached to a POQ request is highlighted with green color, and the product relations are highlighted with a bold font.
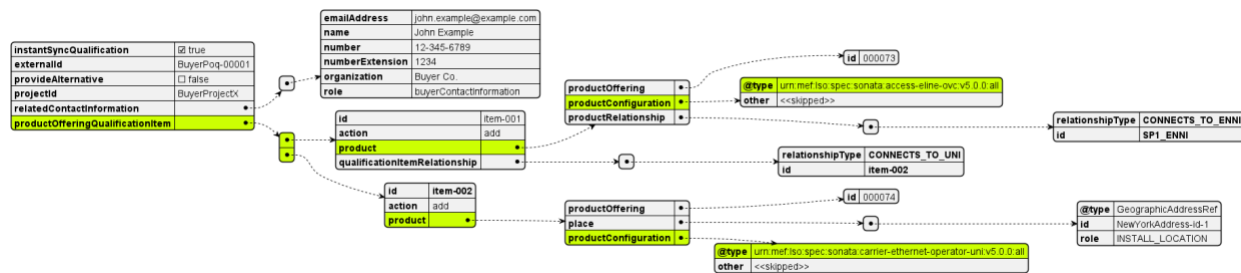
Figure 13 – UC2: POQ Request, envelope part

The sequence diagram below (Figure 14) shows a set of logical steps of building the POQ request:
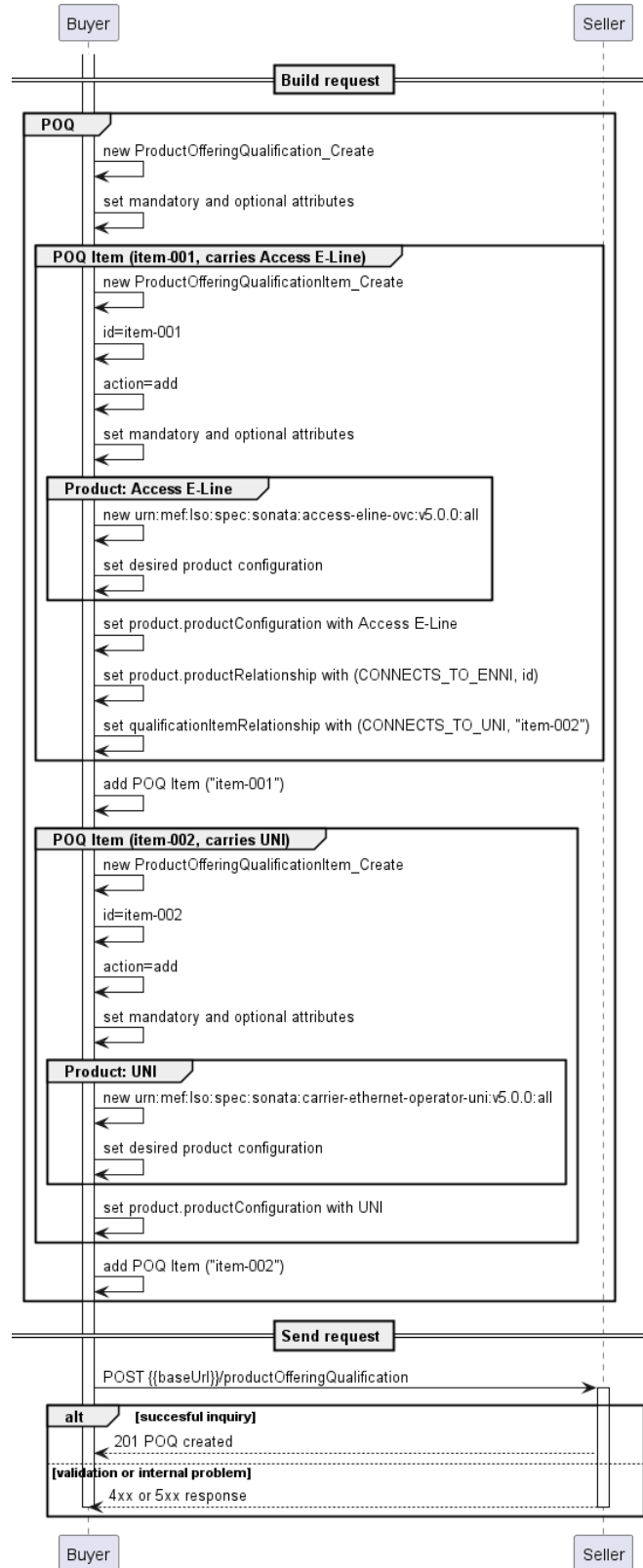
Figure 14 – UC2: POQ request building steps

The products' integration with the API is covered. Let's go to some details of the products' configuration. The setup of the Use Case 2 is presented in Figure 15.
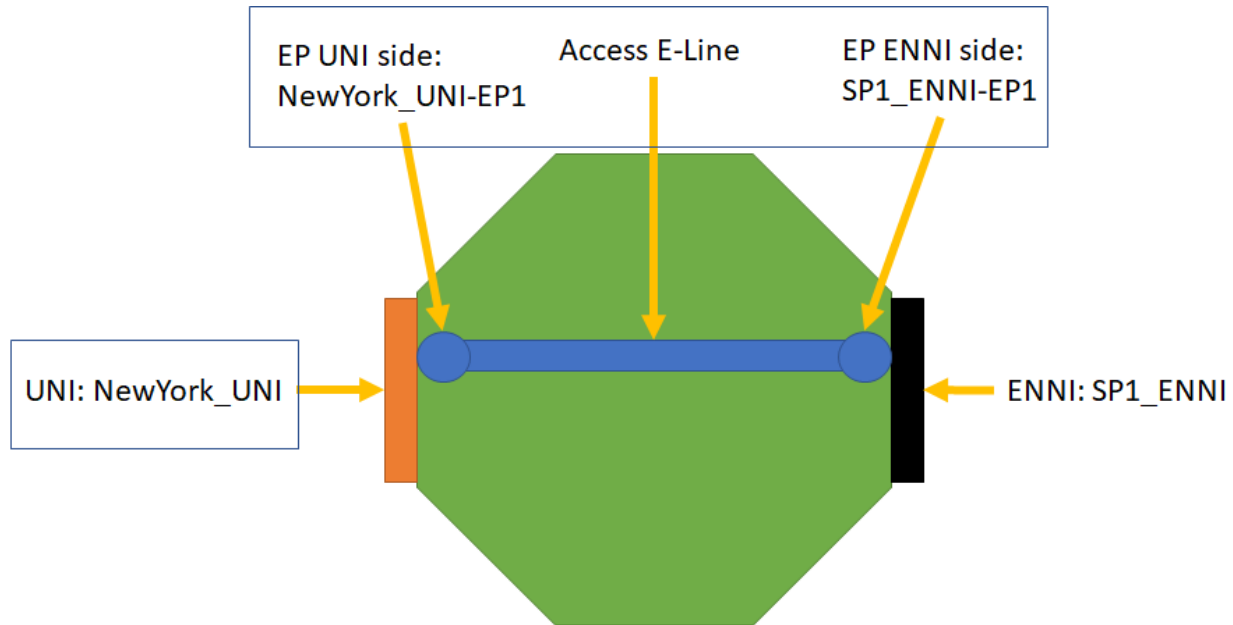
Figure 15 – UC2: Setup Diagram

This setup involves:
- Creation of the UNI
  - place: New York (Address id acquired in Use Case 1)
  - id="NewYork_UNI"
- Creation of the Access E-Line, including:
  - configuration of a new UNI Endpoint with id="NewYork_UNI-EP1", at the UNI with id="NewYork_UNI", which is also created within the same request.
  - configuration of a new ENNI Endpoint with id="SP1_ENNI-EP1", at the already existing ENNI with id="SP1_ENNI"

The diagram aggregates the scope of a particular product configuration into rectangles. This is to stress that the UNI and ENNI endpoints are parts of the Access E-Line configuration. They are not individual orderable products (this is the case in point-to-point connections).

The instance diagram for the whole Access E-Line (AEL) configuration is too big to be presented as a whole so it is split and presented in parts. Figure 16 shows the basic AEL attributes. This diagram as attached to Figure 13 as the node with "@type=urn:mef:lso:spec:sonata:access-eline-ovc:v5.0.0:all". The attributes that are skipped on this level are marked with a "<<skipped>>" label and will be presented on the next diagrams.
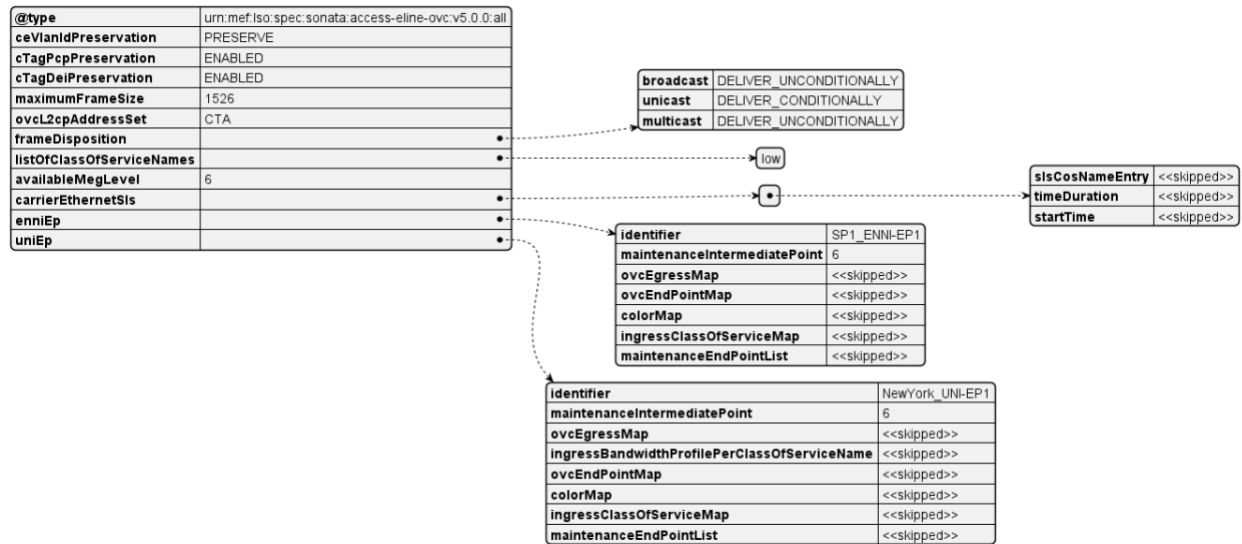
Figure 16 – UC2: Access E-Line, basic attributes

The structures defining the "carrierEthernetSls", "enniEp", and the "uniEp" are complex and presented in the following figures:
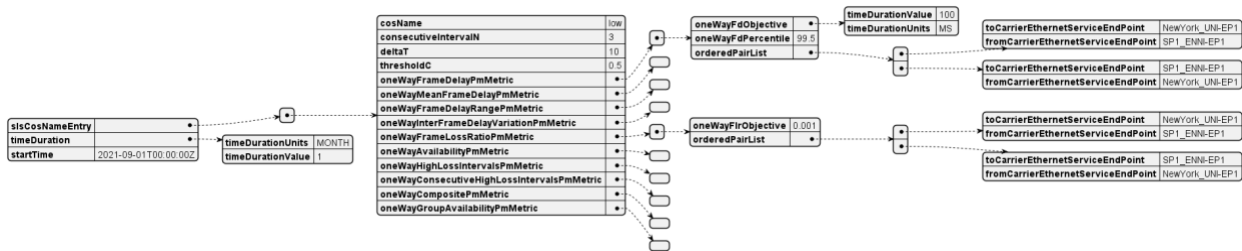

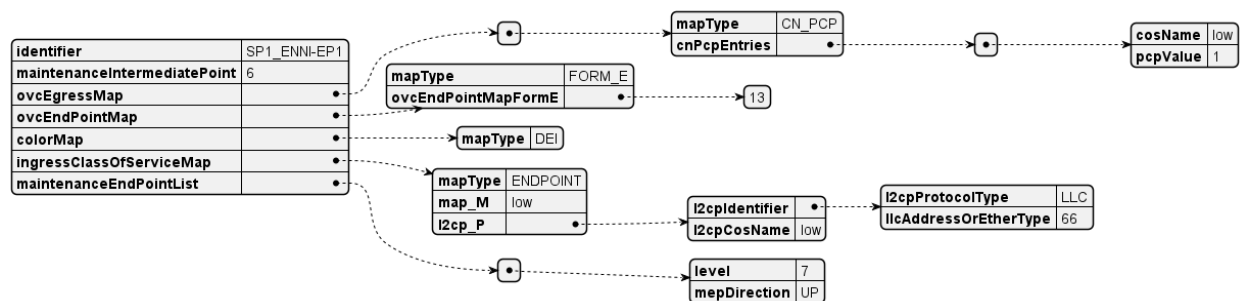**Figure 17 – UC2: Access E-Line, Carrier Ethernet SLS**
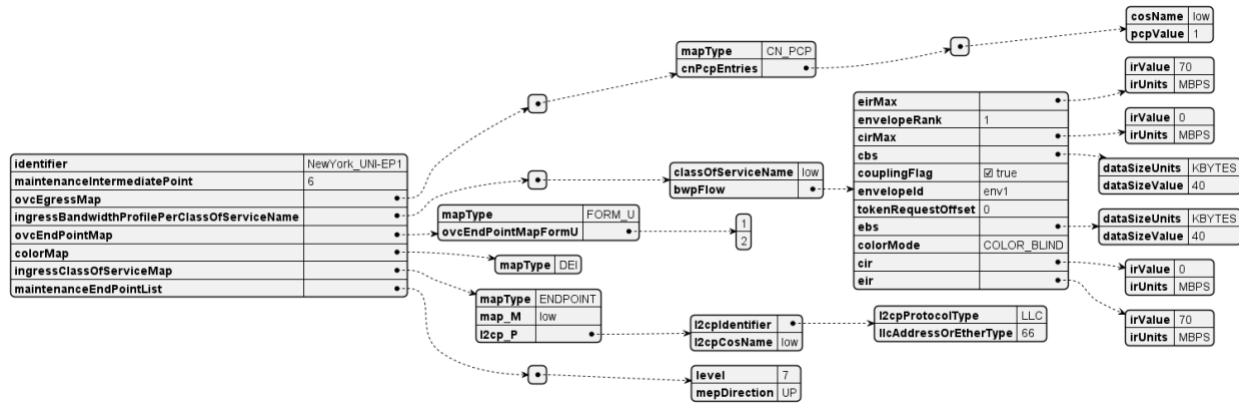

Figure 18 – UC2: Access E-Line, ENNI Endpoint

Figure 19 – UC2: Access E-Line, UNI Endpoint

The last figure in this use case presents the UNI product configuration. There are two envelopes configured in advance. The second will be used in Use Case 3 so that the UNI does not need to be updated.
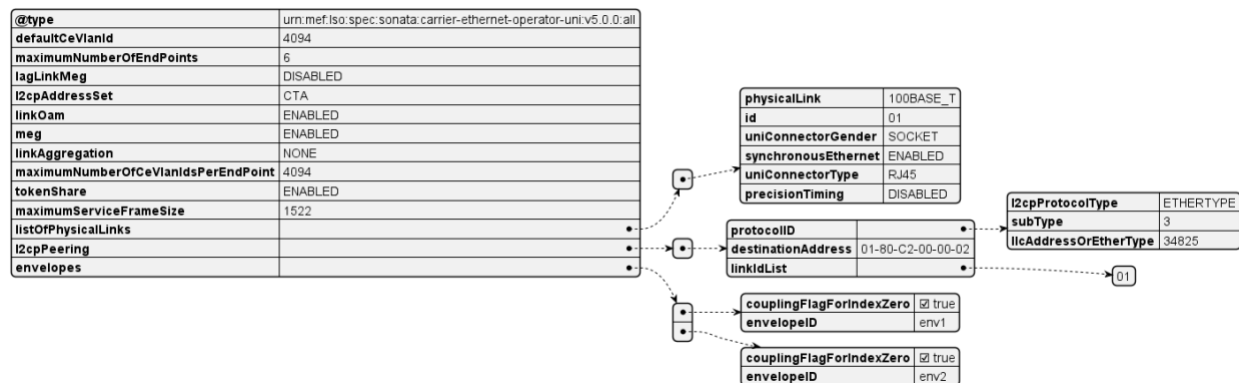
Figure 20 – UC2: UNI

POQ Request example:
```
{
  "instantSyncQualification": true,
  "externalId": "BuyerPoq-00001",
  "provideAlternative": false,
  "projectId": "BuyerProjectX",
  "relatedContactInformation": [
    {
      "emailAddress": "john.example@buyer.com",
      "name": "John Example",
      "number": "12-345-6789",
      "numberExtension": "1234",
      "organization": "Buyer Co.",
      "role": "buyerContactInformation"
```

```
      }
    ],
    "productOfferingQualificationItem": [
      {
        "id": "item-001",
        "action": "add",
        "product": {
          "productOffering": {
            "id": "000073"
          },
          "productConfiguration": {
            "@type": "urn:mef:lso:spec:sonata:access-eline-ovc:v5.0.0:all",
            "ceVlanIdPreservation": "PRESERVE",
            "cTagPcpPreservation": "ENABLED",
            "cTagDeiPreservation": "ENABLED",
            "maximumFrameSize": 1526,
            "ovcL2cpAddressSet": "CTA",
            "frameDisposition": {
              "broadcast": "UNCONDITIONAL",
              "unicast": "CONDITIONAL",
              "multicast": "UNCONDITIONAL"
            },
            "listOfClassOfServiceNames": ["low"],
            "availableMegLevel": "6",
            "carrierEthernetSls": [
              {
                "slsCosNameEntry": [
                  {
                    "cosName": "low",
                    "oneWayFrameDelayPmMetric": [
                      {
                        "oneWayFdObjective": {
                          "timeDurationValue": 100,
                          "timeDurationUnits": "MS"
                        },
                        "oneWayFdPercentile": 99.5,
                        "orderedPairList": [
                          {
                            "toCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1",
                            "fromCarrierEthernetServiceEndPoint": "SP1_ENNI-EP1"
                          },
                          {
                            "toCarrierEthernetServiceEndPoint": "SP1_ENNI-EP1",
                            "fromCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1"
                          }
                        ]
```

```
        }
      ],
      "oneWayFrameLossRatioPmMetric": [
        {
          "oneWayFlrObjective": 0.001,
          "orderedPairList": [
            {
              "toCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1",
              "fromCarrierEthernetServiceEndPoint": "SP1_ENNI-EP1"
            },
            {
              "toCarrierEthernetServiceEndPoint": "SP1_ENNI-EP1",
              "fromCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1"
            }
          ]
        }
      ]
    }
  ],
  "timeDuration": {
    "timeDurationUnits": "MONTH",
    "timeDurationValue": 1
  },
  "startTime": "2021-09-01T00:00:00Z"                }
  }
],
"enniEp": {
  "identifier": "SP1_ENNI-EP1",
  "maintenanceIntermediatePoint": "6",
  "ovcEgressMap": [
    {
      "mapType": "CN_PCP",
      "cnPcpEntries": [
        {
          "cosName": "low",
          "pcpValue": "1"
        }
      ]
    }
  ],
  "ovcEndPointMap": {
    "mapType": "FORM_E",
    "ovcEndPointMapFormE": [13]
  },
  "colorMap": {
    "mapType": "DEI"
```

```
        },
        "ingressClassOfServiceMap": {
          "mapType": "ENDPOINT",
          "map_M": "low",
          "l2cp_P": {}
        },
        "maintenanceEndPointList": [
          {
            "level": 7,
            "mepDirection": "UP"
          }
        ]
      },
      "uniEp": {
        "identifier": "NewYork_UNI-EP1",
        "maintenanceIntermediatePoint": "6",
        "ovcEgressMap": [
          {
            "mapType": "CN_PCP",
            "cnPcpEntries": [
              {
                "cosName": "low",
                "pcpValue": "1"
              }
            ]
          }
        ],
        "ingressBandwidthProfilePerClassOfServiceName": [
          {
            "classOfServiceName": "low",
            "bwpFlow": {
              "eirMax": {
                "irValue": 70,
                "irUnits": "MBPS"
              },
              "envelopeRank": 1,
              "cirMax": {
                "irValue": 0,
                "irUnits": "MBPS"
              },
              "cbs": {
                "dataSizeUnits": "KBYTES",
                "dataSizeValue": 40
              },
              "couplingFlag": true,
              "envelopeId": "env1",
```

```
              "tokenRequestOffset": 0,
              "ebs": {
                "dataSizeUnits": "KBYTES",
                "dataSizeValue": 40
              },
              "colorMode": "COLOR_BLIND",
              "cir": {
                "irValue": 0,
                "irUnits": "MBPS"
              },
              "eir": {
                "irValue": 70,
                "irUnits": "MBPS"
              }
            }
          }
        ],
        "ovcEndPointMap": {
          "mapType": "FORM_U",
          "ovcEndPointMapFormU": [1, 2]
        },
        "colorMap": {
          "mapType": "DEI"
        },
        "ingressClassOfServiceMap": {
          "mapType": "ENDPOINT",
          "map_M": "low",
          "l2cp_P": {
            "l2cpIdentifier": {
              "l2cpProtocolType": "LLC",
              "llcAddressOrEtherType": "66"
            },
            "l2cpCosName": "low"
          }
        },
        "maintenanceEndPointList": [
          {
            "level": 7,
            "mepDirection": "UP"
          }
        ]
      }
    },
    "productRelationship": [
      {
        "relationshipType": "CONNECTS_TO_ENNI",
```

```
              "id": "SP1_ENNI"
            }
          ]
        },
        "qualificationItemRelationship": [
          {
            "relationshipType": "CONNECTS_TO_UNI",
            "id": "item-002"
          }
        ]
      },
      {
        "id": "item-002",
        "action": "add",
        "product": {
          "productOffering": {
            "id": "000074"
          },
          "place": [
            {
              "@type": "GeographicAddressRef",
              "id": "NewYorkAddress-id-1",
              "role": "INSTALL_LOCATION"
            }
          ],
          "productConfiguration": {
            "@type": "urn:mef:lso:spec:sonata:carrier-ethernet-operator-
uni:v5.0.0:all",
            "defaultCeVlanId": 4094,
            "maximumNumberOfEndPoints": 6,
            "lagLinkMeg": "DISABLED",
            "l2cpAddressSet": "CTA",
            "linkOam": "ENABLED",
            "meg": "ENABLED",
            "linkAggregation": "NONE",
            "maximumNumberOfCeVlanIdsPerEndPoint": 4094,
            "tokenShare": "ENABLED",
            "maximumServiceFrameSize": 1522,
            "listOfPhysicalLinks": [
              {
                "physicalLink": "100BASE_T",
                "id": "01",
                "uniConnectorGender": "SOCKET",
                "synchronousEthernet": "ENABLED",
                "uniConnectorType": "SC",
                "precisionTiming": "DISABLED"
```

```
                    }
                ],
                "l2cpPeering": [
                    {
                        "protocolID": {
                            "l2cpProtocolType": "ETHERTYPE",
                            "subType": 3,
                            "llcAddressOrEtherType": 34825
                        },
                        "destinationAddress": "01-80-C2-00-00-02",
                        "linkIdList": [
                            "01"
                        ]
                    }
                ],
                "envelopes": [
                    {
                        "couplingFlagForIndexZero": true,
                        "envelopeID": "env1"
                    },
                    {
                        "couplingFlagForIndexZero": true,
                        "envelopeID": "env2"
                    }
                ]
            }
        }
    }
  ]
}
```

A.3.3   Use Case 3: POQ - new Access E-Line, existing UNI, low+high class of service

This use case presents the same step (POQ) as Use Case 2 but with two differences:
- an existing UNI is being used, and
- a more complex configuration of an Access E-Line is presented.

The setup of the Use Case 3 is presented in Figure 21:
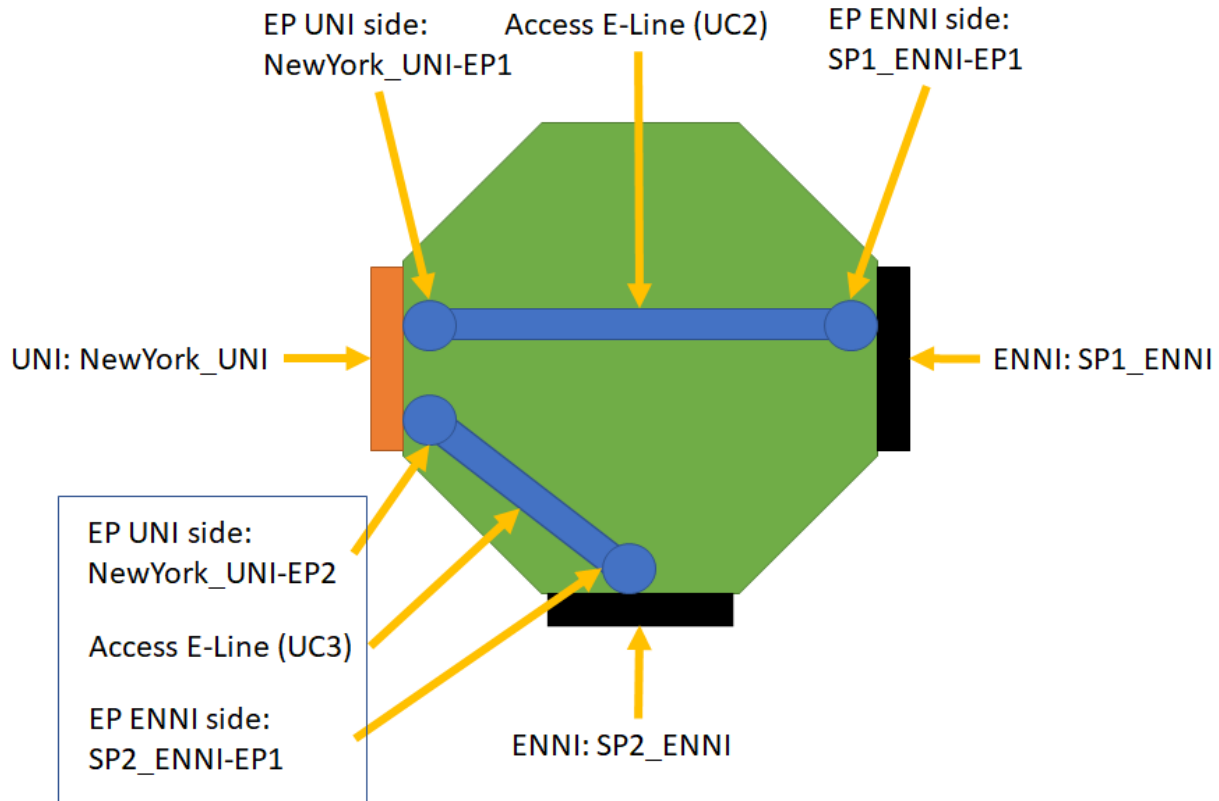
Figure 21 – UC3: Setup diagram

This setup involves:

- Creation of the new Access E-Line, including:

    o configuration of a new UNI Endpoint with id="NewYork_UNI-EP2", at the already existing UNI with id="NewYork_UNI", the one that was created in Use Case 2 (assuming it was successfully ordered)

    o configuration of a new ENNI Endpoint with id="SP2_ENNI-EP1", at the already existing ENNI with id="SP2_ENNI" (other than in use case 1)

This results in POQ Request having only one POQ Item, as now both ENNI and UNI are existing and referenced with the use of "productOfferingQualificationItem.product.productRelationship". This is presented on truncated POQ request example in Figure 22:
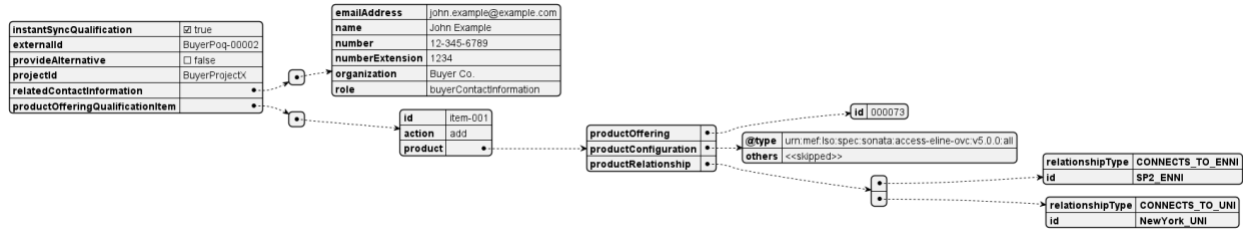
Figure 22 – UC3: Access E-Line relationships

The configuration of the UNI endpoint introduces a configuration with two classes of service: low and high. The difference is first noticed at the root Access E-Line type configuration by having a list in the "listOfClassOfServiceNames" attribute:
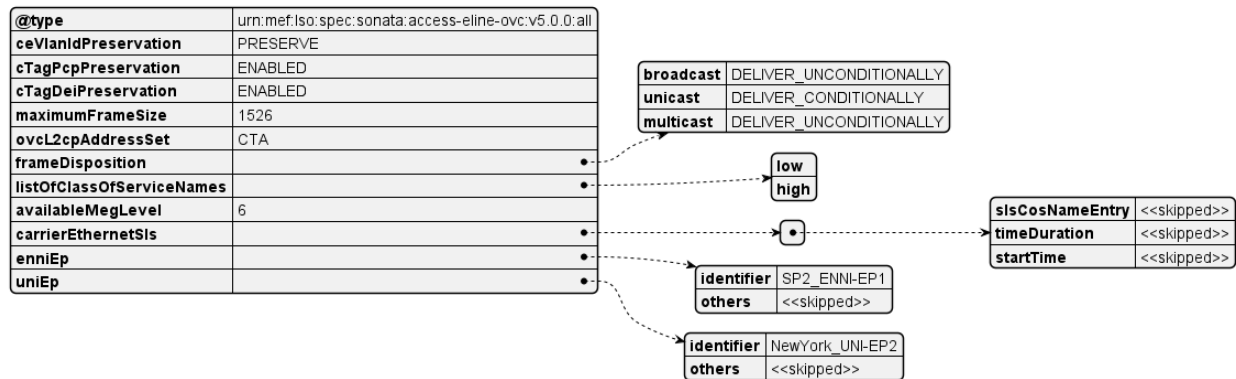
Figure 23 – UC3: Access E-Line, basic attributes

Next, the "carrierEthernetSls" provides the Service Level Specification by defining the performance metrics per class of service. Notice the different metrics and values per low and high ones.
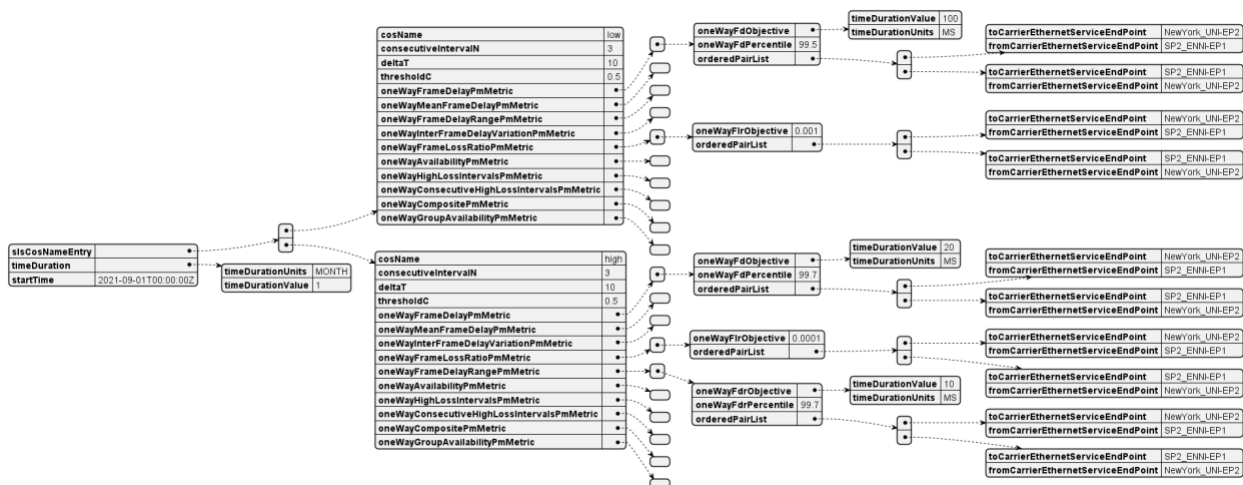
Figure 24 – UC3: Access E-Line, Carrier Ethernet SLS
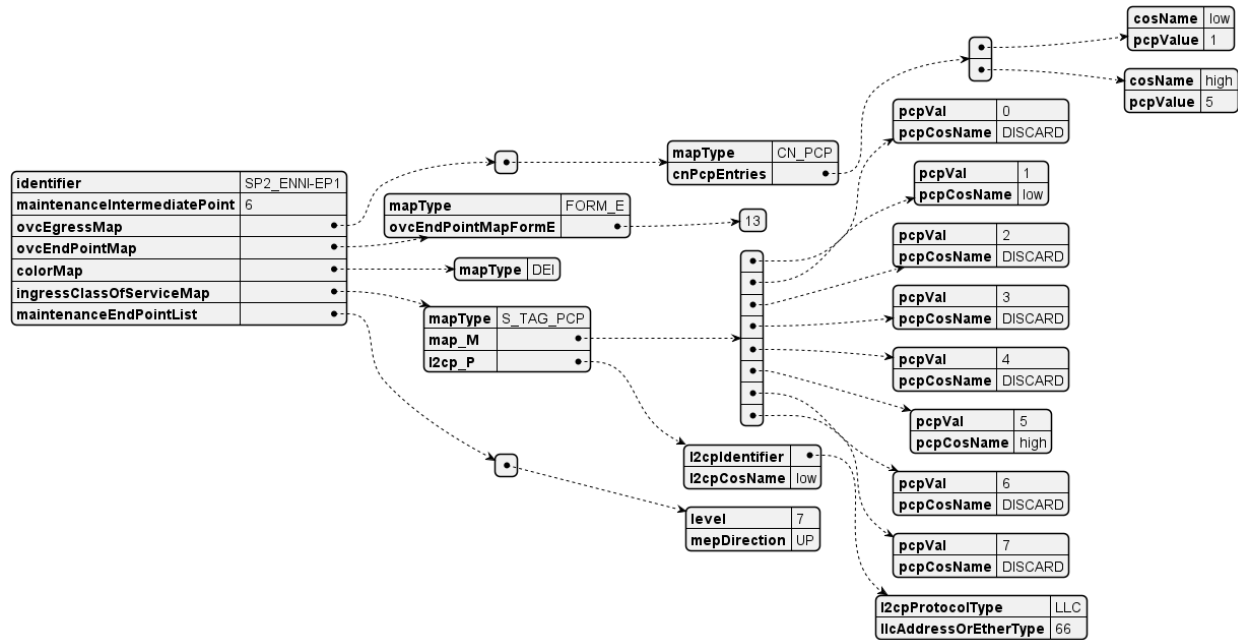
Figure 25 – UC3: Access E-Line, ENNI Endpoint
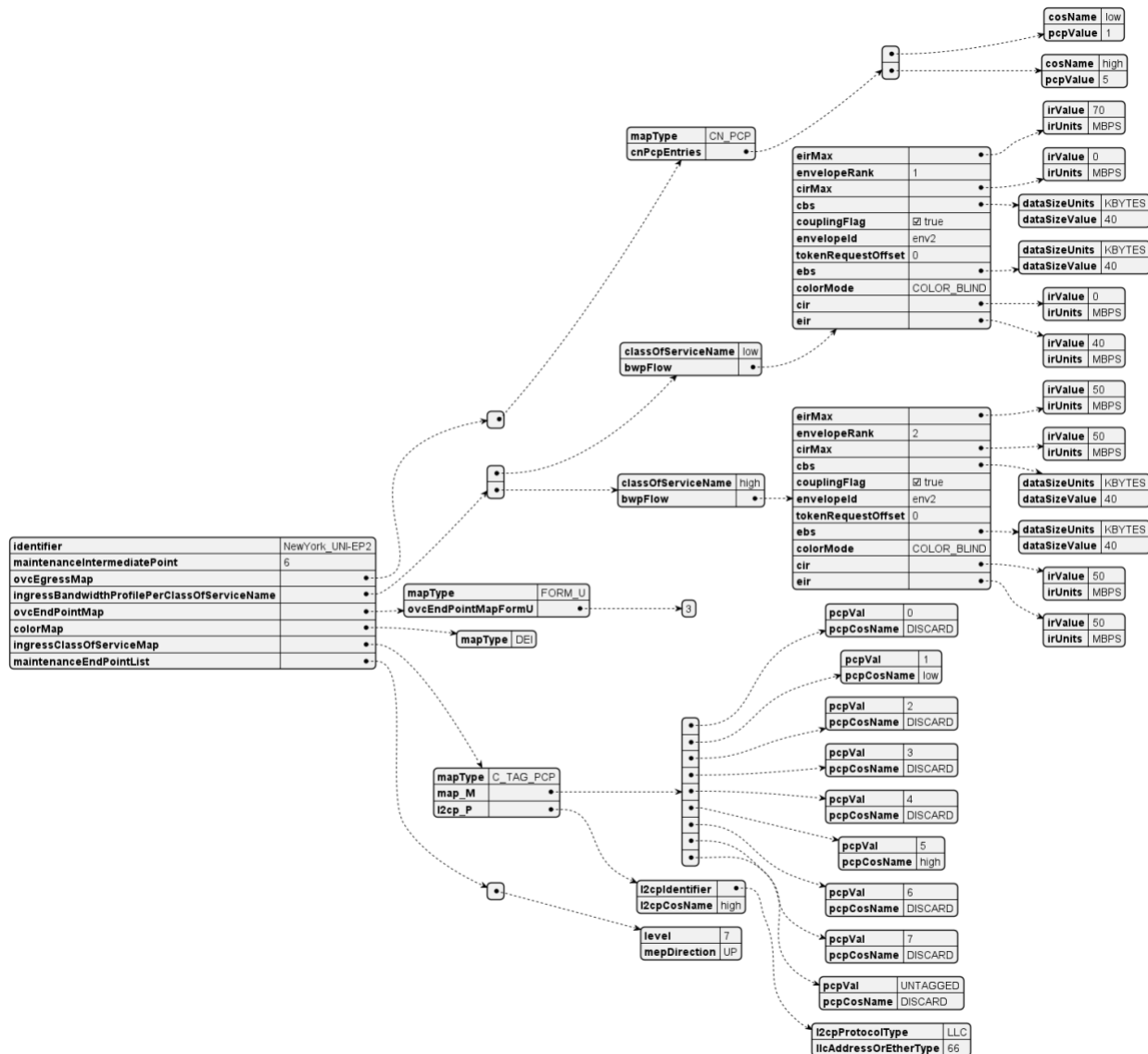
Figure 26 – UC3: Access E-Line, UNI Endpoint

The bandwidth profiles defined for both classes of service share the same envelope "env2" which means they share the same bandwidth "pipe". The "high" profile defines the "cirMax" and "eirMax" on the same level as corresponding "cir" and "eir" values, which is 50 MBPS. This means that the 50 MBPS bandwidth is guaranteed for this profile. The "low" profile defines the "cir" and "cirMax" on the level of 0 MBPS which means no bandwidth is guaranteed. The values of and the "eir" of 40 MBPS that a maximum of 40 MBPS can be used when free and, because "eirMax" is set to 70 MBPS, additional 30 MBPS from the "high" CoS reservation can be used, when the traffic is not utilizing the full guaranteed 50 MBPS.

Note that the names of the classes of service – "low" and "high" are used several times across the payload and must match respectively in all of the places.

The request example, as huge and repeating can be found in the attached postman collection.

A.3.4 Use Case 4: Quote

For detailed guidance on how to use the Quote Management API, please refer to MEF 115 [19].

The aim of the Quote step is to allow the Buyer to submit a request to find out how much the installation of an instance of a Product Offering, an update to an existing Product, or a disconnect of an existing Product will cost.

This use case is the next step after use case 2. It asks for a quotation of the installation of the Access E-Line and UNI products, with configuration as described in use case 2 (A.3.2).

The Quote API carries product information exactly the same way as the POQ. The same steps in request building (Figure 14) and rules of referencing existing products or ones in the same request, as described in section A.3.2, apply.

Figure 27 presents a diagram of a Quote request, with product information skipped.
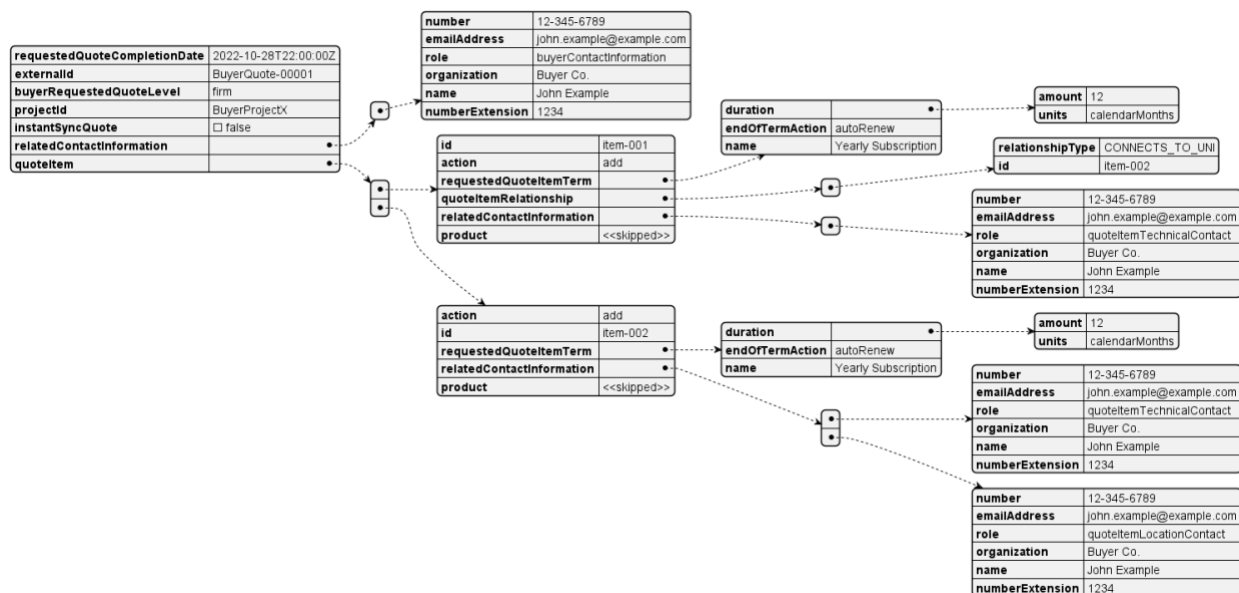
Figure 27 – UC4: Quote Request

The most important attributes to set in the quote request are:

"instantSyncQuote" – to state the preference of receiving an instant (synchronous) response or a deferred (asynchronous) one. In the latter case, the Seller only sends back an acknowledge response and proceeds with the quotation. The Buyer may choose to register for notification or perform a periodical poll.

"requestedQuoteCompletionDate" – If an instant response is not required this specifies the requested response time.

"buyerRequestedQuoteLevel" - 3 different types of quotes are managed:

- **Budgetary**: A Quote that is provided quickly and with very little analysis such that the Buyer can get an idea of how much the requested Product Offering could cost. Any charges specified are subject to change.

- **Firm - Subject to Feasibility Check**: A Quote that is provided to the Buyer based on some, but not a complete, pre-order analysis. At this stage, the Seller may not be willing to perform any further work on the Quote and requests that the Buyer use the Firm – Subject to Feasibility Check Quote to proceed to the Order process. Ordering is possible based on the Firm – Subject to Feasibility Check Quote with some stipulations as to how cost identified during delivery is addressed. The Monthly Recurring Charges specified in the Quote Response are final. Non-Recurring Charges specified in the Quote Response are subject to change and new Non-Recurring Charges may be identified during fulfillment.

- **Firm**: A Quote provided to the Buyer based on complete pre-order analysis. All Monthly Recurring Charges and Non-Recurring Charges specified on a Firm Quote are committed. A Firm Quote may expire at some date specified by the Seller.

"requestedQuoteItemTerm" – to specify the term (also known as commitment)

In the response, the Seller confirms (most likely) the "quoteLevel", "quoteItemTerm" and provides a price per each quote item. An example of price specification is shown below:

```
"quoteItemPrice": [
 {
   "name": "Monthly Plan 25",
   "priceType": "recurring",
   "recurringChargePeriod": "month",
   "price": {
     "taxRate": 16,
     "dutyFreeAmount": {
       "unit": "EUR",
       "value": 25,
     },
     "taxIncludedAmount": {
       "unit": "EUR",
       "value": 29,
     },
   },
 }
],
```

Note: The Seller may require the Buyer to perform POQ prior to sending a Quote request.

A.3.5 Use Case 5: Product Order

Product Order allows the Buyer to request the Seller to initiate and complete the fulfillment process of an installation of a Product Offering, an update to an existing Product, or a disconnect of an existing Product at the address defined by the Buyer.

This use case is the next step after use case 4. It places an order for the installation of the Access E-Line and UNI products, which were qualified and quoted in use cases 2 and 4.

The Order API carries product information exactly the same way as the POQ and Quote. The same steps in request building (Figure 14) and rules of referencing existing products or ones in the same request, as described in section A.3.2, apply.

An example Product Order request can be found in the postman collection. Figure 28 presents it with product information skipped for readability.
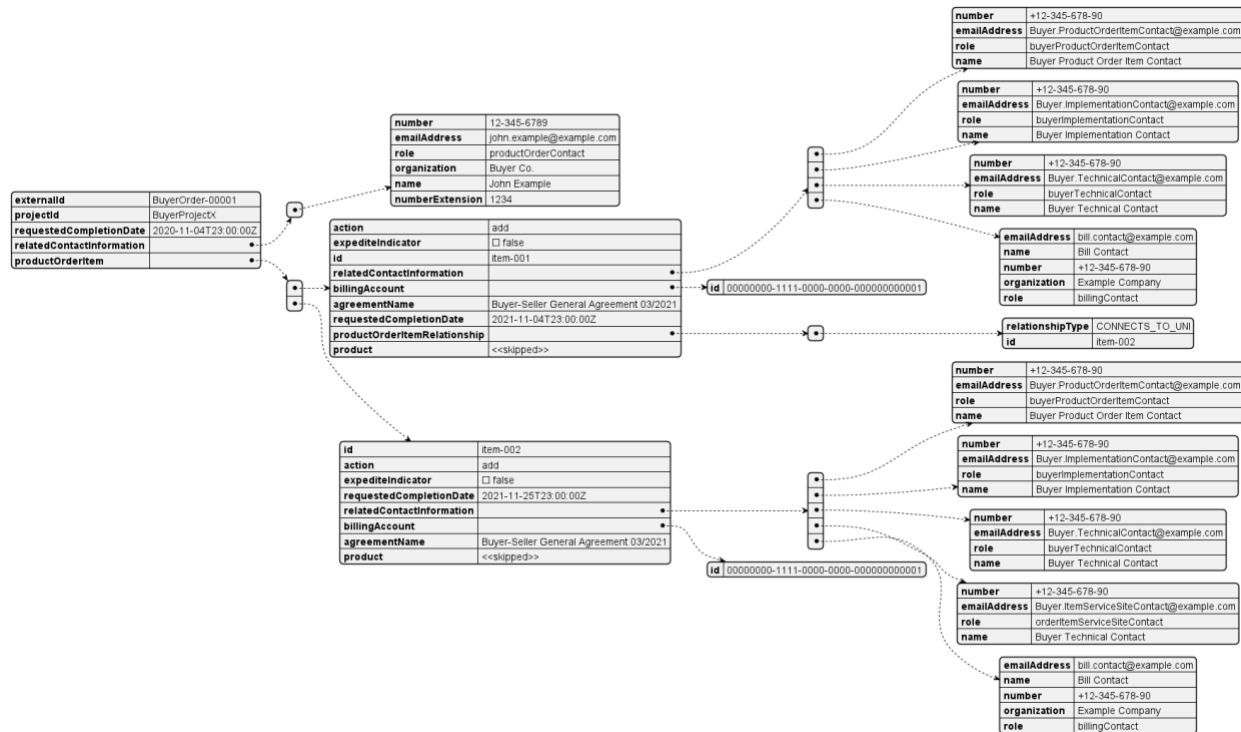
Figure 28 – UC5: Product Order request

Again, there are a few attributes to be set by the Seller in the request like "requestedCompletionDate", "expediteIndicator" or "billingAccount" together with required contact information.

The Seller responds with an acknowledge confirmation and then starts processing the order. The order fulfillment process is longer than a simple request-response one of the previous steps (POQ, Quote) and the state machine is more complex. The process may also be more interactive due to charge negotiation, possible request updates, etc.

Product order API offers much more use cases like updating, expediting, or canceling an order request and additional charge negotiation. For detailed guidance on how to use the Product Order Management API, please refer to MEF 123.

A.4 action: modify

The mechanism of building a modification request for both envelope and payload for all steps are practically the same as for the create request.

The differences are in the following common rules (POQ, Quote, Order):

- "item.action" must be set to "modify"

- "item.product.id" of the product to be updated must be provided

- "product.productConfiguration" must contain all desired configuration (not only the updated values)

- "product.productOffering" must not be changed

- The Access E-Line Product does not allow "product.productRelationship", and "product.place" to be changed.

-

A.4.1 Use Case 6: POQ: Bandwidth change

Use cases 6, 7, and 8 present POQ, Quote, and Order for an Access E-Line bandwidth change. The change is made only for the attributes of the Access E-Line product, so requests contain only one item (UNI product is not modified). The change is made by updating the "eir.irValue" and "eirMax.irValue" from 70 to 100.

Note that since there is no accompanying item with the UNI, the relationship information "CONNECTS_TO_UNI" must be provided with the use of "product.productRelationship" attribute to point to the existing UNI instance with "id"="NewYork_UNI"

The diagram below shows a POQ request for modification, highlighting the changes compared to the creation request.
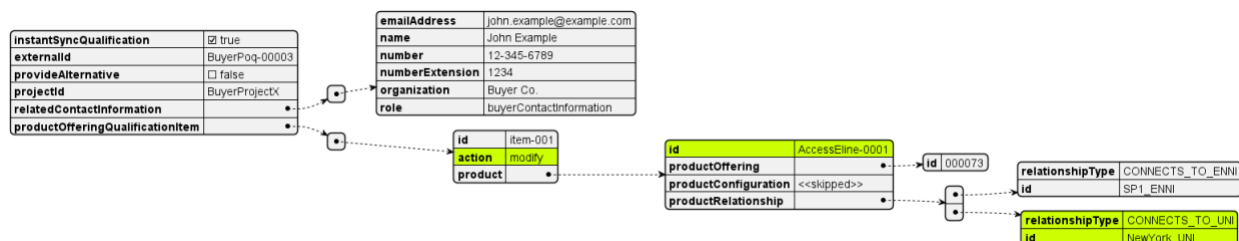

Figure 29 – UC6: POQ modify request

Figure 30 underlines the necessity of providing full product configuration for the modify action. The highlighted attributes are the only ones that are requested to be changed.
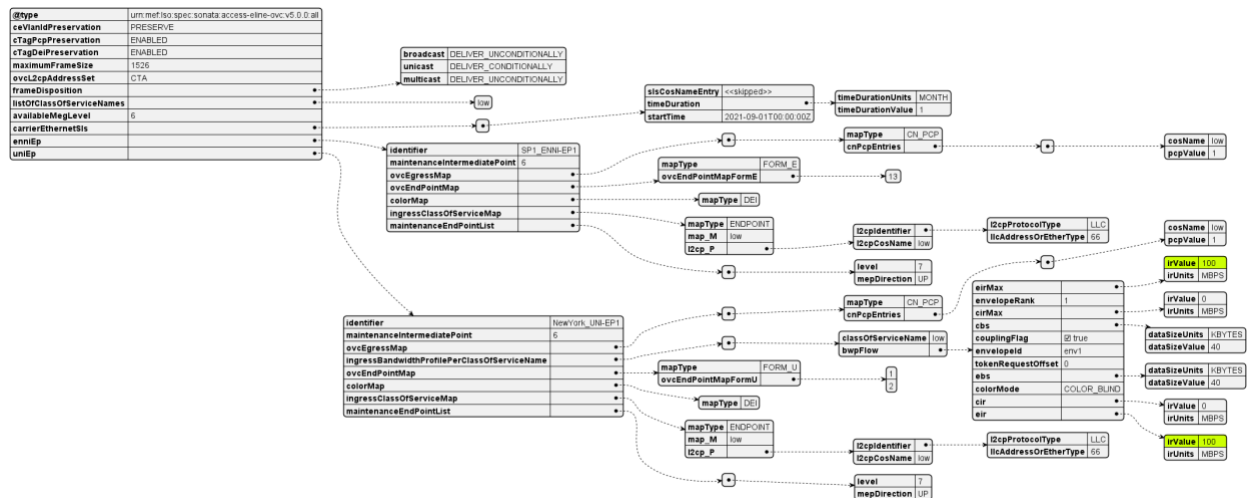

Figure 30 – UC6: Access E-Line modified attributes

A full example request can be found in the attached postman collection.

A.4.2 Use Case 7: Quote: Bandwidth change

As the details of the product modification are already described in the previous chapter, this use case will only highlight the changes in the quote request, compared to the create request (Figure 31)
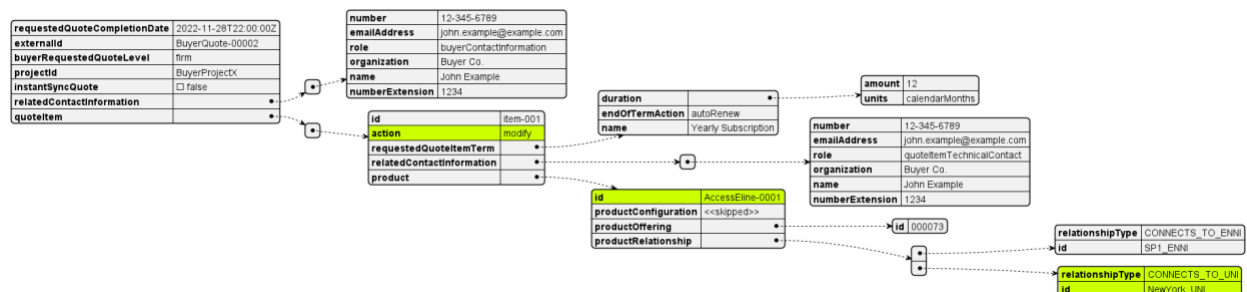
Figure 31 – UC7: Quote modification request

A full example request can be found in the attached postman collection.

A.4.3 Use Case 8: Product Order: Bandwidth change

All rules were described in the two above chapters. Figure 32 presents the Order request with highlighted changes:
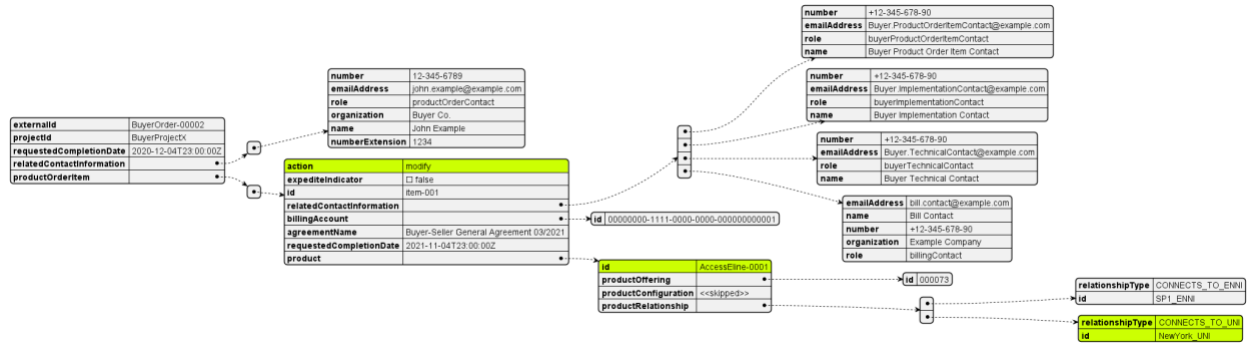
Figure 32 – UC8: Order modification request

A full example request can be found in the attached postman collection.

A.4.4   Use Case 9: Product Order – VLAN change at the UNI

In this case, an order to enhance the list of VLAN IDs are mapped to the UNI End Point is enhanced from [1, 2] to [1, 2, 3]. This is done with the order request as above, but with a slightly modified product payload, which is presented in Figure 33:
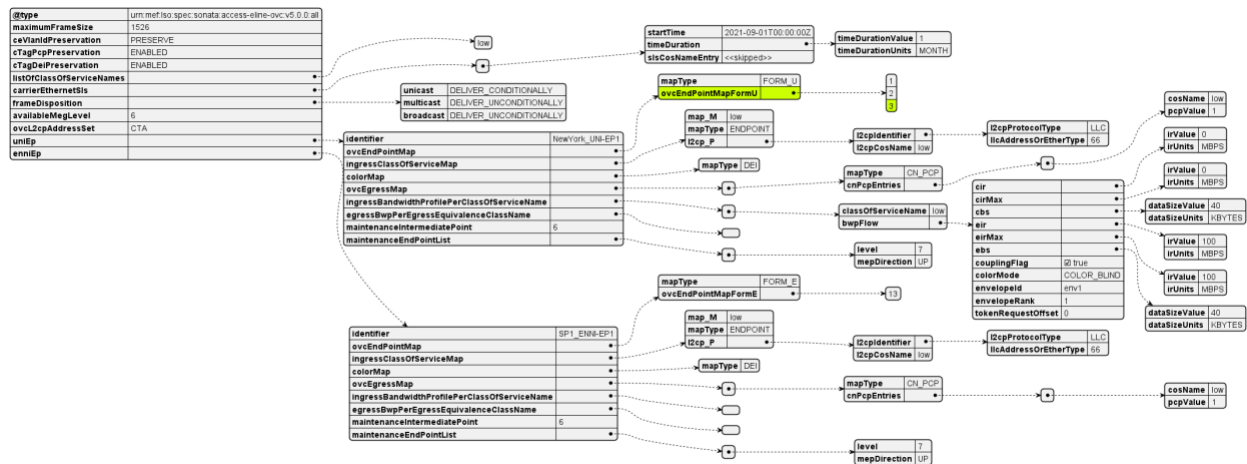

Figure 33 – UC9: Order modification request

A full example request can be found in the attached postman collection.

A.5   action: delete

Delete requests are for all steps are very straightforward, as they only carry the product "id".

Following common rules apply for disconnect operation:

- "item.action" must be set to "delete"

- "item.product.id" of the product to be deleted must be provided

---

- "product.productConfiguration" must not be provided

- no other item attribute may be provided (except for optional "billingAccount" in Order)

A.5.1   Use Case 10: Product Order – delete both Access E-Line and UNI

Deletion of both Access E-Line and UNI products can be ordered with a request that is presented in Figure 34:
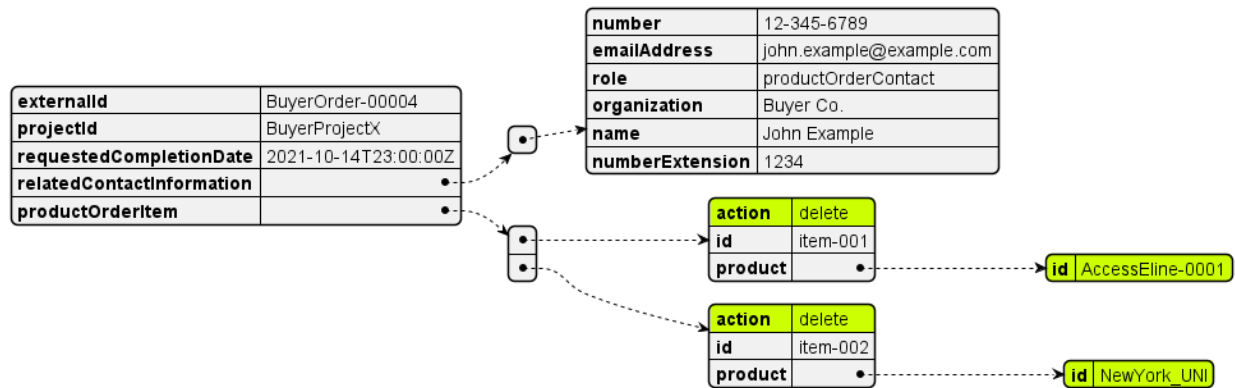

Figure 34 – UC10: Order deletion request

Note: A disconnect request may result in additional charges (if not quoted earlier).

JSON representation of this request:

```
{
  "externalId": "BuyerOrder-00004",
  "projectId": "BuyerProjectX",
  "requestedCompletionDate": "2021-10-14T23:00:00Z",
  "relatedContactInformation": [
    {
      "number": "12-345-6789",
      "emailAddress": "john.example@buyer.com",
      "role": "productOrderContact",
      "organization": "Buyer Co.",
      "name": "John Example",
      "numberExtension": "1234"
    }
  ],
  "productOrderItem": [
    {
      "action": "delete",
      "id": "item-001",
      "product": {
        "id": "AccessEline-0001"
      }
    }
```

```
    },
    {
      "action": "delete",
      "id": "item-002",
      "product": {
        "id": "NewYork_UNI"
      }
    }
  ]
}
```

A.5.2 Use Case 11: Move Access E-Line to a different Location

The case of moving the office to another building cannot be realized by a single update of the "place" attribute of the UNI product. One of the points in Chapter 9 states:

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

Nor it can be realized by updating Access E-Line's product reference to another UNI. Chapter 11:

Changing the UNI and ENNI Reference or the UNI Location is not supported for an Access E-Line Ser-vice. The value included in a Change request must be identical to the value in the Inventory.

So, the argument is both business and technical. In order to realize this use case, the following requests must be performed:

1. Creation of new UNI at the new location

2. Creation of a new Access E-Line

3. Deletion of an old Access E-Line

4. Deletion if an old UNI (optionally, if not used by other connections)

Step 1 as potentially requiring physical installation should be performed earlier to prepare for a switchover. Steps 2 and 3 should be coordinated to assure minimal downtime.

This use case as being built upon already described steps is not part of the attached postman collection.

Appendix B   Acknowledgements

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

- Manfred **ARNDT**
- David **BALL**
- Mike **BENCHECK**
- Michał **ŁĄCZYŃSKI**
- Jack **PUGACZEWSKI**
- Patrick **ROOSEN**
- Fahim **SABIR**
- Larry **SAMBERG**