Operational Structure and Organization of YANG Models
draft-openconfig-netmod-model-structure

June 2015 NETMOD interim
OpenConfig network operators group
Anees Shaikh (Google), Rob Shakir (BT), Kevin D'Souza (AT&T), Luyuan Fang (Microsoft)

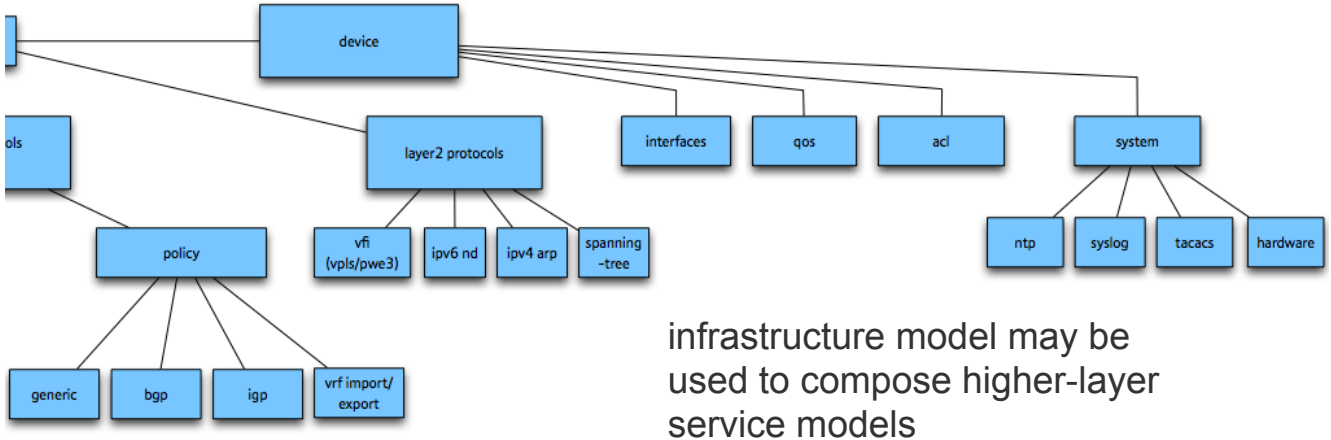
Recap of draft motivations

- YANG models mostly being developed individually and in isolation
  - some narrow efforts to reconcile / relate models, e.g., routing, interfaces
  - no well-defined granularity of individual modules / models
- Managing devices and services requires a multitude of models
  - models must be structured coherently and work together to be useful
  - models often must reference each other -- where are they in the tree?
  - difficult to define higher layer services that use an arbitrary collection of individual models
- Need for a ‘service catalog’ approach for managing models
  - many different individuals, SDOs, and OSS projects creating models
  - which models are supported (or not) on various platforms
  - where to find authoritative source, license terms, dependencies, ...

Proposed approach

Develop device-level structure to organize infrastructure models in a predictable way

- devices, their subsystems, and relevant protocols operating at the link and network layers
- similar problem / approach applies at multiple levels
- consider ‘push’ and ‘pull’ approaches for building model structure



infrastructure model may be used to compose higher-layer service models

Questions / issues

- Why do we need a top-level device container? Why not have multi-rooted collection?
  - model roots are arbitrarily chosen, making it hard to program across them
  - need an anchor point for further composition (e.g., devices in PoP)
  - device is a natural place for general information (e.g., device name, location), hardware inventory info, operator-specific data about a device, etc.
- What is the implication of having all these NP containers?
  - no implication -- published YANG model is simply an illustration of proposed structure to compose models

Further development of proposed structure

- IETF routing design team charter includes discussion and refinement of the model structure (‘meta-model’)
- Ongoing discussions in DT meetings to generalize structure (logical elements, routing instances, VRFs, etc.)

Meta-Model Status Update
Routing Area Yang Architecture Design Team

Members: Acee Lindem, Anees Shaikh, Christian Hopps, Dean Bogdanovic,
Lou Berger, Qin Wu, Rob Shakir, Stephane Litkowski, Yan Gang

Wiki: http://trac.tools.ietf.org/area/rtg/trac/wiki/RtgYangArchDT
Repo: https://github.com/ietf-rtg-area-yang-arch-dt/meta-model

Design Team Meta-Model Structure

- Work in Progress
- Many ways one could organize
- Driving towards reaching consensus among design team - needs more vetting and discussion
- Two level of hierarchy below the device level
  - logical-network-element - (think virtual router, virtual-device-context, etc)
  - networking-instance - (think VRF, routing-instance, or VPN Context). Didn’t call it routing-instance to allow for layer-2 definitions.
  - Today’s models are mostly at the top or routing instance level.

Design Team Meta-Model Structure (continued)

- Interfaces Configured/Managed as silos - consistent with RFC 7233 and RFC 7277.
  - Operational Preference
  - Interfaces bound to logical-networking-elements
  - IPv4/IPv6 Configuration bound to networking-instance
  - Details to be worked out - not necessary for model to enforce all structure
  - May be side effects of moving interfaces/IP interface configuration among logical-network-elements and networking-instances.
- Management at logical-network-element level with some instances able to see/manage the whole device.
  - networking-instance for management specified at logical-network-element level.

Design Team Meta-Model Structure (continued)

- Policies at the networking-instance level
  - Exceptions are ACL and key-chain - since they are not necessarily bound to an IP/IPv6 address space
- Protocol organization largely unchanged from original OpenConfig draft.
  - Flatten somewhat as this aligns with the existing VRF-centric protocol models.

Design Team Model Open Issues

- Protocol specific policies have been moved under the protocols (e. g., BGP). Thoughts?
- Do we need specific RIB policy or is there only RIB client policy? One requirement could be FIB installation policy. We are assuming RIB client policy is in the client.
- The argument as to whether various models should be rooted at the logical-network-instance or networking-instance is an interesting one. For example, key-chain and network-services. Thoughts?
- Need to verify interface binding to logical-network-elements and networking-instances will work. Use keys or references - right now we're using keys.

Design Team Model Open Issues (continued)

- Need to revisit VRF policy definition and relationship to L3VPN Config/Policy.
- This model may not support the zone-based policy firewall - TBD to figure this out.
- Need to describe how logical-network-instance administrators can access all interfaces assigned to logical-network-instance.
  - An administrator with the global-view must assign the respective interfaces to the logical-network-instance.
  - Systems that only support allocation of interfaces to VRFs (aka, networking-instances) will require assignment to specific VRFs rather than logical-networking-elements.
- YANG Model corresponding to tree needed.

Design Team Model Next Steps

- Finalize and document meta model
- Finalize Operational State recommendation
- Revisit YANG recommendations
- Revisit YANG model conventions