

NETCONF and YANG Tutorial

Session #1 – Technology Deep Dive

...updated from NANOG 61

Carl Moberg <calle@tail-f.com>, <camoberg@cisco.com>

...also the next session will use NETCONF and RESTCONF (and some python) for protocol level examples.

Today's Topic: Market Leaders in Configuration Management



Agenda

- Brief Overview
- Background and Motivation
- NETCONF Deeper Dive
- YANG Deeper Dive

Brief Overview

Origins of NETCONF and YANG

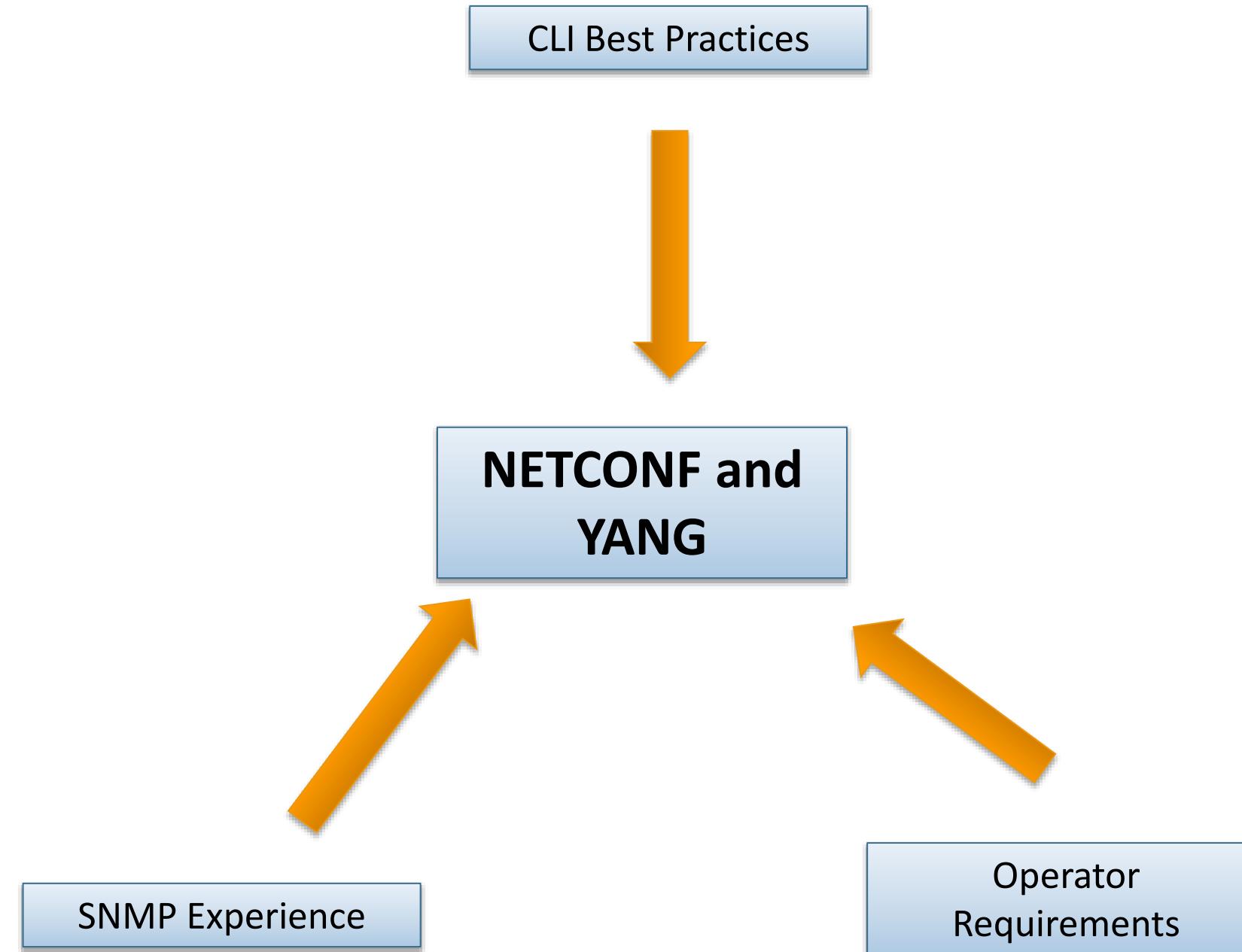
- Several meetings at events in 2001 (NANOG-22, RIPE-40, LISA-XV, IETF 52)
 - Operators expressing opinion that the developments in IETF do not really address requirements configuration management.
- June of 2002, the Internet Architecture Board (IAB) held invitational workshop on Network Management (RFC3535) to
 - Identify a list of technologies relevant for network management with their strengths and weaknesses
 - Identify the most important operator needs.



Personal Favorites from RFC 3535

- It is necessary to make a clear distinction between configuration data, and data that describes operational state and statistics.
- It is necessary to enable operators to concentrate on the configuration of the network as a whole rather than individual devices.
- Support for configuration transactions across a number of devices would significantly simplify network configuration management
- A mechanism to dump and restore configurations is a primitive operation needed by operators
- There is no common database schema for network configuration, although the models used by various operators are probably very similar. It is desirable to extract, document, and standardize the common parts of these network wide configuration database schemas.

Best Practices Coming Together



NETCONF – A Protocol to Manipulate Configuration

- IETF network management protocol
- Distinction between configuration and state data
- Multiple configuration data stores (candidate, running, startup)
- Configuration change validations
- Configuration change transactions
- Selective data retrieval with filtering
- Streaming and playback of event notifications
- Extensible remote procedure call mechanism

Why you should care:

NETCONF provides the fundamental programming features for comfortable and robust automation of network services

YANG – A Data Modeling Language for Networking

- Human readable and easy to learn
- Hierarchical configuration data models
- Reusable types and groupings (structured types)
- Extensibility through augmentation
- Formal constraints for configuration validation
- Data modularity through modules and sub-modules
- Well defined versioning rules

Why you should care:

YANG is a full, formal contract language with rich syntax and semantics to build applications on

```
list interface {  
    key "name";  
    unique "type location";  
  
    leaf name {  
        type string;  
        reference  
            "RFC 2863: The Interfaces Group MIB - ifName";  
    }  
  
    leaf description {  
        type string;  
    }  
    ...  
    container statistics {  
        config false;  
        leaf discontinuity-time {  
            type yang:date-and-time;  
        }  
    }  
  
    leaf in-octets {  
        type yang:counter64;  
        reference  
            "RFC 2863: The Interfaces Group MIB - ifHCInOctets";  
    }  
}
```

A Data Model and a Protocol

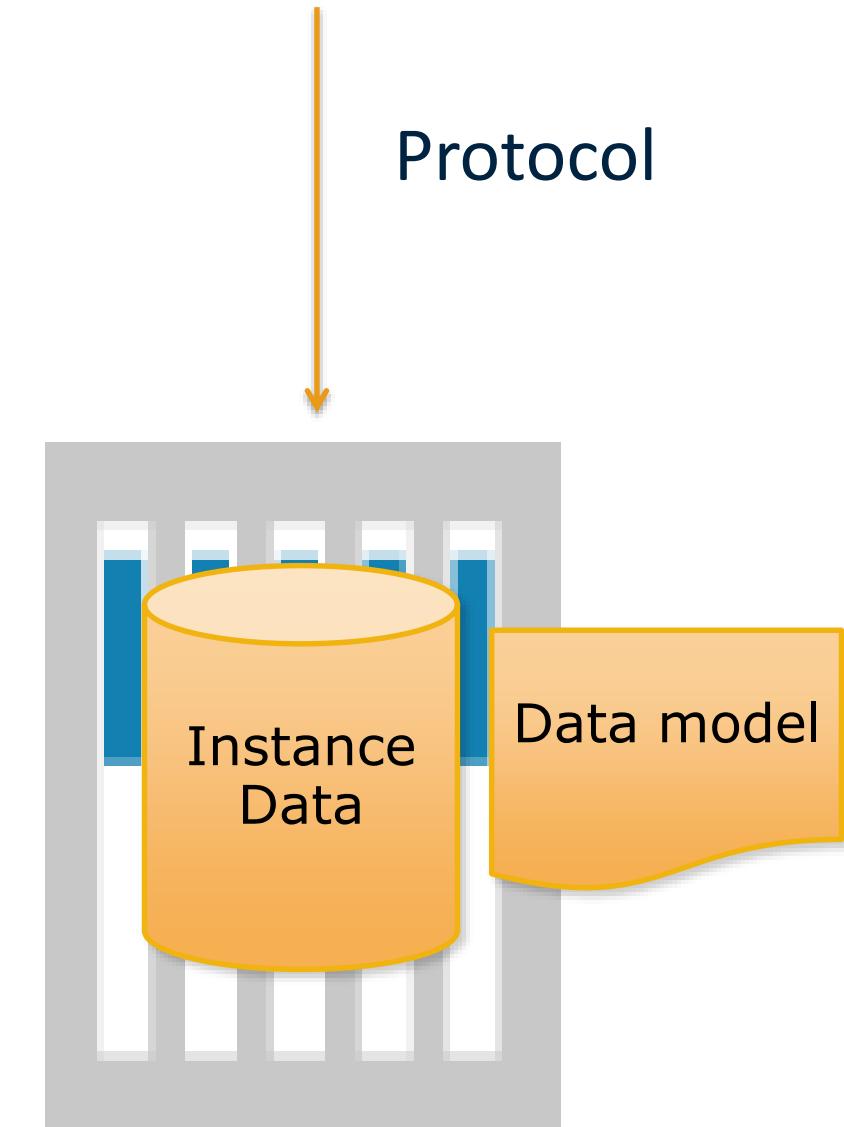
- A Data Model

- Explicitly and precisely determines the structure, syntax and semantics of the data...
- ...that is externally visible
- Consistent and complete

- A Protocol

- Provides primitives to view and manipulate data
- Encoding of the data as defined by the data model

- Beware false corollary that models can be used to validate *protocol message contents*
 - Why is this not true?



Beware Confusions on IM vs DM

“The main purpose of an IM is to model managed objects at a conceptual level, independent of any specific implementations or protocols used to transport the data.

[...]

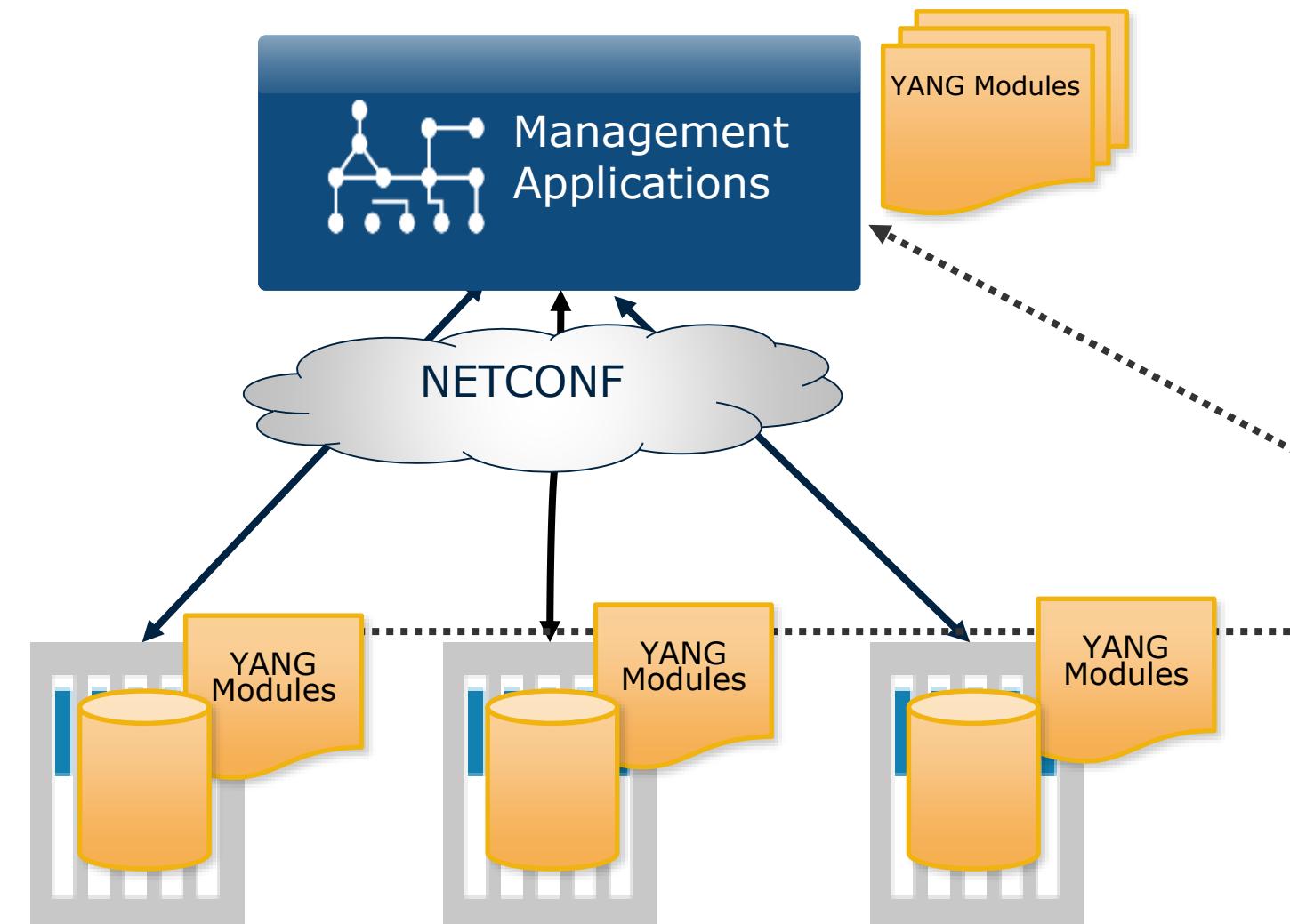
DMs, conversely, are defined at a lower level of abstraction and include many details. They are intended for implementors and include protocol-specific constructs.”

-- *RFC 3535 On the Difference between Information Models and Data Models*

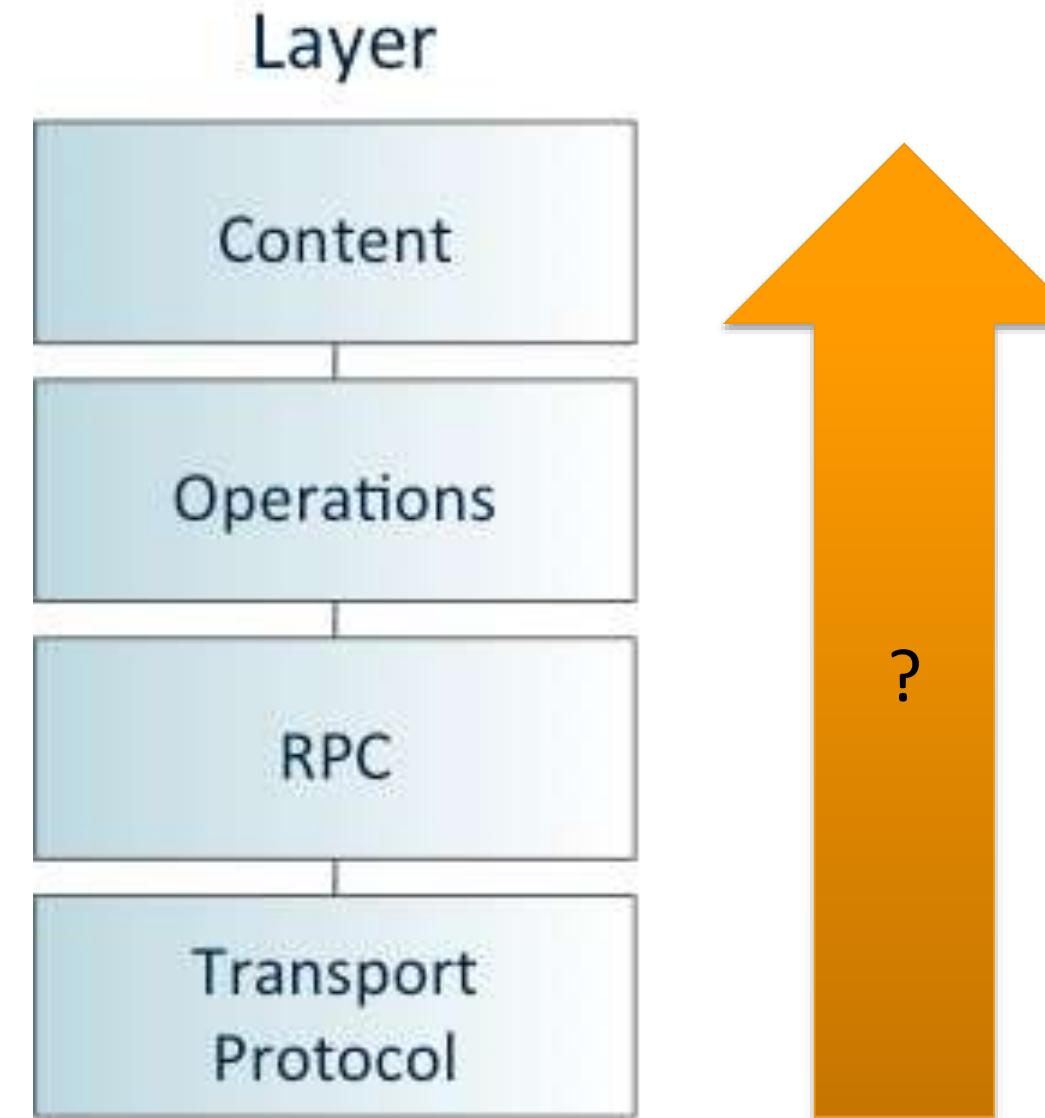
Beware Confusions on Protocol Layers

| | SNMP | NETCONF | SOAP | REST |
|------------------------|---------------|-----------------------|-------------------------------------|--|
| Standard Resources | IETF OIDs | IETF Paths | W3C | - |
| Data models | Standard MIBs | Standard YANG Modules | | |
| Data Modeling Language | SMI | YANG | (WSDL, not data) | Undefined, (WSDL), WADL, text... |
| Management Operations | SNMP | NETCONF | In the XML Schema, not standardised | HTTP operations, (POST, GET, PUT, PATCH) |
| Wire Encoding | BER | XML | XML | XML, JSON, ... |
| Transport Stack | UDP | SSH TLS TCP | SSL HTTP TCP | SSL HTTP TCP |

NETCONF and YANG in Context



What About \$PROTO? I Prefer \$PROTO over NETCONF!



A Little Bit of History (RFC 3535)

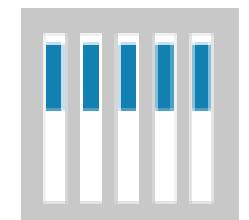
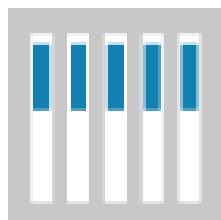
Abstract

This document provides an overview of a workshop held by the Internet Architecture Board (IAB) on Network Management. The workshop was hosted by CNRI in Reston, VA, USA on June 4 thru June 6, 2002. The goal of the workshop was to continue the important dialog started between network operators and protocol developers, and to guide the IETFs focus on future work regarding network management.

- SNMP had failed
 - For configuration, that is
 - Extensive use for fault and performance
- CLI scripting was and is dominant



configuration



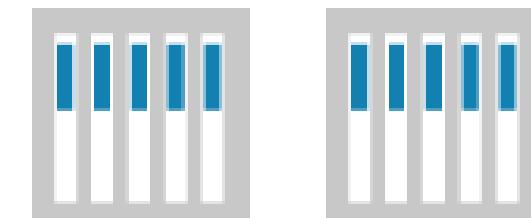
Operator Requirement #1/14

1. **Ease of use** is a key requirement for any network management technology from the operators point of view.

Maybe reduce assumption on integrators and software developers for any and all changes?



Manage



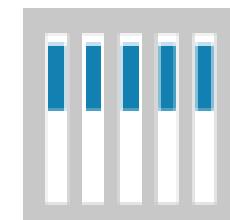
Operator Requirement #2-3/14

2. It is necessary to make a **clear distinction** between **configuration data**, data that describes **operational state and statistics**.
3. It is required to be able to **fetch separately configuration data**, operational state data, and statistics from devices, and to be able to compare these between devices.

- Clearly separate configuration and state data
- Ability to compare across devices



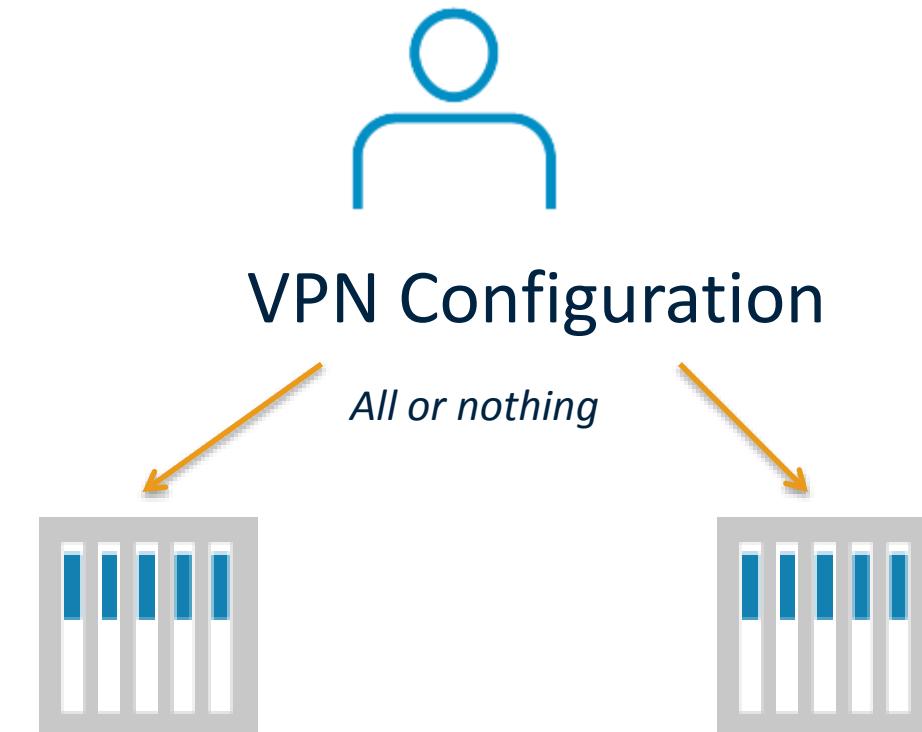
\$ show running-config



Operator Requirement #4-5/14

- 4. It is necessary to enable operators to concentrate on the **configuration of the network** as a whole rather than individual devices.
- 5. Support for **configuration transactions** across a number of devices would significantly simplify network configuration management.

- Service and Network management, not only device management
- Network wide transactions



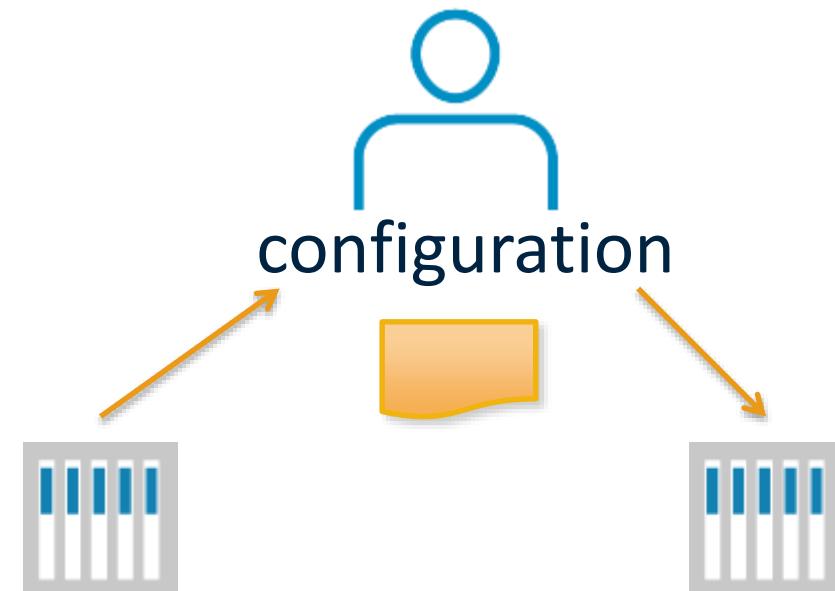
Operator Requirement #6-7/14

6. Given configuration A and configuration B, it should be possible to generate the **operations necessary to get from A to B** with minimal state changes and effects on network and systems. It is important to minimize the impact caused by configuration changes.

7. A mechanism to dump and restore configurations is a primitive operation needed by operators. Standards for **pulling and pushing configurations** from/to devices are desirable.

- Devices figure out ordering
- Minimal diffs
- Finally: backup and restore of configuration

The litmus tests of a management interface

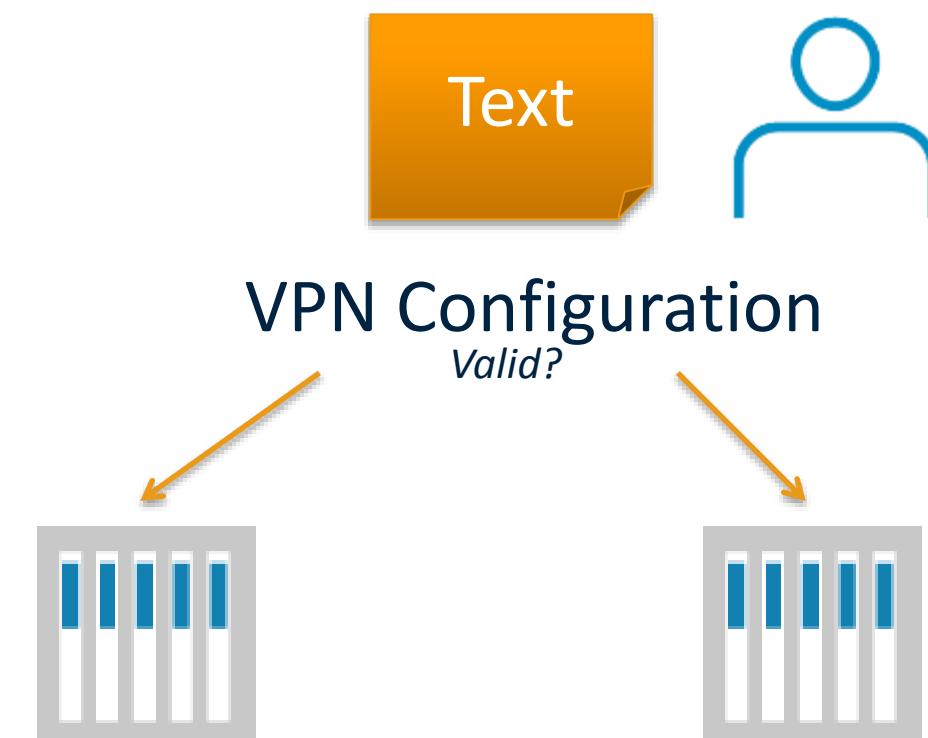


Operator Requirement #8, 10/14

8. It must be easy to do **consistency** checks of configurations over time and between the ends of a link in order to determine the changes between two configurations and whether those configurations are consistent.

10. It is highly desirable that **text** processing tools such as diff, and version management tools such as RCS or CVS, can be used to process configurations, which implies that devices should not arbitrarily reorder data such as access control lists.

- Validation of configuration
- Validation at network level
- Textual encoding



Operator Requirement #9/14

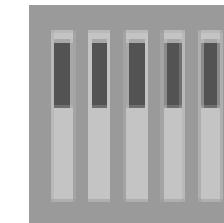
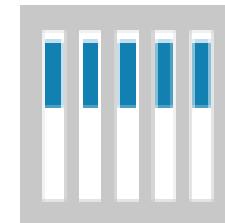
9. Network wide configurations are typically stored in central master databases and transformed into formats that can be pushed to devices, either by generating sequences of CLI commands or complete configuration files that are pushed to devices. There is no **common database schema** ..., although the models used by various operators are probably very similar.

It is desirable **to extract, document, and standardize** the common parts of these network wide configuration database schemas.

- Standardized data models



Interfaces Data-Model



Operator Requirement #13/14

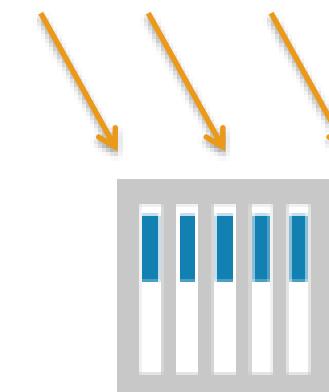
13. It is important to distinguish between the **distribution** of configurations and the **activation** of a certain configuration.

Devices should be able to hold multiple configurations.

- Support for multiple configuration sets
- Delayed, orchestrated activation



Config, Config, Commit



Operator Requirement #11,12,14/14

11. ... Typical requirements are a role-based access control model and the principle of least privilege, where a user can be given only the minimum access necessary to perform a required task.

12. It must be possible to do consistency checks of access control lists across devices.

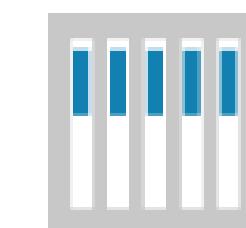
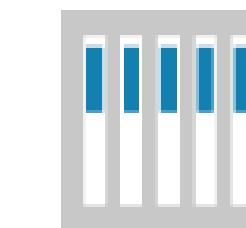
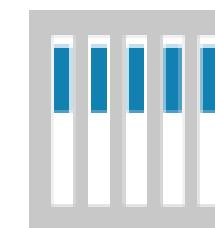
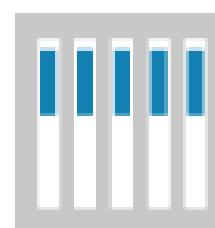
14. SNMP access control is data-oriented, while CLI access control is usually command (task) oriented. ... As such, it is a requirement to support both data-oriented and task-oriented access control

- Role-Based Access Control (RBAC)
 - CRUD...
 - Data oriented
 - Task oriented

Implications of not meeting RFC 3535 (Legacy Situation)



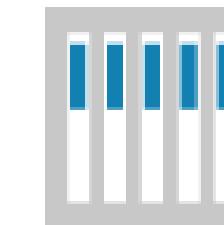
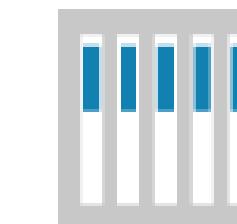
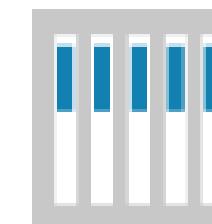
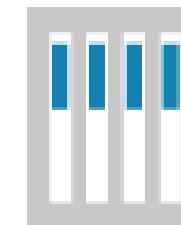
- No well-defined protocols and data-models
- Lack of transactions
- Ordering challenges
- ~USD 13B issue



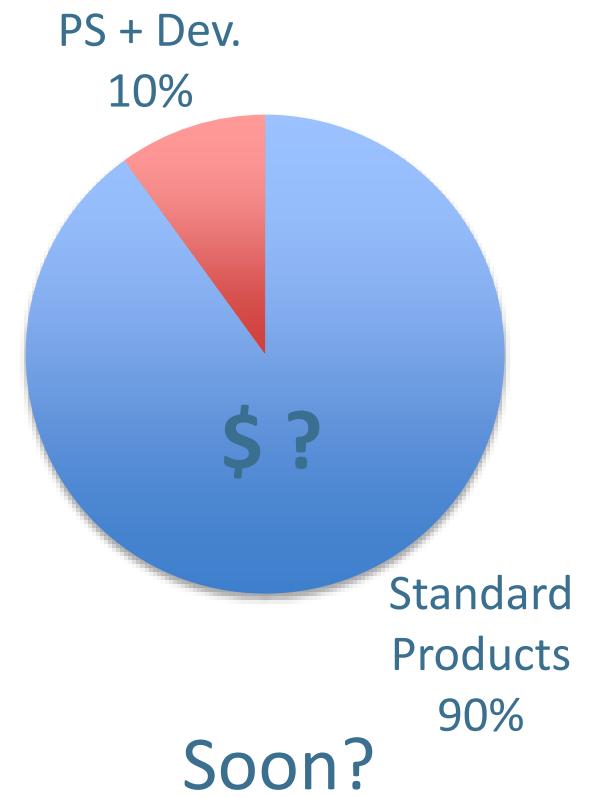
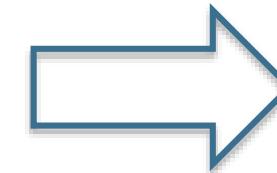
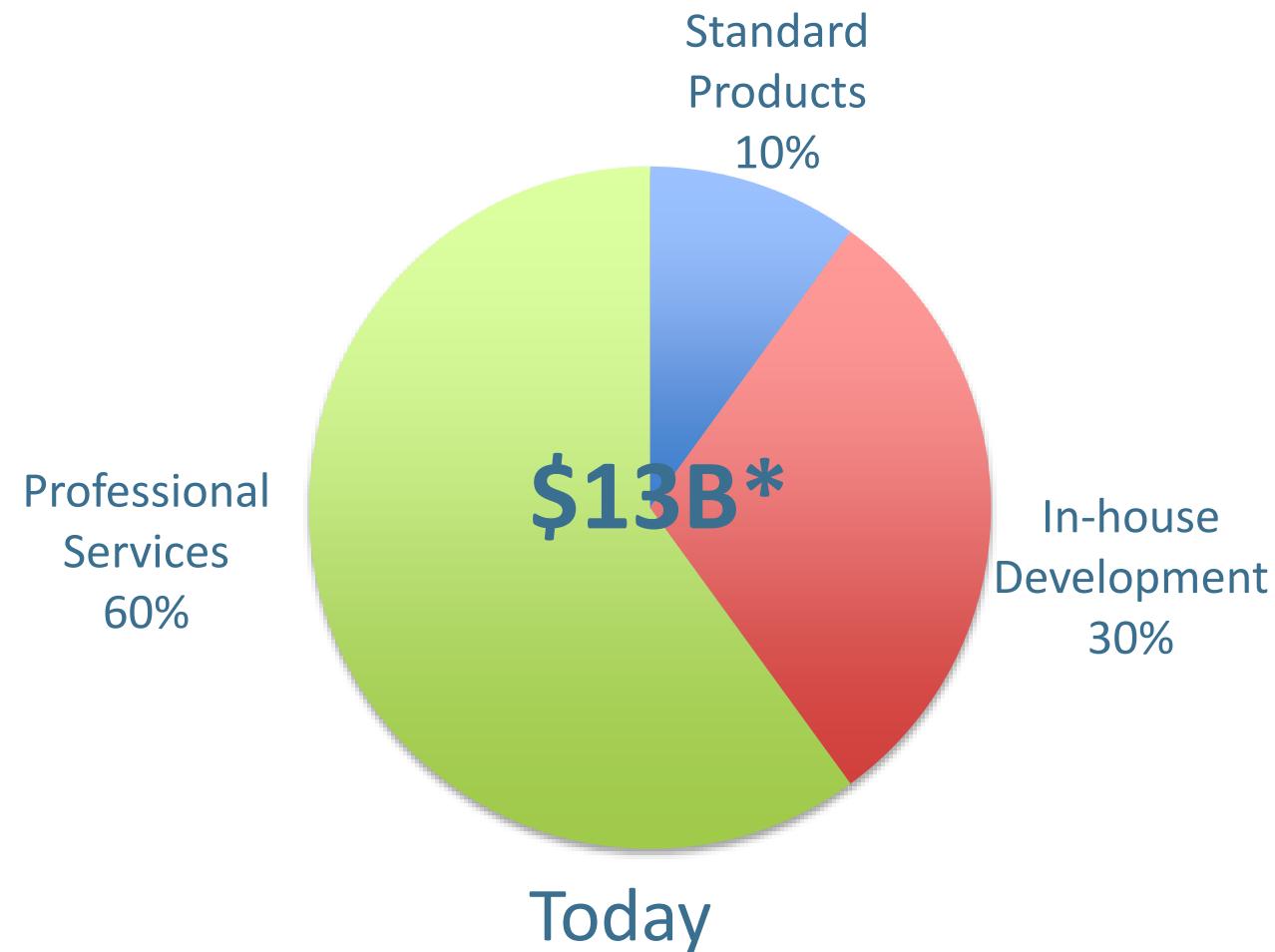
Implications of meeting RFC 3535, With Transactions



- Transactions
- Standard models
- Standardized Protocol



Or, to be more explicit...



*Gartner: Telecom Operations Management Systems (BSS, OSS and SDP), Worldwide, 3Q13 Update

NETCONF Deeper Dive

NETCONF Layering Model

