Enabling the zero touch network - programmatic management plane

Ina Minei
on behalf of Google Technical Infrastructure

3/1/2016


For the past **15 years**, Google has been building out the largest cloud infrastructure **on the planet.**
Images by Connie Zhou


In the process, we have built a global software defined network Infrastructure
Google | google.com/datacenters


And scaled it!

Google Backbone(s)

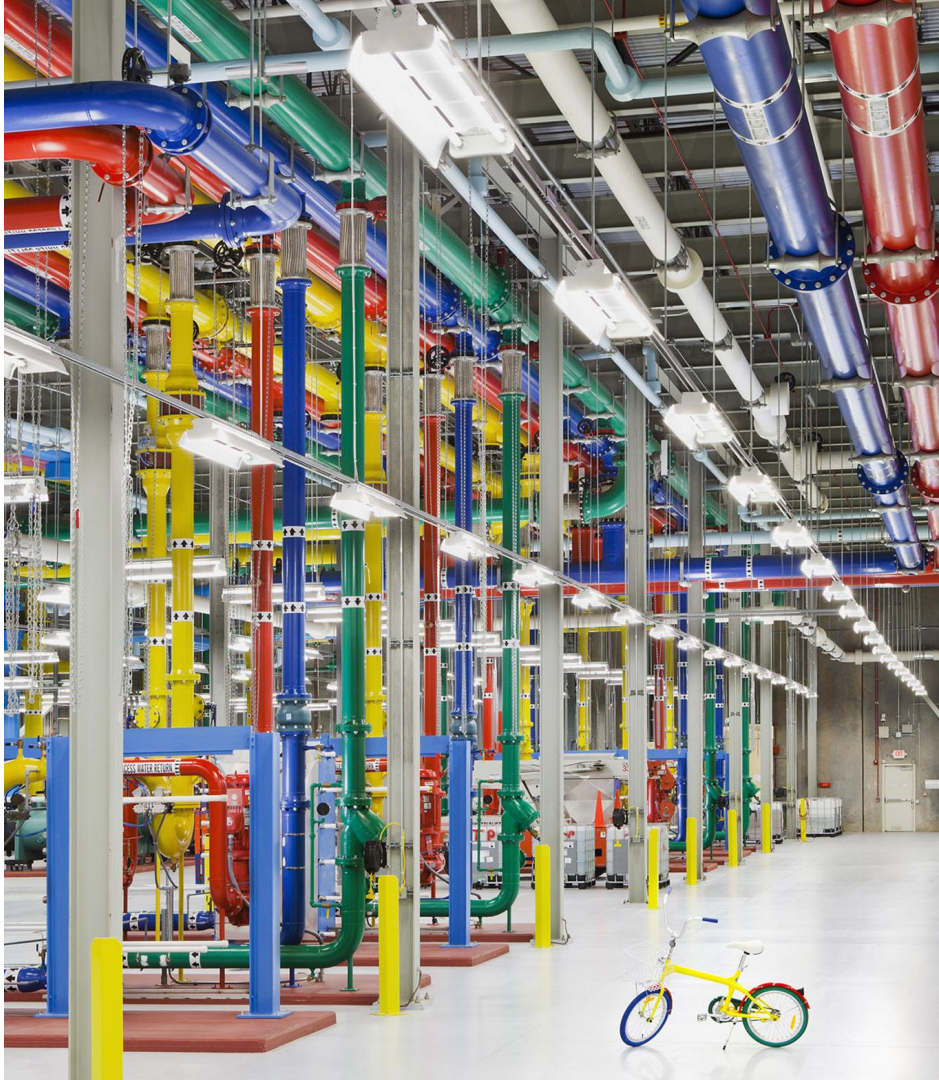
Internet facing Backbone, B2:
70+ locations in 33 countries

Global Software Defined Inter-DC Backbone: B4

Sigcomm papers:
B4
Datacenter - clos topologies and centralized control
Bandwidth enforcement

Plumbing underneath

- tens of network device roles
- more than half dozen vendors, multiple platforms
- many protocols
- many tools, and multiple generations of software



Operational scale

- 4M lines of configuration files
- up to ~30K configuration changes per month
- more than 8M OIDs collected every 5 minutes
- more than 20K CLI commands issued and scraped every 5 minutes

The show must go on...

Cannot limit the rate of change in the network

- Expensive - capacity has to be deployed significantly ahead of demand
- Ineffective- application needs, the network must provide the features required
- Brittle - limiting the number of change operations makes the network more brittle, not more robust as it fails to exercise these code paths

The zero touch network

Successful automation - the zero touch network

- Operator-independent - all network operations are automated, requiring no operator steps beyond the instantiation of high-level intent
- Safe to fail
- Constrained in scope

Operator independent

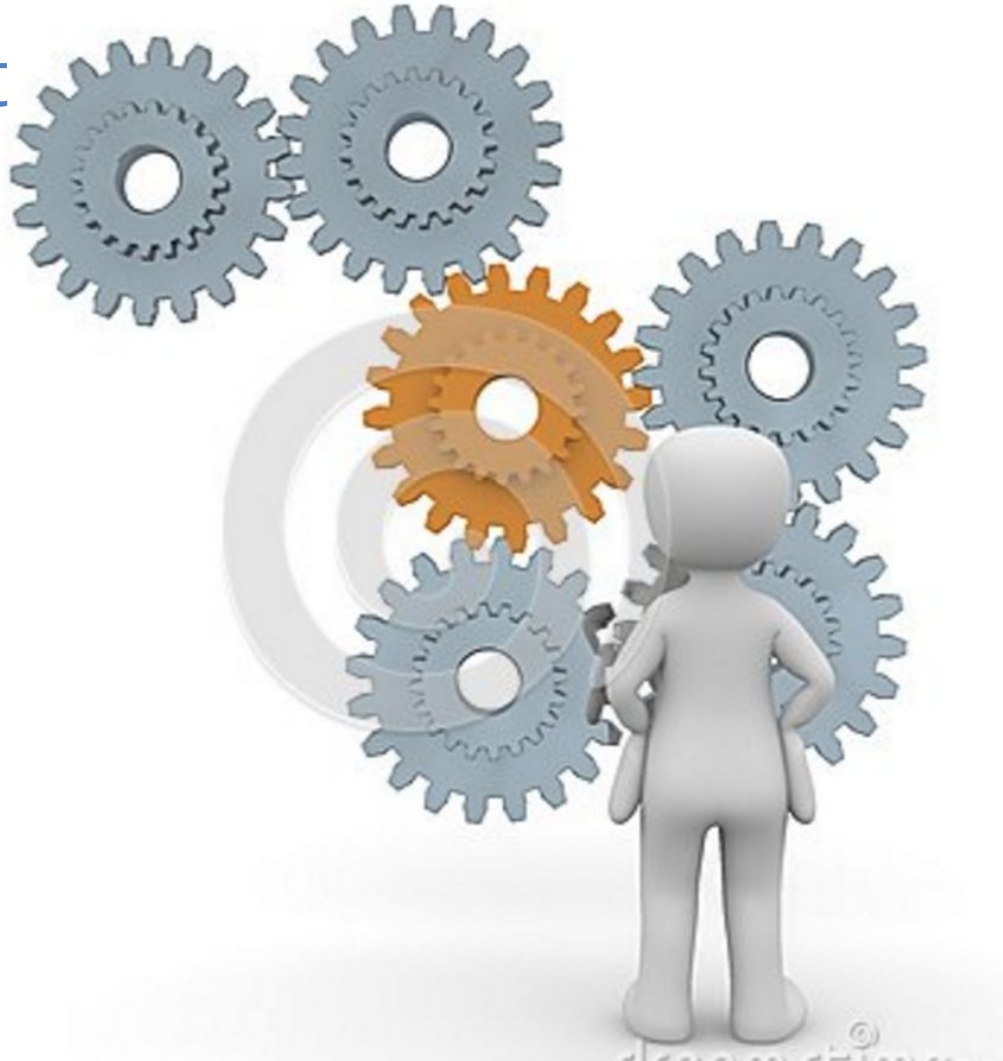
Taking the humans out of the loop - they open the opportunity for errors:

- Human-edited and reviewed goals, but not human-entered data
- Every cut and paste is an opportunity for error

Operator independent

Taking the humans out of the loop - they open the opportunity for delays:

- No hand-offs between humans
- Operational efficiency by concurrent workflows



Expressing intent

- Complete, explicit, and machine-readable expressions of goals
- Express intent at the network-level, not at individual device level
- Configuration changes driven by updates to the intent

What is needed for expressing intent

- Network topology
  - A logical topology model
- Validation rules - intent checking
  - Design rules - what has to hold true at various layers of the network
  - Network policy - what packets can/can't use which paths
- Inputs - abstractions
  - Configuration intent - high level config intent and device level abstractions
  - Topology intent
- Business logic: how to combine topology+rules+inputs to create config
  - code co-written by software engineers and network engineers

Successful automation - the zero touch network

Google

- Operator independent - All network operations are automated, requiring no operator steps beyond the instantiation of high-level intent
- Safe to fail - Any network changes are automatically halted and rolled-back if the network displays unintended behavior
- Constrained in scope

Safe to fail

- Ability to automatically check the safety of operations
- Ability to repeatedly validate the network state against the stated intent
- Ability to recognize “bad” network behavior
- Ability to roll back to the original state

Successful automation - the zero touch network

Google

- Operator independent - all network operations are automated, requiring no operator steps beyond the instantiation of high-level intent
- Safe to fail - any network changes are automatically halted and rolled-back if the network displays unintended behavior
- Constrained in scope - the infrastructure does not allow operations which violate network policies

Constrained in scope

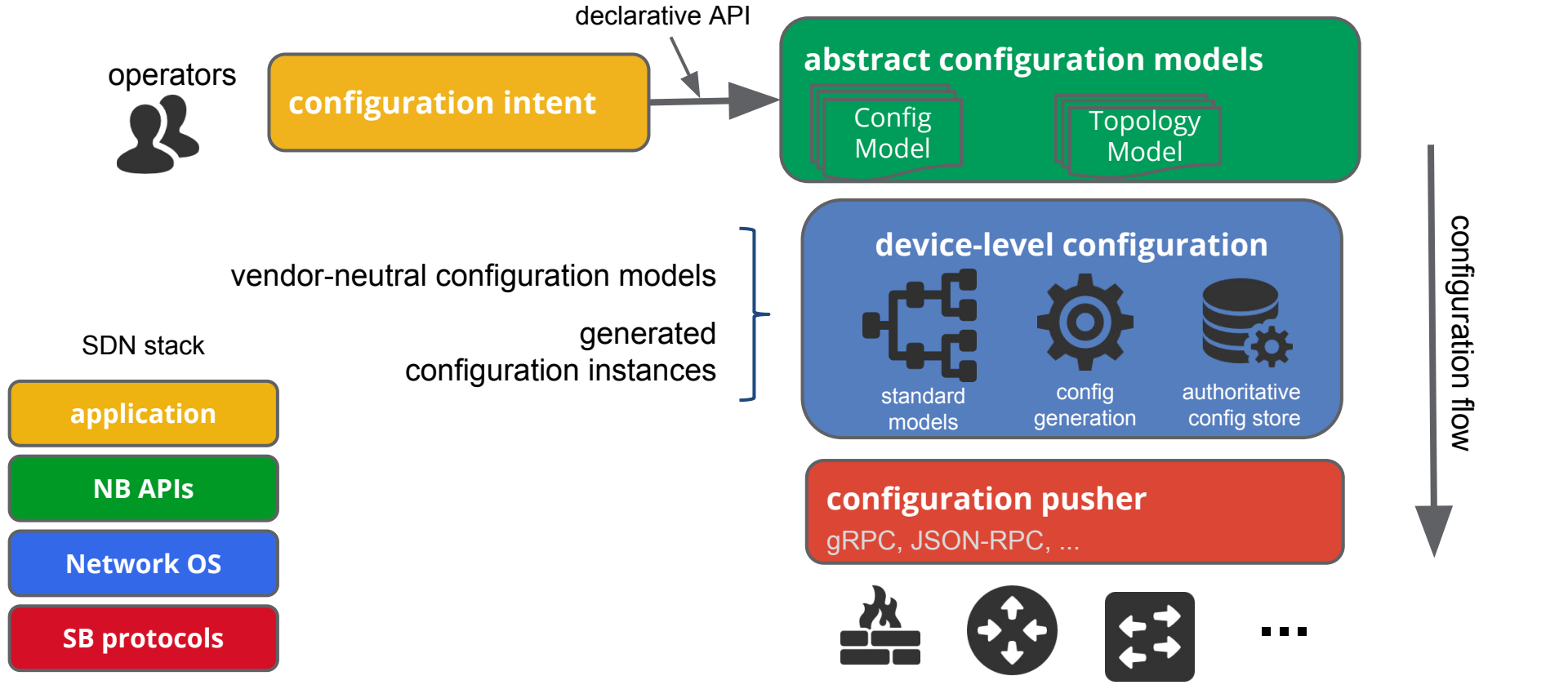
- Staged deployment of large changes
- Concurrency control - prevent clash of concurrently running workflows
- Minimum survivable topology - prevent operations which would violate it
- Policy based blast radius containment - granular authorization checks
- Each change logged and traceable to source

Towards a programmable management plane

SDN applied to configuration management

SDN principle	Applied to configuration
separation of control and data	authoritative configuration outside of devices; scraped config is not authoritative !
centralized control	network-wide, coordinated, configuration and visibility
network abstractions and APIs	knowledge in data models, pub/sub API for obtaining network state
standard protocols and interop	adoption of standard configuration data models is crucial to enable vendor-neutral configuration

Intent-based configuration flow

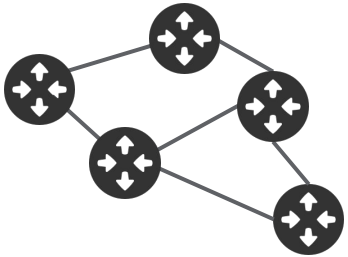


Model-driven network management

Google

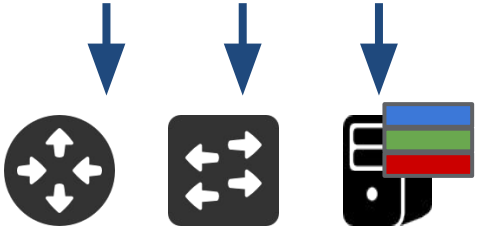
Topology

- describes structure of the network
- common modeling language: ?
- data encoding: protobuf, ...



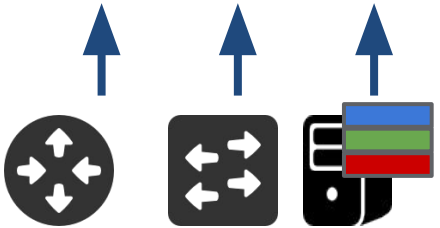
Configuration

- describes configuration data structure and content
- common modeling language: YANG
- multiple data encodings: protobuf, XML, JSON, ...



Telemetry

- describes monitoring data structure and attributes
- common modeling language: YANG
- data encoding: JSON, protobuf, ...



Configuration independence through models

Google

- Software today often uses a hardware abstraction layer (HAL) to simplify development against changing hardware constructs
- Google's modeling goal is to create a CAL (configuration abstraction layer) using models
- Advantages of vendor-independent configuration
  - Increase qualification velocity
  - Decrease operational complexity
  - Refocus staff knowledge on networking, reduce knowledge in vendor CLI languages

Telemetry solutions today

What do we use? Often SNMP is the default choice.

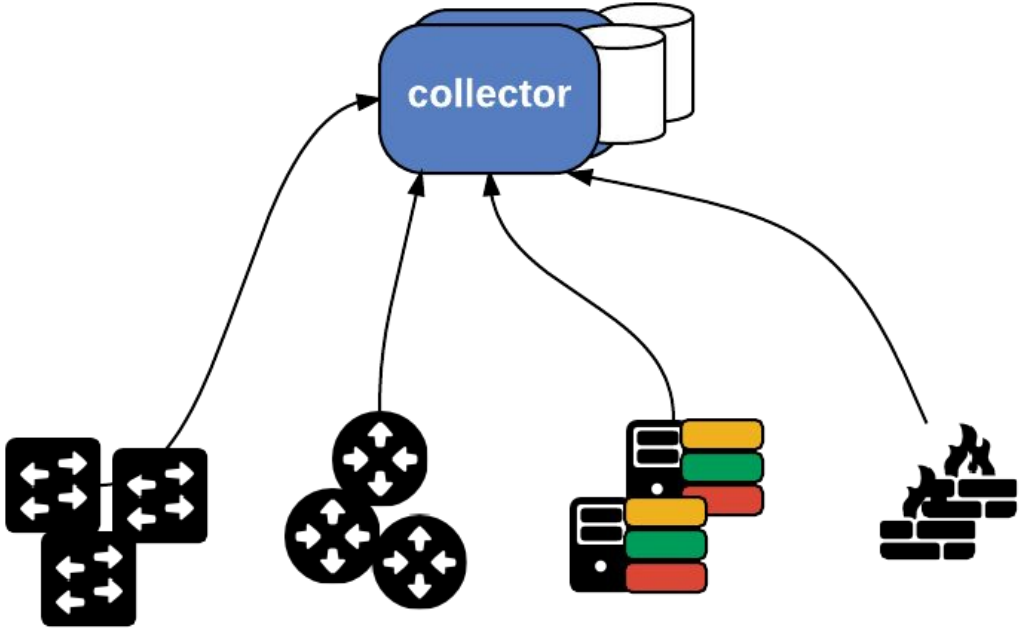
- legacy implementations -- designed for limited processing and bandwidth
- expensive discoverability -- re-walk MIBs to discover new elements
- no capability advertisement -- test OIDs to determine support
- rigid structure -- limited extensibility to add new data
- proprietary data -- require vendor-specific mappings and multiple requests to reassemble data
- protocol stagnation -- no absorption of current data modeling and transmission techniques

Telemetry challenges

- SNMP object collection growing with each platform generation
  - e.g., 100K objects on current platforms, expected to grow 3x over next 2 generations
  - similar for object collection frequency
- Future devices continue to grow in density and drive this trend
  - scale limitations in data acquisition at high frequencies
- Near-real-time acquisition and access to monitoring data is a requirement for <insert buzzword here>
  - traffic management, tight control loops, fast recovery

Streaming telemetry - reverse the flow

network state changes observed by analyzing comprehensive time-series data stream



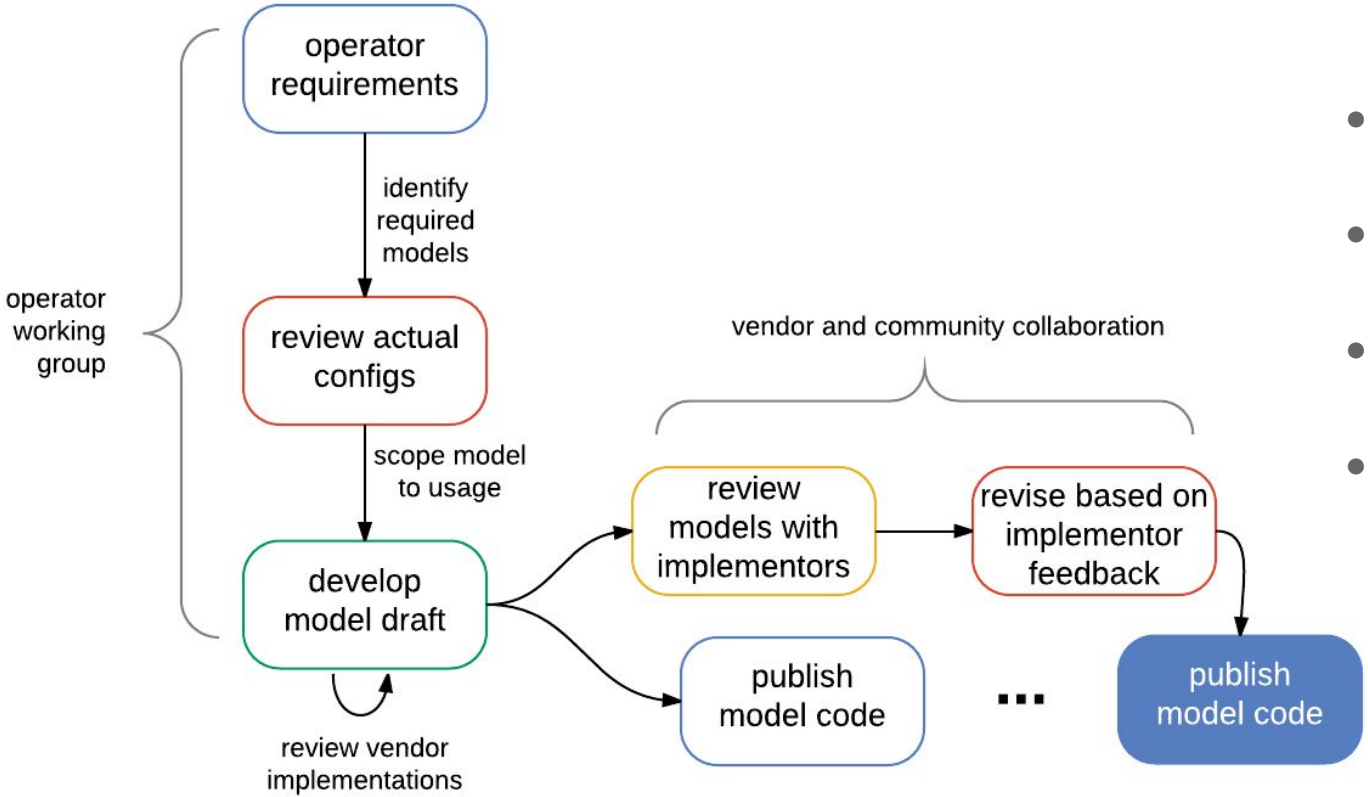
- devices programmed with a data model describing desired structure and content
- stream data continuously -- with incremental updates
- support for efficient, secure transport protocols, e.g. gRPC (http://www.grpc.io/)

Configuration models

OpenConfig

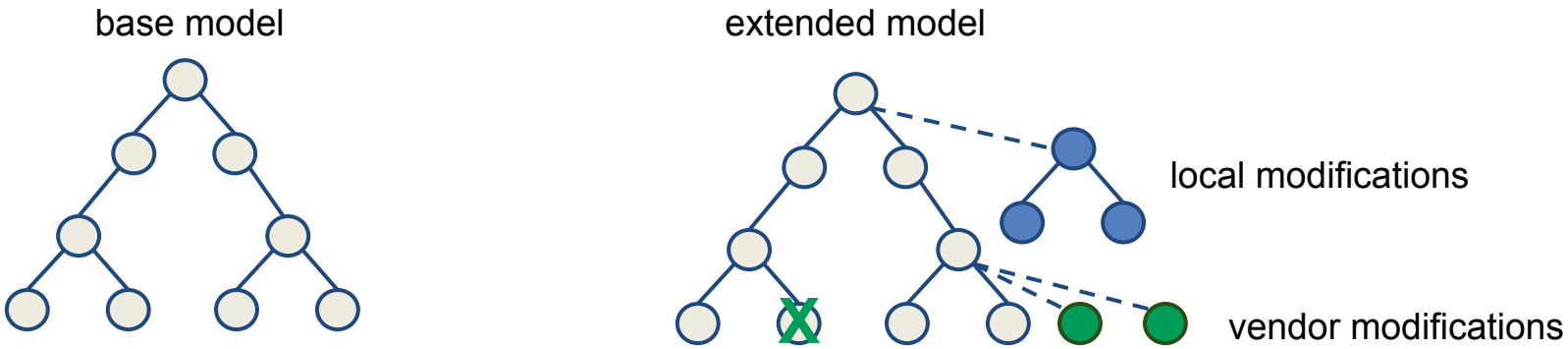
- Informal industry collaboration of network operators
- Focus: define vendor-neutral configuration and operational state models based on real usage
  - Adopted YANG data modeling language (RFC 6020)
- Key Participants: Apple, AT&T, BT, Comcast, Cox, Facebook, Google Level3, Microsoft, Verizon, Yahoo!
- Primary output is model code, published as open source via public github repo : https://github.com/openconfig/
- Ongoing interactions with standards and OSS community (e.g., IETF, ONF, ONOS, ODL)

Current OpenConfig development process



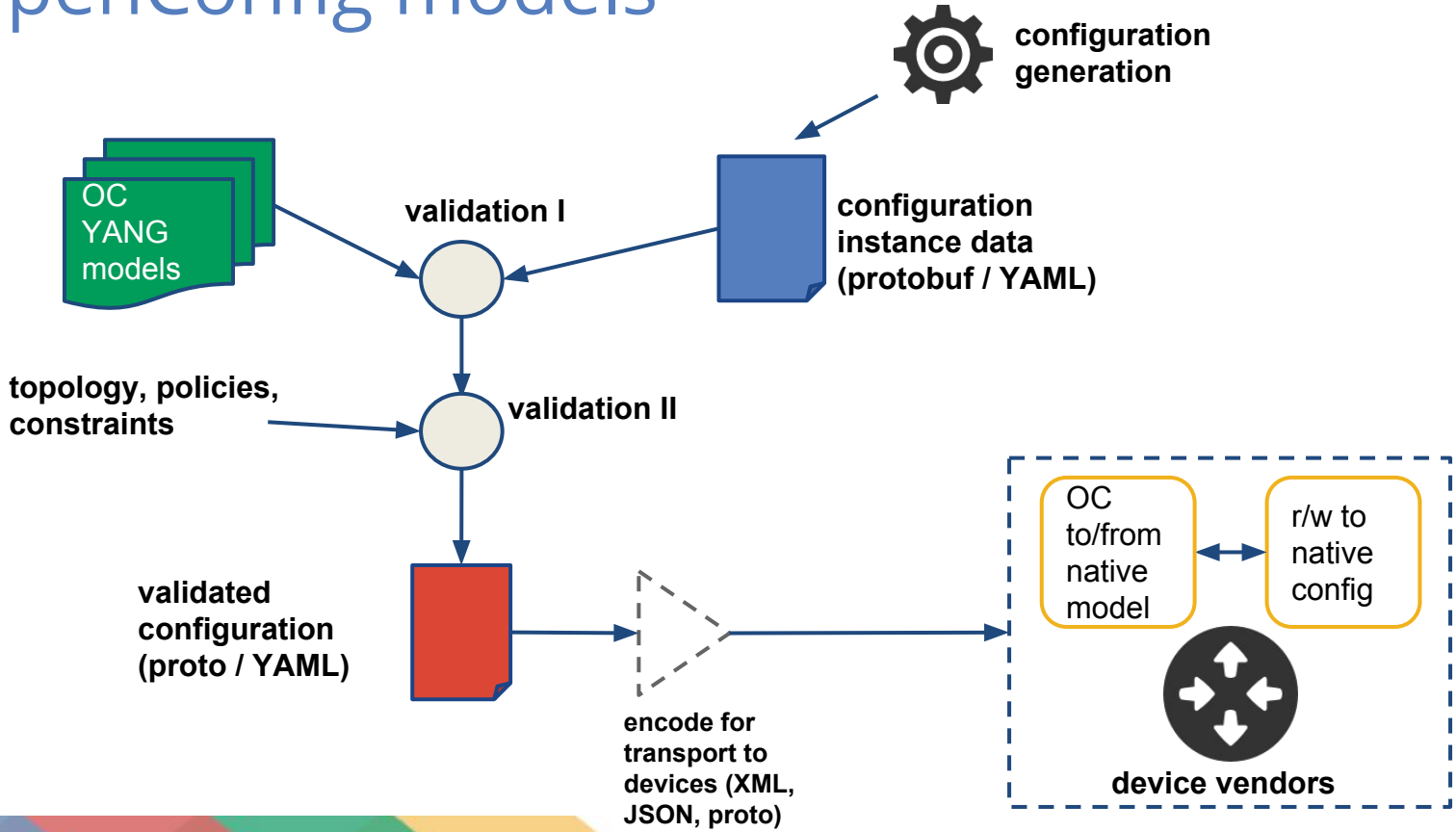
- initial models developed by OpenConfig
- extensive collaboration with vendors
- leverage existing work where possible
- publish models and docs

Extending OpenConfig models



- base OpenConfig model as a starting point
- vendors can offer augmentations / deviations
- operators can add locally consumed modifications

Example configuration pipeline with OpenConfig models



Progress on model development

Published OpenConfig models

- BGP
- routing policy
- locally generated routes
- interfaces, VLANs
- MPLS, RSVP / TE
- networking / forwarding instances, VRFs
- system management
- RIB contents
- terminal optics
- streaming telemetry configuration
- top-level device structure

Models adopted by IETF for standardization

Models in development / review

- system inventory / hardware
- ACLs
- line optics (EDFAs and ROADMs)
- IS-IS
- QoS
- tunnels / encapsulation
- LLDP
- FIB / LFIB

Summary

Research opportunities in the zero touch world

Google

- Intent expression - and intent checking
- Design-rule expression and checking
- Configuration validation
- Impact of operations - anomaly detection - identifying “bad” outcomes vs “normal” outcomes from the state of the network
- Expressing safety rules
- Big data analysis - using streaming telemetry to drive automation (closing the control loop)
- Safety of concurrent workflows

Summary

- Zero touch approach to network automation
- Time for the management plane to join the age of SDN
- Core principles: intent-driven configuration, models for configuration, operation and topology, streaming telemetry
- A lot of open research topics in this space