Contribution Number

Letter Ballot

MEF W139

Internet Access Product Schemas and Developer Guide

November 2023

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

Disclaimer

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

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

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

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

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

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

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

Table of Contents

List of Figures

List of Tables

1 List of Contributing Members

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

- Amartus
- Cisco
- Colt
- Lumen

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 [17] and MEF 61.1.1 [18]. The Basic and Advanced services are specified in MEF 69.1 [19] based on the Service Attributes defined in MEF 61.1 [17].

MEF 61.1 [17] and MEF 61.1.1 [18] specify Service Attributes 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 the GitHub repository https://github.com/MEF-GIT/MEF-LSO-Sonata-SDK, commit id: c43f8884c305631717cf9eea08be5937d9b48198:

productSchema/ip/

- common/ipCommon.yaml
- common/ipSls.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
- ipUni/ipUniCommon.yaml

Also included in the GitHub repository is a Postman file that contains informative examples illustrating use of the Internet Access. This file is not part of this standard but is referred to in Appendix A.

- documentation/productSchema/ip/internetAccess/MEF 139 - Appendix A.postman_collection.json

3 Terminology and Abbreviations

This document does not define any new terms or definitions. All of the terms defined in the standards below are included in this document by reference:

- MEF 55.1 Lifecycle Service Orchestration (LSO): Reference Architecture and Framework [14]
- MEF 55.1.1 Amendment to MEF 55.1: Reference Architecture and Framework - Terminology [15]
- MEF 57.2 Product Order Management Requirements and Use Cases [16]
- MEF 61.1 IP Service Attributes [17]
- MEF 61.1.1 Amendment to MEF 61.1: UNI Access Link Trunks, IP Addresses, and Mean Time to Repair Performance Metric [18]
- MEF 69.1 Subscriber IP Service Definitions [19]
- MEF 79 Address, Service Site, and Product Offering Qualification Management, Requirements and Use Cases [20]
- MEF 106 LSO Sonata Access E-Line Product Schemas and Developer Guide [24]

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 [5], RFC 8174 [10]) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

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

5 Numerical Prefix Conventions

This document uses the prefix notation to indicate multiplier values as shown in Table 1.

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

Table 1 Numerical Prefix Conventions

6 Introduction

LSO Cantata and LSO Sonata provide a programmatic interface for establishing (quoting, ordering, etc.) products between a Seller and a Buyer). In the case of LSO Cantata the Seller is a Service Provider and the Buyer is a Subscriber. In the case of LSO Sonata the Buyer is a Service Provider and the Seller is a Partner. 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 [17] and MEF 61.1.1 [18]. The Service definitions for Basic and Advanced Internet Access which are, in effect, a set of constraints on the values of the Service Attributes, are provided in MEF 69.1 [19].

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 IPVC 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 7)
- An overview of the Internet Access Service (section 8)
- Data Model Design Principles (section 9)
- Order Milestones (section 0)
- An abbreviated Information Model for Internet Access and explanation of the organization of the Service Attributes in MEF 61.1 [17] and MEF 61.1.1 [18] (section 10)
- Organization of the Data Model for Internet Access (section 12)
- The relationship between the entities in the service (section 13)
- The detailed comparison of Service Attributes of Basic and Advanced Products with a list of Service Attributes that are not included in the Data Model (section 0)

These are followed by two sections that contain tables that describe the details of the data model. The tables include information about each class and a list of properties in each class. For each property, the JSON Name, description, data type, cardinality, details about allowed values, and, in

some cases, some additional information about relationships between Service Attributes are provided.

7 Overview of LSO Cantata and LSO Sonata

MEF 55.1 [14] 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 [20]
- Quote, MEF 80 [21]
- Product Ordering, MEF 57.2 [16]
- Product Inventory, MEF 81 [22]

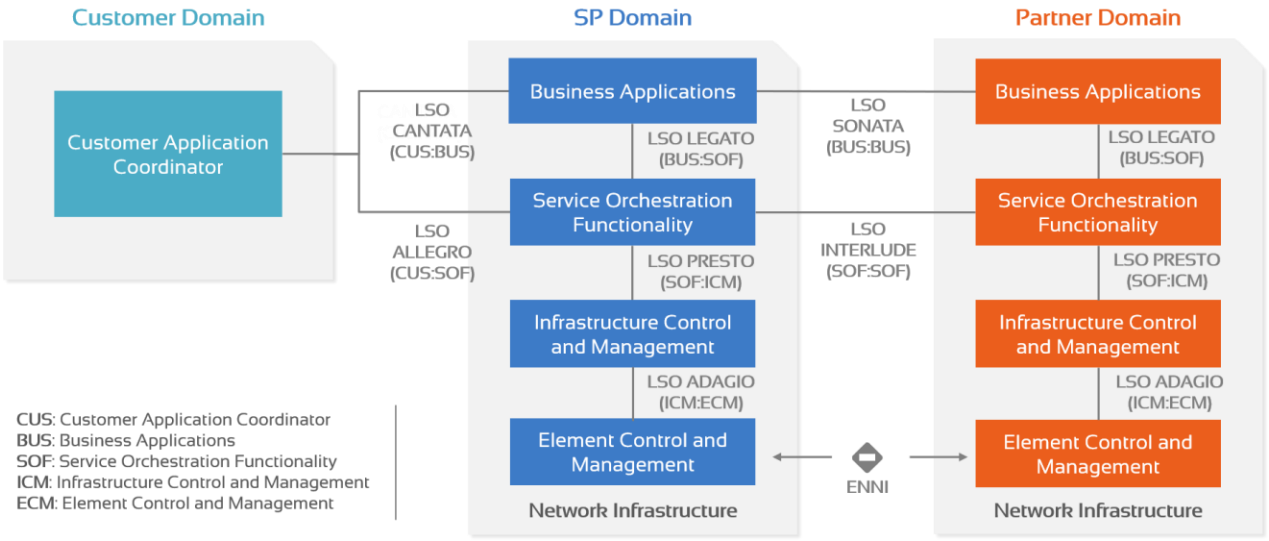


Figure 1 LSO Cantata and LSO Sonata Reference Diagram

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

- Product Offering Qualification, MEF 87 [23]
- Quote, MEF 115 [25]

- Product Ordering, MEF 123 [28]
- Product Inventory, MEF 116 [26]

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

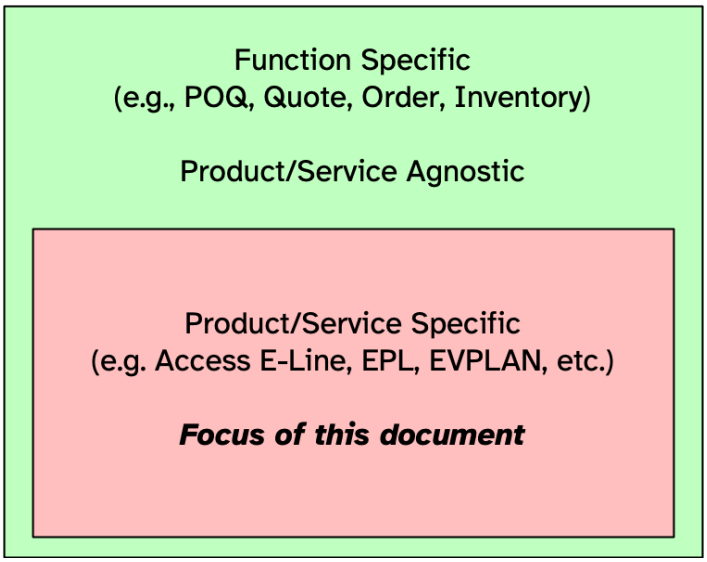


Figure 2 LSO Cantata and LSO Sonata API Structure

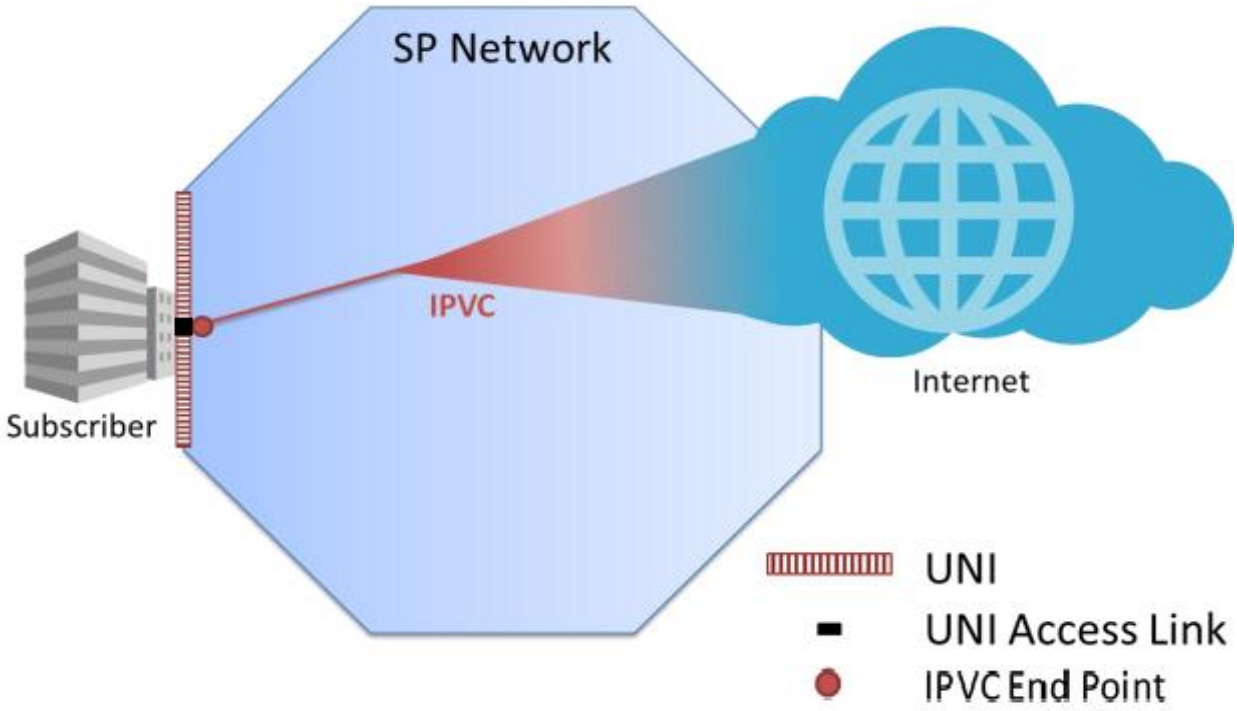
This document describes the product-specific Data Model for a MEF Internet Access service as defined in MEF 69.1 [19].

8 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 that UNI 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 UNI Access 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 (MEF 61.1, IP Service Attributes [17])**

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 provides access to the Internet,
535 and the Internet itself (as shown in the Figure above). The IPVC thus has only one IPVC End Point
536 at the UNI that connects to the Subscriber but does not have one that would connect it to the
537 Internet.

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

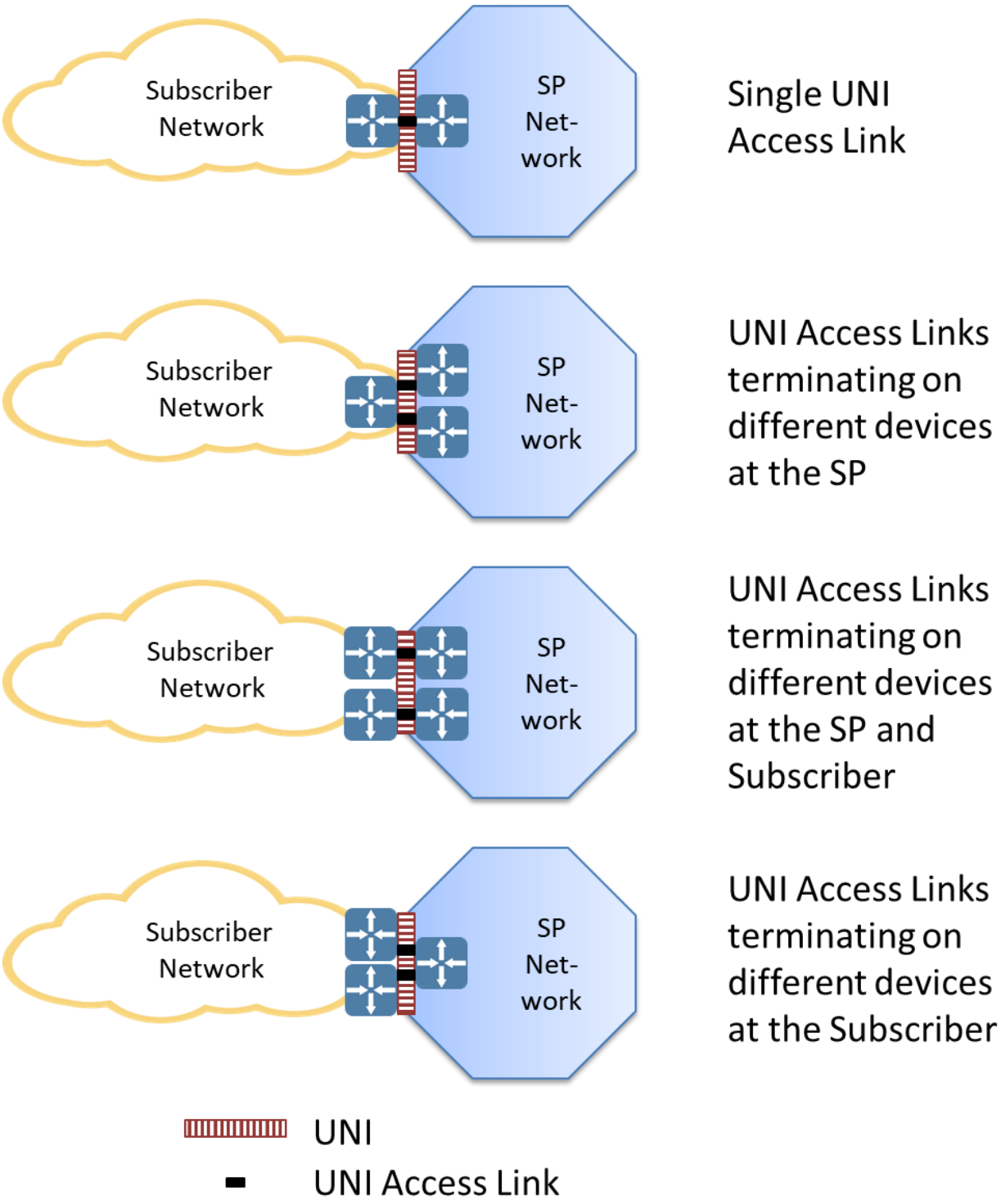


Figure 4 Examples of UNI Access Links in a Single UNI concept (MEF 61.1, IP Service Attributes [17]

Two types of Internet Access are defined in MEF 69.1 [19]: Basic and Advanced. The possible values for certain Service Attributes differ between these two types.

Basic Internet Access is typically delivered to residential dwellings. It may be offered to small/medium businesses. Its service characteristics typically include:

- plug-and-play ease of use
- low-cost
- for IPv4, a few (or shared) publicly routed addresses

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

- redundancy features
- dynamic routing protocol support (e.g., BGP [1] routing)
- options for Subscriber-supplied IP addressing
- proactive monitoring to support a Service Level Specification (SLS)

This standard additionally introduces a version of Advanced Internet Access, called Exclusive Advanced Internet Access. The details and rationale are presented in section 10.

9 Data Model Design Principles and Assumptions

A Service Attribute for a Product can have a value that is a simple datatype such as an integer or string (or list of simple datatypes) or a value that is an object with multiple properties such as a Bandwidth Profile or a composition of objects such as an 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.
    - Mandatory - attributes that must be provided by the Buyer in a POQ/Quote/Order request or must be returned by the Seller for an Inventory request as specified in section 9.1.
    - Optional - attributes that may be provided by the Buyer in a POQ/Quote/Order request and may be returned by the Seller for an Inventory request as specified in section 9.2.
    - Fixed - attributes that are hard-coded and may be specified by the Buyer in a POQ/Quote/Order request (subject to agreement between the Buyer and Seller) and may be returned by the Seller for an Inventory request (subject to agreement between the Buyer and Seller) as specified in section 9.3.

As noted above, the classification may depend on:

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

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

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

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

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

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

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

The 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.

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

9.1 Mandatory Product-Specific Attributes

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

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

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

9.2 Optional Product-Specific Attributes

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

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

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

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

9.3 Fixed Product-Specific Attributes

A Product-Specific Attribute may be classified as Fixed for a Business Function, Product Action, and Product Offering when only one value is applicable for the Seller. This can be the case for example if:

- the Seller supports only a single value, or
- the value is derived by the Seller from the value of one or more other Product-Specific Attributes, or
- the Seller specifies a single value in the Product Catalog for a specific Product Offering, or
- the Buyer and the Seller agree on a single value during onboarding

Since these are Product-Specific Attributes, each value must still be agreed 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.

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

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

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

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

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

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

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

694

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 15.4)
- One IP UNI where the Subscriber accesses the service (see section 15.5).
- Exactly one IPVC End Point for the IPVC at this IP UNI. (see section 15.4.3).
- One (for Basic and Exclusive Internet Access) or more UNI Access Links in each UNI, (see section 15.6).
- One (for Basic and Exclusive Internet Access) or more UNI Access Link Trunks each carrying one or more UNI Access Links (see section 15.7).

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.
- 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 not provided as Product Specific Attributes. They are handled by the API (POQ, Quote, Order, Inventory) model attributes (as specified in section 13). The source and target of such relations on the diagrams are bound to objects that are their logical sides, yet technically the relation is on the root product level.

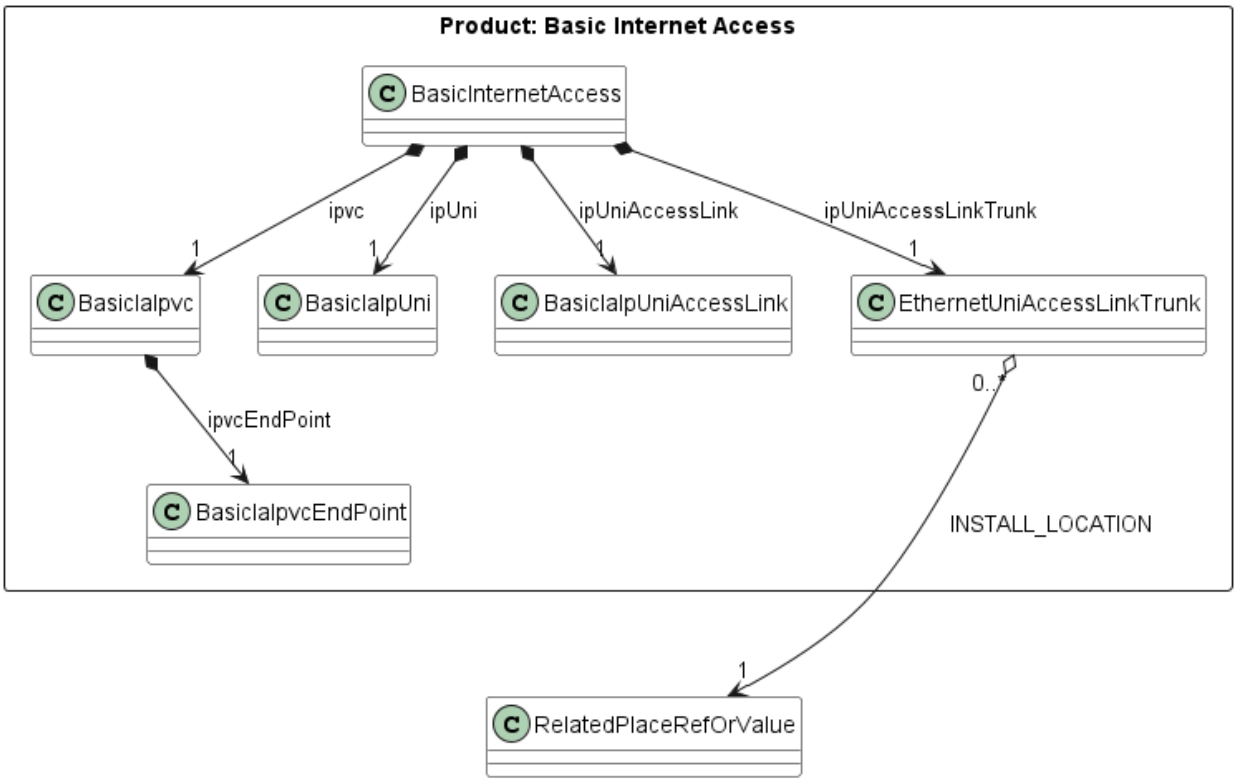


Figure 5 Information model for Basic Internet Access product

Figure 5 presents the information model for Basic Internet Access. MEF 69.1 [19] defines restrictions for Basic Internet Access such that all relations' cardinalities have exactly the value of 1 and that they are exclusive for a given product instance (meaning that all components serve only one IPVC). In other words, a Basic Internet Access Service comprises one IPVC with one IPVC End Point located at one UNI, and that UNI comprises one UNI Access Link which is transported over one UNI Access Link Trunk; moreover, all of these are dedicated to the Basic Internet Access Service and cannot be used for or shared with any other service. MEF 61.1.1 [18], which introduces the UNI Access Link Trunk, does not specify such requirement as it only defines the Service Attributes, yet this document adds such restriction for consistency with the Basic Internet Access specifics. This allows this product to be modeled in a simplified way as one main type (BasicInternetAccess) having all components as single ref attributes. This means that all components (IPVC, IPVC End Point, UNI, UNI Access Link, and UNI Access Link Trunk) are ordered with a single Product Order Item. Since all components are within the same order, the only API-level relation is the one to a place. It is the UNI Access Link Trunk that is the logical owner of the relationship

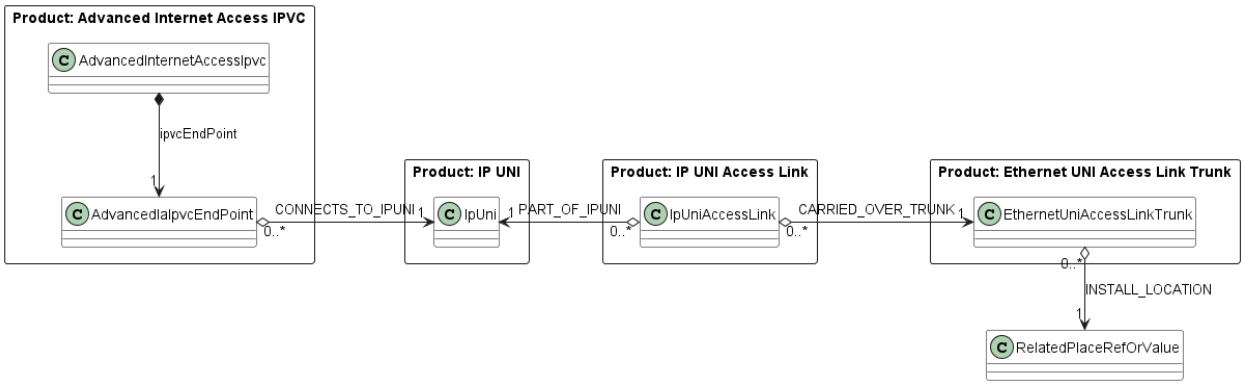


Figure 6 Information model for Advanced Internet Access product

Figure 6 shows the building blocks of an Advanced Internet Access product. It implements the Advanced flavor of Internet Access as specified by MEF 69.1 [19]. It is a set of 4 distinct products that must be ordered separately by different Product Order Items of one (or more) Product Orders. Note the main differences compared to Basic Internet Access:

- The IPVC and IPVC End Point contain product specific attributes for Advanced Internet Access, per MEF 69.1.
- IP UNI can serve more than one Advanced Internet Access product (and possibly other IP products such as IP VPN).
- IP UNI can be provided by more than one IP UNI Access Link.
- Ethernet UNI Access Link Trunk can serve more than one IP UNI Access Link.
- All relations between components are specified by API product or item relationships (as specified in section 13).
- The place relationship is specified by the Ethernet UNI Access Link Trunk product.

Note an Advanced Internet Access service can use the same IP UNI (and hence the same IP UNI Access Links and UNI Access Link Trunks) as other IP Services; hence there is nothing specific to Internet Access in the definition of the IP UNI, IP UNI Access Link or Ethernet UNI Access Link Trunk products.

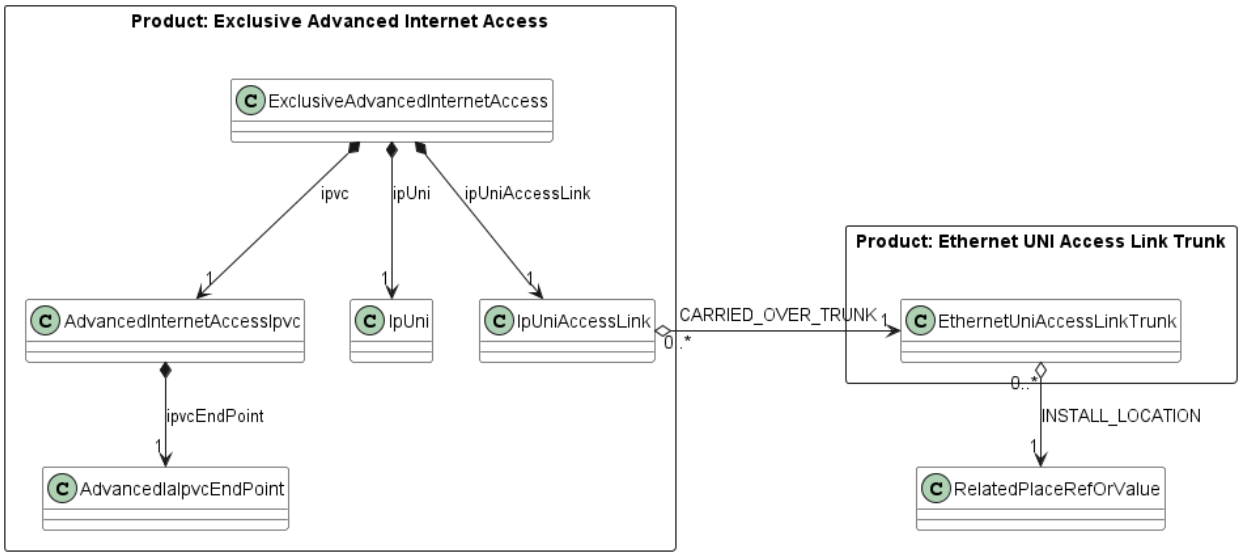


Figure 7 Information model for Exclusive Advanced Internet Access product

MEF 69.1 [19] defines 2 types of Internet Access: Basic and Advanced. They differ by several requirements summarized in section 14. However, the flexibility of Advanced Internet Access comes with the burden of having to order 4 different products. This burden is partially mitigated by introducing the Exclusive Advanced Internet Access product. It is still an Advanced Internet Access as specified by MEF 69.1 [19] but adds some assumptions that cover most of the probable common deployment configurations. These are:

- The IP UNI is dedicated exclusively (hence the product name) to the Advanced Internet Access service (IPVC), no other services can be run over that IP UNI.
- The IP UNI comprises one IP UNI Access Link.

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

The three cases described above are composed of six products defined in this document:

- Basic Internet Access
- Advanced Internet Access IPVC
- IP UNI
- IP UNI Access Link
- Ethernet UNI Access Link Trunk
- Exclusive Advanced Internet Access

10.1 Organization of Service Attributes

The data model of Internet Access products is based on Service Attributes defined in MEF 61.1 [17], and MEF 61.1.1 [18], and implements Service Definition Requirements as specified in MEF 69.1 [19] Section 9. These requirements result in Basic and Advanced versions being a variation of Service Attributes defined in MEF 61.1 [17]. A set of Common classes is introduced in the data model to gather the attributes shared by Basic and Advanced flavors. Note that the Common types are not as specified by MEF 61.1 [17] or MEF 61.1.1 [18] but only subsets of them.

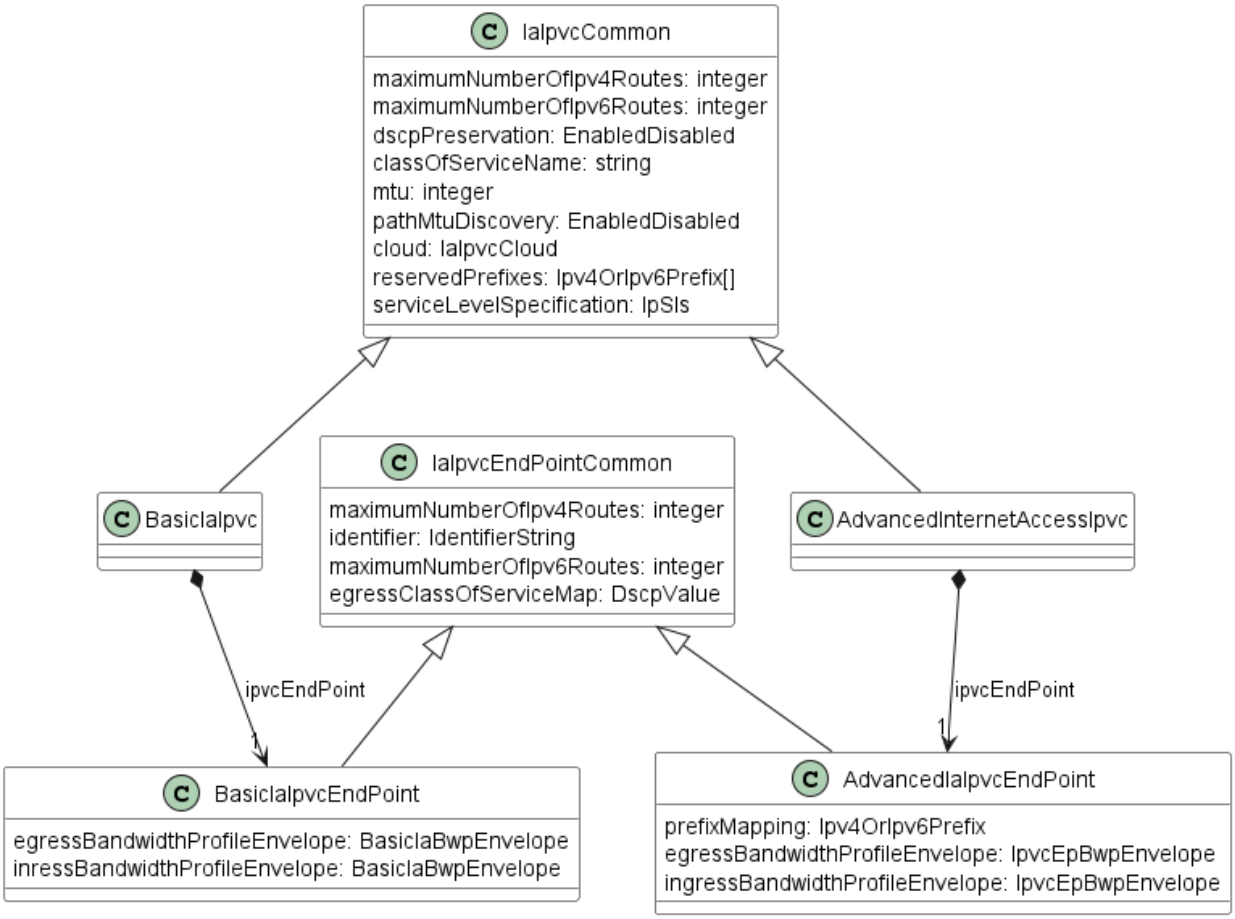


Figure 8 IPVC and IPVC End Points Common classes

Figure 8 presents the organization of Common IPVC and IPVC End Point types and differences in their respective Basic and Advanced subtypes. The IPVC flavors differ only by the type of referenced IPVC End Points which have different types used for ingress and egress bandwidth profiles envelopes and the AdvanceIaIpvcEndPoint additionally specifies prefixMapping. The details of the differences are described in section 0.

The naming convention is to have a full version of “InternetAccess” in the types that are orderable products and the abbreviation “Ia” in others.

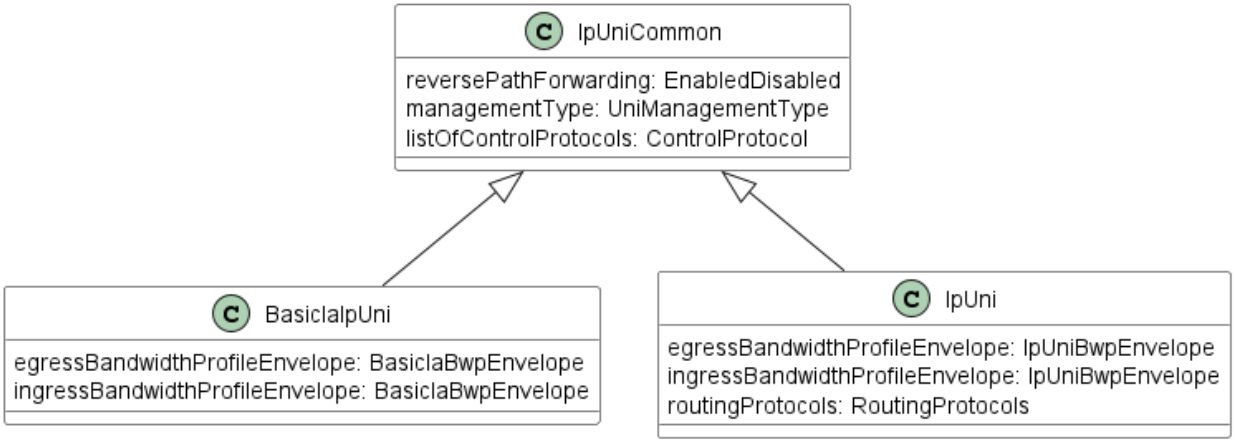


Figure 9 IP UNI Common class

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

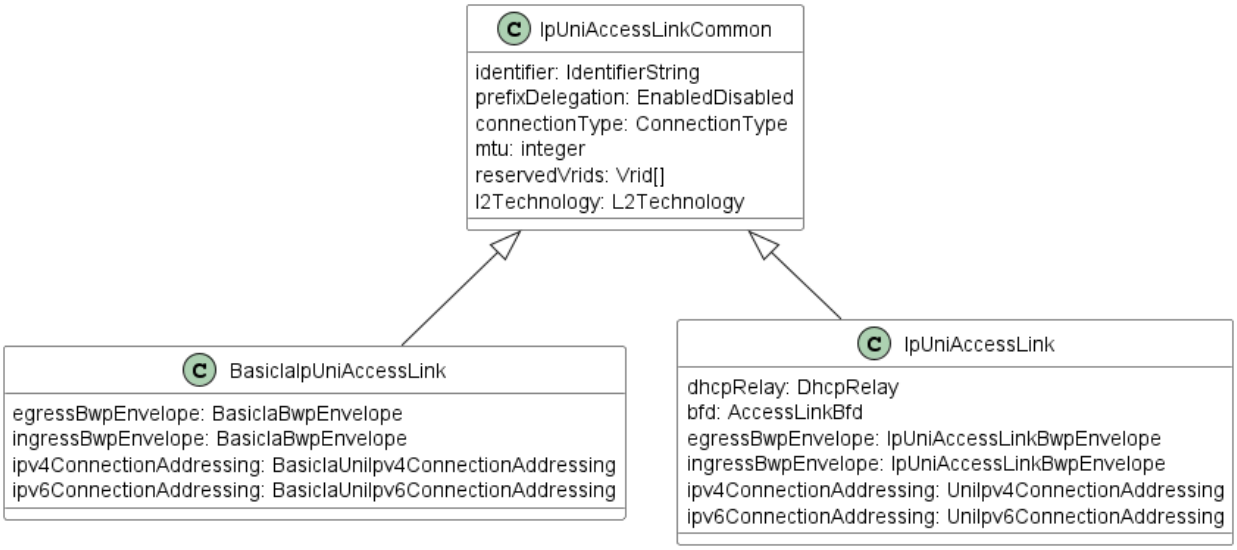
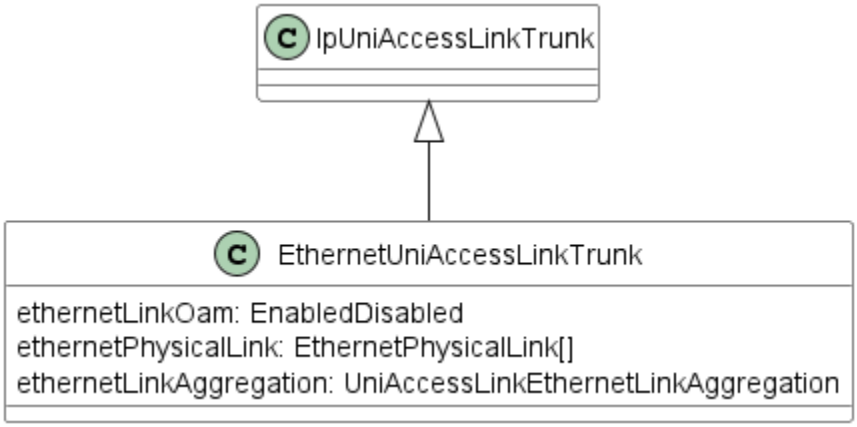


Figure 10 IP UNI Access Link

Figure 10 shows the differences between the Basic and Advanced flavors of IpUniAccessLink. They differ in how the Bandwidth profile envelope and IPv4/IPv6 Connection Addressing are specified. Additionally, IpUniAccessLink allows the specification of DHCP relay and BFD attributes. As in the IpUni case - only BasicIaIpUniAccessLink is Internet Access specific. IpUniAccessLink may also be used by other IP products.



811

812 **Figure 11 Ethernet UNI Access Link Trunk**

813 The Ethernet UNI Access Link Trunk has the same representation in all three Internet Access
814 models. Figure 11 shows its inheritance from IpUniAccessLinkTrunk. MEF 61.1.1 [18] specifies
815 Ethernet UNI Access Link Trunk as the only available implementation of the
816 IpUniAccessLinkTrunk.

817

11 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 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).

The Milestone Value in the first column of **Błąd! Nie można odnaleźć źródła odwołania.** is included in ProductOrderItem.milestone and ProductOrderEventPayload.milestoneName in the Product Order API (see MEF 123 [28]).

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

Milestone Value	Description	Basic Internet Access	Advanced Internet Access Ipvc	Ip Uni	Ip Uni Access Link	Ethernet Uni Access Link Trunk	Exclusive Advanced Internet Access
SITE_SURVEY_SCHEDULED	Site Survey Scheduled	X				X	
SITE_SURVEY_COMPLETE	Site Survey Complete	X				X	
PLANNING_COMPLETE	Planning Complete	X	X	X	X	X	X
FIRM_DELIVERY_DATE_PROVIDED	Firm Delivery Date Provided	X	X	X	X	X	X
AWAITING_MUNICIPAL_APPROVAL	Awaiting Municipal Approval	X				X	
MUNICIPAL_APPROVAL_GRANTED	Municipal Approval Granted	X				X	
AWAITING_LANDLORD_APPROVAL	Awaiting Landlord Approval	X				X	
LANDLORD_APPROVAL_GRANTED	Landlord Approval Granted	X				X	
CONSTRUCTION_STARTED	Construction Started	X				X	
CONSTRUCTION_COMPLETED	Construction Completed	X				X	
AWAITING_ACCESS	Awaiting Site Access Permission (for the end-to-end test)	X	X		X	X	X
ACCESS_DENIED	Site Access Denied (for the end-to-end test).	X	X		X	X	X
AWAITING_WIRING	Awaiting Installation of Inside Wiring by Landlord	X				X	
WIRING_COMPLETE	Installation of Inside Wiring by Landlord Complete	X				X	
EQUIPMENT_DISPATCHED	Equipment Dispatched	X				X	
EQUIPMENT_DELIVERED	Equipment Delivered	X				X	
EQUIPMENT_INSTALLED	Equipment Installed	X				X	
E2E_TESTING_SCHEDULED	End-to-End Testing Scheduled	X	X		X	X	X
E2E_TESTING_COMPLETED	End-to-End Testing Completed	X	X		X	X	X
E2E_TESTING_FAILED	End-to-End Testing Failed.	X	X		X	X	X

828 **Table 2 Order Milestones for Internet Access**

12 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.

12.1 Organization and Structure of the Schemas

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

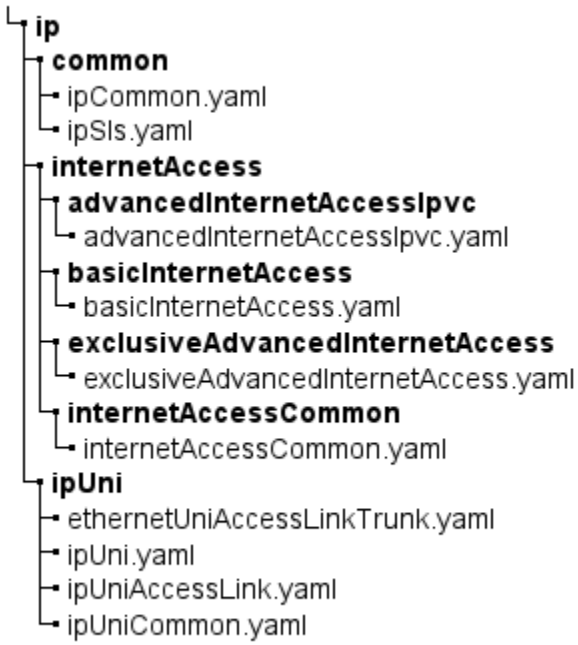


Figure 12 Schema Files Organization

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 directory holds schemas for separately orderable products that are building blocks for Advanced and Exclusive Advanced Internet Access Products, and also the ipUniCommon.yaml. The common directory keeps the definition of types that are shared among other IP services.

12.2 Additional Details

This section includes an explanation of some naming conventions and other patterns used.

12.2.1 Naming Conventions

In the schemas following naming conventions are used:

- class and type names are UpperCamelCase,
- Service Attribute/property names are lowerCamelCase,
- enumeration values are defined using UPPER_SNAKE_CASE.

12.2.2 IPVC End Point Service Attribute

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

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

Note that one of the IPVC End Point Service Attributes is IPVC End Point EI Type ([17] section 11.2) which can be “UNI” or “ENNI”. Since this information is implicit, this Service Attribute is not included in the schema for Internet Access but likely would be included for other IP Services.

12.2.3 Identifiers

There are two patterns of identifying objects defined in this document.

For objects that are separately orderable products, there is no explicit Identifier attribute defined. These products are identified by product.id attribute of the ProductOfferingQualificationItem, QuoteItem, or ProductOrderItem of POQ, Quote, Product Order APIs (respectively), and also Product.id in the Product Inventory API. This identifier attribute is set by the Seller. This applies to all defined products. The bracket shows which components the identifier applies to:

- BasicInternetAccess (IPVC)
- AdvancedInternetAccessIpvc (IPVC)
- IpUni (IP UNI)
- IpUniAccessLink (IP UNI Access Link)
- EthernetUniAccessLinkTrunk (Ethernet Uni Access Link Trunk)
- ExclusiveAdvancedInternetAccess (IPVC)

For entities that require to be referenced by other entities that are not separate products, an explicit identifier attribute is provided which is set by the Buyer. This allows the Buyer to specify the relationships prior to identifiers being set by the Seller. This applies to:

- IaIpvcEndPointCommon, subclassed by:
  - BasicIaIpvcEndPoint
  - AdvancedIaIpvcEndPoint
- IpUniAccessLinkCommon, subclassed by:
  - BasicIaIpUniAccessLink
  - IpUniAccessLink

IpUniAccessLink as a result may be identified in two ways.

There are cases where an object has no identifier assigned because it is included in another product structure and can be uniquely identified by the corresponding product.id:

- IP UNI used in Basic Internet Access
- IP UNI used in Exclusive Advanced Internet Access
- Ethernet Uni Access Link Trunk used in Basic Internet Access

13 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 Product 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.

The INSTALL/CHANGE column specifies whether the given relation is mandatory or optional to be provided per respective operation.

The final column notes if 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**	Multiple Allowed at POQ and Quote
Advanced Internet Access IPVC	CONNECTS_TO_IPUNI	Mandatory	**IP UNI**	1	NO
IP UNI Access Link	PART_OF_IPUNI	Mandatory	**IP UNI**	1	NO
IP UNI Access Link	CARRIED_OVER_TRUNK	Mandatory	**Ethernet UNI Access Link Trunk**	1	NO
Exclusive Advanced Internet Access IPVC	CARRIED_OVER_TRUNK	Mandatory	**Ethernet UNI Access Link Trunk**	1	NO

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.

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

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

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

Table 4 Place Relationship Role

[R20] For Basic Internet Access or Ethernet UNI Access Link Trunk products, the Role field (role) of the Related Place (RelatedPlaceRefOrValue) type **MUST** contain the INSTALL_LOCATION value shown in the Place Relationship Role column in Table 4.

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

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

14 Basic vs. Advanced Service Attributes requirements

There are several Service Attributes defined by MEF 61.1 [17] on which MEF 69.1 [19] puts additional requirements when applying to Basic or Advanced Internet Access definition. This results in some attributes differing from their original definition or missing from the Product Schema specified by this document.

These variations are presented for both Basic and Advanced versions, side by side in the tables below (all numbered requirements come from MEF 69.1 [19] and thus the document number is not mentioned each time). This is not a full list of attributes. Only those modified by MEF 69.1 [19]are listed.

Service Attribute	Basic Internet Access (BasicIaIpvc)	Advanced Internet Access (AdvancedIaIpvc)
IPVC Identifier	**Not present** There is no need for an additional Identifier. The IPVC product instance gets the product.id assigned upon creation in the Seller's system, which then can be used for inter-product references.	
IPVC Topology	**Not present** [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 **Ref type: BasicIaIpvcEndPoint**	**Single attribute instead of a list: AdvancedIaIpvc.ipvcEndPoint** **Ref type: AdvancedIaIpvcEndPoint**
IPVC Packet Delivery	**Not present.** 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 Standard Routing the only available option, so there is no need to specify it.	
IPVC DSCP Preservation	[D3] IPVC DSCP Preservation SHOULD be Disabled. **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 **Single attribute instead of a list: classOfServiceName**	
IPVC Fragmentation	**Not present.** [R8] IPVC Fragmentation MUST be Enabled. 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.	
IPVC Cloud Cloud Type	**Not present.** [R9] IPVC Cloud. Cloud Type MUST be Internet Access.	
IPVC Cloud Cloud Ingress Class of Service Map	**Not present.** [R10] Cloud Ingress Class of Service Map (F, M, D), map M MUST be empty. [R11] Cloud Ingress Class of Service Map (F, M, D), default CoS name, D, MUST NOT be Discard. 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 Cloud DNS Service	[R12] For a Basic Internet Access Service, Cloud DNS MUST NOT be None. **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 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. [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. **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. **The requirement is stated in the attribute's description.**

951 **Table 5 IPVC Service Attributes requirements**

Service Attribute	Basic Internet Access (BasicIaIpvcEndPoint)	Advanced Internet Access (AdvancedIaIpvcEndPoint)
IPVC EP EI	**Not present.** **IpUni is a composite of BasicInternetAccess there is no need to use additional references.**	**Not present.** **IpUni is either a composite of ExclusiveAdvancedInternetAccess and there is no need to use additional references or is referenced on the API level in the case of AdvancedInternetAccessIpvc product**
IPVC EP EI Type	**Not present.** Always the value of UNI	
IPVC EP Role	**Not present.** [R16] IPVC EP Role MUST be Root.	
IPVC EP ENNI Service Mapping Identifier	**Not present.** Not relevant for Internet Access	
IPVC EP Ingress Class of Service Map	**Not present.** [R17] IPVC Ingress EP Class of Service Map (F, M, D), map M MUST be empty. [R18] IPVC Ingress EP Class of Service Map (F, M, D), default CoS name, D, MUST NOT be Discard. 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 Egress Class of Service Map	**Type: DscpValue** Since there is only one class of service fo Internet Access there is no need to keep the mapping of Cos names to DSCP Value. If set, the CoS name is explicit hence only the DscpValue is sufficient to provide**.**	
IPVC EP Ingress Band-width Profile Envelope	**Ref type: BasicIaBwpEnvelope** [D5] For a Basic Internet Access Service, the IPVC EP Ingress Bandwidth Profile Envelope SHOULD be None. **The requirement is stated in the attribute's description.**	**Ref type: IpvcEpBwpEnvelope**
IPVC EP Egress Band-width Profile Envelope	**Ref type: BasicIaBwpEnvelope** [D6] For a Basic Internet Access Service, the IPVC EP Egress Bandwidth Profile Envelope SHOULD be None. **The requirement is stated in the attribute's description.**	**Ref type: IpvcEpBwpEnvelope**
IPVC EP Prefix Mapping	**Not present.** [R19] For a Basic Internet Access Service, the IPVC EP Prefix Mapping MUST be Empty.	--

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

Service Attribute	Basic Internet Access (BasicIaIpUni)	Advanced Internet Access (IpUni)
UNI Identifier	**Not present** There is no need for an additional Identifier. The IpUni product instance gets the product.id assigned upon creation in the Seller's system, which then can be used for inter-product references	
UNI List of UNI Access Links Service Attribute	**Not present.** IpUniAccessLink is a composite of BasicInternetAccess there is no need to use additional references.	**Not present.** IpUniAccessLink is either a composite of ExclusiveAdvancedInternetAccess and there is no need to use additional references or is referenced on the API level in the case of IpUni product
UNI Ingress Bandwidth Profile Envelope	**Ref type: BasicIaBwpEnvelope** [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). **The requirement is stated in the attribute's description.**	**Ref type: IpUniBwpEnvelope**
UNI Egress Bandwidth Profile Envelope	**Ref type: BasicIaBwpEnvelope** [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 61.1Table 28). **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. [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. **The requirement is stated in the attribute's description.**	
UNI Routing Protocols	**Not present.** [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 Forwarding	[D11] At a UNI with an IPVC EP for an Internet Access Service, UNI Reverse Path Forwarding SHOULD be Enabled. **The requirement is stated in the attribute's description.**

953 **Table 7 IP UNI Service Attributes requirements**

954

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. **Ref type: BasicIaUniIpv4ConnectionAddressing does not have the ipv4AddressType attribute, as if the ipv4ConnectionAddressing is set to not null the ipv4AddressType attribute 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. **IpUniAccessLink is a type that is shared among other IP Services so it does not contain Internet Access-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. **Ref type: 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. **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. **BasicIaUniIpv6ConnectionAddressing: if not null, the 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. **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. **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. **The requirement is stated in the attribute's description.**	--
UNI Access Link DHCP Relay	**Not present.** [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.** [R31] For a Basic Internet Access Service, UNI Access Link BFD MUST be None.	--
UNI Access Link Ingress Bandwidth Profile Envelope	**Ref type: BasicIaBwpEnvelope** [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. **The requirement is stated in the attribute's description.**	**Ref type: IpUniAccessLinkBwpEnvelope**
UNI Access Link Egress Bandwidth Profile Envelope	**Ref type: BasicIaBwpEnvelope** [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. **The requirement is stated in the attribute's description.**	**Ref type: IpUniAccessLinkBwpEnvelope**

UNI Access Link Reserved 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. **The requirement is stated in the attribute's description.**	--

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

956 There is no table for Ethernet Uni Access Link Trunk as it was introduced by MEF 61.1.1 [18] and
957 for the moment of this standard creation, it is not yet reflected in MEF 69.1 [19] revision so there
958 are no additional requirements to refer to.

15 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 Service Attributes of the main components (IPVC, IP UNI, IP UNI Access Link, and Ethernet UNI Access Link). Some parts of the data model representing complex technological structures are extracted to their separate subsections of section 16.

Not all MEF 61.1 [17] and MEF 61.1.1[18] Service Attributes are included in the data model. The Service Attributes that are not included are also listed in section 0. 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 the figures below some classes' attributes or further class structures are skipped for diagram readability. This is denoted by the "<<skipped>>" clause.

For readability on the diagrams the default multiplicity of relations is 1.

Some requirements define Service Attributes as mutually exclusive. This means that either one or the other must be provided, but not two (or more) of them at the same time. This is defined in the schema using the "oneOf" statement in 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 the schema that defines this requirement looks as follows:

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

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

Tables listing the attributes have the following columns:

- Name - attribute name as present in the schema file,
- Type - the data type of the attribute. All additional constraints are also listed in this column if they are defined in the schema (ex. minItems, maxItems, minimum, etc.)
    - List attributes are designated by square bracket "[]" next to the type name. E.g. "Ipv4OrIpv6Prefix[]" means the attribute is a list of objects of type "Ipv4OrIpv6Prefix"
- M/O - specifies if the attribute is mandatory or optional to provide.
- Description - description of the attribute.

It is often the case that an attribut is defined as a list with a maximum number of items equal to 1. This is a patter used to fulfil the case when given attribute has a meaningful value outside of the normal range or data type. E.g. IaIpvcCommon.maximumNumberOfIpv4Routes is an integer attribute limiting the maximum supported number of Ipv4 routes that when not provided means "Unlimited" (see Table 12). Setting an attribute explicitly to an empty list has a different meaning than not providing the value at all, in which case the default value is applied (see 9.2).

15.1 BasicInternetAccess

File: /ip/internetAccess/basicInternetAccess/basicInternetAccess.yaml

URN: urn:mef:lso:spec:cantata-sonata:basic-internet-access:v1.0.0:all

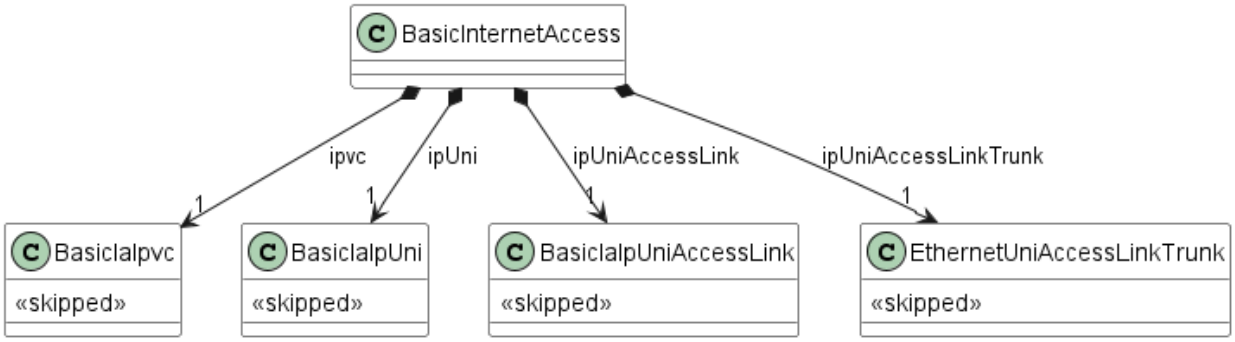


Figure 13 Basic Internet Access product

Figure 13 presents the model of BasicInternetAccess, as specified in basicInternetAccess.yaml. As described in section 17, it gathers the configuration of all product components (BasicIaIpvc, BasicIaIpUni, BasicIaIpUniAccessLink, and EthernetUniAccessLinkTrunk) in a single "top-level" product. The details of components are skipped for readability and are described in later sections (15.4.2, 15.5.3, 15.6.3, and 15.7.2).

Name	Type	M/O	Description
ipvc	BasicIaIpvc	M	Configuration of Service Attributes for Basic Internet Access IPVC
ipUni	BasicIaIpUni	M	Configuration of Service Attributes for Basic Internet Access IP UNI
ipUniAccessLink	BasicIaIpUniAccessLink	M	Configuration of Service Attributes for Basic Internet Access IP UNI Access Link
ipUniAccessLinkTrunk	EthernetUniAccessLinkTrunk	M	Configuration of Service Attributes for Basic Internet Access IP UNI Access Link Trunk

Table 9 BasicInternetAccess

15.2 AdvancedInternetAccessIpvc

File: /ip/internetAccess/advancedInternetAccessIpvc/advancedInternetAccessIpvc.yaml

URN: urn:mef:lso:spec:cantata-sonata:advanced-internet-access-ipvc:v1.0.0:all

The Advanced Internet Access IPVC is a MEF 69.1 defined version of MEF 61.1 IPVC. Reference: MEF 69.1 Section 9.1 Note that a complete Advanced Internet Access product setup requires also separate ordering of IpUni, IpUniAccessLink, EthernetUniAccessLinkTrunk (Figure 6). In case of Exclusive Advanced Internet Access, the Advanced Internet Access IPVC is part of the "top product" configuration and requires only EthernetUniAccessLinkTrunk to be ordered separately. (Figure 7). Please refer to Figure 15 to see the model diagram.

Inherits from: IaIpvcCommon

Name	Type	M/O	Description
ipvcEndPoint	AdvancedIaIpvcEndPoint	M	Advanced IPVC End Point. Reference - MEF 61.1 Section 10.3. This is narrowed to multiplicity = 1 and to AdvancedIaIpvcEndPoint type. Reference - MEF 69.1 Section 9.1 [R5]

Table 10 AdvancedInternetAccessIpvc

15.3 ExclusiveAdvancedInternetAccess

File:
/ip/internetAccess/exclusiveAdvancedInternetAccess/exclusiveAdvancedInternetAccess.yaml

URN: urn:mef:lso:spec:cantata-sonata:exclusive-advanced-internet-access:v1.0.0:all

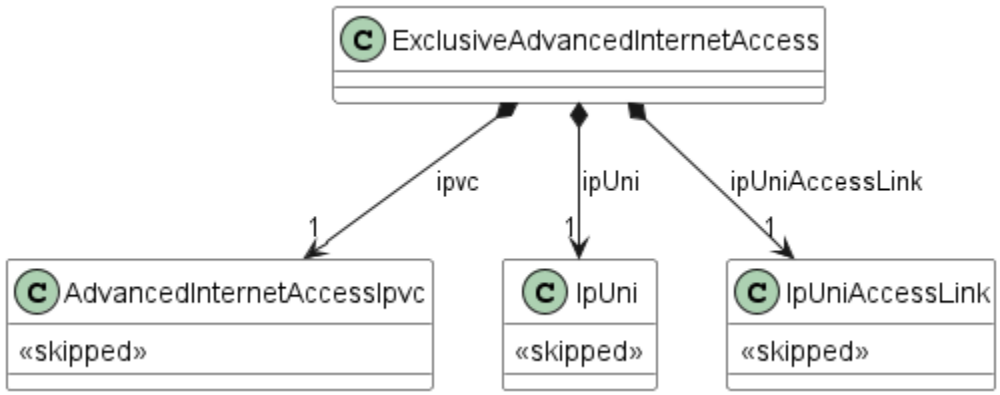


Figure 14 Exclusive Advanced Internet Access product

Figure 14 presents the model of ExclusiveAdvancedInternetAccess, as specified in exclusiveAdvancedInternetAccess.yaml. As described in section 17, for simplicity it gathers the configuration of AdvancedInternetAccessIpvc, IpUni, and IpUniAccessLink components in a single "top-level" product. The details of these components are skipped for readability and are described in their dedicated sections (15.2, 15.5, 15.6). A reference to EthernetUniAccessLinkTrunk must be provided on the product level.

Name	Type	M/O	Description
ipvc	AadvancedInternetAccessIpvc	M	Configuration of Service Attributes for Advanced Internet Access IPVC
ipUni	IpUni	M	Configuration of Service Attributes for IP UNI
ipUniAccessLink	IpUniAccessLink	M	Configuration of Service Attributes for IP UNI Access Link

Table 11 ExclusiveAdvancedInternetAccess

15.4 IPVC

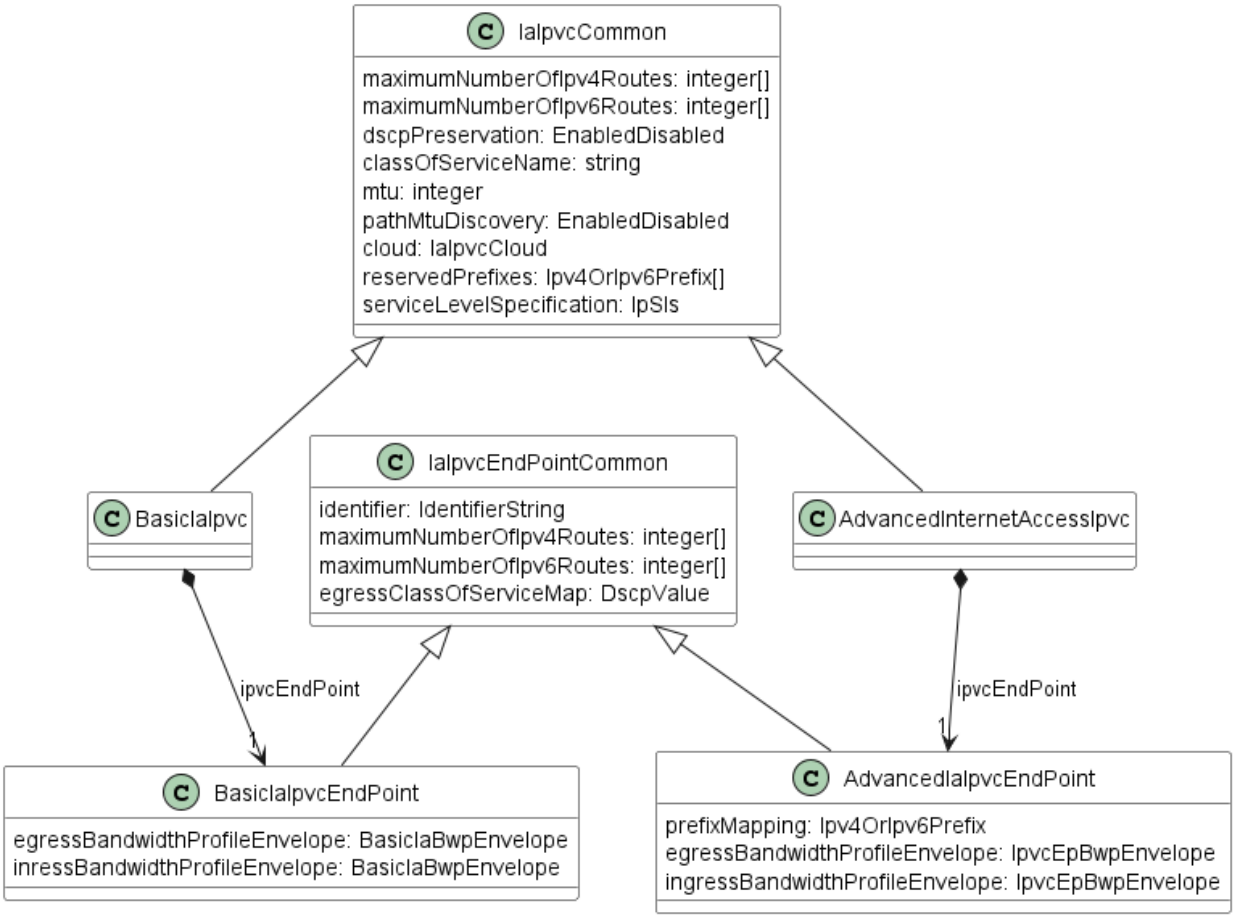


Figure 15 IPVC

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

15.4.1 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

Name	Type	M/O	Description
maximumNumberOfIpv4Routes	integer[] minimum = 0 maxItems=1	O	Maximum number of IPv4 routes supported by the service as a whole. Empty listcorresponds to a value of "Unlimited". Reference - MEF 61.1 Section 10.5
maximumNumberOfIpv6Routes	integer[] minimum = 0 maxItems=1	O	Maximum number of IPv6 routes supported by the service as a whole. Empty listcorresponds to a value of "Unlimited". Reference - MEF 61.1 Section 10.6
dscpPreservation	EnabledDisabled	O	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. MEF 69.1 [D3] For an Internet Access Service, IPVC DSCP Preservation SHOULD be Disabled.
classOfServiceName	string	O	The Class of Service Name supported by the IPVC. Reference - MEF 61.1 Section 10.8. This is "listOfClassOfServiceNames" attribute narrowed to single ref per Reference - MEF 69.1 Section 9.1 [R7]
serviceLevelSpecification	IpSls[] maxItems=1	O	The set of performance objectives for CoS Name in the IPVC. Empty list corresponds to the value of None Reference MEF 61.1 Section 10.9
mtu	integer minimum = 576	O	Indicates the maximum size (in octets) of an IP packet that can traverse the IPVC without fragmentation. Reference - MEF 61.1 Section
pathMtuDiscovery	EnabledDisabled	O	Indicates whether the Path MTU Discovery is supported for the IPVC. Reference - MEF 61.1 Section 10.11
cloud	IaIpvcCloud	O	Details of the cloud service being accessed. Reference - MEF 61.1 Section 10.13.
reservedPrefixes	Ipv4OrIpv6Prefix[]	O	Reference - MEF 61.1 Section 10.14. 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])

Table 12 IaIpvcCommon

15.4.2 BasicIaIpvc

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

The Basic 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.

Inherits from: IaIpvcCommon

Name	Type	M/O	Description
ipvcEndPoint	BasicIaIpvcEndPoint	M	Basic IPVC End Point. Reference - MEF 61.1 Section 10.3. This is narrowed to multiplicity = 1 and to BasicIaIpvcEndPoint type. Reference - MEF 69.1 Section 9.1 [R5]

Table 13 BasicIaIpvc

15.4.3 IaIpvcEndPointCommon

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

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

Name	Type	M/O	Description
identifier	IdentifierString	O	IPVC End Point identifier as described in MEF 61.1 Section 11.1.
maximumNumberOfIpv4Routes	integer[] minimum = 0 maxItems=1	O	Maximum number of IPv4 routes supported by this IPVC End Point. Reference - MEF 61.1 Section 11.7. Empty list corresponds to a value of "Unlimited".
maximumNumberOfIpv6Routes	integer[] minimum = 0 maxItems=1	O	Maximum number of IPv6 routes supported by this IPVC End Point. Reference - MEF 61.1 Section 11.8. Empty list corresponds to a value of "Unlimited".
egressClassOfServiceMap	DscpValue[] maxItems=1	O	DSCP value. Reference - MEF 61.1 Section 11.10

Table 14 IaIpvcEndPointCommon

15.4.4 BasicIaIpvcEndPoint

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

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

Inherits from: IaIpvcEndPointCommon

Name	Type	M/O	Description
egressBandwidthProfileEnvelope	BasicIaBwpEnvelope[] maxItems=1	O	Egress Bandwidth Profile Envelope for the IPVC End Point. Empty list corresponds to the value of None. Reference - MEF 61.1 Section 11.12. Reference - MEF 69.1 Section 9.2. [D6] For a Basic Internet Access Service, the egressBandwidthProfileEnvelope SHOULD be None.
ingressBandwidthProfileEnvelope	BasicIaBwpEnvelope[] maxItems=1	O	Ingress Bandwidth Profile Envelope for the IPVC End Point. Empty list corresponds to the value of None. Reference - MEF 61.1 Section 11.11. Reference - MEF 69.1 Section 9.2. [D5] For a Basic Internet Access Service, the ingressBandwidthProfileEnvelope SHOULD be None.

Table 15 BasicIaIpvcEndPoint

15.4.5 AdvancedIaIpvcEndPoint

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

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

Inherits from: IaIpvcEndPointCommon

Name	Type	M/O	Description
prefixMapping	Ipv4OrIpv6Prefix[]	O	Indicates which IP Prefixes can send and receive traffic to/from the IPVC. Reference - MEF 61.1 Section 11.5
egressBandwidthProfileEnvelope	IpvcEpBwpEnvelope[] maxItems=1	O	Egress Bandwidth Profile Envelope for the IPVC End Point. Empty list corresponds to the value of None. Reference - MEF 61.1 Section 11.12
ingressBandwidthProfileEnvelope	IpvcEpBwpEnvelope[] maxItems=1	O	Ingress Bandwidth Profile Envelope for the IPVC End Point. Empty list corresponds to the value of None. Reference - MEF 61.1 Section 11.11

Table 16 AdvancedIaIpvcEndPoint

15.4.6 IPVC Cloud

This section groups types modelling the IPVC Cloud.

15.4.6.1 IaIpvcCloud

Figure 16 presents a class diagram of IaIpvcCloud

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

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

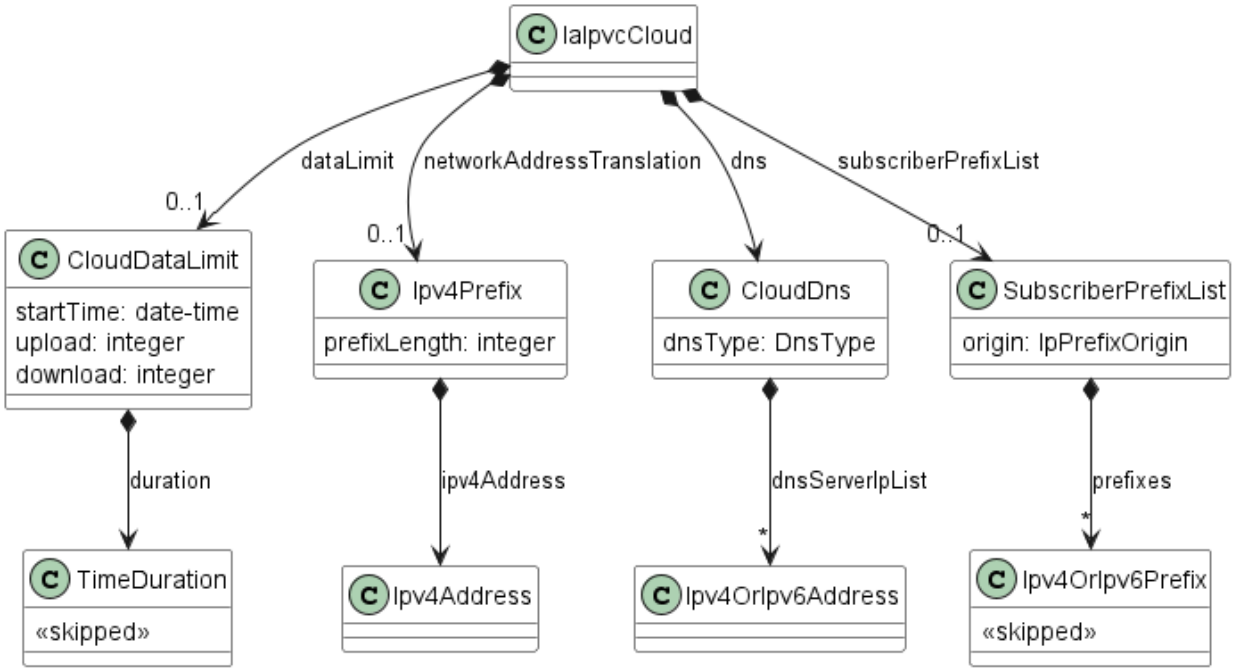


Figure 16 IaIpvcCloud

Name	Type	M/O	Description
dataLimit	CloudDataLimit[] maxItems=1	O	Limit on the amount of Data traffic sent to/received from the cloud service. Unlimited or a 4-tuple (scdl, Tcdl, ucdl, dcdl). Empty list corresponds to Unlimited. Reference - MEF 61.1 Section 10.13.3
networkAddressTranslation	Ipv4Prefix[] maxItems=1	O	Specifies whether Network Address Translation is used, and if so the IPv4 Prefix. Empty list corresponds to `Disabled`. Reference - MEF 61.1 Section 10.13.4. Reference - MEF 61.1.1 Section 9: [R55] "If the value of the Cloud Type parameter is Internet Access, when and the value of the Cloud NAT parameter is not Disabled, an IPv4 Prefix, then it MUST be a publicly assigned IPv4 Prefix.
dns	CloudDns	O	Specifies whether and how DNS is provided for the service. Reference MEF 61.1 Section 10.13.5. [R12] "For a Basic Internet Access Service, Cloud DNS MUST NOT be null.
subscriberPrefixList	SubscriberPrefixList[] maxItems=1	O	2-tuple containing the list of IP Prefixes and the origin of the IP Prefixes. Reference - MEF 61.1 Section 10.13.6. Reference - MEF 61.1.1 Section 9.

Table 17 IaIpvcCloud

15.4.6.2 CloudDataLimit

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
startTime	string format = date-time	O	Specifies a start time.
duration	TimeDuration	O	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.
upload	integer minimum = 0	O	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.
download	integer minimum = 0	O	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.

Table 18 CloudDataLimit

15.4.6.3 CloudDns

File: /ip/common/ipCommon.yaml

Data type representing a Domain Name System. Reference: MEF 61.1 Sn 10.13.5. Reference: MEF 69.1 Section 9.1:

- [R12] “For a Basic Internet Access Service, Cloud DNS MUST NOT be None”.
- [R13] “For an Internet Access Service, 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”.
- [D4] “For an Internet Access Service, 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”.

Name	Type	M/O	Description
dnsServerIpList	Ipv4OrIpv6Address[]	O	DNS server IP addresses list. If `dnsType` is STATIC this list must have at least one entry. Otherwise, it must be empty.
dnsType	DnsType	O	Domain Name System type.

Table 19 CloudDns

15.4.6.4 DnsType

File: /ip/common/ipCommon.yaml

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

Value
NONE
DHCP
PPP
STATIC
SLAAC

Table 20 DnsType

15.4.6.5 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 obtained by the Subscriber from another source.
  Reference - MEF 61.1 Section 10.13.6.
  Reference - MEF 61.1.1 Section 10.13

Name	Type	M/O	Description
prefixes	Ipv4OrIpv6Prefix[] minItems = 1	O	Non-empty list of public IP Prefixes that are used in the Subscriber Network
origin	IpPrefixOrigin	O	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.

Table 21 SubscriberPrefixList

15.4.6.6 IpPrefixOrigin

File: /ip/common/ipCommon.yaml

Enumeration of possible values of Ip Prefix Origin.

- SP: The prefix(es) have been allocated to the Subscriber by the Service Provider.
- OTHER: The prefix(es) have been allocated to the Subscriber by other source (e.g. another SP or a Regional Internet Registry).

Value
SP
OTHER

Table 22 IpPrefixOrigin

15.5 IP UNI

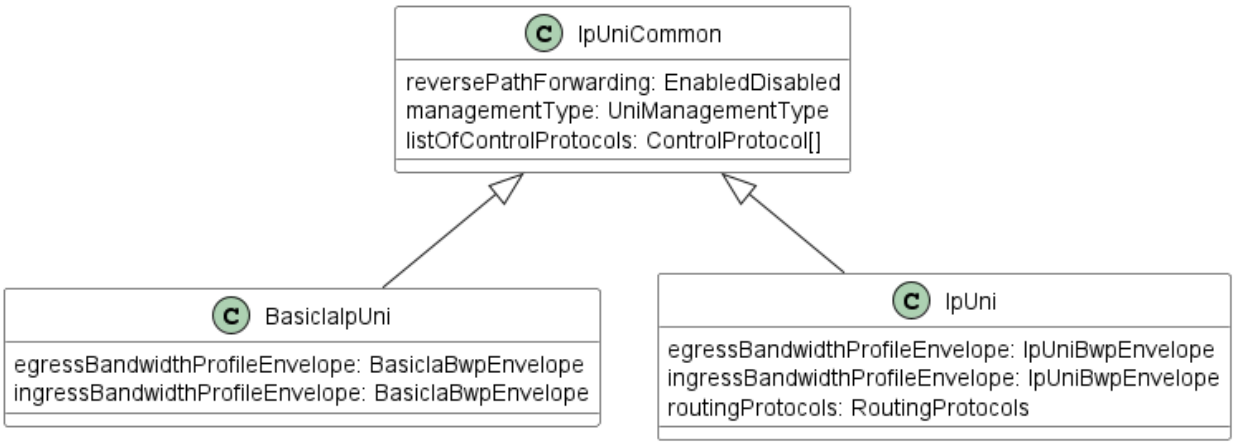


Figure 17 IP UNI

Figure 17 shows the model of the IP UNI and also the differences between the Basic and Advanced versions.

15.5.1 IpUniCommon

File: /ip/common/ipCommon.yaml

A User Network Interface (UNI) is 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. Reference - MEF 61.1 Section 12

Name	Type	M/O	Description
managementType	UniManagementType	O	Attribute indicating whether the CE is the responsibility of the Subscriber or the Service Provider. Reference - MEF 61.1 Section 12.2
listOfControlProtocols	ControlProtocol[]	O	Indication of IP Control Protocols that are not forwarded transparently by the SP. Reference - MEF 61.1 Section 12.6. [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 provided, 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 provided, the UNI List of Control Protocols SHOULD include ICMPv6 with a list of applicable SP IP addresses. Reference - MEF 69.1 Section 9.3
reversePathForwarding	EnabledDisabled	O	Indicates whether Reverse Path Forwarding checks are used by the SP at the UNI. Reference - MEF 61.1 Section 12.8. [D11] At a UNI with an IPVC EP for an Internet Access Service, reversePathForwarding SHOULD be ENABLED. Reference - MEF 69.1 Section 9.3

Table 23 IpUniCommon

15.5.2 IpUni

File: /ip/ipUni/ipUni.yaml

URN: urn:mef:lso:spec:cantata-sonata:ip-uni:v1.0.0:all

Inherits from: IpUniCommon

A User Network Interface (UNI) is 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. Reference - MEF 61.1 Section 12

Name	Type	M/O	Description
egressBandwidthProfileEnvelope	IpUniBwpEnvelope[] maxItems=1	O	Attribute used for an egress UNI Bandwidth Profile. Reference - MEF 61.1 Section 12.5. Empty list corresponds to the 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 provided, 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
ingressBandwidthProfileEnvelope	IpUniBwpEnvelope[] maxItems=1	O	Attribute used for an ingress UNI Bandwidth Profile. Reference - MEF 61.1 Section 12.4. Empty list corresponds to the 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 provided, 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
routingProtocols	RoutingProtocols[] maxItems=1	O	List of Routing Protocols used across the UNI. Reference - MEF 61.1 Section 12.7. [R21] "At a UNI with an IPVC EP for a Basic Internet Access Service, the UNI Routing Protocols list MUST be empty."

Table 24 IpUni

15.5.3 BasicIaIpUni

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

The Basic Internet Access IP UNI is a MEF 69.1 defined version of MEF 61.1 IP UNI. Reference: MEF 69.1 Section 9.3

Inherits from: IpUniCommon

Name	Type	M/O	Description
egressBandwidthProfileEnvelope	BasicIaBwpEnvelope[] maxItems=1	O	Attribute used for an egress UNI Bandwidth Profile. Reference - MEF 61.1 Section 12.5. Empty list corresponds to the 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 provided, 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
ingressBandwidthProfileEnvelope	BasicIaBwpEnvelope[] maxItems=1	O	Attribute used for an ingress UNI Bandwidth Profile. Reference - MEF 61.1 Section 12.4. Empty list corresponds to the 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 provided, 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

Table 25 BasicIaIpUni

15.5.4 ControlProtocol

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
addressing	ControlProtocolAddressing	O	Enumeration representing the addressing.
protocolName	string	O	Protocol name.
reference	string[] minItems = 1	O	Protocol reference.

Table 26 ControlProtocol

15.5.5 ControlProtocolAddressing

File: /ip/common/ipCommon.yaml

Enumeration representing the Address type for the Control Protocols data type. Reference: MEF 61.1 Section 12.6

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

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

Value
SP_OPERATOR_ADDRESSES
ANY

Table 27 ControlProtocolAddressing

15.5.6 UniManagementType

File: /ip/common/ipCommon.yaml

Enumeration representing the UNI Management Type options. Indicates 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.

- SUBSCRIBER_MANAGED - the CE is managed by the Subscriber, and the UNI Access Links correspond with the IP Attachment Circuits between the CE and the PE

- PROVIDER_MANAGED - the CE is managed (logically) by the SP, and the UNI Access Links correspond with the links from the CE to the devices within the Subscriber Network. In this latter case, the IP Attachment Circuits between the CE and the PE are internal to the SP Network.

Value
SUBSCRIBER_MANAGED
PROVIDER_MANAGED

Table 28 UniManagementType

15.6 IP UNI Access Link

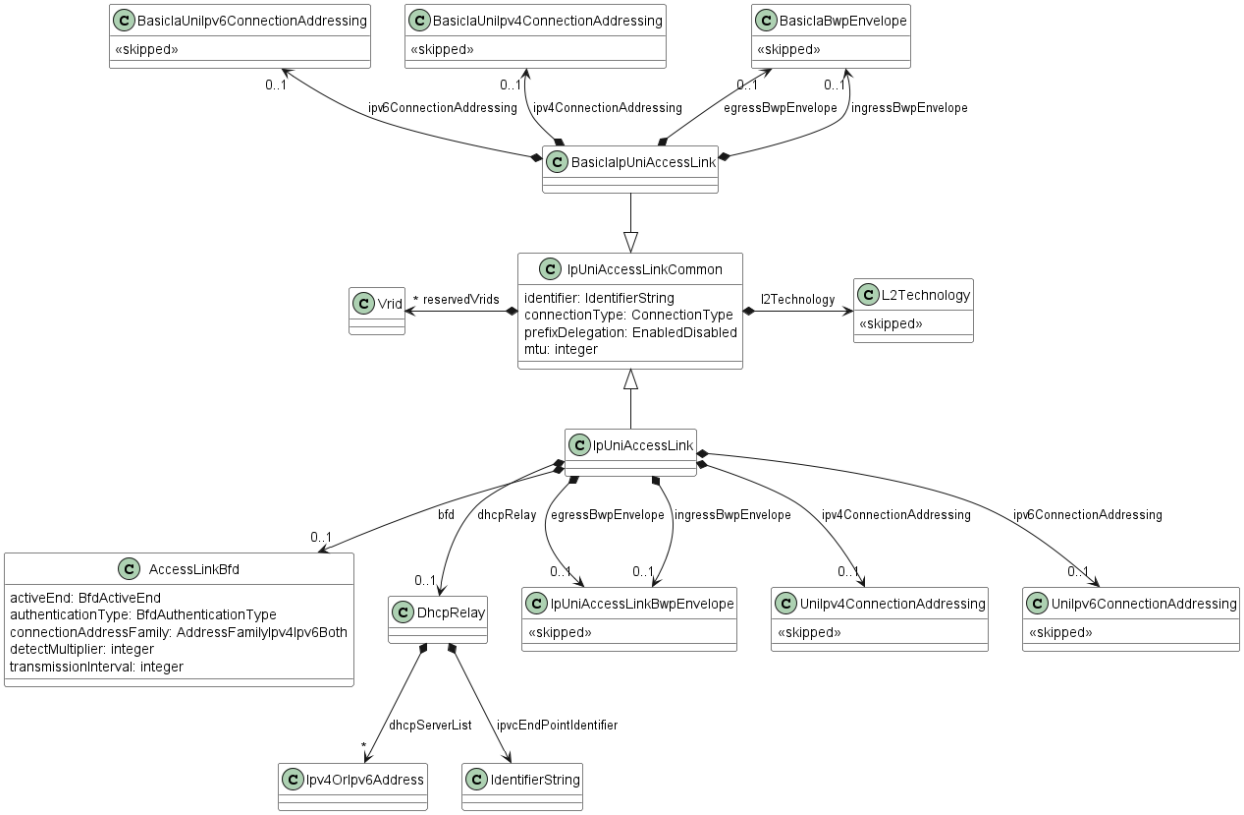


Figure 18 IP UNI Access Link

Figure 18 depicts the model of Basic and Advanced IP UNI Access Links and their differences.

15.6.1 IpUniAccessLinkCommon

File: /ip/common/ipCommon.yaml

An individual connection between the Subscriber and the SP that forms part of a UNI. Reference - MEF 61.1 Section 7.3

Name	Type	M/O	Description
identifier	IdentifierString	O	IPVC UNI Access Link identifier as described in MEF 61.1 Section 13.1. Note - it is not the same thing as the potential Product identifier if IpUniAccessLink is an instance of a Product.
connectionType	ConnectionType	O	Indicates whether the UNI Access Link is point-to-point or multipoint.
l2Technology	L2Technology	O	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	O	Indicates whether DHCP Prefix delegation is enabled. Reference - MEF 61.1 Section 13.7
mtu	integer minimum = 576	O	Maximum size, in octets of an IP Packet that can traverse the UNI Access Link. Reference - MEF 61.1 Section 13.9
reservedVrids	Vrid[]	O	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

Table 29 IpUniAccessLinkCommon

15.6.2 IpUniAccessLink

File: schema/productSchema/ip/ipUni/ipUniAccessLink.yaml

URN: urn:mef:lso:spec:cantata-sonata:ip-uni-access-link:v1.0.0:all

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

Inherits from: IpUniAccessLinkCommon

Name	Type	M/O	Description
bfd	AccessLinkBfd[] maxItems=1	O	Indication of whether BFD is used on the Uni Access Link. Reference - MEF 61.1 Section 13.8 Empty list corresponds to the value of None.
dhcpRelay	DhcpRelay[] maxItems=1	O	Indicates whether DHCP Relay functionality is enabled. Reference - MEF 61.1 Section 13.6. Empty list corresponds to a value of "Disabled".
egressBwpEnvelope	IpUniAccessLinkBwpEnvelope[] maxItems=1	O	Egress Bandwidth Profile Envelope for the UNI Access Link. Reference MEF 61.1 Section 13.11. Empty list corresponds to the value of None
ingressBwpEnvelope	IpUniAccessLinkBwpEnvelope[] maxItems=1	O	Ingress Bandwidth Profile Envelope for the UNI Access Link. Reference MEF 61.1 Section 13.10. Empty list corresponds to the value of None
ipv4ConnectionAddressing	UniIpv4ConnectionAddressing[] maxItems=1	O	IPv4 Connection Addressing. Reference - MEF 61.1 Section 13.4. Empty list corresponds to the value of None
ipv6ConnectionAddressing	UniIpv6ConnectionAddressing[] maxItems=1	O	IPv6 Connection Addressing. Reference - MEF 61.1 Section 13.5. Empty list corresponds to the value of None

Table 30 IpUniAccessLink

15.6.3 BasicIaIpUniAccessLink

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

The Basic Internet Access UNI Access Link is a MEF 69.1 defined version of MEF 61.1 UNI Access Link. Reference - MEF 69.1 Section 9.4 Internet Access UNI Access Link Requirements.

Inherits from: IpUniAccessLinkCommon

Name	Type	M/O	Description
egressBwpEnvelope	BasicIaBwpEnvelope[] maxItems=1	O	Egress Bandwidth Profile Envelope for the UNI Access Link. Reference - MEF 61.1 Section 13.11. Empty list corresponds to the value of None
ingressBwpEnvelope	BasicIaBwpEnvelope[] maxItems=1	O	Ingress Bandwidth Profile Envelope for the UNI Access Link. Reference - MEF 61.1 Section 13.10. Empty list corresponds to the value of None
ipv4ConnectionAddressing	BasicIaUniIpv4ConnectionAddressing[] maxItems=1	O	IPv4 Connection Addressing. Reference - MEF 61.1 Section 13.4. Empty list corresponds to the value of None
ipv6ConnectionAddressing	BasicIaUniIpv6ConnectionAddressing[] maxItems=1	O	IPv6 Connection Addressing. Reference - MEF 61.1 Section 13.5. Empty list corresponds to the value of None

Table 31 BasicIaIpUniAccessLink

15.6.4 UNI Access Link BFD

This section groups types modelling the UNI Access Link Bidirectional Forwarding Detection (BFD)

15.6.4.1 AccessLinkBfd

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
connectionAddressFamily	AddressFamilyIpv4Ipv6Both	O	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 and 16.5
transmissionInterval	integer minimum = 0	O	Transmission Interval Reference - MEF 61.1 Section 13.8 and 16.5
detectMultiplier	integer minimum = 0	O	BFD Detect multiple as an Integer. Reference - MEF 61.1 Section 13.8 and 16.5 Attribute.
activeEnd	BfdActiveEnd	O	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
authenticationType	BfdAuthenticationType	O	BFD Authentication as described in RFC 5880. Reference - MEF 61.1 Section 13.8 and 16.5

Table 32 AccessLinkBfd

15.6.4.2 AddressFamilyIpv4Ipv6Both

File: /ip/common/ipCommon.yaml

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

Value
IPV4
IPV6
BOTH

Table 33 AddressFamilyIpv4Ipv6Both

15.6.4.3 BfdActiveEnd

File: /ip/common/ipCommon.yaml

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

- SUBSCRIBER: Subscriber takes active BFD role.
- SP: Service Provider takes active BFD role.
- BOTH: Subscriber and Service Provider take active BFD role.

Value
SUBSCRIBER
SP
BOTH

Table 34 BfdActiveEnd

15.6.4.4 BfdAuthenticationType

File: /ip/common/ipCommon.yaml

Enumeration of possible BFD Authentication Type, as specified by RFC 5880 [9]. In case other than "NONE" is specified additional specific parameters need to be agreed between the Buyer and the Seller.

- 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 RFC 5880 [9] 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: RFC 5880 [9] 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: RFC 5880 [9] 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: RFC 5880 [9] 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: RFC 5880 [9] 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

Table 35 BfdAuthenticationType

15.6.5 ConnectionType

File: /ip/common/ipCommon.yaml

An enumeration representing the connection type.

- POINT_TO_POINT indicates that the link is logically point to Point.
- MULTIPOINT indicates the link is logically multipoint.

Value
POINT_TO_POINT
MULTIPOINT

Table 36 ConnectionType

15.6.6 DhcpRelay

File: /ip/common/ipCommon.yaml

Dynamic Host Configuration Protocol (DHCP) Relay functionality is useful when the Subscriber uses DHCP (per RFC 2131 and RFC 8415) in the Subscriber Network but does not want to place a DHCP server (or possibly a pair of redundant DHCP servers) in each part of the network. Reference - MEF 61.1 Section 13.6

Name	Type	M/O	Description
dhcpServerList	Ipv4OrIpv6Address[] minItems = 1	O	Non-empty list of IP addresses for DHCP Servers belonging to the Subscriber. Reference - MEF 61.1 Section 13.6
ipvcEndPointIdentifier	IdentifierString	O	IPVC End Point identifier as described in MEF 61.1 Section 11.1. In case of Exlusive Advanced Internet Access it points to the “identifier” of the IPVC End Point that is part of the product configuration. In case of Advanced Internet Access it points to the “identifier” attribute of the related IPVC End Point of the Advanced Internet Access product or an IPVC End Point for a different product at the same IP UNI.

Table 37 DhcpRelay

15.6.7 Vrid

File: /ip/common/ipCommon.yaml

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

15.6.8 Connection Addressing

This section groups types modelling the UNI Access Link Connection Addressing.

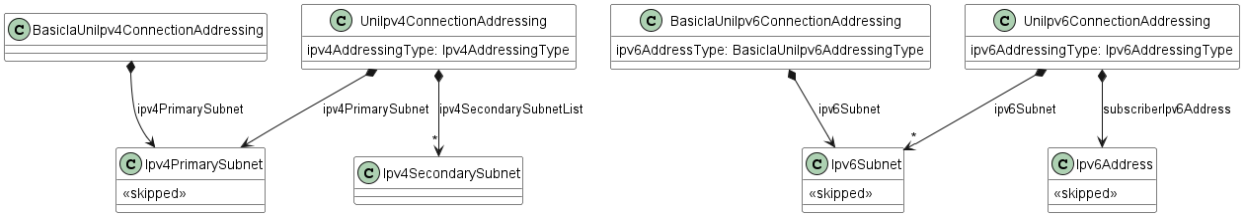


Figure 19 IPV4 and IPV6 Connection Addressing

Figure 19 shows both IPv4 and IPv6 versions of Connection Addressing.

15.6.8.1 BasicIaUnIpv4ConnectionAddressing

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

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

- [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." Reference - MEF 69.1 Section 9.4

- [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 Section 9.4

Name	Type	M/O	Description
ipv4PrimarySubnet	Ipv4PrimarySubnet	O	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 Section 9.4

Table 38 BasicIaUniIpv4ConnectionAddressing

15.6.8.2 UnIpv4ConnectionAddressing

File: /ip/common/ipCommon.yaml

UniIpv4ConnectionAddressing is a data type representing how IPv4 addresses are allocated to the devices on the UNI Access Link. Reference - MEF 61 Section 13.4.

- [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 null." Reference - MEF 69.1 Section 9.4

Name	Type	M/O	Description
ipv4AddressingType	Ipv4AddressingType	O	IPv4 Connection Addressing.
ipv4PrimarySubnet	Ipv4PrimarySubnet	O	Primary IPv4 Subnet. Includes IPv4 Prefix and Service Provider IPv4 Addresses.
ipv4SecondarySubnetList	Ipv4SecondarySubnet[]	O	Secondary IPv4 Subnet List. Includes IPv4 Prefix and Service Provider IPv4 Addresses.

Table 39 UniIpv4ConnectionAddressing

15.6.8.3 BasicIaUniIpv6ConnectionAddressing

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

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

- [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 null." Reference - MEF 69.1 Section 9.4.

- [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." Reference - MEF 69.1 Section 9.4.

Name	Type	M/O	Description
ipv6AddressType	BasicIaUniIpv6AddressingType	O	Basic Internet Access IPv6 Connection Address mechanism
ipv6Subnet	Ipv6Subnet	O	Ipv6 Subnet [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." Reference - MEF 69.1 Section 9.4.

Table 40 BasicIaUniIpv6ConnectionAddressing

15.6.8.4 UniIpv6ConnectionAddressing

File: /ip/common/ipCommon.yaml

UniIpv6ConnectionAddressing is a data type representing how IPv6 addresses are allocated to the devices on the UNI Access Link. Reference - MEF 61 Section 13.5.

- [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 null. Reference - MEF 69.1 Section 9.4

Name	Type	M/O	Description
ipv6AddressingType	Ipv6AddressingType	O	IPv6 Connection Addressing.
subscriberIpv6Address	Ipv6Address	O	Subscriber IPv6 address.
ipv6Subnet	Ipv6Subnet[]	O	Ipv6 Subnet

Table 41 UniIpv6ConnectionAddressing

15.6.8.5 Ipv4AddressingType

File: /ip/common/ipCommon.yaml

Enumeration representing IPv4 Address Types specific for UNI Access Links.

- DHCP: Dynamic Host Configuration Protocol (DHCP) is used the Subscriber devices to request IPv4 addresses in a given subnet from the SP or Operator.
- STATIC: IPv4 addresses in a given IPv4 subnet are statically assigned to the SP or Operator and to the Subscriber.
- UNNUMBERED: The SP or Operator and the Subscriber each assigned an IPv4 address (from their own address pools) independently. These addresses can be on different subnets, and so an interface-based routing protocol is needed to ensure reachability.

Value
DHCP
STATIC
UNNUMBERED

Table 42 Ipv4AddressingType

15.6.8.6 BasicIaUnIpv6AddressingType

File: /ip/internetAccess/internetAccessCommon/internetAccessCommon.yaml

Enumeration representing IPv6 Address Types specific for UNI Access Links.

- DHCP: Dynamic Host Configuration Protocol (DHCP) is used by the Subscriber devices to request IPv6 addresses in a given subnet from the SP or Operator.
- SLAAC: Stateless Address Autoconfiguration (SLAAC) is used by the Subscriber devices to create unique IPv6 global addresses within an IP Prefix advertised by the SP or Operator as describer in RFC 4862.

Value
DHCP
SLAAC

Table 43 BasicIaUniIpv6AddressingType

15.6.8.7 Ipv6AddressingType

File: /ip/common/ipCommon.yaml

Ipv6AddressingType

Enumeration representing IPv6 Address Types specific for UNI Access Links.

- DHCP: Dynamic Host Configuration Protocol (DHCP) is used by the Subscriber devices to request IPv6 addresses in a given subnet from the SP or Operator.
- SLAAC: Stateless Address Autoconfiguration (SLAAC) is used by the Subscriber devices to create unique IPv6 global addresses within an IP Prefix advertised by the SP or Operator as describer in RFC 4862.
- STATIC: IPv6 addresses in a given IPv6 subnet are statically assigned to the SP or Operator and to the Subscriber.
- LL_ONLY: If the value is LL-only, these are only IPv6 addresses used onthe UNI Access Link.

Value
DHCP
SLAAC
STATIC
LL_ONLY

Table 44 Ipv6AddressingType

15.6.9 L2Technology

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
demux	VlanId[]	O	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

Table 45 L2Technology

15.6.10 VlanId

File: /ip/common/ipCommon.yaml

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

15.7 Ethernet UNI Access Link Trunk

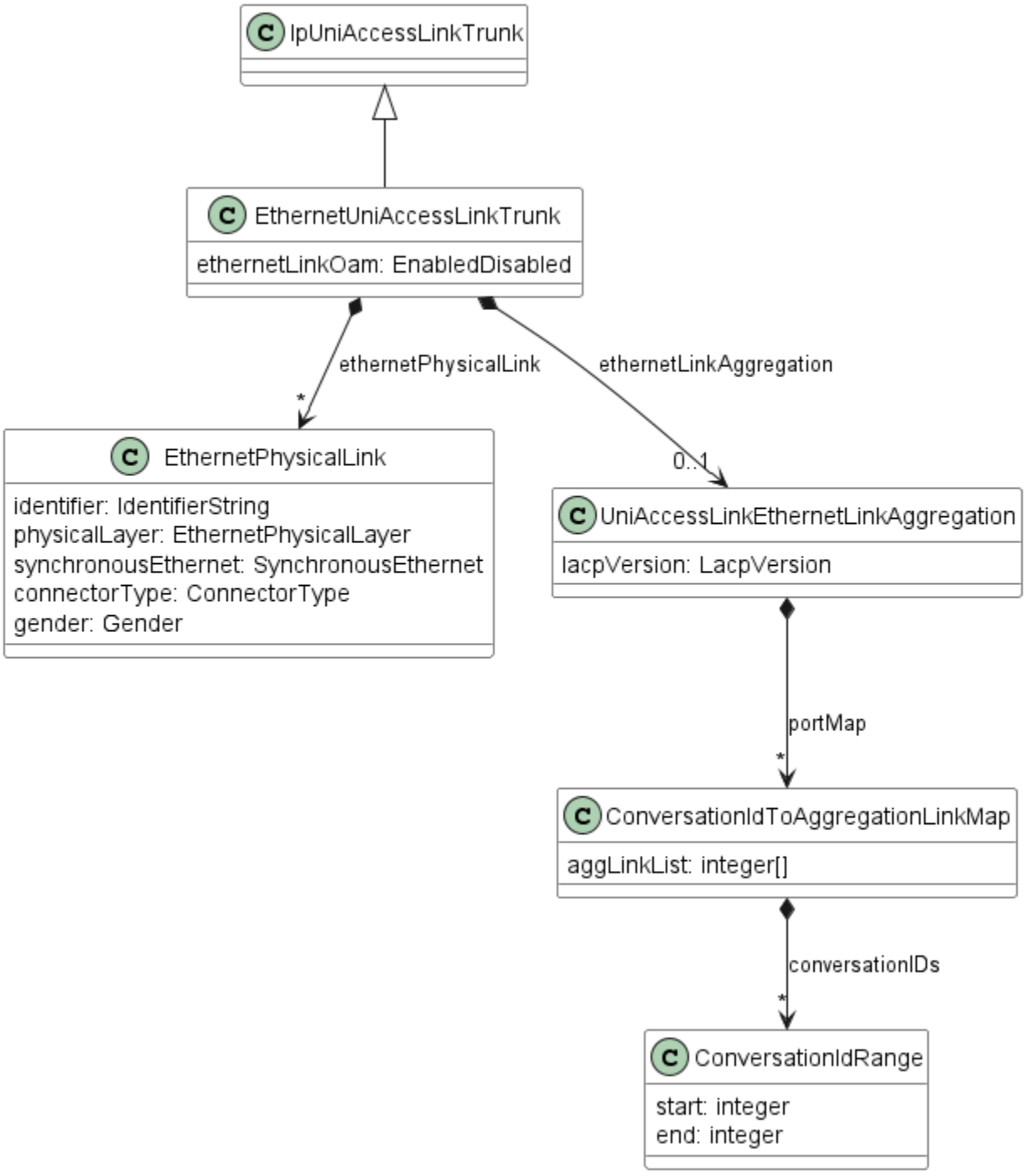


Figure 20 EthernetUniAccessLinkTrunk

Figure 20 Shows the diagram of the Ethernet UNI Access Link Trunk. It is the only specified subclass of an abstract class IP UNI Access Link Trunk. It is used by all 3 Internet Access Product flavors without any changes.

15.7.1 IpUniAccessLinkTrunk

File: /ip/ipUni/ipUniCommon.yaml

A UNI Access Link Trunk is a construct that encapsulates the details of Layer 1 and Layer 2 configuration shared by one or more UNI Access Links. Reference: MEF 61.1.1 Section A1-1. It has no attributes.

15.7.2 EthernetUniAccessLinkTrunk

File: /ip/ipUni/ethernetUniAccessLinkTrunk.yaml

URN: urn:mef:lso:spec:cantata-sonata:ethernet-uni-access-link-trunk:v1.0.0:all

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

Inherits from: IpUniAccessLinkTrunk

Name	Type	M/O	Description
ethernetPhysicalLink	EthernetPhysicalLink[] minItems = 1	O	A list of the physical link types along with some additional capabilities
ethernetLinkAggregation	UniAccessLinkEthernetLinkAggregation[] maxItems=1	O	Configuration of Link Aggregation for the UNI Access Link Trunk. Empty list corresponds to the value of None.
ethernetLinkOam	EnabledDisabled	O	Indicates whether Link OAM is used on the UNI Access Link Trunk

Table 46 EthernetUniAccessLinkTrunk

15.7.3 EthernetPhysicalLink

File: /ip/common/ipCommon.yaml

Data type representing UNI Access Link Trunk List of Ethernet Physical Links as defined in MEF 61.1.1 Section A1-1.3.1.

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

Table 47 EthernetPhysicalLink

15.7.4 ConnectorType

File: /ip/common/ipCommon.yaml

Enumeration representing type of connector presented to Subscriber.

- RJ45 - Copper. Standard: IEC 60603-7 [1], TIA568 [29]
- SC - Fiber. Standard: IEC 61754-4 [2]
- LC - Fiber. Standard: IEC 61754-20 [3]
- OTHER - any other connector type

Value
RJ45
SC
LC
OTHER

Table 48 ConnectorType

15.7.5 EthernetPhysicalLayer

File: /ip/common/ipCommon.yaml

Enumeration representing the different Ethernet physical layers. Reference: MEF 61.1.1 Table A1

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

Table 49 EthernetPhysicalLayer

15.7.6 Gender

File: /ip/common/ipCommon.yaml

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

- SOCKET - Socket
- PLUG - Plug

Value
SOCKET
PLUG

Table 50 Gender

15.7.7 SynchronousEthernet

File: /ip/common/ipCommon.yaml

Enumeration indicating if the physical link supports Synchronous Ethernet.

DISBALED - Synchronous Ethernet is disabled on the corresponding physical link.

ESMC - Synchronous Ethernet as defined in ITU-T G.8262/Y.1362 [12] is used on the corresponding physical link with synchronization provided by the Service Provider to the Subscriber. SSM for Synchronous Ethernet using the Ethernet Synchronous Messaging Channel (ESMC) protocol as defined in ITU-T G.8264/Y.1364 [13] is used on the corresponding physical link.

NO ESMC - Synchronous Ethernet as defined in ITU-T G.8262/Y.1362 [12] is used on the corresponding physical link with synchronization provided by the Service Provider to the Subscriber. SSM for Synchronous Ethernet using the Ethernet Synchronous Messaging Channel (ESMC) protocol as defined in ITU-T G.8264/Y.1364 [13] is not used on the corresponding physical link.

Value
DISABLED
ESMC
NO_ESMC

Table 51 SynchronousEthernet

15.7.8 UniAccessLinkEthernetLinkAggregation

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
lacpVersion	LacpVersion	O	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[]	O	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.

Table 52 UniAccessLinkEthernetLinkAggregation

15.7.9 LacpVersion

File: /ip/common/ipCommon.yaml

Indicates which version of the Link Aggregation Control Protocol, LACP, is used. (See clause 6.4 in IEEE Std 802.1AX-2020 [A1-4].). The possible values are LACPv1, LACPv2, or Static. If the value is Static, LACP is not used.

Value
LACPV1
LACPV2
STATIC

Table 53 LacpVersion

15.7.10 ConversationIdToAggregationLinkMap

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
conversationIDs	ConversationIdRange[] minItems = 1 uniqueItems = true	O	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[] minimum = 1 minItems = 1 uniqueItems = true	O	802.1AX-2014 sec. 6.6.2.1 - An ordered list of Aggregation Link Numbers

Table 54 ConversationIdToAggregationLinkMap

15.7.11 ConversationIdRange

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
start	integer minimum = 0 maximum = 4094	O	The starting Conversation ID of the range or the only Conversation ID if there is no end value
end	integer minimum = 0 maximum = 4094	O	The final Conversation ID in the range

Table 55 ConversationIdRange

16 Ancillary Constructs Service Attributes

This section presents the complex data model structures and sets of data types used in the modelling of Internet Access Service Attributes. They are put in their separate subsections to provide more readability.

16.1 IP SLS

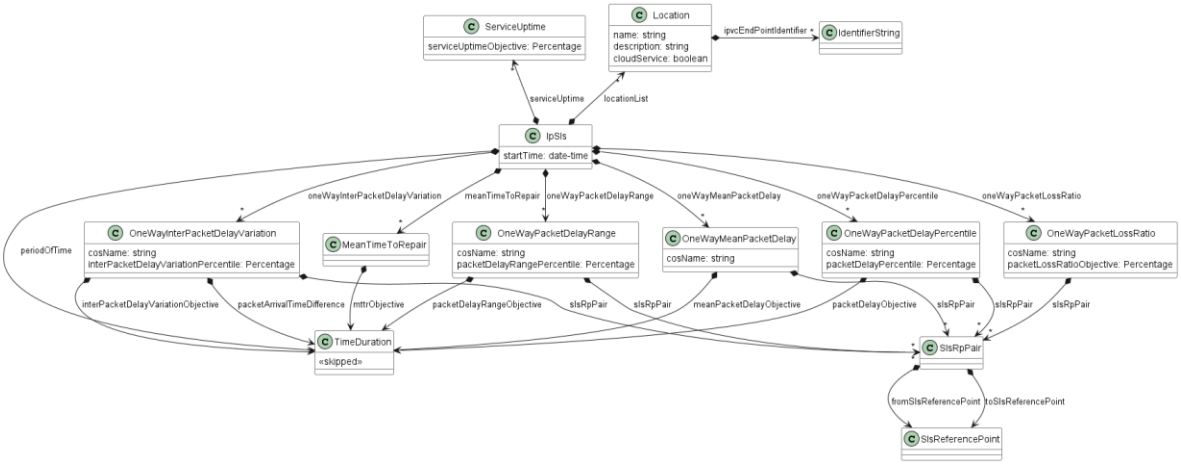


Figure 21 IpSls

Figure 21 shows the model of the IP SLS with all available metrics.

16.1.1 IpSls

File: /ip/common/ipSls.yaml

The IPVC Service Level Specification (SLS) describes the performance objectives for the performance of conformant IP Data Packets that flow over the IPVC. The IPVC Service Level Specification Service Attribute is either empty, or a set of three attributes (`startTime`, `periodOfTime`, `locationList`) followed by attributes per every applicable performance metric, providing metric's specific attributes. Reference - MEF 61.1 Section 10.9

Name	Type	M/O	Description
startTime	string format = date-time	O	Start time of IP SLS.
periodOfTime	TimeDuration	O	Period of time over which IP SLS is measured.
locationList	Location[]	O	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.
oneWayPacketDelayPercentile	OneWayPacketDelayPercentile[]	O	List of SLS Entries for the One-way Packet Delay Percentile metric.
oneWayMeanPacketDelay	OneWayMeanPacketDelay[]	O	List of SLS Entries for the One-way Mean Packet Delay metric.
oneWayInterPacketDelayVariation	OneWayInterPacketDelayVariation[]	O	List of SLS Entries for the One-way Inter-Packet Delay Variation metric.
oneWayPacketDelayRange	OneWayPacketDelayRange[]	O	List of SLS Entries for the One-way Packet Delay Range metric.
oneWayPacketLossRatio	OneWayPacketLossRatio[]	O	List of SLS Entries for the One-way Packet Loss Ratio metric.
serviceUptime	ServiceUptime[]	O	List of SLS Entries for the Service Uptime metric
meanTimeToRepair	MeanTimeToRepair[]	O	List of SLS entries for the Mean Time to Repair metric.

Table 56 IpSls

16.1.2 OneWayPacketDelayPercentile

File: /ip/common/ipSls.yaml

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

Name	Type	M/O	Description
cosName	string	O	One of the values in the IPVC List of Class of Service Names Service Attribute. Reference - MEF 61.1 Section 10.9.4
slsRpPair	SlsRpPair[] minItems = 1	O	Set of ordered SLS-RP pairs. Reference - MEF 61.1 Section 10.9.4
packetDelayPercentile	Percentage	O	Packet Delay Percentile. Reference - MEF 61.1 Section 10.9.4
packetDelayObjective	TimeDuration	O	Packet Delay Objective. Reference - MEF 61.1 Section 10.9.4

Table 57 OneWayPacketDelayPercentile

16.1.3 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

Name	Type	M/O	Description
cosName	string	O	One of the values in the IPVC List of Class of Service Names Service Attribute. Reference - MEF 61.1 Section 10.9.5
slsRpPair	SlsRpPair[] minItems = 1	O	Set of ordered SLS-RP pairs. Reference - MEF 61.1 Section 10.9.5
meanPacketDelayObjective	TimeDuration	O	Mean Packet Delay Objective. Reference - MEF 61.1 Section 10.9.5, Table-5.

Table 58 OneWayMeanPacketDelay

16.1.4 OneWayInterPacketDelayVariation

File: /ip/common/ipSls.yaml

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

Name	Type	M/O	Description
cosName	string	O	One of the values in the IPVC List of Class of Service Names Service Attribute. Reference - MEF 61.1 Section 10.9.6
slsRpPair	SlsRpPair[] minItems = 1	O	Set of ordered SLS-RP pairs. Reference - MEF 61.1 Section 10.9.6
packetArrivalTimeDifference	TimeDuration	O	Difference in the time of arrival of packets. Reference - MEF 61.1 Section 10.9.6
interPacketDelayVariationPercentile	Percentage	O	Inter-Packet Delay Variation Percentile. Reference - MEF 61.1 Section 10.9.6
interPacketDelayVariationObjective	TimeDuration	O	Inter-Packet Delay Variation Objective. Reference - MEF 61.1 Section 10.9.6

Table 59 OneWayInterPacketDelayVariation

16.1.5 OneWayPacketDelayRange

File: /ip/common/ipSls.yaml

The One-way Packet Delay Range Performance Metric is the maximum, over all the ordered pairs of SLS-RPs in a given set S, of the difference between the rth percentile of one-way packet delay and the minimum 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 10.9.7

Name	Type	M/O	Description
cosName	string	O	One of the values in the IPVC List of Class of Service Names Service Attribute. Reference - MEF 61.1 Section 10.9.7
slsRpPair	SlsRpPair[] minItems = 1	O	Set of ordered SLS-RP pairs. Reference - MEF 61.1 Section 10.9.7
packetDelayRangePercentile	Percentage	O	Packet Delay Range Percentile. Reference - MEF 61.1 Section 10.9.7
packetDelayRangeObjective	TimeDuration	O	Packet Delay Range Objective. Reference - MEF 61.1 Section 10.9.7

Table 60 OneWayPacketDelayRange

16.1.6 OneWayPacketLossRatio

File: /ip/common/ipSls.yaml

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

Name	Type	M/O	Description
cosName	string	O	One of the values in the IPVC List of Class of Service Names Service Attribute. Reference - MEF 61.1 Section 10.9.8
slsRpPair	SlsRpPair[] minItems = 1	O	Set of ordered SLS-RP pairs. Reference - MEF 61.1 Section 10.9.5
packetLossRatioObjective	Percentage	O	Packet Loss Ratio Objective. Reference - MEF 61.1 Section 10.9.8

Table 61 OneWayPacketLossRatio

16.1.7 ServiceUptime

File: /ip/common/ipSls.yaml

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

Name	Type	M/O	Description
serviceUptimeObjective	Percentage	O	Service Uptime Objective. Reference - MEF 61.1 Section 10.9.9

Table 62 ServiceUptime

16.1.8 Percentage

File: /ip/common/ipSls.yaml

This is a number of percent - a number (not necessarily an integer) between 0 and 100.

16.1.9 Location

File: /ip/common/ipSls.yaml

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.

Name	Type	M/O	Description
name	string	O	Location name
description	string	O	Location description
ipvcEndPointIdentifier	IdentifierString[]	O	A list of IPVC End Point identifier as described in MEF 61.1 Section 11.1.
cloudService	boolean	O	Attribute to indicate if associated with a cloud service.

Table 63 Location

16.1.10 MeanTimeToRepair

File: /ip/common/ipSls.yaml

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. Reference - MEF 61.1.1. Section 10.9.10

Name	Type	M/O	Description
mttrObjective	TimeDuration	O	Mean Time To Repair Objective

Table 64 MeanTimeToRepair

16.1.11 SlsReferencePoint

File: /ip/common/ipSls.yaml

A reference SlsReferencePoint which is either a Location.name or IpvcEndPoint.identifier. Reference - MEF 61.1 Section 10.9.1.

Name	Type	M/O	Description
referencedType	SlsReferencePointType	O	The type of referenced SlsReferencePoint. Either a Location or IpvcEndPoint.
identifier	string	O	When referencedType is IPVC_END_POINT then the identifier matches the IpvcEndPoint.identifier. When referencedType is LOCATION then the identifier matches the Location.name

Table 65 SlsReferencePoint

16.1.12 SlsReferencePointType

File: /ip/common/ipSls.yaml

Enumeration representing the possible SlsReferencePoint types.

- IPVC_END_POINT - The SlsReferencePoint.identifier points to IpvcEndPoint

- LOCATION - The SlsReferencePoint.identifier points to Location

Value
IPVC_END_POINT
LOCATION

Table 66 SlsReferencePointType

16.1.13 SlsRpPair

File: /ip/common/ipSls.yaml

Service Level Specification Reference Point Pair. In an IPVC, performance objectives are specified as applying between pairs of SLS Reference Points, each of which can be an IPVC End Point or a Location. The SlsRpPair is a representation of this association. Reference MEF 61.1 Section 10.9.1

Name	Type	M/O	Description
fromSlsReferencePoint	SlsReferencePoint	O	Pointer to the "from" SLS Reference Point.
toSlsReferencePoint	SlsReferencePoint	O	Pointer to the "to" SLS Reference Point.

Table 67 SlsRpPair

16.2 Routing Protocols

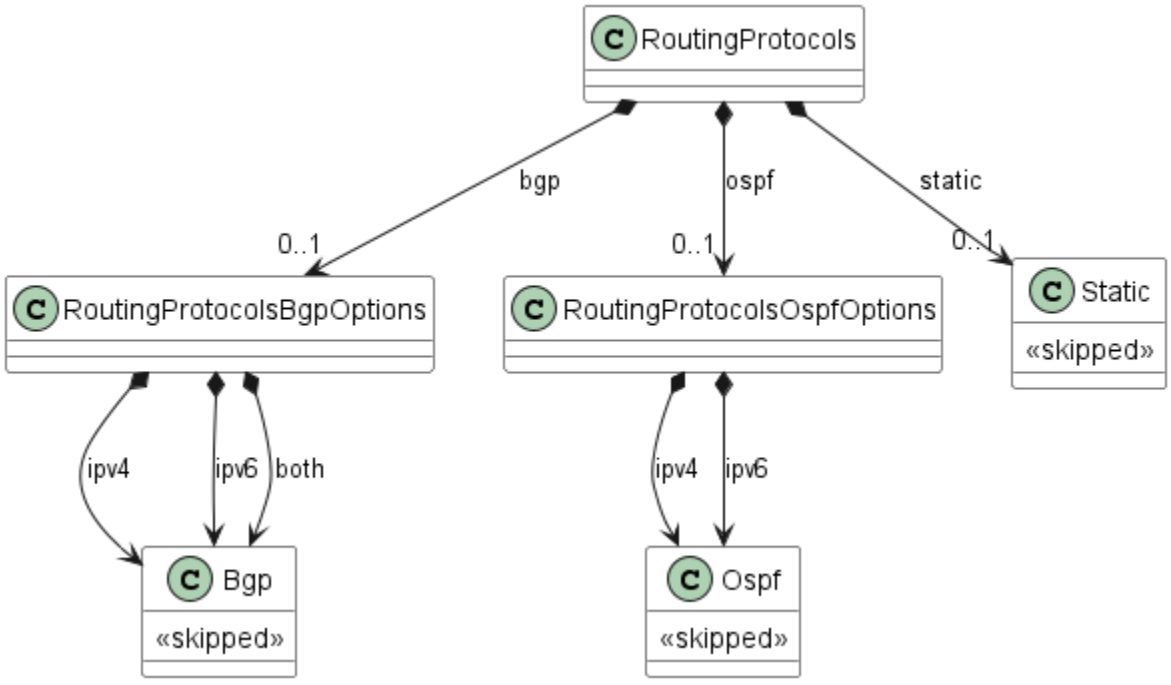


Figure 22 Routing Protocols

The UNI Routing Protocols Service Attribute specifies the routing protocols and associated parameters that are used to exchange IP routes across the UNI. The value is a list of protocols (possibly empty), where each entry consists of the protocol name (one of Static, OSPF or BGP), the type of routes that will be exchanged (one of IPv4, IPv6 or Both), and a set of additional

parameters as specified in the subsections below. According to [R109] The value of the UNI Routing Protocols Service Attribute MUST NOT contain more than one entry for the same protocol name, except when there are exactly two entries with a given protocol name, one with route type IPv4 and one with route type IPv6.

16.2.1 RoutingProtocols

File: /ip/common/ipCommon.yaml

Data type to support routing protocols and associated parameters that are used to exchange IP routes across the UNI. It has three attributes allowing for providing configuration of BGP, OSPF and Static routing protocols. Reference - MEF 61.1 Section 12.7

Name	Type	M/O	Description
bgp	RoutingProtocolsBgpOptions[] maxItems=1	O	BGP routing protocol configuration options.
ospf	RoutingProtocolsOspfOptions[] maxItems=1	O	OSPF routing protocol configuration options.
static	Static[] maxItems=1	O	Static routing configuration options.

Table 68 RoutingProtocols

16.2.2 RoutingProtocolsBgpOptions

File: /ip/common/ipCommon.yaml

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

- ipv4, or
- ipv6, or
- both (one BGP session exchanging both IPv4 and IPv6) , or
- ipv4 and ipv6 (separate BGP session for exchanging IPv4 and IPv6)

Name	Type	M/O	Description
ipv4	Bgp	O	Configuration for exchanging IPv4 types of routes.
ipv6	Bgp	O	Configuration for exchanging IPv6 types of routes.
both	Bgp	O	Common configuration for exchanging both IPv4 and IPv6 types of routes.

Table 69 RoutingProtocolsBgpOptions

16.2.3 RoutingProtocolsOspfOptions

File: /ip/common/ipCommon.yaml

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

- ipv4, or
- ipv6, or

- ipv4 and ipv6

Name	Type	M/O	Description
ipv4	Ospf	O	Configuration for exchanging IPv4 types of routes.
ipv6	Ospf	O	Configuration for exchanging IPv6 types of routes.

Table 70 RoutingProtocolsOspfOptions

16.2.4 BGP

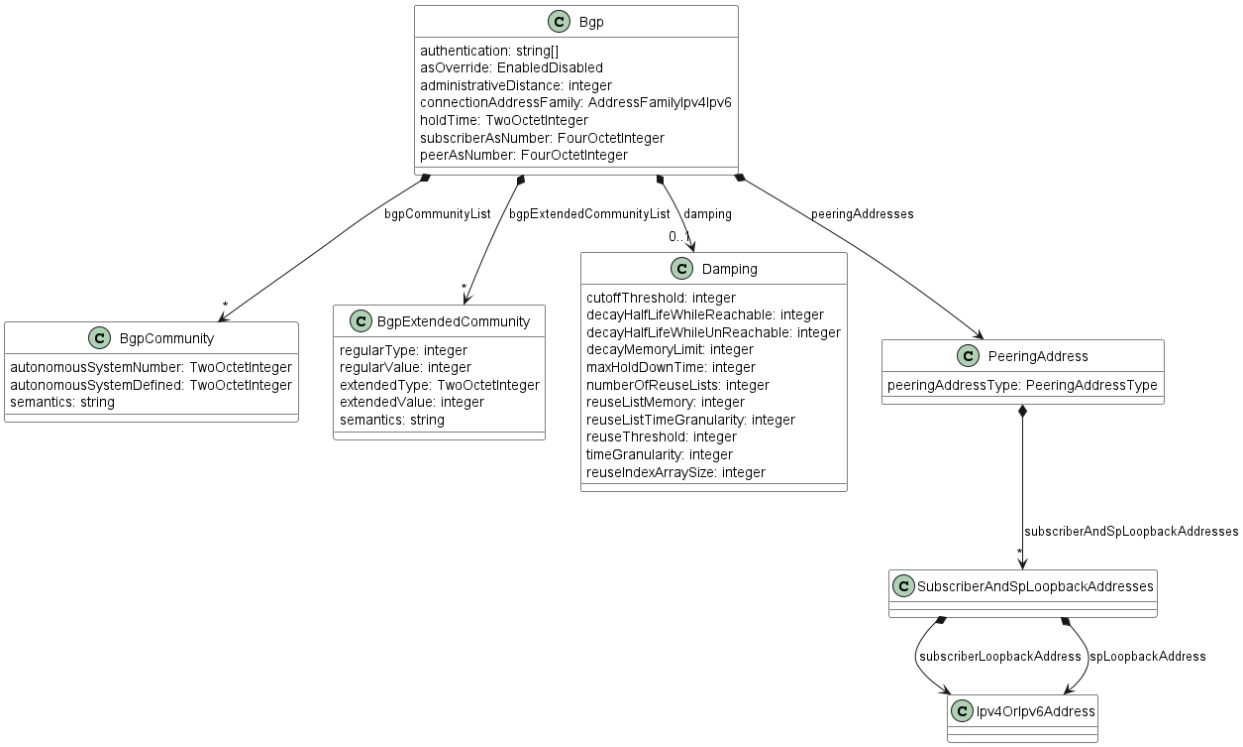


Figure 23 Bgp

Figure 23 depicts the model of BGP routing protocol configuration model.

16.2.4.1 Bgp

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
subscriberAsNumber	FourOctetInteger	O	BGP Subscriber Autonomous System number.
peerAsNumber	FourOctetInteger	O	BGP Peer Autonomous System Number.
connectionAddressFamily	AddressFamilyIpv4Ipv6	O	Connection Address Family (IPv4 or IPv6).
peeringAddresses	PeeringAddress	O	Peering Addresses.
authentication	string[] maxItems=1	O	BGP Authentication. It is either empty or if present is it a value of MD5 Password. It is assumed that an encrypted channel is used when this data is passed across the API so that the password is protected.
bgpCommunityList	BgpCommunity[]	O	Used to control which routers are accepted, preferred, distributed, or advertised.
bgpExtendedCommunityList	BgpExtendedCommunity[]	O	Mechanism for labeling information carried in BGP-4. Provide enhancement over existing BGP Community Attribute an extended range, the addition of type field.
holdTime	TwoOctetInteger	O	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.
damping	Damping[] maxItems=1	O	Route flap damping. When the Damping parameter is empty, the attribute is not set. When not empty a single set of parameters described in Section 4.3 of RFC 2430 MUST be agreed.
asOverride	EnabledDisabled	O	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.
administrativeDistance	integer minimum = 1	O	BGP Administrative Distance.

Table 71 Bgp

16.2.4.2 BgpCommunity

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
autonomousSystemNumber	TwoOctetInteger	O	The first two octets encoding the Autonomous System value.
autonomousSystemDefined	TwoOctetInteger	O	The remaining octets.
semantics	string	O	Text describing how the Seller will handle routes tagged with this Community

Table 72 BgpCommunity

16.2.4.3 BgpExtendedCommunity

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
regularType	integer minimum = 0 maximum = 255	O	Regular Type Field, 1 octet length
regularValue	integer minimum = 0 maximum = 72057594037927935	O	Octets 2 - 8 of the value part of the address. Used in case only Regular Type is provided.
extendedType	TwoOctetInteger	O	Extended Type Field, 2 octets length
extendedValue	integer minimum = 0 maximum: 281474976710655	O	Octets 3 - 8 of the value part of the address. Used in case only Extended Type is provided.
semantics	string	O	Text describing how the Seller will handle routes tagged with this Community

Table 73 BgpExtendedCommunity

oneOf:

- required: [regularType, regularValue]

- required: [extendedType, extendedValue]

16.2.4.4 AddressFamilyIpv4Ipv6

File: /ip/common/ipCommon.yaml

Specifies whether the session is established over IPv4 or IPv6.

Value
IPV4
IPV6

Table 74 AddressFamilyIpv4Ipv6

16.2.4.5 Damping

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
cutoffThreshold	integer minimum = 0	O	This value is expressed as a number of route withdrawals. It is the value above which a route advertisement will be suppressed.
decayHalfLifeWhileReachable	integer minimum = 0	O	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).
decayHalfLifeWhileUnReachable	integer minimum = 0	O	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 minimum = 0	O	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.
maxHoldDownTime	integer minimum = 0	O	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.
numberOfReuseLists	integer minimum = 0	O	This is the number of reuse lists. It may be determined from reuse-list-max or set explicitly.
reuseListMemory	integer minimum = 0	O	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.
reuseListTimeGranularity	integer minimum = 0	O	This is the time (in seconds) interval between evaluations of the reuse lists. Each reuse lists corresponds to an additional time increment.
reuseThreshold	integer minimum = 0	O	This value is expressed as a number of route withdrawals. It is the value below which a suppressed route will now be used again.
timeGranularity	integer minimum = 0	O	This is the time granularity in seconds used to perform all decay computations.
reuseIndexArraySize	integer minimum = 0	O	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.

Table 75 Damping

16.2.4.6 PeeringAddress

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
peeringAddressType	PeeringAddressType	O	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.
subscriberAndSpLoopbackAddresses	SubscriberAndSpLoopbackAddresses[]	O	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.

Table 76 PeeringAddress

16.2.4.7 PeeringAddressType

File: /ip/common/ipCommon.yaml

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.

Value
CONNECTION_ADDRESSES
LOOPBACKS

Table 77 PeeringAddressType

16.2.4.8 SubscriberAndSpLoopbackAddresses

File: /ip/common/ipCommon.yaml

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.

Name	Type	M/O	Description
subscriberLoopbackAddress	Ipv4OrIpv6Address	O	Subscriber's loopback Address for BGP establishing a session
spLoopbackAddress	Ipv4OrIpv6Address	O	Service Provider's loopback Address for BGP establishing a session

Table 78 SubscriberAndSpLoopbackAddresses

16.2.5 OSPF

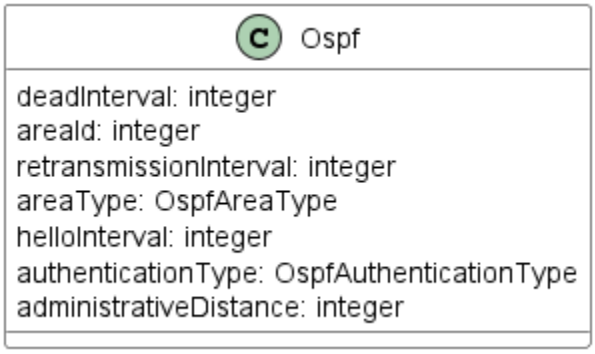


Figure 24 Ospf

Figure 24 Presents the model of OSPF configuration. It consists only of simple attributes and enumerations.

16.2.5.1 Ospf

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
areaId	Ipv4Address	O	Area ID expressed as an IPv4 address.
areaType	OspfAreaType	O	OSPF Area Type enumeration.
authenticationType	OspfAuthenticationType	O	OSPF Authentication Type.
helloInterval	TwoOctetInteger	O	Hello interval (0-65535, in seconds)
deadInterval	integer minimum = 0 maximum = 429967295	O	Dead interval (0-429496295, in seconds)
retransmissionInterval	integer minimum = 0	O	Retransmit interval (integer greater than 0, in seconds)
administrativeDistance	integer minimum = 1	O	Administrative distance (integer greater than 0)

Table 79 Ospf

16.2.5.2 OspfAreaType

File: /ip/common/ipCommon.yaml

OSPF Area Type enumeration. Reference: MEF 61.1 Section 12.7.2

NORMAL - the Area is not a stub or NSSA (Not So Stubby Area)

STUB - the Area is a stub

NSSA - the Area is NSSA (see RFC 3101[7])

Value
NORMAL
STUB
NSSA

Table 80 OspfAreaType

16.2.5.3 OspfAuthenticationType

File: /ip/common/ipCommon.yaml

Enumeration of possible OSPF Authentication Type. In case other than “NONE” is specified additional specific parameters need to be agreed between the Buyer and the Seller.

- NONE - No authentication is used.
- PASSWORD - the 64-bit clear password is used which is inserted into the OSPF packet header
- MESSAGE_DIGEST - Cryptographic authentication is used as specified in RFC 2828 [6]

Value
NONE
PASSWORD
MESSAGE_DIGEST

Table 81 OspfAuthenticationType

16.2.6 Static

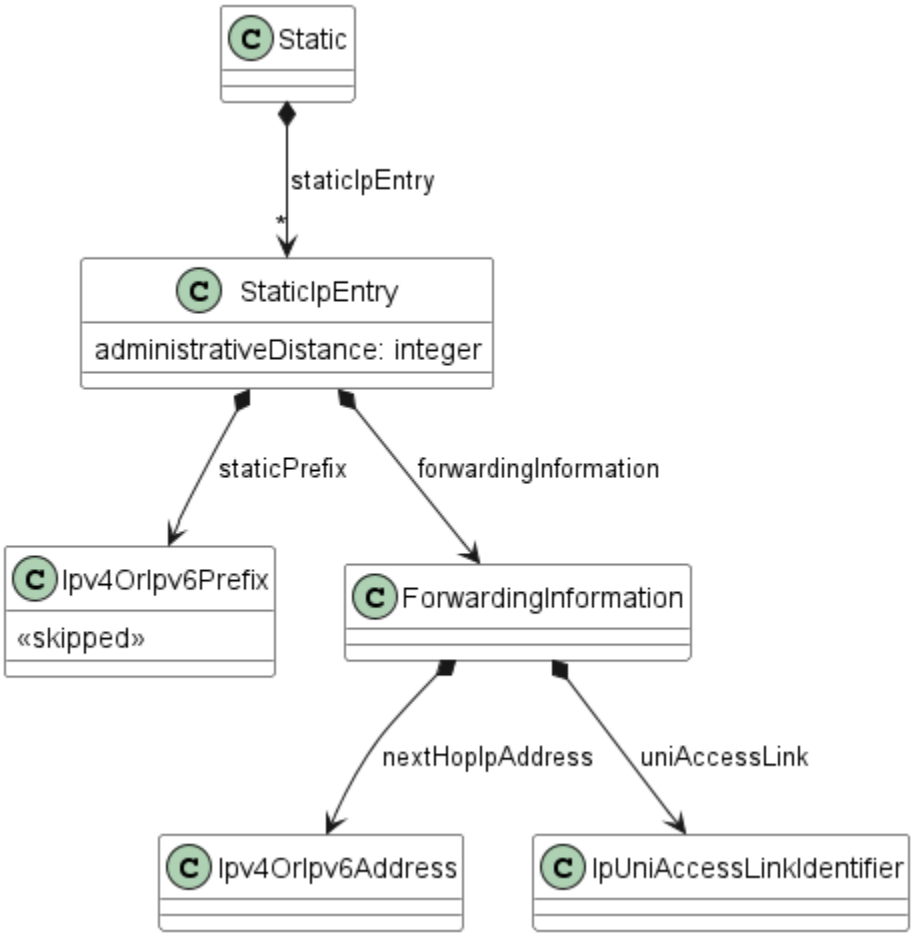


Figure 25 Static

Figure 25 shows the resource model for Static routing configuration.

16.2.6.1 Static

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
staticIpEntry	StaticIpEntry[] minItems = 1	O	Static IP address entry.

Table 82 Static

16.2.6.2 StaticIpEntry

File: /ip/common/ipCommon.yaml

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

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

Table 83 StaticIpEntry

16.2.6.3 ForwardingInformation

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
nextHopIpAddress	Ipv4OrIpv6Address	O	Next hop IP address.
uniAccessLink	IdentifierString	O	UNI Access Link identifier as set by the Buyer in IpUniAccessLink.identifier attribute.

Table 84 ForwardingInformation

oneOf:

- required: [nextHopIpAddress]

- required: [uniAccessLink]

16.3 Bandwidth Profiles

16.3.1 Bandwidth Profile Envelopes

16.3.1.1 BasicIaBwpEnvelope

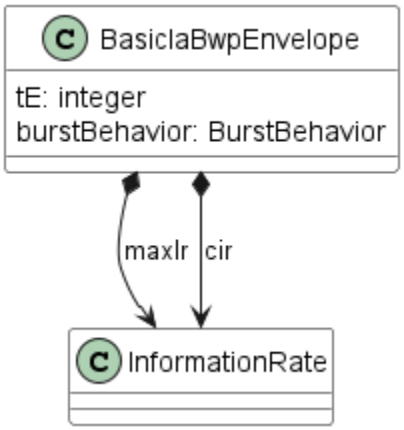


Figure 26 BasicIaBwpEnvelope

Błąd! Nie można odnaleźć źródła odwołania. shows a simple model of BasicIaBwpEnvelope. It l everages MEF 69.1 [19] requirements to Basic Internet Access and simplifies the model, comparing to the advanced one.

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
burstBehavior	BurstBehavior	O	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	O	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.
maxIr	InformationRate	O	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 minimum = 0	O	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

Table 85 BasicIaBwpEnvelope

16.3.1.2 IpBwpEnvelope

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
maxIrE	InformationRate	O	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
tE	integer minimum = 0	O	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

Table 86 IpBwpEnvelope

16.3.1.3 IpvcEpBwpEnvelope

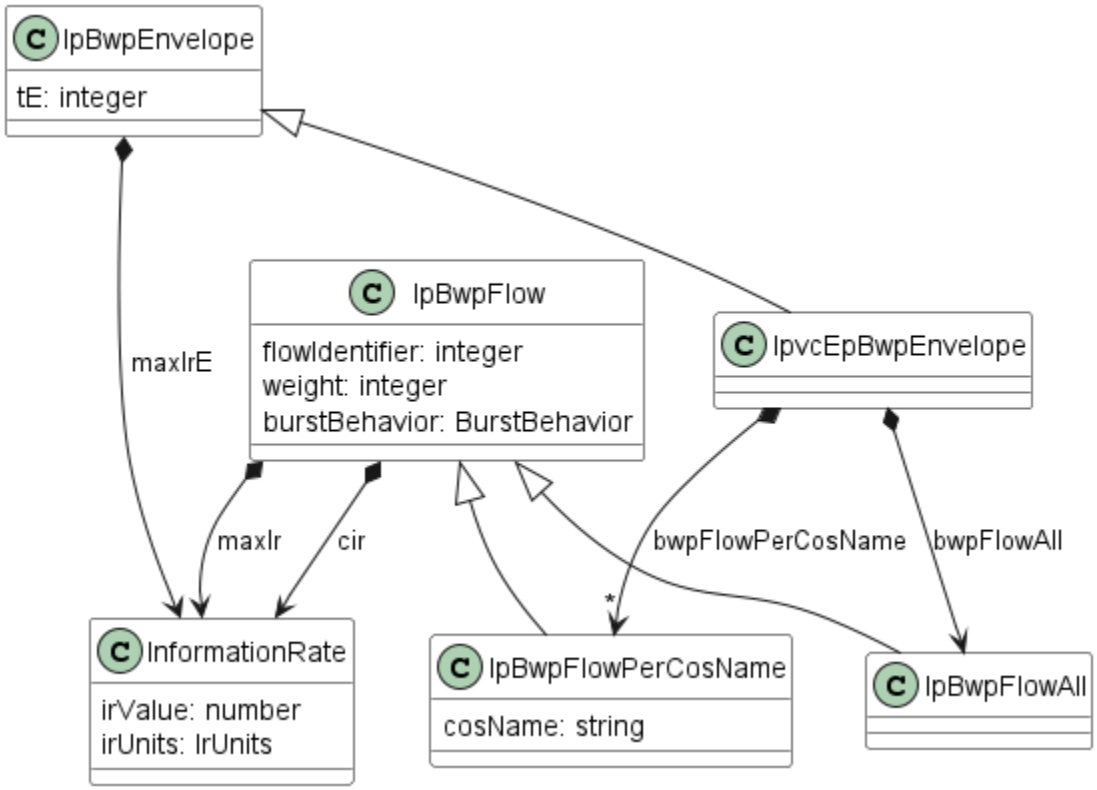


Figure 27 IpvcEpBwpEnvelope

IPVC End Point Bandwith Profile Envelope extends the IpBwpEnvelope to specify possiblies of Flow configurations that can be applied at the IPVC End Point.

File: /ip/common/ipCommon.yaml

A single Bandwidth Profile Envelope consisting of parameters and Bandwidth Profile specifications. A Bandwidth Profile Envelope can be specified for one of a UNI, a UNI Access Link, an ENNI Link, or an IPVC End Point. Reference - MEF 61.1 Section 11.11, 11.12.

Inherits from: - IpBwpEnvelope

Name	Type	M/O	Description
bwpFlowPerCosName	IpBwpFlowPerCosName[] minItems = 1	O	List of BWP flows, each matching one of a set of CoS Names.
bwpFlowAll	IpBwpFlowAll	O	All IP Packets mapped to the IPVC End Point

Table 87 IpvcEpBwpEnvelope

oneOf:

- required: [bwpFlowPerCosName]

- required: [bwpFlowAll]

16.3.1.4 IpUniBwpEnvelope

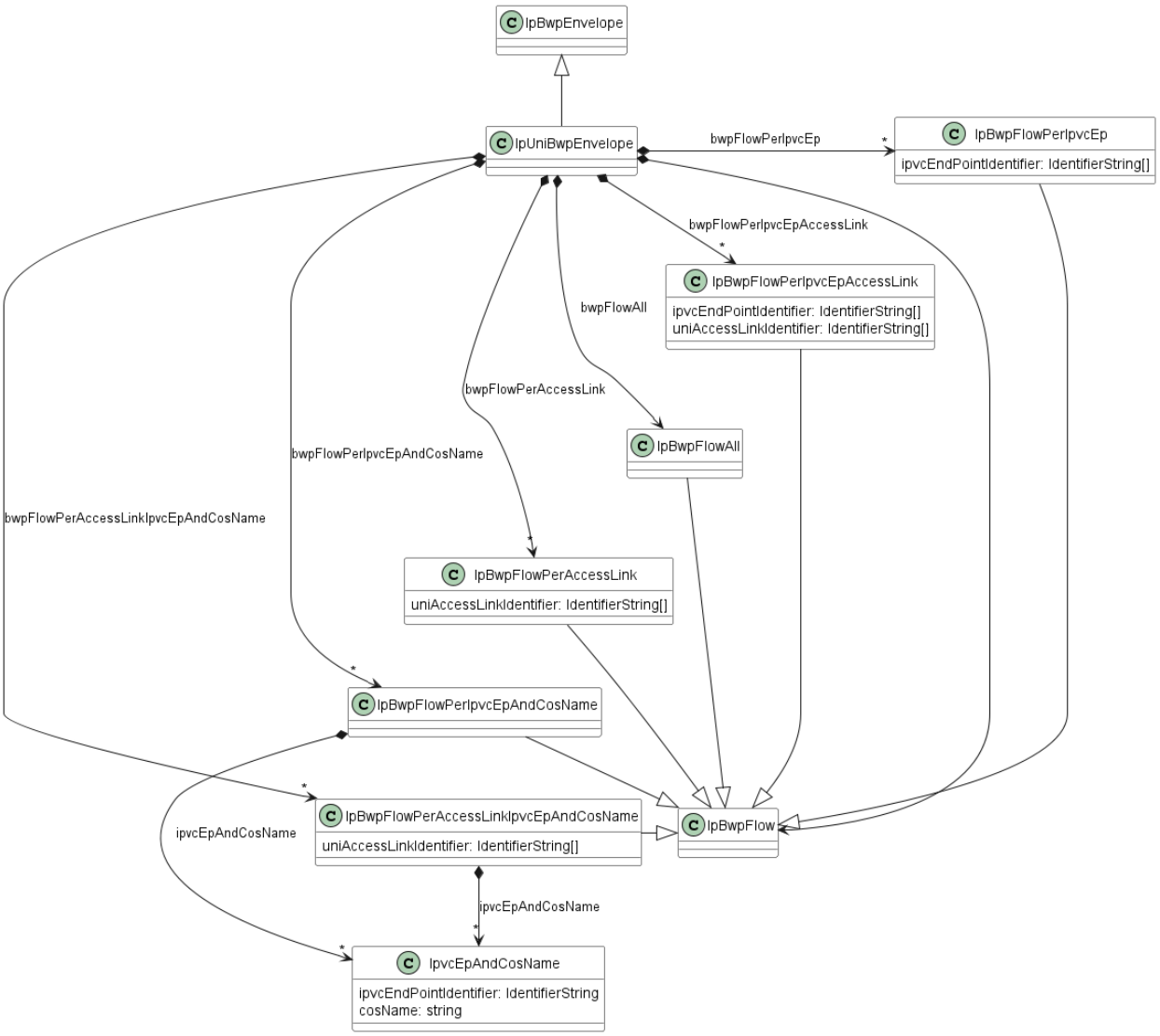


Figure 28 IpUniBwpEnvelope

IP UNI Bandwidth Profile Envelope extends the IpBwpEnvelope to specify possibilities of Flow configurations that can be applied at the IP UNI.

File: /ip/common/ipCommon.yaml

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

Inherits from: IpBwpEnvelope

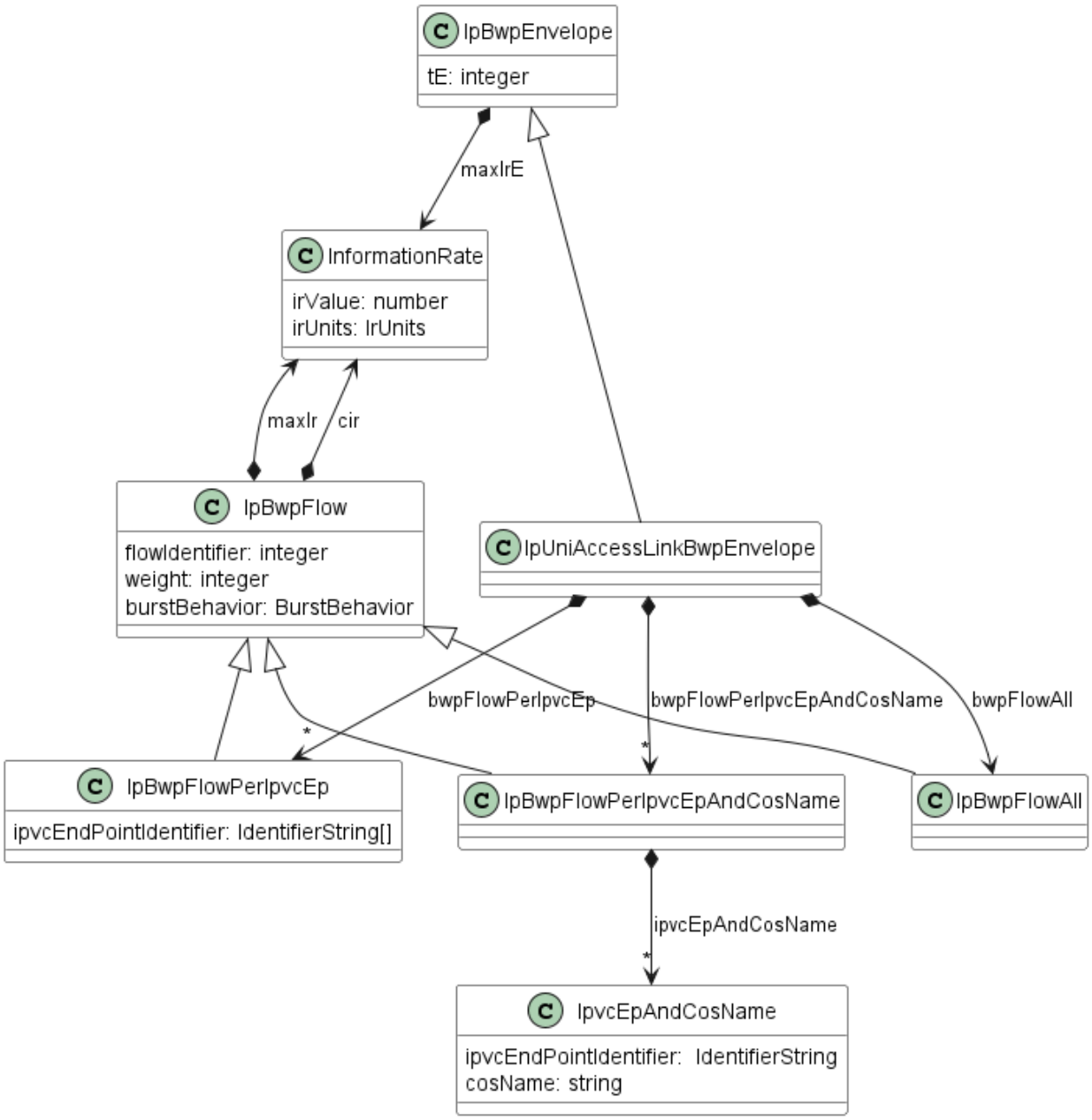
Name	Type	M/O	Description
bwpFlowPerAccessLink	IpBwpFlowPerAccessLink[] minItems = 1	O	A list of BWP Flows for IP Packets that are received over one of a given set of UNI Access Links. Reference - MEF 61.1 Section 12.5
bwpFlowPerIpvcEp	IpBwpFlowPerIpvcEp[] minItems = 1	O	A list of BWP Flows that are mapped to any of a given set of IPVC EPs. Reference - MEF 61.1 Section 12.5
bwpFlowPerIpvcEpAccessLink	IpBwpFlowPerIpvcEpAccessLink[] minItems = 1	O	A list of BWP Flows for IP Packets that are received over one of a given set of UNI Access Links and are mapped to any of a given set of IPVC EPs. Reference - MEF 61.1 Section 12.5
bwpFlowPerIpvcEpAndCosName	IpBwpFlowPerIpvcEpAndCosName[] minItems = 1	O	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
bwpFlowPerAccessLinkIpvcEpAndCosName	IpBwpFlowPerAccessLinkIpvcEpAndCosName[] minItems = 1	O	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
bwpFlowAll	IpBwpFlowAll	O	A BWP Flow for all IP Data Packets at the UNI. Reference - MEF 61.1 Section 12.5

Table 88 IpUniBwpEnvelope

oneOf:

- required: [bwpFlowPerAccessLink]
- required: [bwpFlowPerIpvcEp]
- required: [bwpFlowPerIpvcEpAccessLink]
- required: [bwpFlowPerIpvcEpAndCosName]
- required: [bwpFlowPerAccessLinkIpvcEpAndCosName]
- required: [bwpFlowAll]

1735 #### 16.3.1.5 IpUniAccessLinkBwpEnvelope



1736

1737 **Figure 29 IpUniAccessLinkBwpEnvelope**

1738 IP UNI Access Link Bandwidth Profile Envelope extends the IpBwpEnvelope to specify
1739 possibilities of Flow configurations that can be applied at the IP UNI Access Link.

1740 File: /ip/common/ipCommon.yaml

1741 A single Bandwidth Profile Envelope consisting of parameters and Bandwidth Profile Flow
1742 specifications. An Ingress Bandwidth Profile Envelope can be specified for one of a UNI, a UNI
1743 Access, or an IPVC EP. Reference - MEF 61.1 Section 13.10

Inherits from: IpBwpEnvelope

Name	Type	M/O	Description
bwpFlowAll	IpBwpFlowAll	O	BWP Flow for all IP Data Packets at the UNI that are transmitted or received over the UNI Access Link.
bwpFlowPerIpvcEp	IpBwpFlowPerIpvcEp[]	O	List of BWP Flows matching IPVC End Point Identifier(s) for an IPVC EP located at the UNI Access Link.
bwpFlowPerIpvcEpAndCosName	IpBwpFlowPerIpvcEpAndCosName[]	O	List of BWP Flows matching pairs of IPVC End Point Identifier and CoS Name.

Table 89 IpUniAccessLinkBwpEnvelope

oneOf:

- required: [bwpFlowAll]

- required: [bwpFlowPerIpvcEp]

- required: [bwpFlowPerIpvcEpAndCosName]

16.3.2 Bandwidth Profile Flows

16.3.2.1 IpBwpFlow

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
flowIdentifier	integer minimum = 1	O	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.
cir	InformationRate	O	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.
maxIr	InformationRate	O	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.
weight	integer minimum = 0	O	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	O	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.

Table 90 IpBwpFlow

16.3.2.2 IpBwpFlowAll

File: /ip/common/ipCommon.yaml

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

Inherits from: IpBwpFlow

16.3.2.3 IpBwpFlowPerAccessLink

File: /ip/common/ipCommon.yaml

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

Inherits from: IpBwpFlow

Name	Type	M/O	Description
uniAccessLinkIdentifier	IdentifierString[] minItems = 1	O	List of UNI Access Link Identifiers.

Table 91 IpBwpFlowPerAccessLink

16.3.2.4 IpBwpFlowPerAccessLinkIpvcEpAndCosName

File: /ip/common/ipCommon.yaml

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

Inherits from: IpBwpFlow

Name	Type	M/O	Description
ipvcEpAndCosName	IpvcEpAndCosName [] minItems = 1	O	List of pairs of IPVC End Point Identifier and Class of Service Name. Reference - MEF 61.1 Table 28.
uniAccessLinkIdentifier	IdentifierString[] minItems = 1	O	List of UNI Access Link Identifiers.

Table 92 IpBwpFlowPerAccessLinkIpvcEpAndCosName

16.3.2.5 IpBwpFlowPerIpvcEp

File: /ip/common/ipCommon.yaml

All Egress/Ingress IP Data Packets at the UNI that are are mapped to any of a given set of IPVC End Points. Reference - MEF 61.1 Section 13.10, 13.11

Inherits from: IpBwpFlow

Name	Type	M/O	Description
ipvcEndPointIdentifier	IdentifierString[] minItems = 1	O	List of IPVC End Point Identifiers for IPVC End Points. Reference - MEF 61.1 Table 28.

Table 93 IpBwpFlowPerIpvcEp

16.3.2.6 IpBwpFlowPerIpvcEpAccessLink

File: /ip/common/ipCommon.yaml

All Ingress IP Data Packets at the UNI that are received over one of a given set of UNI Access Links and are mapped to one of a given set of IPVC End Points. Reference - MEF 61.1 Section 12.4

Inherits from: IpBwpFlow

Name	Type	M/O	Description
ipvcEndPointIdentifier	IdentifierString[] minItems = 1	O	List of IPVC End Point identifiers as described in MEF 61.1 Section 11.1.
uniAccessLinkIdentifier	IdentifierString[] minItems = 1	O	List of UNI Access Link Identifiers.

Table 94 IpBwpFlowPerIpvcEpAccessLink

16.3.2.7 IpBwpFlowPerIpvcEpAndCosName

File: /ip/common/ipCommon.yaml

All Ingress IP Data Packets at the UNI that are mapped to any of a given set of (IPVC EP, CoS Name) pairs. Inherits from: IpBwpFlow

Name	Type	M/O	Description
ipvcEpAndCosName	IpvcEpAndCosName[] minItems = 1	O	List of pairs of IPVC End Point Identifier and Class of Service Name. Reference - MEF 61.1 Table 28.

Table 95 IpBwpFlowPerIpvcEpAndCosName

16.3.2.8 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

16.3.2.9 IpBwpFlowPerCosName

File: /ip/common/ipCommon.yaml

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

Inherits from: IpBwpFlow

Name	Type	M/O	Description
cosName	string[] minItems = 1	O	List of Class of Service names.

Table 97 IpBwpFlowPerCosName

16.3.2.10 IpvcEpAndCosName

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
ipvcEndPointIdentifier	IdentifierString	O	IPVC End Point identifier as described in MEF 61.1 Section 11.1.
cosName	string	O	Class of Service Name.

Table 98 IpvcEpAndCosName

16.4 IP Addressing

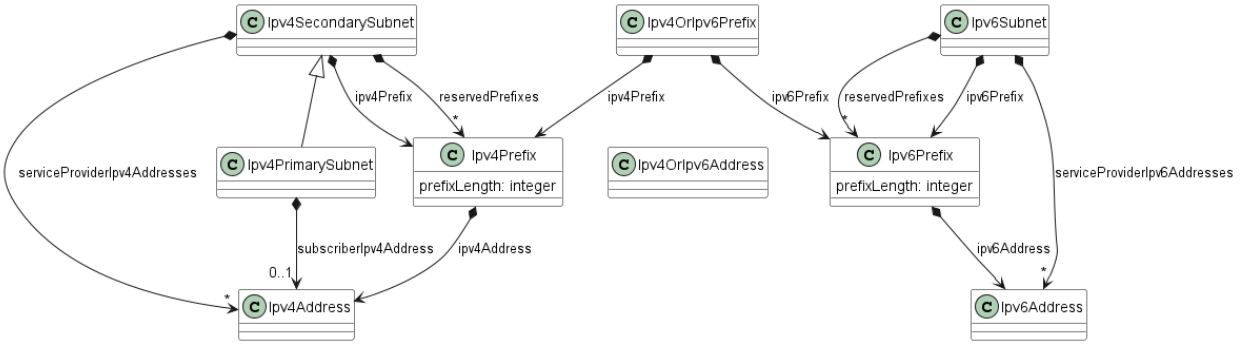


Figure 30 IP Addressing

Figure 30 illustrates the model of Ipv4 and Ipv6 addressing. Note that the API schema leverages the OAS embedded ipv4 and ipv6 string formats and uses them to specify the Ipv4Address and Ipv6Address data types that are uses whenever an address value must be provided.

16.4.1 Ipv4Address

File: /ip/common/ipCommon.yaml

Data type representing Ipv4 address.

Format: ipv4

16.4.2 Ipv4Prefix

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
ipv4Address	Ipv4Address	O	IPv4 address.
prefixLength	integer minimum = 0 maximum = 31	O	IPv4 address prefix. Length 0-31.

Table 99 Ipv4Prefix

16.4.3 Ipv4PrimarySubnet

File: /ip/common/ipCommon.yaml

IPv4 Subnet used in context of Primary Ipvc subnet. It adds the subscriberIpv4Address attribute to the Ipv4SecondarySubnet.

Inherits from: Ipv4SecondarySubnet

Name	Type	M/O	Description
subscriberIpv4Address	Ipv4Address[] maxItems=1	O	Subscriber IPv4 Address

Table 100 Ipv4PrimarySubnet

16.4.4 Ipv4SecondarySubnet

File: /ip/common/ipCommon.yaml

Data type representing an IPv4 subnet logical partition of an IP network. Included is list of Service Provider IPv4 addresses.

Name	Type	M/O	Description
ipv4Prefix	Ipv4Prefix	O	IPv4 address prefix (IPv4 address prefix and mask length between 0 and 31 in bits).
serviceProviderIpv4Addresses	Ipv4Address[] minItems = 1	O	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 Section 9.4
reservedPrefixes	Ipv4Prefix[]	O	List of IPv4 Prefixes, possibly empty

Table 101 SecondarySubnet

16.4.5 Ipv6Address

File: /ip/common/ipCommon.yaml

Data type representing IPv6 address.

Format: ipv6

16.4.6 Ipv6Prefix

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
ipv6Address	Ipv6Address	O	IPv6 address.
prefixLength	integer minimum = 0 maximum = 127	O	IPv6 address prefix. Length 0-127.

Table 102 Ipv6Prefix

16.4.7 Ipv6Subnet

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
ipv6Prefix	Ipv6Prefix	O	IPv6 Prefix (IPv6 address prefix and mask length between 0 and 127 in bits).
serviceProviderIpv6Addresses	Ipv6Address[] minItems = 1	O	List of IPv6 Service Provider addresses.
reservedPrefixes	Ipv6Prefix[]	O	List of IPv6 Prefixes, possibly empty

Table 103 Ipv6Subnet

16.4.8 Ipv4OrIpv6Address

File: /ip/common/ipCommon.yaml

Data type representing IPv4 or IPV6 address.

oneOf:

- format: ipv4

- format: ipv6

16.4.9 Ipv4OrIpv6Prefix

File: /ip/common/ipCommon.yaml

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

Name	Type	M/O	Description
ipv4Prefix	Ipv4Prefix	O	IPv4 prefix.
ipv6Prefix	Ipv6Prefix	O	IPv6 prefix.

Table 104 Ipv4OrIpv6Prefix

oneOf:

- required: [ipv4Prefix]

- required: [ipv6Prefix]

16.5 Common Classes

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

16.5.1 EnabledDisabled

File: /ip/common/ipCommon.yaml

Enumeration to indicate Enabled/Disabled state of an attribute

Value
ENABLED
DISABLED

Table 105 EnabledDisabled

16.5.2 IdentifierString

File: /ip/common/ipCommon.yaml

Data type used for common identifier string requirements definition.

A string; maxLength: 53; pattern: "[\x20-\x7F]+".

16.5.3 InformationRate

File: /ip/common/ipCommon.yaml

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

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

Table 106 InformationRate

16.5.4 IrUnits

File: /ip/common/ipCommon.yaml

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

Value
BPS
KBPS
MBPS
GBPS
TBPS
PBPS
EBPS
ZBPS
YBPS

Table 107 IrUnits

16.5.5 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	M/O	Description
timeDurationValue	integer minimum = 0	O	The value of the duration. For example, if the duration is 20 ms, this element is 20.
timeDurationUnits	TimeDurationUnits	O	The unit of measure in the duration. For example, if an interval is 2ms, this element is MS.

Table 108 TimeDuration

16.5.6 TimeDurationUnits

File: /ip/common/ipCommon.yaml

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

Table 109 TimeDurationUnits

16.5.7 TwoOctetInteger

File: /ip/common/ipCommon.yaml

A two octet integer. Value range 0 - 65535

16.5.8 FourOctetInteger

File: /ip/common/ipCommon.yaml

A four-octet value range integer 0-4294967295

17 References

[1] IEC 60603-7, Connectors for electronic equipment - Part 7: Detail specification for 8-way, unshielded, free and fixed connectors, Edition 4.0, October 2020

[2] IEC 61754-4, Fiber optic interconnecting devices and passive component - Fiber optic connector interfaces - Part 4: Type SC connector family, Edition 2.0, July 2013

[3] IEC 61754-20, Fiber optic interconnecting devices and passive component - Fiber optic connector interfaces - Part 20: Type LC connector family, Edition 2.0, April 2012

[4] 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.

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

[6] IETF RFC 2828, Internet Security Glossary, May 2000

[7] IETF RFC 3101, The OSPF Not-So-Stubby Area (NSSA) Option, January 2003

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

[9] IETF RFC 5880, Bidirectional Forwarding Detection (BFD), June 2010

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

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

[12] ITU-T Recommendation G.8262/Y.1362, Timing characteristics of synchronous Ethernet equipment slave clock, January 2015

[13] ITU-T Recommendation G.8264/Y.1364, Distribution of timing information through packet networks, May 2014

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

[15] MEF W55.1.1 Amendment to MEF 55.1: Reference Architecture and Framework - Terminology, June 2023

[16] MEF 57.2 Product Order Management Requirements and Use Cases, October 2022

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

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

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

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

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

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

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

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

[25] MEF 115, LSO Cantata and LSO Sonata Quote Management API - Developer Guide, May 2022

[26] MEF 116, LSO Cantata and LSO Sonata Inventory Management API - Developer Guide, May 2022

[27] MEF 121, LSO Cantata and LSO Sonata Address Management API - Developer Guide, May 2022

[28] MEF 123, LSO Cantata and LSO Sonata Product Order Management API - Developer Guide, December 2022

[29] TIA-568.0-E-2020, Generic Telecommunications Cabling for Customer Premises, March 2020

Appendix A Usage examples (Informative)

This appendix aims to provide an extensive set of examples to cover:

- configurations for each Internet Access product
- basic all APIs steps walkthrough to order an Internet Access product
- modification use cases
- deletion of products

The full examples are delivered as a Postman collection file available at:

- documentation/productSchema/ip/internetAccess/MEF 139 - Appendix A.postman_collection.json

A.1 High-Level flow

The Cantata and Sonata Interface Reference Points are formed from a set of APIs that serve different functions in the end-to-end flow. Figure 31 shows all of the functions and their sequence.

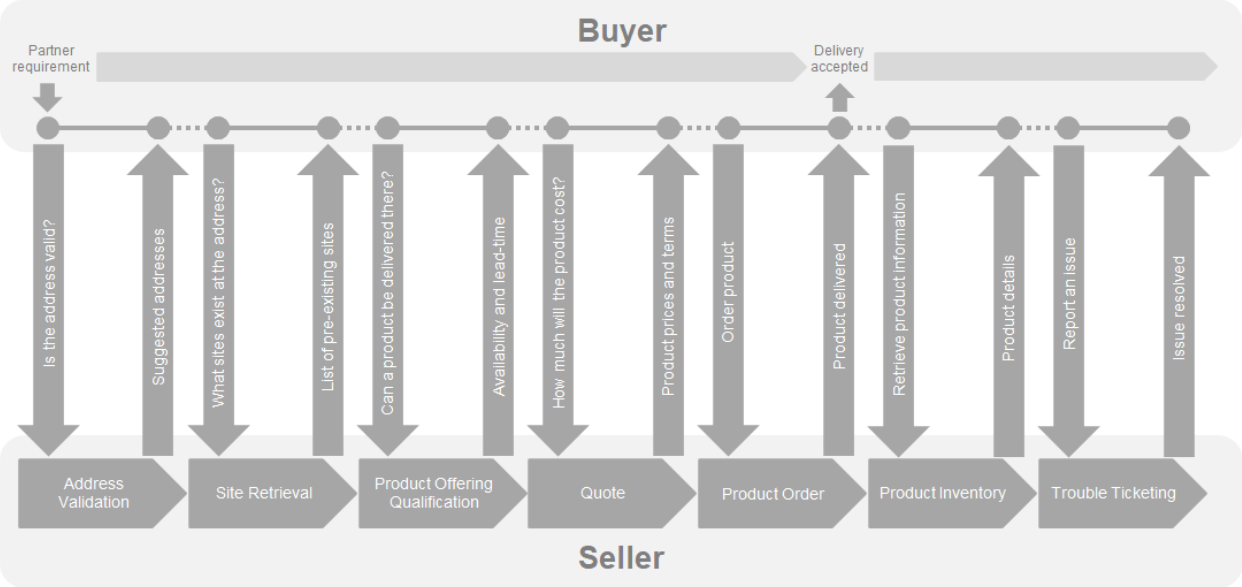


Figure 31 Cantata and Sonata End-to-End Function Flow

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

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

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

A.2 Integration of product specifications into the APIs.

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

The extension hosting type in the API data model is “MEFProductConfiguration”. The “@type” attribute of that type must be set to a value that uniquely identifies the product specification (Figure 32). A unique identifier for MEF standard product specifications is in URN format and is assigned by MEF. This identifier is provided as root schema “$id” and in product specification documentation. In case of Internet Access, this will be one of:

- urn:mef:lso:spec:cantata-sonata:basic-internet-access:v1.0.0:all
- urn:mef:lso:spec:cantata-sonata:advanced-internet-access-ipvc:v1.0.0:all
- urn:mef:lso:spec:cantata-sonata:exclusive-advanced-internet-access:v1.0.0:all
- urn:mef:lso:spec:cantata-sonata:ip-uni:v1.0.0:all
- urn:mef:lso:spec:cantata-sonata:ip-uni-access-link:v1.0.0:all
- urn:mef:lso:spec:cantata-sonata:ethernet-uni-access-link-trunk:v1.0.0:all

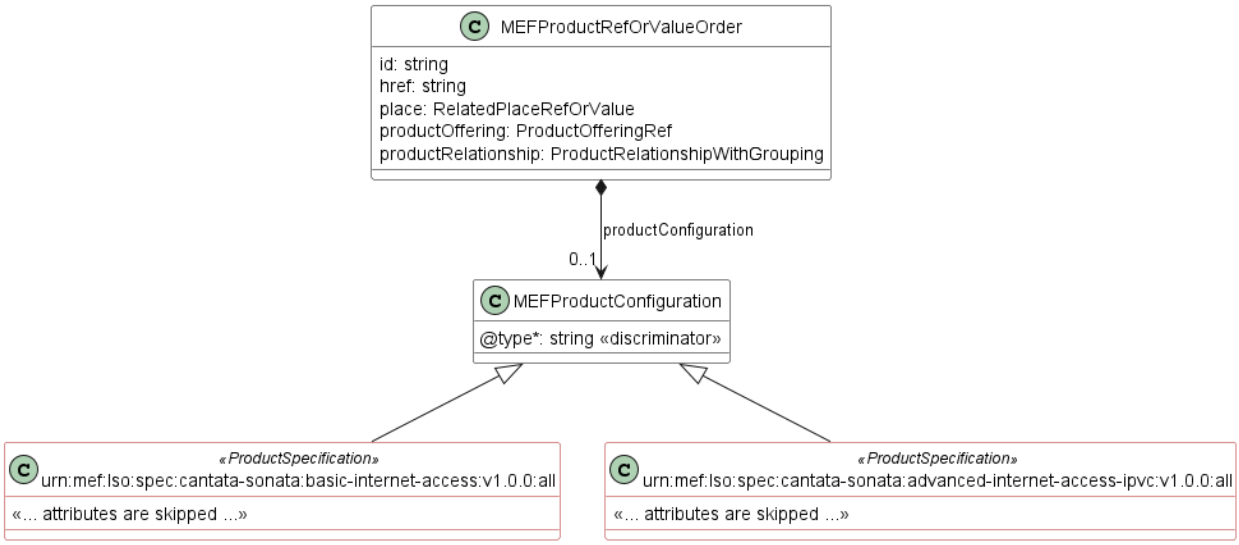


Figure 32 The Extension Pattern

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

```
{
  "externalId": "BuyerPoq-00002",
  "instantSyncQualification": false,
  "provideAlternative": false,
  "projectId": "BuyerProject2",
  "productOfferingQualificationItem": [
    {
      "action": "add",
      "id": "item-00001",
      "product": {
        "productOffering": {
          "id": "BasicInternetAccessOffering-0001"
        },
        "productConfiguration": {
          "@type": "urn:mef:lso:spec:cantata-sonata:basic-internet-
access:v1.0.0:all",
          "ipvc": {
            "maximumNumberOfIpv4Routes": [1],
            "maximumNumberOfIpv6Routes": [1],
            "dscpPreservation": "DISABLED",
            "classOfServiceName": "Best-effort",
            "mtu": 1500,
```

POQ API part

Internet Access Product part

A.3 action: add

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

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

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

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

- "item.action" must be set to "add"
- "item.product.id" must not be provided
- "product.productConfiguration" must contain all desired configurations.

A.3.1 Use Case 1: Address Validation

For detailed guidance on how to use the Address Validation API, please refer to MEF 121 [27]

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

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

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

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

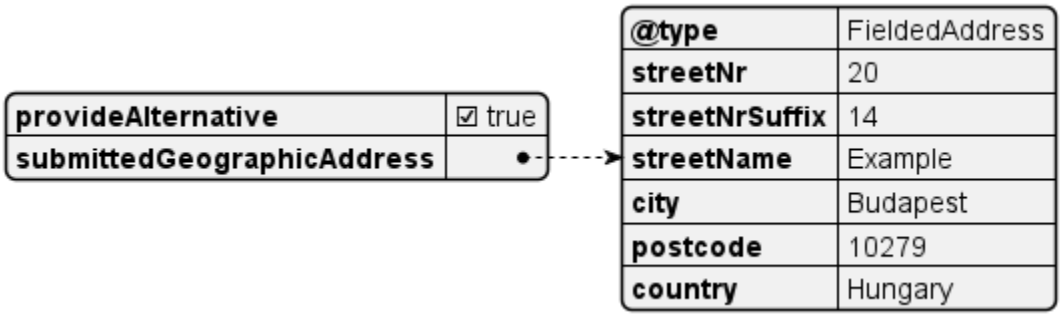


Figure 33 UC1: Address Validation request

Example Address Validation Request:

```
{
  "provideAlternative": true,
  "submittedGeographicAddress": {
    "@type": "FieldedAddress",
    "streetNr": "20",
    "streetNrSuffix": "14",
    "streetName": "Example",
    "city": "Budapest",
    "postcode": "10279",
    "country": "Hungary"
  }
}
```

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

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

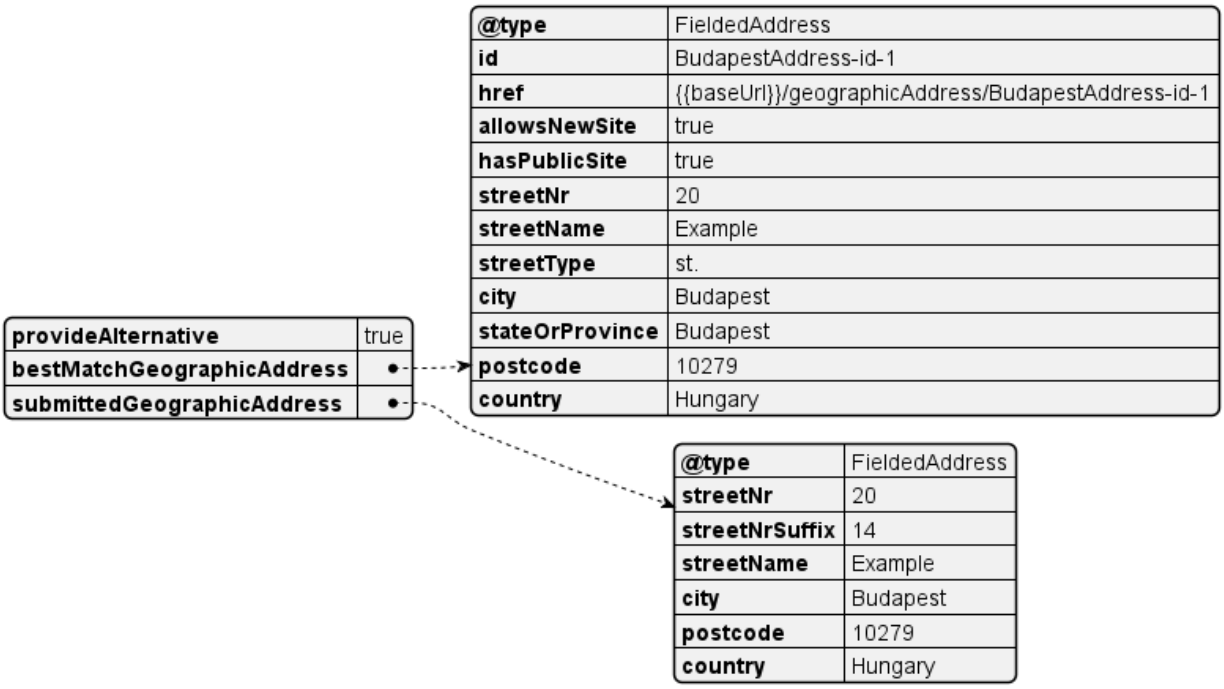


Figure 34 UC1: Address Validation response

Seller's response:


```
{
  "provideAlternative": "true",
  "bestMatchGeographicAddress": {
    "@type": "FieldedAddress",
    "id": "BudapestAddress-id-1",
    "href": "{{baseUrl}}/geographicAddress/BudapestAddress-id-1",
    "allowsNewSite": "true",
    "hasPublicSite": "true",
    "streetNr": "20",
    "streetName": "Example",
    "streetType": "st.",
    "city": "Budapest",
    "stateOrProvince": "Budapest",
    "postcode": "10279",
    "country": "Hungary"
  },
  "submittedGeographicAddress": {
    "@type": "FieldedAddress",
    "streetNr": "20",
    "streetNrSuffix": "14",
    "streetName": "Example",
    "city": "Budapest",
```

```
    "postcode": "10279",
    "country": "Hungary"
  }
}
```

A.3.2 Use Case 2: POQ - Basic Internet Access

For detailed guidance on how to use the Product Offering Qualification (POQ) API, please refer to MEF 87 [23]

The Product Offering Qualification step is designed for the Buyer to ask the question “Dear Seller, are you able to provide a certain product (based on “productOffering”) with specific configuration (provided as “productConfiguration”) at a given location”? The Seller responds with one of qualification confidences:

- green - The Seller has high confidence that this Product can be delivered,
- yellow - The Seller believes they can deliver the Product but is not highly confident,
- red - The Seller cannot deliver the Product as specified.

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

It is very important to understand the pattern of integrating the product-specific configuration with the functional product-agnostic API like POQ. As explained in chapter 10 the Internet Access product model is composed of 4 elements:

- IPVC (incl. IPVC End Point)
- IP UNI
- IP UNI Access Link
- IP UNI Access Link Trunk

A topology diagram is presented in Figure 35. All 4 components are additionally labeled and covered with a single grey rectangle to designate they are all covered by single Basic Internet Access product configuration.

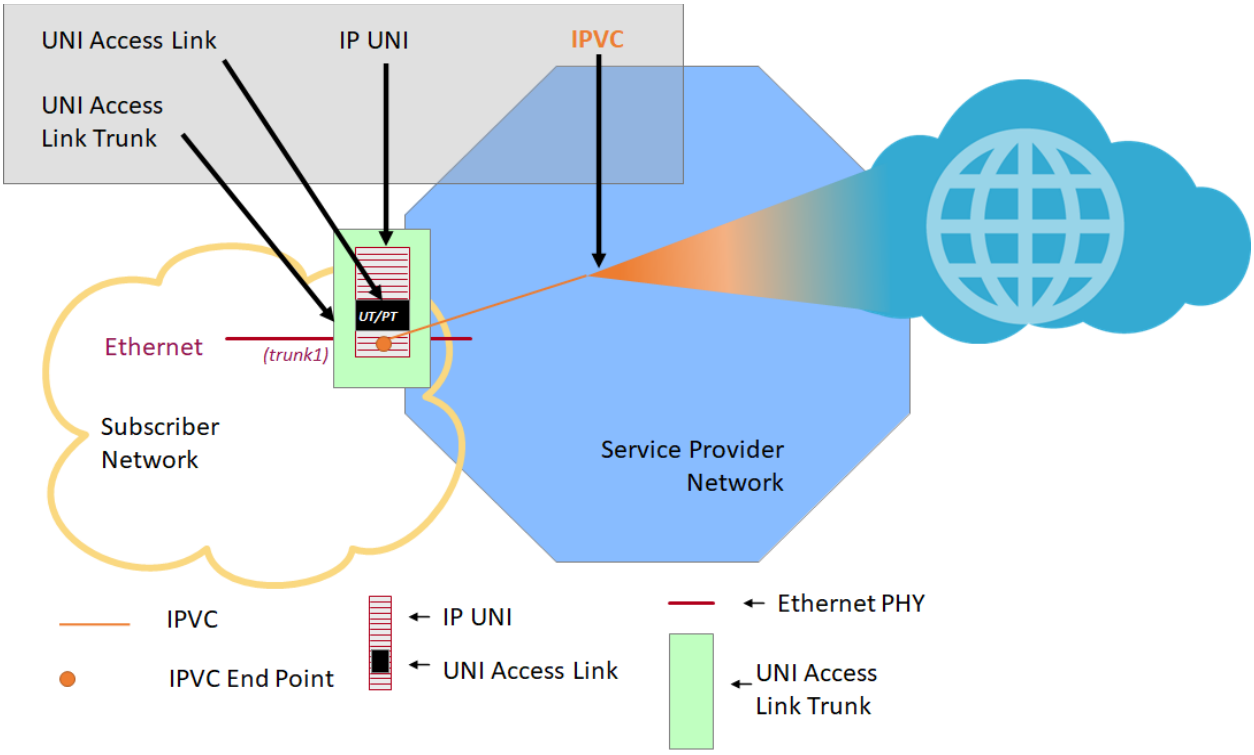


Figure 35 Basic Internet Access Topology

Depending on version (Basic, Advanced, Exclusive), they are either aggregated into one single product definition or managed separately. This maps to a POQ request having one POQ item for Basic, four POQ items for Advanced or two POQ Items in Exclusive Advanced case. This will be covered by examples in this and subsequent sections.

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

There are 2 ways to reference products:

- existing Products - present in the Product Inventory at the moment of issuing the request, to which the Buyer has the "product.id". These must be referenced by "productOffer-ingQualificationItem.product.productRelationship" with appropriate "product.id" and "relationshipType". Product Specification defines what relationship types must be used during referencing other products. E.g. the Advanced Internet Access IPVC points to the IP UNI product with the "relationshipType" value: "CONNECTS_TO_IPUNI" (as specified in Chapter 13).
- newly created or modified products - the ones being created or modified by other POQ Item in the same POQ request, so there is a relation between the Items within a POQ. These must be referenced using the "productOfferingQualificationItem. qualificationItemRelationship" by the target Item "id" and the "relationshipType".

All configurations presented by Use Cases 2 to 6 base on this topology. The attribute values are taken from MEF 61.1[17], section C.3. and applied minimum required changes.

In this use case the Basic Internet Access aggregates all components' configuration into a single product with four main attributes keeping respective configurations. Thus, there is only one POQ Item. Model diagram is presented in Figure 36 to remind the structure of Basic Internet Access.

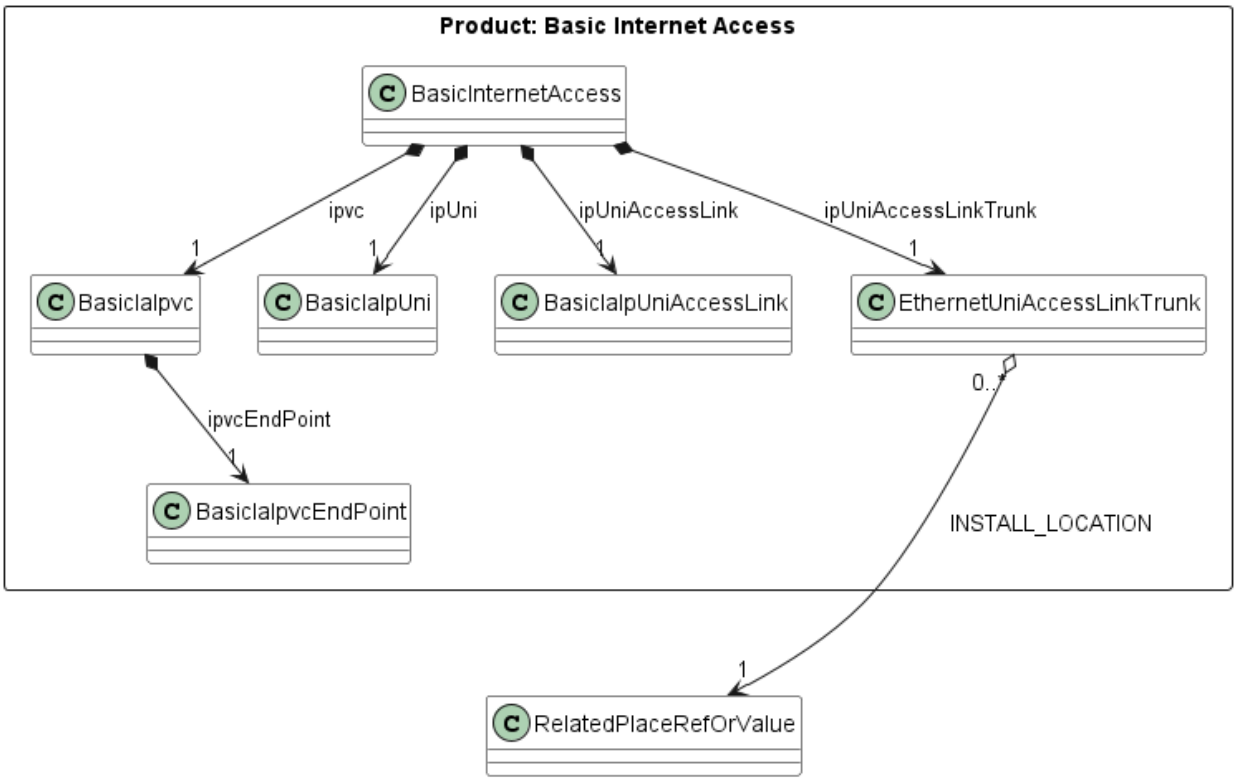


Figure 36 Information model for Basic Internet Access product

The outer rectangle represents the coverage of single product specification.

An instance diagram in Figure 37 shows an extracted part from the request, to present the most important attributes.

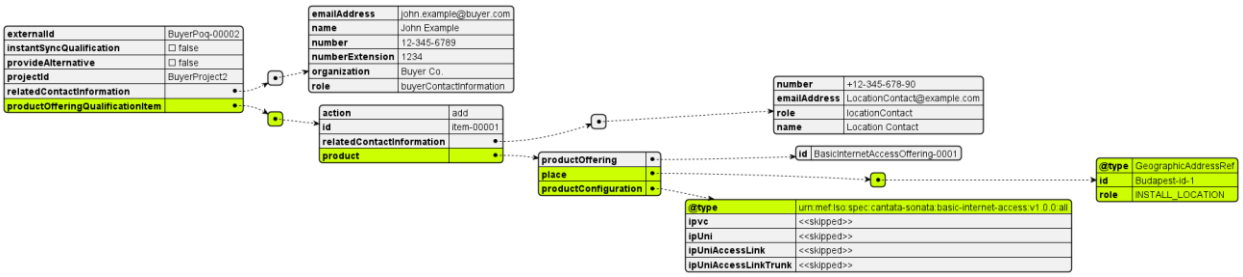


Figure 37 UC2: POQ Request, product-agnostic part

The green color highlights key aspects:

- there is only one productOfferingQualificationItem
- the type of the product is Basic Internet Access: urn:mef:lso:spec:cantata-sonata:basic-internet-access:v1.0.0:all

- the configuration of building components in stored a simple attribute values (<<skipped>> for the sake of readability)
- since it the is only one product - it also holds the relation to install location

Figure 38 shows the IPVC configuration:

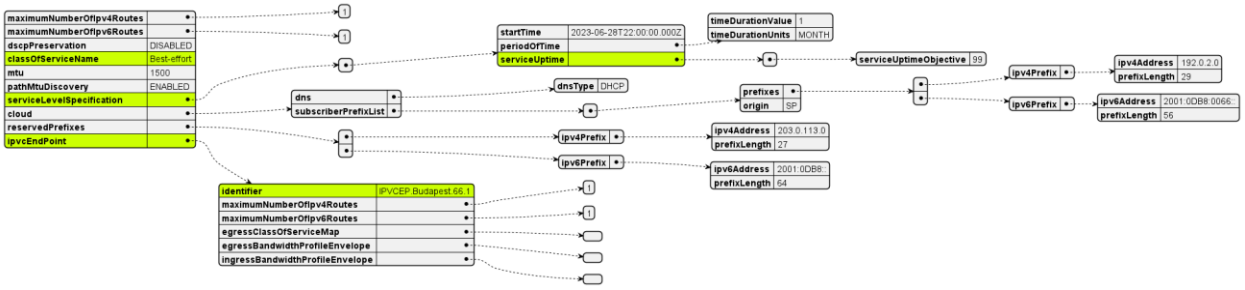


Figure 38 UC2: IPVC configuration

There is one Best-effort class of service defined, one SLS metric: serviceUptime, and one IPVC End Point with identifier: IPVCEP.Budapest.66.1.

Figure 39 shows IP UNI product configuration:

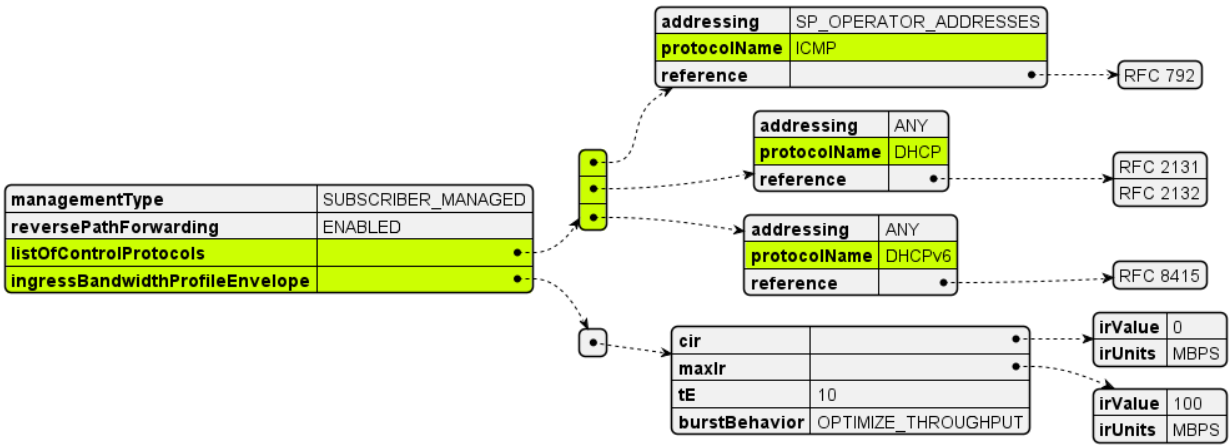


Figure 39 UC2: IP UNI configuration

This UNI has ICMP and DHCP control protocols enabled and best effort maximum bandwidth of 100 MBPS with no committed information rate.

Figure 40 presents the IP UNI access Link product configuration:

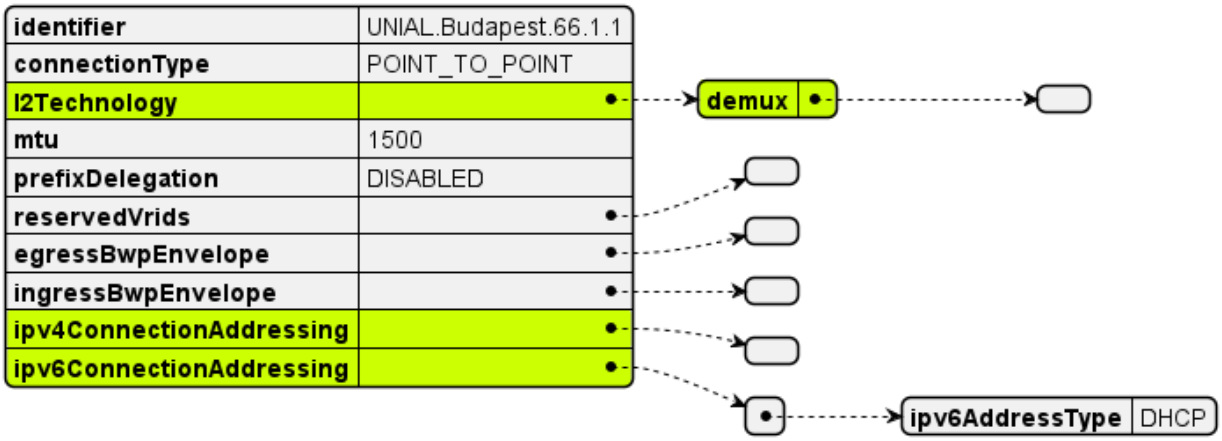


Figure 40 UC2: IP Uni Access Link configuration

The demux has no value provided, which means UT/PT is used. The value UT/PT refers to untagged and priority tagged frames and when set that means that the UNI Access Link Trunk must not be used for any other UNI Access Link. This is the case for the Basic Internet Access, where all resources are dedicated to single IPVC. ipv6ConnectionAddressing is using DHCP explicitly. The ipv6ConnectionAddressing is also using DHCP, yet implicitly. This by the rule that if set, the it must be DHCP. If IPv4 was not used on this UNI Access link, then it would have been provided as an empty list .

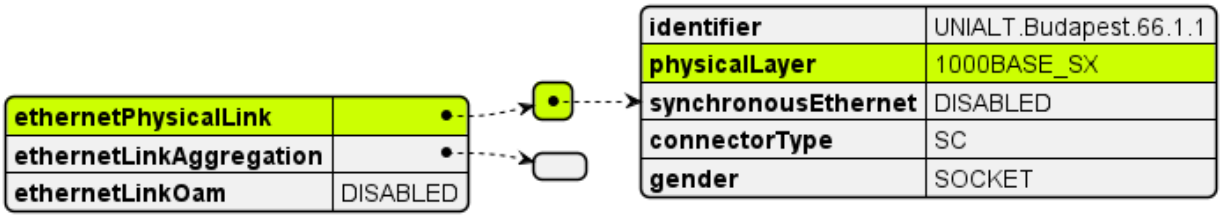


Figure 41 UC2: IP Uni Access Link Trunk configuration

Figure 41 shows IP UNI access Link Trunk part configuration specifying a single ethernet connection with a 1000BASE_SX interface.

A.3.3 Use Case 3: POQ - Advanced Internet Access

The Advanced Internet Access is built from same components as the Basic one. The difference is that in Advanced case all of them are managed separately, can be ordered separately and the cardinality of the relations between them is not restricted to only one, thus they can serve more products (following relations cardinalities defined in section 13).

A topology diagram is presented in Figure 42. All four components are now covered be separate rectangles to underline that each of them is now a different product.

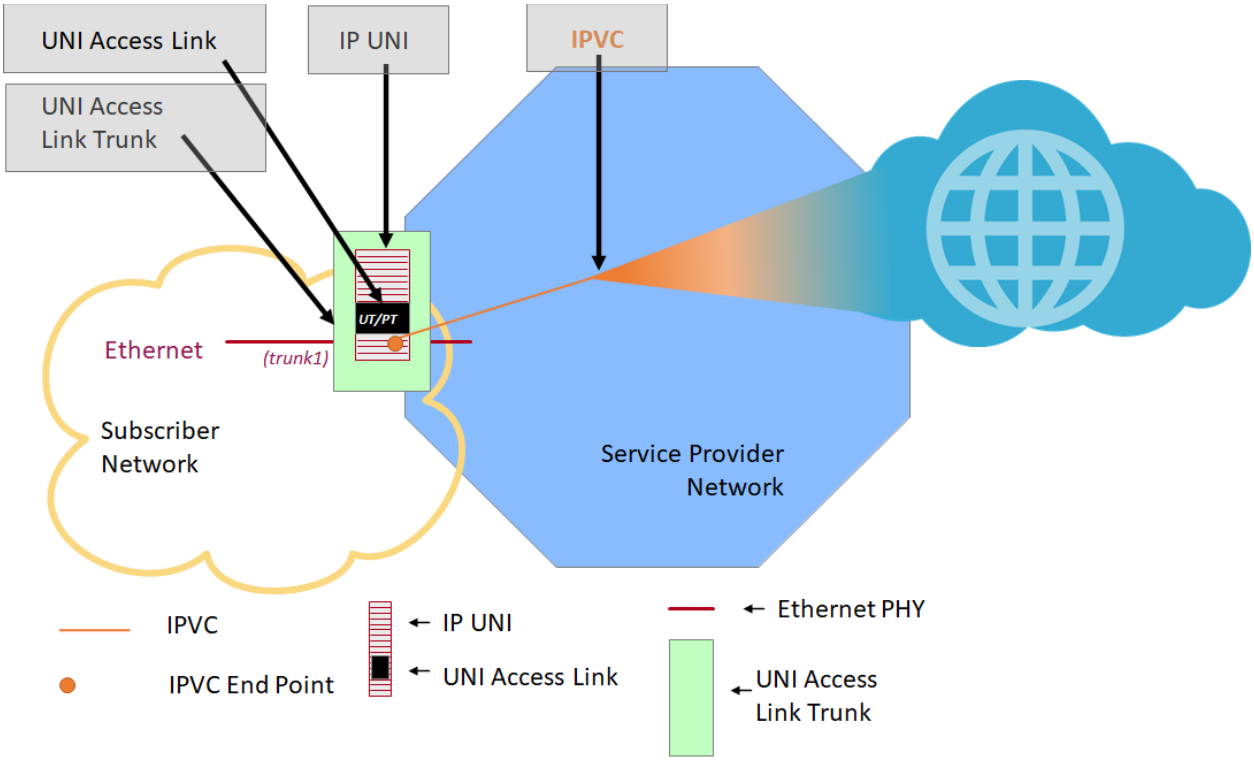


Figure 42 Advanced Internet Access topology

The example provided in request collection attached to this document covers topology using single cardinalities that is similar to the topology of Use Case 2. This is to pinpoint the differences between them.

The greatest difference is the structure of the request, as presented in Figure 43. Now there four distinct POQ items each carrying respective product configuration, having own URN, and specifying relations between them. Note also that the place relationship is now defined by Ethernet UNI Access Link Trunk.

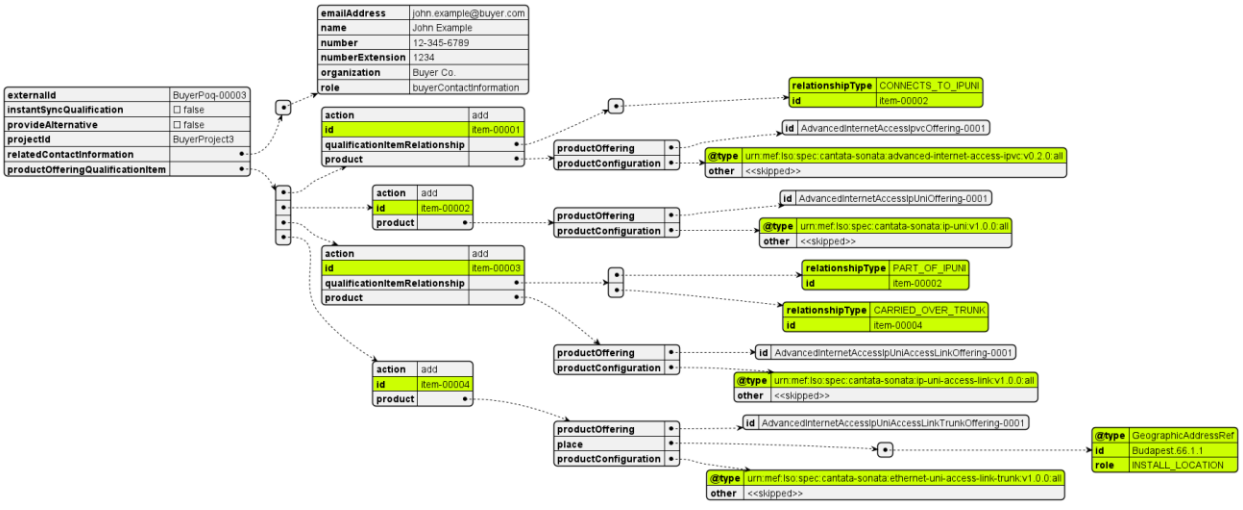


Figure 43 UC3: POQ Request, product-agnostic part

Figure 44 present a diagram of the Advanced Internet Access IPVC product configuration. The only difference comparing to the Basic one is the presence of the @type and the ipvcEndPoint.prefixMapping attribute, which in this use case is an empty list. Please refer to Table 5 and **Błąd! Nie można odnaleźć źródła odwołania.** which list the details of all discrepancies between the Basic and Advanced versions.

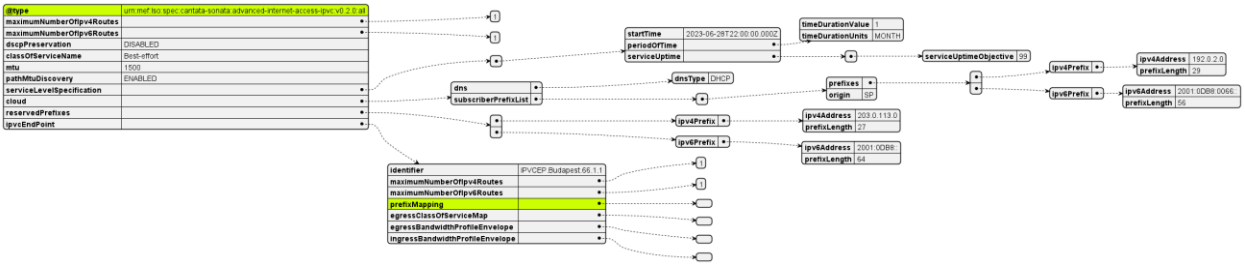


Figure 44 UC3: IPVC configuration

There are a little bit more differences in IP UNI product. As highlighted in Figure 45:

- the egressBandwidthProfileEnvelope - is now present,
- ingressBandwidthProfileEnvelope - is not simplified as in Basic and now can define flow with regards to End Point, Access Link or class of service name
- routingProtocols - is now present (for Basic it must be empty)

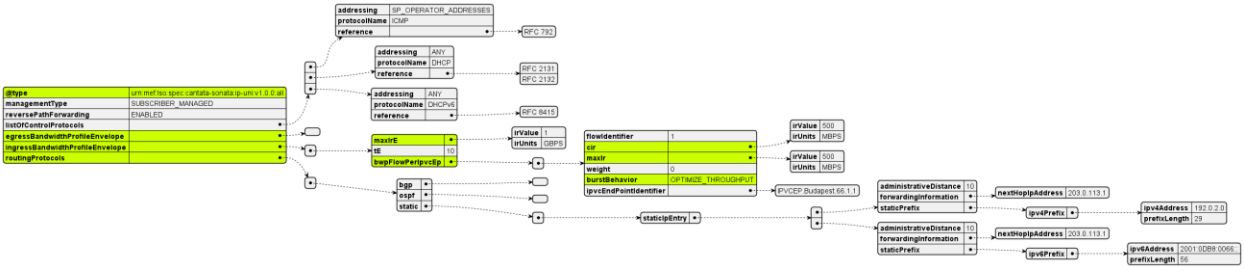


Figure 45 UC3: IP UNI configuration

Figure 46 presents the configuration of an IP UNI Access Link product and highlights discrepancies comparing to the Basic Internet Access version. These are:

- @type is present pointing to IpUniAccessLink product schema
- dhcpRelay, bfd, egressBwpEnvelope, ingressBwpEnvelope are present, yet empty in the example configuration.
- l2Technology.demux provides vlanId, so that more than one IP UNI Access Link can be carried on the trunk.
- ipv4ConnectionAddressing.ipv4AddressingType - now available for definition. For Basic it was assumed to be DHCP.
- ipv4ConnectionAddressing.ipv4AddressingType - set to ask for given prefix length (number of possible IP Addresses)

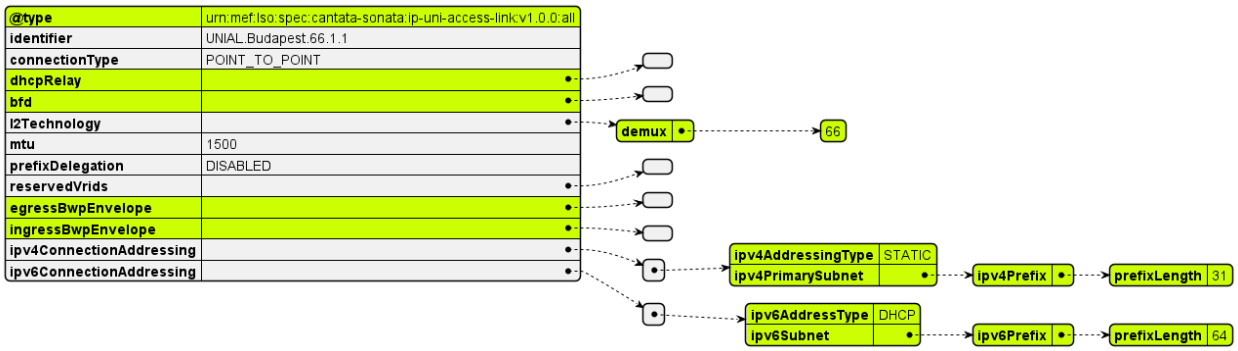


Figure 46 UC3: IP UNI Access Link configuration

As shown in Figure 47 the configuration of Ethernet UNI Access Link Trunk is the same as in the Basic use case. The only difference is the presence of the @type.

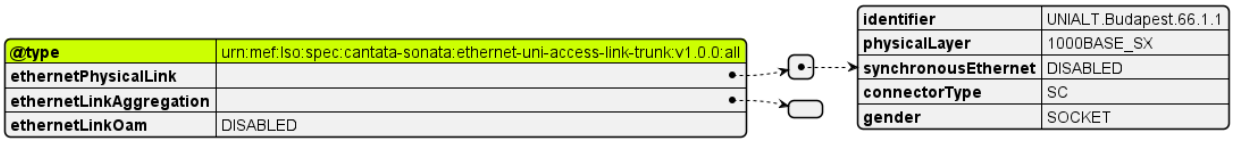


Figure 47 UC3: IP UNI Access Link Trunk configuration

Figure 48 recaps the relations' names and cardinalities. It will help to understand more complex scenario presented in Figure 49

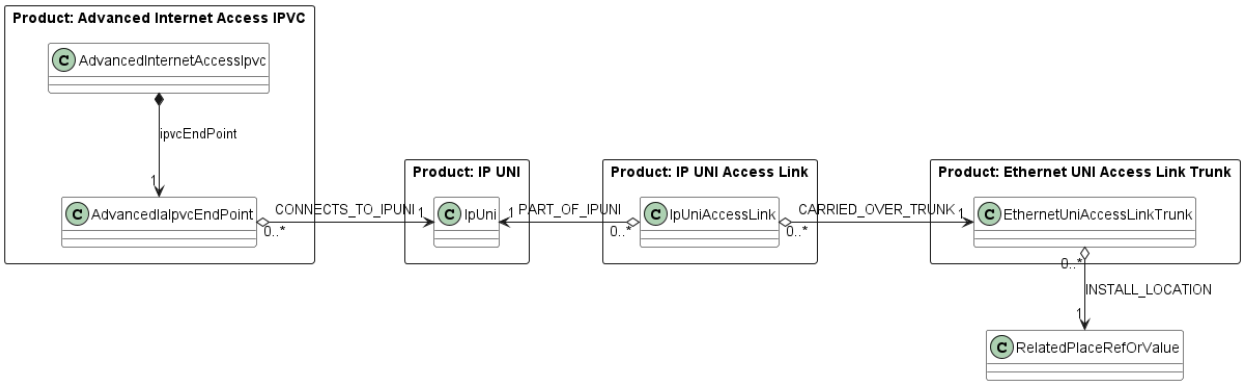


Figure 48 Information model for Advanced Internet Access product

An example topology using these cardinalities in presented on Figure 49. Here the Internet Access IPVC has an End Point that connects to IP UNI 2. This IP UNI 2 consists of two IP UNI Access Links. One of them is provided by Ethernet UNI Access Link Trunk 1 with use of VLAN ID = 66 and the other one is exclusively provided by Ethernet UNI Access Link 2 that is using LAG.

There is also an IP VPN Product that has 2 End Points, one per UNI. First End Point connects to UNI 1 that consist only of one IP UNI Access Link that is provided by Ethernet UNI Access Link 1 with use of VLAN ID = 55. The second End Point of the IP VPN connects to same UNI 2 as the one of IPVC.

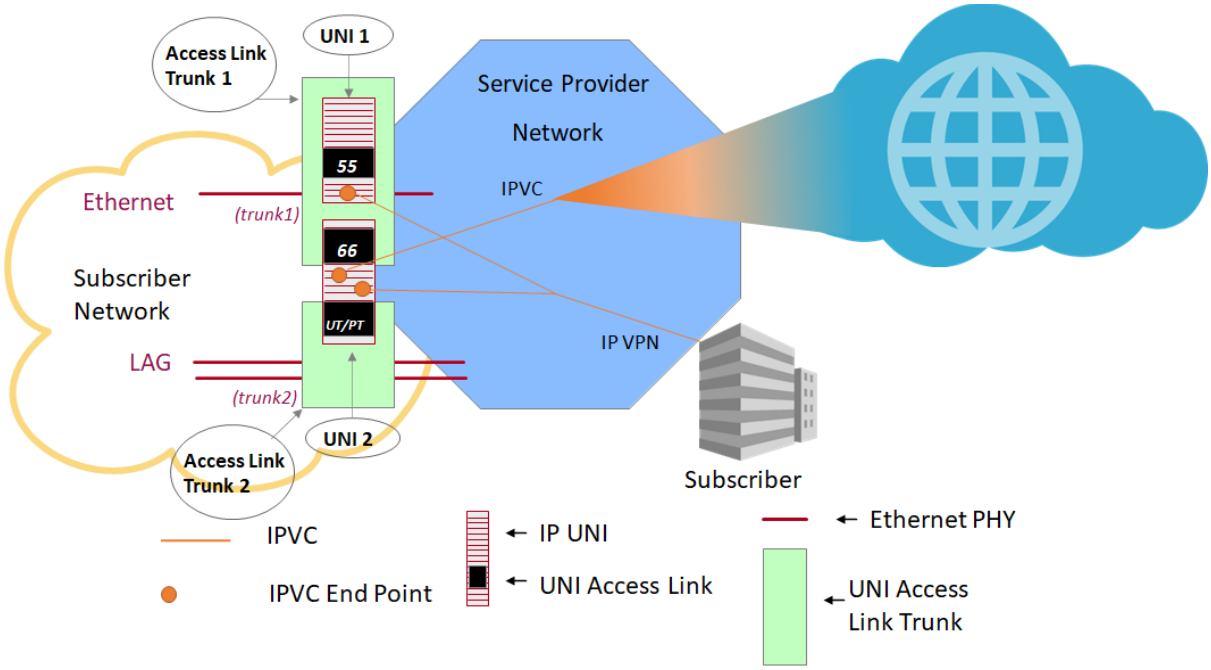


Figure 49 Complex topology example of Advanced Internet Access

A.3.4 Use Case 4: POQ - Exclusive Advanced Internet Access

The Exclusive version of the Advanced Internet Access aims to cover the presumed most common use case when the IP UNI and the IP UNI Access Link are used exclusively by one IPVC. This allows to aggregate IPVC, IP UNI and IP UNI Access Links thus reducing the number of items needed to be ordered.

A topology diagram is presented in Figure 50. Three components are covered by common rectangle (IPVC, IP UNI, and IP UNI Access Link) and ordered as one product. Ethernet UNI Access Link is covered by separate rectangle to underline that it is ordered separately and can be shared by multiple IP Uni Access Links being part of Exclusive Internet Access or an Advanced Internet Access products.

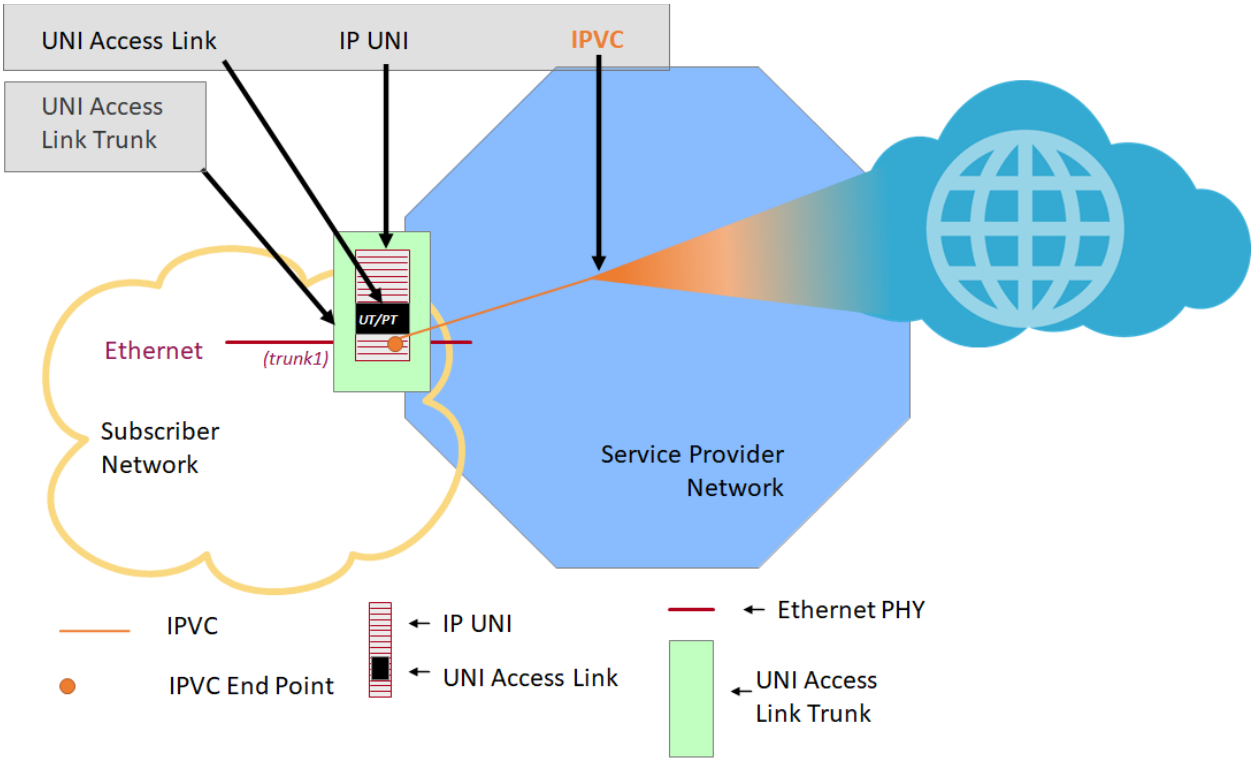


Figure 50 Exclusive Advanced Internet Access topology

Figure 51 shows the structure of the POQ request for creation of Exclusive Advanced Internet Access products. Note that there are 2 items. The Exclusive Advanced Internet Access Product points to Ethernet UNI Access Link Trunk with a relation "CARRIED_OVER_TRUNK" and the Ethernet UNI Access Link Trunk with a relation points to an "INSTALL_LOCATION".

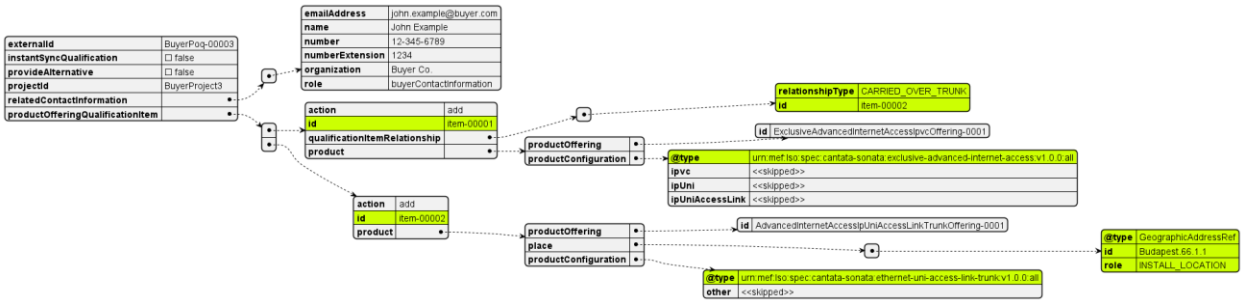


Figure 51 UC4: POQ Request, product-agnostic part

The configuration of the components is identical to one in Advanced Internet Access product (despite the lack of @type) so it will not be discussed further here.

A.3.5 Use Case 5: Quote - Basic Internet Access

For detailed guidance on how to use the Quote Management API, please refer to MEF 115 [25]

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

This use case is the next step after use case 2. It asks for a quotation of the installation of the Basic Internet Access product, with configuration that was previously checked for availability.

The Quote API carries product information exactly the same way as the POQ in terms of building the request of items, referencing other product, referencing locations, and attaching the product information. The "product" part will be the same as in POQ and will not be discussed further in this chapter.

Figure 52 presents a diagram of a Quote request for creation of Basic Internet Access, with product information skipped.

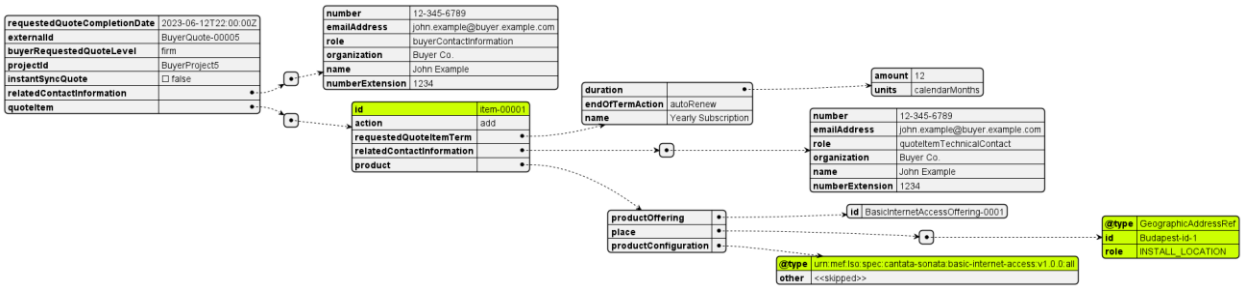


Figure 52 UC5: Quote Request, product-agnostic part

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

- instantSyncQuote - to state the preference of receiving an instant (synchronous) response or a deferred (asynchronous) one. In the latter case, the Seller only sends back an acknowledge response and proceeds with the quotation. The Buyer may choose to register for notification or perform a periodical poll.
- requestedQuoteCompletionDate - If an instant response is not required this specifies the requested response time.
- buyerRequestedQuoteLevel - 3 different types of quotes are managed:
    - **Budgetary**: A Quote that is provided quickly and with very little analysis such that the Buyer can get an idea of how much the requested Product Offering could cost. Any charges specified are subject to change.
    - **Firm - Subject to Feasibility Check**: A Quote that is provided to the Buyer based on some, but not a complete, pre-order analysis. At this stage, the Seller may not be willing to perform any further work on the Quote and requests that the Buyer use the Firm - Subject to Feasibility Check Quote to proceed to the Order process. Ordering is possible based on the Firm - Subject to Feasibility Check Quote with some stipulations as to how cost identified during delivery is addressed. The Monthly Recurring Charges specified in the Quote Response are final. Non-Recurring Charges specified in the Quote Response are subject

to change and new Non-Recurring Charges may be identified during fulfillment.

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

- requestedQuoteItemTerm - to specify the term (also known as commitment)

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

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

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

A.3.6 Use Case 6: Product Order - Basic Internet Access

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

This use case is the next step after use case 5. It places a Product Order for the installation of the Basic Internet Access product, which was qualified and quoted in use cases 2 and 5.

The Order API carries product information exactly the same way as POQ and Quote in terms of building the request of items, referencing other product, referencing locations, and attaching the product information. The “product” part will be the same as in Quote and will not be discussed further in this chapter.

An example Product Order request can be found in the postman collection. Figure 53 presents it with product information skipped for readability.

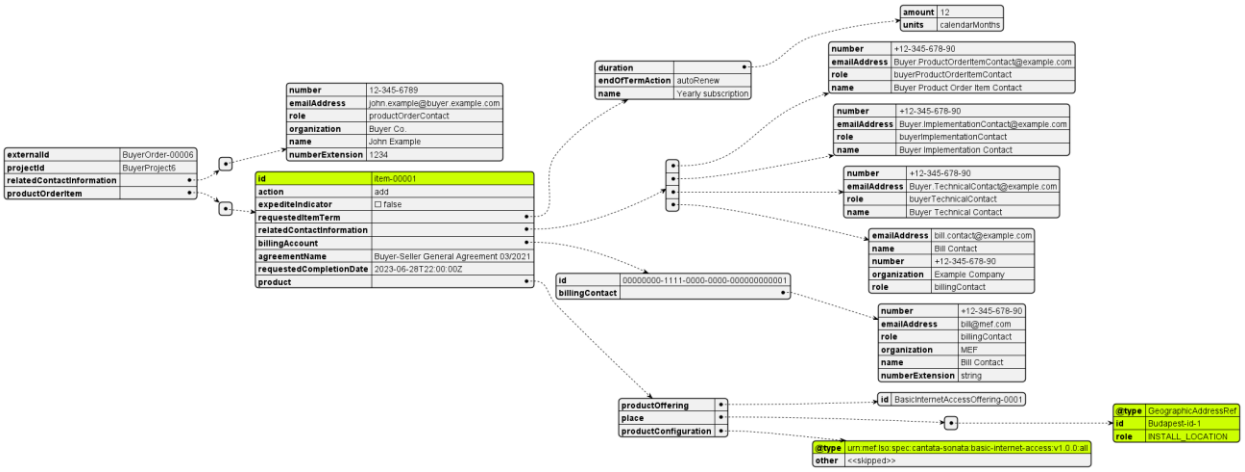


Figure 53 UC6: Product Order Request, product-agnostic part

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

Product Order API offers much more use cases like updating, expediting, or canceling an order request and additional charge negotiation. For detailed guidance on how to use the Product Order Management API, please refer to MEF 123 [28].

A.4 action: modify

The mechanism of building a modification request for both product-independent and product-specific parts for all steps are practically the same as for the create request.

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

- "item.action" must be set to "modify"
- "item.product.id" of the product to be modified must be provided
- "product.productConfiguration" must contain all desired configuration (not only the modified values)
- "product.productOffering" must not be changed
- The "place" and "productRelationship" lists must comply to Product Specification requirements with regards to possibility of modification. In most cases it's prohibited.

A.4.1 Use Case 7: POQ - Advanced Internet Access: Bandwidth change

This use case presents POQ for an Advanced Internet Access product instance bandwidth change. The assumption is that the change is not significant and can be provided only with an update of

configuration without a need of any installation of new equipment (the "1000BASE_SX" interface is used).

This use cases is "applied" to configuration from Use case 2 to an IP UNI product instance with id=SellerIpUniId-0001. There the POQ request had 4 items to create all four components of Advanced Internet Access. When the modification is to be applied only to one of them - only single POQ item is required in the POQ request. The IP UNI product did not define any place or product relations (it is the IPVC and IP UNI Access Link that define relations towards IP UNI) so they are also not provided in this request

Figure 54 shows the structure of the POQ product-agnostic part.

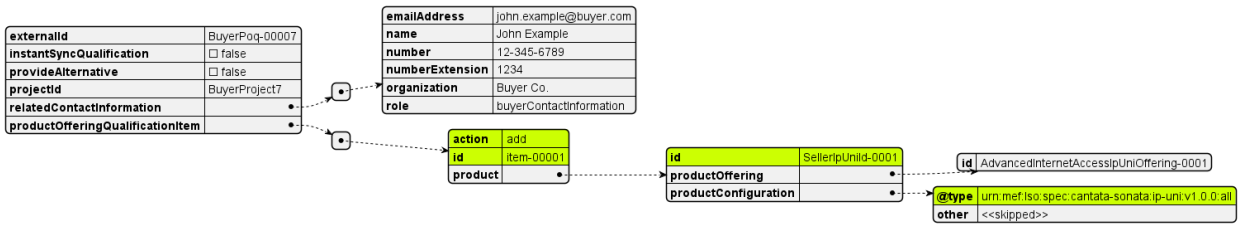


Figure 54 UC7: POQ Request, product-agnostic part

Figure 55 shows the configuration of the IP UNI, with highlighted attributes that are to be modified.

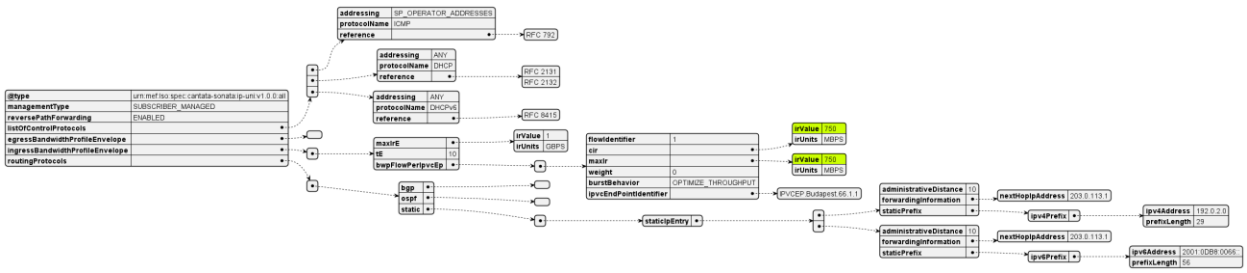


Figure 55 UC7: IP UNI configuration

The cir and maxIr attributes change from 500 to 750 MBPS. Note that the new value is still lower than ingressBandwidthProfileEnvelope.maxIrE which sets the limit for the sum of all flow at this IP UNI.

A.5 action: delete

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

Following common rules apply for delete operation:

- "item.action" must be set to "delete"
- "item.product.id" of the product to be deleted must be provided
- "product.productConfiguration" must not be provided

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

A.5.1 Use Case 8: Quote - Basic Internet Access - delete

This example attempts to quote a deletion of a Basic Internet Access product instance that was ordered in Use Case 6. Since there was only one Product Order Item, there will also be one Quote item in the deletion request. Figure 56 shows presents a diagram of a full Quote request for deletion:

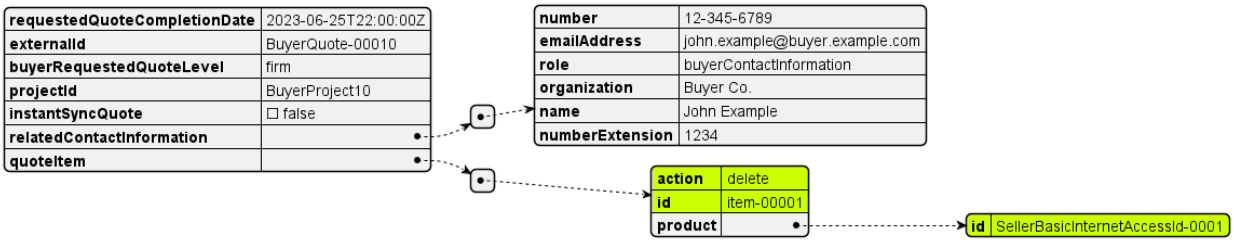


Figure 56 UC8: Quote request

A.5.2 Use Case 9: Product Order - Advanced Internet Access - delete IPVC and End Points only

In Advanced Internet Access case each product can be managed separately and be potentially used by many other products. This use case shows how to delete an IPVC, leaving all other product available for further reuse. Note that the IP UNI product carries IPVC End Point related configuration of bandwidth profiles in ingressBandwidthProfileEnvelope.bwpFlowPerIpvcEp. The relevant entry (the only one in this example) needs to be deleted - this requires a modify action on IP UNI product. The structure of the Product Order request is presented in Figure 57:

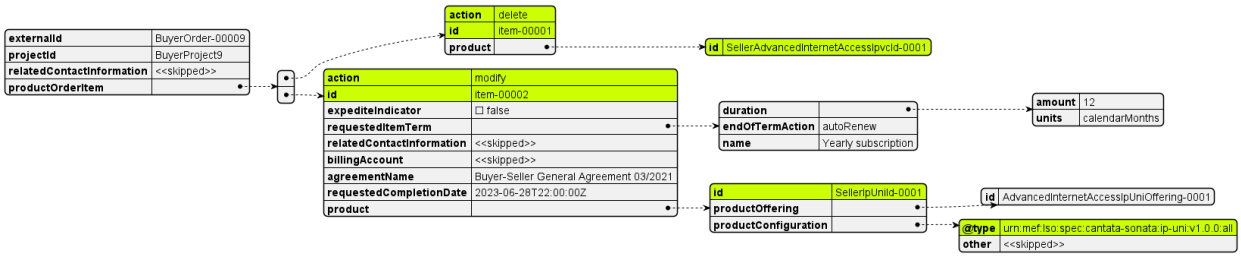


Figure 57 UC9: Product Order, product-agnostic part

And the configuration of IP UNI in Figure 58.

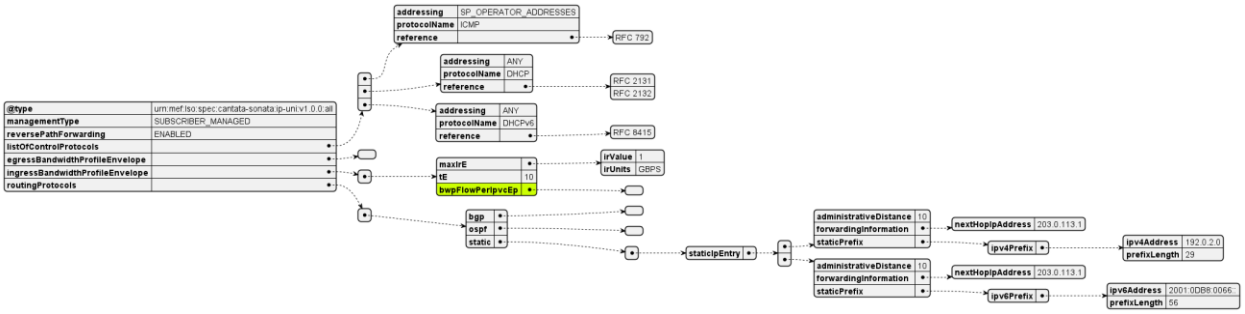


Figure 58 UC9: IP UNI configuration

A.5.3 Use Case 10: Product Order - Exclusive Advanced Internet Access - delete all of items at once

The last use case presents a deletion of all components of Exclusive Advanced Internet Access product. This includes 2 items - Exclusive Advanced Internet Access and the Advanced Internet Access Ip Uni Access Link Trunk. Figure 59 presents the full Product Order request:

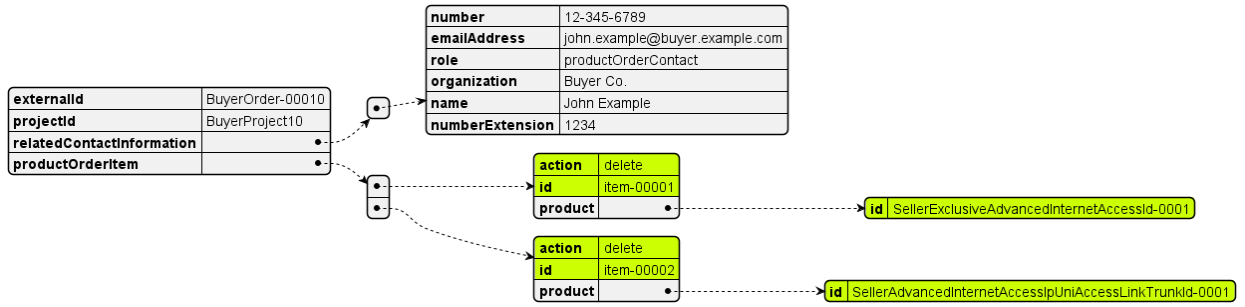


Figure 59 UC10: Product Order request

Appendix B Acknowledgements

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

- David **BALL**
- Mike **BENCHECK**
- Michał **ŁĄCZYŃSKI**
- Jack **PUGACZEWSKI**
- Fahim **SABIR**
- Larry **SAMBERG**