METRO-HAUL

METRO High bandwidth, 5G Application-aware optical network, with edge storage, compute and low latency

Grant No. 761727

Deliverable D4.1

METRO-HAUL Control and Management Requirements and Framework.

Editor:	Ramon Casellas, CTTC
Deliverable nature:	Report
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	2018-05-31
Actual delivery date:	2018-06-07
Suggested readers:	Optical Networks Researchers and Engineers; Product Managers; Architects and Technology specialists.
Version:	1.0
Total number of pages:	107
Keywords:	SDN/NFV; Network Control Plane; Multilayer and Optical

	Transport Services; Management and Orchestration (MANO); Disaggregated Optical Networks; Monitoring and Data Analytics; Dynamic Network Planning

Abstract

This document (D4.1) reports the initial design of the METRO-HAUL control platform. It provides an overview of the control and management requirements and framework justifying design choices and deployment models. D4.1 covers a preliminary specification of the interfaces in view of integration and interoperability, listing selected components to be developed and to be integrated in demonstration scenarios. For each component, D4.1 details its implementation roadmap, functional regression and integration tests, and the means to validate the implementation, including control KPIs.

The METRO-HAUL control, orchestration and management system that enables the dynamic provisioning of services, such as network data connectivity or network slicing instances, is presented from a high-level point of view. The COM is based on Software Defined Networking principles, with a centralized network control plane. This control plane is hierarchical, with an SDN control per technology domain. Along the SDN control plane for the provisioning of connectivity, a European Telecommunications Standards Institute Network Function Virtualization Management and Orchestration MANO part is responsible for the instantiation of NFV-based Network Services, understood in this context as interconnected Virtual Network Functions. As key components of the system and as examples of the innovation behind METRO-HAUL, there is Monitoring and Data Analytics (MDA) subsystem, able to recommend actions to the controllers based on performance monitoring and telemetry data and a service and network planner component, which enables the COM system to off-load computations such as VNF placement or network reconfiguration to task optimized entities.

The research leading to these results has received funding from the European Union Horizon 2020 Programme under grant agreement number METRO-HAUL 761727

Impressum

[Full project title] METRO High bandwidth, 5G Application-aware optical network, with edge storage, compute and low latency

[Short project title] METRO-HAUL

[Number and title of work-package] WP4

[Number and title of task] T4.1, T4.2, T4.3

[Document title] METRO-HAUL Control and Management Requirements and Framework.

[Editor: Name, company] Ramon CASELLAS, CTTC

[Work-package leader: Name, company] Ramon CASELLAS, CTTC

Executive Summary

The overall aims of the METRO-HAUL project are to architect and design cost-effective, energy-efficient, agile and programmable metro networks that are scalable for 5G access and future requirements. The project also encompasses the design of all-optical metro nodes (including full compute and storage capabilities), which interface effectively with both 5G access and multi-Tbit/s elastic core networks. *Furthermore, and consideration of the 5GPP control requirements associated with 5G networks, service provisioning and traffic dynamicity management - a key requirement of the metro network – has entailed the development of a METRO-HAUL Control, Orchestration and Management (COM) system that allows the dynamic provisioning of services and efficient resource optimisation.*

We present the COM requirements and architecture designed to allow the automatic deployment of 5G services (including support for European Telecommunications Standards Institute (ETSI) NFV (Network Function Virtualization) based network services). These services are envisaged to cross metropolitan networks, and conceived to interface 5G access networks with disaggregated elastic core optical networks at multi Tb/s. This network segment, referred to as "**METRO-HAUL"**, is composed of various resources, including: infrastructure nodes with networking capabilities, augmented with storage and processing resources, which are in turn interconnected by open and disaggregated optical networks. We also provide further detail on ancillary subsystems like the Monitoring and Data Analytics (MDA) or the in-operation planning backend that extend current SDN based network control to accommodate existing and emerging use cases. At the heart of METRO-HAUL, we apply the philosophy of a disaggregated optical network.

Typically, Software Defined Network (SDN) solutions for transport networks are often associated to single-vendor optical domains, managed as single entities, and commonly referred to as fully aggregated. Controllers do export and expose interfaces for the limited control of abstracted resources and operations via North-Bound Interface (NBI), but such Application Programme Interfaces (APIs) are often vendor specific and internal control aspects related to provisioning, monitoring and resource management remain proprietary and not open for further development unless strictly controlled by the vendor. Disaggregation of optical networks refers to a deployment model of optical systems, by composing and assembling open APIs, general hardware components, open devices and sub-systems. This disaggregation is driven by multiple factor (the mismatch between the needs of operators and the ability to deliver adapted solutions by vendors or the increase in hardware commoditization) and disaggregated networks are an excellent use case for open and standards-based (formal and informal) interfaces, showing the benefits of a unified, model-driven development. In addition to this, recent advances related to the concepts of Artificial Intelligence (AI) and Machine Learning (ML) and with applications across multiple technology domains, have gathered significant attention due to the overall performance improvement of such automated systems when compared to methods relying on human operation. Using AI/ML for managing, operating and optimizing transport networks is increasingly seen as a potential opportunity for automation and accelerated deployment times, especially applicable to large and complex environments. *Such AI-assisted automated network operation is expected to facilitate innovation in multiple aspects a promising milestone in the evolution towards autonomous networks, where networks self-adjust parameters such as transceiver configuration*. To accomplish

this goal, current network control, management and orchestration systems need to facilitate the integration and application of AI/ML techniques. It is arguable that SDN principles, favouring centralized control deployments, featured application programming interfaces and the development of a related application ecosystem are well positioned to facilitate the progressive introduction of such techniques, therefore allowing efficient massive monitoring and data collection, and utilising this vast knowledge.

The underlying infrastructure (spanning multiple geographic locations) relies on macroscopic nodes, combining networking, processing and storage resources. Such modular devices are composed of different components operating at different layers and technologies, and of different vendors, realizing hardware and software disaggregation inside the node. It implements layer 0-1 (optical domain), layer 2 transmission and switching and computing capabilities, provided by local pool of servers to instantiate Virtual Network Functions (VNFs) with configurable amount of processing, memory and storage. Concrete specializations of the generic architecture are at the METRO-HAUL *Access Metro Edge nodes (AMEN nodes*) to interface with heterogeneous access technologies and at the *Metro Core Edge nodes (MCEN nodes*).

From a control and management point of view, we have identified **a set of core (basic) services** to be developed, being the main targets: 1) *Network connectivity services* include connectivity provisioning between MAC endpoints, IP addresses or optical layer ports; 2) *Virtual Services*, where METRO-HAUL include cloud resources to support the instantiation of Virtual Machines (VMs) and VNFs, being able to configure VMs and attach virtual NICs to soft switches; 3) *ETSI / Network Function Virtualization (NFV) Network Services and slicing*, a set of interconnected VNFs across the METRO-HAUL infrastructure and 4) *Monitoring Services*, since cognitive network architectures have been proven to self-adapt the network in a cost-effective manner (autonomic networking).

Specifically, by applying data analytics to monitored data, control loops can be enabled, where analysis outcomes can be used to recommend network reconfiguration actions to the SDN controller. This is especially useful for the **following use cases**: i) *Open Line Systems (OLS) monitoring,* where an OLS considers the disaggregation of a network into functional blocks (such as transponder, ROADMs and/or degrees in a switching fabric). Such disaggregation benefits from monitoring e.g. for alien wavelengths, and to optimize the power levels. Similarly, for coherent transmission, additional parameters (dispersion, differential group delay, line errors (including BER)) can be monitored and estimated, enabling closed loop control and optimization of bandwidth-variable transponders; ii) *Transport Network Re-optimization*, where the collection of statistics and historical data regarding service requests allows to predict demands (e.g. using ML) and to re-optimize periodically ensuring an efficient network-resource optimization, towards an autonomic infrastructure that proactively self-configures and self-tunes. Traffic matrices may be estimated periodically with dedicated passive and active probes, and algorithms may re-arrange connections. Finally, iii) *VNF placement across transport networks*, since 5G services requiring the deployment of virtualized network functions (VNFs) may be constrained in terms of latency, jitter or bandwidth, especially if deployed in a metropolitan or WAN network. The joint network orchestration may rely on an external planner, optimizing provisioning while implementing medium and long-term deployment and operation goals.

We adopt SDN/NFV principles and frameworks, extended with additional dedicated systems, listed below:

Network Control and Orchestration relies on an over-arching control adopting hierarchical control architectures with a parent SDN controller abstracting the underlying complexity. We adopt open interfaces exporting programmability along with unified and systematic information and data modelling. Finally, the coordinated control of the packet layer and the optical layer is needed to provision end-to-end services and steering traffic (based on a number of optimisation objectives) coming from aggregation networks.

Compute and Storage Integration via SDN/NFV: Joint IT/Cloud and Network Orchestration is used to refer to the coordination of resources to deploy services and applications that require storage, computing and networking resources. The approach involves integrating the NFV Management and Orchestration (MANO) functional elements over multiple Virtual Infrastructure Managers (VIMs), with hierarchical control planes which are abstracted as WAN infrastructure managers (WIM)

ETSI MANO / Slicing Integration: the ETSI NFV framework can be used as a starting point for generic slicing architecture, in which network slice instances are NFV Network Services (NS), encompassing NS endpoints and one or more VNFs interconnected by logical links, forming VNF forwarding graphs (VNFFGs).

Monitoring and Data Analytics: Autonomic networking entails the capability to do measurements on the data plane and generating data records that are collected and analysed to discover patterns (knowledge) from data. Such knowledge can be used to issue re-configuration/re-optimization recommendations toward COM modules, such as an SDN controller or orchestrator.

Network Planning: The aim is to optimize the resource allocation. The traffic generated by end-user oriented services have different requirements in terms of bandwidth, delay and QoS. For this reason, the intelligent optimization algorithms of the optimization module will apply different policies and strategies and will need data coming from the data plane.

The core of SDN control work is the optical transport network. The approach is based on having an SDN controller controlling one or more devices, which are characterized by their data model, which determines the structure, syntax and semantics of the data that is externally visible. We are considering mainly two sets of data models for abstracting optical hardware devices: OpenConfig, for terminal optical devices (i.e., transponders) and an Open Line System (OLS) and the OpenROADM multi-source agreement, which focuses on functional disaggregation and covers pluggable optics, transponders and ROADMs.

This document (D4.1) reports the initial design of the METRO-HAUL control architecture, justifying design choices and deployment models. D4.1 covers a preliminary specification of the interfaces in view of integration and interoperability, listing selected components to be developed and to be integrated in demonstration scenarios. For each component, D4.1 details its implementation roadmap, functional regression and integration tests, and the means to validate the implementation, including control KPIs.

For the proposed architecture, we identify the interfaces between components, along with the macroscopic requirements associated to these interfaces. Where applicable, existing interfaces (from Standards Development Organisations (SDOs) or from OSS projects) are reused (extended if need be). Finally, we list the WP4 components that are being designed and implemented in view of demonstration and proof-of-concepts. Each component is characterized in terms of functional roles, exported interfaces, and KPIs, as well as the associated methodology, schedule and means of validation.

Document structure

This document is structured as follows:

- Section 1 is the introduction. We briefly present the **goals of the control, orchestration and management system** of the METRO-HAUL project, insisting on key innovations such as the extended use of **telemetry, monitoring and data analytics** enabling artificial intelligence and machine learning applications, use cases, and the deployment of advanced network planner subsystems for placement, reconfiguration and re-optimization. We enumerate the main **services exported by / from the point of view of the COM system**, such as data connectivity, virtualized functions, cloud services, ETSI NFV network services and network slicing, as the main ones.
- Section 2 summarizes, for the purposes of this document**, the METRO-HAUL infrastructure, encompassing, notably, multiple cloud enabled optical nodes (termed AMEN and MCEN in the document) interconnected by an SDN controlled disaggregated optical network**. Advanced active monitoring probes are deployed for considered cases such as end-to-end bandwidth estimation. A state of art identifying the main related projects and initiatives such as Open and Disaggregated Transport Network (ODTN) or Telecom Infra Project (TIP) is also presented.
- Section 3 is the first main section of the deliverable. It presents the **COM Architecture**. After an overview of relevant standards and projects such as ETSI NFV, Broadband Forum (BBF) Cloud Central Office (CO) and Central Office Re-architected as a Data center (CORD), we enumerate the design considerations for METRO-HAUL, and the different systems that compose the architecture: i) At the lowest level, the control of the optical layer, the control of the intra-DC network within each METRO-HAUL macro node and the control of the PON network aggregating users traffic; ii) the hierarchical system for network orchestration allowing the end-to-end control of the network architecture; iii) the integration with IT and cloud resources based on ETSI NFV MANO and how METRO-HAUL supports the concept of slicing; iv) the Monitoring and Data Analytics (MDA) subsystem and v) the Placement, Planning, and Reconfiguration subsystem (NP for short). The section concludes identifying the **main interfaces between functional components**.
- Section 4 is fully dedicated to present the **macroscopic requirements** for each of the interfaces between components. **Relevant considerations** are also listed, such as selection of existing interfaces to be extended, frameworks to be used, etc.
- Section 5 is dedicated to list the **main components of the METRO-HAUL COM**, which are being designed and developed in the context of METRO-HAUL. By functional component we refer to specific subsystems or elements in the WP4/COM architecture (SDN agents, SDN

controllers, net2plan planning tools, MDA systems, or applications) which are developed by project members. Selected components will be candidate for integration targeting experiments in WP5. Several of these components will be released as open source. For each component, the tables below provide a description of the component, its role and functions within the COM system and main goals. A Roadmap of the component is also provided, listing key releases and functions, in line with the project and work package schedule. The tables also provide, for each one, a testing methodology and the means of validation, as well as a set of per component KPIs.

- Section 6 concludes the deliverable.

List of Authors

	Name	Partner
	Ramon Casellas, Ricardo Martínez, Ricard Vilalta, Raül Muñoz, Michela Svaluto	CTTC
	Luis Velasco, Gabriel Junyent, Jaume Comellas, Marc Ruiz	UPC
	Francisco Javier Moreno Muro, Miquel Garrich, Pablo Pavón	UPCT
	Guido Maier, Massimo Tornatore, Francesco Musumeci, Sebastian Troia	POLIMI
	Alessio Giorgetti, Filippo Cugini	CNIT
	Roberto Morro	TIM
	Oscar González de Dios, Rafael Alejandro Lopez da Silva	TID
	Jorge López de Vergara, Rafael Leira, Mario Ruiz	Naudit
	Danish Rafique	ADVA
	Daniel King	Old Dog
	Antonio Napoli	Coriant
	Navdeep Uniyal	UNIVBRIS
	Anna Lisa Morea	NI
	Chris Matrakidis, Evangelos Kosmatos	OLC
	Antonio D'Errico	TEI
Checked by:		
	Alessio Giorgetti	CNIT
	Daniel King	ODC
	Victor López	TID
		EURESCOM

Revision History

Revision	Date	Responsible	Comment
0.0	01/04/2018	Editor	Initial version, ToC
0.2	14/05/2018		First integration of Contributions
0.4	15/05/2018		Circulated version
0.6	22/05/2018		All contributions integrated, including partial reviews
0.6b	22/05/2018		Refinements on the CORD and Cloud CO architectures
0.7b	23/05/2018		Version for internal Review
0.8	25/05/2018		Alignment in terminology with D3.1
0.8b,c,d,e	29/05/18		Integrated reviews.
1.0	31/05/2018		First release for Eurescom review

Table of contents

List of Figures

List of Tables

1 Introduction

The constant need to dynamically provision services in a cost-effective way, within complex end-to-end scenarios, spanning multiple knowledge domains, technologies and administrative boundaries has driven the evolution of architectures and protocols for the operation of networks (more recently and generically, telecommunication infrastructures), referred to as their *control, management and orchestration*. Such services have grown in scale and complexity, from conceptually simple voice and data connections in homogeneous networks within the scope and control of a single administrative entity, to services requiring the allocation of heterogeneous resources with complex placement constraints and highly dynamic usage patterns in an environment characterized by having multiple actors and stakeholders. This automation requires the development of service and resource orchestration platforms that extend, integrate and build on top of existing ones, macroscopically adopting Software Defined Networking (SDN) principles and are conceived combining centralized and distributed elements.

In this line, the overall METRO-HAUL objective is to architect and to design cost-effective, energy-efficient, agile and programmable metro networks that are scalable for 5G access and future requirements, encompassing the design of all-optical metro nodes (including full compute and storage capabilities), which interface effectively with both 5G access and multi-Tbit/s elastic core networks.

In view of the requirements associated to 5G networks, related to service provisioning and traffic dynamicity, a key component of the metro network is the control, orchestration and management (COM) system that allows the dynamic provisioning of services. In particular, the COM system augments the concept of network control plane (CP), where a CP is a system and set of functions, specially dedicated to the dynamic and on-demand provisioning of network connectivity services between endpoints, with standard interfaces operating across domains ensuring vendor inter-operability. The CP is responsible for configuring associated switching and forwarding state at the data plane level.

Software Defined Networking (SDN) is simplistically defined as a centralized control model architecture and protocols, highlighting the CP and data plane (DP) separation, and enabling an application layer. The simplest architectures encompass a single, logically centralized Controller (control layer) on top of the data plane network elements (NE) or devices (infrastructure or data plane layer), with the control logic placed within the controller. The interface and (associated protocol) by which a controller communicates with devices is referred to as the South Bound Interface (SBI), while the set of Application Programming Interfaces (API) offered to applications is named the North Bound Interface (NBI).

A device information model macroscopically describes device capabilities, in terms of operations, not detailed and using high level abstractions – more recently this is known as declarative or intent driven networking. A data model determines the structure, syntax and semantics of the data that is externally visible. YANG [YANG] is a data modelling language, where a model includes a header, imports and include statements, type definitions, configurations and operational data declarations as well as actions (RPC) and notifications. The language is expressive enough to structure data into data trees within the so called datastores, by means of encapsulation of containers and lists, and to define constrained data types (e.g. following a given textual pattern); to condition the presence of specific data to the support of optional features and to allow the refinement of models by extending and constraining existing models (by inheritance/augmentation), resulting in a hierarchy of models.

YANG has become the data modelling language of choice for multiple network control and management aspects (covering devices, networks, and services, even pre-existing protocols). For example, an SDN controller may export the underlying optical topology in a format that is unambiguously determined by its associated YANG schema, or a high-level service may be described so that an SDN controller is responsible for mediating and associating high-level service operations to per-device configuration operations.

The goal of the WP is to design and implement a scalable, flexible and modular control plane for a cost-efficient optical metro network, in cooperation with WP2 related to architectural aspects, and with WP3 in order to define abstracted devices, nodes and network models, to be demonstrated in integrated scenarios within WP5. The control plane should have support for multi-tenancy, network virtualization and slicing and, consequently, the key WP4 objectives are: i) to identify control plane requirements and related KPIs from a set of functional and requirements, starting from a set of targeted METRO-HAUL services and use cases; ii) to define the network control and management architecture for an abstracted METRO-HAUL network and infrastructure, including service management aspects as well as network and system monitoring, iii) to define information and data models, covering from devices to systems (nodes), and to provide as inputs to standards defining organizations (SDO) in cooperation with WP6 and finally, iv) to develop a control plane prototype apt for integrated demonstration with node and network prototypes as implemented within other work-packages, apt for integrated demonstration within WP5, including prototype an enhanced SDN controller with a monitoring framework, ETSI/MANO stack and a multi-tenancy/slicing layer. The macroscopic view of this goal and relationship to tasks is shown in Figure 1.

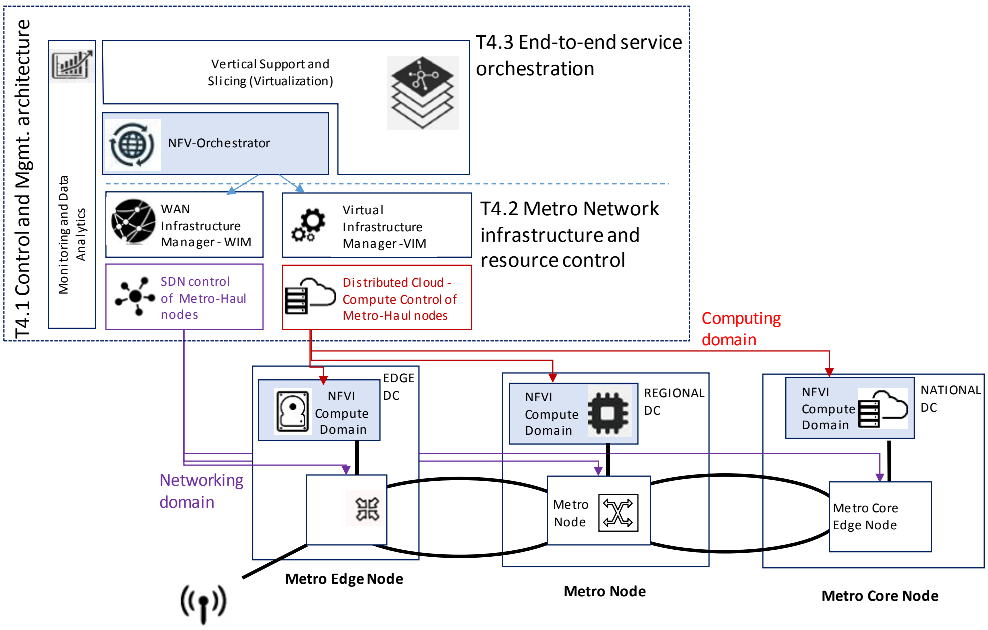


Figure 1 METRO-HAUL WP4 Macroscopic View

1.1 Monitoring and Telemetry as Key innovations

The ever-increasing number of connected devices and the proliferation of multimedia services imposes a major impact on network resources and operation, reflected in an increased traffic dynamicity and network incidents, that might impact the quality of service. In consequence, deploying a COM system to automate service provisioning and reactively proceed against failures, does not always guarantee network resiliency and optimal resource usage. To improve the probability of recovery from catastrophic failure and resource efficiency, the COM system must be augmented with a telemetry capability.

Network and IT performance needs to be continuously measured so that events like degradations or capacity exhaustion can be anticipated and resources can be proactively managed. Monitoring and telemetry are automated processes by which measurements data from network and IT devices, applications and services, are collected, transmitted, and stored in a certain repository, to be used as input of Data Analytics (DA) algorithms that provide outputs like failure detection and localization and daily traffic models, just to mention some examples.

Traditional network management platforms maintain visibility of network behaviour, via control plane and data plane SNMP probes, providing analytical data to the operator or high-layer Service Layer Agreement (SLA) monitor. In highly dynamic environments, information provided via SNMP is either too restrictive, and very often too slow. Furthermore, SNMP is based on a “pull model”, which requires additional processing to periodically poll the network elements, and limits scaling in large environments with numerous collection points for statistical data. Other monitoring methods include protocol-based solutions, like NetFlow and IPFIX [IPFIX], but these are not accurate enough to detect issues caused by short-lived events or extremely bursty traffic, that may impact higher-layer applications.

In addition, the use of distributed physical traffic monitoring probes may be possible, but in optical environments these devices to monitor traffic require significant interface bandwidth support, and often lack the ability to correlate metadata and history, which makes tracking and responding to events impossible at scale.

1.1.1 Applicability to SDN

The discussions around SDN have almost exclusively focused on separation of data and control planes, leverage programmability and the use of open Application Programming Interfaces (APIs) with little to no attention on overall operational feedback loop, including monitoring, intelligence and management functionalities. Figure 2 captures this theme in an Autonomic Networking (see [RFC 7575]) architecture, where network resources—physical or virtual—are continuously monitored using a telemetry engine, exposing real-time network states to the analytics stage, which in turn feeds into the control and management planes. This holistic platform not only caters for centralized and programmable control, but also makes ML-driven decisions to trigger actions, essentially connecting data-driven automation with policy-based orchestration and management.

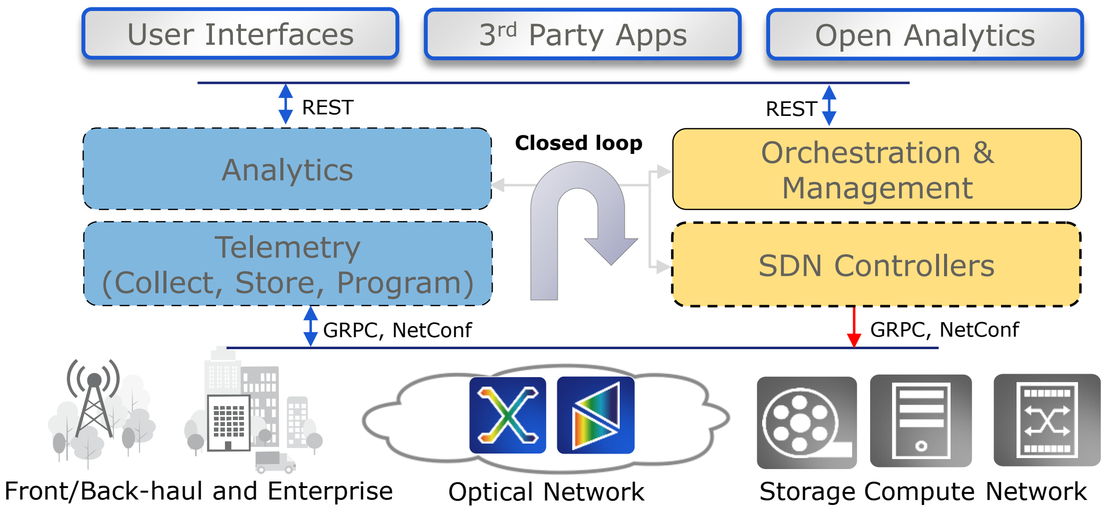


Figure 2 Autonomic Networking Architecture

1.1.2 Targeted Monitoring and Data Analytics use cases

1.1.2.1 Open Line Systems monitoring

An Open Line System (OLS) is a key-element of disaggregated networks and opposes to traditional integrated wavelength division multiplexing (WDM), characterized by single vendor solutions. An OLS divides a network into functional blocks (such as transponder, switching fabric and routing) that allow service providers to select the best-in-class product for each of them. An OLS includes important monitoring functions. A typical use case is the detection of alien wavelengths generated by third-party interfaces within disaggregated networks. This functionality has been widely utilized in proprietary WDM and it is needed to optimize the power levels of the channels along the transmission line, for example by varying the attenuation through a wavelength selective switch (WSS) according to the rules of the hosting WDM line system's and its link control algorithms.

With the advent of coherent technology, a vast series of parameters, besides the ones provided by individual network elements, have been made available to the SDN control plane. Channel parameters, e.g., dispersion, differential group delay (DGD), polarization dependent loss (PDL), can be now precisely estimated, together with component characteristics such as carrier frequency offset through digital signal processing (DSP) algorithms. This knowledge enables dynamic control of the bandwidth variable transponders (BVT) and thus design of sophisticated real-time network management with different level of optimization. Missing parameters could be obtained via machine-learning, by correlating them with other parameters provided by the network.

Besides the above parameters, the receiver also delivers the pre-forward error correction bit error rate (pre-FEC BER). This unbiased information mirrors the status of the lightpath under test, with little possibility to determine the causes of fault or of need for configuration update. A more relevant parameter is the optical signal-to-noise ratio (OSNR), which could be derived from the definition or by using the Gaussian Noise model using the parameters estimated via DSP and provided by other network elements, e.g., the amplifier output powers. The OSNR is a sign of the health of the system and it is widely used by operators to evaluate whether actions are required to guarantee full operation.

All this information cannot be properly used without the possibility to manage the BVT and all network elements via software. For example, within the BVT, we can vary: modulation formats, symbol rates and FEC, and adapt them to instantaneous quality of transmission, assessed through the OSNR. We envision that this information require standardization for what concerns the parameters to be provided and their accuracy, so that a proper monitoring of the network can be guaranteed. Multi-vendor interoperability represents a fundamental enabler for the adoption of OLS. This has been part of industrial initiatives such as OpenConfig [OpenConfig], Open ROADM [OpenROADM] and the Open Optical Packet Transport project within the Telecom Infra Project (TIP) [TIP]. The first defines the interoperability specifications for ROADMs, transponders, and pluggable optics with specifications for YANG [Yang] data models as well in a metro scenario. The second focuses on DWDM open packet transport architectures enabling innovation and avoiding vendor lock-in. All this cross-companies and institutions work dealt also with the standardization of the FEC for interoperability, which has enabled WDM interface optics from different vendors at each end of individual wavelengths. The main issue for standardization is that the derived staircase FEC (with 7% hard decision) can provide only 9.38 dB net coding. On the other end, best-in-class proprietary FECs are typically closer to 12 dB, so interoperability comes at an excessive cost in terms of reach/capacity, and interoperable FEC is likely to be restricted to metro applications, because in this scenario power consumption plays a fundamental role.

1.1.2.2 Transport Network Re-Optimization with Machine Learning and Data Analytics

Physical layer network design and planning process is a cumbersome one. It includes laying out all possible combinations of modulation formats, fiber types, forward error correction codes, channel grids, etc.; conducting exhaustive simulations and lab experiments to come up with carefully tuned engineering rules, and finally using these approximate analytical models to propose transmission feasibility. The fundamental challenges associated with such solutions are their static nature, conservative design principles, leading to resource underutilization, and finally their lack of scalability —neither from planning viewpoint nor computationally—to next-generation highly granular and flexible networks.

ML/AI may be applied to solve some of issues highlighted in this section, by allowing data-driven model development, and consequent physical layer optimization based on real-time transmission quality monitoring. To enable such an approach, however, optical performance monitoring needs to be performed with uninterrupted access to the diverse networking infrastructure. This necessitates disaggregated systems, served by common interfaces, interoperable across different equipment suppliers and vendors. Exploiting an extended SDN architecture, model-driven telemetry has the power to disrupt the way transport networks are designed and operated today. While a challenging prospect, both in terms of level of disaggregation and interoperability requirements, early research and collaborative projects show promise and drive by various industrial players.

Key events that the METRO-HAUL system will manage on the network side are the requests for connections. The simplest form of connection that can be requested is point-to-point, form a source to a destination. By source and destination, we generically mean host interfaces, where a host can be either a physical or a virtual machine, located in site-based, geographical locations or METRO-HAUL nodes either an Access Metro Edge Nodes or in a Metro Core Edge Node (MCEN).

Thus, the connection will comprise segments internal to such nodes and segments between such nodes. The internal (or intra-node) segments correspond to allocation of resources in a packet-switched network, while the external (or inter-node) segments require allocation of resources on the optical network. The function of translating a request into the needed set of intra- and inter-node resources is normally carried out by the Network Planner (NP). Requests to the NP may come directly from the Operations Support System (OSS) of the network operator, or from the NP itself. The NP satisfies all connection requests on demand and dynamically by performing path computation and resource allocation according to suitable matheuristic[1] algorithms. The framework adopted to run these algorithms is the Net2Plan open-source software [net2plan].

1.1.2.3 Proactive network re-optimization

The COM system collects statistics and historical data regarding all the requests, to predict demands by machine learning and to re-optimize periodically all connections in order to ensure an efficient network-resource optimization on the long run. Prediction also allows enhancing the algorithms used for on-demand provisioning. This technique is particularly effective on the tidal components of the traffic, such as those generated by daily movement of citizens in the urban area, or by regular jobs in enterprise applications.

In this regard, it is also important to consider the concept of autonomic infrastructure that proactively self-configures and self-tunes in a cost-effective manner (near) real-time to adapt its resources to future conditions. Thanks to the application of DA to monitored data, control loops can be enabled, where outcomes of such analysis can be used for event notifications together with recommended actions to SDN controllers and/or NFV orchestrators.

As an example of control loop, optical layer monitoring along the whole path helps to detect BER degradations in advance as well as to localize the cause of soft-failures, as demonstrated in [1]. Once the system has localized the cause of the failure, e.g., an optical link, a notification can be sent toward the SDN controller with a recommended action to restore connections affected by the failure.

Another example is in the case of multilayer networks, where virtual network topologies (VNT) are usually created to improve resource utilization or provide per-service isolation. In such cases, daily variation in traffic patterns forces operators to overprovision VNT capacity to be able to accommodate peak-hour traffic. As an alternative, traffic patterns can be modelled by using monitoring data collected from the nodes. With such models, prediction of the traffic can help to dynamically manage the capacity of the VNT, thus greatly reducing overprovisioning.

[1] Optimization algorithms made by the interoperation of metaheuristics and mathematical programming (MP) techniques

1.1.2.4 VNF placement across transport networks

In a METRO-HAUL network, enabling the execution of Virtual Network Functions (VNFs) close to the users, extending the concept of NFV towards the edge is an emerging requirement, often due to latency and jitter requirements of a service and/or application. The Network Service (NS) concept plays a central role, as a set of VNFs and traffic flows between them that should be atomically allocated (or blocked). As such, a NS can involve both IT and network resources, handled by per-metro node VIMs (Virtual Infrastructure Managers,), and the transport network SDN controller/s, respectively. Note that deciding the VIM/s where the VNFs of a NS are allocated, is unavoidably coupled to the dimensioning of the transport flows between the metro nodes hosting the VIMs.

The joint network orchestration may rely on an external planner. Its role is optimizing NS provisioning in the short scale and implement medium and long term joint capacity planning of the IT and transport network resources. In this view, the dimensioning of the Data Center (DCs) and the dimensioning of the transport network connecting them is jointly addressed. This requires active monitoring of heterogeneous resources encompassing IT and network ones.

1.2 METRO-HAUL Control, Orchestration and Management Services

The METRO-HAUL COM is responsible for the dynamic provisioning of services. At the time being, and from a control of management point of view, we have identified a set of basic services to be developed and on top of which main use cases will be demonstrated.

1.2.1 Network connectivity services

A key focus for the project will be the dynamic provisioning of data connections in a multi-layer network, with a disaggregated optical network at the core. In a priority order, the main services to be addressed are:

- Connectivity provisioning between Optical Layer ports (media channels).
- Connectivity provisioning between MAC endpoints. These MAC endpoints can correspond to physical devices (e.g. hosts) or NICs of VMs inside data centers. This may include VLAN segmentation.
- Connectivity provisioning between IP addresses (ex. VXLAN VTEP at each data center node)
- Configuration of PON

Similarly, the WIM (WAN Infrastructure Manager - see Section 4) is responsible for provisioning connectivity provisioning for VNF (Virtual Network Function) paths. The WIM will rely on the services of the SDN control plane for this.

1.2.1.1 Optical Transport Connectivity

As further detailed in [D31] such services involve the transport of client signals over an optical WDM transport Network infrastructure, in the METRO-HAUL context. Basic connectivity services are the reservation / creation / modification / deactivation of end-to-end transparent transport "Optical Tunnels" in the network between two access points (also referred to as Service Interface Points or SIP), for well-specified digital client or analogue CBR signals. Each optical transport network domain is abstracted toward higher hierarchy entities as a single managed topological entity with a defined number of Service Interface Points. Two main types of services are expected:

- **Digital Optical Services:** The activation / deactivation of a transparent point-to-point digital transport channel is requested between two SIPs of the OTN network for a specific customer technology (Eth, OTN, SDH, etc ...). Upon activation, the details of the characteristic information to be transported are also specified (customer type, bit-rate, etc.); the requests and the constraints to be respected for the selection of the routing (differences respect from other flows, latency, etc.); the characteristics of desired resilience and the QoS level.
- **Analogue optical transport services** This transport service is of an analogical nature and requires the presence of optical continuity with acceptable degradation of the client optical signal between the two SIPs of the analogue optical layer of the server network domain. The service is aimed at optical channel customers OTSiG (or Och) [G.872] that carry OTU digital information modulated on one or more carriers and consists in providing a media channel between the two media port of access to the service, configured to carry the whole OTSiG optical signal (Och), that is the whole set of modulated optical carriers on which the digital information is distributed. It therefore assigns a single media channel to contain all the OTSiG which is then routed as a single entity within a flexi-grid network. [G.694.1]**.**

1.2.2 Cloud computing (e.g. VIM) Services

METRO-HAUL relies on having a dedicated VIM (Virtual Infrastructure Manager) per location / Central Offices. Such VIMs are responsible for the instantiation of Virtual Machines (VMs) and VNFs in specific METRO-HAUL nodes; ability to configure VMs and attach virtual NICs to soft switches.

The COM system shall enable VIM services, which may or may not be part of a NFVO (Network Function Virtualization Orchestration) Network Service. Such services are related to the instantiation of VMs, etc.

1.2.3 Network Virtualization, ETSI/NFV Network Services (NS) and slicing

As transport networks evolve, the need to provide network abstraction and virtualization has emerged as a key requirement for operators. Network virtualization refers to the process by which multiple logical (virtual) networks are supported over a common, shared physical network infrastructure [Nej11]] (note that the physical aspect is understood at the lowest level. Since network virtualization can become recursive, a logical virtual network may also be virtualized). A common research problem is e.g. virtual optical network embedding (see [Gon14]). Network virtualization is an enabler for multi-tenancy [Li17], an ownership concept in which tenants are given a different partial and abstracted topology view and are allowed to utilize and independently control allocated virtual network resources as if resources were real. The granularity level of control given to tenants can vary, depending on the involved new business models. The mechanisms to actually support network virtualization are diverse, and strongly depend on the uses cases and associated requirements, notably in terms of traffic isolation, service level agreements and performance guarantees. In most cases, such mechanisms rely on a combination of i) actual hardware device support for multi-user and virtualization ensuring resource and traffic isolation and ii) software layers and middleware that perform the necessary control functions.

For example, an optical flexi-grid network can be partitioned, based on a selection of NE or ROADMs, physical ports or link fibers and nominal central frequencies of the DWDM grid (hard partitioning) so a virtual network is thus a subgraph of the underlying network topology graph. A Bandwidth Variable Transceiver (BVT) can tune its bit-rate and bandwidth dynamically with a trade-off between reach and capacity. A BVT may, in turn, be composed of multiple transceivers, each one of such sub-transceivers being configured independently. Such sliceable BVT (S-BVT) enables transmitting from one point to multiple destinations, changing the traffic rate to each destination and the number of destinations on demand. Consequently, a set of such sub-transponders can be assigned to support one or more logical links of the virtual network. In another scenario, a virtual network can be an L2/L3 Overlay Network, i.e., an arbitrary L2/L3 network in which software-based switches and routers are instantiated in specific hosting nodes and virtual network links are supported over physical network paths provisioned with actual resource reservation or relying on overprovisioning and statistical multiplexing.

Control plane architectures, which initially assumed direct control over the physical resources need to be extended to explicitly support multitenancy and control of virtualized resources (e.g. an SDN controller for a transport network may need to implement a dedicated interface towards a network hypervisor instead of the SBI to the network element). From the automation perspective, a challenge is to conceive not only systems able to allocate, manage and deallocate a given slice during its life-time (as in a Slice-as-a-Service or SlaaS), but also to be able to provision potentially dynamically, control plane instances for the specific control of the allocated slice, supporting a wide range of control models, i.e., from basic monitoring of the slice operation to a full control on the slice down to the constituting elements of the slice. For example, a tailored ETSI NFV MANO system can be instantiated associated to the slice lifetime for the instantiation of Network Services over the actual elements of the slice. Another relevant challenge is to support this concept across multiple (federated) domains across administrative boundaries.

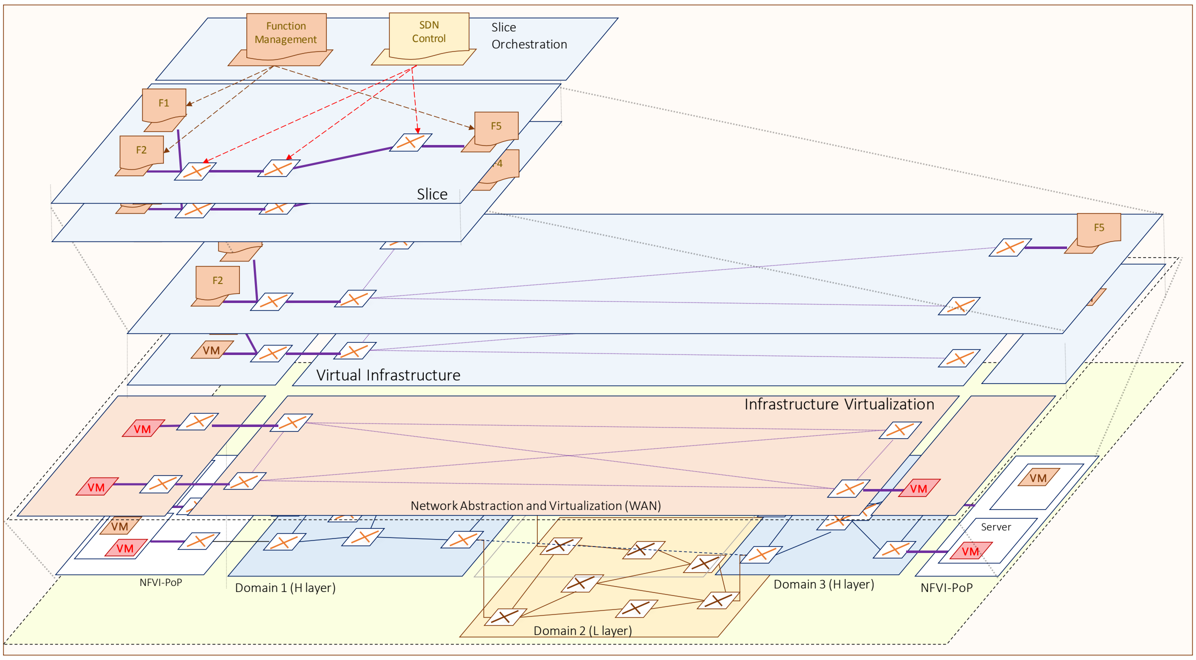


Figure 3. Network Slicing Concept: Virtualize an infrastructure encompassing network, computing and storage resources, so virtual infrastructures can support interconnected functions tailored for a service or customer, with dedicated control, management and orchestration

Figure 3 illustrates the concept of having a physical infrastructure (composed of network, computing and storage resources) that can be virtualized, resulting in multiple virtual infrastructures. Such virtual infrastructures are composed of virtual inks that interconnect VMs, the latter supporting generic functions (F1, F2…) forming a logical network or construct (slice). Resources within this slice can also be orchestrated, with systems dedicated to managing the composing functions or instantiated SDN control planes for the virtualized network elements.

Generalizing the concept of network virtualization and driven by recent standardization work at SDOs such as 3GPP, IETF, ETSI, the term *Network Slicing* has appeared as an emerging requirement for future 5G networks. While the roots of the concept are related to network virtualization, including the partitioning (slicing) of a single (commonly physical) infrastructure in order to construct multiple (logical) infrastructures, there are important differences that are worth highlighting. In particular, more emphasis is given to the actual network functions and how they are arranged and configured, forming a complete logical construct or network, tailored, customized and optimized for a given service or service set, or to support a given actor or customer (e.g. vertical industry). Second, a given slice can combine both data and control plane functions and functional elements, which are inherent part of the slice. In this context, concepts such as traditional data connectivity services such as Virtual Private Networks (VPN), Network Virtualization or NFV Network Services become specific cases of this generic construct.

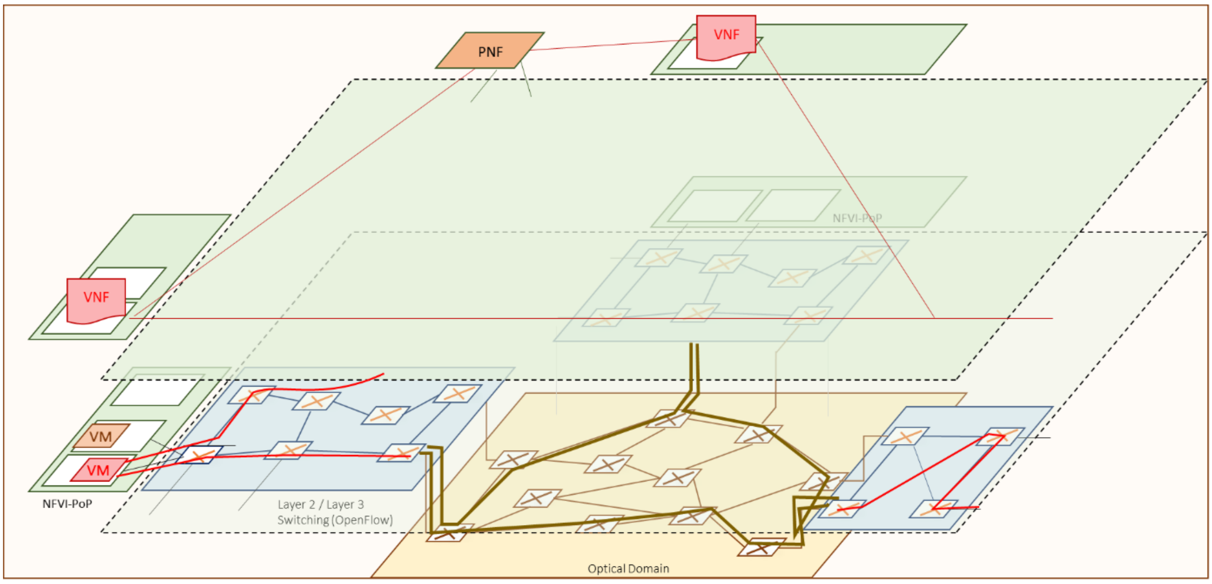


Figure 4. ETSI NFV Network Service as canonical METRO-HAUL 5G slice (scope)

1.2.4 Monitoring Services and Data Analytics

Cognitive network architectures have been proven to self-adapt the network in a cost-effective manner (*autonomic networking*). Specifically, by applying data analytics to monitored data, *observe-analyse-act* (OAA) loops can be enabled in the network, where analysis outcomes can be used to recommend network reconfiguration actions to the SDN controller. Autonomic networking entails thus the capability to do measurements on the data plane and generating data records that are collected and analysed to discover patterns (knowledge) from data (KDD).

To be able to define autonomic network architectures, a number of interfaces need to be defined between the MDA subsystem and other systems within the METRO-HAUL architecture. Specifically, these interfaces must support:

- to collect measurements from the devices using monitoring / telemetry capabilities.
- to correlate measurements with operational entities, e.g., topology, connections, etc.
- to synchronize network status and provide recommendations.
- and to retrieve metered data and knowledge.

1.2.5 Network Planning and Optimization services

In the METRO-HAUL architecture there is a subsystem dedicated to network planning and optimization, which will provide these two important services to the whole network system. These services deal with resource assignment to requests that come to the network from the operator and to demands that come from the users. By resources we mean: a) computing resources, w.r.t. deployment of virtual network functions in suitable virtual machines in the compute nodes; b) network resources, w.r.t. routing of connections or instantiation of traffic flows over the nodes and links of the network infrastructure. There are two main operational modes to exploit the service offered by the planning and optimization subsystem.

The first mode is *reactive* and it is operated each time there is a request for a service that is coming to the network. Probably the best way to explain how this mode works, is to describe the workflow that is associated to its utilization. The request is issued by the OSS to the service orchestrator, which converts it into a service chain, i.e. a sequence of VNFs to instantiate. The service orchestrator then sends a request to the network planner to obtain a solution of optimal placement of the VNFs in the network compute nodes: this is a first invocation of the planning tool. The tool will send the reply to the orchestrator and at the same time to the VIM controllers, so that the VNFs can be instantiated in the suggested compute nodes. The service orchestrator will forward a request to the SDN top-level controller to set up suitable connections to support the service chain in terms of network connectivity, i.e. to create the set of flows that allow the connection of the VNFs of the chain. The SDN top-level controller will invoke the network planner a second time asking for the network resources to be allocated to the connections. The planner will propose an optimized solution of routing and resource allocation for those connections. The second operational mode is *proactive*: in this case, the role of the network planner is to propose rearrangements of existing active service chains and network connections with the purpose of globally reoptimizing resource allocation to improve network performance. The reoptimization is not simply designed considering the current state, but in the light of the evolution of the network traffic in the next future period, as it can be predicted by observing traffic for a given number of past epochs. Predictions are carried out by suitable algorithms based on machine learning. The proactive reoptimization service can be carried out periodically or it can be triggered by monitoring utilization of resources and detecting degradation above certain thresholds.

This approach of implementing the planning and optimization service into a dedicated subsystem has several advantages that can be summarized as follows:

- The service and network orchestrators – especially in case of open-source software – often do not have built-in optimization algorithms or, if they do have, they commonly rely on well-known algorithms, not easily replaceable with others. By developing dedicated planning and optimization algorithms, we have full control on the math and the logic and we can easily upgrade and modify them in order to make them more powerful.
- The network operator has easy access to the algorithms and thus can customize the service by deciding his own policies and criteria.
- Since the METRO-HAUL network is an experimental environment, it is important to have the possibility of easily testing new optimization and prediction algorithms.
- We have decided to implement the network planner exploiting Net2Plan, a software framework purposely developed to favor testing and implementation of optimization algorithms.

1.2.6 Other relevant services

The project will also consider other services that a COM system can potentially offer. In particular, of special interest are the L2VPN and L3VPN. L2VPN and L3VPN services have been at the core of most of the connectivity services provided by network operators in the last decades. They have been used as an integral part of the network infrastructure of the network operator itself, as well as being a key part of the service portfolio offering to the customers. WAN connectivity technologies have been mainly exposed to the customers, whether internal or external, under the paradigm of Layer 2 or Layer 3 VPN services for them to consume them.

As such, METRO-HAUL will keep the L2VPN and L3VPN paradigm as a way of packaging the connectivity services to external customers. That way METRO-HAUL can benefit from the previous work on these paradigms in terms of modelling. In addition to this, L2VPN and L3VPN services also ease the extension of connectivity services to other non-METRO-HAUL, “legacy” nodes on an E2E fashion. At the same time, internal to the METRO-HAUL architecture, multisite underlay connectivity between METRO-HAUL nodes can be modelled for some use cases as L2 or L3 VPN services to be consumed by the WIM on behalf of the NFVO.

For that purpose, METRO-HAUL must analyse the existing service models suitable for the L2VPN and L3VPN services and decide which ones are applicable to the METRO-HAUL use cases and complement them if needed. IETF data models for VPN service delivery can be the starting point in that respect.

From the service models, the network models applicable to the METRO-HAUL nodes must be derived aligned with the METRO-HAUL nodes implementation of L2 and L3 VPN connectivity services and its support must be defined across the different METRO-HAUL architecture interfaces.

2 Overview of the METRO-HAUL Data Plane Infrastructure for 5G

2.1 Macroscopic view

The METRO-HAUL infrastructure (spanning multiple geographic locations) relies on macroscopic nodes, combining networking, processing and storage resources. Such modular devices are composed of different components operating at different layers and technologies, and of different vendors, realizing hardware and software disaggregation inside the node. It implements layer 0-1 (optical domain), layer 2 transmission and switching (frame domain) for add/drop purposes, and moreover Edge Computing capabilities, provided by local pool of servers to instantiate VNFs with configurable amount of processing, memory and storage. The aforementioned devices are controlled by an SDN/NFV Node Controller handling the integration of such disaggregated components.

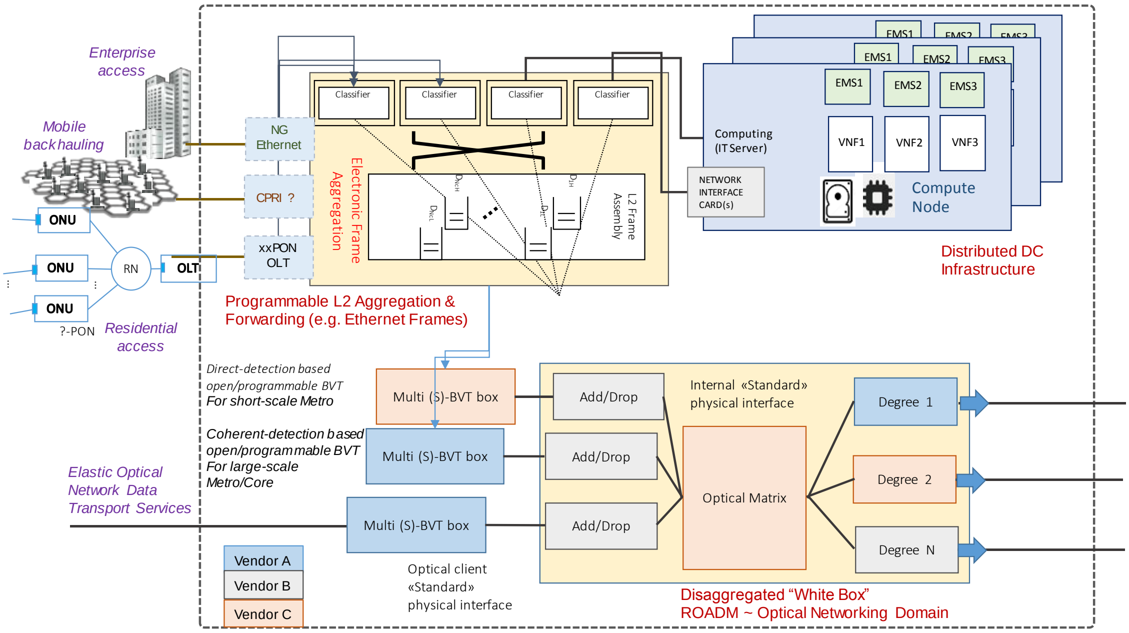


Figure 5. Disaggregated Optical Networks: METRO-HAUL Node.

Figure 5 shows the architecture of the METRO-HAUL Node. The general node architecture includes VNFI capabilities with a local Compute Node, realizing caching and virtualized services running close to the node to serve users with reduced latency. The electronic frame aggregation implements classifiers and switching directing the traffic from/to access and metro-core sections. In particular, in the AMEN version, 5G access (e.g., fronthaul interfaces, CPRI) and optical access (e.g. xPON) are directly attached to the aggregation stage. Traffic to the metro network is directed to the optical white box. Multi-vendor SBVT with different capabilities are employed. In fact, in the edge version, SBVT with direct detection for short reach metro traffic (e.g., 10, 40 Gbps bitrate) are used, whereas in the core version, SBVT with coherent detection and high speed (e.g., 100, 400 Gbps bitrate) for long reach core transport are utilized. The SBVT transponders are connected to Add-Drop stages, optical switching matrix devices and degree devices (including WSS and line amplifiers). Each device, equipped with a dedicated SDN agent (i.e., NETCONF agent with specific YANG models), may be of different vendor and connected by means of standard data plane interfaces. Concrete specializations of the generic architecture are:

- AMEN nodes to interface with heterogeneous access technologies
- MCEN nodes

AMEN nodes implement local/regional edge computing, aggregation and optical metro communication, along with access networks gateway, with both 5G-based wireless and optical capabilities (e.g., fronthaul interfaces, CPRI, xPON) and MCEN nodes are gateways towards the core transport network and comprise core-oriented capabilities, among which the possibility to perform EON-based multi-vendor alien wavelength provisioning and control.

METRO-HAUL nodes support the instantiation on user-oriented and operational functions supporting heterogeneous services and are interconnected through a disaggregated flexi-grid optical network supporting dynamic optical bandwidth allocation and computing resources. Figure 6 shows the main pillars of the METRO-HAUL infrastructure, spanning a regional or metropolitan network. In simple terms, the infrastructure is composed of a set of heterogeneous resources, mainly:

- *An optical network* (in yellow) the Disaggregated Optical Layer that provides high bandwidth, low latency connections between nodes and constitutes the core part. This optical network is obtained as a combination of Optical-Network Elements (O-NEs), encompassing optical components with a varying degree of abstraction, providing flexibility regarding deployment models.
- *Packet switched networks* (in blue), providing connectivity to customers, NVFI servers and related endpoints. networks that transport traffic coming from access and aggregation networks (such as PON) and which provides connectivity to functions and applications running in the local DC.
- *Passive Optical Networks* (PON) that provide connectivity to users.
- *ETSI NFVI Points of Presence* (NFVI PoP) encompassing multiple compute nodes in which virtualized network functions can be executed: A Computing and Storage Infrastructure, with a variable number of computing, storage, and virtualization servers that are potentially available at every node (loosely referred to as a local Data-Center, or DC). It is assumed that such compute nodes (i.e., virtualization servers) are attached to the packet switched networks e.g. by means of Ethernet interfaces.

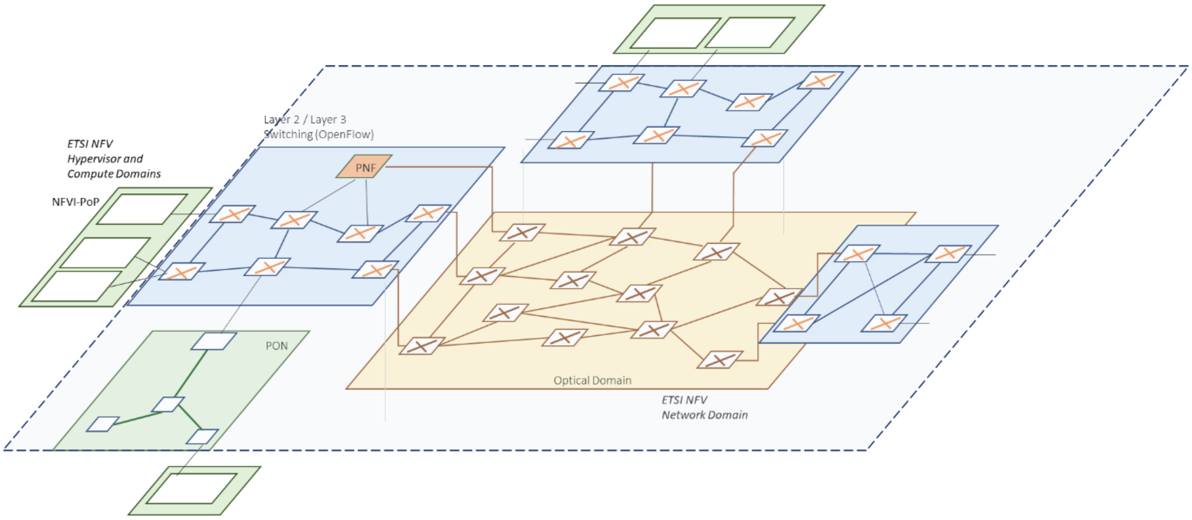


Figure 6 Macroscopic METRO-HAUL infrastructure

2.1.1 Related Initiatives and Projects on Distributed Cloud

2.1.1.1 BBF Cloud CO NFVI

A related initiative that METRO-HAUL architecture will consider for the integration of packet switched network infrastructure and VNFs in the METRO-HAUL architecture is the Broadband Forum CloudCO (Cloud Central Office) project.

The Broadband Forum (BBF) has defined in its TR-384 [TR-384] a Cloud Central Office Reference Architectural Framework that re-architects the broadband network using SDN and NFV technologies. It defines a CloudCO Domain as a set CloudCO Macro-Nodes that share a CloudCO Domain Orchestrator, which includes an ETSI MANO NFVO as depicted in Figure 7.

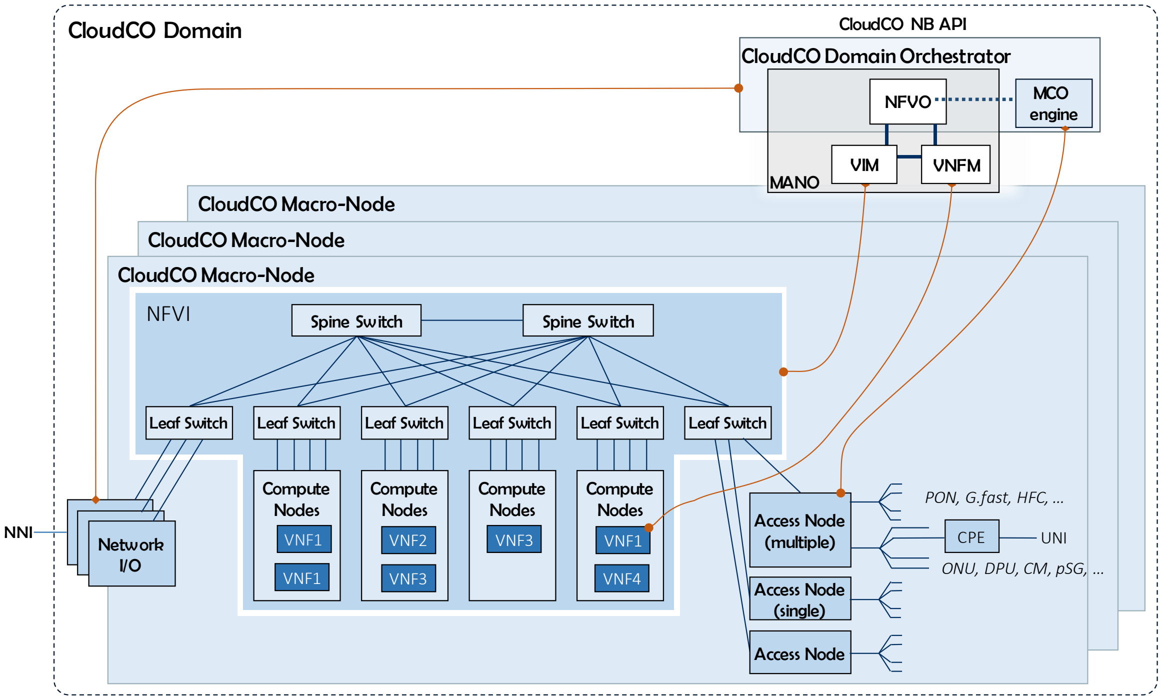


Figure 7.BBF CloudCO macro-node structure (figure 10 from BBF TR-384).

Inside each CloudCO Macro-Node there is an NFVI composed of switches and compute nodes. The leaf-spine switching fabric shown in figure is exemplary, not normative, so other topologies are possible. Network I/O of a CloudCO Macro-Node connects the NFVI to an external "core-facing" network and may be implemented via the existing equipment in the Central Office and/or uplink modules integrated into the leaf switches and/or spine switches. Subscribers are connected to the CloudCO Macro-Node by means of Access Nodes connected to the switches of the NFV Infrastructure (NFVI).

2.1.1.2 CORD

CORD (Central Office Re-architected as a Data-center) is an initiative originally promoted by AT&T and now under Open Networking Foundation (ONF) control combining NFV and SDN to bring data-center economics and cloud agility to the Telco Central Office. It has three main 'flavours' addressing residential, enterprise and mobile services named, respectively, R-CORD, E-CORD and M-CORD; actually, the plan is to overcome this separation in the next releases. Figure 8 shows a typical CORD POD data plane configuration. The main components are the switching fabric and at least two standard x86 servers (one head and one compute node).

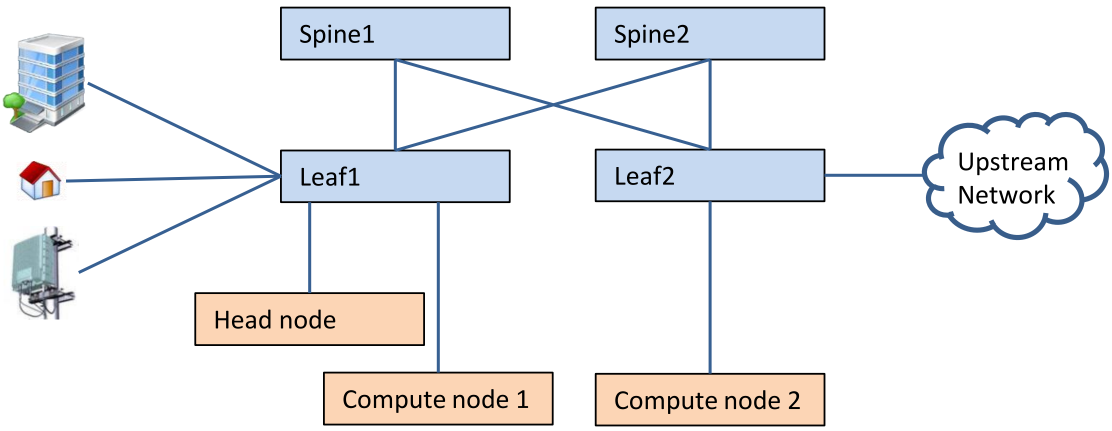


Figure 8. CORD POD architecture

The switching fabric is typically made of four switches assembled to form a leaf/spine configuration; however, both a higher number of switches (when the available ports on the leaf switches are not enough) and a single switch fabric are allowed. Leaf switches are connected to the servers and to the access devices collecting traffic from the users. Finally, one of the leaf switches is connected upstream to the metro network. User traffic, coming from the access devices, goes through different leaves, spines and compute nodes, depending on the location of the needed services. The CORD data plane is completed by an instance of Open VSwitch (OVS) running on every compute node realizing the overlay network interconnecting the VMs and/or the containers that execute the services. For R- and M-CORD use cases, after the manipulation performed by the services (VLAN and/or tunnel termination, NAT, etc.), user traffic is conveyed towards an upstream router as L3 IP traffic. A typical E-CORD deployment is made of multiple (minimum two) CORD sites connected by means of a transport network providing L2 (Carrier Ethernet) connectivity (Figure 9).

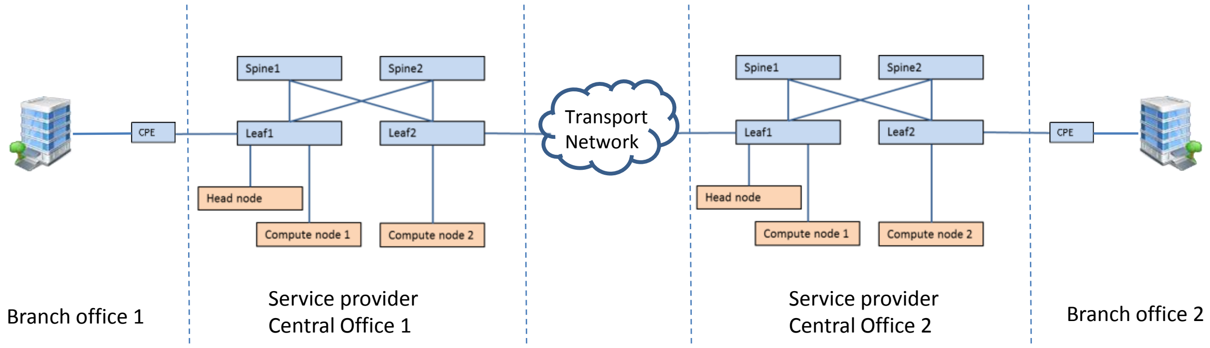


Figure 9. E-CORD scenario

2.1.2 Related Initiatives and Projects on Disaggregated Optical Networks

2.1.2.1 ODTN

Open and Disaggregated Transport Network [ODTN] project, also managed by the ONF, is an operator-led initiative to build data center interconnects using disaggregated optical equipment, open and common standards, and open source software. ODTN aims to drive innovation and become the optical network of choice by disaggregating the components of the network and providing open software to control a multi-vendor assembly of components.

ODTN will enable a white-box optical ecosystem that allows multiple components to be combined and built into complete solutions. Vendors can focus on building a specific component (e.g. transponder) without having to build a complete solution leading to accelerated innovation and lower costs. Operators will have the freedom to select best-in-class components and avoid vendor lock-in, thereby gaining flexibility as their network needs grow.

Technical complexity posed by the analogue nature of long distance DWDM optical communications has driven the need for the current vertically integrated solutions in the market. ODTN's approach is designed to change this by assuming that every optical link will use a matched pair of transponders from a single vendor. However, unlike single vendor solutions, the network can use a different brand of transponder for each link, and these transponders can run over an open line system from yet another vendor.

Leveraging the ONOS SDN Controller, ODTN will automatically and transparently discover the disaggregated components and will control the entire transport network as a unified whole, thus enabling multi-vendor choice. ODTN will rely open industry standards like TAPI (Transport API) and OpenConfig – OpenROADM to achieve a truly vendor-neutral solution.

The ODTN project will start with relatively simple point-to-point open line systems to more complex network scenarios, and end with a meshed network consisting of disaggregated optical equipment.

METRO-HAUL partners are co-founding and participant members of the Open Networking Foundation (ONF) Open Disaggregated Transport Networks (ODTN) project, which also includes operators and companies from US (Infinera, Nokia, Coriant, …) and Asia (NTT, China Telecom, NEC…). METRO-HAUL members' participation includes the Use Case, Testing and Software developments teams. METRO-HAUL has already contributed into ODTN Open source applications and the cooperation is supposed to strengthen in the following months / years.

2.1.2.2 TIP

The Telecom Infrastructure Project (TIP) [TIP] is a Facebook collaborative initiative launched in February 2016 aiming to drive innovation across the entire telecom landscape. TIP counts on over 500 Member organizations and leverages on Facebook's expertise from the Open Compute Project [OCP], which generated Wedge, a fully open and disaggregated top-of-rack Ethernet network switch [Eck16]. Within TIP, the Open Optical Packet Transport (OOPT) project group aims to define DWDM systems including open line system & control, transponder & network management and packet-switch and router technologies. A remarkable outcome in TIP's OOPT group is Voyager [Lyu16], an adaptation of the Wedge top-of-rack switch with pluggable optical transceiver modules supporting up to 200Gb/s for point-to-point transmission links of ~150 km. Additionally, OOPT [Sch17] is divided into working groups that address disaggregated transponder chips (DTG), a Common-API, Open Line Systems (OLS), Disaggregated Cell Site Gateways and a Physical Simulation Environment (PSE).From the METRO-HAUL perspective, the TIP initiative and its efforts toward collaborative openness and disaggregation of traditional vertically-integrated and vendor-locked systems is attractive, especially for the telecom operators in the project [Ric18].

2.2 METRO-HAUL Optical Network

The METRO-HAUL optical network is organized in different hierarchical levels of control, see Figure 10. This hierarchical structure is intended to improve scalability performance by reducing the number of NETCONF sessions required at the METRO-HAUL optical network controller to control all the elements of the (disaggregated) optical network.

At the first level, the optical network is disaggregated into several Optical-Network Elements (O-NEs), typically interconnected by optical fiber links supporting fixed or flexible grids [ITU-T G694]. One or more O-NEs are physically located in each Central Office. Each O-NE includes an O-NE Agent (i.e., implemented using a NETCONF server) that has a direct NETCONF connection to the METRO-HAUL optical network controller (e.g., implemented using ONOS framework), and optionally a telemetry connection to the network control and management system.

Initial work within METRO-HAUL identified two basic O-NEs:

- ROADM in N:1 relationship to Central Office (i.e., N ROADMs located in each Central Office)
- XPONDER Box in N:1 relationship to Central Office

In turn, each O-NE is typically composed of a set of devices/sub-systems (i.e., second hierarchical level in Fig. 9), called Optical Infrastructural Elements (OIEs). The following OIEs were identified:

- **ROADM Degree** (in N:1 relationship to ROADM O-NE)
- **ROADM Add/Drop modules** (in N:1 relationship to ROADM O-NE)
- **Muxponder**: N clients-1 line interfaces (in N:1 relationship to XPONDER Box O-NE)
- **Transponder**: 1 client-1 line interfaces (in N:1 relationship to XPONDER Box O-NE)
- **Switchponder**: N clients-M line interfaces (in N:1 relationship to XPONDER Box O-NE)

Moreover, "pluggable" devices (i.e., WDM transceiver to be directly inserted in an SFP/SFP+ slot of an IP/MPLS router) could be also considered as possible optical devices in the METRO-HAUL optical network.

The O-NE Controller coordinates the OIEs. Specifically, each O-NE is presented to the METRO-HAUL optical controller as a single YANG modelled device but can be implemented using two approaches: (i) O-NE coordinates the included OIEs using proprietary mechanisms or; (ii) each OIE can be provided with an OIE agent using a direct NETCONF connection (and optionally a telemetry connection) with the parent O-NE agent, see Figure 10. Specifically, Figure 10 illustrates (on the left side) a generic O-NE including a number of OIEs each one provided with a local agent connected to the southbound interface of the O-NE agent. With this latter solution, the O-NE agent receives (on its southbound) the YANG model of each connected OIE, and consolidates the models in a single model to be exposed to the METRO-HAUL optical controller.

From a research perspective it is worth to investigate a further hierarchical level of disaggregation where an OIE is further disaggregated in Optical Sub-Systems (OSTs). Every OST can be provided with an OST Agent with a NETCONF connection (and optionally a telemetry connection) to the parent OIE Controller. The following OSTs have already been identified:

- Switching elements (e.g., WSS, BV-WSS)
- Add/drop module components (e.g., splitter/coupler, AWG)
- Monitors (e.g., OSA, OTDR)
- Optical amplifier (e.g., EDFA line/pre/booster)
- VOA

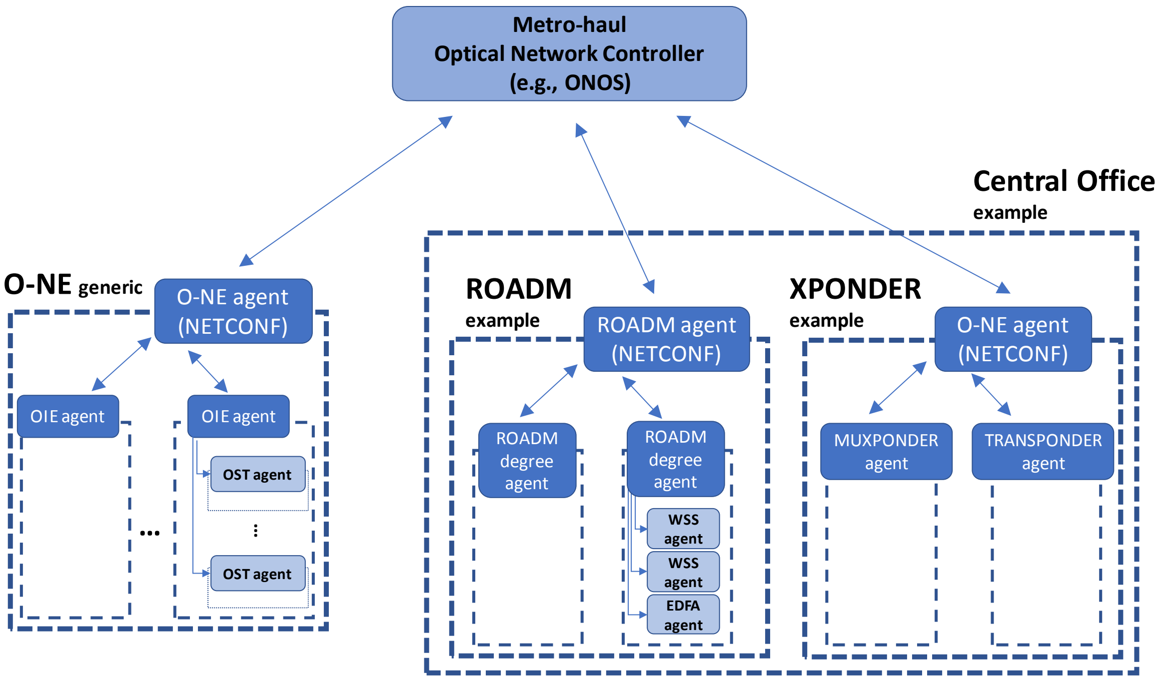


Figure 10. O-NE, O-IE and OST hierarchy.

Specifically, Figure 10 (on the right side) illustrates a Central Office example including two O-NEs: a ROADM and an XPONDER. The ROADM O-NE is composed of two OIEs (i.e., two ROADM degrees). In turn, the first ROADM degree manages included sub-systems using proprietary mechanisms, whereas the second ROADM degree uses a NETCONF connection to each OST agent. As described for the O-NE agent, also the OIE agent performs the consolidation of the OSF YANG models.

As mentioned above, all the NETCONF communications among agents (i.e., O-NEs, OIEs and OSTs) utilize the proper YANG models. The following YANG model references have been identified:

- OpenROADM as general reference for ROADM devices (i.e., not including all details)
- OpenConfig as general reference for XPONDER devices (i.e., not including all details)
- IETF, OpenROADM and OpenConfig for specific augmented versions of YANG models

Extensions to these models will be provided to improve the model definitions as well as to encompass M-H innovative solutions.

Finally, it is worth noting that in the initial phases of the project and in such demonstration scenarios where total number of devices is limited (not causing scalability issue at the controller) OIE and OST agents may be allowed to directly communicate with the METRO-HAUL optical network controller. This intermediate approach is mainly intended to face the implementation with a gradual approach, where enabling actual configuration of devices is the first step, and the consolidation of the YANG models in the proposed hierarchy is the final step.

Let is note that this of O-NE elements is just preliminary work and reported here for illustrative purposes. METRO-HAUL also focuses significantly on filter-less nodes and degree 2 low cost ROADMs, with additional constraints and technology limitations. O-NE and OIE categories can be extended, including ND-ROADM to include any form of Reconfigurable ROADM irrespectively of CDC and so on; F-ROADM for fix ROADM, etc.

The complete analysis and description is provided in [D31]. From the point of view of the SDN control, the functional capabilities of each O-NE are described in its associated data model(s). We can thus refer to O-NE and O-IE without loss of generality, without excluding different deployment models and disaggregation choices.

Likewise, the optical layer can be subject to further refinements, including the distinction between the A-DWDM (OLS) and DtoWDM layers [D31] in all aspects including horizontal and vertical interoperability and type of services offered. The SDN Control can also, in turn, become hierarchical with dedicate controllers to specific subsystems (e.g. orchestrating and controlling different layers as in a sub-controller for the OLS.

2.3 Packet Switched Networks

At each geographic location or Central-Office (CO), there is a Layer 2/Layer 3 packet switched network that aggregates traffic coming from access and aggregation networks (such as PON) and which provides connectivity to functions and applications running in local data-centers.

Unless stated otherwise, there are little assumptions on the specific topology of such L2/L3 networks. Initial designs will consider common and existing practice such as leaf-and-spine topologies and related fat-tree based topologies. The actual topology of the network will also constrain what different control plane architectures can be deployed, and it is part of the METRO-HAUL project to specify which switches are under the control of the site or location VIM and which switches are part of the transport network.

For those infrastructure elements that in METRO-HAUL are decided to be part of the location Layer 2/Layer 3 packet switched network, the work of the BBF CloudCO can be considered to be leveraged since, from the description in Section 2.1.1.1, it can be noted that the assumptions regarding the Data Plane infrastructure for the packet switched domain are in principle very much aligned between both architectures. At the same time, since many of the interfaces of the control architecture in CloudCO are still work in progress, there is an opportunity to take contributions from METRO-HAUL to the BBF CloudCO applicable Working Texts.

2.4 Active Monitoring

As part of the monitoring and telemetry system, active probes working at 10 and 100 Gbps will be developed within METRO-HAUL. Such probes will enable the characterization of network paths and the identification of capacity bottlenecks. Active probes inject traffic to the network and measure such traffic at the receiver. Probes will be able to provide at least one the following measurements:

Table 1. Measurements provided by the active network probe to be developed.

Measurement	Units	Meaning
Network path capacity	Bits per second, up to 100 Gbps speed	The capacity in the path from the source to the destination. It can include several links.
Delay	Seconds (ns granularity)	The delay from a source to a destination, since the packet/frame is sent until it is received. RTT (Round Trip Time) can be measured, and also OWD (One Way Delay) if there is GPS synchronization.
Jitter	Seconds (ns granularity)	The delay variation. Several definitions exist, we can adapt to them.
Packet/frame loss	Packets percentage	Fraction of packets that are not received from those that are transmitted at the sender. Accuracy will depend on the number of packets that is sent. For instance, to measure a 1% of packet loss it will be necessary to send 1000 packets.

To perform all these measurements, we will follow the packet train technique [Rui16]. A group of *N* packets is sent back-to-back, at maximum speed, from a sender to a receiver. The average dispersion of the *N* packets is used to calculate the network path capacity. Additionally, delay, jitter and packet loss rate may also be estimated by including timestamps and sequence numbers on the packets. This technique is based on flooding the network with *N* packet, but, since the number of packets per second required is negligible compared to the packets per second at 100 Gbps, the technique interference with the rest of the traffic is very low. This technique will be implemented in a high-end Field ProGrammable ASIC (FPGA)that allows a high precision in the measurement, and the accuracy will depend on the quality of the synchronization. The following settings can be done:

Table 2. Measurements parameters to be set in the active network probe to be developed by Naudit.

Parameter	Units	Meaning

Parameter	Units	Meaning
Frame length	Bytes	For each frame in the packet train, the number of bytes it contains, excluding preamble and interframe gap.
Train length	Packets	Number of packets that are sent, back-to-back, in the packet train.
Source and destination addresses and ports	IP addresses (32 bits) and UDP ports (16 bits) used for the packets	The packet is sent from a probe to another probe, with different addresses and ports.

The hardware design in the FPGA will have several modules. An ARP module will be implemented to fulfil MAC request and reply. A packet filter will also be implemented to discard unnecessary packets. The core of the system will be split into three modules: the packet generator, time-stamping and the computation of the measurement. Finally, an exporter will provide the result to the user of the active probe.

Additionally, the packet train technique can be complemented with a Bit Error Rate Test (BERT), which consists in sending a well-known pattern of bits, in this case in the payload of the packet. If the frame has not been previously discarded based on the CRC, the receiver counts the differences between the pattern and the received bits, with this count a bit error rate can be computed. If the frame has been discarded, only a Packet Error Rate (PER) can be measured.

Another well know active probe is a traffic replay, which is based on sending a real or synthetic traffic. A trace is stored in the memory, after is read and sent to the network. This probe is flexible, because the trace can contain as much flows as the memory capacity can store. A traffic replay can be built both in an FPGA or commodity hardware. The main challenge of building a traffic player in an FPGA is the limited available memory, which can store just few seconds. However, the maximum transmission speed is guaranteed, transmitting the traffic at line rate. Stressing a network with specific flows could arise corner cases that without this probe will be arduous to debug.

2.5 Computing and Storage Infrastructure as NFVI

At each location or CO, there is a variable number of storage computing and virtualization servers, which constitute part of the NFVI of the METRO-HAUL infrastructure. The set of servers in a given geographical location are managed by a dedicated Virtual Infrastructure Manager (VIM), within a unique administrative domain that spans the whole METRO-HAUL. Such NFVI is commonly referred to as a local DC.

For the purposes of METRO-HAUL, we assume a variable number of compute nodes or servers (N) per location, each having a certain amount of RAM (X Gb) and CPU cores (M vCPUs) that can be dynamically allocated to support virtual machine instances. The exact dimensioning of each location is not specified.

2.6 Passive Optical Networks

Finally, the integration of Passive Optical Networks (PON) is part of the METRO-HAUL COM system. The network orchestrator, as shown in the next session will coordinate the configuration of the PON during the provisioning of connectivity and QoS services.

The unified control of the PONs is realized through the adoption of an abstraction scheme which represents the PON as a legacy OF/NETCONF-enabled switch, as depicted in Figure 11(a). Under this abstraction, the physical ports a PON is using to interconnect to the network infrastructure i.e. the upstream port of OLT (p1) and the downstream ports of ONUs (p2-p5), are portrayed as the logical ports of a switch. The proposed abstraction scheme is offering two advantages: a) it is hiding the PON specific details related to forwarding and control/management operations; b) vendor-specific configuration commands are automatically translated and executed by the PON components. Therefore, the forwarding command messages are exchanged by means of the standard OF protocol, while (re)configuration is realized by means of the NETCONF protocol.

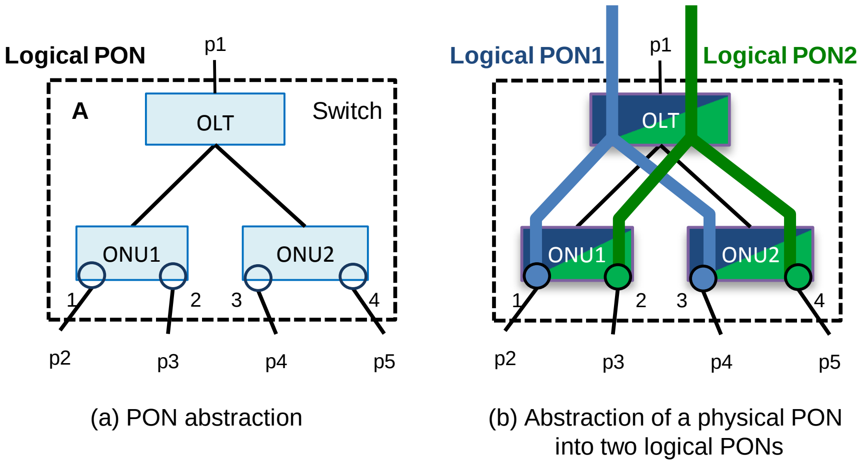


Figure 11. Schematic representation of PON abstraction

Under this scheme, the PON is operated as an OF-controlled L2 switch. The novelty of this approach is that it is a truly plug-and-play solution in the sense that standard protocols are used; there is no need for OF protocol extensions or upgrades to the already deployed SDN control/management tools. Operators may simply connect the OF-enabled PON infrastructure into their network and integrate the provided YANG model into the NETCONF infrastructure. Moreover, this approach allows to capitalize on the increased penetration of OF/NETCONF in other network segments (WAN/Metro).

Further, this scheme is extended to support PON slicing since several OF-controlled switches can be abstracted over the same physical GPON. Under this mode, each of these switches can dynamically control a subset of the GPON resources. Thus, a PON is partitioned into several logical PONs where each one of them consists of a logical-OLT (having a slice of the physical OLT) and from several logical-ONUs that may include any combination of ONUs, ONU ports or even ONU port slices. The objective of this slicing approach is to come up with an infrastructure ready to support multi-tenancy in the context of Infrastructure-as-a-Service. This is shown in Figure 11(b), where the logical PON of Figure 11(a) is partitioned in two logical PONs/SDN-controlled switches. Under such partitioning each slice may deploy different configurations and policies as the two slices are assigned to two different operators. As an example, the logical PON 1 assigned to operator A, includes a slice of the upstream OLT port (p1), and the ports p2 and p4 of two ONUs, while operator's B logical PON consists of a slice of the upstream OLT port (p1) and the ports p3 and p5 of two ONUs.

3 COM High Level Architecture

As stated, the COM system relies on recent advances in SDN and NFV, trying to adapt existing frameworks to the specifics of the project, that is, the applicability to Metropolitan networks, the deployment of disaggregated optical networks, the importance of monitoring, telemetry and data analytics and the interest of externalizing the algorithmic aspects (network optimization, function placement, resource allocation) to dedicated subsystems.

3.1 Related Initiatives and projects

3.1.1 ETSI NFV

ETSI NFV architecture considers that, for multi-site network services, MANO NFVO will request the required resources to the WAN network by means of a WAN Infrastructure Manager (WIM) functional module, as it is described in ETSI GS NFV-MAN 001 ("Management and Orchestration") and shown in Figure 12.

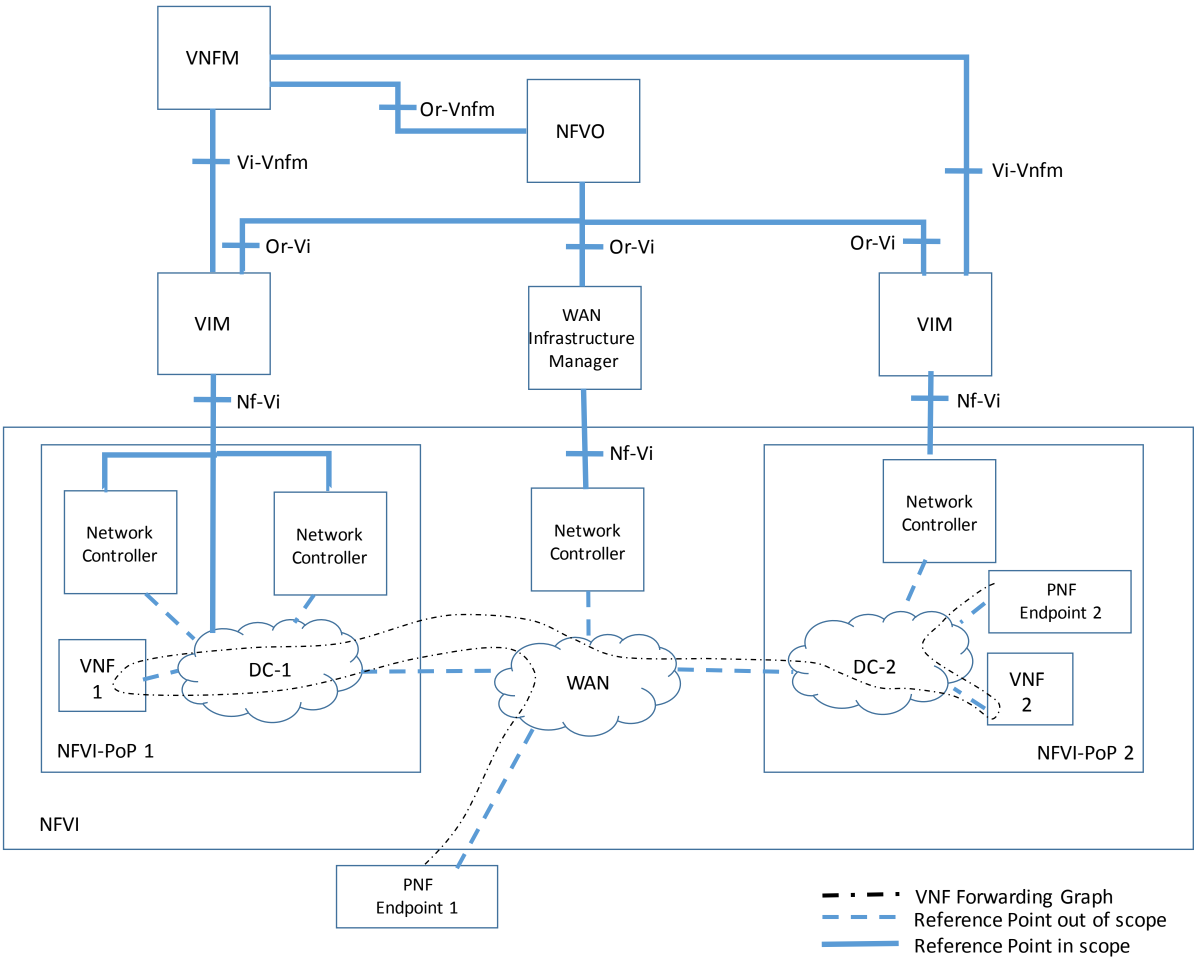


Figure 12.WIM in ETSI MANO architecture.

In order to advance the definition of the interactions and the modelling required between the NFVO and the WIM, ETSI NFV published ETSI GR NFV-IFA 022 [IFA022] that collects a set of use cases and derived recommendations for further normative work in ETSI NFV regarding multi-site services.

These recommendations are to be elaborated in ETSI NFV IFA Release 3 specifications as part of Feature 10 ("Management and Connectivity Multi-Site Services"), having an impact on existing ETSI NFV standards [IFA005, IFA007, IFA010, IFA011, IFA013, IFA014, IFA022] as well as triggering work on a new GS document NFV [IFA032] ("Interface and Information Model Specification for Multi-Site Connectivity Services").

ETSI work on NFV IFA Rel 3 Feature 10 is relevant for METRO-HAUL architecture, as network services deployed on top of the METRO-HAUL architecture in the packet layer will very likely involve VNFs deployed in different METRO-HAUL nodes and as such can be considered Multi-site Services. Work on a standard interface between the NFVO and the WIM will allow for a generalized Or-Wi interface between them that decouples the architecture from the use of the specific NB API of a specific Transport Network SDN controller or tied to a specific Transport Network technology. Since Feature 10 is still work in progress there is an opportunity for METRO-HAUL contributions addressing this ETSI work.

3.1.2 BBF Cloud CO Control and Orchestration

Regarding control and orchestration a CloudCO domain will look like Figure 13 at the functional level

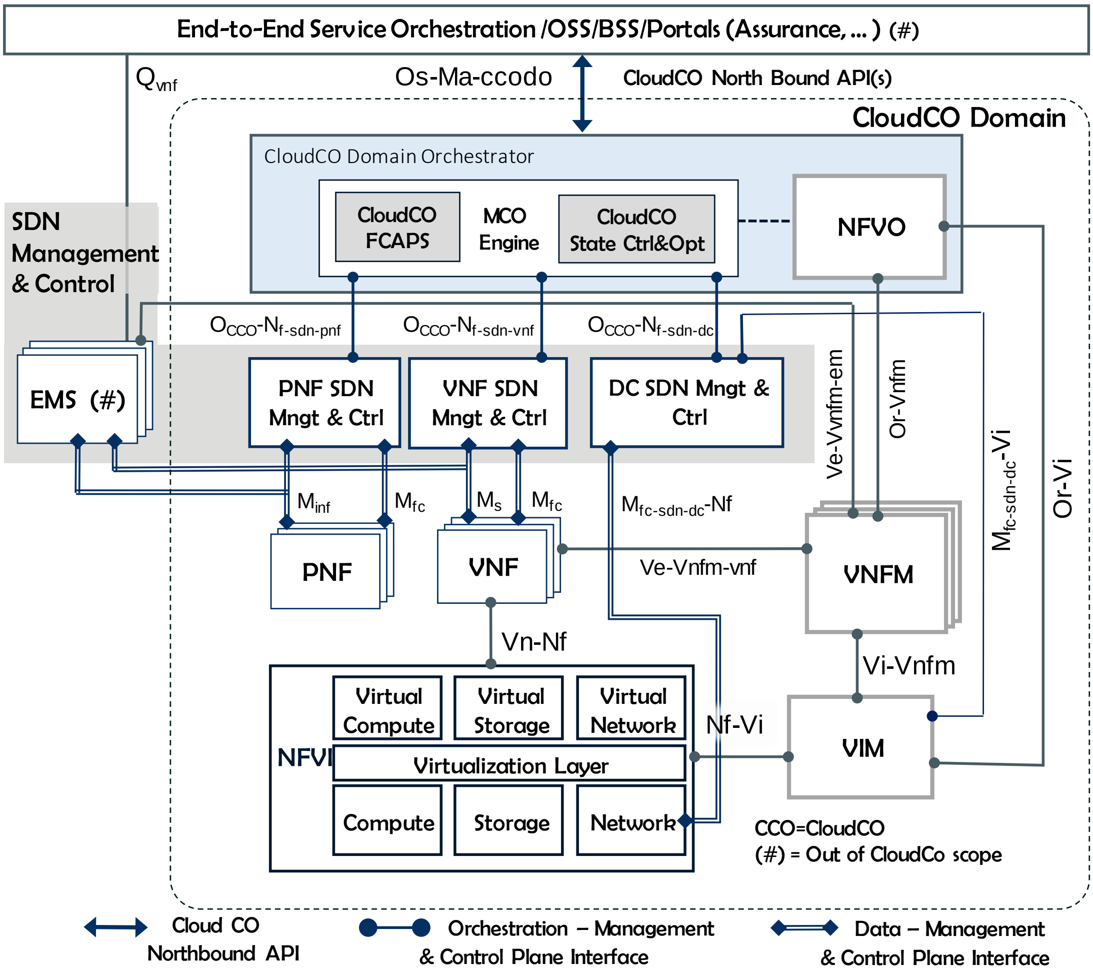


Figure 13. CloudCO reference architecture (figure 13 from BBF TR-384).

PNFs in the CloudCO domain include the Access I/O PNFs, Network I/O PNFs and any other PNF that is deployed inside the CloudCO and used in the deployed network services making use of the CloudCO infrastructure. The DC SDN Manager & Controller directly accesses the NFVI networking resources to implement functions (e.g., L3 routes in the switch fabric) that the VIM is not supposed to do.

The CloudCO Domain orchestrator will not be present in all the macro-nodes in a domain, being this CCO Domain orchestrator deployed centralized typically in one or some (for redundancy purposes) of the macro-nodes in a domain, as shown in Figure 14. Each macro-node in a domain will be connected to a WAN network that provides the external connectivity to the Network Functions deployed in that macro-node.

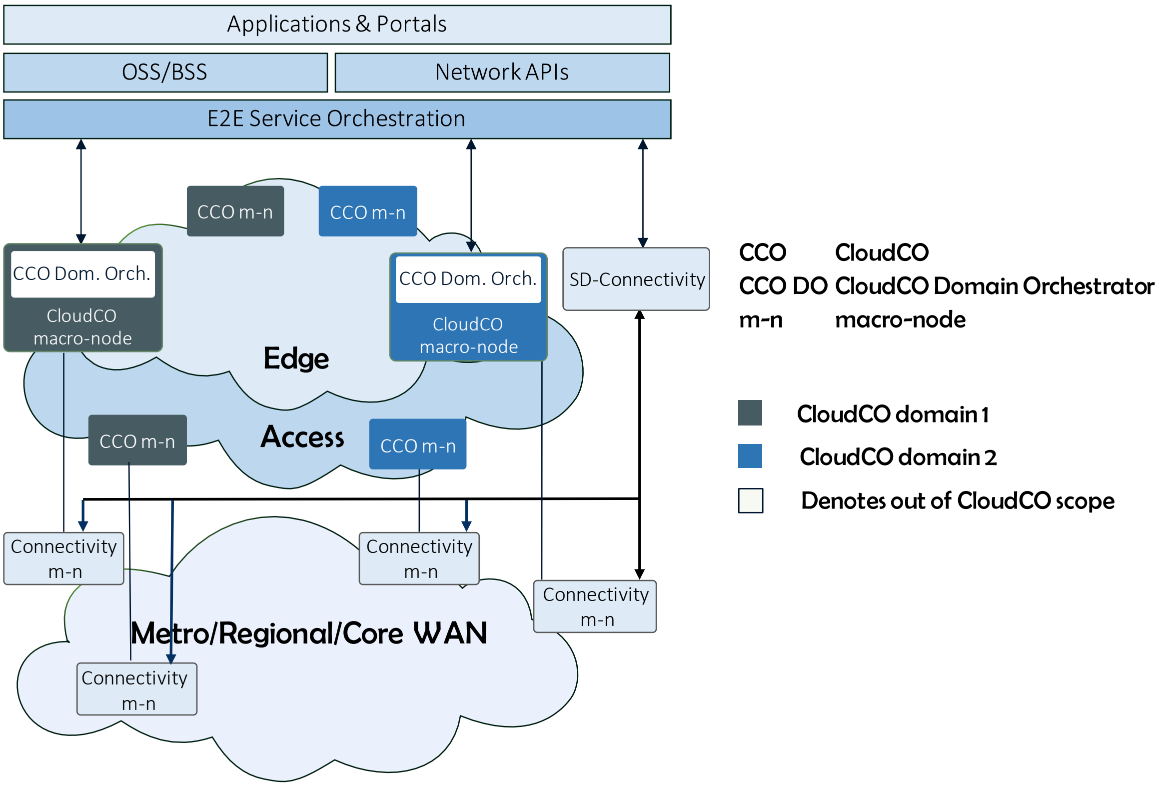


Figure 14. Example of deployment of CloudCO Domain Orchestrators (figure 9 from BBF TR-384).

However, control of the WAN connectivity of the different CloudCO Macro-nodes is out of the CloudCO scope. For that purpose, the CloudCO relies on an external control SD-Connectivity module shown in Figure 14 that is managed from an out-of-scope E2E Service Orchestrator (other than the fact that it would be a client of the CCO Northbound API). It can be inferred then that the Network I/O functional module in the CloudCO architecture only covers the adaptation (e.g. physical media adaptation or VLAN delineation) between the Packet Switched environment of the CloudCO and the WAN environment, but is not part of the WAN environment.

3.1.3 CORD Control Architecture

Figure 15 shows the CORD control architecture. It encompasses ONOS for managing the POD switching fabric, OpenStack for the creation of Virtual Machines and Virtual Networks and XOS as the orchestration layer.

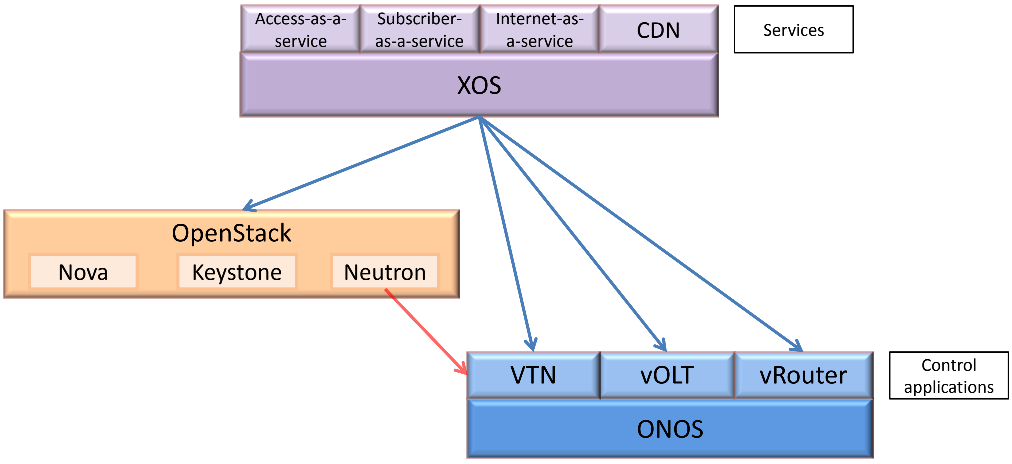


Figure 15. CORD control architecture

ONOS plays two roles in CORD. It both interconnects VMs (this includes implementing VNs and managing flows across the switching fabric) and provides a platform for hosting control programs that implement CORD services. Two examples of the latter are the vOLT and the vRouter. The vOLT implements authentication on a per-subscriber basis, establishes and manages VLANs connecting the subscriber's devices to the central office switching fabric and manages other control plane functions of the OLT. The vRouter provides each subscriber with their own "private virtual router," where the underlying fabric can be viewed as a distributed router. The vRouter control program then routes between the attached per-subscriber subnets and also peers with legacy routers (e.g., advertising BGP routes). Another ONOS application, the VTN, installs flow rules in the OvS running on each server to implement a service composition overlay, using VxLAN tunnels and a custom OvS pipeline. For this purpose, it coordinates with Neutron, the OpenStack module managing VNs, by means of a dedicated ML2 plug-in. Actually, ONOS is deployed as two instances, the first, named onos-fabric, controlling the underlay network made of the switching fabric and the second, named onos-cord, controlling the OpenvSwitch instances running in the compute-nodes realizing the overlay network. The reason for this is simply an implementation issue that allows separation of specific applications like VTN, SegmentRouting, vRouter according to their roles in controlling the overlay or the underlay network.

OpenStack is responsible for creating and provisioning Virtual Machines and Virtual Networks. The CORD implementation supports services running in VMs, in containers running directly on bare metal and in containers nested inside VMs. XOS is a framework for assembling and composing services. It unifies infrastructure services (provided by OpenStack), control plane services (provided by ONOS) and any data plane or cloud service (running in VMs or containers), adopting the Everything-as-a-Service (XaaS) organizing principle. It uses a design philosophy similar to Unix: both are organized around a single cohesive idea—everything is a file in Unix and everything is a service in XOS. Both also aim to have a minimal core (kernel) and can be extended to include new functionalities.

It's worth noting that the described CORD control architecture applies to a single POD covering all control aspects of the R-CORD and M-CORD scenarios. For the E-CORD use case, however, coordination among the PODs offering connectivity to the subscriber sites is necessary. For this purpose, the current E-CORD architecture employs another ONOS instance (named ONOS-global in Figure 16) interfacing with both ONOS-fabric and ONOS-cord of all the involved PODs; ONOS-global has also control over the SDN controller of the transport network for intra-POD connections management.

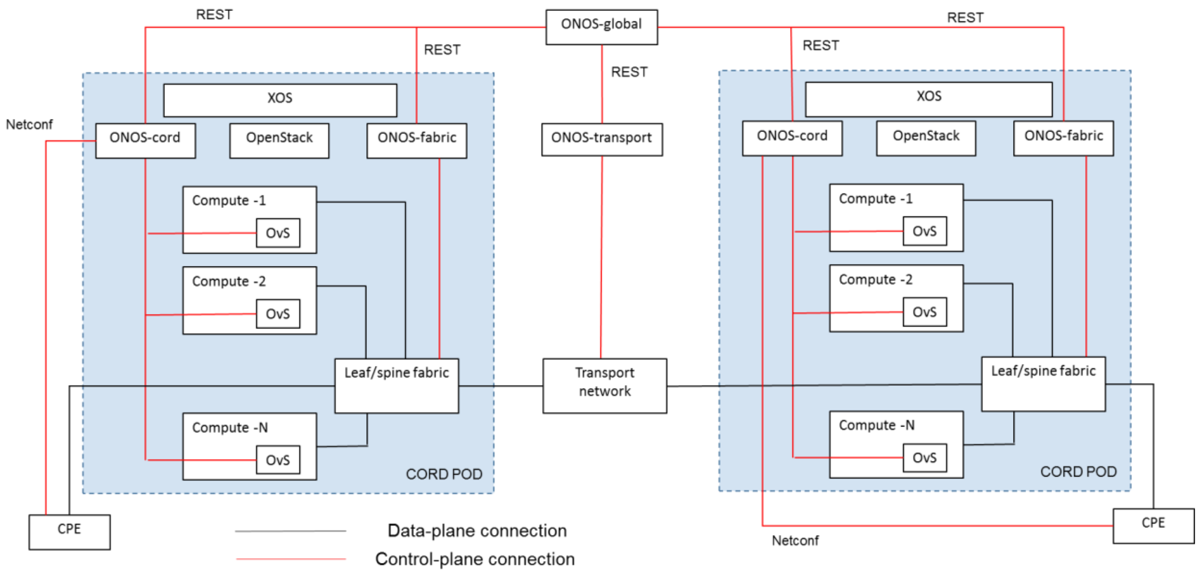


Figure 16. E-CORD control overview

3.2 Design considerations and initial requirements

The ETSI NFV MANO [NFVArch] provides a suitable framework but, as shown in Figure 17, it needs to be extended in multiple ways. The main limitations, as show, can be briefly summarized as follows:

- First, the main focus is the network service, which does not cover all use cases (e.g. METRO-HAUL wants to support a limited form of network slicing). In other words, the main NBI of the NFVO deals with the deployment of VNFs.
- The role of the WIM is still not fully specified. The entity appears in ETSI normative documents, but not the relevant interfaces or how this entity communicates with other entities. Scenarios with more than one WIM are not fully explored. There are no proof-of-concepts with a fully integrated WIM.
- It is not easy to integrate optical network control, since the ETSI NFV MANO is focused on L2/L3 management.

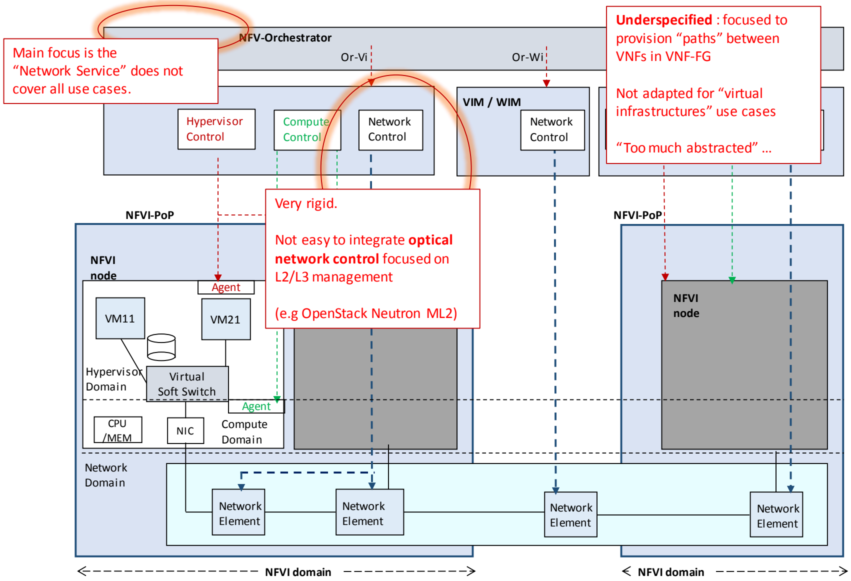


Figure 17. Key requirements to adopt ETSI NFV MANO for METRO-HAUL COM framework

The design of the METRO-HAUL COM has taken into account existing approaches and architectures, with a few compromises in view of the target scenarios and use cases. METRO-HAUL is aligned with ETSI and related standards and relies on the following (see Figure 18 and Figure 19):

- A single, centralized MANO system, with a centralized NVFO, for the deployment of network services and slices. We do not consider in scope the deployment scenarios with more than one NFVO (as opposed to CORD XOS or Cloud CO local controller).
- The single NFVO becomes the entry point for the system. The slicing system is deployed on top of the single, centralized NFVO, and on top of it, using its APIs and exported interfaces.
- We do consider a single VIM per location (METRO-HAUL node) given that VNFs logic (e.g. ISO files with the VNF software) needs to be stored locally at each geographic location. Having a single centralized VIM could raise issues on scalability, and delays / latencies in service deployments, for example, when having to transfer large volumes of data.
- A single centralized control of the disaggregated optical transport network.
- The centralized control, including centralized MANO, centralized SDN controller (at least, from the point of view of the parent controller, and from the point of view of the SDN controller for the optical network) makes it easier to integrate with a hierarchical monitoring and data analytics system as well as with a network planning system.

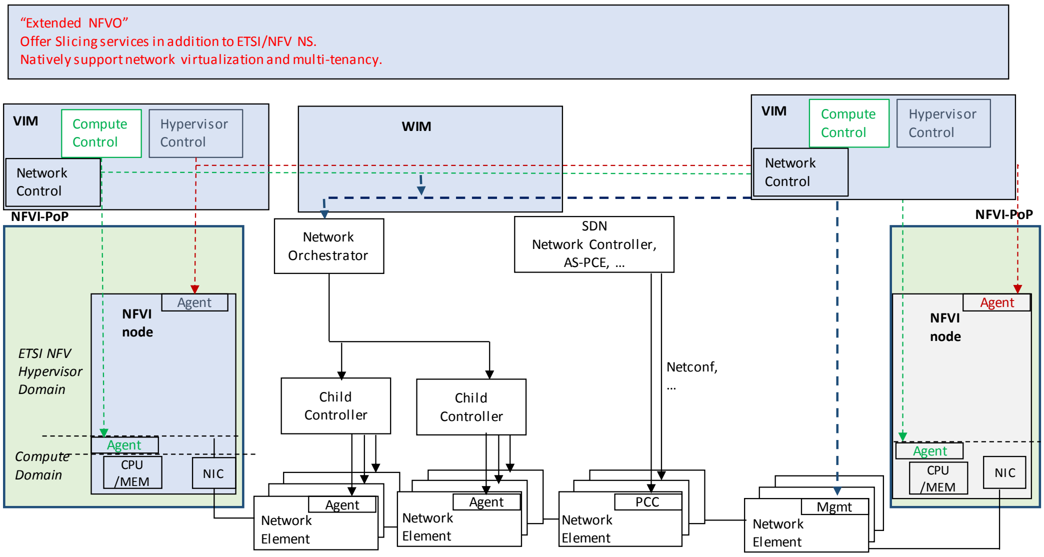


Figure 18. METRO-HAUL COM high level design with centralized MANO

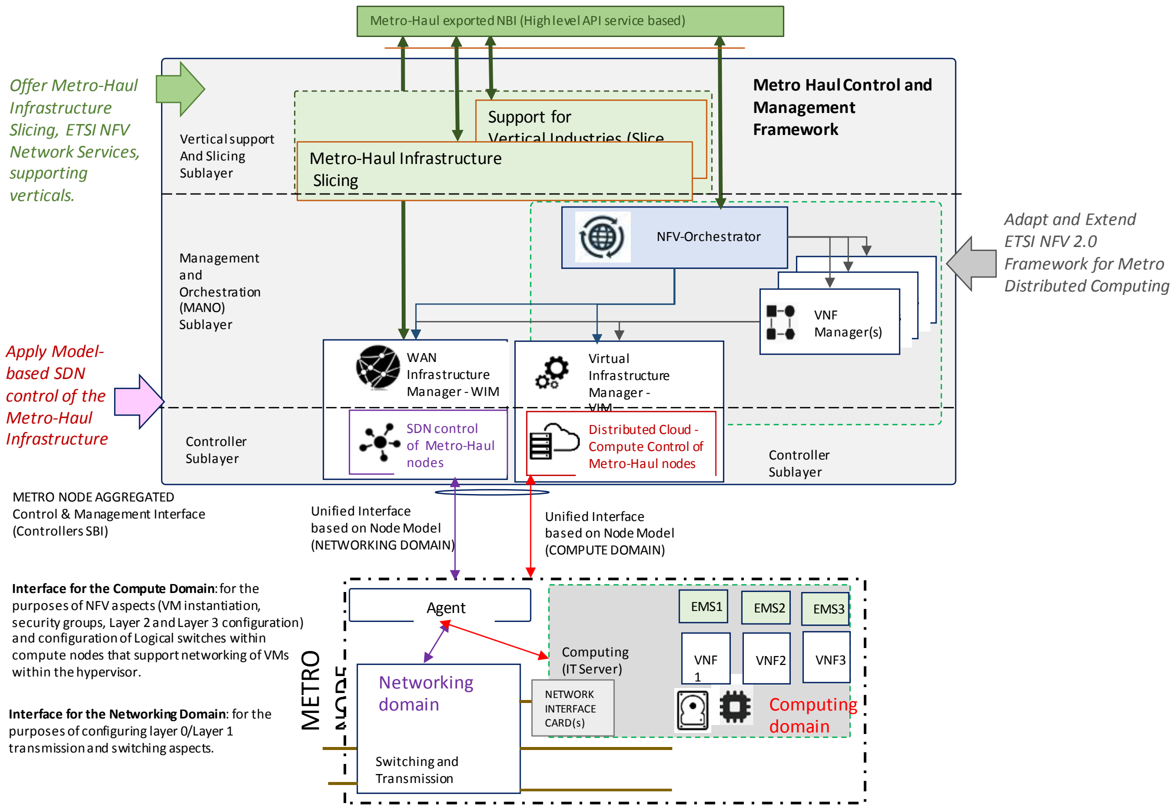


Figure 19. METRO-HAUL COM high level design (focusing on the COM and NFVO)

The different elements of the COM are detailed in Section 3.3.

3.2.1 Relationship between CORD and ETSI NFV architecture

One point about the CORD control architecture is how it relates to the ETSI NFV architecture.

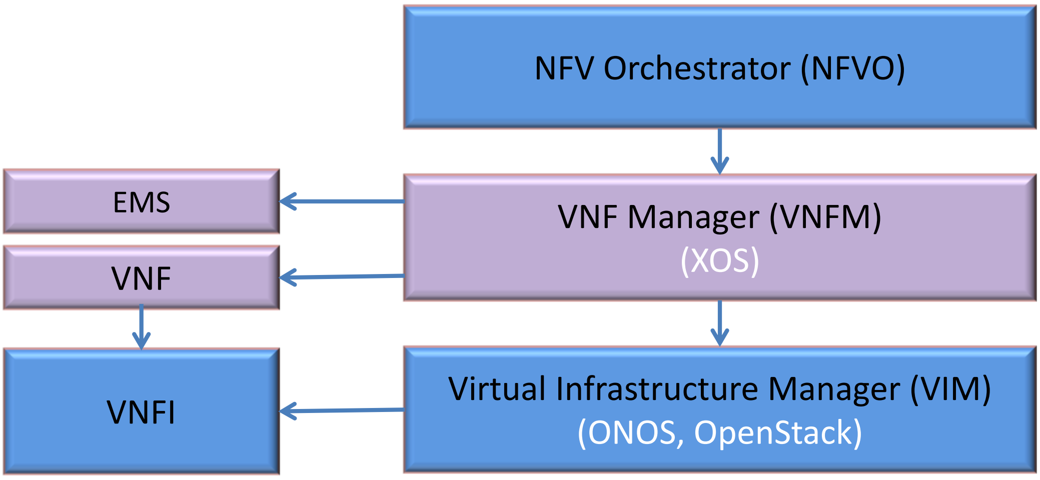


Figure 20: Mapping of CORD onto the ETSI NFV architecture

Figure 20 gives a simplified depiction of the ETSI NFV architecture, annotated with the three key software components of CORD (XOS, OpenStack, and ONOS). It's quite straightforward that the Virtual Infrastructure Manager (VIM) corresponds to OpenStack and ONOS. In this framework ONOS plays a dual role: it both creates virtual networks on the underlying switching fabric and it hosts SDN based services, i.e. all those services implemented by a control application running in it. These services are not shown in Figure 20, but their service controllers (vOLT, vBNG, …) would be examples of Element Managers (EM). However, in CORD, some VNF may be a combination of "program-based VNFs" running in virtual machines and "Openflow-based VNFs" running in the switching fabric. An example for this could be the vOLT, where VLAN tags are removed by instructing ONOS to push an appropriate flow rule to the switching fabric while other packet processing (admission control, traffic shaping, …) is performed by a virtual machine instantiated by OpenStack. XOS coordinates a collection of service controllers and, in this sense, it plays the role of a VNF Manager (VNFM). Interestingly, having the VNFM control both a set of EMs and the VIM is one of the main innovations XOS brings to the NFV architecture: it defines a single locus of control for coordinating both the infrastructure that hosts a VNF and the controller that manages a VNF.

3.3 METRO-HAUL COM Core Platform

3.3.1 Control of the Optical Layer

Traditional optical transport networks are proprietary, integrated and closed, where the entire transport network acts as a single vendor managed domain. It can export high-level interfaces and open NBI, yet the internal details and interfaces are hidden from the operator.

Disaggregation of optical networks refers to a deployment model of optical systems, by composing and assembling open, available components, devices and sub-systems. This disaggregation can be partial or total (down to each of the optical components) and is driven by multiple factors, notably, the mismatch between the needs of operators and the ability to deliver adapted solutions by vendors; the increase in hardware commoditization; the different rate of innovation for different components; the promised acceleration on the deployment of services and the consequent reduction in operational and capacity expenses.

Disaggregation imposes new challenges in the control and management system. It is clearly a use case for open interfaces exporting programmability, and the increase of unified and systematic information and data modelling activities is a crucial step in this regard. However, optical networks are particularly challenging to model due to the lack of agreed-upon hardware models, and this is critical for the development of an interoperable ecosystem around disaggregated hardware.

In the next sections, we briefly detail the main deployment models. For a detailed overview, see [Ric18].

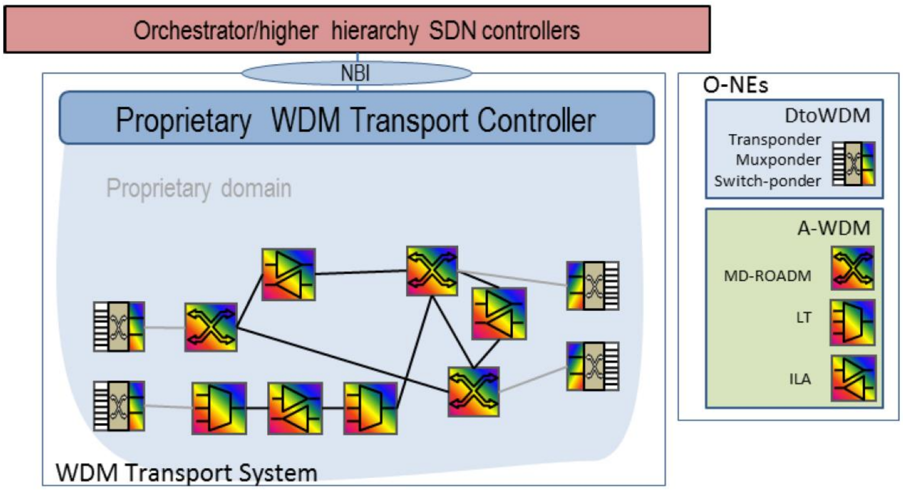


Figure 21. Fully aggregated WDM transport System [Ric18].

3.3.1.1 Aggregated optical domains

This is the current evolution of mono-vendor WDM-Sys, with the introduction of an open NBI for control and managing the whole network island in a more flexible way (Figure 21 : Controller and O-NEs are provide by the same vendor. A black box lifecycle model is implied. Only the NBI needs to be specified/standardized). The optical system lifecycle management is responsibility of the system vendor in the pure black box approach. System vendor provides both a proprietary WDM transport controller and all the O-NEs. SBI to O-NEs may run proprietary protocols with tailored equalization algorithms. Typically, these networks support digital transport services (e.g. Ethernet or OTN from client-side ports of TPs): mapping of digital clients and activation of network media channels is under the control of the proprietary domain. NBI translates aggregated information of the underlying optical network in a simplified abstract model to be used by higher order controllers or orchestrators. Standardization of this NBI could assist the integration between the controller and the orchestrator.

3.3.1.2 Partial Disaggregation: Open Line System and Multi-Vendor Transponders

In this approach (Figure 22), the disaggregation applies to the DtoWDM layer (i.e. to TPs) whose lifecycle is decoupled from that of a mono-vendor and proprietary A-WDM layer. The A-WDM layer remains a proprietary black box analogue transport system supporting Optical Channels from external TPs as client signals. Thus, to this Open Line System (OLS) applies all the considerations made in the previous paragraph; the term 'Open' refers to the fact that it is open to be used by any signal which follows a given behavior, specified by the Single Wavelength Interfaces (SWI). An OLS-NBI API is needed to configure and report events from the OLS. Note in the figure, A) is an Open

Line System as part of a partial disaggregated WDM transport system: OLS and controller are from a single vendor (1-2); TPs may be in pair form the same supplier (3) or mixed (4); the WDM Transport Controller interfaces directly with TPs (5) and through an NBI (7) to the OLS. Single Wavelength Interface (SWI) need to be standardized (6); B) Alternative partial disaggregated WDM transport system: OLS and WDM controller are proprietary from a single vendor (1-2); TPs may be in pair form the same supplier (3) or mixed (4); the proprietary WDM Transport Controller interfaces directly with TPs with standard SBI (5). Single Wavelength Interface (SWI) (6) and SBI (5) need to be standardized.

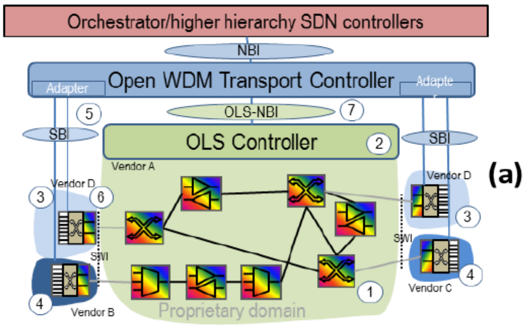


Figure 22. A) An Open Line System as part of a partial disaggregated WDM transport system. B) Alternative partial disaggregated WDM transport system [Ric18].

3.3.1.3 Full Disaggregation: Multi-Vendor Optical Network elements

The fully disaggregated WDM transport system is shown in **Error! Reference source not found.**. In this approach, the involvement of the operators in the WDM-Sys lifecycle is strong, certainly not limited to vertical integration of control and management SW. Indeed, O-NEs from both the A-WDM and DtoWDM layers are potentially purchased from different vendors, leaving interworking at the control and data plane to the system integrator. Therefore, most of the control intelligence is moved to the WDM controller (necessarily vendor agnostic) which becomes the most critical element of the whole chain, having to face also all the analogue transmission issues (equalization, transient suppression, etc.). In the figure, a fully disaggregated WDM transport system [Ric18] is shown: O-NEs can be from the same (1-2) or from different suppliers. No separation between DtoWDM and A-WDM layers exist. A standard SBI (5) is needed to simplify the direct control of the whole WDM-Sys by the controller (4). Both Single Wavelength (6) and Multi Wavelength Interfaces (7) need standardization.

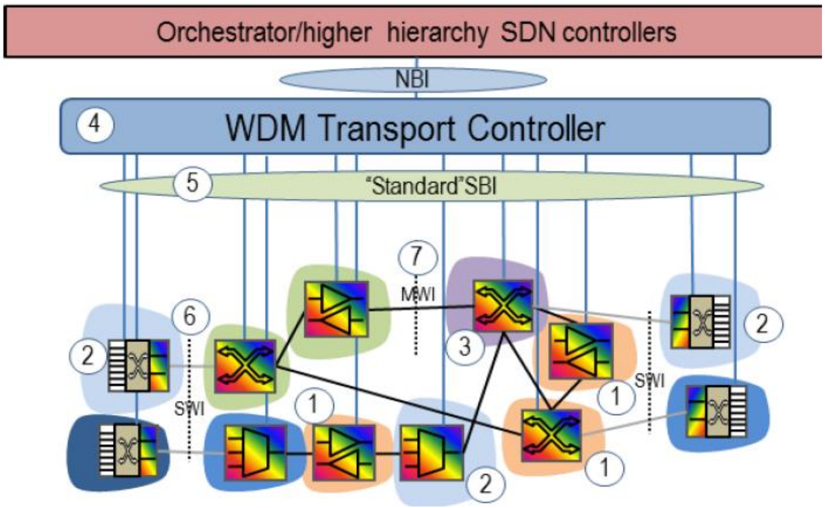


Figure 23. Fully disaggregated WDM transport system [Ric18].

Regardless of the actual deployments, from the rest of the COM system we can consider the SDN control of the optical layer as a single component, provided the right abstractions and interfaces. In particular, it is useful to consider the METRO-HAUL architecture for the control of the disaggregated Optical Network relies on a centralized SDN controller and the use of YANG/NETCONF as data modelling language and protocol (Figure 24), without excluding deployments combining the aforementioned choices.

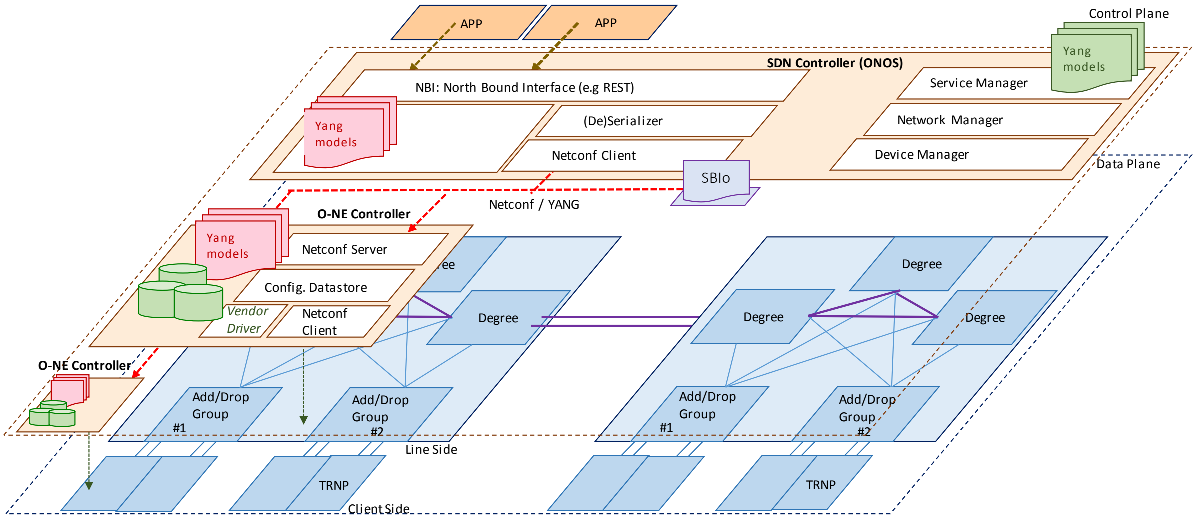


Figure 24. NETCONF / YANG control of the partially disaggregated optical network.

Element	Functions
SDN Network Controller	• Responsible for the centralized control of the optical infrastructure. • Exports a North Bound Interface to applications, to instantiate services. The services are described using YANG models. • Configures optical devices (Optical Network Elements) by using a NETCONF/YANG protocol and data models.

	• Provisions network connectivity in the optical layer between transponders (with optional recovery mechanisms)
Local node controller agent	• Implements a NETCONF Server, accessed by the SDN Network controller. • Configures individual components within the O-NE. • Notifies the SDN controller about events • Notifies the Monitoring and Data Analytics system.

In cooperation with WP3, we have closed MS4.2, "specification of abstracted HW models for remote programmability and configuration". As main takeaway message, three relevant worldwide open initiatives are considered of high relevance for METRO-HAUL: OpenROADM, OpenConfig and ODTN. In the definition of HW (device) models, METRO-HAUL will consider OpenROADM and OpenConfig to model the hardware devices. Extensions to these models will be provided to improve the model definitions as well as to encompass METRO-HAUL innovative solutions and the following implementation steps are identified: First phase, implement software components compliant with M4.2 Agreement (i.e., models) and second phase, enhance/add/complement models and implemented software components to include METRO-HAUL advanced models/solutions.

The baseline architecture foresees a single SDN controller for the whole optical domain. However, we are not excluding refinements of the architecture including deployment model with multiple co-ordinated controllers.

3.3.2 Control of the Packet Layer

The control of the packet layer is needed to provision services end-to-end involving e.g. instantiated VMs or NFVs or traffic coming from aggregation networks such as PONs. METRO-HAUL will rely on exiting controllers and frameworks for the packet layer. Existing approaches based on the OpenFlow protocol will be reused. Depending on partners' interest, the newer P4 protocol may be considered as an alternative to OpenFlow.

Element	Functions
SDN Network Controller	• Responsible for the METRO-HAUL Central Office control of the packet switching network. • Under the coordination of the network orchestrator **Remarks:** • Current VIMs such as OpenStack may delegate part of the interconnection of the VMs and VNFs to a dedicated underlying SDN control. It is still a work in progress to define the demarcation points and how this is mapped to a transport network. The initial design consideration is to cover the intra-DC network within the AMEN/MCEN nodes as a leaf-spine architecture under the responsibility of the VIM at the PoP. We can then have packet based transport networks under the control of a hierarchical SDN controller interconnecting AMEN/MCEN

	nodes.

3.3.3 Control of the PON

The architecture for the control of the PON network is schematically depicted in *Figure 25*. The main building blocks of the architecture is the Management Agent Software Framework (ConfD)[ConfD], the PON Configuration Agent (PCA), the PON Network Flow Agent (PNFA) and the SDN PON Controller which is based on Open Network Operating System (ONOS). The SDN PON Controller platform comprised a set of components as illustrated in Figure 25 and is the central management system/orchestrator of the PON network.

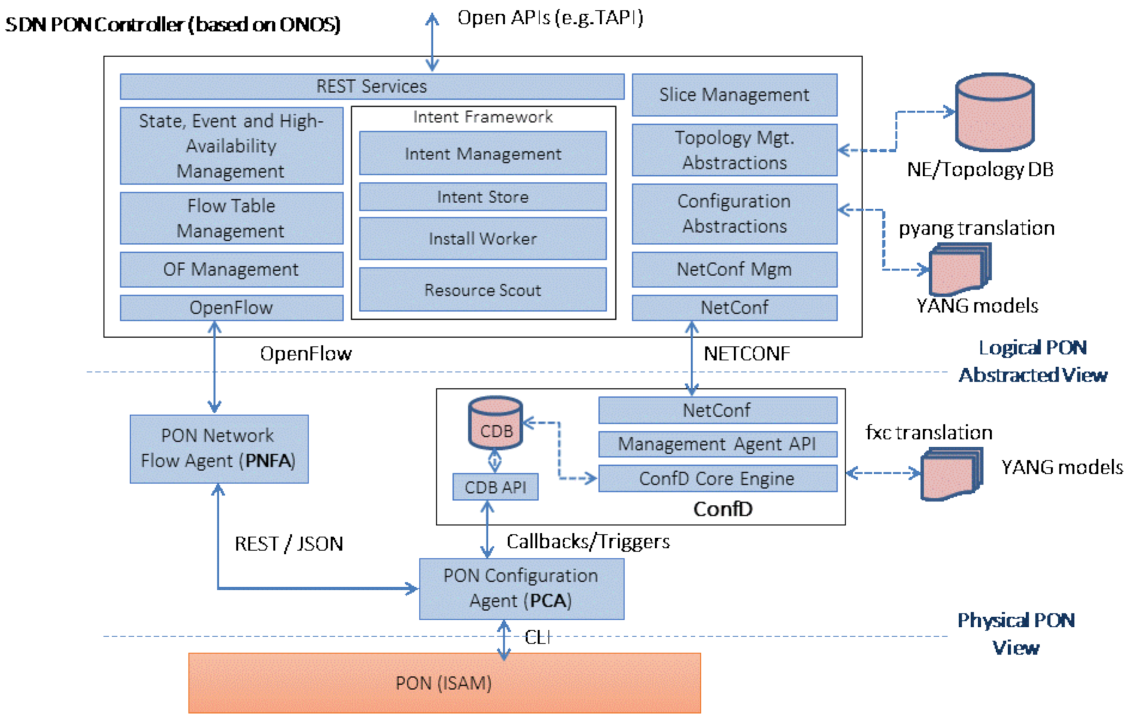


Figure 25. SDN Control of PON network

Element	Functions
SDN PON Controller	• Responsible for the METRO-HAUL control of the PON network • Under the coordination of the parent controller (Orchestrator) **Remarks:** • The SDN PON Controller will be based on ONOS. Required extensions to ONOS will be implemented in order to be integrated with the PNFA and PCA.
PON Configuration Agent (PCA)	• Communicates with ConfD via the CDB API • Realizes the requested configuration changes to the PON elements ○ when the configuration is modified, the CDB engine informs the PCA of the changes and the PCA generates the appropriate CLI

	commands and executes them directly on the corresponding PON elements. • Translates the OpenFlow requests from PNFA to the appropriate element mapping (e.g. VLAN to queue) and executes the appropriate commands.

PON Network Flow Agent (PNFA)	• Implements the flow management of the infrastructure.
	• Uses OpenFlow
	  ○ to receive OpenFlow flow requests from the OpenFlow controller of ONOS and to forward them to the PCA
	  ○ to inform PON Controller of the activation of a new switch (PON) and to establish the initial OpenFlow flows.

ConfD	• Retains the configuration status of the PON. • When configuration requests are arriving from the application layer via the North Bound Interface (NBI), the ConfD Core Engine Agent retrieves the data stored in CDB and updates the configuration, while through a CDB API informing the other entities in real time of configuration status changes. **Remarks:** • The abstraction of the PON as a L2 switch was based on YANG modelling and in particular on the legacy-switch model in which the GPON QoS queue models of Broadband Forum [BBF1] were incorporated. This integration allows to (re)configure the PON using the NETCONF protocol, with the configuration status stored into the Core Engine Database (CDB).

3.3.4 Network Orchestration

Transport networks are increasingly segmented in domains, e.g., to enhance scalability, due to confidentiality reasons or by virtue of having non-interoperable vendor islands. Regardless of the looseness of the term Domain (admitting multiple definitions, such as the set of elements as defined by management boundaries, vendor or technology islands, topology visibility or path computational responsibility), CP entities in multi-domain networks have inherent limited topology visibility outside a given domain and interoperability issues for cross-domain signalling. Exchange of topological information between domains is limited to the dissemination of reachability yielding sub-optimal choices and domain local optimality does not imply end-to-end optimality. In fully distributed models, such exchange of information takes place between inter-domain neighbours, which, in turn, inject part of the information to their respective domains.

The term Orchestration often appears when referring to control and management architectures, but there is only a rough consensus on its actual meaning and scope. The Open Networking Foundation (ONF) defines orchestration [ONFArch] as the selection of resources to satisfy service demands in an optimal way, where the available resources, the service demands and the optimization criteria are all subject to change. The orchestration function adjusts the state of the resources under its control to move toward that optimum. Within Networks Function Virtualization [NFVArch], the term refers to the coordination of the resources and networks needed to set up cloud-based services and applications, a process using a variety of virtualization software and industry standard hardware. In particular, Network Orchestration addresses the over-arching control across multiple heterogeneous domains (both in terms of control and data plane domains). It commonly relies on hierarchical control architectures, where the parent controller is referred to as the Orchestrator ensuring over-arching control relying on multi-domain abstracted topology and service databases, with each specific domain performs its own topology abstraction and control adaptation

Joint IT/Cloud and Network Orchestration is used to refer to the coordination of resources to deploy services and applications that require storage, computing and networking resources. The former is exemplified by the need to provision network connectivity services across heterogeneous domains (e.g. OpenFlow islands inter-connected by a GMPLS/PCE optical network) [Lop15]. The latter, by the provisioning of cloud services that require end-to-end Intra/Inter data center (DC) control and inter-connection of Virtual Machines (VMs) located in distributed data centers, in multiple geographically disperse sites or locations, where cloud locations are optimized for applications delivery [May16].

METRO-HAUL preliminary designs have considered the ACTN architecture [draft-ietf-actn]. The architecture for network orchestration is shown in Figure 26. To be considered in view of Task T4.3 is the fact that ACTN can facilitate virtual network operation via the creation of a single virtualized network or a seamless service. This supports operators in viewing and controlling different domains (at any dimension: applied technology, administrative zones, or vendor- specific technology islands) and presenting virtualized networks to their customers. Finally, regarding the interfaces, IETF models and or ONF TAPI candidates are being evaluated in parallel.

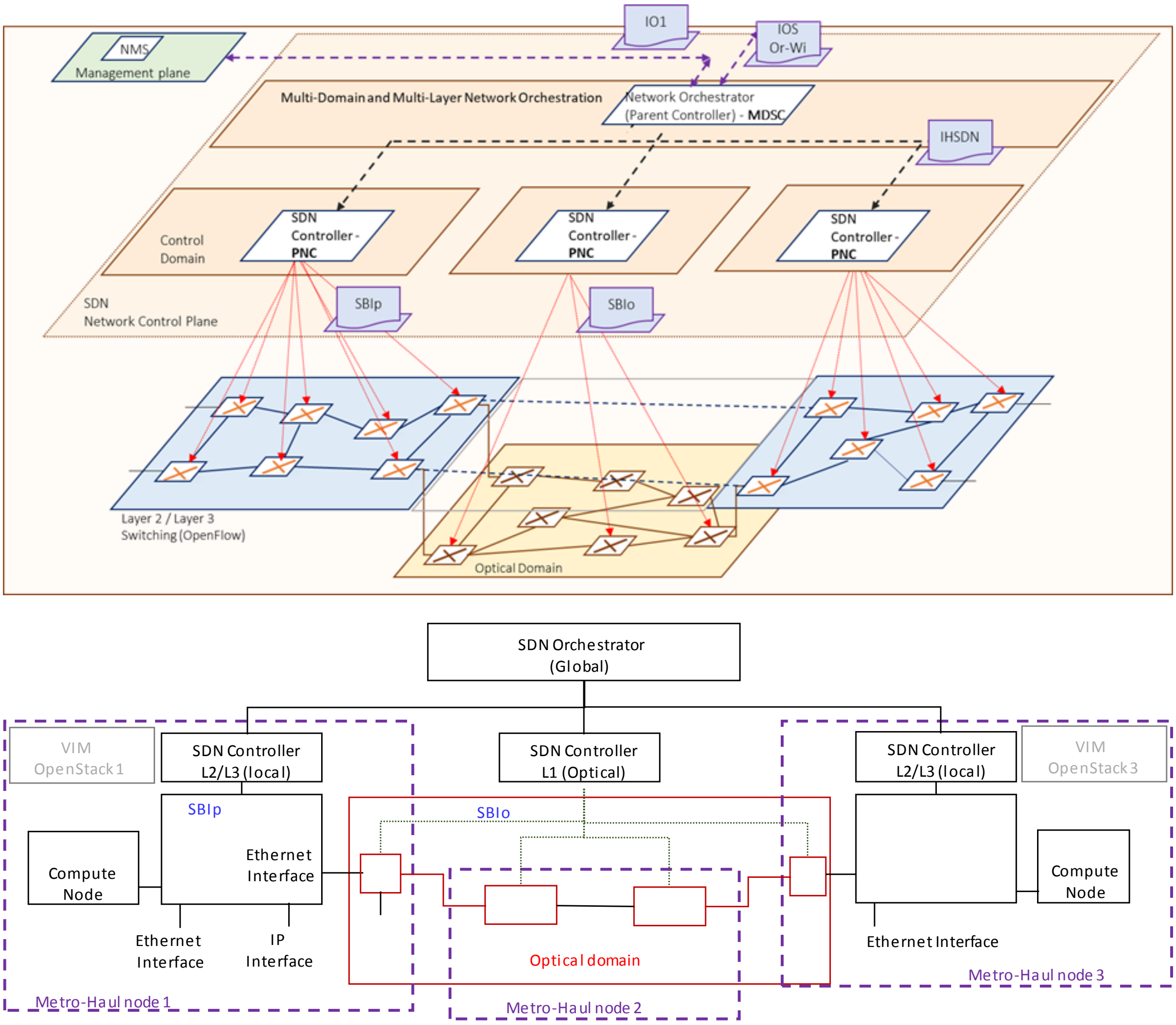


Figure 26. Network Orchestration Architecture.

Element	Functions
Network Orchestrator or MDSC (parent)	• A Multi-Domain Service Coordinator (MDSC) is a functional block that implements multi domain coordination and virtualization/abstraction (referred to as network-related functions) and customer mapping/translation and virtual service coordination (service-related functions). • The MDSC responds to connectivity requests and detaches the network and service control from underlying technology
PNC (children)	• Provisioning Network Controllers (PNCs) manage the network resources. They correspond to the controllers within their domains

3.3.5 Compute and Storage Integration

Joint IT/Cloud and Network Orchestration is used to refer to the coordination of resources to deploy services and applications that require storage, computing and networking resources. NFV can be initially defined as an architecture and deployment model around the idea of replacing dedicated network appliances — such as routers and firewalls — with software implementations (guests) running on common shared hardware (hosts), becoming Virtualized Network Functions (VNFs). NFV relies and builds on top the state of art and advances regarding servers "virtualization" and cloud computing and management, i.e., the ability to allocate VM, or containers over a common, shared infrastructure by means of a hypervisor, with direct control over the hardware resources. It is important to highlight that it is the function that is virtualized, keeping the same function logic but executed in a virtualized environment. The benefits have been well established, including lower costs, replacing dedicated appliances with shared servers; use capacity on demand and efficient resource usage, reduce operational costs with fewer appliances to deploy and maintain, enable e.g. migrations, support on-demand, and pay-as-you-go deployment models and enable innovation by making it easier to develop and deploy network functions.

The ETSI NFV architecture defines the NFV Infrastructure (NFVI) deployed across multiple points of presence (NFVI-PoP) for supporting the instantiation of VMs, along with the Management and Orchestration (MANO) subsystem, which deals with the orchestration of VNFs and how to deploy them as components of the so-called Network Services.

Element	**Functions**
NFVO	• The NFVO performs Service Orchestration, that is, the part of service instantiation involving the functional split of the service into/amongst different VNFs – that may be managed by different managers (VNFMs) by different vendors – and their logical interconnection (called VNF forwarding graphs) and Resource Orchestration that deals with the allocation of resources to support the VNFs and the logical links
WIM	• Responsible for provisioning of connectivity paths between VNFs.
VIM	• Responsible for the management of the NFVI and the instantiation of the VMs of the VNFs in a single data center domain (AMEN/MCEN).

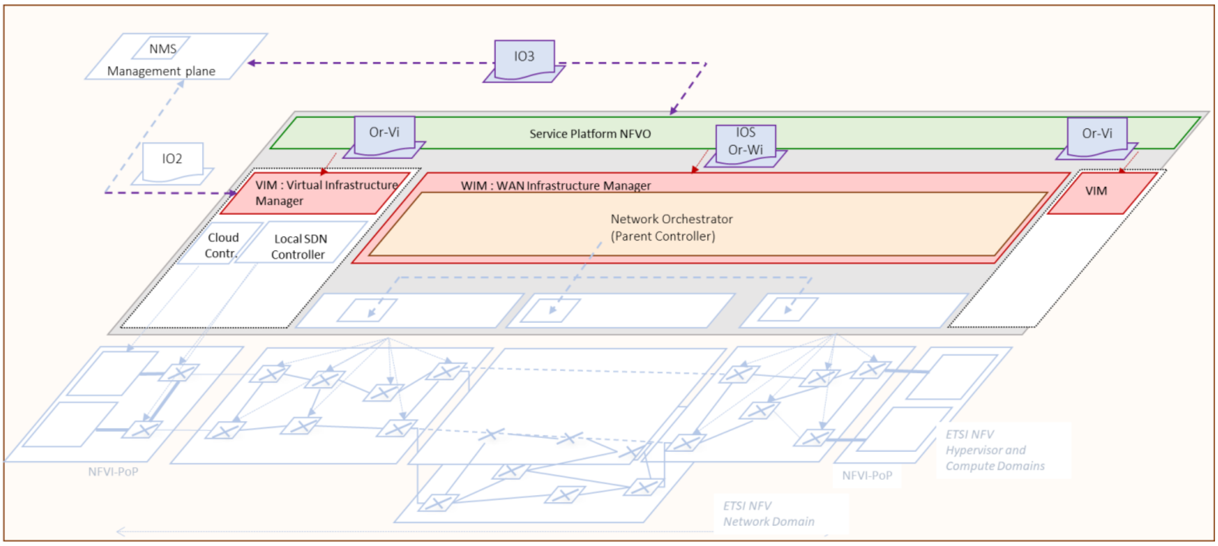


Figure 27. Integration with Computing and Storage, ETSI/NFV MANO

3.4 ETSI MANO / Slicing Integration

The ETSI NFV framework can be used as a starting point for a concrete implementation of a generic slicing architecture, in which network slice instances are NFV Network Services (NS), encompassing NS endpoints and one or more VNFs interconnected by logical links, forming VNF forwarding graphs (VNFFGs). Logical links are thus mapped to supporting network connectivity services which may, in turn, span multiple network segments.

A key service offered by the COM system is network slices. The ETSI NFV framework can be used as a starting point for a concrete implementation of a generic slicing architecture, in which network slice instances are NFV Network Services (NS), encompassing NS endpoints and one or more VNFs interconnected by logical links, forming VNF Forwarding Graphs (VNFFGs). Logical links are thus mapped to supporting network connectivity services which may, in turn, span multiple network segments. This is shown in, where multiple NFVO (green, blue), potentially managed by different users or operators can have shared access to a common NFVI managed by their respective VIM/WIMs, and each NFVO instantiated network service (with its corresponding VNFs) is a slice instance.

In this context, there are still a few challenges, some expected to be addressed in successive refinements of the architecture. For example, the focus on the Network Service may not cover all use cases, and additional functions are required to render the NFVO/VIM a full featured service platform for SlaaS, which requires specific support for slicing and multi-tenancy. Reference implementations of the architecture have been focused on centralized deployments until recently, where the use of multiple VIMs is considered and the integration with transport networks is still an ongoing debate; current specifications mention the need to provision paths between VNFs in VNFFG and such paths have oftentimes been assumed to be L2 and L3 tunnelling technologies. The current trend is reflected in considering multiple VIMs interconnected by a WIM which delegates its functions or is implemented in terms of an SDN controller or orchestrator. Such SDN controllers may, in turn, have specific capabilities supporting network virtualization, although overlapping functionalities are not uncommon. For example, OpenDayLight supports Virtual Tenant Networks (VTN), which allow users to define the network with a look and feel of conventional L2/L3 network.

Within the scope of METRO-HAUL, the main focus of a network slice is the ETSI NFV Network Services, that is, a set of interconnected VNFs, across the METRO-HAUL infrastructure (see, for example, Figure 4). It is worth noting that METRO-HAUL aims at extending the current state of art in key projects in order to include PNFs (Physical Network Function) and VNFs (ex. PNFs are physically attached to packet switches in METRO-HAUL nodes).

Multi-tenancy support can be added by thin wrappers on top of (possibly dedicated) MANO systems, relying on the multi-tenancy capabilities of the underlying VIM / WIM (see Figure 28). Alternatively, a single MANO may implement all the functions required to support this requirement.

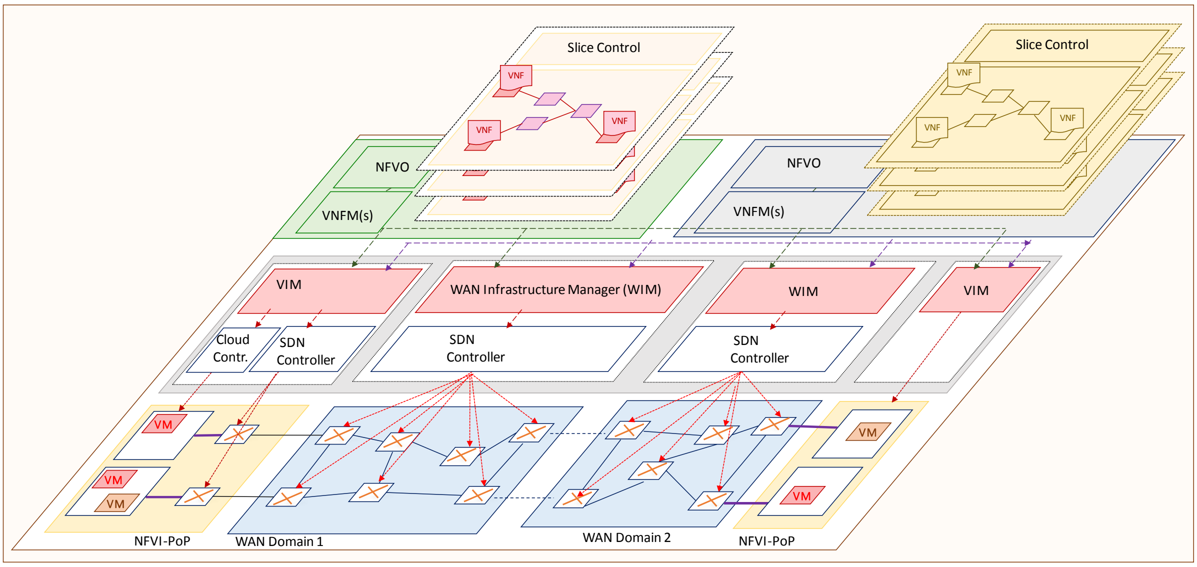


Figure 28. Network Slicing using the integrated SDN/NFV framework. Different Tenants (e.g. Green / Blue) manage their NFVO to deploy Network Services and Slices over a set of shared VIM/WIM spanning multiple PoP and domains. Each slice has a dedicated Control Plane instance.

3.5 Monitoring and Data Analytics Subsystem

Autonomic networking entails the capability to do measurements on the data plane and generating data records that are collected and analysed to discover patterns (knowledge) from data (KDD). Such knowledge can be used to issue re-configuration/re-optimization recommendations toward control, orchestration and management (COM) modules such as an SDN controller or an NFV orchestrator. For instance, bit error rate (BER) degradation can be detected on a lightpath and a recommendation to the SDN controller can be issued to re-route such lightpath [Vel17]; this is referred to as a *control loop*. BER, optical power, and other parameters can be metered in optical transponders. In addition, specific monitoring devices can be included to complement the parameters metered by optical devices, e.g., Optical Spectrum Analysers were used in [Vel18] for failure localization purposes.

In METRO-HAUL, data analytics is distributed; optical, MPLS, and IT nodes export measurements to a collocated monitoring and data analytics (MDA) agent with local data analytics and decision making capabilities, whereas a centralized MDA controller running in the control and management plane include domain-wide decision making capabilities. Figure 29 illustrates the hierarchical architecture of the MDA system, as well as communication interfaces with COM modules and monitorable devices.

Collection of data directly from the control plane is critical as it provides visibility into network operational state, rather than the traditional statistical information related to interface performance. Obtaining a real-time view of network control information, such as reachability and traffic classification; would also allow the formulation of proactive strategies to dealt with transient issues that arise due to performance bottlenecks, network failures, or even operational configuration errors. Streaming telemetry directly from the control plane would facilitate network layer fault correlation and trending in 5G networks. Especially when it's not clear why applications are performing poorly, and there is a need to identify bottlenecks at specific layers, or interfaces, that are impacting higher-layer network connectivity or applications. Overall the design framework required for streaming telemetry is different from traditional monitoring mechanisms, protocol requirements would include:1) Streaming of data in real-time, as fast as possible; 2) Supporting a push model; 3) Provide full access to operational data and 4) Support a policy model for defining time periods for collecting data, including tunable parameters (scale up or down).

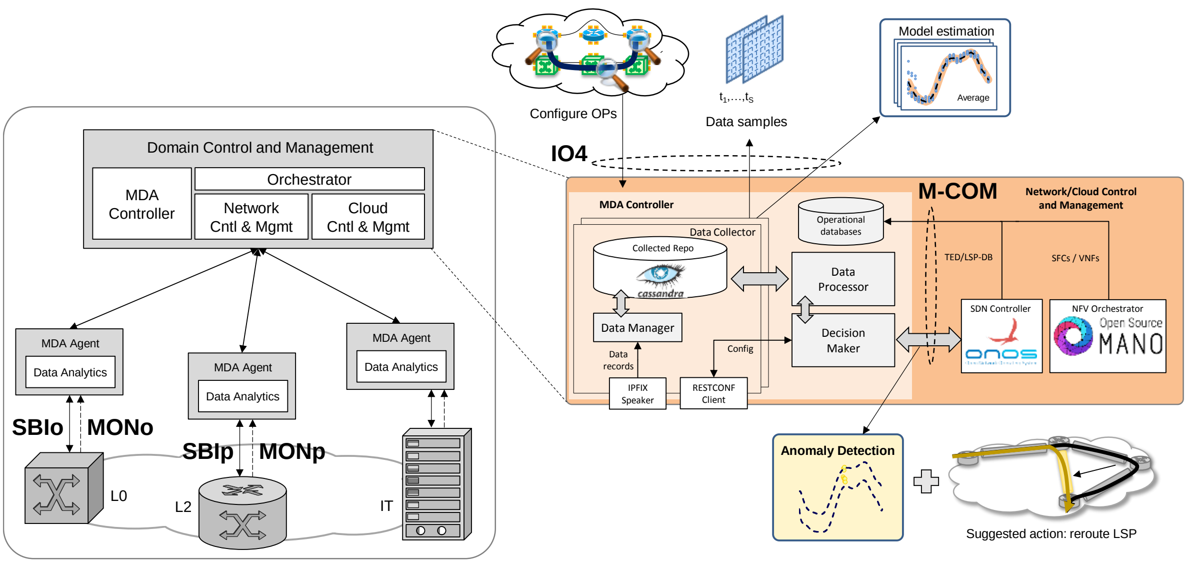


Figure 29. Monitoring and Data Analytics Architecture

3.5.1 MDA Agents

MDA Agents receive messages from physical nodes containing data records with data from observation points belonging to an observation domain (network slice). The MDA agent includes a local KDD module containing KDD applications for handling and processing data records.

MDA agents include a local configuration module to allow local control loops implementation, e.g., collecting monitoring data and applying re-configuration/re-tuning to devices in the local node.

3.5.2 MDA Controller

Data records and notifications from the MDA agents are received in the centralized MDA controller, where they are stored and processed in correlation with operational databases. Eventually, the

MDA controller may issue notifications and suggested actions to COM modules, like an SDN controller for the application of primitives, e.g., such as reconfiguration, through the M-COM interface.

3.5.3 Interfaces

Interaction between the monitoring service and the rest of METRO-HAUL entities is enabled through the following interfaces:

- IO4 (Section 4.1.4). Interface associated to external configuration of observation points (OPs) and data retrieval from other METRO-HAUL services.
- M-COM (Section 4.1.11). Interface used to retrieve of operational databases and notify data-driven events from/to COM modules.
- SBIp/SBIo (Sections 4.1.12, 4.1.13). Interface with network devices for OP (de-)activation and/or local reconfiguration loops.
- MONp/MONo (Section 4.1.14). Interface for data collection from network devices.

Interfaces between monitoring service and IT devices are to be defined similarly to the other SBIx and MONx interfaces.

3.5.4 Slicing support

The MDA system must be aligned with the slicing approaches within METRO-HAUL, currently under study. A preliminary proof-of-concept (PoC) has been developed in the context of multi-operator multilayer optical networks [Vel18Sl]. To this respect, some major changes are required inside the architecture in Figure 29.

On the one hand, a different local KDD application is created for each network slice in the related MDA agents. To isolate KDD application execution, each KDD application runs inside a container.

On the other hand, the MDA controller introduces a new slice manager which forwards the received data from MDA agents to local modules or to other MDA instances, so-called customer network controller (CNC), in charge of the corresponding slice for remote processing, through a new north-bound interface (NBI), to be defined in a further stage.

3.6 Placement, Planning, and Reconfiguration Subsystem

The aim of the Placement, Planning, and Reconfiguration Subsystem (alias, in short, Network Planner) is to optimize the resource allocation in the optical metro network in order to effectively provision services featured by heterogeneous requirements. This task comprises the provisioning of Virtual Network Functions in specific computing nodes (e.g. the mini-datacenter distributed in the metro network) and network resources allocation.

The traffic generated by end-user oriented services (such as Smart Factories, Live TV distribution, Content Delivery Network, etc.) have different requirements in terms of bandwidth, delay and QoS. For this reason, the intelligent optimization algorithms of the optimization module will apply different policies and strategies for the aforementioned services in order to optimize VNFs and network resources in advance, reacting to specific events. For all the cases, however, the optimizer will need data coming from the data plane (optical layer, layer 2 and IP layer) through the front-end interfaces, to learn the state of the network. Figure 30 shows the METRO-HAUL Service Platform, where the dashed red circle highlights the Placement, Planning, and Reconfiguration Subsystem. In the following subsections, the front-end and back-end interfaces will be explained, and especially how they interact to implement planning and to provide solutions to the METRO-HAUL control plane.

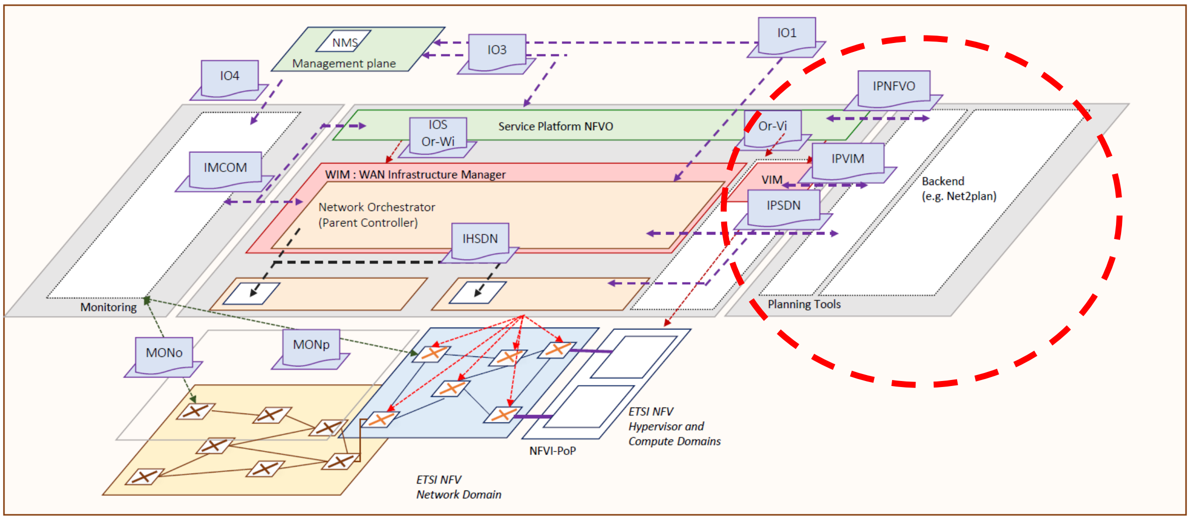


Figure 30. METRO-HAUL unified service platform

The open-source Net2Plan tool has been chosen as the common framework for materializing the placement, planning and reconfiguration of the VNFs, IT and network resources. Specialized algorithms will be used or developed in order to optimize the aforementioned resources.

3.6.1 Front-end

The Front-end contains the interfaces that allow the exchange of information necessary for the planning tool (the back-end) to carry out its operations. Thanks to the IPNFVO, IPVIM, IPSDN interfaces, the planning tool is able to query the Service Platform NFVO and the WIM in order to plan the network properly and provides network resource-planning solutions. The IPNFVO interface is used to obtain information about service chain characteristics such as start/end points, type and number of VNFs (requested service chain), etc. The IPSDN interface is in charge of providing the topology information and monitoring data. Thanks to this interface it is possible to provide to the planning tool information on the physical structure of the network, e.g., identify the forwarding and NFV-capable nodes, provide the transmission delay of physical nodes (e.g., L2/L1 switches), available bandwidth resources, propagation delay, etc. Eventually, the IPVIM interface will provide the resource requirements information in terms of computation, storage and networking capability of the METRO-HAUL nodes (AMEN, MCEN), that is, the VNF resources requirements (CPU cores, RAM, storage), scaling constraints, VIM instances real-time status. The last two allow the planning tool to decide whether it is possible to provision the new services by deciding whether a new service can be accommodated in an existing VNF by performing a VM upgrade (scale up, scale down) or whether other nodes must be utilized to provision other VNFs (scale in, scale out).

The Net2Plan graphical user interface will be adapted allowing users interacting with the network emulating OSS behaviour.

3.6.2 Back-end

The back-end provides the common framework to run the planning and provisioning algorithms. In order to develop the back-end, Net2Plan has been selected as integration environment. Net2Plan is a free and open-source Java-based tool devoted to the planning, optimization and evaluation of communication networks. It runs two main algorithms: VNF Placement and Scaling Optimizer (NPSO) and Network Resource Allocation Optimizer (NRAO).

Given a new service request, the NPSO will provide the VNF placement on the METRO-HAUL nodes (AMEN, MCEN). The request is characterized by the end-points of the service (source and destination), and by service chains, i.e., a sequential concatenation of VNFs to provide a specific Internet service (e.g., VoIP, Web Service, etc.) to the users. Besides, the NRAO will perform the resources assignment in the underlining optical network.

In both the modules, the solutions suggested to the METRO-HAUL control plane by the Network Planner will be based on traffic predictions returned by the machine-learning algorithms. These algorithms exploit self-learning procedures to evaluate the history of the network and the plans that have been adopted in the past. Based on machine learning, they are then capable of providing the best solutions of network/IT resource assignment and routing. The introduction of artificial intelligence will allow the planning tool to learn the best network configurations, so that input to the control plane can be provided on demand with low computational complexity.

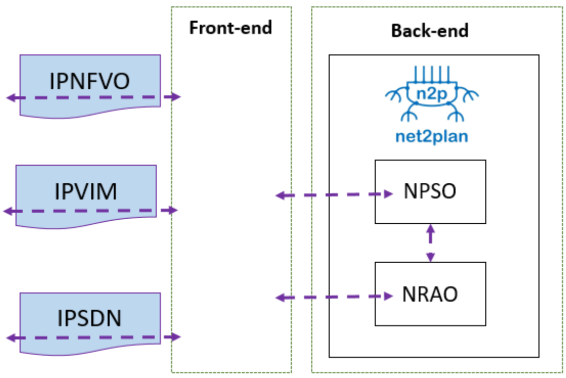


Figure 31. METRO-HAUL NP (planning) architecture

Figure 30 shows the interaction of the different components that form the planning architecture. In particular, the output of the two planning algorithms NPSO and NRAO provides the mapping of service chains and the resources allocation, respectively.

Both NPSPO and NRAO will be able to operate into two modes: a) reactive: response to a single event of request coming from the NFVO (in case of service request) or from the VIM or WIM (in case of direct connection setup; b) proactive: periodic re-optimization of the network based on traffic and service-chain predictions. It will be up to the network operator to decide which of the two modes to activate, being understood that it would be possible to activate both the modes, i.e. periodical re-optimization and reactive operations between one re-optimization session and the next re-optimization.

The tool, Net2Plan, will be integrated in the provisioning workflow. It is under consideration the particular interactions to be completed. The key interactions for the planning tool will be with the NFVO and also with the SDN controllers. It is mandatory that Net2Plan is aware of the overall overview of the network resources state in order to provide optimality. Performance information might be also accessible from the monitoring and data analytics system in case another agent might request such information for its purposes.

3.6.3 Integration with Monitoring and Data analytics

The Monitoring and Data analytics system illustrated in Sec. 4.3 supports an Autonomic Decision Maker Subsystem. Actually, we can say that also the Placement, Planning, and Reconfiguration Subsystem (or Network Planner) consumes monitoring data and performs data analytics. But, as explained in previous section, the Network Planner and the Autonomic Decision Maker are distinct and different subsystems.

The interaction between the Network Planner and the Autonomic Decision Maker will be defined in details and precisely regulated later on in the course of the project. At this time, we can say that the general principle will be to avoid any possible conflict between the suggestions provided to the control plane by the two systems. Several methods can be adopted to ensure this: just for sake of for example, let us mention the following possibilities:

- the two systems operate on different time-scales, i.e. long term for the Network Planner (strategic optimization), short term for the Autonomic Decision Maker (immediate reaction to unexpected events such as failures, etc.)
- the network operator can decide explicitly to switch between a Planned and an Autonomic operation mode. In the first mode the control plane will get suggestion from the network planner, while in the second mode will take inputs from the Autonomic Decider
- the network planner can operate only in proactive mode, proposing periodic re-optimizations, while the Autonomic Decider is responsible for reactive actions

3.7 Summary of Interfaces

Figure 32 summarizes the interfaces within the COM components. These interfaces correspond to interfaces between logical functional entities and, at the time being, we are only concerned with the macroscopic, high-level requirements. It is thus possible that some of these interfaces end up unified / coalesced or reused from existing implementations or interfaces. Let us note that we only highlight externally visible interfaces and interfaces that are extended in the context of METRO-HAUL. Internal interfaces, interfaces used as-is are not described, and they are part of the respective components.

Table 3 Summary of Interfaces

NAME	Description
IO1	Operator – SDN NBI exported by the SDN controller for service provisioning, topology mgmt., etc.
IO2/ Or-Vi	Operator – VIM for Authentication (Keystone), Instantiation of VMs (Nova), Basic Intra-node connectivity (Neutron) as per the OpenStack existing API.
IO3	Operator – NFVO (OSM). Based on OSM interfaces and APIs.
IO4	Operator – Monitoring

IPON	Interface between the parent controller (Orchestrator) and the SDN PON controller – Controller Specific
IPNFVO	NFVO – Planning Tool – Interface for planning and placement
IPSDN	SDN – Planning Tool – Interface for planning (SDN) [Complements IPNFVO]
IPVIM	VIM – Planning Tool – Interface for planning and placement (VIM) [Complements IPNFVO]
IHSDN	Interface between the parent controller (Orchestrator) and the packet / optical controllers.
Or-Wi / I-OS	Interface between Orchestrator – WIM [NFVO-TSDN, follow OSM plugin]
MCOM	Interface between the monitoring system and the SDN controller (parent) or the NFVO
SBIp	Interface to packet switches -- OpenFlow
SBIo	Interface to optical infrastructure elements (NETCONF / YANG)
MONp	Interface from packet switches to the Monitoring component
MONo	Interface from optical switches to the Monitoring component
IO2/ Or-Vi	Operator – VIM for Authentication (Keystone), Instantiation of VMs (Nova), Basic Intra-node connectivity (Neutron) OpenStack API
IO3	Operator – NFVO (OSM). Based on OSM interfaces and APIs. Req1: specify a Network Service Descriptor
IO4	Operator – Monitoring, interface used by operator to access data on the MDA and related databases. The default implementation of this interface will be given by the existing software used to implement the component. More specific interfaces can be implemented depending on subsystems requirements.

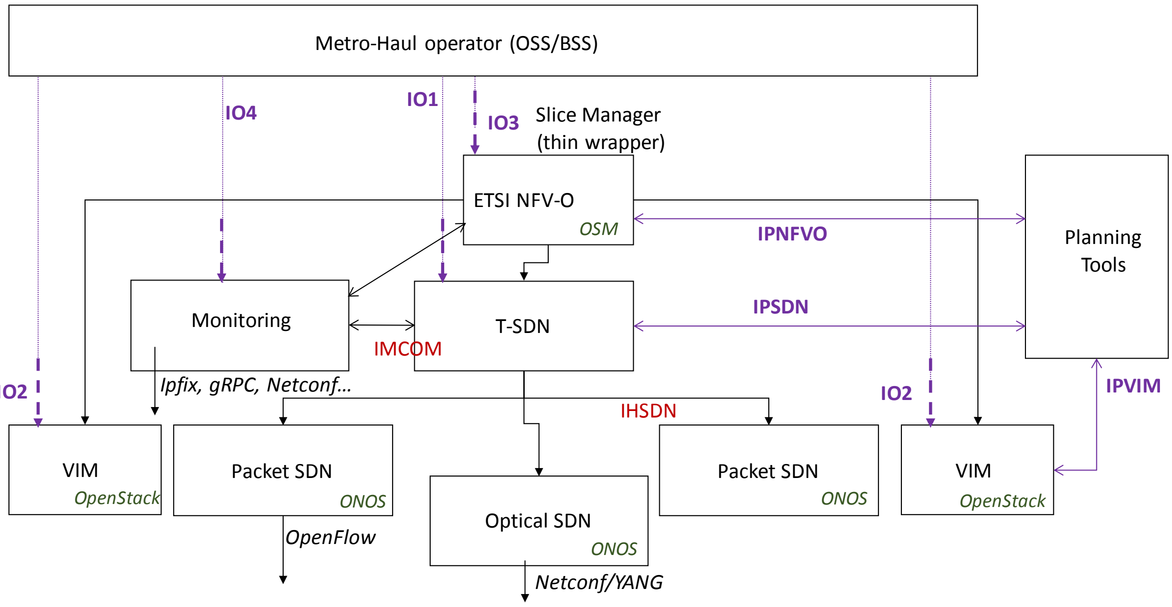


Figure 32. METRO-HAUL unified service platform main external interfaces

3.8 Final considerations w.r.t. CORD and CloudBBF architectures

The main difference between the CORD and the METRO-HAUL architectures is the fact that CORD relies on the presence of an orchestrator (XOS) within each network node, not considering, for the time being, the need for coordinating computing resources among multiple nodes. Nevertheless, even if switching of residential and mobile traffic is performed at IP level, an orchestrator coordinating the metro network resources and all the nodes is necessary, at least for enterprise traffic. The current approach is to rely on ONOS for this purpose, but this has the limitation that only networking resources can be controlled. However, within the CORD community, it is currently under evaluation the use of ONAP [ONAP].

METRO-HAUL combines Packet and Optical technologies and therefore control of both kinds of devices are in scope of the Control, Management and Orchestration (COM). As such, METRO-HAUL architecture and the CloudCO reference architecture cannot be fully aligned, since METRO-HAUL considers in scope some aspects that are left out of CloudCO and it does this following the ETSI NFV approach where the NFVO (embedded in the CCO Domain Orchestrator) invokes a WIM to request WAN resources.

A possible mapping of the METRO-HAUL architecture to the CloudCO reference architecture could be achieved considering a merge of the CloudCO reference architecture with the extra elements required for the extended scope considered by METRO-HAUL. Such an exercise is shown in Figure 32.

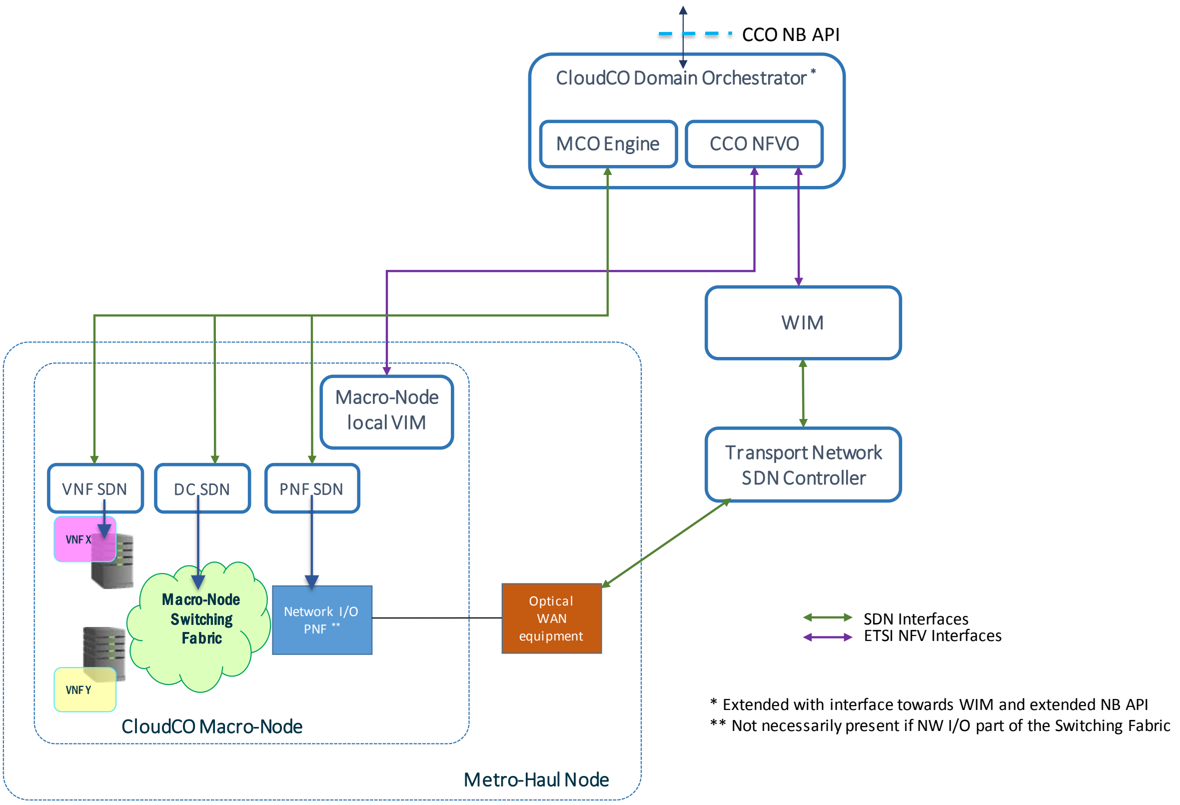


Figure 33. Possible mapping of CloudCO to METRO-HAUL architecture.

The CloudCO Domain Orchestrator could be augmented in METRO-HAUL having an interface in the CloudCO NFVO towards an ETSI WIM (WAN Infrastructure Manager) that puts in scope the control of the WAN packet and optical resources as needed by METRO-HAUL architecture. It is very likely that this extension of the scope of CloudCO by METRO-HAUL will result at least in extensions to the CCO NB API, or a complete different NB API can be proposed by METRO-HAUL, depending on the kind of operations finally exposed by the CCO NB API. The CCO NB API has not been yet defined, but it will very likely focus initially on broadband access use cases (e.g. PPP residential subscriber operations, etc).

Southbound Interfaces towards the Macro-Node SDN controlled elements (green lines to CloudCO Macro-Node in Figure 33) would not be impacted by such an extension, so an alignment between the two architectures can be preserved for that part. In fact, CloudCO still misses the definition of both the external (e.g NB API) and the internal interfaces in the architecture. The definition of these interfaces is the subject of active Working Texts in the BBF (WT-411, WT-412, WT-413 and WT-416). Therefore, contribution from METRO-HAUL to the definition of these interfaces is still possible as well as the reuse by Metro Haul of CloudCO interfaces as these are defined, if there is a match in the requirements of both architectures.

4 Initial Requirements and relevant considerations

4.1 Interface Requirements

4.1.1 IO1

IO1 is defined as the interface between the operator and the hierarchical SDN system (without excluding that it can be used by additional interface users, such as WIM components, or higher layer orchestrators). The main functions are related to the Service provisioning and topology and inventory management.

Function	High level requirements
Service Provisioning	• Provision an SDN Network Connectivity Service, that is a point to point connection between endpoints. Such endpoints may be packet based (e.g. MAC addresses corresponding to hosts or OpenFlow switches) or optical, corresponding to transceivers or ROADM ports for media channel services. • Retrieve the status of active network connectivity services, based on the provisioning actions in the previous point. • Delete an active network connectivity service.
Topology Management	• Enable the configuration of a network topology, encompassing links and nodes/ports • Retrieve the network resource status • Retrieve the network topology • Retrieve detailed link attributes

	• Retrieve detailed node attributes
Remarks	
• Service Interface defined by YANG models. This “service” interface will take into account e.g. TAPI models [TAPI], IETF models, or models specific for optical devices [OpenROADM] • IO1 sub-interfaces to build upon existing interfaces in relevant projects (e.g. to retrieve a network topology) • Let us note that METRO-HAUL will aim at applying recursive interfaces that can be used in the IO1 as well as the IHSDN. Interfaces like TAPI are designed in this regard.	

Transport API [TAPI] is a standard API defined by the Open Networking Foundation (ONF) that allows a TAPI client, such as a carrier’s orchestration platform or a customer’s application, to retrieve information from and control a domain of transport r. In other words, TAPI is a standard NBI for a Transport SDN Controller. It supports both high level technology independent service (i.e., intent-like) and detailed technology-specific service, depending on policy.

As a component of Transport SDN, TAPI enables programmatic control of the carrier’s transport network to support faster and more flexible allocation of network resources to support application demands (e.g., bandwidth or latency). The benefits include reduction of cost due to operational simplification and reduced delay for the introduction of new equipment and services, as well as the ability to develop and offer new revenue-producing services such as network slicing and virtualization for 5G and IoT applications.

TAPI supports unified control of domains with different technologies using a common technology-agnostic framework based on abstracted information models. This allows the carrier to deploy SDN broadly across equipment from different vendors and with different vintages, integrating both greenfield and brownfield environments as opposed to requiring major turnover and investment in new equipment.

In the context of METRO-HAUL, TAPI will be demonstrated as NBI interface and framework, for the different transport layers suitable extended where appropriate. The main focus will be the Topology Service, which supports retrieval of Topology information from the Controller in the form of Node, Link & Edge-Point details. This information can be used for path computation, planning and analysis purposes and supports virtualization of network resources for particular client applications and the Connectivity Service, which allows the client to retrieve information about and request new point-to-point, connectivity service across the transport network. Support for both single layer and multi-layer connectivity services is included.

4.1.2 IO2 / Or-Vi

Function	High level requirements
NFVI management	In practical terms, this API is the one defined by the OpenStack cloud management system. From the ETSI NFV architecture, this interface

	corresponds to the Or-Vi interface. The interface must be able to: • Configure security groups, user networks and subnetworks, software routers and related information • Instantiate Virtual Machines given their requirements in terms of size (memory, CPU, etc.) and for a given software image • Retrieve status and information of the nodes / servers part of the NFVI and be able to monitor the instantiated VMs. • Retrieve status and monitor the status of the pooled hypervisors that constitute the NFVI.

4.1.3 IO3

Function	**High level requirements**
Network service management	A standard OSM northbound API will be used or adapted for that interconnect the Operator with the NFV-O. This interconnection is illustrated on the left side of Figure 36. It is targeted to permit the definition, instantiation and management of network services via ETSI network service descriptors (NSD). OSM is technology agnostic in its principle. It is being explored how to introduce considerations coming from the transport network in the network services management. Additionally, a variant of this interface is under study to provide the NSD to the Network Planner so that additional specific details of the demand characteristics can be introduced. This interconnection is illustrated on the right side of Figure 36.

Service data information	• Labelled traffic (based on the slices) in terms of network demands.
	• Demand characteristics: start/end points, bandwidth, latency, sorted sequence of VNFs. Two options are under consideration:
	    ○ This information is provided within the NSD directly from the operator.
	    ○ Demand features are, in the first instance, sent to the Network Planner from the Operator with regular NSDs and additional information may be provided in parallel. Subsequently, the Network Planner notifies the NFV-O with a modified NSD in the instantiation time via the IPNFVO.
	• Number of service requests in real time.

4.1.4 IO4

Function	**High level requirements**
Monitoring Procedures	This interface is in charge of managing monitoring procedures through enable/disable operations for OPs, retrieving monitoring data, as well as retrieving predictive models computed by the monitoring module. Specifically, the next functional requirements are considered:

	• Get list of OPs: Returns the list of Observation Points. • Enable/disable OP. After an OP is enabled, monitoring data starts to be collected and stored. OPs disable entails stopping collecting monitoring data samples. • Retrieve OP data: Returns available data in the datastore for a given time window for a given OP. Simple operations are available to be applied on the retrieved data, for instance average, max, min, percentile, etc. • Subscribe to OP. Subscribes to new data received for a given OP. • Notify new OP data: Notifies about new OP data in the datastore. • Get list of models: Returns the list of Models. • Get model: Return a given Model.

Figure 34 illustrates the consumer (arrow head in dashed lines, for notifications) and the provider (arrow head in solid lines, for request/response interactions) of each of the functionalities of the interfaces, whereas all of them are defined in the next tables.

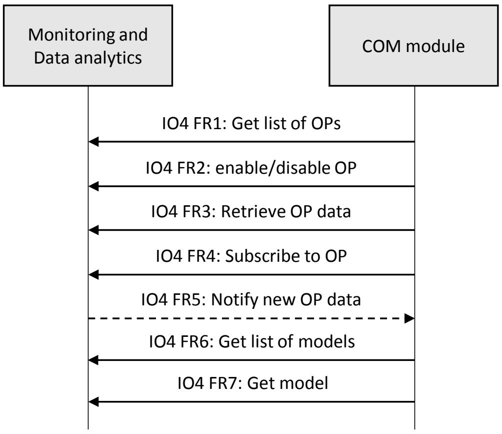


Figure 34. Operations supported by IO4 interface

4.1.5 IPON

Function	**High level requirements**

PON Control	This interface is used to control the Passive Optical Network from the Network Orchestrator. The main functions are related to:
	• Configure PON elements
	  ○ Management of VLANs
	  ○ Management of Virtual Private LAN Services (VPLSs)
	  ○ Management of ports
	    ▪ Bridging/ unbridging of ports
	  ○ Management of bandwidth profiles
	    ▪ QoS parameters: committed-info-rate (CIR), assured-info-rate (AIR), excess-info-rate (EIR), delay-tolerance
	  ○ Management of queues

	▪ Generation of priority bit to queue mapping ▪ Assignment of bandwidth profiles to upstream queues ▪ Management of ingress QoS queue profiles and assignment to VLANs • Retrieve status of PON elements ○ OLT, ONU, VPLS, VLAN, port, queue, bandwidth profile status

4.1.6 IPNFVO

This interface is in charge of the ETSI NFV-O and Network Planner intercommunication procedures. Network Planner needs to know the information coming from the requests made by the OSS / BSS, therefore the IPNFVO interface maps the requests in order to be used by the algorithms implemented in the Network Planner in order to provide network optimization and reconfiguration (see Figure 35).

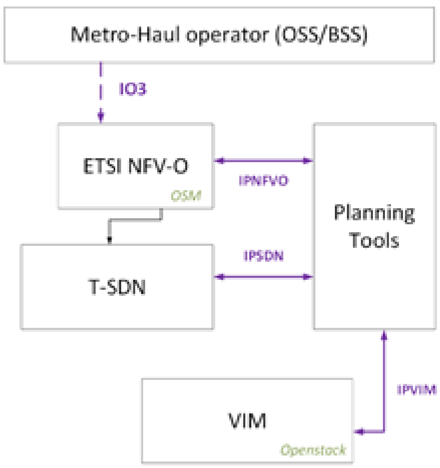


Figure 35. Network Optimization and Reconfiguration interfaces

This interface depends on IO3 and particularly on the demand characteristics (start/end points, bandwidth, latency, sorted sequence of VNFs) included in the NSD. Two options are under consideration:

- Invasive option regarding the OSM code modification (left side Figure 36): This option exploits the interconnection between the operator and the NFV-O (ETSI-OSM). First, (1) the operator requirements are sent to the NFVO in the NSD through IO3. Then (2) the NFVO sends a modified (transport technology specific) NSD to the Network Planner via IPNFVO, the planning tool executes the algorithms. The Network Planner (3) returns the VNFs placements in pre-instantiation time to the NFV-O also including the VLDs the inherent transport connectivity needed by the WIM in the final time (4) which corresponds to the VNFs instantiation. New features have to be added to ETSI OSM in this option.
- Less invasive option regarding the OSM code modification (right side Figure 36): In this case, (1) the Network Planner receives the request information in a NSD directly from the operator as a variant of IO3. Then, (2) the planning tool computes the optimal NFV

allocation and notifies OSM including a token for post-allocation verification processes. Subsequently, (3) the NFV-O sends to the WIM the VNFs instantiation commands, and finally (4) the WIM can query the Network Planner if needed with the token information to perform verification procedures.

The Network Planner requires information from the NVF-O as input of the algorithms to be executed in its Back-End. In particular, the Network Planner has to be aware of the operator demand requirements in terms of end-to-end nodes, sequence of VNFs, bandwidth and latency. On the other way, the results obtained from the execution of the algorithms need to be sent back to the NFV-O though this (IPNFVO) interface. In practice, two different variations of the following models are considered:

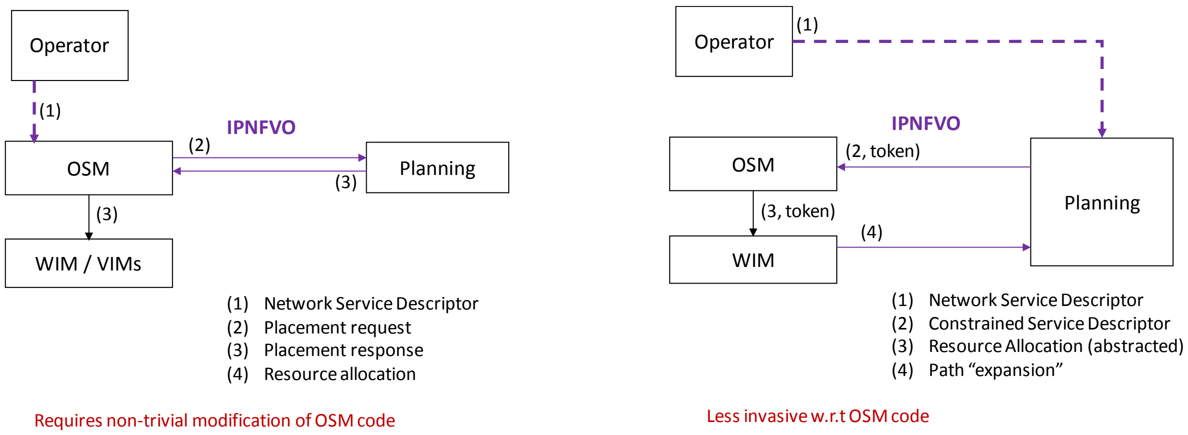


Figure 36. IPNFVO alternatives: (left) Invasive and (right) non-invasive OSM code modification

Function	**High level requirements**
Service chain characteristics	• Start/end points • Type and number of VNFs • Type of Interconnections (forwarding rules between VNFs of the same Service Chains) • Specific requirements of the virtual link (data rate, tolerated latency) • Cost to instantiate a new VNF • Disjoint placement of active/standby VNFs on physically separated machines • VNF Hardware accelerations requirement (hardware acceleration mechanisms) Manual placement of specific VNFs (location constraints)
NVF control	The OSM REST/API will be used or adapted for that purpose. The NFV-O will provide information about virtual resources to the planning tool. Providing IT resources information coming from the VIMs through this interface is under investigation.
NFV Instantiation Features	The results of the execution of the optimal algorithm will be sent to the NFV-O indicating the placement chosen for the service chain requests, and also some transport information in the NSD, if needed.

4.1.7 IPSDN

This is the NBI standard of the SDN controller (Network Control Transport). In particular, the planning tool will have read access to the network information to maintain an updated view of the topology and network flows. We are investigating two ways to read the above information and send network configuration information to the T-SDN. The first is the ability to directly issue control orders on the SDN network controllers; the second is via the NSD. Therefore, the latter will have the necessary entries concerning the definition of network paths with the relative topology.

In other words, it is possible that the definition of the flows (transport paths) to follow come from an updated version of the NSD and that information provided by the NVF-O such that it is passed to the SDN-Controller / WIM. In the other case, the IPSDN is used by the SDN-Controller/WIM to query the network planner about optimal NFV-placement-aware transport paths.

Function	**High level requirements**
Topology Information	• Forwarding nodes and NFV-capable Nodes • Transmission delays of physical nodes (e.g., L2/L1 switches) • Physical links in the optical domain and packet domain • Bandwidth resources • Propagation delay • Power consumption model of MCEN/AMEN
Monitoring data management	• Nodes connectivity status • Links load status • Failures alert

4.1.8 IPVIM

IPVIM is the interface between the VIM at each site and the NP sub-system. Macroscopically, the planning and placement components need to know the status of the resources, in terms of CPU usage, memory usage, etc. METRO-HAUL intent is to reuse existing VIM native interfaces as much as possible, provided they meet the main requirements.

Function	**High level requirements**
Compute/storage/networking resource requirements	• VNF resources requirements (CPU cores, RAM, storage) • Scaling constraints • VIM instances real-time status
VIM control	In the first instance, VIMs information and state will be provided by OSM to the planning tool via IPNVFO

4.1.9 IHSDN

This interface enables the end-to-end orchestration between the parent controller or network orchestrator and the children at each domain (e.g. packet switched domains, optical disaggregated networks or PON).

The terminology is based on ACTN [ACTN] although we decouple the terms and the actual interfaces, models and protocols. The ACTN architecture is to a large extent, agnostic of the actual used protocol and interfaces. This flexibility is used in METRO-HAUL to accommodate multiple possible interfaces.

Function	High level requirements
Network control Topology Management and Service Provisioning	• This Interface is an interface between an MDSC and a PNC. • It communicates requests for new connectivity or for bandwidth changes in the physical network. • The MDSC needs to communicate with multiple PNCs each responsible for control of a domain. • The interface presents an abstracted topology to the MDSC hiding technology specific aspects of the network and hiding topology according to policy.

4.1.10 Or-Wi

Function	High level requirements
WIM control	This interface is defined between the NFVO and the WIM, which will wrap and rely on a hierarchical SDN controller. The high level requirements are: • Define a logical network to which multiple endpoints / ports can be connected. This logical network supports a given ETSI network service or slice • Attach a given port from one site or location to the logical network. When two or more ports are attached to the logical network, it is assumed that the endpoints have network connectivity. Remarks: • This interface will be complemented by the interface exported by the hierarchical SDN controller and orchestrator, which is assumed to be accessible by the NFVO functional element. This includes topology and connection management.

4.1.11 M-COM

Function	High level requirements
Monitoring and	This interface is devoted to synchronize information from operational

Data Analytics Synchronization	databases from the COM module to the monitoring platform. Then, this information can be correlated with monitoring data and notify the COM module about different events, even providing recommended actions. Specifically, the following functional requirements are considered: • *Get list of databases:* returns the list of databases in the COM module. Each database is defined by the tuple <Database-Identifier, Database-Name, Resource-Type list>, where the defined resource types include node, link, and connection. • *Get database:* returns the contents of the specified database. Each resource is represented by a JSON object that contains the attributes of the resource. • *Subscribe/unsubscribe to database changes*: subscribes/unsubscribes to asynchronous notifications on changes in the selected database. • *Notify database change:* notifies about changes in resources within the specified database. The new JSON objects representing the changed resources are included. • *Subscribe/unsubscribe to network event*: subscribes/unsubscribes to asynchronous events issued during monitoring and data analytics processes. • *Notify event:* notifies about specific conditions or patterns found in the network during monitoring and data analytics processes. The notification can include a recommendation to the COM module.

Figure 37 illustrates the consumer (arrow head in dashed lines, for notifications) and the provider (arrow head in solid lines, for request/response interactions) of each of the functionalities of the interfaces.

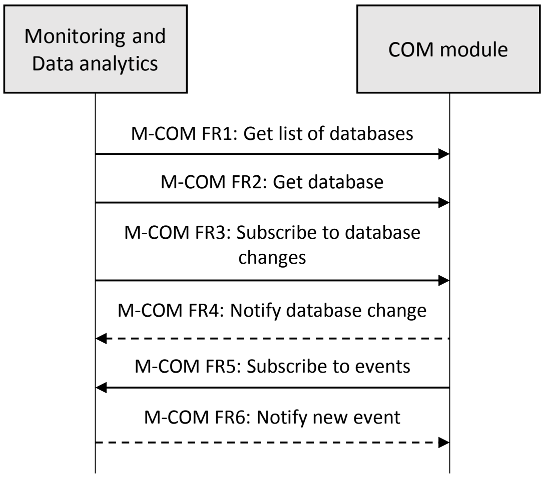


Figure 37. Operations supported by M-COM interface

4.1.12 SBIp

The South bound protocol for the packet layer (SBIp) will be based on OpenFlow, given the protocol maturity and availability of open source implementations that can be used in the project. OpenFlow enables network controllers to determine the path of network packets across a network of switches. It allows switches from different vendors to be managed remotely using a single, open protocol, being an enabler of software defined networking (SDN). OpenFlow allows remote administration of a switch's packet forwarding tables, by adding, modifying and removing packet matching rules and actions. The OpenFlow protocol is layered on top of the Transmission Control Protocol (TCP) and prescribes the use of Transport Layer Security (TLS). Controllers should listen on TCP port 6653 for switches that want to set up a connection. Earlier versions of the OpenFlow protocol unofficially used port 6633.

4.1.13 SBIo

SBIo is the south bound interface towards the network elements.

Function	**High level requirements**
Control and configuration of optical devices	• This client/server interface is an interface between the southbound METRO-HAUL Optical Network controller (client) and each agent controlling optical devices (server), i.e., O-NEs, OIEs and OSTs. • NETCONF protocol • Required operations change depending on the optical device.
ROADM O-NE	1. Retrieve overall configuration 2. Configure/Edit/Delete a cross-connection between an input optical port and an output optical port 3. Retrieve the list of established cross-connections 4. Retrieve state/monitoring parameters
XPONDER O-NE	1. Retrieve overall configuration 2. Configure/Edit/Delete optical signal 3. Retrieve the list of active optical signals 4. Configure/Edit/Delete a connection between tributary and optical

	port 5. Retrieve the list of established connections 6. Retrieve state/monitoring parameters

For the optical layer, the communication between the southbound interface of the METRO-HAUL optical network controller and the optical network elements and subsystems (O-NEs, OIEs and OSTs) will exploit the NETCONF protocol. The controller will act as NETCONF client, while each devices will be provided with a NETCONF server as described in Section 2.2. Optical devices will be modeled using YANG data models and the controller will be able to configure and monitor the optical devices through NETCONF operations (e.g., edit-config and RPCs) following the YANG description of the device.

In a first implementation phase, the METRO-HAUL optical controller has been extended with a set of southbound modules (i.e., device drivers) to enable monitoring and configuration of each specific device. For example, the first version of the ONOS driver for controlling the Lumentum ROADM-20 Whitebox has been already released and is available at https://gerrit.onosproject.org/#/c/17102/.

In a second implementation phase, a more generic driver will be implemented also in collaboration with the ONF ODTN Working Group for the support of all devices respectively based on OpenConfig and OpenROADM YANG models.

4.1.13.1 OpenConfig

OpenConfig is an informal working group of network operators sharing the goal of moving networks toward a more dynamic, programmable infrastructure by adopting software-defined networking principles such as declarative configuration and model-driven management and operations [OpenConfig].
Also following the choice of the ODTN group (phase 1), the METRO-HAUL consortium decides to adopt the OpenConfig YANG model for XPONDER O-NEs whose agent is developed within the project. However, the developed agents could not include all the optional modules described within the OpenConfig models.

4.1.13.2 OpenROADM

The OpenROADM Multi-Source Agreement (MSA) defines interoperability specifications for Reconfigurable Optical Add/Drop Multiplexers (ROADM) [OpenRoadm]. The consortium includes the modeling of the ROADM switch as well as transponders and pluggable optics. Specifications consist of both Optical interoperability as well as YANG data models.

The OpenROADM models will be used within the METRO-HAUL project to model ROADM O-NEs. However, the developed agents could not include all the optional modules described within the OpenROADM models.

4.1.13.3 Additional prototype models for experimental devices

To enable the support of all experimental optical devices that will be developed within the METRO-HAUL project, the aforementioned OpenConfig and OpenROADM models could be extended within the project (e.g., some fields can be added/modified).

Many commercial devices currently available within the project's partners are based on YANG models not consistent with OpenConfig or OpenROADM devices (e.g., the Lumentum ROADM-20 Whitebox). Such kind of devices will be supported at the controller developing a device specific NETCONF driver.

4.1.14 MONp and MONo

In METRO-HAUL, two different approaches for performing and collecting measurements from the devices have been considered: *monitoring* and *telemetry*. While monitoring is focused on performing measurements at regular intervals, e.g., every minute, telemetry is intended for the continuous measurement that is sent as a data stream and no regular intervals are strictly needed, i.e., measurements are sent as soon they are available or at a rate much shorter than for the monitoring approach, e.g., one per second. For both, monitoring and telemetry, every measurement might convey values for a set of parameters that are collected simultaneously and have a complete meaning. Therefore, measurements are defined as data records that follow a given data structure defined as a template, i.e., different templates are defined for different measurements to be collected.

Table 4 presents the template (*templateId*: 256) defined for MPLS-TP LSPs, where number of packets and number of bytes are measured.

Table 4 Template 256 for MPLS-TP LSPs

Field Name	**Description**
packetDeltaCount	Number of packets since the previous report.
layer2OctetDeltaCount	Number of L2 octets since the previous report.

For the optical later, two different templates have been defined so far for BER and optical power measured at the optical transponders (*templateId*: 310) and for the spectrum acquired by Optical Spectrum Analysers (*templateId*: 330).

Table 5 presents *templateId* 310, where BER and optical power are metered.

Table 5 Template 310 for Lightpaths at the transponders

Field Name	**Description**
ber	Bit error rate
rxPowerDecibelMilliwatts	Received optical power (dBm)

txPowerDecibelMilliwatts	Transmitted optical power (dBm)

Finally, Table 6 defines *templateId* 330 for optical spectrum monitoring.

Table 6 Template 330 for optical spectrum

Field Name	Description
opticalSpectrum	Ordered vector of tuples <frequencyGigaHertz, powerDecibelMilliwatts>

Although the previous templates are defined to be protocol independent, other fields are needed to be included when monitoring is used, since individual messages are sent for every measurement. Specifically, the Observation Point (OP) id and the direction in which the measurement has been performed are commonly needed, as defined in Table 7.

Table 7 Additional fields for monitoring messages

Field Name	Description
observationPointId	Id of the observation point
Direction	Direction of the connection / link

4.2 Preliminary selection of frameworks, projects and Interfaces

4.2.1 Hierarchical Network Orchestration

4.2.1.1 ACTN and / or ONF TAPI Frameworks

We have considered different choices for the development and deployment of a hierarchical SDN infrastructure. The main choice is related to the service models and actual protocols. We are considering different options for the hierarchical network orchestration:

- Design from scratch. This option has been discarded, due to the fact that several models that have been proposed within the community can be the basis for METRO-HAUL modelling work-
- OpenROAM Network and Service models. This option is not adapted to the use case, since it clearly focuses a single domain with OpenROADM devices.
- IETF ACTN - IETF topology models could be used within METRO-HAUL.
- ONF TAPI models Covering the main functions of topology management and connectivity provisioning. The preliminary PoC are being developed with this interface. It is expected that this interface will be the basis for the developments, but we are not excluding additional activities for specific demonstrators targeting IETF models.

4.2.2 SDN Controllers

The selection of SDN controllers in the scope of METRO-HAUL is given by the following requirements:

- Open Source Software, with a thriving community and the existence of sample applications and projects that demonstrate the main capabilities of the platform. This also includes availability of resources (tutorials, mailing lists, forums) to ease development.
- A mature code base that can be easily extended for the purposes of the project objectives,
- Previous experience with the software by the members of the consortium.
- Availability of frameworks, APIs and tools that can be reused in the project. An example of this is tools that allow to compile, generate, etc. YANG models and how they are integrated in the software.

4.2.2.1 ONOS

The ONOS [ONOS] controller framework has been selected for the implementation of the functionalities required in the SDN controller, both for the optical layer as well as the packet and aggregation layer.

4.2.2.2 OpenDaylight

Alternatively, and to support already existing deployments, OpenDaylight [OpenDaylight] will also be considered as a SDN controller, in view of the existence of devices already integrated in this platform. OpenDaylight (ODL) is a modular open platform for customizing and automating networks of any size and scale. The OpenDaylight Project arose out of the SDN movement, with a clear focus on network programmability. It was designed from the outset as a foundation for commercial solutions that address a variety of use cases in existing network environments.

4.2.3 VIM

4.2.3.1 OpenStack

OpenStack [OpenStack] is Open source software for creating private and public clouds. OpenStack software controls large pools of compute, storage, and networking resources throughout a datacenter, managed through a dashboard or via the OpenStack API. OpenStack works with popular enterprise and open source technologies making it ideal for heterogeneous infrastructure.

OpenStack has been selected as the de facto choice for a METRO-HAUL VIM. It is if course possible that PoC that target other concepts or systems may select another VIM for practical purposes (e.g. if demonstrating application of the METRO-HAUL system to other commercial or available VIMs).

4.2.4 ETSI NVFO

4.2.4.1 OSM

OSM [OSM] is delivering an open source Management and Orchestration (MANO) stack aligned with ETSI NFV Information Models. As an operator-led community, OSM is offering a production-quality open source MANO stack that meets the requirements of commercial NFV networks. OSM has been selected as the reference implementation of a MANO system to be used in METRO-HAUL, as Multi-site NFV orchestrator.

In particular, T4.3 has been focused on the integration between OSM and main METRO-HAUL NFVO, the hierarchical SDN controller and the backend planning tools. This includes the aforementioned feature proposal: Integration of OSM and WIM (https://osm.etsi.org/gerrit/#/c/5945), titled "Enable dynamic connectivity setup in multi-site Network Services", since in a multi-VIM environment, deployments are likely to be geographically distributed across different sites, interconnected through a (wide area) transport network. Until recently, provisioning connectivity across such a transport network has been done in a way that is decoupled from MANO orchestration and is taken as a static input from its point of view. With the scope of METRO-HAUL, we have proposed an integrated WIM solution in OSM. In this regard, an initial Proof-of-Concept is done and the initial design based on the scenario figure 28 is being created.

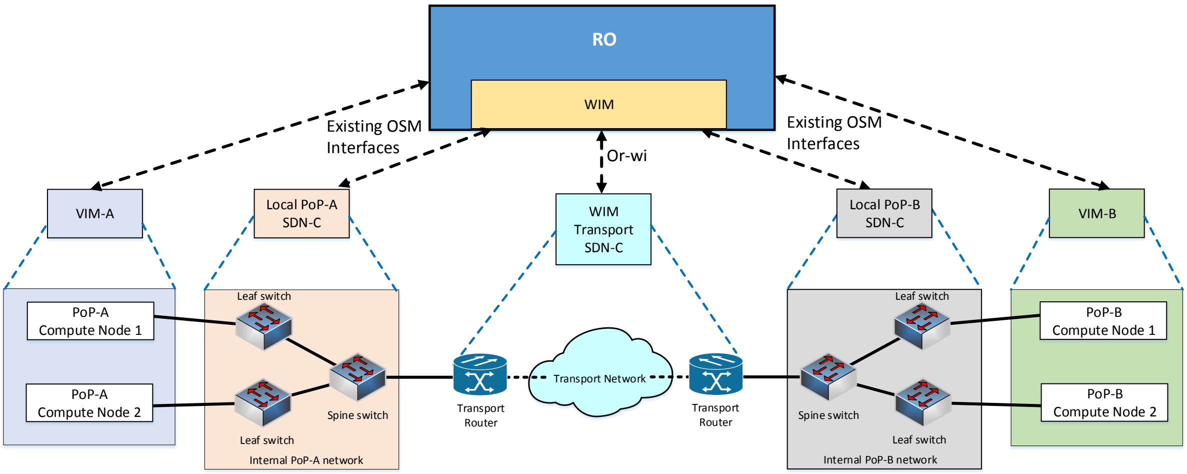


Figure 28. OSM WIM integration scenario

4.2.5 Monitoring and Data Analytics

Monitored and telemetry data exported from the MDA agents is collected by the domain MDA controller and stored into a distributed big data repository. This big data repository requires scalability and high availability without compromising performance. In the context of METRO-HAUL, Apache Cassandra [Cassandra] has been selected because of its capacity to manage massive amounts of data, fast.

To enable centralized data analytics and KDD implementation in the domain controller, a KDD module is defined; it contains KDD processes that execute intensive data analytics tasks, either locally or delegated to a computing cluster. In the context of METRO-HAUL, the Apache Spark [Spark] framework will be considered as it is a fast and general engine for large-scale data processing.

Regarding protocols, IPFIX [IPFIX] for monitoring and gRPC [gRPC] for telemetry have been selected.

4.2.6 Placement, Planning and Re-configuration

4.2.6.1 Net2Plan

Net2Plan (www.net2plan.com) is a free, java-based tool with open-source (BSD 2 Clause) license, founded and developed by UPCT since 2011. Its use is widespread in a number of world-wide Universities and centres for teaching & research, and has also attracted industrial interest.

Net2Plan has been chosen the common framework for network optimization & planning in other H2020 projects (e.g. ACINO, XHaul). As a tool devoted to the planning, optimization and evaluation of communication networks, it has been selected as framework for the development of the back-end section of the Network Planner (Placement, Planning, and Reconfiguration Subsystem – see Sec. 4.4). It will include two main algorithms: NPSO and NRAO. Net2Plan will provide the data structures to support the network abstractions needed by the algorithms and also to support the collection of monitoring historical series needed by the machine-learning algorithms.

Net2Plan may also be used for the front-end of the Network Planner, as all interfaces needed for this front-end will be developed as Net2Plan communication interfaces with the other software components of the METRO-HAUL control plane (i.e. the service orchestrators (e.g. OSM), the network controllers (e.g. ONOS), etc. Net2Plan already models Networks Services, and service chains, easing algorithm development and testing for the new challenges coming from the transport & IT joint network management. In short, Net2Plan will be extended to interface with OSM, the SDN controllers and potentially with monitoring tools to keep a live network view, and provide allocation decisions. The different algorithms developed by the given partners can then be integrated as standard Net2Plan plugins, and then tested in the platform.

In [Mor18] a proof-of-concept in NFV was made where some “Back-end” and “Front-end” functionalities of the network planner were tested and proven: VNF Placement and Scaling Optimizer (NPSO), some of the network planner-related interfaces, IPNFVO and IPVIM, in a not only a preliminary but also in advance versions of such interfaces.

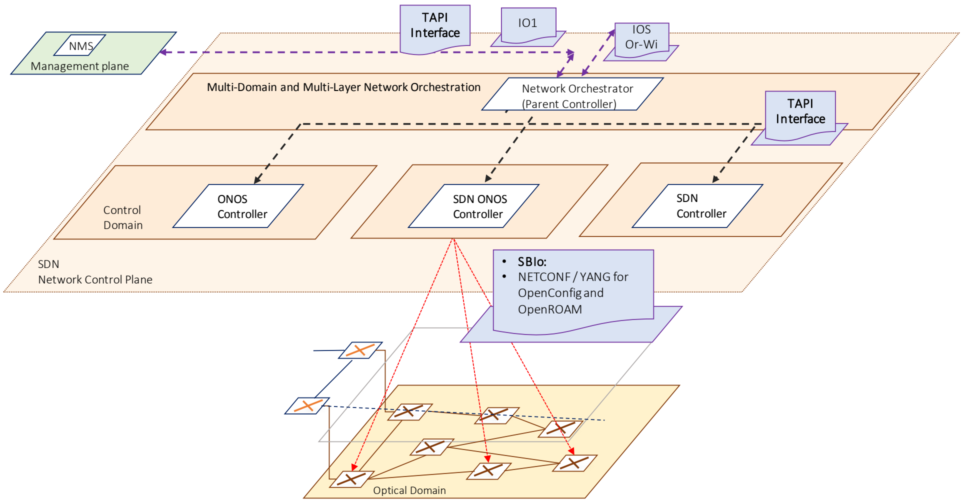
5 METRO-HAUL COM Functional Components

This section enumerates the main components that are being designed and developed in the context of METRO-HAUL. For each component, the tables below provide a description of the component, its role and functions within the COM system and main goals. A Roadmap of the component is also provided, listing key releases and functions, in line with the project and work package schedule. The tables also provide, for each one, a testing methodology and the means of validation, as well as a set of per component KPIs.

For each component we list main involved partners. This does not preclude additional partners contributing to and/or integrating a given component in a broader scope.

5.1 Hierarchical SDN System

Description	This component involves the design and development of the hierarchical SDN system. The METRO-HAUL SDN controller is being developed within the ONOS framework [ONOS], relying on ONOS support for model driven development, which applies both at the NBI and at the SBI levels. The use of YANG is possible to describe devices as well as services or other control and management constructs (topologies, inventory). ONOS Yang tools will be used consistently to register within ONOS service models that are consumed by applications and device models for the actual ONOS devices.

	 *Figure 38: schema of the Hierarchical SDN system.* Figure 38 depicts the main elements of the Hierarchical SDN system. In particular, the system encompasses: **SDN controller for the optical network**. The minimum is to control devices that are either OpenConfig or OpenROADM based. It will be possible to also control a subset of devices if their Yang data model is known and registered. The main target is a single SDN Controller for the set of devices in the domain. The software can accommodate hybrid deployments in which the control of the transceivers is decoupled from the control of the Open Line System. The SBI will be based on NETCONF [NETCONF] over SSH transport. The devices will either be using an OpenConfig device model for platform and terminal device or OpenROADM device version 2.2. The NBI for the SDN controller (and adopted as well for the parent orchestrator) will be using TAPI version 2.0 or 2.2. The implementation will cover mostly the connectivity and topology models. Topology models will be read-only, to be discovered by other subsystems by means of REST operations. The integration with WIM / OSM will also define a dedicated NBI that will need to be implemented. **SDN controller for the packet domains.** METRO-HAUL scope does not include extending the SDN control of packet switched networks, since the current features are considered enough. OpenFlow switches will be controlled from one or more dedicated instances of SDN controller. If applicable, SDN controllers will either implement basic TAPI services to be controlled by the orchestrator or the orchestrator may already consume the exported native interfaces.

	SDN controller for the parent orchestrator. The controller is responsible for the end-to-end service provisioning. The orchestrator will reuse as much as possible developments for the controllers.
Roadmap	The roadmap for the component is aligned with WP4 activities. • M6: PoC demonstrating specific functions (e.g., NETCONF control of devices, selected functions; • M9: Initial tests with TAPI components. Validation of YANG service models (first, simple connectivity services). • M12 (May 2018) : PoC with several NETCONF devices, validation of YANG device models; • M18 (Nov 2018): alpha release, SDN control of optical networks with configurable parameters, networks and devices. Graphical User interface; • M24: (May 2019): beta release of the hierarchical SDN controller; • M36: final release of the hierarchical SDN controller, including validation in demos and integrated scenarios.
Testing Methodology	**Functional tests**: Preliminary tests to be done locally, at the different partners' premises. This includes validation with emulated and real devices, isolated. For this, agents will be used using devices provided with a NETCONF agent, e.g., using ConfD or Net2peer tools; **Additional Integration Tests**: the system will be developed by multiple partners, with well-defined external interfaces. Secured connections (e.g. IPSEC, GRE) tunnels will be established between partners premises to integrate the different components; • Isolated testing of interfaces with e.g. mock agents and CLI commands for TAPI control.
Means of Validation	• Experimental Validation in multi-partners' testbeds; • Lab testing for specific functions; • Integration into WP5 demos.
KPI	The control KPIs are: • Service Setup delay; • Blocking probability in specific scenarios; Estimate on Control plane overhead.
Component KPIs	• Estimate on the number of supported NETCONF sessions; • Sizes of networks; • Memory and CPU requirements.
Other Remarks	• The software will be released as Open Source. A project Gitlab has been setup for the purposes of code sharing; • Developments are complementary and in cooperation with the ODTN

	project. METRO-HAUL source code has been contributed.
Main Involved Partners	**CTTC**, TID, CNIT

5.2 SDN for Passive Optical Networks

Description	SDN architecture for enabling a SDN-compatible control-plane of commercially deployed PONs using standard protocols, without introducing new extensions, to avoid deployment complications. The main building blocks of the platform are: ConfD tool, the PON Configuration Agent (PCA), the PON Network Flow Agent (PNFA) and the PON Controller based on an extension of ONOS controller. The SDN PON architecture is described in detail in Section 3.3.3.
Roadmap	• Implementation of selected functionalities of the involved components to create a Proof of Concept; • Definition of the interface/APIs (e.g. TAPI) toward the parent controller; • Implementation of the interface/APIs and integration with the parent controller; • Integration with other subsystems toward the delivery of a project wise demo in WP5.
Testing Methodology	• **Functional tests:** execution of a set of functional tests to ensure that the implemented functions are working properly and their outcomes are in line with the project objectives; • **Integration tests**: execution of a set of integration tests to validate the appropriate integration of the SDN controller with the other subsystem of the project (e.g. Orchestrator); • **Performance tests:** Execution of a well-defined set of tests to validate the performance of the platform under a set of project scenarios and against a set of predefined KPIs.
Means of Validation	• Experimental validation using a testbed including (among others) commercial PON equipment (e.g. GPON); • Testing of the different subsystems of the architecture; • Testing of the different functionalities of the architecture; • Integration into the WP5 demos.

KPI	• Control plane latency:
	    o Latency in PCA for PON reconfiguration;
	    o Latency in PCA for PON reconfiguration causing DATA interruption;
	    o Total latency in PCA;
	    o Total NETCONF configuration cycle (request-update-response).
	• Data plane latency:
	    o Data interruption time using different traffic types (UDP and TCP)

	and service types (video streaming and FTP download/upload).
	• Scalability
	  ○ Number of supported NETCONF sessions;
	  ○ Number of supported OF flows;
	  ○ Number of configuration actions (assuming a NETCONF session may include several configuration actions on the actual devices);

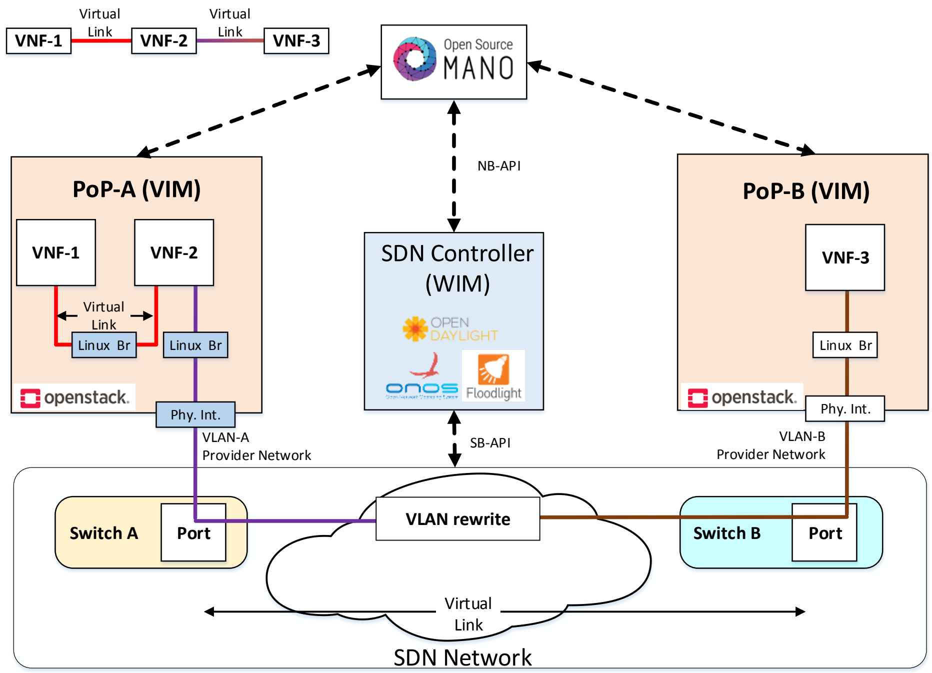
Main Involved Partners	**OLC**

5.3 SDN agents for O-NEs, OIEs and OSTs

Description	The ONOS controller of the optical domain will utilize NETCONF/YANG to control each optical device as explained in Sec. 2.2. Thus, each optical device has to be equipped with a NETCONF server based on the YANG model description of the specific device. A NETCONF server implementing OpenConfig and OpenROADM YANG models, will be therefore implemented and deployed for each optical device considered in the METRO-HAUL architecture.
Roadmap	• M2 setup of a NETCONF server able to manage generic YANG models; • M6 implementation of NETCONF agents for XPONDERs and ROADMs using generic YANG modelling, i.e., not fulfilling OpenConfig/OpenROADM specifications; • M12 (associated with milestone M4.2) implementation of basic version of XPONDER/ROADM agents fulfilling the main OpenConfig/OpenROADM specifications; • M18 (associated with milestone M4.3) implementation of stable version of XPONDER and ROADM agents fulfilling OpenConfig/OpenROADM specifications; • M24 (associated with milestone M4.4) hierarchy of agents as described in section 2.2.
Testing Methodology	• **Functional tests**: Preliminary tests to be done locally, at the premises of the developing partners. This includes validation of communication between the ONOS-based optical controller and the developed agents; • **Additional Integration Tests**: the agents, and related optical controller drivers will be released and tested at other partners' premises.
Means of Validation	• Experimental Validation in single-partner testbeds; • Experimental Validation in multi-partners' testbeds; • Lab testing for specific functions; Integration into WP5 demos.
KPI	• Time required to perform each configuration including only data plane

	delays (i.e. configure a new cross-connection on a ROADM O-NE); • Time required to perform each configuration including control and data plane delays (i.e. configure a new cross-connection on a ROADM O-NE); • Scalability in terms of number of NETCONF agents that can be used in the network without degradation of controller performance.
Main Involved Partners	**CNIT**, CTTC

5.4 Integrated SDN and ETSI NFV/MANO system

Description	Management and Network Orchestration of Virtualized Network Services running over NFVI (Openstack) with an integrated WIM component and a WIM SDN Controller to create L2 network slicing over the WAN Infrastructure, as depicted in Figure 39.  *Figure 39: schema of the Management and Network Orchestration plane*.
Roadmap	• Creation of a Proof of Concept; • Design of a generic solution to cover all the networking cases currently supported in OSM; • Finalizing interfaces/APIs to integrate the WIM solution with Net2Plan for planning and placement of VNFs; • Using the tool to deliver the demos in WP5.

Testing Methodology	• Create Network Service Descriptors covering cases:
	  ○ Explicitly selecting VIM Network Name while deploying a network service;
	  ○ Selecting only physical network name and segmentation id while

	deploying a network service; ◦ Allow OSM to create the default virtual links. • In all the three cases, instantiate and deploy multiple Virtualized Network Services using single instance of OSM connected to multiple VIMs(Openstack) having an underlying L2 connectivity; • Check if the network service is deployed and operational.
Means of Validation	• The OSM must be able to insert appropriate OpenFlow rules in the SDN switches to create the network services chain. • The traffic must flow from one end to the other in the network service irrespective of the VIM deployed.
KPI	• Time required to instantiate a network service; • Time required to create and install all required flow entries.
Main Involved Partners	**UNIVBRIS**, TID, CTTC, UPCT

5.5 System for network virtualization and slicing

Description	Integrated environment for VNF deployment across data-centers using single or multiple OSM instances. The OSM must orchestrate Network Functions and create service chain between them automatically using the ETSI NFV descriptors. Non-interfering co-existence of multiple network service chains between the two of more data-centers would reflect the network slicing capabilities.
Roadmap	• Infrastructure tools (OSM, Openstack) integration; • Creation of appropriate ETSI NFV Network Service Descriptors; • Deploy VNFs over the established infrastructure and chain them to run over the network links simultaneously; • Leverage the output from component 5.4 (WIM) to deploy it over the multiple data-centers; • Integrate solution for WP5 demonstrations.
Testing Methodology	• Create descriptors for multiple chains of Virtualized Network Services using OSM; • Instantiate and deploy the Network Services using OSM; • Deploy multiple Network Services over the same underlying physical network to test the network slicing capabilities; • Traffic must flow from end-to-end without interfering with other network services.
Means of Validation	• The VNFs must be deployed successfully using OSM; • The traffic must flow from one end to the other in the Network Service (over VNF/PNF) irrespective of the VIM deployed; • Multiple Network Service chains should be operational over the same link at the same time.

KPI	NS instantiation time.
Main Involved Partners	**UNIVBRIS**, TID, UPCT, CTTC

5.6 Monitoring and Data Analytics subsystem

Description	The MDA subsystem includes MDA agents running close to the network nodes, and a big data centralized MDA controller running in the control and management plane. MDA agents are multi-node agents that collect monitoring data records from configured Observation Point (OP) in the nodes, which can be used for KDD to proactively implement local control loops to tune parameters in the network devices and to notify the MDA controller about network anomalies and degradations. The MDA agent has been conceived to support multilayer disaggregated scenarios. An auto-discovery function allows retrieving the available monitoring capabilities for each node, as well as already configured OPs. This functionality is of paramount importance in brown-field scenarios, where the network is already in operation. The MDA controller collates measurements from the active OPs in the network and stores them in a (big data) repository. The protocol used for monitoring configuration is RESTCONF that it is based on an extended YANG data model. In addition, the IPFIX protocol used for monitoring purposes, has been extended by defining new templates, specifically to convey measurements from the optical data plane. The MDA controller implements a set of functionalities, including: i) the configuration of the monitoring and telemetry, defined in the new YANG data models, ii) collating measurements from the MDA agents and storing them into a scalable multi-master database (we use Apache Cassandra), and iii) applying data analytics algorithms to the monitoring data. When monitoring data or notifications are received in in the MDA controller, a process manager module is notified, and the corresponding KDD process is executed; KDD processes can run locally or in a cluster using big-data processing engines, such as Apache Spark. In addition, operational databases (e.g., topology and LSP databases in the SDN controller) synchronization is supported. With operational databases synchronized, the MDA controller is able not only to collate measurements from the nodes, but also to correlate them with operational data, e.g., with the route of an LSP for failure localization purposes. Therefore, more sophisticated procedures can be developed correlating measurements with topology and LSP data.
Roadmap	The development of the MDA controller and of the interfaces with the MDA

	agents are part of this WP, whereas the MDA agents are being developed in WP3. The roadmap of the MDA developments is as follows: • M04: Initial designs; • M08: First prototype showing some integration with data plane components released. Identification of improvement points; • M16: Second prototype showing some integration with COM components released. Identification of improvement points; • M24: Field tests on available equipment. First developments tested and validated. Identification of improvement points; • M30: Final development released; • M36: Field tests on available equipment. Final development tested and validated.
Testing Methodology	A testing setup has been deployed at UPC premises. The testbed includes several MDA agents and one MDA controller running in VMs. Monitoring generators are used to generate monitoring measurements that are collected by the MDA agents. Those measurements are aggregated and sent toward the MDA controller, where monitoring data can be visualized. IPSec tunnels will be used to connect the local testbed with other partners. Specifically, the M-COM interface will be tested though a tunnel established with CTTC, monitoring interfaces will be tested using tunnels with CNIT and Naudit. In addition, other tunnels can be established ad-hoc to test other interfaces.
Means of Validation	A very first prototype of the MDA controller together with MDA agents just collecting measurements (i.e., without configurations capabilities) was demonstrated at OFC 2018 [Vel18o]. Integration with Naudit passive probes generating packet traffic measurements, OSA generating optical spectrum captures, and transponders generating BER measurements, was exhibited; monitoring data was graphically visualized. A second prototype of the MDA controller showing integration with the SDN controller is targeted for ECOC 2018 demonstration. Integration with other systems in the COM through the M-COM interface is also targeted. The IO4 interface will enable external systems to retrieve data. A demonstration of an application running on top of METRO-HAUL COM will be carried out; the application will be able to retrieve monitoring data for their specific resources, including network (VLAN) and IT (VMs), and application logs.
KPI	• Time to detect an anomaly or degradation • Amount of data being conveyed to the centralized MDA controller
Main Involved Partners	**UPC**, NAUDIT, CNIT, CTTC

5.7 Placement, planning and reconfiguration subsystem

Description	As pointed out in Section 3.6, the aim of the Placement, Planning, and Reconfiguration Subsystem (namely Network Planner) is to optimize the resource allocation in the optical metro network to effectively provision VNFs in specific computing nodes considering heterogeneous requirements. The Network Planner is divided in a Front-end containing the interfaces (detailed in Section3.6.1) that allow the exchange of information necessary for the planning tool (the Back-end). The open-source Java-based Net2Plan tool has been chosen as the planning tool / Back-end for placement, planning and reconfiguration of: VNFs, IT and network resources. This including off-line algorithms for traffic-based dynamic metro resource activation, with correlation of service traffic monitoring, resource availability and service requirements. Net2Plan runs two main algorithms: VNF Placement and Scaling Optimizer (NPSO) and Network Resource Allocation Optimizer (NRAO), as described in Section 3.6.2.
Roadmap	The development of the placement, planning and reconfiguration subsystem and the interfaces with the Network Planner are a relevant part of WP4. The proposed roadmap follows: • M04. Definition, development and test the algorithms; • M08. Definition and development of the IPNFVO and IPVIM interfaces. Test VFN optimal instantiation; • M14. Definition and development of the IPSDN interface. Test transport flow commanded by the Network Planner algorithms results; • M20. Joint IT and transport integration. Merge algorithms and interfaces. Test joint subsystem; • M26. Deployment of NSD considering algorithms results; • M30. Preliminary deployment of an advance version of the Network Planner to the METRO-HAUL Network; M36. Final integration. Fix bugs and increase performance to achieve the final KPIs.
Testing Methodology	The testing procedure for this set of tasks can be split into three main groups: (i) testing the planning tool software, i.e., Net2plan; (ii) testing the interfaces which involve the Network Planner; (iii) testing the NPSO and NRAO algorithms: • Testing Net2plan: the development of the releases of the planning tool has an inherent testing process, mainly based in JUnits testing methods. Such test procedures are automatically executed during the development and before releasing a new version of Net2plan. Besides, Net2plan is a globally used tool that the use itself of Net2plan by the community assures the rapid identification of problems to be solved in next releases. • Testing Interfaces: - IPNFVO: This is a regular REST/API interface which implies a server-client deployment. In this way, the methodology suggested to test it is proving

	the correct interconnectivity of both agents bidirectionally put the focus on assuring the correct transmission and reception of the data from both sides and test the functionalities required for the given purposes. The NFV-O chosen (OSM, release three) provide not only a server, but also a suggested client to implement. The structure of this client is used in this interface and determine the commands (or capabilities) to test. Common acknowledge messages will be sent until successful communication is achieved. - IPSDN: much like to the IPNFVO, this interface is REST/API-based and the SDN controller chosen to handle with the related tasks, ONOS, provides a well-documented set of instructions to use the API. It is assumed as a tested interface. In case further effort are needed to provide new functionalities to the SDN controller or WIM, such functionalities have to be implemented in the API and tested. - IPVIM: The OpenStack REST/API is widely used and tested by the OpenStack community. The network planner uses this interface to exchange data with the VIMs. • Testing Algorithms: a usual method to test the algorithms (NPSO and NRAO) is to assume a set of expectable and feasible results (provided by well-known benchmark algorithms) given a set of inputs in and compare the outputs of the execution of the algorithms to the expectable ones.
Means of Validation	The Network Planner will be validated defining a set of expected outcomes and performing the testing methods until the desired outcomes are obtained, for its three main components: • The programmatic interfaces in the Network Planner front-end will be validated establishing the required connectivity between the software modules involved (e.g., the IPSDN will be validated establishing proper connectivity between the planning tool and the WIM SDN controllers). • The planning tool (back-end) will be validated through a set of sample cases that will be carried out during testing until the expected (and correct) outcome is achieved and revealed through its interfaces (e.g. JAVA console module, automatic report extension or graphical user interface). • The algorithms will be validated analysing the obtained results in large topologies and comparing them with the expected outcomes. In case the results are within a valid confidence interval provided by the benchmarks for a wide range of tests, the algorithm is considered as valid.
KPI	• NPSO and NRAO algorithms execution time; • Number of operator service-chain-based requests; • Interfaces communication delay.
Main	**UPCT**, **POLIMI**, TID-JRU(UC3M)

Involved Partners	

5.8 Service and traffic monitoring system

Description	In order to check that the communication equipment developed in the project is working correctly, it is necessary to monitor it, both actively and passively. This element will be in charge of monitoring the system at network layer, watching which service is being provided by the underlying layers. The obtained measurements will be made available to the monitoring and data analytics system, which can also request active measurements. These elements are needed to achieve objective 6 of the project (to design a monitoring with big-data analytics framework supporting cognition). With respect to the active probes, the following use cases are foreseen: 1. Periodic active monitoring of links. In this way, the state of these links is assessed on every time interval. If problems are detected at packet level, this information can be used by the monitoring and data analytics system to locate and correct a failure at the optical layer. 2. Under demand. The monitoring and data analytics system can also request an active probe to check and locate problems at layer 2. For instance: a. Creation of new lightpaths. Once the lightpath has been created, the active probe checks its behaviour transmitting layer 2 frames over the lightpath. b. Analysis of links and correlation with the optical layer. In this case, a failure in the optical layer has been detected and it is necessary to check its impact on layer 2 transmission.
Roadmap	The development of an active probe at 100 Gbps is part of this WP, to be covered within T4.2.2 subtask, whereas a passive probe is being developed in WP3. The roadmap of these developments is the following: • M12: Initial designs; • M18: First developments released; • M24: Field tests on available equipment. First developments tested and validated. Identification of improvement points; • M30: Final development released; • M36: Field tests on available equipment. Final development tested and validated.
Testing Methodology	To test our passive probes, it is necessary to generate traffic at line rate (100 Gbps is our objective), which will be captured by the probe. Thus, we can check that the probe is receiving all sent traffic, and that the time series are similar to what is really transmitted. As pointed out in Section 2.4, a very precise FPGA-based implementation that replays network traces is feasible for this testing, but the amount of traffic to be replayed will be very small for a 100 Gbps probe (in

	the units of GBytes). A less precise but more affordable approach makes use of a software solution based on DPDK. Testing of active probes will be based on the methodology presented in [Rui16]. In this case, the most challenging part is to calibrate the measured delay, due to the uncertainty of the delay that is generated by the NIC transceivers and their elastic buffers. On the other hand, the FPGA implementation provides deterministic results that can be measured with the number of cycles that a task takes to be completed. FPGA development tools such as ILA (Integrated Logic Analyzer, formerly known as ChipScope) provide also means to debug and measure how the implementation is working on.
Means of Validation	A first prototype of our testing development for passive probes was shown at OFC 2018 [Vel18o], sending traffic simulating the behaviour of a network during one day, modifying at regular intervals the data rate. If we use an array of NVMe drives, our current developments based on DPDK show that we can reach 75 Gbps when reading from them network traces and replaying with a Mellanox NIC. We think we can reach 100 Gbps if we work on it during the rest of the project. This improvement is not trivial, because to achieve this goal we need to use more than one transmission queue, which can disorder packets. To validate our active probes, we will need a traffic emitter and a traffic receiver. The development card we are targeting is the Xilinx's VCU-118, which provides two QSF28 interfaces. If necessary, other 100 Gbps interfaces can also be plugged to that platform through the FMC+ connector. Then, we will use the same FPGA card to send and receive the traffic, which would provide a very precise synchronization between both processes, and let us calibrate the measurements if we measure a loopback link.
KPI	As exposed in the DoW, the main KPI is MH4 – WP4 development of end to end SDN controlled application management combined with WP5 vertical demonstrations. Our specific KPIs for passive probes are: • Traffic capture at line rate speed; • Generation of time series of packets and bytes in a 100 Gbps link with second granularity. Our specific KPIs for active probes are: • Sending packet trains at 100 Gbps; • Receiving packet trains at 100 Gbps; • Measuring such packet trains at line rate speed.
Main Involved Partners	**NAUDIT**

5.9 SDN Application for managing spectrum fragmentation in a multilayer optical network

<table>
<tr><td>Description</td><td>Traffic generated by 5G will have stringent requirements in terms of bandwidth, delay and QoS. Moreover, it will be highly dynamic and therefore, the optical network has to be able to use as better as possible the avilable optical resources. For this reason, intelligent optimization modules are required at the optical layer (including both the OTN and WDM layers) for reallocate resources without impacting the service QoS.

To this aim, the proposed Defragmentation application will monitor the fragmentation state of the network at the WDM layer and point out the links that present a high fragmentation state. A defragmentation operation is run following network operator specification that will be specified using a web-application (e.g., defining the maximum tolerated link fragmentation and the minimum time interval between two consecutive reconfiguration).

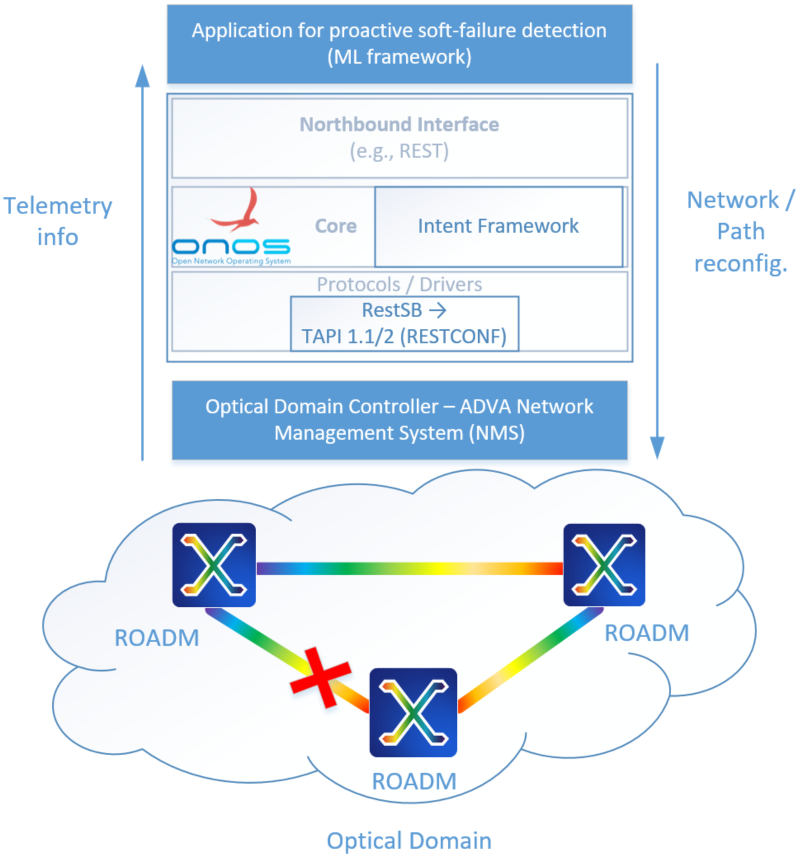
When a reconfiguration operation is required, the defragmentation routine evaluates the interest of performing a network reconfiguration by firstly reconfiguring at the OTN layer the services with high QoS (i.e., services that cannot be interrupted) and then providing the reconfiguration at the WDM layer of remaining services. Only if the network fragmentation decreases thanks to the new resource allocation, the network reconfiguration will be actually started.

This work is divided into two main activities:
1. Define a web-application to monitor the state of the network, both at the OTN and WDM layers, and drive the reconfiguration operation to be performed by the SDN controller;
2. Provide a defragmentation routine to scans the routed traffic over HO-ODUs and decide if it can be rerouted and at which layer.</td></tr>
<tr><td>Roadmap</td><td>The development of the application is divided into two separate parts, one concerning the web-application and the other the defragmentation routine. The roadmap of the developments and release of these parts is the following:
• M18: provide a web-application interface that to monitor the network fragmentation following the fragmentation metric chosen by the network operator. This phase will include the definition of the inputs/outputs data required by the proposed application to interact with the network;
• M20: delivery of a dll with the management of the optical defragmentation for only L0 HO-ODUs;
• M28: delivery of a dll with the management of the optical defragmentation with rerouting of L1 and L0 HO-ODUs;
• M30: Final development released;
• M34: Final development tested and validated (check that the</td></tr>
</table>

	reconfiguration is performed following the user rerouting preferences); • M36: Field tests on available equipment.
Testing Methodology	To test the proposed application, it is possible to generate different traffic matrices, i.e., dynamic services (L0 and / or L1 services) with diverse duration / distribution / QoS and the user can define constraints on the reconfiguration events. After rerouting, it will be verified vthat the rerouting is done by respecting the specified QoS constraints.
Means of Validation	To allow the validation of the proposed application, the fragmentation state of the network will be periodically measured. If the fragmentation overpasses a defined threshold, it is possible to check that the reconfiguration happens with the periodicity specified by the user. After each reconfiguration, the list of rerouted HO-ODUs and their services is provided to the user, so that it is possible to check that the rerouting respects the service constraints. A tool able to read provided reports and check that the user / service constraints are respected will be provided.
KPI	The specific KPI of this application are: • Capture of the fragmentation state of the network • Defragmentation reconfiguration output o List of reconfigured L0/L1 HO-ODU at every reconfiguration state o List of the reconfigured L0/L1 services
Main Involved Partners	**NI**

5.10 SDN application for proactive soft-failure detection

Description	With increasing network complexity and heterogeneity, business economics dictate the need for improved asset management to reduce downtime and improve resource usage. Typical examples of network faults may include cooling unit failure, laser degradation, subsystem control unit failure, etc. Early detection of equipment failure states and consequent remedial actions can prevent network downtime and enable scheduled preventive maintenance. Many commercial equipment tolerates some errors until automatically tearing down the connection when some system thresholds are exceeded. While a restoration procedure could be initiated to recover the affected traffic, it would be desirable to anticipate such events and taking relevant actions.
Roadmap	In context of T4.2 we aim to demonstrate an analytics application enabling cognitive network assurance through proactive soft-failure detection. In particular, we will cater real-life network fault use cases and identify them using

	the ML framework. The roadmap follows the following timeline, M06: Initial requirements; M12: Design and proof-of-concept; M22: Prototype with initial integration with ONOS; M26: Functional and integration tests; M30: Integrated demo with WP5 use case.
Testing Methodology	Figure 40 shows the overall architecture of the discussed framework which will be used to develop and test our approach.  *Figure 40. SDN application for proactive soft-failure detection* Going from bottom to top, optical transport network configuration and telemetry information from the optical domain is extracted inside the Network Management System and made available to the ONOS controller through a RESTCONF interface. The information is modelled according to the ONF Transport API (TAPI) and will support either version 1.1 or v. 2. Relevant telemetry information is provided northbound to the application for proactive soft-failure detection. Upon detection of a soft-failure, the application can proactively submit an application centric intent (ACI) to the northbound interface of the ONOS controller. The orchestrator routes the request through the intent framework, compiles the submitted intent and selects the actions that need to be taken to satisfy the intent. Since, in the case of an optical network, multiple devices are controlled through a single entry point, i.e., the mediation layer, the compiler

	needs to be able to create a special intent type, called domain intent. Those domain intents are then installed through a protocol, e.g., RESTCONF, and the corresponding driver, e.g., TAPI. The communication with the optical devices is done through ADVA's NMS, which exposes an experimental TAPI interface. • **TAPI Driver** Even though ONOS provides numerous implementations of protocols out of the box, they only cover the common behaviour in most cases. To apply a protocol based on a YANG description, it is currently necessary to extend one of those protocols with the details of a particular implementation. The REST southbound protocol was used as a basis and starting point. One advantage is that it supports intermediate controllers. The TAPI is used for topology discovery, which includes nodes and links, as well as service setups. The topology is exposed by the underlying controller, the individual elements are extracted and then exposed to ONOS by the driver. Most information is directly mapped to available entities; some additional data is needed to set up connections. This information is stored in the form of annotations, compliant with ONOS' architectural requirements. • **Optical Domain Intents** Domain intents are an important prerequisite for installing services in the optical domain e.g. to setup of lightpaths through a network. Optical intent concept is already available in ONOS, but important extension is required to fulfil the METRO-HAUL scenario. Besides configuring analog parameters, like launch power, the process itself needs a particular structure, e.g., equalization and hop-by-hop lightpath setup. Additionally, a local controller might have a better knowledge about the domain because of additional information or a view without abstractions. To handle intents for domains a specialized processing needs to be applied. A domain intent contains the ingress and egress points and might also contain information on a preferred path inside the domain. It is assumed that the local controller is able to do all the required steps to fulfil the incoming requests. Without an included path the domain controller can compute a valid path for provisioning. In the second case, it needs to verify the feasibility of the requested path / configuration before installation. In order to activate the domain processing an NBI intent needs to indicate this via an internal flag (e.g., DomainConstraint). • **Soft-failure application** The machine learning framework consists of several stages including data access through the RESP API, pre-processing features, neural networks for computational mapping of inputs to failure probability, and post-processing to trigger intents.
Means of Validation	The initial architecture of our approach was shown in [Raf18], identifying commercial fault scenarios. We will extend that to the WP5 vertical demonstrator use case. As shown in Figure 40, the presented system architecture can be implemented

	and evaluated with commercial hardware in a test-bed. Minimum requirements on validation include an optical ROADM ring (e.g., ADVA FSP3000 ROADMs or ADVA MicroROADMs), a network management system (domain controller), a custom ONOS implementation and the SDN Application for proactive soft-failure detection.
KPI	• Receiving PMs via the REST API and sending ML outcomes; • Rate of correct and incorrect soft-failure detection; • Control plane reaction time to a soft-failure detection; Network reconfiguration time in case of soft-failure detection.
Main Involved Partners	**ADVA**

6 Conclusions

As stated in the description of work, D4.1 reports the initial design of the METRO-HAUL control architecture, justifying design choices and deployment models. D4.1 covers a preliminary specification of the interfaces in view of integration and interoperability, listing selected components to be developed and to be integrated in demonstration scenarios. For each component, D4.1 details its implementation roadmap, functional regression and integration tests, and the means to validate the implementation, including control KPIs.

This deliverable briefly presented the goals of the control, orchestration and management system of the METRO-HAUL project, insisting on key innovations such as the extended use of telemetry, monitoring and data analytics enabling artificial intelligence and machine learning applications, use cases and targeted services. The COM system is introduced for the dynamic deployment of services over the METRO-HAUL infrastructure, encompassing, notably, multiple cloud enabled optical nodes (termed AMEN and MCEN in the document) interconnected by a SDN controlled disaggregated optical network. We enumerate the design considerations for METRO-HAUL, and the different systems that compose the architecture: i) At the lowest level, the control of the optical layer, the control of the intra-DC network within each METRO-HAUL macro node and the control of the PON network aggregating users traffic; ii) the hierarchical system for network orchestration allowing the end-to-end control of the network architecture; iii) the integration with IT and cloud resources based on ETSI NFV MANO and how METRO-HAUL supports the concept of slicing; iv) the Monitoring and Data Analytics (MDA) subsystem and v) the Placement, Planning, and Reconfiguration subsystem (NP for short). The section concludes identifying the main interfaces between functional components. Each interface has its macroscopic requirements and relevant considerations. Finally, main METRO-HAUL COM components are listed.

Let us note the following: i) the architecture is considered stable, at least from a functional, high level point of view. We foresee that minor modifications will be needed to include emerging requirements or new use cases (for example, to accommodate for sub-controllers in a hierarchical setting); ii) the KPIs have been defined per component and are significantly specific. It is a work in progress with WP2 to define the METRO-HAUL architecture and to elaborate of the KPIs and how different KPIs are composed into meaningful KPIs and the relationship with project-wide KPIs and 5GPPP program KPIs.

The COM architecture has been presented in several events and conferences, including, for example, the presentation “METRO-HAUL: Enabling 5G Services across Disaggregated Multi-Layer Transport Networks: Ongoing Challenges for NFV Orchestration across Metro Networks” within the Zero Touch & Carrier Automation Congress; the paper "Dichotomy of Distributed and Centralised Control: METRO-HAUL, when control planes collide for 5G networks" to be prepared for Elsevier; the 5GPPP journal section related to METRO-HAUL or the contributions to IETF 101 YANG models and T-NBI meetings. It is also worth noting the publication [Vel18b].

The main work now is devoted to implementation of PoC. At this stage, development of WP4 components is being done to validate proof-of-concepts, demonstrate existing baseline functionality and testing selected open source software and frameworks. Notably, let’s mention an analysis of OpenConfig Models for Disaggregated Optical Networks; YANG Models of services and North Bound Interface requirements for partial disaggregation use cases enabling alien wavelength restoration, and the investigation of the use of MEF Legato interface for service provisioning. There are standardization contributions like IETF WSON and Flexi-Grid YANG models or the IETF drafts. The initial PoCs have had a huge impact at the OFC2018 conference, a key event relevant to METRO-HAUL, with very high impact in terms of regular papers and SDN/NFV demos to the Open Platform Summit. Of special interest are the 4 demos conducted at OFC showing different parts of the WP4 COM architecture: SDN control of disaggregated optical networks; Hierarchical SDN Control, Monitoring and ML and NFVO/OSM integration: a) “Fully Disaggregated ROADM White Box with NETCONF/YANG Control, Telemetry, and Machine Learning-based Monitoring”; b) “CASTOR: An Architecture to Bring Cognition to Transport Networks”; c) “Towards IP & Transport Network Transformation Using Standardized Transport NorthBound Interfaces” and d) “Joint Optimal Service Chain Allocation, VNF instantiation and Metro Network Resource Management Demonstration”.

6.1 Main take-away messages

- METRO-HAUL is developing a dynamic, intelligent control architecture which allows 5G slices to be provisioned E2E, taking account of their individual KPIs, in scenarios involving multiple network segments and layers, spanning multiple geographical data centre (DC) locations and addressing resource heterogeneity including, notably, the optical transport. Without this architecture and Control Plane technology, network resources for future 5G services would require enormous over provisioning, both of optical transport bandwidth across metro and core networks, but also of edge DC resources such as compute and storage.
- METRO-HAUL aims at highlighting the benefits of applying a systematic and unified approach based on model driven development for the SDN control of multilayer disaggregated and open transport networks, while allowing flexibility in deployment choices, extensibility for the integration of new technologies and agility in migration processes without vendor lock-in.
- METRO-HAUL has implemented a telemetry / monitoring framework which provides a global, real-time view of the E2E 5G network performance without requiring a huge data

communications channel bandwidth or storage requirements. This new technology will enable 5G services to be set up and then maintained reliably, whilst providing pro-active early warning of reliability issues. Machine Learning within the decision engine allows this new METRO-HAUL technology to continually learn and improve as real network data is collected.

- METRO-HAUL has also integrated such dynamic control systems, telemetry/monitoring frameworks and Machine Learning approaches with state-of-art advanced planning, placement and re-optimization/re-configuration tools, enabling holistic (joint) optimization across heterogeneous resources.

7 List of acronyms

3DoF	Three(3) Degrees of Freedom
3GPP	Third Generation Partnership Project
5G PPP	5G Infrastructure Public Private Partnership
6DoF	Six(6) Degrees of Freedom
ABNO	Application Based Network Operations
ACI	Application Centric Intent
ACTN	Abstraction and Control of Traffic-Engineered Networks
ADAS	Advanced Driver Assistance Systems
AI	Artificial Intelligence
AMEN	Access-Metro Edge Node
API	Application Programming Interface
AR	Augmented Reality
ARP	Address Resolution Protocol
ASON	Automatically Switched Optical Networks
AWG	Arrayed Waveguide Grating
BBF	Broadband Forum
BBU	Base Band Unit
BER	Bit Error Rate
BSS	Business Support System
BVT	Bandwidth Variable Transceivers
CAGR	Compound Annual Growth Rate
CBR	Constant Bit Rate
CCTV	Closed Circuit Television
CDB	Core Engine Database
CDN	Content Delivery Network
CLI	Command Line Interface
CloudCO	Cloud Central Office
CN	Core Network
CNC	Customer Network Controller
CO	Central Office
COM	Control, Orchestration and Management
CORD	Central Office Re-architected as a Data-center
CoS	Class of Service
CP	Control Plane

CPE	Customer Premises Equipment
CPRI	Common Public Radio Interface
CPS	Cyber Physical System
CPU	Central Processing Unit
CRC	Cyclic Redundancy Code
CriC	Critical Connections
CU	Centralized Unit
DA	Data Analytics
DASH	Dynamic Adaptive Streaming over HTTP
DB	Data Base
DC	Data Centre
DGD	Differential Group Delay
DM	Data Mining
DP	Data Plane
DPDK	Data Plane Development Kit
DSP	Digital Signal Processing
DTG	Disaggregated Transponder Chips
DU	Distributed Unit
DWDM	Dense Wavelength Division Multiplexing
EB	Exabyte
E-CORD	Enterprise CORD
EDFA	Erbium Doped Fiber Amplifier
eMBB	Enhanced Mobile Broad Band
EON	Elastic Optical Network
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
FLOPS	FLoating point Operations Per Second
FMC+	FPGA Mezzanine Card plus
FPGA	Field Programmable Gate Array
GMPLS	Generalized Multi-Protocol Label Switching
gNMI	gRPC Network Management Interface
GPON	Gigabit Passive Optical Network
GPS	Global Positioning Sysmte
GPU	Graphic Processing Unit
GRE	Generic Routing Encapsulation
gRPC	Google's Remote Procedure Call
GSMA	Global Mobile Supplier Association (Industrial Association)
HD	High Definition
HO-ODU	High Order ODU
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure-as-a-Service
IETF	Internet Engineering Task Force
IFA	Interfaces and Architecture
IHSDN	Interfaceparent controller (Orchestrator) and domain controllers
ILA	Integrated Logic Analyzer

IO2	Operator – VIM interface
IO3	Operator – NFVO (OSM). Based on OSM interfaces and APIs.
IO4	Operator – Monitoring
IoE	Internet of Everything
IOS	Interface between Orchestrator – WIM [NFVO-TSDN, follow OSM]
IoT	Internet of Things
IPFIX	Internet Protocol Flow Information Export
IPNFVO	NFVO – Planning Tool – Interface for planning and placement
IPON	Interface to the SDN PON controller – Controller Specific
IPSDN	SDN – Planning Tool – Interface for planning (SDN)
IPSEC	IP Security
IPVIM	VIM – Planning Tool – [Complements IPNFVO]
IT	Information Technology
ITS	Intelligent Transportation System
JSON	JavaScript Object Notation
KB	Kilobyte
KDD	Knowledge Discovery from Data
KPI	Key Performance Indicators
L2VPN	Layer2 VPN
L3VPN	Layer3 VPN
LSP	Label Switched Path
MAC	Medium Access Control
MANO	Management and Orchestration
MB	Megabyte
MCEN	Metro-Core Edge Node
MCOM	Interface monitoring system - SDN controller (parent) or the NFVO
M-CORD	Mobile CORD
MDA	Monitoring and Data Analytics
MDSC	Multi-Domain Service Coordinator
mIoT	Massive Internet of Things
ML	Machine Learning
MP	Management Plane
MPD	Media Presentation Description
MPEG	Moving Picture Experts Group
MPLS	Multi Protocol Label Switching
MPLS-TP	MPLS Transport Profile
MR	Mixed Reality
MSA	Multi-Source Agreement
NAT	Network Address Translation
NBI	North-Bound Interface
NE	Network Element
NFV	Network Function Virtualization
NFVI, NFV-I	NFV Infrastructure
NFVO, NFV-O	Network Function Virtualization Orchestrator
NGMN	Next Generation Mobile Networks (Industrial Association)
NIC	Network Interface Card

NMS	Network Management System
NP	Network Planner
NPSO	VNF Placement and Scaling Optimizer
NRAO	Network Resource Allocation Optimizer
NS	Network Service
NSD	Network Service Descriptor
NSI	Network Slice Instance
OAA	Observe-Analyze-Act
ODL	OpenDaylight SDN Controller
ODTN	Open and Disaggregated Transport Networks
ODU	Optical Demultiplexing Unit
OEM	Original Equipment Manufacturers
OF	OpenFLow
OFC	Optical Fiber Communication
OIE	Optical Infrastructure Element
OLS	Open Line System
OLT	Optical Line Terminal
ONAP	Open Network Automation Platform
O-NE	Optical Network Element
ONF	Open Networking Foundation
ONU	Optical Network Unit
OOPT	Open Optical & Packet Transport
OP	Observation Points
OSM	Open Source MANO
OSNR	Optical Signal To Noise Ratio
OSS	Operations and Support System
OSS	Open Source Software
OST	Optical Sub-System
OTDR	Optical Time-Domain Reflectometer
OTN	Optical Transport Network
OvS	Open vSwitch
OWD	One Way Delay
PaaS	Platform-as-a-Service
PCA	PON Configuration Agent
PCE	Path Computation Element
PDL	Polarization Dependent Loss
PER	Packet Error Rate
PERT	Packet Error Rate Test
PLC	Programmable Logic Controller
PNC	Provisioning Network Controller
PNF	Physical Network Function
PNFA	PON Network flow Agent
PoC	Proof-of-Concept
PON	Passive Optical Network
PoP	Points-of-Presence
PSE	Physical Simulation Environment

PTZ	Pan Tilt Zoom
QKD	Quantum Key Distribution
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random-Access Memory
R-CORD	Residentential CORD
REST	Representational State Transfer
RN	Radio Network
ROADM	Reconfigurable Optical Add Drop Multiplexer
RPC	Remote Procedure Call
RTT	Round-Trip-Time
RU	Radio Unit
SaaS	Software-as-as-Service
SBI	South Bound Interface
SBIo	SBI for Optical Networks
SBIp	SBI for Packet Switched Networks
S-BVT	Sliceable BVT
SD	Standard Definition
SDN	Software-Defined Networking
SDO	Standards Defining Organization
SFP	Small Form Factor Pluggable
SIP	Service Interface Point
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TAPI	Transport API
TCP	Transmission Control Protocol
TIP	Telecom Infra Project
T-SDN	Transport-SDN
UDP	User Datagram Protocol
UHD	Ultra-High Definition
URLLC	Ultra-Reliable Low Latency Connections
V2I	Vehicle to Infrastructure
V2N	Vehicle to Network (Internet servers)
V2P	Vehicle to Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNF	Virtualized Network Function
VNFFG	VNF Forwarding Graph
VNT	Virtual Network Topology
VOA	Variable Optical Attenuator
VPLS	Virtual Private LAN Service
VPN	Virtual Private Network
VR	Virtual Reality
VTEP	Virtual Tunnel End Point

VTN	Virtual Transport Network
VXLAN	Virtual Extensible LAN
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WIM	WAN Infrastructure Manager
WSS	Wavelength Selective Switch
xPON	X version of Passive Optical Network
XR	Extended Reality
ZB	Zetabyte

8 References

[BBF1] "ITU-T PON YANG Modules", Broadband Forum, WT-385, May 2017

[Cas18] R. Casellas, R. Martínez, R. Vilalta, R. Muñoz, "Control, Management, and Orchestration of Optical Networks: Evolution, Trends, and Challenges", Journal of Lightwave Technology, Volume: 36, Issue: 7, April1, 1 2018

[Cassandra] Apache Cassandra: [online] http://cassandra.apache.org/

[Cha17] B. Chatras, S. Tsang Kwong U, N. Bihannic, "NFV Enabling Network Slicing for 5G", Innovations in Clouds, Internet and Networks (ICIN), 2017 20th Conference on, March 2017 Paris.

[ConfD] ConfD, a powerful management agent software framework for network elements, online [http://www.tail-f.com/confd-basic/]

[D31] METRO-HAUL D3.1, "Deliverable D3.1, Selection of metro node architectures and optical technology options", May 2018

[draft-ietf-actn] D. Cecarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", IETF draft: draft-ietf-teas-actn-framework-13, April 3, 2018.

[draftietf-i2rs-network-topo A Data Model for Network Topologies, IETF draft, work in progress, [online] https://datatracker.ietf.org/doc/draft-ietf-i2rs-yang-network-topo/

[Eck16] A. Eckert, L. Martin Garcia, R. Niazmand, and X. Wang, "Wedge 100: More open and versatile than ever", 18 October 2016. https://code.facebook.com/posts/1802489260027439/wedge-100-more-open-and-versatile-than-ever/

[G.694.1] International Telecommunications Union, "ITU-T Recommendation G.694.1, Spectral grids for WDM applications: DWDM frequency grid", 2011.

[G.872] International Telecommunications Union, "ITU-T Recommendation G.872, Architecture of optical transport networks, draft v0.16 2012/09 (for discussion)", 2012.

[Gon14] L. Gong, Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks", Journal of Lightwave Technology 32 (3), 450-460, 2014

[gRPC] gRPC: a high performance, open-source universal RPC framework: [online] https://grpc.io

[IFA005] ETSI GS NFV-IFA 005 V2.4.1; Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification. February 2018 http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/005/02.04.01_60/gs_NFV-IFA005v020401p.pdf

[IFA007] ETSI GS NFV-IFA 007 V2.4.1; Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification. February 2018 http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/007/02.04.01_60/gs_NFV-IFA007v020401p.pdf

[IFA010] ETSI GS NFV-IFA 010 V2.4.1; Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Functional requirements specification. February 2018 http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/010/02.04.01_60/gs_NFV-IFA010v020401p.pdf

[IFA011] ETSI GS NFV-IFA 011 V2.4.1; Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; VNF Descriptor and Packaging Specification. February 2018 http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/011/02.04.01_60/gs_NFV-IFA011v020401p.pdf

[IFA013] ETSI GS NFV-IFA 013 V2.4.1; Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification. February 2018 http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/013/02.04.01_60/gs_NFV-IFA013v020401p.pdf

[IFA014] ETSI GS NFV-IFA 014 V2.4.1; Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Network Service Templates Specification. February 2018 http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/014/02.04.01_60/gs_NFV-IFA014v020401p.pdf

[IFA022] ETSI GR NFV-IFA 022 V3.1.1; Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on Management and Connectivity for Multi-Site Services. April 2018 http://www.etsi.org/deliver/etsi_gr/NFV-IFA/001_099/022/03.01.01_60/gr_NFV-IFA022v030101p.pdf

[IPFIX] B. Claise, B. Trammell, P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol," IETF RFC 7011, 2013.

[Leira_1] Rafael Leira, Javier Aracil, Jorge E. López de Vergara, Paula Roquero, Iván González, "High-speed optical networks latency measurements in the microsecond timescale with software-based traffic injection," Optical Switching and Networking, Volume 29, July 2018, pp. 39-45, Elsevier, ISSN 1573-4277, doi:10.1016/j.osn.2018.03.004

[Li17] X. Li, R. Casellas, G. Landi, A. de la Oliva, X. Costa-Perez, A. Garcia-Saavedra, T. Deiss, L. Cominardi, R. Vilalta, 5G-Crosshaul Network Slicing: Enabling Multi-Tenancy in Mobile Transport Networks, IEEE Communications Magazine, special issue on Network Slicing in 5G systems, August 2017, Vol. 55, No. 8, August 2017

[Lop15] V. López, et al, "Demonstration of SDN orchestration in optical multi-vendor scenarios", OFC2015, paper Th2A.41.

[Lyu16] I. Lyubomirsky, B. Taylor, and H.-J. W. Schmidtke,"An open approach for switching, routing, and transport", Nov. 2016. Available: https://code.facebook.com/posts/1977308282496021/an-open-approach-for-switching-routing-and-transport/

[May16] A. Mayoral, R. Vilalta, R. Muñoz, R. Casellas, R. Martínez, SDN orchestration architectures and their integration with Cloud Computing Applications, Optical Switching and Networking, February 2016

[Mor18] F. J. Moreno-Muro, C. San-Nicolas-Martinez, E. Martin-Seoane, M. Garrich, P. Pavon-Marino, O. G. de Dios, and V. López, "Joint Optimal Service Chain Allocation, VNF instantiation and Metro Network Resource Management Demonstration," OFC, Tu3D.10, 2018.

[Nej11] R. Nejabati, E. Escalona, S. Peng, D. Simeonidou, "Optical network virtualization", Optical Network Design and Modeling (ONDM), 2011

[net2plan] Net2Plan, the Open Source Network Planne, [online]: http://www.net2plan.com/

[Netconf] R. Enns, ed. "Network Configuration Protocol NETCONF", IETF Request for Comments 6241, June 2011.

[NFVArch]	ETSI, "Network Functions Virtualisation (NFV); Virtual Network Functions Architecture," December 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_nfv-swa001v010101p.pdf
[OCP]	Open Compute Project (OCP) web site: http://www.opencompute.org/
[ODTN]	https://www.opennetworking.org/solutions/odtn/
[ONAP]	ONAP: Open Network Automation Platform: [online] https://www.onap.org
[ONFArch]	ONF ONF TR-521, "SDN Architecture, Issue 1.1, 2016.
[ONOS]	ONOS, Open-source Network Operating System: [online] https://www.onosproject.org/
[OOPT]	Open Optical & Packet Transport (OOPT), TIP project web site: https://telecominfraproject.com/project/backhaul-projects/open-optical-packet-transport/
[OpenConfig]	The OpenConfig project, online: http://openconfig.net/
[OpenDaylight]	OpenDaylight: [online] https://www.opendaylight.org/
[OpenROADM]	The Open ROADM Multi-Source Agreement (MSA), online: http://www.openroadm.org
[OpenStack]	OpenStack: [online] https://www.openstack.org/
[OSM]	OSM, open source MANO: [online] https://osm.etsi.org/
[Raf18]	D. Rafique, "Cognitive Assurance Architecture for Optical Network Fault Management", in IEEE JLT, vol. 36, no. 7, pp. 1443-1450, April1, 1 2018
[RFC7575]	M. Behringer et al, "Autonomic Networking: Definitions and Design Goals", IETF RFC 7575, June 2015.
[Ric18]	E. Riccardi, P. Gunning, O. Gonzalez de Dios, M. Quagliotti, V. Lopez, A. Lord, "An Operator's view on introduction of White Boxes in Optical Networks", Journal of Lightwave Technology, 12 March 2018, DOI: 10.1109/JLT.2018.2815266.
[Rom18]	J. L. Romero-Gázquez, M. V. Bueno-Delgado, F. J. Moreno-Muro, and P. Pavon-Marino, "Net2plan-GIS: An open-source Net2Plan extension integrating GIS data for 5G network planning" ICTON 2018.
[Rui18]	Mario Ruiz, Javier Ramos, Gustavo Sutter, Jorge E. López de Vergara, Sergio López-Buedo, Javier Aracil, "Accurate and affordable packet-train testing systems for multi-Gb/s networks," IEEE Communicatons Magazine, Vol. 54, Issue 3, March 2016. ISSN 0163-6804. doi:10.1109/MCOM.2016.7432152
[Sch17]	H.-J. Schmidtke and L. M. Garcia, "Driving Openness in Optical Networks: An Update from the OOPT Project", Nov. 2017. Available: http://telecominfraproject.com/riving-openness-in-optical-networks-an-update-from-the-oopt-project-group/
[Spark]	Apache Spark: [online] http://spark.apache.org/
[TAPI]	Open Networking Foundation (ONF) Transport API (TAPI) 2.0 Overview, August 2017 online : https://www.opennetworking.org/wp-content/uploads/2017/08/TAPI-2-WP_DRAFT.pdf
[TIP]	Telecom Infrastructure Project (TIP) web site: http://telecominfraproject.com/
[TR-384]	Broadband Forum Technical Report, TR-384, "Cloud Central Office Reference Architectural Framework", Issue: 1, January 2018. online https://www.broadband-forum.org/technical/download/TR-384.pdf
[Vel17]	A. P. Vela, M. Ruiz, F. Fresi, N. Sambo, F. Cugini, G. Meloni, L. Potì, L. Velasco, and P. Castoldi, "BER Degradation Detection and Failure Identification in Elastic Optical Networks," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 35, pp. 4595-4604, 2017.
[Vel18]	A. P. Vela, B. Shariati, M. Ruiz, F. Cugini, A. Castro, H. Lu, R. Proietti, J. Comellas, P. Castoldi, S. J. B. Yoo, and L. Velasco, "Soft Failure Localization during Commissioning Testing and Lightpath Operation [Invited]," IEEE/OSA Journal of Optical Communications and Networking (JOCN), 2018.
[Vel18a]	Ll. Gifre, J-L. Izquierdo-Zaragoza, M. Ruiz, and L Velasco, "Autonomic Disaggregated Multilayer Networking," IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 10, pp. 482-492, 2018.

[Vel18b]	L. Velasco, A. Sgambelluri, R. Casellas, Ll. Gifre, J-l. Izquierdo-Zaragoza, F. Paolucci, R. Martínez, E. Riccardi, "Building Autonomic Optical Whitebox-based Networks," IEEE/OSA Journal of Lightwave Technology (JLT), PP April 2018.
[Vel18o]	Luis Velasco, Lluis Gifre, Jose Luis Izquierdo-Zaragoza, Guillermo Julián, Jorge López de Vergara, "CASTOR: An Architecture to Bring Cognition to Transport Networks," Proceedings of the Optical Fiber Communication Conference, OFC'2018, San Diego, CA, USA, 11-15 March 2018
[Vel18Sl]	L. Velasco, Ll. Gifre, J.-L. Izquierdo-Zaragoza, F. Paolucci, A. P. Vela, A. Sgambelluri, M. Ruiz, and F. Cugini, "An Architecture to Support Autonomic Slice Networking [Invited]," IEEE/OSA Journal of Lightwave Technology (JLT), 2018.
[Yang]	M. Bjorklund, Ed., "The YANG 1.1 Data Modeling Language", IETF Request for Comments 7950, August 2016.

[end of document]