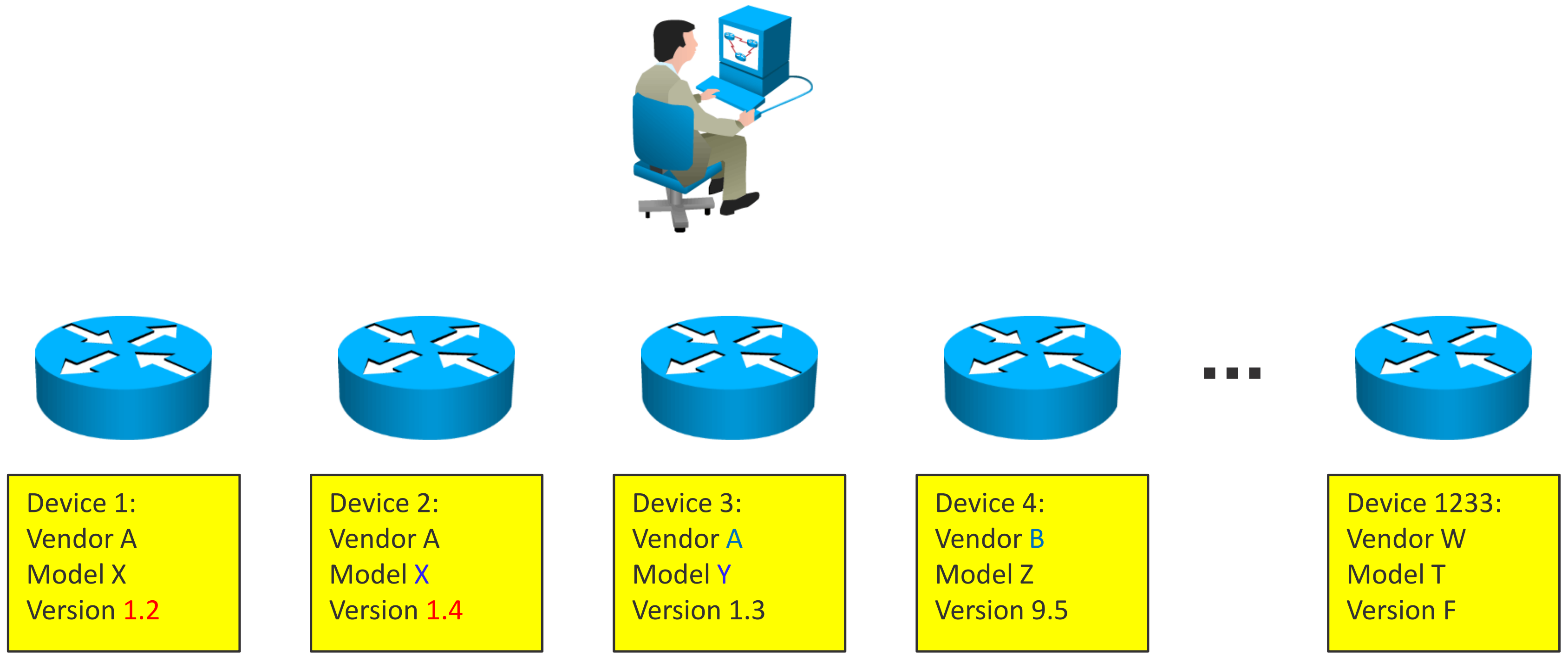
nil.com

YANG and NETCONF

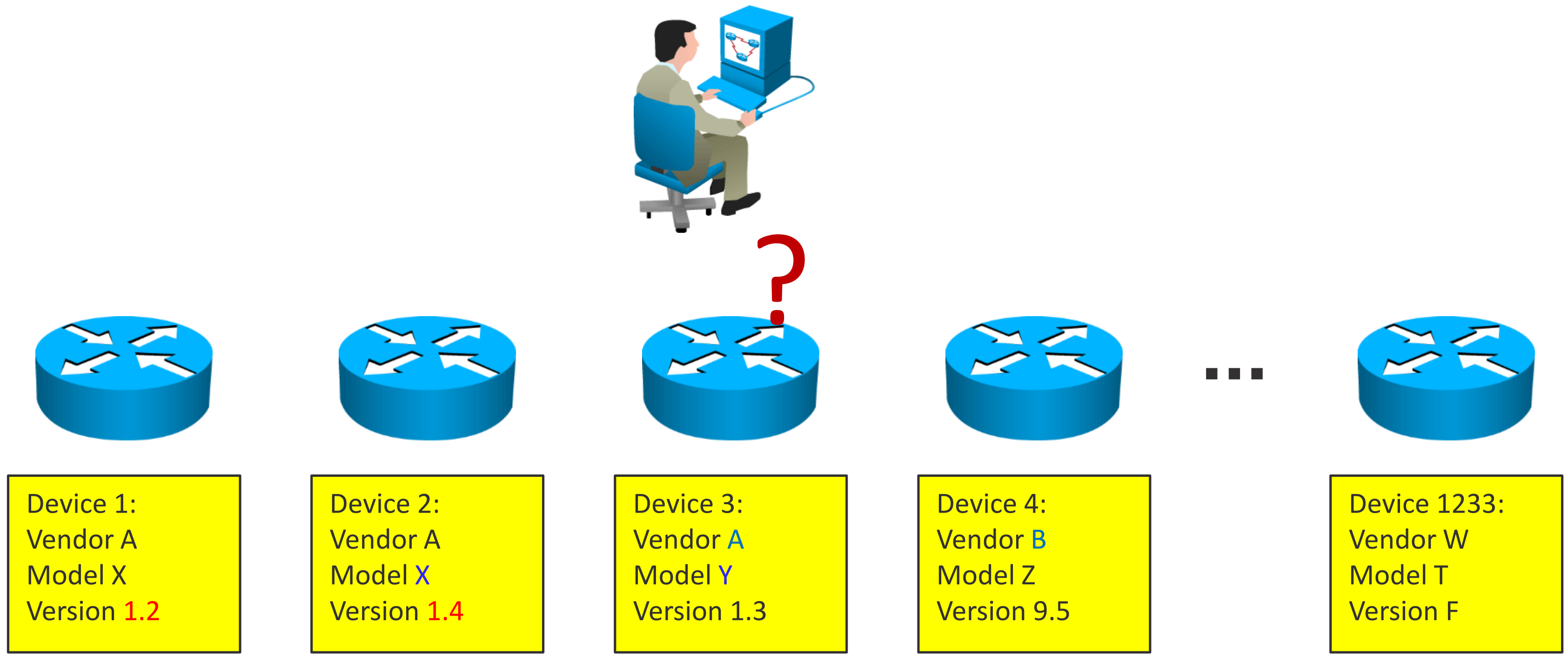
YET ANOTHER NEXT GENERATION

Network Management Characteristics



- Different management interfaces
- Different capabilities
  - CLI for configuration, SNMP for monitoring, HTTP for RBAC…

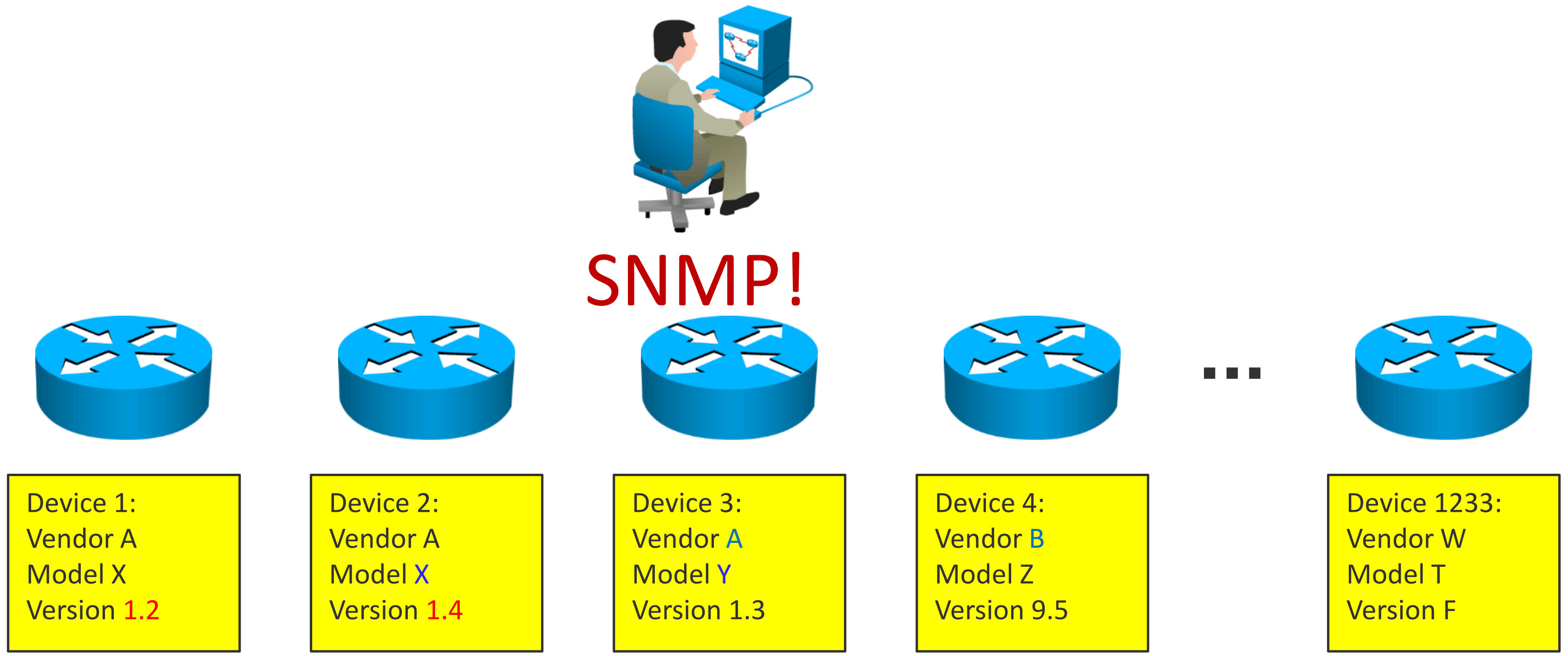
Network Management Challenges



- Learn all management interfaces
- Understand the capabilities and limitations of each device group
- Ensure consistency and reliability of configurations
- Backup configurations (and restore if/when needed)

NIL

Network Management Solution

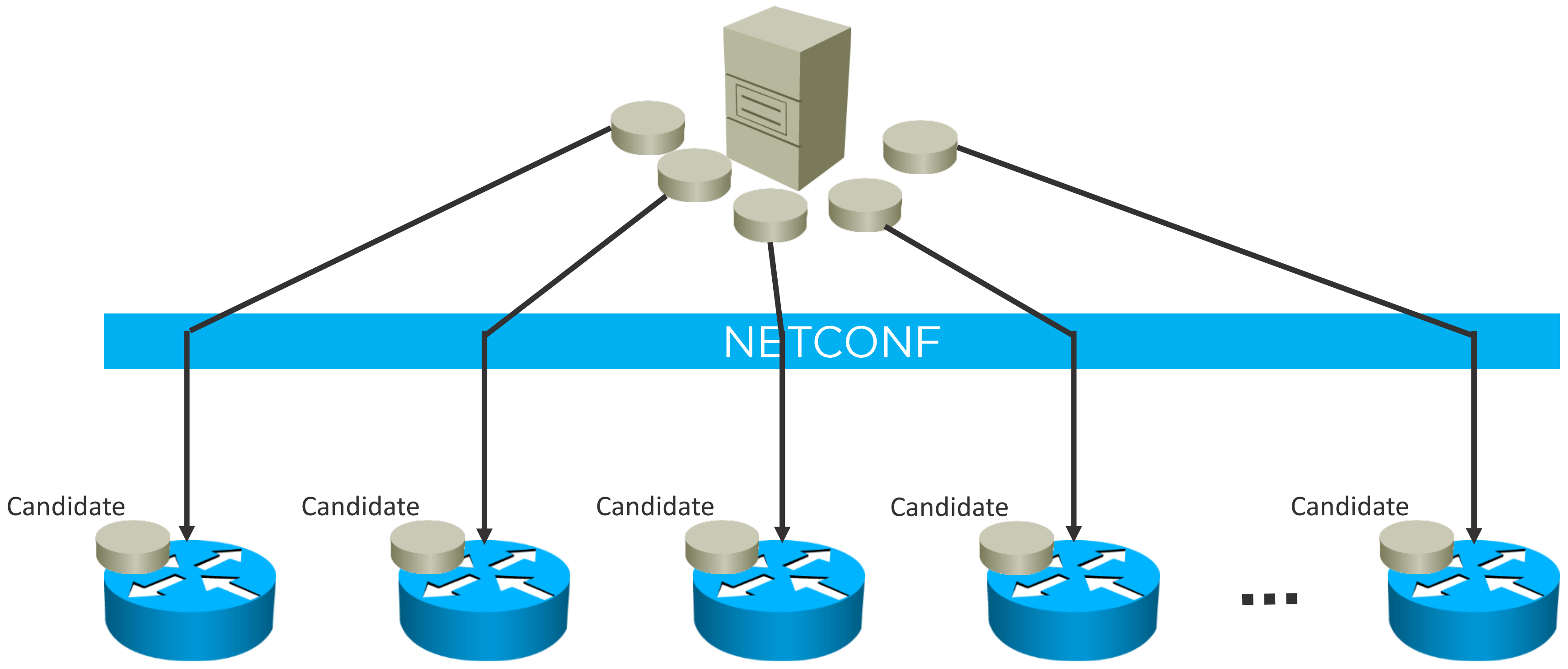


- Good intentions, but poor result in practice
- Today, SNMP is primarily only used for monitoring
- Possible reasons:
  - Most IETF standards are influenced by network equipment vendors
  - Operators' input was missing in determining what features are required for efficient network management

RFC 3535 – Operator Requrements for network management

- Ease of use!
- Clearly separating configuration from other data
- Ability to compare across devices
- Service and network management, not device management
- Network wide transactions
- Devices figure out ordering
- No unnecessary changes
- Backup and restore of configuration
- Validation of configuration which should be text based
- Standardized data models
- Support for multiple configuration sets
- Delayed, orchestrated activation
- Role-Based Access Control (RBAC): data and task oriented

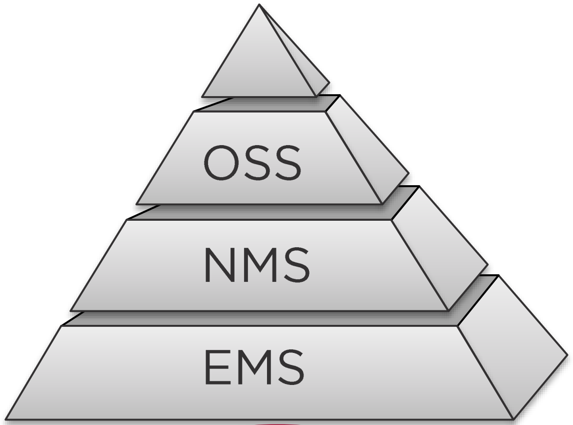
Transactional Network Management

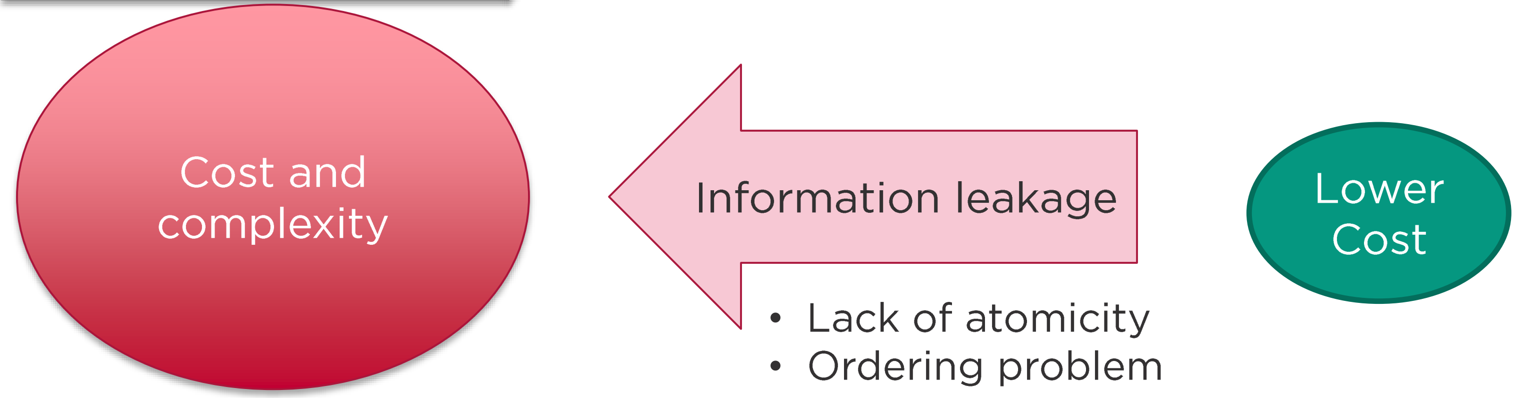


- Candidate data stores enable data-wide transactions:
  - Send configuration changes to all participating devices
  - Commit candidates if all participating devices successfully validate changes

Legacy Network Management

Operations Vs. Hardware

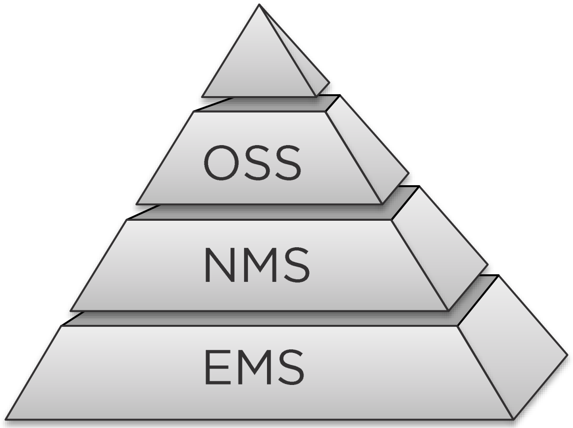




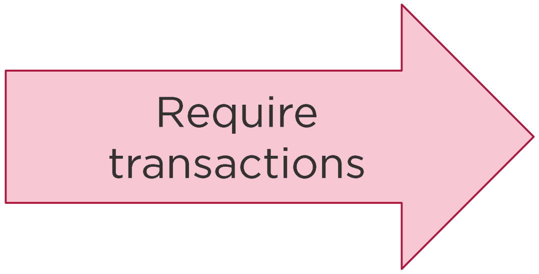
- Lack of atomicity
- Ordering problem

Transactional Network Management

Operations Vs. Hardware



Reduced Cost and complexity



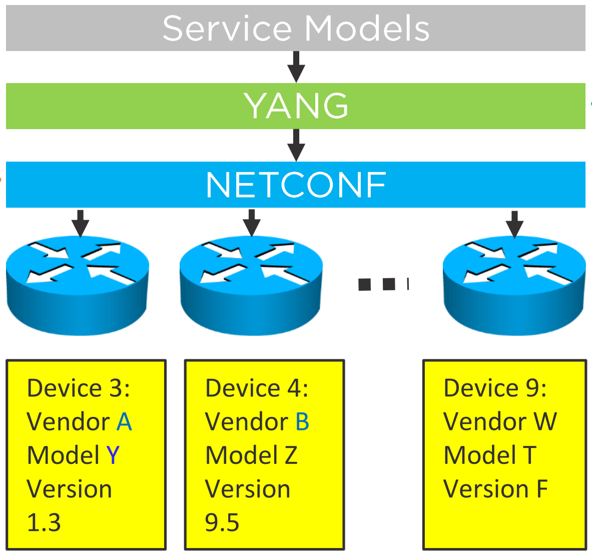
NIL

YANG

"YANG is a data modeling language used to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF), NETCONF remote procedure calls, and NETCONF notifications. YANG is used to model the operations and content layers of NETCONF"

IETF RFC 6020

The NETCONF protocol allows a manager to set configuration, query configuration and state and execute actions on the device (similar to SNMP).



Service Models
YANG
NETCONF
Device 3: Vendor A Model Y Version 1.3
Device 4: Vendor B Model Z Version 9.5
Device 9: Vendor W Model T Version F

The YANG models describe everything there is to …

- Configure
- Monitor
- Admin actions
- Notifications

… for each device type and version (similar to MIBs).

NIL

YANG Characteristics

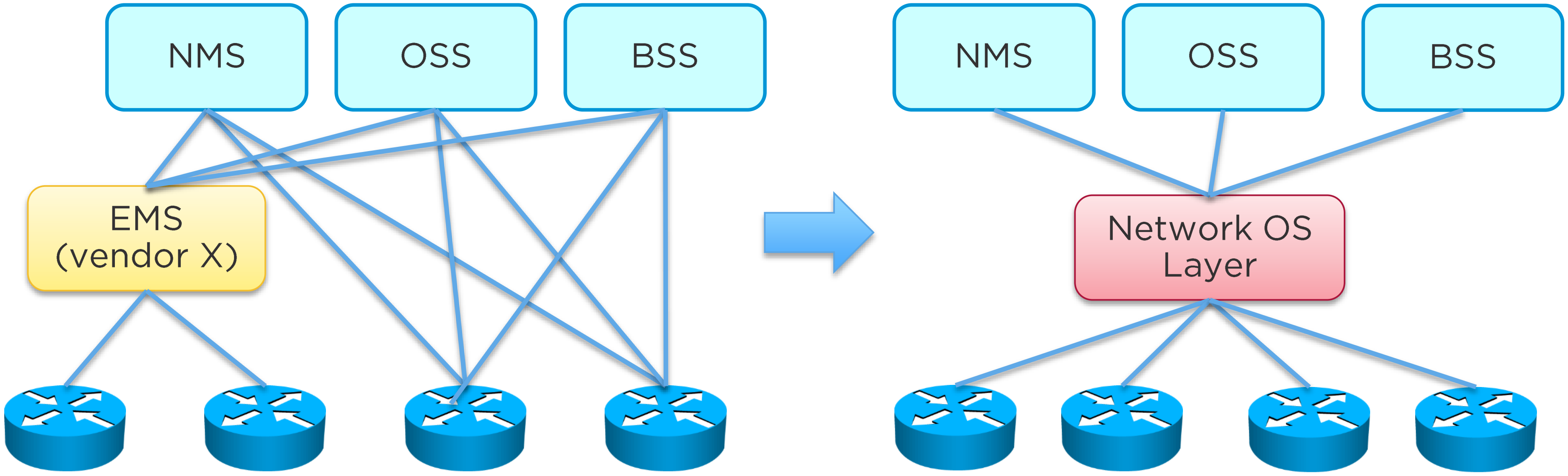
- Designed for readability
- Uses UTF-8 encoding to support international character sets
- Simplicity enables fast service model development
- Modularity and hierarchy enable reuse of code

```
list loopback-interface {
  key Loopback;
  unique ip-address;

  leaf Loopback {
    type string;
  }

  leaf ip-address {
    type uint32;
  }
}
```

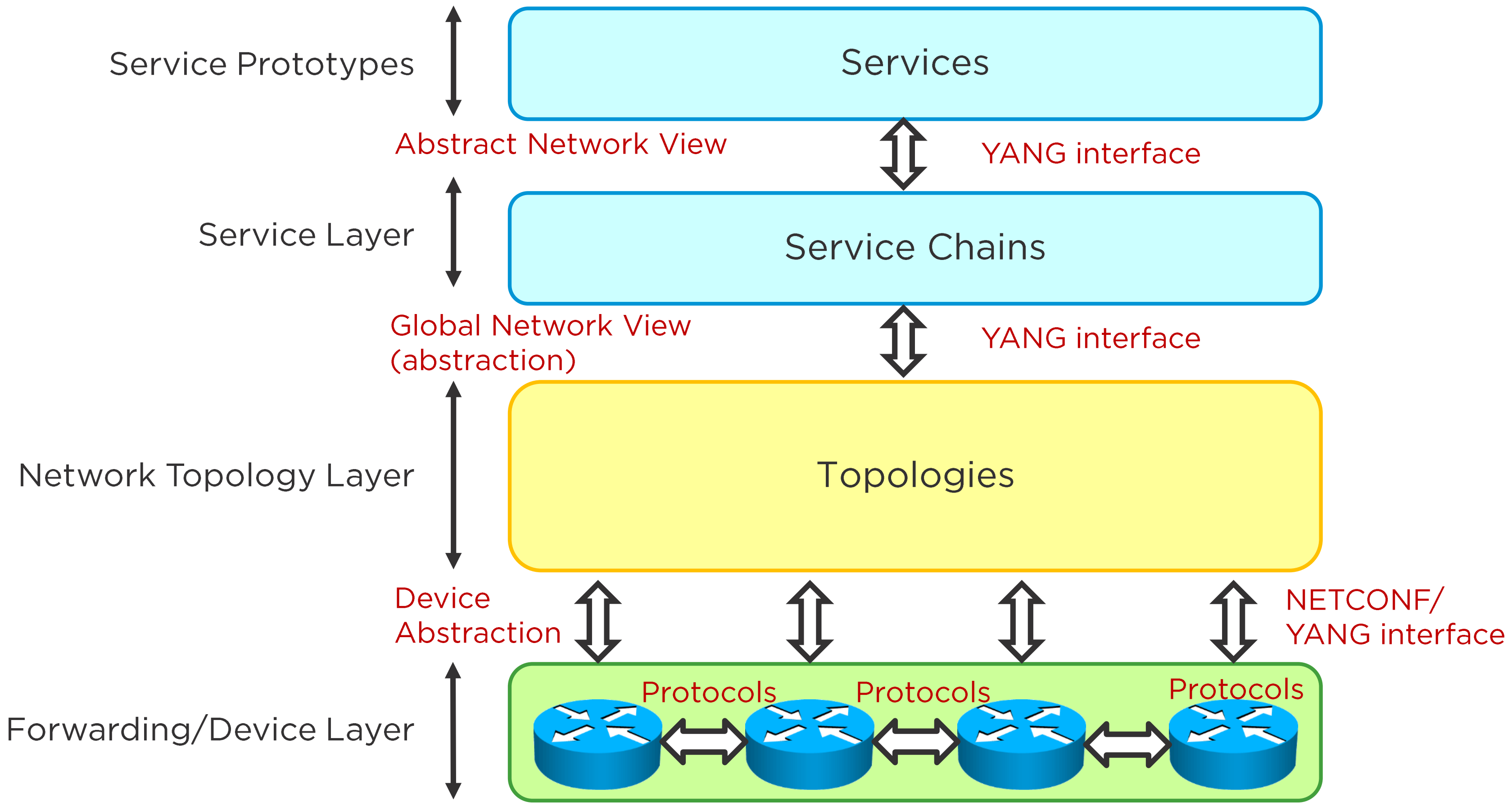
```
 <list name="loopback-interface">
    <key value="Loopback"/>
    <unique tag="ip-address"/>
    <leaf name="Loopback">
      <type name="string"/>
    </leaf>
    <leaf name="ip-address">
      <type name="uint32"/>
    </leaf>
  </list>
```

The Power of Abstractions

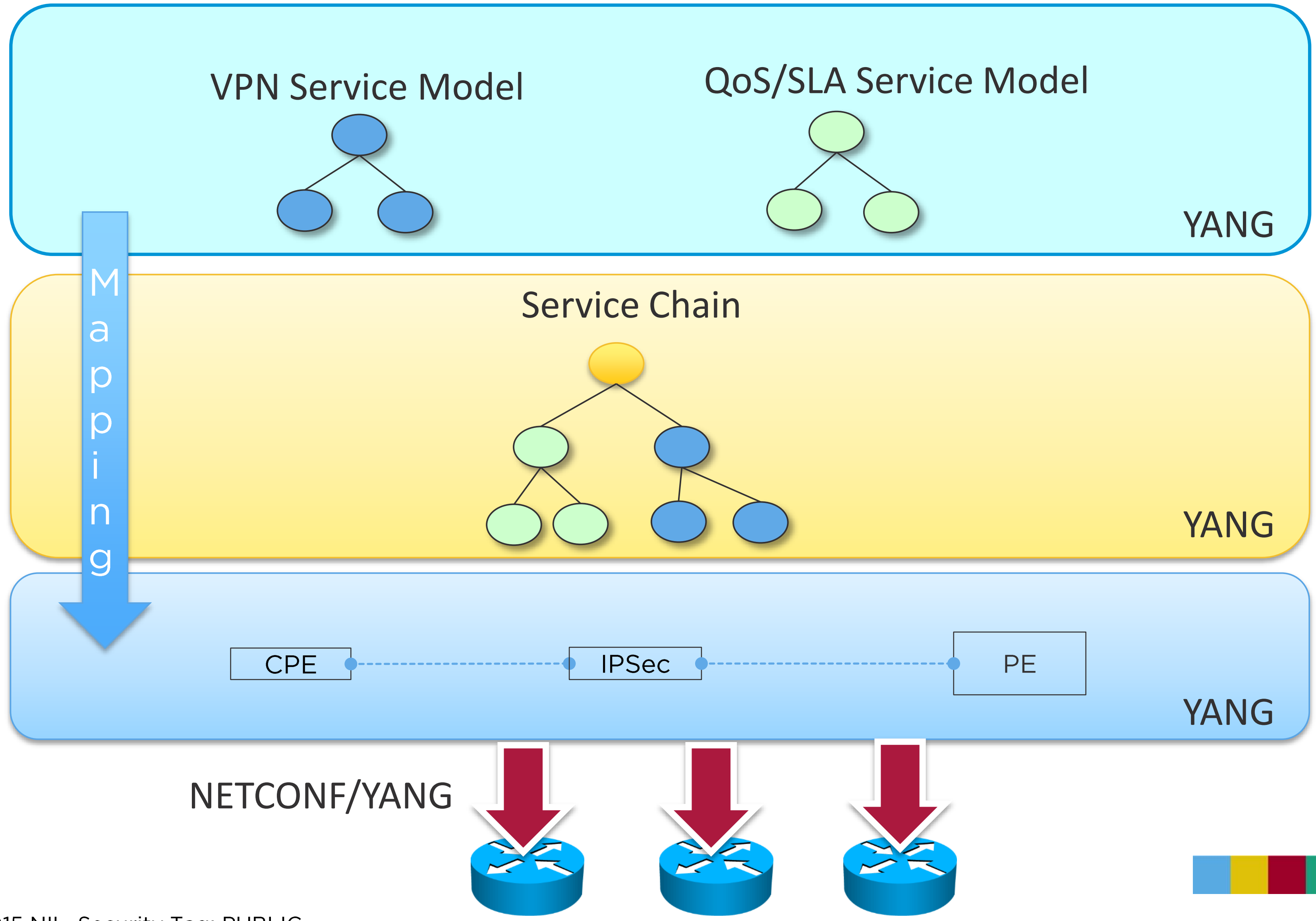
- Networking currently lacks fundamental abstractions
- As a result, networks are hard to manage
- Abstractions simplify and speed up development
- YANG can provide a global network view abstraction



Layered Architecture



Example


VPN Service Model
QoS/SLA Service Model
YANG
Mapping
Service Chain
YANG
CPE
IPSec
PE
YANG
NETCONF/YANG

What we‘ve learned so far

- The protocol is mature (survived 10+ years)
- Can effectively replace CLI/other interfaces
- Abstractions can be done on many levels
- Configurational integrity is guaranteed
- Provides complete and partial rollbacks
- Provides network wide transactions
- Simplifies configuration management and service lifecycle management
- Service chaining is much easier to implement
- Enables truly modular service design
- Enables reusability of service components
- Reduces complexity of multi-doman network orchestration

What we‘ve learned so far

- Actual support varies from vendor to vendor
  - Danger of poor implementation
- In order to maintain transaction based management, different agents have to be implemented (ConfD…) for non-NETCONF devices
  - Implementation quality varies!
  - Requires additional CPU/Memory!
- YANG models can become very large – memory implications
- YANG still requires certain extensions (30+ open drafts)
- Device-level YANG models require standardisation
- NETCONF/YANG are designed for configuration management in mind
  - Changes are less dynamic than operational
  - What about network monitoring?

nil.com