API based Network Automation Orchestration

JUN-2023
Mahesh Jethanandani
- Chair NETCONF WG, IETF
- Distinguished Engineer, Arrcus
Orchestration vs Automation

Orchestration

Automating many tasks as a process or workflow.

Automation

Setting up one task to run on its own.

Source: g2.com
Agenda

• Motivation
• What are
  • NETCONF,
  • RESTCONF, and
  • YANG
• Landscape of YANG Models
• Tools
• Development Resources
• Conclusions & References
Facts

- 95% of network changes are manual
- Manual changes are error prone
- Expanding network at scale is a problem
  - 75 billion IoT devices by 2025
Key Drivers

• End to end management
  • Access, transport and core
• Biggest area for capex and opex reduction
  • 60 of 76 operators (survey conducted between 8/22-10/22)
• Benefits include reduction of call-centers
  • 15-20% reduction
• Agility
• Time to Market
Where We Need To Go

- Networking is well known, we can build stable abstractions
- Modern software practices gives us reusability
- The competitive edge is in the applications

- Professional Services 60%
- In-house Development 30%
- Software 10%

- In-house Development 10%
- Software 90%
Best Practices Coming Together

- SNMP Experience
- CLI Best Practices
- Operations Requirements

NETCONF, RESTCONF and YANG
Agenda

• Motivation
• What are
  • *NETCONF*,
  • RESTCONF, and
  • YANG
• Landscape of YANG Models
• Tools
• Development Resources
• Conclusions & References
NETCONF/RESTCONF and YANG

Apps

APIs

Protocol

Encoding

Transport

Models

Devices

YANG Data Models

Apps

API

NETCONF

RESTCONF

gRPC

XML

JSON

Binary

SSH

HTTP(S)

Arrcus

Cisco

Juniper
NETCONF Protocol (RFC 6241)

- Network Management Protocol
- Distinction between configuration and state data
- Multiple configuration data stores (candidate, running, startup, operational)
- Configuration change validations and transactions
- Selective data retrieval with filtering
- Streaming and playback of event notifications

Why you should care:
NETCONF provides fundamental programming features for comfortable and robust orchestration of network services
NETCONF Transactions

Transaction Definition: the “ACID test”
- Atomicity: all-or-nothing, great for error handling
- Consistency: all-at-once, great for integrity
- Independence: no-crosstalk, great for many concurrent clients
- Durability: done-is-done, great for reliability

Introduction of transactions + SQL caused a DB industry boom in the 80’s. Applications got reliable. Could run against many different DBMS’

*NETCONF makes the network a distributed database*
Agenda

• Motivation
• What are
  • NETCONF,
  • RESTCONF, and
  • YANG
• Landscape of YANG Models
• Tools
• Development Resources
• Conclusions & References
RESTCONF Protocol (RFC 8040)

- Network Management Protocol
- Not intended to replace NETCONF, but rather provide an additional simplified REST interface
- Defines HTTP-based Create, Retrieve, Update, Delete (CRUD) operations
- Configuration data and state data exposed as resources
- Operations defined with YANG rpc invoked with the POST method
- Simplified transaction model

Why you should care:
RESTCONF provides a lighter-weight interface to NETCONF data stores leveraging the well known combination of HTTP and JSON/XML/CBOR
Agenda

• Motivation
• What are
  • NETCONF,
  • RESTCONF, and
  • YANG
• Landscape of YANG Models
• Tools
• Development Resources
• Conclusions & References
The YANG 1.1 Language (RFC 7950)

A Data Modeling Language for Networking

- Human readable and easy to learn
- Hierarchical configuration/state data models
- Reusable types and groupings (structured types)
- Extensibility through augmentation
- Formal constraints for configuration validation
- Data modularity through modules and sub-modules
- Well defined versioning rules

Why you should care:

YANG is a full, formal contract language with rich syntax and semantics to build applications on
Service vs Infra vs Device Level Models

- Services Level Models
  - End-to-End Services
  - E-Line, E-LAN, E-Tree ([MEF 58](https://github.com/MEF-GIT/))
  - Legato
- Infra Level Models
  - Topology Models
  - Presto
- Device Level Models
  - Device Specific
  - Adagio
- [https://github.com/MEF-GIT/](https://github.com/MEF-GIT/)
Agenda

• Motivation
• What are
  • NETCONF,
  • RESTCONF, and
  • YANG
• Landscape of YANG Models
• Tools
• Development Resources
• Conclusions & References
YANG Tsunami in the industry
Standards Body Models

- IETF YANG models
  - ~1500 YANG Models
  - GitHub ([https://github.com/YangModels/yang](https://github.com/YangModels/yang))
- BBF, IEEE, ETSI, IANA, MEF
OPENCONFIG

• Operators-led YANG models
  • Google, AT&T, British Telecom, Microsoft, Facebook, Comcast, Verizon, Level3, Cox Communications, Yahoo!, Apple, Jive Communications, Deutsche Telekom / TeraStream, Bell Canada
  • (https://github.com/openconfig/public)
• YANG models not aligned with other SDO, e.g. IETF
VENDOR SPECIFIC MODELS

• Augmentation of Standard or OpenConfig YANG models
• Vendor Specific Features
  • Missing capabilities
  • Support existing CLI
• Not compatible with other vendor models
Agenda

• Motivation
• What are
  • NETCONF,
  • RESTCONF, and
  • YANG
• Landscape of YANG Models
• Tools
• Development Resources
• Conclusions & References
Validating YANG models

- **pyang**
  - An extensible YANG validator written in Python. (Video training: [pyang](https://github.com/mbj4668/pyang))
  - Can be used standalone as a validator of YANG modules, or to generate YIN, YANG, DSDL and XSD from YANG and YIN.
  - [https://github.com/mbj4668/pyang](https://github.com/mbj4668/pyang)

- **libyang**
  - yanglint – tool to validate YANG models and examples
  - [https://github.com/CESNET/libyang](https://github.com/CESNET/libyang)
Client tools

• Interacting with servers (NETCONF and RESTCONF)
  • netconf-client - A NETCONF client for Python 3.6+.
  • ncclient - Python library for NETCONF clients (ncclient.org)
  • curl - command line tool and library for transferring data with URLs (curl.haxx.se)
  • Python requests - the only Non-GMO HTTP library for Python (python-requests.org)
  • Postman – REST(conf) requests
Orchestrators

- OpenDaylight
  - Enables auto-generation of RESTconf APIs from YANG models, with NETCONF client support
  - APIdocs feature provides a way to explore both local and mounted YANG models
- Network System Orchestrator (NSO)
- Others
Agenda

- Motivation
- What are
  - NETCONF,
  - RESTCONF, and
  - YANG
- Landscape of YANG Models
- Tools
- Development Resources
- Conclusions & References
NAPALM

- Network Automation and Programmability Abstraction Layer with Multivendor support
- Python 3 library
- Support for multiple vendors including Arrcus
- Template based
  - Replace - `device.load_replace_candidate(filename='new_good.conf')`
  - Merge - `device.load_merge_candidate(filename='new_good.conf')`
  - Commit - `device.commit_config()`
  - Rollback - `device.rollback()`
API documentation

- Swagger Editor

```plaintext
GET /data/openconfig-interfaces:interfaces/interface={interface-name}/subinterfaces/subinterface={subinterface-index}/state

Example:
```curl
curl -X 'GET' \
'http://127.0.0.1:8888/restconf/data/openconfig-interfaces:interfaces/interface=eth0' \
-H 'accept: application/yang-data+json'
```

Server response:

```json
{
  "openconfig-interfaces:state": {
    "index": 0,
    "name": "string",
    "description": "string",
    "enabled": true,
    "ifindex": 0,
    "admin-status": "UP",
    "oper-status": "UP",
    "last-change": 0,
    "counters": {
      "in-octets": 0,
      "in-unicast-pkts": 0,
      "in-broadcast-pkts": 0,
      "in-multicast-pkts": 0,
      "in-discard": 0,
      "in-errors": 0,
      "in-unknown-protos": 0,
      "out-octets": 0,
      "out-unicast-pkts": 0,
      "out-broadcast-pkts": 0,
      "out-multicast-pkts": 0,
    }
  }
}```
Programming Resources

- libyang-cpp – C++ bindings for YANG
- libyang-python – Python bindings for YANG
Agenda

• Motivation
• In Configuration, what are
  • NETCONF,
  • RESTCONF, and
  • YANG
• Standard, Opensource, and Vendor Models
• Management
• Tools
• Development Resources
• Conclusions & References
Conclusions

• Critical mass of models available
• Model first
  • Consistent interface between CLI and API
• Network orchestration is where dB was 30 years ago
• Services to be deployed in minutes instead of days
References

- YANG Catalog  
  https://yangcatalog.org
Thank you

13-June-2023
CLI to API

CLI Challenge
- No deterministic results (expect scripts)
- Any errors result in human intervention
- Change scripts are scary to use (no trust)
- Limited to read-only or reporting
- No true automation, always ends with human involvement

API Automation
- APIs offer a much better method of automating
- APIs are made for machine-first interfaces
- Standardization and definition (OpenAPI)
- Rich set of exposed methods
- Clear pass/fail results with more details
- Modern network systems have APIs
- APIs enable more automations across multiple systems that operate in a way you can trust

Source: itential.com
Agenda

• Motivation
• In Configuration, what are
  • NETCONF,
  • RESTCONF, and
  • YANG
• Standard, Opensource, and Vendor Models
• Management
• Tools
• Conclusions & References
RESTCONF Datastores (before NMDA)

Complete and active configuration and operational data
Beware Confusion on Information vs Data Models

Information Models
• Conceptual models
• Independent of implementations or protocols

Data Models
• Precise models
• Include protocol-specific constructs.

RFC3444 On the Difference between Information Models and Data Models
# Management

- Do not forget monitoring

<table>
<thead>
<tr>
<th></th>
<th>SNMP</th>
<th>Telemetry</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>How it works</strong></td>
<td>Polling (Pull)</td>
<td>Push</td>
</tr>
<tr>
<td><strong>Protocol</strong></td>
<td>UDP</td>
<td>UDP or TCP</td>
</tr>
<tr>
<td><strong>Use Cases</strong></td>
<td>Static data, inventory</td>
<td>Performance, high speed data</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>Simple, widely supported</td>
<td>Higher rate, selective</td>
</tr>
<tr>
<td><strong>Challenges</strong></td>
<td>Pull</td>
<td>Can overwhelm the server</td>
</tr>
</tbody>
</table>
Model Driven Telemetry
Exploring and using YANG

- Editor plug-ins
  - emacs (yang-mode.el)
  - vim (yang.vim)
  - sublime text (sublime-yang-syntax)

- libsmi
  - A library allowing the generation of YANG models from SMI/SMIv2 compliant MIBs