1

2

3

4 # Working Draft

5 # MEF W139 v0.2

6

7 # Internet Access Product Schemas and
8 # Developer Guide

9

10 # December 2022

11

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

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

21 Disclaimer

## Table of Contents

209

# List of Figures

242

# List of Tables

360

# 1    List of Contributing Members

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

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

## 2  Abstract

The MEF Standard consisting of this schema guide and its associated software artifacts (JSON Schemas) defines and describes the product-specific information used in LSO Cantata and LSO Sonata APIs for a set of Business Functions - specifically, Product Offering Qualification, Quote, Product Ordering, and Product Inventory - for Basic and Advanced Internet Access product. The document starts with an overview of LSO Cantata, LSO Sonata, and Internet Access services. It then provides a basic information model for the MEF Internet Access 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 Internet Access Data Model and the schemas provided that embody the Data Model. MEF Services are described by a set of Service Attributes. Each Service Attribute describes an aspect of the service that is agreed upon between the provider and the user of the service. The documents that describe the Service Attributes for Internet Access Services are MEF 61.1 **Błąd! Nie można odnaleźć źródła odwołania.** and MEF 61.1.1 [9]. The Basic and Advanced services are specified in MEF 69.1 [10] based on the Service Attributes defined in MEF 61.1 **Błąd! Nie można odnaleźć źródła odwołania.**.

MEF 61.1 **Błąd! Nie można odnaleźć źródła odwołania.** and MEF 61.1.1 [9] specify Service A ttributes to describe the various components that compose a Basic Internet Access service and Advanced Internet Access. This document defines a data model that includes these Service Attributes respectively and also lists the Service Attributes that are not included in the data model or are present in modified form, and the reason why each is not included or modified.

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:

productSchema/ip/

- common/ipCommon.yaml

- common/ipCsl.yaml

- internetAccess/advancedInternetAccessIpvc/advancedInternetAccessIpvc.yaml

- internetAccess/basicInternetAccess/basicInternetAccess.yaml

- internetAccess/exclusiveAdvancedInternetAccess/exclusiveAdvancedInternetAccess .yaml

- internetAccess/internetAccessCommon/internetAccessCommon.yaml

- ipUni/ethernetUniAccessLinkTrunk.yaml

- ipUni/ipUni.yaml

- ipUni/ipUniAccessLink.yaml

# 3  Terminology and Abbreviations

This section defines the terms used in this document. In many cases, the normative definitions of 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.

In addition, terms defined in the standards referenced below are included in this document by reference and are not repeated in the table below:

- MEF 55.1 Lifecycle Service Orchestration (LSO): Reference Architecture and Framework [6]
- MEF 57.2 Product Order Management Requirements and Use Cases [7]
- MEF 61.1 IP Service Attributes [8]
- MEF 61.1.1 Amendment to MEF 61.1: UNI Access Link Trunks, IP Addresses, and Mean Time to Repair Performance Metric [9]
- MEF 69.1 Subscriber IP Service Definitions [10]
- MEF 78.1 MEF Core Model [11]
- MEF 79 Address, Service Site, and Product Offering Qualification Management, Requirements and Use Cases [12]
- MEF 106 LSO Sonata Access E-Line Product Schemas and Developer Guide [16]

| Term | Definition | Reference |
|---|---|---|
| **IP UNI Access Link** | A UNI Access Link for an IP Service, i.e. a subnetwork corresponding to a distinct IP subnet, that forms part of a UNI.  The subnet might use both IPv4 and IPv6 addressing. | MEF 61.1 [8] |
| **IP User Network Interface** | A UNI at which an IP Service is accessed. | MEF 61.1 [8] |
| **IP Virtual Connection** | An association of two or more IPVC EPs that limits the exchange of IP Packets to IPVC EPs for the IPVC. | MEF 61.1 [8] |
| **IPVC End Point** | A logical entity at a given External Interface to which a distinct subset of IP Packets passing over that External Interface is mapped. | MEF 61.1 [8] |
| **UNI Access Link Trunk** | A construct that encapsulates the details of the Layer 1 and Layer 2 configuration shared by one or more UNI Access Links. | MEF 61.1.1 [9] |

**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 [2] , 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   Introduction

LSO Cantata provides a programmatic interface for establishing (quoting, ordering, etc.) products between Service Provider (Seller) and the Subscriber (Buyer). 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, in this specification Basic or Advanced Internet Access.

Internet Access is a Subscriber IP Service that connects the Subscriber to the Internet. The Service Attributes that are agreed to between the parties are defined in MEF 61.1 **Błąd! Nie można odnaleźć źródła odwołania.** and MEF 61.1.1 [9] The Service definition for Basic and Advanced Internet Access which is, in effect, a set of constraints on the values of the Service Attributes, is provided in MEF 69.1 [10].

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

- IPVC, including exactly one End Point

- IP UNI

- IP UNI Access Link

- IP UNI Access Link Trunk

The document contains the following sections:

- An overview of LSO Cantata and LSO Sonata (section 6)

- An overview of the Internet Access Service (section 7)

- Data Model Design Principles (section 8)

- Order Milestones (section 9)

- An abbreviated Information Model for Internet Access and explanation of the organization of the Service Attributes in MEF 61.1 **Błąd! Nie można odnaleźć źródła odwołania.** and MEF 61.1.1 [9] (section 10)

- Organization of the Data Model for Internet Access (section 11)

- The relationship between the entities in the service (section 12)

- The detailed comparison of Service Attributes of Basic and Advanced Products with a list of ones that are not included in the Data Model (section 13)

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

468    • Section 14 contains the details of all of the Service Attributes

# 6    Overview of LSO Cantata and LSO Sonata

MEF 55.1 [6] describes the Reference Architecture for Lifecycle Service Orchestration (LSO) of MEF-defined services. MEF 55.1 defines seven LSO Interface Reference Points (see Figure 1) that are abstract interconnection points between different entities—either within the Service Provider domain (intra-domain) or between Service Provider and other business entities (inter-domain). One of these LSO Reference Points is LSO Cantata which defines the abstract interconnection point between a Subscriber (Buyer) and a Service Provider (Seller) and another is LSO Sonata which defines the abstract interconnection point between a Service Provider (Buyer) and an Operator (Seller). It is at these Interface Reference Points – LSO Cantata and LSO Sonata – that the Buyer and the Seller interact to orchestrate business transactions for the different Business Functions. Inter-provider Business Functions include Address Qualification, Site Query, Product Offering Qualification, Quote, Product Ordering, Product Inventory, Trouble Ticketing, and Billing In the context of this document, the following 4 business functions are relevant as ones exchanging product information:

- Product Offering Qualification, MEF 79 [12]

- Quote, MEF 80 [13]

- Product Ordering, MEF 57.2 [7]

- Product Inventory, MEF 81 [14]
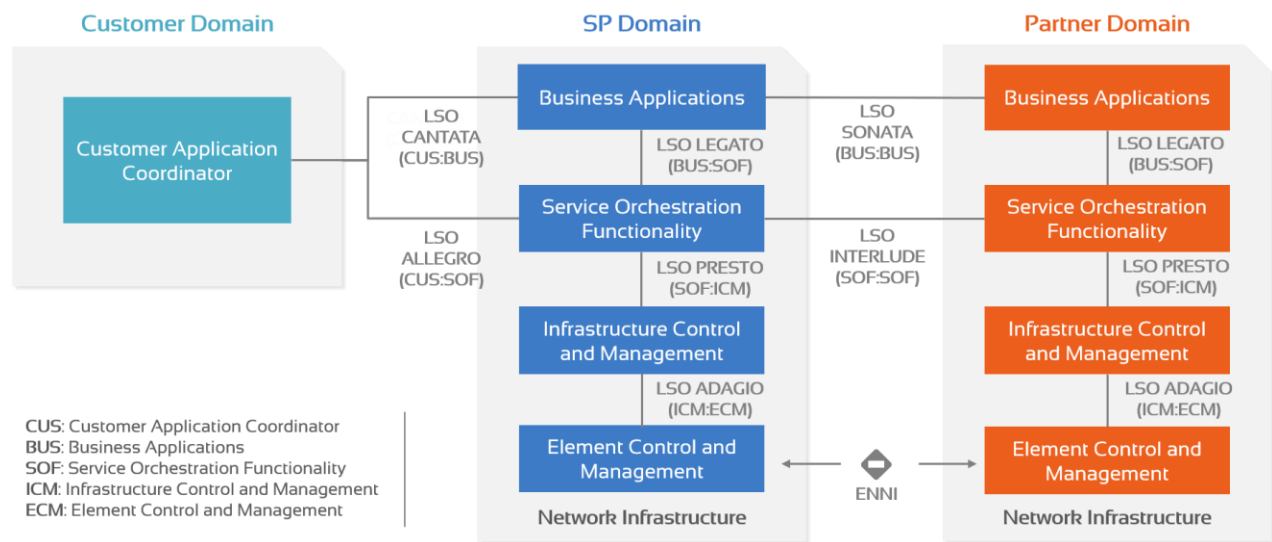


**Figure 1 LSO Cantata and LSO Sonata Reference Diagram**

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

492 SDK and LSO Sonata SDK. The APIs are standardized by following API and Developer Guide
493 documents:

494 • Product Offering Qualification, MEF 87 [15]

495 • Quote, MEF 115 [17]

496 • Product Ordering, MEF 123 [19]

497 • Product Inventory, MEF 116 [18]

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

500



Function Specific
(e.g., POQ, Quote, Order, Inventory)

Product/Service Agnostic

Product/Service Specific
(e.g. Access E-Line, EPL, EVPLAN, etc.)

*Focus of this document*
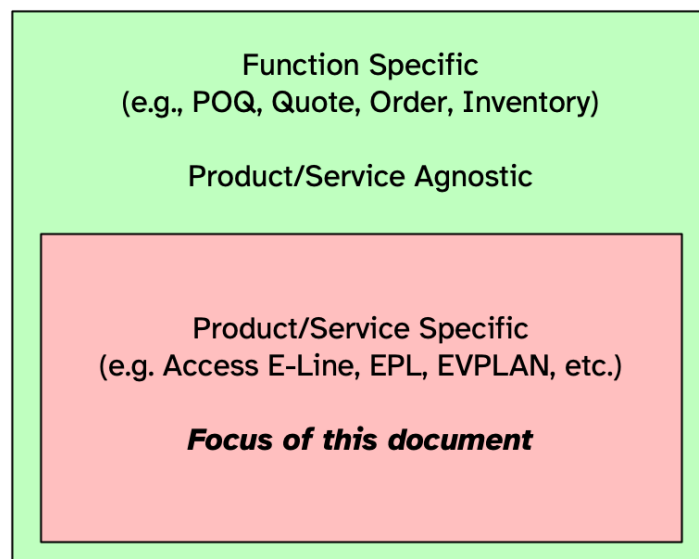
501 **Figure 2 LSO Cantata and LSO Sonata API Structure**

502 This document describes the product-specific Data Model for a MEF Internet Access service as
503 defined in MEF 61.1**Błąd! Nie można odnaleźć źródła odwołania.**.

504

# 7   Overview of Internet Access Services

This specification describes a data model for MEF-defined Internet Access Services. An Internet Access Service is a Subscriber IP Service which means it is provided to an end-user (the Subscriber) by a Service Provider. A Subscriber can be an enterprise, a mobile operator, an IT system integrator, a government department, etc. An Internet Access Service provides the Subscriber with connectivity to the global Internet. In this case, the Service Provider is acting as an Internet Service Provider.

Internet Access is composed of 5 main building blocks:

- IPVC: An IP Service is formed of an IP Virtual Connection (IPVC) that links together IPVC End Points at a UNI (or IPVC End Points and "the Internet" as in the Internet access case).

- IPVC End Point: A logical entity at a UNI, to which a subset of packets that traverse thatUNI is mapped.

- UNI – A User Network Interface (UNI), the demarcation point between the responsibility of the SP and the responsibility of the Subscriber.  Note that a given UNI always relates to a single SP and a single Subscriber.

- IP UNI Access Link: An individual IP connection (i.e. a subnetwork corresponding to a distinct IP subnet) between the Subscriber and the Service Provider that forms part of that UNI.

- IP UNI Access Link Trunk: An underlying construct that encapsulates the Layer 1 and Layer 2 characteristics of the link. A UNI Access Link Trunk may carry packets for a single UNI Access Link, as in the case where the UNI Access Link is a direct physical connection, or may carry packets for multiple UNI Access Links, for example when the UNI Access Link is an Ethernet VLAN. The UNI Access Link Trunk itself may be a single physical link, may comprise multiple physical links such as an Ethernet Link Aggregation Group, or may be logical such as an IP tunnel.

530
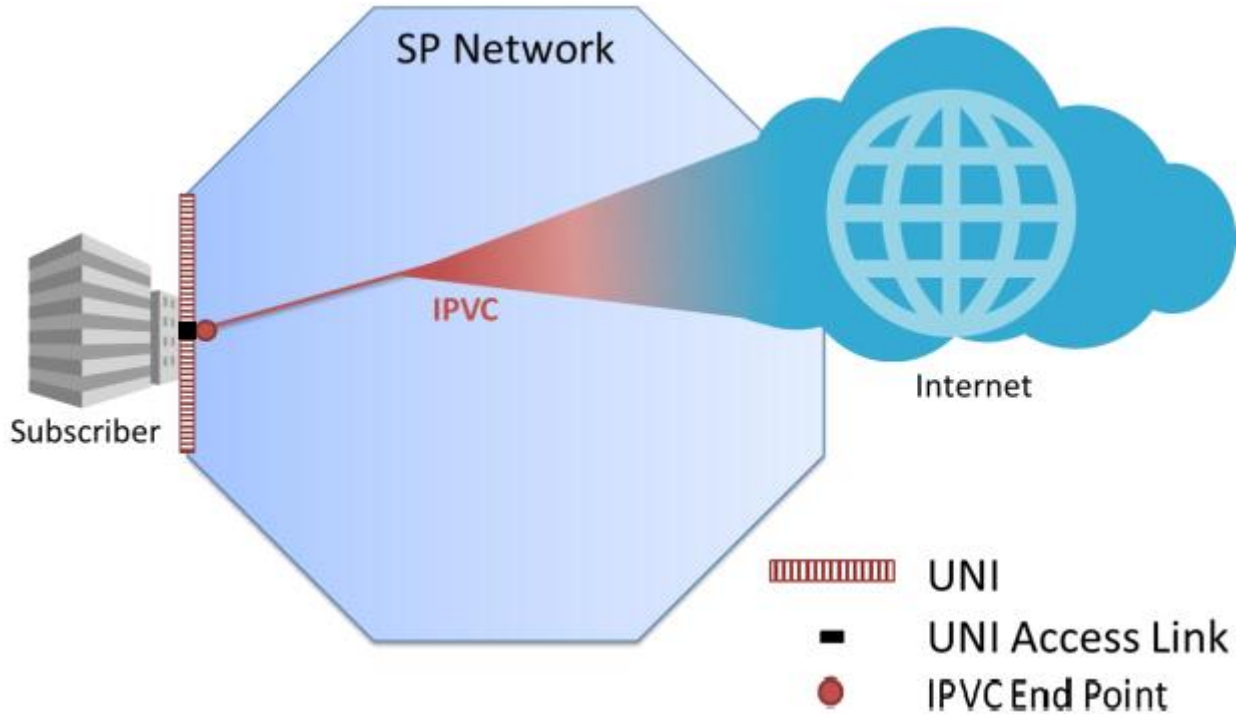
531 **Figure 3 Internet Access Service  - concept** Błąd! Nie można odnaleźć źródła odwołania.

532 Subscribers' perception of Internet access is that it allows general access to a range of content
533 available on the Internet. The content can be served from within the SP Network, or typically from
534 outside of it. There is no strict boundary between the IPVC that provide access to the Internet, and
535 the Internet itself (as shown in the Figure above). The IPVC thus has only one IPVC End Point at
536 the UNI that connects to the Subscriber but does not have one that would connect it to the Internet.

537 Figure 4 shows some examples of how UNI Access Links in a given UNI can be connected to one
538 or multiple devices at the Subscriber and at the Service Provider. Other arrangements are also
539 possible.

Single UNI
Access Link

UNI Access Links
terminating on
different devices
at the SP

UNI Access Links
terminating on
different devices
at the SP and
Subscriber

UNI Access Links
terminating on
different devices
at the Subscriber

||||||||||||  UNI

▬  UNI Access Link

540
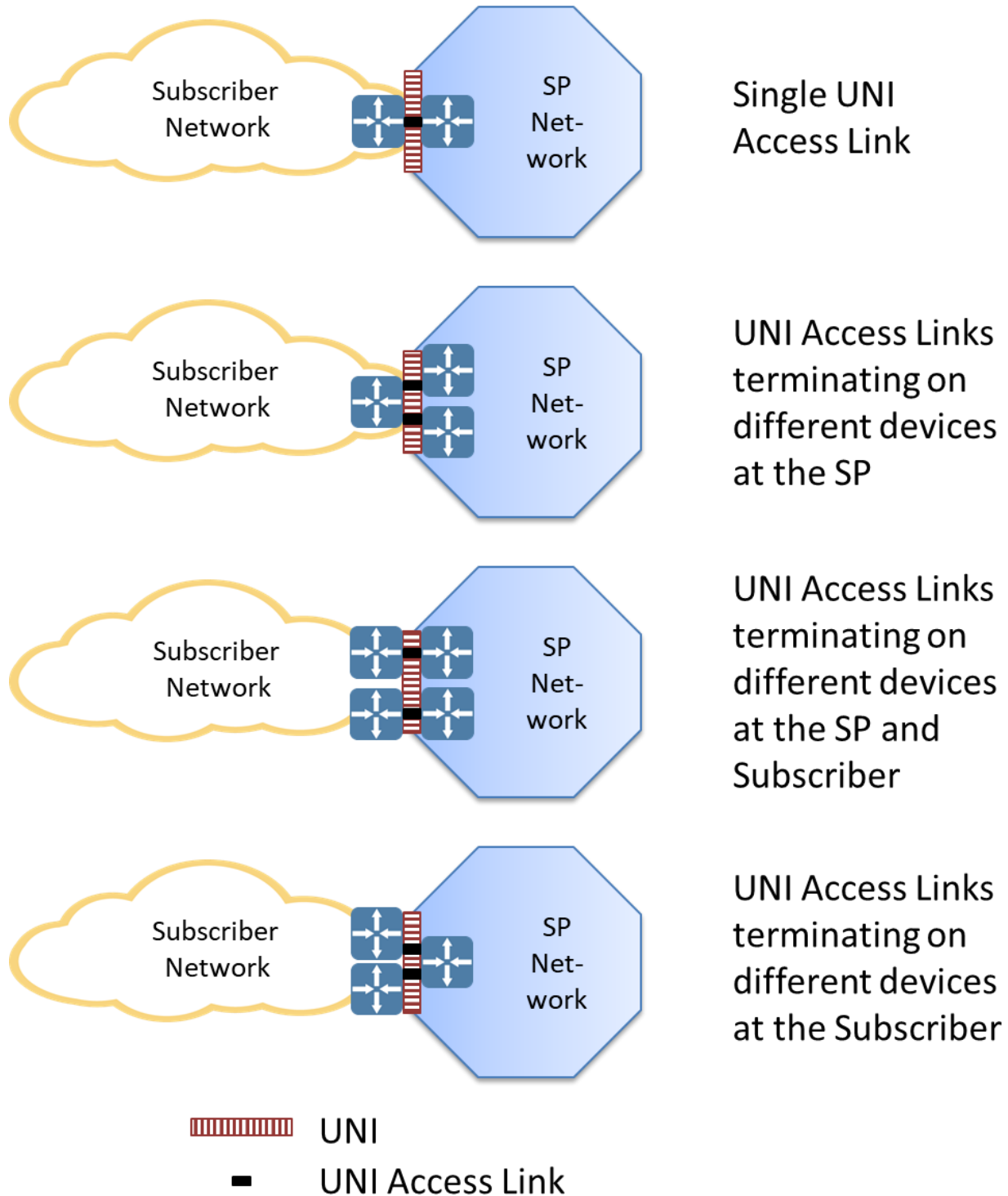
541    **Figure 4 Examples of UNI Access Links in a Single UNI** Błąd! Nie można odnaleźć źródła odwołania.

542

543 Two types of Internet Access can be offered: Basic and Advanced. The possible values for certain
544 Service Attributes differ between these two types. Basic Internet Access is typically delivered to
545 residential dwellings. It may be offered to small/medium businesses. Its service characteristics
546 typically include:

547 • plug-and-play ease of use

548 • low-cost

549 • For Ipv4, a few (or shared) publicly routed addresses

550 Advanced Internet Access is typically delivered to business locations. Its service characteristics
551 include:

552 • redundancy features

553 • dynamic routing protocol support (e.g., BGP [1] routing)

554 • options for Subscriber-supplied IP addressing

555 • proactive monitoring to support a Service Level Specification (SLS)
556

# 8  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 Ipv4 Secondary Subnet List. 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 Internet Access 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 is the IPVC End Point defined for the IPVC (and subclasses). This must be included in the Internet Access data model since the IPVC is incomplete without them and is therefore tagged as "Required".

The design for the Internet Access data model is based on several 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 Functions, Product Actions, and Product Offerings.

    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 8.1.

    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 8.2.

    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 8.3.

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 Internet Access data model supports both INSTALL and CHANGE actions for POQ, Quote, and Order for all specified products. Note that the DISCONNECT action does not require support from the data model.

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

## 8.1     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.

**[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.

## 8.2   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.

## 8.3   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 upon 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.

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

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

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

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

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

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

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

688  ———

# 9 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 Internet Access Service (IPVC 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, IPVC |
| FIRM_DELIVERY_DATE_PROVIDED | Firm Delivery Date Provided | UNI, IPVC |
| 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, IPVC |
| 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, IPVC |
| 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 | IPVC |
| E2E_TESTING_COMPLETED | End-to-End Testing Completed | IPVC |
| E2E_TESTING_FAILED | End-to-End Testing Failed. Issue is to be resolved and testing may be re-attempted. | IPVC |

**Table 2 Order Milestones for Internet Access**

696     The Milestone Value in the first column of Table 2 is included in *ProductOrderItem.milestone* and
697     *ProductOrderEventPayload.milestoneName* in the Product Order API (see MEF 123 [19]).

698     Note: Milestones and their notifications are independent of Product Order Item's state.

## 10  Information Model for Internet Access Product Data Model

Internet Access Services are composed of five primary classes of objects: IPVC, IPVC End Point, IP UNI, IP UNI Access Link, and IP UNI Access Link Trunk. A complete Internet Access product consists of:

- Exactly one IPVC (see section 14.3)
- One IP UNI where the Subscriber accesses the service (see section 14.4).
- Exactly one IPVC End Point for the IPVC at this IP UNI. (see section 14.3.4).
- One (for Basic and Exclusive Internet Access) or more UNI Access Links in each UNI, (see section 14.5).
- One (for Basic and Exclusive Internet Access) or more UNI Access Link Trunks each carrying one or more UNI Access Links (see section 14.6).

Based on the above there are two main types of Internet Access defined – Basic and Advanced. The Advanced one comes with an additional flavor called "Exclusive". The differences between them are explained by the following figures. The convention is as follows:

- The surrounding rectangle designates the scope of a given product and provides its name and urn.
- The model shows only the main components listed above and the relations between them, including cardinalities. All other attributes and types are hidden.
- Relations between other products (crossing the big rectangles) or locations are provided on the envelope level (as specified in section 12). The source and target of such relations on the diagrams are bound to objects that are their logical sides, yet technically the relation on the envelope is on the root product level.

724
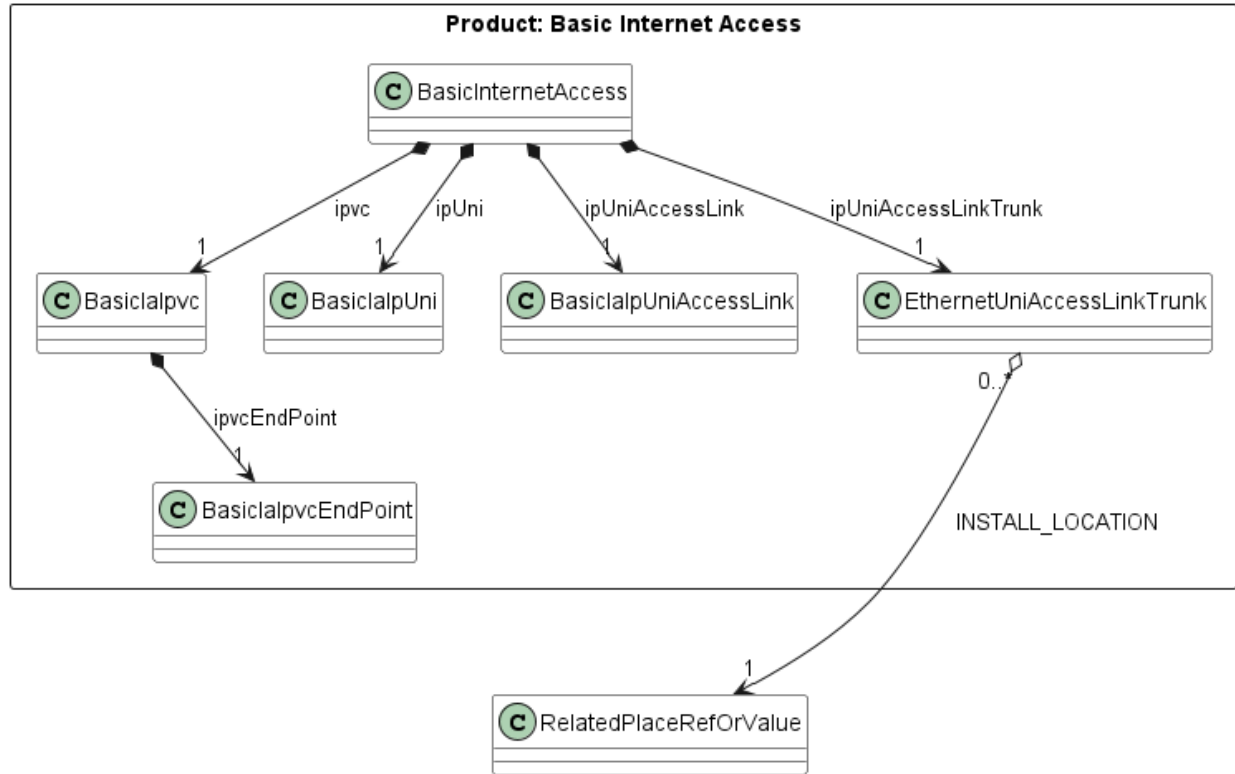
725                    **Figure 5 Information model for Basic Internet Access product**

726    Figure 5 presents the information model for Basic Internet Access. MEF 69.1 [10] defines
727    restrictions for Basic Internet Access such that all relations' cardinalities have exactly the value of
728    1 and that they are exclusive for a given product instance (meaning that all components serve only
729    one IPVC). This allows this product to be modeled in a simplified way as one main type
730    (BasicInternetAccess) having all components as single ref attributes. This means that all
731    components (IPVC, IPVC End Point, UNI, UNI Access Link, and UNI Access Link Trunk) are
732    ordered with a single Product Order Item. Since all components are within the same order, the only
733    envelope-level relation is the one to a place. It is the UNI Access Link Trunk that is the logical
734    owner of the relationship

735



736
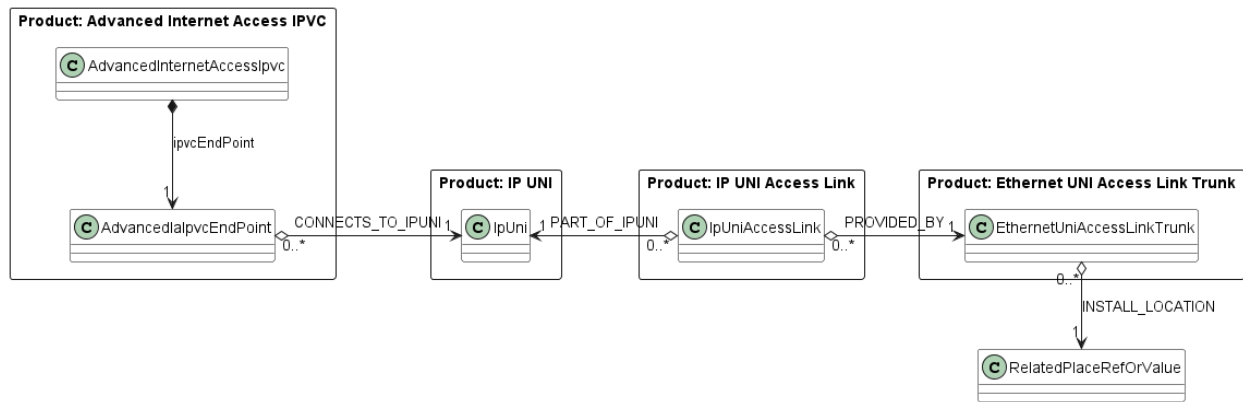
**Figure 6 Information model for Advanced Internet Access product**

738  Figure 6 shows the building blocks of an Advanced Internet Access product. It implements the
739  Advanced flavor of Internet Access as specified by MEF 69.1 [10]. In fact, it is a set of 4 distinct
740  products that must be ordered separately by different Product Order Items of one (or more) Product
741  Orders. Note the main differences compared to Basic Internet Access:

742  • IP UNI can serve more than one Advanced Internet Access product (and possibly other IP
743    products such as IP VPN).

744  • IP UNI can be provided by more than one IP UNI Access Link.

745  • Ethernet UNI Access Link Trunk can serve more than one IP UNI Access Link.

746  • All relations between components are specified by envelope-level product or order item
747    relationships (as specified in section 12).

748  • The place relationship is specified by the Ethernet UNI Access Link Trunk product.

749
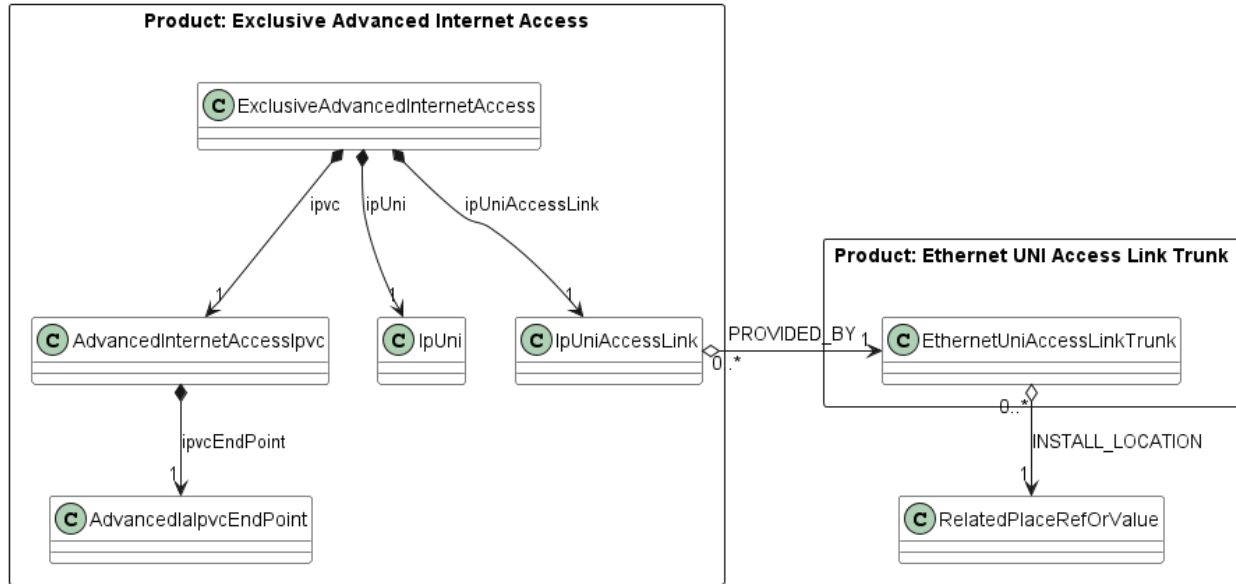
**Figure 7  Information model for Exclusive Advanced Internet Access IPVC product**

751  MEF 69.1 [10] defines 2 types of Internet Access: Basic and Advanced. They differ by several
752  requirements summarized in section 13. However, the flexibility of Advanced Internet Access
753  comes with the burden of the necessity of ordering 4 different products. This is addressed by
754  introducing the Exclusive Advanced Internet Access product. It is still an Advanced Internet
755  Access as specified by MEF 69.1 [10], but adds some assumptions that cover most of the probable
756  common deployment configurations. These are:

757  • The IPVC is served exclusively by one IP UNI (hence the product name)

758  • The IP UNI is served exclusively by one IP UNI Access Link.

759  This allows merging the IPVC, IP UNI, and IP UNI Access Link into one product definition called
760  Exclusive Advanced Internet Access. This also reduces the number of product relations defined
761  on the envelope level to only 1. The Ethernet UNI Access Link Trunk remains a separately ordered
762  product, allowing for serving multiple IP UNI Access Links.

763  **10.1    Organization of Service Attributes**

764  The data model of Internet Access products is based on Service Attributes defined in MEF 61.1
765  [8], and MEF 61.1.1 [9], and implements Service Definition Requirements as specified in MEF
766  69.1 [10] Section 9. These requirements result in Basic and Advanced versions being a variation
767  of Service Attributes defined in MEF 61.1 **Błąd! Nie można odnaleźć źródła odwołania.**. A set
768  of Common classes was introduced to gather the attributes shared by Basic and Advanced flavors.
769  Note that the Common types are not as specified by MEF 61.1 [8] or MEF 61.1.1 [9] but only
770  subsets of them.

771



772
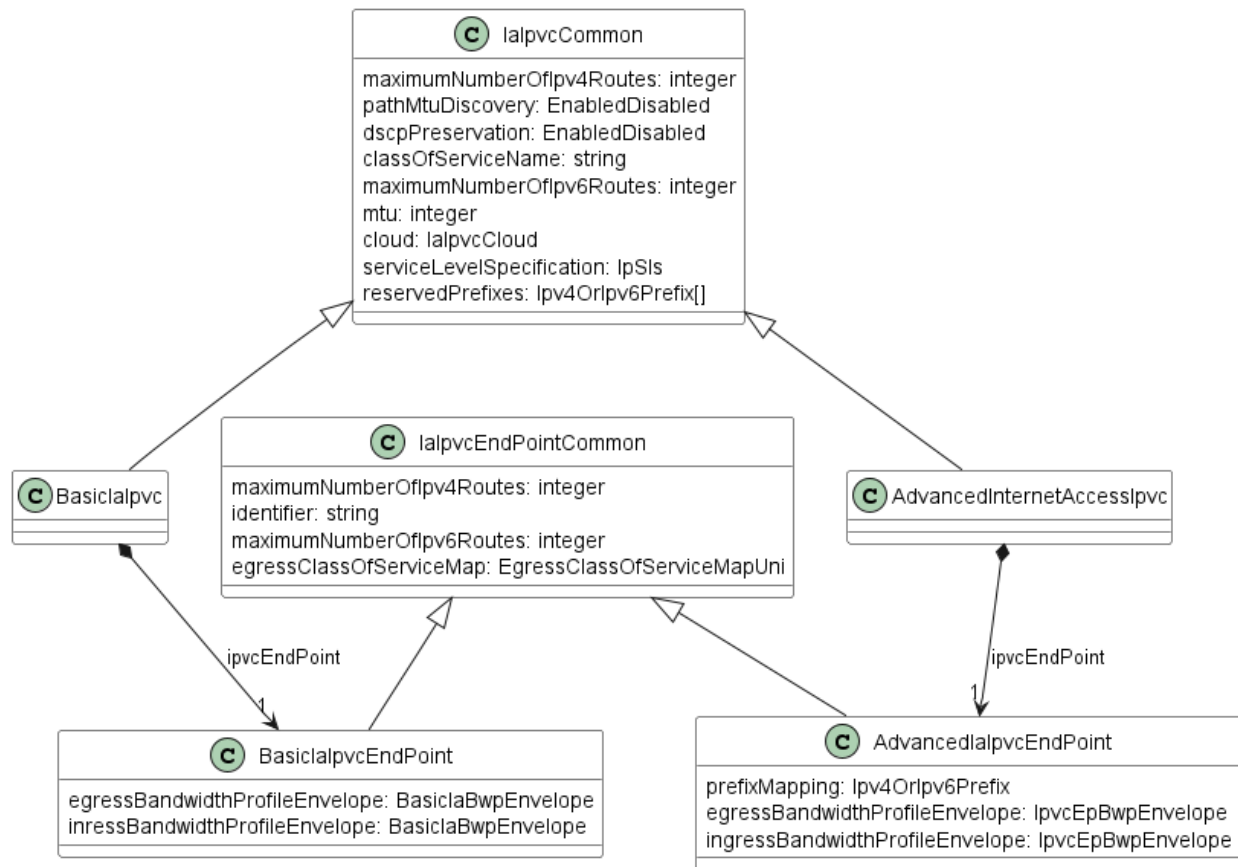
773  **Figure 8 IPVC and IPVC End Points Common classes**

774  Figure 8 presents the organization of Common IPVC and IPVC End Point types and differences
775  in their respective Basic and Advanced subtypes. The IPVC flavors differ only by the type of
776  referenced IPVC End Points which have different types used for ingress and egress bandwidth
777  profiles envelopes and the AdvanceIaIpvcEndPoint additionally specifies prefixMapping. The
778  details of the differences are described in section 13.

779
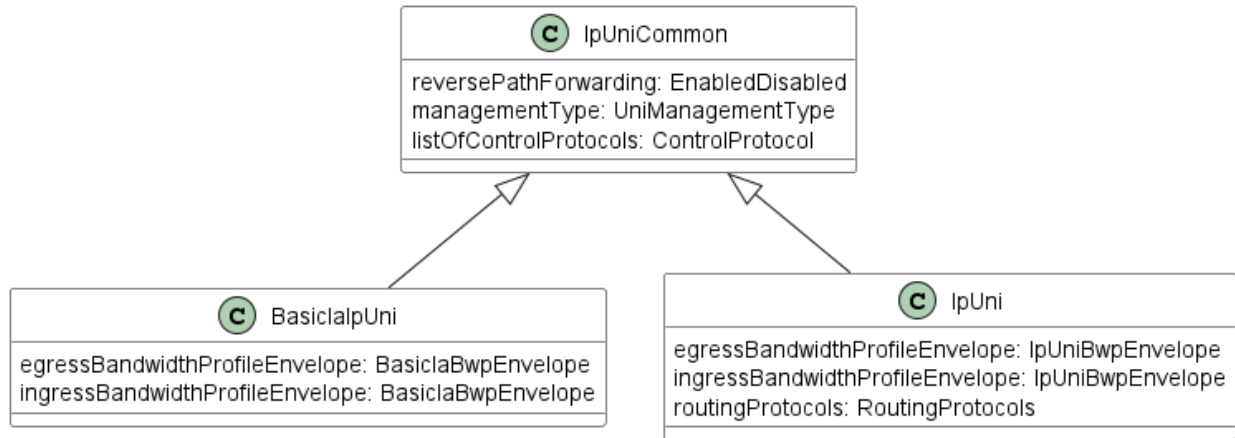
**Figure 9 IP UNI Common class**

780

781    Figure 9 shows that the difference between the Basic and Advanced flavors of UNI is how the
782    bandwidth profiles envelopes are specified and the IpUni can additionally provide
783    routingProtocols configuration. Note that the Advanced prefix is not present for the IpUni model
784    used by Advanced Internet Access. This is because this form does not introduce any Internet
785    Access specific restrictions and can be shared by different IP products (i.e. IP VPN) both on the
786    data model and instance level.



787
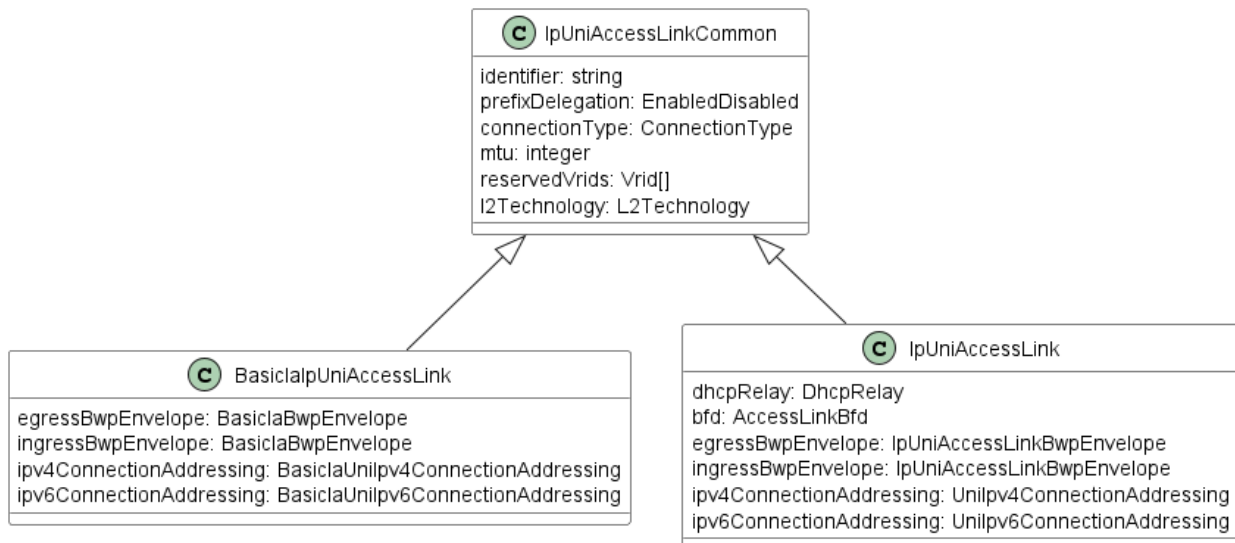
788    **Figure 10 IP UNI Access Link**

789    Figure 10 shows the differences between the Basic and Advanced flavors of IpUniAccessLink.
790    They differ on how the Bandwidth profile envelope and Ipv4/Ipv6 Connection Addressing are
791    specified. Additionally, IpUniAccessLink allows the specification of DHCP relay and BFD
792    attributes. As in the IpUni case – only BasicIaIpUniAccessLink is Internet Access specific.
793    IpUniAccessLink may also be used by other IP products.

794
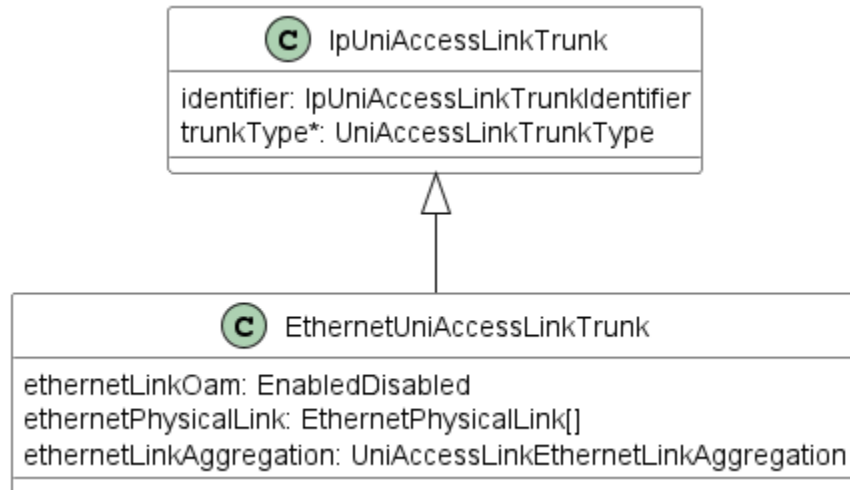
**Figure 11 Ethernet Uni Access Link Trunk**

796 The Ethernet Uni Access Link Trunk has the same representation in all flavors. Figure 11 shows
797 its inheritance from IpUniAccessLinkTrunk. MEF 61.1.1 [9] specifies Ethernet Uni Access Link
798 Trunk as the only available implementation of the IpUniAccessLinkTrunk.

# 11 Data Models for Internet Access Product

The data models for the Internet Access product configuration are expressed as a set of JSON schemas based on JSON schema draft 7 [1] and encoded in YAML. These schemas accompany this document. This section explains the organization and structure of these schemas.

## 11.1 Organization and Structure of the Schemas

The schemas are organized into a file structure as shown in Figure 12.

```
ip
  common
    ipCommon.yaml
    ipSls.yaml
  internetAccess
    advancedInternetAccessIpvc
      advancedInternetAccessIpvc.yaml
    basicInternetAccess
      basicInternetAccess.yaml
    exclusiveAdvancedInternetAccess
      exclusiveAdvancedInternetAccess.yaml
    internetAccessCommon
      internetAccessCommon.yaml
  ipUni
    ethernetUniAccessLinkTrunk.yaml
    ipUni.yaml
    ipUniAccessLink.yaml
```

**Figure 12 Schema Files Organization**

There are 3 root product specifications for Internet Access, namely *BasicInternetAccess*, *AdvancedInternetAccessIpvc*, and *ExclusiveAdvancedInternetAccess*. They are specified by schemas in separate dedicated directories and files inside the *internetAccess* directory. There is also the *internetAccessCommon.yaml* that holds the definitions of types shared among the Internet Access products. The *ipUni* catalog holds schemas for separately orderable products that are building blocks for Advanced and Exclusive Advanced Internet Access Products. The *common* catalog keeps the definition of types that are shared among other IP services.

Note: A CHANGE request cannot change a single Service Attribute. The Buyer must send a full product configuration including all Mandatory Service Attributes (section 8.1) and all Optional Service Attributes (section 8.2) that were previously specified by the Buyer (in an INSTALL request or previous CHANGE request). Any Optional Service Attributes that are not specified in a CHANGE request are reset to their default value.

819 ## 11.2    Additional Details

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

823 ### 11.2.1    Naming Conventions

824 In the schemas, class and type names are UpperCamelCase and Service Attribute/property names
825 are lowerCamelCase. Enumeration values are defined using UPPER_SNAKE_CASE.

826 ### 11.2.2    IPVC End Point Service Attributes

827 IPVC End Points are not separately orderable items. They are part of the IPVC. The IPVC End
828 Points are the repositories for IPVC Service Attributes that can be different at each UNI.whereas
829 the IPVC Service Attributes have the same value at every point in the IPVC. The Internet Access
830 information model requires the IPVC to include exactly one IPVC End Point of type UNI hence
831 there is an explicit single attribute defined for IaIpvcCommon: `ipvcEndPoint.`

832 Internet Access allows this simplified coding since it has exactly one End Point and exactly of type
833 UNI. In the general case of a service that allows an arbitrary number of End Points (e.g., a
834 multipoint service) or where the external interface types are not predetermined, the End Points will
835 most likely be modeled as separately orderable products instead of being attributes of the IPVC.

836 Note that one of the IPVC End Point Service Attributes is IPVC End Point EI Type (**Błąd! Nie
837 można odnaleźć źródła odwołania.** section 11.2) which can be "UNI" or "ENNI". Since this
838 information is implicit, this Service Attribute is not included in the schema for Internet Access,
839 but likely would be included for other IP Services.

## 12 Relationships Between Entities

This section describes the relationships (and their constraints) between the separately orderable products and between the products and places.

The use case for Advanced Internet Access is based on purchasing the AdvancedIaIpvc, AdvancedIaUni, and an AdvancedIaUniAccessLink

The relationship between separately managed products 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 is mandatory at INSTALL and CHANGE of the product.

The final column notes that during POQ and Quote, a list of references might be provided or not. The list denotes that a range of related objects is provided to choose from.

| Source Product | Relationship Type | INSTALL | CHANGE | Target Product | Cardinality |
|---|---|---|---|---|---|
| Advanced Internet Access IPVC | CONNECTS_TO_IPUNI | Mandatory | Mandatory | **IP UNI** | 1 |
| IP UNI Access Link | PART_OF_IPUNI | Mandatory | Mandatory | **IP UNI** | 1 |
| IP UNI Access Link | PROVIDED_BY | Mandatory | Mandatory | **Ethernet UNI Access Link Trunk** | 1 |
| Exclusive Advanced Internet Access IPVC | PROVIDED_BY | Mandatory | Mandatory | **Ethernet UNI Access Link Trunk** | 1 |

**Table 3 Product Relationship Roles**

**[R16]** For a product listed in the Source Product column of Table 3, the *Relationship Type* field of the *Product Relationship, POQ Item Relationship, Quote Item Relationship, and Order Item Relationship* types **MUST** contain the corresponding value shown in the Relationship Type column.

**[R17]** For POQ, Quote, and Order, relationships listed in Table 3 **MUST** be specified for every INSTALL of, or CHANGE to, a product listed in the Source Product column of Table 3.

**[R18]** For a product listed in the Source Product column of Table 3, the relationship **MUST** reference the respective product listed in the Target Product column or an equivalent POQ Item, Quote Item, or Order Item.

863     **[R19]**      For a CHANGE to a product listed in the Source Product column of Table 3
864          the specified relationship **MUST NOT** be changed from the value present in
865          the Product Inventory.

866 Note that [R19] indicates that once Advanced Internet Access IPVC or Ip Uni Access Link product
867 is associated with an IP UNI, it cannot be associated with a different IpPUNI.

868 The Ethernet UNI Access Link Trunk is the location-oriented component that builds the Internet
869 Access product. In the case of Basic Internet Access, the Ethernet UNI Access Link Trunk is part
870 of the whole product definition, thus it is the Basic Internet Access product that needs to have a
871 relationship to the location. In Advanced Internet Access cases, the Ethernet UNI Access Link
872 Trunk is a separately orderable product, so the location relation must be set from this product. The
873 Ethernet UNI Access Link Trunk is associated with a specific INSTALL_LOCATION and as
874 noted below, it is required at INSTALL and CHANGE and once it is associated with a specific
875 location, the INSTALL_LOCATION cannot be changed. The install location is captured in the
876 product-agnostic part of the POQ, Quote, and Order APIs. The value in the Place Relationship
877 Role column in the table below is used in the *role* field of the *RelatedPlaceRefOrValue* type.

878                                                           **Table 4**

| Product | Place Relationship Role | INSTALL | CHANGE |
|---|---|---|---|
| **Basic Internet Access** | INSTALL_LOCATION | Mandatory | Mandatory |
| **Ethernet UNI Access Link Trunk** | INSTALL_LOCATION | Mandatory | Mandatory |

879                                          **Table  Place Relationship Role**

880     **[R20]**      For Basic Internet Access or Ethernet UNI Access Link Trunk products, the
881          Role field (*role*) of the Related Place (*RelatedPlaceRefOrValue)* type **MUST**
882          contain the INSTALL_LOCATION value shown in the Place Relationship
883          Role column in.

884     **[R21]**      For POQ, Quote, and Order, the Related Place (*RelatedPlaceRefOrValue)*
885          **MUST** be specified for every INSTALL of, or CHANGE to, a Basic Internet
886          Access and Ethernet UNI Access Link Trunk.

887     **[R22]**      For a CHANGE to a Basic Internet Access or Ethernet UNI Access Link Trunk
888          product, the Related Place **MUST NOT** be changed from the value present in
889          the Product Inventory.

890

891 # 13 Basic vs. Advanced Service Attributes requirements

892 There are several Service Attributes defined by MEF 61.1 **Błąd! Nie można odnaleźć źródła**
893 **odwołania.** that MEF 69.1 [10] puts additional requirements when applying to Basic or Advanced
894 Internet Access definition. This results in some attributes differing from their original definition
895 or missing from the Product Schema specified by this document.

896 These variations are presented for both Basic and Advanced versions, side by side in the Tables
897 below (all numbered requirements come from MEF 69.1 [10] and thus the document number is
898 not mentioned each time):

| Service Attribute | Basic Internet Access (BasicIaIpvc) | Advanced Internet Access (AdvancedIaIpvc) |
|---|---|---|
| IPVC Identifier | **Not present**<br><br>There is no need for an additional Identifier. The IPVC product instance gets the product.is assigned upon creation in the Seller's system, which then can be used for inter-product references. | |
| IPVC Topology | **Not present**<br>[R4] IPVC Topology MUST be *Cloud Access* | |
| IPVC End Point List | [R5] IPVC End Point List MUST have exactly one entry. | |
| | **Single attribute instead of a list: BasicIaIpvc.ipvcEndPoint**<br><br>**Ref type: BasicIaIpvcEndPoint** | **Single attribute instead of a list: AdvancedIaIpvc.ipvcEndPoint**<br><br>**Ref type: AdvancedIaIpvcEndPoint** |
| IPVC Packet Delivery | **Not present.**<br><br>Packet Delivery is an enumeration with 2 values: *Static Routing* and *Policy Based Routing*. But according to the description "*The behavior and requirements when the IPVC Packet Delivery Service Attribute is set to Policy-Based Routing are deferred to a future revision of this specification*". That leaves the *Standard Routing* the only available option, so there is no need to specify it. | |
| IPVC DSCP Preservation | [D3]IPVC DSCP Preservation SHOULD be Disabled.<br><br>**The requirement is stated in the attribute's description.** | |
| IPVC List of Class of Service Names | [R7] The IPVC List of Class of Service Names MUST have exactly one entry<br><br>**Single attribute instead of a list:** *classOfServiceName* | |
| IPVC Fragmentation | **Not present.**<br><br>[R8] IPVC Fragmentation MUST be Enabled.<br><br>Note: Fragmentation is necessary for an Internet Access Service as the Subscriber has no control over the size of packets received from the Internet. IPVC Fragmentation Enabled ensures the ISP will not discard any packets destined for the Subscriber that exceed the allowable IPVC MTU size. | |

| Service Attribute | Basic Internet Access (BasicIalpvc) | Advanced Internet Access (AdvancedIalpvc) |
|---|---|---|
| IPVC Cloud<br>Cloud Type | **Not present.**<br><br>[R9] IPVC Cloud. Cloud Type MUST be *Internet Access.* | |
| IPVC Cloud<br>Cloud Ingress Class of Service Map | **Not present.**<br><br>[R10] Cloud Ingress Class of Service Map (F, M, D), map M MUST be empty.<br><br>[R11] Cloud Ingress Class of Service Map (F, M, D), default CoS name, D, MUST NOT be *Discard*.<br><br>When map M is empty, the F has no effect. Additionally, only one Class of Service can be specified, so with R11, that means there is no point in specifying the whole Ingress Class of Service Map. | |
| IPVC Cloud<br>Cloud DNS Service | [R12] For a Basic Internet Access Service, Cloud DNS MUST NOT be *None*.<br>**The requirement is stated in the attribute's description.** | For an Advanced Internet Access Service, a value of None for Cloud DNS is not precluded. |
| IPVC Cloud<br>Cloud DNS Service | [R13] If the Cloud DNS parameter of the IPVC Cloud Service Attribute is *Static*, the associated list of DNS Servers MUST have at least one entry.<br><br>[D4] If the Cloud DNS parameter of the IPVC Cloud Service Attribute is *Static*, the associated list of DNS Servers SHOULD contain at least two DNS servers**.**<br><br>**Requirements are stated in the attribute's description.** | |
| IPVC Reserved Prefixes | [R14] IPVC Reserved Prefixes MUST be either empty or free from any public address prefixes.<br><br>**The requirement is stated in the attribute's description.** | |

899                                   **Table 5 IPVC Service Attributes requirements**

900

| Service Attribute | Basic Internet Access (BasicIalpvcEndPoint) | Advanced Internet Access (AdvancedIalpvcEndPoint) |
|---|---|---|
| IPVC EP EI | **Not present.**<br><br>**IpUni is a composite of BasicInternetAccess there is no need to use additional references.** | **Not present.**<br><br>**IpUni is either a composite of ExclusiveAdvancedInternetAccess and there is no need to use additional references or is referenced on the envelope level in the case of AdvancedInternetAccessIpvc product** |
| IPVC EP EI Type | **Not present.**<br><br>Always the value of *UNI* | |

| Service Attribute | Basic Internet Access (BasicIaIpvcEndPoint) | Advanced Internet Access (AdvancedIaIpvcEndPoint) |
|---|---|---|
| IPVC EP Role | **Not present.**<br><br>[R16] IPVC EP Role MUST be *Root*. | |
| IPVC EP ENNI Service Mapping Identifier | **Not present.**<br><br>Not relevant for Internet Access | |
| IPVC EP Ingress Class of Service Map | **Not present.**<br><br>[R17] IPVC Ingress EP Class of Service Map (F, M, D), map M MUST be empty.<br><br>[R18] IPVC Ingress EP Class of Service Map (F, M, D), default CoS name, D, MUST NOT be *Discard*.<br><br>When map M is empty, the F has no effect. Additionally, only one Class of Service can be specified, so with R11, that means there is no point in specifying the whole Ingress Class of Service Map. | |
| IPVC EP Ingress Band-width Profile Envelope | **Ref type: BasicIaBwpEnvelope**<br><br>[D5] For a Basic Internet Access Service, the IPVC EP Ingress Bandwidth Profile Envelope SHOULD be *None*.<br><br>**The requirement is stated in the attribute's description.** | **Ref type: IpvcEpBwpEnvelope** |
| IPVC EP Egress Band-width Profile Envelope | **Ref type: BasicIaBwpEnvelope**<br><br>[D6] For a Basic Internet Access Service, the IPVC EP Egress Bandwidth Profile Envelope SHOULD be *None*.<br><br>**The requirement is stated in the attribute's description.** | **Ref type: IpvcEpBwpEnvelope** |
| IPVC EP Prefix Mapping | **Not present.**<br><br>[R19] For a Basic Internet Access Service, the IPVC EP Prefix Mapping MUST be Empty. | -- |

901                          **Table 6 IPVC End Point Service Attributes requirements**

902

| Service Attribute | Basic Internet Access (BasicIaIpUni) | Advanced Internet Access (IpUni) |
|---|---|---|
| UNI Identifier | **Not present**<br><br>There is no need for an additional Identifier. The IpUni product instance gets the product.is assigned upon creation in the Seller's system, which then can be used for inter-product references | |

| Service Attribute | Basic Internet Access (BasicIaIpUni) | Advanced Internet Access (IpUni) |
|---|---|---|
| UNI List of UNI Access Links Service Attribute | **Not present.**<br><br>IpUniAccessLink is a composite of BasicInternetAccess there is no need to use additional references. | **Not present.**<br><br>IpUniAccessLink is either a composite of ExclusiveAdvancedInternetAccess and there is no need to use additional references or is referenced on the envelope level in the case of IpUni product |
| UNI Ingress Bandwidth Profile Envelope | **Ref type: BasicIaBwpEnvelope**<br><br>[D7] At a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Ingress Bandwidth Profile Envelope is not *None*, it SHOULD have Bandwidth Profile Flows that contain all Ingress IP Data Packets at the UNI that are mapped to any of a given set of IPVC EPs (as defined in MEF 61.1 Table 28).<br><br>**The requirement is stated in the attribute's description.** | **Ref type: IpUniBwpEnvelope** |
| UNI Egress Bandwidth Profile Envelope | **Ref type: BasicIaBwpEnvelope**<br><br>[D8] At a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Egress Bandwidth Profile Envelope is not *None*, it SHOULD have Bandwidth Pro-file Flows that contain all Egress IP Data Packets at the UNI that are mapped to any of a given set of IPVC EPs (as defined in MEF 61.1Table 28).<br><br>**The requirement is stated in the attribute's description.** | **Ref type: IpUniBwpEnvelope** |
| UNI List of Control Protocols | [D9] At a UNI with an IPVC EP for an Internet Access Service, if the UNI has at least one UNI Access Link where the UNI Access Link IPv4 Connection Addressing is not None, the UNI List of Control Protocols SHOULD include ICMP with a list of applicable ISP IP addresses.<br><br>[D10] At a UNI with an IPVC EP for an Internet Access Service with at least one UNI Access Link where the UNI Access Link IPv6 Connection Addressing is not None, the UNI List of Control Protocols SHOULD include ICMPv6 with a list of applicable SP IP addresses.<br><br>**The requirement is stated in the attribute's description.** | |
| UNI Routing Protocols | **Not present.**<br><br>[R21] At a UNI with an IPVC EP for a Basic Internet Access Service, the UNI Routing Protocols list MUST be empty. | -- |
| UNI Reverse Path For-warding | [D11] At a UNI with an IPVC EP for an Internet Access Service, UNI Reverse Path Forwarding SHOULD be Enabled.<br><br>**The requirement is stated in the attribute's description.** | |

903

**Table 7 IP UNI Service Attributes requirements**

| Service Attribute | Basic Internet Access (BasicIaIpUniAccessLink) | Advanced Internet Access (IpUniAccessLink) |
|---|---|---|
| UNI Access Link IPv4 Connection Addressing | [R23] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, UNI Access Link IPv4 Connection Addressing MUST be DHCP or None.<br><br>**Ref type:**<br>***BasicIaUniIpv4ConnectionAddressing* does not have the *ipv4AddressType* attribute, as if set it MUST be *DHCP*** | [R22] At a UNI Access Link in a UNI with an IPVC EP for an Advanced Internet Access Service, UNI Access Link IPv4 Connection Addressing MUST be Static or None.<br><br>**IpUniAccessLink is a type that is shared among other IP Services so it does not contain InternetAccess-specific restrictions, thus the requirement is only stated in the attribute's description.** |
| UNI Access Link IPv4 Connection Addressing | [R24] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Access Link IPv4 Connection Addressing is DHCP, the UNI Access Link IPv4 Connection Addressing Secondary Subnet List parameter MUST be empty.<br><br>**Ref type:**<br>***BasicIaUniIpv4ConnectionAddressing* does not have the *ipv4SecondarySubnetList* attribute.** | -- |
| UNI Access Link IPv4 Connection Addressing | [R25] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Access Link IPv4 Connection Addressing is DHCP, the UNI Access Link IPv4 Connection Addressing Primary Subnet parameter MUST contain only a single Service Provider IPv4 Address.<br><br>**The requirement is stated in the attribute's description.** | -- |
| UNI Access Link IPv6 Connection Addressing | [R27] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, UNI Access Link IPv6 Connection Addressing MUST be DHCP or SLAAC or None.<br><br>**BasicIaUniIpv6ConnectionAddressing:** *ipv6AddressType* **attribute only contains possible values:** *DHCP, SLAAC* | [R26] At a UNI Access Link in a UNI with an IPVC EP for an Advanced Internet Access Service, UNI Access Link IPv6 Connection Addressing MUST be Static or None.<br><br>**IpUniAccessLink is a type that is shared among other IP Services so it does not contain InternetAccess-specific restrictions, thus the requirement is only stated in the attribute's description.** |
| UNI Access Link IPv6 Connection Addressing | [R28] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Access Link IPv6 Connection Addressing is DHCP or SLAAC, the UNI Access Link IPv6 Connection Address Subnet List parameter MUST contain a single entry. | -- |

| Service Attribute | Basic Internet Access (BasicIaIpUniAccessLink) | Advanced Internet Access (IpUniAccessLink) |
|---|---|---|
| | **BasicIaUniIpv6ConnectionAddressing:** *ipv6Subnet* **is a single attribute instead of a list** | |
| UNI Access Link IPv6 Connection Addressing | [R29] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Access Link IPv6 Connection Addressing is DHCP or SLAAC, the UNI Access Link IPv6 Connection Addressing Subnet List parameter MUST contain only a single Service Provider IPv6 Address.<br><br>**The requirement is stated in the attribute's description.** | -- |
| UNI Access Link DHCP Relay | **Not present.**<br><br>[R30] For a Basic Internet Access Service, where the UNI contains only a single IP Service, the UNI Access Link DHCP Relay MUST be empty. | -- |
| UNI Access Link BFD | **Not present.**<br><br>[R31] For a Basic Internet Access Service, UNI Access Link BFD MUST be *None*. | -- |
| UNI Access Link Ingress Bandwidth Profile Envelope | **Ref type: BasicIaBwpEnvelope**<br><br>[D12] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, UNI Access Link Ingress Bandwidth Profile Envelope SHOULD be *None*.<br><br>**The requirement is stated in the attribute's description.** | **Ref type: IpUniAccessLinkBwpEnvelope** |
| UNI Access Link Egress Bandwidth Profile Envelope | **Ref type: BasicIaBwpEnvelope**<br><br>[D13] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, UNI Access Link Egress Bandwidth Profile Envelope SHOULD be *None*.<br><br>**The requirement is stated in the attribute's description.** | **Ref type: IpUniAccessLinkBwpEnvelope** |
| UNI Access Link Re-served VRIDs Service Attribute | [D14] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, UNI Access Link Reserved VRIDs Service Attribute SHOULD be None.<br><br>**The requirement is stated in the attribute's description.** | -- |

904                    **Table 8 IP UNI Access Link Service Attributes requirements**

## 14  Internet Access Service Attributes

This section provides a guide to the detailed model of the Internet Access product in all flavors. It starts with the model of the top-level product types, then dives into the models of main components (IPVC, IP UNI, etc.).

Not all MEF 61.1 [8] and MEF 61.1.1[9] Service Attributes are included in the data model. The Service Attributes that are not included are also listed in section 13. 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 in the context of Internet Access (i.e., can only have one possible value) or are simple attributes instead of lists because the cardinality is restricted to 1.

In figures below some classes' attributes or further class tree are skipped for diagram readability. This is denoted by the *"<<skipped>>"* clause.

Some requirements define Service Attributes  as mutually exclusive. This means that either one or the other must be provided, but 2 (or more) of them at the same time. This is defined in the schema using the "oneOf" statement at the "required" section of the type definition.

For example the IpvcEpBwpEnvelope has 2 attributes: bwpFlowPerCosName and bwpFlowAll, but only one of them must be set at the same time. This part of schema defines this requirement looks as follows:

```
oneOf:
  - required: [bwpFlowPerCosName]
  - required: [bwpFlowAll]
```

In following sections, where applicable,  this information is provided by including this section after the table with the attributes.

### 14.1    BasicInternetAccess

File: \ip\internetAccess\basicInternetAccess\basicInternetAccess.yaml

URN: urn:mef:lso:spec:cantata-sonata:basic-internet-access:v0.2.0:all

931
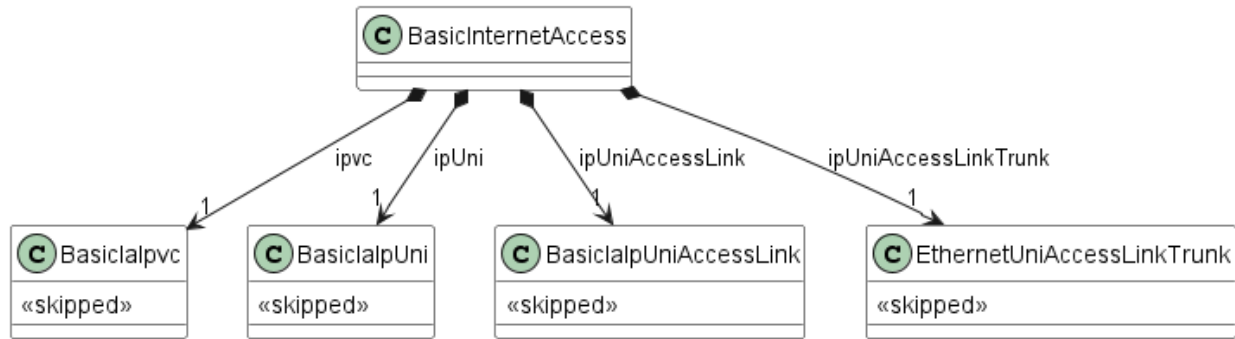
**Figure 13 Basic Internet Access product**

932

933 Figure 13 presents the model of *BasicInternetAccess*, as specified in *basicInternetAccess.yaml*. As
934 described in section 19, it gathers the configuration of all main product components in a single
935 "top-level" product. The details of components are skipped for readability and will be described
936 in later sections.

937 The Basic Internet Access IPVC is a MEF 69.1 defined version of MEF 61.1 IPVC. Reference
938 MEF 69.1 Section 9.1 Internet Access IPVC Requirements. It holds a complete configuration of
939 all needed product components: BasicIaIpvc, BasicIaIpUni, BasicIaIpUniAccessLink, and
940 ethernetUniAccessLinkTrunk.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| ipUniAccessLink* | BasicIaIpUniAccessLink | 1 | Configuration of Service Attributes for Basic Internet Access IP UNI Access Link |
| ipUni* | BasicIaIpUni | 1 | Configuration of Service Attributes for Basic Internet Access IP UNI |
| ipUniAccessLinkTrunk* | ethernetUniAccessLinkTrunk | 1 | Configuration of Service Attributes for Basic Internet Access IP UNI Access Link Trunk |
| ipvc* | BasicIaIpvc | 1 | Configuration of Service Attributes for Basic Internet Access IPVC |

941

**Table 9 BasicInternetAccess**

942 **14.2 ExclusiveAdvancedInternetAccess**

943 File:                                                                                                           \
944 ip\internetAccess\exclusiveAdvancedInternetAccess\exclusiveAdvancedInternetAccess.yaml

945 URN: urn:mef:lso:spec:cantata-sonata:exclusive-advanced-internet-access:v0.2.0:all

946



947
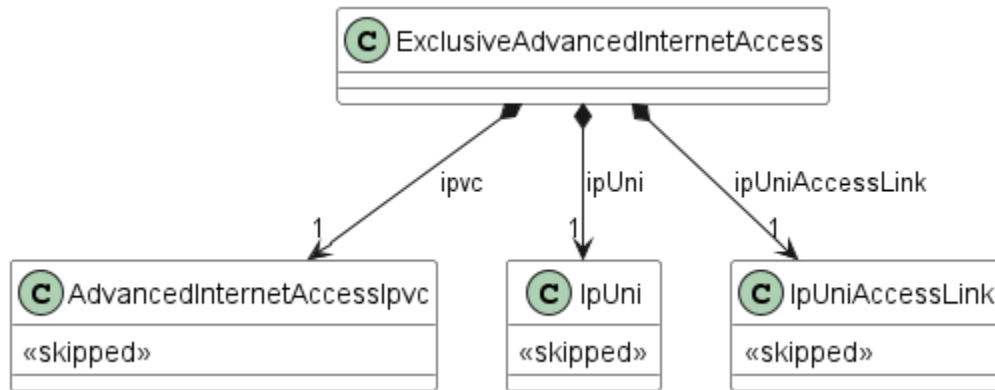
**Figure 14 Exclusive Advanced Internet Access product**

Figure 14 Exclusive Advanced Internet Access product presents the model of *ExclusiveAdvancedInternetAccess*, as specified in *exclusiveAdvancedInternetAccess.yaml*. As described in section 19, for simplicity it gathers the configuration of IPVC, IP UNI, and IP UNI Access Link components in a single "top-level" product. The details of components are skipped for readability and will be described in later sections.

The Advanced Internet Access IPVC is a MEF 69.1 defined version of MEF 61.1 IPVC. Reference MEF 69.1 Section 9.1 Internet Access IPVC Requirements. It holds the configuration for most of the required components: AdvancedInternetAccessIpvc, IpUni, and IpUniAccessLink. A reference to EthernetUniAccessLinkTrunk must be provided on the product level.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| ipUniAccessLink* | ipUniAccessLink | 1 | Configuration of Service Attributes for IP UNI Access Link |
| ipUni* | ipUni | 1 | Configuration of Service Attributes for IP UNI |
| ipvc* | advancedInternetAccessIpvc | 1 | Configuration of Service Attributes for Advanced Internet Access IPVC |

958

**Table 10 ExclusiveAdvancedInternetAccess**

959 **14.3    IPVC**

960

961
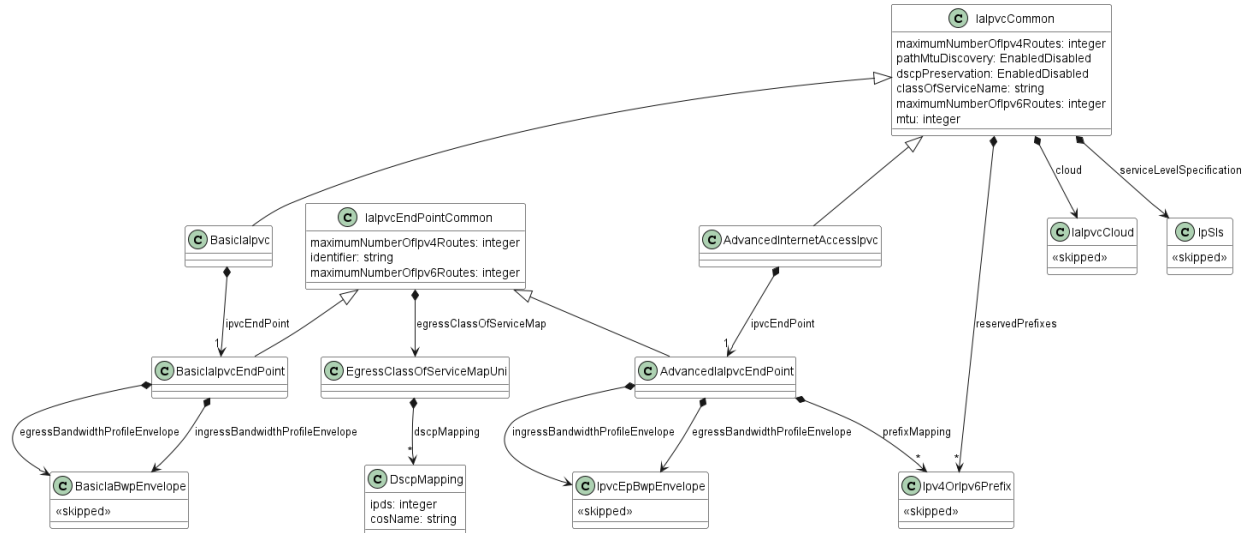
962                              **Figure 15 IPVC**

Figure 15 shows the model of the IPVC. In the case of Internet Access, the list of endpoints is
restricted to having only 1 item, so the endpoint relations are modeled as simple ones. Also,
differences between the Basic and Advanced versions are depicted.

### 14.3.1     IaIpvcCommon

File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

An IP Service is formed of an IP Virtual Connection (IPVC) that links together IPVC End Points
at External Interfaces (EIs). Reference MEF 61.1 Section 7.4 IP Virtual Connections and IPVC
End Points

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| maximumNumberOfIpv4Routes | integer | 0..1 | Maximum number of IPv4 routes supported by the service as a whole. Absence of this attribute corresponds to a value of "Unlimited". Reference MEF 61.1 Section 10.5 IPVC Maximum Number of IPv4 Routes Service Attribute. |
| maximumNumberOfIpv6Routes | integer | 0..1 | Maximum number of IPv6 routes supported by the service as a whole. Absence of this attribute corresponds to a value of "Unlimited". Reference MEF 61.1 Section 10.6 IPVC Maximum Number of IPv6 Routes Service Attribute. |
| dscpPreservation | EnabledDisabled | 0..1 | Indicates whether the Service Provider is allowed to modify the value of the IP DS field in the IP header of the Subscriber's traffic as it traverses the IPVC. Reference MEF 61.1 Section 10.7 IPVC DSCP Preservation Service Attribute. MEF 69.1 [D3] For an Internet Access Service, IPVC DSCP Preservation SHOULD be Disabled. |
| classOfServiceName | string | 0..1 | The Class of Service Name supported by the IPVC. Reference MEF 61.1 Section 10.8 IPVC List of Class of Service Names Service Attribute. This is "listOfClassOfServiceNames" attribute narrowed to single ref per Reference MEF 69.1 Section 9.1 [R7] |
| serviceLevelSpecification | IpSls | 0..1 | The set of performance objectives for CoS Name in the IPVC. The absence of this attribute corresponds to a value of "None". Reference MEF 61.1 Section 10.9 IPVC Service Level Specification Service Attribute. |
| mtu | integer | 0..1 | Indicates the maximum size (in octets) of an IP packet that can traverse the IPVC without fragmentation. Reference MEF 61.1 Section 10.10 IPVC MTU Service Attribute. |
| pathMtuDiscovery | EnabledDisabled | 0..1 | Indicates whether the Path MTU Discovery is supported for the IPVC. Reference MEF 61.1 Section 10.11 IPVC Path MTU Discovery Service Attribute. |
| cloud | IaIpvcCloud | 0..1 | Details of the cloud service being accessed. Reference MEF 61.1 Section 10.13 IPVC Cloud Service Attribute. |
| reservedPrefixes | Ipv4OrIpv6Prefix[] | 0..1 | Reference MEF 61.1 Section 10.14 IPVC Reserved Prefixes Service Attribute. For an Internet Access Service, IPVC Reserved Prefixes MUST be either empty, or free from any public address prefixes. (Reference MEF 69.1 Section 9.1 [R14]) |

971 **Table 11 IaIpvcCommon**

972    **14.3.2    BasicIaIpvc**

973    File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

974    The Basic Internet Access IPVC is a MEF 69.1 defined version of MEF 61.1 IPVC. Reference
975    MEF 69.1 Section 9.1 Internet Access IPVC Requirements.

976    Inherits from: IaIpvcCommon

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipvcEndPoint* | BasicIaIpvcEndPoint | 1 | Basic IPVC End Point. Reference MEF 61.1 Section 10.3 IPVC End Point List Service Attribute. This is narrowed to multiplicity = 1 and to BasicIaIpvcEndPoint type. Reference MEF 69.1 Section 9.1 [R5] AdvancedIaIpvc |

977                                                **Table 12 BasicIaIpvc**

978    **14.3.3    AdvancedInternetAccessIpvc**

979    File: \ip\internetAccess\advancedInternetAccessIpvc\advancedInternetAccessIpvc.yaml

980    URN: urn:mef:lso:spec:cantata-sonata:advanced-internet-access-ipvc:v0.2.0:all

981    The Advanced Internet Access IPVC is a MEF 69.1 defined version of MEF 61.1 IPVC. Reference
982    MEF 69.1 Section 9.1 Internet Access IPVC Requirements. A complete product setup requires
983    also ordering of IpUni, IpUniAccessLink, EthernetUniAccessLinkTrunk.

984    Inherits from: IaIpvcCommon

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipvcEndPoint* | AdvancedIaIpvcEndPoint | 1 | Advanced IPVC End Point. Reference MEF 61.1 Section 10.3 IPVC End Point List Service Attribute. This is narrowed to multiplicity = 1 and to AdvancedIaIpvcEndPoint type. Reference MEF 69.1 Section 9.1 [R5] AdvancedIaIpvc |

985                                         **Table 13 AdvancedInternetAccessIpvc**

986    **14.3.4    IaIpvcEndPointCommon**

987    File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

988    The Advanced Internet Access IPVC End Point is a MEF 69.1 defined version of MEF 61.1 IPVC
989    End Point. Reference MEF 69.1 Section 9.2 Internet Access IPVC End Point Requirements.

| Name | Type | Multiplicity | Description |
|------|------|-------------|-------------|
| maximumNumberOfIpv4Routes | integer | 0..1 | Maximum number of IPv4 routes supported by this IPVC End Point. Reference MEF 61.1 Section 11.7 IPVC EP Maximum Number of IPv4 Routes Service Attribute. Absence of this attribute corresponds to a value of "Unlimited". |
| identifier | string | 0..1 | IPVC End Point identifier as described in MEF 61.1 Section 11.1. Note - it is not the same thing as the potential Product identifier if IPVC Endpoint is an instance of a Product. |
| egressClassOfServiceMap | EgressClassOfServiceMapUni | 0..1 | Pair of values (D,P). D specifies how to set the DS field in Egress IP Data Packets based on CoS Name. It is either None, or a mapping from CoS Names to DSCP values. P specifies how to set the PCP field in VLAN Tagged Ethernet Frames containing Egress IP Data Packets based on CoS Name. It is either None, or a mapping from CoS Names to PCP values. Reference MEF 61.1 Section 11.10 IPVC EP Egress Class of Service Map Service Attribute. |
| maximumNumberOfIpv6Routes | integer | 0..1 | Maximum number of IPv6 routes supported by this IPVC End Point. Reference MEF 61.1 Section 11.8 IPVC EP Maximum Number of IPv6 Routes Service Attribute. Absence of this attribute corresponds to a value of "Unlimited". |

990 **Table 14 IaIpvcEndPointCommon**

### 14.3.5    BasicIaIpvcEndPoint

992    File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

993    The Basic Internet Access IPVC End Point is a MEF 69.1 defined version of MEF 61.1 IPVC End
994    Point. Reference MEF 69.1 Section 9.2 Internet Access IPVC End Point Requirements.

995    Inherits from: IaIpvcEndPointCommon

| Name | Type | Multiplicity | Description |
|------|------|-------------|-------------|
| egressBandwidthProfileEnvelope | BasicIaBwpEnvelope | 0..1 | Egress Bandwidth Profile Envelope for the IPVC End Point. The absence of this attribute corresponds to a value of "None". Reference MEF 61.1 Section 11.12 IPVC EP Egress Bandwidth Profile Envelope Service Attribute. Reference MEF 69.1 Section 9.2. [D6] For a Basic Internet Access Service, the egressBandwidthProfileEnvelope SHOULD be empty. |
| ingressBandwidthProfileEnvelope | BasicIaBwpEnvelope | 0..1 | Ingress Bandwidth Profile Envelope for the IPVC End Point. The absence of this attribute corresponds to a value of "None". Reference MEF 61.1 Section 11.11 IPVC EP Ingress Bandwidth Profile Envelope Service Attribute. Reference MEF 69.1 Section 9.2. [D5] For a Basic Internet Access Service, the ingressBandwidthProfileEnvelope SHOULD be empty. |

996 **Table 15 BasicIaIpvcEndPoint**

### 14.3.6    AdvancedIaIpvcEndPoint

998    File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

999    The Advanced Internet Access IPVC End Point is a MEF 69.1 defined version of MEF 61.1 IPVC
1000    End Point. Reference MEF 69.1 Section 9.2 Internet Access IPVC End Point Requirements.

1001    Inherits from: IaIpvcEndPointCommon

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| prefixMapping | Ipv4OrIpv6Prefix[] | 0..* | Indicates which IP Prefixes can send and receive traffic to/from the IPVC. Reference MEF 61.1 Section 11.5 IPVC EP Prefix Mapping Service Attribute. |
| egressBandwidthProfileEnvelope | IpvcEpBwpEnvelope | 0..1 | Egress Bandwidth Profile Envelope for the IPVC End Point. The absence of this attribute corresponds to a value of "None". Reference MEF 61.1 Section 11.12 IPVC EP Egress Bandwidth Profile Envelope Service Attribute. Reference MEF 69.1 Section 9.2. [D6] For a Basic Internet Access Service, the egressBandwidthProfileEnvelope SHOULD be empty. |
| ingressBandwidthProfileEnvelope | IpvcEpBwpEnvelope | 0..1 | Ingress Bandwidth Profile Envelope for the IPVC End Point. The absence of this attribute corresponds to a value of "None". Reference MEF 61.1 Section 11.11 IPVC EP Ingress Bandwidth Profile Envelope Service Attribute. Reference MEF 69.1 Section 9.2. [D5] For a Basic Internet Access Service, the ingressBandwidthProfileEnvelope SHOULD be empty. |

**Table 16 AdvancedIaIpvcEndPoint**

### 14.3.7    EgressClassOfServiceMapUni

File: \ip\common\ipCommon.yaml

Pair of values (D,P). D specifies how to set the DS field in Egress IP Data Packets based on CoS Name. It is either None, or a mapping from CoS Names to DSCP values. P specifies how to set the PCP field in VLAN Tagged Ethernet Frames containing Egress IP Data Packets based on CoS Name. It is either None, or a mapping from CoS Names to PCP values. Reference MEF 61.1 Section 11.10 IPVC EP Egress Class of Service Map Service Attribute. The "Uni" version of this class does not have the "pcpMapping" attribute - MEF 61.1 [R95] The value of P in the IPVC EP Egress Class of Service Map Service Attribute MUST be None unless the following conditions are met: The IPVC EP is at an ENNI using Option A; and All of the ENNI Links in the ENNI have a value for the ENNI Link L2 Technology Attribute (section 16.2) that indicates that VLAN Tagged Ethernet Frames are used to carry IP Packets across the ENNI.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| dscpMapping | DscpMapping[] | 0..* | Reference to CoS to IP DSCP mapping. |

1015                                    **Table 17 EgressClassOfServiceMapUni**

1016    **14.3.8    DscpMapping**

1017    File: \ip\common\ipCommon.yaml

1018    IP DSCP mapping of CoS name to DSCP value Reference MEF 61.1 Section 11.9 IPVC EP Ingress
1019    Class of Service Map Service Attribute, 11.10 IPVC EP Egress Class of Service Map Service
1020    Attribute

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipds | integer | 0..1 | DSCP value |
| cosName | string | 0..1 | Class of Service name |

1021                                          **Table 18 DscpMapping**

1022    **14.3.9    IaIpvcCloud**

1023

1024

1025
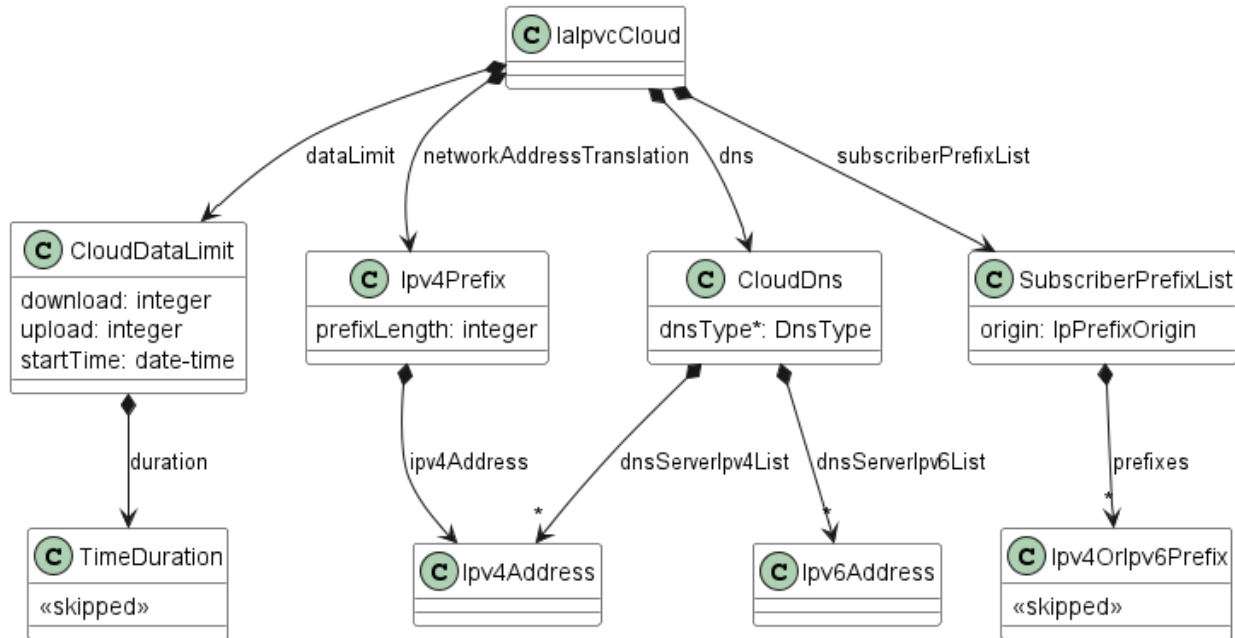
**Figure 16 IaIpvcCloud**

1026    Figure 16 presents a class diagram of IaIpvcCloud

1027    File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

1028    The IPVC Cloud Service Attribute is a set of parameters describing the access connectivity to the
1029    cloud service. Reference MEF 61.1 Section 10.13 IPVC Cloud Service Attribute.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| networkAddressTranslation | Ipv4Prefix | 0..1 | Specifies whether Network Address Translation is used, and if so the IPv4 Prefix. If not selected, then Disabled. Reference MEF 61.1 Section 10.13.4 Cloud Network Address Translation. |
| dataLimit | CloudDataLimit | 0..1 | Limit on the amount of Data traffic sent to/received from the cloud service. Unlimited or a 4-tuple (scdl, Tcdl, ucdl, dcdl). If not provided, then Unlimited. Reference MEF 61.1 Section 10.13.3 Cloud Data Limit. |
| dns | CloudDns | 0..1 | Specifies whether and how DNS is provided for the service. Reference MEF 61.1 Section 10.13.5 Cloud DNS Service. [R12] For a Basic Internet Access Service, Cloud DNS MUST NOT be None. |
| subscriberPrefixList | SubscriberPrefixList | 0..1 | List of public IP Prefixes used in the Subscriber Network and their origin. 2-tuple containing the list of IP Prefixes and the origin of the IP Prefixes. Reference MEF 61.1 Section 10.13.6 Cloud Subscriber Prefix List. Reference MEF 61.1 Section 10.13 IPVC Cloud Service Attribute |

1030 **Table 19 IaIpvcCloud**

### 14.3.10 CloudDataLimit

1032 File: \ip\common\ipCommon.yaml

1033 Specifies an absolute limit on the amount of data the Subscriber can transmit to, or receive from,
1034 the cloud service in a given time period. It is either Unlimited or a 4-tuple (scdl, tcdl, ucdl, dcdl).
1035 Reference MEF 61.1 Section 10.13.3 Cloud Data Limit.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| duration | TimeDuration | 0..1 | Specifies a duration. Together with the start time, it describes a service of contiguous time intervals, starting at the specified start time and each lasting for the specified duration. |
| download | integer | 0..1 | An integer indicating a limit, in octets, on the amount of IP traffic received from the cloud service that can be delivered to the Subscriber during each time interval described by startTime and duration. |
| upload | integer | 0..1 | An integer indicating a limit, in octets, on the amount of IP traffic that can be transmitted towards the cloud service during each time interval described by startTime and duration. |
| startTime | date-time | 0..1 | Specifies a start time. |

1036 **14.3.11 CloudDns**

1037 File: \ip\common\ipCommon.yaml

1038 Data type representing a Domain Name System. Reference MEF 61.1 Sn 10.13.5. Reference MEF
1039 69.1 Section 9.1[R12] For a Basic Internet Access Service, Cloud DNS MUST NOT be NONE.
1040 [R13] For an Internet Access Service, if the Cloud DNS parameter of the IPVC Cloud Service
1041 Attribute is STATIC, the associated list of DNS Servers MUST have at least one entry. [D4] For
1042 an Internet Access Service, if the Cloud DNS parameter of the IPVC Cloud Service Attribute is
1043 STATIC, the associated list of DNS Servers SHOULD contain at least two DNS servers.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| dnsType* | DnsType | 1 | Domain Name System type. |
| dnsServerIpv6List | Ipv6Address[] | 0..* | DNS server list an IPv6 addresses |
| dnsServerIpv4List | Ipv4Address[] | 0..* | DNS server list an IPv4 addresses |

1044 **Table 20 CloudDns**

1045 **14.3.12 DnsType**

1046 File: \ip\common\ipCommon.yaml

1047 Enumeration representing the different types of DNS. Reference MEF 61.1 10.13.5 Cloud DNS
1048 Service

| Value |
| --- |
| NONE |
| DHCP |
| PPP |
| STATIC |
| SLAAC |

**Table 21 DnsType**

### 14.3.13 IpPrefixOrigin

File: \ip\common\ipCommon.yaml

Enumeration of possible values of Ip Prefix Origin.

| Value |
| --- |
| SP |
| OTHER |

**Table 22 IpPrefixOrigin**

### 14.3.14 SubscriberPrefixList

File: \ip\common\ipCommon.yaml

The value of the Cloud Subscriber Prefix List parameter is None or a 2-tuple <prefixes, origin>, where:

- prefixes is a non-empty list of public IP Prefixes that are used in the Subscriber Network, and

- origin is either SP or Other and indicates whether the IP Prefixes are assigned to the Subscriber by the SP or obtainedby the Subscriber from another source. Reference MEF 61.1 Section 10.13.6 Cloud Subscriber Prefix List.Reference MEF 61.1.1 Section 10.13 IPVC Cloud Service Attribute

| Name | Type | Multiplicity | Description |
|------|------|-------------|-------------|
| prefixes | Ipv4OrIpv6Prefix[] | 1..* | Non-empty list of public IP Prefixes that are used in the Subscriber Network |
| origin | IpPrefixOrigin | 0..1 | The origin of the IP Prefixes. Either `SP` or `Other` and indicates whether the IP Prefixes are assigned to the Subscriber by the SP or obtained by the Subscriber from another source. |

1064 **Table 23 SubscriberPrefixList**

1065 ## 14.4　IP UNI



1066
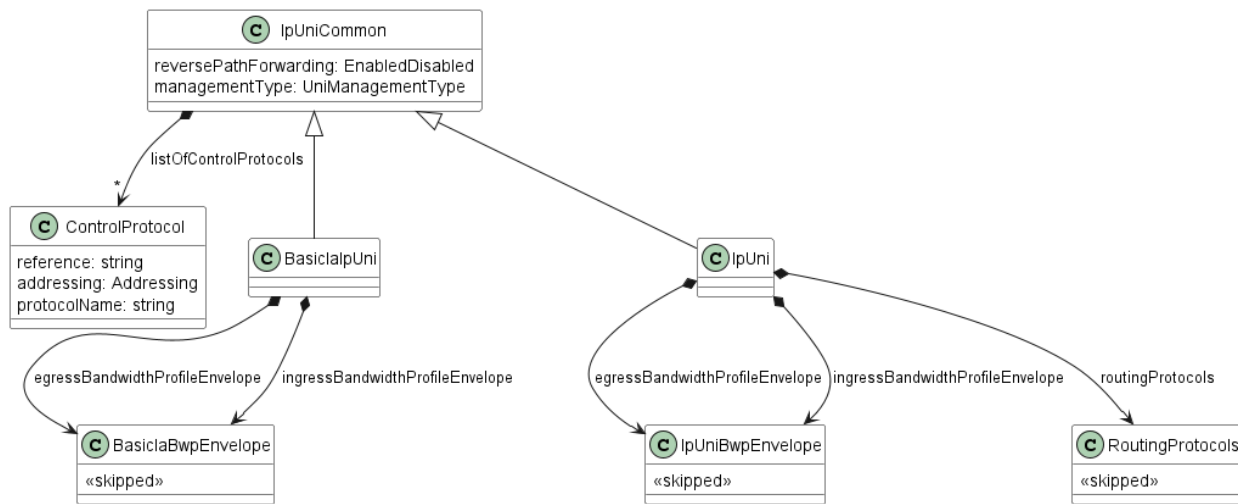
1067 **Figure 17 IP UNI**

1068 Figure 17 Shows the model of the IP UNI and also the differences between the Basic and Advanced
1069 versions.

1070 ### 14.4.1　IpUni

1071 File: \ip\ipUni\ipUni.yaml

1072 URN: urn:mef:lso:spec:cantata-sonata:ip-uni:v0.2.0:all

1073 The IP UNI is a MEF 69.1 defined version of MEF 61.1 UNI. Reference MEF 69.1 Section 8.3
1074 Internet Access UNI Requirements.

1075 Inherits from: IpUniCommon

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| egressBandwidthProfileEnvelope | IpUniBwpEnvelope | 0..1 | Attribute used for an egress UNI Bandwidth Profile. Reference MEF 61.1 Section 12.5 UNI Egress Bandwidth Profile Envelope Service Attribute. Absence of this attribute corresponds to a value of "None". [D8] At a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Egress Bandwidth Profile Envelope is not None, it SHOULD have Bandwidth Profile Flows that contain all Egress IP Data Packets at the UNI that are mapped to any of a given set of IPVC EPs (as defined in MEF 61.1 [8] Table 28). Reference MEF 69.1 Section 9.3 Internet Access UNI Requirements. |
| routingProtocols | RoutingProtocols | 0..1 | List of Routing Protocols used across the UNI. Reference MEF 61.1 Section 12.7 UNI Routing Protocols Service Attribute. Absence of this attribute corresponds to a value of "None". [R21] At a UNI with an IPVC EP for a Basic Internet Access Service, the UNI Routing Protocols list MUST be empty. |
| ingressBandwidthProfileEnvelope | IpUniBwpEnvelope | 0..1 | Attribute used for an ingress UNI Bandwidth Profile. Reference MEF 61.1 Section 12.4 UNI Ingress Bandwidth Profile Envelope Service Attribute. Absence of this attribute corresponds to a value of "None". [D7] At a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Ingress Bandwidth Profile Envelope is not None, it SHOULD have Bandwidth Profile Flows that contain all Ingress IP Data Packets at the UNI that are mapped to any of a given set of IPVC EPs (as defined in MEF 61.1 [8] Table 28). Reference MEF 69.1 Section 9.3 Internet Access UNI Requirements. |

1076 **Table 24 IpUni**

1077 **14.4.2     IpUniCommon**

1078     File: \ip\common\ipCommon.yaml

1079     A User Network Interface (UNI) is the demarcation point between the responsibility of the SP and
1080     the responsibility of the Subscriber. Note that a given UNI always relates to a single SP and a
1081     single Subscriber. Reference MEF 61.1 Section 12. UNI Service Attributes

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| reversePathForwarding | EnabledDisabled | 0..1 | Indicates whether Reverse Path Forwarding checks are used by the SP at the UNI. Reference MEF 61.1 Section 12.8 UNI Reverse Path Forwarding Service Attribute. [D11] At a UNI with an IPVC EP for an Internet Access Service, reversePathForwarding SHOULD be ENABLED. Reference MEF 69.1 Section 9.3 Internet Access UNI Requirements. |
| listOfControlProtocols | ControlProtocol[] | 0..* | Indication of IP Control Protocols that are not forwarded transparently by the SP. Reference MEF 61.1 Section 12.6 UNI List of Control Protocols Service Attribute. Absence of this attribute corresponds to a value of "None". [D9] At a UNI with an IPVC EP for an Internet Access Service, if the UNI has at least one UNI Access Link where the UNI Access Lin IPv4 Connection Addressing is not None, the UNI List of Control Protocols SHOULD include ICMP with a list of applicable ISP IP addresses. [D10] At a UNI with an IPVC EP for an Internet Access Service with at least one UNI Access Link where the UNI Access Link IPv6 Connection Addressing is not None, the UNI List of Control Protocols SHOULD include ICMPv6 with a list of applicable SP IP addresses. Reference MEF 69.1 Section 9.3 Internet Access UNI Requirements. |
| managementType | UniManagementType | 0..1 | Attribute indicating whether the CE is the responsibility of the Subscriber or the Service Provider. Reference MEF 61.1 Section 12.2 UNI Management Type Service Attribute. |

1082                                   **Table 25 IpUniCommon**

1083   ### 14.4.3    BasicIalpUni

1084   File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

1085   The Basic Internet Access IP UNI is a MEF 69.1 defined version of MEF 61.1 IP UNI. Reference
1086   MEF 69.1 Section 9.3 Subscriber Internet Access Service: UNI Requirements

1087   Inherits from: IpUniCommon

| Name | Type | Multiplicity | Description |
|------|------|------|------|
| egressBandwidthProfileEnvelope | BasicIaBwpEnvelope | 0..1 | Attribute used for an egress UNI Bandwidth Profile. Reference MEF 61.1 Section 12.5 UNI Egress Bandwidth Profile Envelope Service Attribute. Absence of this attribute corresponds to a value of "None". [D8] At a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Egress Bandwidth Profile Envelope is not None, it SHOULD have Bandwidth Profile Flows that contain all Egress IP Data Packets at the UNI that are mapped to any of a given set of IPVC EPs (as defined in MEF 61.1 [8] Table 28). Reference MEF 69.1 Section 9.3 Internet Access UNI Requirements. |
| ingressBandwidthProfileEnvelope | BasicIaBwpEnvelope | 0..1 | Attribute used for an ingress UNI Bandwidth Profile. Reference MEF 61.1 Section 12.4 UNI Ingress Bandwidth Profile Envelope Service Attribute. Absence of this attribute corresponds to a value of "None". [D7] At a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Ingress Bandwidth Profile Envelope is not None, it SHOULD have Bandwidth Profile Flows that contain all Ingress IP Data Packets at the UNI that are mapped to any of a given set of IPVC EPs (as defined in MEF 61.1 [8] Table 28). Reference MEF 69.1 Section 9.3 Internet Access UNI Requirements. |

1088                                     **Table 26 BasicIaIpUni**

1089    **14.4.4    ControlProtocol**

1090    File: \ip\common\ipCommon.yaml

1091    Data type representing Control Protocol. Each entry consists of a 3-tuple containing the protocol
1092    name, addressing information (either SP/Operator Addresses or Any) and one or more references.
1093    Reference MEF 61.1 Section 12.6 UNI List of Control Protocols Service Attribute

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| reference | string[] | 1..* | Protocol reference. |
| addressing | Addressing | 0..1 | Enumeration representing the addressing. |
| protocolName | string | 0..1 | Protocol name. |

1094                                 **Table 27 ControlProtocol**

1095 **14.4.5   Addressing**

1096 File: \ip\common\ipCommon.yaml

1097 Enumeration representing the Address type for the Control Protocols data
1098 type. Reference MEF 61.1 Section 12.6 UNI List of Control Protocols
1099 Service Attribute

1100 • SP_OPERATOR_ADDRESSES: If the addressing information is SP/Operator Addresses,
1101 then Ingress IP Packets for the specified protocol that have a multicast or broadcast
1102 destination address, or a unicast destination address that is reachable within the SP's or
1103 Operator's network, are considered to be IP Control Protocol Packets, and Egress IP
1104 Packets for the specified protocol that have a source address that is reachable within the
1105 SP's or Operator's network are considered to be IP Control Protocol Packets.

1106 • ANY: If the addressing information is Any, then all IP Packets for the specified protocol
1107 that cross the UNI are considered to be IP Control Protocol Packets.

| Value |
|-------|
| SP_OPERATOR_ADDRESSES |
| ANY |

1108                                 **Table 28 Addressing**

1109 **14.4.6   UniManagementType**

1110 File: \ip\common\ipCommon.yaml

1111 Enumeration representing the UNI Management Type options. Reference MEF 61.1 Section 12.2
1112 UNI Management Type Service Attribute.

| Value |
|-------|
| SUBSCRIBER_MANAGEMENT |
| PROVIDER_MANAGEMENT |

1113                                    **Table 29 UniManagementType**

## 14.5     IP UNI Access Link

1114



1115
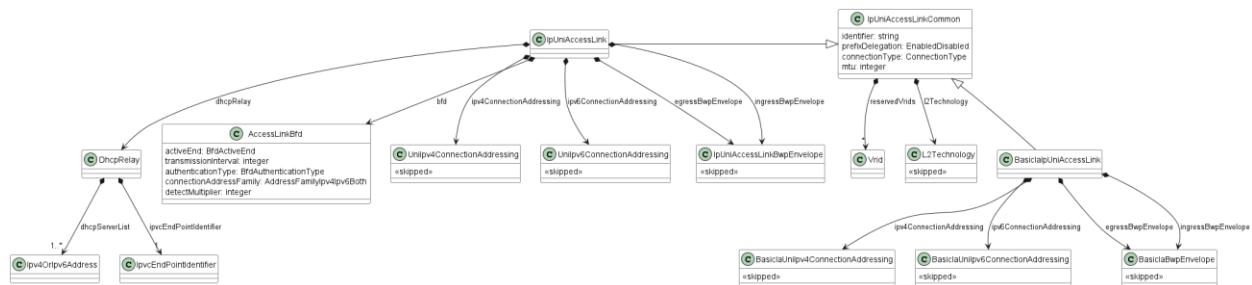
1116                                    **Figure 18 IP UNI Access Link**

1117    Figure 18 depicts the model of Basic and Advanced IP UNI Access Links and differences thei
1118    dofferences.

### 14.5.1     IpUniAccessLinkCommon

1119

1120    File: \ip\common\ipCommon.yaml

1121    An individual connection between the Subscriber and the SP that forms part of a UNI. Reference
1122    MEF 61.1 Section 7.3 UNIs and UNI Access Link.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| reservedVrids | Vrid[] | 0..* | List of VRRP (Virtual Router Redundancy Protocol) VRIDs (Virtual Router Identifier) reserved for use by the SP or Operator. Reference MEF 61.1 Section 13.12 UNI Access Link Reserved VRIDs Service Attribute. |
| identifier | string | 0..1 | IPVC End Point identifier as described in MEF 61.1 Section 11.1. Note - it is not the same thing as the potential Product identifier if IPVC Endpoint is an instance of a Product. |
| l2Technology | L2Technology | 0..1 | Specifies the UNI Access Link Trunk (61.1.1 section A1-1) used to carry IP Packets across the UNI along with information needed to identify IP Packets for this UNI Access Link. |
| prefixDelegation | EnabledDisabled | 0..1 | Indicates whether DHCP Prefix delegation is enabled. Reference MEF 61.1 Section 13.7 UNI Access Link Prefix Delegation Service Attribute. |
| connectionType | ConnectionType | 0..1 | Indicates whether the UNI Access Link is point-to-point or multipoint. |
| mtu | integer | 0..1 | Maximum size, in octets of an IP Packet that can traverse the UNI Access Link. Reference MEF 61.1 Section 13.9 UNI Access Link IP MTU Service Attribute. |

1123 **Table 30 IpUniAccessLinkCommon**

1124 **14.5.2 IpUniAccessLink**

1125 File: schema\productSchema\ip\ipUni\ipUniAccessLink.yaml

1126 URN: urn:mef:lso:spec:cantata-sonata:ip-uni-access-link:v0.2.0:all

1127 An individual connection between the Subscriber and the SP that forms part of a UNI. Reference
1128 MEF 61.1 Section 7.3 UNIs and UNI Access Link. Inherits from: IpUniAccessLinkCommon

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| dhcpRelay | DhcpRelay | 0..1 | Indicates whether DHCP Relay functionality is enabled. Reference MEF 61.1 Section 13.6 UNI Access Link DHCP Relay Service Attribute. Absence of this attribute corresponds to a value of "Disabled". |
| bfd | AccessLinkBfd | 0..1 | Indication of whether BFD is used on the IpServicesExternalInterfaceLink. Reference MEF 61.1 Section 16.5 ENNI Link BFD Attribute and MEF 61.1 Section 13.8 UNI Access Link BFD Service Attribute. |
| ipv4ConnectionAddressing | UniIpv4ConnectionAddressing | 0..1 | IPv4 Connection Addressing. Reference MEF 61.1 Section 13.4 UNI Access Link IPv4 Connection Addressing Service Attribute. Absence of this attribute corresponds to a value of "None". |
| ipv6ConnectionAddressing | UniIpv6ConnectionAddressing | 0..1 | IPv6 Connection Addressing. Reference MEF 61.1 Section 13.5 UNI Access Link IPv6 Connection Addressing Service Attribute. Absence of this attribute corresponds to a value of "None". |
| egressBwpEnvelope | IpUniAccessLinkBwpEnvelope | 0..1 | Egress Bandwidth Profile Envelope for the UNI Access Link. Reference MEF 61.1 Section 13.11 UNI Access Link Egress Bandwidth Profile Envelope Service Attribute. Absence of this attribute corresponds to a value of "None". |
| ingressBwpEnvelope | IpUniAccessLinkBwpEnvelope | 0..1 | Ingress Bandwidth Profile Envelope for the UNI Access Link. Reference MEF 61.1 Section 13.10 UNI Access Link Ingress Bandwidth Profile Envelope Service Attribute. Absence of this attribute corresponds to a value of "None". |

1129                                  **Table 31 IpUniAccessLink**

1130   **14.5.3   BasicIapUniAccessLink**

1131   File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

1132   The Basic Internet Access UNI Access Link is a MEF 69.1 defined version of MEF 61.1 UNI

1133   Access Link. Reference MEF 69.1 Section 9.4 Internet Access UNI Access Link Requirements.

1134   Inherits from: IpUniAccessLinkCommon

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipv4ConnectionAddressing | BasicIaUniIpv4ConnectionAddressing | 0..1 | IPv4 Connection Addressing. Reference MEF 61.1 Section 13.4 UNI Access Link IPv4 Connection Addressing Service Attribute. Absence of this attribute corresponds to a value of "None". |
| ipv6ConnectionAddressing | BasicIaUniIpv6ConnectionAddressing | 0..1 | IPv6 Connection Addressing. Reference MEF 61.1 Section 13.5 UNI Access Link IPv6 Connection Addressing Service Attribute. Absence of this attribute corresponds to a value of "None". |
| egressBwpEnvelope | BasicIaBwpEnvelope | 0..1 | Egress Bandwidth Profile Envelope for the UNI Access Link. Reference MEF 61.1 Section 13.11 UNI Access Link Egress Bandwidth Profile Envelope Service Attribute. Absence of this attribute corresponds to a value of "None". |
| ingressBwpEnvelope | BasicIaBwpEnvelope | 0..1 | Ingress Bandwidth Profile Envelope for the UNI Access Link. Reference MEF 61.1 Section 13.10 UNI Access Link Ingress Bandwidth Profile Envelope Service Attribute. Absence of this attribute corresponds to a value of "None". |

1135 **Table 32 BasicIaIpUniAccessLink**

1136 ### 14.5.4 AccessLinkBfd

1137 File: \ip\common\ipCommon.yaml

1138 The Access Link BFD Service Attribute indicates whether Bidirectional Forwarding Detection
1139 (BFD) is enabled on the UNI Access Link. Reference MEF 61.1 Section 13.8 UNI Access Link
1140 BFD Service Attribute and Section 16.5 ENNI Link BFD Attribute.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| activeEnd | BfdActiveEnd | 0..1 | BFD Active End. At least one end of BFD session has to have an active role, meaning that it sends out asynchronous control messages regardless of whether it has received any. Reference MEF 61.1 Section 13.8 UNI Access Link BFD Service Attribute. |
| transmissionInterval | integer | 0..1 | Transmission Interval Reference MEF 61.1 Section 13.8 UNI Access Link BFD Service Attribute and Section 16.5 ENNI Link BFD Attribute. |
| authenticationType | BfdAuthenticationType | 0..1 | BFD Authentication as describer in RFC 5880. Reference MEF 61.1 Section 13.8 UNI Access Link BFD Service Attribute and Section 16.5 ENNI Link BFD Attribute. |
| connectionAddressFamily | AddressFamilyIpv4Ipv6Both | 0..1 | The Connection Address Family parameter specifies whether the session is established over IPv4 or IPv6 or whether two separate sessions are established using IPv4 and IPv6. Reference MEF 61.1 Section 13.8 UNI Access Link BFD Service Attribute and Section 16.5 ENNI Link BFD Attribute. |
| detectMultiplier | integer | 0..1 | BFD Detect multiple as an Integer. Reference MEF 61.1 Section 13.8 UNI Access Link BFD Service Attribute and Section 16.5 ENNI Link BFD Attribute. |

1141
**Table 33 AccessLinkBfd**

1142 **14.5.5    AddressFamilyIpv4Ipv6Both**

1143    File: \ip\common\ipCommon.yaml

1144    Specifies whether the session is established over IPv4 or IPv6 or whether two separate session are
1145    established using IPv4 and IPv6.

| Value |
|-------|
| IPV4 |
| IPv6 |
| BOTH |

1146 **Table 34 AddressFamilyIpv4Ipv6Both**

### 14.5.6 BfdActiveEnd

1148 File: \ip\common\ipCommon.yaml

1149 At least one end of the BFD session must have an active role, meaning that it sends out
1150 asynchronous control messages regardless of whether it has received any. This enumeration
1151 represents the values that can be set for the BFD Active End. Reference MEF 61.1 Section 13.8
1152 UNI Access Link BFDService Attribute [R171] and [R172].

1153 SUBSCRIBER: Subscriber takes active BFD role.

1154 SP: Service Provider takes active BFD role.

1155 BOTH: Subscriber and Service Provider take active BFD role.

| Value |
|-------|
| SUBSCRIBER |
| SP |
| BOTH |

1156 **Table 35 BfdActiveEnd**

### 14.5.7 BfdAuthenticationType

1158 File: \ip\common\ipCommon.yaml

1159 RFC 7419 specifies a set of common intervals which are used to ensure interoperability.

1160 • NONE: No BFD authentication.

- SIMPLE_PASSWORD: Simple Password Authentication is the most straightforward (and weakest) form of authentication. In this method of authentication one or more Passwords (with corresponding Key IDs) are configured in each system and one of these Password/ID pairs is carried in each BFD Control packet. The receiving system accepts the packet if the Password and Key ID matches one of the Password/ID pairs configured in that system. Reference IETF RFC5880 Section 6.7.2.

- KEYED_MD5: The Keyed MD5 and Meticulous Key MD5 Authentication mechanisms are very similar to those used in other protocols. In these methods of authentication, one or more security keys (with corresponding key IDs) are configured in each system. Reference RFC5880 Section 6.7.3 Keyed MD5 and Meticulous Keyed MD5 Authentication.

- METICULOUS_KEYED_MD5: The Keyed MD5 and Meticulous Key MD5 Authentication mechanisms are very similar to those used in other protocols. In these methods of authentication, one or more security keys (with corresponding key IDs) are configured in each system. Reference RFC5880 Section 6.7.3 Keyed MD5 and Meticulous Keyed MD5 Authentication.

- KEYED_SHA1: The Keyed SHA1 and Meticulous Key SHA1 Authentication mechanisms are very similar to those used in other protocols. In these methods of authentication, one or more secret keys (with corresponding key IDs) are configured in each system. Reference RFC5880 Section 6.7.4 Keyed SHA1 and Meticulous Keyed SHA1 Authentication.

- METICULOUS_KEYED_SHA1: The Keyed SHA1 and Meticulous Key SHA1 Authentication mechanisms are very similar to those used in other protocols. In these methods of authentication, one or more secret keys (with corresponding key IDs) are configured in each system. Reference RFC5880 Section 6.7.4 Keyed SHA1 and Meticulous Keyed SHA1 Authentication.

| Value |
| --- |
| NONE |
| SIMPLE_PASSWORD |
| KEYED_MD5 |
| METICULOUS_KEYED_MD5 |
| KEYED_SHA1 |
| METICULOUS_KEYED_SHA1 |

1186                                    **Table 36 BfdAuthenticationType**

1187

1188    **14.5.8    ConnectionType**

1189    File: \ip\common\ipCommon.yaml

1190    An enumeration representing the connection type.

1191    • POINT_TO_POINT indicates that the link is logically point to Point.

1192    • MULTIPOINT indicates the link is logically multipoint.

| Value |
| --- |
| POINT_TO_POINT |
| MULTIPOINT |

1193                                    **Table 37 ConnectionType**

1194    **14.5.9    DhcpRelay**

1195    File: \ip\common\ipCommon.yaml

1196    Dynamic Host Configuration Protocol (DHCP) Relay functionality is useful when the Subscriber
1197    uses DHCP (per RFC 2131 and RFC 8415) in the Subscriber Network but does not want to place

1198 a DHCP server (or possibly a pair of redundant DHCP servers) in each part of the network.
1199 Reference MEF 61.1 Section UNI Access Link DHCP Relay Service Attribute

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| dhcpServerList | Ipv4OrIpv6Address[] | 1..1 | Non-empty list of IP addresses for DHCP Servers belonging to the Subscriber. Reference MEF 61.1 Section 13.6 UNI Access Link DHCP Relay Server Attribute. |
| ipvcEndPointIdentifier | IpvcEndPointIdentifier | 0..1 | IPVC End Point identifier as described in MEF 61.1 Section 11.1. Note - It points to the value of IpvcEndPoint.identifier Service Attribute. It is not intended to point to the potential Product identifier if IPVC Endpoint is an instance of a Product. |

1200 **Table 38 DhcpRelay**

1201 oneOf:

1202   - required: [dhcpServerList]

1203   - required: [ipvcEndPointIdentifier]

1204 **14.5.10   Vrid**

1205 File: \ip\common\ipCommon.yaml

1206 Data type definition: VRID (Virtual Router ID) as defined in RFC 5798 is a number between 1
1207 and 255

1208 **14.5.11   BasicIaUniIpv4ConnectionAddressing**

1209 File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml



1210
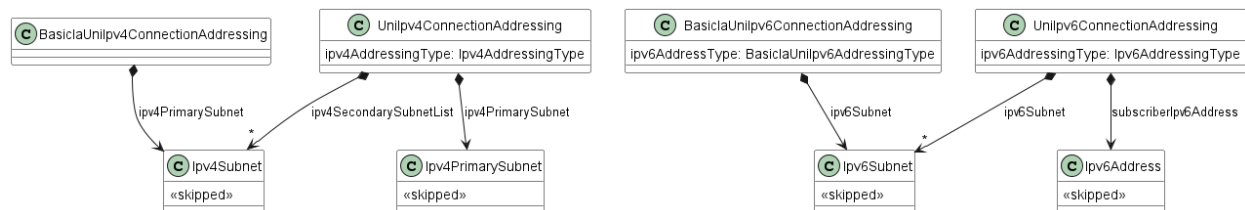
1211 **Figure 19 IPV4 and IPV6 Connection Addressing**

1212 Figure 19 shows the both IPv4 anf IPv6 versions of Connection Addressing.

1213 Represents how IPv4 addresses are allocated to the devices on the UNI Access Link in case of
1214 Basic Internet Access. Reference MEF 61 Section 13.4 UNI Access Link IPv4 Connection

1215 Addressing Service Attribute. [R23] At a UNI Access Link in a UNI with an IPVC EP for a Basic
1216 Internet Access Service, UNI Access Link IPv4 Connection Addressing MUST be DHCP or None.
1217 [R25] If IPv4 Connection Addressing is DHCP, the UNI Access Link IPv4 Connection Addressing
1218 Primary Subnet parameter MUST contain only a single Service Provider IPv4 Address. Reference
1219 MEF 69.1[1] Section 9.4 Subscriber Internet Access Service: UNI Access Link Requirements.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| ipv4PrimarySubnet | Ipv4Subnet | 0..1 | Primary IPv4 Subnet. Includes IPv4 Prefix and Service Provider IPv4 Addresses. [R25] If IPv4 Connection Addressing is DHCP, the UNI Access Link IPv4 Connection Addressing Primary Subnet parameter MUST contain only a single Service Provider IPv4 Address. Reference MEF 69.1[1] Section 9.4 Subscriber Internet Access Service: UNI Access Link Requirements |

1220 **Table 39 BasicIaUniIpv4ConnectionAddressing**

### 14.5.12    UniIpv4ConnectionAddressing

1221

1222 File: \ip\common\ipCommon.yaml

1223 UniIpv4ConnectionAddressing is a data type representing how IPv4 addresses are allocated to the
1224 devices on the UNI Access Link. Reference MEF 61 Section 13.4 UNI Access Link IPv4
1225 Connection Addressing Service Attribute. [R22] At a UNI Access Link in a UNI with an IPVC EP
1226 for an Advanced Internet Access Service, UNI Access Link IPv4 Connection Addressing MUST
1227 be Static or None. Reference MEF 69.1 Section 9.4

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| ipv4AddressingType | Ipv4AddressingType | 0..1 | IPv4 Connection Addressing. |
| ipv4SecondarySubnetList | Ipv4Subnet[] | 0..* | Secondary IPv4 Subnet List. Includes IPv4 Prefix and Service Provider IPv4 Addresses. |
| ipv4PrimarySubnet | Ipv4PrimarySubnet | 0..1 | Primary IPv4 Subnet. Includes IPv4 Prefix and Service Provider IPv4 Addresses. |

1228 **Table 40 UniIpv4ConnectionAddressing**

### 14.5.13    BasicIaUniIpv6ConnectionAddressing

1229

1230 File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

1231 Represents how IPv6 addresses are allocated to the devices on the UNI Access Link in case of
1232 Basic Internet Access. Reference MEF 61 Section 13.5 UNI Access Link IPv6 Connection

1233  Addressing Service Attribute. [R27] At a UNI Access Link in a UNI with an IPVC EP for a Basic
1234  Internet Access Service, UNI Access Link IPv6 Connection Addressing MUST be DHCP or
1235  SLAAC or None. Reference MEF 69.1[1] Section 9.4 Subscriber Internet Access Service: UNI
1236  Access Link Requirements. [R29] At a UNI Access Link in a UNI with an IPVC EP for a Basic
1237  Internet Access Service, if the UNI Ac-cess Link IPv6 Connection Addressing is DHCP or
1238  SLAAC, the UNI Access Link IPv6 Connection Ad-dressing Subnet List parameter MUST contain
1239  only a single Service Provider IPv6 Address.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipv6AddressType | BasicIaUniIpv6AddressingType | 0..1 | Basic Internet Access IPv6 Connection Address mechanism. |
| ipv6Subnet | Ipv6Subnet | 0..1 | IPv6 Subnet. [R29] At a UNI Access Link in a UNI with an IPVC EP for a Basic Internet Access Service, if the UNI Ac-cess Link IPv6 Connection Addressing is DHCP or SLAAC, the UNI Access Link IPv6 Connection Ad-dressing Subnet List parameter MUST contain only a single Service Provider IPv6 Address. |

1240                      **Table 41 BasicIaUniIpv6ConnectionAddressing**

1241  **14.5.14    UniIpv6ConnectionAddressing**

1242  File: \ip\common\ipCommon.yaml

1243  UniIpv6ConnectionAddressing is a data type representing how IPv6 addresses are allocated to the
1244  devices on the UNI Access Link. Reference MEF 61 Section 13.5 UNI Access Link IPv6
1245  Connection Addressing Service Attribute. [R26] At a UNI Access Link in a UNI with an IPVC EP
1246  for an Advanced Internet Access Service, UNI Access Link IPv6 Connection Addressing MUST
1247  be Static or None. Reference MEF 69.1 Section 9.4

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipv6AddressingType | Ipv6AddressingType | 0..1 | IPv6 Connection Addressing. |
| ipv6Subnet | Ipv6Subnet[] | 0..* | Ipv6 Subnet |
| subscriberIpv6Address | Ipv6Address | 0..1 | Subscriber IPv6 address. |

1248                        **Table 42 UniIpv6ConnectionAddressing**

1249  **14.5.15    Ipv4AddressingType**

1250  File: \ip\common\ipCommon.yaml

1251 Enumeration representing IPv4 Address Types specific for UNI Access Links.

1252 • DHCP: Dynamic Host Configuration Protocol (DHCP) is used the Subscriber devices to
1253    request IPv4 addresses in a given subnet from the SP or Operator.

1254 • STATIC: IPv4 addresses in a given IPv4 subnet are statically assigned to the SP or
1255    Operator and to the Subscriber.

1256 • UNNUMBERED: The SP or Operator and the Subscriber each assigned an IPv4 address
1257    (from their own address pools) independently. These addresses can be on different subnets,
1258    and so an interface-based routing protocol is needed to ensure reachability.

| Value |
| --- |
| DHCP |
| STATIC |
| UNNUMBERED |

1259                         **Table 43 Ipv4AddressingType**

1260 **14.5.16   BasicIaUniIpv6AddressingType**

1261 File: \ip\internetAccess\internetAccessCommon\internetAccessCommon.yaml

1262 Enumeration representing IPv6 Address Types specific for UNI Access Links.

1263 • DHCP: Dynamic Host Configuration Protocol (DHCP) is used by the Subscriber devices
1264    to request IPv6 addresses in a given subnet from the SP or Operator.

1265 • SLAAC: Stateless Address Autoconfiguration (SLAAC) is used by the Subscriberdevices
1266    to create unique IPv6 global addresses within an IP Prefix advertised by the SP or Operator
1267    as describer in RFC 4862.

| Value |
| --- |
| DHCP |
| SLAAC |

1268                   **Table 44 BasicIaUniIpv6AddressingType**

1269 **14.5.17    Ipv6AddressingType**

1270    File: \ip\common\ipCommon.yaml

1271    Ipv6AddressingType

1272    Enumeration representing IPv6 Address Types specific for UNI Access Links.

1273    • DHCP: Dynamic Host Configuration Protocol (DHCP) is used by the Subscriber devices
1274       to request IPv6 addresses in a given subnet from the SP or Operator.

1275    • SLAAC: Stateless Address Autoconfiguration (SLAAC) is used by the Subscriber devices
1276       to create unique IPv6 global addresses within an IP Prefix advertised by the SP or Operator
1277       as describer in RFC 4862.

1278    • STATIC: IPv6 addresses in a given IPv6 subnet are statically assigned to the SP or
1279       Operator and to the Subscriber.

1280    • LL_ONLY: If the value is LL-only, these are only IPv6 addresses used onthe UNI Access
1281       Link.

| Value |
|-------|
| DHCP |
| SLAAC |
| STATIC |
| LL_ONLY |

1282                    **Table 45 Ipv6AddressingType**

1283 **14.5.18    L2Technology**
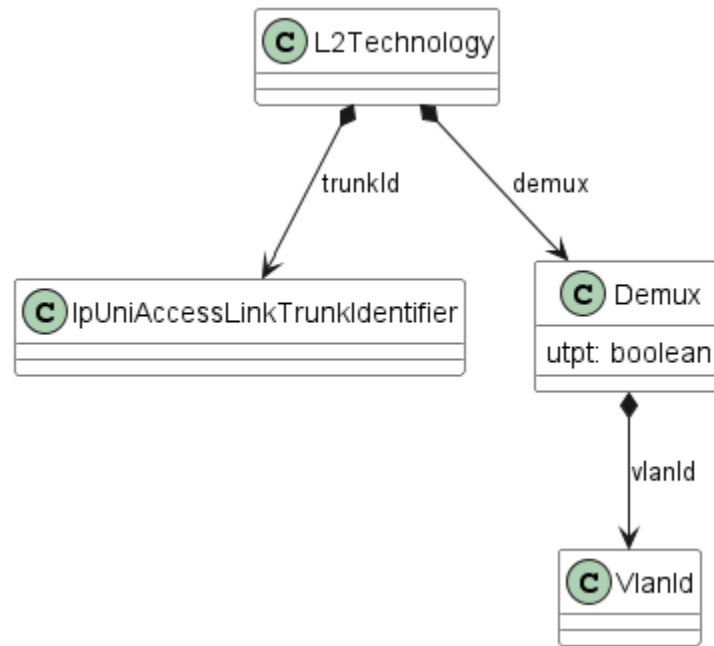


1284

1285                                **Figure 20 L2Technology**

1286    Figure 20 presents the diagram of L2 Technology Service Attribute.

1287    File: \ip\common\ipCommon.yaml

1288    Specifies the UNI Access Link Trunk used to carry IP Packets across the UNI along with
1289    information needed to identify IP Packets for this UNI Access Link. Reference MEF 61.1.1 Section
1290    13.3 UNI Access Link Trunk Service Attributes

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| trunkId | IpUniAccessLinkTrunkIdentifier | 0..1 | UNI Access Link Trunk Identifier. Reference MEF 61.1.1 described in section A1-1 |
| demux | Demux | 0..1 | Value that is specific to each type of UNI Access Link Trunk and indicates which Layer 2 sub-channel should be selected for this UNI Access Link1 |

1291                                **Table 46 L2Technology**

1292    **14.5.19    IpUniAccessLinkTrunkIdentifier**

1293    File: \ip\common\ipCommon.yaml

1294 IPVC End Point identifier as described in MEF 61.1 Section 11.1. Note - It points to the value of
1295 IpvcEndPoint.identifier Service Attribute. It is not intended to point to the potential Product
1296 identifier if IPVC Endpoint is an instance of a Product.

1297 maxLength: 53

1298 pattern: "[\x20-\x7F]+"

### 14.5.20    Demux

1300 File: \ip\common\ipCommon.yaml

1301 Is a value that is specific to each type of UNI Access Link Trunk and indicates which Layer 2 sub-
1302 channel should be select for this UNI Access Link. [A1-R8] If the UNI Access Link Trunk
1303 identified by trunkID is of type ETHERNET, then the value of the demux element MUST be either
1304 UT/PT or a VLAN ID in the range 1 to 4094.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| utpt | boolean | 0..1 | Untagged and priority tagged frames. |
| vlanId | VlanId | 0..1 | VLAN ID. |

1305                                                              **Table 47 Demux**

### 14.5.21    VlanId

1307 File: \ip\common\ipCommon.yaml

1308 Data type used for VLAN id configuration. Defined as a Integer. Value 1 to 4094.

1309 **14.6 Ethernet Uni Access Link Trunk**
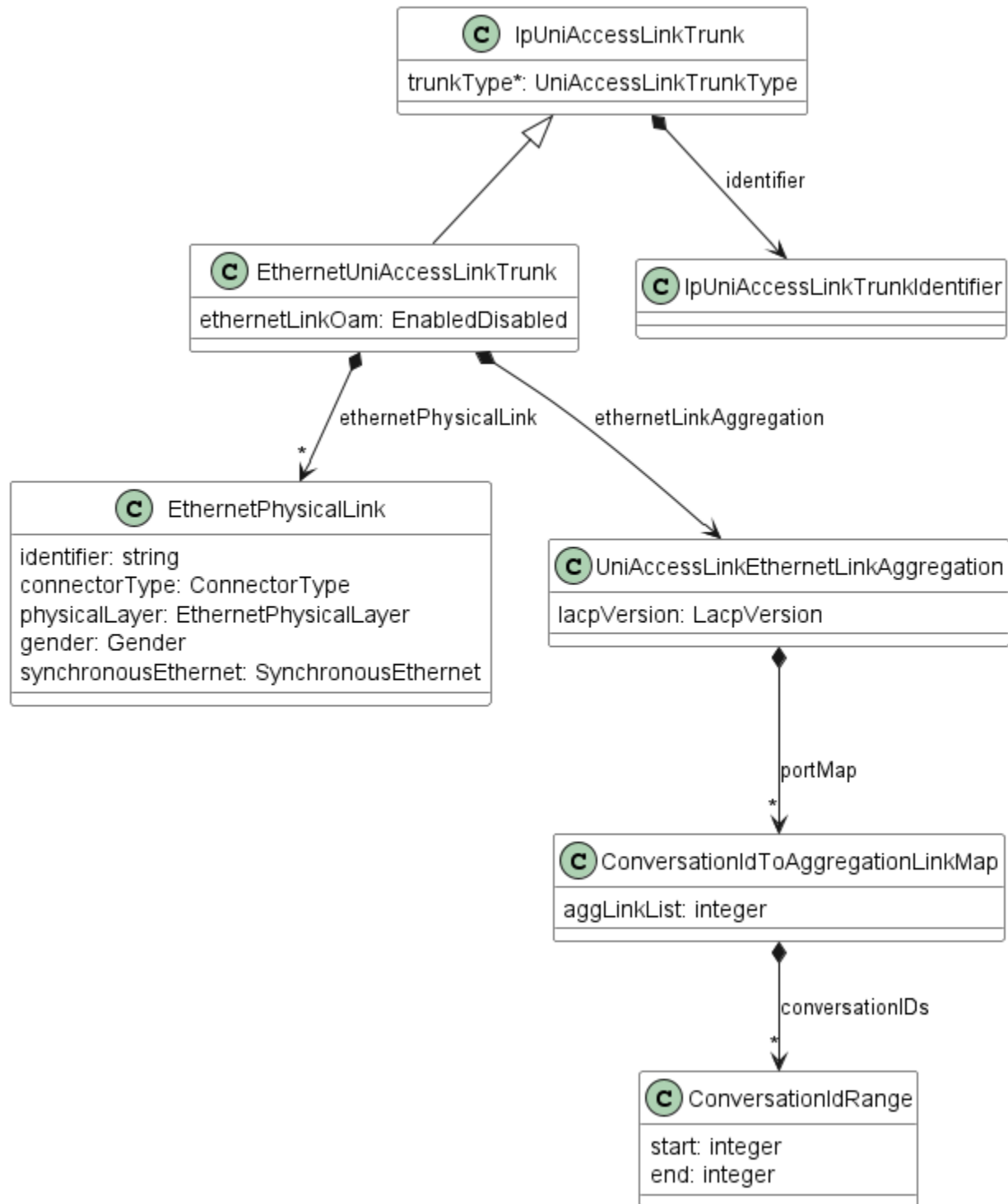


1310

1311 **Figure 21 EthernetUniAccessLinkTrunk**

1312 Figure 21 Shows the diagram of Thernet Uni Access Link Trunk. It is the only specified sublass
1313 of an abstract class IP UNI Access Link Trunk. It is used by all 3 Internet Access Product flavors
1314 without any changes.

### 14.6.1    IpUniAccessLinkTrunk

1316 File: \ip\common\ipCommon.yaml

1317 A UNI Access Link Trunk is a construct that encapsulates the details of Layer 1 and Layer 2
1318 configuration shared by one or more UNI Access Links. Reference MEF 61.1.1 Section A1-1 UNI
1319 Access Link Trunk Service Attributes.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| identifier | IpUniAccessLinkTrunkIdentifier | 0..1 | Unique identifier for the UNI Access Link Trunk for management purposes. Reference MEF 61.1.1 Section A1-1.1 UNI Access Link Trunk Identifier Service Attribute. |
| trunkType* | UniAccessLinkTrunkType | 0..1 | The type of Layer 2 technology of the UNI Access Link Trunk |

1320                                **Table 48 IpUniAccessLinkTrunk**

### 14.6.2    EthernetUniAccessLinkTrunk

1322 File: \ip\ipUni\ethernetUniAccessLinkTrunk.yaml

1323 URN: urn:mef:lso:spec:cantata-sonata:ethernet-uni-access-link-trunk:v0.2.0:all

1324 A single point-to-point physical Ethernet channel or multiple physical Ethernet links combined
1325 into a Link Aggregation Group. The Ethernet frames associated with a given UNI Access Link can
1326 be either untagged/priority-tagged or VLAN tagged. Reference MEF 61.1.1 A1-1.3 Ethernet UNI
1327 Access Link Trunk Service Attributes

1328 Inherits from: IpUniAccessLinkTrunk

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ethernetPhysicalLink | EthernetPhysicalLink[] | 1..* | A list of the physical link types along with some additional capabilities |
| ethernetLinkAggregation | UniAccessLinkEthernetLinkAggregation | 0..1 | Configuration of Link Aggregation for the UNI Access Link Trunk |
| ethernetLinkOam | EnabledDisabled | 0..1 | Indicates whether Link OAM is used on the UNI Access Link Trunk |

1329                                   **Table 49 EthernetUniAccessLinkTrunk**

### 14.6.3    UniAccessLinkTrunkType

1331    File: \ip\common\ipCommon.yaml

1332    Enumeration representing the UNI Access Link Trunk Type Service Attribute. MEF 61.1 Specifies
1333    value of ETHERNET and OTHER. Yet since OTHER means to practical implementation, only
1334    ETHERNET is used. The enumeration is remained though for future enhancements.

| Value |
|-------|
| ETHERNET |

1335                                   **Table 50 UniAccessLinkTrunkType**

### 14.6.4    EthernetPhysicalLink

1337    File: \ip\common\ipCommon.yaml

1338    Data type representing UNI Access Link Trunk List of Ethernet Physical Links of form
1339    <id,pl,fs,ct,gn> as defined in MEF 61.1.1 Section A1-1.3.1.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| identifier | string | 0..1 | Identifier of the Physical LInk |
| connectorType | ConnectorType | 0..1 | Enumeration representing type of connector presented to Subscriber. |
| physicalLayer | EthernetPhysicalLayer | 0..1 | Enumeration representing the different Ethernet physical layers. Reference MEF 61.1.1 Table A1-4 Ethernet PHYs for UNI Access Link Trunks. |
| gender | Gender | 0..1 | Enumeration representing the gender of the connector presented to the Subscriber. |
| synchronousEthernet | SynchronousEthernet | 0..1 | Enumeration indicating if the physical link supports Synchronous Ethernet. |

1340 **Table 51 EthernetPhysicalLinkEthernetPhysicalLink**

1341 **14.6.5    ConnectorType**

1342    File: \ip\common\ipCommon.yaml

1343    Enumeration representing type of connector presented to Subscriber.

| Value |
|---|
| RJ45 |
| SC |
| LC |
| OTHER |

1344 **Table 52 ConnectorTypeConnectorType**

1345 **14.6.6    EthernetPhysicalLayer**

1346    File: \ip\common\ipCommon.yaml

1347    Enumeration representing the different Ethernet physical layers. Reference MEF 61.1.1 Table A1-
1348    4 Ethernet PHYs for UNI Access Link Trunks.

| | | |
|---|---|---|
| 10BASE_FB | 10BASE_FL | 10BASE_FP |
| 10BASE_T | 10BASE_T1L | 10BASE_T1S |
| 10BASE_TE | 10BROAD36 | 10PASS_TS |
| 100BASE_BX10 | 100BASE_FX | 100BASE_LX10 |
| 100BASE_T | 100BASE_T1 | 100BASE_T2 |
| 100BASE_T4 | 100BASE_TX | 100BASE_X |
| 1000BASE_BX10 | 1000BASE_CX | 1000BASE_LX |
| 1000BASE_LX10 | 1000BASE_PX10 | 1000BASE_PX20 |
| 1000BASE_RHA | 1000BASE_RHB | 1000BASE_RHC |
| 1000BASE_SX | 1000BASE_T | 1000BASE_T1 |
| 1000BASE_X | 2_5GBASE_T | 2_5GBASE_T1 |
| 5GBASE_T | 5GBASE_T1 | 10GBASE_CX4 |
| 10GBASE_E | 10GBASE_ER | 10GBASE_EW |
| 10GBASE_L | 10GBASE_LR | 10GBASE_LRM |
| 10GBASE_LW | 10GBASE_LX4 | 10GBASE_R |
| 10GBASE_S | 10GBASE_SR | 10GBASE_SW |
| 10GBASE_T | 10GBASE_T1 | 10GBASE_W |
| 10GBASE_X | 25GBASE_CR | 25GBASE_CR_S |
| 25GBASE_ER | 25GBASE_LR | 25GBASE_SR |
| 25GBASE_T | 40GBASE_CR4 | 40GBASE_ER4 |
| 40GBASE_FR | 40GBASE_LR4 | 40GBASE_R |
| 40GBASE_SR4 | 40GBASE_T | 50GBASE_CR |
| 50GBASE_ER | 50GBASE_FR | 50GBASE_LR |
| 50GBASE_SR | 100GBASE_CR10 | 100GBASE_CR2 |
| 100GBASE_CR4 | 100GBASE_DR | 100GBASE_ER4 |
| 100GBASE_LR4 | 100GBASE_R | 100GBASE_SR10 |
| 100GBASE_SR2 | 100GBASE_SR4 | 200GBASE_CR4 |
| 200GBASE_DR4 | 200GBASE_ER4 | 200GBASE_FR4 |
| 200GBASE_LR4 | 200GBASE_SR4 | 400GBASE_DR4 |

| 400GBASE_ER8 | 400GBASE_FR8 | 400GBASE_LR8 |
| --- | --- | --- |
| 400GBASE_SR16 | 400GBASE_SR4_2 | 400GBASE_SR8 |

1349                                  **Table 53 EthernetPhysicalLayer**

### 14.6.7    Gender

1351    File: \ip\common\ipCommon.yaml

1352    Enumeration representing the gender of the connector presented to the Subscriber.

| **Value** |
| --- |
| SOCKET |
| PLUG |

1353                                         **Table 54 Gender**

### 14.6.8    SynchronousEthernet

1355    File: \ip\common\ipCommon.yaml

1356    Enumeration indicating if the physical link supports Synchronous Ethernet.

| **Value** |
| --- |
| DISABLED |
| ESMC |
| NO_ESMC |

1357                                  **Table 55 SynchronousEthernet**

### 14.6.9    UniAccessLinkEthernetLinkAggregation

1359    File: \ip\common\ipCommon.yaml

1360    Link Aggregation, as described in IEEE Std. 802.1AX-2020 allows one or more parallel instances
1361    of full-duplex point-to-point Ethernet links to be aggregated to form a Link Aggregation Group
1362    (LAG) such that the MAC Client (the UNI Access Link) can treat the LAG as if it were a single
1363    link. Reference MEF 61.1.1 Section A1-1.3.2 UNI Access Link Trunk Ethernet Link Aggregation
1364    Service Attribute.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| lacpVersion | LacpVersion | 0..1 | The value of LACPv1, LACPv2, or Static and indicates which version of the Link Aggregation Control Protocol, LACP, is used. (See clause 6.4 in IEEE Std 802.1AX-2020 [A1-4].). If the value is Static, LACP is not used. |
| portMap | ConversationIdToAggregationLinkMap[] | 0..* | A list of 2-tuples ⟨vid, lspl⟩ that represents a VLAN ID to Aggregation Link Map (in clause 6.6 of IEEE Std 802.1AX-2020 this is referred to as "Admin_Conv_Link_Map"). The first element, vid, is a VLAN ID, and the second element, lspl, (Link Selection Priority List) is a list of Link Number IDs. |

1365                                **Table 56 UniAccessLinkEthernetLinkAggregation**

1366   **14.6.10   LacpVersion**

1367   File: \ip\common\ipCommon.yaml

1368   This is a 2-tuple <x,y> where x is a list of Port Conversation IDs or ranges of Port Conversation
1369   IDs (a Port Conversation ID is a VLAN ID or 0 for untagged frames) and y is a list of Link
1370   Numbers. This is used in the Port Conversation to Aggregation Link Map for the UNI and ENNI.

| Value |
|-------|
| LACPV1 |
| LACPV2 |
| STATIC |

1371                                          **Table 57 LacpVersion**

1372   **14.6.11   ConversationIdToAggregationLinkMap**

1373   File: \ip\common\ipCommon.yaml

1374   This is a 2-tuple <x,y> where x is a list of Port Conversation IDs or ranges of Port Conversation
1375   IDs (a Port Conversation ID is a VLAN ID or 0 for untagged frames) and y is a list of Link
1376   Numbers. This is used in the Port Conversation to Aggregation Link Map for the UNI and ENNI.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| conversationIDs | ConversationIdRange[] | 1..* | 802.1AX-2014 sec. 6.6.2.1 - A Port Conversation ID is a VLAN ID (1 to 4094) or 0 to represent untagged and priority-tagged frames. |
| aggLinkList | integer[] | 1..* | 802.1AX-2014 sec. 6.6.2.1 - An ordered list of Aggregation Link Numbers |

1377 **Table 58 ConversationIdToAggregationLinkMap**

1378 ### 14.6.12 ConversationIdRange

1379 File: \ip\common\ipCommon.yaml

1380 A range of ConversationID (either a VLAN Id or 0 for untagged frames)

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| start | integer | 0..1 | The starting Conversation ID of the range or the only Conversation ID if there is no end value |
| end | integer | 0..1 | The final Conversation ID in the range |

1381 **Table 59 ConversationIdRange**

1382 ## 14.7 IP SLS



1383
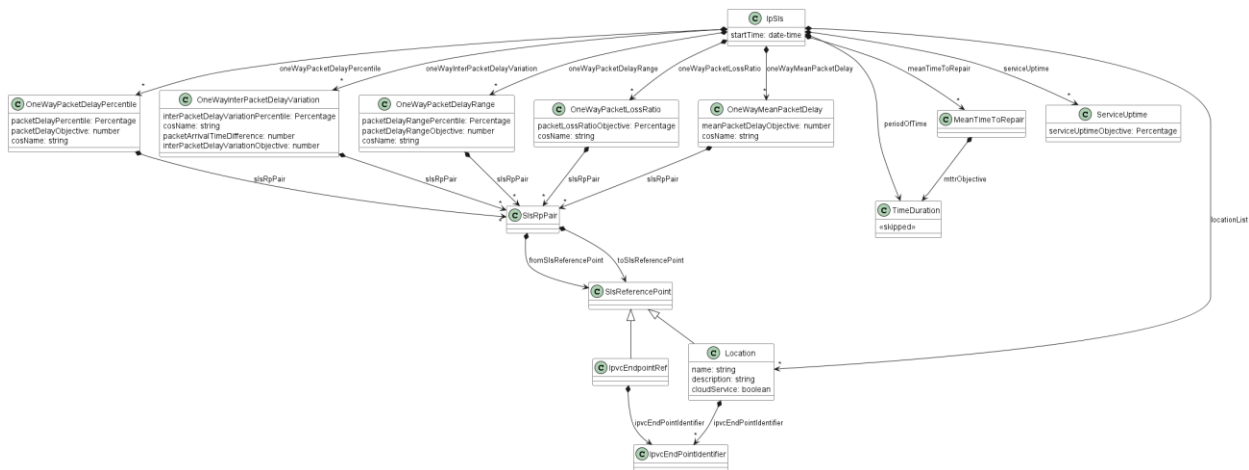
1384 **Figure 22 IpSls**

1385 Figure 22 shows the model of the IP SLS with all available metrics.

1386 ### 14.7.1     IpSls

1387 File: \ip\common\ipSls.yaml

1388 The IPVC Service Level Specification Service Attribute is either None, or a four-tuple of the form
1389 (s, T, E, L) where s is the start time, T is a period of time, E is a set of SLS entries, and L is a set
1390 of locations as described in section 10.9.1. Each SLS entry in E contains the Performance Metric,
1391 the CoS Name, and number of other parameters specific to the Performance Metric, as described
1392 in the subsections below. Reference MEF 61.1 Section 10.9 IPVC Service Level Specification
1393 Service Attribute.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| oneWayPacketDelayPercentile | OneWayPacketDelayPercentile[] | 0..* | List of SLS Entries for the One-way Packet Delay Percentile metric. |
| locationList | Location[] | 1..* | A Location is associated with one or more IPVC EPs or with a cloud service. A Location can refer to a specific address (such as the SP's premises where the PE is located), a city, a region, or even a country. |
| oneWayInterPacketDelayVariation | OneWayInterPacketDelayVariation[] | 0..* | List of SLS Entries for the One-way Inter-Packet Delay Variation metric. |
| oneWayPacketDelayRange | OneWayPacketDelayRange[] | 0..* | List of SLS Entries for the One-way Packet Delay Range metric. |
| serviceUptime | ServiceUptime[] | 0..* | Service uptime metric. |
| oneWayPacketLossRatio | OneWayPacketLossRatio[] | 0..* | List of SLS Entries for the One-way Packet Loss Ratio metric. |
| startTime | date-time | 0..1 | Start time of IP SLS. |
| oneWayMeanPacketDelay | OneWayMeanPacketDelay[] | 0..* | List of SLS Entries for the One-way Mean Packet Delay metric. |

| periodOfTime | TimeDuration | 0..1 | Period of time over which IP SLS is measured. |
|---|---|---|---|
| meanTimeToRepair | MeanTimeToRepair[] | 0..* | The Mean Time To Repair Performance Metric is the arithmetic mean of the durations of all outages that start in a given time period, excluding any pre-agreed maintenance periods. |

1394                                          **Table 60 IpSls**

1395   **14.7.2     OneWayInterPacketDelayVariation**

1396   File: \ip\common\ipSls.yaml

1397   The One-way Inter-Packet Delay Variation Performance Metric is the maximum, over all the
1398   ordered pairs of SLS-RPs in a given set S, of the vth percentile of differences between the one-
1399   way packet delays of Qualified Packets that arrive at time separated by a given interval tau, for a
1400   given ordered pair of SLS-RPs, a given CoS Name, and a given time period Tk. Reference MEF
1401   61.1 Section 10.9.6 One-way Inter-Packet Delay Variation Performance Metric.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| slsRpPair | SlsRpPair[] | 1..* | Set of ordered SLS-RP pairs. Reference MEF 61.1 Section 10.9.5 One-way Mean Packet Delay Performance Metric, Table-5. |
| interPacketDelayVariationPercentile | Percentage | 0..1 | Inter-Packet Delay Variation Percentile. Reference MEF 61.1 Section 10.9.6 One-way Inter-Packet Delay Variation Performance Metric, Table 6. |
| cosName | string | 0..1 | One of the values in the IPVC List of Class of Service Names Service Attribute. Reference MEF 61.1 Section 10.9.6 One-way Inter-Packet Delay Variation Performance Metric, Table-6. |
| packetArrivalTimeDifference | number | 0..1 | Difference in the time of arrival of packets. Reference MEF 61.1 Section 10.9.6 One-way Inter-Packet Delay Variation Performance Metric, Table 6. |
| interPacketDelayVariationObjective | number | 0..1 | Inter-Packet Delay Variation Objective. Reference MEF 61.1 Section 10.9.6 One-way Inter-Packet Delay Variation Performance Metric, Table 6. |

**Table 61 OneWayInterPacketDelayVariation**

### 14.7.3    OneWayMeanPacketDelay

File: \ip\common\ipSls.yaml

The One-way Mean Packet Delay Performance Metric is the maximum, over all the ordered pairs of SLS-RPs in a given set S, of the arithmetic mean of one-way packet delay for Qualified Packets for a given ordered pair of SLS-RPs, a given CoS Name, and a given time period Tk. Reference MEF 61.1 Section 10.9.5 One-way Mean Packet Delay Performance Metric.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| slsRpPair | SlsRpPair[] | 1..* | Set of ordered SLS-RP pairs. Reference MEF 61.1 Section 10.9.5 One-way Mean Packet Delay Performance Metric, Table-5. |
| meanPacketDelayObjective | number | 0..1 | Mean Packet Delay Objective. Reference MEF 61.1 Section 10.9.5 One-way Mean Packet Delay Performance Metric, Table-5. |
| cosName | string | 0..1 | One of the values in the IPVC List of Class of Service Names Service Attribute. Reference MEF 61.1 Section 10.9.5 One-way Mean Packet Delay Performance Metric, Table-5. |

1409 **Table 62 OneWayMeanPacketDelay**

1410 **14.7.4    OneWayPacketDelayPercentile**

1411    File: \ip\common\ipSls.yaml

1412    The One-way Packet Delay Percentile Performance Metric is the maximum, over all the order
1413    pairs of SLS-RPs in a given set S, of the pth percentile of one-way packet delay for Qualified
1414    Packets for a given order pair of SLS-RPs, a given CoS Name and a given time period Tk.
1415    Reference MEF 61.1 Section 10.9.4 One-way Packet Delay Percentile Performance Metric.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| slsRpPair | SlsRpPair[] | 1..* | Set of ordered SLS-RP pairs. Reference MEF 61.1 Section 10.9.5 One-way Mean Packet Delay Performance Metric, Table-5. |
| packetDelayPercentile | Percentage | 0..1 | Packet Delay Range Percentile. Reference MEF 61.1 Section 10.9.7 One-way Packet Delay Range Performance Metric, Table 7. |
| packetDelayObjective | number | 0..1 | Packet Delay Objective. Reference MEF 61.1 Section 10.9.4 One-way Packet Delay Percentile Performance Metric, Table-4. |
| cosName | string | 0..1 | One of the values in the IPVC List of Class of Service Names Service Attribute. Reference MEF 61.1 Section 10.9.4 One-way Packet Delay Percentile Performance Metric, Table-4. |

1416 **Table 63 OneWayPacketDelayPercentile**

1417 **14.7.5    OneWayPacketDelayRange**

1418    File: \ip\common\ipSls.yaml

1419 The One-way Packet Delay Range Performance Metric is the maximum, over all the ordered
1420 pairs of SLS-RPs in a given set S, of the difference between the rth percentile of one-way packet
1421 delay and the minimum one-way packet delay, for Qualified Packets for a given ordered pair of
1422 SLS-RPs, a given CoS Name, and a given time period Tk. Reference MEF 61.1 Section 10.9.7
1423 One-way Packet Delay Range Performance Metric.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| slsRpPair | SlsRpPair[] | 1..* | Set of ordered SLS-RP pairs. Reference MEF 61.1 Section 10.9.5 One-way Mean Packet Delay Performance Metric, Table-5. |
| packetDelayRangePercentile | Percentage | 0..1 | Packet Delay Range Percentile. Reference MEF 61.1 Section 10.9.7 One-way Packet Delay Range Performance Metric, Table 7. |
| cosName | string | 0..1 | One of the values in the IPVC List of Class of Service Names Service Attribute. Reference MEF 61.1 Section 10.9.7 One-way Mean Packet Delay Performance Metric, Table-7. |
| packetDelayRangeObjective | number | 0..1 | Packet Delay Range Objective. Reference MEF 61.1 Section 10.9.7 One-way Packet Delay Range Performance Metric, Table 7. |

1424                                    **Table 64 OneWayPacketDelayRange**

### 14.7.6    OneWayPacketLossRatio

1426 File: \ip\common\ipSls.yaml

1427 The One-way Packet Loss Ratio Performance Metric is the maximum, over the ordered pairs of
1428 SLS-RPs in a given set S, of the ratio of lost packets to transmitted packets for a given ordered
1429 pair of SLS-RPs, a given CoS Name and a given time period Tk. Reference MEF 61.1 Section
1430 10.9.8 One-way Packet Loss Ratio Performance Metric.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| packetLossRatioObjective | Percentage | 0..1 | Packet Loss Ratio Objective. Reference MEF 61.1 Section 10.9.8 One-way Packet Loss Ratio Performance Metric, Table 8. |
| slsRpPair | SlsRpPair[] | 1..* | Set of ordered SLS-RP pairs. Reference MEF 61.1 Section 10.9.5 One-way Mean Packet Delay Performance Metric, Table-5. |
| cosName | string | 0..1 | One of the values in the IPVC List of Class of Service Names Service Attribute. Reference MEF 61.1 Section 10.9.8 One-way Packet Loss Ratio Performance Metric, Table-8. |

1431                                      **Table 65 OneWayPacketLossRatio**

1432 **14.7.7    Percentage**

1433    File: \ip\common\ipSls.yaml

1434    his is a number of percent - a  number between 0 and 100.

1435 **14.7.8    Location**

1436    File: \ip\common\ipSls.yaml

1437    A Location is associated with one or more IPVC EPs or with a cloudservice.  A Location can refer
1438    to a specific address (such as the SP'spremises where the PE is located), a city, a region, or even
1439    acountry.

1440    Inherits from: SlsReferencePoint

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipvcEndPointIdentifier | IpvcEndPointIdentifier[] | 1..* | IPVC End Point identifier as described in MEF 61.1 Section 11.1. Note - It points to the value of IpvcEndPoint.identifier Service Attribute. It is not intended to point to the potential Product identifier if IPVC Endpoint is an instance of a Product. |
| name | string | 0..1 | Location name |
| description | string | 0..1 | Location description |
| cloudService | boolean | 0..1 | Attribute to indicate if associated with a cloud service. |

1441                              **Table 66 Location**

1442 **14.7.9    MeanTimeToRepair**

1443    File: \ip\common\ipSls.yaml

1444    The Mean Time To Repair Performance Metric is the arithmetic mean of the durations of all
1445    outages that start in a given time period, excluding any pre-agreed maintenance periods. Reference
1446    MEF 61.1.1. Section 10.9.10 Mean Time To Repair Performance Metric

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| mttrObjective | TimeDuration | 0..1 | Mean Time To Repair Objective |

1447                        **Table 67 MeanTimeToRepair**

1448 **14.7.10    ServiceUptime**

1449    File: \ip\common\ipSls.yaml

1450    The Service Uptime Performance Metric is the proportion of time, during a given time period
1451    Tk, that the service is working from the perspective of the Subscriber (for a Subscriber IP
1452    Service) or the perspective of the SP/SO (for an Operator IP Service), excluding any pre-agreed
1453    exceptions, for example maintenance intervals. Reference MEF 61.1[1] Section 10.9 Service
1454    Uptime Performance Metric.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| serviceUptimeObjective | Percentage | 0..1 | Service Uptime Objective. Reference MEF 61.1 Section 10.9.9 Service Uptime Performance Metric, Table 9. |

1455                                                        **Table 68 ServiceUptime**

1456    **14.7.11    SlsReferencePoint**

1457    File: \ip\common\ipSls.yaml

1458    SlsReferencePoint is an abstract data type that can be subclassed to IpvcEndPoint and Location.
1459    Reference MEF 61.1 Section 10.9.1 SLS Reference Points.

1460    **14.7.12    SlsRpPair**

1461    File: \ip\common\ipSls.yaml

1462    Service Level Specification Reference Point Pair. In a multipoint or rooted multipoint IPVC,
1463    performance objectives are ideally specified as applying between pairs of IPVC EPs - in other
1464    words, they apply to the performance that IP Data Packets experience as they flow from one EI
1465    to another. The SlsRpPair is a representation of this association. Reference MEF 61.1 Section
1466    10.9.1

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| toSlsReferencePoint | SlsReferencePoint | 0..1 | Pointer to the "to" SLS Reference Point. |
| fromSlsReferencePoint | SlsReferencePoint | 0..1 | Pointer to the "from" SLS Reference Point. |

1467                                                          **Table 69 SlsRpPair**

1468    **14.7.13    IpvcEndpointRef**

1469    File: \ip\common\ipSls.yaml

1470    A subclass of a SlsReferencePoint pointing to an instance of IPVC Endpoint

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| ipvcEndPointIdentifier | IpvcEndPointIdentifier | 0..1 | IPVC End Point identifier as described in MEF 61.1 Section 11.1.<br><br>Note - It points to the value of IpvcEndPoint.identifier Service Attribute. It is not intended to point to the potential Product identifier if IPVC Endpoint is an instance of a Product. |

1471                                    **Table 70 IpvcEndpointRef**

1472    **14.7.14    IpvcEndPointIdentifier**

1473    File: \ip\common\ipCommon.yaml

1474    IPVC End Point identifier as described in MEF 61.1 Section 11.1. Note – It points to the value of
1475    IpvcEndPoint.identifier Service Attribute. It is not intended to point to the potential Product
1476    identifier if IPVC Endpoint is an instance of a Product.

1477    maxLength: 53

1478    pattern: "[\x20-\x7F]+"

1479    **14.8      Routing Protocols**
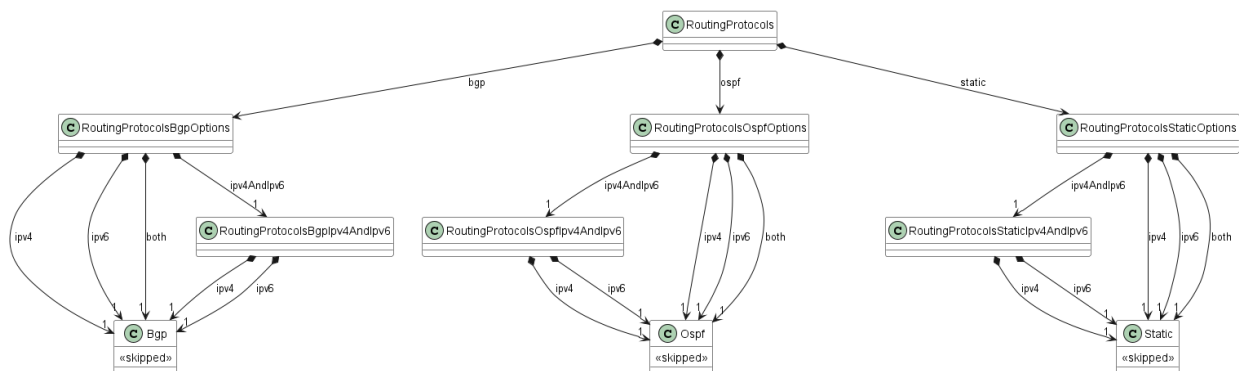


1481                                    **Figure 23 Routing Protocols**

1482    The UNI Routing Protocols Service Attribute specifies the routing protocols and associated
1483    parameters that are used to exchange IP routes across the UNI.  The value is a list of protocols
1484    (possibly empty), where each entry consists of the protocol name (one of Static, OSPF or BGP),
1485    the type of routes that will be exchanged (one of IPv4, IPv6 or Both), and a set of additional
1486    parameters as specified in the subsections below. According to [R109] The value of the UNI
1487    Routing Protocols Service Attribute MUST NOT contain more than one entry for the same

1488 protocol name, except when there are exactly two entries with a given protocol name, one with
1489 route type IPv4 and one with route type IPv6.. This means per given type of routing protocols one
1490 out of four possible sets of configuration can be provided: IPv4, IPv6, IPv4 and IPv6, Both. In
1491 order to model that in the API resource model additional type has been added
1492 *RoutingProcotolsXXXOptions* that has four mutually exclusive attributes: *ipv4, ipv6, ipv4AndIpv6,*
1493 *both*, respectively to handle this requirement.

### 14.8.1    RoutingProtocols

1495 File: \ip\common\ipCommon.yaml

1496 Data type to support routing protocols and associated parameters that are used to exchange IP
1497 routes across the UNI. The value is a list of protocols (possibly empty), where each entry consists
1498 of the protocol name (one of Static, OSPF or BGP) the type of routes that will be exchanged (one
1499 of IPv4 or IPv6 or Both) and set of additional parameters as specified. Reference MEF 61.1 Section
1500 12.7 UNI Routing Protocols Service Attributes.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| static | RoutingProtocolsStaticOptions | 0..1 | Static routing configuration options. |
| bgp | RoutingProtocolsBgpOptions | 0..1 | BGP routing protocol configuration options. |
| ospf | RoutingProtocolsOspfOptions | 0..1 | OSPF routing protocol configuration options. |

1501                               **Table 71 RoutingProtocols**

### 14.8.2    RoutingProtocolsBgpIpv4AndIpv6

1503 File: \ip\common\ipCommon.yaml

1504 Configuration for exchanging IPv4 and IPv6 types of routes.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipv4* | Bgp | 1 | Configuration for exchanging IPv4 types of routes. |
| ipv6* | Bgp | 1 | Configuration for exchanging IPv6 types of routes. |

1505                       **Table 72 RoutingProtocolsBgpIpv4AndIpv6**

### 14.8.3    RoutingProtocolsBgpOptions

1507 File: \ip\common\ipCommon.yaml

1508 BGP routing protocol configuration options. The configuration of the BGP can be provided for the
1509 following type pf routes that will be exchanged:

1510 • ipv4, or

1511 • ipv6, or

1512 • both, or

1513 • ipv4 and ipv6

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipv4 | Bgp | 0..1 | Configuration for exchanging IPv4 types of routes. |
| Ipv6 | Bgp | 0..1 | Configuration for exchanging IPv6 types of routes. |
| ipv4AndIpv6 | RoutingProtocolsBgpIpv4AndIpv6 | 0..1 | Configuration for exchanging IPv4 and IPv6 types of routes. |
| Both | Bgp | 0..1 | Configuration for exchanging both IPv4 and IPv6 types of routes. |

1514 **Table 73 RoutingProtocolsBgpOptions**

1515 oneOf:

1516 - required: [ipv4]

1517 - required: [ipv6]

1518 - required: [both]

1519 - required: [ipv4AndIpv6]

1520 **14.8.4    RoutingProtocolsOspfIpv4AndIpv6**

1521 File: \ip\common\ipCommon.yaml

1522 Configuration for exchanging IPv4 and IPv6 types of routes.

| Name | Type | Multiplicity | Description |
|------|------|------|------|
| ipv4* | Ospf | 1 | Configuration for exchanging IPv4 types of routes. |
| ipv6* | Ospf | 1 | Configuration for exchanging IPv6 types of routes. |

1523                                    **Table 74 RoutingProtocolsOspfIpv4AndIpv6**

1524  **14.8.5    RoutingProtocolsOspfOptions**

1525  File: \ip\common\ipCommon.yaml

1526  OSPF   routing   protocol   configuration   options.   The   configuration   of   the   BGP
1527  can be provided for the following type pf routes that will be exchanged:

1528      • ipv4, or

1529      • ipv6, or

1530      • both, or

1531      • ipv4 and ipv6

| Name | Type | Multiplicity | Description |
|------|------|------|------|
| ipv4* | Ospf | 0..1 | Configuration for exchanging IPv4 types of routes. |
| ipv6* | Ospf | 0..1 | Configuration for exchanging IPv6 types of routes. |
| ipv4AndIpv6* | RoutingProtocolsOspfIpv4AndIpv6 | 0..1 | Configuration for exchanging IPv4 and IPv6 types of routes. |
| both* | Ospf | 0..1 | Configuration for exchanging both IPv4 and IPv6 types of routes. |

1532                                    **Table 75 RoutingProtocolsOspfOptions**

1533  oneOf:

1534    - required: [ipv4]

1535    - required: [ipv6]

1536    - required: [both]

1537    - required: [ipv4AndIpv6]

### 14.8.6    RoutingProtocolsStaticIpv4AndIpv6

1539    File: \ip\common\ipCommon.yaml

1540    Configuration for exchanging IPv4 and IPv6 types of routes.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipv4* | Static | 1 | Configuration for exchanging IPv4 types of routes. |
| ipv6* | Static | 1 | Configuration for exchanging IPv6 types of routes. |

1541                    **Table 76 RoutingProtocolsStaticIpv4AndIpv6**

### 14.8.7    RoutingProtocolsStaticOptions

1543    File: \ip\common\ipCommon.yaml

1544    Static routing configuration options. The configuration of the BGP can be provided for the
1545    following type pf routes that will be exchanged: - ipv4, or - ipv6, or - both, or - ipv4 and ipv6

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipv4* | Static | 0..1 | Configuration for exchanging IPv4 types of routes. |
| ipv6* | Static | 0..1 | Configuration for exchanging IPv6 types of routes. |
| ipv4AndIpv6* | RoutingProtocolsStaticIpv4AndIpv6 | 0..1 | Configuration for exchanging IPv4 and IPv6 types of routes. |
| both* | Static | 0..1 | Configuration for exchanging both IPv4 and IPv6 types of routes. |

1546                      **Table 77 RoutingProtocolsStaticOptions**

1547    oneOf:

1548      - required: [ipv4]

1549      - required: [ipv6]

1550      - required: [both]

1551        - required: [ipv4AndIpv6]

1552    **14.9      BGP**



1553
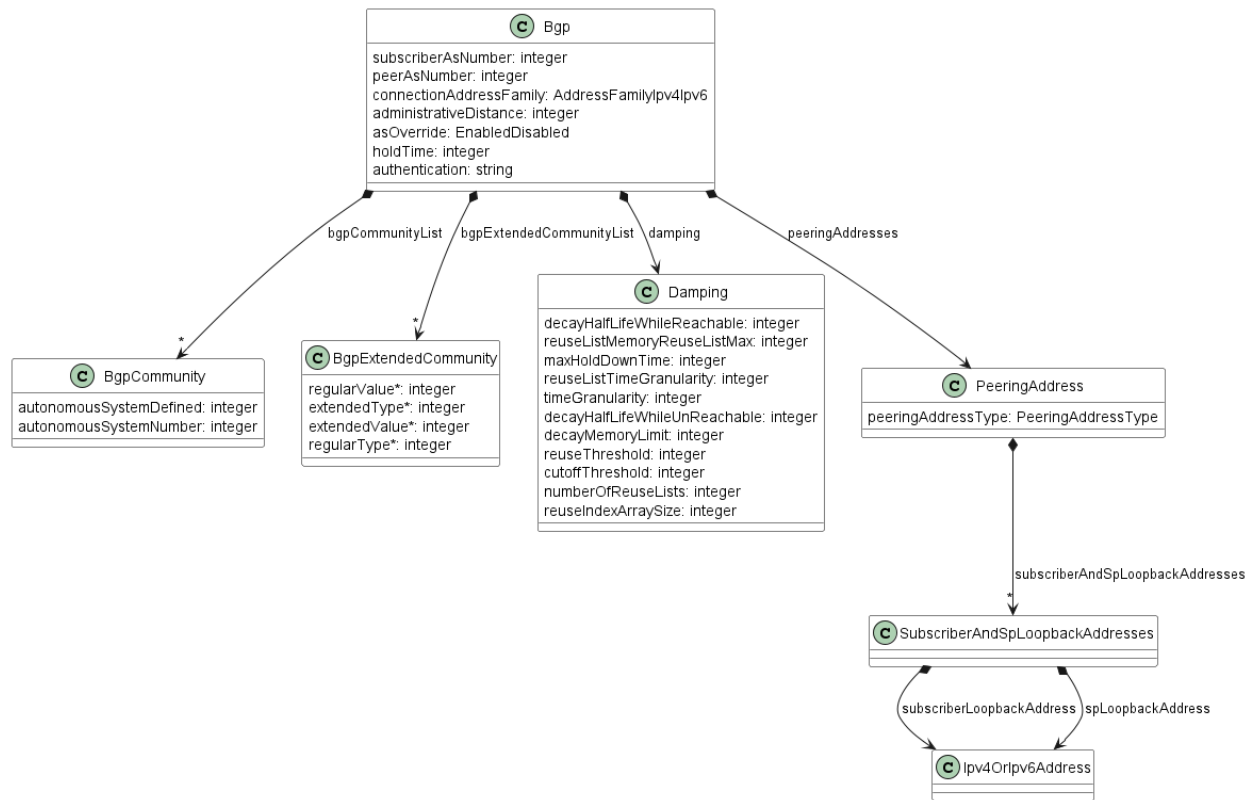
1554                                    **Figure 24 Bgp**

1555    Figure 24 depicts the model of BGP routing protocol configuration model.

1556    **14.9.1      Bgp**

1557    File: \ip\common\ipCommon.yaml

1558    When an entry in the UNI Routing Protocol is for BGP, BGP as specified in RFC 4271 is used
1559    across the UNI to exchange information. Reference MEF 61.1 Section 12.7.3 BGP.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| subscriberAsNumber | integer | 0..1 | BGP Subscriber Autonomous System number. |
| peerAsNumber | integer | 0..1 | BGP Peer Autonomous System Number. |
| bgpExtendedCommunityList | BgpExtendedCommunity[] | 0..* | Mechanism for labeling information carried in BGP-4. Provide enhancement over existing BGP Community Attribute an extended range, the addition of type field. |
| connectionAddressFamily | AddressFamilyIpv4Ipv6 | 0..1 | Connection Address Family (IPv4 or IPv6). |
| damping | Damping | 0..1 | Route flap damping. When the Damping parameter is NONE, the attribute is not set. When not NONE a single set of parameters described in Section 4.3 of RFC 2430 MUST be agreed. |
| administrativeDistance | integer | 0..1 | BGP Administrative Distance. |
| asOverride | EnabledDisabled | 0..1 | Autonomous System Override. The SP (or Operator) can overwrite instances of the Subscriber's AS Number in the AS Path with their own AS Number, when advertising routes to the Subscriber. This needs to be explicitly agreed between the SP and the Subscriber, and/or between an SP/SO and an Operator. |
| peeringAddresses | PeeringAddress | 0..1 | Peering Addresses. |
| holdTime | integer | 0..1 | Hold time in seconds. Indicates the agreed Hold Time used for BGP sessions. The possible values are 0 or an integer in the range 3-65535. |
| authentication | string | 0..1 | BGP Authentication. It is either None or if present is it a value of MD5 Password. It is assumed that an encrypted channel is used for API session so that the password is protected. |
| bgpCommunityList | BgpCommunity[] | 0..* | Used to control which routers are accepted, preferred, distributed, or advertised. |

1560                                         **Table 78 Bgp**

1561 **14.9.2    BgpCommunity**

1562 File: \ip\common\ipCommon.yaml

1563 A community is a group of destinations which share some common property. Each autonomous
1564 system administrator may define which communities a destination belongs to.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| autonomousSystemDefined* | integer | 1 | The remaining octets. |
| autonomousSystemNumber* | integer | 1 | The first two octets encoding the Autonomous System value. |

1565                                    **Table 79 BgpCommunity**

1566 **14.9.3    BgpExtendedCommunity**

1567 File: \ip\common\ipCommon.yaml

1568 This attribute provides a mechanism for labeling information carried in BGP-4. These labels can
1569 be used to control the distribution of this information, or for other applications.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| regularValue* | integer | 0..1 | Octets 2 - 8 of the value part of the address. Used in case only Regular Type is provided. |
| extendedType* | integer | 0..1 | Extended Type Field, 2 octets length |
| extendedValue* | integer | 0..1 | Octets 3 - 8 of the value part of the address. Used in case only Extended Type is provided. |
| regularType* | integer | 0..1 | Regular Type Field, 1 octet length |

1570                              **Table 80 BgpExtendedCommunity**

1571 oneOf:

1572    - required: [regularType, regularValue]

1573    - required: [extendedType, extendedValue]

1574 **14.9.4    AddressFamilyIpv4Ipv6**

1575 File: \ip\common\ipCommon.yaml

1576    Specifies whether the session is established over IPv4 or IPv6.

| Value |
|-------|
| IPV4 |
| IPV6 |

1577                        **Table 81 AddressFamilyIpv4Ipv6**

1578    **14.9.5    Damping**

1579    File: \ip\common\ipCommon.yaml

1580    Damping parameters as defined in RFC 2439 BGP Route Flap Damping, Section 4.2

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| decayHalfLifeWhileReachable | integer | 0..1 | This value is the time duration in seconds during which the accumulated stability figure of merit will be reduced by half if the route if considered reachable (whether suppressed or not). |
| reuseListMemoryReuseListMax | integer | 0..1 | This is the time (in seconds) value corresponding to the last reuse list. This may be the maximum value of T-hold for all parameter sets of may be configured. |
| maxHoldDownTime | integer | 0..1 | This value is the maximum time a route can be suppressed no matter how unstable it has been prior to this period of stability. In seconds. |
| reuseListTimeGranularity | integer | 0..1 | This is the time (in seconds) interval between evaluations of the reuse lists. Each reuse lists corresponds to an additional time increment. |
| timeGranularity | integer | 0..1 | This is the time granularity in seconds used to perform all decay computations. |
| decayHalfLifeWhileUnReachable | integer | 0..1 | This value is the time duration in seconds during which the accumulated stability figure of merit will be reduced by half if the route if considered unreachable. If not specified or set to zero, no decay will occur while a route remains unreachable. |
| decayMemoryLimit | integer | 0..1 | This is the maximum time (in seconds) that any memory of previous instability will be retained given that the route's state remains unchanged, whether reachable or unreachable. This parameter is generally used to determine array sizes. |
| reuseThreshold | integer | 0..1 | This value is expressed as a number of route withdrawals. It is the value below which a suppressed route will now be used again. |
| cutoffThreshold | integer | 0..1 | This value is expressed as a number of route withdrawals. It is the value above which a route advertisement will be suppressed. |
| numberOfReuseLists | integer | 0..1 | This is the number of reuse lists. It may be determined from reuse-list-max or set explicitly. |
| reuseIndexArraySize | integer | 0..1 | This is the size of reuse index arrays. This size determines the accuracy with which suppressed routes can be placed within the set of reuse lists when suppressed for a long time. |

1581 **Table 82 Damping**

1582 **14.9.6  PeeringAddress**

1583  File: \ip\common\ipCommon.yaml

1584  Peering Addresses. Connection Addresses, or Loopbacks plus a list of pairs of IP addresses.
1585  Reference MEF 61.1 Section 12.7.3 BGP.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| subscriberAndSpLoopbackAddresses | SubscriberAndSpLoopbackAddresses[] | 0..* | A list of pairs of IP addresses, each pair containing the Subscriber's loopback address and the SP's or Operator's loopback address. A single BGP peering session is established for each pair of addresses. |
| peeringAddressType | PeeringAddressType | 0..1 | If the Peering Addresses parameter is CONNECTION_ADDRESSES, a separate BGP peering session is established over each UNI Access Link, using the primary IPv4 addresses in the UNI Access Link IPv4 Connection Addressing Service Attribute (section 13.4) or the first IPv6 addresses in the UNI Access Link IPv6 Connection Addressing Service Attribute (section 13.5), as indicated by the Connection Address Family parameter. If the Peering Addresses parameter is LOOPBACKS, a list of pairs of IP addresses is additionally specified, each pair containing the Subscriber's loopback address and the SP's or Operator's loopback address. A single BGP peering session is established for each pair of addresses. |

1586  **Table 83 PeeringAddress**

1587  **14.9.7  PeeringAddressType**

1588  File: \ip\common\ipCommon.yaml

1589  If the Peering Addresses parameter is CONNECTION_ADDRESSES, a separate BGP peering
1590  session is established over each UNI Access Link, using the primary IPv4 addresses in the UNI
1591  Access Link IPv4 Connection Addressing Service Attribute (section 13.4) or the first IPv6
1592  addresses in the UNI Access Link IPv6 Connection Addressing Service Attribute (section 13.5),
1593  as indicated by the Connection Address Family parameter. If the Peering Addresses parameter is
1594  LOOPBACKS, a list of pairs of IP addresses is additionally specified, each pair containing the
1595  Subscriber's loopback address and the SP's or Operator's loopback address. A single BGP peering
1596  session is established for each pair of addresses.

| Value |
| --- |
| CONNECTION_ADDRESSES |
| LOOPBACKS |

1597  **Table 84 PeeringAddressType**

### 14.9.8  SubscriberAndSpLoopbackAddresses

1599  File: \ip\common\ipCommon.yaml

1600  A list of pairs of IP addresses, each pair containing the Subscriber's loopback address and the SP's
1601  or Operator's loopback address. A single BGP peering session is established for each pair of
1602  addresses.

| Name | Type | Multiplicity | Description |
| --- | --- | --- | --- |
| subscriberLoopbackAddress | Ipv4OrIpv6Address | 0..1 | Subscriber's loopback Address for BGP establishing a session |
| spLoopbackAddress | Ipv4OrIpv6Address | 0..1 | Service Provider's loopback Address for BGP establishing a session |

1603  **Table 85 SubscriberAndSpLoopbackAddresses**

1604 **14.10 OSPF**
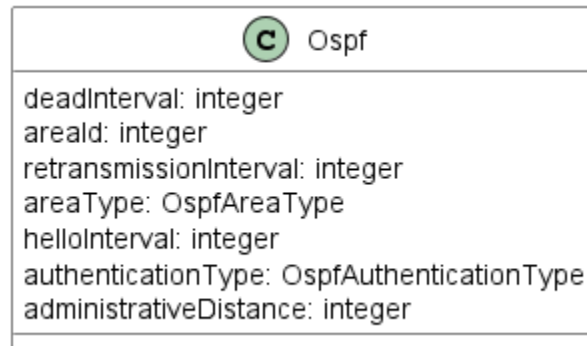


1605

1606 **Figure 25 Ospf**

1607 Figure 25 Presents the model of OSPF configuration. It consists only of simple attributes and
1608 enumerations.

1609 **14.10.1 Ospf**

1610 File: \ip\common\ipCommon.yaml

1611 When an entry in the UNI Routing Protocols is for OSPF, OSPF as specified in RFC 2328 (for
1612 IPv4) and/or RFC 5340 (for IPv6) is used across each UNI Access Link to exchange routing
1613 information. Reference MEF 61.1 Section 12.7.2 OSPF.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| deadInterval | integer | 0..1 | Dead interval (0-429496295, in seconds) |
| areaId | integer | 0..1 | Area ID (0-429967295), normally expressed as an IPv4 address. |
| retransmissionInterval | integer | 0..1 | Retransmit interval (integer greater than 0, in seconds) |
| areaType | OspfAreaType | 0..1 | OSPF Area Type enumeration. |
| helloInterval | integer | 0..1 | Hello interval (0-65535, in seconds) |
| authenticationType | OspfAuthenticationType | 0..1 | OSPF Authentication Type. |
| administrativeDistance | integer | 0..1 | Administrative distance (integer greater than 0) |

1614 **Table 86 Ospf**

1615  **14.10.2   OspfAreaType**

1616  File: \ip\common\ipCommon.yaml

1617  OSPF Area Type enumeration. Reference MEF 61.1 Section 12.7.2 OSPF

| Value |
|-------|
| NORMAL |
| STUB |
| NSSA |

1618  **Table 87 OspfAreaType**

1619  **14.10.3   OspfAuthenticationType**

1620  File: \ip\common\ipCommon.yaml

1621  OSPF Authentication Type enumeration.

| Value |
|-------|
| NONE |
| PASSWORD |
| MESSAGE_DIGEST |

1622  **Table 88 OspfAuthenticationType**

1623 **14.11    Static**



1624
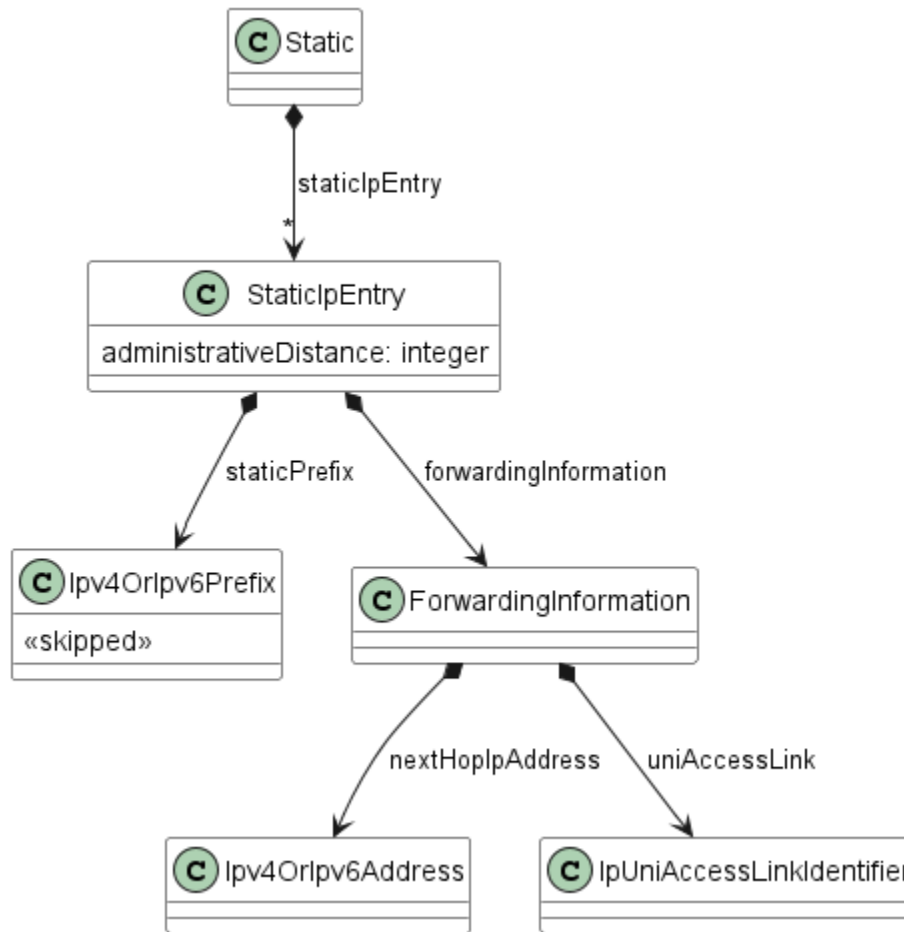
1625                                              **Figure 26 Static**

1626    Figure 26 shows the resource model for Static routing configuration.

1627    **14.11.1    Static**

1628    File: \ip\common\ipCommon.yaml

1629    When an entry in the UNI Routing Protocols list is for Static, the IP Prefixes used in the Subscriber
1630    Network that are reachable via this UNI are specified as additional parameters in the entry. These
1631    are known as Static IP Prefixes. Reference MEF 61.1 Section 12.7.1 Static.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| staticIpEntry | StaticIpEntry[] | 1..* | Static IP address entry. |

1632                                              **Table 89 Static**

1633 **14.11.2    StaticIpEntry**

1634   File: \ip\common\ipCommon.yaml

1635   StaticIpEntry data type including IPv4/IPv6 prefixes, forwarding information and administrative
1636   distance.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| staticPrefix | Ipv4OrIpv6Prefix | 0..1 | IPv4 or IPv6 Prefix that is advertised. |
| administrativeDistance | integer | 0..1 | Administrative distance, an integer > 0. |
| forwardingInformation | ForwardingInformation | 0..1 | Forwarding information with either Next Hop IP address or UNI Access Link identifier. |

1637                                               **Table 90 StaticIpEntry**

1638   **14.11.3    ForwardingInformation**

1639   File: \ip\common\ipCommon.yaml

1640   Forwarding information, consisting of either a nexthop IP address in the Subscriber Network (if
1641   the access medium is multipoint capable, e.g., Ethernet), or a specific UNI Access Link (if the
1642   access medium is strictly point-to-point, e.g., HDLC, PPP over DSL).

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| nextHopIpAddress | Ipv4OrIpv6Address | 0..1 | Next hop IP address. |
| uniAccessLink | IpUniAccessLinkIdentifier | 0..1 | UNI Access Link unique identifier. |

1643                                            **Table 91 ForwardingInformation**

1644   oneOf:

1645     - required: [nextHopIpAddress]

1646     - required: [uniAccessLink]

1647   **14.11.4    IpUniAccessLinkIdentifier**

1648   File: \ip\common\ipCommon.yaml

1649 Ip Uni Access Link identifier as described in MEF 61.1 Section 11.1. Note - It points to the value
1650 of IpUniAccessLink.identifier Service Attribute. It is not intended to point to the potential Product
1651 identifier if IpUniAccessLink is an instance of a Product.

1652 maxLength: 53

1653 pattern: "[\x20-\x7F]+"

1654 **14.12    IPVC Endpoint Bandwidth Profile Envelope**



1655
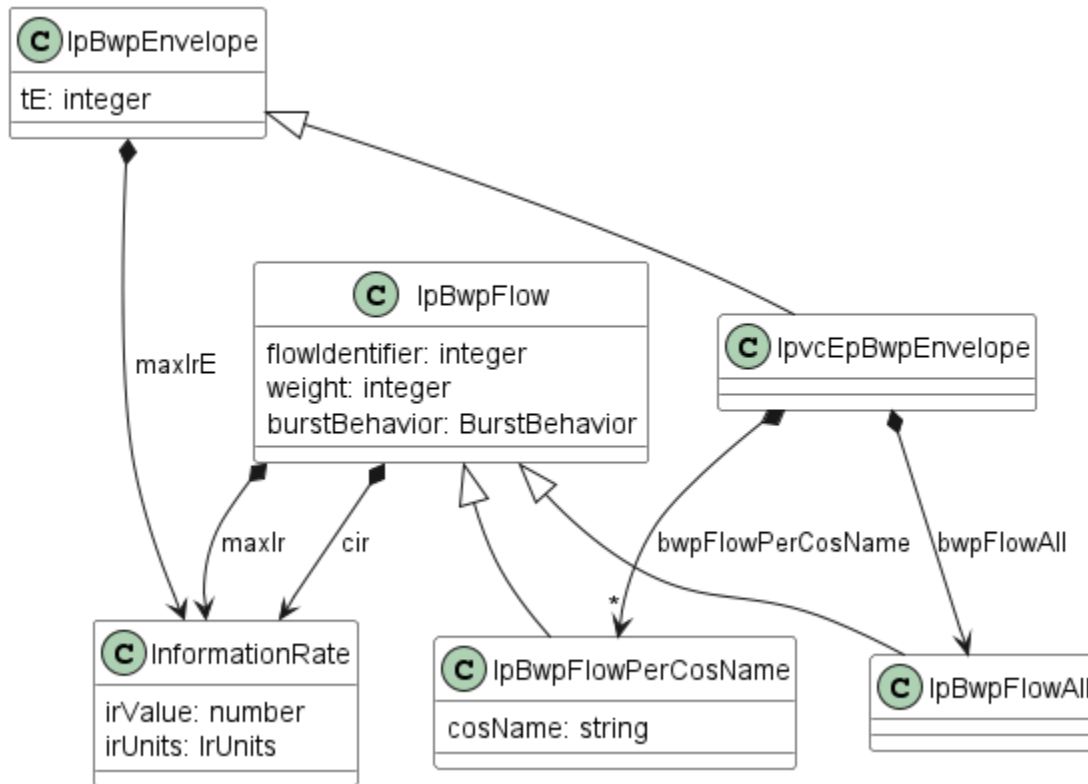
1656 **Figure 27 IpvcEpBwpEnvelope**

1657 Figure 27 IPVC End Point Bandwith Profile Envelope exiends the IpBwpEnvelope to specify
1658 possiblies of Flow configurations that can be applied at the IPVC End Point.

1659 **14.12.1    IpBwpEnvelope**

1660 File: \ip\common\ipCommon.yaml

1661 A BWP Envelope is a list of Bandwith Profile Flows, plus additional parameters for the BWP as
1662 a whole. A BWP Envelope is a set of one or more BWP Flows that are associated such that the
1663 amount of traffic for one flow can affect the amount that is permitted for another flow. This is an
1664 abstract superclass. There subclasses of IPVC Endpoint, IP UNI and IP NI Access Link Envelopes.
1665 Reference MEF 61.1 Section 17.3 Bandwidth Profile Envelopes.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| tE | integer | 0..1 | The Envelope IR Time in milliseconds. This is the time period over which average Information Rates are calculated and thus it limits the size of a burst. Reference MEF 61.1 Section 17.3 Bandwidth Profile Envelopes. |
| maxIrE | InformationRate | 0..1 | The Envelope Maximum Information Rate in bits per second. This is the limit on the total aggregate information rate of traffic across all BWP Flows in the Envelope. Reference MEF 61.1 Section 17.3 Bandwidth Profile Envelopes. |

1666                                     **Table 92 IpBwpEnvelope**

1667    **14.12.2    IpBwpFlow**

1668    File: \ip\common\ipCommon.yaml

1669    A Bandwidth Profile Flow is a stream of IP Packets meeting certain criteria. This is an abstract
1670    superclass. It has subclasses depending on the criteria used. The criteria than can be used depends
1671    on which BWP Envelope the BWP Flow is a part of. Reference MEF 61.1 Section 17.2 Bandwidth
1672    Profile Flows.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| maxIr | InformationRate | 0..1 | Maximum Information Rate in bits per second. Limit on the average information rate of IP Packets for this BWP Flow. Reference MEF 61.1 Table 29 - Bandwidth Profile Parameters for a Bandwidth Profile Flow. |
| flowIdentifier | integer | 0..1 | Identifier for the BWP Flow within the BWP Envelope. Unique integer between 1 and n where n is the number of BWP Flows in the BWP Envelope. Reference MEF 61.1 Table 29 - Bandwidth Profile Parameters for a Bandwidth Profile Flow. |
| weight | integer | 0..1 | Weight as an integer greater than or equal to 0. Relative weight for this BWP Flow compared to other BWP Flows in the BWP Envelope. Reference MEF 61.1 Table 29 - Bandwidth Profile Parameters for a Bandwidth Profile Flow. |
| burstBehavior | BurstBehavior | 0..1 | Burst Behavior either Optimize-Delay or Optimize-Throughput. Whether the SP is requested to optimize the delay characteristic of this flow, or the throughput. Reference MEF 61.1 Table 29 - Bandwidth Profile Parameters for a Bandwidth Profile Flow. |
| cir | InformationRate | 0..1 | Committed Information Rate in bits per second. Average information rate of IP Packets that is committed to this BWP Flow. Reference MEF 61.1 Table 29 - Bandwidth Profile Parameters for a Bandwidth Profile Flow. |

1673                                     **Table 93 IpBwpFlow**

1674 **14.12.3   IpvcEpBwpEnvelope**

1675   File: \ip\common\ipCommon.yaml

1676   A single Bandwidth Profile Envelope consisting of parameters and Bandwidth Profile
1677   specifications. A Bandwidth Profile Envelope can be specified for one of a UNI, a UNI Access
1678   Link and ENNI Link or an IPVC End Point. Reference MEF 61.1 Section 11.11 IPVC EP Egress
1679   Bandwidth Profile Envelope Service Attribute. 11.12 IPVC EP Ingress Bandwidth Profile
1680   Envelope Service Attribute.

1681   Inherits from: IpBwpEnvelope

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| bwpFlowAll | IpBwpFlowAll | 0..1 | Pointer to IpvcEpBwpAll |
| bwpFlowPerCosName | IpBwpFlowPerCosName[] | 1..* | List of BWP flows matching given CoS Name |

1682                                  **Table 94 IpvcEpBwpEnvelope**

1683   oneOf:

1684     - required: [bwpFlowPerCosName]

1685     - required: [bwpFlowAll]

1686   **14.12.4   IpBwpFlowAll**

1687   File: \ip\common\ipCommon.yaml

1688   All IP Data Packets. NOTE: No attributes are needed.

1689   Inherits from: IpBwpFlow

1690   **14.12.5   IpBwpFlowPerCosName**

1691   File: \ip\common\ipCommon.yaml

1692   A Bandwidth for and IPVC End Point with an associated Class of Service identifier.

1693   Inherits from: IpBwpFlow

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| cosName | string[] | 1..* | List of Class of Service names. |

**Table 95 IpBwpFlowPerCosName**

### 14.12.6   BurstBehavior

File: \ip\common\ipCommon.yaml

Enumeration used to select the Bandwidth Profile Flow Burst Behavior attribute. Reference MEF 61.1 Section 17.3 Table 29 Bandwidth Profile Parameters for a Bandwidth Profile Flow.

- OPTIMIZE_DELAY: Enumeration representing the Burst Behavior of optimization of delay.

- OPTIMIZE_THROUGHPUT: Enumeration representing the Burst Behavior of optimization of throughput.

| Value |
|-------|
| OPTIMIZE_DELAY |
| OPTIMIZE_THROUGHPUT |

**Table 96 BurstBehavior**

1705 **14.13    IP UNI Bandwidth Profile Envelope**



1706
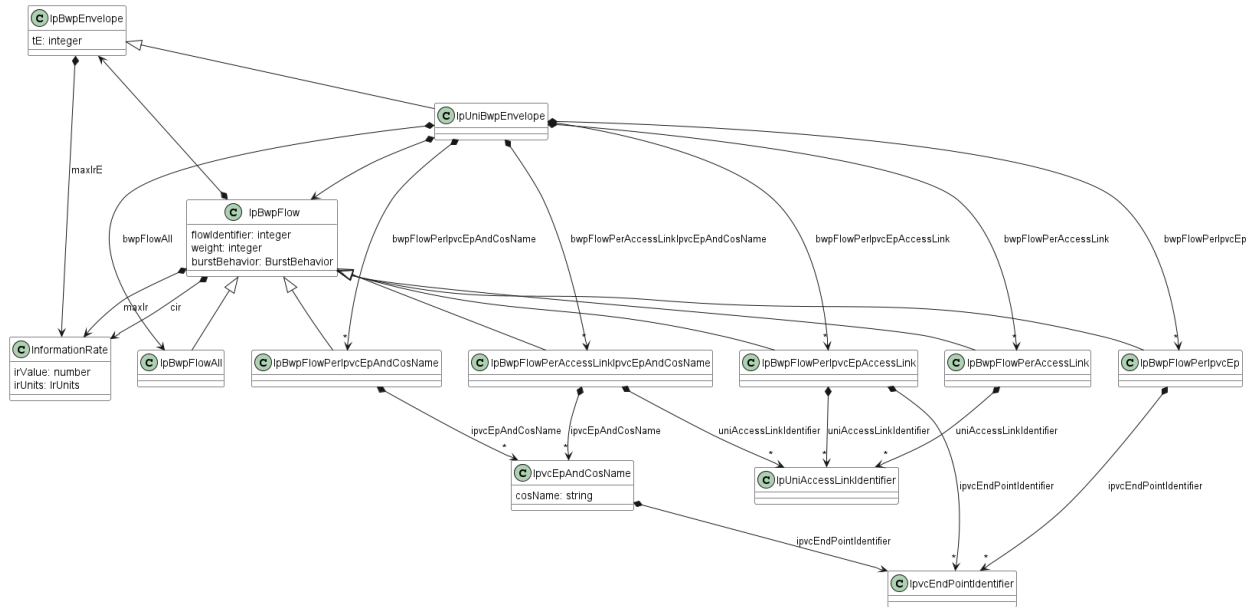
1707                                **Figure 28 IpUniBwpEnvelope**

1708    Figure 28 IP UNI Bandwith Profile Envelope exiends the IpBwpEnvelope to specify possiblies of
1709    Flow configurations that can be applied at the IP UNI.

1710    **14.13.1    IpUniBwpEnvelope**

1711    File: \ip\common\ipCommon.yaml

1712    A single Bandwidth Profile Envelope consisting of parameters and Bandwidth Profile Flow
1713    specifications. The BWP Flows can be defined per UNI, per IPVC EP, per UNI Access Link, per
1714    CosName, etc. Reference MEF 61.1 Sections 12.4 UNI Ingress Bandwidth Profile Envelope
1715    Service Attribute, 12.5 UNI Egress Bandwidth Profile Envelope Service Attribute.

1716    Inherits from: IpBwpEnvelope

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| bwpFlowPerIpvcEp | IpBwpFlowPerIpvcEp[] | 1..* | A list of BWP Flows that are mapped to any of a given set of IPVC EPs. Reference MEF 61.1 Section 12.5 UNI Egress BWP Envelope. |
| bwpFlowPerIpvcEpAccessLink | IpBwpFlowPerIpvcEpAccessLink[] | 1..* | A list of BWP Flows for IP Packets that are received over one of a given set of UNI Access Links and re mapped to any of a given set of IPVC EPs. Reference MEF 61.1 Section 12.5 UNI Egress BWP Envelope. |
| bwpFlowAll | IpBwpFlowAll | 0..1 | A BWP Flow for all IP Data Packets at the UNI. Reference MEF 61.1 Section 12.5 UNI Egress BWP Envelope. |

| | | | |
|---|---|---|---|
| bwpFlowPerAccessLink | IpBwpFlowPerAccessLink[] | 1..* | A list of BWP Flows for IP Packets that are received over one of a given set of UNI Access Links. Reference MEF 61.1 Section 12.5 UNI Egress BWP Envelope. |
| bwpFlowPerIpvcEpAndCosName | IpBwpFlowPerIpvcEpAndCosName[] | 1..* | A list of BWP Flows that are mapped to any of a given set of (IPVC EP, CoS Name) pairs. Reference MEF 61.1 Section 12.5 UNI Egress BWP Envelope. |

| bwpFlowPerAccessLinkIpvcEpAndCosName | IpBwpFlowPerAccessLinkIpvcEpAndCosName[] | 1..* | A list of BWP Flows that are mapped to the UNI Access Link and any of a given set of (IPVC EP, Cos Name) pairs. Reference MEF 61.1 Section 12.5 UNI Egress BWP Envelope. |
|---|---|---|---|

1717                                                    **Table 97 IpUniBwpEnvelope**

1718     oneOf:

1719       - required: [bwpFlowPerAccessLink]

1720       - required: [bwpFlowPerIpvcEp]

1721       - required: [bwpFlowPerIpvcEpAccessLink]

1722       - required: [bwpFlowPerIpvcEpAndCosName]

1723       - required: [bwpFlowPerAccessLinkIpvcEpAndCosName]

1724       - required: [bwpFlowAll]

1725    **14.13.2    IpBwpFlowPerAccessLink**

1726    File: \ip\common\ipCommon.yaml

1727    All Ingress IP Data Packets at the UNI that are received over one of a give set of UNI Access
1728    Links. Reference MEF 61.1 Section 12.5 UNI Egress BWP Envelope.

1729    Inherits from: IpBwpFlow

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| uniAccessLinkIdentifier | IpUniAccessLinkIdentifier[] | 1..* | List of UNI Access Link Identifiers. |

1730                                          **Table 98 IpBwpFlowPerAccessLink**

1731    **14.13.3    IpBwpFlowPerAccessLinkIpvcEpAndCosName**

1732    File: \ip\common\ipCommon.yaml

1733    All Ingress IP Data Packets at the UNI that are received over one of a given set of UNI Access
1734    Links,and that are mapped to the any of a given set of (IPVC EP, Cos Name) pairs. Reference
1735    MEF 61.1 Section 12.4 UNI Ingress BWP Envelope.

1736    Inherits from: IpBwpFlow

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipvcEpAndCosName | IpvcEpAndCosName[] | 1..* | List of pairs of IPVC End Point Identifier and Class of Service Name. Reference MEF 61.1 Table 28. |
| uniAccessLinkIdentifier | IpUniAccessLinkIdentifier[] | 1..* | List of UNI Access Link Identifiers. |

1737                                  **Table 99 IpBwpFlowPerAccessLinkIpvcEpAndCosName**

1738    **14.13.4    IpBwpFlowPerIpvcEp**

1739    File: \ip\common\ipCommon.yaml

1740    All Egress/Ingress IP Data Packets at the UNI that are received over the UNI Access Link, and are
1741    mapped to any of a given set of IPVC End Points. Reference MEF 61.1 Section 13.10 UNI Access
1742    Link Ingress BWP Envelope, Section 13.11 UNI Access Link Egress BWP Envelope

1743    Inherits from: IpBwpFlow

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipvcEndPointIdentifier | IpvcEndPointIdentifier[] | 1..* | List of IPVC End Point Identifiers for an IPVC End Point located at the UNI Access Link. Reference MEF 61.1 Table 28. |

1744                                          **Table 100 IpBwpFlowPerIpvcEp**

1745 **14.13.5    IpBwpFlowPerIpvcEpAccessLink**

1746    File: \ip\common\ipCommon.yaml

1747    All Ingress IP Data Packets at the UNI that are received over one of a give set of UNI Access
1748    Links. Reference MEF 61.1 Section 12.4 UNI Ingress BWP Envelope.

1749    Inherits from: IpBwpFlow

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| ipvcEndPointIdentifier | IpvcEndPointIdentifier[] | 1..* | List of IPVC End Point identifiers as described in MEF 61.1 Section 11.1. Note - It points to the value of IpvcEndPoint.identifier Service Attribute. It is not intended to point to the potential Product identifier if IPVC Endpoint is an instance of a Product. |
| uniAccessLinkIdentifier | IpUniAccessLinkIdentifier[] | 1..* | List of UNI Access Link Identifiers. |

1750                              **Table 101 IpBwpFlowPerIpvcEpAccessLink**

1751    **14.13.6    IpBwpFlowPerIpvcEpAndCosName**

1752    File: \ip\common\ipCommon.yaml

1753    CoS Name from the IPVC List of Class of Service Names (section 10.8) for the IPVC that has the
1754    IPVC EP.

1755    Inherits from: IpBwpFlow

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| ipvcEpAndCosName | IpvcEpAndCosName[] | 1..* | List of pairs of IPVC End Point Identifier and Class of Service Name. Reference MEF 61.1 Table 28. |

1756                              **Table 102 IpBwpFlowPerIpvcEpAndCosName**

1757    **14.13.7    IpvcEpAndCosName**

1758    File: \ip\common\ipCommon.yaml

1759    Data type representing IPVC End Point Identifier and CoS name use for Bandwidth Profiles.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| ipvcEndPointIdentifier | IpvcEndPointIdentifier | 0..1 | IPVC End Point identifier as described in MEF 61.1 Section 11.1. Note - It points to the value of IpvcEndPoint.identifier Service Attribute. It is not intended to point to the potential Product identifier if IPVC Endpoint is an instance of a Product. |
| cosName | string | 0..1 | Class of Service Name. |

1760

**Table 103 IpvcEpAndCosName**

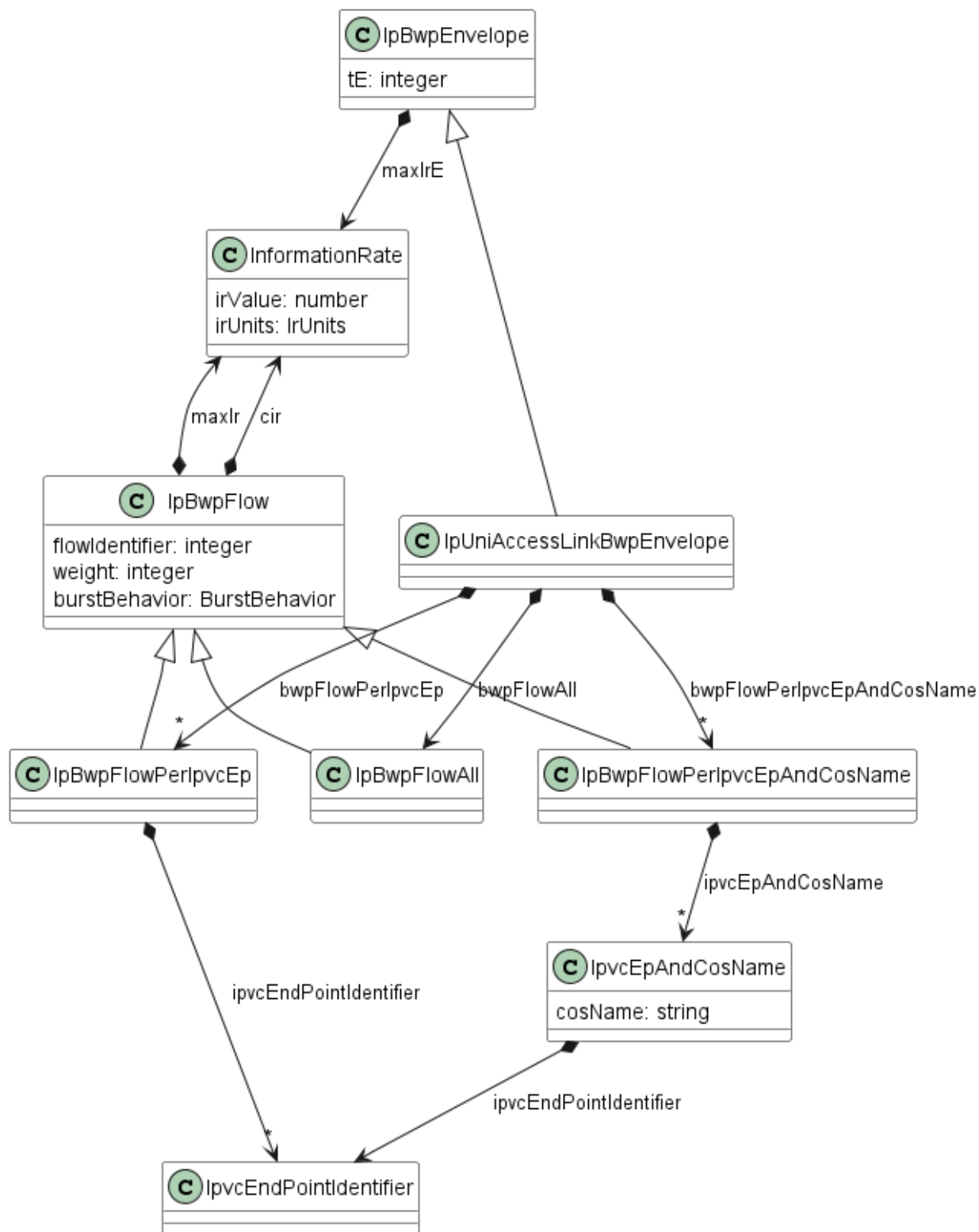1761 ## 14.14    IP UNI Access Link Bandwidth Profile Envelope

1762

1763 **Figure 29 IpUniAccessLinkBwpEnvelope**

1764 Figure 29 IP UNI Access Link Bandwith Profile Envelope exiends the IpBwpEnvelope to specify
1765 possiblies of Flow configurations that can be applie at the IP UNI Access Link.

### 14.14.1 IpUniAccessLinkBwpEnvelope

1767 File: \ip\common\ipCommon.yaml

1768 A single Bandwidth Profile Envelope consisting of parameters and Bandwidth Profile Flow
1769 specifications. An Ingress Bandwidth Profile Envelope can be specified for one of a UNI, a UNI
1770 Access, or an IPVC EP. Reference MEF 61.1 Section 13.10 UNI Access Link Ingress Bandwidth
1771 Profile Envelope Service Attribute.

1772 Inherits from: IpBwpEnvelope

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| bwpFlowPerIpvcEp | IpBwpFlowPerIpvcEp[] | 0..* | List of BWP FLows matching IPVC Endpoint Identifier(s) for an IPVC EP located at the UNI. |
| bwpFlowAll | IpBwpFlowAll | 0..1 | BWP Flow for all IP Data Packets at the UNI that are transmitted or received over the UNI Access Link. |
| bwpFlowPerIpvcEpAndCosName | IpBwpFlowPerIpvcEpAndCosName[] | 0..* | List of BWP FLows matching pairs of IPVC Endpoint Identifier and CoS Name. |

1773 **Table 104 IpUniAccessLinkBwpEnvelope**

1774 oneOf:

1775 - required: [bwpFlowAll]

1776 - required: [bwpFlowPerIpvcEp]

1777 - required: [bwpFlowPerIpvcEpAndCosName]

1778 **14.15     BasicIaBwpEnvelope**
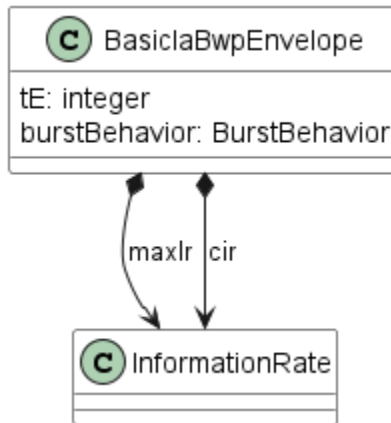


1779

1780                                    **Figure 30 BasicIaBwpEnvelope**

1781   Figure 30 shows a simple model of BasicIaBwpEnvelope. It leverages MEF 69.1 [10] requirements
1782   to Basic Internet Access and simplifies the model , comaring to the advanced one.

1783   **14.15.1     BasicIaBwpEnvelope**

1784   File: \ip\common\ipCommon.yaml

1785   A single Bandwidth Profile Envelope simplified for the use of Basic Internet Access. For Basic
1786   Internet Access there must always be exactly one Class of Service Name, exactly one IPVC End
1787   Point at the UNI and exactly one UNI Access Link, none of the other options are needed. There
1788   can also be one flow, so the flowIdentifier and weight are also omitted for the flow. maxIr is omitted
1789   from the Envelope - resulting in flattened BasicIaBwpEnvelope class containing four attributes:
1790   the Envelope IR Time tE, and the cir, maxIr and burstBehavior for the single BWP Flow. This special
1791   case envelope is used for the UNI, IPVC End Point and UNI Access Links cases for Basic Internet
1792   Access. Reference MEF 61.1 Section 11.11 IPVC EP Egress Bandwidth Profile Envelope Service
1793   Attribute.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| maxIr | InformationRate | 0..1 | Maximum Information Rate in bits per second. Limit on the average information rate of IP Packets for this BWP Flow. Reference MEF 61.1 Table 29 - Bandwidth Profile Parameters for a Bandwidth Profile Flow. |
| tE | integer | 0..1 | The Envelope IR Time in milliseconds. This is the time period over which average Information Rates are calculated and thus it limits the size of a burst. Reference MEF 61.1 Section 17.3 Bandwidth Profile Envelopes. |
| burstBehavior | BurstBehavior | 0..1 | Burst Behavior either Optimize-Delay or Optimize-Throughput. Whether the SP is requested to optimize the delay characteristic of this flow, or the throughput. Reference MEF 61.1 Table 29 - Bandwidth Profile Parameters for a Bandwidth Profile Flow. |
| cir | InformationRate | 0..1 | Committed Information Rate in bits per second. Average information rate of IP Packets that is committed to this BWP Flow. Reference MEF 61.1 Table 29 - Bandwidth Profile Parameters for a Bandwidth Profile Flow. |

1794 **Table 105 BasicIaBwpEnvelope**

1795 ## 14.16    IP Addressing



1796
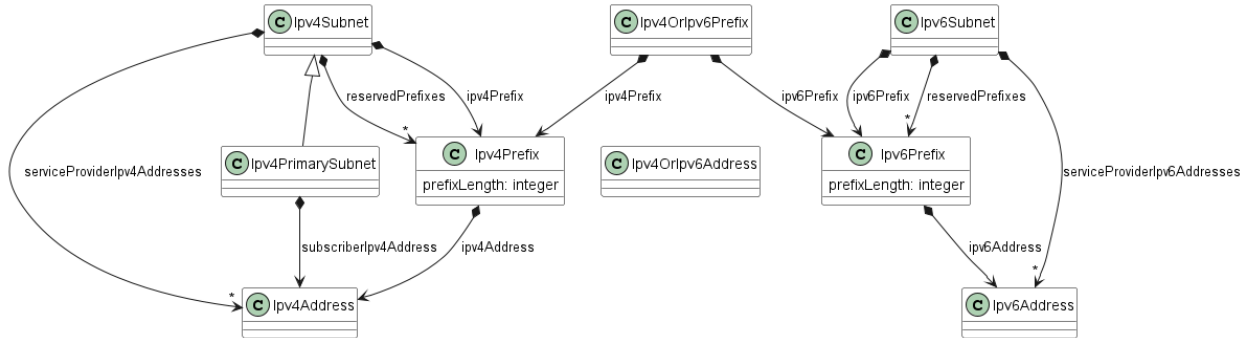
1797 **Figure 31 IP Addressing**

1798 Figure 31 Illustrates the model of IPv4 and IPv6 addressing. Note that the API schema leverages
1799 the OAS embedded *ipv4* and *ipv6* string formats and uses them to specify the *Ipv4Address* and
1800 *Ipv6Address* data types that are uses whenever an address value must be provided.

1801 ### 14.16.1    Ipv4Address

1802 File: \ip\common\ipCommon.yaml

1803 Data type representing IPv4 address.

1804 Format: ipv4

1805    **14.16.2    Ipv4Prefix**

1806    File: \ip\common\ipCommon.yaml

1807    Data type representing IPv4 address prefix and mask length between 0 and 31 bits.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| prefixLength | integer | 0..1 | IPv4 address prefix. Length 0-31. |
| ipv4Address | Ipv4Address | 0..1 | IPv4 address. |

1808                                    **Table 106 Ipv4Prefix**

1809    **14.16.3    Ipv4Subnet**

1810    File: \ip\common\ipCommon.yaml

1811    Ipv4Subnet is a data type representing an IPv4 subnet logical partition of an IP network. Included
1812    is list of Service Provider IPv4 addresses.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| serviceProviderIpv4Addresses | Ipv4Address[] | 1..* | List of Service Provider IPv4 addresses. [R25] If IPv4 Connection Addressing is DHCP, the UNI Access Link IPv4 Connection Addressing Primary Subnet parameter MUST contain only a single Service Provider IPv4 Address. Reference MEF 69.1[1] Section 9.4 Subscriber Internet Access Service: UNI Access Link Requirements |
| reservedPrefixes | Ipv4Prefix[] | 0..* | List of IPv4 Prefixes, possibly empty |
| ipv4Prefix* | Ipv4Prefix | 1 | IPv4 address prefix (IPv4 address prefix and mask length between 0 and 31 in bits). |

1813                                    **Table 107 Ipv4Subnet**

1814    **14.16.4    Ipv4PrimarySubnet**

1815    File: \ip\common\ipCommon.yaml

1816    Ipv4Subnet used in context of Primary Ipvc subnet. It adds the subscriberIpv4Address attribute to
1817    the Ipv4Subnet.

1818    Inherits from: Ipv4Subnet

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| subscriberIpv4Address | Ipv4Address | 0..1 | Subscriber IPv4 Address |

1819                                   **Table 108 Ipv4PrimarySubnet**

1820  **14.16.5   Ipv6Address**

1821  File: \ip\common\ipCommon.yaml

1822  Data type representing IPv6 address.

1823  Format: ipv6

1824  **14.16.6   Ipv6Prefix**

1825  File: \ip\common\ipCommon.yaml

1826  Data type representing IPv6 address prefix and mask length between 0 and 127 in bits.

| Name | Type | Multiplicity | Description |
|------|------|--------------|-------------|
| prefixLength | integer | 0..1 | IPv6 address prefix. Length 0-127. |
| ipv6Address | Ipv6Address | 0..1 | IPv6 address. |

1827                                   **Table 109 Ipv6Prefix**

1828  **14.16.7   Ipv6Subnet**

1829  File: \ip\common\ipCommon.yaml

1830  IPv6Subnet is a data type representing an IPv6 subnet logical partition of an IP network. Included
1831  is list of Service Provider IPv6 addresses.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| reservedPrefixes | Ipv6Prefix[] | 0..* | List of IPv6 Prefixes, possibly empty |
| serviceProviderIpv6Addresses | Ipv6Address[] | 1..* | List of IPv6 Service Provider addresses. |
| ipv6Prefix* | Ipv6Prefix | 1 | IPv6 Prefix (IPv6 address prefix and mask length between 0 and 127 in bits). |

1832                                   **Table 110 Ipv6Subnet**

1833    **14.16.8    Ipv4OrIpv6Address**

1834    File: \ip\common\ipCommon.yaml

1835    Data type representing IPv4 or IPV6 address.

1836    oneOf:

1837      - format: ipv4

1838      - format: ipv6

1839    **14.16.9    Ipv4OrIpv6Prefix**

1840    File: \ip\common\ipCommon.yaml

1841    IPv4 or IPv6 prefix. Includes subnet address and prefix length.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| ipv6Prefix | Ipv6Prefix | 0..1 | IPv6 prefix. |
| ipv4Prefix | Ipv4Prefix | 0..1 | IPv4 prefix. |

1842                                   **Table 111 Ipv4OrIpv6Prefix**

1843    oneOf:

1844      - required: [ipv4Prefix]

1845      - required: [ipv6Prefix]

1846 ## 14.17    Common IP Classes

1847 This section describes classes that are present in the ipCommon.yaml file, yet are not strictly
1848 related to IP technology.

1849 ### 14.17.1    EnabledDisabled

1850 File: \ip\common\ipCommon.yaml

1851 Enumeration to indicate Enabled/Disabled state of an attribute

| Value |
|---|
| ENABLED |
| DISABLED |

1852 **Table 112 EnabledDisabled**

1853 ### 14.17.2    InformationRate

1854 File: \ip\common\ipCommon.yaml

1855 A value and a unit of measure that specifies an Information Rate.

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| irValue | number | 0..1 | The value in the information rate. For example if the information rate is 70 kbps this element is 70. |
| irUnits | IrUnits | 0..1 | The unit of measure for the Information Rate. For example if the Information Rate is 70KBPS this element is KBPS. Note that the values are decimal values. 1 KBPS is 1000 bits per second and 1MBPS is 1,000,000 bits per second. |

1856 **Table 113 InformationRate**

1857 ### 14.17.3    IrUnits

1858 File: \ip\common\ipCommon.yaml

1859 The unit of measure for the Information Rate. For example if the Information Rate is 70KBPS this
1860 element is KBPS. Note that the values are decimal values. 1 KBPS is 1000 bits per second and
1861 1MBPS is 1,000,000 bits per second.

---

| Value |
| --- |
| BPS |
| KBPS |
| MBPS |
| GBPS |
| TBPS |
| PBPS |
| EBPS |
| ZBPS |
| YBPS |

**Table 114 IrUnits**

### 14.17.4 Percentage

File: \ip\common\ipCommon.yaml

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

### 14.17.5 TimeDuration

File: \ip\common\ipCommon.yaml

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

| Name | Type | Multiplicity | Description |
|---|---|---|---|
| timeDurationValue* | integer | 1 | The value of the duration. For example, if the duration is 20 ms, this element is 20. |
| timeDurationUnits* | TimeDurationUnits | 1 | The unit of measure in the duration. For example, if an interval is 2ms, this element is MS. |

1870

**Table 115 TimeDuration**

1871 ### 14.17.6 TimeDurationUnits

1872 File: \ip\common\ipCommon.yaml

1873 The unit of measure in the duration. For example, if an interval is 2ms, this element is MS.

| Value |
|---|
| NS |
| US |
| MS |
| SEC |
| MIN |
| HOUR |
| DAY |
| WEEK |
| MONTH |
| YEAR |

1874

**Table 116 TimeDurationUnits**

## 15 References

[1] IETF JSON Schema draft 7, *JSON Schema: A Media Type for Describing JSON Documents* and associated documents, by Austin Wright and Henry Andrews, March 2018. Copyright © 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

[2] IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, March 1997

[3] IETF RFC 4862, *IPv6 Stateless Address Autoconfiguration,* September 2007

[4] IETF RFC 8174, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, May 2017

[5] Internet Engineering Task Force RFC 4271, *A Border Gateway Protocol 4 (BGP-4),* by Dr. Yakov Rekhter, January 2006. Copyright © The Internet Society (2006). All Rights Reserved.

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

[7] MEF 57.2, Draft Release 4 Product Order Management Requirements and Use Cases, May 2022

[8] MEF 61.1, IP Service Attributes, May 2019

[9] MEF 61.1.1, Amendment to MEF 61.1: UNI Access Link Trunks, IP Addresses, and Mean Time to Repair Performance Metric, July 2022

[10] MEF 69.1, Subscriber IP Service Definitions, February 2022

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

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

[13] MEF 80, Quote Management Requirements and Use Cases, July 2021

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

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

[16] MEF 106, LSO Sonata Access E-Line Product Schemas and Developer Guide, October 2022

1906     [17]  MEF 115, LSO Cantata and LSO Sonata Quote Management API – Developer Guide,
1907           May 2022

1908     [18]  MEF 116, LSO Cantata and LSO Sonata Inventory Management API – Developer
1909           Guide, May 2022

1910     [19]  MEF W123, LSO Cantata and LSO Sonata Product Order Management API –
1911           Developer Guide, December 2022

1912