Streaming telemetry
part II

Vlad Hanciuta, Sergey Kolobov
iNOG::F @ ARISTA Networks
Dublin, Ireland
February 28, 2018

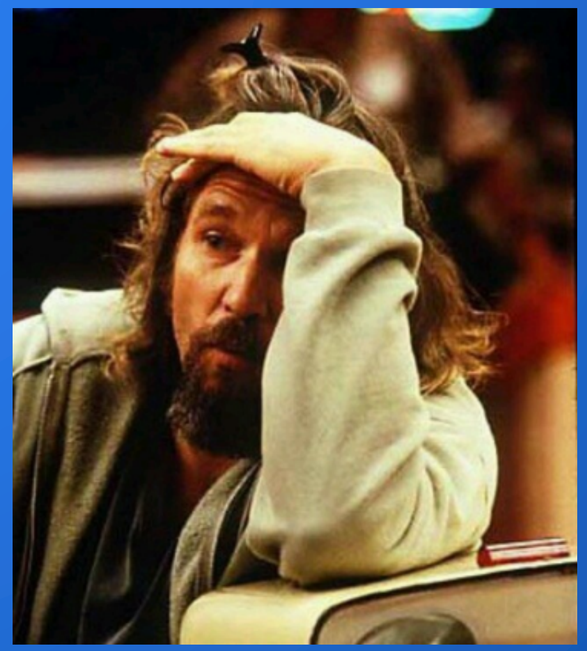
A blast from the past:

Why can’t we just keep using traditional methods?

- scalability and efficiency challenges
- reactive approach challenges
- challenges to get granular data

What majority of network operators are still doing today?

- Cut-and-paste config management
- One step at a time: box-by-box-by-box ….
- Polling based monitoring



Moving forward:

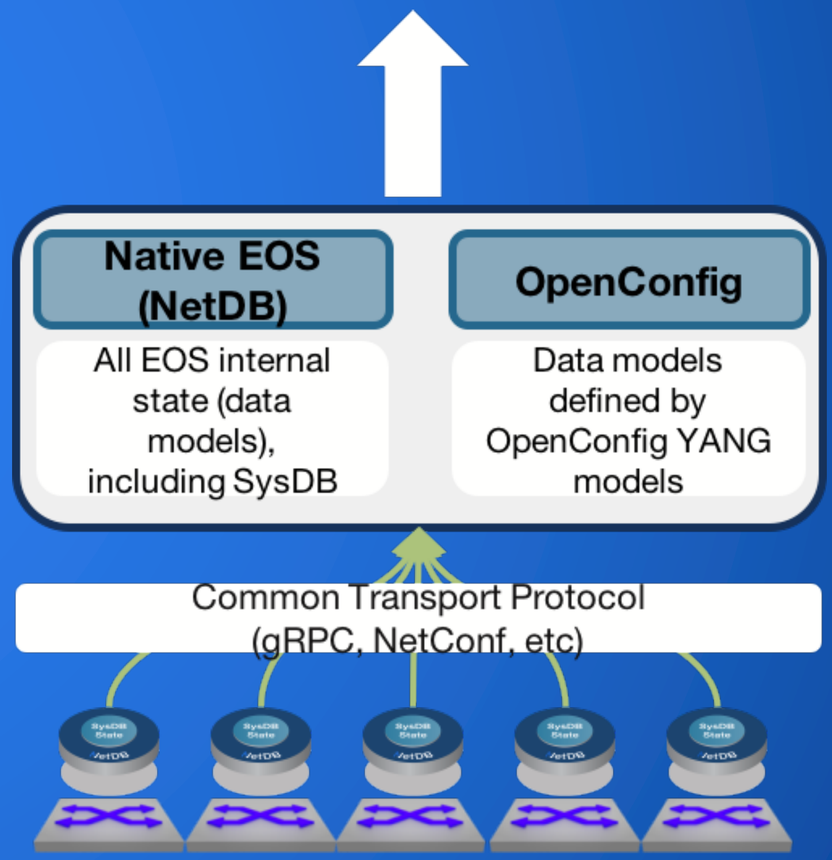
1. What alternatives are available?
   - Arista Networks ST agent: TerminAttr
   - OpenConfig

ARISTA

+

OPENCONFIG

1. How these alternatives are working?
   - Streaming out to subscribers
   - Trigger notification upon value change
   - Get data in real time



Recommended approach:

- Arista TerminAttr for all the raw state
- OpenConfig for vendor-independent state
- *BMP for BGP with Performance targets
  a. 20M paths
  b. 1000 changes/sec

OPENCONFIG

driven by a number of innovative participants in community:

SK telecom

Bell

T··

COMCAST

Google

ORACLE

Microsoft

NETFLIX

facebook

verizon

AT&T

BT

Level (3) COMMUNICATIONS

JIVE

Tencent 腾讯

CLOUDFLARE

Baidu 百度

Consistent set of vendor-neutral data models written in Yang:

- Network instances
- Routing policies
- Segment Routing
- BFD
- Interfaces
- L2
- Platform
- System
- ACL
- QoS
- Optical Transport
- Wi-Fi
- System Management

https://github.com/openconfig/public/tree/master/release/models

```
+-- rw interfaces
   +-- rw interface* [name]
      +-- rw config
      |  +-- rw description?       String
      |  +-- rw enabled?           Boolean
      |  +-- rw mtu?               Uint16
      |  +-- rw type_              Identityref
      |  +-- rw name?              String
      +-- ro state
         +-- ro description?       String
+-- ro name?                       String
+-- ro enabled?                    Boolean
+-- ro oper_status                 Enumeration
+-- ro mtu?                        Uint16
+-- ro type_                       Identityref
+-- ro admin_status                Enumeration
+-- ro ifindex?                    Uint32
+-- rw counters
   +-- rw in_octets?               Counter64
   +-- rw in_discards?             Counter64
   +-- rw last_clear?              DateAndTime
```

YANG types

- **leaf**: a leaf holds a strongly typed basic type, such as a a number, string, or network address this is a value inside a key value pair.
- **leaf-list**: this is simply a list of leaf nodes
- **container**: holds other types of information. Very similar to a dictionary where you can see KV pairs of information.
- **list**: a keyed container that can contain any other data type in a container
- **Config**: Configuration for a device
- **State**: State of a device

BGP Neighbor yang model:

```
leaf neighbor-address {
    type oc-inet:ip-address;
    description
      "Address of the BGP peer, either in IPv4 or IPv6";
```

```
leaf-list supported-capabilities {
  type identityref {
    base oc-bgp-types:BGP_CAPABILITY;
  }
  description
    "BGP capabilities negotiated as supported with the peer";
```

```
container prefixes {
  description "Prefix counters for the BGP session";
  leaf received {
    type uint32;
    description
      "The number of prefixes received from the neighbor";
  }
```

```
list neighbor {
  key "neighbor-address";
  description
    "List of BGP neighbors configured on the local system,
    uniquely identified by peer IPv[46] address";
```

https://github.com/openconfig/public/tree/master/release/models

Building device interaction layer:

	REST	gRPC	NETCONF	RESTCONF
Terminattr	✓	✓		
OpenConfig		✓	✓	✓

The exact same open-source interface

https://github.com/openconfig/reference/tree/master/rpc/gnmi

…It’s 2018, we can do better than XML...

Streaming Telemetry and Configuration

- **get current state/configuration:**
  - gnmi -addr host.ip:6030 -username admin get /interfaces/interface[name=Ethernet20]

- **subscribe to a stream:**
  - gnmi -addr host.ip:6030 -username admin subscribe /interfaces/interface[name=Ethernet20]

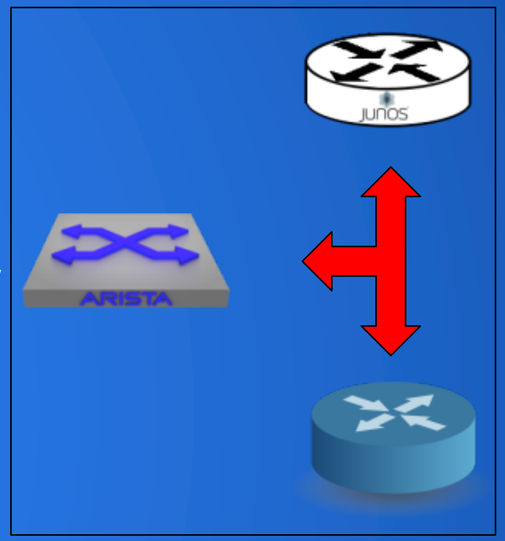
- **change the configuration:**
  - gnmi -addr host.ip:6030 -username admin replace /network-instances/network-instance[name=default]/protocols/protocol[identifier=BGP][name=BGP]/bgp/global' '{"config":{"as" : 123, "router-id" : "1.2.3.4"}}'
  - gnmi -addr host.ip:6030 -username admin update '/interfaces/interface[name=Ethernet20]/config/enabled' 'false'

How to enable community to make use of created data models:

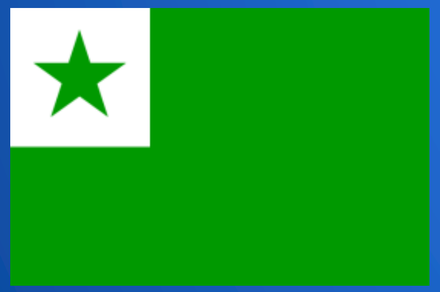
- operators and engineers are writing automation
- focus is on config artifacts

ANSIBLE SALTSTACK puppet

- modeling language is often out of the picture
- but YANG is fundamental w.r.t. vendors interoperability



- tools needed in common languages to help build model validation



Thank you!
Questions?