Open ROADM Service Model MSA V4

April 24, 2019

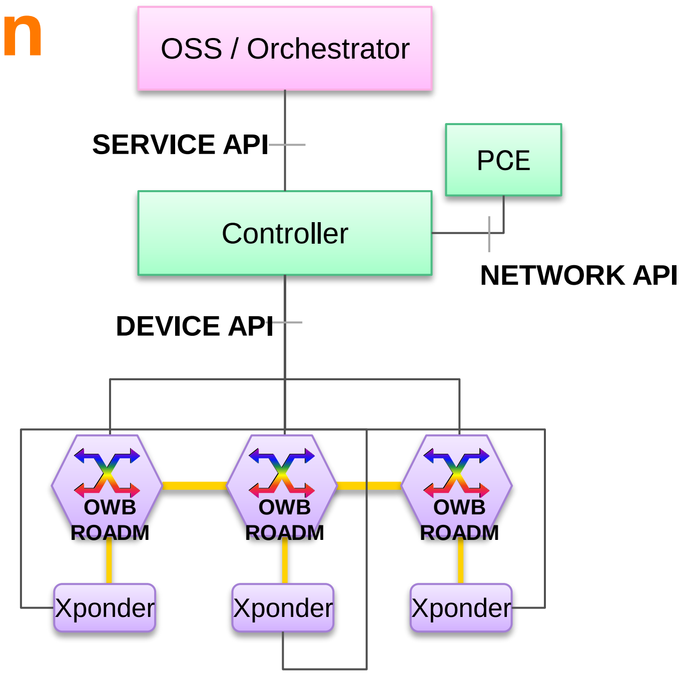
Olivier Renais
Orange Labs
olivier.renais@orange.com

orange™

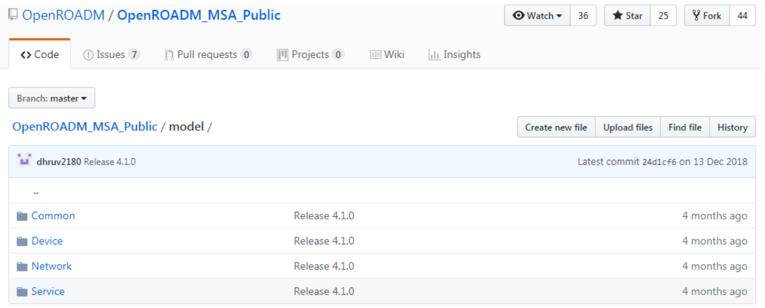
Open ROADM

Open ROADM model Organization

- **Device Model**
  - Interface to Open ROADM devices
  - Also basis for the inventory database
- **Network Model**
  - Abstract network model for routing
- **Service Model**
  - Optical service creation, modification, and deletion
- **Common Models**
  - Models that apply to both the Open ROADM device and controller
  - Alarms, PM, TCA
  - Common type definitions



https://github.com/OpenROADM/OpenROADM_MSA_Public



OpenROADM / OpenROADM_MSA_Public   Watch 36 Star 25 Fork 44

Code Issues 7 Pull requests 0 Projects 0 Wiki Insights

Branch: master

OpenROADM_MSA_Public / model /   Create new file Upload files Find file History

dhruv2180 Release 4.1.0   Latest commit 24d1cf6 on 13 Dec 2018

..

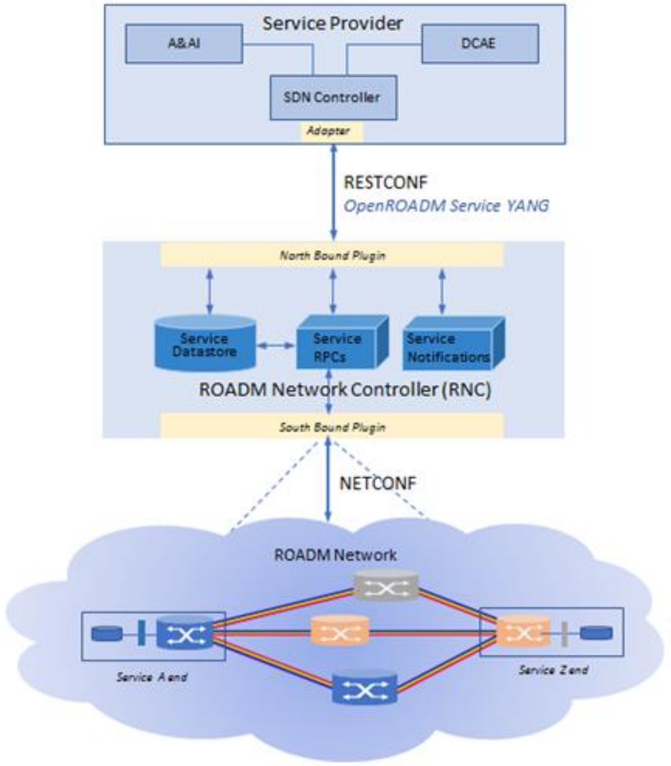
Common	Release 4.1.0	4 months ago
Device	Release 4.1.0	4 months ago
Network	Release 4.1.0	4 months ago
Service	Release 4.1.0	4 months ago

On courtesy of Fred Gruman

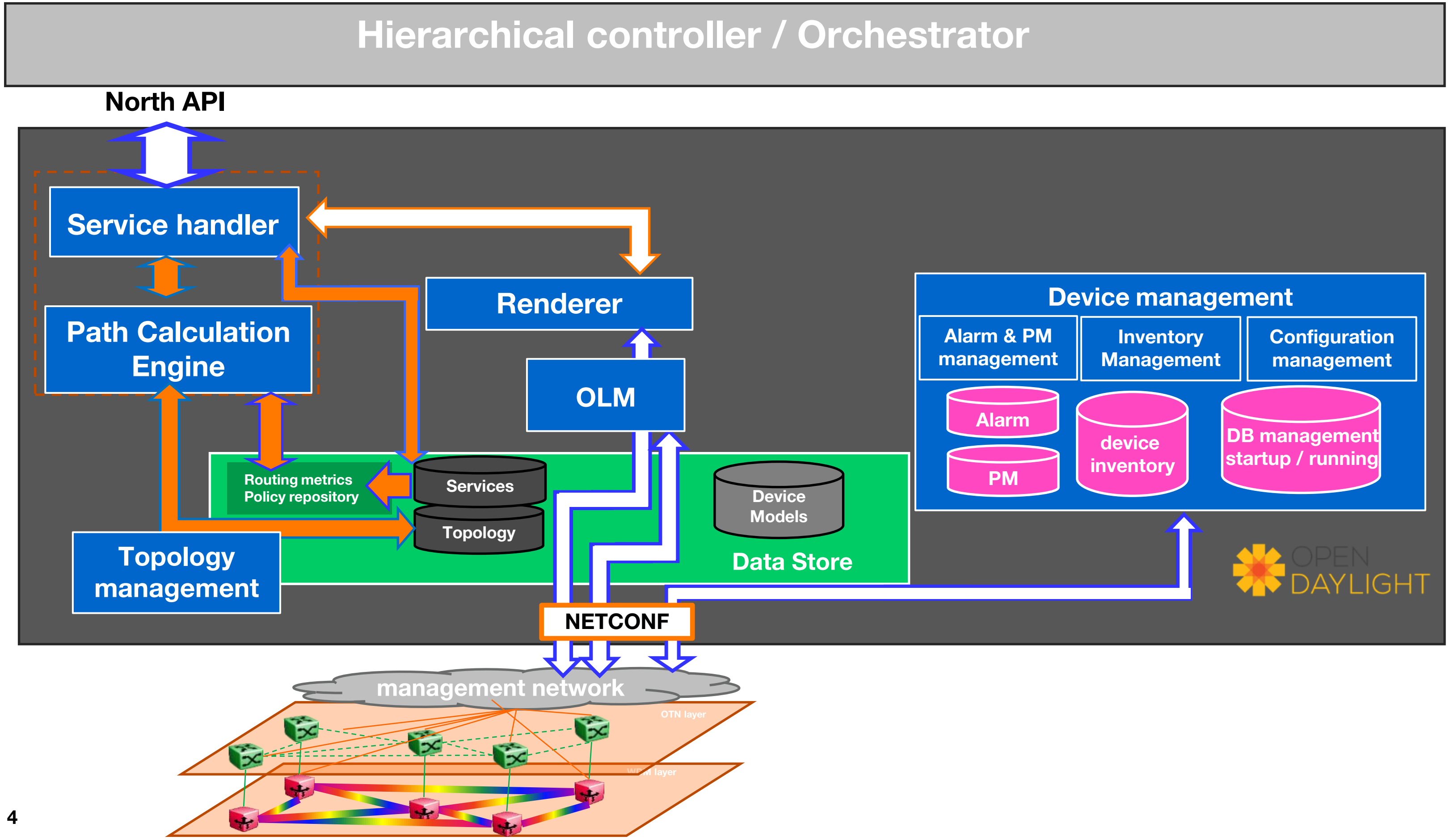
orange™

Service Model general description

- **Open ROADM Service model defines the rpc for the Northbound API of a ROADM Network Controller (RNC), and the yang models associated with the Service Data Store**
- **North API**
  - The Open ROADM Service Model defines northbound API from ROADM Network Controller (RNC) to SDN Controller :
  - Service interface is based on RESTCONF (RFC8040)
  - Remote Procedure Calls (RPCs), call back and notification defined in Open ROADM Service model
    - service create/delete, temp-service create/delete
    - service-roll, service-reconfigure,
    - service-reroute …
- **Service Data Store**
  - service-lists, temp-service-list, versioned-service-list

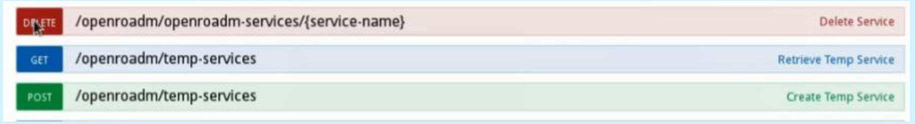


A generic ROADM Network Controller (RNC) architecture



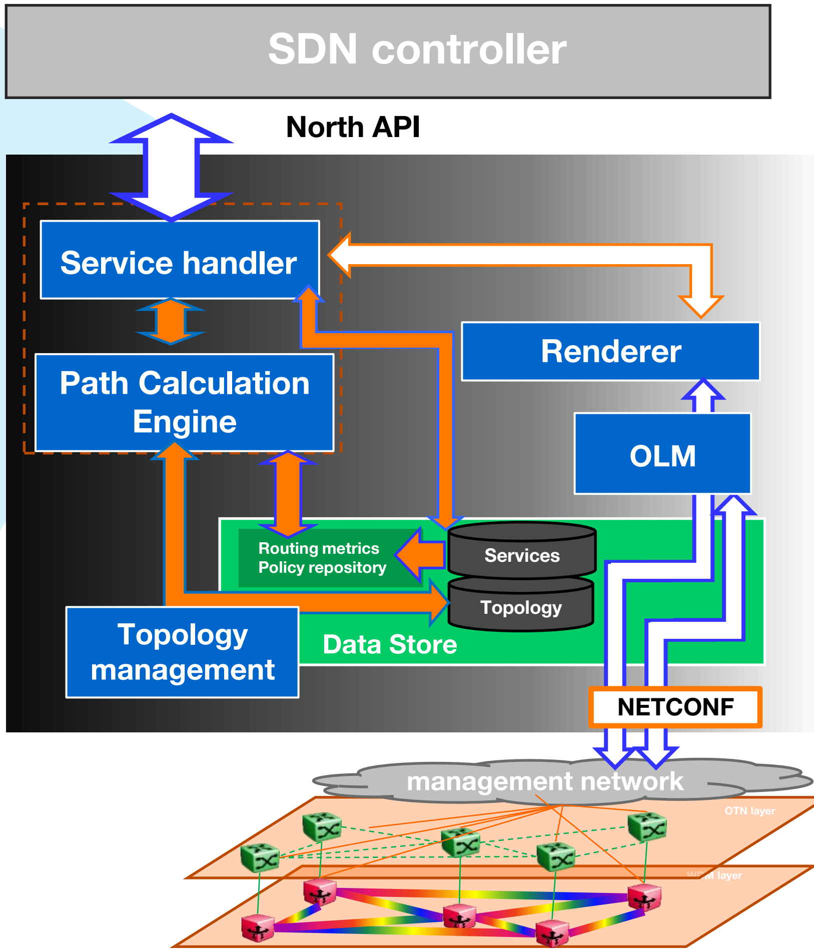
North API

- **OR Service Model defines north API from RNC to SDN Controller :**
  - **Service interface is based on RESTCONF (RFC8040) as stated in the OR Service White paper**
    - RESTCONF protocol allows interrogating elements on their state using procedures based on http primitives
    - It uses POST, GET, PUT, DELETE… HTTP methods to provide CRUD (create, read, update, delete) operations on a conceptual Data Store containing YANG-defined data



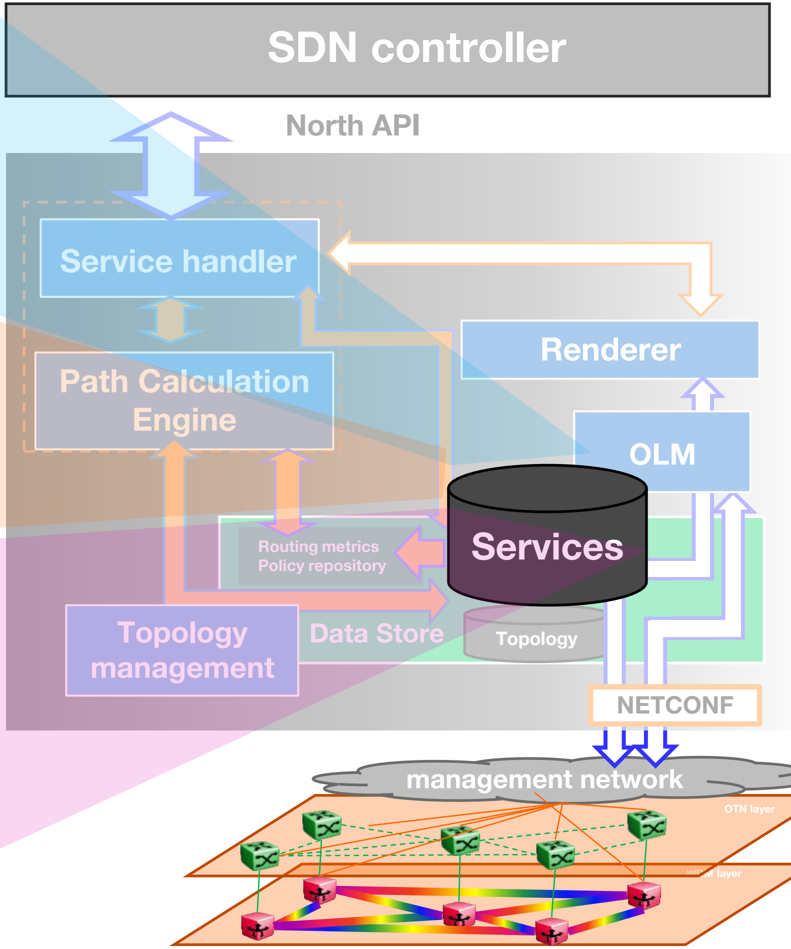
DELETE	/openroadm/openroadm-services/{service-name}	Delete Service
GET	/openroadm/temp-services	Retrieve Temp Service
POST	/openroadm/temp-services	Create Temp Service

  - **Open ROADM service model defines Remote Procedure Calls (RPCs).**
    - service create/delete, temp-service create/delete
    - service-create/delete-result-notification-request
    - service-roll, service-reconfigure, service-reroute (-confirm),
    - network-re-optimization
    - service-feasibility-check (-bulk)
    - service-restoration, service-reversion
    - ber-test, ber-test-async-callback



Service Data Store

- **Service List**
  - List of services/service names requested or created in the ROADM network
    - Service-name is unique
  - Associated configuration and operational parameters
    - does not contain historical (deleted or those past their end time) or temp/draft services
- **Temporary service list**
  - Temp service list represents services to be provisioned in the future
  - Temp service list is a list of temporary services.
  - Temp services can only be created, deleted, modified, etc. using special RPCs.
- **Versioned service list**
  - Adds version number(s) to the service list, while keeping the same service names as in service list
  - Contains versioned services, regardless of their lifecycle state.
  - May contain deleted services, multiple versions of the same service, as identified by its name
  - Versioned services can be created, deleted, modified, etc. using standard RPCs, stating the version-number

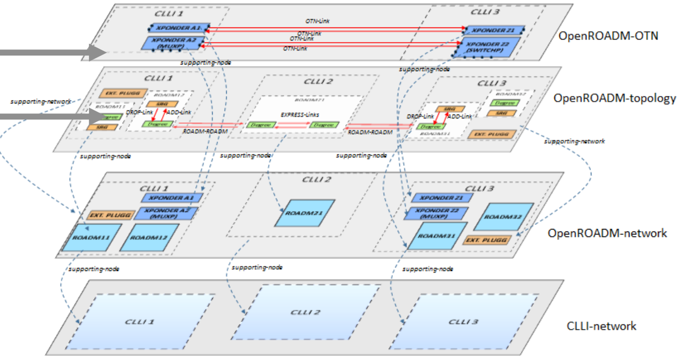


Service Model structure

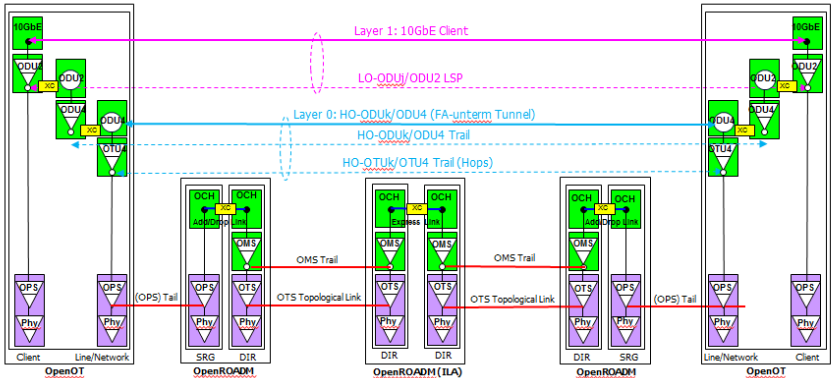
- **org-openroadm-ber-test.yang, org-openroadm-common-ber-test.yang**
  - Model for RPCs, Call back and notifications associated with Ber Test
- **org-openroadm-common-service-types.yang**
  - Common base for the service model : defines all grouping used in RPCs and service lists
- **org-openroadm-routing-constraints.yang**
  - Defines the structure (groupings) of the routing constraints (hard and soft constraints)
- **org-openroadm-service.yang**
  - Defines the model for the main RPCs and notification
  - Defines the models for the different service list
- **org-openroadm-topology.yang**
  - Defines Topology (detailed path information)
  - Defines Network-topology (abstracted path information)
    - Relies on resources/network-resources defined in the Common model

Service Levels

- **According to topology**
  - defined by connection-type and service-layer-type
    - service-layer-type :
      - otn
      - wdm
    - connection-type :
      - roadm-line : OMS service created in the initial phase at line commissioning
      - infrastructure : defined between ROADM PPs
        - first step of an OCh service creation, when WDM line (ROADMs) is in place, but transponders are not provisioned
      - service : End to end service at OTN or WDM layer
        - Second step of an OCh service creation, after Transponder have been put in place.
        - Network-Tails must be created between transponders and ROADMS, as well as client tails between transponders and client equipment



- **Supporting services**
  - The service that the considered service is running over the top
  - As an example : a client 10 GE service on a muxponder is running over an OTU4 service, itself relying on roadm-line service



On courtesy of Nokia

orange™

Service lifecycle

- **The status of a service can change, according to the rpc received by the RNC from SDN controller**
- **Lifecycle : deployed(1),planned(2), maintenance(3), deploying(4), undeploying(5), undeployed(6), proposed(7), draft(8)**
  - Planned : a service has been planned through service create rpc (with a due date) and the resources associated to it are reserved (the service is in the service-list).
  - Draft : the service feasibility has been checked, and the service is feasible, but some resources are missing. Available and needed resources are allocated to it. Draft services are only in temp-service list
    - *A typical use case is when end resources (transponders) have not been provisioned. Resources are however reserved to make sure the service can be deployed when transponders are available at both ends*
  - Proposed : A service is feasible, with no missing resources. No resource is allocated to it. The service path is populated. A service create rpc will lead to check if resources are still available and update accordingly the status of the service in the list
    - *A typical use case is when several path constraints might be experimented (service-feasibility-check). Several implementation might be proposed, but only one will be implemented. Several temporary services can be populated as proposed in temp-service-list by the controller. At a later step, after a decision was made, a service creation can be launched using best option, and corresponding routing constraints.*
  - Deploying : the service is in the creation process, but all the tests (BER) have not been performed
  - Deployed : the service has been successfully created and all associated tests successfully passed. A notification service-traffic-flow has been received by the controller
  - Undeploying : the controller received an order to delete the service, the operation is ongoing
  - Undeployed : the service has been successfully deleted, all resources are released
  - Maintenance : the service is not active. Some equipment on the path is supporting (or may support) maintenance activities

Parameters exchanged during service creation and modification process (1/3)

- **General information about the service**
  - due and end date
  - customer information
  - service layer
  - reference network and topology
  - bandwidth calendaring option (see next slide)
- **SLA (Resiliency)**
  - resiliency type :
    - unprotected, unprotected-diversely-routed
    - protected, restorable, higher-level-restorable
  - switching/ reversion parameters

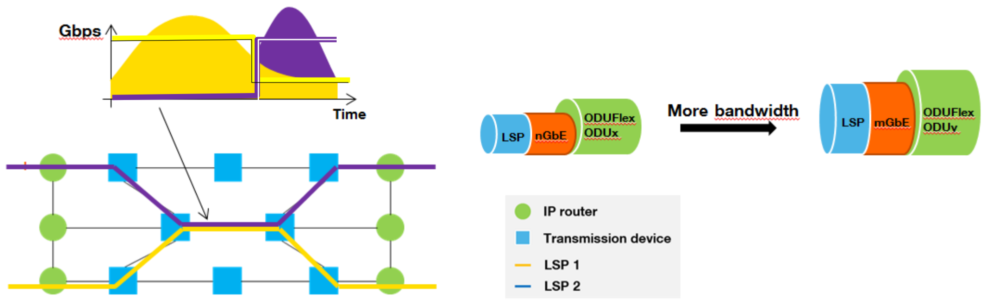
```
|    +--rw due-date?                 yang:date-and-time
|    +--rw end-date?                 yang:date-and-time
|    +--rw eventHorizonStart?        yang:date-and-time
|    +--rw eventHorizonEnd?          yang:date-and-time
|    +--rw nc-code?                  string
|    +--rw nci-code?                 string
|    +--rw secondary-nci-code?       string
|    +--rw customer?                 string
|    +--rw customer-contact?         string
|    +--rw operator-contact?         string
|    +--rw service-layer?            service-layer-type
|    +--rw clli-network-ref?         string
|    +--rw openroadm-network-ref?    string
|    +--rw openroadm-topology-ref?   string
|    +--rw bandwidth-calendaring?    boolean
|    +--rw bw-calendaring-parameters
|              +-- …..
```

```
|    +--rw service-resiliency
|    |  +--rw resiliency?                            identityref
|    |  +--rw revertive?                             boolean
|    |  +--rw wait-to-restore?                       uint64
|    |  +--rw holdoff-time?                          uint64
|    |  +--rw pre-calculated-backup-path-number?     uint8
|    |  +--rw coupled-service
|    |     +--rw coupled-services* [service-index]
|    |        +--rw service-index                    uint16
|    |        +--rw service-name?                    string
|    |        +--rw common-id?                       string
|    |        +--rw version-number?                  uint64
```

orange™

Bandwidth calendaring option

- **Regular services are activated from a due date to an end date**
- **Bandwidth calendaring option allows implementing use cases where services are activated across time according to specific profiles**
  - The service is activated according to a recurrence pattern
  - The service is complementary to a set of coupled services with different activation profiles
    - Each service correspond to one rate
    - Each recurrence pattern defines several time slots on a daily basis during which the service is active
    - Coupled service at another rate can be defined on complementary time slots (same customer)
    - Concurrent coupled service sharing partially or completely the same bandwidth on complementary slots (routing-constraint use for co-routing) can also be defined (same/other customer)



```
|     +--rw bandwidth-calendaring?      boolean
  |     +--rw bw-calendaring-parameters
  |     |  +--rw bw-calendaring-coupled-services* [service-index]
  |     |  |  +--rw service-index      uint16
  |     |  |  +--rw service-name?      string
  |     |  |  +--rw common-id?         string
  |     |  |  +--rw version-number?   uint64
  |     |  +--rw recurrence-pattern* [recurrence-id]
  |     |     +--rw recurrence-id      uint32
  |     |     +--rw day-of-the-week*   enumeration
  |     |     +--rw start-time?        string
  |     |     +--rw end-time?          string
```

orange™

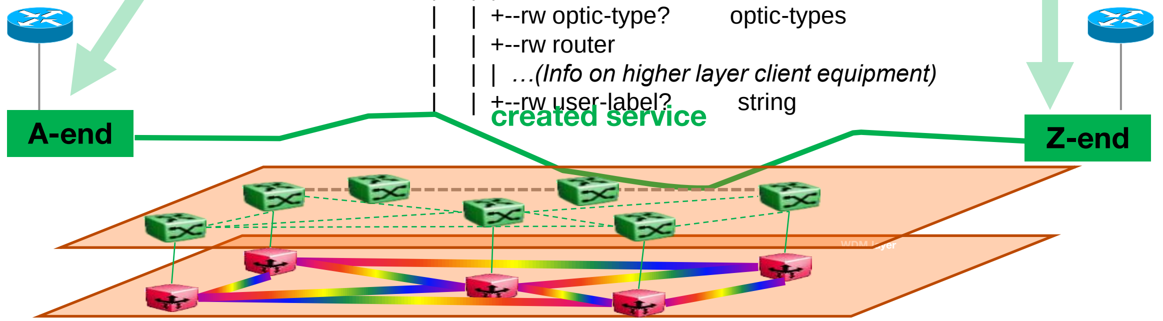
Service Resiliency

Resiliency type	triggering	Reversion	Description
Unprotected	NA	No	The service is not protected
Unprotected-Diversely-Routed	NA	No	The service is associated to another service for disjoint routing via coupled-service attributes. Used when the protection is handled in an upper layer. A typical use case is when IP adjacencies (odd/even) are routed on diverse optical paths.
Protected	Device level	Yes	A working and a backup path are defined. The protection is triggered at the device level. Reversion can be defined.
Restorable	ROADM network Controller	Yes	One working and potentially one or several backup paths • No backup-path → On the fly restoration • Backup-path → path pre-calculation. Reversion can be defined. It is also handled at the controller level. Either at photonic, or OTN layer.
Higher-level-restorable	SDN Controller	Yes	Same as restorable, except that restoration is triggered at the SDN controller level through service-restoration rpc. Reversion can also be defined. it is triggered by the SDN controller through service-reversion-rpc

Parameters exchanged during service creation and modification process (2/3)

- **Technical requirements**
  - rate /format (attributes of A and Z end)
  - routing constraints
    - hard (absolute) /soft (best-effort)
  - routing metrics

```
|      +--rw service-a-end / rw service-z-end
|      |  +--rw service-format          service-format
|      |  +--rw service-rate?           uint32
|      |  +--rw otu-service-rate?       identityref
|      |  +--rw odu-service-rate?       identityref
|      |  +--rw ethernet-encoding?   ethernet-encoding-type
|      |  +--rw mapping-mode?        mapping-mode-type
|      |  +--rw clli                 string
|      |  +--rw node-id?          node-id-type
|      |  +--rw tx-direction / rw tx-direction
|      |  |  +--rw port
|      |  |  …port name & type; device-name; circuit-pack-name
                rack; shelf, slot; subslot….
|      |  |  +--rw lgx
|      |  |  …
|      |  |  +--rw tail
|      |  |  …
|      |  +--rw optic-type?        optic-types
|      |  +--rw router
|      |  |  …(Info on higher layer client equipment)
|      |  +--rw user-label?        string
```

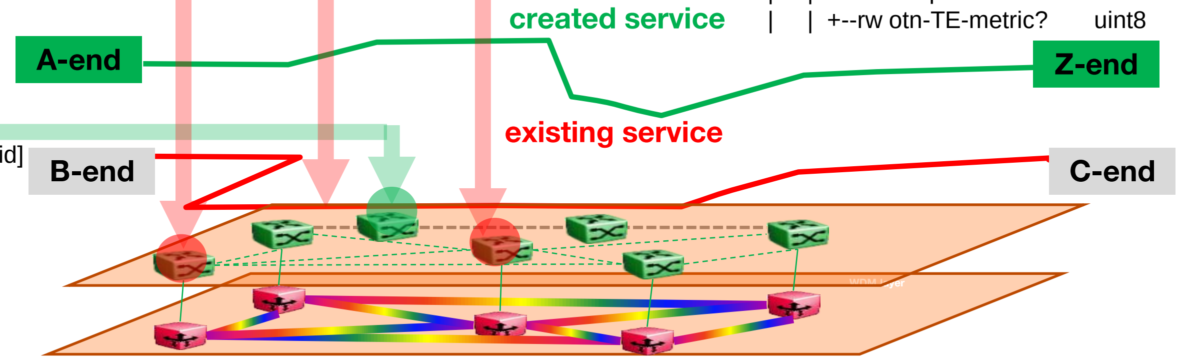


Parameters exchanged during service creation and modification process (3/3)

```
|     +--rw hard-constraints  / soft-constraints
   |     |  +--rw customer-code*   string
   |     |  +--rw (co-routing-or-general)?
   |     |     +--:(general)
   |     |     |  +--rw diversity
   |     |     |  |  +--rw existing-service*            string
   |     |     |  |  +--rw existing-service-applicability
   |     |     |  |     +--rw site?   boolean
   |     |     |  |     +--rw node?   boolean
   |     |     |  |     +--rw srlg?   boolean
   |     |     |  |     +--rw link?   boolean
   |     |     |  +--rw exclude
   |     |     |  |  +--rw fiber-bundle*              string
   |     |     |  |  +--rw site*                      string
   |     |     |  |  +--rw node-id*                   node-id-type
   |     |     |  |  +--rw link-identifier* [link-network-id link-id]
   |     |     |  |  |  +--rw link-network-id    string
   |     |     |  |  |  +--rw link-id            string
   |     |     |  |  +--rw supporting-service-name*   string
   |     |     |  +--rw include
   |     |     |  |  +--rw fiber-bundle*              string
   |     |     |  |  +--rw site*                      string
   |     |     |  |  +--rw node-id*                   node-id-type
   |     |     |  |  +--rw link-identifier* [link-network-id link-id]
   |     |     |  |  |  +--rw link-network-id    string
   |     |     |  |  |  +--rw link-id            string
   |     |     |  |  +--rw supporting-service-name*   string
   |     |     |  +--rw latency
   |     |     |     +--rw max-latency?   uint32
   |     |     +--:(co-routing)
   |     |        +--rw co-routing
   |     |           +--rw existing-service*   string
```

- **Routing constraint (diversity, include, exclude)**
  - **Configuration of routing algorithm : routing metrics**
    - metric priority : 0=not used, 1 Highest, 255 Lowest
      - composite metric :
        - Example : wdm-hop-count →1, wdm-load → 2

```
|     +--rw routing-metric
|     |  +--rw wdm-hop-count?       uint8
|     |  +--rw otn-hop-count?       uint8
|     |  +--rw wdm-load?            uint8
|     |  +--rw otn-load?            uint8
|     |  +--rw latency?             uint8
|     |  +--rw distance?            uint8
|     |  +--rw wdm-TE-metric?       uint8
|     |  +--rw adaptation-number?   uint8
|     |  +--rw otn-TE-metric?       uint8
```



Parameters returned by controller during service creation and modification process (1/2)

- **In the service lists, and through service-notification**
- **Path general information**
  - total latency
  - current active-path
  - supporting-service-name
  - followed Shared Risk Groups (SRGs)

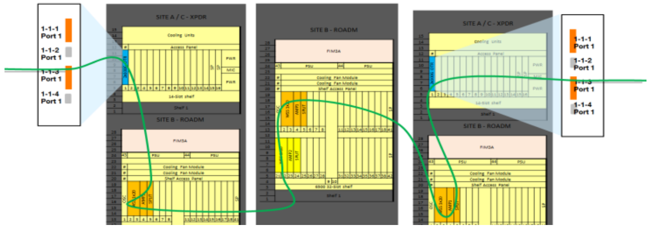
```
|     +--rw latency?                  uint32
|     +--rw fiber-span-srlgs*          string
|     +--rw equipment-srgs* [srg-number]
|     |  +--rw srg-number    uint16
|     +--rw supporting-service-name*     string
|     +--rw current-active-path-id?      uint8
```

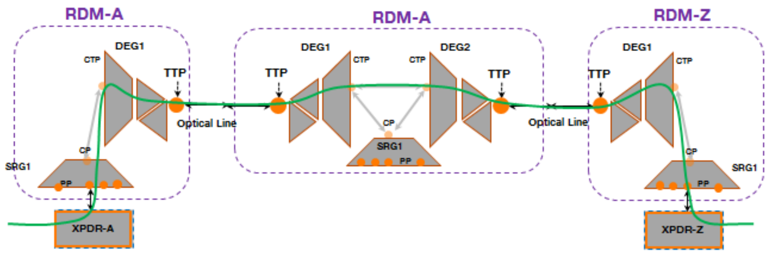
- **Detailed path description**
  - **Topology**
    - Provides a detailed path description based on resources such as circuit-packs and port (device oriented)
      - Fits for rendering function, responsible for configuring interfaces and connections on devices
    - One working path, several (in case of path pre-calculation) backup-paths, and an indication on what is the current active path
  - **Network-Topology (added in R3.0)**
    - Provides an abstracted path description based on resources such as Nodes, Links and tps
      - Fits for Path computation : Path Computation Engine relies on an abstracted topology view (as modeled in ORD Network model)
      - One working path, several (in case of path pre-calculation) backup-paths, and an indication on what is the current active path

Topology versus Network-topology (1/2)

```
|     +--rw topology
|     | +--rw aToZ* [id] : List of resources used on the path from A to Z
|     | | +--rw id              string
|     | | +--rw hop-type?       enumeration
|     | | +--rw device
|     | | | +--rw node-id?   org-openroadm-common-node-types:node-id-type
|     | | +--rw resource
|     | | | +--rw (resource)?
                  circuit-pack/port/connection/physical-link/internal-link/shelf/
                  srg/degree/service/interface/odu-sncp-pg/node-id/amplifier/
                  xponder/other-resource/versionned-service/temp-service
|     | | |   +--resource unique identifier
|     | | +--rw resourceType
|     | |   +--rw type         resource-type-enum
|     | |   +--rw extension?   string
|     | +--rw zToA* [id] : List of resources used on the path from A to Z
                    …. List of ressources described as for Z to A path
|     +--rw backup-topology
|     | +--rw backup-path* [backup-path-id] : List of potential backup-path
|     |   +--rw backup-path-id     uint8
|     |   +--rw failure-case-id?   string
|     |   +--rw aToZ* [id]
|     |    ...
|     |   +--rw aToZ* [id]
```

```
|     +--rw network-topology
|     | +--rw aToZ* [id] : List of resources used on the path from A to Z
|     | | +--rw id              string
|     | | +--rw network-resource
|     | | | +--rw (network-resource)?
|     | | |   +--:(network-resource-tp)
|     | | |   | +--rw tp-network-id      string
|     | | |   | +--rw tp-node-id         string
|     | | |   | +--rw tp-id              string
|     | | |   +--:(network-resource-link)
|     | | |     +--rw link-network-id    string
|     | | |     +--rw link-id            string
|     | | +--rw network-resource-type   identityref
|     | +--rw zToA* [id] : List of resources used on the path from A to Z
                    …. List of ressources described as for Z to A path
|     +--rw backup-network-topology
|     | +--rw backup-path* [backup-path-id] : List of potential backup-path
|     |   +--rw backup-path-id     uint8
|     |   +--rw failure-case-id?   string
|     |   +--rw aToZ* [id]
|     |    ...
|     |   +--rw aToZ* [id]
```

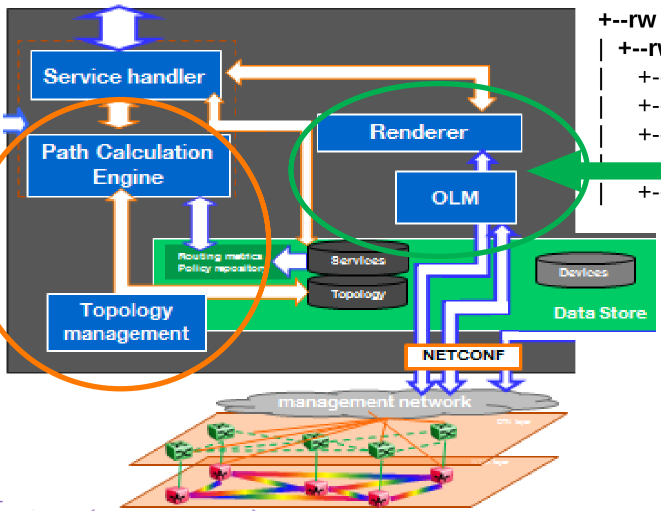


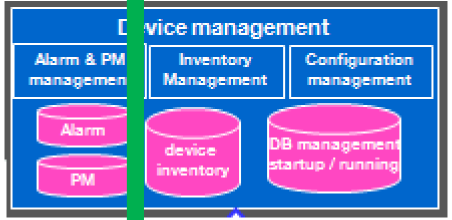


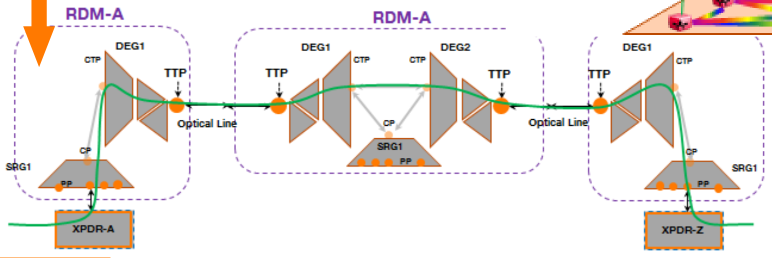
Topology versus Network-topology (2/2)

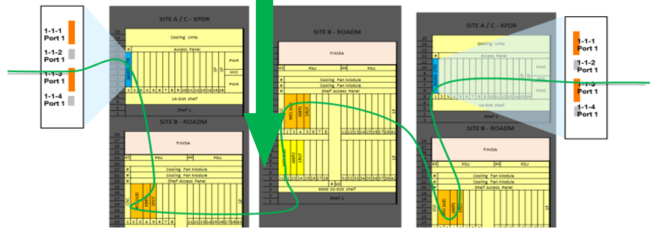
```
|    +--rw network-topology
|    | +--rw aToZ* [id] : List of resources
|    | | +--rw id            string
|    | | +--rw network-resource
             tp-network-id; tp-node-id; link-network-id; link-id
|    | +--rw zToA* [id] : List of resources
|    +--rw backup-network-topology
|    | +--rw backup-path* [backup-path-id] :
                   List of potential backup-path
|    |    +--rw backup-path-id     uint8
|    |    +--rw failure-case-id?   string
|    |    +--rw aToZ* [id]
|    |     ...
|    |    +--rw aToZ* [id]
```

```
|    +--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/versionned-service/temp-service
|    | +--rw zToA* [id] : List of resources used on the path from A to Z
              …. List of ressources described as for Z to A path
     +--rw backup-topology
     | +--rw backup-path* [backup-path-id] : List of potential backup-path
     |    +--rw backup-path-id     uint8
     |    +--rw failure-case-id?   string
     |    +--rw aToZ* [id]
     |
     |    +--rw aToZ* [id]
```









Service model rpcs (1/7)

- **service-create, temp-service-create**
  - service-create : SDN Controller requests RNC to create a new service either immediately or in the future.
  - temp-service-create : SDN Controller requests RNC to compute a service path and reserve the wavelengths assigned to the service.
  - If this request passes the initial validation, a service-rpc-result notification is sent by the RNC, once processed.
  - Main Input parameters :
    - Service general information : due and end date, customer information, service layer, reference network and topology, bandwidth calendaring option
    - Technical requirements : Aend, Zend, rate /format, port-information, lgx and tail information, routing constraints, routing metrics
    - SLA : resiliency type, switching/ reversion parameters
- **service-delete, temp-service-delete**
  - SDN Controller requests RNC to remove an existing service either immediately or in future.
  - If this request passes the initial validation, a service-rpc-result notification is sent by the RNC, once processed.
  - Once the service has been deleted, it no longer appears in the service list
  - Main Input parameters : service-name/common-id, due-date, tail-retention
- **service-create/delete-result-notification-request**
  - Callback notification sent by the RNC to the SDN controller
  - Main Input parameters : ber-test results

Service model rpcs (2/7)

- **service-feasibility-check, service-feasibility-check-bulk**
  - SDN controller calls RNC to check whether a/multiple service(s) can be provisioned in the existing network,
    - RNC checks connectivity, equipment availability, and reachability.
    - For service-feasibility-check-bulk rpc, the RNC analyzes feasibility collectively and return results, ensuring that a given resource is not used more than once
  - Options are made available to choose from one of the following for routing:
    - Using only deployed and planned equipment
    - Using existing equipment first, then proposing new equipment as needed
    - Using proposed equipment
  - Main Input parameters :
    - Technical requirements : Aend, Zend, rate /format, routing constraints, routing metrics
    - SLA : resiliency type, switching/ reversion parameters
  - Main Output parameters :
    - Equipment required at Aend, Zend and intermediate sites
  - No resources will be reserved, provisioned or planned because of this RPC

Service model rpcs (3/7)

- **service-restoration**
  - SDN Controller calls RNC to restore a service disrupted by a regenerator failure :
    - The restoration shall be carried out immediately, using spare regen(s) that can be on the same or on another path
  - This rpc can be used to reroute a service in 2 ways :
    - Case 1 : neither the backup-path-id nor the failure-case-id are provided
      - A new path shall be calculated according to the provided routing metrics. Initial routing constraints (at the time of service creation) shall be applied if permanent rerouting is asked , and the resulting path may use the same resources apart from regens
      - If path computation is successful, a new path is provided, and the service is rerouted according to that path.
    - Case 2 : backup-path-id or a failure-case-id are provided
      - the service shall be rerouted according to the corresponding path.
      - If selected backup-path is valid (not affected by the failure), the service is rerouted according to that path.
  - Main Input parameters : service-name, option (temporary, permanent), backup-path-id, failure-case-id , routing metrics
  - If this request passes the initial validation, a service-rpc-result notification is sent by the RNC, once processed
  - *Currently considering to apply this rpc to photonic/OTN restoration to overcome optical failures, even in case no regen are used on the path*
    - *backup path has been introduced in the meanwhile*
    - *dedicating service-reroute to maintenance operation*

orange™

Service model rpcs (4/7)

- **service-reroute**
  - SDN Controller calls RNC to restore a service that is affected or may be affected later on by ROADM line failures
    - fiber cut, optical amplifier failure, maintenance operation…
  - Service reroute is to be carried out immediately, on a temporary basis, without consideration of any routing constraints.
  - Since service re-route is always on a temporary basis, the RNC must mark the equipment and wavelengths in the original path as "Out of Service Maintenance" so that the rerouted service can be reverted back through "Service Reversion".
  - The RNC returns the SDN Controller the initial routing constraints applying to service
  - Main Input parameters : service-name, SLA (service-resiliency), routing metrics
  - Main Output parameters : routing constraints
  - If this request passes the initial validation, a service-rpc-result notification is sent by the RNC, once processed
- **service-reroute-confirm**
  - SDN Controller calls RNC to confirm the restoration of a service
  - Service restoration becomes permanent.
  - A new path calculation is performed by RNC according to routing constraints (that might have been updated by the SDN-controller) provided in the RPC and routing metrics provided previously through service-reroute-rpc
  - Main Input parameters : service-name, routing constraints
  - If this request passes the initial validation, a service-rpc-result notification is sent by the RNC, once processed

Service model rpcs (5/7)

- **service-reversion**
  - SDN controller requests RNC to revert the service that was restored or rerouted temporarily to the original equipment or path.
  - Expected to be performed in a maintenance window with a due date.
  - Main Input parameters : service-name, due date
  - If this request passes the initial validation, a service-rpc-result notification is sent by the RNC, once processed
- **service-reconfigure**
  - SDN controller requests the RNC to change the service to different terminating equipment:
    - re-home the service, to change the service path, and to route the service with different routing constraints etc.
  - Main Input parameters :
    - Service general information : service current & new name, common-id, connection-type
    - Technical requirements : Aend, Zend, rate /format, routing constraints, routing metrics
    - SLA : resiliency type, switching/ reversion parameters
  - If this request passes the initial validation, a service-rpc-result notification is sent by the RNC, once processed

Service model rpcs (6/7)

- **network-reoptimization**
  - As the network topology changes over time, the SDN Controller can periodically request the RNC to check whether any embedded services can be routed more efficiently without violating any routing constraints imposed on the services.
  - Main Input parameters : service-name, A & Z ends, customer-code, pass-through, routing metrics
    - Not only the identified service, but also all services having same-customer code, A or Z end and same pass-through designated sites shall be checked by the RNC
  - Main Output parameters : optimization –candidate
- **service-roll**
  - SDN controller requests the RNC to change the path of a service while keeping the same A and Z end points.
    - Mostly exercised by the SDN Controller following a network re-optimization request through which the RNC identified more optimal paths for some embedded services
  - A new path calculation shall be triggered, even in the case pre-calculated paths exist since the goal is to optimize network use
  - The new path must comply with the routing constraints that were imposed on the service initially.
  - Main Input parameters : service-name, due-date

Service model rpcs (7/7)

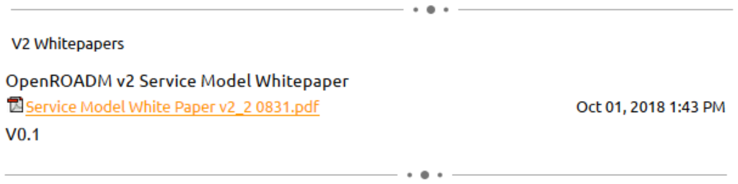
- **service-create-result-notification-request, service-delete-result-notification-request**
  - The RNC confirms the SDN controller that a service has been successfully created/deleted
  - Main Input parameters : service-name, common-id, version-identifier
- **ber-test**
  - The SDN controller requests RNC to perform a BER test on a service from the service-list
  - Main Input parameters : service-name, ber-options (duration, polling-timeout, retry-attempts, target-prefec-ber)
- **service-rpc-ber-test-async-callback RPC,**
  - The RNC provides BER test results to the SDN controller confirming that a service has been successfully created
  - Main Input parameters :
    - initial ber-test rpc input parameters, timestamps (initial rpc / callback), service line rate
    - details of the BER test results : measured-prefec-ber (a-end & z-end), status of the BER tests (pass/fail)

Service model notifications

- ***Notifications***
  - *A Notification is a Request object without an "id" member. No Response object needs to be returned to the client.*
  - *Notifications are not confirmable by definition, since they do not have a Response object to be returned.*
- **service-rpc-result**
  - This Notification indicates result of service RPC
  - Provides service-name and actual-date, status (successful/failed), status message, and notification-type
- **service-traffic-flow**
  - Sent by RNC to SDN controller to indicate that traffic is flowing again on the service after an administrative action has completed
  - Provides service-name and actual-date
- **service-notification**
  - Sent by RNC to SDN controller to advertise that a service has been added, modified or removed.
  - Provides information about the service in its entirety (creation), the modified field and the service identifier (modified), the service identifier (deletion)
- **service-notification-ber-test,**
  - Sent by RNC to provide SDN controller with BER test results, confirming the that a service has been successfully created
  - Main Input parameters :
    - initial ber-test rpc input parameters, timestamps (initial rpc / callback), service line rate
    - details of the BER test results (measured-prefec-ber for a-end & z-end, status of the BER tests)

Thanks

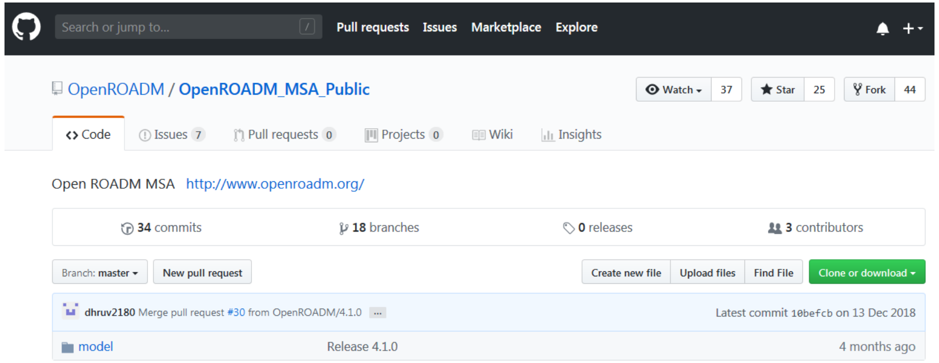
- **For more information :**
  - Open ROADM public site : http://www.openroadm.org/home.html
  - White papers available using Download menu : http://www.openroadm.org/download.html
  - Yang Models : https://github.com/OpenROADM/OpenROADM_MSA_Public


V2 Whitepapers

OpenROADM v2 Service Model Whitepaper

Service Model White Paper v2_2 0831.pdf — Oct 01, 2018 1:43 PM

V0.1



Open ROADM MSA http://www.openroadm.org/

34 commits · 18 branches · 0 releases · 3 contributors

dhruv2180 Merge pull request #30 from OpenROADM/4.1.0 ... Latest commit 10befcb on 13 Dec 2018

model — Release 4.1.0 — 4 months ago