2020 Transport SDN API Interoperability Demonstration

OIF Whitepaper
February 2021
www.oiforum.com

Table of Contents

Table of Figures

1 EXECUTIVE SUMMARY

Manageability and flexibility of the network are critical to allow network operators to successfully deliver a range of cloud-based services, meet dynamic bandwidth demands, and accelerate transport network transformation for the 5G era. Transport SDN gives network operators flexibility and more responsive central control of network traffic through a programmable network. Established open Transport SDN APIs can help network operators:

- improve network agility to adapt to dynamic service demands and traffic patterns
- improve service provisioning and time-to-revenue
- reduce maintenance and management with simplified control and automation

By working through specifications, rigorous interoperability testing, and validation, the Optical Internetworking Forum (OIF) 2020 Transport SDN Application Programming Interface (API) Interoperability Demonstration focused on SDN-based programmability, control, and automation using open, standardized APIs – the established Open Networking Foundation (ONF) Transport-API (T-API) northbound interface for the SDN Controller and the OpenConfig interface for the devices.

The 2020 Transport SDN API interoperability demonstration builds on OIF's 2018 interoperability test and demonstration, which substantiated T-API as the northbound interface (NBI) of choice and established a foundation for open, programmable networks that allow network operators to efficiently deliver dynamic multi-domain connectivity services to the market. OIF tested new operator-defined use cases and deployment scenarios in the host lab to test multi-vendor interoperability of Layer 1 OTN, Layer 0 OTN control using ONF T-API 2.1.3 and OpenConfig APIs.

In 2013, the OIF developed and published a Transport SDN Framework that defined key functions and interfaces. In 2014, the OIF partnered with the ONF to conduct an interop demo that tested pre-standard ONF OpenFlow extensions for the Southbound Interfaces or APIs and prototype transport Northbound APIs to support Service and Topology requests. That work led to the initiation of standards work in ONF on the Transport API (T-API) NBI and approval of T-API specs in 2H2016.

In the 2016 OIF SDN Transport API Interoperability Demonstration, the OIF and ONF partnered to lead the industry toward the wide-scale deployment of commercial SDN by testing ONF T-API standards. The interoperability test and demonstration, managed by the OIF, addressed multi-layer and multi-domain environments in global carrier labs located in Asia, Europe, and North America. That work led to the enhanced T-API 2.0 spec published by ONF in 2017.

In 2018 the OIF, in collaboration with MEF, introduced new dynamic-behavior use cases and deployment scenarios into network operator labs around the world to test multi-vendor interoperability of the T-API 2.0 NBI. The 2018 SDN T-API interoperability demonstration built on the OIF's 2016 interoperability test and demonstration which addressed multi-layer and multi-domain environments as well as on the 2014 demo which prototyped the use of Northbound APIs and helped advance transport SDN standardization. The demo also incorporated service provisioning scenarios at the MEF LSO (Lifecycle Service Orchestration) Presto reference point in the MEF LSO architecture, using the MEF NRP Interface Profile Specification (MEF 60), which defines T-API extensions in support of MEF Carrier Ethernet services.

The OIF 2020 Transport SDN API Interoperability Demonstration achieved the objectives to evaluate interoperability over standard interfaces. Participants complemented limited on-site technical support with innovative virtual lab testing to overcome resource, lab access, and communication challenges presented by the coronavirus pandemic. Despite the challenges, participants supported (21) integrations between vendor solutions in (29) T-API use cases and (31) OpenConfig operations over a 10-week testing period.

Findings and results of the testing include:

- Vendor products demonstrated a high level of maturity and support of operator-favored use cases, based on T-API version 2.1.3 and OpenConfig APIs. Vendors have adopted the ONF 2.1.3 Reference Implementation Agreement (ONF TR-547) rapidly and with few deviations.
- Inconsistent interpretation of the Internet Engineering Task Force (IETF) RESTCONF standard was one interoperability issue identified, especially standard authentication. Clarification to RESTCONF provisioning behavior and implementation scalability were also identified as issues.
- A particularly good level of compliance to the OpenConfig models was demonstrated by implementations, however, some areas such as key exchange algorithms for IETF NETCONF and complex use cases for device provisioning and commissioning did generate issues.

The testing shows that the T-API spec is at an acceptable maturity to allow participating, early adopter network operators to continue aggressive commercial deployment plans and for all operators to adopt T-API as a standard NBI in planning open optical transport SDN networks deployments. Continued enhancements to the spec will enable accelerated, broad adoption of T-API.

The testing also demonstrated a very good level of OpenConfig model compliance and that both NETCONF and gNMI are widely supported. The OIF continues to help establish a foundation for open, programmable networks that allow operators to efficiently deliver dynamic multi-domain connectivity services to the market.

2 INTRODUCTION

Currently, network operators deploy optical networks on a regional basis. This means that each vendor uses its technology in a given area and operates it, which is very relevant for production environments. Towards the OSS, the situation is similar. The traditional OSS-network integration is presented in Figure 1. Each vendor exposes to the OSS layer its northbound interface, thus increasing the integration complexity and cost.

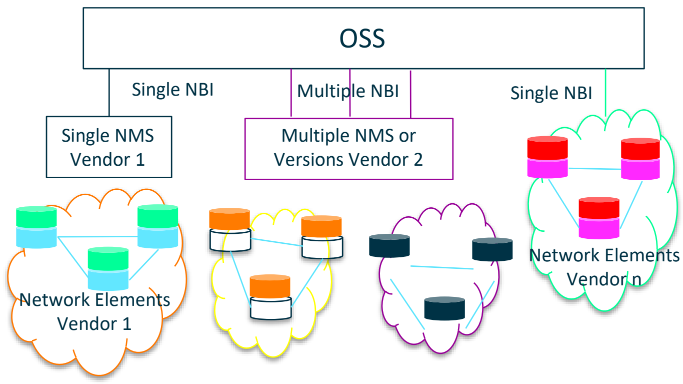


Figure 1: Traditional OSS-Network Integration Model

The use of T-API as a Northbound Interface (NBI) of the optical controllers allows the utilization of a common abstraction model to support the optical services. Figure 2 illustrates how a Transport API facilitates the integration with OSS enabling an Open Transport Integration Model.

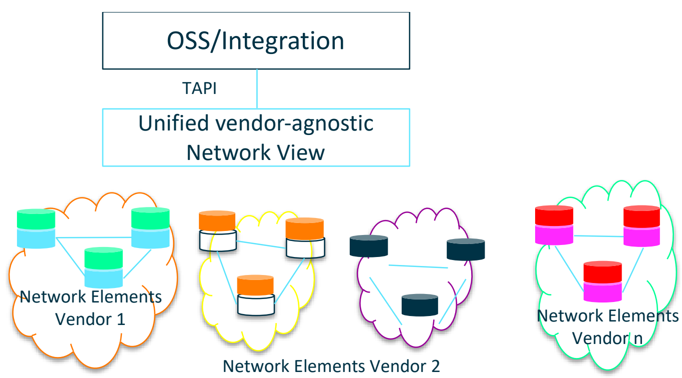


Figure 2: Open Transport Integration Model

This Open Transport Integration model allows operators to decouple the network evolution from the OSS evolution. Consequently, network and OSS teams can select the best technology for their needs, simplifying the migration process, when updating either the OSS stack or the network deployment.

Transport API allows this evolution because it standardizes not just the interface, but also the information model. It creates a model that supports multi-vendor solutions with a common network view for resources and services.

3 Demo Setup

This interop was carried out in a single operator lab in Europe. The following are the details of the demo participants and roles.

Host Network Operator:

- Telefonica

Consulting Network Operators:

- China Telecom
- Telia Company
- TELUS

Five vendors participated with software and hardware, providing a variety of types of equipment and software functions. The testing involved lab and cloud-deployed systems. Participating vendors:

- ADVA
- Ciena
- Cisco
- Infinera
- Nokia

The details of hardware and software provided by each vendor participant are included in Figure 3.

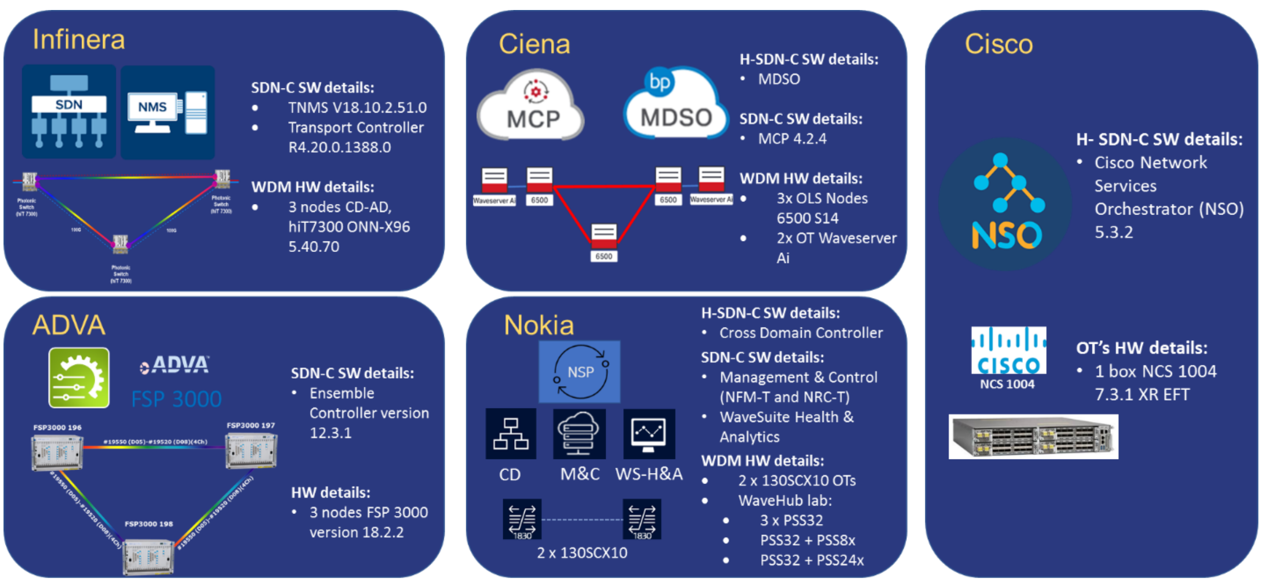


Figure 3: OIF 2020 Transport SDN API Interoperability demo testbed setup

4 TECHNICAL SPECIFICATIONS

4.1 TRANSPORT API (T-API)

T-API is a standard API defined by the Open Networking Foundation (ONF) that allows a T-API client to retrieve information from and control a domain of transport network equipment controlled by a T-API server. T-API is designed to be deployable in a multi-level hierarchy of controllers, offering control over network resources at different levels of abstraction. A typical deployment of T-API would be as the interface between a set of network *Domain Controllers* and an upper-level network *Orchestrator* that acts as a parent controller, supporting multi-domain, multi-vendor, multi-layer end-to-end network programmability.

The set of primitives supported by T-API for transport network control includes:

- Topology Service: retrieval of network topology, resource availability, and status
- Inventory Service: retrieval of the relationship of logical network objects and their physical location in, for example, chassis, slot, and port
- Connectivity Service: control of creation, modification, and deletion of connectivity services between service endpoints, with specified path constraints
- OAM Services: ability to instantiate OAM monitoring points and control fault and performance monitoring for network troubleshooting
- Notification Service: subscription to autonomous or on-demand information about network events and monitoring data
- Higher Level Features: access to additional services such as network virtualization/slicing and path computation services

This set of primitives is implemented in T-API in ways that allow for varying degrees of information and control based on operator policy. This allows for different business relationships to be supported by the API, including clients that may be allowed very limited "black box" type views of the network to internal operator applications that may need full details of the network topology and the ability to trace services across this topology.

T-API work is based on the ONF's Core Information Model (CIM). The [ONF-CIM] provides a representation of data plane resources for management-control by the operator. The ONF T-API derives its Information Model by pruning and refactoring the ONF-CIM as a purpose-specific realization of the ONF-CIM for transport network control.

4.1.1 T-API 2.1.3

T-API 2.1.3 is the latest version of the T-API Software Development Kit and was published by ONF in June 2020. The main new features in T-API 2.1.3 are as follows:

Equipment Inventory

- Allows the client to determine the relationship between logical objects and their physical location, i.e., chassis, slot, port

Restconf Alignment

- Previous T-API versions followed REST patterns. Based on operator input this version aligns the T-API specification with the Restconf standards defined in IETF.

Photonic Media Model

- The extension of T-API to support a photonic media model allows the operator to separate the control of the media channel created within the line system and the photonic signal created by the endpoints that then traverses the media channel

Streaming Telemetry

- This extension allows the use of streaming technology to improve the efficiency and scalability of the interactions between T-API client and server, making T-API more scalable for real-world deployments involving potentially large quantities of information (e.g., topology details or event reports) being transferred across the T-API interface.

4.1.2 T-API 2.1.3 Reference Implementation Agreement

[ONF TR-547] (published July 2020) is the T-API v2.1.3 Reference Implementation Agreement. The purpose of the document is to describe a set of guidelines and recommendations for standard use of the T-API models in combination with RESTCONF protocol for the implementation of the interface between network systems in charge of the control/management of networks based on WDM/OTN technologies.

The T-API v2.1.3 Reference Implementation Agreement provides details of the models to be supported by compliant systems and the use cases to be supported for topology retrieval, connectivity service creation/deletion, recovery, inventory services, and event notification (including via streaming).

4.2 OpenConfig

OpenConfig is a set of well-defined and consistent common data models for device configuration and streaming telemetry. The data models are written in YANG and model both configuration and operational data in vendor-neutral structures.

OpenConfig is not a standards body; the project is a collaborative effort between network operators to develop programmatic interfaces and tools for managing networks in a dynamic, vendor-neutral way. The project is driven by the operational needs from use cases and requirements across these operators.

OpenConfig defines a large set of YANG modules to be supported by a network device. These contain models to represent device capabilities and features, such as hardware components hierarchy, interfaces, OSPF configuration, QoS, etc. A device may already support many of these features through proprietary vendor-specific YANG models. An OpenConfig model provides a common way to configure and manage the same features across vendors. For all vendors that support a feature through OpenConfig, a network operator can use the same YANG model to configure and monitor the feature across those different vendor devices.

OpenConfig has adopted a structural convention for YANG models that emphasizes the importance of modeling operational state (i.e., monitoring and telemetry data), in addition to configuration data. At a high level, this convention uses specially named config and state containers, in every sub-tree to explicitly indicate configuration and operational state data.

4.2.1 Optical Transport YANG models

There are a large number of OpenConfig YANG models; only models in the platform components module and the terminal device module are used for the interoperability testing.

The platform components module defines a data model for representing a system component inventory, which can include hardware or software elements arranged in an arbitrary structure.

Every element in the inventory is termed a *component* with each component expected to have a unique name. Where other parts of the device model include an element name, it is a reference to this component list, therefore all elements must be present in the list.

The component list defines a generic tree structure with a simple parent/child relationship. Each component has a single parent and a list of *subcomponents*, which are references to other components. Appearance in a list of subcomponents indicates a containment relationship. For example, a line card may have subcomponent references to port components that reside on the line card. However, the YANG model itself doesn't define or enforce this relationship; a device will populate the generic YANG component structure with the specific containment relationship between the device inventory elements.

The component model is augmented with nodes for the different component types: line card, port, and transceiver. A transceiver is expected to be a subcomponent of a port and allows configuration and operational state data for pluggable optics. The components module also includes augments for CPU, Fan, and PSU component types.

The terminal device module adds augments for the transceiver physical channels and an optical channel component type. An optical channel corresponds to an optical carrier and is assigned a wavelength/frequency, and has properties such as power, BER, and operational mode. The terminal device module defines a separate top-level tree that is a list of operational modes supported by the device.

The terminal device module also defines another top-level tree for the list of logical channels configured on the device. These represent the logical grouping of logical grooming elements that may be assigned to subsequent grooming stages for multiplexing / de-multiplexing, or to an optical channel for line side transmission. At the client side, the logical channel references the ingress transceiver component and physical channel.

The list of main YANG models used for the interoperability testing is shown in the following two tables:

OpenConfig Platform (Components) Module

Name	Yang Model	Description
Platform	openconfig-platform.yang	The main model to define the hardware components hierarchy of a network device.
CPU	openconfig-platform-cpu.yang	Augments the platform model to add specific parameters of a CPU component.
Fan	openconfig-platform-fan.yang	Augments the platform model to add specific parameters of a fan component.
Line card	openconfig-platform-linecard.yang	Augments the platform model to add specific parameters of a line card component.
Port	openconfig-platform-port.yang	Augments the platform model to add specific parameters of a port component.
PSU	openconfig-platform-psu.yang	Augments the platform model to add specific parameters of a Power Supply Unit component.
Transceiver	openconfig-platform-transceiver.yang	Augments the platform model to add specific parameters of a transceiver component.
Platform Types	openconfig-platform-types.yang	Defines the types and common groupings used in the platform module.

OpenConfig Terminal Device (Optical Transport) Module

Name	Yang Model	Description
Terminal Device	openconfig-terminal-device.yang	The main model to define a terminal optics device.
Optical Transport Types	openconfig-transport-types.yang	Defines the types and common groupings used in the terminal device module.

4.2.2 Transport Protocol

OpenConfig YANG models can be used over NETCONF, RESTCONF, and gNMI transport protocols. The interoperability testing focuses on device OpenConfig configuration over NETCONF and some testing of streaming telemetry over gNMI.

5 Use Cases

The target of this interop was to evaluate interoperability over standard interfaces of different SDN Controllers (SDN-C), Hierarchical SDN-C, and disaggregated Open Terminals, through the evaluation of different use cases over different network scenarios.

5.1 Agnostic network control over standard NBI (T-API)

The first objective was to evaluate how the standard APIs can provide uniform and vendor-agnostic network control and management functionalities over single domain WDM/OTN network scenarios. Network operators target the uniform exposure of all network control and management functions from all its network suppliers to be integrated into their Operations Support Systems (OSS) seamlessly. The solution evaluated in this interop is based on ONF T-API (see Figure 4).

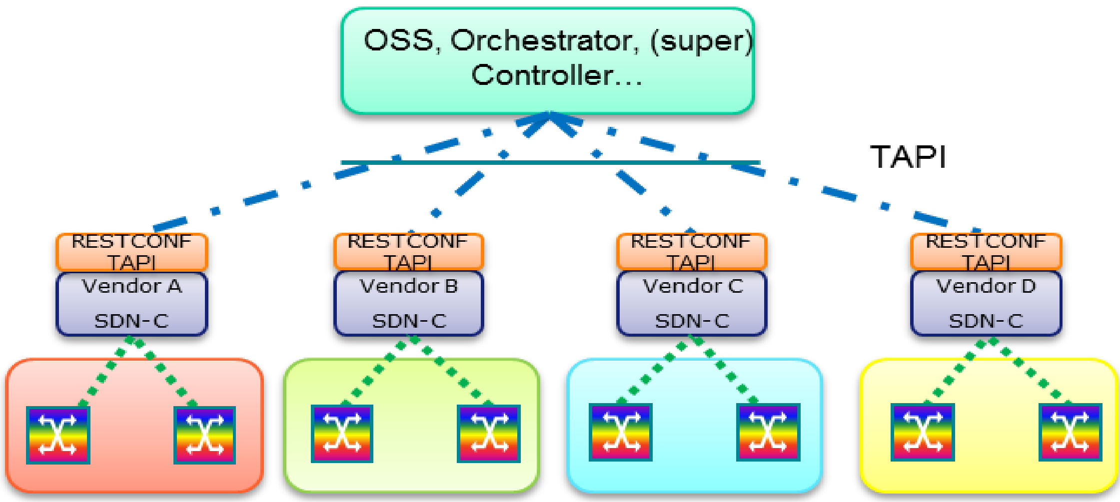


Figure 4: Multi-vendor agnostic network control over standard NBI (T-API).

As described in section 4.1, the [ONF TR-547] has been defined as a standard reference implementation agreement to unify several implementation aspects over the standard T-API models. The main aspects which required further standardization efforts beyond the data model itself are the topology and connectivity abstractions required to be implemented. These abstractions defined how to expose the network elements capabilities, logical connectivity, and its topology relationships, through the different T-API models, based on its network capabilities, independently to the network manufacturer.

In section 5.3 all identified use cases for this architecture are detailed.

5.2 Optical Partial Disaggregation

Optical disaggregation is a current trend in optical networking that aims to provide interoperability between SDN controllers and Network Management Systems (NMS), and network elements to enable building multi-vendor optical networks. It breaks the silos established by proprietary information models and protocols used today to configure the network elements by the network suppliers.

The SDN paradigm has introduced new standard information models (T-API and OpenConfig among others) which allow standard and vendor-agnostic network modeling at different management/control abstraction levels. This standardization is key to achieve the target disaggregation by allowing interoperability between control/management systems and network elements from different suppliers.

Optical Partial Disaggregation is an architectural solution that is being adopted by different network operators, including Telia Company and Telefonica among others, where the optical terminal devices (Transponders/Muxponders) are disaggregated from the rest of the network (the Line System). This approach is perceived by the industry as the most cost-effective solution, given its relative simplicity compared with more aggressive disaggregation architectures (e.g., the full disaggregation approach which requires the standardization of the whole optical device's portfolio included within the line system, such as Mux/Demux boards, Filters, Amplifiers…), and because it provides great benefits given the portion the optical terminals represents within the optical transport market segment (around an 80%).

The partial disaggregation solution imposes a set of requirements to be effective. The participating operators divide their requirements into those to be fulfilled to realize the Open Line System (OLS) and the Open Terminals (OT) solutions:

Open Line System.

1. **Openness.** This means that the Line System supplier shall allow transporting third-party alien wavelength services over its hardware. This implies the OLS filters allow transmission of third-party signals using the required hardware configuration, adjusting the filters to the required band. There should not be a difference between the alien-wavelength bandwidth combination possible by the ITU-T standard grids defined in G.694.2, including if the underlying OLS filters support it, the Flexgrid with 6.25 GHz adjustment granularity.
2. **Support of a standard NBI** over the OLS management entity (OLS SDN Controller or SDN-C), for the network deployment processes and integration with OSS and other controllers. This requirement explicitly includes the support of the independent management/control of media elements as defined by the [ITU-T G.872] - Section 8, including but not limited to Media-channels, OMS, and OTS layers (supported in T-API 2.1.3 in the demo).
3. **Open Optical Planning**. The open planning and impairment validation is required to determine the end-of-life quality of transmission for a given service manage by the OLS by a vendor-agnostic planning tool. The OLS segment must expose through the standard NBI the required data for planning and impairment validation of service by an external planning tool. This information MUST include, but is not limited to:
    a. Routing and spectrum allocation map across all OMS sections within the OLS.
    b. Media-channel cross-connections configuration, including the explicit reference to hardware equipment ports.
    c. Power monitoring status at OTS, OMS, and Media Channel logical termination points.
    d. Optical amplifiers across the media-channel path (including degree amplifiers, booster, pre-amplifiers, in line-amplifiers) configuration and status data, including operational frequency ranges, gain max/min levels, max output power, tilt, and noise figures. This requirement applies to all optical amplifiers of the supplier OLS equipment portfolio.
    e. Hardware detailed information (equipment type, part-numbers, performance metrics, management information) about all the equipment composing the OLS. Including all

connectivity information between these components (i.e., filters, WSS, amplifiers, OSC boards…).

   f. Connector loss static information on each optical connector along the media-channel path.
   g. Fiber strands information including, distance, fiber type, fiber spans, lineal attenuation figures, and PMD and CD degradation.

4. **Media Channel's management**. The OLS SDN-C segment must expose, through the standard NBI, control/management capabilities for the media channel services. This control/management MUST include, but is not limited to:
   a. Service provisioning of Media Channels from ROADM add/drop port to ROADM add/drop port.
   b. PCE functions for non-explicit defined Media Channel service provisioning:
      i. Routing and spectrum allocation for end-to-end map across all OMS sections within the OLS.
      ii. Support for target input/output power constraints, defined at Media Channel service's boundary endpoints, for the PCE function.
      iii. Support for target OTSi signal sensitivity constraints (OSNR, PMD, CD tolerances) for PCE function.
   c. Explicit defined Media Channel service provisioning including, explicit OMS, OTS routing and spectrum allocation, intermediate amplifier target configurations (variable gain, target output power).
   d. Media Channel, OMS, OTS, OSC fault management alarms exposure.
   e. Media Channel, OMS, OTS, OSC performance monitoring through OAM and STREAMING interfaces.
   f. The last step is the support of a standard model and open interface on the Open Terminals and the SBI (OpenConfig) of the controllers.

Open Terminals

1. **The support of the Standard SBI interface,** by the SDN-C to manage and control Open Terminals. This MUST be supported by the supplier SDN Controller solution which will provide the Open Terminals management. In this interop we use a NETCONF/gNMI interface based on OpenConfig models.
2. **Transparent management**. Exposure of third-party Open Terminals control/management capabilities through the SDN-C standard NBI for seamless integration with upper layers. This means the SBI model shall be translated/abstracted by the SDN-C solution to be exposed through the NBI.
3. **Full control/management capabilities.** The Open Terminals control/management shall include the following functionalities:
   a. Full hardware and software inventory discovery.
   b. Full device configuration discovery.
   c. Full configurability of the device.
   d. Full fault management features.
   e. Full performance monitoring and OAM functions.

Assuming these requirements, two possible architectures are possible to realize the partial disaggregation solution (see Figure 5).

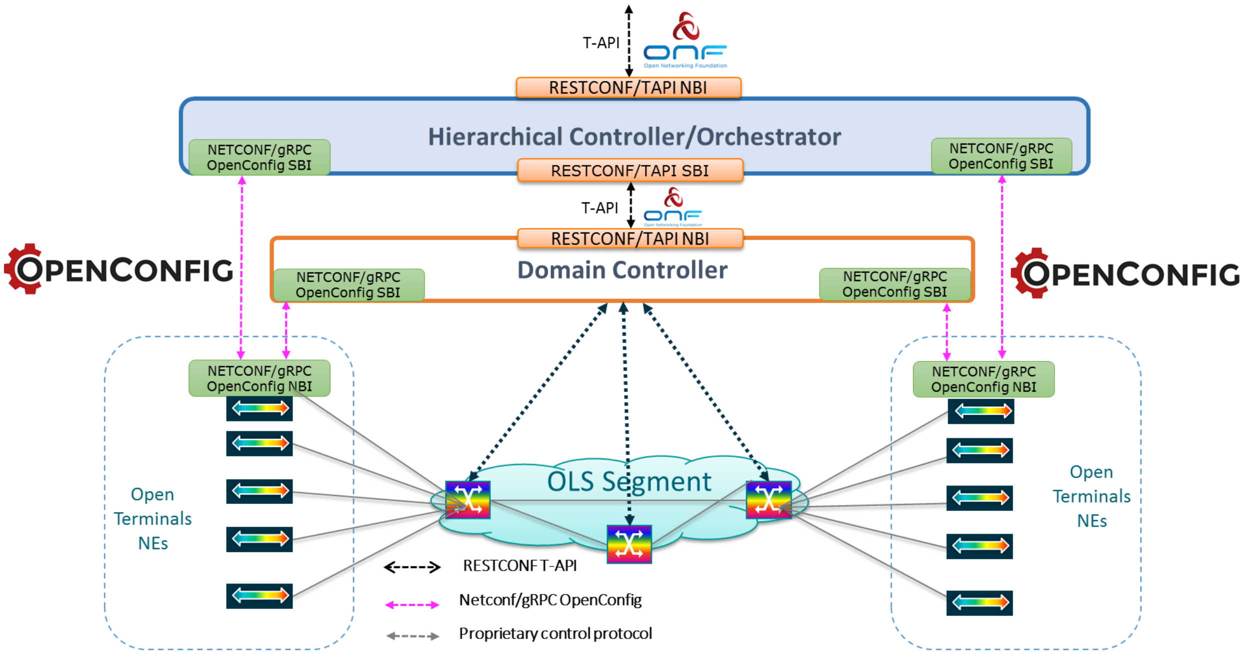


Figure 5: Partially disaggregated SDN architecture.

5.2.1 Option A: Single SDN Controller approach.

In this scenario, the Open Line System (OLS) supplier's SDN-C supports the management and control of third-party Open Terminals (OTs) through standard SBI (see Section 4.2).

In this case, the main technical requirements are:

1. The third-party OTs MUST support the proposed standard management interface (SBI OpenConfig) for the configuration, state data retrieval, fault management, and performance monitoring functionalities.
2. The OLS supplier MUST provide an SDN-C solution that supports the integration (including the configuration, state data retrieval, fault management, and performance monitoring functionalities) of Third-Party OTs through the proposed standard interface.

5.2.2 Option B: Hierarchical SDN Controller approach

The second possible architecture requires a Hierarchical SDN-C (H-SDN-C) to manage the Open Terminals (OTs) over an already deployed Open Line System (OLS). In this architecture, the H-SDN-C is responsible to perform the whole control/management workflow to support the disaggregation uses case.

Additionally, to the requirements included in option A, the proposed standard SDN-C NBI (T-API) is required to be implemented as SBI of the H-SDN-C for the control of the OLS, along with the implementation of OT's management interface (OpenConfig) as SBI.

5.3 Common Use Cases

For the set of use cases described in this section, it is assumed that the SDN-C is responsible for the control and management of a network domain. The network architecture controlled by the SDN-C might be aggregated (all the network equipment within the network domain belongs to the same supplier) or partially disaggregated (following one of the two architectures described in the previous section). All the Use Cases (UCs) defined in this section MUST be supported by the SDN-C solution in any of the network architectures proposed.

In this interop the target use cases have been selected from the definition in ONF TR-547 based on its maturity, they are organized within the following categories:

5.3.1 Network discovery use cases

- Use case 0a**:** Context and service interface points discovery.
  - The discovery of the service interface points available in the SDN-C context is a basic requirement to request connectivity services.
  - The SDN-C MUST support the exposure of Service Interface Points at the following networking layers:
    - Digital Signal Rate (DSR) services between client interfaces of terminal devices i.e., 1 GE, 10 GE, 100GE, FlexE, STM-X, Fibre Channel.
    - ODU/OTU digital services.
    - OCh/OTSi/OTSiA which can be single or multi-carrier.
    - Media Channels (MC/OTSiMC) defined as: "the media channel is a topological construct that represents both the path through the media and the resource (frequency slot) that it occupies", in [ITU G.872]
  - Moreover, the rest of the services i.e., topology, connectivity, notifications, and path-computation SHALL be available within the tapi-common:context root tree, information exposed by the SDN-C.
- Use case 0b**:** Topology and services discovery.
  - This use case is aimed to be supported through the standard SDN-C NBI's and consists of the retrieval of the logical representation of the network topology. The WDM/OTN network segment consists of all the networking layers according to the architecture and optical transport network hierarchy equipment functional blocks, defined in [ITU-T G.798] and [ITU-T G.872] recommendations.
- Use case 0c**:** Connectivity services discovery.
  - This use case is aimed to be supported through the standard SDN-C NBI's and consists of the retrieval of the logical representation of the connectivity-services and their underlying supporting connections.
  - The network connectivity shall be exposed by the SDN-C as a nested set of layered connections and relationships which describe the actual end-to-end route from the source to the destination connectivity-service-end-points (CSEPs). The hierarchical relationship between layer connections shall follow the connectivity abstraction model defined in TR-547.

5.3.2 Connectivity services provisioning

- Use case 1: Unconstrained E2E Service Provisioning.
  - This use case consists of the provisioning of connectivity services between two service endpoints at any network layer i.e., DSR, ODU, OTSi/OTSiA, MC/MCA, OTSiMC. This service MAY include intermediate regeneration if necessary.

- The underlying connection provisioning and management (including lower layer connections e.g., OTSi, Photonic Media, and regeneration nodes) SHALL be performed by the SDN-C.
- The path of each lower layer connection (e.g., OTSi, Photonic Media) across the network topology is computed by the SDN-C and the connection is automatically provisioned.
- The "unconstrained" term refers that the NBI-Client is not allowed to introduce any routing constraint in the service request, thus relays completely into the routing capabilities of the SDN-C to select the network resources employed to provide the desired service characteristics.
- Moreover, the SDN-C should be able to export stateful information about the services deployed in its control domain. Here the services included in the scope of the WDM/OTN SDN-C are:
    - Digital Signal Rate (DSR) services between client interfaces of terminal devices i.e., 1 GE, 10 GE, 100GE, FlexE, STM-X, Fibre Channel.
    - ODU/OTU digital services.
    - OTSi/OTSiA which can be single or multi-carrier.
    - Media Channels (MC) defined as: "the media channel is a topological construct that represents both the path through the media and the resource (frequency slot) that it occupies." in [ITU-T G.872]

In detail, the SDN-C MUST be able to support the provisioning of the following sub-use cases:

- <u>Use case 1a: Unconstrained DSR Service Provisioning single wavelength (<100G).</u>
    - This use case consists of the provisioning of connectivity services between two service endpoints, i.e., client interfaces of terminal devices (i.e., transponders, muxponders) at a Digital Signal Rate (i.e., GE, 10GbE, STM-X, Fibre Channel) less than 100Gbps data rate based on a single wavelength OTSi transport signal. This service can include intermediate regeneration if necessary.
- <u>Use Case 1b: Unconstrained DSR Service Provisioning multi wavelength (beyond 100G).</u>
    - This use case consists of the provisioning of connectivity services between two service endpoints, i.e., client interfaces of terminal devices (i.e., transponders, muxponders) at a Digital Signal Rate (i.e., XGbE, STM-X, Fibre Channel) beyond 100Gbps data rate based on a multi-wavelength OTSiA/OTSiG transport signal. This service can include intermediate regeneration if necessary.
- <u>Use case 1c: Unconstrained ODU Service Provisioning</u>
    - This use case consists of the provisioning of connectivity services between two service endpoints, i.e., client interfaces of terminal devices (i.e., transponders, mux-ponders) at an ODU rate (e.g., ODU0, ODU1, ODU2, ODU2E, ODU3, ODU4, ODUCN, ODUFlex).
- <u>Use case 1d: Unconstrained PHOTONIC_MEDIA/OTSi Service Provisioning</u>
    - This use case intends the provisioning of a connectivity service between service-interface-points exposed by the SDN-C at the PHOTONIC_MEDIA (OTSi/OTSiA) networking layer. This service can include intermediate regeneration if necessary.
    - The scope of this use case is just single wavelength OTSI channel transmission for services with less than 100G.
- <u>Use case 1e: Unconstrained PHOTONIC_MEDIA/ OTSiA/OTSiG Service Provisioning</u>

- This use case intends the provisioning of a connectivity service between service-interface-points exposed by the SDN-C at the PHOTONIC_MEDIA (OTSi/OTSiA) networking layer. This service can include intermediate regeneration if necessary.
- The scope of this use case is multi-wavelength OTSI channel transmission for support of services beyond 100G.

- Use Case 1f: Unconstrained PHOTONIC_MEDIA/MC Service Provisioning
  - The SDN-C must accept requests for connectivity service between service-interface-points at the PHOTONIC_MEDIA (MC/MCA) networking layer.

- Use case 3: Constrained Provisioning.
  - The SDN-C must accept requests of a connectivity service between two service endpoints (at the DSR, ODU, or OCh/OTSi layers) subject to a node exclusion or inclusion in the path (for example a regeneration node). This service can include a regeneration node.
  - The underlying connection generation (including lower layer connections e.g., OCh/OTSi, Photonic Media, and regeneration nodes) is made by the SDN-C.
  - The SDN-C calculates the path of each lower layer connection (e.g., OCh/OTSi, Photonic Media) across the network topology considering the path constraints introduced in the service request.

Sub-use cases for consideration:

- Use case 3a: Include/exclude a node or group of nodes at any transport layer (ODU, OTSi, Photonic layers (OMS, OTS)).
- Use case 3b: Include/exclude a link or group of links at any transport layer (ODU, OTSi, Photonic layers (OMS, OTS)).
- Use case 3c: Include/exclude the path used by another service.
  - The route in each layer is recursively calculated from the lower to the upper layer, excluding those components of the topology crossed by the connections created to support the selected service.
  - The route of each lower layer connection (e.g., OCh/OTSi, Photonic Media) across the network topology is calculated by the SDN Domain controller considering the constraints introduced in the service request. The route in each layer is recursively calculated from the lower layer to the upper layer, excluding/including ALL those components of the topology shared with the connections created to support the selected service.

5.3.3 Physical inventory management

- Use case 4: Inventory model for NBI Interface
  - Use case 4a: Introduction of references to external inventory model.
    - Hardware identifiers currently stored in Telefonica OSS inventory systems must be correlated with SDN-C NBI logical endpoint's identifiers. This information MUST be provided by the SDN-C suppliers.
    - For every inventory element represented as a logical element in T-API by the SDN Domain controller, an INVENTORY_ID tapi-common:name property shall be included in the logical element construct.
    - The INVENTORY_ID tag will be included for the following T-API objects:
      - *tapi-topology:node-edge-point*

            - *tapi-common:service-interface-point*
    - Use case 4b: Complete Inventory model for NBI Interface
        - Introduction of the new complete inventory model defined within the information model of the NBI.
        - The model MUST present detailed information about all the physical devices controlled through the SDN-C. This information SHALL cover but NOT BE limited to Site identifier (Location information), Network Element identifier (Serial Number), equipment management IP address, equipment manufacturer, equipment part number, equipment structure (chassis, slots, ports), hardware characteristics (i.e., CPU, Memory, Disk), FAN information (temperature, operational power levels, consumption…)
        - The model MUST integrate the physical devices' inventory information and integrate it into the current logical topology definition.
        - Currently, the tapi-equipment model is required to implement this use case.

5.3.4 Resilient services management

- Use case 5: 1+1 Diverse Service Provisioning
    - Use case 5a: 1+1 OLP OTS/OMS Protection
        - The SDN-C must be able to discover and map into the PHOTONIC_MEDIA layer topology elements, the OLP components at the different protection levels. In this test we will cover the OTS and OMS sections.
        - The UC covers the available partially disaggregated architectures, but they do not impact the workflow as the PHOTONIC_MEDIA layer is entirely managed by the OLS.
        - Thus, the only requirement in the case of the disaggregated architecture Option B is, the OLS SDN-C MUST report the topology features of the PHOTONIC_MEDIA layer through the standard NBI. It must include the protection capabilities of those OMS and OTS links which are equipped by OLP components.
    - Use case 5b: 1+1 OLP Line Protection with Diverse Service Provisioning
        - The SDN-C must support the provisioning of a connectivity service between two service (ETH/ODU) endpoints that will be split using an OLP element to create a disjoint path
        - The underlying connection provisioning and management (including lower OCh/OTSi layer connections) is performed by the SDN-C.
        - The working and protected paths of each lower layer connection (e.g., OCh, Photonic Media) across the network topology is calculated by the SDN-C.
    - Use case 5c: 1+1 Diverse Service Provisioning (eSNCP)
        - The SDN-C must accept requests of 2 connectivity disjoint services between two service endpoints (tributary ports)
        - The underlying connection provisioning and management (including lower OCh/OTSi layer connections) is performed by the SDN-C.
        - The working and protected paths across the OTSi network topology is calculated by the controller.
- Use case 6a: Dynamic restoration for unconstrained and constrained use cases (1, 2, 3)
        - The SDN-C must support providing and management of services between two service endpoints (DSR/ODU) with restoration capabilities.

  - The underlying connection provisioning and management (including lower OCh/OTSi layer connections) is performed by the SDN-C. Also, the new path calculation in case of failure of the working path
  - The path across the OCh/OTSi network topology is calculated by the controller.
  - The SDN-C must provide asynchronous notifications of the events generated by the network failure which triggers the restoration including, but not limited to, topology changes, service status changes, triggered alarms…
- Use case 7a: Dynamic restoration and protection 1+1 for any of the previous cases (1, 2, 3)
  - The SDN-C must support the provisioning and management of services according to use case 5a, 5b, or 5c with a dynamic restoration of the backup service in case of failure.
  - The underlying connection provisioning and management (including lower OCh/OTSi layer connections) is performed by the SDN-C. Also, the new path calculation in case of failure of the working path
  - The path across the OCh/OTSi network topology is calculated by the controller.
  - The SDN-C must provide asynchronous notifications of the events generated by the network failure which triggers the restoration including, but not limited to, topology changes, service status changes, triggered alarms…
- Use case 8: Permanent protection 1+1 for any of the previous cases (1, 2, 3)
  - The SDN-C must support the provisioning and management of services according to use 5a, 5b, or 5c with a persistent dynamic restoration of the backup service in case of failure.
  - The underlying connection provisioning and management (including lower OCh/OTSi layer connections) is performed by the SDN-C. Also, the new path calculation in case of failure of the working path
  - The path across the OCh/OTSi network topology is calculated by the controller.
  - The SDN-C must provide asynchronous notifications of the events generated by the network failure which triggers the restoration including, but not limited to, topology changes, service status changes, triggered alarms…
- Use case 9: Reverted protection.
  - The SDN-C must allow different reversion capabilities for all resiliency services described in UCs 5a, 5b, 5c, 6a, 6b, 7a, 7b, and 8.

5.3.5 Service maintenance

- Use case 10: Service Deletion for any previous use cases (1, 2, 3, 5, 6, 7, and 8)
  - The SDN-C must support the removal/deletion of an already created service.
  - The SDN-C must notify an error when a non-existing service is attempted to be removed/deleted.
  - The SDN-C makes the underlying deletion.

5.3.6 Notification use cases

- Use case 13a: Subscription to notification stream service.
  - The SDN-C MUST offer a notification subscription service that allows several client applications to subscribe to asynchronous notifications about the changes that occurred in the network.

  - The notification stream service SHALL allow notification filtering to select portions of the information available upon the use case implemented. The filtering system MUST allow filtering by object-type (i.e., Connectivity-Service, Connection…), by networking layer, by notification-type (OBJECT_CREATION, ATTRIBUTE_VALUE_CHANGE, OBJECT_DELETION, ALARM_EVENT, THRESHOLD_CROSSING_ALERT).
  - The same client application MAY request more than one notification stream with different filtering characteristics.
- Use case 14a: Notification of new topology element (topology, link, node, node-edge-point) inserted/removed in/from the network
  - The Notification system MUST emit events exposing the creation/deletion of Topology object-types such as topology, link, node, and node-edge-points.
- Use case 14b: Notification of new connectivity-service element inserted/removed in/from the network.
  - The Notification system MUST emit events exposing the creation/deletion of Connectivity object-types such connectivity-services, connections, and connection-end-points.
- Use case 14c: Notification of new path-computation element inserted/removed in/from the network.
  - The Notification system MUST emit events exposing the creation/deletion of Path-Computation object-types such path-comp-service, or paths.
- Use case 15a: Notification of status change on existing topology element (topology, link, node, node-edge-point) in the network.
  - The Notification system MUST emit events exposing the attribute changes of Topology object-types such as topology, link, node, and node-edge-points.
- Use case 15b: Notification of status change on an existing connectivity-service element in the network.
  - The Notification system MUST emit events exposing the attribute changes of Connectivity object-types such connectivity-services, connections and connection-end-points
- Use case 15c: Notification of status change on the routing conditions of an existing connection element in the network.
  - The Notification system MUST emit events exposing the attribute changes of Connection sub-object-types such ROUTE, SWITCH, and NODE-RULE-GROUPS.

6 Testing

The 2020 Demonstration largely followed the testing methodology used in previous OIF SDN interop testing. The use cases tested this time were significantly more detailed than previous OIF interop tests, led by the development of reference implementation agreements which greatly improved the homogeneity of the use of the APIs tested.

In the preparation phase, participants cooperated in selecting the target use cases from the specifications and define a common test specification to evaluate T-API and OpenConfig API. This included the modeling validation as well as the protocol details, defining the set of procedures and sequence of protocol messages to be exchanged between systems.

This interop has been organized in three different tracks:

1. **Phase I: T-API Testing (SDN-C NBI).** Focus on testing the SDN-C use cases defined by the ONF in TR-547 specification over a single domain, single vendor scenario (no disaggregated).
2. **Phase II: Open Terminals Testing**. Focus on testing automatic operations on the Open Terminals NBI based on OpenConfig models and NETCONF protocol, and its interoperability with commercial SDN-C from a third-party vendor. In this phase, both domain SDN-C and Hierarchical SDN-C interoperability with OTs has been tested.
3. **Phase III: Open Terminals + Open Line System (OLS) End-to-End Demonstration**. Focus on demonstrating end-to-end use cases over a partially disaggregated network using T-API-based RESTCONF NBI for the control of the OLS SDN-C and OpenConfig based NETCONF NBI for the control of the Open Terminals. Due to time and resource constraints, only a subset of the vendors was able to participate in this phase.

The roles of each vendor in this interop for the three testing phases are depicted in Figure 6.

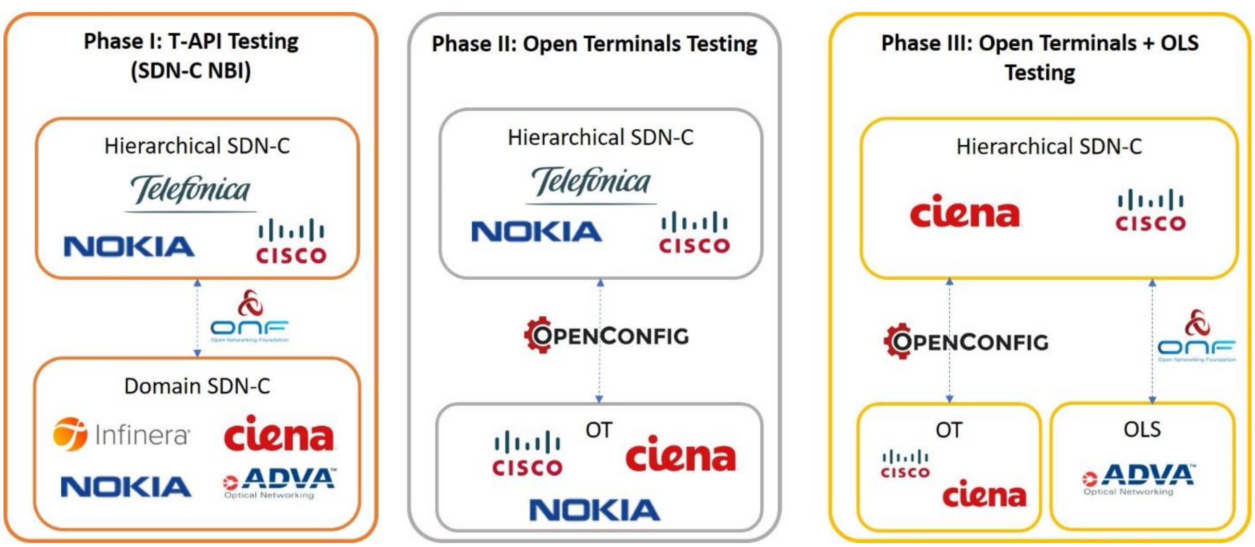


Figure 6: Multi-vendor agnostic network control over standard NBI (T-API)[1].

[1] Telefonica has participated in the interop with a proprietary testing platform.

6.1 PHASE I: SDN-C NBI - RESTCONF T-API TESTING

The first phase was dedicated to the testing of the SDN-C NBI use cases defined by the [ONF TR-547] specification.

6.1.1 RESTCONF validation

- Evaluation criteria:
    - RESTCONF API URIs compliance validation.
    - RETCONF API query filters are supported.
    - RESTCONF Body messages JSON encoding compliance validation (including namespace prefixing rules – [RFC 7951].
    - Validation of the RESTCONF Root tree discovery mechanism defined in [RFC 6415].
    - Validation of the YANG model's discovery mechanism defined in [RFC 7895] according to YANG library model [RFC 8525].

6.1.2 Topology and services Discovery

- Use cases:
    - Use case 0a: Context & Service Interface Points discovery (synchronous mode)
    - Use case 0b: Topology discovery (synchronous mode)
    - Use case 0c: Connectivity Service discovery (synchronous mode)
- Evaluation criteria:
    - Evaluate that all mandatory API URI entries defined in [ONF TR-547] related to tapi-common, tapi-topology and tapi-connectivity model's resources are correctly exposed and fully functional.
    - Evaluate compliance with the Topology abstraction model defined in [ONF TR-547].
        - All network layers must be present as part of the topology OTS, OMS, MC, OTSiMC, OTSi, ODU, DSR according to the standard (NEPs Layer Protocol qualifier validation)
        - Check that the topology model for optical terminals follows either the multi-layer approach or transitional link approach.
        - Review that the number of node objects is according to the testbed scenario considering the T-API Topology abstraction model.
        - Review that the number of link objects is according to the testbed scenario considering the T-API Topology abstraction model.
    - Evaluate compliance with the Connectivity abstraction model defined in [ONF TR-547].
        - Validate that the connection generated to support the Connectivity Service follows the Multi-Layer Hierarchical model defined in the standard.
        - Valid implementation of automatic discovery retrieval lower connection relations.
        - Multi-Layer semantic model validation (all layers are present OTS, OMS, MC, OTSi, ODU, DSR).

- Evaluate the consistency of the implementation, checking that all mandatory attributes for each object, included in [ONF TR-547]., are correctly exposed.
- Retrieve response body does not contain any proprietary extension (if present they must be properly qualified according to standard model)

6.1.3 Connectivity Provisioning

- Use cases:
    - Use case 1a: Unconstrained DSR Service Provisioning single wavelength.
    - Use case 1b: Unconstrained DSR Service Provisioning multi-wavelength.
    - Use case 1d: Unconstrained PHOTONIC_MEDIA/OTSi Service Provisioning
    - Use case 1e: Unconstrained PHOTONIC_MEDIA/OTSiA Service Provisioning
    - Use case 1f: Unconstrained PHOTONIC_MEDIA/MC Service Provisioning
    - Use case 3a: Constrained Provisioning - Include/exclude a node or group of nodes.
    - Use case 3b: Constrained Provisioning -Include/exclude a link or group of links.
    - Use case 3c: Constrained Provisioning - Include/exclude the path used by other service.
- Evaluation criteria:
    - Basic validations for all provisioning operations:
        - Evaluation of the correct creation of the connectivity-service including all the mandatory parameters in the Body of the POST call.
        - Validate that the data retrieved after creation is consistent with all the information provided in the T-API request by the T-API client and that there are not missing attributes or different values.
        - Validate that all the connections generated by the T-API Server after the creation of the CS are discoverable through the lower-connection hierarchy and not further connections are generated without explicit association with the CS.
        - Validate Connectivity Service lower-connection hierarchy. Retrieve response body and validate that the connection generated to support the Connectivity Service follow the Multi-Layer Hierarchical model defined in [ONF TR-547] - Multi-Layer semantic model validation
        - Connectivity-service provisioning is available for all networking service layers: DSR, ODU, OTSi/OTSiA, MC, OTSiMC.
    - Evaluate that topology constraints (where applicable, use case 3x) are correctly applied (routing).

6.1.4 Inventory

- Use cases:

    - Use case 4a: Introduction of references to external inventory model.
    - Use case 4b: Complete Inventory model for NBI Interface.
- Evaluation criteria:
    - Evaluate the correct correlation with legacy inventory models. The INVENTORY_ID attribute must correctly detail absolute inventory references using native inventory IDs of the devices (shelf, slot, port identifiers).
    - Evaluate the tapi-equipment model is fully implemented, including device, equipment, access-ports and physical-span objects, and all logical relationships between objects.
    - Evaluate the correct correlation between logical (tapi-topology, tapi-connectivity) and physical inventory (tapi-equipment) models. This relationship is present through the Node-edge-point to Access-port objects.

6.1.5 Resiliency

- Use cases:
    - Use case 5a: 1+1 OLP OTS/OMS Protection
    - Use case 5b: 1+1 OLP Line Protection with Diverse Service Provisioning
    - Use case 5c: 1+1 Protection with Diverse Service Provisioning (eSNCP)
    - Use case 6a: Dynamic restoration policy for unconstrained and constrained connectivity services.
    - Use case 7a: Dynamic restoration and 1+1 protection of DSR/ODU unconstrained service provisioning.
    - Use case 8: Permanent protection 1+1
- Evaluation criteria:
    - Evaluation of all the basic provisioning validations.
    - Validate traffic after service resilience is triggered according to each use case corresponding resilience mechanism.
    - 1+1 Protection:
        - Validate service disruption of the Connectivity Service when 1+1 Protection (Diverse Service Provisioning) is applied is under 50ms for the first service outage.
        - After the switch re-route is activated, validate the Connection object's switch control is updated accordingly and that the Connection objects affected, reflect the corresponding changes into the Route attribute.
    - Dynamic Restoration:

- Measure service disruption of the Connectivity Service Dynamic restoration after each first cut.
- After the restoration, validate the Connection objects affected, reflect the corresponding changes into the Route attribute.

6.1.6 Planning & Maintenance

- Use cases:
  - Use Case 10: Service deletion (applicable to all previous use cases)
- Evaluation criteria:
  - Deletion of a target connectivity-service and all the related connections created by this service.
  - Evaluate that after deletion, the GET operation of the target connectivity service returns an HTTP 400 internal error (Connectivity Service deleted) have been deleted.

6.1.7 Notifications

- Use cases:
  - Use case 13a: Subscription to notification service.
  - Use case 14a: Notification of new topology element (topology, link, node, node-edge-point) inserted/removed in/from the network
  - Use case 14b: Notification of new connectivity-service element inserted/removed in/from the network.
  - Use case 14c: Notification of new path-computation element inserted/removed in/from the network.
  - Use case 15a: Notification of status change on existing topology element (topology, link, node, node-edge-point) in the network.
  - Use case 15b: Notification of status change on an existing connectivity-service element in the network.
  - Use case 15c: Notification of status change on the switching conditions of an existing connection element in the network.
- Evaluation criteria:
  - Evaluate that the T-API server MUST support the RESTCONF Notifications subscription mechanism is defined in Section 6.3 of [RFC 8040].
  - T-API server must support the "filter" Query Parameter, as defined in Section 4.8.4 of [RFC 8040], to indicate the target subset of the possible events being advertised by the RESTCONF server stream.

- Evaluate that the T-API server supports the tapi-notification subscription mechanism, alternative to RESTCONF Notifications subscription.
- Evaluation that a notification subscription correctly send notifications through Server Sent Events (SSE) [W3C.REC-xml-20081126] standard protocol.
- Evaluate that a given notification subscription service where filtering has been applied does correctly retrieve only T-API Notifications according to the filter.
- Evaluate that the following filters are correctly supported:
    - notification-type = OBJECT_CREATION, OBJECT_DELETION, ATTRIBUTE_VALUE_CHANGE
    - object-type = SERVICE-INTERFACE-POINT, TOPOLOGY, NODE, LINK or NODE_EDGE_POINT, CONNECTIVITY-SERVICE, CONNECTION, CONNECTION-END-POINT, ROUTE, SWITCH, NODE-RULE-GROUPS.

The complete list of use cases is included in the table below.

Use Cases	
Discovery use cases	
Use case 0a:	Context & Service Interface Points discovery (synchronous mode)
Use case 0b:	Topology discovery (synchronous mode)
Use case 0c:	Connectivity Service discovery (synchronous mode)
Provisioning use cases	
Use case 1a:	Unconstrained DSR Service Provisioning single wavelength
Use case 1b:	Unconstrained DSR Service Provisioning multi wavelength
Use case 1d:	Unconstrained PHOTONIC_MEDIA/OTSi Service Provisioning
Use case 1e:	Unconstrained PHOTONIC_MEDIA/OTSiA Service Provisioning
Use case 1f:	Unconstrained PHOTONIC_MEDIA/MC Service Provisioning
Use case 3a:	Constrained Provisioning - Include/exclude a node or group of nodes
Use case 3b:	Constrained Provisioning -Include/exclude a link or group of links
Use case 3c:	Constrained Provisioning - Include/exclude the path used by other service
Inventory use cases	
Use case 4a:	Introduction of references to external inventory model
Use case 4b:	Complete Inventory model for NBI Interface
Resilience use cases	
Use case 5a:	1+1 OLP OTS/OMS Protection
Use case 5b:	1+1 OLP Line Protection with Diverse Service Provisioning
Use case 5c:	1+1 Protection with Diverse Service Provisioning (eSNCP)
Use case 6a:	Dynamic restoration policy for unconstrained and constrained connectivity services

Use case 7a:	Dynamic restoration and 1+1 protection of DSR/ODU unconstrained service provisioning
Use case 8a:	Permanent protection 1+1 for use cases
Service maintenance use cases	
Use case 10a:	Service deletion (applicable to all previous use cases)
Alarms and Notification use cases	
Use case 13a:	Subscription to notification service
Use case 14a:	Notification of new topology element (topology, link, node, node-edge-point) inserted/removed in/from the network
Use case 14b:	Notification of new connectivity-service element inserted/removed in/from the network
Use case 14c:	Notification of new path-computation element inserted/removed in/from the network
Use case 15a:	Notification of status change on existing topology element (topology, link, node, node-edge-point) in the network
Use case 15b:	Notification of status change on existing connectivity-service element in the network
Use case 15c:	Notification of status change on the switching conditions of an existing connection element in the network

6.2 Phase II: Open Terminals NBI – OpenConfig Testing

The second phase was dedicated to the testing of the OpenConfig interface interoperability between SDN-C and OTs. To this aim, a test plan based on Telefonica's specification based on atomic NETCONF/OpenConfig operations has been used for the evaluation in this interop.

6.2.1 NETCONF/OpenConfig interface atomic operations validation

The first testing part required for interoperability is the protocol validation. The NETCONF protocol has been employed for configuration and data retrieval, the correctness of the implementation has been evaluated based on the following criteria.

- Evaluation criteria:
    - Check if the authentication with username and password is enabled.
    - Check if port 830 is enabled to exchange NETCONF messages.
    - Check if the capabilities exchange is supported through the <hello> rpc message.
    - Check if the supported NETCONF version is 1.0. Version 1.1 can be supported too, but it will not be used.
    - Check if the <get-config> rpc is correctly implemented. Only configuration parameters must be retrieved. It will be tested over both configurations, candidate and running.
    - Check if the <get> rpc is correctly implemented. Both types of parameters, configuration, and running state must be retrieved.
    - Check if the <edit-config> rpc is correctly implemented. Only parameters that can be modified by using this rpc. It will be tested over both datastores, candidate, and running. For the edition

of the configuration, a candidate and running datastore are required. Figure 7 describes the correct workflow required for editing the configuration of a device.

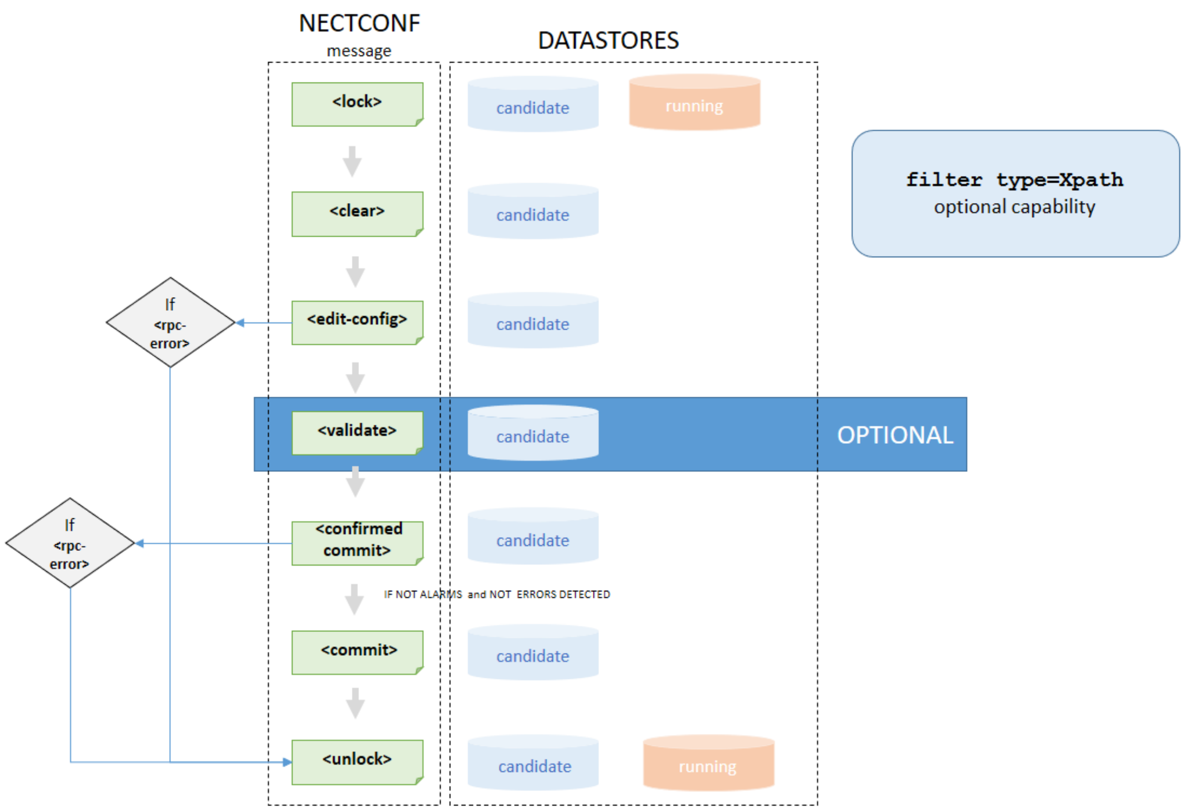


Figure 7: NETCONF Configuration workflow

- Check if the <commit> rpc is correctly implemented. Running datastore configuration should be fully overwritten by the candidate configuration.
- Evaluate NETCONF capabilities are correctly implemented according to [RFC 6241]:
    - Check if capability "Writable-running" is correctly implemented.
    - Check if capability "Candidate Config" is correctly implemented.
    - Check if capability "Confirmed-commit" is correctly implemented.
    - Check if capability "Rollback-on-error" is correctly implemented.

The following table includes the complete list of atomic operations validated during the interop.

Discovery operations <get>, <get-config> operations	
Discovery-1:	Discovery of components name
Discovery-2:	Discovery type of components
Discovery-3:	Discovery of line and client ports and subcomponents

Discovery-4:	Discovery of operational modes available
Discovery-5:	Discovery of admin and operational status of port
Discovery-6:	Discovery of index of logical channel in client side
Discovery-6b:	Discovery of capabilities of client logical channel
Discovery-7:	Discovery of line port capabilities (protocol, bit rate, granularity available and allocation)
Discovery-8:	Discovery of frequency, target-out-power and operational mode configured in line port
Discovery-9:	Discovery of type of logical channel assignment (preset or flexible)
Discovery-10:	Discovery of logical channel-assignment
Discovery-10b:	Discovery of logical channel-assignment
Discovery-11:	Discovery of location of a component and mapping to INVENTORY_ID (TAPI)
Discovery-12:	Retrieve of global structure of HW inventory
Discovery-13:	Retrieve of HW information of each component
Discovery-15:	Retrieval of q-value of optical channel
Provisioning operations	
Provisioning-1:	Provisioning of admin-state enable of client port.
Provisioning-2:	Provisioning of admin-state enable of line port
Provisioning-3:	Provisioning of frequency in line port
Provisioning-4:	Provisioning of optical output power in line port
Provisioning-5:	Provisioning of operational mode
Provisioning-6:	Basic configuration of a logical channel
Provisioning-8:	Assignment of logical channel in another logical
Provisioning-11:	Configuration of rate (rate-class), protocol (trib-protocol and logical channel type) on client port
Service deletion operations	
Delete-1:	Disable the client port
Delete-2:	Disable the line port
Delete-3:	Disable the logical channel and change assignment

6.2.2 gNMI/gRPC OpenConfig interface atomic operations validation

For the performance monitoring, functionality validation has been tested over the gNMI/gRPC protocol, employing streaming telemetry. Streaming telemetry is a new paradigm in monitoring the health of a network. It provides a mechanism to efficiently stream configuration and operational data of interest from NE. This streamed data is transmitted in a structured format to remote management systems for monitoring and troubleshooting purposes.

The following operations have been tested to evaluate the streaming telemetry functionality over gNMI/gRPC interface:

Streaming (gNMI) operations	
gRPC-1:	Subscribe for all relevant paths with gNMI/gRPC. See table
gRPC-2:	Create subscription for specific path with gNMI
gRPC-3:	Create subscription of a specific path with a defined SAMPLE (sample_interval)

gRPC-4:	Subscribe for all paths with gNMI/gRPC

The subscriptions to test the telemetry capabilities of the OTs devices are listed in the following table:

PM counters	
severely-errored-seconds	/terminal-device/logical-channels/channel/otn/state/severely-errored-seconds
unavailable-seconds	/terminal-device/logical-channels/channel/otn/state/unavailable-seconds
fec-uncorrectable-blocks	/terminal-device/logical-channels/channel/otn/state/fec-uncorrectable-blocks
fec-uncorrectable-words	/terminal-device/logical-channels/channel/otn/state/fec-uncorrectable-words
fec-corrected-bytes	/terminal-device/logical-channels/channel/otn/state/fec-corrected-bytes
fec-corrected-bits	/terminal-device/logical-channels/channel/otn/state/fec-corrected-bits
background-block-errors	/terminal-device/logical-channels/channel/otn/state/background-block-errors
Other counters	
oc-opt-term:chromatic-dispersion	/components/component/oc-opt-term:optical-channel/oc-opt-term:state/oc-opt-term:chromatic-dispersion
oc-opt-term:polarization-mode-dispersion	/components/component/oc-opt-term:optical-channel/oc-opt-term:state/oc-opt-term:polarization-mode-dispersion
Monitoring counters	
output-power	/…/output-power/* (transceiver, optical-channel…)
input-power	/…/input-power/* (transceiver, optical-channel…)
pre-fec-ber	/terminal-device/logical-channels[index=A]/channel/otn/state/pre-fec-ber/*
post-fec-ber	/terminal-device/logical-channels/channel/otn/state/post-fec-ber/*
temp	/…/temp/*
q-value	/…/state/q-value/*
General counters	
interface	/interfaces/interface/state/counters/*

6.3 PHASE III: OPEN TERMINALS + OPEN LINE SYSTEM (OLS) END-TO-END (E2E)-DEMONSTRATION

The last phase of testing was dedicated to demonstrating the partially disaggregated architecture solution end-to-end over multi-vendor setups. The general architectures demonstrated were the ones described in Section 5.2, the vendor distribution is generally shown in Figure .

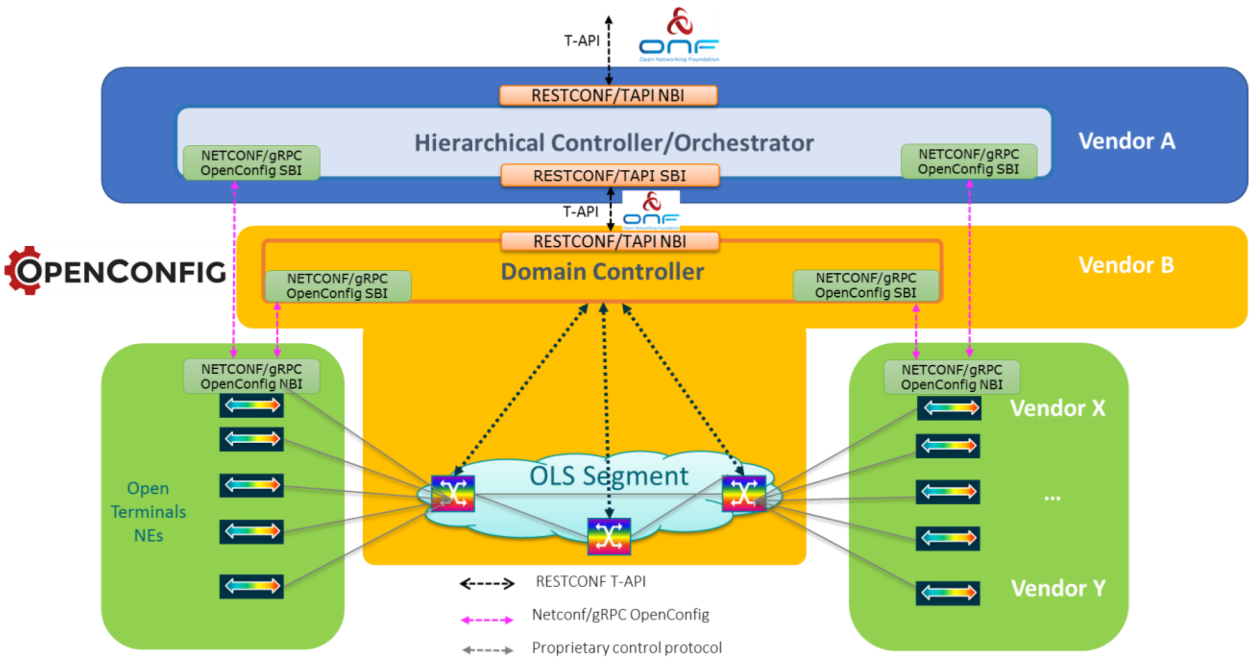


Figure 8: Vendors' distribution in a Phase III testing.

The list of target use cases to be demonstrated in this Phase III were all included in Section 5.3, however, given the level of maturity of the existing vendors' solutions, the target was refined to the following shortlist:

Use Cases	
Discovery use cases:	
Use case 0a:	Context & Service Interface Points discovery (synchronous mode)
Use case 0b:	Topology discovery (synchronous mode)
Use case 0c:	Connectivity Service discovery (synchronous mode)
Provisioning use cases:	
Use case 1a:	Unconstrained DSR Service Provisioning single wavelength.
Use case 3a:	Constrained Provisioning - Include/exclude a node or group of nodes
Use case 3b:	Constrained Provisioning - Include/exclude a link or group of links
Use case 3c:	Constrained Provisioning - Include/exclude the path used by other service
Inventory use cases:	

Use case 4a:	Introduction of references to external inventory model
Use case 4b:	Complete Inventory model for NBI Interface
Resilience use cases:	
Use case 6a:	Dynamic restoration policy for unconstrained and constrained connectivity services
Service maintenance use cases:	
Use case 10a:	Service deletion (applicable to all previous use cases)

The resulting E2E setups and demonstrations between suppliers were internally showcased among the participants and other invited operators.

The common evaluation criteria adopted for the selected use cases are described in the following subsections.

6.3.1 E2E Network discovery validation

- Use cases:
    - Use case 0a: Context & Service Interface Points discovery (synchronous mode)
    - Use case 0b: Topology discovery (synchronous mode)
    - Use case 0c: Connectivity Service discovery (synchronous mode)
- Evaluation criteria:
    - Validate the automatic discovery of OLS network resources into the H-SDN-C or OLS SDN-C through T-API.
    - Validate the automatic discovery of OT's configuration and state data by the H-SDN-C or OLS SDN-C through T-API.
        - Verify correct translation into NBI's model (T-API).
    - Validate the discovery between OLS and OT interfaces and its characterization.
    - Validate the discovery of E2E services over the disaggregated network including the correct representation of DSR, ODU, OTSi, and Media-channel layers.
        - Verify correct translation into NBI's model (T-API).

6.3.2 E2E Connectivity Services validation

- Use cases:
    - Use case 1a: Unconstrained DSR Service Provisioning single wavelength.
    - Use case 3a: Constrained Provisioning - Include/exclude a node or group of nodes.
    - Use case 3b: Constrained Provisioning -Include/exclude a link or group of links.
    - Use case 3c: Constrained Provisioning - Include/exclude the path used by another service.
- Evaluation criteria:
    - Validate that the E2E service is implemented by a tapi-connectivity:connectivity-service request between service-interface-points exposed by the H-SDN-C or OLS SDN-C NBI.

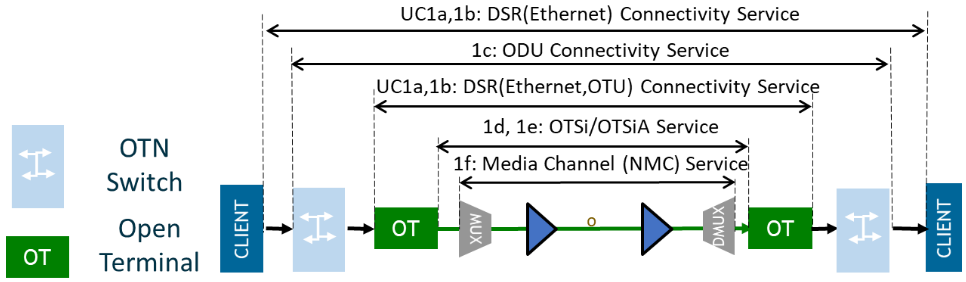
- Depending on the disaggregated architecture the H-SDN-C or OLS SDN-C will:
    - Setup a bidirectional Media Channel (MC) in the OLS segment between the corresponding Add/Drop ports through a call to the OLS SDN-C (Option B) or by directly configuring the Network elements (Option A).
    - And, to configure (using OpenConfig model and NETCONF protocol) the line ports OTSI transmission and the client-to-line logical channel configurations representing the cross-connection at the disaggregated OTs.
- Validates that SDN-C performs the underlying connection provisioning and management, including lower layer connections e.g., ODU, OTSi/OTSiA, MC:
    - Digital Signal Rate (DSR) services between client interfaces of terminal devices i.e., 1 GE, 10 GE, 100GE, FlexE, STM-X, Fibre Channel.
    - ODU/OTU digital services.
    - OCh/OTSi/OTSiA, which can be single or multi-carrier.
    - Network Media Channels (NMC) defined as: "the media channel is a topological construct that represents both the path through the media and the resource (frequency slot) that it occupies." in [ITU-T G.872].
    - 

Figure 9: T-API UC1 network layer stack over a partially disaggregated network.

- Validates that the path of each lower layer connection (e.g., ODU or OCh/OTSi, OMS) across the network topology is calculated by the SDN-C, and the connection is automatically provisioned.
    - The route of each lower layer connection (e.g., OCh/OTSi, Photonic Media) across the network topology is calculated by the SDN Domain controller considering the constraints introduced in the service request. The route in each layer is recursively calculated from the lower layer to the upper layer, excluding/including ALL those components of the topology shared with the connections created to support the selected service.

6.3.3 E2E Inventory validation

- Use cases:
    - Use case 4a: Introduction of references to external inventory model.

    - Use case 4b: Complete Inventory model for NBI Interface.

- Evaluation criteria:
    - Validates that the OLS T-API equipment model is correctly reflecting the OLS network physical inventory.
        - Validates that the OLS T-API equipment model is correctly exposed through the NBI of the H-SDN-C and OLS SDN-C.
    - Validates that InventoryID tags are appropriately included within the tapi-topology/owned-node-edge-points and tapi-common/service-interface-points within the OLS T-API implementation.
        - Validates that InventoryID tags are correctly exposed through the NBI of the H-SDN-C and OLS SDN-C.
    - Validates that the H-SDN-C or OLS SDN-C is correctly discovering OT's physical inventory through the OpenConfig model's interface.
    - Validates that the H-SDN-C or OLS SDN-C is correctly translating OT's physical inventory OpenConfig information into T-API NBI models.

6.3.4 E2E Resiliency validation

- Use cases:
    - Use case 6a: Dynamic restoration policy for unconstrained and constrained connectivity services.
- Evaluation criteria:
    - Evaluation of all the basic provisioning validations.
    - Evaluation of the restoration mechanism is implemented in a centralized manner by the H-SDN-C or OLS SDN-C depending on the architecture.
    - Evaluate that the restoration control mechanism is correctly triggered in case of a failure of:
        - Fiber cut/damage
        - Amplifier outage.
        - Mux/ROADM, outage
        - Transponder outage
    - Evaluate that the SDN-C does correctly implement the restoration workflow when:
        - There is no change of frequency in the new route.
        - Change of frequency in the new route (OT's optical channel dynamic re-configuration is needed).

    - Evaluate that connection's route after the restoration process occurs is correctly discoverable through the H-SDN-C and/or OLS SDN-C NBIs.

6.3.5 E2E Service deletion validation

- Use cases:
    - Use case 10: Service deletion (applicable to all previous use cases)
- Evaluation criteria:
    - Evaluate that the H-SDN-C and/or OLS SDN-C is correctly processing the connectivity-service DELETE RESTCONF operation over any of the previous connectivity-services.
    - Validate that after correctly processing the request, both OLS connectivity-service (media-channel) and OT's configuration are correctly removed.
    - Evaluate that if the user is not authorized to delete the target resource, then an error response containing a "403 Forbidden" status-line SHOULD be returned. The error-tag value "access-denied" is returned in this case. A server MAY return a "404 Not Found" status-line, as described in Section 6.5.4 in [RFC7231]. The error-tag value "invalid-value" is returned in this case.

7 DETAILED FINDINGS

Key findings of the testing program follow. Interoperability issues that required the most attention are cited as well as observations related to compliance and adoption of the API specs by the participating vendor pool.

7.1 T-API

- Quick adoption of ONF TR-547 reference implementation agreement was found.
    - RESTCONF compliant API was supported on all vendors with few deviations found.
    - T-API v2.1.3 models were implemented by all vendors
    - Notification subscription mechanism based on tapi-notification model was widely supported.
    - SSE notification format was widespread across implementations. Websockets (as an alternative) was supported as well.
- Maturity:
    - Quite a high percentage of use cases were supported across the vendor participants (64% use cases supported on average).
    - A high level of maturity was found across products. Production-ready products were interoperable to a large extent and product implementation supported exhaustive testing.
- Main interoperability issues:
    - IETF RESTCONF
        - Standard authentication for RESTCONF APIs was not consistently enforced.
        - API scalability for large data trees needs to be addressed by RESTCONF filters or Streaming rather than through RESTCONF deviations.
        - RESTCONF provisioning behavior for POST operations into lists needs to be clarified.
        - UUID keys provisioning by RESTCONF client was not consistently supported.
        - Standard RESTCONF error codes must be sent if a request cannot be served.
    - ONF TR-547
        - Provisioning lifecycle is still not uniform. Once the RESTCONF request is accepted, there must be a standardized lifecycle for the connectivity-service, uniform for all implementations.
        - Reduction in the number of API operations required when following relations between tables through T-API enhancement is desired.

7.2 OPENCONFIG

- Very good level of compliance to standard OpenConfig models
    - Uniform HW components hierarchy relation using parent and location attributes.
    - Cross-connection logic discovery was widely implemented.
    - Optical channel configuration (Frequency, power, and operational mode) 100% supported.
    - Performance indicators are not fully supported yet. At least pre-fec-ber and/or q-value need to be supported.
- NETCONF and gNMI
    - NETCONF standard is widely supported across the industry.

    - Subscription and notifications for performance streaming telemetry compliant with both gNMI and NETCONF were demonstrated.
- Main interoperability issues:
    - IETF NETCONF:
        - Standard handshake for SSH was not consistently supported. Key exchange algorithms were different and were not fully standard.
        - Candidate datastore MUST be supported and is not fully supported across the industry.
    - OpenConfig:
        - openconfig-transport-line-common version 0.6.0 is not widely supported; instead, a previous version is commonly adopted (0.5.2)
    - Further standardization needed for complex use cases:
        - The commissioning and provisioning process for devices MUST be uniform. OTN Logical channel cross-connections flexibility is hard to manage due to strong dependency on different hardware capabilities
        - HW Dependencies does not allow some configurations. This must be clarified for understanding configuration workflows.

8 BENEFITS

The interoperability testing program provides several benefits for network operators and vendors.

Participant and industry benefits:

- Realize operator targets for network control and management
- Provide a venue for Operator and Vendor cooperation
- Identify and solve interoperability issues using standard APIs
- Accelerate maturity of Transport SDN standards and deployment

Established open Transport SDN APIs can help network operators:

- Improve network agility to adapt to dynamic service demands and traffic patterns
- Improve service provisioning and time-to-revenue
- Reduce maintenance and management with simplified control and automation

9 CONCLUSION

The OIF 2020 Transport SDN API Interoperability Demonstration achieved the objectives to evaluate interoperability over standard interfaces. Participants complemented limited on-site technical support with innovative virtual lab testing to overcome resource, lab access, and communication challenges presented by the coronavirus pandemic. Despite the challenges, participants supported (21) integrations between vendor solutions in (29) T-API use cases and (31) OpenConfig operations over a 10-week testing period.

Specifically, the testing successfully evaluated agnostic control over a standard NBI (T-API) using an open optical partially disaggregated network architecture utilizing open line systems and open terminals with single and hierarchical SDN controller approaches in common use cases identified by the ONF TR-547 T-API Reference Implementation Agreement.

The testing found that the T-API spec is at an acceptable maturity level to allow participating, early adopter network operators to continue aggressive commercial deployment plans and for all operators to adopt T-API as a standard NBI in planning open optical transport SDN networks deployments. Continued enhancements to the spec will enable accelerated, broad adoption of T-API.

The testing demonstrated a very good level of OpenConfig model compliance and both NETCONF and gNMI are widely supported. A few major interoperability issues identified gaps in the model that require further work to allow broader adoption, especially for complex use cases.

10 Appendices

10.1 List of Contributors

- Dave Brown (Editor), Nokia
- Nigel Davis, Ciena
- Victor Lopez, Telefónica
- Michael Maddern, Cisco
- Arturo Mayoral, Telefónica
- Lyndon Ong, Ciena
- Jonathan Sadler, Infinera

10.2 About the OIF

OIF is where the optical networking industry's interoperability work gets done. Building on 20 years of effecting forward change in the industry, OIF represents the dynamic ecosystem of 100+ industry-leading network operators, system vendors, component vendors, and test equipment vendors collaborating to develop interoperable electrical, optical, and control solutions that directly impact the industry's ecosystem and facilitate global connectivity in the open network world. Connect with OIF at @OIForum, on LinkedIn, and at http://www.oiforum.com.

10.3 T-API Supplement

T-API is a standard NorthBound Interface 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 discussed below, T-API has also been adopted by other industry SDOs and forums for their specific needs.

As a component of Transport SDN, T-API 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.

Some of the unique benefits of T-API include:

- T-API 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.
- T-API is based on telecom management models that are familiar to telecom equipment vendors and network operations staff, making its adoption easier and reducing disruption of network operations.

- T-API combines both standards specification development and open-source software development, providing code for implementation and testing that allows faster feature validation and incorporation into vendor and carrier software and equipment.

10.3.1 History

The T-API project was initiated at ONF in 2014, following a successful multi-vendor multi- carrier prototype demonstration of Transport SDN architecture jointly held with OIF. A determination from the testing was that the NBI presented a critical interface for SDN deployment across domains with different equipment and capabilities.

At the same time, a technology-independent Common Information Model (CIM) project in ONF provided a framework for the T-API information model built on many years of experience in the development of management interfaces at TMF, ITU-T, and other SDOs. T-API was agreed to be a purpose-specific, use-case-driven Interface Profile Specification of the ONF Core Information Model that would reflect the concepts, patterns, and evolution of the Core IM.

10.3.2 Design/Framework

The work on CIM and T-API is part of an overall vision to develop interfaces and APIs for SDN that are based on an "invariant" model, i.e., one that is independent of the protocols and technologies in current fashion and can be flexibly mapped to the protocol using model-based automated code generation rather than costly, time-consuming handcrafting of interface software. This approach aims to avoid the creation of "software silos" that could be restrictive to operators just as vendor-based silos are today.

The T-API model is accordingly defined in the same manner as the CIM, using the Unified Modeling Language (UML). UML is a well-known, stable standard (ISO) and is familiar to people with experience in transport network management standards, as well as many other industries and sectors close to telecom. The use of UML allows an easy bridge to ITU-T, TMF, MEF, and NFV management modeling work – it should be noted that UML is also in use in some IETF groups.

UML is supported by open source tooling which allows the graphical presentation of the models. The two-dimensional representation in UML makes objects and especially the relationships between objects easier to visualize than a linear text specification such as a YANG model. Tooling also supports constriction of consistent views that ease explanation and understanding

UML is protocol-independent and can be mapped to different data schema languages and associated wireline protocols. UML provides similar descriptive properties to data modeling languages such as YANG, including specification of object classes, superclasses, and stereotypes, attribute types, rw vs. ro characteristics, ability to transfer attributes by reference, interfaces, operations, and notifications. A multi-organizational effort has been underway to specify the mapping from UML to YANG to make the mapping process capable of being done via software rather than being a manually intensive process [ISOMII].

10.4 REFERENCES

[ONF CIM] ONF Core Information Model, TR-512, https://www.opennetworking.org/wp-content/uploads/2018/12/TR-512_v1.4_OnfCoreIm-info.zip

[ONF TR-547] TAPI v2.1.3 Reference Implementation Agreement, Version 1.0 June 2020, https://opennetworking.org/wp-content/uploads/2020/08/TR-547-TAPI-v2.1.3-Reference-Implementation-Agreement-1.pdf

[ITU-T G.872] G.872: Architecture of optical transport networks (12/19)

[RFC 7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>

[RFC 6415] Hammer-Lahav, E. and B. Cook, "Web Host Metadata", RFC 6415, October 2011, <http://www.rfc-editor.org/info/rfc6415>.

[RFC 7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", RFC 7895, DOI 10.17487/RFC7895, June 2016, <http://www.rfc-editor.org/info/rfc7895>.

[RFC 8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/info/rfc8525>.

[RFC 8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[W3C.REC-eventsource-20150203] Hickson, I., "Server-Sent Events", World Wide Web Consortium Recommendation REC-eventsource-20150203, February 2015 Considerations <http://www.w3.org/TR/2015/REC-eventsource-20150203>

[RFC 6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.