



# ODL and YANG

LFN Developer & Testing Forum - Feb 2021

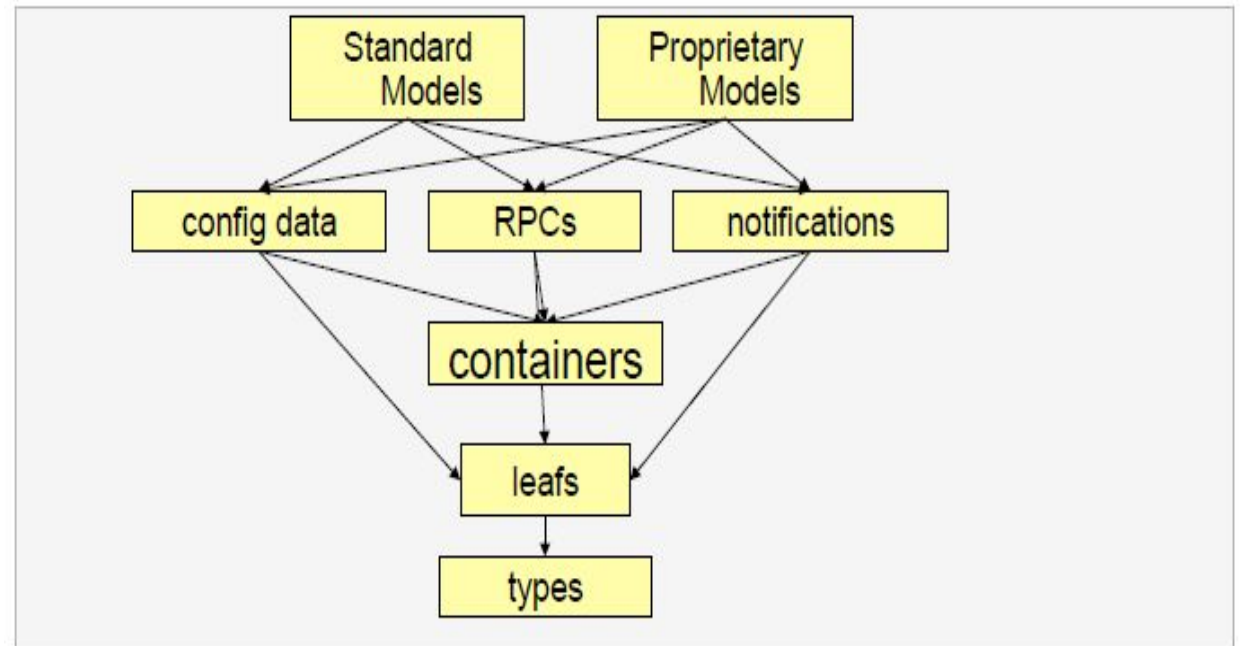
Luis Gomez Palacios  
Kratos Defense & Security Solutions



**YANG**

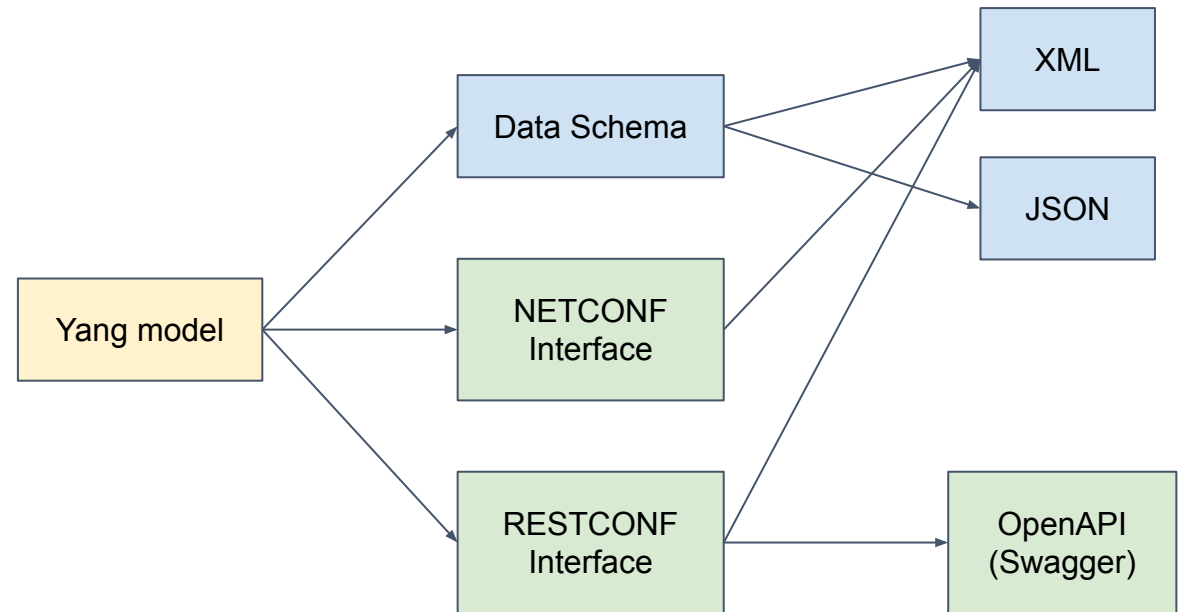
# What is YANG

- YANG (RFC 6020, RFC 7950) is a data modeling language used to model configuration and state data manipulated by the NETCONF protocol (RFC 6241), NETCONF remote procedure calls, and NETCONF notifications.
- YANG models the hierarchical organization of data as a tree in which each node has a name, and either a value or a set of child nodes.



# YANG Coding and Protocols

- YANG modeled data can be encoded in XML (RFC 6020, RFC 7950) or JSON (RFC 7951).
- YANG modeled data can be manipulated by the NETCONF protocol (RFC 6241) or by the RESTCONF protocol (RFC 8040).



# What is YANG used for?

“To model Network Information and Interfaces”

- **Device Management:** Originally designed to model the management interface to a network device (NETCONF) and by so to replace SNMP.

It has expanded to other areas like:

- **Network Topology:** Used to model different network topologies (IP, Ethernet, Optical, etc) and their relations (overlay, underlay, etc).
- **Network Services:** Used to model connection services (MPLS, Ethernet, Optical), VPN services, NFV services, etc.

# What is YANG used for?

“To standardize Network Information and Interfaces ”

- Standard Developing Organizations (SDOs) use YANG models to standardize network objects and application programmable interfaces (APIs).
- Interface standardization has enormous value as it allows for multiple vendors network functions to coexist and interact with each other.



# Device Management Models

- **IETF** models for basic types, interfaces, IP, routing, etc:  
<https://github.com/YangModels/yang/tree/master/standard/ietf/RFC>
- **OpenConfig** models for interfaces, routing, L2 and optical configuration:  
<https://www.openconfig.net/projects/models>
- **OpenROADM** models for WDM and OTN configuration:  
[https://github.com/OpenROADM/OpenROADM\\_MSA\\_Public/tree/master/model/Device](https://github.com/OpenROADM/OpenROADM_MSA_Public/tree/master/model/Device)
- **ONF TR 532** model for microwave configuration:  
<https://github.com/OpenNetworkingFoundation/5G-xHaul/blob/master/models/yang/microwave-model.yang>
- **BBF** models for Broadband Access (FTTx) configuration:  
<https://github.com/BroadbandForum/yang>
- **O-RAN** models for RAN configuration:  
<https://gerrit.o-ran-sc.org/r/gitweb?p=scp/oam/modeling.git;a=tree;f=data-model/yang/working/o-ran-sc>

# Network Topology Models

- **IETF** models for basic, L2, L3, TE, DC fabric topologies:  
<https://github.com/YangModels/yang/tree/master/standard/ietf/RFC>
- **OpenROADM** models for WDM and OTN topologies:  
[https://github.com/OpenROADM/OpenROADM\\_MSA\\_Public/tree/master/model/Network](https://github.com/OpenROADM/OpenROADM_MSA_Public/tree/master/model/Network)
- **ONF T-API** models for Ethernet and Optical topologies:  
<https://github.com/OpenNetworkingFoundation/TAPI/tree/v2.2.0/YANG>



# Network Services Models

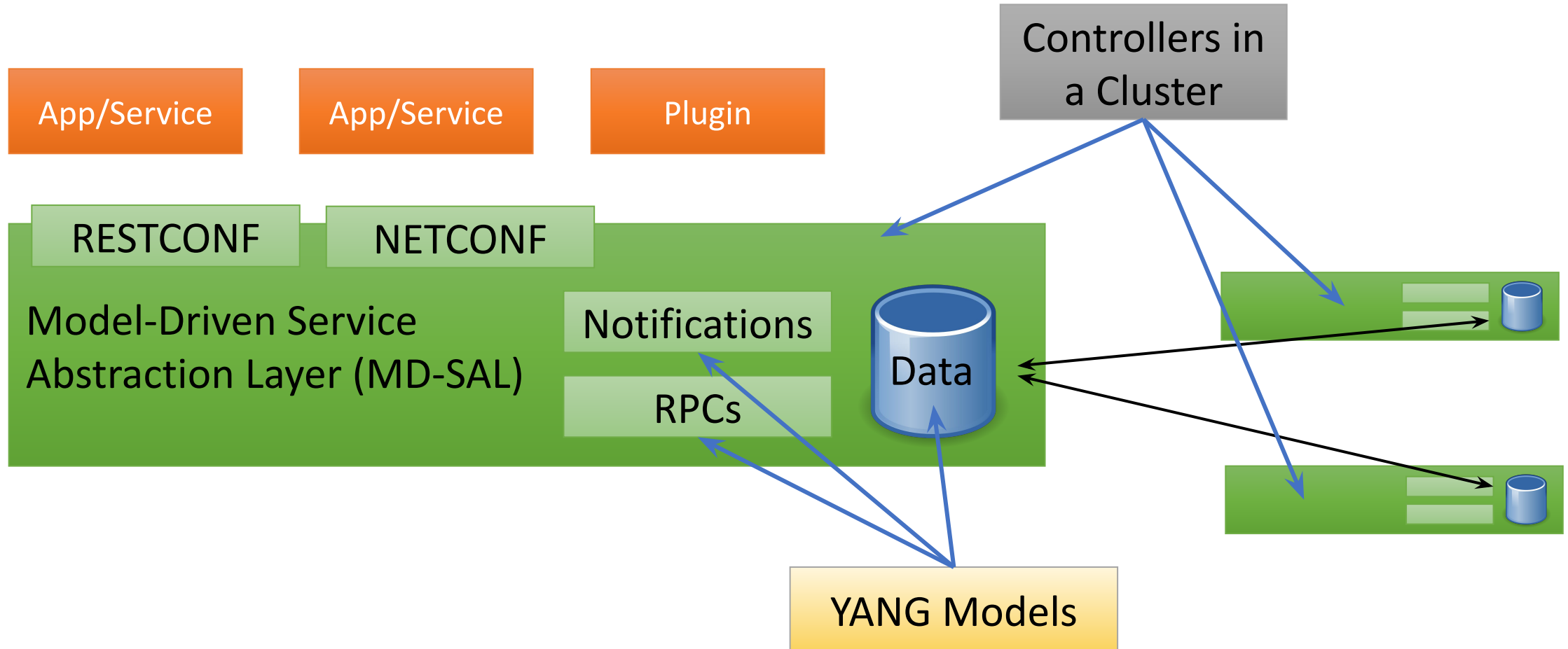
- **IETF** models for L2VPN and L3VPN services:  
<https://github.com/YangModels/yang/tree/master/standard/ietf/RFC>
- **OpenROADM** models for WDM and OTN services:  
[https://github.com/OpenROADM/OpenROADM\\_MSA\\_Public/tree/master/model/Service](https://github.com/OpenROADM/OpenROADM_MSA_Public/tree/master/model/Service)
- **ONF T-API** models for Connection services:  
<https://github.com/OpenNetworkingFoundation/TAPI/tree/v2.2.0/YANG>
- **MEF** models for Ethernet services:  
<https://github.com/MEF-GIT/YANG-public/tree/master/src/model/standard>
- **ETSI MANO** models for NFV services:  
<https://github.com/etsi-forge/NFV-SOL006/tree/master/src/yang>

# YANG Resources:

- Yang Catalog (model search):
  - <https://yangcatalog.org>
- Yand Repository:
  - <https://github.com/YangModels/yang>
- Pyang (Yang validator, Yang to XML/JSON schema, Yang to UML):
  - <https://github.com/mbj4668/pyang>
  - <https://github.com/cmoberg/pyang-json-schema-plugin>
- UML To Yang:
  - <https://github.com/OpenNetworkingFoundation/EagleUmlYang>
- Yang To OpenAPI:
  - <https://github.com/OpenNetworkingFoundation/EagleYangOpenApi>
  - <https://github.com/corenova/yang-swagger>
  - <https://github.com/bartoszm/yang2swagger>

# ODL and YANG

# YANG at the Core of ODL



# YANG at the Core of ODL

“Model-Driven SAL = YANG-driven SAL”

OpenDaylight provides a framework for applications and inter-application communication:

- YANG datastore (configuration + operational) and data change notifications (DCN).
- YANG RPCs routing and notifications pub/sub mechanism.
- YANG auto-generated Java Classes (interface) to access the data.
- Automatic RESTCONF/NETCONF external interface.
- Support for user-defined transport and payload formats, including payload serialization and adaptation (e.g. binary, XML or JSON).

# ODL YANG Leverage

ODL can be used to:

- Manage NETCONF/YANG devices via RESTCONF (NETCONF plugin).
- Quickly develop YANG defined API applications (Archetype).
- Generate OpenAPI (Swagger) UI and documents from YANG.
- Test RESTCONF interface.
- Test YANG models.

# ODL NETCONF Plugin

## The ODL NETCONF plugin:

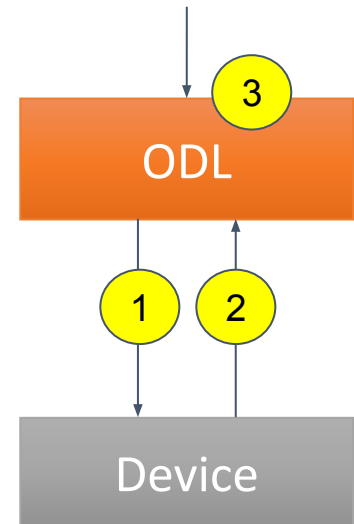
1. Connects to a NETCONF/YANG device.
2. Imports the device YANG models.
3. Generates RESTCONF API from YANG models.

Device data schema is mounted on the NETCONF topology:

<http://localhost:8181/rests/data/network-topology:network-topology/topology/topology-netconf/node/<node-id>/yang-ext:mount/>

More information in:

<https://docs.opendaylight.org/projects/netconf/en/latest/user-guide.html>







# ODL API Explorer

You can open this URL in a browser and point to ODL (odl-mdsal-apidocs installed) to access the OpenAPI UI and docs for your YANG models: <http://localhost:8181/apidoc/explorer/index.html>

The screenshot displays the OpenDaylight RestConf API Documentation interface. At the top, there is a black header with the OpenDaylight logo and the text "OpenDaylight RestConf API Documentation". To the right of the header, there is a dropdown menu labeled "Select controller/mounted resources of specific RestConf version". Below the header, the main content area shows "Controller modules of RestConf version RFC8040" with two version tags: "1.0.0" and "OAS3". A URL is provided: <http://localhost:8181/apidoc/openapi3/18/apis/single>. Below this, there is a "Servers" section with a dropdown menu showing "http://localhost:8181/". The main content area is titled "aaa" and contains a list of API endpoints for the "aaa" module. The endpoints are:

- POST** /rests/data/aaa:authentication POST - aaa - authentication
- GET** /rests/data/aaa:authentication GET - aaa - authentication
- DELETE** /rests/data/aaa:authentication DELETE - aaa - authentication
- PUT** /rests/data/aaa:authentication PUT - aaa - authentication



Thank You