MEF 89

Resource Model - Common

January 2022

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 Standard. MEF is not responsible or liable for any modifications to this Standard made by any other party.

The receipt or any use of this Standard 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 Standard.

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 Standard and have requested to be included in this list.

- Fujitsu
- Kratos Defense
- NEC
- Nokia

2 Abstract

This Standard describes the MEF Resource Model (MRM) Common, which essentially is the selection of ONF Transport API Information Model (TAPI IM [19], [21])

- currently used and extended by more specific MEF Resource Models like Ethernet Connectivity Info Model (MEF 59 [12]), Subscriber and Operator Layer 1 Info Model (MEF 72.1 [15]), Ethernet OAM Info Model (MEF 83 [16]),
- potentially useful for the Resource management at PRESTO IRP.

Lifecycle Service Orchestration Reference Architecture (LSO RA, MEF 55.1 [11]) extends the traditional MEF scope concerning Service Modeling, from a pure view "from outside the network" to cover a range of Operational, Orchestration, and Network Management behaviors, including SDN and NFV paradigms.

The MRM Common and its specific models (MEF 59, MEF 72.1, MEF 83) have been defined to manage the Network Infrastructure, through SDN Controllers, WAN Controllers, OTN Subnetwork Managers, and other legacy Network Management Systems. Considering LSO architecture, these management models are applicable to PRESTO Interface Reference Point.

The MRM structure is based on current and developing best network management solutions by ITU-T, ONF, TM Forum, to allow wider and future proof interoperability across multi-vendor and multi-technology networks. Examples of reference network management solutions are ITU-T G.7711/Y.1702 [5], ONF TR-512 [18], ONF TR-527 [19], TM Forum MTNM [24] and MTOSI [25].

This Standard normatively includes the content of the following Papyrus [22] UML and YANG files as if they were contained within this Standard (GitHub Repository [17]):

- TapiCommon.di / .notation / .uml / .yang / .tree
- TapiTopology.di / .notation / .uml / .yang / .tree
- TapiConnectivity.di / .notation / .uml / .yang / .tree
- TapiPathComputation.di / .notation / .uml / .yang / .tree
- TapiFm.di / .notation / .uml / .yang / .tree
- TapiOam.di / .notation / .uml / .yang / .tree
- TapiNotification.di / .notation / .uml / .yang / .tree

3 Terminology and Abbreviations

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

Term	Definition	Reference
Connection	A *Connection* represents an enabled (provisioned) potential for forwarding (including all circuit/packet forms) between two or more *Node Edge Points* of a *Node*. The bounding *Node* of a *Connection* may be explicit or be conceptually implicit. *Connection* is a container for provisioned connectivity that tracks the state of the allocated resources and is distinct from the *Connectivity Service* request.	ONF TR-527 [19]
CEP	*Connection End Point*	ONF TR-527 [19]
Connection End Point	The *Connection End Point* encapsulates information related to a *Connection* at the ingress/egress points of every *Node* that the *Connection* traverses in a *Topology*. Thus they represent the ingress/egress port functions (including termination, encapsulation, processing, mapping, etc) of the *Connection*.	ONF TR-527 [19]
Connectivity Service	A *Connectivity Service* represents an "intent-like" request for connectivity between two or more *Service Interface Points*. *Connectivity Service* is a container for connectivity request details and is distinct from *Connection* that realizes the request.	ONF TR-527 [19]
Connectivity Service End Point	The *Connectivity Service End Point* encapsulates information related to a *Connectivity Service* at the ingress/egress points of that *Connectivity Service.*	ONF TR-527 [19]
Context	The *Context* provides a scope of control, naming and information exchange between particular instances of SOF & ICM.	ONF TR-527 [19]
CS	*Connectivity Service*	ONF TR-527 [19]
CSEP	*Connectivity Service End Point*	ONF TR-527 [19]
DSR	Digital Signal Rate	MEF 72.1 [15]
Digital Signal Rate	Generic term indicating Layer 1 coding functions. Examples are 10GBASE-W, 100GBASE-R, FC-400, FC-800, STM-16, OC-192, etc. as defined by MEF 63 [13].	MEF 72.1 [15]
ICM	Infrastructure Control and Management	MEF 55.1 [11]

Term	Definition	Reference
Infrastructure Control and Management	The set of functionality providing domain specific network and topology view resource management capabilities including configuration, control and supervision of the network infrastructure.	MEF 55.1 [11]
INNI	Internal Network-to-Network Interface	MEF 4 [7] MEF 55.1 [11]
Internal Network-to-Network Interface	A reference point representing the boundary between two networks or network elements that are operated within the same administrative domain. Note: In this specification, the "networks or network elements" refers to those in a given ICM Domain, hence, between two ICM domains.	MEF 4 [7] MEF 55.1 [11]
Link	A *Link* is an abstract representation of the effective adjacency between two or more *Nodes* (specifically *Node Edge Points*) in a *Topology*.	ONF TR-527 [19]
LSO	Lifecycle Service Orchestration	MEF 55.1 [11]
LSO RA	Lifecycle Service Orchestration Reference Architecture	MEF 55.1 [11]
Lifecycle Service Orchestration	Open and interoperable automation of management operations over the entire lifecycle of Services. This includes fulfillment, control, performance, assurance, usage, security, analytics and policy capabilities, over all the network domains that require coordinated management and control in order to deliver the Service.	MEF 55.1 [11]
Lifecycle Service Orchestration Reference Architecture	A layered abstraction architecture that characterizes the management and control domains and entities, and the interfaces among them, to enable cooperative orchestration of Connectivity Services.	MEF 55.1 [11]
MRM	MEF Resource Model	This Standard
MEG	Maintenance Entity Group	ITU-T G.8001 [6]
MEP	Maintenance Entity Group End Point	ITU-T G.8001 [6]
MIP	Maintenance Entity Group Intermediate Point	ITU-T G.8001 [6]
NEP	*Node Edge Point*	ONF TR-527 [19]
Node	The *Node* is an abstract representation of the forwarding capabilities of a particular set of network resources. It is described in terms of the aggregation of set of ports (*Node Edge Point*) belonging to those network resources and the potential to enable forwarding of information between those edge ports.	ONF TR-527 [19]

Term	Definition	Reference
Node Edge Point	The *Node Edge Point* represents the ingress-egress edge-port functions that access the forwarding capabilities provided by the *Node*. Hence it provides an encapsulation of addressing, mapping, termination, adaptation and OAM functions of one or more transport layers (including circuit and packet forms) performed at the entry and exit points of the *Node*.	ONF TR-527 [19]
ODU	Optical Data Unit	ITU-T G.709 [4]
OTN	Optical Transport Network	ITU-T G.709 [4]
Product Instance	Specific implementation of a Product Offering dedicated to the benefit of a party.	TMF GB922 [23]
Product Offering	An externally facing representation of a Service and/or Resource procurable by the Customer.	TMF GB922 [23]
Product Specification	The detailed description of product characteristics and behavior used in the definition of Product Offerings.	TMF GB922 [23]
Resource	A physical or non-physical component (or some combination of these) within a Service Provider's infrastructure or inventory.	TMF GB922 [23]
Service	Represents the Customer experience of a Product Instance that has been realized within the Service Provider's and / or Partners' infrastructure.	TMF GB922 [23]
Service Component	A segment or element of a Service that is managed independently by the Service Provider.	MEF 55.1 [11]
Service Interface Point	A *Service Interface Point* represents the network-interface-facing aspects of the edge-port functions that access the forwarding capabilities provided by the *Node*. Hence it provides a limited, simplified view of interest to external clients (e.g. shared addressing, capacity, re-source availability, etc.), that enable the clients to request connectivity without the need to understand the provider network internals.	ONF TR-527 [19]
Service Orchestration Functionality	The set of service management layer functionality supporting an agile framework to streamline and automate the service lifecycle in a sustainable fashion for coordinated management supporting design, fulfillment, control, testing, problem management, quality management, usage measurements, security management, analytics, and policy-based management capabilities providing coordinated end-to-end management and control of Services.	MEF 55.1 [11]
SIP	*Service Interface Point*	ONF TR-527 [19]

Term	Definition	Reference
SOF	Service Orchestration Functionality	MEF 55.1 [11]
Route	The *Route* of a *Connection* is modeled as a collection of *Connection End Points*.	This Standard
TAPI or T-API	Transport API Information Model	ONF TR-527 [19] ONF TAPI IM [21]
Topology	The *Topology* is an abstract representation of the topological aspects of a particular set of network resources. It is described in terms of the underlying topological network of *Nodes* and *Links* that enable the forwarding capabilities of that particular set of network resources.	ONF TR-527 [19]
Transitional Link	A *Link* that is formed by abstracting one or more termination and adaptation functions to focus on the flow and deemphasize the protocol transformation. This abstraction is relevant when considering multi-layer routing.	ONF TR-527 [19]
UML	Unified Modeling Language	OMG UML, Infrastructure, Version 2.5
UUID	An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity.	This Standard

Table 1 – Terminology and Abbreviations

4 Compliance Levels

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this Standard are to be interpreted as described in BCP 14 (RFC 2119 [1], RFC 8174 [2]) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

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

5 Introduction

5.1 The MEF Common Resource Model

The development of MEF Resource Models, applicable to PRESTO IRP, needs several common resource management items, both technology agnostic and technology specific. Examples of technology agnostic constructs are:

- *Node*, *Node Edge Point, Link* for the topology model,
- *Connection, Connection End Point, Route, Path* for the forwarding model,
- *Meg, Mep*, *Mip* for the OAM model.

To accelerate the development of MEF Resource Models, a selection of the ONF Transport API Information Model (TAPI IM [19], [21]) model has been reused by:

- MEF 59 [12], which defines Carrier Ethernet Connectivity resource management features, identified as the set of management features supported by MEF 7.4 [8] at service level.
- MEF 72.1 [15], which defines Subscriber and Operator Layer 1 Connectivity resource management features, according to the requirements defined by MEF 63 [13] and MEF 64 [14].
- MEF 83 [16], which defines OAM management features for Carrier Ethernet Connectivity, according to the requirements defined by MEF 30.1 [9] and MEF 35.1 [10].

The scope of this Standard is the specification of the ONF TAPI technology agnostic model items which are

- currently used by the above listed MEF Standard Resource Models,
- potentially useful for the Resource management at PRESTO IRP.

This selection of ONF TAPI forms the MEF Resource Model - Common. Figure 1 depicts the federation of ONF and MEF information models. The MRM – Common (subset of TAPI technology agnostic model) is augmented by TAPI and MEF technology specific models. See Section 5.3 for extension, or augmentation, mechanism.

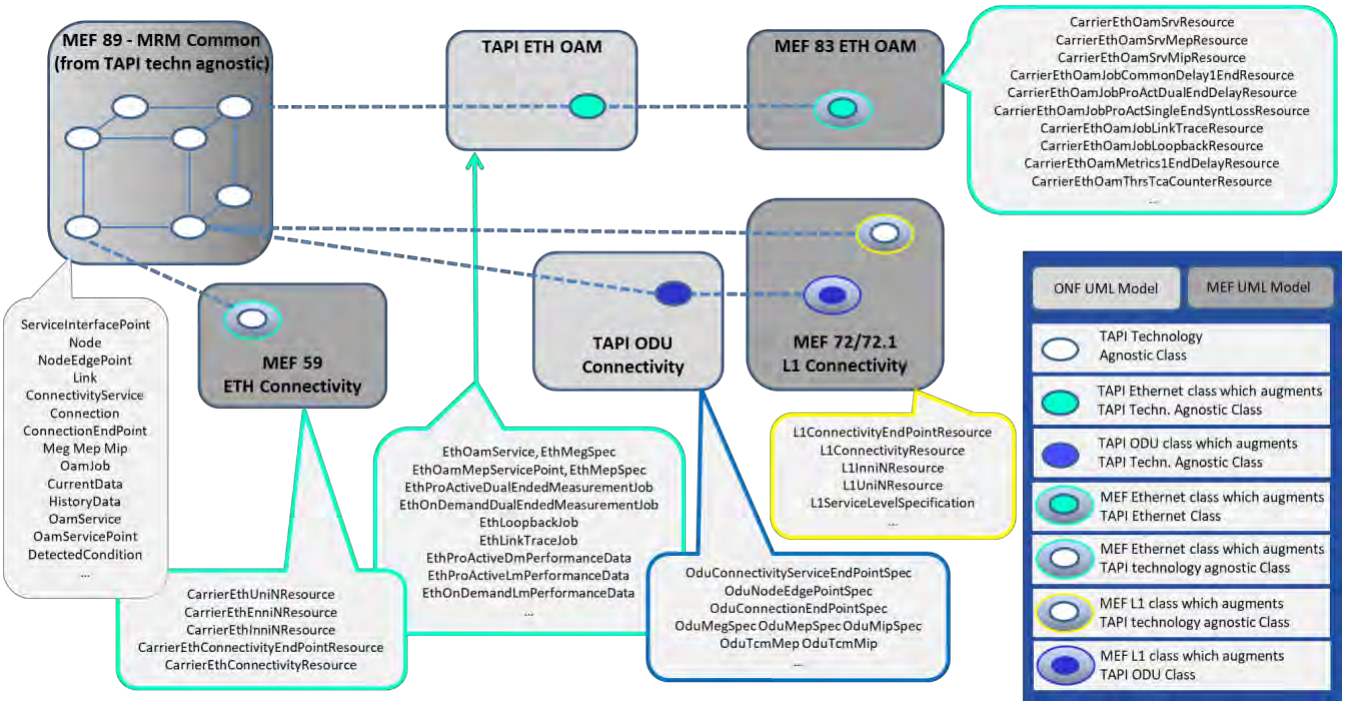

Figure 1 – MEF MRM Common augmented by technology specific models

Figure 1 shows that MEF 83 augments TAPI Ethernet specific model, while MEF 59 does not. This asymmetry is mainly due to the ONF TAPI different state-of-art at the time of MEF 59 development. MEF 83 has leveraged a more mature TAPI Ethernet model, and conversely the review process of MEF 83 included comments on TAPI OAM/Ethernet model, which was then accordingly enhanced in ONF, eventually allowing MEF approved and consistent reuse.

Note that MEF 72.1 augments TAPI OTN model for the management of Operator L1 Services, which include OTN E-NNI.

The MRM Common classes and their extensions are applicable to PRESTO Interface Reference Point (MEF 55.1 [11]), see Figure 2.

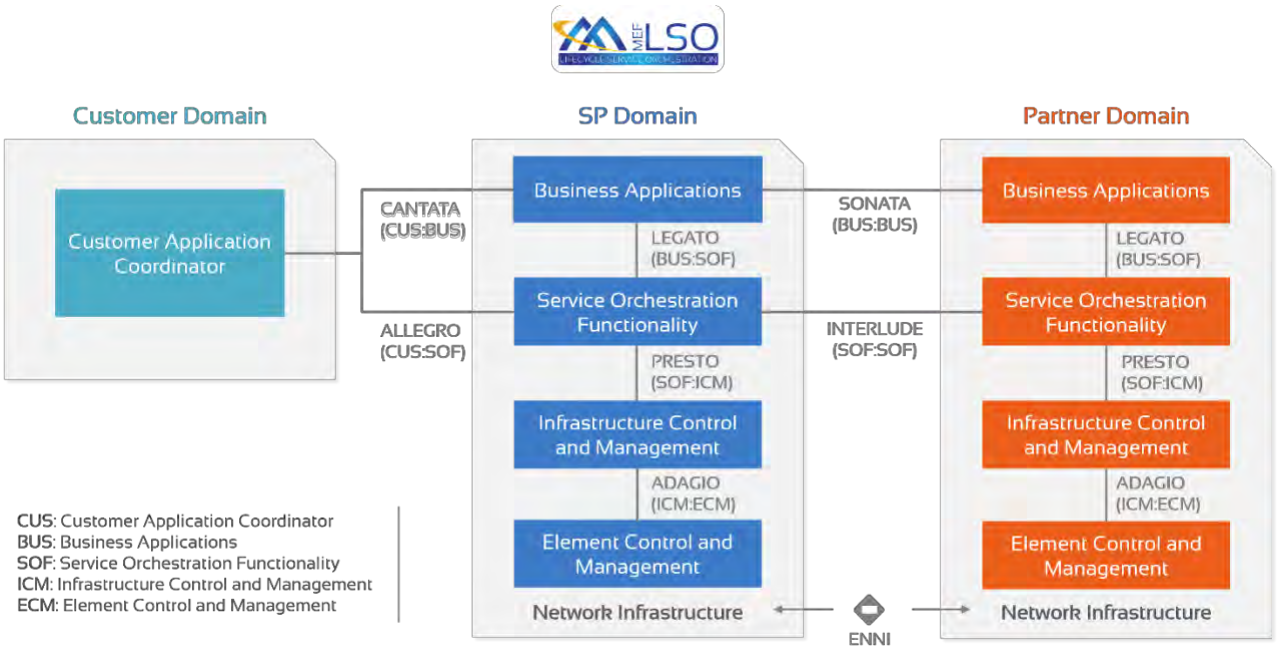

Figure 2 – MEF Common Resource Model positioning in LSO RA

The following sections of this Standard include:

- TAPI overview (Section 5.2)
- Augmentation overview (Section 5.3)
- Operations API (RPC) vs Data API (Section 5.4)
- MRM - Common descriptions and examples (Section 6)
- The list of all defined object classes, their attributes and data types, associations, and abstractions (augmentations) for:
  - Common Model (Section 7)
  - Topology Model (Section 8)
  - Connectivity Model (Section 9)
  - Path Computation Model (Section 10)
  - Fault Management Model (Section 11)
  - OAM Model (Section 12)
  - Notification Model (Section 13)
- References (Section 14)

5.2 ONF TAPI General Overview

TAPI is defined to be applicable on the interface between a Transport SDN controller "Black Box" and its client application. The actors involved in the information exchange over this interface include transport network provider domain controllers in the role of producers (e.g., ICM) and the transport network application systems in the role of the consumers (e.g., SOF), see Figure 3.

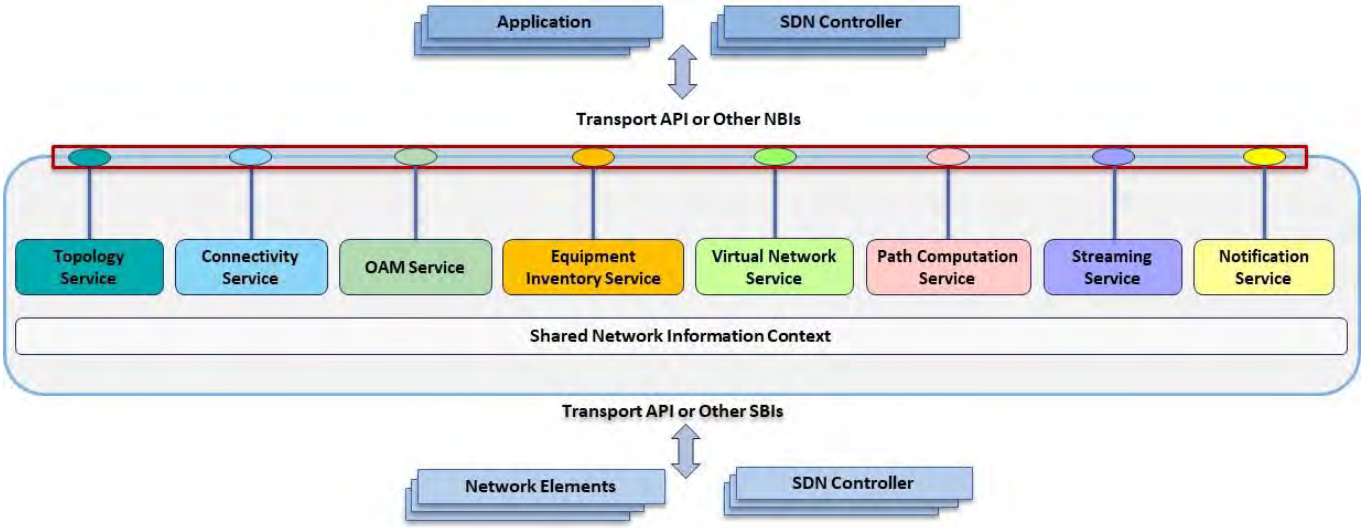

Figure 3 – Transport API Functional Architecture

TAPI model is structured in *service-oriented* classes and *resource-oriented* classes.

- The *service-oriented* classes represent the *intent*, i.e., the *provisioning aspects*. It is the model of the desired result, which allows to specify from minimal (opaque view) to detailed or prescriptive constraints. Examples of *service-oriented* classes are:
  - *Service Interface Point*, *Connectivity Service*, *Connectivity Service End Point*, *Path Computation Service*, *Path Service End Point, Oam Service*, *Oam Service Point.*
- The *resource-oriented* classes represent the *network state*. Examples of *resource-oriented* classes are:
  - *Topology, Node*, *Node Edge Point*, *Link, Connection*, *Connection End Point*, *Route, Path, Meg*, *Mep*, *Mip*.

For example, when the client controller provisions a *Connectivity Service*, the server controller will create one or more *Connections* to represent how the network has implemented the *Connectivity Service*. In other words, the *Connections* represent a view of the network resources allocated to support the requested *Connectivity Service*. This structure applies also to OAM, i.e., an *Oam Service* is the representation of the OAM *intent*, the server controller will create *Meg*, *Meps* and *Mips* representing the OAM network resources allocated to support the requested *Oam Service*.

The *TAPI Context* provides a scope of control, naming and information exchange for particular TAPI provider & client interaction using the Network Topology, Connectivity, OAM, Equipment Inventory, Virtual Network, Path Computation, Streaming and Notification Service APIs.

Offline negotiation and agreement between TAPI provider and its client determine the setup of this shared *TAPI Context* and in turn the type and degree of abstraction that is provided. A *TAPI Context* is distinct from the TAPI provider's or TAPI client's internal context. A TAPI client assumes exclusive control over all information provided to it within its *TAPI Context*.

A *TAPI Context* is defined by a set of *Service Interface Points* and includes one or more top-level *Topologies* that are either statically assigned by the provider or dynamically created by the client.

Figure 4 and Figure 5 show two examples of possible hierarchy of controllers in the LSO RA ICM domain. There are two Administration Contexts (both Blue) and three Client Contexts (Red, Green and Yellow). All UNI interfaces are Ethernet (e.g., 10GE), while INNI interfaces are ODU (e.g., 100G OTN). All UNI devices are ODU plus Ethernet switch capable, while internal devices are only ODU switch capable. In Figure 4 the SOF is administrator of ICM Domains, hence has the holistic view of network resources and manages per customer resource allocation / isolation. Only Admin Contexts are exposed at PRESTO. This allows more sophisticated resource management capability.

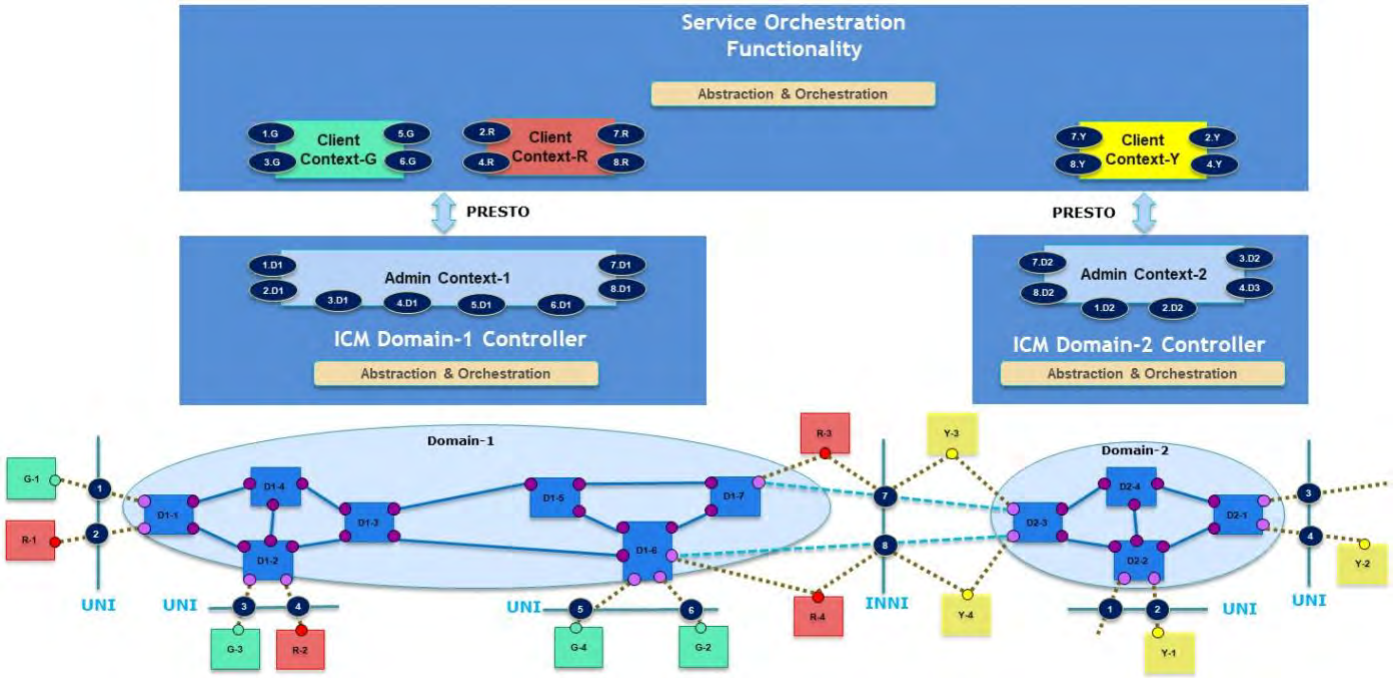

Figure 4 – Reference Hierarchical Control Example 1

In Figure 5 the SOF delegates per customer resource mapping / allocation / isolation to ICM. Besides Admin Context, ICM exposes per Client Contexts (Red, Green, Yellow).

Figure 5 – Reference Hierarchical Control Example 2

Figure 6 depicts the complete topology potentially exposed by ICM Domain-1 Controller to the SOF. In this example, the exposed MRM *Topology Context* (which augments *TAPI Context* for topology model, see Section 6.2 for more details) shows single-layer *Nodes* abstraction of domain-controller's internal context. The Ethernet *Node* D1-1e is linked to OTN *Node* D1-1o by a *Transitional Link*.

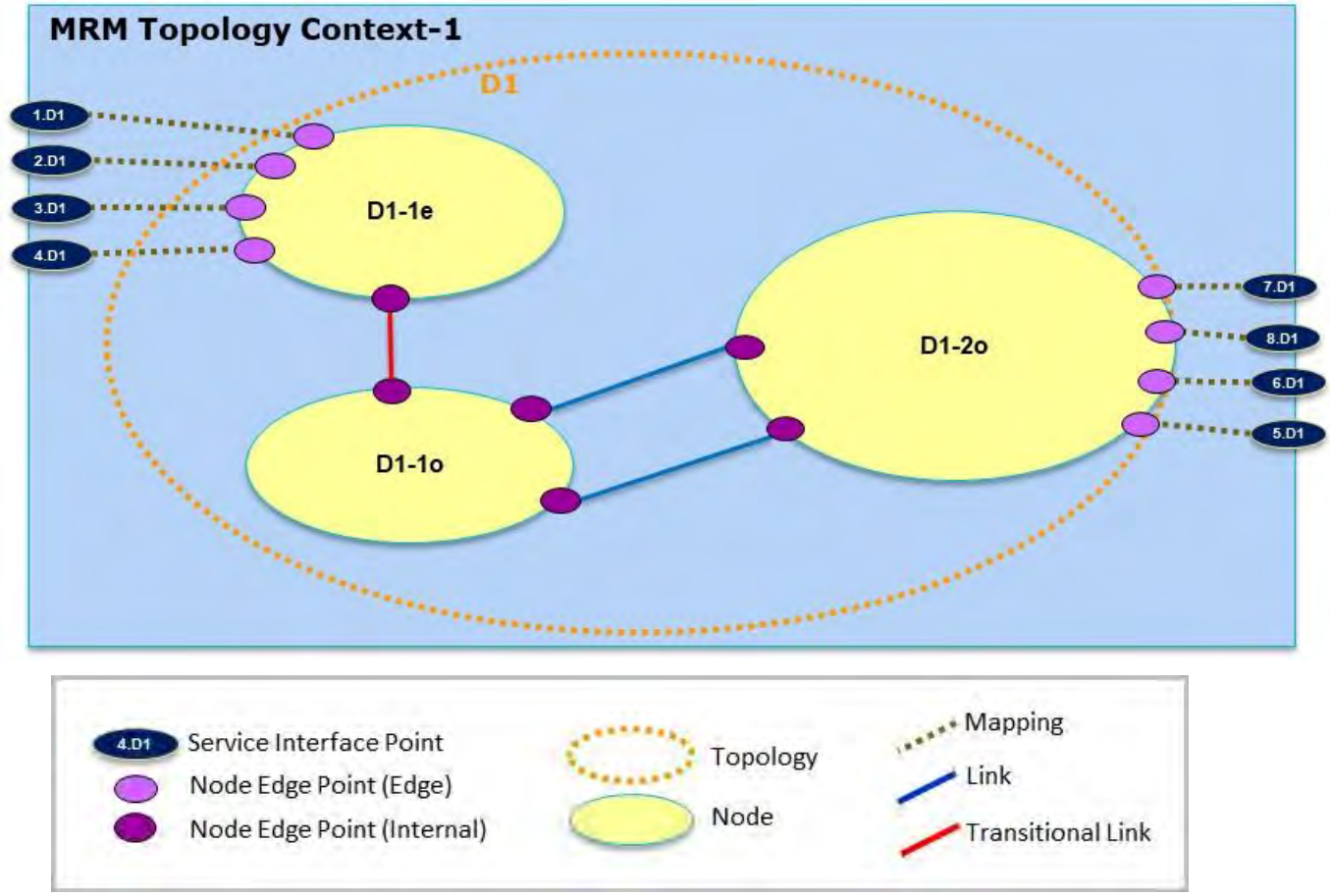

Figure 6 – Domain 1 Context Topology

The *Node* represents the forwarding potential between its *Node Edge Points* (*NEP*). The *Link* associates NEPs of different *Nodes*, *Transitional Link* in case of *Nodes* of different layer networks. A *Link* may offer parameters such as capacity and delay, depending on the type of technology that supports the *Link*.

Figure 7 depicts the complete topology potentially exposed by ICM Domain-1 Controller to the SOF as 2-level hierarchy. This view is constructed by recursively traversing/expanding the internal topology of top-level *Nodes*.

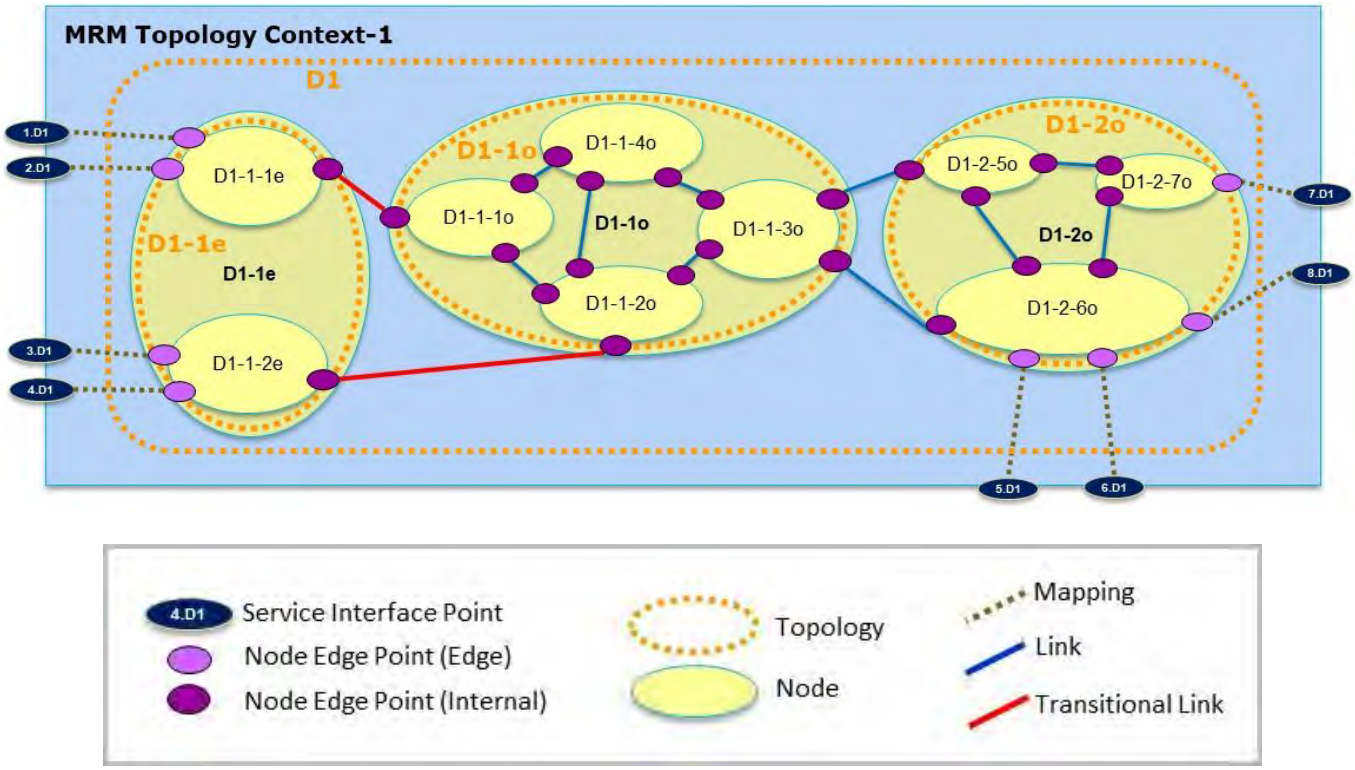

Figure 7 – Domain 1 Context Topology, 2-level hierarchy

The D1 *Topology* encompasses the D1-1e, D1-1o and D1-2o *Nodes*.

The *Nodes* D1-1e, D1-1o and D1-2o each encapsulate an internal *Topology* (with same name, in orange font), which in turn encompasses *Nodes* at lower partitioning level. So, a *Node* at a top-level could abstract the *Topology* of an entire network while a *Node* at the bottom-most level could abstract a switch matrix within a device.

Figure 8 depicts the complete topology potentially exposed by ICM Domain-2 Controller to the SOF. In this example, the exposed MRM *Topology Context* is same as the domain-controller's internal context, which is modeled by multi-layer *Nodes*.

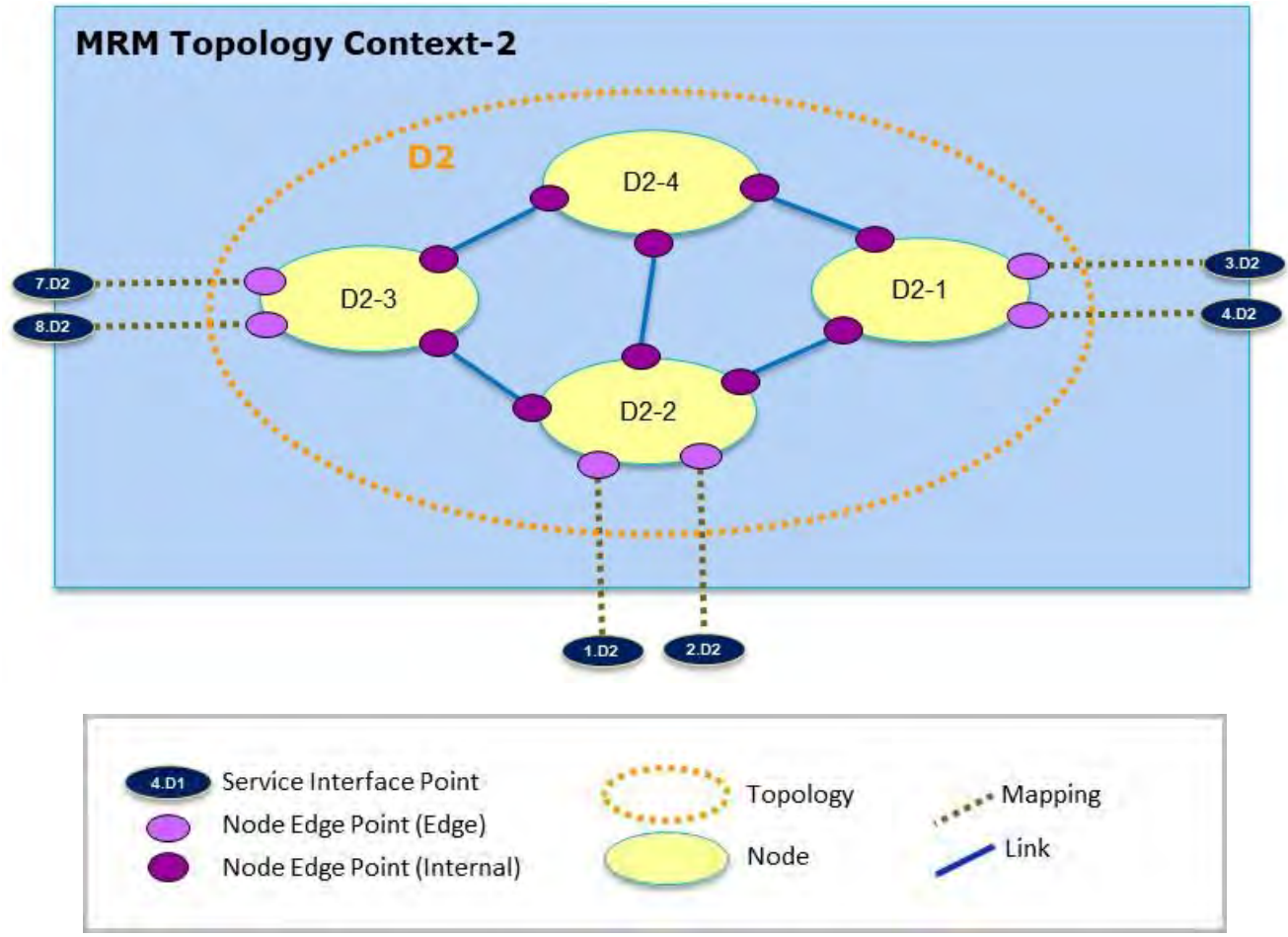

Figure 8 – Domain 2 Context Topology

Multi-layer *Nodes* apply when the *Transitional Link* is not suitable to describe complex transmission functions which e.g., imply multiplexing between layer protocols. In other words, *Transitional Link* is suitable when the focus is on the flow (e.g,. for multi-layer routing) rather than on the protocol transformation.

Figure 9 shows how the Transitional Link splits G.800 *adaptation* and *termination* functions.

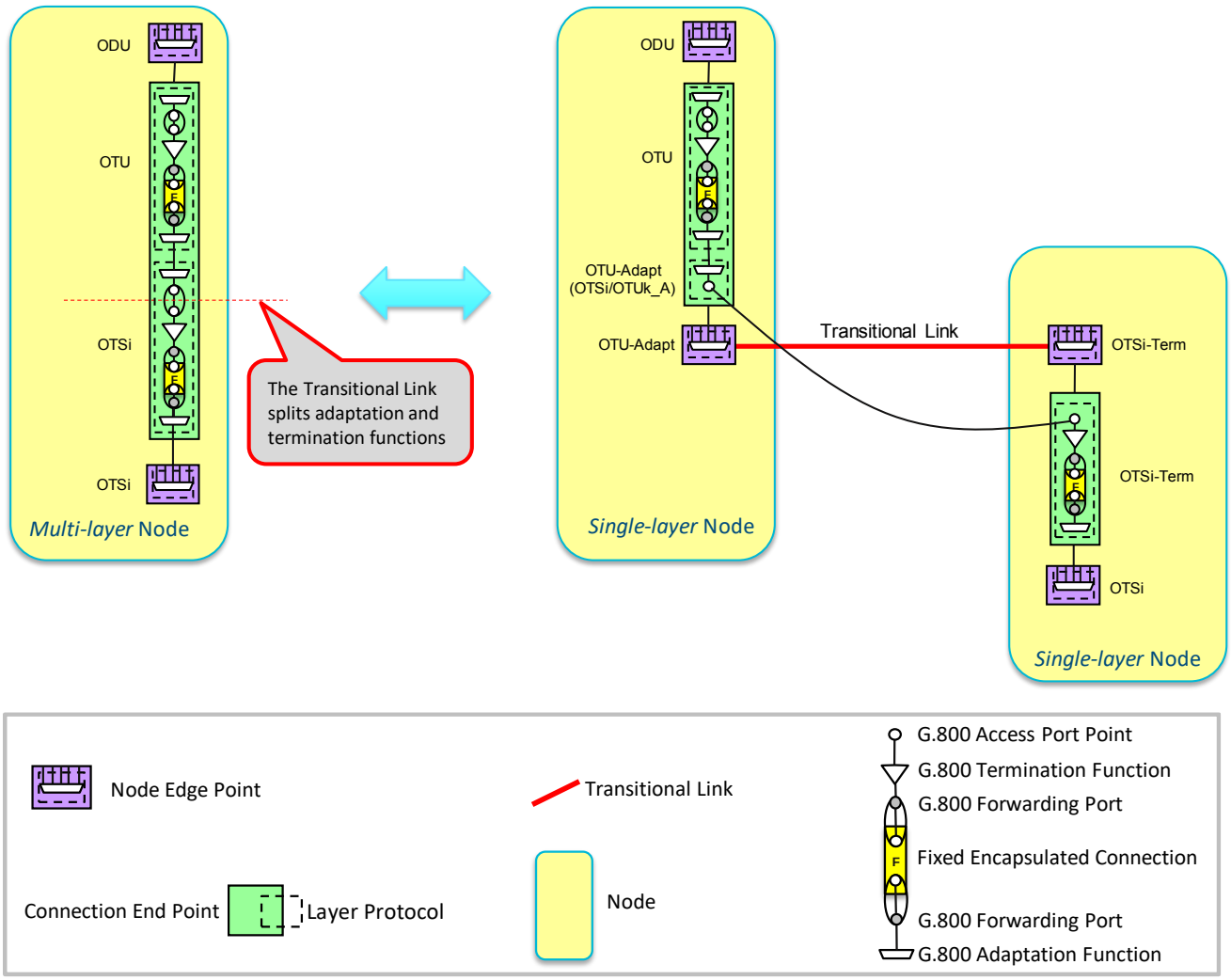

Figure 9 – Transitional Link

Figure 10 depicts the topology of Green Client Context exposed by ICM Domain-1 Controller to the SOF. In this example, the exposed MRM *Topology Context* is a single *Node* abstraction of multi-domain-controller's internal context. The *Node* represents an Ethernet layer *Topology*.

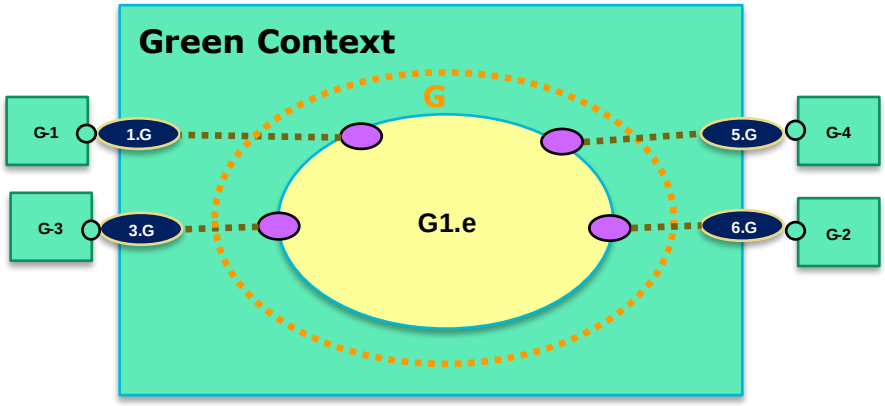

Figure 10 – Green Context Topology

Figure 11 depicts a different example of topology of Green Client Context exposed by ICM Domain-1 Controller to the SOF. The exposed MRM *Topology Context* is an edge-*Node* abstraction of multi-domain-controller's internal context, i.e., only *Nodes* with an edge Ethernet *NEP* are represented, internal OTN topology is hidden. The Ethernet layer *Links* are an abstraction of the potential connectivity at this layer.

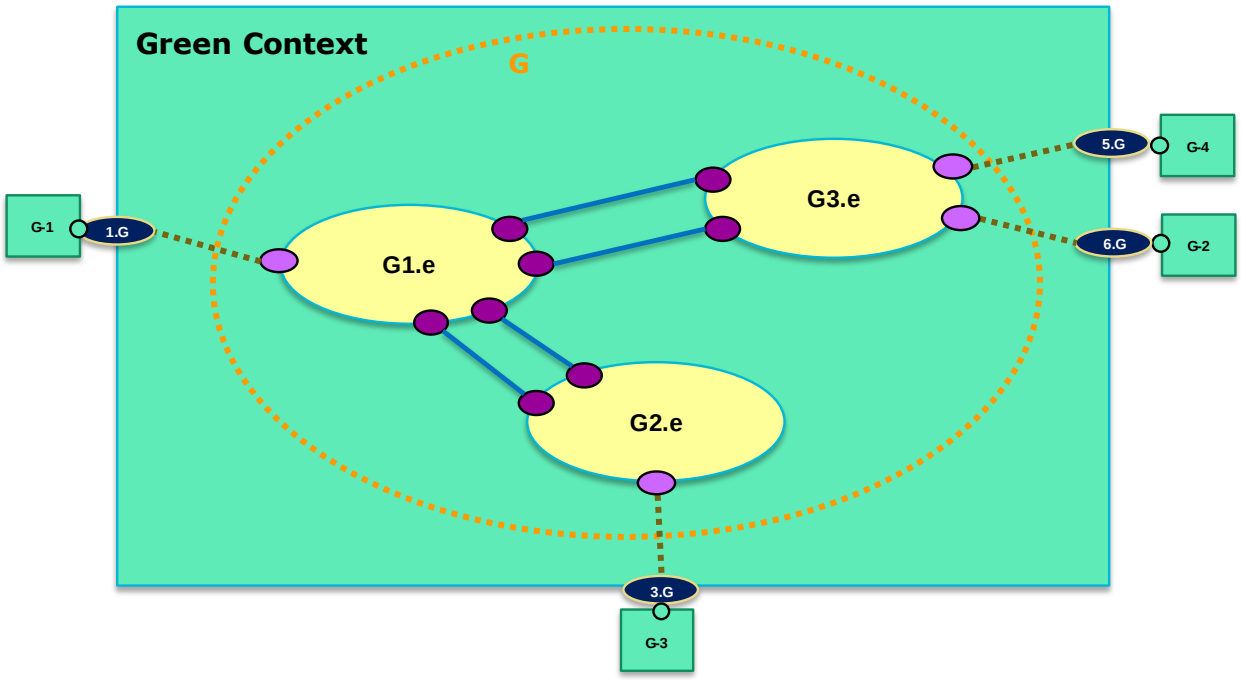

Figure 11 – Green Context Topology, alternative view

Figure 12 depicts the topology of Red Client Context exposed by ICM Domain-1 Controller to the SOF. In this example the exposed MRM *Topology Context* is per-layer-*Node* abstraction of multi-domain-controller's internal context. This is an Ethernet + ODU layer *Topology*, the ODU layer network being abstracted by a single *Node*. There are *Transitional Links* between Ethernet and ODU *Nodes*, but initially no *Links* in the Ethernet layer, as they are the result of underlying OTN *Connections*.

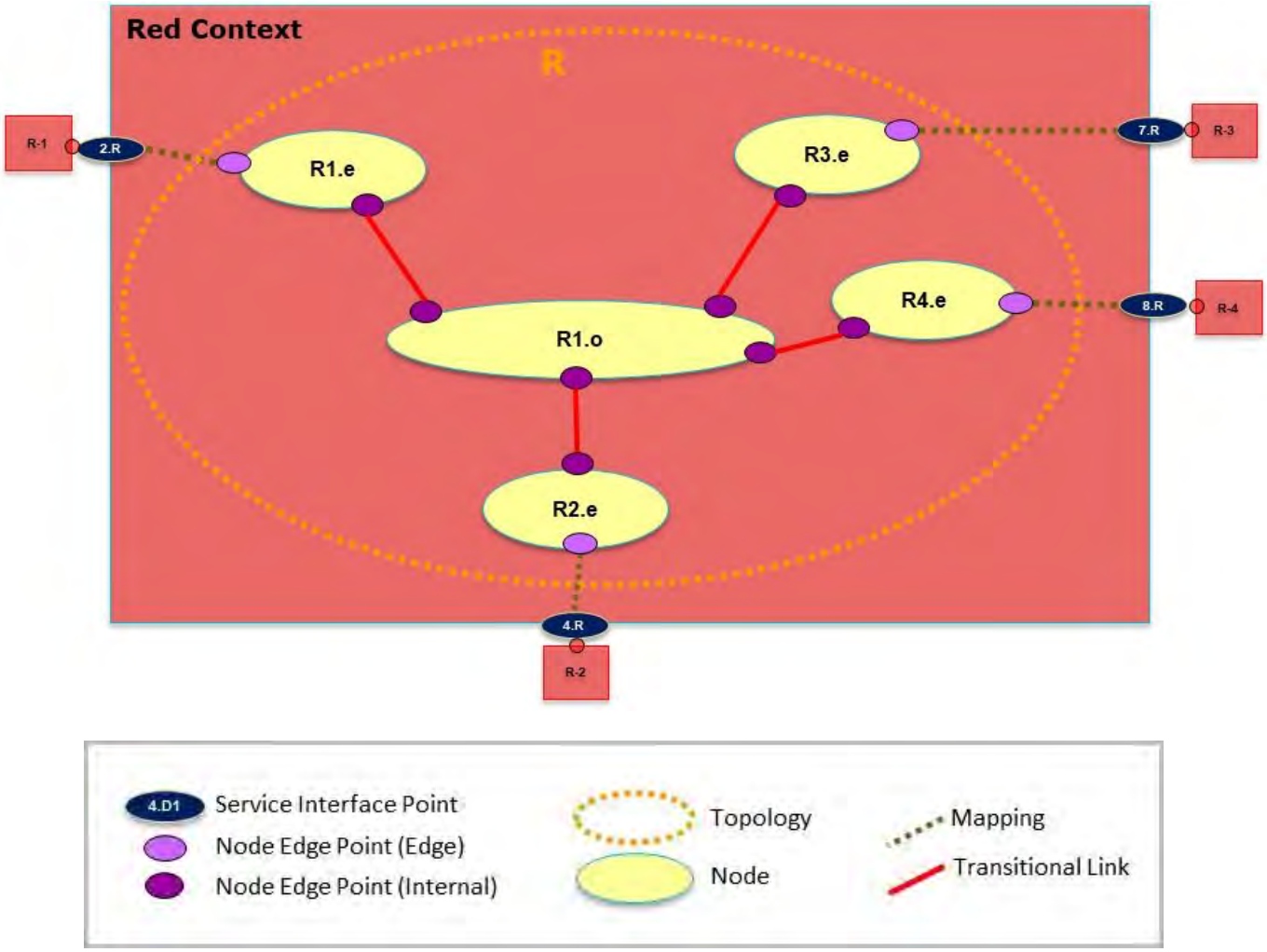

Figure 12 – Red Context Topology

Note that in Figure 6 and Figure 7 the SIP 7 and 8 are at ODU layer, while in Figure 12 and Figure 13 they are at Ethernet layer.

When SOF requests a 10G EPL *Connectivity Service* in Red Client Context, between *Service Interface Points* (*SIPs* 2.R and 7.R), then:

1) The ICM computes and provisions end-to-end connectivity within its internal topology.
2) The ICM creates the following objects at Presto IRP:
   a. The *Connectivity Service*
   b. The ODU *Connection* in *Node* R1.o
   c. The Ethernet *Link* between R1.e and R3.e *Nodes*
   d. The *Connections* in R1.e and R3.e plus the supported "top" *Connection* at higher partitioning level.

Figure 13 depicts the resulting scenario, including the "equipment oriented" view below.

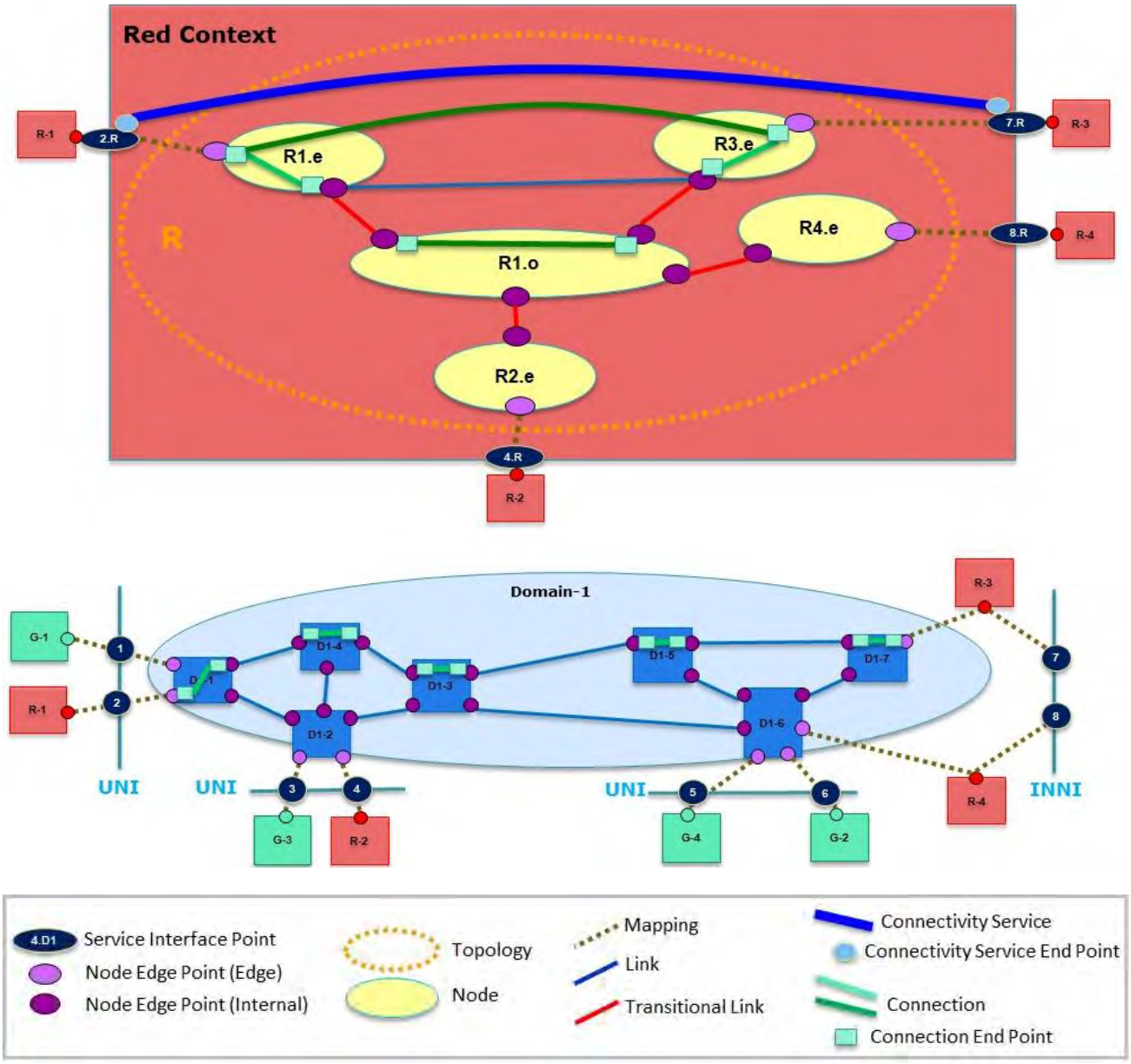

Figure 13 – 10G Connectivity Service in Red Context

5.3 Augmentation Approach – Model Federation

TAPI model is structured into a technology agnostic core and a number of technology specific augmentations, see Figure 14.

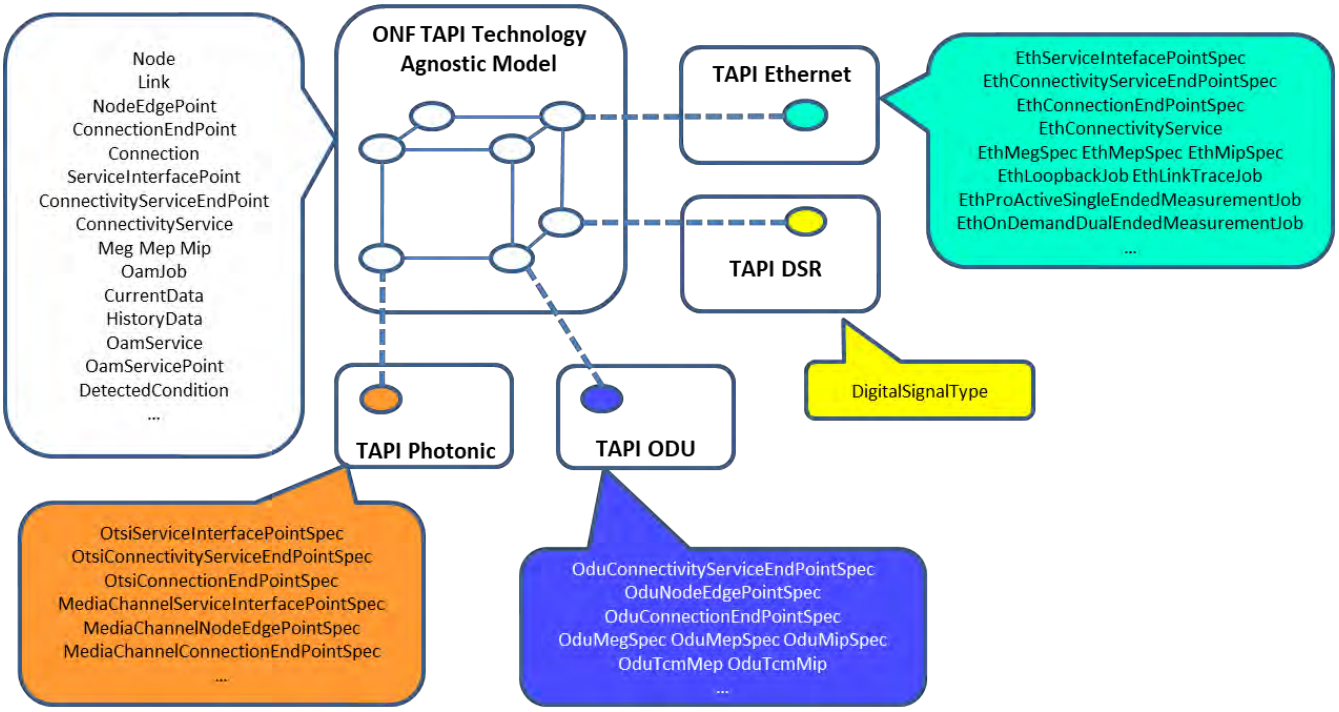

Figure 14 – Federation of technology agnostic and technology specific models

The model capabilities are extended through the augmentation approach. The essential concept of augmentation is to enable an instance of a base class to be extended with a set of attributes that characterize the specific case of augmenting class.

Augmentation does not require that base class is aware of the extensions, which allows for easy incorporation of newer technology or a vendor specific feature. Thus an implementation (if desired) can be developed against just the base/standard model (schema) and ignore the augmentations (e.g., vendor specific augmentations).

In this way, the augmentation approach is flexible and does not lead to a proliferation of specialized classes (which are difficult to maintain) and does not lock children to all modifications of parents. In other words, augmentation allows for a flexible federated relationship between models.

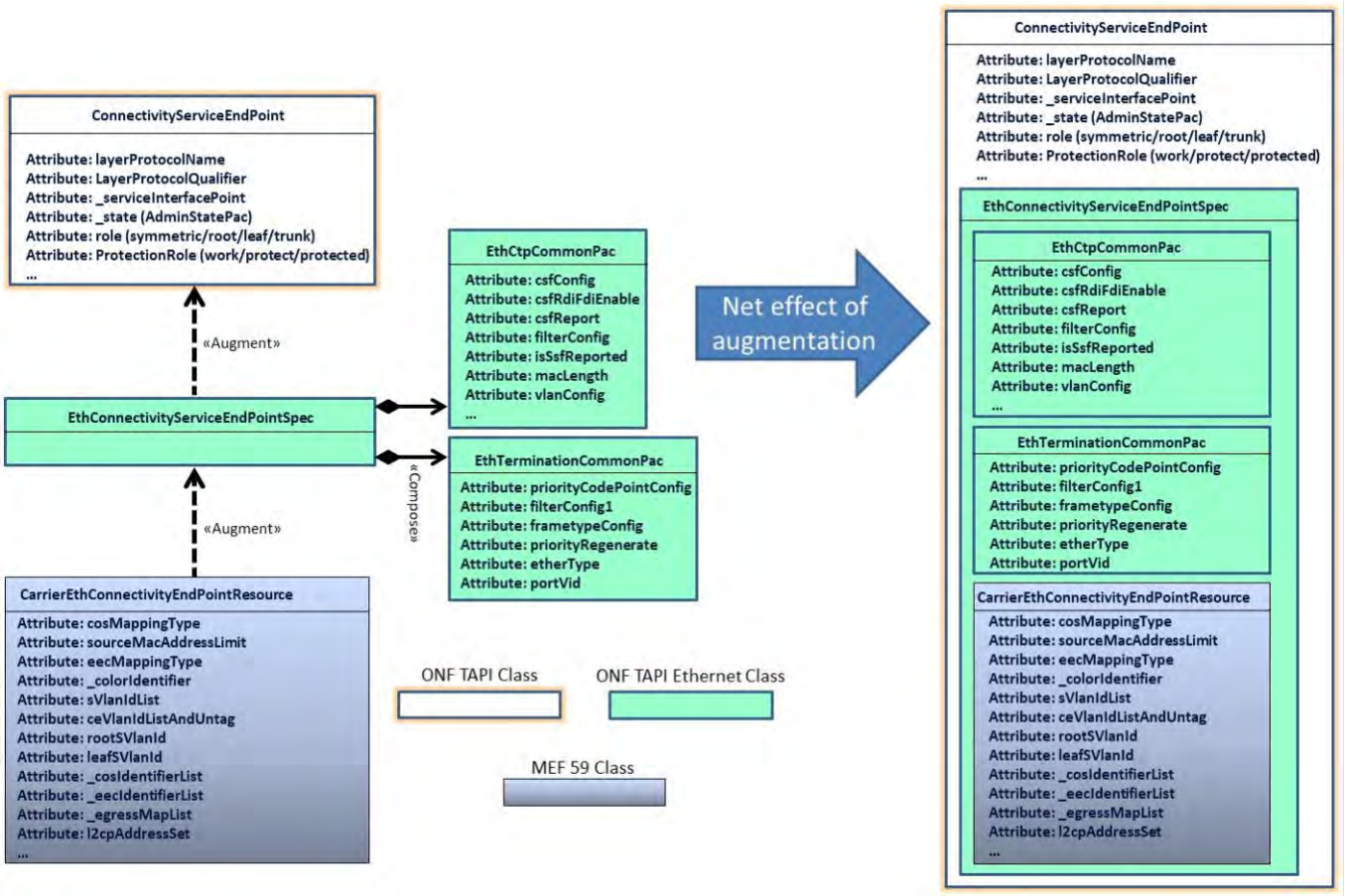

Figure 15 – Example of Augmentation

Figure 15 shows how the technology independent *Connectivity Service End Point* class is augmented by both ONF and MEF Ethernet specific classes.

5.4 Operations API (RPC) vs Data API

Given the low penetration in the industry of the RPC-based API implementation, ONF is deprecating RPCs from TAPI, leveraging for example RESTCONF (RFC 8040 [2]) which uses HTTP methods to provide CRUD operations on a conceptual datastore containing YANG defined data, i.e. GET, DELETE, PATCH, POST, and PUT methods.

TAPI does not mandate direct access to all data nodes defined by its YANG models, as typically only a subset of *configurable* objects are required to provide full CRUD support. Reference implementation agreements can specify the required RESTCONF Data API, an example being ONF TR-547 [20].

6 Model Descriptions and Examples

The purpose of this section is to provide an overview of the MRM Common through a selection of UML diagrams with key classes, their relationships and attributes.

Some of the UML diagrams are very dense. To view them either zoom (sometimes to 400%) or open the corresponding UML diagram via Papyrus [22] (for each figure with a UML diagram the UML model diagram name is provided under the figure in *italic* font).

6.1 Common Model

This section provides an overview of the model constructs defined in Common Model (Section 7).

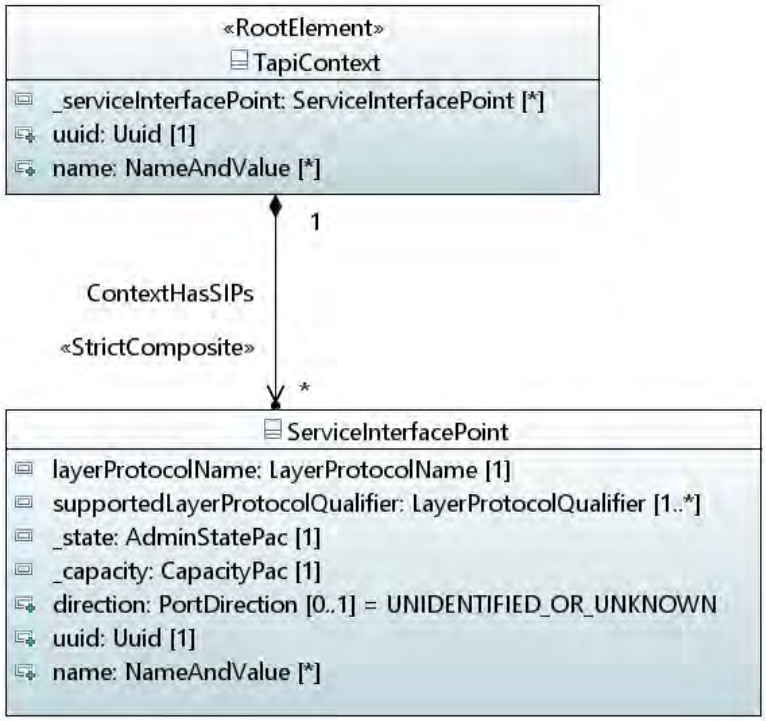

Figure 16 – *Context*

All interactions between an API provider (e.g., SDN Controller) and an API Client (e.g., Application, Orchestrator or another SDN Controller) occur within a shared context. In other words, the *Tapi Context* represents the scope of control that a particular SDN controller has with respect to a particular network.

The *Tapi Context* is defined by a set of *Service Interface Points* (*SIPs*) and is a container for all classes, *service* and *resource oriented*.

The *Tapi Context* can be augmented by *Topology Context*, which contains zero, one or more top-level *Topology* instances (see Section 6.2). Note that the "opaque view" case does not see any *Topology*, i.e., only *SIPs* are visible.

The *Service Interface Points* are the entry points for connectivity provisioning.

Common implementations foresee that *Service Interface Point* are mostly read-only, i.e. made available by the server controller. An example of provisionable attribute is their *Administrative State*.

6.2 Topology and Connectivity Models

This section provides an overview of the model constructs defined in Topology (Section 8) and Connectivity (Section 9) Models.

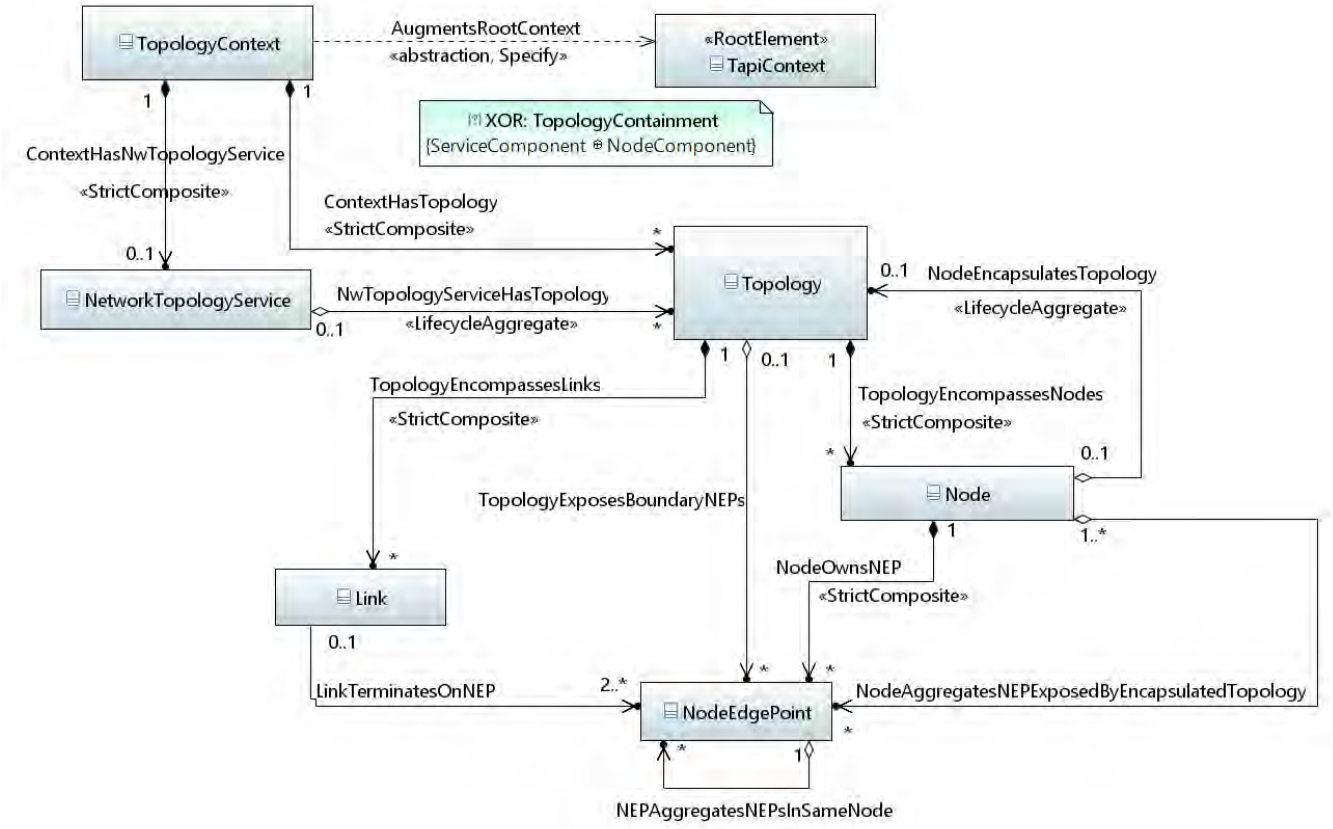

Figure 17 – *TopologyServiceSkeleton*

A *Topology* resource may either be:

- Statically assigned on Server side – is referenced by (belongs to) *Network Topology Service.*
- Dynamically created by Client – is referenced by (belongs to) *Virtual Network Service*.

Figure 17 shows *Network Topology Service* class, the *Virtual Network Service* class is defined in TAPI Virtual Network model, which is not included in this Standard. Hence all Topology objects are considered read-only in this Standard.

The *Topology Context* represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the topology description. An instance of this class includes its *Topology* object instances, which are an abstract representation of the topological aspects of a particular set of network resources.

The model supports topology decomposition, which allows for information hiding and abstraction and allows to recursively decompose a top-level *Topology* down to the lowest-level *Nodes* and *Links*:

- A *Node* can encapsulate an internal *Topology*, which recursively encompasses lower-level *Nodes* and *Links* (partitioning).
- A *Node* either "owns" or "aggregates" *Node Edge Points* (*NEPs*).
- *NEPs* are instantiated only at the lowest partitioning level.
    - Lowest partitioning level *Nodes* "own" the *NEPs*.
    - Higher partitioning level *Nodes* "aggregate" the *NEPs* which are visible at that level.
- A *NEP* can be specified as a *pool* of more *NEPs*, e.g., when a set of *NEP* instances are equivalent for usage. For example, a higher partitioning level Node can "aggregate" only the *pool NEP*, abstracting the component *NEPs* at this partitioning level.
- A *Link* terminates on *Node Edge Points*.

In other words, the *Node* is simply an opaque view of a *Topology* wherein just the edge ports (*Node Edge Points*) on the boundary are exposed. Hence it can be recursively decomposed over the API to expose its "internal" *Topology*, if allowed by the policies.

The lowest level *Node* is simply just an abstraction wherein the policy does not allow exposure of its internal details, e.g., it could be that a *Node* cannot be further decomposed due to physical constraints (switch matrix).

One of the exposed top-level *Topology* could be the provider's own internal topology (1-1 mapping).

Note that a *Topology* instance may be referenced either by a *Network Topology Service* instance (i.e., case of top-level *Topology*) or by a *Node* instance (i.e., a Topology at lower partitioning level) exclusively, as indicated by the XOR box in Figure 17.

The *Node* and *Link* express their transfer characteristics via the following packages (ancillary classes):

- *Capacity, Cost, Latency, Integrity* packages.

The *Link* includes also:

- *Risk Parameter* package.
- *Layer Protocol Transition* package, in case of *Transitional Link*.

A *Node* may refer to zero, one or more *Node Rule Groups*. A *Node Rule Group* object specifies different rules applicable to the *Node* or to a group of *NEPs* of the *Node*. For example, it is possible

to specify whether forwarding is allowed between two groups of *NEPs*, but forbidden between *NEPs* of the same group. It is possible to specify complex rules by associating more *Node Rule Group* and *Inter Rule Group* objects, see Figure 18.

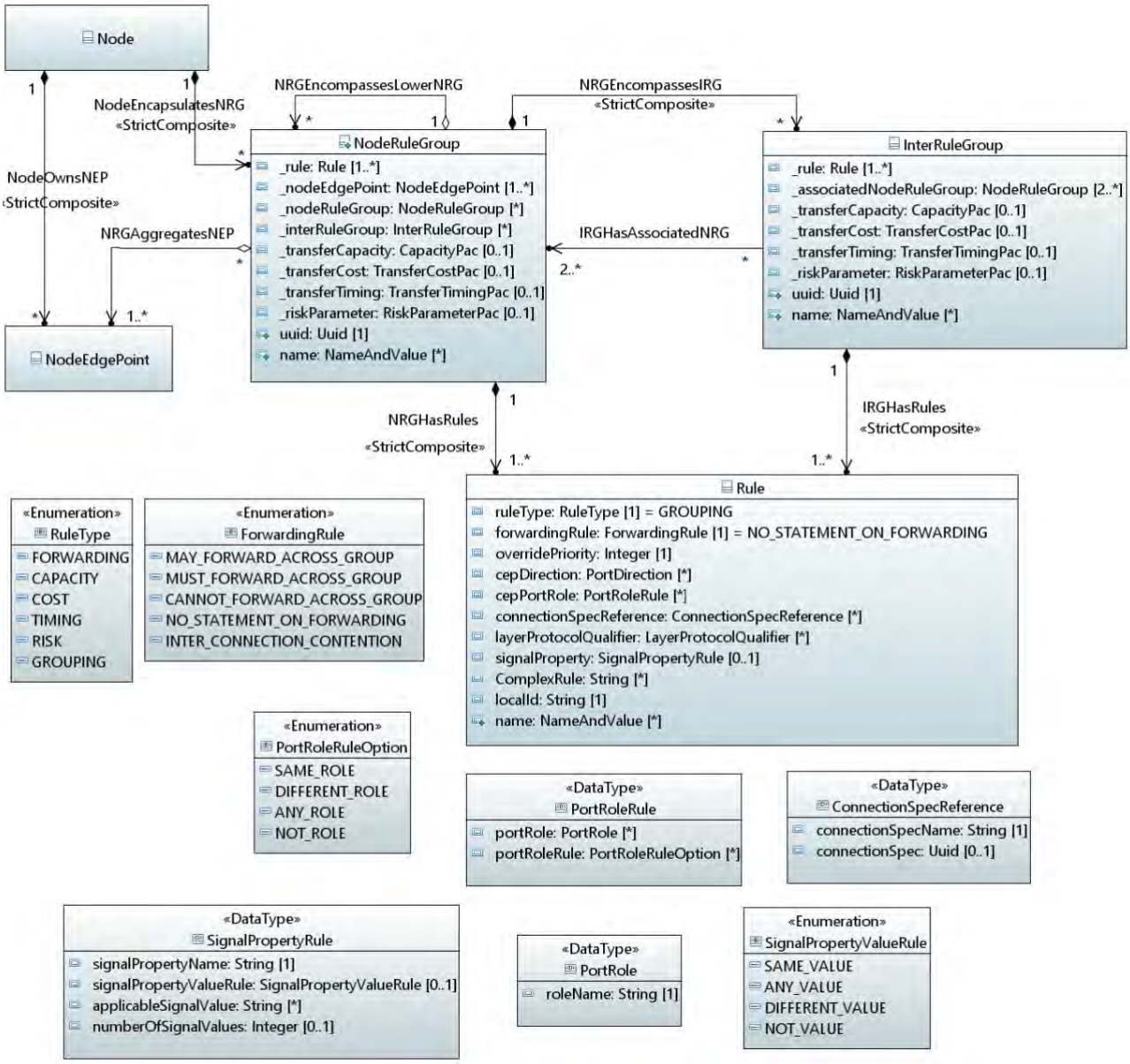

Figure 18 – *NodeConstraints*

The *Node Edge Point* represents the node-facing aspect of the edge-port functions for accessing the forwarding capabilities provided by the *Node*. In other words, the *Node Edge Point* provides an encapsulation of addressing, mapping, termination, adaptation and OAM functions of one or more transport layers (including circuit and packet forms) performed at the entry/exit points of *Node*. *Node Edge Point* has one or more *Layer Protocols*:

- Example of *Layer Protocols* are: Photonic, Digital OTN, Ethernet, DSR.
- *Layer Protocol* supports technology extensions via the augmentation pattern.

The *Node Edge Point* specifies the potentially supported *Connection End Points* (similarly to "payload structure" concept) at each supported *Layer Protocol Qualifier*, e.g., {80 *CEPs* at ODU0 rate}; {40 *CEPs* at ODU1 rate}.

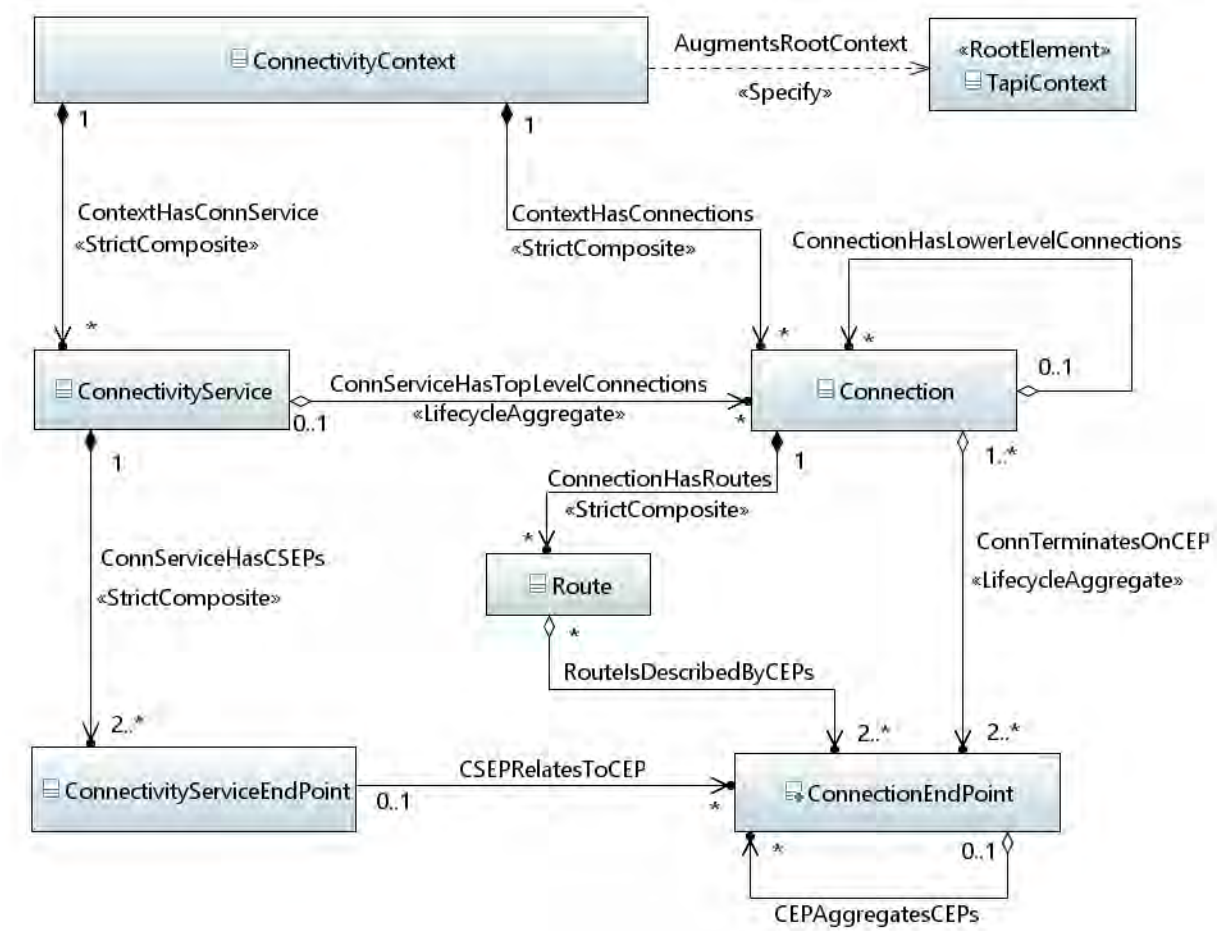

Figure 19 – *ConnectivityServiceSkeleton*

The *Connectivity Context* represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the connectivity description. An instance of this class includes its *Connectivity Service* and *Connection* object instances.

As a result of the *Connectivity Service* provisioning between two or more *Connectivity Service End Points*, the Server creates one or more *Connections*, which are the containers that track the state of the allocated resources.

The *Connectivity Service End Point*, supported by a *Service Interface Point,* is used to provision the transmission functions at the ingress/egress point of the *Connectivity Service*. In other words, the *Connectivity Service End Point* abstracts the configuration of *Connection End Points*.

The *Connection End Points* encapsulate information related to a *Connection* at the ingress/egress points of every *Node* that the *Connection* traverses in a *Topology*. Every *Connection End Point* is supported by a specific parent *Node Edge Point*.

The *Connectivity Service* provisioning may include routing and topology constraints, resilience properties, other constraints on cost, latency, etc.

Figure 20 shows the relationships between topology and connectivity classes.

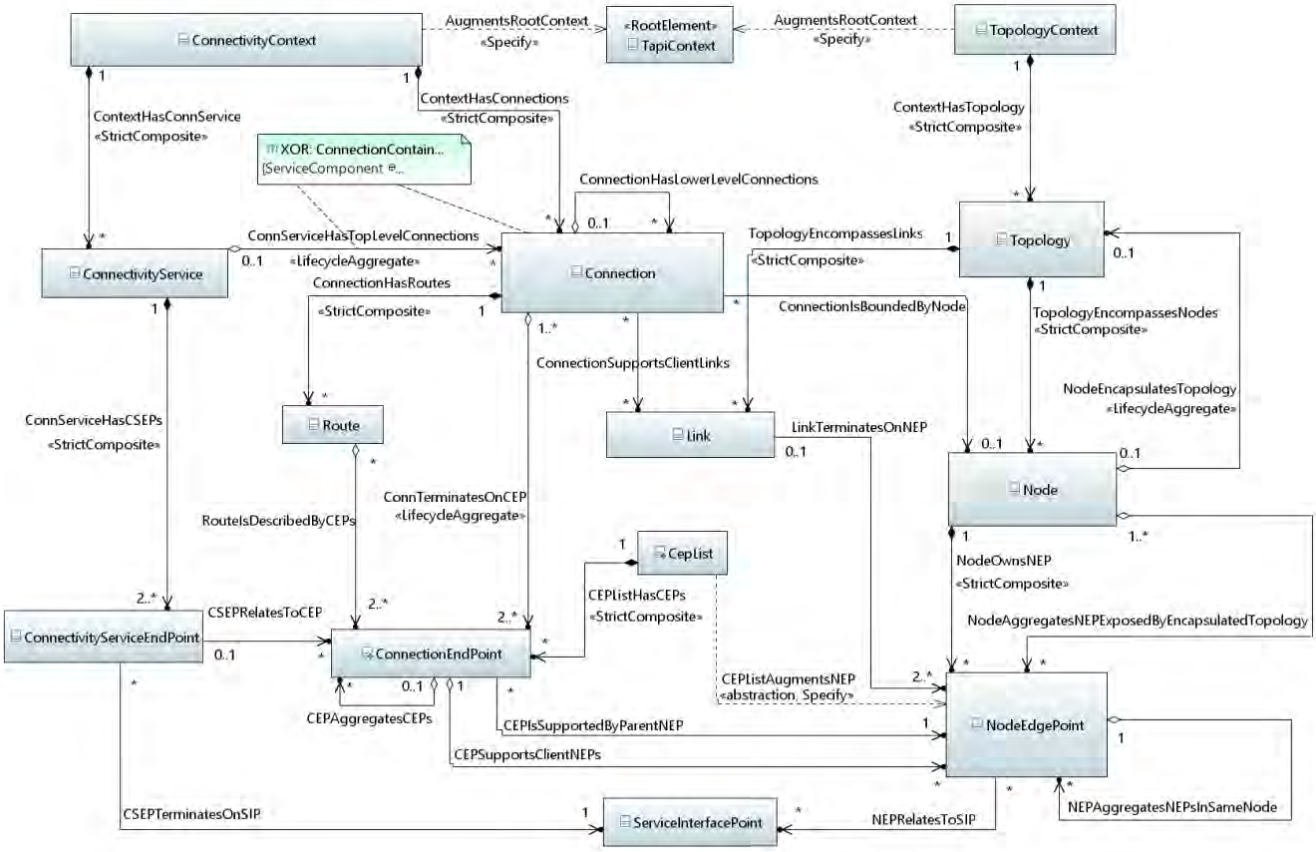

Figure 20 – *ConnectivityTopologySkeleton*

The *Connection* spanning a *Node* at a given partitioning level can be recursively decomposed into *Connections* which span *Nodes* at the lower partitioning level. A simple example is an end-to-end *Connection* spanning the whole management domain, decomposed into its constituent "cross connections" which represent the smallest *managed* flexibility. Note that equipment may include even more elementary/atomic connections, which are purposely hidden to connectivity management. Figure 21 shows an example of highest partitioning level where the top-level *Topology* D1 encompasses one top level *Node* D1, which defines the forwarding scope of a top-level *Connection*.

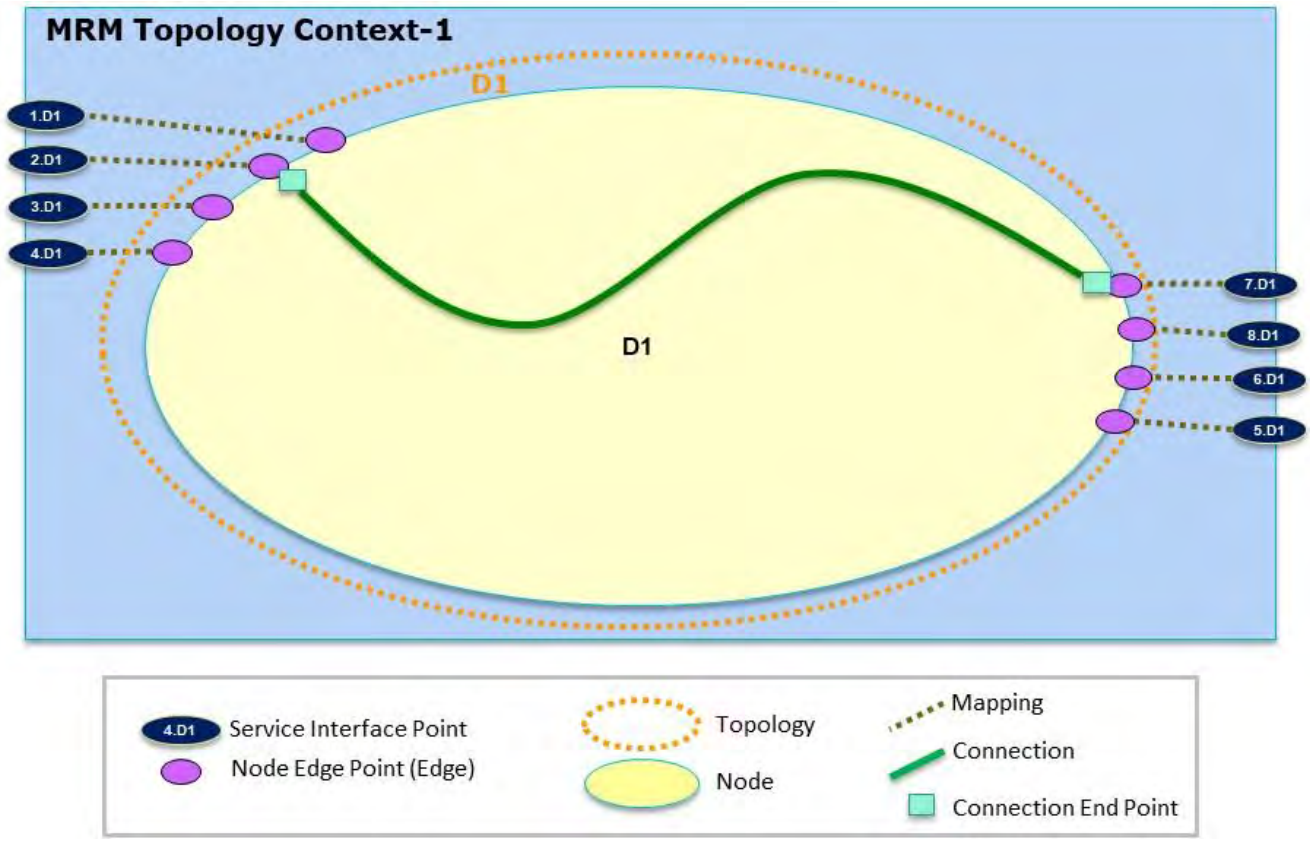

Figure 21 – Top-level *Topology* D1 encompasses one top-level *Node* D1

Figure 22 shows the partitioning of *Node* D1, which encapsulates *Topology* D1-1.

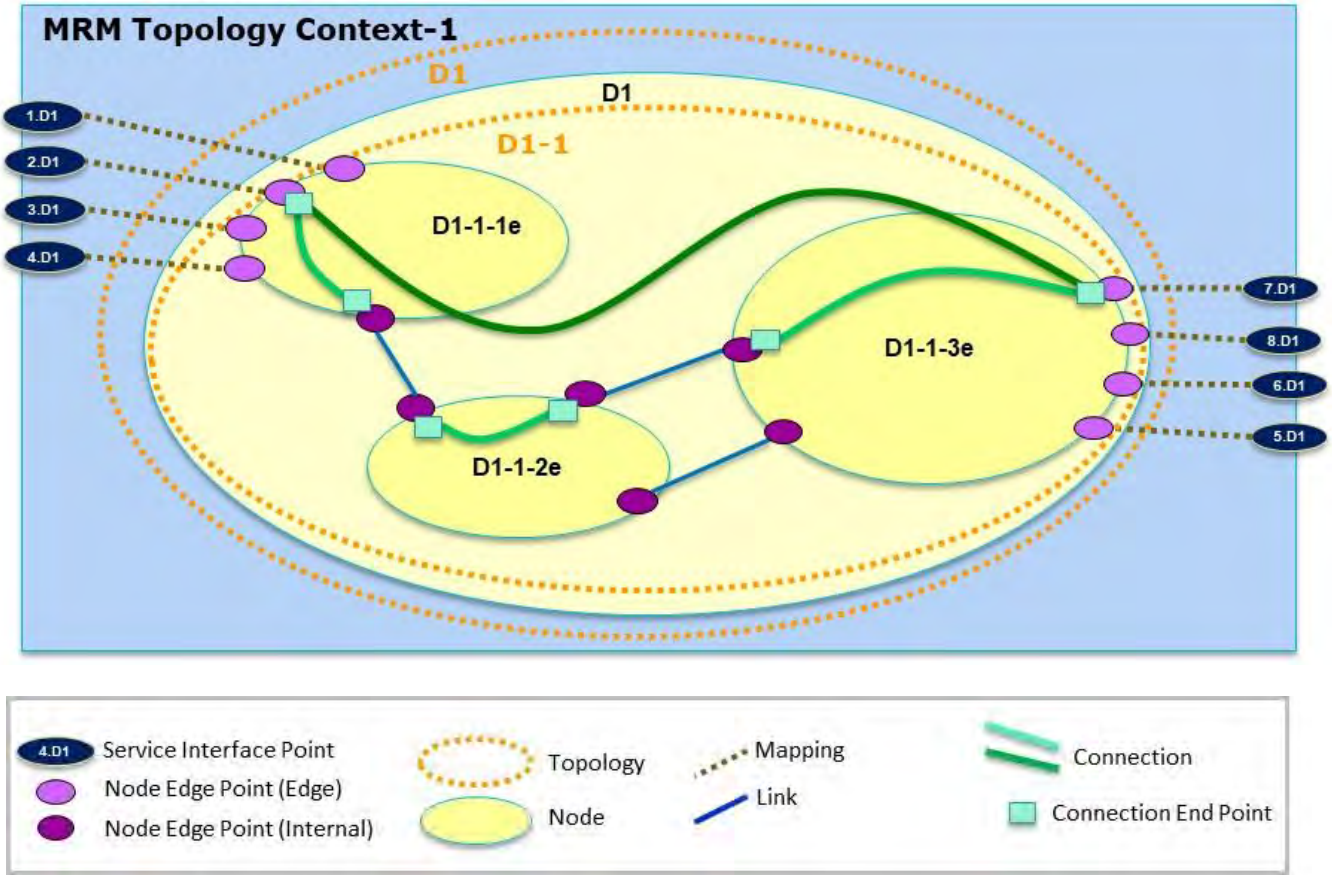

Figure 22 – Top-level *Node* D1 encapsulates *Topology* D1-1

Topology D1-1 describes the topology at lower partitioning level, which includes three *Nodes* and the *Links* between them. The top-level *Connection* is partitioned into three *Connections* at lower partitioning level.

Note that the *Connection* at highest partitioning level (top-level *Connection*) may not be bounded by a Node when it is assumed that the top-level *Topology* encompasses only one *Node*. Hence a *Topology*-*Node* recursion is skipped, see Figure 23.

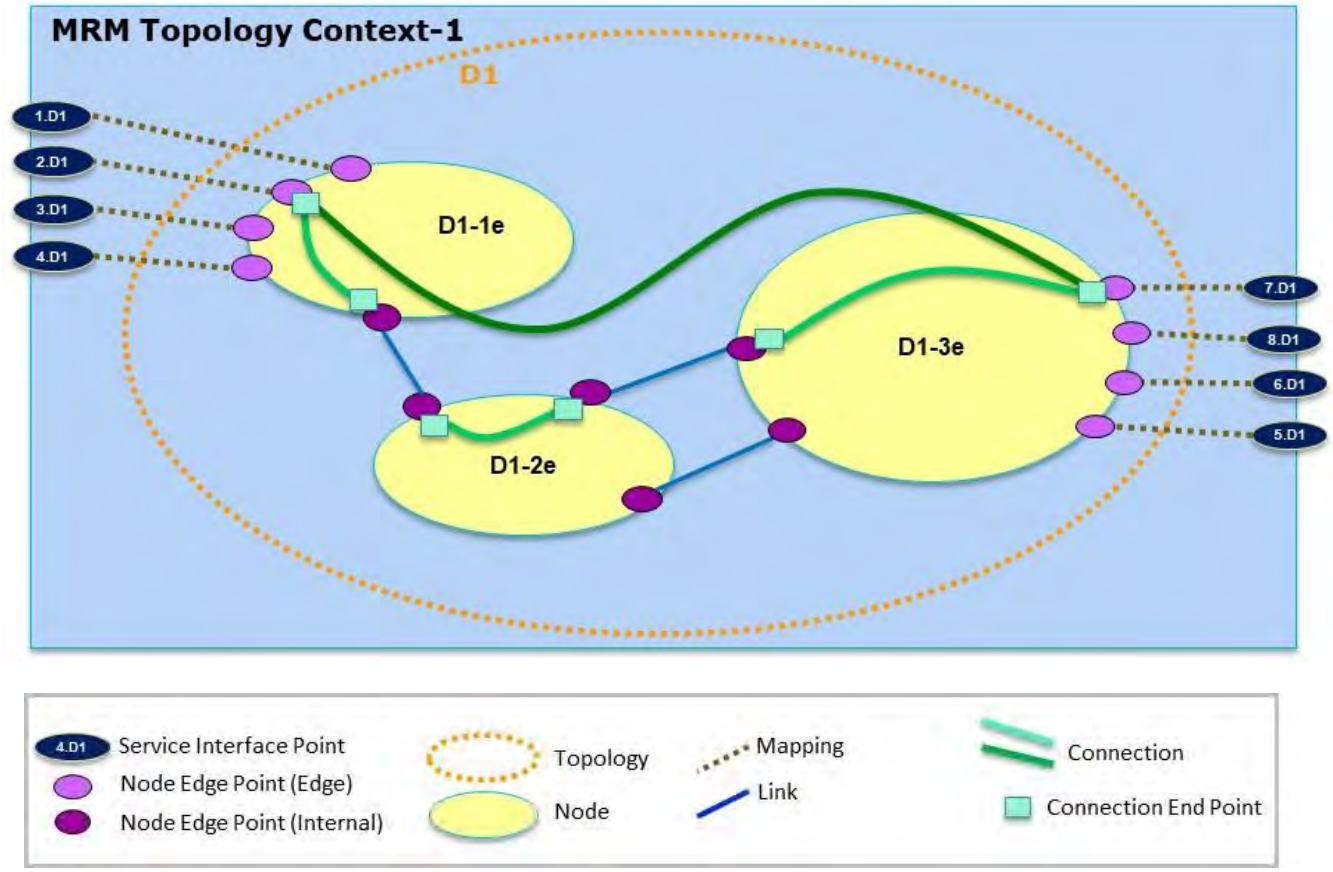

Figure 23 – Simplified Partitioning

The *Route* represents the path of a *Connection*, described as a list of *Connection End Points*. The logical order of the *Connection End Points* within the *Route* object can be inferred by the TAPI client by the knowledge of the topology. Resilient *Connections* may have more distinct *Resilience Routes*.

6.3 Path Computation Model

This section provides an overview of the model constructs defined in Path Computation Model (Section 10).

The Path Computation management is part of Connectivity Management, together with Topology Management: SOF implements Service Decomposition towards ICM layer, the Service Decomposition process can rely on Path Computation to check Connectivity feasibility/constraints.

The *Path Computation Context* represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the path computation description. An instance of this class includes its *Path Computation Service* and *Path* object instances. The *Path Computation Service* purpose regards the planning aspects of forwarding services.

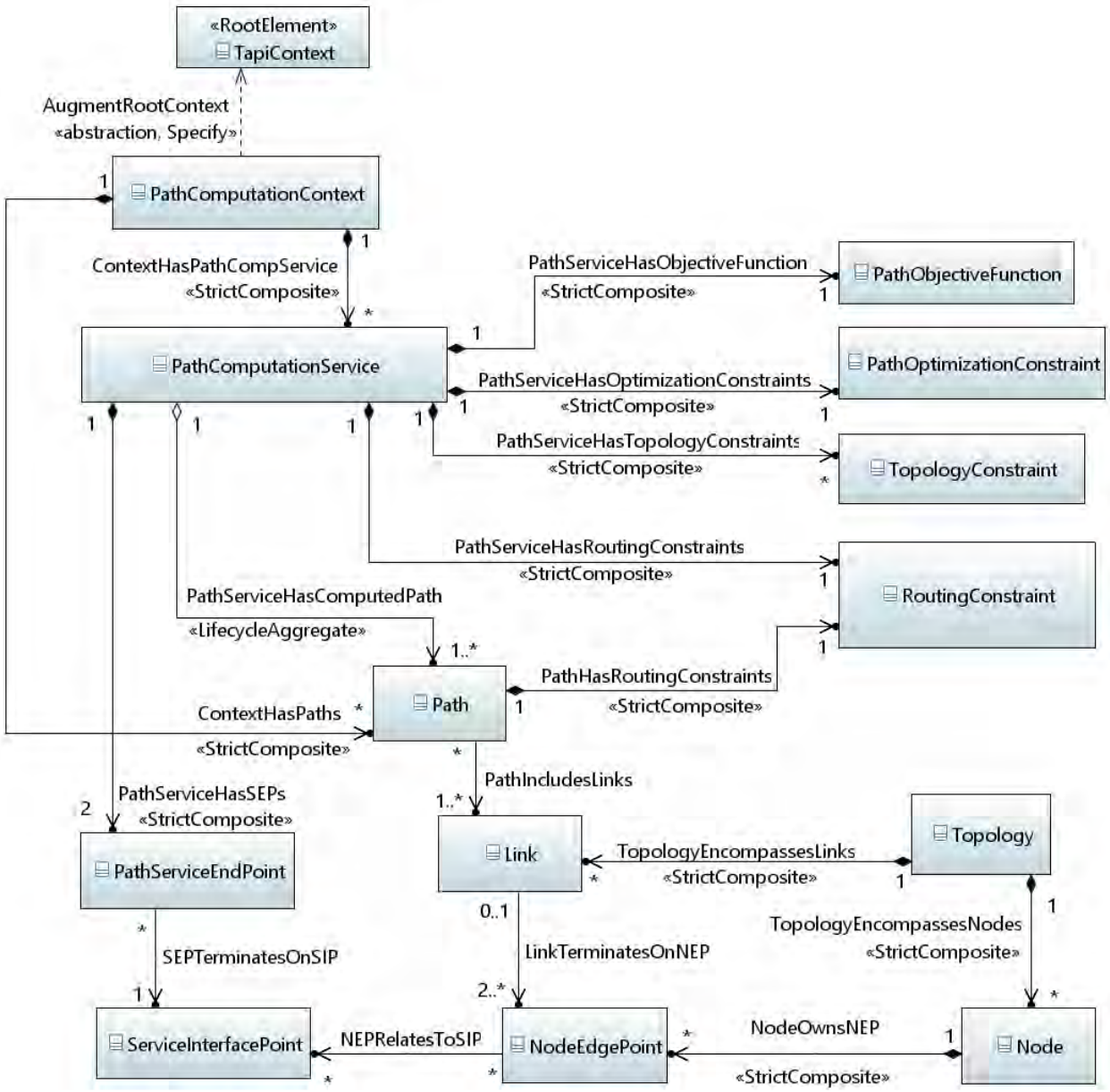

Figure 24 – *PathComputationServiceSkeleton*

As a result of the *Path Computation Service* provisioning between two *Path Service End Points*, the Server creates one or more *Paths*, which can be used as routing constraint criteria for *Connectivity Services*. For example, it is possible to create a *Connectivity Service* with *Route* constrained to follow or avoid a *Path* instance.

- The *Path Service End Point* encapsulates information related to a *Path Computation Service* at the ingress/egress points of that *Path Computation Service.*
- The *Path* is described by an ordered list of *Links*. A *Link* is conceptually defined by a pair of *Node Edge Points*.

6.4 Fault Management Model

This section provides an overview of the model constructs defined in FM Model (Section 11).

The *Detected Condition* data structure augments the *Event Notification* signal, defined by Notification Management Model, with the information related to an alarm or a threshold crossing alert detected on a given resource. See Figure 25.

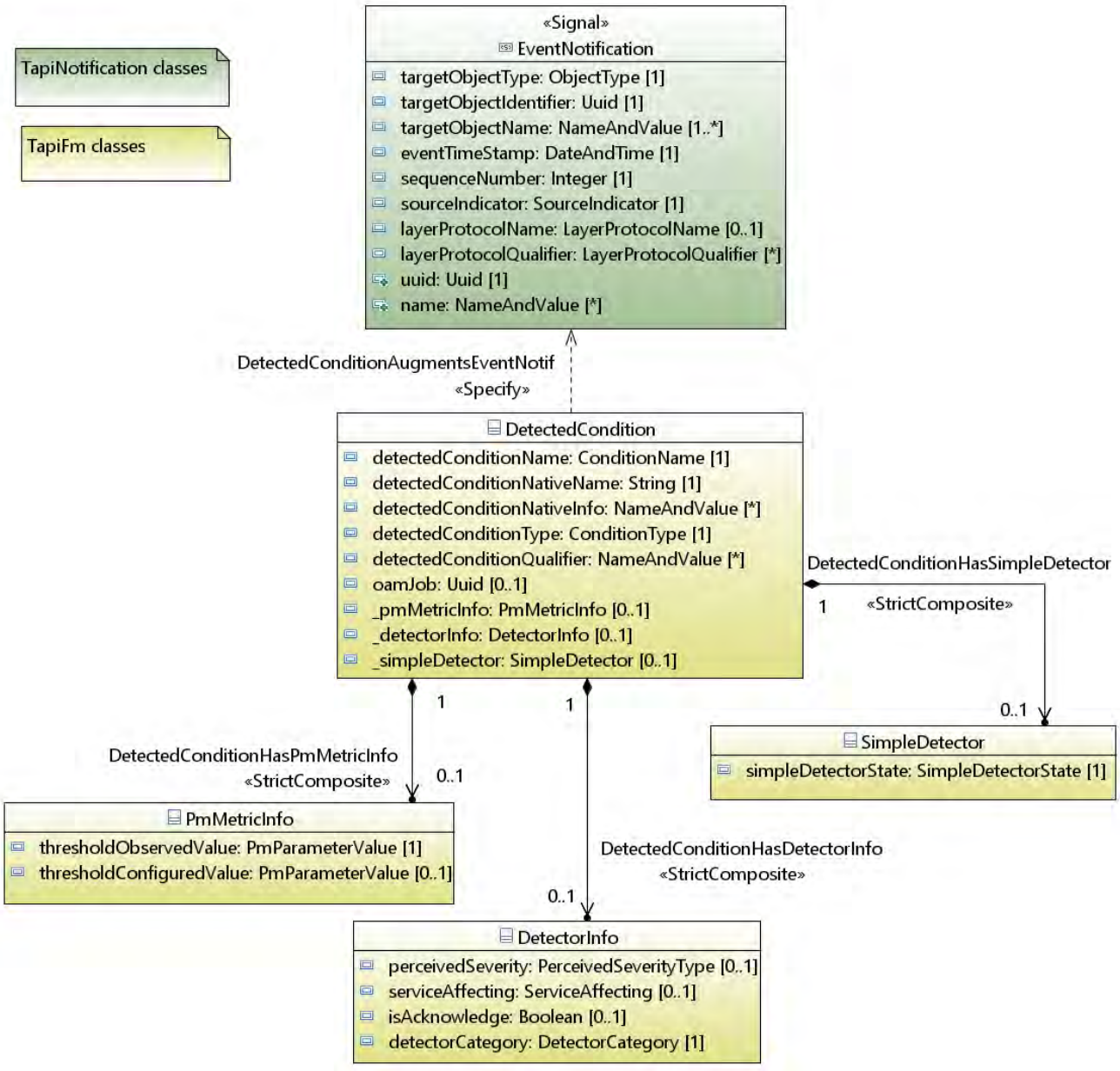

Figure 25 – *FmDetails*

The *ConditionName* data type can be extended by technology specific alarm probable causes or PM metrics, e.g. OTN Open Connection Indication, Ethernet Loss Of Continuity or OTN Background Block Error, Ethernet Maximum Frame Delay.

6.5 OAM Model

This section provides an overview of the model constructs defined in OAM Model (Section 12).

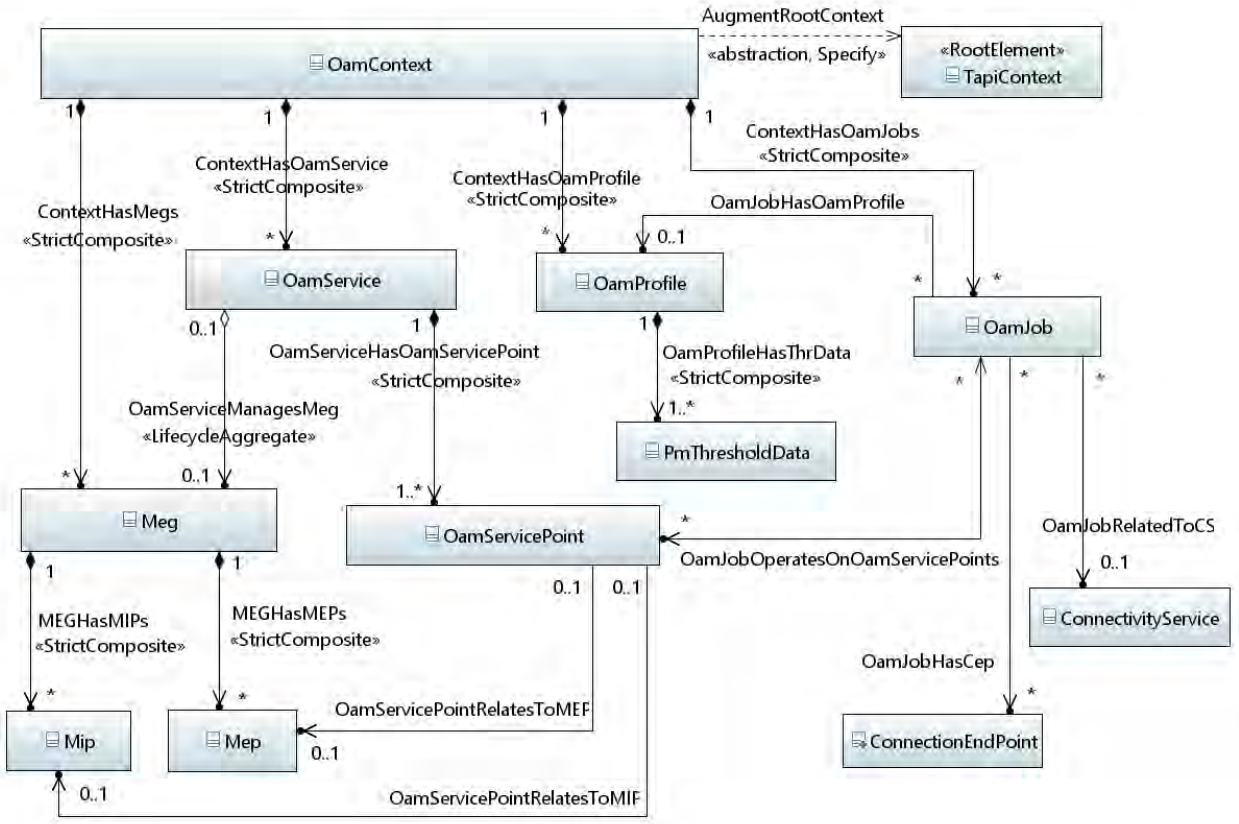

Figure 26 – *OamSkeleton*

The *Oam Context* represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the OAM description. An instance of this class includes its *Oam Service*, *Oam Profile*, *Oam Job* and *Meg* object instances.

As a result of the *Oam Service* provisioning, the Server creates one *Meg* instance (with its *Mep* and *Mip* instances) which are the containers that track the state of the allocated resources.

The *Oam Service Point* is used to provision OAM functions, abstracting the configurations of *Meps* and *Mips*. *Service Interface Points, Connectivity Service End Points* and *Connection End Points* may be monitored by one or more *Oam Service Points*.

The monitoring of:

- *Service Interface Points* applies to the monitoring at UNI or ENNI scope, e.g., reachability of remote interfaces independent from supported services.
- *Connectivity Service End Points* applies to the monitoring of *Connectivity Services*.
- *Connection End Points* applies to the monitoring of *Connections*.

On service-oriented side, *the Oam Service* and *Oam Service Point* are augmented by technology specific classes, e.g., *EthOamService*, *EthOamMepServicePoint*, *EthOamMipServicePoint* (see [16]).

Analogously on resource-oriented side, the *Meg*, *Mep* and *Mip* are augmented by technology specific classes, e.g., *EthMegSpec*, *EthMepSink*, *EthMepSource* (see [16]).

Figure 27 shows the relationships between OAM and connectivity classes.

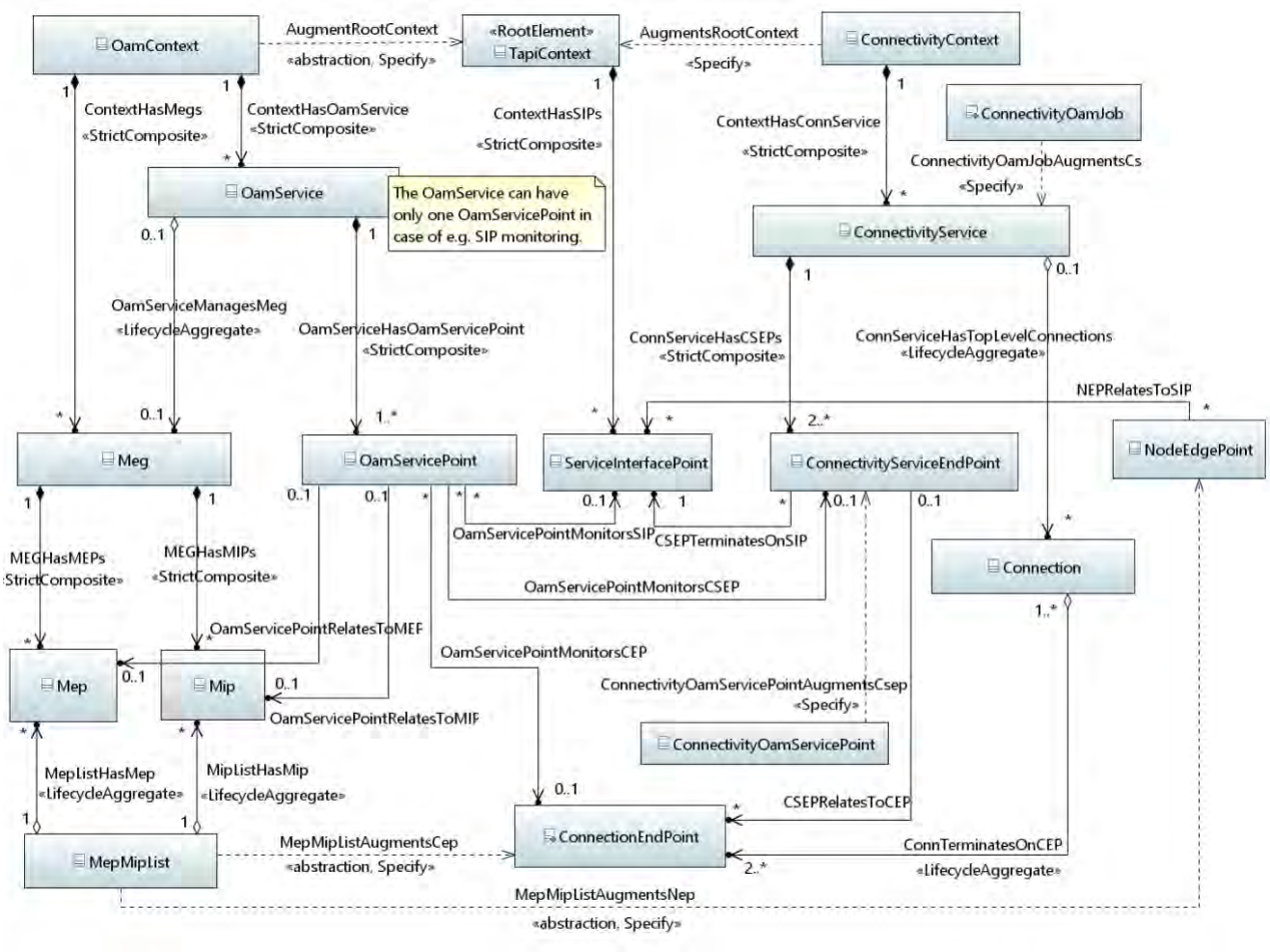

Figure 27 – *OamConnSkeleton*

One or more *Oam Job* instances can be provisioned on a *Oam Service* instance. Each technology specific model augments the *Oam Job* class with technology specific features, e.g., OTN Tandem Connection Monitoring, Ethernet Pro-Active Single Ended Measurement Job, Ethernet Link Trace Job, etc.

The OAM model defines data structures for :

- *Pm Threshold Data*: the PM metrics, their measurement interval and threshold values.
- *Current Data*: the PM metrics and their values under current measurement period.

- *History Data*: the PM metrics and their values of past measurement periods.

Pm Threshold Data, Current Data and *History Data* classes are augmented by technology specific classes, e.g., OTN Error Performance Data, Ethernet Pro-Active Loss Measurement Performance Data.

An *Oam Job* instance (see Figure 28) may refer to:

- One *Oam Profile* instance, which includes one or more *Pm Threshold Data* instances.
- One or more *Current Data* instances, each *Current Data* instance referring to zero, one or more *History Data* instances.

It is also possible to provision OAM aspects together with *Connectivity Service* provisioning:

- The *Connectivity Oam Service Point* class augments the *Connectivity Service End Point*, providing a hook to technology specific *MEP* and *MIP* configuration.

- The *Connectivity Oam Job* class augments the *Connectivity Service*, providing a hook to technology specific OAM job types.

This compact provisioning option is specially applicable to technologies where OAM is mostly embedded with forwarding, e.g., OTN.

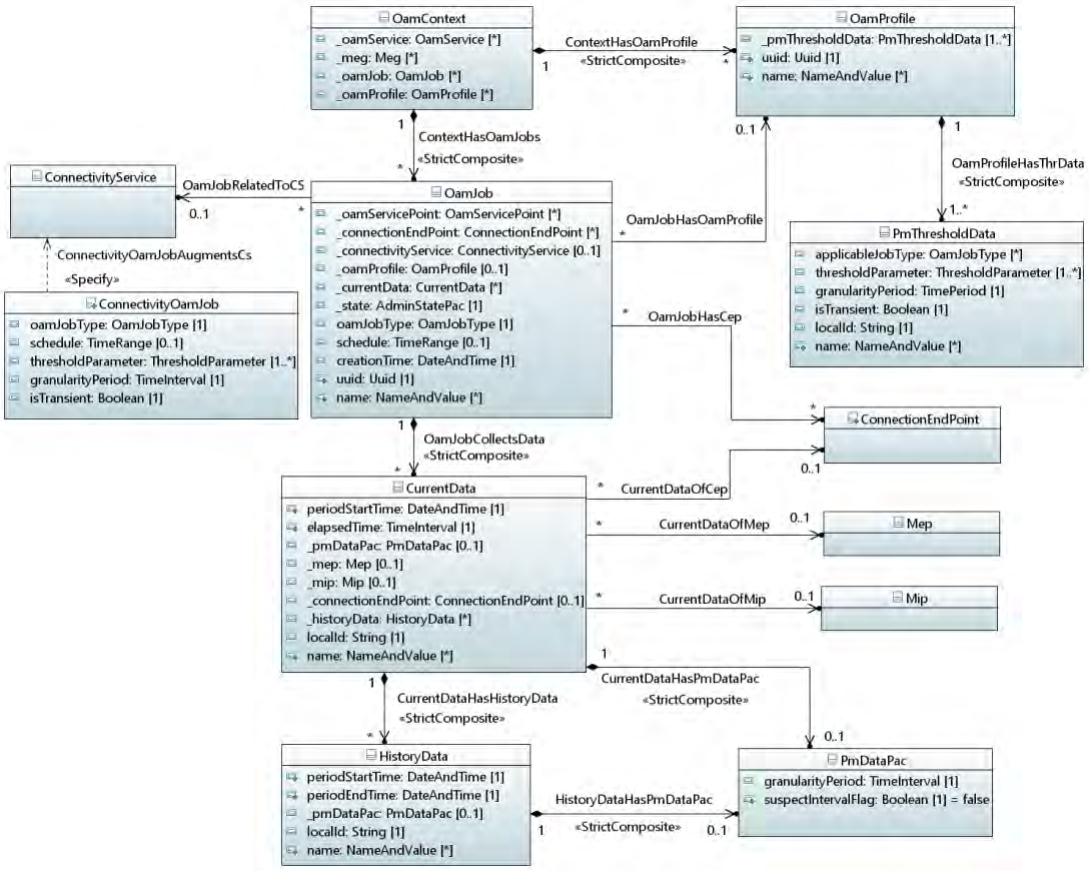

Figure 28 – *OamJobDetails*

6.6 Notification Model

This section provides an overview of the model constructs defined in Notification Model (Section 13).

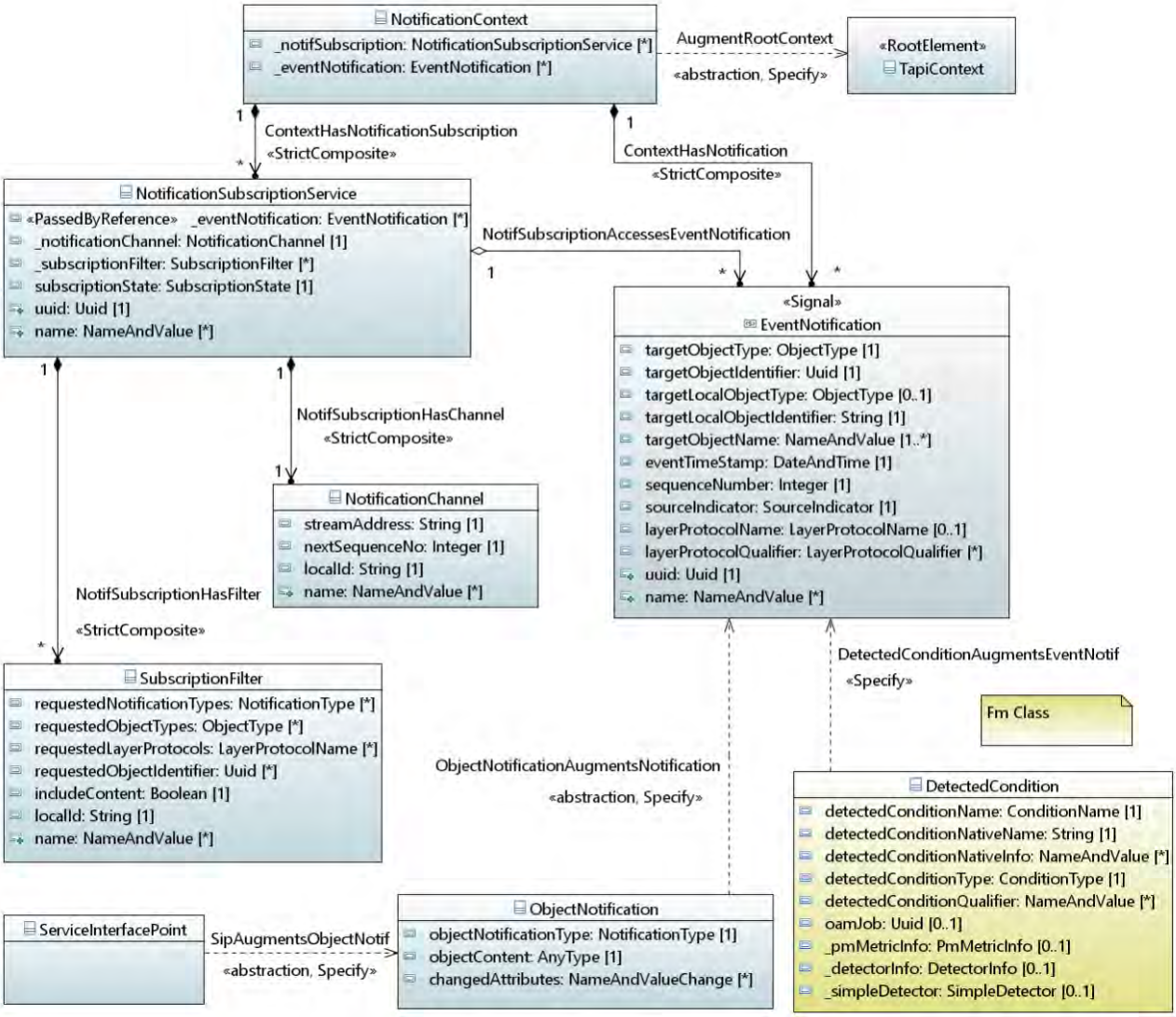

Figure 29 – *NotificationServiceDetails*

The *Notification Context* represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the notification description. An instance of this class includes its *Notification Subscription Service* and *Event Notification* instances.

As a result of the *Notification Subscription Service* provisioning, the Server is enabled to raise *Event Notification* signals according to the following subscription parameters:

- *Notification Channel*
- *Notification Types*:
  - OBJECT_CREATION

  - OBJECT_DELETION
  - ATTRIBUTE_VALUE_CHANGE
  - DETECTED_CONDITION
- *Object Types* and *Object Identifiers*
- *Layer Protocol Names*

The *Event Notification* signal includes the following generic parameters:

- *Target Object Type*, the type (class) of the object instance to which the *Event Notification* instance is related to.
- *Target Object Identifier*, the UUID of the object instance to which the *Event Notification* instance is related to.
- *Event Time Stamp*, the best knowledge of the time of the event which originated the *Event Notification* instance.
- *Layer protocol name* and *qualifiers* of the resource to which the *Event Notification* instance is related to.

The *Event Notification* signal can be augmented with function or technology specific contents, e.g., the *Detected Condition* defined in Fault Management Model, or all the object classes which e.g. creation can be notified, for example see Figure 30 for Connectivity objects.

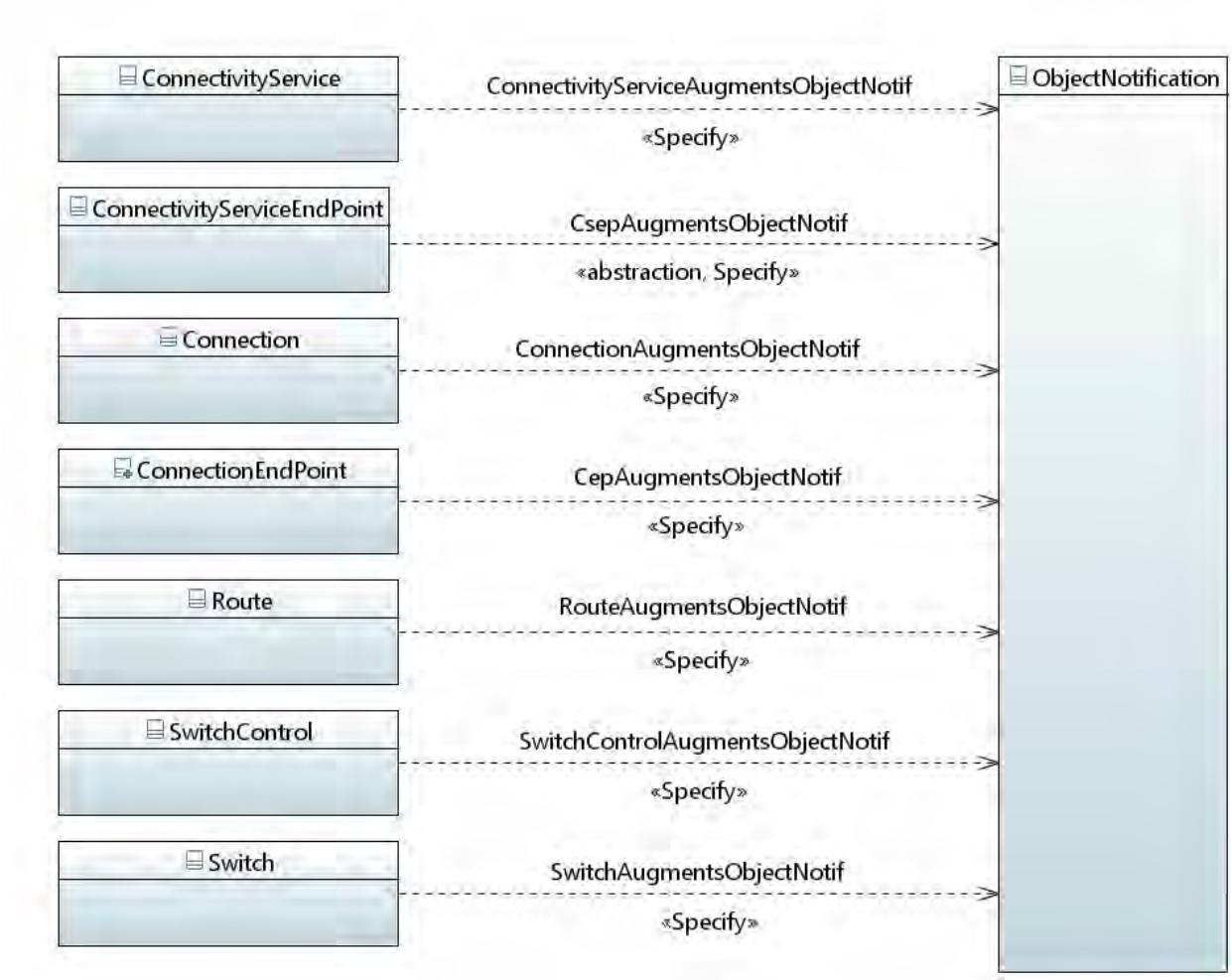

Figure 30 – *ConnectivityNotif*

6.7 Profiles of Utilization

A set of profiles are defined to restrict functionalities for different use cases :

- *Full-topology-management*: The full management of topology, resilience, and path computation features.
- *Single-node-management*: The management is restricted to the "top" *Topology* (i.e. the *Topology* instance which is not encapsulated in a *Node*), its encompassed *Node* and owned *NEPs*.
- *Top-connection-management*: This profile, which requires the *single-node-management* profile, allows the visibility of the "top" *Connection* management, i.e. the end to end *Connection* (supporting a *Connectivity Service*) and its end points (*CEPs* and their supporting *NEPs*). The *NEPs* are restricted to the ones allowed by the *single-node-management* profile.
- *Notification-management*: Enables the notification management.
- *Detected-condition-management*: This profile, which requires the n*otification-management* profile, enables the alarm and threshold crossing alert management (restricted to the objects allowed by the other applied profiles).

These profiles are coded as *condition* stereotypes in the UML modules, and by *feature* and *if-feature* statements in the YANG modules.

The MEF Resource Model (MRM) Common includes TAPI management items which are

- currently used and extended by MEF 59 [12], MEF 72.1 [15] and MEF 83 [16], and
- potentially useful for the Resource management at PRESTO IRP.

The first set is isolated if neither of *full-topology-management*, *single-node-management* and *top-connection-management* profiles are supported. Concerning topology and connectivity, only the following objects are supported:

- *Service Interface Point,*
- *Connectivity Service, Connectivity Service End Point.*

Concerning OAM, only the following objects are supported:

- *Oam Service, Oam Service Point, Oam Job,*
- *Current Data, History Data,*
- *Pm Parameter Name and Pm Parameter Value enumerations.*

In case of *single-node-management* profile support, the *Mep* and *Mip* objects can be shown on monitored UNI/ENNI *NEPs*.

In case of *top-connection-management* profile support, the *Mep* and *Mip* objects can be shown on monitored UNI/ENNI *CEPs*.

The Path Computation Model is supported only if *full-topology-management* profile is supported.

7 Common Model

7.1 Diagrams

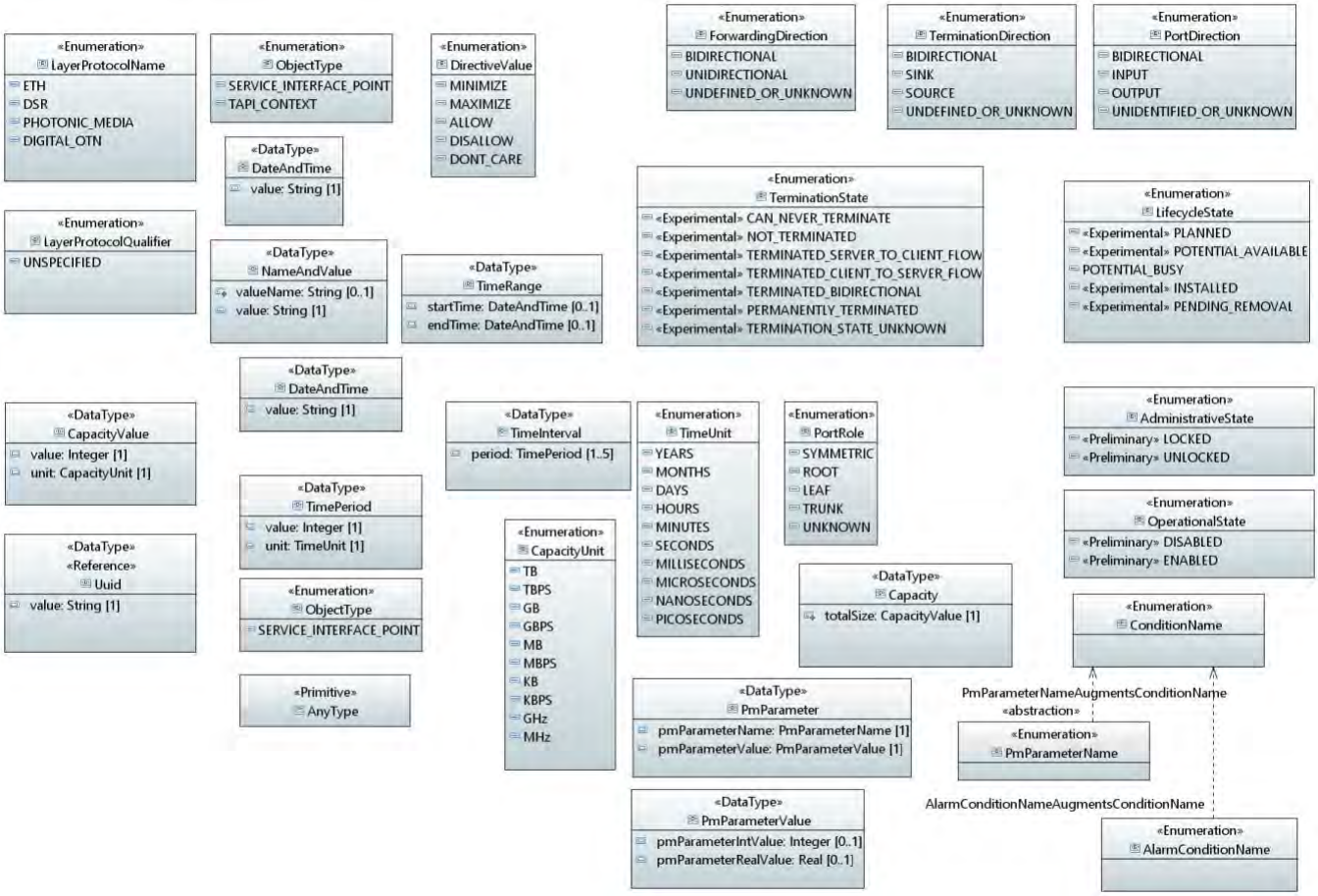

Figure 31 – *CommonDataTypes*

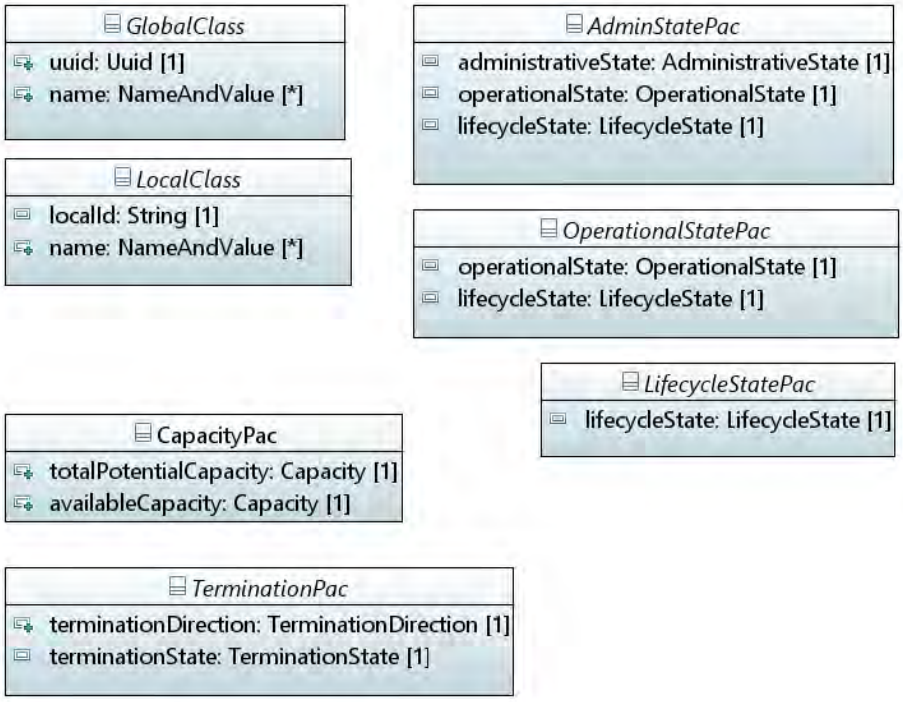

Figure 32 – *CommonPacs*

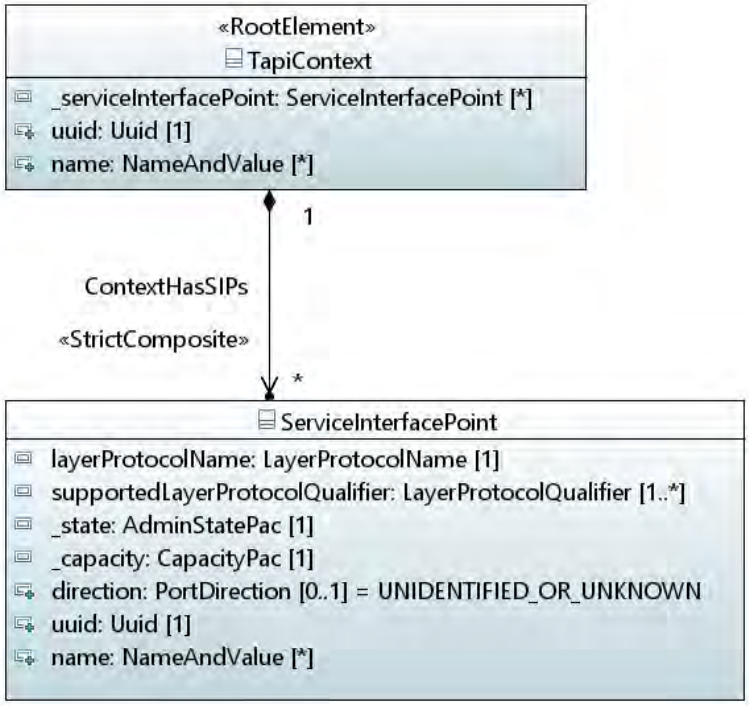

Figure 33 – *Context*

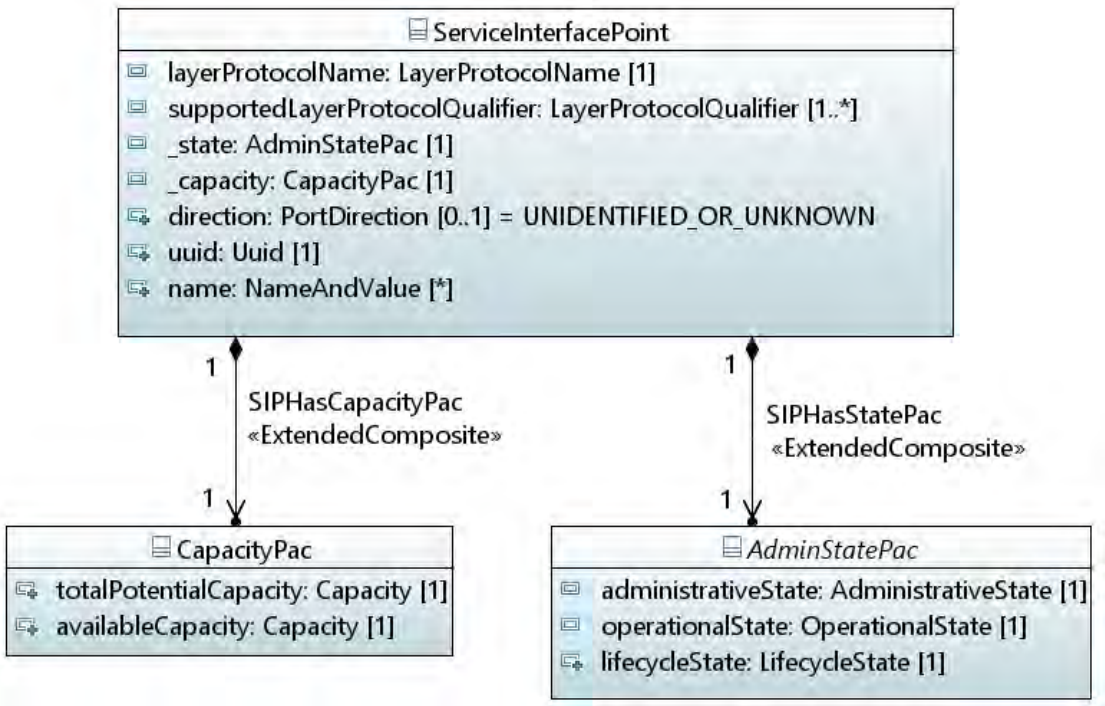

Figure 34 – *ServicePointDetails*

7.2 Classes

7.2.1 AdminStatePac

Provides state attributes that are applicable to an entity that can be administered. Such an entity also has operational and lifecycle aspects.

Attribute Name	Attribute Info			
administrativeState	**Type**: AdministrativeState	**Mult**: 1	RW	**Key**: No
	Description: The administration of managed objects operates independently of the operability and usage of managed objects and is described by the administrative state attribute. The administrative state is used by the operator to make a resource available for service, or to remove a resource from service.			
operationalState	**Type**: OperationalState	**Mult**: 1	R	**Key**: No
	Description: The operational state gives the information about the real capability of a resource to provide or not provide service.			
lifecycleState	**Type**: LifecycleState	**Mult**: 1	R	**Key**: No
	Description: Used to track the planned deployment, allocation to clients and withdrawal of resources.			

7.2.2 CapacityPac

Provides capacity related attributes.

Attribute Name	Attribute Info			
totalPotentialCapacity	**Type**: Capacity	**Mult**: 1	R	**Key**: No
	Description: An optimistic view of the capacity of the entity assuming that any shared capacity is available to be taken.			
availableCapacity	**Type**: Capacity	**Mult**: 1	R	**Key**: No
	Description: Capacity available to be assigned.			

7.2.3 GlobalClass

This class serves as the super class for all TAPI entities that can be directly retrieved by their ID. As such, these are first class entities and their ID is expected to be globally unique.

Attribute Name	Attribute Info			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

7.2.4 LifecycleStatePac

Provides state attributes for an entity that has lifecycle aspects only.

Attribute Name	Attribute Info			
lifecycleState	**Type**: LifecycleState	**Mult**: 1	R	**Key**: No
	Description: Used to track the planned deployment, allocation to clients and withdrawal of resources.			

7.2.5 LocalClass

This class serves as the super class for all TAPI entities that are ancillary of first class entities, i.e. their ID is not expected to be globally unique.

Attribute Name	Attribute Info			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1

Attribute Name	Attribute Info			
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

7.2.6 OperationalStatePac

Provides state attributes that are applicable to an entity that reflects operational aspects. Such an entity is expected to also have lifecycle aspects.

Attribute Name	Attribute Info			
operationalState	**Type**: OperationalState	**Mult**: 1	R	**Key**: No
	Description: The operational state gives the information about the real capability of a resource to provide or not provide service.			
lifecycleState	**Type**: LifecycleState	**Mult**: 1	R	**Key**: No
	Description: Used to track the planned deployment, allocation to clients and withdrawal of resources.			

7.2.7 ServiceInterfacePoint

A Service Interface Point represents the network-interface-facing aspects of the edge-port functions that access the forwarding capabilities provided by the Node. Hence it provides a limited, simplified view of interest to external clients (e.g. shared addressing, capacity, resource availability, etc.), that enable the clients to request connectivity without the need to understand the provider network internals.

Attribute Name	Attribute Info			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	R	**Key**: No
	Description: The layer protocol of the ServiceInterfacePoint (SIP). Usage of layerProtocolName [>1] in the ServiceInterfacePoint should be considered experimental.			
supportedLayerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1..*	R	**Key**: No
	Description: The supported sub-layer(s) or rate(s) of Layer Protocol.			
_state	**Type**: AdminStatePac	**Mult**: 1	RW	**Key**: No
	Description: The ServiceInterfacePoint (SIP) status information.			
_capacity	**Type**: CapacityPac	**Mult**: 1	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: The ServiceInterfacePoint (SIP) capacity information.			
direction	**Type**: PortDirection	**Mult**: 0..1	R	**Key**: No
	Description: The orientation of flow at the (conceptual) port of the potentially supported ConnectivityService(s). If direction attribute is missing the ServiceInterfacePoint (SIP) instance is to be intended as "BIDIRECTIONAL"			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

7.2.8 TapiContext

This object class represents the scope of control that a particular SDN controller has with respect to a particular network, (i.e., encompassing a designated set of interconnected (virtual) network elements). This class includes the list of Service Interface Points. This class can be augmented by specific contexts, e.g. topology context.

Attribute Name	Attribute Info			
_serviceInterfacePoint	**Type**: ServiceInterfacePoint	**Mult**: 0..*	RW	**Key**: No
	Description: The ServiceInterfacePoint (SIP) instances belonging to this context.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

7.2.9 TerminationPac

Specifies the direction and layer termination state of a termination entity, e.g. CEP, NEP.

Attribute Name	Attribute Info			
terminationDirection	**Type**: TerminationDirection	**Mult**: 1	R	**Key**: No
	Description: The overall directionality of the termination entity.			
terminationState	**Type**: TerminationState	**Mult**: 1	R	**Key**: No
	Description: Indicates whether the layer is terminated and if so how.			

7.3 Associations

7.3.1 ContextHasSIPs

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_serviceInterfacePoint	composite	Yes	ServiceInterfacePoint	0..*
context	none	No	TapiContext	1

7.3.2 SIPHasCapacityPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_capacity	composite	Yes	CapacityPac	1
serviceinterfacepoint	none	No	ServiceInterfacePoint	1

7.3.3 SIPHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	AdminStatePac	1
_serviceEndPoint	none	No	ServiceInterfacePoint	1

7.4 Abstractions

7.4.1 AlarmConditionNameAugmentsConditionName

Enumeration Augment.

7.4.2 PmParameterNameAugmentsConditionName

Enumeration Augment.

7.5 Data Types

7.5.1 Capacity

Information on capacity of a particular entity.

Attribute Name	Attribute Info			
totalSize	**Type**: CapacityValue	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: Total capacity of the entity. In case of bandwidthProfile, this is expected to be the same as the committedInformationRate.			

7.5.2 CapacityValue

The Capacity (Bandwidth) values that are applicable for digital layers.

Attribute Name	Attribute Info			
value	**Type**: Integer	**Mult**: 1	RW	**Key**: No
	Description: The specific value of the capacity.			
unit	**Type**: CapacityUnit	**Mult**: 1	RW	**Key**: No
	Description: The specific unit of measurement of the capacity.			

7.5.3 DateAndTime

This primitive type defines the date and time according to ISO 8601 with the following structure: yyyyMMddhhmmss.s[Z|{+|-}HHMm] where: yyyy 0000..9999 year MM 01..12 month dd 01..31 day hh 00..23 hour mm 00..59 minute ss 00..60 second (60 for leap seconds) s .0...9 tenth of second (set to .0 if EMS or NE cannot support this granularity) Z Z indicates UTC (rather than local time) {+|-} + or - delta from UTC HH 00..23 time zone difference in hours Mm 00..59 time zone difference in minutes.

Attribute Name	Attribute Info			
value	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: The specific value of the date and time.			

7.5.4 NameAndValue

A scoped name-value pair.

Attribute Name	Attribute Info			
valueName	**Type**: String	**Mult**: 0..1	RW	**Key**: yes – part: 1
	Description: The name of the value. Optional, the value need not to have a name.			

Attribute Name	Attribute Info			
value	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: The specific value.			

7.5.5 PmParameter

PM metric name and value.

Attribute Name	Attribute Info			
pmParameterName	**Type**: PmParameterName	**Mult**: 1	RW	**Key**: No
	Description: The name of the PM metric. Technology specific modules may define specific PM metrics.			
pmParameterValue	**Type**: PmParameterValue	**Mult**: 1	RW	**Key**: No
	Description: The value of the PM metric.			

7.5.6 PmParameterValue

PM metric value.

Attribute Name	Attribute Info			
pmParameterIntValue	**Type**: Integer	**Mult**: 0..1	RW	**Key**: No
	Description: Integer value, e.g. for counters.			
pmParameterRealValue	**Type**: Real	**Mult**: 0..1	RW	**Key**: No
	Description: Real value, e.g. for gauges.			

7.5.7 TimeInterval

Interval of time, duration. Q.821: The Interval attribute type indicates the time between occurrences of a given activity described by an instance of the Management Operations Schedule object class. The interval can be specified in seconds, minutes, hours, or days.

Attribute Name	Attribute Info			
period	**Type**: TimePeriod	**Mult**: 1..5	RW	**Key**: No

Attribute Name	Attribute Info
	Description: The specific interval of time. Each TimePeriod occurrence specifies a duration in years, months, days, hours, minutes etc. The 1..5 occurrences complies with Q.821. ITU-T Q.821 (02/2000): TimeInterval ::= SEQUENCE { day [0] INTEGER (0..31) DEFAULT 0, hour [1] INTEGER (0..23) DEFAULT 0, minute [2] INTEGER (0..59) DEFAULT 0, second [3] INTEGER (0..59) DEFAULT 0, msec [4] INTEGER (0..999) DEFAULT 0 } Examples: A duration of 1 hour, 15 minutes and 30 seconds should be coded as 3 occurrences of TimePeriod: - HOURS; 1 - MINUTES; 15 - SECONDS; 30 A duration of 1550 milliseconds as two occurrences of TimePeriod: - SECONDS; 1 - MILLISECONDS: 550

7.5.8 TimePeriod

Period of time.

Attribute Name	Attribute Info			
value	**Type**: Integer	**Mult**: 1	RW	**Key**: No
	Description: The specific value of the time period.			
unit	**Type**: TimeUnit	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: The unit of measurement of the time period.			

7.5.9 TimeRange

Range of time.

Attribute Name	Attribute Info			
startTime	**Type**: DateAndTime	**Mult**: 0..1	RW	**Key**: No
	Description: Date and time of the range start.			
endTime	**Type**: DateAndTime	**Mult**: 0..1	RW	**Key**: No
	Description: Date and time of the range end.			

7.5.10 Uuid

The univeral ID value where the mechanism for generation is defned by some authority not directly referenced in the structure. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6

Attribute Name	Attribute Info			
value	**Type**: String	**Mult**: 1	RW	**Key**: No

Attribute Name	Attribute Info
	Description: The specific value of the universal id.

7.6 Enumerations

7.6.1 AdministrativeState

The possible values of the administrativeState.

Contains Enumeration Literals:

- LOCKED:
  - Users are administratively prohibited from making use of the resource.
- UNLOCKED:
  - Users are allowed to use the resource.

7.6.2 AlarmConditionName

The alarm condition name, or alarm probable cause. This extensible enumeration can be augmented with specific alarm condition names in the other modules.

Contains Enumeration Literals: none

7.6.3 CapacityUnit

Units of measurement of the capacity.

Contains Enumeration Literals:

- TB:
  - Indicates that the integer CapacityValue is in TeraBytes
- TBPS:
  - Indicates that the integer CapacityValue is in Terabit-per-second
- GB:
  - Indicates that the integer CapacityValue is in GigaBytes
- GBPS:
  - Indicates that the integer CapacityValue is in Gigabit-per-second
- MB:
  - Indicates that the integer CapacityValue is in MegaBytes
- MBPS:
  - Indicates that the integer CapacityValue is in Megabit-per-second
- KB:
  - Indicates that the integer CapacityValue is in KiloBytes
- KBPS:
  - Indicates that the integer CapacityValue is in Kilobit-per-second
- GHz:
  - Indicates that the integer CapacityValue is in gigahertz (spectrum)
- MHz:

- Indicates that the integer CapacityValue is in megahertz (spectrum)

7.6.4 ConditionName

The Condition names. This extensible enumeration can be augmented with - specific PM metric names - specific alarm condition names, or alarm probable causes

Contains Enumeration Literals: none

7.6.5 DirectiveValue

Types of directives.

Contains Enumeration Literals:

- MINIMIZE:
  - Directive to minimize.
- MAXIMIZE:
  - Directive to maximize.
- ALLOW:
  - Directive to allow.
- DISALLOW:
  - Directive to disallow
- DONT_CARE:
  - Directive is do not care.

7.6.6 ForwardingDirection

The directionality of a forwarding entity, e.g. Link, ConnectivityService, Connection, PathComputationService, Path.

Contains Enumeration Literals:

- BIDIRECTIONAL:
  - The fowarding entity supports BIDIRECTIONAL flows at all its (conceptual) ports (i.e. all ports have both an INPUT flow and an OUTPUT flow defined).
- UNIDIRECTIONAL:
  - The forwarding entity has (conceptual) ports that are either INPUT or OUTPUT. It has no BIDIRECTIONAL (conceptual) ports.
- UNDEFINED_OR_UNKNOWN:
  - Not a normal state. The system is unable to determine the correct value.

7.6.7 LayerProtocolName

Provides a controlled list of layer protocol names and indicates the naming authority. Note that it is expected that attributes will be added to this structure to convey the naming authority name, the name of the layer protocol using a human readable string and any particular standard reference.

Contains Enumeration Literals:

- ETH:

- Models the ETH layer as per ITU-T G.8010
- DSR:
  - Models a Digital Signal of an unspecified rate (Layer 1 coding functions). This value can be used when the intent is to represent a generic digital layer signal without making any statement on its format or overhead (processing) capabilities.
- PHOTONIC_MEDIA:
  - Models the optical signal and media channel layer as per ITU-T G.807
- DIGITAL_OTN:
  - Models the OTU/ODU OTN digital layers as per ITU-T G.872

7.6.8 LayerProtocolQualifier

This enumeration is used to qualify the sub-layers (if applicable) for a specific LayerProtocol. This extensible enumeration can be augmented with layer-specific values in the respective technology-specific modules.

Contains Enumeration Literals:

- UNSPECIFIED:
  - No sub-layer is specified.

7.6.9 LifecycleState

The possible values of the lifecycleState.

Contains Enumeration Literals:

- PLANNED:
  - The resource is planned but is not present in the network.
- POTENTIAL_AVAILABLE:
  - The supporting resources are present in the network but are shared with other clients; or require further configuration before they can be used; or both. When a potential resource is configured and allocated to a client it is moved to the INSTALLED state for that client. If the potential resource has been consumed (e.g. allocated to another client) it is moved to the POTENTIAL_BUSY state for all other clients.
- POTENTIAL_BUSY:
  - The supporting resources are present in the network but have been allocated to other clients.
- INSTALLED:
  - The resource is present in the network and is capable of providing the service expected.
- PENDING_REMOVAL:
  - The resource has been marked for removal.

7.6.10 ObjectType

The list of TAPI Global Object Class types on which Notification signals can be raised. This extensible enumeration can be augmented with specific object types/classes in the other modules.

Contains Enumeration Literals:

- SERVICE_INTERFACE_POINT:

- The ServiceInterfacePoint (SIP) class.
- TAPI_CONTEXT:
  - The TapiContext class.

7.6.11 OperationalState

The possible values of the operationalState.

Contains Enumeration Literals:

- DISABLED:
  - The resource is unable to meet the SLA of the user of the resource. If no (explicit) SLA is defined the resource is disabled if it is totally inoperable and unable to provide service to the user.
- ENABLED:
  - The resource is partially or fully operable and available for use.

7.6.12 PmParameterName

The PM metric names. This extensible enumeration can be augmented with specific PM metric names in the other modules.

Contains Enumeration Literals: none

7.6.13 PortDirection

The orientation of flow at the (conceptual) port of a forwarding entity, e.g. Link, ConnectivityService, Connection, PathComputationService, Path, VirtualNetworkService.

Contains Enumeration Literals:

- BIDIRECTIONAL:
  - The Port has both an INPUT flow and an OUTPUT flow defined.
- INPUT:
  - The port only has definition for a flow into the forwarding entity, (i.e. an ingress flow of e.g Link or Connection, hence an egress flow of NEP or CEP, CSEP etc.).
- OUTPUT:
  - The port only has definition for a flow out of the forwarding entity (i.e. an egress flow of e.g. Link or Connection, hence an ingress flow of NEP or CEP, CSEP etc.).
- UNIDENTIFIED_OR_UNKNOWN:
  - Not a normal state. The system is unable to determine the correct value.

7.6.14 PortRole

The role of a (conceptual) port of a forwarding entity, e.g. Link, ConnectivityService, Connection, PathComputationService, Path, VirtualNetworkService.

Contains Enumeration Literals:

- SYMMETRIC:

  - A port that can exchange flows (e.g. distinct packet flows) with any other port(s) in a forwarding entity. The SYMMETRIC role applies to point to point and multipoint to multipoint connection schemes.
- ROOT:
  - A port that can exchange flows (e.g. distinct packet flows) with any other port(s) in a forwarding entity. The ROOT role is unique to the Rooted Multipoint connection scheme.
- LEAF:
  - A port that can only exchange flows (e.g. distinct packet flows) with any other ROOT or TRUNK port(s) in a forwarding entity. The LEAF role is unique to the Rooted Multipoint connection scheme.
- TRUNK:
  - The TRUNK role is unique to the ENNI involved in a Rooted Multipoint connection scheme. It provides a way to extend the concept of ROOT and LEAF bidirectionally across the ENNI without having to create multiple ports (Leaves and Roots) and hairpinning from one to the other.
- UNKNOWN:
  - Not a normal state. The system is unable to determine the correct value.

7.6.15 TerminationDirection

The directionality of a termination entity, e.g. CEP, NEP.

Contains Enumeration Literals:

- BIDIRECTIONAL:
  - A termination entity with both SINK and SOURCE flows.
- SINK:
  - The flow is up the layer stack from the server side to the client side. Considering an example of a termination function within the termination entity, a SINK flow: 1) will arrive at the base of the termination function (the server side) where it is essentially at an INPUT to the termination function, 2) then will be decoded and deconstructed, 3) then the relevant parts of the flow will be sent out of the termination function (the client side) where it is essentially at an OUTPUT from the termination function. A SINK termination function is one that only supports a SINK flow. A SINK termination function can be bound to an OUTPUT (conceptual) port of a forwarding entity, e.g. Link, ConnectivityService, Connection, PathComputationService, Path.
- SOURCE:
  - The flow is down the layer stack from the client side to the server side. Considering an example of a termination function within the termination entity, a SOURCE flow: 1) will arrive at the top of the termination function (the client side) where it is essentially at an INPUT to the termination function, 2) then will be assembled with various overheads etc and will be coded, 3) then the coded form of the assembly of flow will be sent out of the termination function (the server side) where it is essentially at an OUTPUT from the termination function. A SOURCE termination is one that only supports a SOURCE flow. A SOURCE termination can be bound to an INPUT (conceptual) port of a forwarding entity, e.g. Link, ConnectivityService, Connection, PathComputationService, Path.
- UNDEFINED_OR_UNKNOWN:
  - Not a normal state. The system is unable to determine the correct value.

7.6.16 TerminationState

Provides support for the range of behaviours and specific states that the termination function of a termination entity can take with respect to the termination of the signal.

Contains Enumeration Literals:

- CAN_NEVER_TERMINATE:
  - A non-flexible case that can never be terminated.
- NOT_TERMINATED:
  - A flexible termination that can terminate but is currently not terminated.
- TERMINATED_SERVER_TO_CLIENT_FLOW:
  - A flexible termination that is currently terminated for server to client flow only.
- TERMINATED_CLIENT_TO_SERVER_FLOW:
  - A flexible termination that is currently terminated for client to server flow only.
- TERMINATED_BIDIRECTIONAL:
  - A flexible termination that is currently terminated in both directions of flow.
- PERMANENTLY_TERMINATED:
  - A non-flexible termination that is always terminated (in both directions of flow for a bidirectional case and in the one direction of flow for both unidirectional cases).
- TERMINATION_STATE_UNKNOWN:
  - Not a normal state. The system is unable to determine the correct value.

7.6.17 TimeUnit

Units of measurement of the time.

Contains Enumeration Literals:

- YEARS:
- MONTHS:
- DAYS:
- HOURS:
- MINUTES:
- SECONDS:
- MILLISECONDS:
- MICROSECONDS:
- NANOSECONDS:
- PICOSECONDS:

7.7 Primitives

7.7.1 AnyType

This primitive represents the "any data" mechanism.

7.7.2 BinaryType

Represents any binary data, i.e., a sequence of octets. A binary type can be restricted by a length which defines the number of octets it contains.

7.7.3 MacAddress

Pattern: "[0-9a-fA-F]{2}(-[0-9a-fA-F]{2}){5}" Description: "The mac-address type represents a MAC address in the canonical format and hexadecimal format specified by IEEE Std 802. The canonical representation uses lowercase characters. The hexadecimal representation uses uppercase characters."

7.7.4 Timeticks

Type uint32. This type represents a non-negative integer that represents the time, modulo 2^32 (4294967296 decimal), in hundredths of a second between two epochs.

8 Topology Model

8.1 Diagrams

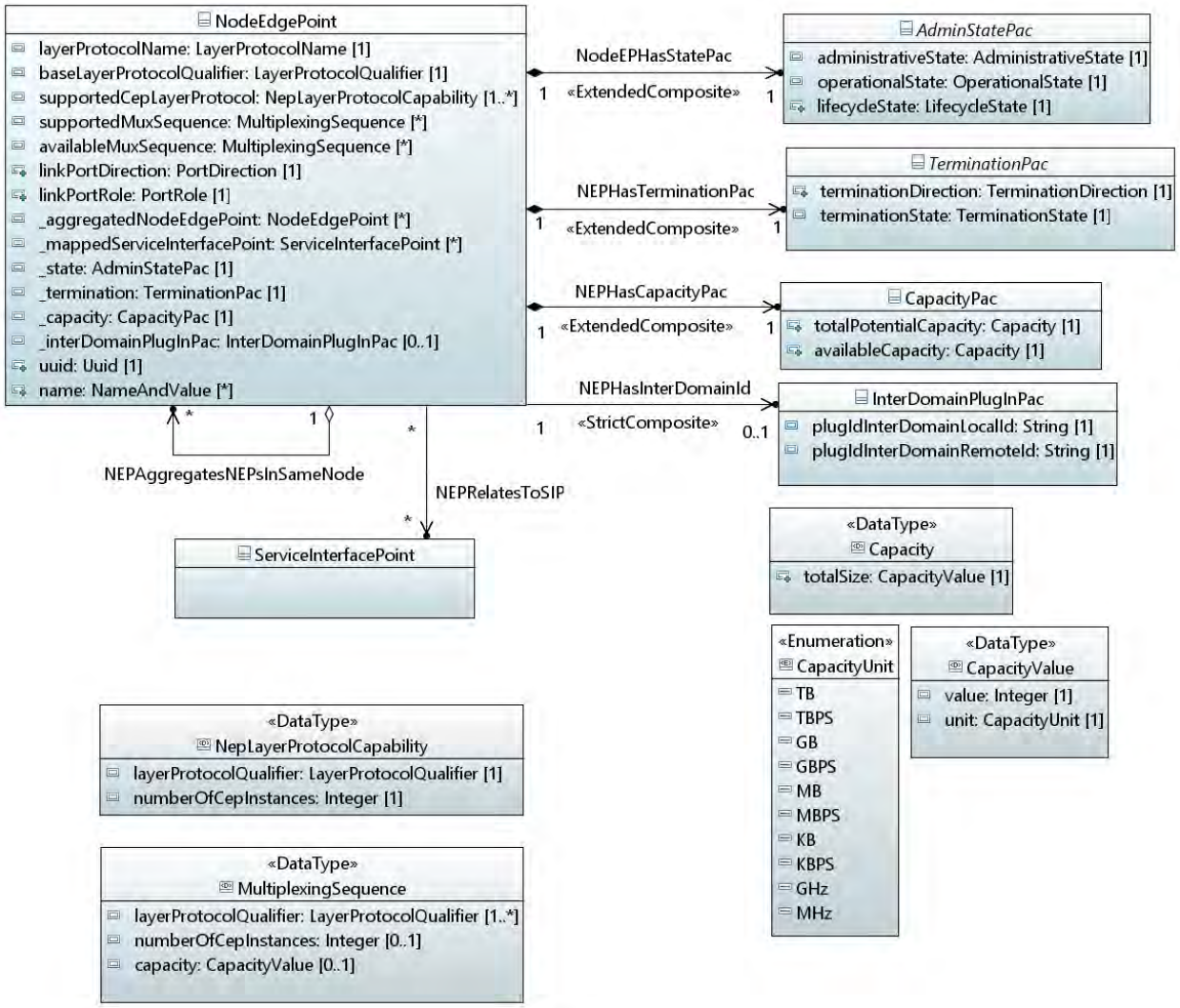

Figure 35 – *EdgePointDetails*

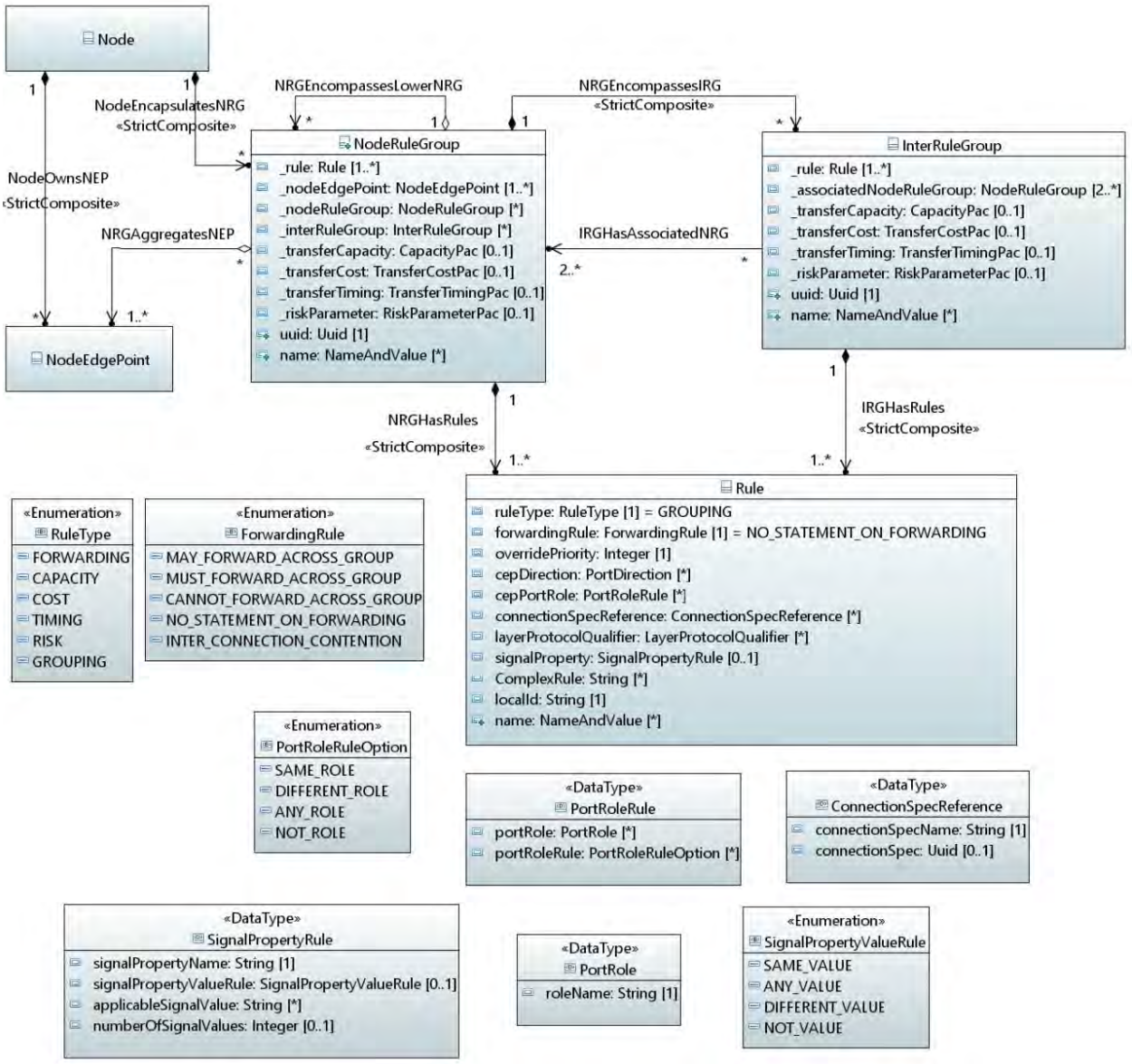

Figure 36 – *NodeConstraints*

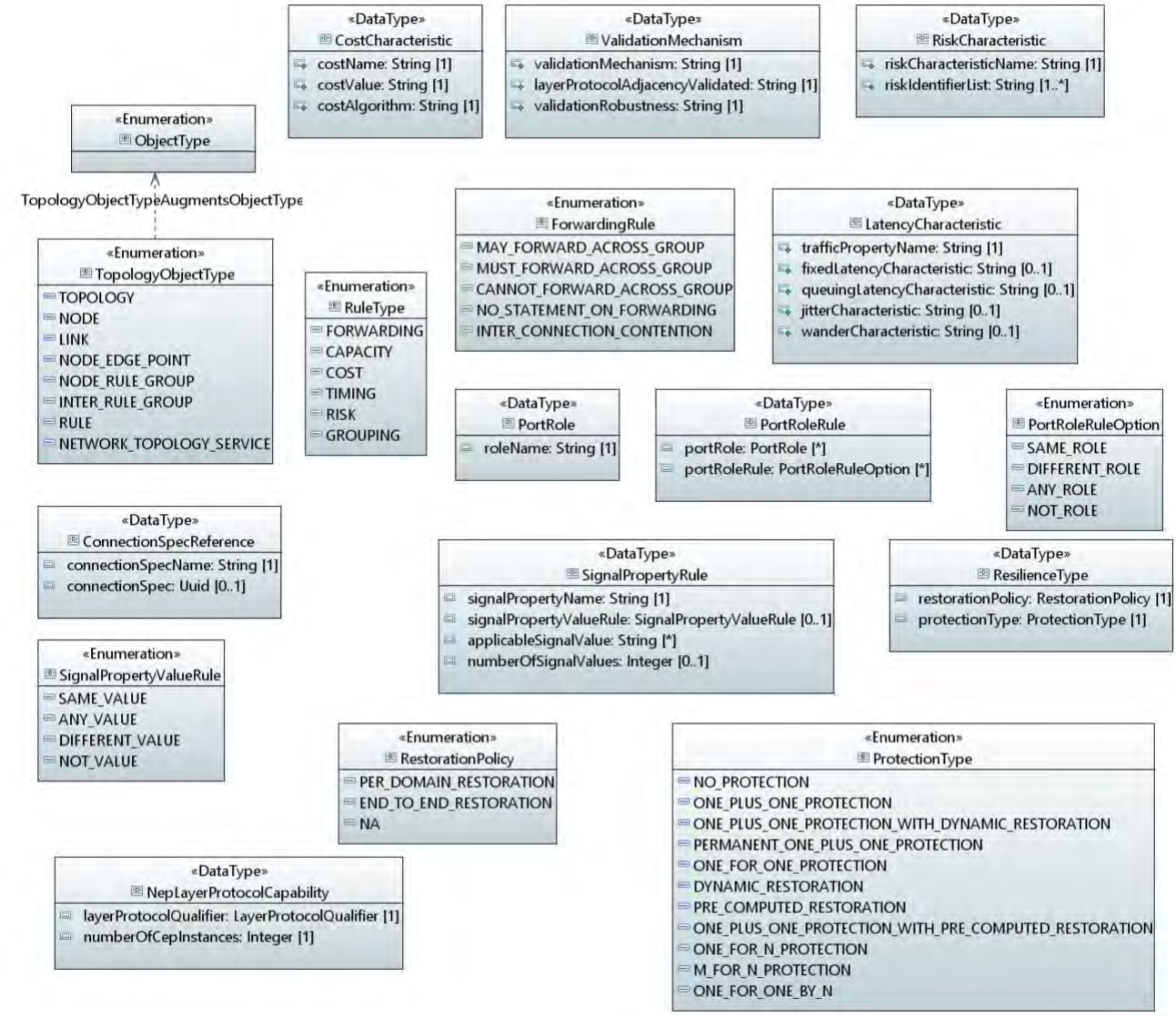

Figure 37 – *TopologyDataTypes*

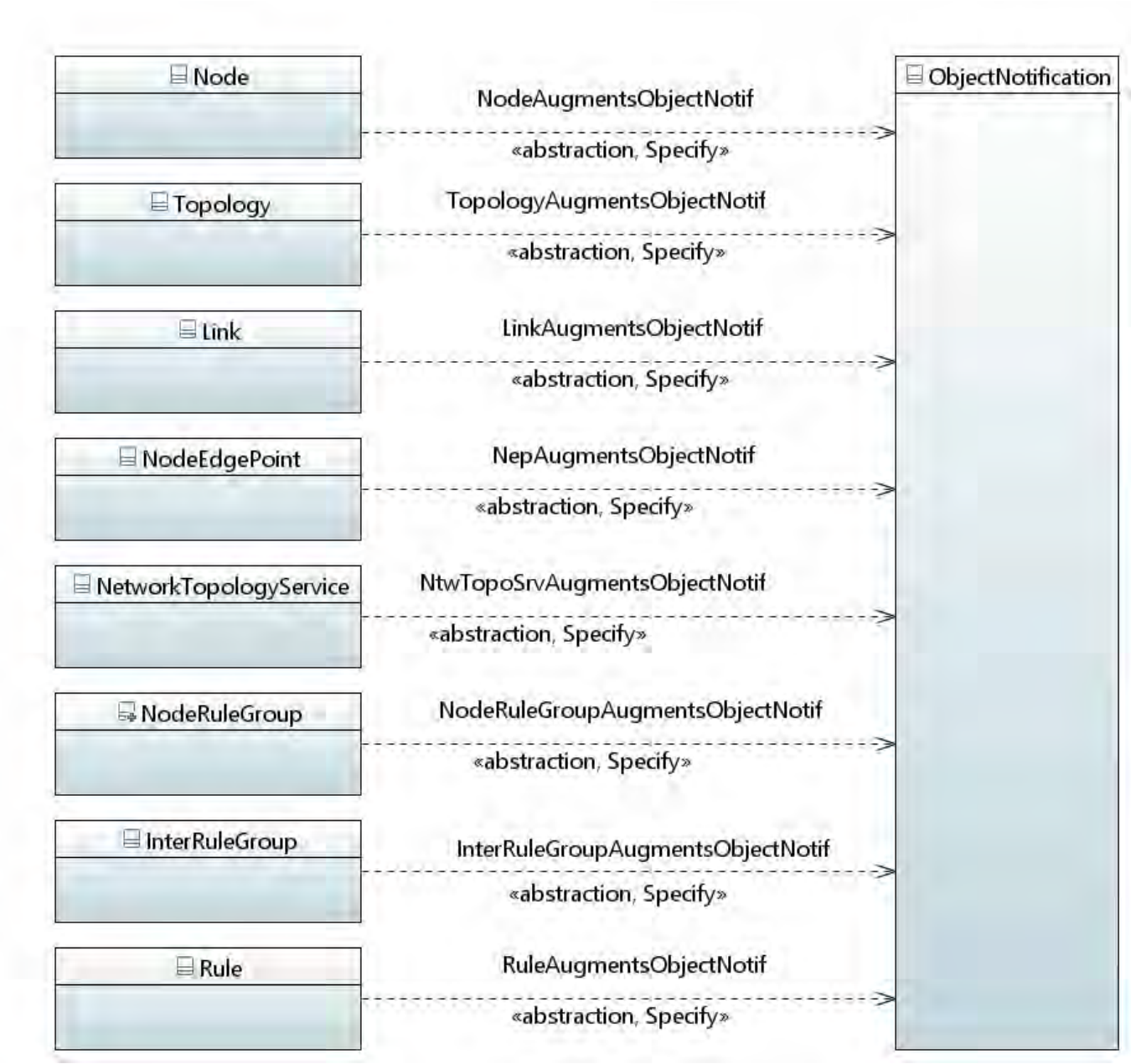

Figure 38 – *TopologyNotif*

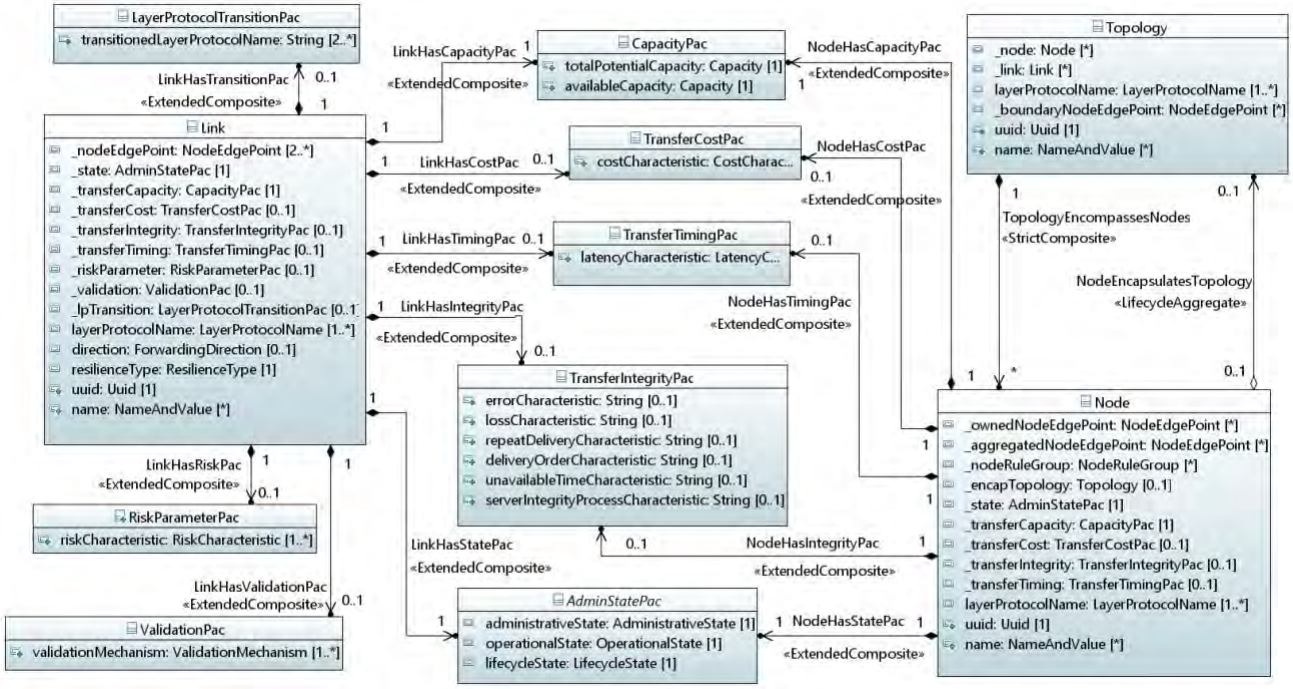

Figure 39 – *TopologyServiceDetails*

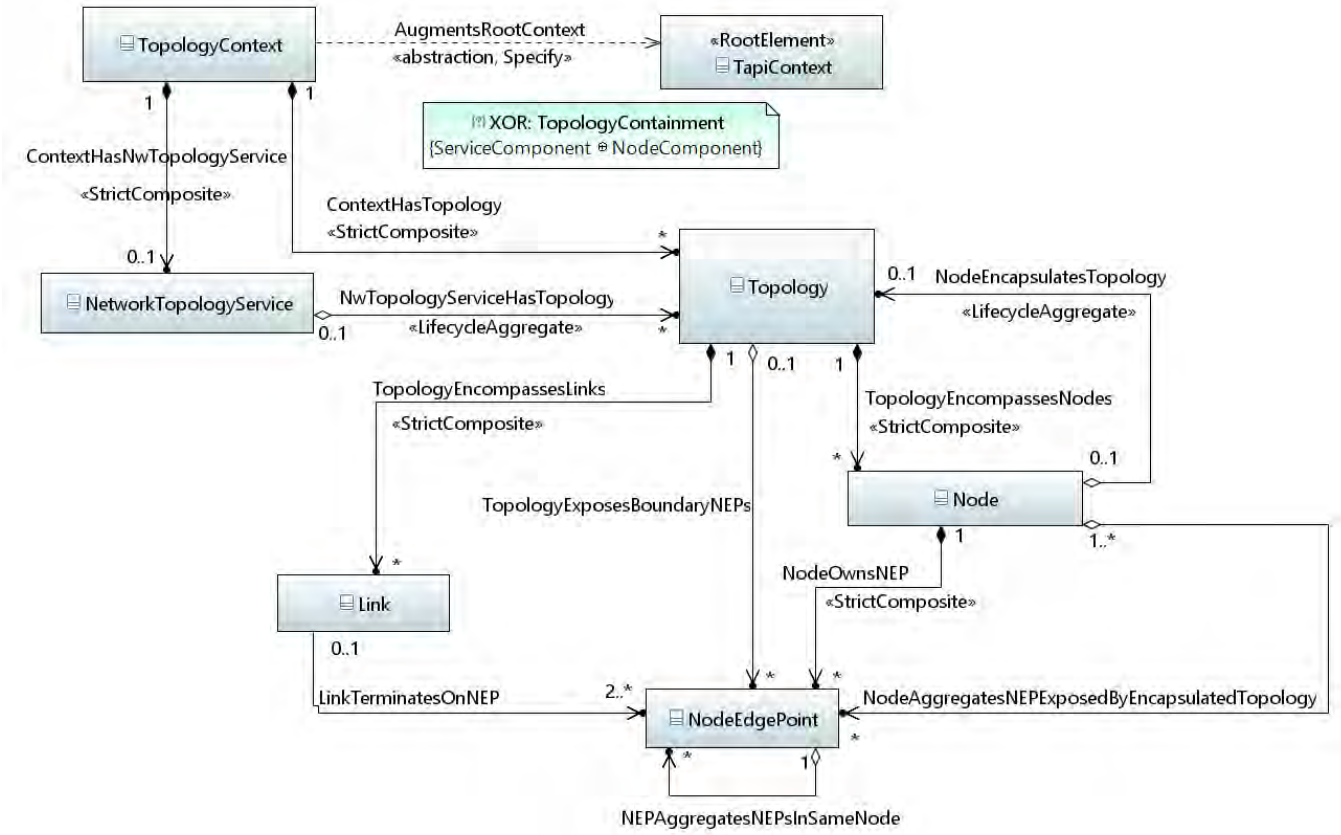

Figure 40 – *TopologyServiceSkeleton*

8.2 Classes

8.2.1 InterDomainPlugInPac

NEP at ENNI shall include an ENNI identifier (inter domain plug id) which must be unique in both the connected managed domains, to support the automatic discovery of interdomain links between E-NNI interfaces of e.g. different network providers. The inter domain plug id can be based on OTN technology (OTU or ODU Trail Trace Identifier, SAPI). ITU-T G.709: The access point identifier shall consist of a three-character international segment and a twelve-character national segment coded according to [ITU-T T.50]. The international segment field provides a three-character ISO 3166 geographic/political country code (G/PCC). The country code shall be based on the three-character uppercase alphabetic ISO 3166 country code. The national segment field consists of two subfields: the ITU carrier code (ICC) followed by a unique access point code (UAPC). The ITU carrier code is assigned to a network operator/service provider and shall consist of 1-6 left-justified characters, alphabetic, or leading alphabetic with trailing numeric [e.g., "USATELCORuapc"].

Attribute Name	Attribute Info			
plugIdInterDomainLocalId	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: Source Access Point Identifier (SAPI) in TxTI. G.709 TxTI: string[64 bytes]: The Trail Trace Identifier (TTI) information, provisioned by the managing system at the termination source, to be placed in the TTI overhead position of the source of a trail for transmission.			
plugIdInterDomainRemoteId	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: Expected Source Access Point Identifier (ExSAPI). G.709 ExSAPI: Provisioned by the managing system, to be compared with the TTI accepted (AcTI) at the overhead position of the sink for the purpose of checking the integrity of connectivity. AcTI: string [64 bytes] The Trail Trace Identifier (TTI) information recovered (Accepted) from the TTI overhead position at the sink of a trail.			

8.2.2 InterRuleGroup

Rules that apply between groups of NodeEdgePoint (NEP) instances.

Attribute Name	Attribute Info			
_rule	**Type**: Rule	**Mult**: 1..*	R	**Key**: No
	Description: The list of rules of the InterRuleGroup.			
_associatedNodeRuleGroup	**Type**: NodeRuleGroup	**Mult**: 2..*	R	**Key**: No
	Description: The NodeRuleGroups that the InterRuleGroup constrains interconnection between. The CEPs of the NEPs of a referenced NodeRuleGroup can interconnect to the CEPs of the NEPs of another referenced NodeRuleGroup constrained by the rules of the InterRuleGroup.			
_transferCapacity	**Type**: CapacityPac	**Mult**: 0..1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: The rule relates to transfer capacity constraint. The connections, matching the properties of the rule, formed between the NEPs, governed by the group, must abide by the transfer capacity statement. The capacity is assumed to be maximum allowed.			
_transferCost	**Type**: TransferCostPac	**Mult**: 0..1	R	**Key**: No
	Description: The rule relates to transfer cost constraint. The connections, matching the properties of the rule, formed between the NEPs, governed by the group, will acquire the cost stated. Several rules may state different costs for the same configuration. This indicated that there is underlying complexity that is not being fully expressed at the level of abstraction of the rules.			
_transferTiming	**Type**: TransferTimingPac	**Mult**: 0..1	R	**Key**: No
	Description: The rule relates to transfer timimg constraint. The connections, matching the properties of the rule, formed between the NEPs, governed by the group, will acquire the timing penalty stated. Several rules may state different timing penalties for the same configuration. This indicated that there is underlying complexity that is not being fully expressed at the level of abstraction of the rules.			
_riskParameter	**Type**: RiskParameterPac	**Mult**: 0..1	R	**Key**: No
	Description: The rule relates to risk constraints. The connections, matching the properties of the rule, formed between the NEPs, governed by the group, will acquire the risk penalty stated. Several rules may state different risk penalties for the same configuration. This indicated that there is underlying complexity that is not being fully expressed at the level of abstraction of the rules.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

8.2.3 LayerProtocolTransitionPac

Relevant for a Link that is formed by abstracting one or more termination entities (in a stack) to focus on the flow and deemphasize the protocol transformation. This abstraction is relevant when considering multi-layer routing and the protocol transformation is not too complex, e.g. there is not multiplexing. This Pac provides the relevant abstractions of the embedded termination entities: The layer protocols of the embedded termination entities and the order of their application to the signal is still relevant and need to be accounted for. Links that included details in this Pac are often referred to as Transitional Links.

Attribute Name	Attribute Info			
transitionedLayerProtocolName	**Type**: String	**Mult**: 2..*	RW	**Key**: No

Attribute Name	Attribute Info
	Description: Provides the ordered structure of layer protocol transitions encapsulated in the Link. The list starts with the client side as the first entry and includes all layer-protocol names (hence the smallest number is 2 as otherwise the Link is not transitional). The ordering relates also to the (conceptual) port role (which emphasizes the orientation).

8.2.4 Link

A Link is a topological entity which is an abstract representation of the effective adjacency between two or more Node instances (specifically NodeEdgePoint instances) in a Topology.

Attribute Name	Attribute Info			
_nodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 2..*	R	**Key**: No
	Description: The NEPs connected by the Link.			
_state	**Type**: AdminStatePac	**Mult**: 1	R	**Key**: No
	Description: The Link status information.			
_transferCapacity	**Type**: CapacityPac	**Mult**: 1	R	**Key**: No
	Description: The Link capacity.			
_transferCost	**Type**: TransferCostPac	**Mult**: 0..1	R	**Key**: No
	Description: The transfer cost of the Link.			
_transferIntegrity	**Type**: TransferIntegrityPac	**Mult**: 0..1	R	**Key**: No
	Description: The transfer integrity of the Link.			
_transferTiming	**Type**: TransferTimingPac	**Mult**: 0..1	R	**Key**: No
	Description: The transfer timing of the Link.			
_riskParameter	**Type**: RiskParameterPac	**Mult**: 0..1	R	**Key**: No
	Description: The risk parameters of the Link.			
_validation	**Type**: ValidationPac	**Mult**: 0..1	R	**Key**: No
	Description: The validation mechanisms of the Link.			

Attribute Name	Attribute Info			
_lpTransition	**Type**: LayerProtocolTransitionPac	**Mult**: 0..1	R	**Key**: No
	Description: The information on encapsulated termination functions, applicable in case of Transitional Link.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1..*	R	**Key**: No
	Description: The layer protocol(s) of the Link.			
direction	**Type**: ForwardingDirection	**Mult**: 0..1	R	**Key**: No
	Description: The directionality of the Link.			
resilienceType	**Type**: ResilienceType	**Mult**: 1	RW	**Key**: No
	Description: The underlying resilience type of the Link.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

8.2.5 NetworkTopologyService

A NetworkTopologyService represents an "intent-like" request for topology related provisioning, for future developments. The NetworkTopologyService is a container for topology request details and is distinct from the Topology that realize the request.

Attribute Name	Attribute Info			
_topology	**Type**: Topology	**Mult**: 0..*	R	**Key**: No
	Description: The Topology instance(s) tracking the state of the allocated resources for the support of the NetworkTopologyService.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1

Attribute Name	Attribute Info			
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

8.2.6 Node

The Node is a topological entity which is an abstract representation of the forwarding capabilities (of transport characteristic information) of a particular set of network resources. It is described in terms of the aggregation of set of ports (NodeEdgePoint) belonging to those network resources and the potential to enable forwarding of information between those edge ports. At the lowest level of recursion, a Node may represent a switch matrix (i.e., a fabric) in an equipment.

Attribute Name	Attribute Info			
_ownedNodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 0..*	R	**Key**: No
	Description: The NEPs belonging to / owned by this Node. By convention, only the Node instances at the lowest partitioning level "own" the NEPs. In other words, each and every NEP instance is owned by a Node at the lowest partitioning level.			
_aggregatedNodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 0..*	R	**Key**: No **Condition**: full-topology-management
	Description: The NEPs aggregated by this Node. By convention, only the Node instances which are not at the lowest partitioning level "aggregate" the NEPs. In other words, each and every NEP instance is owned by a Node at the lowest partitioning level. A subset of NEP instances may be aggregated by Nodes at higher partitioning levels.			
_nodeRuleGroup	**Type**: NodeRuleGroup	**Mult**: 0..*	RW	**Key**: No **Condition**: full-topology-management
	Description: The Node rules applicable to this Node.			
_encapTopology	**Type**: Topology	**Mult**: 0..1	R	**Key**: No **Condition**: full-topology-management
	Description: A Node may encapsulate one Topology instance, which in turn encompasses Nodes at lower partitioning level.			
_state	**Type**: AdminStatePac	**Mult**: 1	R	**Key**: No
	Description: The Node status information.			

Attribute Name	Attribute Info			
_transferCapacity	**Type**: CapacityPac	**Mult**: 1	R	**Key**: No **Condition**: full-topology-management
	Description: The transfer capacity of the Node.			
_transferCost	**Type**: TransferCostPac	**Mult**: 0..1	R	**Key**: No **Condition**: full-topology-management
	Description: The transfer cost of the Node.			
_transferIntegrity	**Type**: TransferIntegrityPac	**Mult**: 0..1	R	**Key**: No **Condition**: full-topology-management
	Description: The transfer integrity of the Node.			
_transferTiming	**Type**: TransferTimingPac	**Mult**: 0..1	R	**Key**: No **Condition**: full-topology-management
	Description: The transfer timing of the Node.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1..*	R	**Key**: No
	Description: The layer protocol(s) of the (multi-layer) Node.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

8.2.7 NodeEdgePoint

The NodeEdgePoint (NEP) is a topological entity which represents the ingress-egress edge-port functions that access the forwarding capabilities provided by the Node. Hence it provides an encapsulation of addressing, mapping, termination, adaptation and OAM functions of one or more transport layers (including circuit and packet forms) performed at the entry and exit points of the Node.

Attribute Name	Attribute Info			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: The layer protocol of the NodeEdgePoint (NEP).			
baseLayerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol qualifier at the bottom of supported stack.			
supportedCepLayerProtocol	**Type**: NepLayerProtocolCapability	**Mult**: 1..*	RW	**Key**: No
	Description: The potentially supported protocols and flows. In ITU-T terms, the potentially supported adaptation and termination functions.			
supportedMuxSequence	**Type**: MultiplexingSequence	**Mult**: 0..*	RW	**Key**: No
	Description: More detailed description of (potential) capability than "supportedCepLayerProtocol".			
availableMuxSequence	**Type**: MultiplexingSequence	**Mult**: 0..*	RW	**Key**: No
	Description: More detailed description of available capability than "supportedCepLayerProtocol".			
linkPortDirection	**Type**: PortDirection	**Mult**: 1	R	**Key**: No **Condition**: full-topology-management
	Description: The orientation of flow at the (conceptual) port of the associated Link.			
linkPortRole	**Type**: PortRole	**Mult**: 1	R	**Key**: No **Condition**: full-topology-management
	Description: The role of the (conceptual) port of the associated Link.			
_aggregatedNodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 0..*	R	**Key**: No **Condition**: full-topology-management
	Description: A NodeEdgePoint (NEP) instance may aggregate one or more other NEP instances for e.g. pooling purposes, when a set of NEP instances are equivalent for usage.			
_mappedServiceInterfacePoint	**Type**: ServiceInterfacePoint	**Mult**: 0..*	R	**Key**: No
	Description: A NodeEdgePoint (NEP) may be associated to a ServiceInterfacePoint (SIP), i.e. when the NEP is the resource oriented view of a SIP. NEP mapped to more than one SIP (slicing/virtualizing) or a SIP mapped to more than one NEP (load balancing/resilience) should be considered experimental.			
_state	**Type**: AdminStatePac	**Mult**: 1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: The NodeEdgePoint (NEP) status information.			
_termination	**Type**: TerminationPac	**Mult**: 1	R	**Key**: No
	Description: Termination direction and termination state of the NodeEdgePoint (NEP).			
_capacity	**Type**: CapacityPac	**Mult**: 1	R	**Key**: No
	Description: The NodeEdgePoint (NEP) capacity information.			
_interDomainPlugInPac	**Type**: InterDomainPlugInPac	**Mult**: 0..1	RW	**Key**: No
	Description: ENNI Identifier.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

8.2.8 NodeRuleGroup

Rules that apply to a group of NodeEdgePoint (NEP) instances.

Attribute Name	Attribute Info			
_rule	**Type**: Rule	**Mult**: 1..*	R	**Key**: No
	Description: The list of rules of the NodeRuleGroup.			
_nodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 1..*	R	**Key**: No
	Description: NEPs and their client CEPs that the rules apply to.			
_nodeRuleGroup	**Type**: NodeRuleGroup	**Mult**: 0..*	R	**Key**: No

Attribute Name	Attribute Info			
	Description: NodeRuleGroups may be nested such that finer grained rules may be applied. A nested rule group should have a subset of the NEPs of the superior rule group.			
_interRuleGroup	**Type**: InterRuleGroup	**Mult**: 0..*	R	**Key**: No
	Description: Nested NodeRuleGroups may have InterRuleGroups. The Superior NodeRuleGroup contains the nested NodeRuleGroups and their associated InterRuleGroups. This is equivalent to the Node-Topology hierarchy.			
_transferCapacity	**Type**: CapacityPac	**Mult**: 0..1	R	**Key**: No
	Description: The rule relates to transfer capacity constraint. The connections, matching the properties of the rule, formed between the NEPs, governed by the group, must abide by the transfer capacity statement. The capacity is assumed to be maximum allowed.			
_transferCost	**Type**: TransferCostPac	**Mult**: 0..1	R	**Key**: No
	Description: The rule relates to transfer cost constraint. The connections, matching the properties of the rule, formed between the NEPs, governed by the group, will acquire the cost stated. Several rules may state different costs for the same configuration. This indicated that there is underlying complexity that is not being fully expressed at the level of abstraction of the rules.			
_transferTiming	**Type**: TransferTimingPac	**Mult**: 0..1	R	**Key**: No
	Description: The rule relates to transfer timimg constraint. The connections, matching the properties of the rule, formed between the NEPs, governed by the group, will acquire the timing penalty stated. Several rules may state different timing penalties for the same configuration. This indicated that there is underlying complexity that is not being fully expressed at the level of abstraction of the rules.			
_riskParameter	**Type**: RiskParameterPac	**Mult**: 0..1	R	**Key**: No
	Description: The rule relates to risk constraints. The connections, matching the properties of the rule, formed between the NEPs, governed by the group, will acquire the risk penalty stated. Several rules may state different risk penalties for the same configuration. This indicated that there is underlying complexity that is not being fully expressed at the level of abstraction of the rules.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

8.2.9 RiskParameterPac

The risk characteristics of a topological entity (e.g. the Link) come directly from the underlying physical realization. The risk characteristics propagate from the physical realization to the client and from the server layer to the client layer, this propagation may be modified by protection. A topological entity may suffer degradation or failure as a result of a problem in a part of the underlying realization. The realization can be partitioned into segments which have some relevant common failure modes. There is a risk of failure/degradation of each segment of the underlying realization. Each segment is a part of a larger physical/geographical unit that behaves as one with respect to failure (i.e. a failure will have a high probability of impacting the whole unit (e.g. all cables in the same duct). Disruptions to that larger physical/geographical unit will impact (cause failure/errors to) all topological entities that use any part of that larger physical/geographical entity. Any topological entity that uses any part of that larger physical/geographical unit will suffer impact and hence each topological entity shares risk. The identifier of each physical/geographical unit that is involved in the realization of each segment of a topological entity can be listed in the RiskParameter_Pac of that topological entity. A segment has one or more risk characteristic. Shared risk between two topological entities compromises the integrity of any solution that use one of those topological entity as a backup for the other. Where two topological entities have a common risk characteristic they have an elevated probability of failing simultaneously compared to two topological entities that do not share risk characteristics.

Attribute Name	Attribute Info			
riskCharacteristic	**Type**: RiskCharacteristic	**Mult**: 1..*	R	**Key**: No
	Description: A list of risk characteristics for consideration in an analysis of shared risk. Each element of the list represents a specific risk consideration.			

8.2.10 Rule

Single complex rule statement. A Node with no rule group has no restrictions and is essentially May/Any. A NodeRuleGroup constrains the CEP connectability in the Node. A Connection from a CEP/NEP must abide by all rules that relate to that CEP/NEP. Rules that are for a particular layerProtocolQualifier, connectionSpecReference, cepPortRole and cepDirection combination must be abided by in combination as dictated by overridePriority. If a particular connectionSpecReference does not have any rule statements then it is not supported and connections of that type are not possible within the rule group. If a particular cepPortRole of a particular connectionSpecReference does not have any rule statements then it is not supported and connections of that connectionSpecReference (type) cannot have that cepPortRole for CEPs from NEPs in that rule group. If a particular cepDirection for a particular connectionSpecReference does not have any rule statements then it is not supported and connections of that connectionSpecReference (type) cannot have that cepPortDirection for CEPs from NEPs in that rule group. Rules that are for different layerProtocolQualifiers or connectionSpecReferences are independent and provide options for Connection in the NodeRuleGroup. Some rules may apply to multiple connectionSpecReferences and all cepPortRoles and all cepDirections.

Attribute Name	Attribute Info			
ruleType	**Type**: RuleType	**Mult**: 1	R	**Key**: No
	Description: The focus of the rule.			
forwardingRule	**Type**: ForwardingRule	**Mult**: 1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: Rule that restricts the creation/deletion of a Connection between points in the NodeRuleGroup or related by the InterRuleGroup between NodeRuleGroups.			
overridePriority	**Type**: Integer	**Mult**: 1	R	**Key**: No
	Description: The overridePriority allows for one rule in a rule group to override another. Priority n rules override priority n+1 rules. Rules of the same priority override as follows (n overrides n+1): 1 - MustNot, 2 - Must, 3 - May, 4 - Null. Within a rule the flexibility rules (signal, port role...) override as follows (n overrides n+1): 1 - Any, 2 - Same, 3 - Different. Where there are two or more "Same" rules, they will form an intersection where all must be met.			
cepDirection	**Type**: PortDirection	**Mult**: 0..*	R	**Key**: No
	Description: The list of CEP directions that the rule applies to, where the CEP direction is the orientation of flow at the (conceptual) port of the associated Connection. No entry means all CEP directions.			
cepPortRole	**Type**: PortRoleRule	**Mult**: 0..*	R	**Key**: No
	Description: Indicates the port role to which the rule applies. The port role is interpreted in the context of the connection type which is identified by the connection spec, if any. The port role is not meaningful in the absence of a connection spec reference. If a NodeRuleGroup carries a port role, that role applies also to the associated InterRuleGroup where the combination of the roles in the NodeRuleGroups at the ends of the InterGroupRule define the Connection orientation. For example a root-and-leaf Connection may be used in a Node where a NodeRuleGroup collects one set of NEPs has the port role "root" and another NodeRuleGroup collects another set of NEPs has the port role "leaf" where these are joined by an InterRuleGroup. This combination specifies an allowed orientation of the root-and-leaf Connection. No port role statement means all port roles are allowed.			
connectionSpecReference	**Type**: ConnectionSpecReference	**Mult**: 0..*	R	**Key**: No
	Description: Identifies the type of Connection that the rule applies to. If the attribute is not present then the rule applies to all types of Connection supported by the device.			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 0..*	R	**Key**: No
	Description: Qualifies a rule for a particular layer protocol identifying the qualifiers that the rule applies to. If the attribute is not present then the rule applies to all relevant qualifiers of the layer protocol of the parent entity.			
signalProperty	**Type**: SignalPropertyRule	**Mult**: 0..1	R	**Key**: No
	Description: The rule only applies to signals with the properties listed. If the attribute is not present then the rule applies to all signals.			
ComplexRule	**Type**: String	**Mult**: 0..*	R	**Key**: No
	Description: Allows for more complex rules where the basic rule system is not sufficient.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1

Attribute Name	Attribute Info			
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

8.2.11 Topology

The Topology is an abstract representation of the topological aspects of a particular set of network resources. It is described in terms of the underlying topological network of Node and Link instances that enable the forwarding capabilities of that particular set of network resources.

Attribute Name	Attribute Info			
_node	**Type**: Node	**Mult**: 0..*	R	**Key**: No
	Description: The list of Nodes which the Topology encompass.			
_link	**Type**: Link	**Mult**: 0..*	R	**Key**: No **Condition**: full-topology-management
	Description: The list of Links which the Topology encompass.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1..*	R	**Key**: No
	Description: The layer protocol(s) of the (multi-layer) Topology.			
_boundaryNodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 0..*	R	**Key**: No
	Description: This list is applicable only in case of a "top" Topology (i.e. a Topology which is not encapsulated in a Node) which does not encompass a single Node. In this case, the list identifies the NEPs which are at the boundary of the Topology, which can be a subset of all the NEPs belonging to encompassed Nodes. It is expected that these boundary NEPs have an associated SIP to allow the provisioning of ConnectivityServices spanning the whole Topology.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No

Attribute Name	Attribute Info
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.

8.2.12 TopologyContext

This object class represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the topology description. An instance of this class includes its Topology object instances.

Attribute Name	Attribute Info			
_nwTopologyService	**Type**: NetworkTopologyService	**Mult**: 0..1	R	**Key**: No **Condition**: full-topology-management
	Description: The defined operations.			
_topology	**Type**: Topology	**Mult**: 0..*	R	**Key**: No **Condition**: single-node-management or full-topology-management
	Description: The included Topology instances.			

8.2.13 TransferCostPac

The cost characteristics of a topological entity (e.g. a Link or a Node) not necessarily correlated to the cost of the underlying physical realization. They may be quite specific to the individual topological entity e.g. opportunity cost. Relates to layer capacity. There may be many perspectives from which cost may be considered for a particular topological entity and hence many specific costs and potentially cost algorithms. Using an entity will incur a cost.

Attribute Name	Attribute Info			
costCharacteristic	**Type**: CostCharacteristic	**Mult**: 1..*	R	**Key**: No
	Description: The list of costs where each cost relates to some aspect of the topological entity.			

8.2.14 TransferIntegrityPac

Transfer integrity characteristic covers expected/specified/acceptable characteristic of degradation of the transfered signal. It includes all aspects of possible degradation of signal content as well as any damage of any form to the total topological entity and to the carried signals. Note that the statement is of total impact to the topological entity so any partial usage of the topological entity (e.g. a signal that does not use full capacity) will only suffer its portion of the impact.

Attribute Name	Attribute Info			
errorCharacteristic	**Type**: String	**Mult**: 0..1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: Describes the degree to which the signal propagated can be errored. Applies to TDM systems as the errored signal will be propagated and not to packet as errored packets will be discarded.			
lossCharacteristic	**Type**: String	**Mult**: 0..1	R	**Key**: No
	Description: Describes the acceptable characteristic of lost packets where loss may result from discard due to errors or overflow. Applies to packet systems and not to TDM (as for TDM errored signals are propagated unless grossly errored and overflow/underflow turns into timing slips).			
repeatDeliveryCharacteristic	**Type**: String	**Mult**: 0..1	R	**Key**: No
	Description: Primarily applies to packet systems where a packet may be delivered more than once (in fault recovery for example). It can also apply to TDM where several frames may be received twice due to switching in a system with a large differential propagation delay.			
deliveryOrderCharacteristic	**Type**: String	**Mult**: 0..1	R	**Key**: No
	Description: Describes the degree to which packets will be delivered out of sequence. Does not apply to TDM as the TDM protocols maintain strict order.			
unavailableTimeCharacteristic	**Type**: String	**Mult**: 0..1	R	**Key**: No
	Description: Describes the duration for which there may be no valid signal propagated.			
serverIntegrityProcessCharacteristic	**Type**: String	**Mult**: 0..1	R	**Key**: No
	Description: Describes the effect of any server integrity enhancement process on the characteristics of the topological entity.			

8.2.15 TransferTimingPac

A topological entity (e.g. a Link or a Node) will suffer effects from the underlying physical realization related to the timing of the information passed by the topological entity.

Attribute Name	Attribute Info			
latencyCharacteristic	**Type**: LatencyCharacteristic	**Mult**: 1..*	R	**Key**: No
	Description: The effect on the latency of a queuing process. This only has significant effect for packet based systems and has a complex characteristic.			

8.2.16 ValidationPac

Validation covers the various adjacency discovery and reachability verification protocols. Also may cover information source and degree of integrity.

Attribute Name	Attribute Info			
validationMechanism	**Type**: ValidationMechanism	**Mult**: 1..*	R	**Key**: No
	Description: Provides details of the specific validation mechanism(s) used to confirm the presence of an intended topological entity.			

8.3 Associations

8.3.1 ContextHasNwTopologyService

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_nwTopologyService	composite	Yes	NetworkTopologyService	0..1
context	none	No	TopologyContext	1

8.3.2 ContextHasTopology

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_topology	composite	Yes	Topology	0..*
context	none	No	TopologyContext	1

8.3.3 IRGHasAssociatedNRG

Association end role name	Aggregation type	Navigable	Target Class	Mult
_associatedNodeRuleGroup	none	Yes	NodeRuleGroup	2..*
interrulegroup	none	No	InterRuleGroup	0..*

8.3.4 IRGHasCapacityPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferCapacity	composite	Yes	CapacityPac	0..1
interrulegroup	none	No	InterRuleGroup	1

8.3.5 IRGHasCostPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferCost	composite	Yes	TransferCostPac	0..1
interrulegroup	none	No	InterRuleGroup	1

8.3.6 IRGHasRiskPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_riskParameter	composite	Yes	RiskParameterPac	0..1
interrulegroup	none	No	InterRuleGroup	1

8.3.7 IRGHasRules

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_rule	composite	Yes	Rule	1..*
interrulegroup	none	No	InterRuleGroup	1

8.3.8 IRGHasTimingPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferTiming	composite	Yes	TransferTimingPac	0..1
interrulegroup	none	No	InterRuleGroup	1

8.3.9 LinkHasCapacityPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferCapacity	composite	Yes	CapacityPac	1
_link	none	No	Link	1

8.3.10 LinkHasCostPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferCost	composite	Yes	TransferCostPac	0..1
_link	none	No	Link	1

8.3.11 LinkHasIntegrityPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferIntegrity	composite	Yes	TransferIntegrityPac	0..1
_link	none	No	Link	1

8.3.12 LinkHasRiskPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_riskParameter	composite	Yes	RiskParameterPac	0..1
_link	none	No	Link	1

8.3.13 LinkHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	AdminStatePac	1
_link	none	No	Link	1

8.3.14 LinkHasTimingPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferTiming	composite	Yes	TransferTimingPac	0..1
_link	none	No	Link	1

8.3.15 LinkHasTransitionPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_lpTransition	composite	Yes	LayerProtocolTransitionPac	0..1
_link	none	No	Link	1

8.3.16 LinkHasValidationPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_validation	composite	Yes	ValidationPac	0..1
_link	none	No	Link	1

8.3.17 LinkTerminatesOnNEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_nodeEdgePoint	none	Yes	NodeEdgePoint	2..*
_linkPort	none	No	Link	0..1

8.3.18 NEPAggregatesNEPsInSameNode

Association end role name	Aggregation type	Navigable	Target Class	Mult
_aggregatedNodeEdgePoint	shared	Yes	NodeEdgePoint	0..*
_nodeEdgePoint	none	No	NodeEdgePoint	1

8.3.19 NEPHasCapacityPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_capacity	composite	Yes	CapacityPac	1
nodeedgepoint	none	No	NodeEdgePoint	1

8.3.20 NEPHasInterDomainId

ENNI NEP may have Inter Domain Plug Id.

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_interDomainPlugInPac	none	Yes	InterDomainPlugInPac	0..1
nodeedgepoint	none	No	NodeEdgePoint	1

8.3.21 NEPHasTerminationPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_termination	composite	Yes	TerminationPac	1
nodeedgepoint	none	No	NodeEdgePoint	1

8.3.22 NEPRelatesToSIP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_mappedServiceInterfacePoint	none	Yes	ServiceInterfacePoint	0..*
_mappedNodeEdgePoint	none	No	NodeEdgePoint	0..*

8.3.23 NRGAggregatesNEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_nodeEdgePoint	shared	Yes	NodeEdgePoint	1..*
noderulegroup	none	No	NodeRuleGroup	0..*

8.3.24 NRGEncompassesIRG

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_interRuleGroup	composite	Yes	InterRuleGroup	0..*
noderulegroup	none	No	NodeRuleGroup	1

8.3.25 NRGEncompassesLowerNRG

Association end role name	Aggregation type	Navigable	Target Class	Mult
_nodeRuleGroup	shared	Yes	NodeRuleGroup	0..*
noderulegroup	none	No	NodeRuleGroup	1

8.3.26 NRGHasCapacityPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferCapacity	composite	Yes	CapacityPac	0..1
noderulegroup	none	No	NodeRuleGroup	1

8.3.27 NRGHasCostPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferCost	composite	Yes	TransferCostPac	0..1
noderulegroup	none	No	NodeRuleGroup	1

8.3.28 NRGHasRiskPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_riskParameter	composite	Yes	RiskParameterPac	0..1
noderulegroup	none	No	NodeRuleGroup	1

8.3.29 NRGHasRules

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_rule	composite	Yes	Rule	1..*
noderulegroup	none	No	NodeRuleGroup	1

8.3.30 NRGHasTimingPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferTiming	composite	Yes	TransferTimingPac	0..1
noderulegroup	none	No	NodeRuleGroup	1

8.3.31 NodeAggregatesNEPExposedByEncapsulatedTopology

Association end role name	Aggregation type	Navigable	Target Class	Mult
_aggregatedNodeEdgePoint	shared	Yes	NodeEdgePoint	0..*
_node	none	No	Node	1..*

8.3.32 NodeEPHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	AdminStatePac	1
_nodeEdgePoint	none	No	NodeEdgePoint	1

8.3.33 NodeEncapsulatesNRG

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_nodeRuleGroup	composite	Yes	NodeRuleGroup	0..*
node	none	No	Node	1

8.3.34 NodeEncapsulatesTopology

LifecycleAggregate

Association end role name	Aggregation type	Navigable	Target Class	Mult
_encapTopology	shared	Yes	Topology	0..1
_forwardingDomain	none	No	Node	0..1

8.3.35 NodeHasCapacityPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferCapacity	composite	Yes	CapacityPac	1
_node	none	No	Node	1

8.3.36 NodeHasCostPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferCost	composite	Yes	TransferCostPac	0..1
_node	none	No	Node	1

8.3.37 NodeHasIntegrityPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferIntegrity	composite	Yes	TransferIntegrityPac	0..1
_node	none	No	Node	1

8.3.38 NodeHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	AdminStatePac	1
_node	none	No	Node	1

8.3.39 NodeHasTimingPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_transferTiming	composite	Yes	TransferTimingPac	0..1
_node	none	No	Node	1

8.3.40 NodeOwnsNEP

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_ownedNodeEdgePoint	composite	Yes	NodeEdgePoint	0..*
_node	none	No	Node	1

8.3.41 NwTopologyServiceHasTopology

LifecycleAggregate

Association end role name	Aggregation type	Navigable	Target Class	Mult
_topology	shared	Yes	Topology	0..*
_nwTopologyService	none	No	NetworkTopologyService	0..1

8.3.42 TopologyEncompassesLinks

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_link	composite	Yes	Link	0..*
_forwardingDomain	none	No	Topology	1

8.3.43 TopologyEncompassesNodes

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_node	composite	Yes	Node	0..*
_upperLevelFd	none	No	Topology	1

8.3.44 TopologyExposesBoundaryNEPs

Association end role name	Aggregation type	Navigable	Target Class	Mult
_boundaryNodeEdgePoint	shared	Yes	NodeEdgePoint	0..*
topology	none	No	Topology	0..1

8.4 Abstractions

8.4.1 AugmentsRootContext

Augments the base TAPI Context with TopologyService model.

Target Class: "/TapiCommon:Context:_context"

8.4.2 InterRuleGroupAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

8.4.3 LinkAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

8.4.4 NepAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

8.4.5 NodeAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

8.4.6 NodeRuleGroupAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

8.4.7 NtwTopoSrvAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

8.4.8 RuleAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

8.4.9 TopologyAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

8.4.10 TopologyObjectTypeAugmentsObjectType

Enumeration Augment.

8.5 Data Types

8.5.1 ConnectionSpecReference

The definition of the type of Connection. This definition will explain the flows in the Connection and how they relate to the roles of (conceptual) ports.

Attribute Name	Attribute Info			
connectionSpecName	**Type**: String	**Mult**: 1	R	**Key**: No
	Description: The name of the Connection type spec. This can be used as a reference to a paper document where full formal machine interpretable specs are not supported.			
connectionSpec	**Type**: Uuid	**Mult**: 0..1	RW	**Key**: No
	Description: The reference to the formal Connection type spec.			

8.5.2 CostCharacteristic

The cost characteristic related to some aspect of a topological entity.

Attribute Name	Attribute Info			
costName	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: The cost characteristic will be related to some aspect of the topological entity (e.g. $ cost, routing weight). This aspect will be conveyed by the costName.			
costValue	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: The specific cost.			
costAlgorithm	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: The cost may vary based upon some properties of the topological entity. The rules for the variation are conveyed by the costAlgorithm.			

8.5.3 LatencyCharacteristic

Provides information on latency characteristic for a particular stated trafficProperty.

Attribute Name	Attribute Info			
trafficPropertyName	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: The identifier of the specific traffic property to which the queuing latency applies.			
fixedLatencyCharacteristic	**Type**: String	**Mult**: 0..1	R	**Key**: No
	Description: A topological entity suffers delay caused by the realization of the servers (e.g. distance related; FEC encoding etc.) along with some client specific processing. This is the total average latency effect of the topological entity.			
queuingLatencyCharacteristic	**Type**: String	**Mult**: 0..1	RW	**Key**: No
	Description: The specific queuing latency for the traffic property.			
jitterCharacteristic	**Type**: String	**Mult**: 0..1	R	**Key**: No
	Description: High frequency deviation from true periodicity of a signal and therefore a small high rate of change of transfer latency. Applies to TDM systems (and not packet).			
wanderCharacteristic	**Type**: String	**Mult**: 0..1	R	**Key**: No
	Description: Low frequency deviation from true periodicity of a signal and therefore a small low rate of change of transfer latency. Applies to TDM systems (and not packet).			

8.5.4 MultiplexingSequence

The supported multiplexing sequences, e.g. - ODU0; ODU1; ODU2; ODU4 : 80 - ODU0; ODU1; ODU2; ODU3; ODU4 : 64 - ODUflex; ODU2; ODU3; ODU4 : 64 [64/ts] : 10G - ODUflex; ODU2; ODU4: 80 [80/ts] : 10G - ODU1; ODU2; ODU3; ODUCn : 40 [mult. for n] - OTS; OMS; MC; OTSiMC : 80 : 50G - OTSi; ODUCn : 2 : 200G - OTSi; ODUCn : 1 : 400G

Attribute Name	Attribute Info			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1..*	RW	**Key**: No
	Description: List of layer protocol qualifiers composing the multiplexing sequence.			
numberOfCepInstances	**Type**: Integer	**Mult**: 0..1	RW	**Key**: No
	Description: The maximum number of CEP instances of the multiplexing sequence.			
capacity	**Type**: CapacityValue	**Mult**: 0..1	RW	**Key**: No

Attribute Name	Attribute Info
	Description: The maximum capacity of the multiplexing sequence.

8.5.5 NepLayerProtocolCapability

Number of CEP instances at the layer protocol qualifier.

Attribute Name	Attribute Info			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: The layer protocol qualifier value.			
numberOfCepInstances	**Type**: Integer	**Mult**: 1	RW	**Key**: No
	Description: The number of CEP instances.			

8.5.6 PortRole

The role of a (conceptual) port in the context of the Connection spec referenced in the rule.

Attribute Name	Attribute Info			
roleName	**Type**: String	**Mult**: 1	R	**Key**: No
	Description: The name of the role of the CEP (associated to the conceptual port) of the Connection.			

8.5.7 PortRoleRule

Constrains which (conceptual) port roles the rule applies to.

Attribute Name	Attribute Info			
portRole	**Type**: PortRole	**Mult**: 0..*	R	**Key**: No
	Description: The role(s) of the port(s) considered in the rule.			
portRoleRule	**Type**: PortRoleRuleOption	**Mult**: 0..*	R	**Key**: No
	Description: Where the rule references more than one (conceptual) port role or where there are rule intersections either as a result of overlay of rules or InterRuleGroup usage indicates role matching criteria for a Connection following the rules. For example if two port roles, "a" and "b", are listed and the port role rule is "different", this means that a Connection connecting CEPs in that group must have port roles that are different for each CEP in that group. In the example if a Connection can have n ports of role "a" and m ports of role "b" then a maximum of two ports can be drawn from the NEPs of the group and where there are two, one must be role "a" and one must be role "b".			

8.5.8 ResilienceType

The type of resiliency (protection/restoration).

Attribute Name	Attribute Info			
restorationPolicy	**Type**: RestorationPolicy	**Mult**: 1	RW	**Key**: No
	Description: The restoration policy.			
protectionType	**Type**: ProtectionType	**Mult**: 1	RW	**Key**: No
	Description: The protection type.			

8.5.9 RiskCharacteristic

The information for a particular risk characteristic where there is a list of risk identifiers related to that characteristic.

Attribute Name	Attribute Info			
riskCharacteristicName	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: The name of the risk characteristic. The characteristic may be related to a specific degree of closeness. For example a particular characteristic may apply to failures that are localized (e.g. to one side of a road) where as another characteristic may relate to failures that have a broader impact (e.g. both sides of a road that crosses a bridge). Depending upon the importance of the traffic being routed different risk characteristics will be evaluated.			
riskIdentifierList	**Type**: String	**Mult**: 1..*	RW	**Key**: No
	Description: A list of the identifiers of each physical/geographic unit (with the specific risk characteristic) that is related to a segment of the topological entity.			

8.5.10 SignalPropertyRule

Rule related to an identified signal property.

Attribute Name	Attribute Info			
signalPropertyName	**Type**: String	**Mult**: 1	R	**Key**: No
	Description: The name of the signal property to which the rule applies.			
signalPropertyValueRule	**Type**: SignalPropertyValueRule	**Mult**: 0..1	R	**Key**: No
	Description: Indicates how the signal properties should be accounted for.			
applicableSignalValue	**Type**: String	**Mult**: 0..*	R	**Key**: No

Attribute Name	Attribute Info			
	Description: Specific values of the signal property to which the rule applies.			
numberOfSignalValues	**Type**: Integer	**Mult**: 0..1	R	**Key**: No
	Description: The number of instances of this specific property that can be supported by the group.			

8.5.11 ValidationMechanism

Identifies the validation mechanism and describes the characteristics of that mechanism.

Attribute Name	Attribute Info			
validationMechanism	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: Name of mechanism used to validate adjacency.			
layerProtocolAdjacencyValidated	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: State of validation.			
validationRobustness	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: Quality of validation (i.e. how likely is the stated validation to be invalid).			

8.6 Enumerations

8.6.1 ForwardingRule

Rule that restricts the creation/deletion of a Connection between points referenced by rule groups.

Contains Enumeration Literals:

- MAY_FORWARD_ACROSS_GROUP:
  - NEPs referenced by the NodeRuleGroup (or indirectly by the InterRuleGroup between NodeRuleGroups) may have Connections created between them unless some other rule overrides this. For an InterRuleGroup points in a NodeRuleGroup at one end of the InterRuleGroup may be connected to points in a NodeRuleGroup at another end of the InterRuleGroup.
- MUST_FORWARD_ACROSS_GROUP:
  - NEPs referenced by the NodeRuleGroup (or indirectly by the InterRuleGroup between NodeRuleGroups) MUST have Connections created between them unless some other rule overrides this. For an InterRuleGroup points in a NodeRuleGroup at one end of the InterRuleGroup MUST be connected to points in a NodeRuleGroup at another end of the InterRuleGroup.
- CANNOT_FORWARD_ACROSS_GROUP:

  - NEPs referenced by the NodeRuleGroup (or indirectly by the InterRuleGroup between NodeRuleGroups) MUST NOT have Connections created between them. For an InterRuleGroup points in a NodeRuleGroup at one end of the InterRuleGroup MUST NOT be connected to points in an NodeRuleGroup at another end of the InterRuleGroup.
- NO_STATEMENT_ON_FORWARDING:
  - The rule group makes no statement on forwarding.
- INTER_CONNECTION_CONTENTION:
  - Connections to NEPs in the Rule Group contend for resources based upon a constraint of some signal property. For example, each Connection to a NEP in the Group must use a different value of the signal property from all other Connections to NEPs in the Rule Group. For example, each Connection to a NEP in the Group must use a same value of the signal property as all other Connections to NEPs in the Rule Group. In this case the first Connection created in the Rule Group sets the value and the Group constraint is freed when the last Connection is deleted.

8.6.2 PortRoleRuleOption

Indicates how to interpret the port role list.

Contains Enumeration Literals:

- SAME_ROLE:
  - The (conceptual) ports of the Connection to which the rule applies must have the same role from the list in port role.
- DIFFERENT_ROLE:
  - The (conceptual) ports of the Connection to which the rule applies must have different roles from the list in port role.
- ANY_ROLE:
  - The (conceptual) ports of the Connection to which the rule applies may take any identified role.
- NOT_ROLE:
  - The (conceptual) ports of the Connection to which the rule applies must not have any of the listed roles.

8.6.3 ProtectionType

The types of protection and restoration.

Contains Enumeration Literals:

- NO_PROTECTION:
- ONE_PLUS_ONE_PROTECTION:
  - Protection scheme where the switches are not required to be coordinated (typically the signal is always bridged).
- ONE_PLUS_ONE_PROTECTION_WITH_DYNAMIC_RESTORATION:
  - Protection scheme where the switches are not required to be coordinated (typically the signal is always bridged). In addition is implemented a second level of resilience, through dynamic restoration of the first connection affected by a failure.
- PERMANENT_ONE_PLUS_ONE_PROTECTION:

    - Extends the ONE_PLUS_ONE_PROTECTION_WITH_DYNAMIC_RESTORATION allowing an indeterminate number of failures to affect either of the 1+1 routes and the respective subsequent dynamic restorations.
- ONE_FOR_ONE_PROTECTION:
    - Protection scheme where the switches are coordinated (e.g. by signalling).
- DYNAMIC_RESTORATION:
    - Restoration scheme where the protection route is computed and implemented only when the current (and only) route is impaired (e.g. by a failure or maintenance command).
- PRE_COMPUTED_RESTORATION:
    - Restoration scheme where the protection route is pre-computed. When the current (and only) route is impaired (e.g. by a failure or maintenance command) the pre-computed route is implemented.
- ONE_PLUS_ONE_PROTECTION_WITH_PRE_COMPUTED_RESTORATION:
    - Protection scheme where the switches are not required to be coordinated (typically the signal is always bridged). In addition a further protection route is pre-computed. When either the current or protection route is impaired (e.g. by a failure or maintenance command), the pre-computed route is implemented to restore resiliency level.
- ONE_FOR_N_PROTECTION:
    - N routes share one protection route. Switches need coordination (e.g. by signalling).
- M_FOR_N_PROTECTION:
    - N routes share M protection routes. Switches need coordination (e.g. by signalling).
- ONE_FOR_ONE_BY_N:
    - N parallel one-for-one schemes.

8.6.4 RestorationPolicy

The restoration policy.

Contains Enumeration Literals:

- PER_DOMAIN_RESTORATION:
    - Restoration is expected to be performed independently within each (restoration) domain scope. This implies that the server is responsible of activating the required control mechanisms to guarantee the restoration of the service autonomously.
- END_TO_END_RESTORATION:
    - Restoration is expected to be performed on end to end basis across all domain(s).
- NA:
    - Not Applicable.

8.6.5 RuleType

The focus of the rule.

Contains Enumeration Literals:

- FORWARDING:
    - The rule applies to the creation of Connections.
- CAPACITY:
    - The rule applies to capacity limitations.
- COST:
    - The rule applies to the cost of the creation of Connections.

- TIMING:
  - The rule applies to timing constraints across the group.
- RISK:
  - The rule applies to risk considerations across the group so as to express shared risk.
- GROUPING:
  - The rule is simply for grouping related to other rules.

8.6.6 SignalPropertyValueRule

Indicates how to interpret the signal property value rule.

Contains Enumeration Literals:

- SAME_VALUE:
  - The signal property of the CEP to which the rule applies must have the same value from the identied list.
- ANY_VALUE:
  - The signal property of the CEP to which the rule applies may take any identified value.
- DIFFERENT_VALUE:
  - The signal property of the CEP to which the rule applies each must have different values from the identified list.
- NOT_VALUE:
  - The signal property of the CEP to which the rule applies must not have any of the identified values.

8.6.7 TopologyObjectType

The list of TAPI Topology Global Object Class types on which Notification signals can be raised.

Contains Enumeration Literals:

- TOPOLOGY:
  - The Topology class.
- NODE:
  - The Node class.
- LINK:
  - The Link class.
- NODE_EDGE_POINT:
  - The NodeEdgePoint (NEP) class.
- NODE_RULE_GROUP:
  - The NodeRuleGroup class.
- INTER_RULE_GROUP:
  - The InterRuleGroup class.
- RULE:
  - The Rule class.
- NETWORK_TOPOLOGY_SERVICE:
  - The NetworkTopologyService class.

9 Connectivity Model

9.1 Diagrams

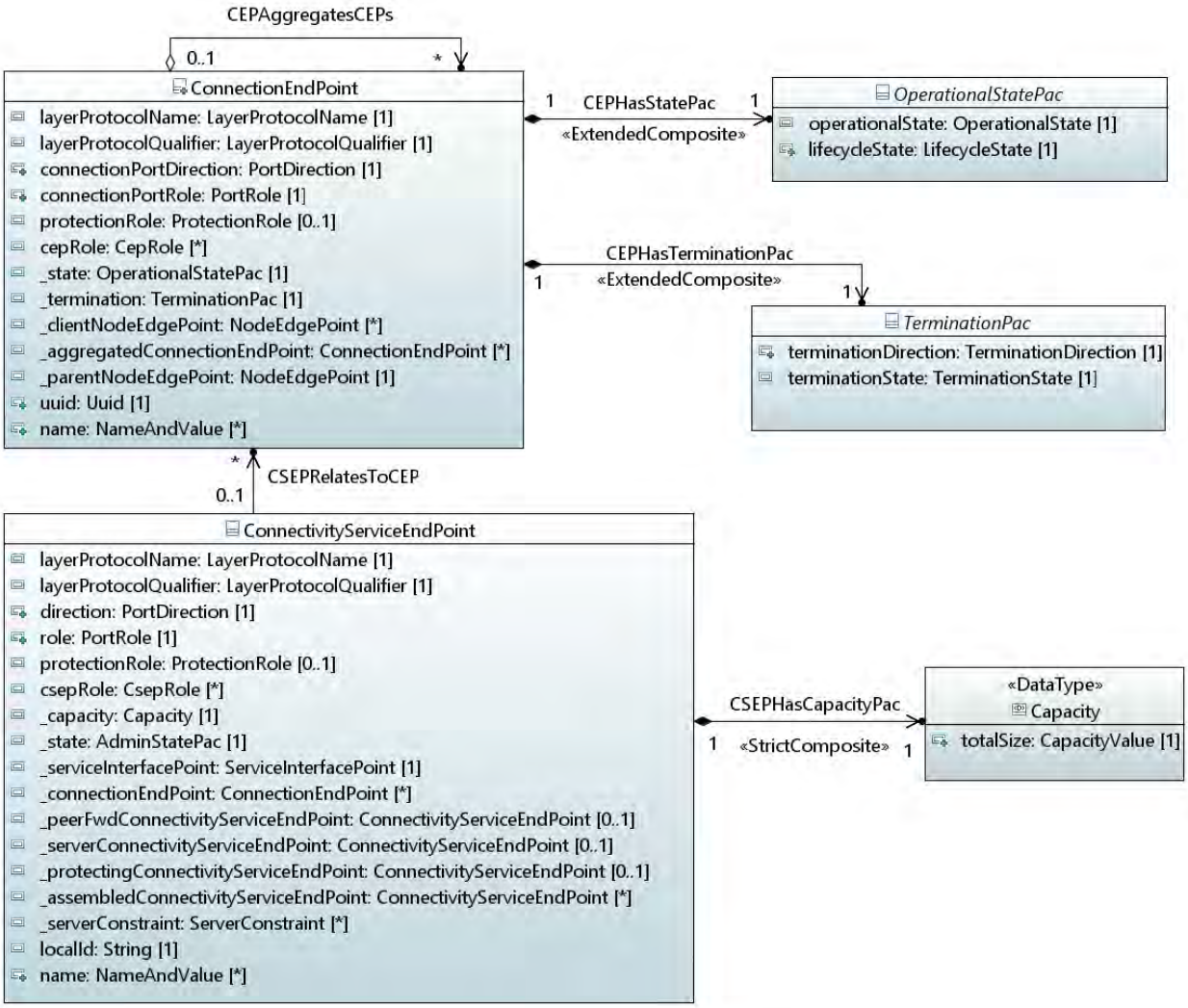

Figure 41 – *ConnectionEndPointDetails*

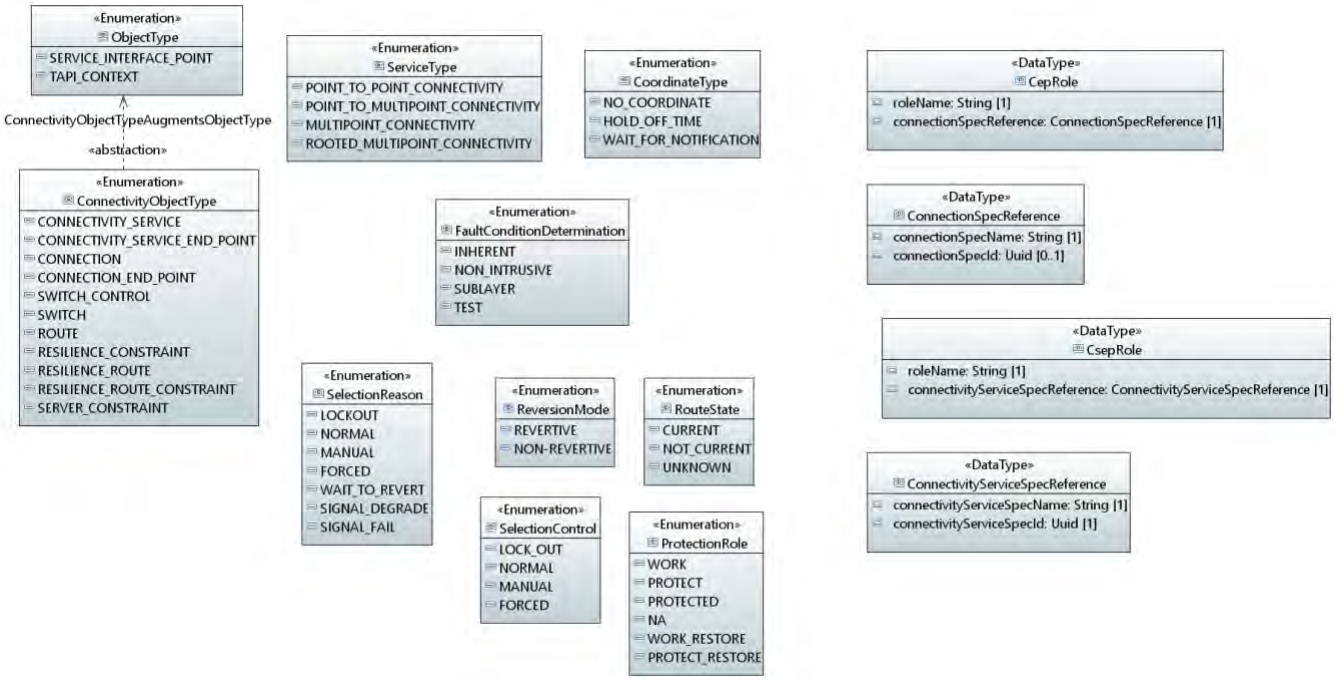

Figure 42 – *ConnectivityDataTypes*

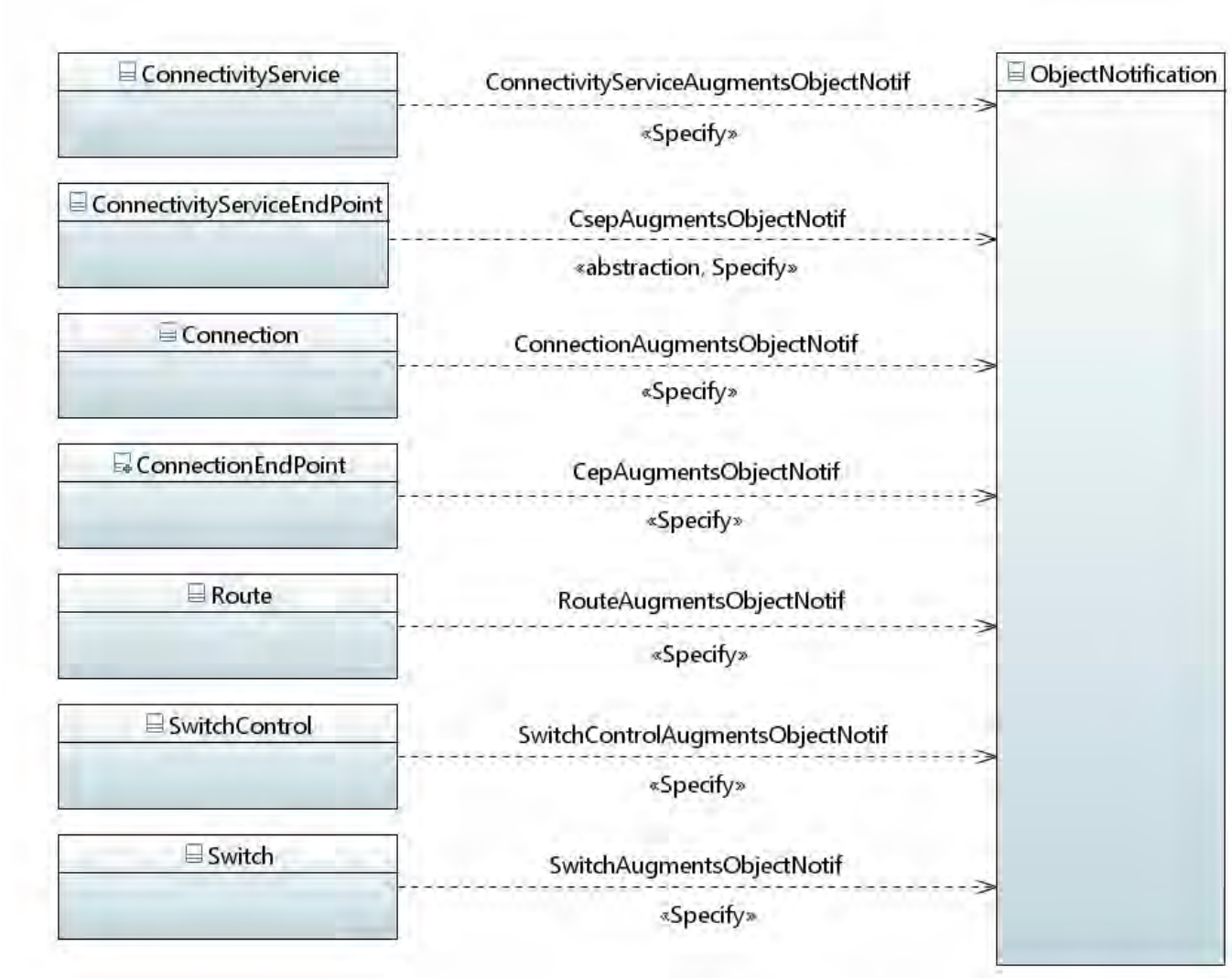

Figure 43 – *ConnectivityNotif*

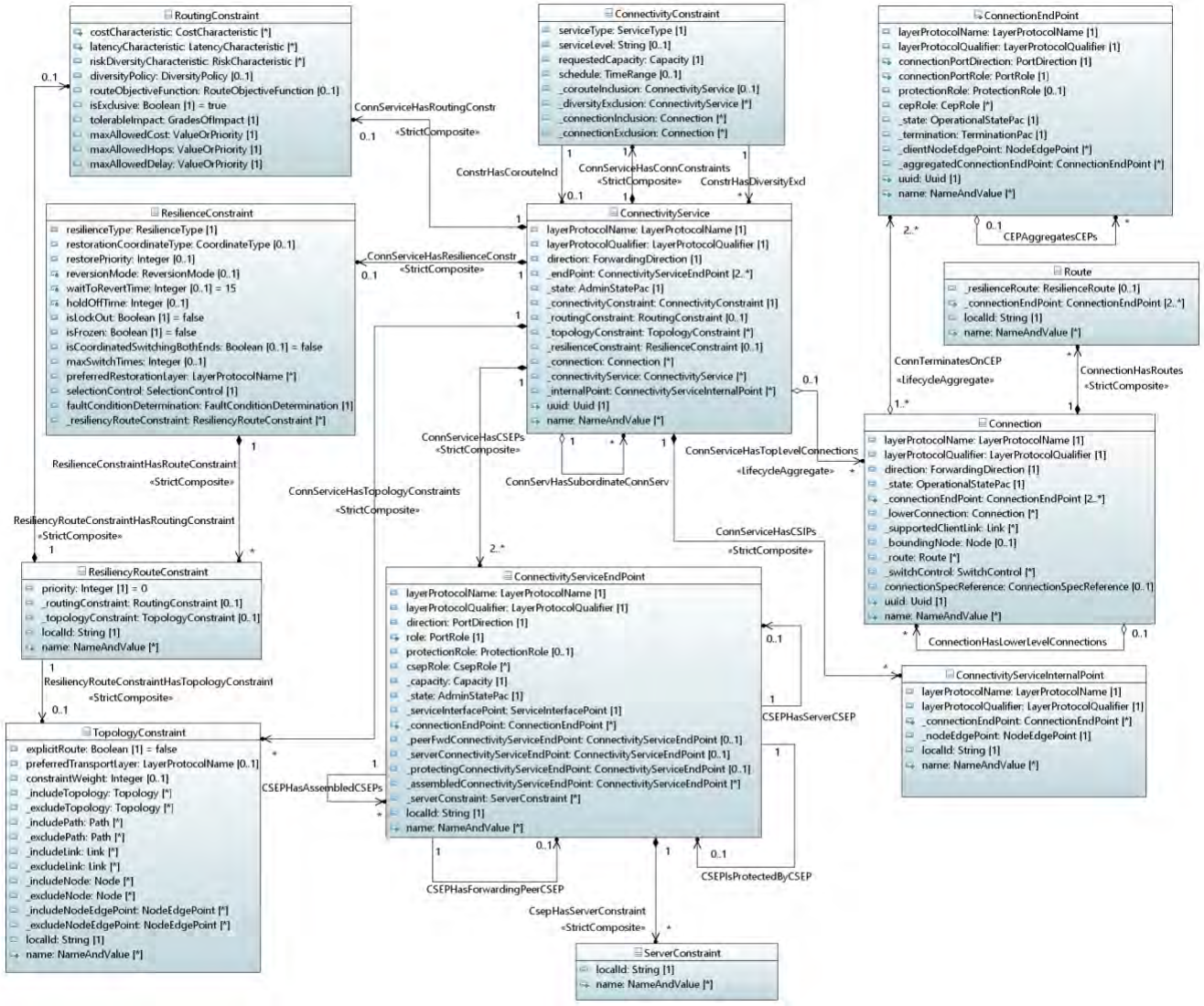

Figure 44 – *ConnectivityServiceDetails*

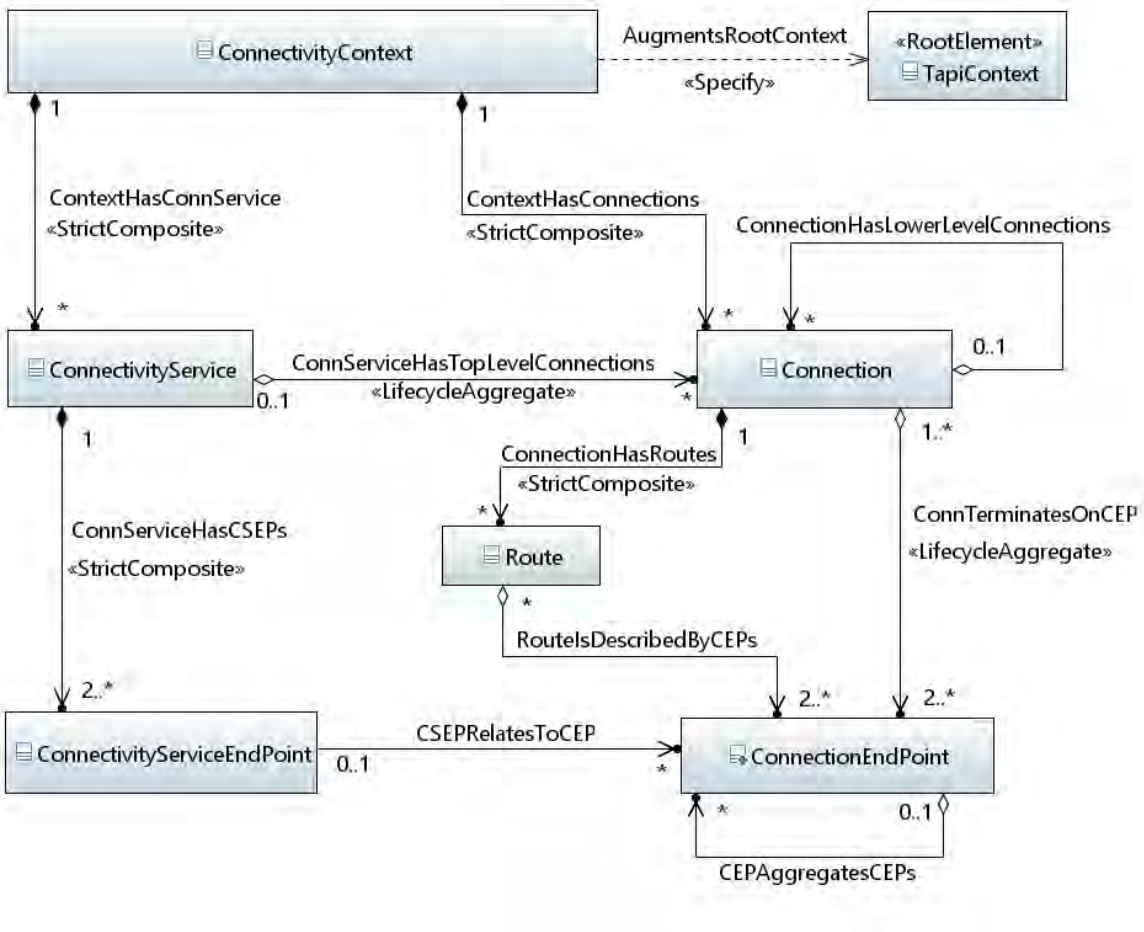

Figure 45 – *ConnectivityServiceSkeleton*

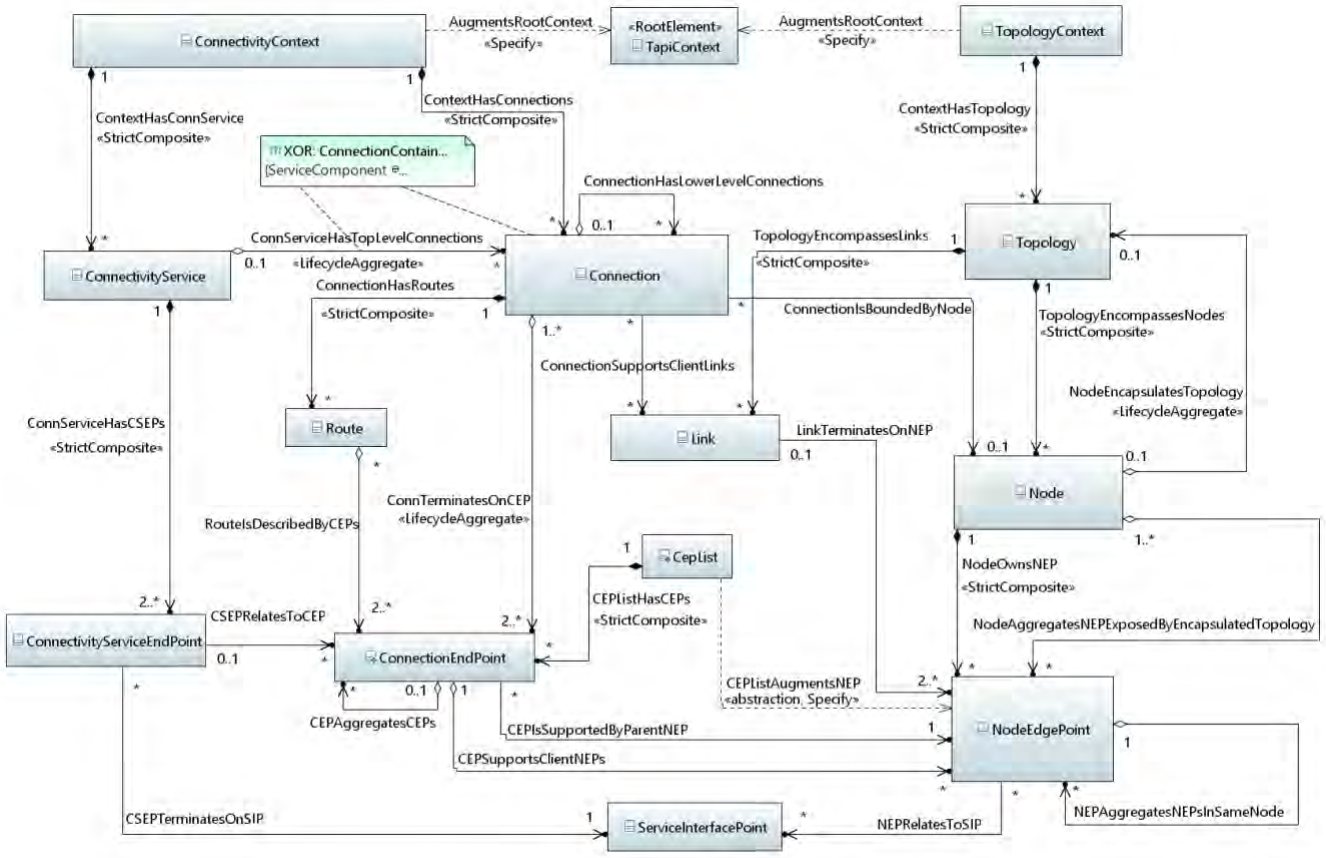

Figure 46 – *ConnectivityTopologySkeleton*

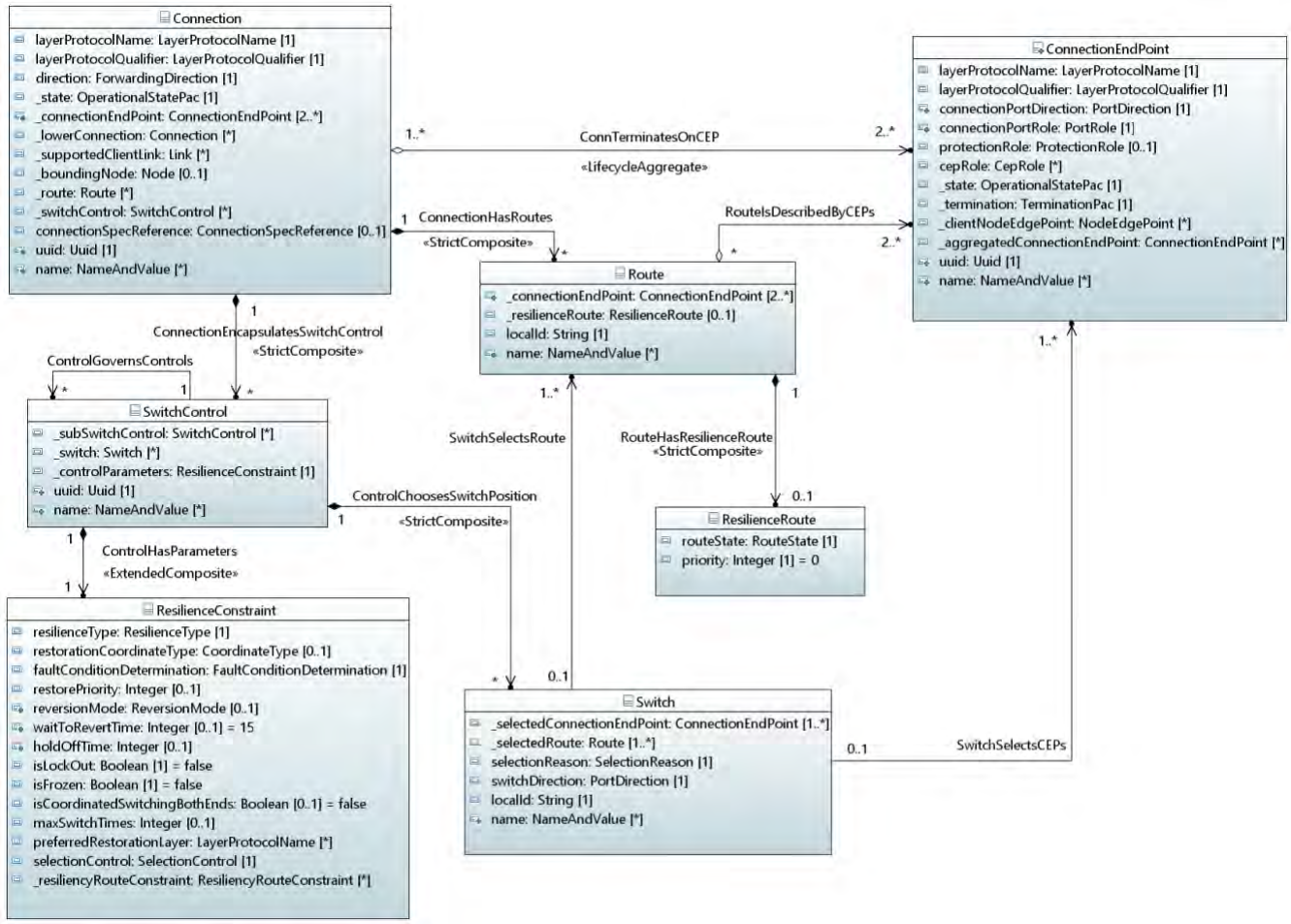

Figure 47 – *Resilience*

9.2 Classes

9.2.1 CepList

This class provides the linkage between the NodeEdgePoint (NEP) instance and its supported ConnectionEndPoint CEP instances. The NEP class, which is defined in TapiTopology module, cannot directly include the reference to its CEPs, because CEP class is defined in another module, TapiConnectivity.

Attribute Name	Attribute Info			
_connectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 0..*	RW	**Key**: No
	Description: The list of supported ConnectionEndPoint (CEP) instances.			

9.2.2 Connection

A Connection represents an enabled (provisioned) potential for forwarding (of transport characteristic information including all circuit/packet forms) between two or more ConnectionEndPoint instances. The bounding Node of a Connection may be explicit or be conceptually implicit. The Connection is a container for provisioned connectivity that tracks the state of the allocated resources and is distinct from the ConnectivityService. At the lowest level of recursion, a Connection may represent a cross-connection in a switch matrix (i.e., a fabric) in an equipment.

Attribute Name	Attribute Info			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	R	**Key**: No
	Description: The layer protocol of the Connection.			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol qualifier of the Connection.			
direction	**Type**: ForwardingDirection	**Mult**: 1	R	**Key**: No
	Description: The forwarding direction of the Connection.			
_state	**Type**: OperationalStatePac	**Mult**: 1	R	**Key**: No
	Description: The Connection status information.			
_connectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 2..*	R	**Key**: No
	Description: The ConnectionEndPoint (CEP) instances of the Connection.			
_lowerConnection	**Type**: Connection	**Mult**: 0..*	R	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: A Connection supports a recursive aggregation relationship such that the internal construction of a Connection can be exposed as multiple lower level Connection objects (partitioning). Aggregation is used as for the Node/Topology to allow changes in hierarchy. Connection aggregation reflects Node/Topology aggregation. Note that a cross-connection in a switch matrix (i.e., a fabric) is not necessarily the lowest level of Connection partitioning.			
_supportedClientLink	**Type**: Link	**Mult**: 0..*	R	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: A Connection instance supports one or more Link instances. G.800: "The links in a client layer network are supported by trails in a server layer network".			
_boundingNode	**Type**: Node	**Mult**: 0..1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: A Connection may or may not be bounded by a Node, which defines the forwarding scope.			
_route	**Type**: Route	**Mult**: 0..*	R	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The Route instances of the Connection.			
_switchControl	**Type**: SwitchControl	**Mult**: 0..*	R	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The SwitchControl instances associated to the Connection.			
connectionSpecReference	**Type**: ConnectionSpecReference	**Mult**: 0..1	R	**Key**: No
	Description: Provides the reference to the spec that defines the connection type and cepRoles.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.2.3 ConnectionEndPoint

The ConnectionEndPoint (CEP) encapsulates information related to a Connection at the ingress/egress points of every Node that the Connection traverses in a Topology. The CEP includes the termination and adaptation functions of one or more transport layers (circuit and packet forms) plus the information of the (conceptual) port of associated Connection.

Attribute Name	Attribute Info			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	R	**Key**: No
	Description: The layer protocol of the ConnectionEndPoint (CEP).			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: The layer protocol qualifier of the ConnectionEndPoint (CEP).			
connectionPortDirection	**Type**: PortDirection	**Mult**: 1	R	**Key**: No
	Description: The orientation of flow at the (conceptual) port of the associated Connection.			
connectionPortRole	**Type**: PortRole	**Mult**: 1	R	**Key**: No
	Description: The role of the (conceptual) port of the associated Connection.			
protectionRole	**Type**: ProtectionRole	**Mult**: 0..1	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The protection role of the (conceptual) port of the associated Connection. It is recommended the alignment with the priority of ResilienceRoute.			
cepRole	**Type**: CepRole	**Mult**: 0..*	R	**Key**: No
	Description: Defines the role of the CEP in the context of the Connection spec. There may be many CEP role - Connection spec combinations for a particular CEP where each corresponds to a specific Connection associated with the CEP.			
_state	**Type**: OperationalStatePac	**Mult**: 1	R	**Key**: No
	Description: The ConnectionEndPoint (CEP) status information.			
_termination	**Type**: TerminationPac	**Mult**: 1	RW	**Key**: No
	Description: Termination direction and termination state of the ConnectionEndPoint (CEP).			
_clientNodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 0..*	R	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The supported NodeEdgePoint instance(s).			
_aggregatedConnectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 0..*	R	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: A ConnectionEndPoint (CEP) instance may aggregate one or more other CEP instances for e.g. pooling purposes, when a set of CEP instances are equivalent for usage.			
_parentNodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: The supporting NodeEdgePoint (NEP) instance.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.2.4 ConnectivityConstraint

The connectivity constraints associated to a ConnectivityService instance.

Attribute Name	Attribute Info			
serviceType	**Type**: ServiceType	**Mult**: 1	RW	**Key**: No
	Description: The ConnectivityService type.			
serviceLevel	**Type**: String	**Mult**: 0..1	RW	**Key**: No
	Description: Class of Service Name. An abstract value the meaning of which is mutually agreed - typically represents metrics such as - Class of service, priority, resiliency, availability.			
requestedCapacity	**Type**: Capacity	**Mult**: 1	RW	**Key**: No
	Description: The ConnectivityService capacity.			
schedule	**Type**: TimeRange	**Mult**: 0..1	RW	**Key**: No
	Description: The ConnectivityService timing.			
_corouteInclusion	**Type**: ConnectivityService	**Mult**: 0..1	RW	**Key**: No
	Description: The reference to another ConnectivityService instance for corouting purposes.			
_diversityExclusion	**Type**: ConnectivityService	**Mult**: 0..*	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: The references to other ConnectivityService instances for routing diversity purposes.			
_connectionInclusion	**Type**: Connection	**Mult**: 0..*	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: A ConnectivityService may use one or more existing Connections. A common traditional strategy is to set up 'stranded' connectivity in the core of the network as "express channels" (this is essentially a serial compound link, but can be treated as simple connections). A Connection inclusion capability allows for adoption of discovered Connections, i.e. will allow discovered Connections with no stated intent to be associated with an intent via the ConnectivityService. A ConnectivityService is requested with a Connection inclusion constraint that identifies a Connection (or chain of Connections) that is bounded by CEPs that each belong to a NEP that references a SIP that is referenced by a CSEP of the ConnectivityService such that all CSEPs are satisfied by CEPs of the existing Connection. The type is generic UUID given read/write constraints, the Connection is a readonly node.			
_connectionExclusion	**Type**: Connection	**Mult**: 0..*	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The list of Connection instances which shall not be used to implement the ConnectivityService. The type is generic UUID given read/write constraints, the Connection is a readonly node.			

9.2.5 ConnectivityContext

This object class represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the connectivity description. An instance of this class includes its ConnectivityService and Connection object instances.

Attribute Name	Attribute Info			
_connectivityService	**Type**: ConnectivityService	**Mult**: 0..*	RW	**Key**: No
	Description: The included ConnectivityService instances.			
_connection	**Type**: Connection	**Mult**: 0..*	R	**Key**: No **Condition**: top-connection-management
	Description: The included Connection instances.			

9.2.6 ConnectivityService

A ConnectivityService represents an intent-like request for connectivity between two or more ConnectivityServiceEndPoint (CSEP) instances. The ConnectivityService is a container for connectivity request details and is distinct from the Connection(s) that realize the request.

Attribute Name	Attribute Info			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: The layer protocol of the CS.			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol qualifier of the CS.			
direction	**Type**: ForwardingDirection	**Mult**: 1	R	**Key**: No
	Description: The forwarding direction of the ConnectivityService.			
_endPoint	**Type**: ConnectivityServiceEndPoint	**Mult**: 2..*	RW	**Key**: No
	Description: The ConnectivityServiceEndPoint (CSEP) instances of the ConnectivityService.			
_state	**Type**: AdminStatePac	**Mult**: 1	RW	**Key**: No
	Description: The ConnectivityService status information.			
_connectivityConstraint	**Type**: ConnectivityConstraint	**Mult**: 1	RW	**Key**: No
	Description: The associated connectivity constraints.			
_routingConstraint	**Type**: RoutingConstraint	**Mult**: 0..1	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The associated routing constraints.			
_topologyConstraint	**Type**: TopologyConstraint	**Mult**: 0..*	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The associated topology constraints. Different instances of TopologyConstraints may be used to specify constraints at different layer networks.			
_resilienceConstraint	**Type**: ResilienceConstraint	**Mult**: 0..1	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The associated resilience constraints.			
_connection	**Type**: Connection	**Mult**: 0..*	R	**Key**: No **Condition**: top-connection-management

Attribute Name	Attribute Info			
	Description: The Connection instance(s) tracking the state of the allocated resources for the support of the ConnectivityService.			
_connectivityService	**Type**: ConnectivityService	**Mult**: 0..*	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: Association to other ConnectivityService instances for complex connectivity provisioning.			
_internalPoint	**Type**: ConnectivityServiceInternalPoint	**Mult**: 0..*	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The ConnectivityServiceInternalPoint (CSIP) instances of the ConnectivityService.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.2.7 ConnectivityServiceEndPoint

The ConnectivityServiceEndPoint (CSEP) encapsulates information related to a ConnectivityService at the ingress/egress points of that ConnectivityService.

Attribute Name	Attribute Info			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol of the ConnectivityServiceEndPoint (CSEP).			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol qualifier of the ConnectivityServiceEndPoint (CSEP).			
direction	**Type**: PortDirection	**Mult**: 1	RW	**Key**: No
	Description: The orientation of flow at the (conceptual) port of the associated ConnectivityService.			

Attribute Name	Attribute Info			
role	**Type**: PortRole	**Mult**: 1	RW	**Key**: No
	Description: The role of the (conceptual) port of the associated ConnectivityService.			
protectionRole	**Type**: ProtectionRole	**Mult**: 0..1	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The protection role of the (conceptual) port of the associated ConnectivityService. It is recommended the alignment with the priority of ResilienceRoute.			
csepRole	**Type**: CsepRole	**Mult**: 0..*	R	**Key**: No
	Description: Defines the role of the CSEP in the context of the Connectivity Service spec. There may be many CSEP role - CS spec combinations for a particular CSEP where each corresponds to a specific Connectivity Service associated with the CSEP.			
_capacity	**Type**: Capacity	**Mult**: 1	RW	**Key**: No
	Description: The ConnectivityServiceEndPoint (CSEP) capacity.			
_state	**Type**: AdminStatePac	**Mult**: 1	RW	**Key**: No
	Description: The ConnectivityServiceEndPoint (CSEP) status information.			
_serviceInterfacePoint	**Type**: ServiceInterfacePoint	**Mult**: 1	RW	**Key**: No
	Description: The supporting ServiceInterfacePoint (SIP) instance.			
_connectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 0..*	R	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The associated ConnectionEndPoint (CEP) instances.			
_peerFwdConnectivityServiceEndPoint	**Type**: ConnectivityServiceEndPoint	**Mult**: 0..1	RW	**Key**: No
	Description: The associated ConnectivityServiceEndPoint (CSEP) instance from forwarding perspective.			
_serverConnectivityServiceEndPoint	**Type**: ConnectivityServiceEndPoint	**Mult**: 0..1	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The associated ConnectivityServiceEndPoint (CSEP) instance at a server layer protocol (qualifier).			

Attribute Name	Attribute Info			
_protectingConnectivityServiceEndPoint	**Type**: ConnectivityServiceEndPoint	**Mult**: 0..1	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The associated ConnectivityServiceEndPoint (CSEP) instance from resilience perspective.			
_assembledConnectivityServiceEndPoint	**Type**: ConnectivityServiceEndPoint	**Mult**: 0..*	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The associated ConnectivityServiceEndPoint (CSEP) instances from assembling perspective, e.g. in inverse multiplexing schemes.			
_serverConstraint	**Type**: ServerConstraint	**Mult**: 0..*	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The server constraints.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.2.8 ConnectivityServiceInternalPoint

Experimental class for complex/detailed provisioning schemes.

Attribute Name	Attribute Info			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol of the ConnectivityServiceInternalPoint (CSIP).			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol qualifier of the ConnectivityServiceInternalPoint (CSIP).			
_connectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 0..*	R	**Key**: No
	Description: The associated ConnectionEndPoint (CEP) instances.			
_nodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: The supporting NodeEdgePoint (NEP) instance.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.2.9 ResilienceConstraint

The parameters of a protection/restoration scheme of a ConnectivityService or Connection.

Attribute Name	Attribute Info			
resilienceType	**Type**: ResilienceType	**Mult**: 1	RW	**Key**: No
	Description: The type of resiliency (protection/restoration).			
restorationCoordinateType	**Type**: CoordinateType	**Mult**: 0..1	RW	**Key**: No
	Description: The coordination mechanism between protection/restoration operations across multiple layers.			
faultConditionDetermination	**Type**: FaultConditionDetermination	**Mult**: 1	RW	**Key**: No
	Description: The types of the determinations of a fault condition on a serial compound link connection within the protected domain. Ref: G.808 Amendment 1 (03/2018)			
restorePriority	**Type**: Integer	**Mult**: 0..1	RW	**Key**: No
	Description: 0 highest priority, 1 lower, etc.			
reversionMode	**Type**: ReversionMode	**Mult**: 0..1	RW	**Key**: No
	Description: Indicates whether the protection/restoration scheme is revertive or non-revertive.			
waitToRevertTime	**Type**: Integer	**Mult**: 0..1	RW	**Key**: No
	Description: If the protection/restoration scheme is revertive, this attribute specifies the time, in minutes, to wait after a fault clears on a higher priority (preferred) resource before reverting to the preferred resource.			
holdOffTime	**Type**: Integer	**Mult**: 0..1	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: This attribute indicates the time, in milliseconds, between declaration of signal degrade or signal fail, and the initialization of the protection/restoration switching algorithm.			
isLockOut	**Type**: Boolean	**Mult**: 1	RW	**Key**: No
	Description: The resource is configured to temporarily not be available for use in the protection/restoration scheme(s) it is part of. This overrides all other control states including e.g. "forced". If the item is locked out then it cannot be used under any circumstances. Note: Only relevant when part of a protection/restoration scheme.			
isFrozen	**Type**: Boolean	**Mult**: 1	RW	**Key**: No
	Description: Temporarily prevents any switch action to be taken and, as such, freezes the current state of the protection/restoration scheme. Until the freeze is cleared, additional near-end external commands are rejected and fault condition changes and signalling (e,g, received APS messages) are ignored. All administrative controls of any aspect of the protection/restoration scheme are rejected.			
isCoordinatedSwitchingBothEnds	**Type**: Boolean	**Mult**: 0..1	RW	**Key**: No
	Description: Is operating such that the switching at both ends of each flow across the resilient forwarding entity (e.g. ConnectivityService or Connection) is coordinated at both ingress and egress ends.			
maxSwitchTimes	**Type**: Integer	**Mult**: 0..1	RW	**Key**: No
	Description: Used to limit the maximum switch times. When the impairment on preferred/intended resource disappears and traffic returns to the preferred/intended resource, switch counter reset.			
preferredRestorationLayer	**Type**: LayerProtocolName	**Mult**: 0..*	RW	**Key**: No
	Description: Indicates which layer protocol this resilience parameters package is configured for.			
selectionControl	**Type**: SelectionControl	**Mult**: 1	RW	**Key**: No
	Description: Degree of administrative control applied to the switch selection.			
_resiliencyRouteConstraint	**Type**: ResiliencyRouteConstraint	**Mult**: 0..*	RW	**Key**: No
	Description: The associated constraints related to resiliency routes.			

9.2.10 ResilienceRoute

This object adds resilience and state attributes to the Route. When this object is not present, then the Route is intended as "current" Route of the Connection.

Attribute Name	Attribute Info			
routeState	**Type**: RouteState	**Mult**: 1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: Current information on the route selection.			
priority	**Type**: Integer	**Mult**: 1	R	**Key**: No
	Description: Value of 0 (zero) means "unspecified priority". Highest priority is 1, sometimes referred as "preferred" or "main" or "intended" route. 2 has lower priority than 1, 3 has lower priority than 2, etc. It is recommended the alignment with the protectionRole of CEP/CSEP.			

9.2.11 ResiliencyRouteConstraint

The constraints related to the Resiliency route.

Attribute Name	Attribute Info			
priority	**Type**: Integer	**Mult**: 1	RW	**Key**: No
	Description: Value of 0 (zero) means "unspecified priority". Highest priority is 1, sometimes referred as "preferred" or "main" or "intended" route. 2 has lower priority than 1, 3 has lower priority than 2, etc.			
_routingConstraint	**Type**: RoutingConstraint	**Mult**: 0..1	RW	**Key**: No
	Description: The associated routing constraints.			
_topologyConstraint	**Type**: TopologyConstraint	**Mult**: 0..1	RW	**Key**: No
	Description: The associated topology constraints.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.2.12 Route

The Route of a Connection is modeled as a collection of ConnectionEndPoint (CEP) instances. The logical order of the ConnectionEndPoint (CEP) instances within the Route object can be inferred by the TAPI client by the knowledge of the topology information.

Attribute Name	Attribute Info			
_resilienceRoute	**Type**: ResilienceRoute	**Mult**: 0..1	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: Provides optional resilience and state attributes to the Route.			
_connectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 2..*	R	**Key**: No
	Description: The ConnectionEndPoint (CEP) instances composing the Route.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.2.13 ServerConstraint

This package allows augmentations for server layer technology specific constraints.

Attribute Name	Attribute Info			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.2.14 Switch

The class models the switched forwarding of traffic (traffic flow) between (conceptual) ports of resilient forwarding entities (e.g. resilient ConnectivityService, resilient Connection), these ports being mapped to ConnectionEndPoint (CEP) instances. A resilient forwarding entity may have two or more (conceptual) ports that provide alternative identical inputs/outputs, and one or more associated Switch instances to represent the alternative flow choices visible at the edge of the forwarding entity. The Switch instance represents and defines a protection switch structure conceptually encapsulated in the forwarding entity. The Switch instance essentially performs one of the functions of the Protection Group in a traditional model. It associates to 2 or more (conceptual) ports each playing the role of a Protection Unit. One or more protection, i.e. standby/backup, conceptual ports provide protection for one or more working (i.e. regular/main/preferred) ports where either protection or working can feed one or more protected port. The switch may be used in revertive or non-revertive (symmetric) mode. When in revertive mode it may define a waitToRestore time. It may be used in one of several modes including source switch, destination switched, source and destination switched, etc. (covering cases such as 1+1 and 1:1). It may be locked out (prevented

from switching), force switched or manual switched. It will indicate switch state and change of state. The Switch can be switched away from all sources such that it becomes open and hence two coordinated switches can both feed the same (conceptual) port or CEP so long as at least one of the two is switched away from all sources (is "open"). The ability for a Switch to be "high impedance" allows bidirectional forwarding entities to be overlaid on the same bidirectional CEP where the appropriate control is enabled to prevent signal conflict. This ability allows multiple alternate routes to be present that otherwise would be in conflict.

Attribute Name	Attribute Info			
_selectedConnectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 1..*	R	**Key**: No
	Description: The ConnectionEndPoint (CEP) instance(s) which is (are) currently selected for traffic flow.			
_selectedRoute	**Type**: Route	**Mult**: 1..*	R	**Key**: No
	Description: The Route instance(s) which is (are) currently selected for traffic flow.			
selectionReason	**Type**: SelectionReason	**Mult**: 1	R	**Key**: No
	Description: The reason for the current switch selection.			
switchDirection	**Type**: PortDirection	**Mult**: 1	RW	**Key**: No
	Description: The orientation of flow at the (conceptual) port of the associated Connection. Indicates whether the Switch selects from ingress to the Connection or to egress of the Connection or both.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.2.15 SwitchControl

Represents the capability to control and coordinate Switch instances, to add/delete/modify Connections and to add/delete/modify CEPs so as to realize a protection scheme.

Attribute Name	Attribute Info			
_subSwitchControl	**Type**: SwitchControl	**Mult**: 0..*	R	**Key**: No
	Description: Recursive association to represents hierarchical schemes.			
_switch	**Type**: Switch	**Mult**: 0..*	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: The Switch instances composing the protection scheme.			
_controlParameters	**Type**: ResilienceConstraint	**Mult**: 1	RW	**Key**: No
	Description: The parameters of the protection scheme.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

9.3 Associations

9.3.1 CEPAggregatesCEPs

Association end role name	Aggregation type	Navigable	Target Class	Mult
_aggregatedConnectionEndPoint	shared	Yes	ConnectionEndPoint	0..*
connectionendpoint	none	No	ConnectionEndPoint	0..1

9.3.2 CEPHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	OperationalStatePac	1
_connectionEndPoint	none	No	ConnectionEndPoint	1

9.3.3 CEPHasTerminationPac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_termination	composite	Yes	TerminationPac	1
connectionendpoint	none	No	ConnectionEndPoint	1

9.3.4 CEPIsSupportedByParentNEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_parentNodeEdgePoint	none	Yes	NodeEdgePoint	1
connectionendpoint	none	No	ConnectionEndPoint	0..*

9.3.5 CEPListHasCEPs

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectionEndPoint	composite	Yes	ConnectionEndPoint	0..*
cepholder	none	No	CepList	1

9.3.6 CEPSupportsClientNEPs

Association end role name	Aggregation type	Navigable	Target Class	Mult
_clientNodeEdgePoint	shared	Yes	NodeEdgePoint	0..*
_connectionEndPoint	none	No	ConnectionEndPoint	1

9.3.7 CSEPHasAssembledCSEPs

Association end role name	Aggregation type	Navigable	Target Class	Mult
_assembledConnectivityServiceEndPoint	none	Yes	ConnectivityServiceEndPoint	0..*
connectivityserviceendpoint	none	No	ConnectivityServiceEndPoint	1

9.3.8 CSEPHasCapacityPac

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_capacity	composite	Yes	Capacity	1
connectivityserviceendpoint	none	No	ConnectivityServiceEndPoint	1

9.3.9 CSEPHasForwardingPeerCSEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_peerFwdConnectivityServiceEndPoint	none	Yes	ConnectivityServiceEndPoint	0..1
connectivityserviceendpoint	none	No	ConnectivityServiceEndPoint	1

9.3.10 CSEPHasServerCSEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_serverConnectivityServiceEndPoint	none	Yes	ConnectivityServiceEndPoint	0..1
connectivityserviceendpoint	none	No	ConnectivityServiceEndPoint	1

9.3.11 CSEPHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	AdminStatePac	1
connectivityserviceendpoint	none	No	ConnectivityServiceEndPoint	1

9.3.12 CSEPIsProtectedByCSEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_protectingConnectivityServiceEndPoint	none	Yes	ConnectivityServiceEndPoint	0..1
connectivityserviceendpoint	none	No	ConnectivityServiceEndPoint	1

9.3.13 CSEPRelatesToCEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectionEndPoint	none	Yes	ConnectionEndPoint	0..*
_connectivityServiceEndPoint	none	No	ConnectivityServiceEndPoint	0..1

9.3.14 CSEPTerminatesOnSIP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_serviceInterfacePoint	none	Yes	ServiceInterfacePoint	1
_connServicePort	none	No	ConnectivityServiceEndPoint	0..*

9.3.15 CSIPTerminatesOnNEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_nodeEdgePoint	none	Yes	NodeEdgePoint	1
connectivityserviceinternalpoint	none	No	ConnectivityServiceInternalPoint	0..*

9.3.16 ConnServHasSubordinateConnServ

Useful to specify constraints for subordinate Connectivity Services, e.g. in case of a protection scheme which does not span the whole parent Connectivity Service.

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectivityService	shared	Yes	ConnectivityService	0..*
connectivityservice	none	No	ConnectivityService	1

9.3.17 ConnServiceHasCSEPs

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_endPoint	composite	Yes	ConnectivityServiceEndPoint	2..*
_service	none	No	ConnectivityService	1

9.3.18 ConnServiceHasCSIPs

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_internalPoint	composite	Yes	ConnectivityServiceInternalPoint	0..*
connectivityservice	none	No	ConnectivityService	1

9.3.19 ConnServiceHasConnConstraints

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectivityConstraint	composite	Yes	ConnectivityConstraint	1
_service	none	No	ConnectivityService	1

9.3.20 ConnServiceHasResilienceConstr

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_resilienceConstraint	composite	Yes	ResilienceConstraint	0..1
connectivityservice	none	No	ConnectivityService	1

9.3.21 ConnServiceHasRoutingConstr

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_routingConstraint	composite	Yes	RoutingConstraint	0..1
connectivityservice	none	No	ConnectivityService	1

9.3.22 ConnServiceHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	AdminStatePac	1
_service	none	No	ConnectivityService	1

9.3.23 ConnServiceHasTopLevelConnections

LifecycleAggregate

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connection	shared	Yes	Connection	0..*
_service	none	No	ConnectivityService	0..1

9.3.24 ConnServiceHasTopologyConstraints

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_topologyConstraint	composite	Yes	TopologyConstraint	0..*
connectivityservice	none	No	ConnectivityService	1

9.3.25 ConnTerminatesOnCEP

LifecycleAggregate

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectionEndPoint	shared	Yes	ConnectionEndPoint	2..*
_connPort	none	No	Connection	1..*

9.3.26 ConnectionEncapsulatesSwitchControl

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_switchControl	composite	Yes	SwitchControl	0..*
connection	none	No	Connection	1

9.3.27 ConnectionHasLowerLevelConnections

Association end role name	Aggregation type	Navigable	Target Class	Mult
_lowerConnection	shared	Yes	Connection	0..*
connection	none	No	Connection	0..1

9.3.28 ConnectionHasRoutes

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_route	composite	Yes	Route	0..*
_connection	none	No	Connection	1

9.3.29 ConnectionHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	OperationalStatePac	1
_connection	none	No	Connection	1

9.3.30 ConnectionIsBoundedByNode

Association end role name	Aggregation type	Navigable	Target Class	Mult
_boundingNode	none	Yes	Node	0..1
connection	none	No	Connection	0..*

9.3.31 ConnectionSupportsClientLinks

Association end role name	Aggregation type	Navigable	Target Class	Mult
_supportedClientLink	none	Yes	Link	0..*
_supportingConnection	none	No	Connection	0..*

9.3.32 ConstrHasCorouteIncl

Association end role name	Aggregation type	Navigable	Target Class	Mult
_corouteInclusion	none	Yes	ConnectivityService	0..1
_connectivityConstraint	none	No	ConnectivityConstraint	1

9.3.33 ConstrHasDiversityExcl

Association end role name	Aggregation type	Navigable	Target Class	Mult
_diversityExclusion	none	Yes	ConnectivityService	0..*
_connectivityConstraint	none	No	ConnectivityConstraint	1

9.3.34 ContextHasConnService

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectivityService	composite	Yes	ConnectivityService	0..*
connectivitycontext	none	No	ConnectivityContext	1

9.3.35 ContextHasConnections

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connection	composite	Yes	Connection	0..*
connectivitycontext	none	No	ConnectivityContext	1

9.3.36 ControlChoosesSwitchPosition

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_switch	composite	Yes	Switch	0..*
switchcontrol	none	No	SwitchControl	1

9.3.37 ControlGovernsControls

Association end role name	Aggregation type	Navigable	Target Class	Mult
_subSwitchControl	none	Yes	SwitchControl	0..*
switchcontrol	none	No	SwitchControl	1

9.3.38 ControlHasParameters

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_controlParameters	composite	Yes	ResilienceConstraint	1
switchcontrol	none	No	SwitchControl	1

9.3.39 CsepHasServerConstraint

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_serverConstraint	composite	Yes	ServerConstraint	0..*
connectivityserviceendpoint	none	No	ConnectivityServiceEndPoint	1

9.3.40 ResilienceConstraintHasRouteConstraint

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_resiliencyRouteConstraint	composite	Yes	ResiliencyRouteConstraint	0..*
resilienceconstraint	none	No	ResilienceConstraint	1

9.3.41 ResiliencyRouteConstraintHasRoutingConstraint

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_routingConstraint	composite	Yes	RoutingConstraint	0..1
resiliencyrouteconstraint	none	No	ResiliencyRouteConstraint	1

9.3.42 ResiliencyRouteConstraintHasTopologyConstraint

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_topologyConstraint	none	Yes	TopologyConstraint	0..1
resiliencyrouteconstraint	none	No	ResiliencyRouteConstraint	1

9.3.43 RouteHasResilienceRoute

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_resilienceRoute	composite	Yes	ResilienceRoute	0..1
route	none	No	Route	1

9.3.44 RouteIsDescribedByCEPs

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectionEndPoint	shared	Yes	ConnectionEndPoint	2..*
route	none	No	Route	0..*

9.3.45 SwitchSelectsCEPs

Association end role name	Aggregation type	Navigable	Target Class	Mult
_selectedConnectionEndPoint	none	Yes	ConnectionEndPoint	1..*
switchgroup	none	No	Switch	0..1

9.3.46 SwitchSelectsRoute

Association end role name	Aggregation type	Navigable	Target Class	Mult
_selectedRoute	none	Yes	Route	1..*
switch	none	No	Switch	0..1

9.4 Abstractions

9.4.1 AugmentsRootContext

Augments the base TAPI Context with ConnectivityContext model.

Target Class: "/TapiCommon:Context:_context"

9.4.2 CEPListAugmentsNEP

This augment allows NEP to refer to its CEPs despite TapiTopology model does not import TapiConnectivity model.

Target Class:
"/TapiCommon:Context:_context/TapiTopology:TopologyContext:_topologyContext/TapiTopology:TopologyContext:_topology/TapiTopology:Topology:_node/TapiTopology:Node:_ownedNodeEdgePoint"

9.4.3 CepAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

9.4.4 ConnectionAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

9.4.5 ConnectivityObjectTypeAugmentsObjectType

Enumeration Augment.

9.4.6 ConnectivityServiceAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

9.4.7 CsepAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

9.4.8 RouteAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

9.4.9 SwitchAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

9.4.10 SwitchControlAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

9.5 Data Types

9.5.1 CepRole

The role of the CEP in the context of the Connection spec.

Attribute Name	Attribute Info			
roleName	**Type**: String	**Mult**: 1	R	**Key**: yes – part: 1
	Description: The name of the CEP role in the context of the referenced spec.			
connectionSpecReference	**Type**: ConnectionSpecReference	**Mult**: 1	R	**Key**: No
	Description: The reference to the spec that defines the CEP role.			

9.5.2 ConnectionSpecReference

The reference to a spec for a type of Connection.

Attribute Name	Attribute Info			
connectionSpecName	**Type**: String	**Mult**: 1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: The name of the Connection spec. This can be used alone (with no spec reference) where there is only a paper spec.			
connectionSpecId	**Type**: Uuid	**Mult**: 0..1	R	**Key**: No
	Description: The reference to a formal spec. This reference need not be provided (e.g., where there is no formal machine interpretable spec for the type of Connection).			

9.5.3 ConnectivityServiceSpecReference

The reference to a spec for a type of Connectivity Service

Attribute Name	Attribute Info			
connectivityServiceSpecName	**Type**: String	**Mult**: 1	R	**Key**: No
	Description: The name of the Connectivity Service spec. This can be used alone (with no spec reference) where there is only a paper spec.			
connectivityServiceSpecId	**Type**: Uuid	**Mult**: 1	R	**Key**: No
	Description: The reference to a formal spec. This reference need not be provided (e.g., where there is no formal machine interpretable spec for the type of Connectivity Service).			

9.5.4 CsepRole

The role of the CSEP in the context of the Connectivity Service spec.

Attribute Name	Attribute Info			
roleName	**Type**: String	**Mult**: 1	R	**Key**: yes – part: 1
	Description: The name of the CSEP role in the context of the referenced spec.			
connectivityServiceSpecReference	**Type**: ConnectivityServiceSpecReference	**Mult**: 1	R	**Key**: No
	Description: The reference to the spec that defines the CSEP role.			

9.6 Enumerations

9.6.1 ConnectivityObjectType

The list of TAPI Connectivity Global Object Class types on which Notification signals can be raised.

Contains Enumeration Literals:

- CONNECTIVITY_SERVICE:
  - The ConnectivityService class.
- CONNECTIVITY_SERVICE_END_POINT:
  - The ConnectivityServiceEndPoint (CSEP) class.
- CONNECTION:
  - The Connection class.
- CONNECTION_END_POINT:
  - The ConnectionEndPoint (CEP) class.
- SWITCH_CONTROL:
  - The SwitchControl class.
- SWITCH:
  - The Switch class.
- ROUTE:
  - The Route class.
- RESILIENCE_CONSTRAINT:
  - The ResilienceConstraint class.
- RESILIENCE_ROUTE:
  - The ResilienceRoute class.
- RESILIENCE_ROUTE_CONSTRAINT:
  - The ResilienceRouteConstraint class.
- SERVER_CONSTRAINT:
  - The ServerConstraint class.

9.6.2 CoordinateType

The types of coordination mechanisms between protection/restoration operations across multiple layers.

Contains Enumeration Literals:

- NO_COORDINATE:
  - No coordination, i.e. each layer network restores independently.
- HOLD_OFF_TIME:
  - The client layer network protection/restoration process is suspended for a certain time to possibly allow server layer network to protect/restore, avoiding useless multi-layer protection/restoration. It is assumed that the server layer network successful protection/restoration operation will inherently cancel the protection/restoration trigger at client layer.
- WAIT_FOR_NOTIFICATION:
  - The client layer network protection/restoration process is suspended until a notification is received from the server layer protection/restoration process. The notification should inform about the success or failure of the protection/restoration process at server layer.

9.6.3 FaultConditionDetermination

ITU-T G.808 Amendment 1 (03/2018) - 3.2.6.8 subnetwork connection protection: "Transport entity protection for the case where the transport entity is a subnetwork connection. The serial compound link connection within the subnetwork connection is protected by adding bridges and selectors in the connection functions at the edges of the protected domain and an additional serial compound link connection between these connection functions. The determination of a fault condition on a serial compound link connection within the protected domain can be performed as follows: (see enumeration entries)."

Contains Enumeration Literals:

- INHERENT:
  - Inherent monitored (/I): The fault condition status of each link connection is derived from the status of the underlying server layer trail.
- NON_INTRUSIVE:
  - Non-intrusive monitored (/N): Each serial compound link connection is extended with a non-intrusive monitoring termination sink function to derive the fault condition status from the traffic signal that is present.
- SUBLAYER:
  - Sublayer monitored (/S): Each serial compound link connection is extended with tandem connection monitoring or segment termination/adaptation functions to derive the fault condition status independent of the traffic signal present.
- TEST:
  - Test monitored (/T): Each serial compound link connection's fault condition status is derived from an additional monitored serial compound link connection transported via the same serial compound link.

9.6.4 ProtectionRole

The protection role of a (conceptual) port of a forwarding entity, e.g. Link, ConnectivityService, Connection, PathComputationService, Path, VirtualNetworkService.

Contains Enumeration Literals:

- WORK:
  - The unreliable/unprotected resource is assumed to be the preferred/intended/nominal/highest priority for usage.
- PROTECT:
  - The unreliable/unprotected resource is assumed to be the spare/protection of a higher priority resource.
- PROTECTED:
  - The resource which is reliable/protected/resilient by the protection/restoration scheme.
- NA:
  - Protection role not applicable to the resource.
- WORK_RESTORE:
  - The unreliable/unprotected resource is assumed to be the preferred/intended/nominal/highest priority for usage. Revertive behavior.
- PROTECT_RESTORE:
  - The unreliable/unprotected resource is assumed to be the spare/protection of a higher priority resource. Revertive behavior.

9.6.5 ReversionMode

The reversion mode associated with protection scheme.

Contains Enumeration Literals:

- REVERTIVE:

  - A Connection switched to a lower priority (non-preferred/spare/protection) resource will revert to a higher priority (preferred/intended/nominal) resource when that recovers (potentially after some wait-to-restore time).
- NON-REVERTIVE:
  - A Connection switched to a lower priority (non-preferred/spare/protection) resource will not revert to a higher priority (preferred/intended/nominal) resource when that recovers. This mode is typically applied when there is no ranking between the redundant resources.

9.6.6 RouteState

Potential Route states concerning the service support.

Contains Enumeration Literals:

- CURRENT:
  - The Route instance identified is the current Route, i.e., is the one that is active and selected to support service.
- NOT_CURRENT:
  - The Route instance is not the one supporting the service.
- UNKNOWN:
  - The Route state is unknown.

9.6.7 SelectionControl

Possible degrees of administrative control applied to the Route selection.

Contains Enumeration Literals:

- LOCK_OUT:
  - The resource is configured to temporarily not be available for use in the protection/restoration scheme(s) it is part of. This overrides all other protection/restoration control states including "forced". If the item is locked out then it cannot be used under any circumstances. Note: Only relevant when part of a protection/restoration scheme.
- NORMAL:
  - Remove of any previous administrative command.
- MANUAL:
  - The traffic is temporarily switched to the spare/protection resource, unless it is in a fault condition state. Note: Only relevant when part of a protection/restoration scheme.
- FORCED:
  - The traffic is temporarily switched to the spare/protection resource, regardless its fault condition state. Note: Only relevant when part of a protection/restoration scheme.

9.6.8 SelectionReason

The cause of the current Route selection.

Contains Enumeration Literals:

- LOCKOUT:
  - A "lockout" administrative command has been issued.

- NORMAL:
  - No administrative command currently issued.
- MANUAL:
  - A "manual" administrative command has been issued.
- FORCED:
  - A "forced" administrative command has been issued.
- WAIT_TO_REVERT:
  - The scheme is waiting for reversion to preferred/intended/nominal resource.
- SIGNAL_DEGRADE:
  - A "signal degrade" condition is active.
- SIGNAL_FAIL:
  - A "signal fail" condition is active.

9.6.9 ServiceType

List of simple connectivity types.

Contains Enumeration Literals:

- POINT_TO_POINT_CONNECTIVITY:
  - Point to point.
- POINT_TO_MULTIPOINT_CONNECTIVITY:
  - Point to multipoint.
- MULTIPOINT_CONNECTIVITY:
  - Multipoint to multipoint.
- ROOTED_MULTIPOINT_CONNECTIVITY:
  - Rooted multipoint.

10 Path Computation Model

10.1 Diagrams

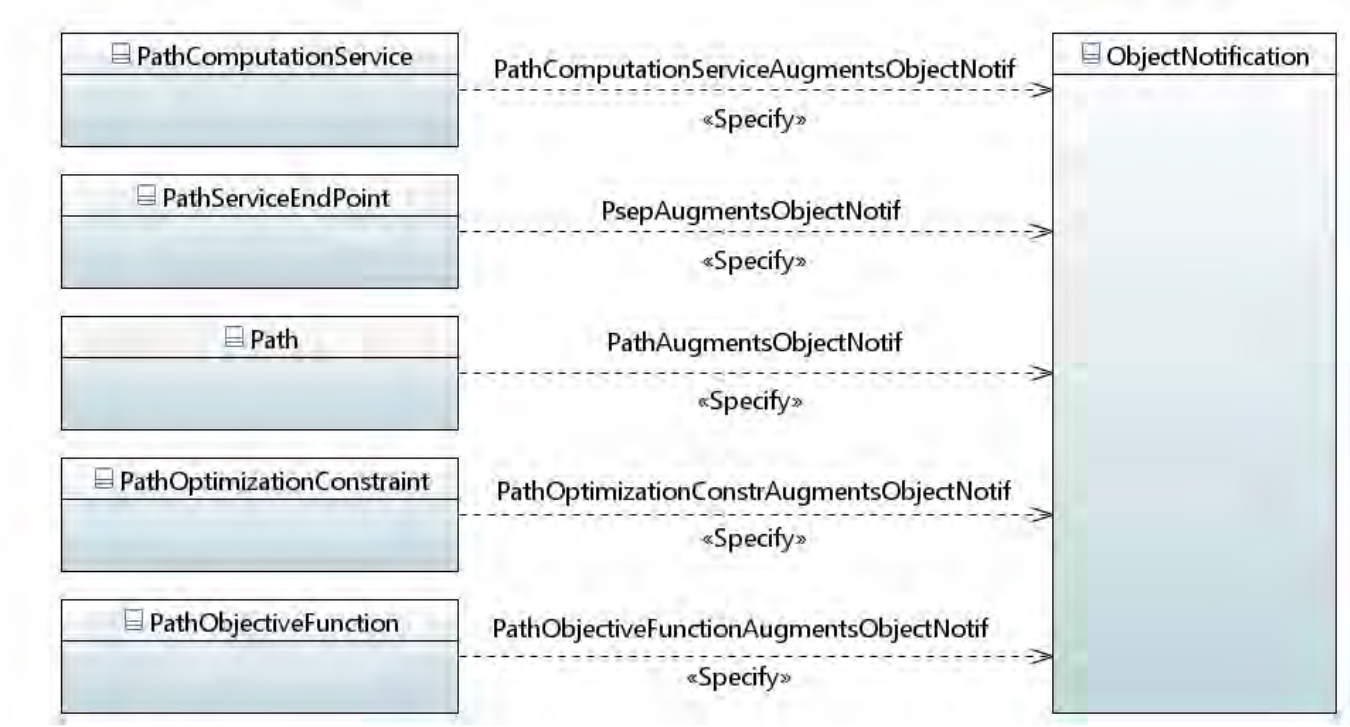

Figure 48 – *PathComputationNotif*

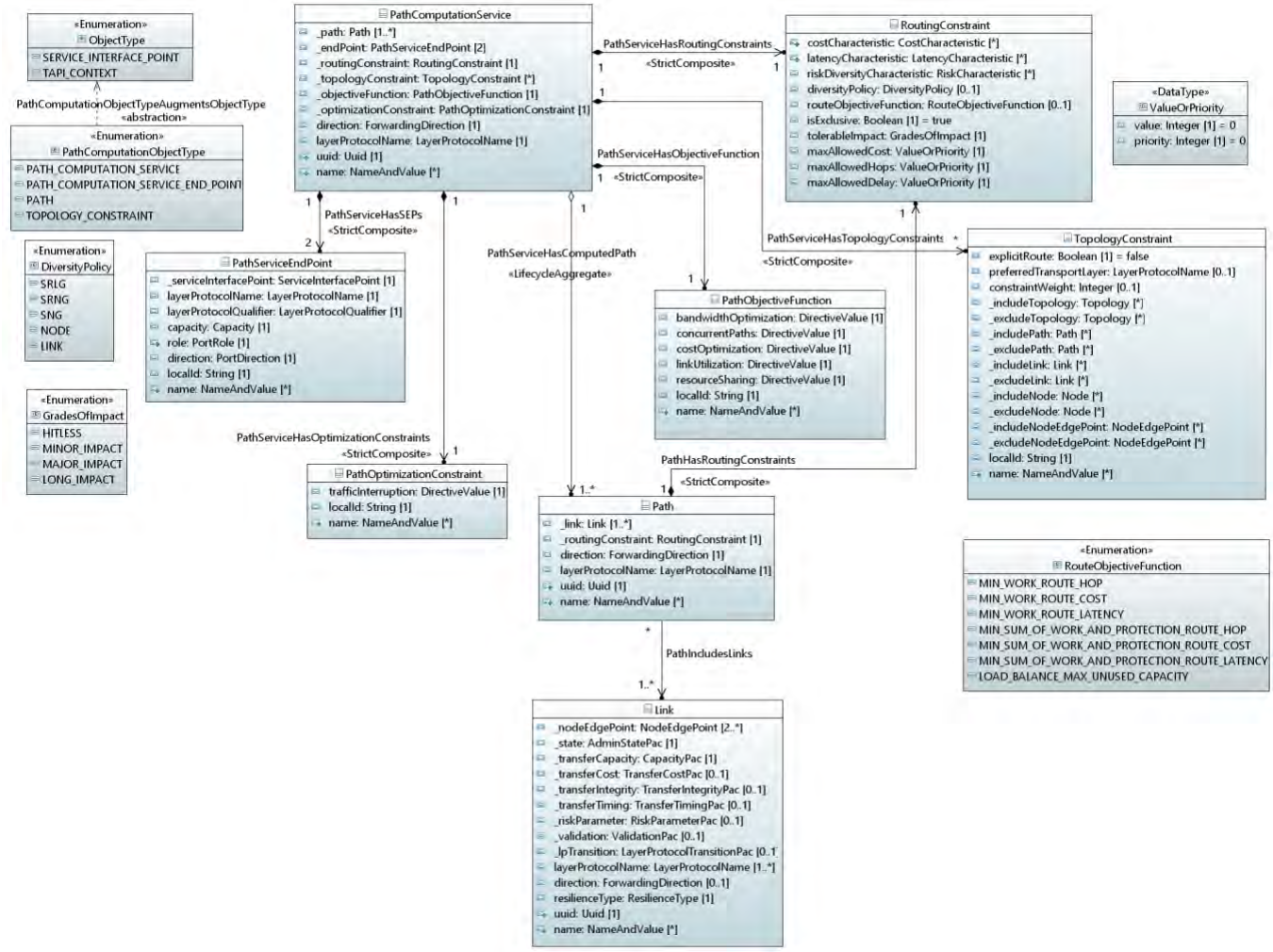

Figure 49 – *PathComputationServiceDetails*

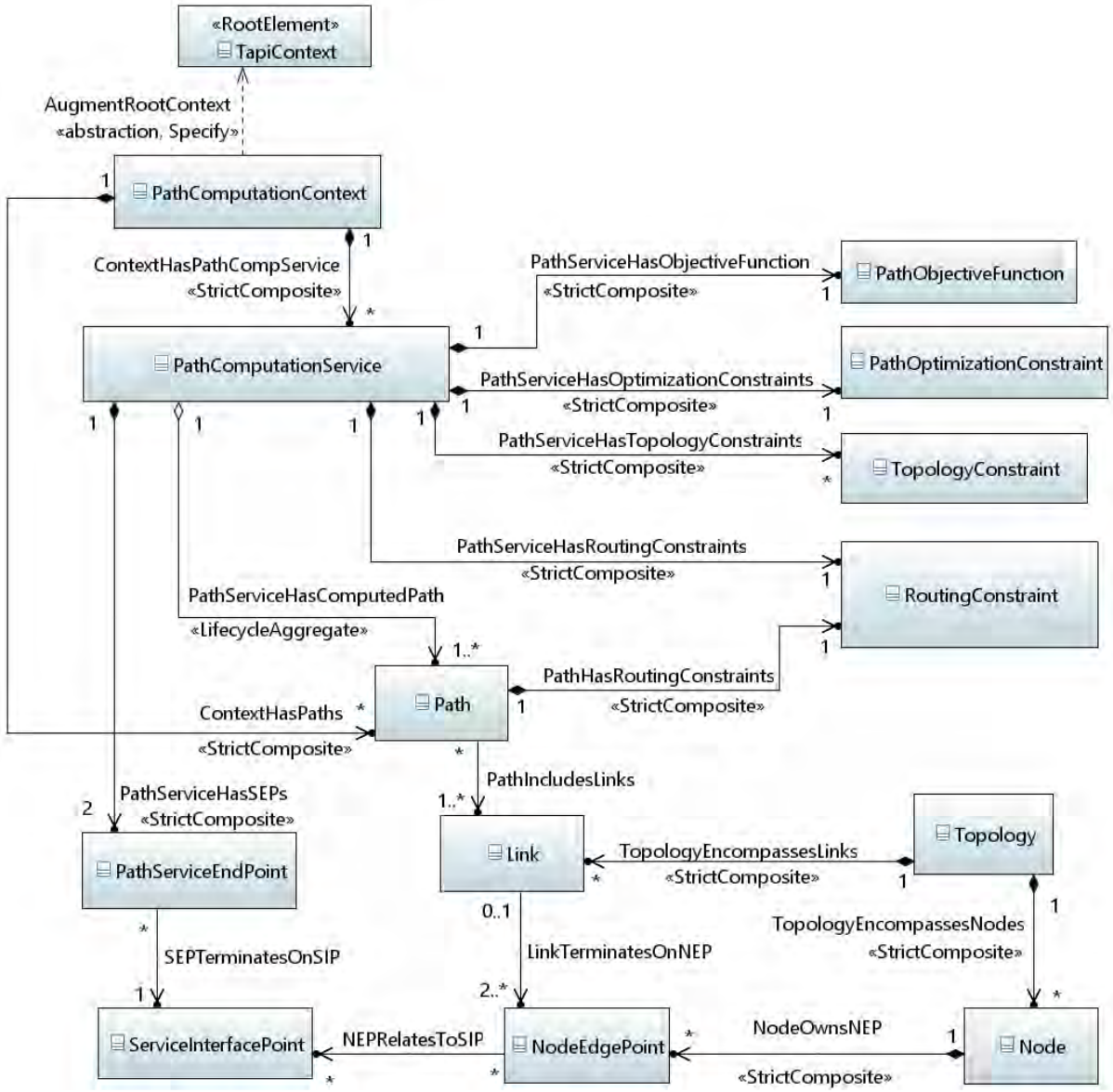

Figure 50 – *PathComputationServiceSkeleton*

10.2 Classes

10.2.1 Path

The Path is described by an ordered list of (TE) Links. A (TE) Link is conceptually defined by a pair of Node/NodeEdgePoint IDs. A Connection is realized by concatenating link resources (associated with a Link) and the lower-level Connections (e.g. cross-connections) in the different Nodes.

Attribute Name	Attribute Info			
_link	**Type**: Link	**Mult**: 1..*	R	**Key**: No

Attribute Name	Attribute Info			
	Description: The list of Link instances composing the Path instance.			
_routingConstraint	**Type**: RoutingConstraint	**Mult**: 1	R	**Key**: No
	Description: The associated routing constraints.			
direction	**Type**: ForwardingDirection	**Mult**: 1	R	**Key**: No
	Description: The forwarding direction of the Path.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	R	**Key**: No
	Description: The layer protocol of the Path.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

10.2.2 PathComputationContext

This object class represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the path computation description. An instance of this class includes its PathComputationService and Path object instances.

Attribute Name	Attribute Info			
_pathCompService	**Type**: PathComputationService	**Mult**: 0..*	RW	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The included PathComputationService instances.			
_path	**Type**: Path	**Mult**: 0..*	R	**Key**: No **Condition**: tapi-topology:full-topology-management
	Description: The included Path instances.			

10.2.3 PathComputationService

A PathComputationService represents an "intent-like" request for connectivity between two or more PathServiceEndPoint (PSEP) instances. The PathComputationService is a container for connectivity request details and is distinct from the Path(s) that realize the request.

Attribute Name	Attribute Info			
_path	**Type**: Path	**Mult**: 1..*	R	**Key**: No
	Description: The Path instance(s) tracking the state of the identified resources for the support of the PathComputationService.			
_endPoint	**Type**: PathServiceEndPoint	**Mult**: 2	RW	**Key**: No
	Description: The PathServiceEndPoint (PSEP) instances of the PathComputationService.			
_routingConstraint	**Type**: RoutingConstraint	**Mult**: 1	RW	**Key**: No
	Description: The associated routing constraints.			
_topologyConstraint	**Type**: TopologyConstraint	**Mult**: 0..*	RW	**Key**: No
	Description: The associated topology constraints. Different instances of TopologyConstraints may be used to specify constraints at different layer networks.			
_objectiveFunction	**Type**: PathObjectiveFunction	**Mult**: 1	RW	**Key**: No
	Description: The associated objective functions.			
_optimizationConstraint	**Type**: PathOptimizationConstraint	**Mult**: 1	RW	**Key**: No
	Description: The associated optimization constraints.			
direction	**Type**: ForwardingDirection	**Mult**: 1	RW	**Key**: No
	Description: The forwarding direction of the PathComputationService.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol of the PathComputationService.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1

Attribute Name	Attribute Info			
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

10.2.4 PathObjectiveFunction

The parameters defining the objective functions.

Attribute Name	Attribute Info			
bandwidthOptimization	**Type**: DirectiveValue	**Mult**: 1	R	**Key**: No
	Description: The directive types regarding bandwidth optimization.			
concurrentPaths	**Type**: DirectiveValue	**Mult**: 1	R	**Key**: No
	Description: The directive types regarding concurrent paths.			
costOptimization	**Type**: DirectiveValue	**Mult**: 1	R	**Key**: No
	Description: The directive types regarding cost optimization.			
linkUtilization	**Type**: DirectiveValue	**Mult**: 1	R	**Key**: No
	Description: The directive types regarding link utilization.			
resourceSharing	**Type**: DirectiveValue	**Mult**: 1	R	**Key**: No
	Description: The directive types regarding resource sharing.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No

Attribute Name	Attribute Info
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.

10.2.5 PathOptimizationConstraint

The parameters defining the optimization constraints.

Attribute Name	Attribute Info			
trafficInterruption	**Type**: DirectiveValue	**Mult**: 1	R	**Key**: No
	Description: The directive types regarding traffic interruption.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

10.2.6 PathServiceEndPoint

The PathServiceEndPoint (PSEP) encapsulates information related to a PathComputationService at the ingress/egress points of that PathComputationService.

Attribute Name	Attribute Info			
_serviceInterfacePoint	**Type**: ServiceInterfacePoint	**Mult**: 1	RW	**Key**: No
	Description: The supporting ServiceInterfacePoint (SIP) instance.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol of the PathServiceEndPoint (PSEP).			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 1	RW	**Key**: No
	Description: The layer protocol qualifier of the PathServiceEndPoint (PSEP).			
capacity	**Type**: Capacity	**Mult**: 1	RW	**Key**: No
	Description: The PathServiceEndPoint (PSEP) capacity.			

Attribute Name	Attribute Info			
role	**Type**: PortRole	**Mult**: 1	RW	**Key**: No
	Description: The role of the (conceptual) port of the associated PathComputationService.			
direction	**Type**: PortDirection	**Mult**: 1	RW	**Key**: No
	Description: The orientation of flow at the (conceptual) port of the associated PathComputationService.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

10.2.7 RoutingConstraint

The parameters of the routing constraints.

Attribute Name	Attribute Info			
costCharacteristic	**Type**: CostCharacteristic	**Mult**: 0..*	RW	**Key**: No
	Description: The list of costs where each cost relates to some aspect of a topological entity.			
latencyCharacteristic	**Type**: LatencyCharacteristic	**Mult**: 0..*	RW	**Key**: No
	Description: The effect on the latency of a queuing process. This only has significant effect for packet based systems and has a complex characteristic.			
riskDiversityCharacteristic	**Type**: RiskCharacteristic	**Mult**: 0..*	RW	**Key**: No
	Description: The diversity risk characteristics.			
diversityPolicy	**Type**: DiversityPolicy	**Mult**: 0..1	RW	
	Description: The diversity policies.			
routeObjectiveFunction	**Type**: RouteObjectiveFunction	**Mult**: 0..1	RW	
	Description: The route objective functions.			
isExclusive	**Type**: Boolean	**Mult**: 1	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: To distinguish if the resources are to be exclusive to the service.			
tolerableImpact	**Type**: GradesOfImpact	**Mult**: 1	RW	**Key**: No
	Description: Grades of maximum tolerable disruption to traffic.			
maxAllowedCost	**Type**: ValueOrPriority	**Mult**: 1	RW	**Key**: No
	Description: The specification of the maximum allowed cost.			
maxAllowedHops	**Type**: ValueOrPriority	**Mult**: 1	RW	**Key**: No
	Description: The specification of the maximum allowed hops.			
maxAllowedDelay	**Type**: ValueOrPriority	**Mult**: 1	RW	**Key**: No
	Description: The specification of the maximum allowed delay, value in microseconds.			

10.2.8 TopologyConstraint

The TopologyConstraint class allows to specify topology entities in order to impose specific constraints (as denoted by the attribute name) on ConnectivityService/PathComputationService realization. The topology entities are specified by their instance UUID rather than using references/path (to allow for mapping to Yang 1.0). This loose typing and reference necessitates that implementations validate not only the presence of the instance, but also that it is of the correct type as implied by the attribute name. If this validation fails, then the implementation is expected to return an error.

Attribute Name	Attribute Info			
explicitRoute	**Type**: Boolean	**Mult**: 1	RW	**Key**: No
	Description: If true, indicates that the route constraints are specified with full detail, i.e. no need for further route computation.			
preferredTransportLayer	**Type**: LayerProtocolName	**Mult**: 0..1	RW	
	Description: Soft constraint requested by client to indicate the layer of transport connection that it prefers to carry the service. This could be same as the service layer or one of the supported server layers.			
constraintWeight	**Type**: Integer	**Mult**: 0..1	RW	**Key**: No
	Description: Zero and positive values: zero means "strongly required to be included", +1 means "less strongly required to be included", etc. For example the work/intended route will be calculated considering the topologies which weights are lowest (but not negative). Negative values: -1 means "strongly required to be excluded", -2 means "less strongly required to be excluded", etc.			

Attribute Name	Attribute Info			
_includeTopology	**Type**: Topology	**Mult**: 0..*	RW	**Key**: No
	Description: The Topology instance to be included in the connectivity route.			
_excludeTopology	**Type**: Topology	**Mult**: 0..*	RW	**Key**: No
	Description: The Topology instance to be excluded from the connectivity route.			
_includePath	**Type**: Path	**Mult**: 0..*	RW	**Key**: No
	Description: The Path instance to be followed by the connectivity route. The type is generic UUID given read/write constraints, the Path is a readonly node.			
_excludePath	**Type**: Path	**Mult**: 0..*	RW	**Key**: No
	Description: The Path instance to be excluded from the connectivity route. The type is generic UUID given read/write constraints, the Path is a readonly node.			
_includeLink	**Type**: Link	**Mult**: 0..*	RW	**Key**: No
	Description: The Link instance to be included in the connectivity route.			
_excludeLink	**Type**: Link	**Mult**: 0..*	RW	**Key**: No
	Description: The Link instance to be excluded from the connectivity route.			
_includeNode	**Type**: Node	**Mult**: 0..*	RW	**Key**: No
	Description: The Node instance to be included in the connectivity route.			
_excludeNode	**Type**: Node	**Mult**: 0..*	RW	**Key**: No
	Description: The Node instance to be excluded from the connectivity route.			
_includeNodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 0..*	RW	**Key**: No
	Description: The NodeEdgePoint (NEP) instance to be included in the connectivity route.			
_excludeNodeEdgePoint	**Type**: NodeEdgePoint	**Mult**: 0..*	RW	**Key**: No
	Description: The NodeEdgePoint (NEP) instance to be excluded from the connectivity route.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1

Attribute Name	Attribute Info			
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

10.3 Associations

10.3.1 ContextHasPathCompService

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_pathCompService	composite	Yes	PathComputationService	0..*
pathcomputationcontext	none	No	PathComputationContext	1

10.3.2 ContextHasPaths

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_path	composite	Yes	Path	0..*
pathcomputationcontext	none	No	PathComputationContext	1

10.3.3 PathHasRoutingConstraints

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_routingConstraint	composite	Yes	RoutingConstraint	1
_path	none	No	Path	1

10.3.4 PathIncludesLinks

Association end role name	Aggregation type	Navigable	Target Class	Mult
_link	none	Yes	Link	1..*
_path	none	No	Path	0..*

10.3.5 PathServiceHasComputedPath

LifecycleAggregate

Association end role name	Aggregation type	Navigable	Target Class	Mult
_path	shared	Yes	Path	1..*
_pathService	none	No	PathComputationService	1

10.3.6 PathServiceHasObjectiveFunction

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_objectiveFunction	composite	Yes	PathObjectiveFunction	1

Association end role name	Aggregation type	Navigable	Target Class	Mult
_path	none	No	PathComputationService	1

10.3.7 PathServiceHasOptimizationConstraints

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_optimizationConstraint	composite	Yes	PathOptimizationConstraint	1
_path	none	No	PathComputationService	1

10.3.8 PathServiceHasRoutingConstraints

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_routingConstraint	composite	Yes	RoutingConstraint	1
_pathService	none	No	PathComputationService	1

10.3.9 PathServiceHasSEPs

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_endPoint	composite	Yes	PathServiceEndPoint	1
_service	none	No	PathComputationService	1

10.3.10 PathServiceHasTopologyConstraints

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_topologyConstraint	composite	Yes	TopologyConstraint	0..*
pathcomputationservice	none	No	PathComputationService	1

10.3.11 SEPTerminatesOnSIP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_serviceInterfacePoint	none	Yes	ServiceInterfacePoint	1
_pathServicePort	none	No	PathServiceEndPoint	0..*

10.4 Abstractions

10.4.1 AugmentRootContext

Augments the base TAPI Context with PathComputationService model.

Target Class: "/TapiCommon:Context:_context"

10.4.2 PathAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

10.4.3 PathComputationObjectTypeAugmentsObjectType

Enumeration Augment.

10.4.4 PathComputationServiceAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

10.4.5 PathObjectiveFunctionAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

10.4.6 PathOptimizationConstrAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

10.4.7 PsepAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

10.5 Data Types

10.5.1 ValueOrPriority

Quantitative target: when a value is specified it is intended as mandatory for fulfilment. If value is specified, priority is not considered. Qualitative target: when priority is specified. Zero means "unspecified", 1 is highest priority, then 2 has lower priority than 1, 3 has lower priority than 2, etc.

Attribute Name	Attribute Info			
value	**Type**: Integer	**Mult**: 1	RW	**Key**: No
	Description: The specified value.			
priority	**Type**: Integer	**Mult**: 1	RW	**Key**: No
	Description: The specified priority.			

10.6 Enumerations

10.6.1 DiversityPolicy

The types of routing diversity policies.

Contains Enumeration Literals:

- SRLG:
  - Shared Risk Link Group.
- SRNG:
  - Shared Risk Node Group.
- SNG:
  - Shared Node Group.
- NODE:
  - Diversity with respect to involved Node instances.
- LINK:
  - Diversity with respect to involved Link instances.

10.6.2 GradesOfImpact

The grades of impact on traffic.

Contains Enumeration Literals:

- HITLESS:
  - No impact on traffic.
- MINOR_IMPACT:
  - Impact less or equal to 50ms.
- MAJOR_IMPACT:
  - Impact order of magnitude: several seconds to minutes.
- LONG_IMPACT:
  - Impact order of magnitude: several minutes to hours.

10.6.3 PathComputationObjectType

The list of TAPI Path Computation Global Object Class types on which Notification signals can be raised.

Contains Enumeration Literals:

- PATH_COMPUTATION_SERVICE:
  - The PathComputationService class.
- PATH_COMPUTATION_SERVICE_END_POINT:
  - The PathServiceEndPoint (PSEP) class.
- PATH:
  - The Path class.
- TOPOLOGY_CONSTRAINT:
  - The TopologyConstraint class.

10.6.4 RouteObjectiveFunction

The types of route objective function.

Contains Enumeration Literals:

- MIN_WORK_ROUTE_HOP:
  - Minimize the number of hops in the working/preferred/intended route.
- MIN_WORK_ROUTE_COST:
  - Minimize the routing cost in the working/preferred/intended route.
- MIN_WORK_ROUTE_LATENCY:
  - Minimize the latency in the working/preferred/intended route.
- MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_HOP:
  - Minimize the total number of hops of the working/preferred/intended and spare/protection routes.
- MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_COST:
  - Minimize the total cost of the working/preferred/intended and spare/protection routes.
- MIN_SUM_OF_WORK_AND_PROTECTION_ROUTE_LATENCY:
  - Minimize the total latency of the working/preferred/intended and spare/protection routes.
- LOAD_BALANCE_MAX_UNUSED_CAPACITY:
  - Balance the unused capacity of the working/preferred/intended and spare/protection routes.

11 Fault Management Model

11.1 Diagrams

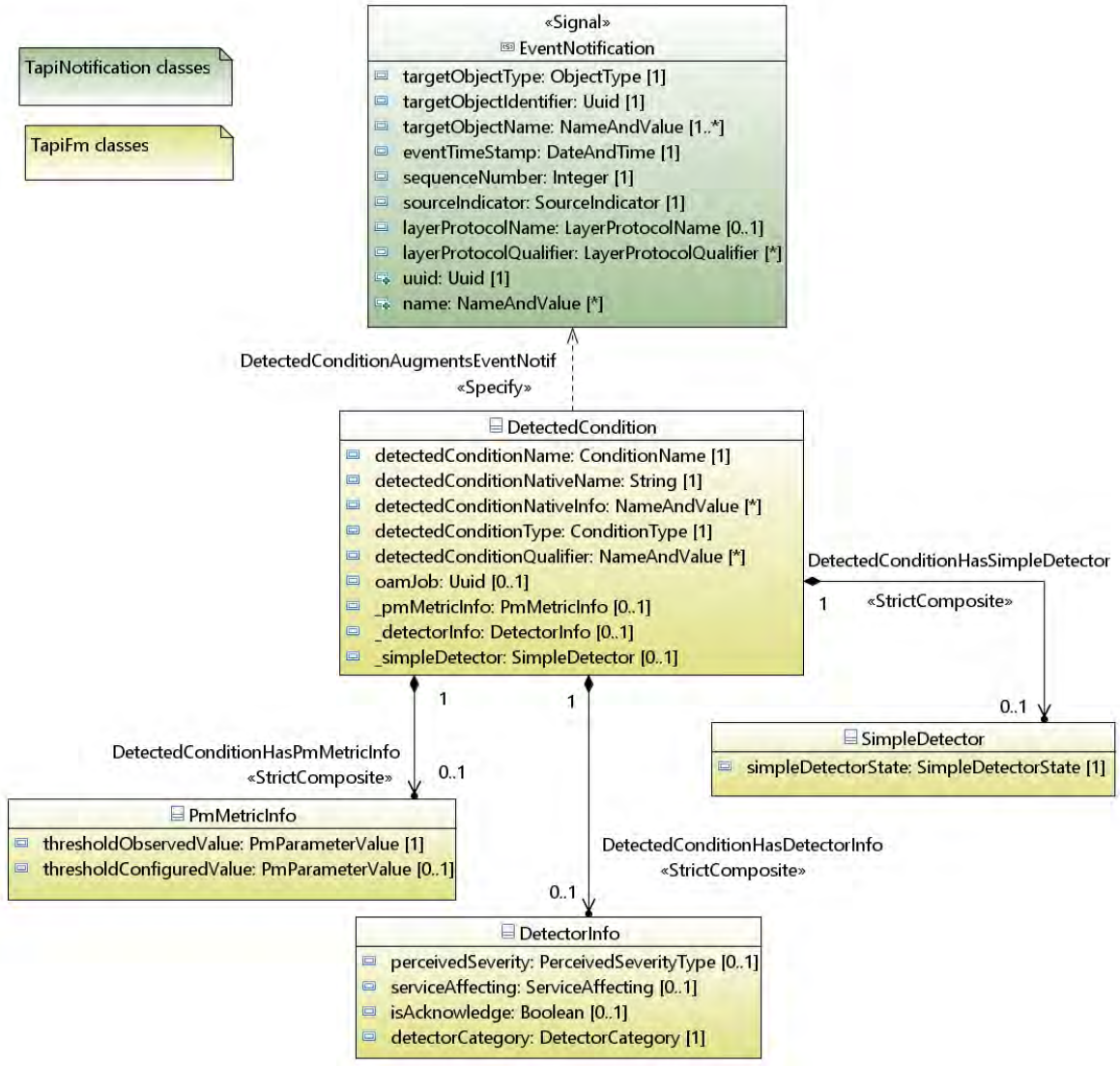

Figure 51 – *FmDetails*

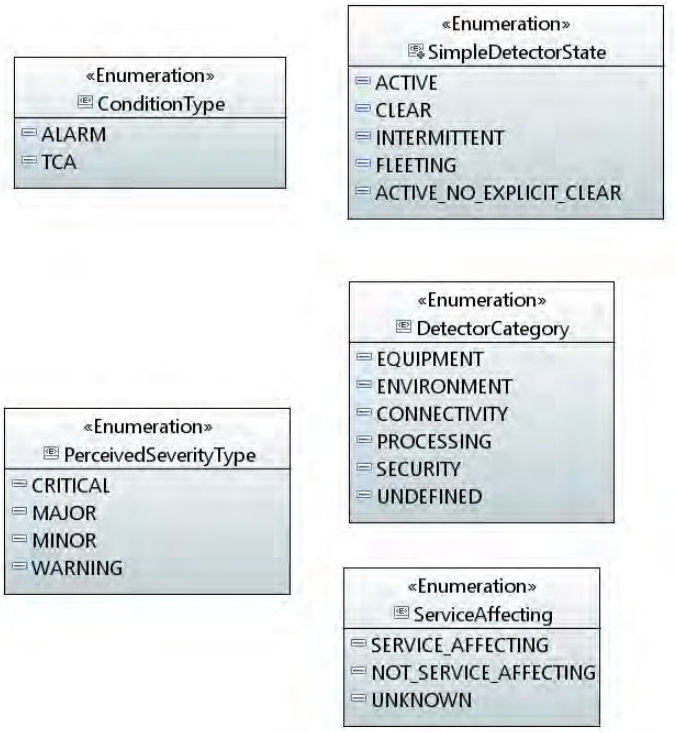

Figure 52 – *FmTypes*

11.2 Classes

11.2.1 DetectedCondition

A record of the state of a Detector where that Detector has two underling states that are of asymmetric importance. For example, an alarm or a threshold crossing alert detected on a given resource. A Condition Detector represents any monitoring component that assesses properties of something and determines from those properties what conditions are associated with the thing. For example, a thing might be "too hot" or might be "unreliable".

Attribute Name	Attribute Info			
detectedConditionName	**Type**: ConditionName	**Mult**: 1	RW	**Key**: No
	Description: The name of the Condition, e.g. an alarm probable cause or the PM metric name which threshold crossing alert refers to. ITU-T probable cause of the failure (detected fault). G.806: - fault: A fault is the inability of a function to perform a required action. This does not include an inability due to preventive maintenance, lack of external resources or planned actions. - fault cause: A single disturbance or fault may lead to the detection of multiple defects. - defect: The density of anomalies has reached a level where the ability to perform a required function has been interrupted. Defects are used as input for performance monitoring, the control of consequent actions and for the determination of fault causes. A fault cause is the result of a correlation process which is intended to identify the defect that is representative of the disturbance or fault that is causing the problem. - failure: The fault cause persisted long enough to consider the ability of an item to perform a required function to be terminated. The item may be considered as failed; a fault has now been detected. - alarm: A human-observable indication that draws attention to a failure (detected fault) usually giving an indication of the severity of the fault.			

Attribute Name	Attribute Info			
detectedConditionNativeName	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: The name used for the Condition by the source of the information.			
detectedConditionNativeInfo	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: Additional info of the Condition provided by the source of the information.			
detectedConditionType	**Type**: ConditionType	**Mult**: 1	RW	**Key**: No
	Description: The type of the Condition.			
detectedConditionQualifier	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: Further information necessary to precisely/uniquely/unambiguously identify the Condition Regarding ITU-T X.733 Alarm Category: For Equipment and Processing Alarm Category, e.g. the local id of the ActualNonFieldReplaceableModule which identifies exact alarm source. For Environment Alarm Category, e.g. on the same Device instance may appear more Environmental alarm notifications with same Alarn Name. For Connectivity Alarm Category in case that same CEP instance includes e.g. both OTS and OMS monitoring layers.			
oamJob	**Type**: Uuid	**Mult**: 0..1	RW	**Key**: No
	Description: Reference to the OamJob instance for which the Condition detection has been configured, e.g. configuration of PM metrics and threshold values and/or of the (alarm) Conditions. The reference is defined as simple UUID because TapiFm does not import TapiOam. MEF 35.1: Identification of the PM Session for which the TCA Function was configured.			
_pmMetricInfo	**Type**: PmMetricInfo	**Mult**: 0..1	RW	**Key**: No
	Description: The PM metric information.			
_detectorInfo	**Type**: DetectorInfo	**Mult**: 0..1	RW	**Key**: No
	Description: The detector info for alarm and TCA.			
_simpleDetector	**Type**: SimpleDetector	**Mult**: 0..1	RW	**Key**: No
	Description: The simple detector state.			

11.2.2 DetectorInfo

(Legacy) information associated to a Condition (alarm).

Attribute Name	Attribute Info			
perceivedSeverity	**Type**: PerceivedSeverityType	**Mult**: 0..1	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: The severity of the detected Condition.			
serviceAffecting	**Type**: ServiceAffecting	**Mult**: 0..1	RW	**Key**: No
	Description: The impact on the service.			
isAcknowledge	**Type**: Boolean	**Mult**: 0..1	RW	**Key**: No
	Description: Information on operator acknowledgement.			
detectorCategory	**Type**: DetectorCategory	**Mult**: 1	RW	**Key**: No
	Description: The Detector (alarm) category, based on ITU-T X.733.			

11.2.3 PmMetricInfo

Information associated to a Threshold Crossing Alert.

Attribute Name	Attribute Info			
thresholdObservedValue	**Type**: PmParameterValue	**Mult**: 1	RW	**Key**: No
	Description: The observed value of PM metric to which TCA refers to.			
thresholdConfiguredValue	**Type**: PmParameterValue	**Mult**: 0..1	RW	**Key**: No
	Description: The configured threshold value of PM metric to which TCA refers to.			

11.2.4 SimpleDetector

Information regarding the (simple) state of the Detector.

Attribute Name	Attribute Info			
simpleDetectorState	**Type**: SimpleDetectorState	**Mult**: 1	RW	**Key**: No
	Description: The (simple) state of the Detector. The Detector state accounts for the time characteristics of the detected Condition.			

11.3 Associations

11.3.1 DetectedConditionHasDetectorInfo

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_detectorInfo	composite	Yes	DetectorInfo	0..1
detectedcondition	none	No	DetectedCondition	1

11.3.2 DetectedConditionHasPmMetricInfo

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_pmMetricInfo	composite	Yes	PmMetricInfo	0..1
detectedcondition	none	No	DetectedCondition	1

11.3.3 DetectedConditionHasSimpleDetector

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_simpleDetector	composite	Yes	SimpleDetector	0..1
detectedcondition	none	No	DetectedCondition	1

11.4 Abstractions

11.4.1 DetectedConditionAugmentsEventNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification"
Condition: "detected-condition-management"

11.5 Enumerations

11.5.1 ConditionType

The types of the Condition.

Contains Enumeration Literals:

- ALARM:
- TCA:
  - Threshold Crossing Alert

11.5.2 DetectorCategory

The Detector (alarm) category, based on ITU-T X.733.

Contains Enumeration Literals:

- EQUIPMENT:
- ENVIRONMENT:
- CONNECTIVITY:
- PROCESSING:
- SECURITY:

- UNDEFINED:

11.5.3 PerceivedSeverityType

The types of perceived severity. ITU-T G.7710: Failures may have been categorized to indicate the severity or urgency of the fault.

Contains Enumeration Literals:

- CRITICAL:
  - ITU-T G.7710/X.733/M.3100: Indication for a service-affecting condition. Immediate corrective action is required.
- MAJOR:
  - ITU-T G.7710/X.733/M.3100: Indication for a service-affecting condition. Urgent corrective action is required.
- MINOR:
  - ITU-T G.7710/X.733/M.3100: Indication for a non-service-affecting condition. Corrective action should be taken in order to prevent more serious fault.
- WARNING:
  - ITU-T G.7710/X.733/M.3100: Indication for a potential or impending service-affecting fault. Further diagnosis should be made.

11.5.4 ServiceAffecting

The possible impact on the service.

Contains Enumeration Literals:

- SERVICE_AFFECTING:
  - The service is affected by the detected Condition.
- NOT_SERVICE_AFFECTING:
  - The service is not affected by the detected Condition.
- UNKNOWN:
  - The impact on the service is unknown.

11.5.5 SimpleDetectorState

The states of the detector.

Contains Enumeration Literals:

- ACTIVE:
  - The detector is indicating the operation of the monitored entity is not within acceptable bounds with respect to the specific condition measured. If INTERMITTENT is supported there may be a requirement for persisted unacceptable operation after a problem occurs before ACTIVE is declared. An alternative may be to declare INTERMITTENT. Where INTERMITTENT is supported, ACTIVE indicates the stable presence of a problem.
- CLEAR:
  - The detector is indicating the operation of the monitored entity is within acceptable bounds with respect to the specific condition measured.
- INTERMITTENT:

- The detector is indicating the operation of the monitored entity is intermittently not within acceptable bounds with respect to the specific condition measured. INTERMITTENT support is optional. Where it is supported there may be a requirement for persisted unacceptable operation after a problem occurs before ACTIVE or INTERMITTENT is declared.

- FLEETING:
  - Event has a very short life (Active-Clear), hence is notified/streamed after its occurrence.
- ACTIVE_NO_EXPLICIT_CLEAR:
  - Same as Active, but an explicit transition to Clear is not foreseen. This e.g. applies to PM metrics which can only increase (counters), hence the "clear" criteria is conventionally the end of a measurement period.

12 OAM Model

12.1 Diagrams

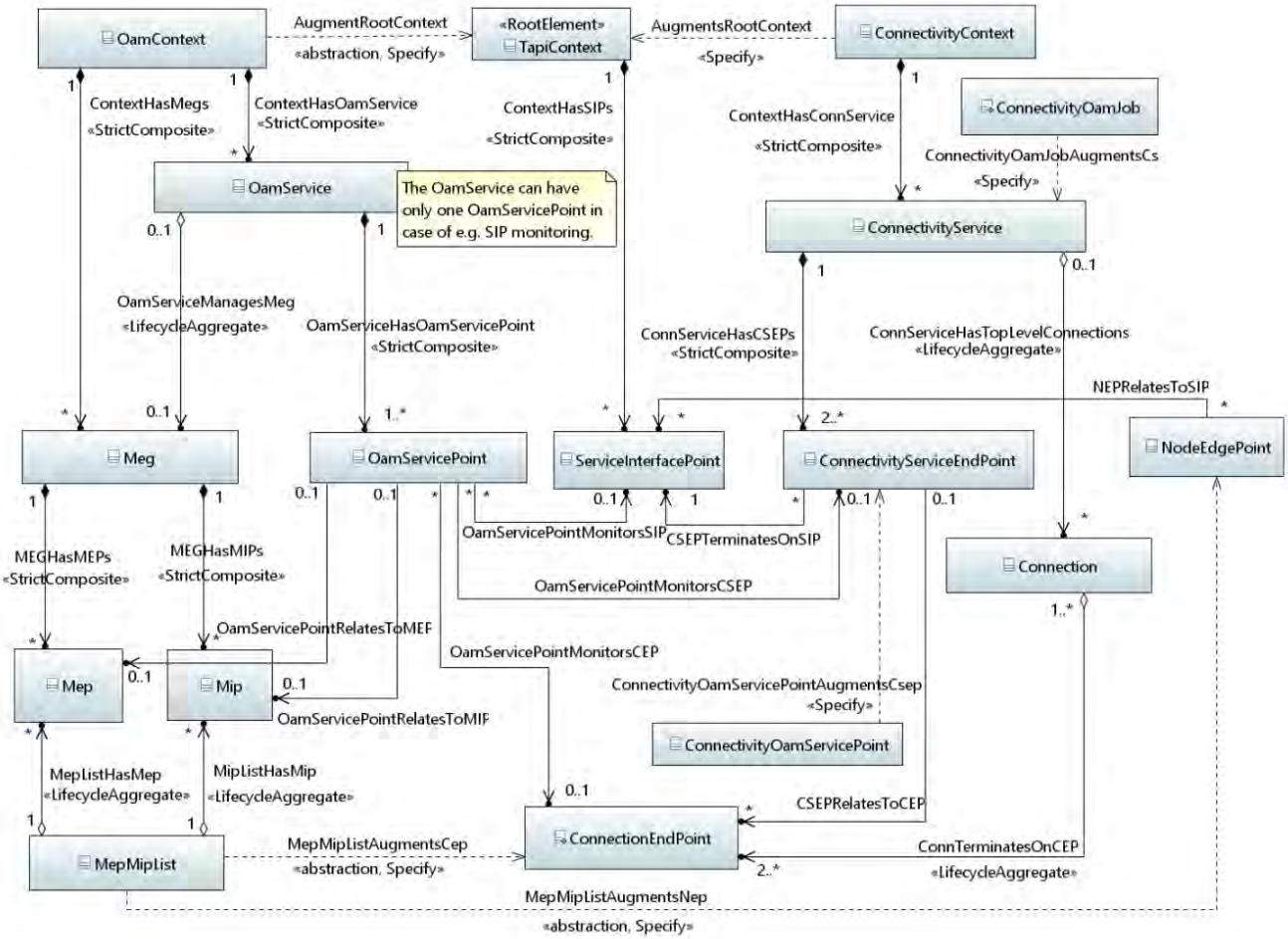

Figure 53 – *OamConnSkeleton*

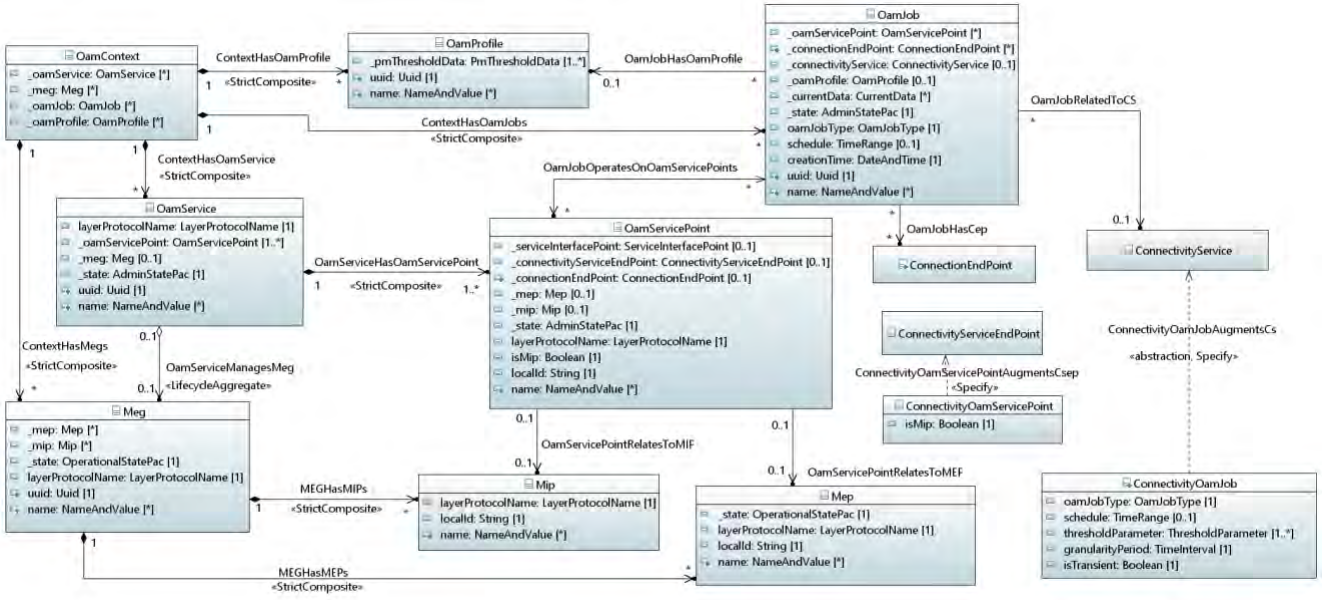

Figure 54 – *OamDetails*

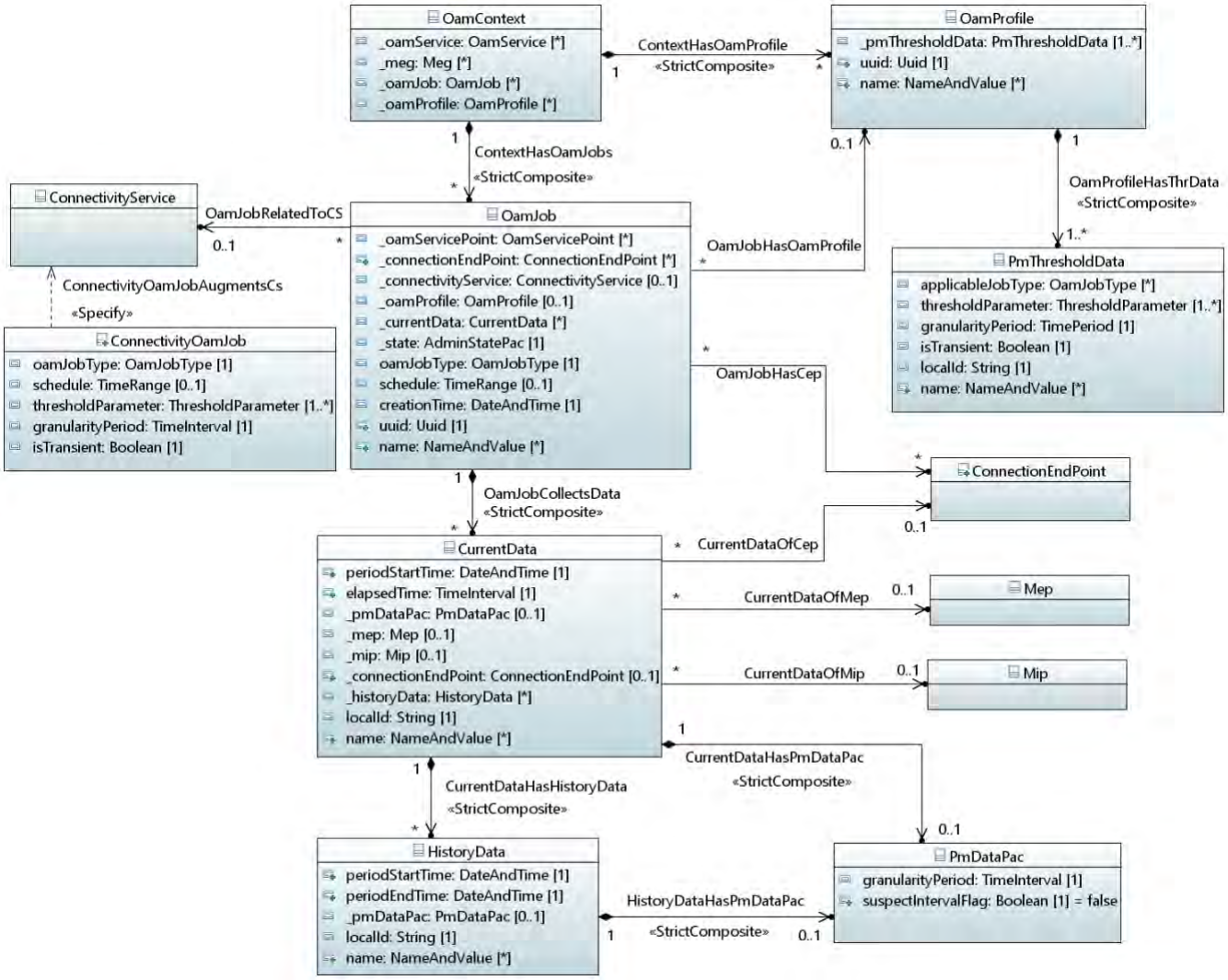

Figure 55 – *OamJobDetails*

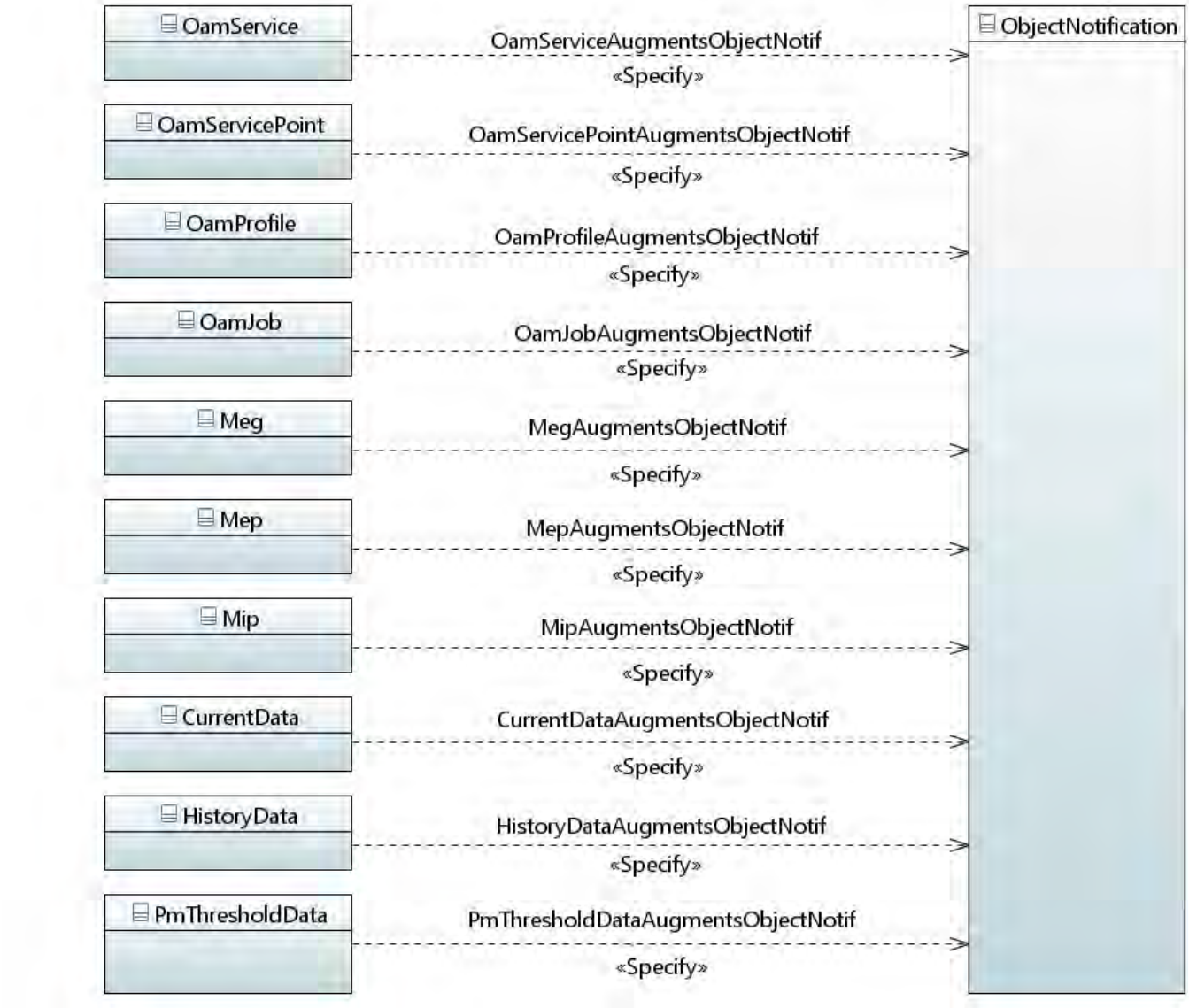

Figure 56 – *OamNotif*

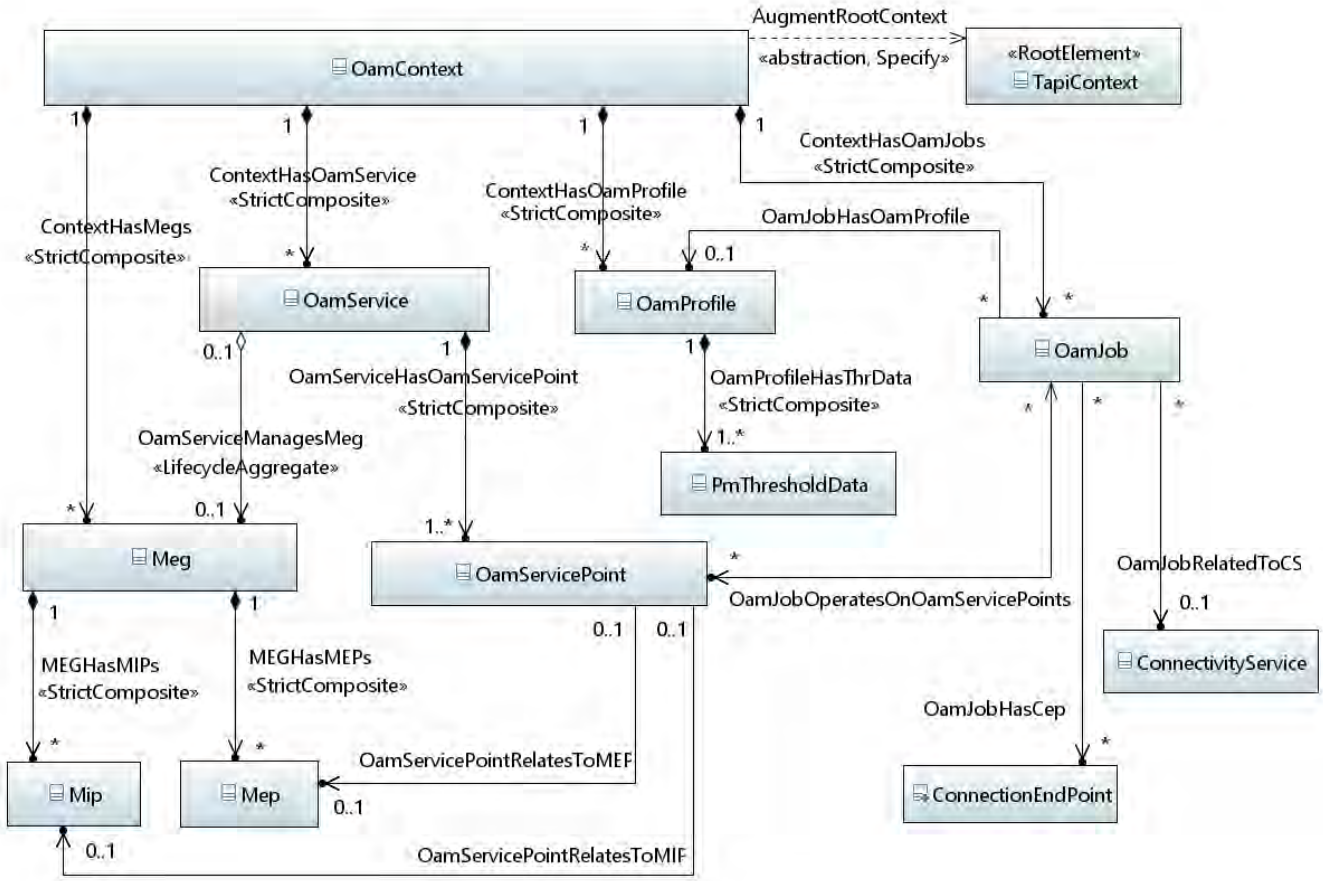

Figure 57 – *OamSkeleton*

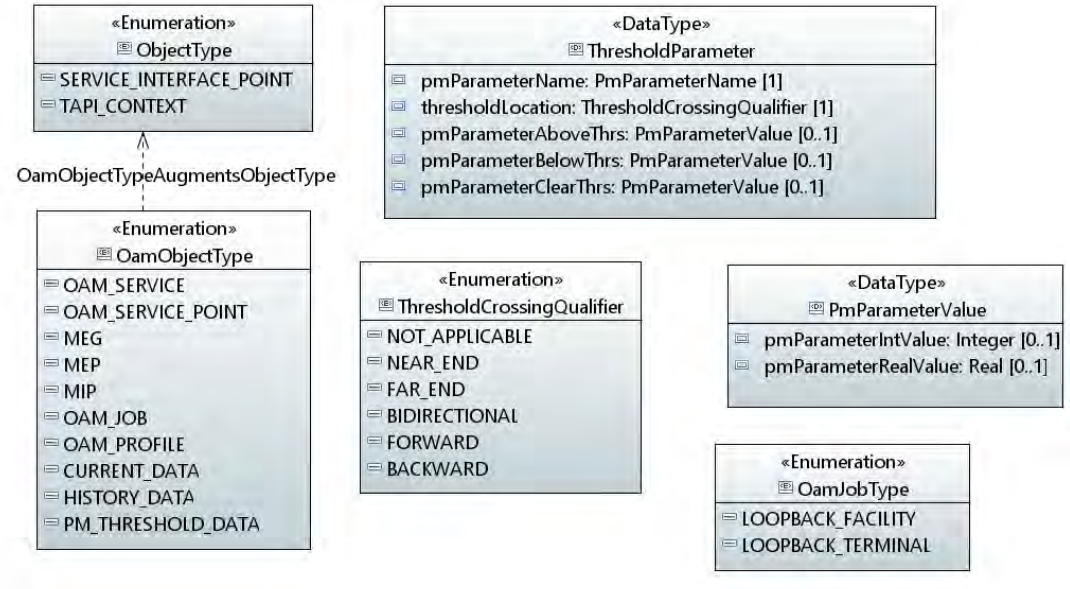

Figure 58 – *OamTypes*

12.2 Classes

12.2.1 ConnectivityOamJob

This class augments the ConnectivityService class to associate OAM job provisioning to ConnectivityService provisioning.

Attribute Name	Attribute Info			
oamJobType	**Type**: OamJobType	**Mult**: 1	RW	**Key**: No
	Description: The type of the OAM job.			
schedule	**Type**: TimeRange	**Mult**: 0..1	RW	**Key**: No
	Description: The schedule of the OAM job.			
thresholdParameter	**Type**: ThresholdParameter	**Mult**: 1..*	RW	**Key**: No
	Description: The PM metrics and their threshold values.			
granularityPeriod	**Type**: TimeInterval	**Mult**: 1	RW	**Key**: No
	Description: The granularity period or measurement interval time of the OAM job.			
isTransient	**Type**: Boolean	**Mult**: 1	RW	**Key**: No
	Description: A threshold crossing alert is transient when stateless, i.e. an explicit clear notification is not foreseen.			

12.2.2 ConnectivityOamServicePoint

This class augments the ConnectivityServiceEndPoint (CSEP) class to associate OAM service provisioning to ConnectivityService provisioning.

Attribute Name	Attribute Info			
isMip	**Type**: Boolean	**Mult**: 1	RW	**Key**: No
	Description: If true, the object is related to a MIP. If false, the object is related to a MEP.			

12.2.3 CurrentData

The CurrentData class. The PM metrics/types can be specified in technology specific augmentations of this class. ITU-T Q.822: This object contains the measurements for the resource being monitored for a specified time interval (measurement interval time / granularity period).

Attribute Name	Attribute Info			
periodStartTime	**Type**: DateAndTime	**Mult**: 1	R	**Key**: No
	Description: This attribute indicates the start time of the current monitoring interval / granularity period. The value is bound to the quarter of an hour in case of a 15 minute interval and bound to the hour in case of a 24 hour interval.			
elapsedTime	**Type**: TimeInterval	**Mult**: 1	RW	**Key**: No
	Description: Q822: This attribute represents the difference between the current time and the start of the present interval.			
_pmDataPac	**Type**: PmDataPac	**Mult**: 0..1	RW	**Key**: No
	Description: Parameters specific to Performance Monitoring functions.			
_mep	**Type**: Mep	**Mult**: 0..1	R	**Key**: No
	Description: The MEP to which the measurements refer to. At least and exclusively one of CurrentDataOfCep, CurrentDataOfMep, CurrentDataOfMip must be referred by the CurrentData instance.			
_mip	**Type**: Mip	**Mult**: 0..1	R	**Key**: No
	Description: The MIP to which the measurements refer to. At least and exclusively one of CurrentDataOfCep, CurrentDataOfMep, CurrentDataOfMip must be referred by the CurrentData instance.			
_connectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 0..1	R	**Key**: No
	Description: The CEP to which the measurements refer to. At least and exclusively one of CurrentDataOfCep, CurrentDataOfMep, CurrentDataOfMip must be referred by the CurrentData instance.			
_historyData	**Type**: HistoryData	**Mult**: 0..*	R	**Key**: No
	Description: The associated HistoryData instances. In case of 24hr CurrentData, at least 1 HistoryData instance shall be maintained. In case of 15min CurrentData, at least 16 HistoryData instances shall be maintained. In case of <15min, the number of HistoryData instances shall be able to cover a span of 4 hours.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

12.2.4 HistoryData

The HistoryData class. The PM metrics/types can be specified in technology specific augmentations of this class. ITU-T Q.822: This object will contain a copy of the performance management and other selected attributes that are present in the CurrentData object at the end of the current interval (measurement interval time / granularity period). A new instance of this object class is created at the end of each interval.

Attribute Name	Attribute Info			
periodStartTime	**Type**: DateAndTime	**Mult**: 1	RW	**Key**: No
	Description: This attribute indicates the start time of the monitoring interval / granularity period. The value is bound to the quarter of an hour in case of a 15 minute interval and bound to the hour in case of a 24 hour interval.			
periodEndTime	**Type**: DateAndTime	**Mult**: 1	RW	**Key**: No
	Description: This attribute indicates the end time of the monitoring interval / granularity period. The value is bound to the quarter of an hour in case of a 15 minute interval and bound to the hour in case of a 24 hour interval.			
_pmDataPac	**Type**: PmDataPac	**Mult**: 0..1	RW	**Key**: No
	Description: Parameters specific to Performance Monitoring functions.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

12.2.5 Meg

The Maintenance Entity Group. ITU-T G.8001: A group defined, for the purpose of fragment or connection monitoring, between a set of flow or connection points within a fragment/connection. This set of flow or connection points may be located at the boundary of one administrative domain or a protection domain, or at the boundaries of two adjacent administrative domains. The maintenance entity group consists of one or more maintenance entities (the entity between two of the flow/connection points in a maintenance entity group).

Attribute Name	Attribute Info			
_mep	**Type**: Mep	**Mult**: 0..*	R	**Key**: No
	Description: The maintenance entity group consists of one or more maintenance entities. There are the following cases: 1. A maintenance entity may have 0 MEPs (case of transit domains where at least 1 MIP is present). 2. A maintenance entity may have 1 MEP (case of edge domains, where the peer MEP is ouside the managed domain). 3. A maintenance entity may have 2 MEPs.			

Attribute Name	Attribute Info			
_mip	**Type**: Mip	**Mult**: 0..*	R	**Key**: No
	Description: The maintenance entity group may have 0, 1, or more MIPs.			
_state	**Type**: OperationalStatePac	**Mult**: 1	RW	**Key**: No
	Description: The Meg status information.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	R	**Key**: No
	Description: The MEG layer protocol.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

12.2.6 Mep

The Maintenance Entity group end Point. ITU-T G.8001: maintenance entity group end point compound sink function: A compound transport processing function that accepts the characteristic information of the layer network at its input, extracts and processes the OAM information related to the monitoring of the maintenance entity group, filters the OAM information from within to the maintenance entity group, adapts the information and presents it as the characteristic information of the layer or a client layer at its output, potentially as a (client) layer maintenance signal (e.g., AIS). ITU-T G.8001: maintenance entity group end point compound source function: A compound transport processing function that accepts the characteristic information of the layer or a client layer network at its input, adapts that information, filters it for OAM information interfering with its own OAM information, adds OAM information to allow the maintenance entity group to be monitored and presents the resulting information at its output.

Attribute Name	Attribute Info			
_state	**Type**: OperationalStatePac	**Mult**: 1	RW	**Key**: No
	Description: The Mep status information.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	R	**Key**: No

Attribute Name	Attribute Info			
	Description: The Mep layer protocol.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

12.2.7 MepMipList

This class provides the linkage between the ConnectionEndPoint (CEP) instance and its associated Mep and Mip instances. The CEP class, which is defined in TapiConnectivity module, cannot directly include the reference to its Mep/Mip, because Mep/Mip classes are defined in another module, TapiOam.

Attribute Name	Attribute Info			
_mep	**Type**: Mep	**Mult**: 0..*	RW	**Key**: No
	Description: The list of associated Mep instances.			
_mip	**Type**: Mip	**Mult**: 0..*	RW	**Key**: No
	Description: The list of associated Mip instances.			

12.2.8 Mip

The Maintenance entity group Intermediate Point. ITU-T G.8001: maintenance entity group intermediate point compound function: A compound transport processing function that accepts the characteristic information of the layer network at its input, reacts to OAM information related to on-demand monitoring of a maintenance entity group and presents the characteristic information without the OAM to which it reacted at its output.

Attribute Name	Attribute Info			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	R	**Key**: No
	Description: The Mip layer protocol.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No

Attribute Name	Attribute Info
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.

12.2.9 OamContext

This object class represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the OAM description. An instance of this class includes its OamService, OamProfile, OamJob and Meg object instances.

Attribute Name	Attribute Info			
_oamService	**Type**: OamService	**Mult**: 0..*	RW	**Key**: No
	Description: The included OamService instances.			
_oamProfile	**Type**: OamProfile	**Mult**: 0..*	RW	**Key**: No
	Description: The included OamProfile instances.			
_oamJob	**Type**: OamJob	**Mult**: 0..*	RW	**Key**: No
	Description: The included OamJob instances.			
_meg	**Type**: Meg	**Mult**: 0..*	R	**Key**: No
	Description: The included Meg instances.			

12.2.10 OamJob

This class allows the provisioning of performance monitoring functions on specified resources.

Attribute Name	Attribute Info			
_oamServicePoint	**Type**: OamServicePoint	**Mult**: 0..*	RW	**Key**: No
	Description: The OamServicePoint (OSP) instances involved in the OamJob.			
_connectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 0..*	RW	**Key**: No
	Description: The ConnectionEndPoint (CEP) instances involved in the OamJob.			
_connectivityService	**Type**: ConnectivityService	**Mult**: 0..1	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: In case the OamJob instance is not related to any OamService/Point but created together with ConnectivityService through ConnectivityOamJob augment.			
_oamProfile	**Type**: OamProfile	**Mult**: 0..1	RW	**Key**: No
	Description: The OamProfile instance referred by the OamJob.			
_currentData	**Type**: CurrentData	**Mult**: 0..*	R	**Key**: No
	Description: The CurrentData instances in the scope of the OamJob.			
_state	**Type**: AdminStatePac	**Mult**: 1	RW	**Key**: No
	Description: The OamJob status information.			
oamJobType	**Type**: OamJobType	**Mult**: 1	RW	**Key**: No
	Description: The OamJob type.			
schedule	**Type**: TimeRange	**Mult**: 0..1	RW	**Key**: No
	Description: The OamJob schedule.			
creationTime	**Type**: DateAndTime	**Mult**: 1	R	**Key**: No
	Description: The OamJob creation time.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

12.2.11 OamProfile

The OamProfile allows centralization of OAM provisioning aspects, e.g. the PM parameters and their threshold values.

Attribute Name	Attribute Info			
_pmThresholdData	**Type**: PmThresholdData	**Mult**: 1..*	RW	**Key**: No
	Description: The PM threshold information associated to the OamProfile.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

12.2.12 OamService

An OamService represents an "intent-like" request for OAM functions between two or more OamServicePoint (OSP) instances. The OamService is a container for OAM request details and is distinct from the Meg that realize the request.

Attribute Name	Attribute Info			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	RW	**Key**: No
	Description: The OamService layer protocol.			
_oamServicePoint	**Type**: OamServicePoint	**Mult**: 1..*	RW	**Key**: No
	Description: The OamServicePoint (OSP) instances of the OamService.			
_meg	**Type**: Meg	**Mult**: 0..1	R	**Key**: No
	Description: The Meg instance tracking the state of the allocated resources for the support of the OamService.			
_state	**Type**: AdminStatePac	**Mult**: 1	RW	**Key**: No
	Description: The OamService status information.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1

Attribute Name	Attribute Info			
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

12.2.13 OamServicePoint

The OamServicePoint (OSP) is a container for OAM request details and is distinct from the Mep and/or Mip instances that realize the request.

Attribute Name	Attribute Info			
_serviceInterfacePoint	**Type**: ServiceInterfacePoint	**Mult**: 0..1	RW	**Key**: No
	Description: The supporting ServiceInterfacePoint (SIP) instance. If neither ConnectivityServiceEndPoint (CSEP) nor ConnectionEndPoint (CEP) are specified, the OamServicePoint (OSP) is intended for SIP monitoring.			
_connectivityServiceEndPoint	**Type**: ConnectivityServiceEndPoint	**Mult**: 0..1	RW	**Key**: No
	Description: The ConnectivityServiceEndPoint (CSEP) instance monitored by the OamServicePoint (OSP). If not specified (and neither CEP is specified), the OamServicePoint (OSP) is intended for SIP monitoring.			
_connectionEndPoint	**Type**: ConnectionEndPoint	**Mult**: 0..1	RW	**Key**: No
	Description: The ConnectionEndPoint (CEP) instance monitored by the OamServicePoint (OSP). If not specified (and neither CSEP is specified), the OamServicePoint (OSP) is intended for SIP monitoring.			
_mep	**Type**: Mep	**Mult**: 0..1	R	**Key**: No
	Description: The associated Mep instance, mutually exclusive wrt Mip instance.			
_mip	**Type**: Mip	**Mult**: 0..1	R	**Key**: No
	Description: The associated Mip instance, mutually exclusive wrt Mep instance.			
_state	**Type**: AdminStatePac	**Mult**: 1	RW	**Key**: No
	Description: The OamServicePoint (OSP) status information.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 1	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: The OamServicePoint (OSP) layer protocol.			
isMip	**Type**: Boolean	**Mult**: 1	RW	**Key**: No
	Description: If true, the object is related to a MIP. If false, the object is related to a MEP.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

12.2.14 PmDataPac

Parameters specific to Performance Monitoring functions.

Attribute Name	Attribute Info			
granularityPeriod	**Type**: TimeInterval	**Mult**: 1	RW	**Key**: No
	Description: The granularity period or measurement interval time.			
suspectIntervalFlag	**Type**: Boolean	**Mult**: 1	RW	**Key**: No
	Description: This attribute is used to indicate that the performance data for the current period may not be reliable. Some reasons for this to occur are: - Suspect data were detected by the actual resource doing data collection. - Transition of the administrativeState attribute to/from the 'lock' state. - Transition of the operationalState to/from the 'disabled' state. - Scheduler setting that inhibits the collection function. - The performance counters were reset during the interval. - The currentData (or subclass) object instance was created during the monitoring period.			

12.2.15 PmThresholdData

The PM threshold information associated to an OamProfile instance. It defines a set of PM metrics, their threshold values, the granularity period or measurement interval time for these PM metrics, the stateful or stateless types of related threshold crossing alert (TCA) reporting.

Attribute Name	Attribute Info			
applicableJobType	**Type**: OamJobType	**Mult**: 0..*	RW	**Key**: No
	Description: This attribute allows an PmThresholdData instance to be constrained to specific job types. If a PmThresholdData instance is so configured to be applicable to more than one job type (worst case ALL), only the parameters relevant for the job instance will be used (non-applicable profile parameters will be ignored).			

Attribute Name	Attribute Info			
thresholdParameter	**Type**: ThresholdParameter	**Mult**: 1..*	RW	**Key**: No
	Description: The PM metrics and their threshold values.			
granularityPeriod	**Type**: TimePeriod	**Mult**: 1	RW	**Key**: No
	Description: The granularity period or measurement interval time.			
isTransient	**Type**: Boolean	**Mult**: 1	RW	**Key**: No
	Description: A threshold crossing alert (TCA) is transient when stateless, i.e. an explicit alarm clear notification is not foreseen. MEF 35.1: Thresholds and associated TCAs are specific to a particular performance metric in a given PM Session (or OAM job). There are two types of TCA reporting: stateless and stateful. With stateless reporting, a TCA is generated in each Measurement Interval in which the threshold is crossed. With stateful reporting, a SET TCA is generated in the first Measurement Interval in which the threshold is crossed, and a CLEAR TCA is subsequently generated at the end of the first Measurement Interval in which the threshold is not crossed. Note: In ITU-T G.7710 terminology, stateless TCA reporting corresponds to a transient condition, and stateful TCA reporting corresponds to a standing condition. Note that threshold management for gauges may be more complex (e.g. out of range function for gauge overflow/underflow detection).			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

12.3 Associations

12.3.1 ContextHasMegs

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_meg	composite	Yes	Meg	0..*
_fc	none	No	OamContext	1

12.3.2 ContextHasOamJobs

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_oamJob	composite	Yes	OamJob	0..*
oamcontext	none	No	OamContext	1

12.3.3 ContextHasOamProfile

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_oamProfile	composite	Yes	OamProfile	0..*
oamcontext	none	No	OamContext	1

12.3.4 ContextHasOamService

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_oamService	composite	Yes	OamService	0..*
oamcontext	none	No	OamContext	1

12.3.5 CurrentDataHasHistoryData

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_historyData	composite	Yes	HistoryData	0..*
_currentData	none	No	CurrentData	1

12.3.6 CurrentDataHasPmDataPac

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_pmDataPac	composite	Yes	PmDataPac	0..1
currentdata	none	No	CurrentData	1

12.3.7 CurrentDataOfCep

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectionEndPoint	none	Yes	ConnectionEndPoint	0..1
currentdata	none	No	CurrentData	0..*

12.3.8 CurrentDataOfMep

Association end role name	Aggregation type	Navigable	Target Class	Mult
_mep	none	Yes	Mep	0..1
currentdata	none	No	CurrentData	0..*

12.3.9 CurrentDataOfMip

Association end role name	Aggregation type	Navigable	Target Class	Mult
_mip	none	Yes	Mip	0..1
currentdata	none	No	CurrentData	0..*

12.3.10 HistoryDataHasPmDataPac

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_pmDataPac	composite	Yes	PmDataPac	0..1
historydata	none	No	HistoryData	1

12.3.11 JobHasAdminStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	AdminStatePac	1
measurementjob	none	No	OamJob	1

12.3.12 MEGHasMEPs

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_mep	composite	Yes	Mep	0..*
_me	none	No	Meg	1

12.3.13 MEGHasMIPs

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_mip	composite	Yes	Mip	0..*
_me	none	No	Meg	1

12.3.14 MEGHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	OperationalStatePac	1
meg	none	No	Meg	1

12.3.15 MEPHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	OperationalStatePac	1
mep	none	No	Mep	1

12.3.16 MepListHasMep

LifecycleAggregate

Association end role name	Aggregation type	Navigable	Target Class	Mult
_mep	shared	Yes	Mep	0..*
oamctppacspec	none	No	MepMipList	1

12.3.17 MipListHasMip

LifecycleAggregate

Association end role name	Aggregation type	Navigable	Target Class	Mult
_mip	shared	Yes	Mip	0..*
oamctppacspec	none	No	MepMipList	1

12.3.18 OSPHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	AdminStatePac	1

Association end role name	Aggregation type	Navigable	Target Class	Mult
oamservicepoint	none	No	OamServicePoint	1

12.3.19 OamJobCollectsData

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_currentData	composite	Yes	CurrentData	0..*
oamjob	none	No	OamJob	1

12.3.20 OamJobHasCep

Direct reference to CEP for simple OAM jobs like loopback.

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectionEndPoint	none	Yes	ConnectionEndPoint	0..*
oamjob	none	No	OamJob	0..*

12.3.21 OamJobHasOamProfile

Association end role name	Aggregation type	Navigable	Target Class	Mult
_oamProfile	none	Yes	OamProfile	0..1
oamjob	none	No	OamJob	0..*

12.3.22 OamJobOperatesOnOamServicePoints

Association end role name	Aggregation type	Navigable	Target Class	Mult
_oamServicePoint	none	Yes	OamServicePoint	0..*
_oamJob	none	Yes	OamJob	0..*

12.3.23 OamJobRelatedToCS

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectivityService	none	Yes	ConnectivityService	0..1
oamjob	none	No	OamJob	0..*

12.3.24 OamProfileHasThrData

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_pmThresholdData	composite	Yes	PmThresholdData	1..*
pmthresholdprofile	none	No	OamProfile	1

12.3.25 OamServiceHasOamServicePoint

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_oamServicePoint	composite	Yes	OamServicePoint	1..*
oamservice	none	No	OamService	1

12.3.26 OamServiceHasStatePac

ExtendedComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_state	composite	Yes	AdminStatePac	1
oamservice	none	No	OamService	1

12.3.27 OamServiceManagesMeg

LifecycleAggregate

Association end role name	Aggregation type	Navigable	Target Class	Mult
_meg	shared	Yes	Meg	0..1
fc	none	No	OamService	0..1

12.3.28 OamServicePointMonitorsCEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectionEndPoint	none	Yes	ConnectionEndPoint	0..1
oamservicepoint	none	No	OamServicePoint	0..*

12.3.29 OamServicePointMonitorsCSEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_connectivityServiceEndPoint	none	Yes	ConnectivityServiceEndPoint	0..1
oamserviceendpoint	none	No	OamServicePoint	0..*

12.3.30 OamServicePointMonitorsSIP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_serviceInterfacePoint	none	Yes	ServiceInterfacePoint	0..1
oamserviceendpoint	none	No	OamServicePoint	0..*

12.3.31 OamServicePointRelatesToMEP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_mep	none	Yes	Mep	0..1
_oamServiceEndPoint	none	No	OamServicePoint	0..1

12.3.32 OamServicePointRelatesToMIP

Association end role name	Aggregation type	Navigable	Target Class	Mult
_mip	none	Yes	Mip	0..1
_oamServiceEndPoint	none	No	OamServicePoint	0..1

12.4 Abstractions

12.4.1 AugmentRootContext

Augments the base TAPI Context with OamService model.

Target Class: "/TapiCommon:Context:_context"

12.4.2 ConnectivityOamJobAugmentsCs

Target Class:
"/TapiCommon:Context:_context/TapiConnectivity:ConnectivityContext:_connectivityContext/TapiConnectivity:ConnectivityContext:_connectivityService"

12.4.3 ConnectivityOamServicePointAugmentsCsep

Target Class:
"/TapiCommon:Context:_context/TapiConnectivity:ConnectivityContext:_connectivityContext/TapiConnectivity:ConnectivityContext:_connectivityService/TapiConnectivity:ConnectivityService:_endPoint"

12.4.4 CurrentDataAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.4.5 HistoryDataAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.4.6 MegAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.4.7 MepAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.4.8 MepMipListAugmentsCep

This augment allows CEP to refer to its MEPs/MIPs despite TapiOam model does not import TapiConnectivity model.

Target Class:
"/TapiCommon:Context:_context/TapiTopology:TopologyContext:_topologyContext/TapiTopology:TopologyContext:_topology/TapiTopology:Topology:_node/TapiTopology:Node:_ownedNodeEdgePoint/TapiConnectivity:CepList:_cepList/TapiConnectivity:Connection:_connectionEndPoint"

12.4.9 MepMipListAugmentsNep

This augment allows NEP to refer to its MEPs/MIPs despite TapiOam model does not import TapiTopology model.

Target Class:
"/TapiCommon:Context:_context/TapiTopology:TopologyContext:_topologyContext/TapiTopology:TopologyContext:_topology/TapiTopology:Topology:_node/TapiTopology:Node:_ownedNodeEdgePoint"

12.4.10 MipAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.4.11 OamJobAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.4.12 OamObjectTypeAugmentsObjectType

Enumeration Augment.

12.4.13 OamProfileAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.4.14 OamServiceAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.4.15 OamServicePointAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.4.16 PmThresholdDataAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

12.5 Data Types

12.5.1 ThresholdParameter

PM metrics, their locations and threshold values.

Attribute Name	Attribute Info			
pmParameterName	**Type**: PmParameterName	**Mult**: 1	RW	**Key**: yes – part: 1

Attribute Name	Attribute Info			
	Description: PM metric name.			
thresholdLocation	**Type**: ThresholdCrossingQualifier	**Mult**: 1	RW	**Key**: yes – part: 2
	Description: PM metric location.			
pmParameterAboveThrs	**Type**: PmParameterValue	**Mult**: 0..1	RW	**Key**: No
	Description: PM metric above threshold.			
pmParameterBelowThrs	**Type**: PmParameterValue	**Mult**: 0..1	RW	**Key**: No
	Description: PM metric below threshold.			
pmParameterClearThrs	**Type**: PmParameterValue	**Mult**: 0..1	RW	**Key**: No
	Description: PM metric clear threshold.			

12.6 Enumerations

12.6.1 OamJobType

The OAM job types. This extensible enumeration can be augmented with specific OAM job types in the other modules.

Contains Enumeration Literals:

- LOOPBACK_FACILITY:
- LOOPBACK_TERMINAL:

12.6.2 OamObjectType

The list of TAPI OAM Global Object Class types on which Notification signals can be raised.

Contains Enumeration Literals:

- OAM_SERVICE:
  - The OamService class.
- OAM_SERVICE_POINT:
  - The OamServicePoint (OSP) class.
- MEG:
  - The Meg class.
- MEP:
  - The Mep class.

- MIP:
  - The Mip class.
- OAM_JOB:
  - The OamJob class.
- OAM_PROFILE:
  - The OamProfile class.
- CURRENT_DATA:
  - The CurrentData class.
- HISTORY_DATA:
  - The HistoryData class.
- PM_THRESHOLD_DATA:
  - The PmThresholdData class.

12.6.3 ThresholdCrossingQualifier

Threshold crossing location or qualifier.

Contains Enumeration Literals:

- NOT_APPLICABLE:
  - Location or qualifier not applicable.
- NEAR_END:
  - Near End detection.
- FAR_END:
  - Far end detection.
- BIDIRECTIONAL:
  - Composition of near and far end detections.
- FORWARD:
  - MEF 35.1: The direction of performance measurements from the Controller MEP towards the Responder or Sink MEP, when One-way measurements are taken using a Single-Ended or Dual-Ended PM Function. MEF 83: In Single-Ended measurements, it is assumed that the the FORWARD and FAR_END qualifiers are equivalent. In Dual-Ended measurements (and in case of TX counters), it is assumed that the FORWARD and NEAR_END qualifiers are equivalent.
- BACKWARD:
  - MEF 35.1: The direction of performance measurements from the Responder MEP towards the Controller MEP, when One-way measurements are taken using a Single-Ended PM Function. Note: this term is not applicable when Dual-Ended PM Functions are used. MEF 83: In Single-Ended measurements, it is assumed that the BACKWARD and NEAR_END qualifiers are equivalent. In Dual-Ended measurements (and in case of TX counters), it is assumed that the BACKWARD and FAR_END qualifiers are equivalent.

13 Notification Model

13.1 Diagrams

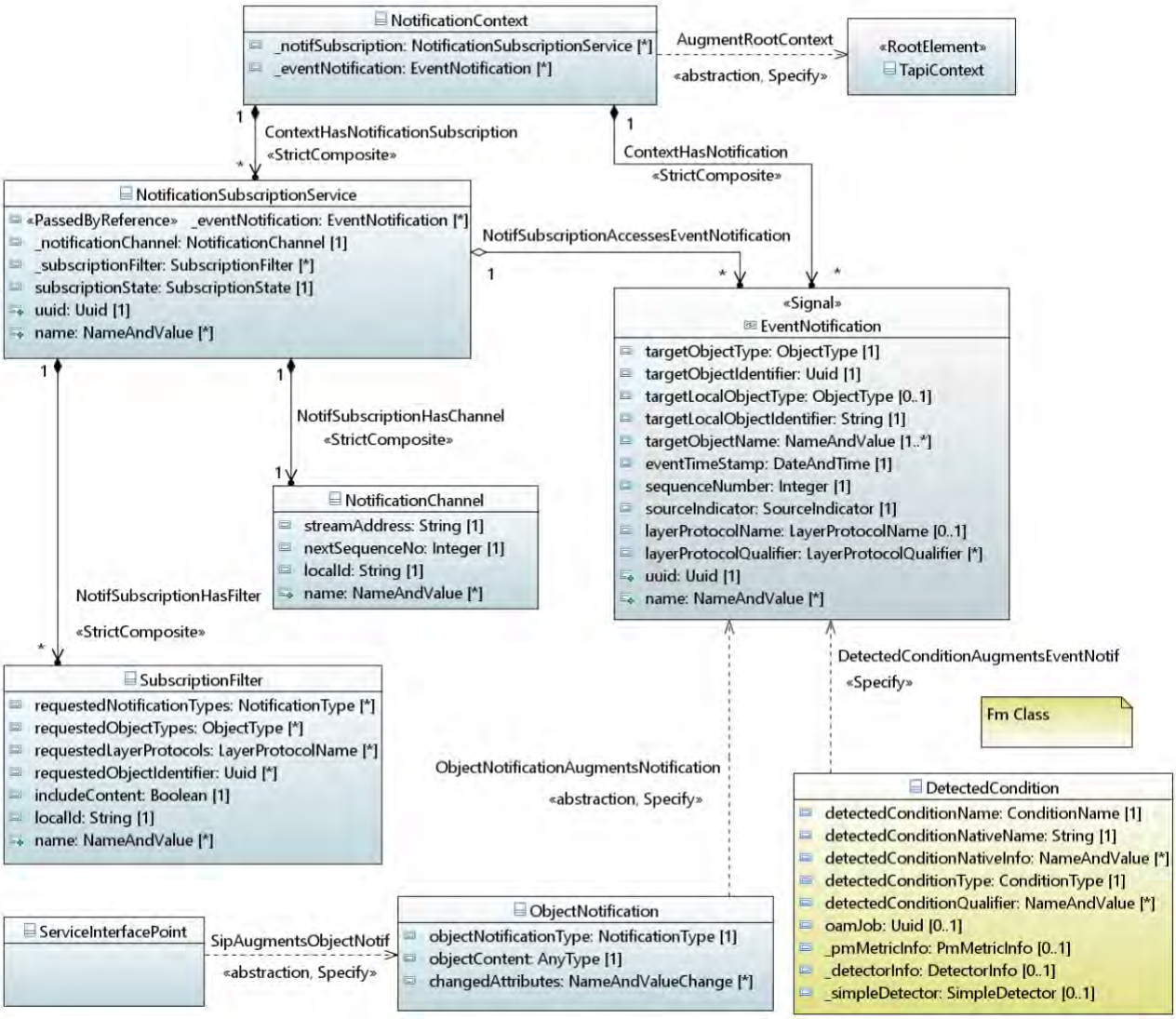

Figure 59 – *NotificationServiceDetails*

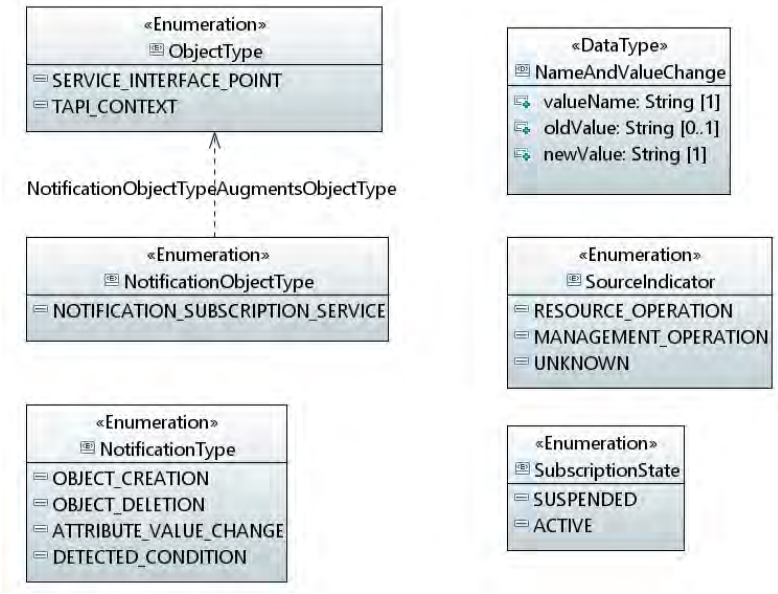

Figure 60 – *NotificationTypes*

13.2 Classes

13.2.1 NotificationChannel

The channel/stream to which the subscribed notifications are published.

Attribute Name	Attribute Info			
streamAddress	**Type**: String	**Mult**: 1	R	**Key**: No
	Description: The address/location/URI of the channel/stream to which the subscribed notifications are published. The format is typically dependent on the implementation protocol & mechanism and hence is typed as a string.			
nextSequenceNo	**Type**: Integer	**Mult**: 1	R	**Key**: No
	Description: The sequence number of the next notification that will be published on the channel.			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

13.2.2 NotificationContext

This object class represents the scope of control that a particular SDN controller has with respect to a particular network, specifically regarding the notification description. An instance of this class includes its NotificationSubscriptionService and Notification instances.

Attribute Name	Attribute Info			
_notifSubscription	**Type**: NotificationSubscriptionService	**Mult**: 0..*	RW	**Key**: No **Condition**: notification-management
	Description: The included NotificationSubscriptionService instances.			
_eventNotification	**Type**: EventNotification	**Mult**: 0..*	R	**Key**: No **Condition**: notification-management
	Description: The included Event Notification instances.			

13.2.3 NotificationSubscriptionService

A NotificationSubscriptionService represents an "intent-like" request for the notification subscription. The NotificationSubscriptionService is a container for subscription request details.

Attribute Name	Attribute Info			
_eventNotification	**Type**: EventNotification	**Mult**: 0..*	RW	**Key**: No
	Description: The EventNotification instances associated to this NotificationSubscriptionService instance.			
_notificationChannel	**Type**: NotificationChannel	**Mult**: 1	R	**Key**: No
	Description: The NotificationChannel instance of this NotificationSubscriptionService instance.			
_subscriptionFilter	**Type**: SubscriptionFilter	**Mult**: 0..*	RW	**Key**: No
	Description: The SubscriptionFilter instance of this NotificationSubscriptionService instance.			
subscriptionState	**Type**: SubscriptionState	**Mult**: 1	RW	**Key**: No
	Description: The SubscriptionState value.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			

Attribute Name	Attribute Info			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

13.2.4 ObjectNotification

Object notification related information.

Attribute Name	Attribute Info			
objectNotificationType	**Type**: NotificationType	**Mult**: 1	R	**Key**: No
	Description: The object notification type.			
objectContent	**Type**: AnyType	**Mult**: 1	R	**Key**: No
	Description: The object content, e.g. all the attributes of a newly created object. The mapping is not specified.			
changedAttributes	**Type**: NameAndValueChange	**Mult**: 0..*	R	**Key**: No
	Description: The list of relevant changed attributes and their values.			

13.2.5 SubscriptionFilter

A SubscriptionFilter represents an "intent-like" request for the filters of the related notification subscription. The SubscriptionFilter is a container for filter request details.

Attribute Name	Attribute Info			
requestedNotificationTypes	**Type**: NotificationType	**Mult**: 0..*	RW	**Key**: No
	Description: The requested NotificationType value(s).			
requestedObjectTypes	**Type**: ObjectType	**Mult**: 0..*	RW	**Key**: No
	Description: The requested ObjectType value(s).			
requestedLayerProtocols	**Type**: LayerProtocolName	**Mult**: 0..*	RW	**Key**: No
	Description: The requested layer protocol value(s).			
requestedObjectIdentifier	**Type**: Uuid	**Mult**: 0..*	RW	**Key**: No
	Description: The requested object identifier (UUID) value(s).			

Attribute Name	Attribute Info			
includeContent	**Type**: Boolean	**Mult**: 1	RW	**Key**: No
	Description: Indicates whether the published Notification includes content or just the Notification Id (which enables retrieval of the notification at the later stage).			
localId	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: An identifier that is unique in the context of the GlobalClass from which it is inseparable.			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

13.3 Signals

13.3.1 EventNotification

The Notification signal. OMG UML(R) Version 2.5.1: "A Signal is a specification of a kind of communication between objects in which a reaction is asynchronously triggered in the receiver without a reply. The data carried by the communication are represented as attributes of the Signal."

Attribute Name	Attribute Info			
targetObjectType	**Type**: ObjectType	**Mult**: 1	RW	**Key**: No
	Description: The Notification instance is related to the object instance (of a global class - with UUID) with this ObjectType value. Alternatively, the Notification is related to the object instance of a local class, whose global object has this ObjectType value.			
targetObjectIdentifier	**Type**: Uuid	**Mult**: 1	RW	**Key**: No
	Description: The Notification instance is related to the object instance (of a global class) with this UUID value. Alternatively, the Notification is related to the object instance of a local class, whose global object has this UUID value.			
targetLocalObjectType	**Type**: ObjectType	**Mult**: 0..1	RW	**Key**: No
	Description: The Notification instance is related to the object instance of a local class, whose global object has targetObjectType value.			
targetLocalObjectIdentifier	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: The Notification instance is related to the object instance of a local class, whose global object has targetObjectIdentifier value.			
targetObjectName	**Type**: NameAndValue	**Mult**: 1..*	RW	**Key**: No

Attribute Name	Attribute Info			
	Description: The Notification instance is related to the object instance with this list of names.			
eventTimeStamp	**Type**: DateAndTime	**Mult**: 1	RW	**Key**: No
	Description: The best knowledge of the time of the event which originated this Notification instance.			
sequenceNumber	**Type**: Integer	**Mult**: 1	RW	**Key**: No
	Description: A monotonous increasing sequence number associated with the Notification instances. The exact semantics of how this sequence number is assigned (per channel or subscription or source or system) is left undefined.			
sourceIndicator	**Type**: SourceIndicator	**Mult**: 1	RW	**Key**: No
	Description: The possible source of this Notification instance.			
layerProtocolName	**Type**: LayerProtocolName	**Mult**: 0..1	RW	**Key**: No
	Description: The Notification instance is related to a resource with this layer protocol value.			
layerProtocolQualifier	**Type**: LayerProtocolQualifier	**Mult**: 0..*	RW	**Key**: No
	Description: The Notification instance is related to a resource with this layer protocol qualifier value.			
uuid	**Type**: Uuid	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: UUID: An identifier that is universally unique within an identifier space, where the identifier space is itself globally unique, and immutable. An UUID carries no semantics with respect to the purpose or state of the entity. UUID here uses string representation as defined in RFC 4122. The canonical representation uses lowercase characters. Pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-' + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12} Example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6			
name	**Type**: NameAndValue	**Mult**: 0..*	RW	**Key**: No
	Description: List of names. This value is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.			

13.4 Associations

13.4.1 ContextHasNotification

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_eventNotification	composite	Yes	EventNotification	0..*
notificationcontext	none	No	NotificationContext	1

13.4.2 ContextHasNotificationSubscription

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_notifSubscription	composite	Yes	NotificationSubscriptionService	0..*
notificationcontext	none	No	NotificationContext	1

13.4.3 NotifSubscriptionAccessesEventNotification

Association end role name	Aggregation type	Navigable	Target Class	Mult
_eventNotification	shared	Yes	EventNotification	0..*
notificationsubscriptionservice	none	No	NotificationSubscriptionService	1

13.4.4 NotifSubscriptionHasChannel

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_notificationChannel	composite	Yes	NotificationChannel	1
_notifSubscription	none	No	NotificationSubscriptionService	1

13.4.5 NotifSubscriptionHasFilter

StrictComposite

Association end role name	Aggregation type	Navigable	Target Class	Mult
_subscriptionFilter	composite	Yes	SubscriptionFilter	0..*
_notifSubscription	none	No	NotificationSubscriptionService	1

13.5 Abstractions

13.5.1 AugmentRootContext

Augments the base TAPI Context with NotificationService model.

Target Class: "/TapiCommon:Context:_context"

13.5.2 NotificationObjectTypeAugmentsObjectType

Enumeration Augment.

13.5.3 ObjectNotificationAugmentsNotification

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification"

13.5.4 SipAugmentsObjectNotif

Target Class:
"/TapiCommon:Context:_context/TapiNotification:NotificationContext:_notificationContext/TapiNotification:NotificationContext:_eventNotification/TapiNotification:EventNotification:_objectNotification"

13.6 Data Types

13.6.1 NameAndValueChange

A scoped name-value triple, including old value and new value.

Attribute Name	Attribute Info			
valueName	**Type**: String	**Mult**: 1	RW	**Key**: yes – part: 1
	Description: The name of the value. The value need not have a name.			
oldValue	**Type**: String	**Mult**: 0..1	RW	**Key**: No
	Description: The old value.			
newValue	**Type**: String	**Mult**: 1	RW	**Key**: No
	Description: The new value.			

13.7 Enumerations

13.7.1 NotificationObjectType

The list of TAPI Notification Global Object Class types on which Notification signals can be raised.

Contains Enumeration Literals:

- NOTIFICATION_SUBSCRIPTION_SERVICE:
  - The NotificationSubscriptionService class.

13.7.2 NotificationType

List of supported notification types.

Contains Enumeration Literals:

- OBJECT_CREATION:
  - The notification of an object instance creation event.
- OBJECT_DELETION:
  - The notification of an object instance deletion event.
- ATTRIBUTE_VALUE_CHANGE:
  - The notification of an attribute value change event.
- DETECTED_CONDITION:
  - The notification of a detected condition event, for example, an alarm or a threshold crossing alert detected on a given resource.

13.7.3 SourceIndicator

The possible source of the notification.

Contains Enumeration Literals:

- RESOURCE_OPERATION:
  - The notification has been raised as a consequence of a generic state change of resource(s) in the managed network.
- MANAGEMENT_OPERATION:
  - The notification has been raised as a consequence of a management operation.
- UNKNOWN:
  - Unknown source of the notification.

13.7.4 SubscriptionState

The SubscriptionState types.

Contains Enumeration Literals:

- SUSPENDED:
  - The subscription is suspended.
- ACTIVE:
  - The subscription is active.

14 References

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

[2] IETF RFC 8040, *RESTCONF Protocol*, January 2017

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

[4] ITU-T Recommendation G.709/Y.1331, *Interfaces for the optical transport network,* June 2020

[5] ITU-T G.7711/Y.1702 *Generic protocol-neutral information model for transport resources*, March 2018

[6] ITU-T G.8001/Y.1354 *Packet over Transport aspects – Ethernet over Transport aspects*, April 2016

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

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

[9] MEF 30.1, *Service OAM Fault Management Implementation Agreement: Phase 2,* April 2013

[10] MEF 35.1, *Service OAM Performance Monitoring Implementation Agreement,* May 2015

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

[12] MEF 59, *Network Resource Management Information Model: Connectivity*, January 2018

[13] MEF 63, *Subscriber Layer 1 Service Attributes,* August 2018

[14] MEF 64, *Operator Layer 1 Service Attributes and Services*, February 2020

[15] MEF 72.1, *Resource Model - Subscriber & Operator Layer 1*, October 2020

[16] MEF 83, *Network Resource Model – OAM*, September 2019

[17] MEF Resource Model GitHub repository: https://github.com/MEF-GIT/MEF-LSO-Presto-SDK-extended

[18] ONF TR-512 *Core Information Model, Version 1.4*, November 2018

[19] ONF TR-527 *Functional Requirements for Transport API*, June 10, 2016

[20] ONF TR-547 *TAPI v2.1.3 Reference Implementation Agreement*, Version 1.0, July 2020.

[21] ONF Transport API (TAPI) Information Model, v2.3.1 - https://github.com/OpenNetworkingFoundation/TAPI/releases/tag/v2.3.1, October 2021

[22] Papyrus UML Tool - https://www.eclipse.org/papyrus/documentation.html Copyright © 2019 The Eclipse Foundation. All Rights Reserved.

[23] TM Forum, Information Framework (SID), GB922, Release 17.0.0, June 2017.

[24] TM Forum MTNM 4.5, July 2015

[25] TM Forum MTOSI 4.0, July 2015