

IEEE 802.1Q YANG Modeling Style

Marc Holness (mholness@ciena.com)
Version 0.1
March 2018

YANG Data Model

- We are using YANG to model the configuration and operational state data in our devices (e.g., Bridges)
- Terminology of types of data
 - a) Intended Configuration

Represents the configuration state that the NMS intends the device to be in (and that has been accepted by the device as valid configuration)
 - b) Applied Configuration

Represents the configuration state that the device is actually in. That is, the configuration state which is currently being used by the device.
 - c) Derived State

Represents information which is generated as part of the devices interactions (e.g., protocol interactions, statistics, counters, etc.)

YANG Modeling Structural Style

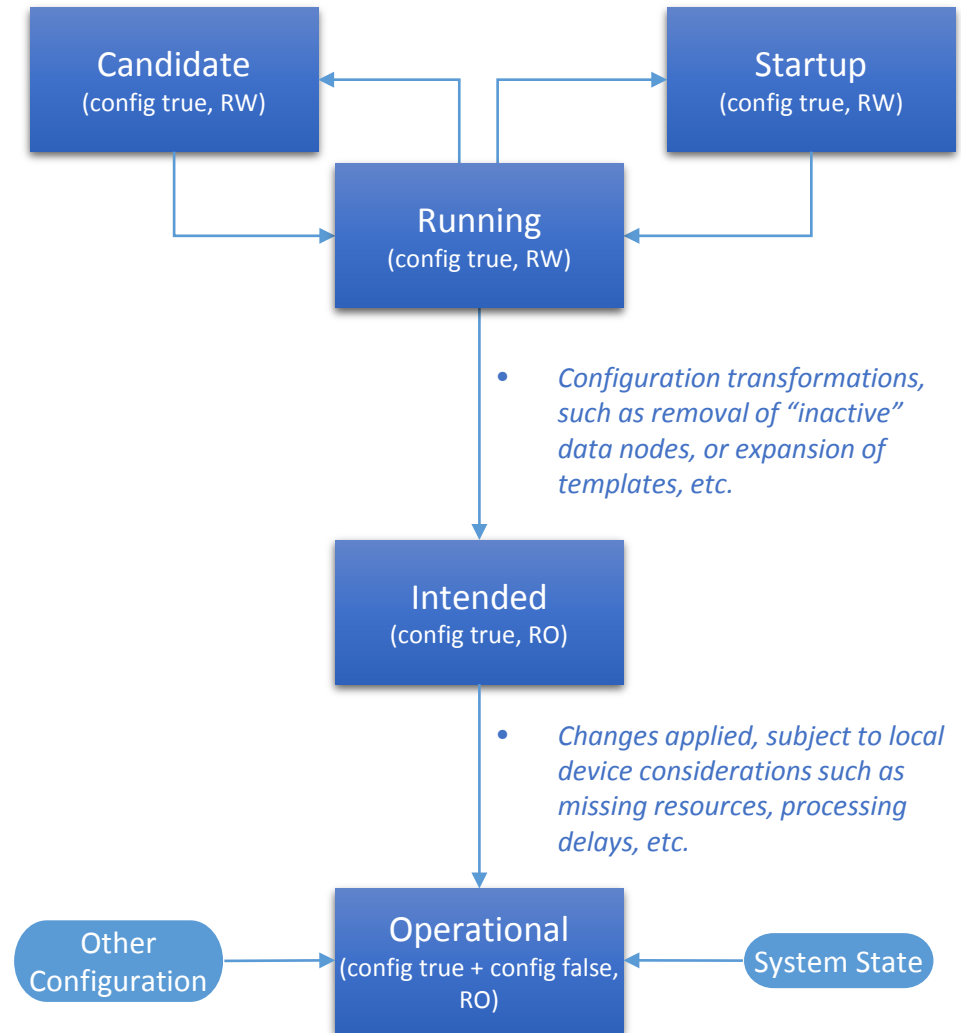
- The style of the YANG model influences how *intended* configuration versus *applied* configuration can be supported
- IETF (Network Management Datastore Architecture WG) is promoting one style (as described in <https://datatracker.ietf.org/doc/draft-ietf-netmod-revised-datastores/>), while OpenConfig is promoting a different style (as described in <https://datatracker.ietf.org/doc/draft-openconfig-netmod-opstate/> and https://github.com/openconfig/public/blob/master/doc/openconfig_style_guide.md)

IETF NMDA YANG Modeling Structure/Style Overview

IETF NMDA Architectural Model of Datastores

EVOLUTION

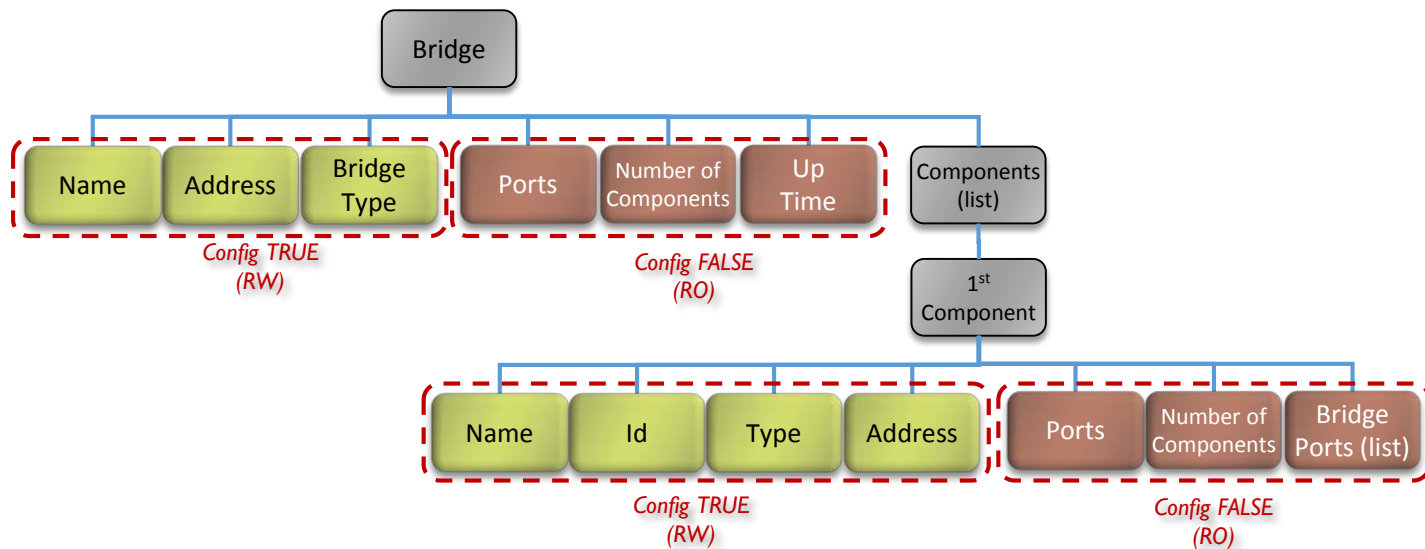
- A datastore is a conceptual place to store and access information
- A datastore maps to an instantiated YANG data tree
- Datastores bind the data models written in the YANG data modeling language to network management protocols such as NETCONF and RESTCONF



IEEE/IETF YANG Model Style

EXAMPLE YANG MODEL

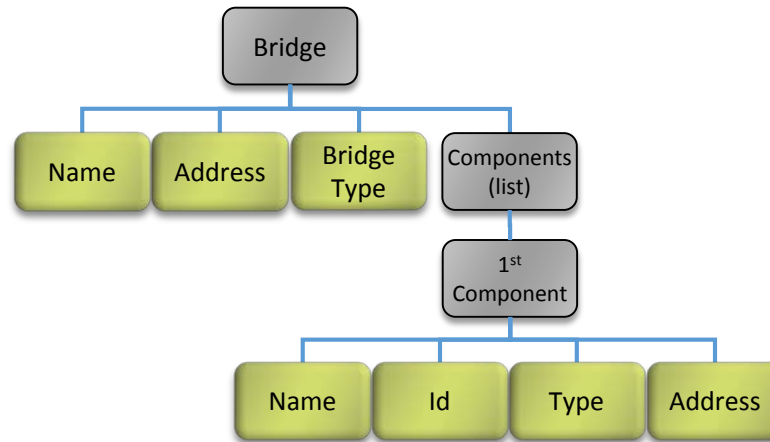
- Consider (a segment of) the 802.1Qcp Bridge YANG model



IEEE/IETF YANG Model Style

DATASTORES

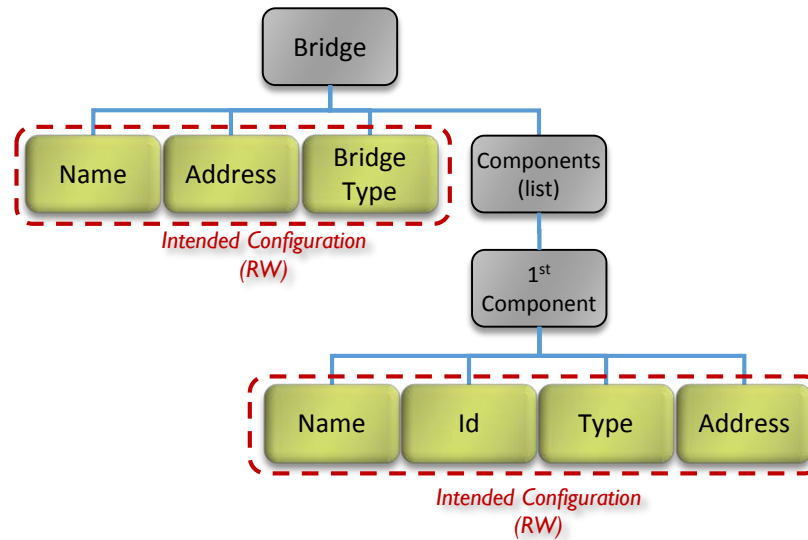
- Example 802.1Qcp **STARTUP** datastore



IEEE/IETF YANG Model Style

DATASTORES

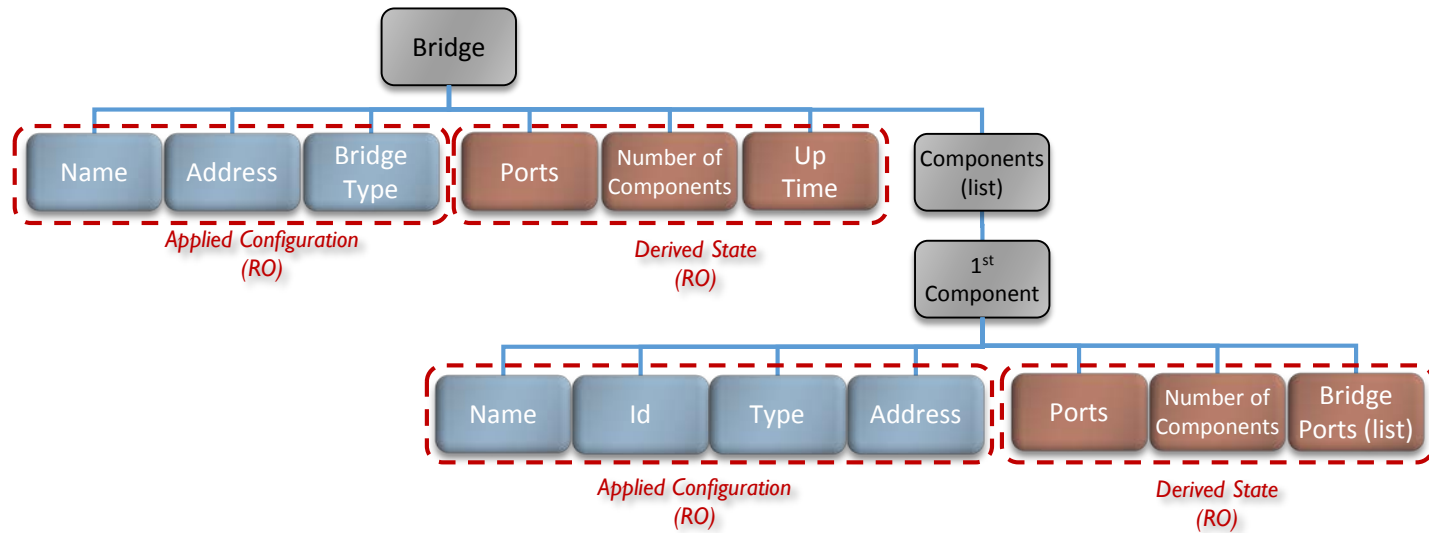
- Example 802.1Qcp **RUNNING/INTENDED** datastore



IEEE/IETF YANG Model Style

DATASTORES

- Example 802.1Qcp **OPERATIONAL** datastore



IEEE/IETF YANG Model Style

- In a **synchronous** configuration management environment

“... A configuration request to update the running configuration of a server that is applied synchronously with respect to the client request (i.e. a blocking call). The server MUST fully attempt to apply the configuration change to all impacted components in the server, updating both the server’s intended and applied configuration, before replying to the client. The reply to the client indicates whether there are any errors in the request or errors from applying the configuration change ...”

- The *intended* configuration is the same as the *applied* configuration

- However, in an **asynchronous** configuration management environment

“... A configuration request to update the running configuration of a server that is applied asynchronously with respect to the client request. The server MUST update its intended configuration before replying to the client indicating whether the request will be processed. This reply to the client only indicates whether there are any errors in the original request. The server’s applied configuration state is updated after the configuration change has been fully effected to all impacted components in the server ...”

- The intended configuration may not be the same as the applied configuration at a given point in time



IEEE/IETF YANG Model Style

ASYNCHRONOUS ENVIRONMENT

- In an **asynchronous** configuration management environment,
 - The NMS can access the **operational** datastore information and compare to the **intended** datastore to determine if the *intended* configuration has been applied, or
 - The NMS can be notified by the device when the **intended** datastore configuration matches that of the **operational** (*applied*) datastore configuration

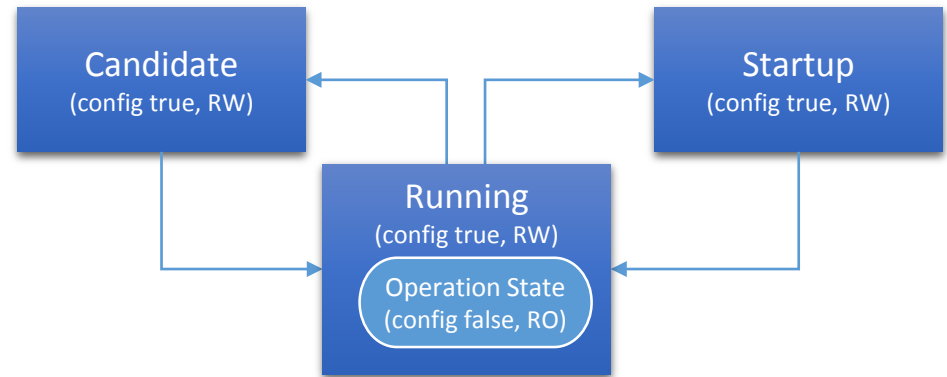
OpenConfig YANG Model Structure/Style Overview

OpenConfig YANG Model Style

- OpenConfig styled YANG models do not make any assumptions about configuration management protocol being used between the NMS and device (i.e., does not assume NETCONF or RESTCONF)
- Models following this style includes the *intended* and *applied* configuration, along with the *desired* state into the same model, and thus same datastore

Architectural Model of Datastores (NETCONF)

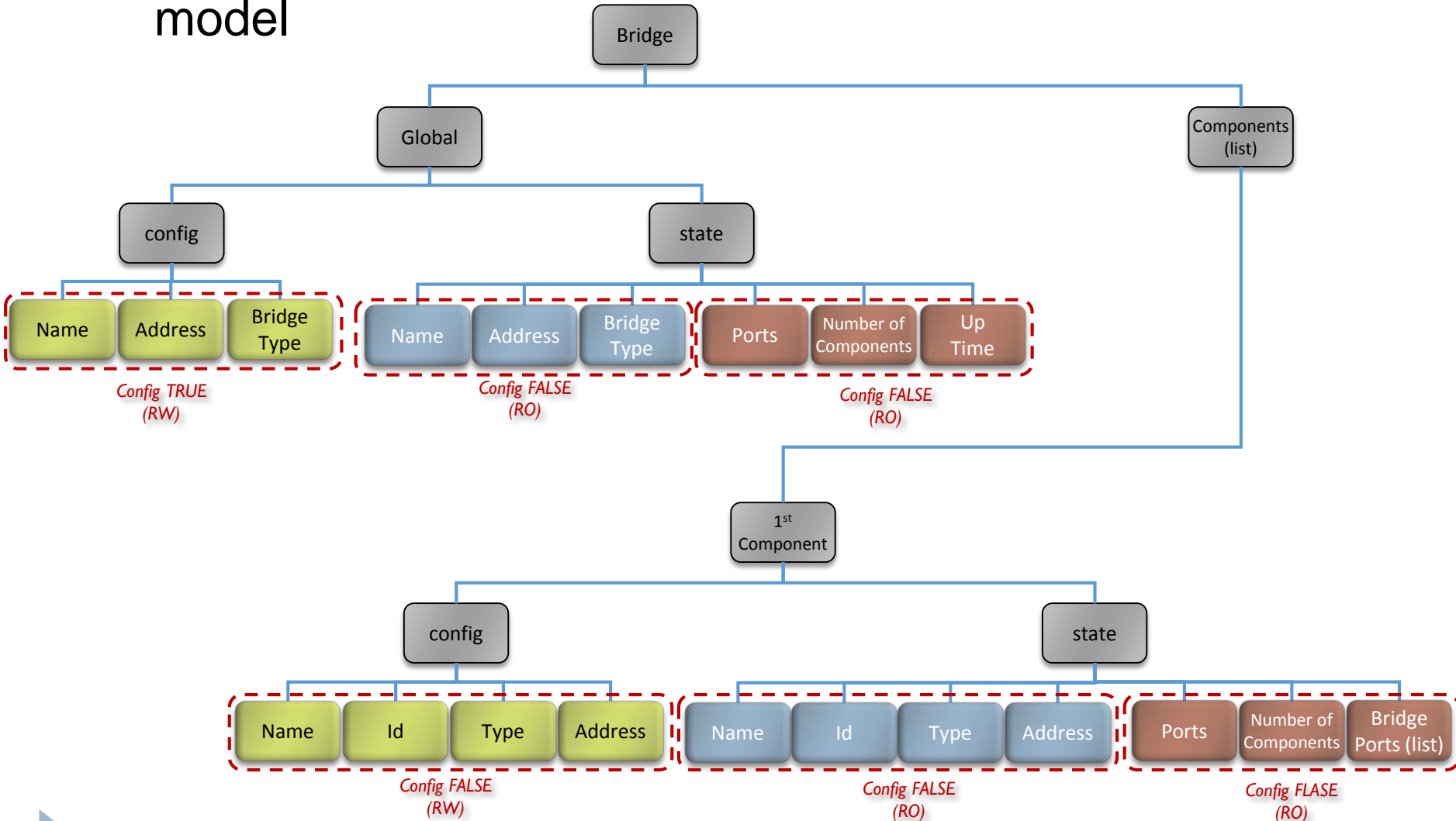
- A datastore is a conceptual place to store and access information
- A datastore maps to an instantiated YANG data tree
- Datastores bind the data models written in the YANG data modeling language to network management protocols such as NETCONF and RESTCONF



OpenConfig YANG Model Style

EXAMPLE YANG MODEL

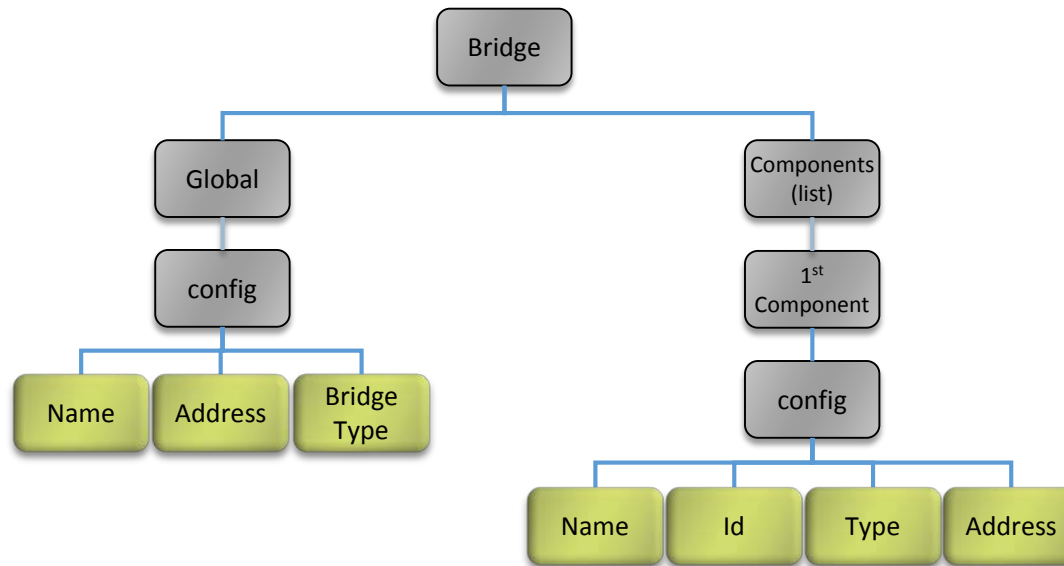
- Consider (a segment of) the 802.1Qcp Bridge YANG model



OpenConfig YANG Model Style

DATASTORES

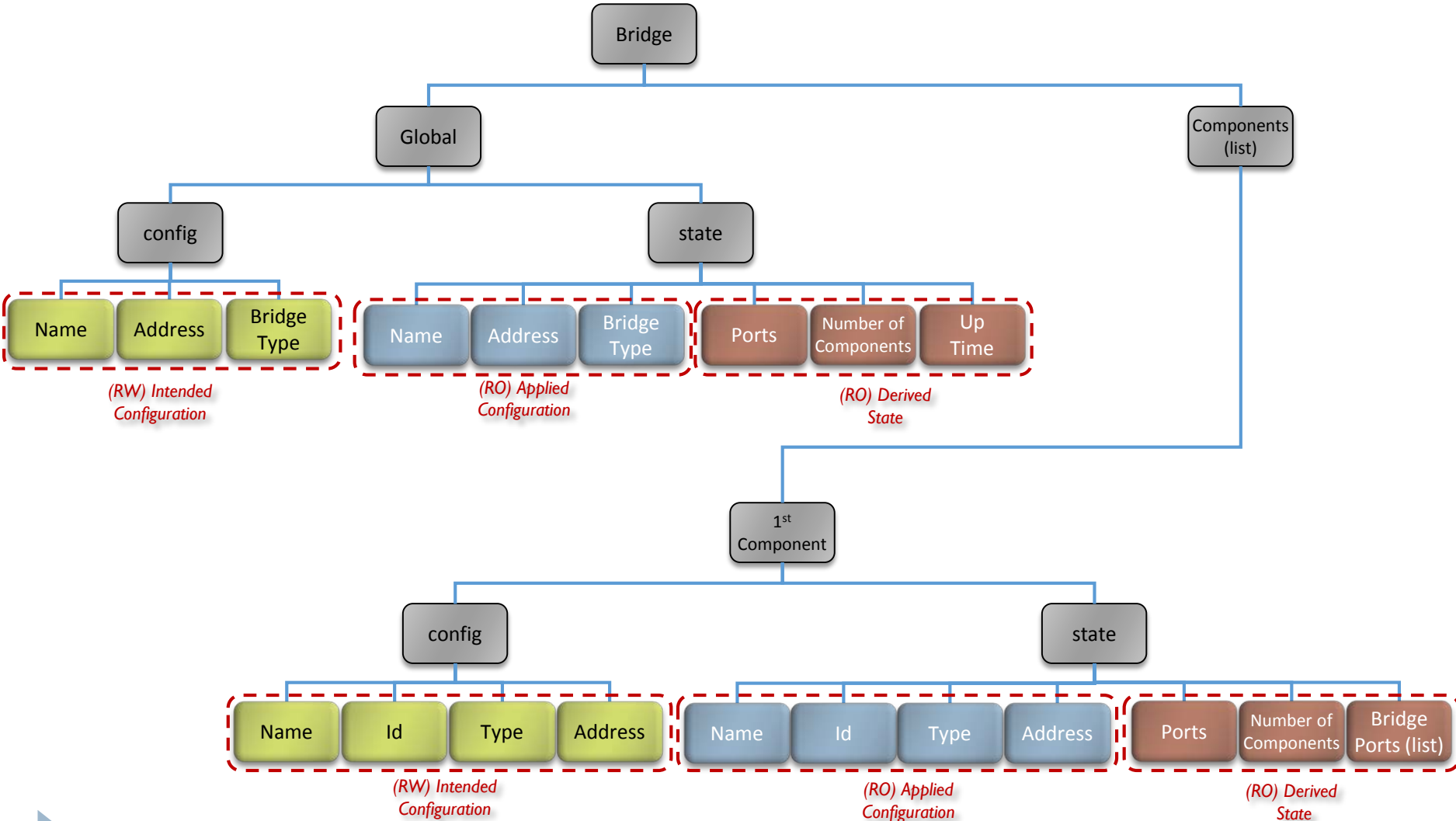
- Example 802.1Qcp **STARTUP** datastore



IEEE/IETF YANG Model Style

DATASTORES

- Example 802.1Qcp **RUNNING** datastore



OpenConfig YANG Model Style

- Both “synchronous” and “asynchronous” configuration management environments can be supported
 - The server (i.e., device) needs to adjust the *applied* configuration elements appropriately
 - The NMS can access the **running** datastore information and compare to the *intended* configuration with the *applied* configuration
 - Alternatively, the NMS can be notified by the device (i.e., server) when the *intended* configuration matches that of the *applied* configuration

Summary

Summary

- The OpenConfig YANG modeling style
 - Is decoupled from NETCONF or RESTCONF as the exclusive configuration management protocol between the client and server to support *intended* and *applied* configurations
 - Can support *intended* and *applied* configuration (and *derived* state) today using current version of NETCONF
 - The server (i.e., device) would need to support the *intended* and *applied* configuration concepts however
 - Is difficult to read primarily due to extensive usage of group statements (e.g., used to address duplication of data nodes within the *intended* configuration and *applied* configuration branches of the YANG tree)
 - Is a coding/convention based scheme, and relies upon all YANG authors getting the convention right
 - Some YANG statements should be avoided (e.g., choice, if-feature, presence, etc.)

Summary

- The IETF NMDA YANG modeling style
 - Results in a simpler YANG module to read and [arguably] maintain
 - Relies upon NETCONF/RESTCONF protocol enhancements to cleanly support the notion of *intended* and *applied* configuration (and *derived* state)
 - The datastores (e.g., **intended/running** and **operational**) provide a clean separation of function
 - Supports the full compliment of YANG statements

IEEE YANG Modeling Structure/Style

- IEEE 802.1 YANG modeling structure/style direction are currently aligning with IETF NMDA