



TIAD

Network Automation with Ansible and OpenConfig / YANG

— Marcel Wiget & Khelil Sator —

Juniper Networks

October 2016

Agenda

- One Protocol – many languages to master ...
- From CLI scraping to NETCONF
- NETCONF Python Tools
- Common Data Model Language YANG
- OpenConfig
- Ansible for Juniper Network Devices
- Demo

One Protocol – many languages to master ...

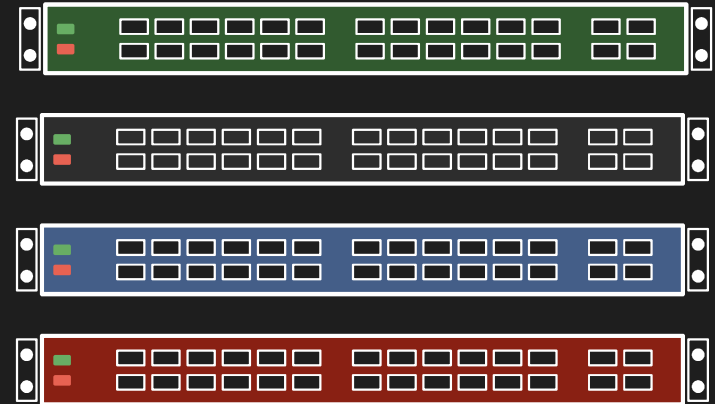


```
router bgp as-number 12345
```

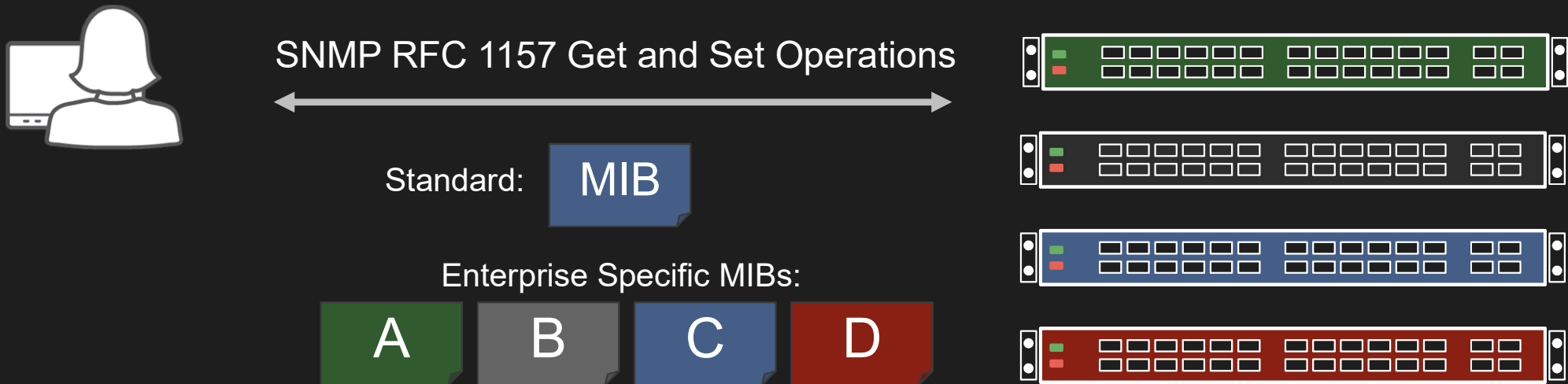
```
router bgp 12345
```

```
configure router autonomous-system 12345
```

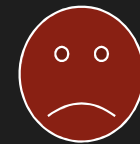
```
set routing-options autonomous-system 1234
```



Wait, we have been here 25 years ago ...



Standard and Enterprise specific MIBs don't mix !



Why CLI scraping is bad ...

Simple Task: Get IP address of a Linux Server

```
IP=$(hostname --ip-address | awk '{print $1}')
```

Host1: hostname --ip-address
172.17.0.3



Host2: hostname --ip-address
172.17.0.4 172.20.0.3 172.18.0.2 172.19.0.2



Host3: hostname --ip-address
2a05:4f8:1130:55b0::2 193.5.1.23



The Network Configuration Protocol - NETCONF

- Protocol to “install, manipulate and delete configuration”
- Extensible Markup Language (XML)-based data encoding for configuration data and protocol messages
- NETCONF protocol operations over a simple RPC layer
- Defined by the IETF in RFC 4741 in 2006, updated in RFC 6241
- NETCONF over SSH first defined in RFC 4742, updated in RFC 6242



NETCONF Python Tools

- ncclient: Python library for NETCONF clients (<http://ncclient.org/>)
Can automate operations, but still low level operation
- Junos PyEZ uses ncclient and adds a level of abstraction
→ “easy” to use, even for network operators
<https://github.com/Juniper/py-junos-eznc>
- Ansible uses PyEZ



YANG

- Data model language for the Network Configuration Protocol (NETCONF)
- Defined by the IETF in RFC 6020 in 2010
- YANG is an acronym for "Yet Another Next Generation"
- Human readable
- Any encoding format, including XML and JSON
- Transport over NETCONF over SSH and recently also over gPRC

<https://tools.ietf.org/html/rfc6020>

YANG Schema Example

```
grouping bgp-global-config {
```

```
...
```

```
    leaf as {
```

```
        type inet:as-number;
```

```
        mandatory true;
```

```
        description
```

```
            "Local autonomous system number of the router. Uses  
            the 32-bit as-number type from the model in RFC 6991.";
```

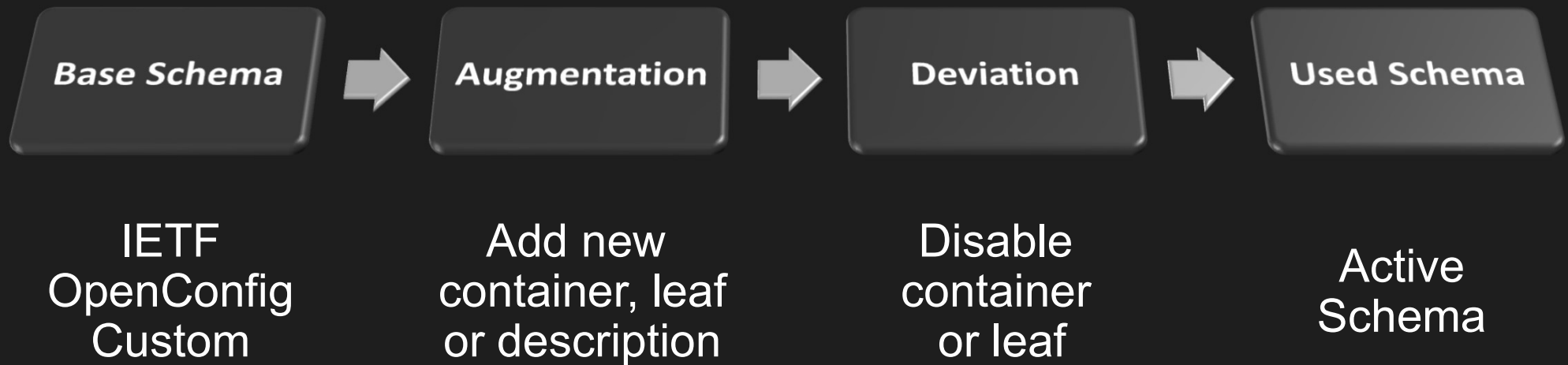
```
    }
```

```
...
```

[edit openconfig-bgp:bgp global config]mwiget@oc1# set as ?

Possible completions: <as> Local autonomous system number of the router. Uses the 32-bit as-number type from the model in RFC 6991.

YANG Augmentation and Deviation



maintained in separate text files

OpenConfig

<http://www.openconfig.net>

- Vendor-neutral, model-driven network management designed by users
- Common Data Models written in YANG
- Streaming Telemetry
- Published on GitHub



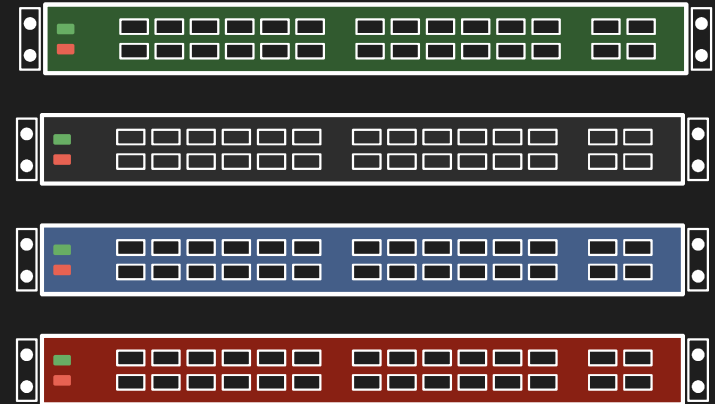
OpenConfig

One Protocol – one language to master all vendors

```
openconfig-bgp:bgp global as 12345
```



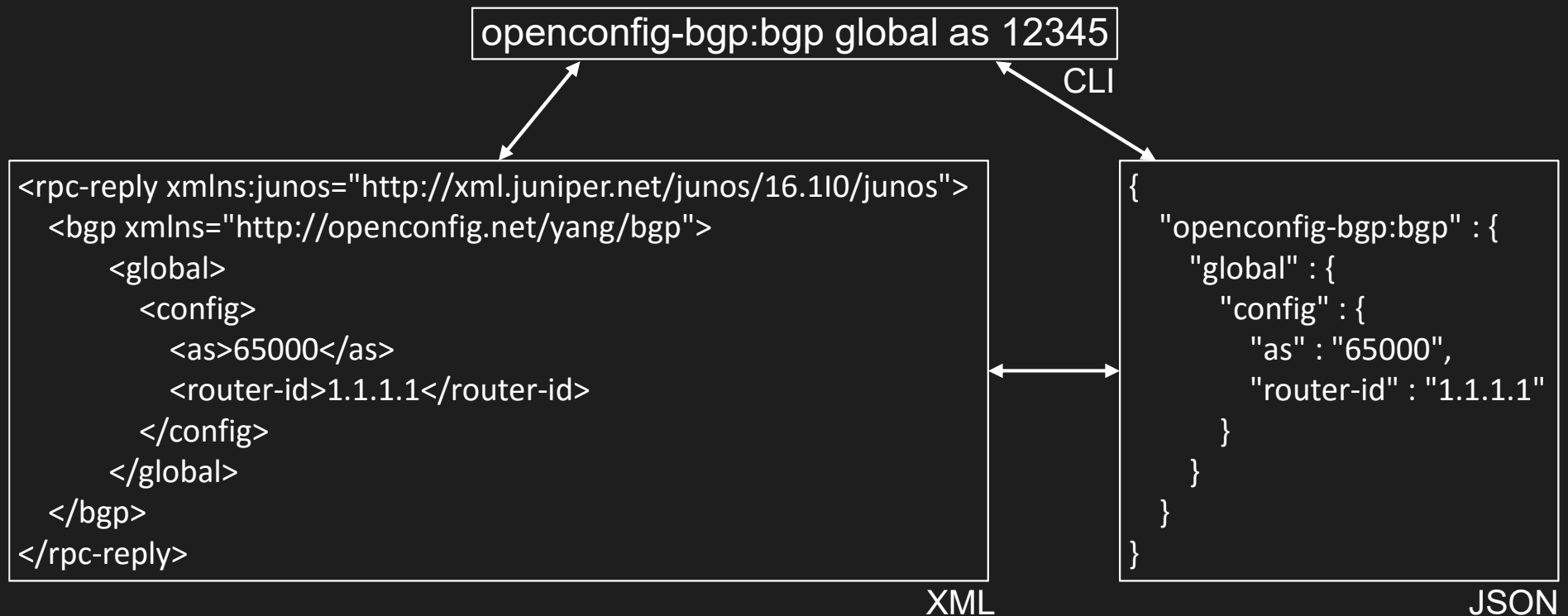
(over CLI, NETCONF/SSH & gRPC)



<http://www.juniper.net/us/en/solutions/automation/>

OpenConfig

One Protocol – one language, many encoding



ANSIBLE

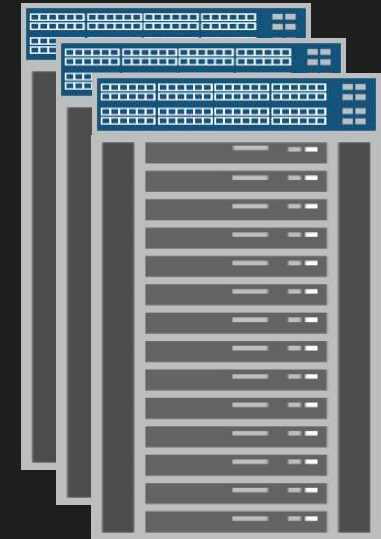


Ansible

**Standard Mode: sends
Python module over SSH,
executes, then removes it**

Python module over SSH

**Network
Devices**



Servers

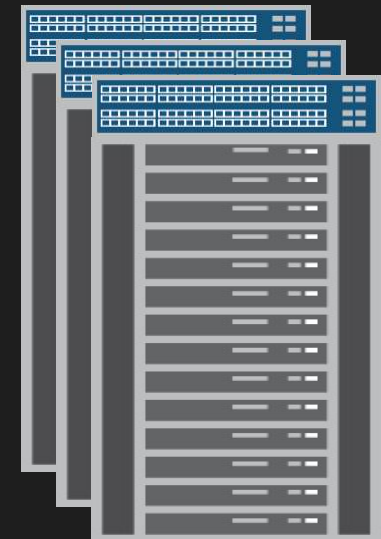
ANSIBLE + JUNIPER NETWORKS



NETCONF over SSH

**API Mode:
Modules run on server,
communicate over an API**

Network
Devices



Servers

<https://www.ansible.com/ansible-juniper>

Junos Core Modules in Ansible 2.1

<code>junos_config</code>	Deploy configuration lines, zeroize, rollback
<code>junos_template</code>	Deploy configuration template or file
<code>junos_facts</code>	Gather device facts and configuration
<code>junos_packages</code>	Deploy and install packages/os on devices
<code>junos_command</code>	Execute any CLI or RPC commands remotely
<code>junos_netconf</code>	Enable NETCONF on devices



DEMO !

<https://github.com/ksator/openconfig-demo>



Follow us on Twitter:

@JunosAutomation

@KhelilSator

@MarcelWiget

- <https://github.com/ksator/openconfig-demo>
- <https://www.ansible.com/ansible-juniper>
- <https://www.juniper.net/developer>
- <https://www.juniper.net/us/en/solutions/automation/>