



Software Defined Networking at Scale

Bikash Koley, Anees Shaikh
on behalf of Google Technical Infrastructure

5/12/2015



Software Defined Networking at Scale

Bikash Koley
on behalf of Google Technical Infrastructure



Software Defined

Networking at Google

Bikash Koley

on behalf of Google Technical Infrastructure



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



In the process, we have
built a global software
defined network
Infrastructure



Software Defined
Networks require
Software Defined
Operations!

Google backbone networks

70+ locations in 33 countries



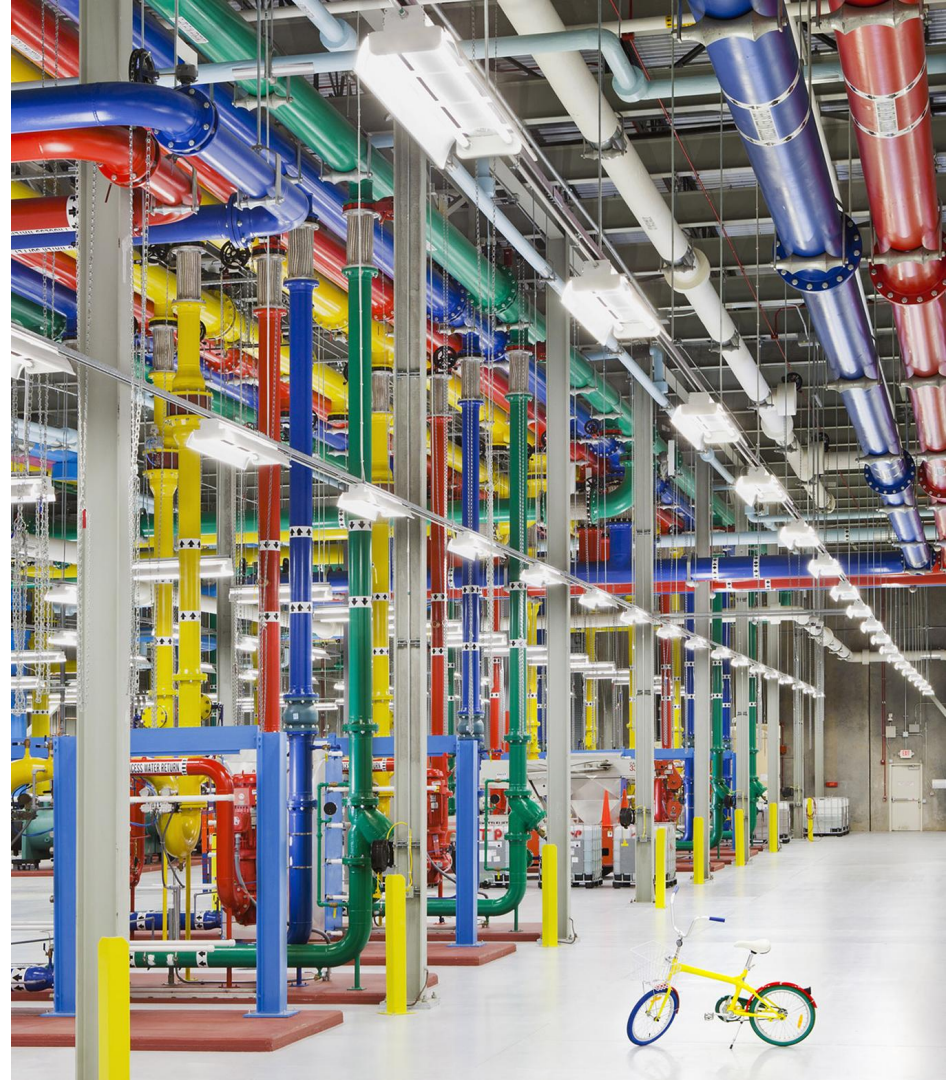
Global Software Defined
Inter-DC Backbone: B4

Google global CDN



A Global SDN Infrastructure

- 50+ network device roles
- 4M lines of configuration files
- more than a dozen vendors, multiple platforms
- ~30K configuration changes per month
- many tools, and multiple generations of software



- SDN \neq Cheap Hardware
- SDN = programmatic decomposition of **control**, **data** and **management** planes
- Well defined APIs \Rightarrow fundamentally easier operational model
- Separation of **control** and **data** planes \Rightarrow much higher uptime
- Network function virtualization \Rightarrow new functions rolled out in days (vs years)

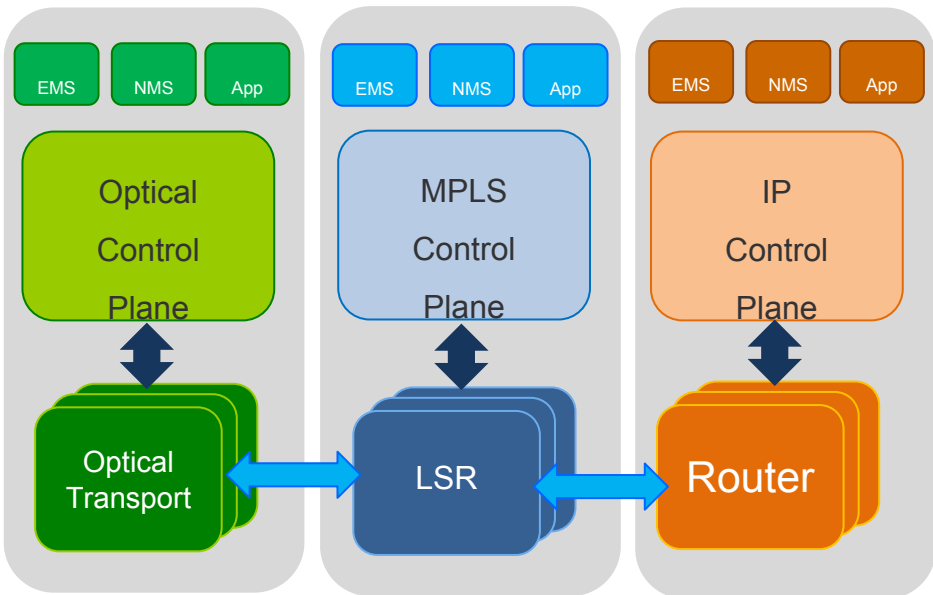
- SDN \neq Cheap Hardware
- SDN = programmatic decomposition of **control, data** and **management** planes
- Well defined APIs \Rightarrow fundamentally easier operational model
- Separation of **control** and **data** planes \Rightarrow much higher uptime
- Network function virtualization \Rightarrow new functions rolled out in days (vs years)

Virtual Network



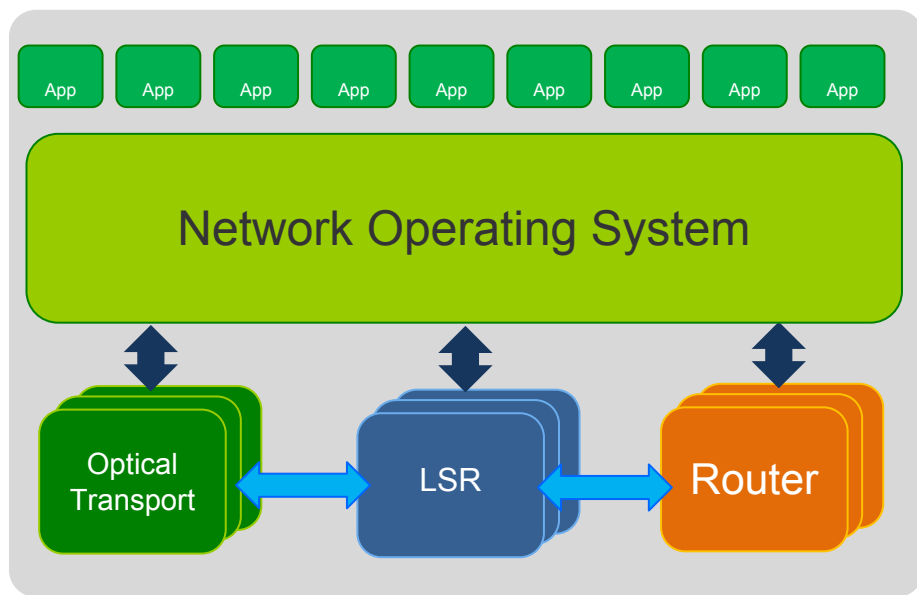
Physical Network

Layer-cake Network

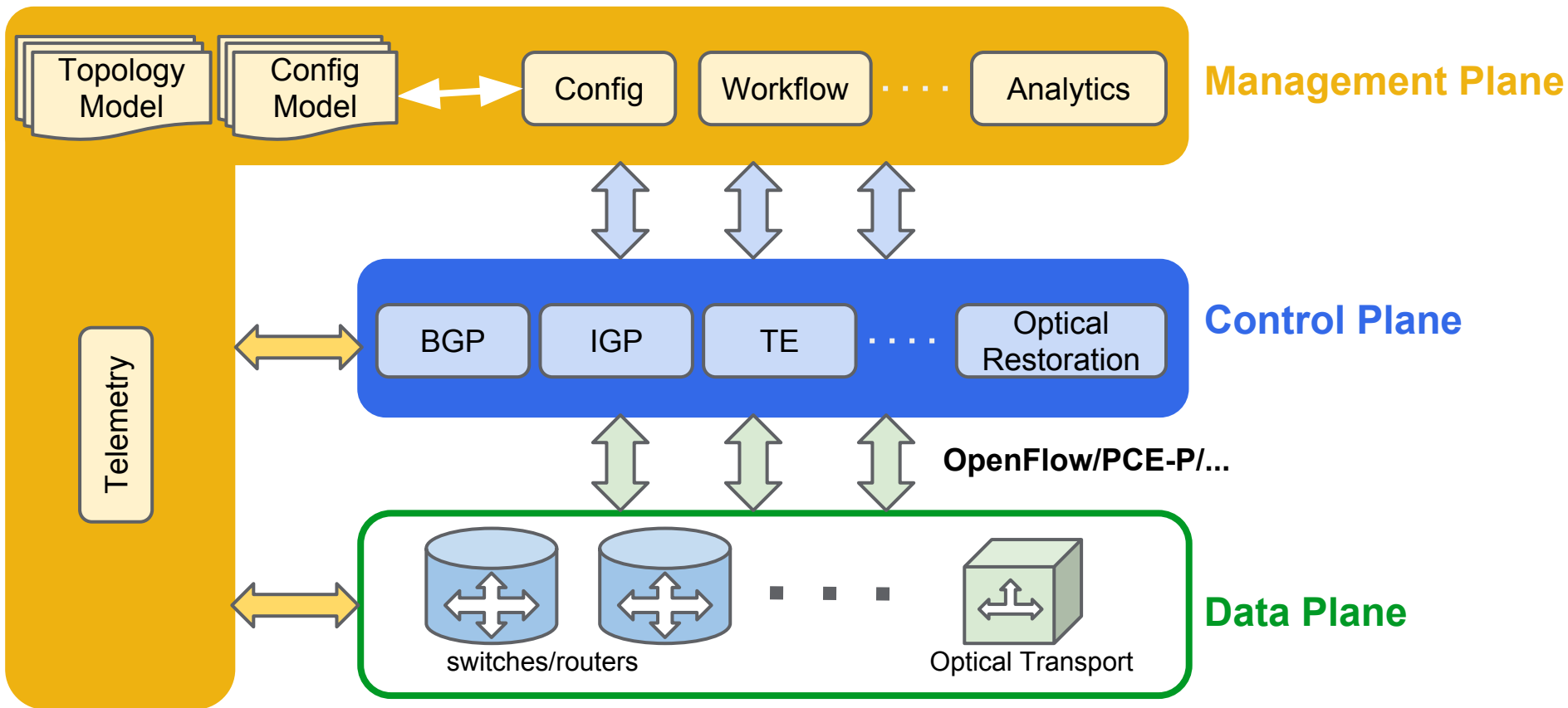


- **Heterogeneous control plane**
- **Heterogeneous network apps**
- **Large inefficiencies**

Software Defined Network



- **Common network OS**
- **Common network apps**
- **Global view of network states**



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**

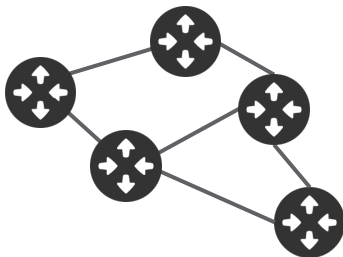
adoption of standard configuration data models is
crucial to enable vendor-neutral configuration



Model-driven network management

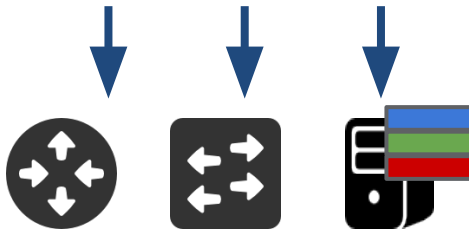
Topology

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



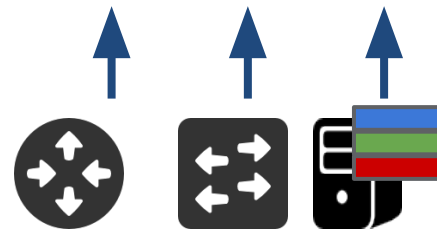
Configuration

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

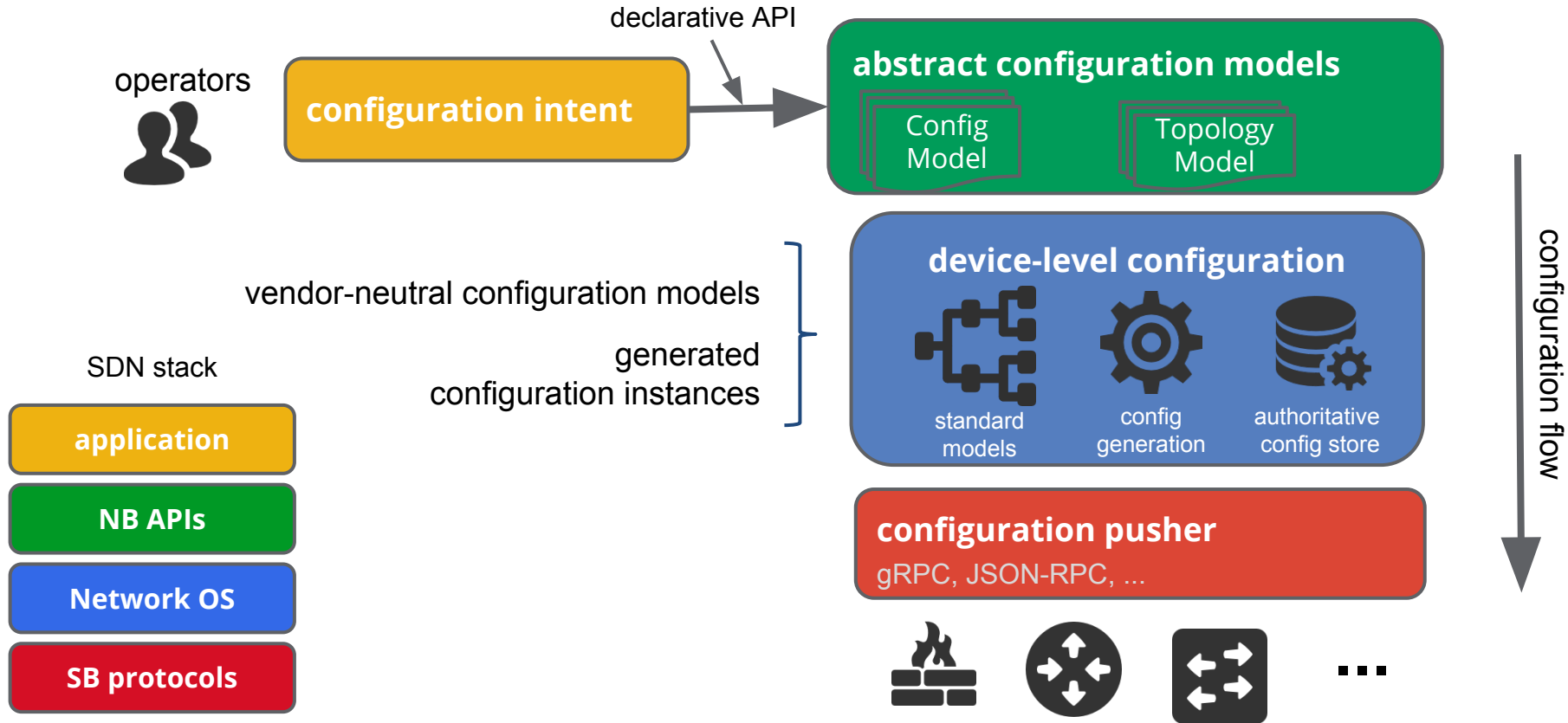


Telemetry

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

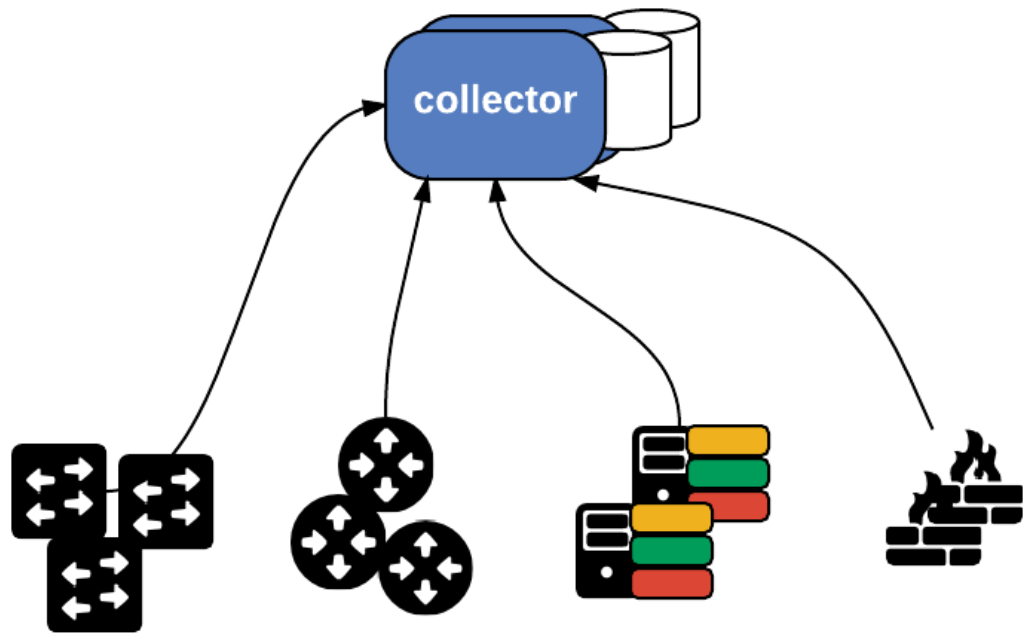


Intent-based configuration flow



Streaming telemetry

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



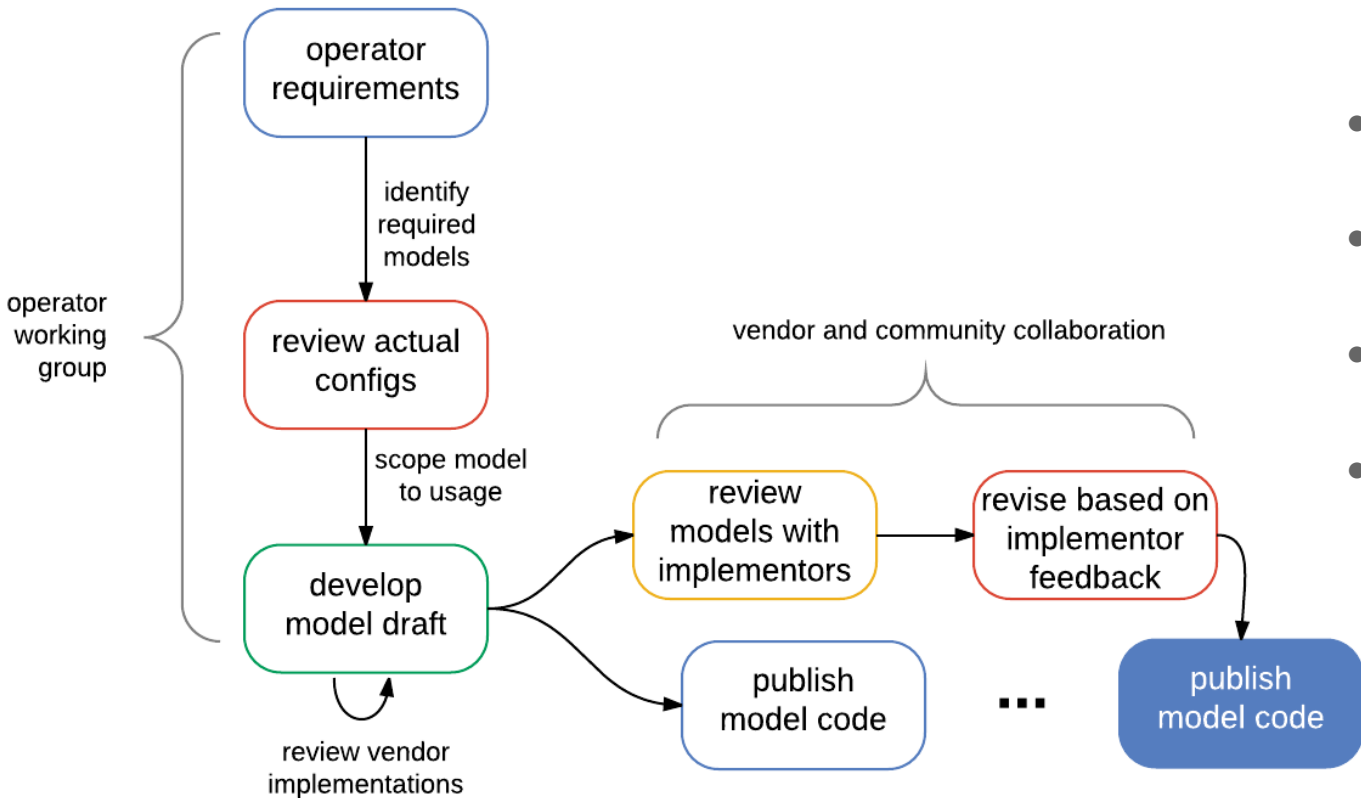
- devices programmed with a data model describing desired structure and content
- stream data continuously -- with incremental updates
- support for efficient, secure transport protocols, e.g. gRPC (<http://www.grpc.io/>)

OpenConfig

OpenConfig

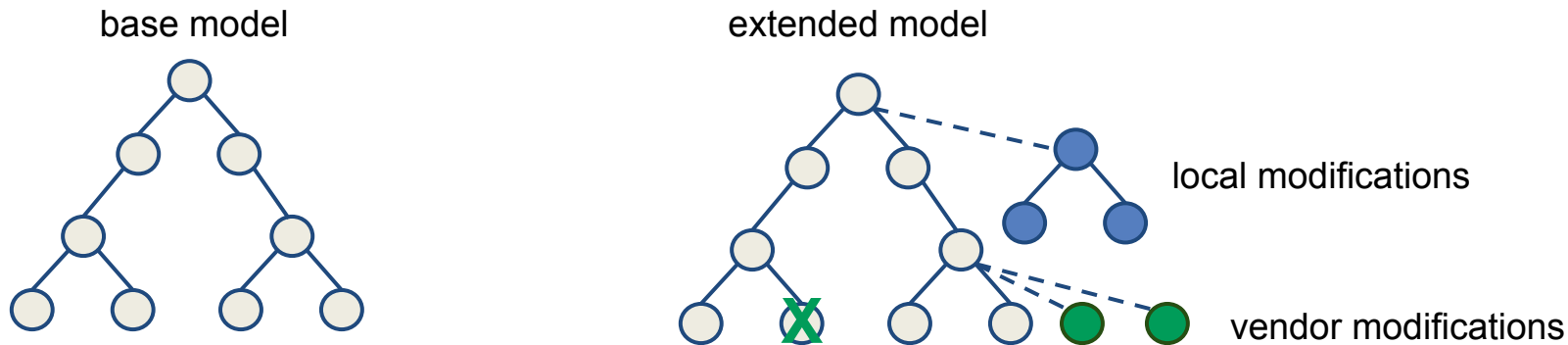
- Informal industry collaboration of network operators
- Focus: define vendor-neutral configuration and operational state models based on real usage
 - Adopted YANG data modeling language (RFC 6020)
- 49 11 key Participants: Apple, AT&T, BT, Comcast, Cox, Facebook, Google Level3, Microsoft, Verizon, 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 “process”



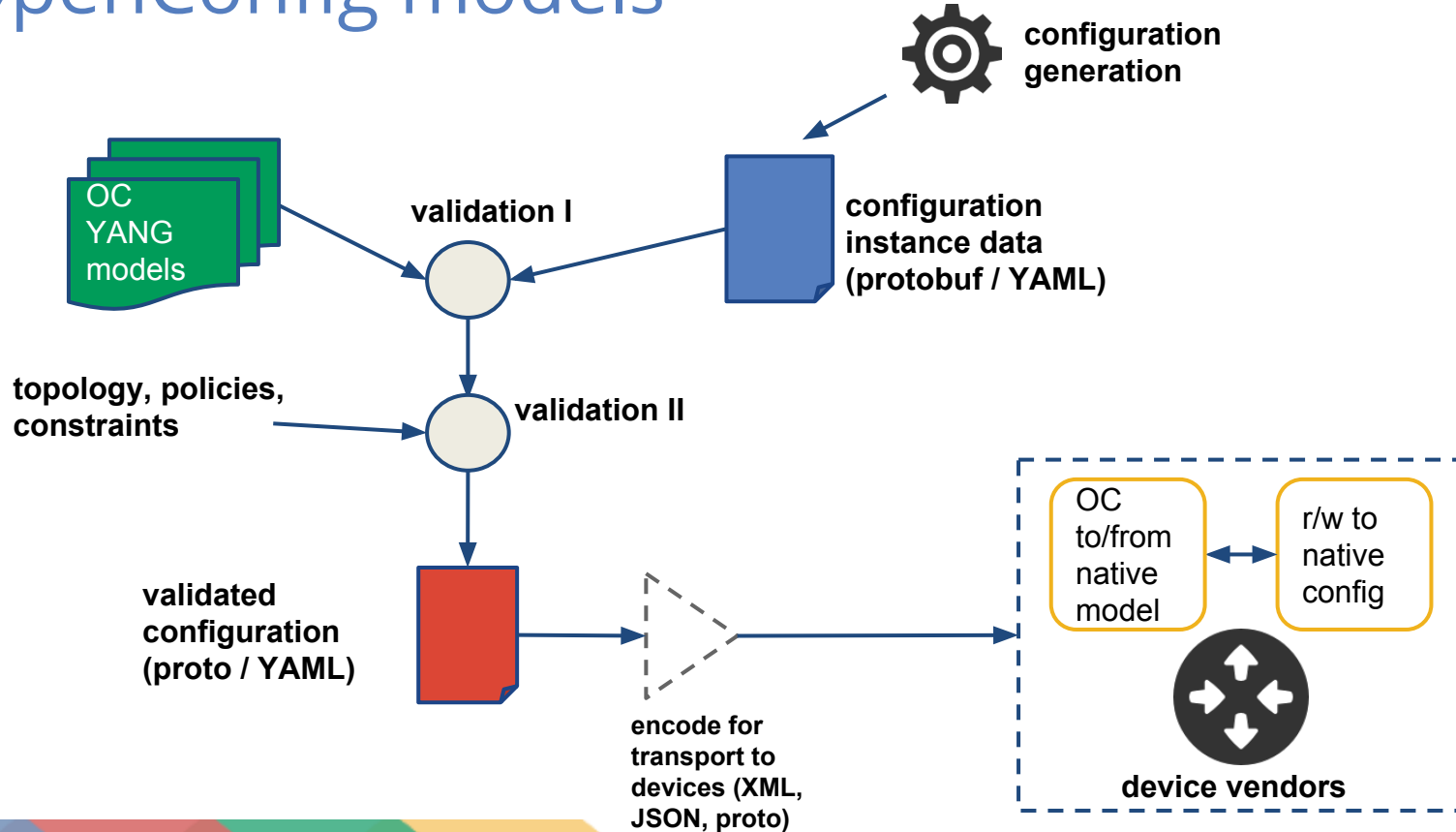
- initial models developed by OpenConfig
- extensive collaboration with vendors
- leverage existing work where possible
- publish models and docs

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



OpenConfig progress

- Initially layer-3 focused
 - BGP and routing policy models published
 - MPLS / TE model in progress
 - device-level structural model published
- interfaces, system, IS-IS, L3VPN, ...
- native implementation ongoing with several major vendors
- Optical Layer model in progress and to be open-sourced

Summary

- Time for the management plane to join the age of SDN
- *Core principles*: configuration and topology models; declarative configuration; streaming telemetry
- OpenConfig is a focused effort by operators to drive model development based on operational use cases

Invitation to additional network operators in all sectors:

leverage OpenConfig to enable your networks with a programmable management plane



Thank you