

***Introducing network programmability into
OTN/WDM networks***

Transport PCE

***a full open source approach based on
OpenDaylight & Open ROADM***

27th April 2020

**Olivier Renais, Guillaume Lambert, Orange Labs
Shweta Vachhani, Dhruv Bhardwaj, Bala Bathula, Martin Birk, AT&T**

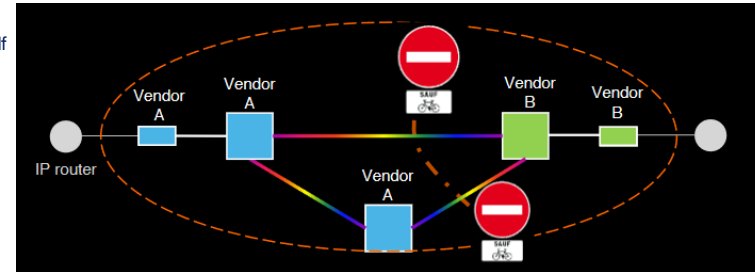
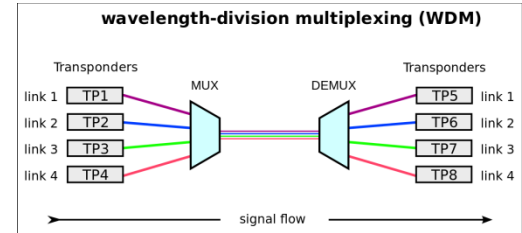


AGENDA

- **Context**
- **Project initial goals**
- **Global architecture**
- **Main modules description**
- **Current status**
- **Planned developments**
- **TransportPCE challenges to become a managed project**

Transport PCE context

- an open-source SDN controller for WDM and OTN optical core networks (not access nor FTTH)
- **WDM = Wavelength-Division Multiplexing**
 - At the edge of the optical domain, packet-based signals are converted into WDM wavelengths (= laser colors), the associated devices are named “Xponders”(transponder or muxponder)
 - Then the signal is “routed” up to another Xponder by ROADMs (Reconfigurable Optical Add Drop Multiplexer)
- **OTN = Optical Transport Network**
 - OTN hierarchy was introduced to address robust higher rate optical transport in the following of SDH/SONET.
 - It is a layer 1 technology (WDM is layer 0). It provides **aggregation capability** (switchponders), alarm & monitoring as well as protection and restoration mechanisms inherited from SDH/SONET
- **Lack of Interoperability**
 - WDM devices interoperability has been a second-choice criteria behind performance for too much time.
 - contrary to IP and Ethernet, different manufacturers cannot be mixed easily inside a same WDM domain.
 - Many reasons for that: most details available at ONS NA 2018 presentation <https://events19.linuxfoundation.org/wp-content/uploads/2018/07/ONSNA-TransportPCE-Full-Interworking-in-Optical-Networks-.pdf>



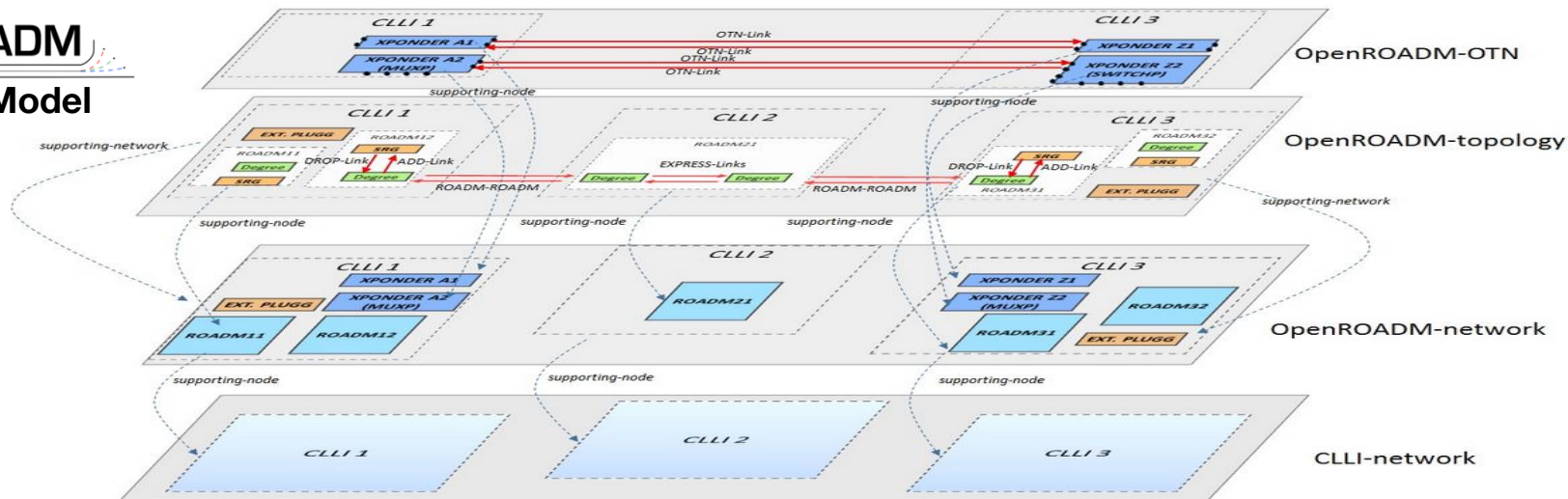
Transport PCE initial Goals

- **Provide the Opensource community with a Controller for Optical infrastructure based on open standards**
 - Allowing Service providers to avoid vendor implementation lock-in
 - Reducing CAPEX by letting the competition acts
 - Reducing the OPEX and the costs of equipment integration
 - Improving time to market
- **Promoting interoperability since open control / device configuration does not imply interoperability**
 - Many good initiatives for open control but very few that guarantees the interoperability because of WDM specificities
 - OpenROADM was the most natural choice for TransportPCE primary implementation
 - It provides the most complete set of models (Device, Network Topology, Service) needed for interop
- **Create an emulation which contributes in accelerating innovation in this domain**
 - Optical domain NMS and controllers are currently deployed in Silos
 - WAN-SDN offers a chance to open data, control and management planes of transport networks
 - Leaning on Openness of data model to develop a common platform for the control a wide diversity of systems/vendors
 - Driving through data plane and control plane specifications interoperability, to limit the development effort
 - Sharing the development effort between multiple contributors to build fast a common base
- **Propose code and tests for a reference implementation that can be reused in third-parties derived products**
 - Standardize with YANG models the API between the various components in an optical controller
 - Providing feedbacks and a proof-of-concept for the most significant Open initiatives

The Open ROADM choice

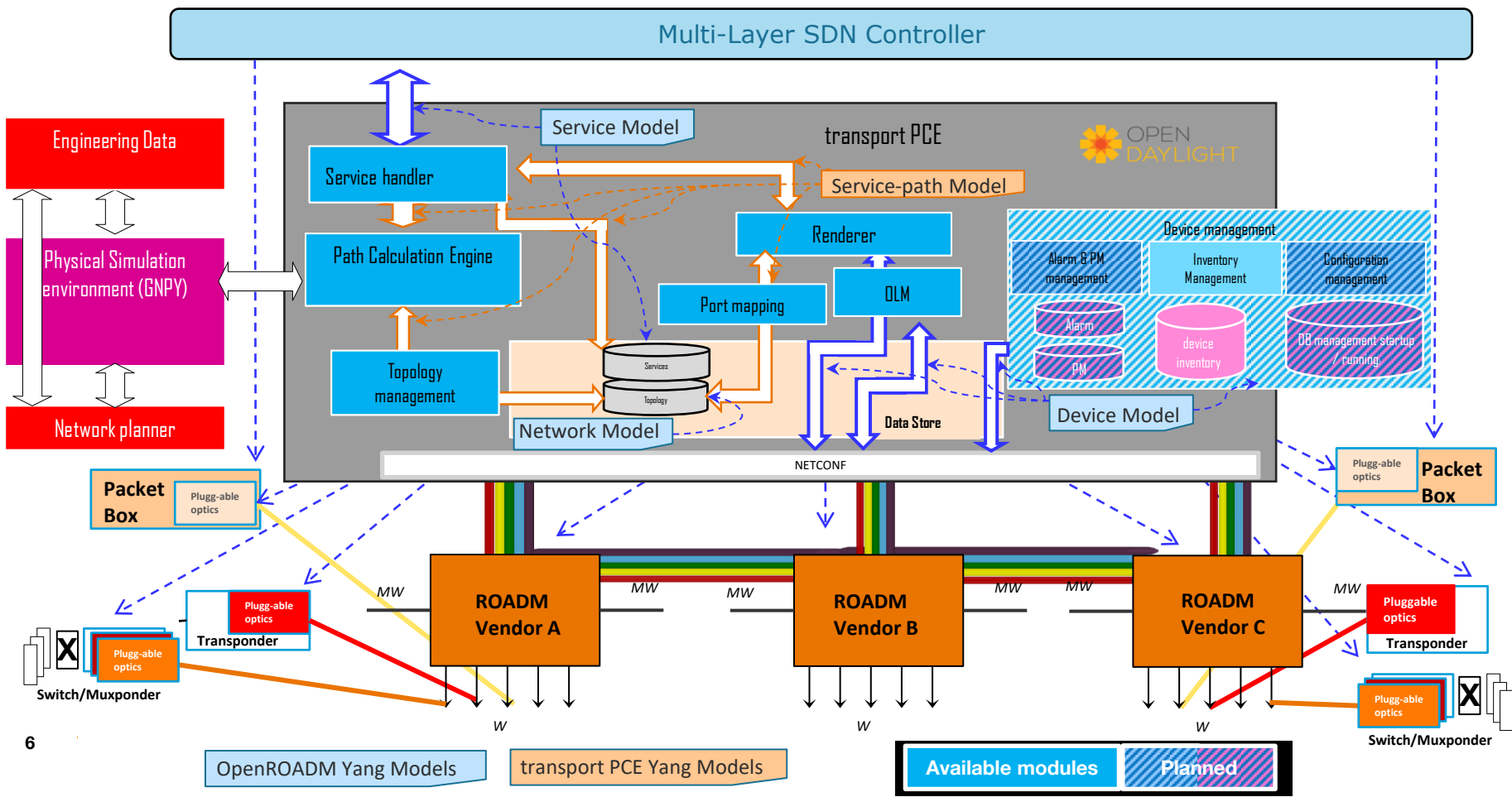
- Open ROADM Multi-Service Agreement (MSA) provides comprehensive and coherent YANG models, not only for device control, but also for network topology & service management.
- It also defines specifications for the optical layer.
- The disaggregation of ROADM & Xponders building blocks, provides a high level of interoperability

Open ROADM Network Model

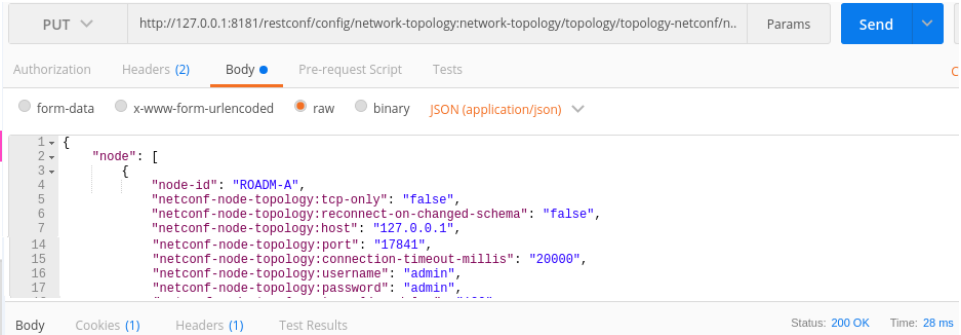


- Interoperability is a key enabler for automation : Open ROADM turn out to be the most natural choice for tpce implementation

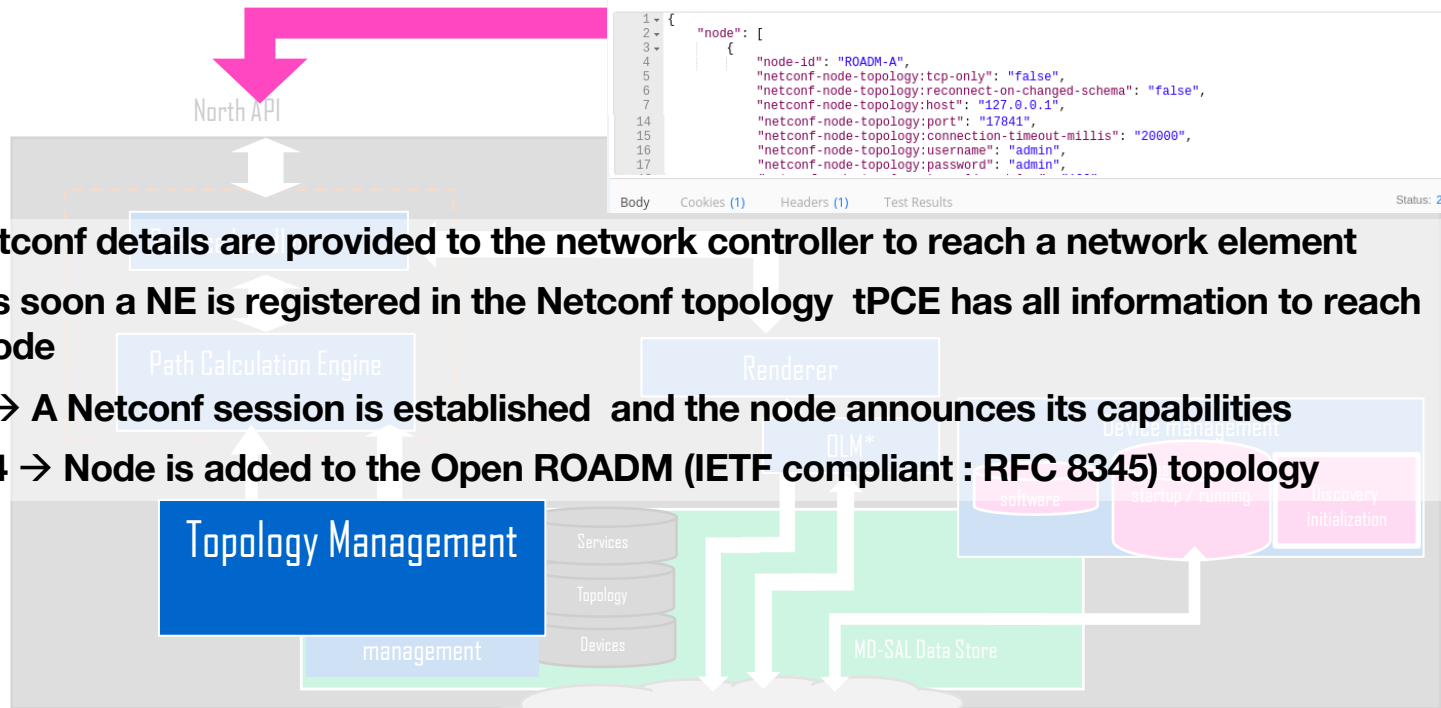
TransportPCE architecture



Topology management



- 1 → Netconf details are provided to the network controller to reach a network element
- 2 → as soon a NE is registered in the Netconf topology tPCE has all information to reach the node
- 3 → A Netconf session is established and the node announces its capabilities
- 4 → Node is added to the Open ROADM (IETF compliant : RFC 8345) topology

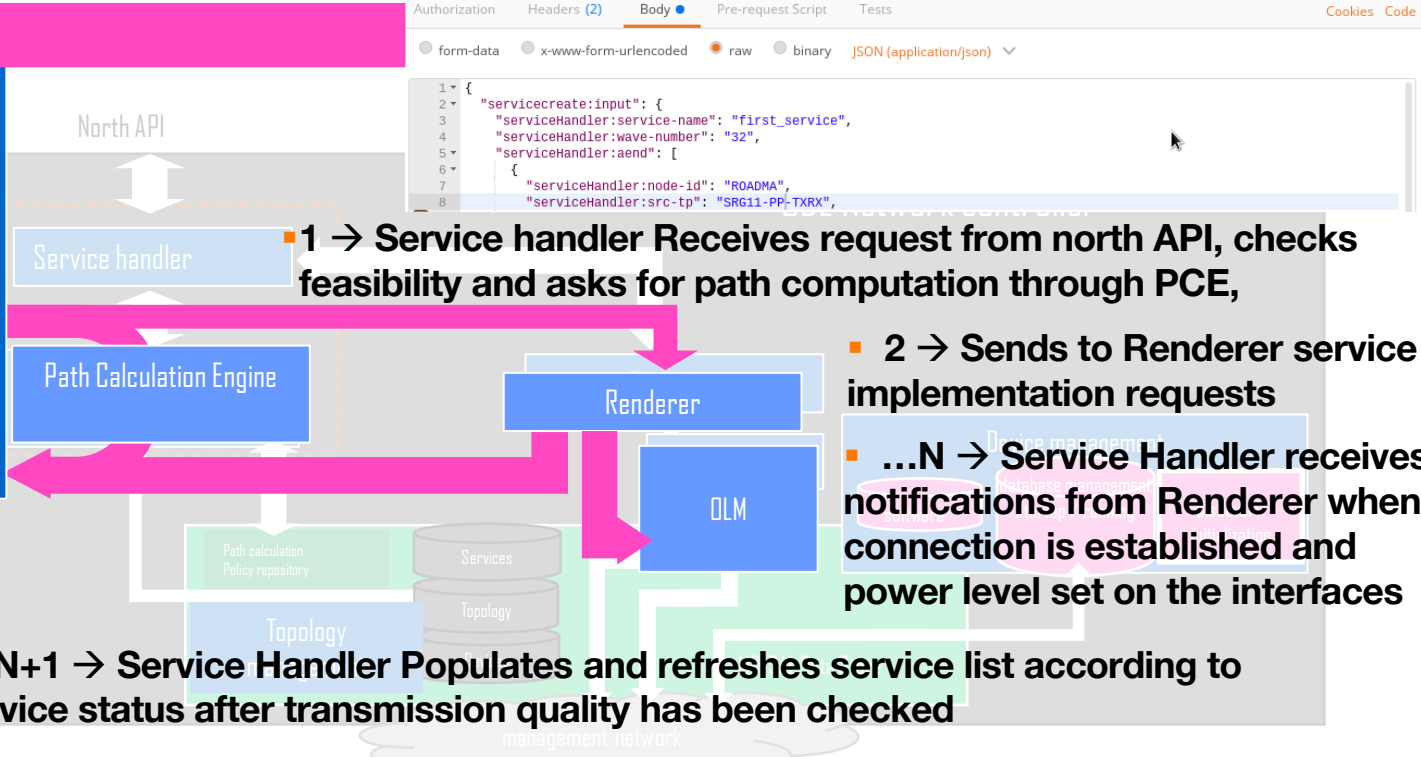


- As soon as Netconf nodes are in the Netconf topology of the controller, node discovery and Open ROADM topology building is based on equipment connection/disconnections
- Link discovery is based on Ildp
 - Each nodes knows its Ildp neighbors



Service Handler

```
PUT http://(odlip):8181/restconf/config/serviceHandler:serviceCreate
Authorization
Headers (2)
Body
Pre-request Script
Tests
Cookies
Code
form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)
1 {
2   "servicecreate:input": {
3     "serviceHandler:service-name": "first_service",
4     "serviceHandler:wave-number": "32",
5     "serviceHandler:aend": [
6     {
7       "serviceHandler:node-id": "ROADMA",
8       "serviceHandler:src-tp": "SRG11-PP1-TXRX",
```



- 1 → Service handler Receives request from north API, checks feasibility and asks for path computation through PCE,
- 2 → Sends to Renderer service implementation requests
- ...N → Service Handler receives notifications from Renderer when connection is established and power level set on the interfaces
- N+1 → Service Handler Populates and refreshes service list according to service status after transmission quality has been checked

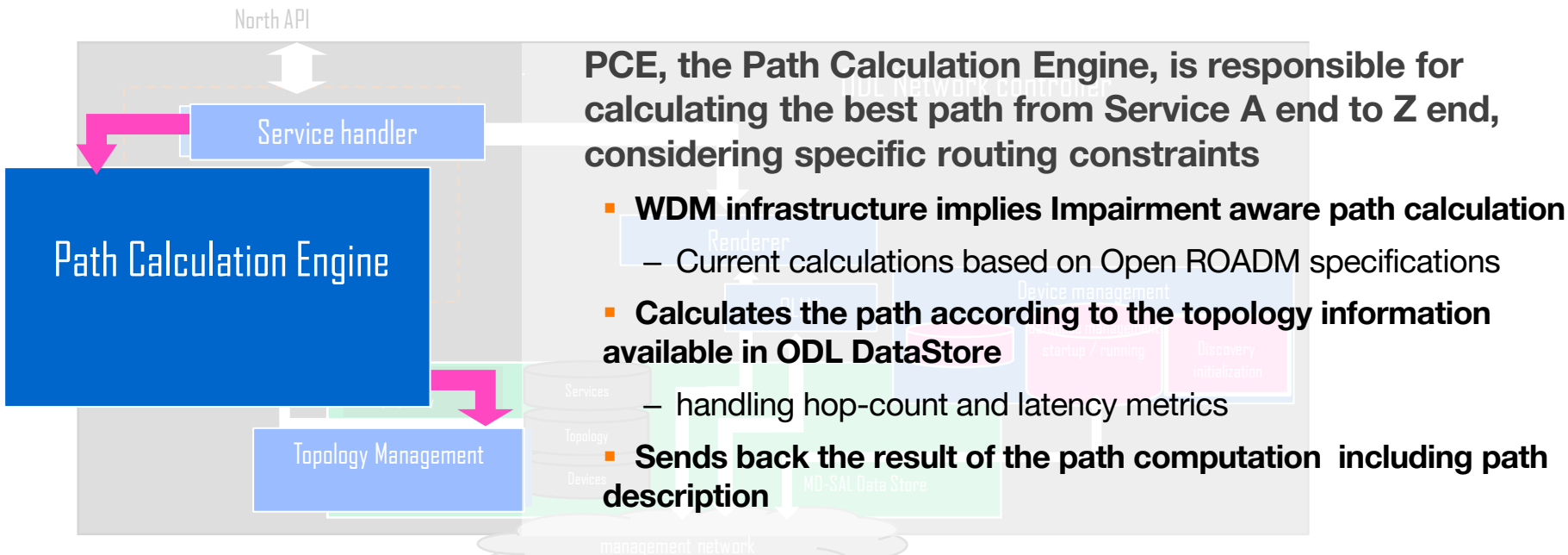
SH Manages service handling request coming from the North RESTCONF API

- Service creation, deletion, restoration, reconfiguration and rerouting...



PCE

- The PCE receives from service handler request for path computation, or service feasibility check



PCE, the Path Calculation Engine, is responsible for calculating the best path from Service A end to Z end, considering specific routing constraints

- **WDM infrastructure implies Impairment aware path calculation**
 - Current calculations based on Open ROADM specifications
- **Calculates the path according to the topology information available in ODL DataStore**
 - handling hop-count and latency metrics
- **Sends back the result of the path computation including path description**

The following constraints are handled in the latest version of tpce

- **General constraints:** exclude Node/SRLG, maximum latency
- **Diversity constraints:** with respect to the path(s) of a specific service or a list of service

Renderer



Service-path (transportPCE)

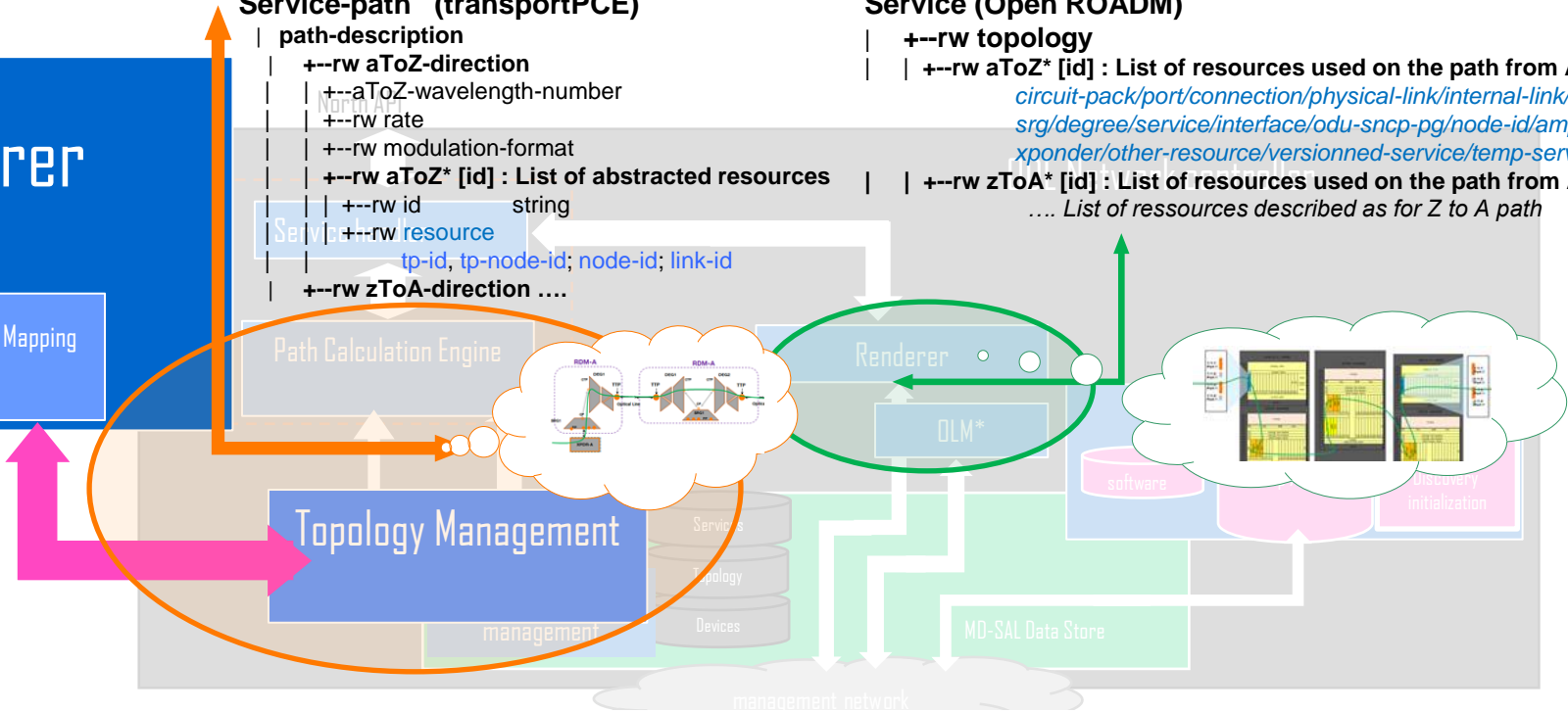
```

| path-description
|   |--rw aToZ-direction
|   |   |--aToZ-wavelength-number
|   |   |--rw rate
|   |   |--rw modulation-format
|   |--rw aToZ* [id] : List of abstracted resources
|   |   |--rw id string
|   |   |--rw resource
|   |   |   |--tp-id, tp-node-id; node-id; link-id
|   |--rw zToA-direction ....
  
```

Service (Open ROADM)

```

| |--rw topology
|   |--rw aToZ* [id] : List of resources used on the path from A to Z
|   |   |--circuit-pack/port/connection/physical-link/internal-link/shelf/
|   |   |--srg/degree/service/interface/odu-sncp-pg/node-id/amplifier/
|   |   |--xponder/other-resource/versioned-service/temp-service
|   |--rw zToA* [id] : List of resources used on the path from A to Z
|   |   .... List of resources described as for Z to A path
  
```



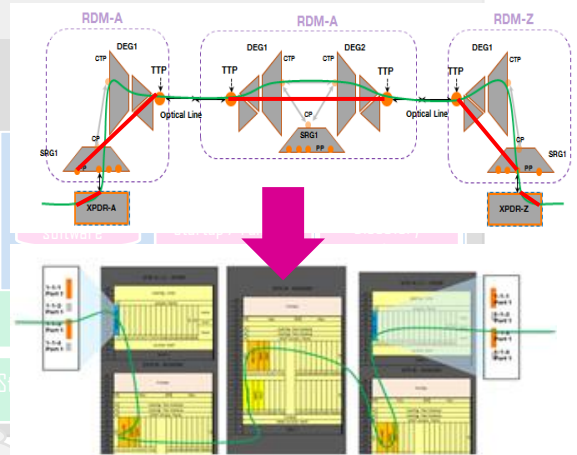
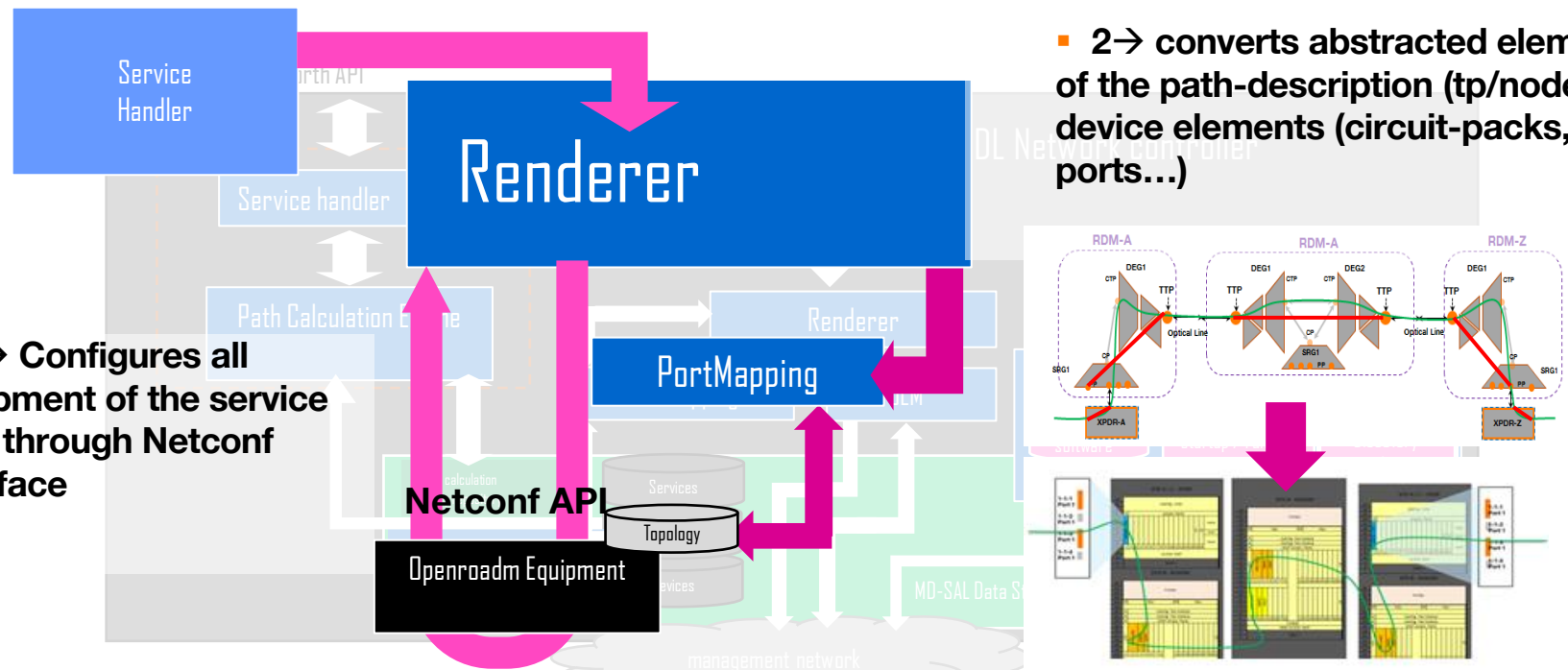
- **Port mapping** module of the Service Handler builds the mapping between **physical elements** as described in the **Open ROADM service-model** and **abstracted elements** used in the topology and in the **path description** of **transport PCE service-path** model

Renderer

- 1 → Receives Service implementation requests from service handler

- 2 → Configures all equipment of the service path through Netconf Interface

- 2 → converts abstracted elements of the path-description (tp/node) into device elements (circuit-packs, ports...)



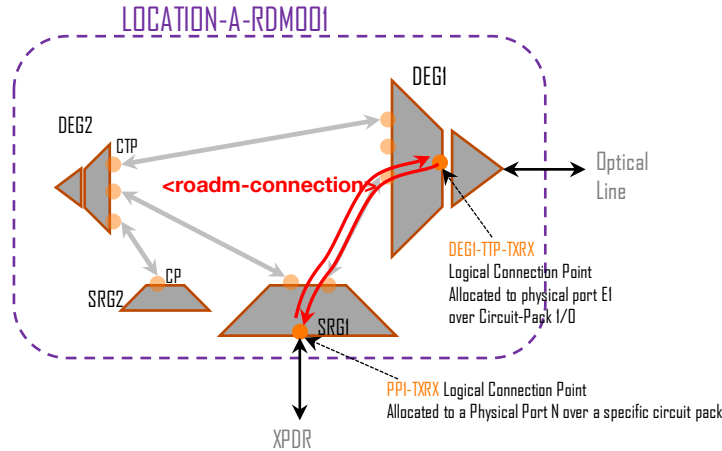
The Renderer configures the service path through the different network elements

- Handles interfaces & connection creation and deletion

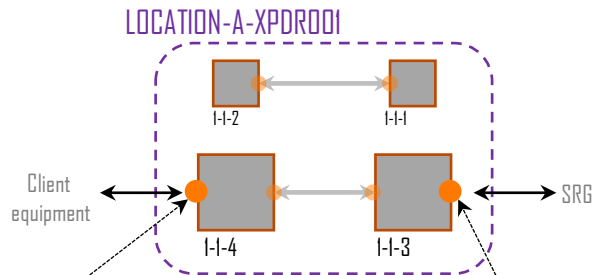


Rendering tasks performed during service creation :

Example of a 100Gbps wavelength (OCH channel) creation



- On **DEGI-TTP-TXRX**
 - Create OTS interface if does not exist
 - Create OMS interface if does not exist
 - Create Och interface
- On **PPI-TXRX**
 - Create Och interface
- On **LOCATION-A-RDMOOI**
 - Create <roadm-connection> between Och interfaces



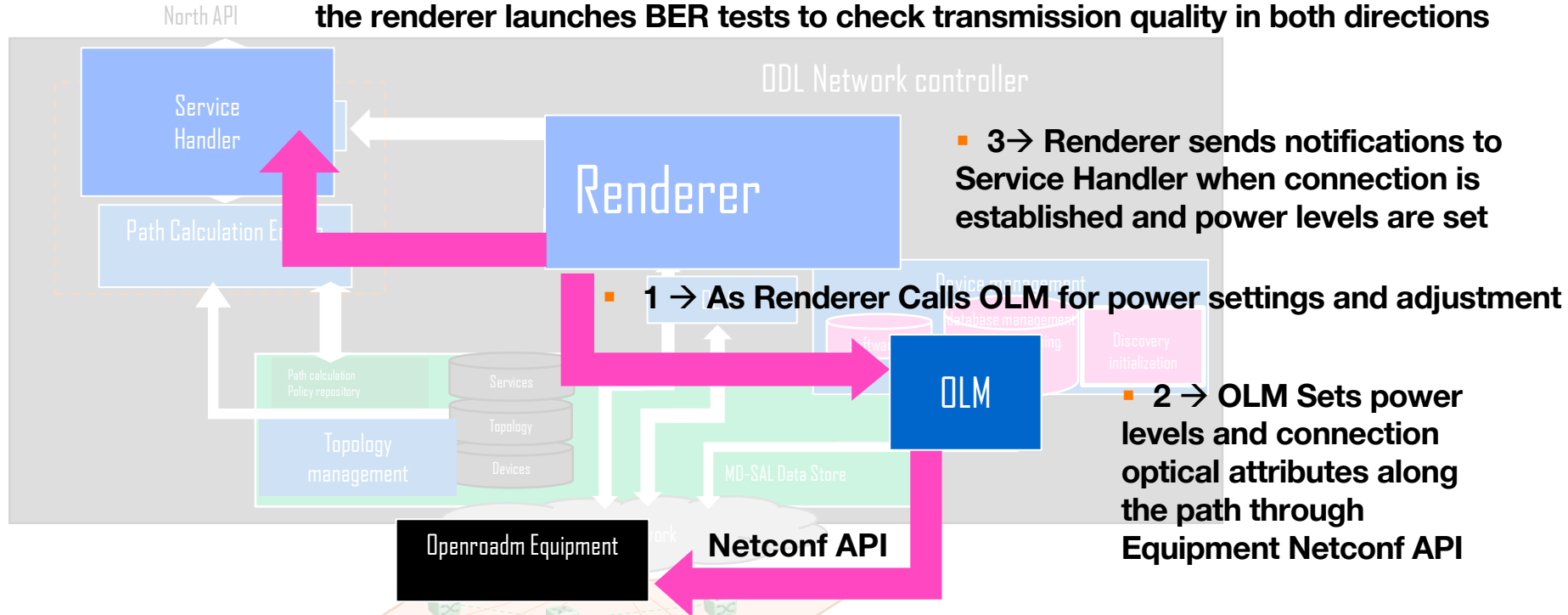
- On **XPDR-NETWORKI**
 - Create Och interface
 - Create OTU interface
 - Create ODU interface
- On **XPDR-CLIENTI**
 - Create 100G Ethernet interface

XPDR-CLIENTI Logical Connection Point allocated to a physical port N over a specific Circuit-Pack I-1-4 (client pluggable module)

XPDR-NETWORKI Logical Connection Point allocated to physical port a physical port M over another specific Circuit-Pack I-1-3 (network pluggable module)

OLM : Optical Line management module

- 4 → After interfaces, connections & power levels has been configured on the NEs, the renderer launches BER tests to check transmission quality in both directions

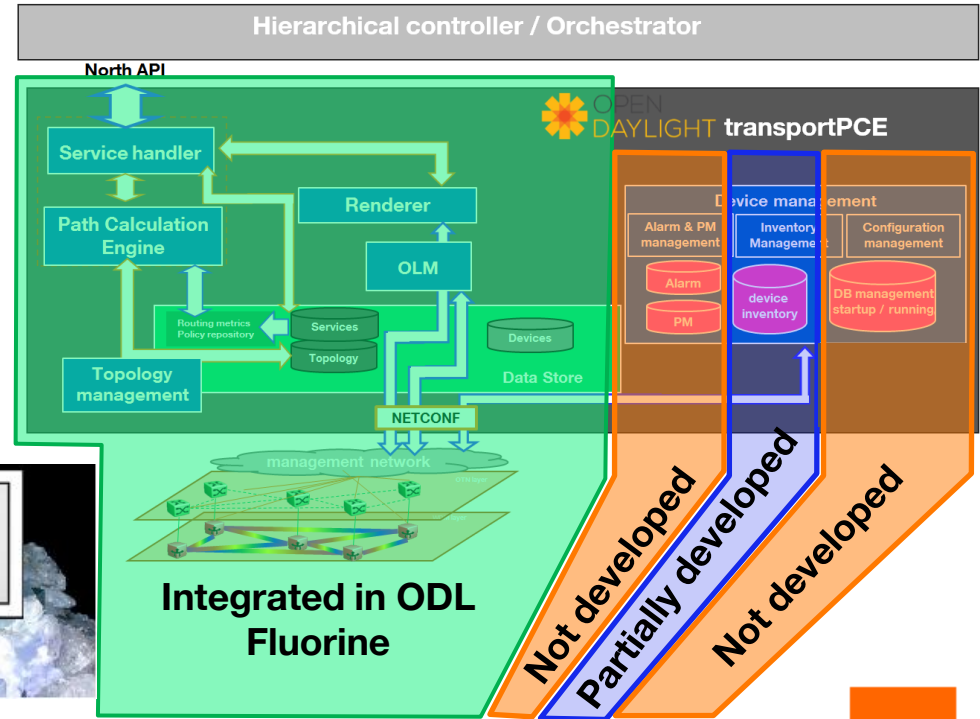


OLM is responsible for setting and controlling optical power levels

- After interfaces & connection are created, OLM manages power settings of the different optical elements

Transport PCE in OpenDaylight Fluorine

- Provides most of the bricks defined in the controller architecture for the WDM layer
- Junit and functional tests developed for the available modules
 - Continuous integration eases collaboration between contributors in different countries, entities & companies



Added in Neon ODL Release

The main features added in Neon release are the following

- Add **support for notification in RPC handling**
- Extension of the coverage for OpenROADM Service RPC handling: **service-reroute, service-restoration, temp-service-create/delete**
- **Impairment aware path calculation** in PCE (OSNR calculation)
- Management of unidirectional ports in path calculation and path configuration
- ROADM to ROADM service creation, for resource reservation when transponders are not present
- Introduction of transportpce-service-path 1.6
- **Device version management** (up to release 2.2.1)



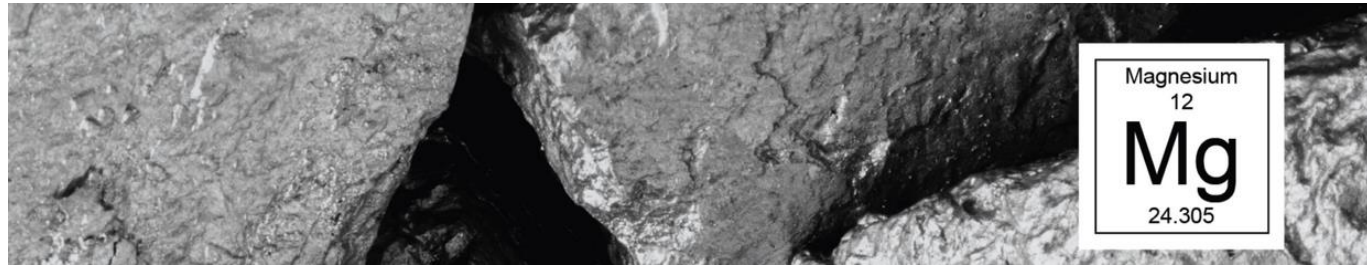
Sodium Release focused on code refactoring

Sodium tPCE release focuses on code refactoring

- Main goal is to get a robust base (fully tested) aligned with latest developments and bug corrections coming from the different contributors
- Hardened support of OpenROADM 1.2.1 and 2.2.1 releases
- Full support of ietf network topology (RFC 8345 / openROADM Network model 4.1) with consolidated topology building and portmapping functions
- **CI/CD environment allowing smooth integration of contributions avoiding regression (+/-1 voting)**



Magnesium



A wide set of new features is introduced as experimental support in Mg SR0 . A hardened support will be provided across next Mg releases

Introduction of OTN

- **OTN topology** : management of OpenROADM OTN devices including switching pool
- **Creation/deletion of OTN services using East/west APIs**
 - path-computation request, otn-service-path
 - 1GE/ODU0, 10GE/ODU2e, ODU4, OTU4 services
- **OTN rendering function** (creation of OTN interfaces and cross connections on devices)

T-API

- **Implementation of get-T-API-topology** allows retrieving an abstracted topology derived from the openroadm-topology and otn-topology layer (Nodes and access-points in SR0)

Device inventory

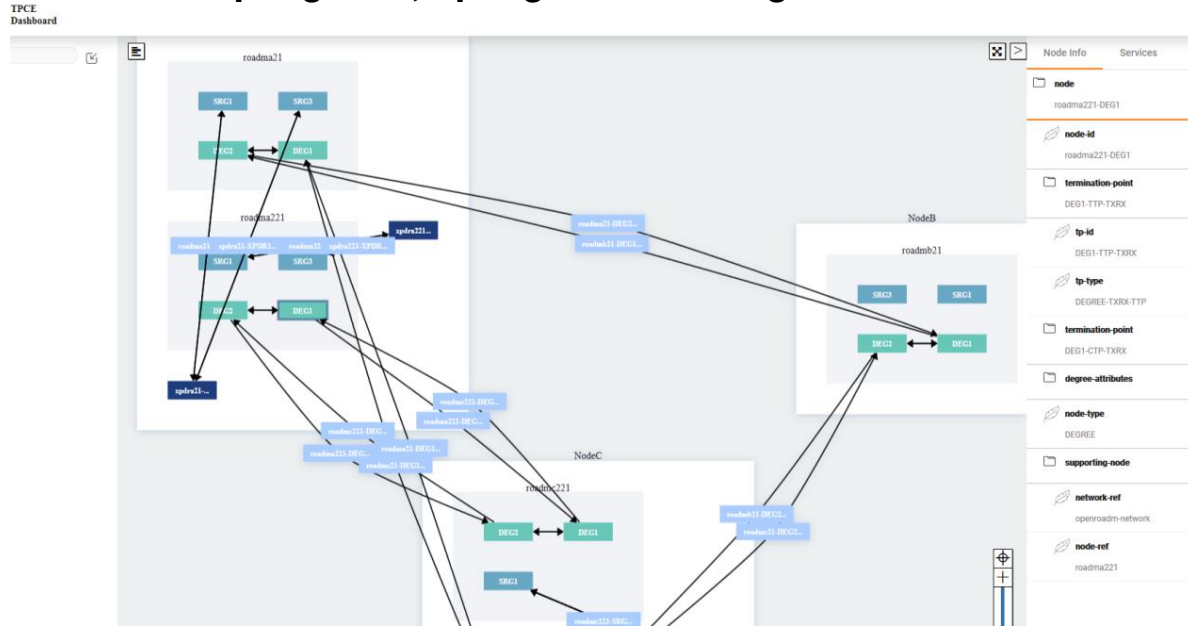
- **Experimental support of device inventory** (limited to OpenROADM device 1.2.1 in Mg SR0)

Interconnection to GNPY

- **An interconnection to GNPY is fully supported, including** :
 - Topology export to GNPY tool
 - Path validation / impairment aware optical path calculation performed in GNPY according to specific constraints

Transport PCE GUI

- Provides a collapsed topological view of the OpenROADM network controlled by Transport PCE
 - Backend interaction with TransportPCE Data store (ODL MD-SAL)
 - Collapsed view based on CLLI, Network and topology layers
- As well as a view of provisioned services
- Based on Spring boot, Spring Data and Angular



Available at :

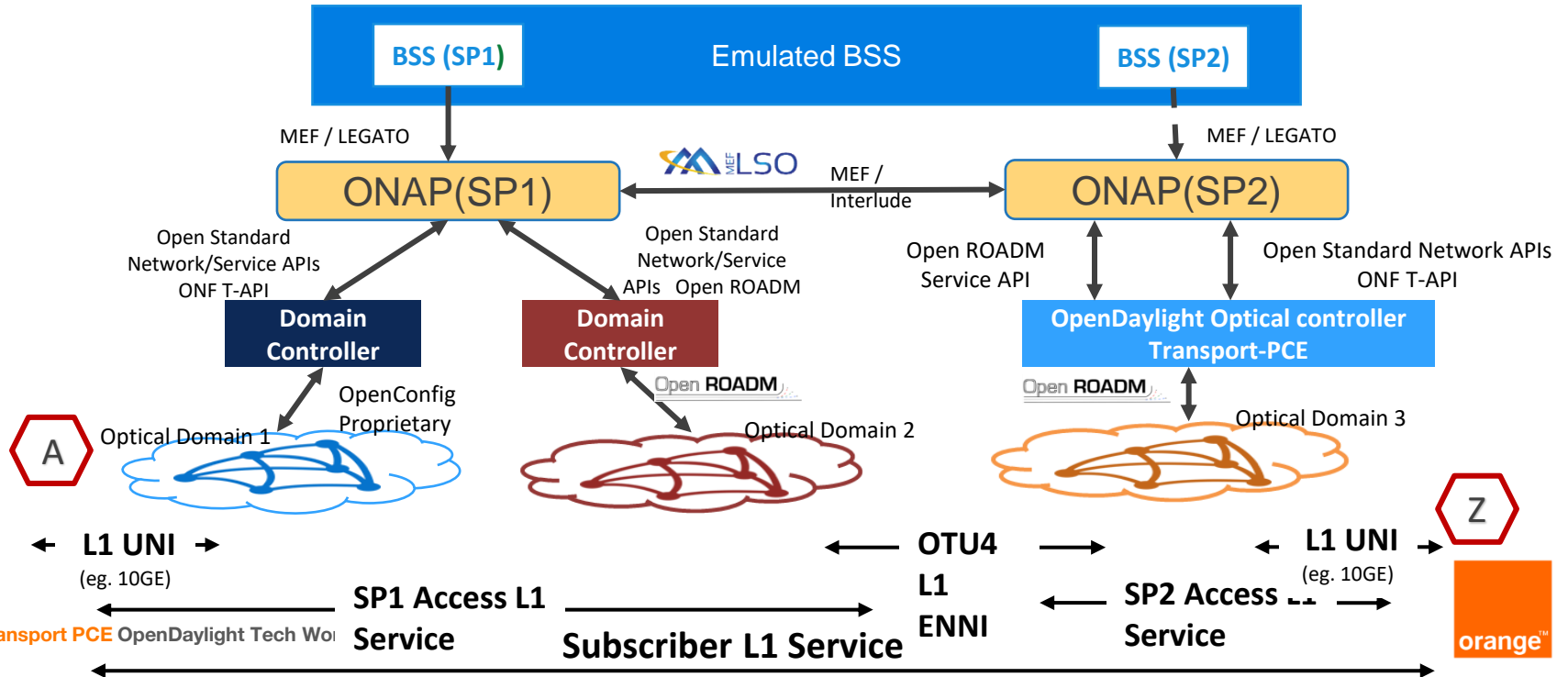
https://gitlab.com/Orange-OpenSource/lfn/odl/tpce_gui

Next steps

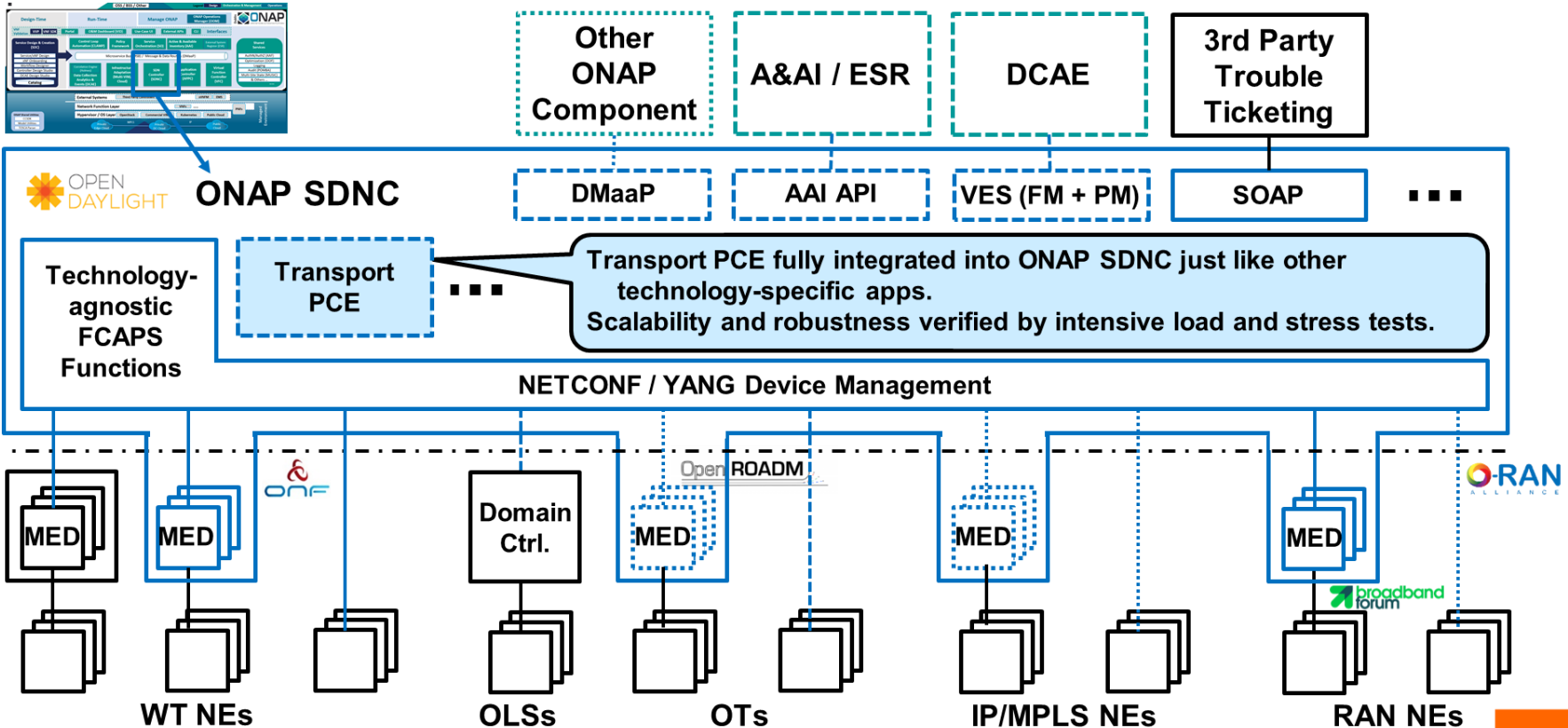
- **Improve OTN support to provide end to end service creation/deletion**
 - The support for OTN will be improved across the different Magnesium Releases
 - The target is to handle E2E service creation/deletion for 1GE, 10GE, 100GE as well as OTU2 and OTU4 services.
- **FLEXGRID and higher rates**
 - Current implementation is based on a 96 wavelengths fixed grid
 - Evolution towards a flexgrid in order to handle services with variable spectrum occupancy (PCE/topology management)
 - Handling Beyond 100Gbps services (200/300/400G)
 - Developments in all modules, with a main impact on Renderer.
- **PCE enhancement**
 - The target is to enrich PCE implementation :
 - Complement constraints handling to implement all what's in OpenROADM service model :
 - handling of soft constraints , handling of co-routing/include constraints
 - Complement the set of metrics handled by the PCE (currently based on Hop count & propagation delay)
 - distance, load, adaptation number, TE-metric, Composite metric

ONAP MDONS Use Case

- Multi Domain Optical Network Services (MDONS) is an ONAP use case which focuses on the ability to automate the creation of L1 (OTN based) services that span across multiple carriers



Transport PCE integrated in ONAP SDNC



Early version of Mg SR0 tPCE code demonstrated at OFC 2020

Optical Fiber Communications Conference (OFC) 2020 in San Diego, CA

- The “PROnet” SDN orchestrator automatically initiates actions to execute datacenter backups and massive Virtual Machine migrations.
- It relies on TransportPCE to control a low latency OpenROADM optical layer build from ROADMs and OTN flexponders coming from six suppliers :
 - FUJITSU, CIENA, CISCO, ECI, JUNIPER, INFINERA
- <https://www.utdallas.edu/oit/about/cyberinfrastructure-research-services/press-release/interoperability-of-innovative-capabilities-demonstrated-at-ofc-20/>



TransportPCE challenges to become a managed project

- **too few active committers compared to releases pace**
 - Most contributors were optical network experts before being developers.
Until recently, almost only the PTL was taking care of reviewing, merging contributions and integrating releases.
 - New blood and decision taken with others teams members to share this load and to develop this skill.
- **upstream dependencies**
 - TPCE strongly depends on the Netconf project, that itself depends on most kernel projects.
 - Bug fixes and stable usable dependencies arrive late in the release cycle, usually around 2 weeks before code freeze.
We usually start working on the new release after code freeze to reduce debug overheads.
- **Snapshot versions**
 - they are unstable, what complicates debugging.
Also since old snapshots versions are removed from nexus, older commits do not work out of the box.
 - Current release process implies working on snapshot daily built version but release can only be staged when non-snapshots versions are published on Nexus => short slot for integrating release

Thank You !

Questions ?

