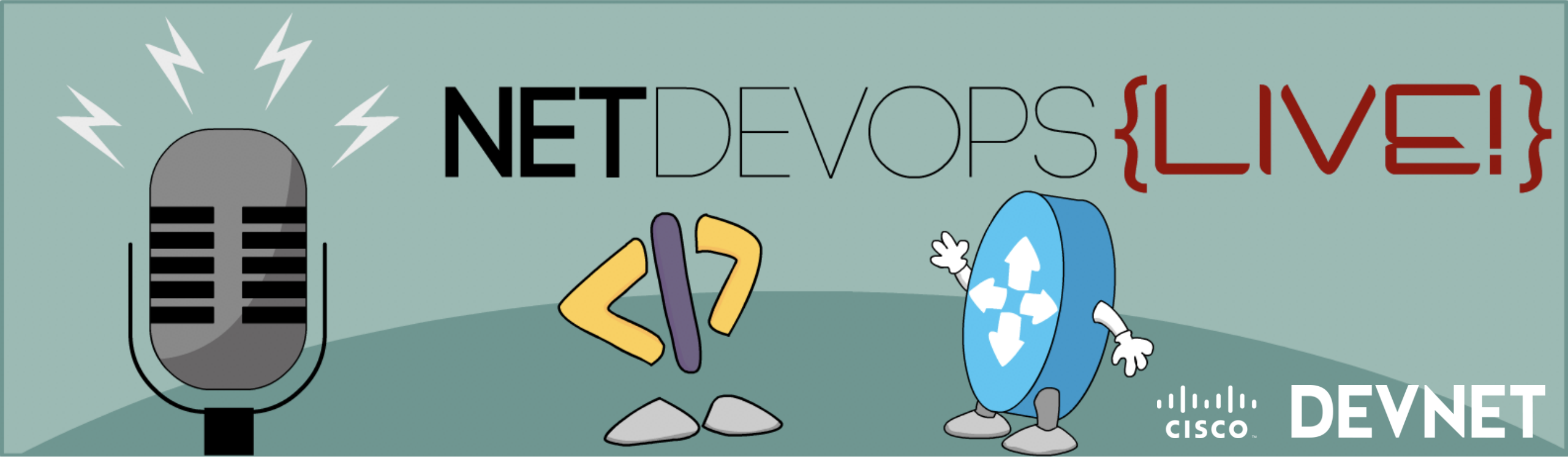


Introduction to Network Services Orchestrator – the single API and CLI for your network

Kevin Corbin, ccie 11577

Cisco Automation Dude

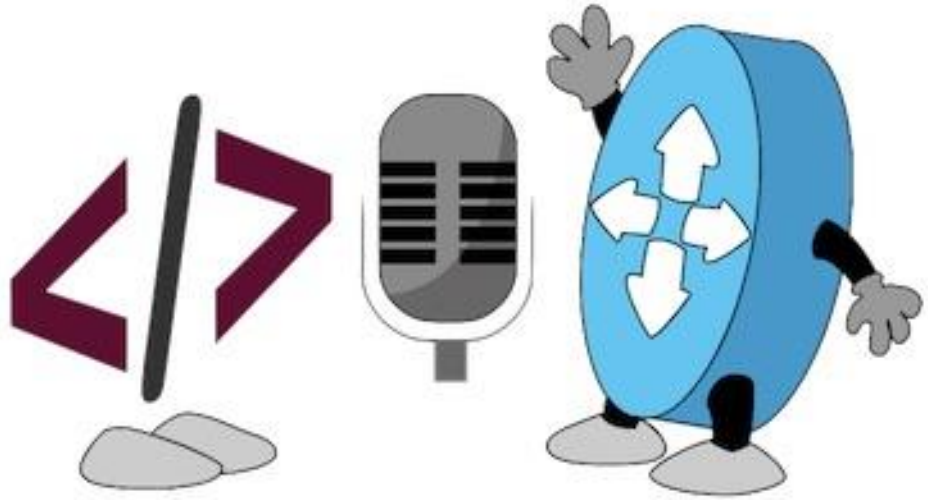
Twitter: @kecorbin

Season 1, Talk 6

https://developer.cisco.com/netdevops/live

What are we going to talk about?

- A brief History of tail-f
- Introduction to NSO
- Using NSO as a single CLI and API to solve common configuration and automation challenges

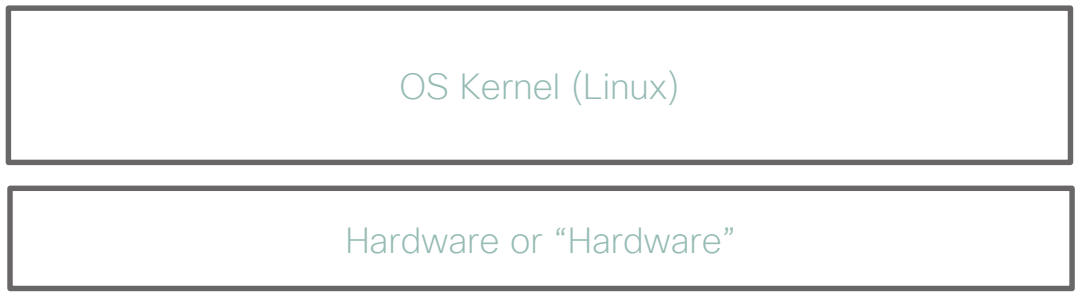


A Brief History of tail-f

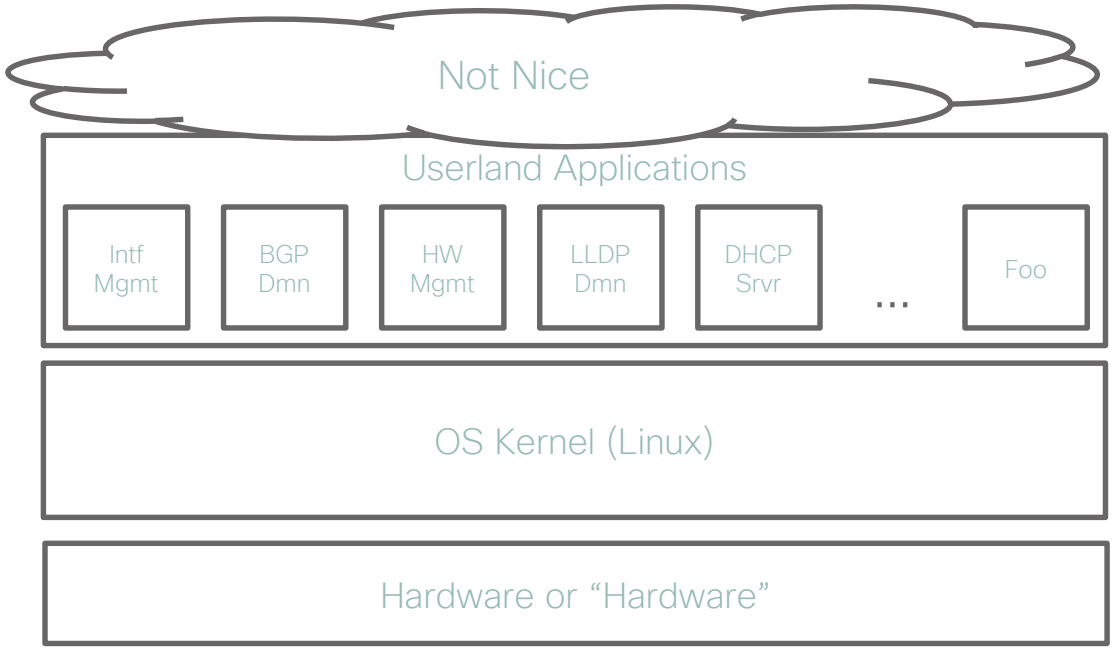
How to Design a Network Device Step-by-step

Hardware or “Hardware”

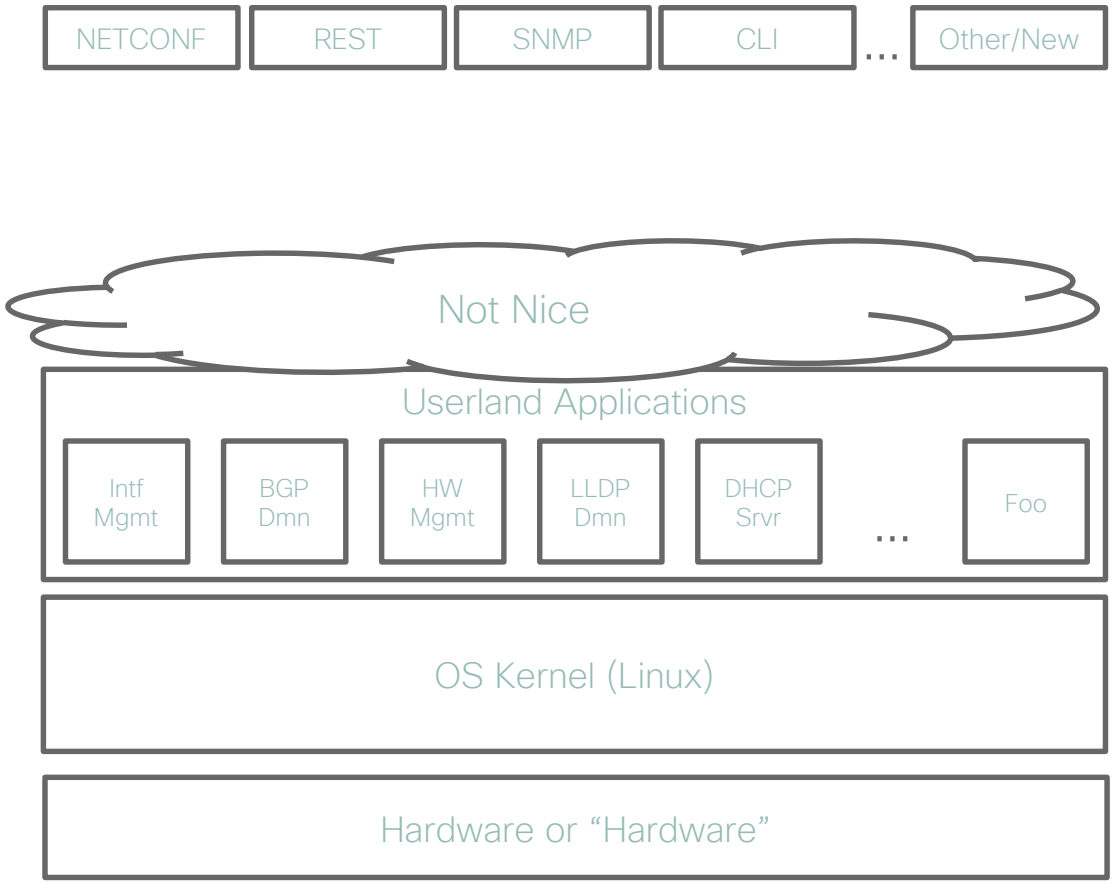
How to Design a Network Device Step-by-step



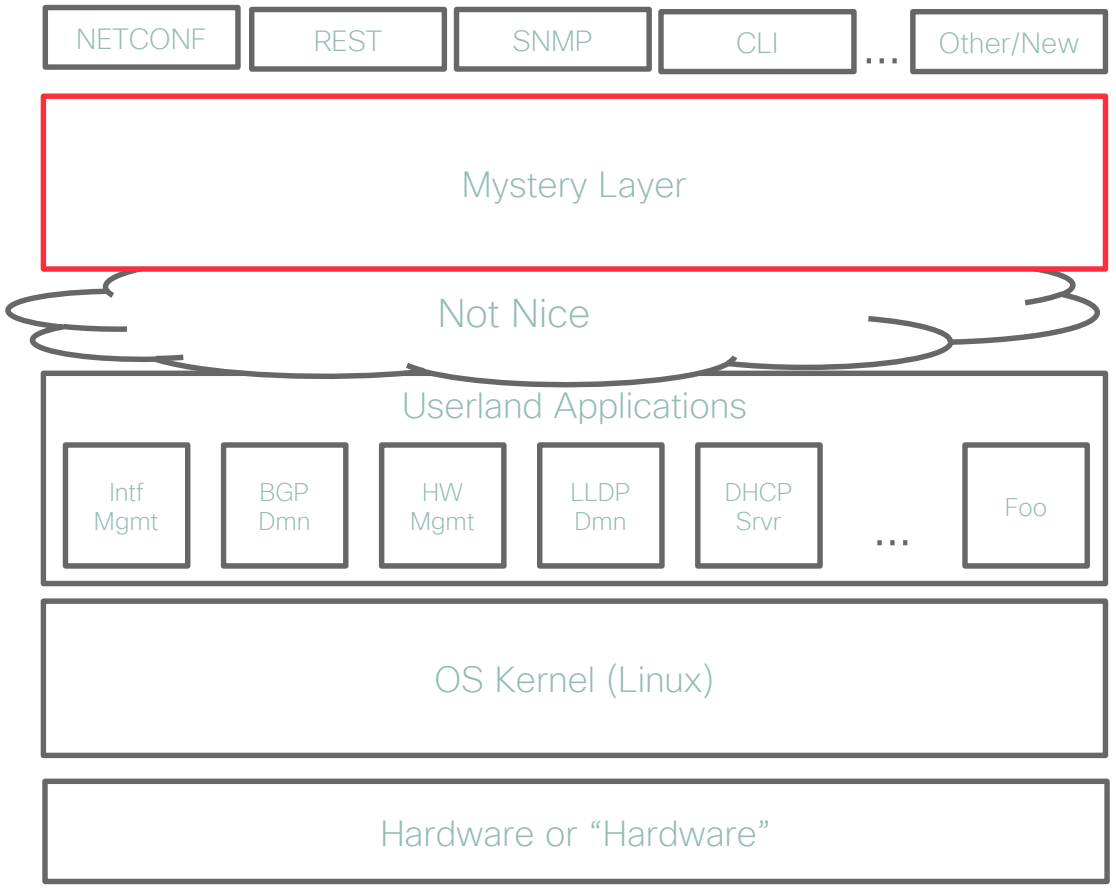
How to Design a Network Device Step-by-step



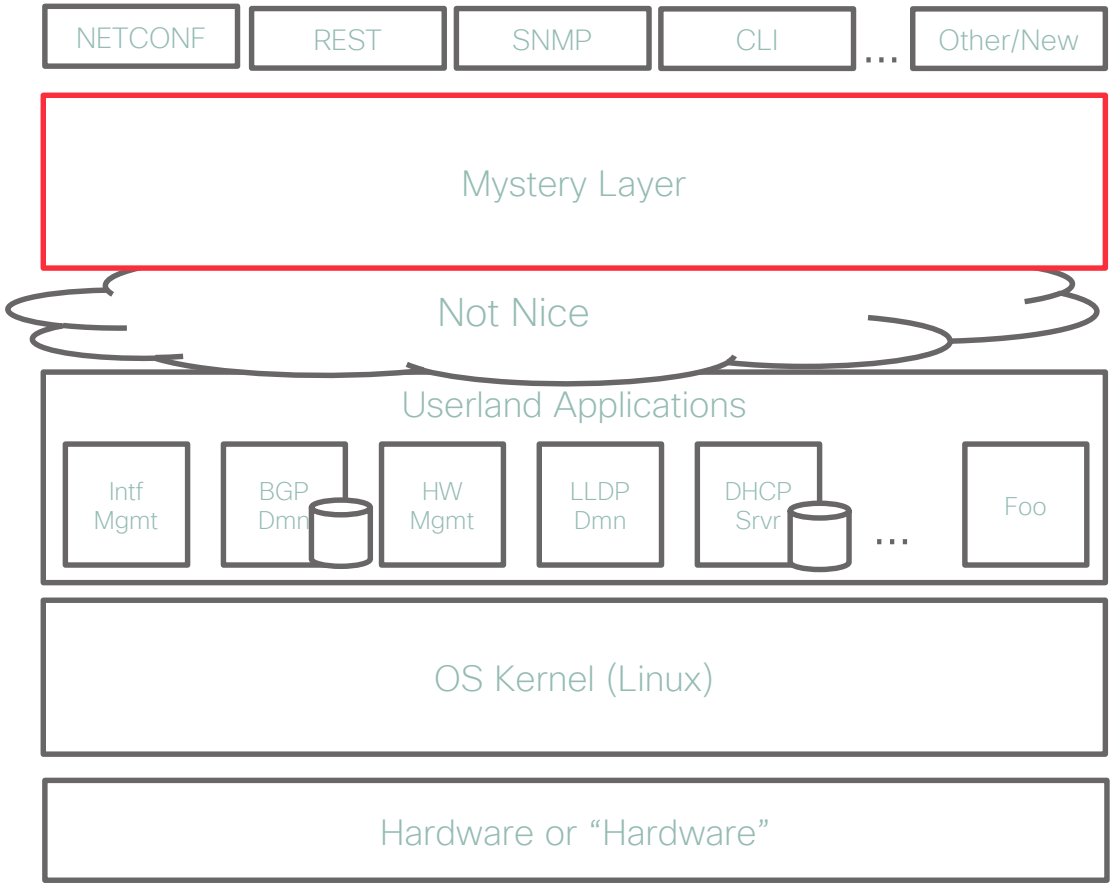
How to Design a Network Device Step-by-step



How to Design a Network Device Step-by-step



How to Design a Network Device Step-by-step

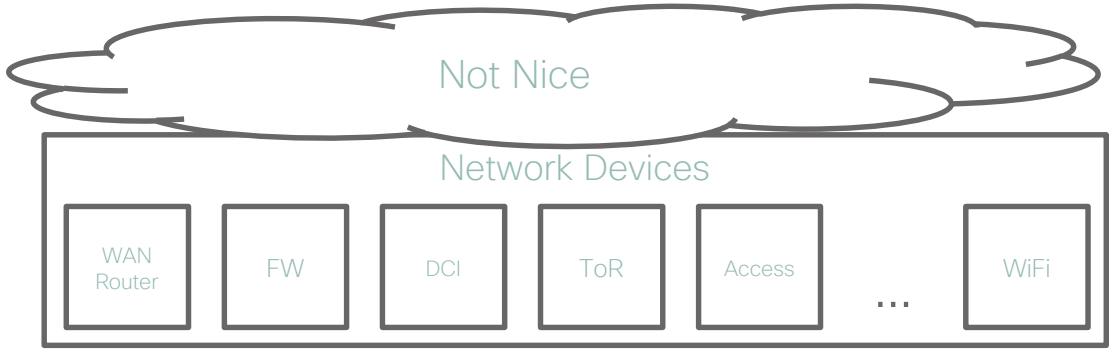


- Many NBI agents
- **Very** messy southbound
- 1:N mapping North-South

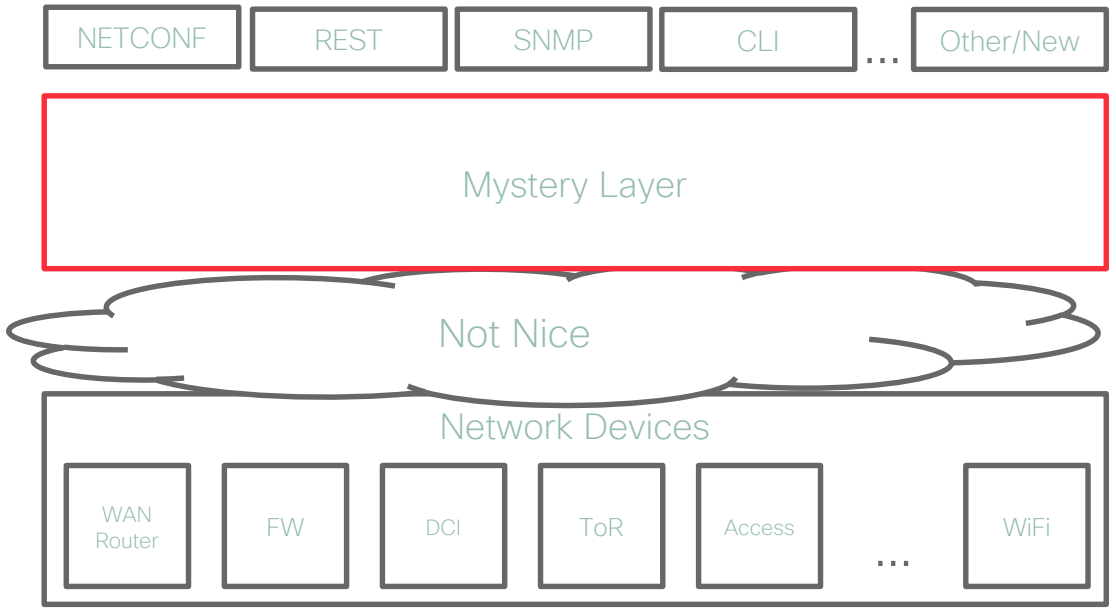
- Model-based APIs
- Transactions
- Subscription-based integration

tail-f.com/confd-basic/

How to Design a Network Orchestrator



How to Design a Network Orchestrator

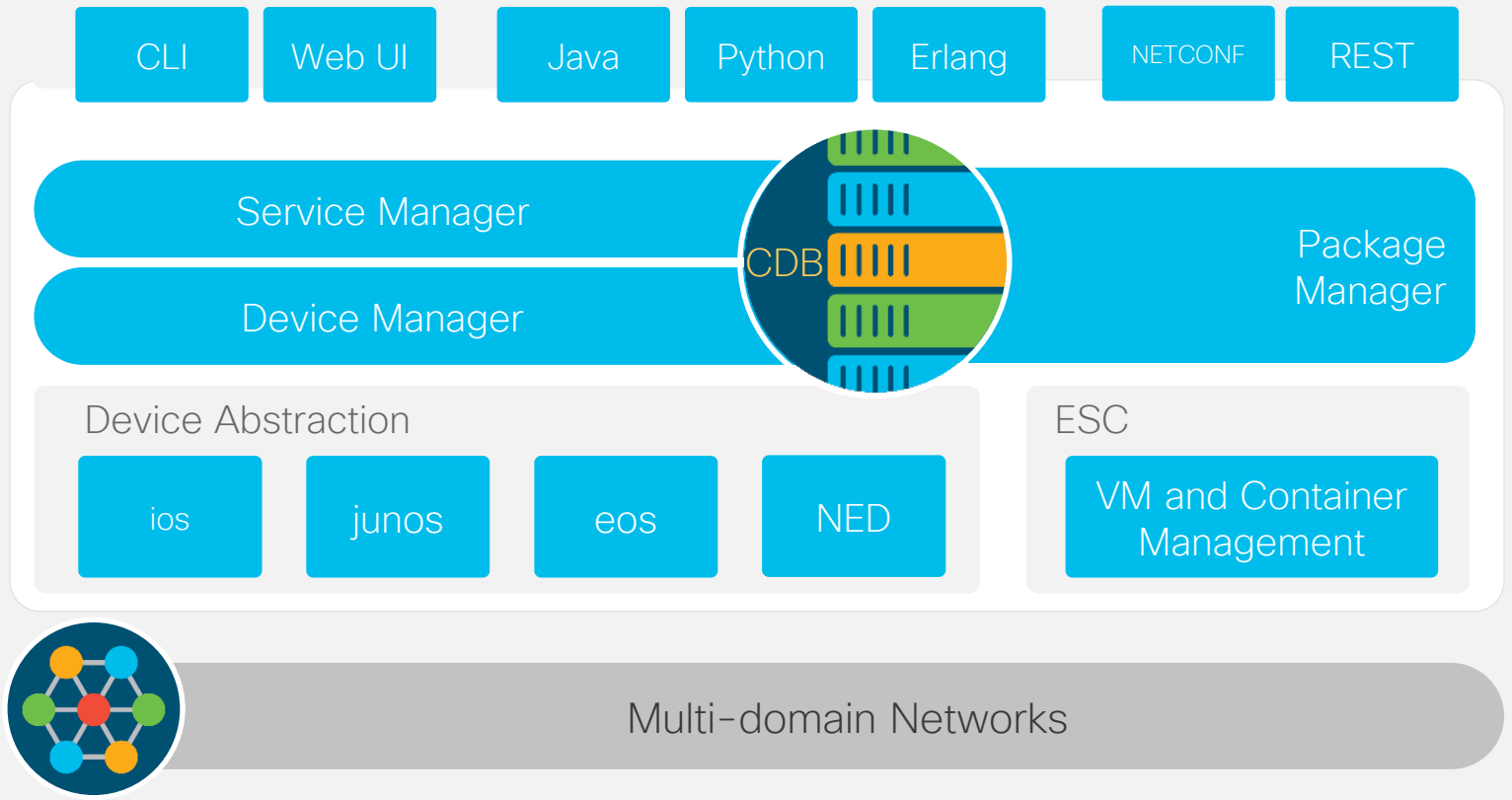


How to Design a Network Orchestrator

- Many NBI agents
- **Very** messy southbound
- 1:N mapping North-South

- Model-based APIs
- Transactions
- Subscription-based integration

Cisco NSO – Architecture



- Model-driven, end-to-end service lifecycle and customer experience focused orchestration
- Seamless integration with order managers and ITSM systems
- Loosely-coupled and modular architecture leveraging open APIs and standard protocols
- Orchestration across multi-domain and multi-layer for network-wide, centralized policy and services

The Industry's Broadest Multivendor Support

Over 100 Supported NEDs

Developer Tools and SDK Content

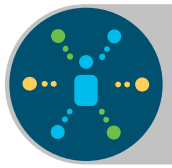
- Dev-local multi-vendor network simulator
- Full production-grade installation in dev environments
- YANG tools including validator, compiler

- Dev-local multi-vendor network simulator
- Build- and runtime validation of package content
- Offline-tools for validating version migration

- Self-contained and versioned package format
- Hitless package installation and version migration
- Local or remote project and package locations

Full runtime and tooling runs in local environment

Automation Journey



Network Engineering

Automation

Day-to-day management of rapidly growing, complex networks

Challenges

- Error-prone, manual tasks
- Growing backlog
- Virtualization is here

Ops and Provisioning

Customer Experience

Provisions services and manages service quality in networks

Challenges

- No service insight
- Lack of automation
- Quality issues in service delivery

Architecture

Time-to-Market

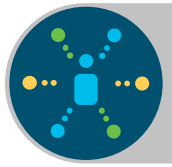
Develops new network services on demand

Challenges

- Implementation time
- Cost of change
- Lack of tooling

Automation Journey

TODAY'S FOCUS



Network Engineering

Automation

Day-to-day management of rapidly growing, complex networks

Challenges
- Error-prone, manual tasks
- Growing backlog
- Virtualization is here

Stage 1

Network API
Utilize a single interface to all network devices

STAYED TUNED! – THESE WILL BE COVERED IN FUTURE EPISODES

Ops and Provisioning

Customer Experience

Provisions services and manages service quality in networks

Challenges
- No service insight
- Lack of automation
- Quality issues in service delivery

Stage 2

Service Abstraction
Leverage one central API for all services

Architecture

Time-to-Market

Develops new network services on demand

Challenges
- Implementation time
- Cost of change
- Lack of tooling

Stage 3

Transformation
Develop your own services

Stage 1 – The Network API

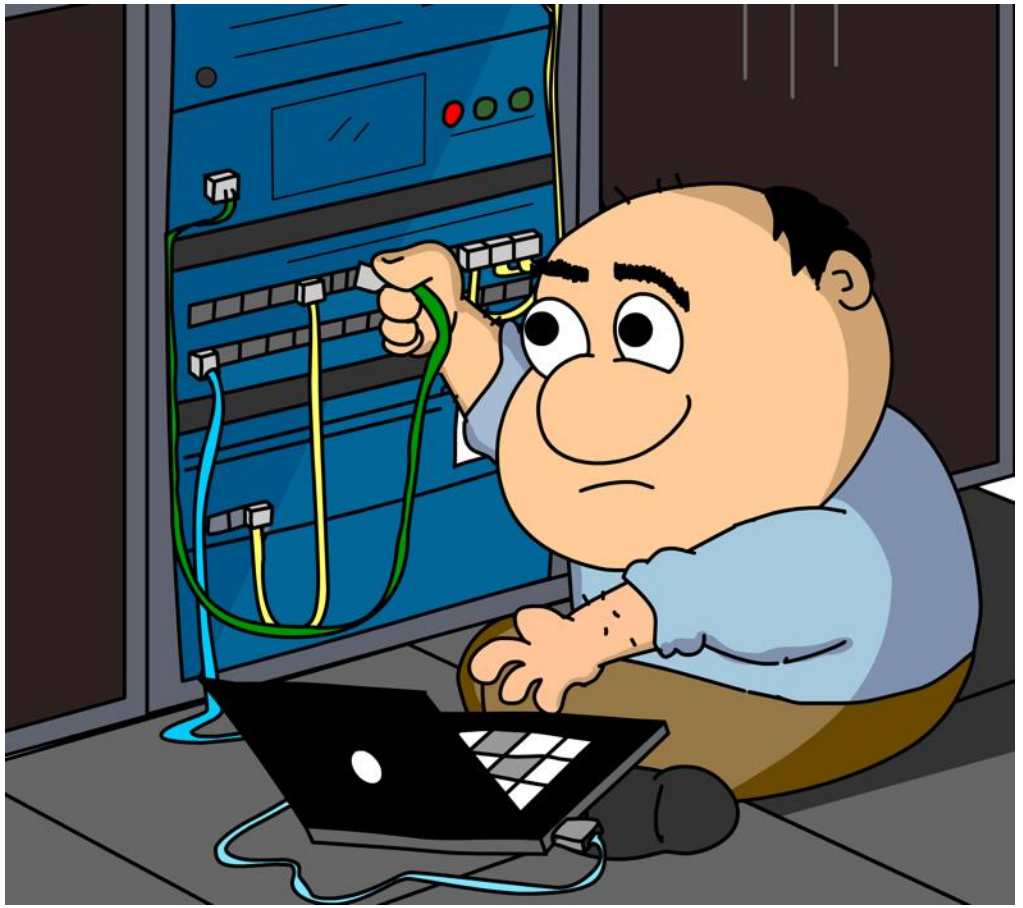
Fixes these chronic issues:

- Lack of automation, Managing device configuration
- Quality issues in delivery
- Inflexibility to change existing configuration (create and delete only)
- CLI Scripting–inflexible and high fallout

- A YANG-Based Configuration Database
- Sync-From/Sync-To Devices
- Rich set of Northbound APIs rendered from the database / devices
- Consistent and Network-wide CLI, UI, REST
- Start with CLI but gradually introduce others e.g. REST for scripting trivial tasks
- Transaction-safe operations and rollback!
- **Device Configuration Management AND Accurate network configuration state**

- Choice of technology up to the **consumer**!!

Demo Time!



Governance – Current State

Engineering teams create standard templates of device configurations

Devices are audited for configuration standards

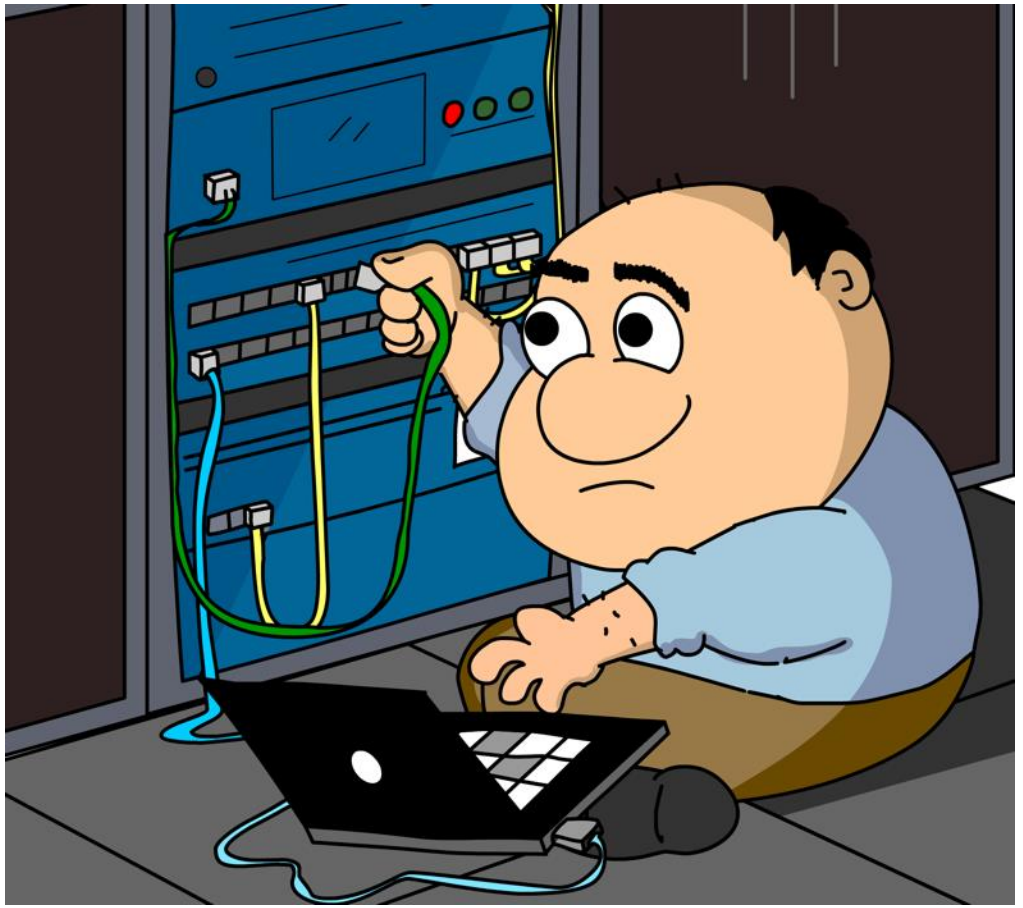
Out of compliance devices are remediated (sometimes)

Governance policies actually prevent future out of compliance changes

Governance – with NSO

- Templates are created directly from “golden devices”
- Compliance Reports audit template against groups of devices
- Template is re-applied to any devices out of compliance
- Policies are created against the configuration database preventing future non-compliant changes

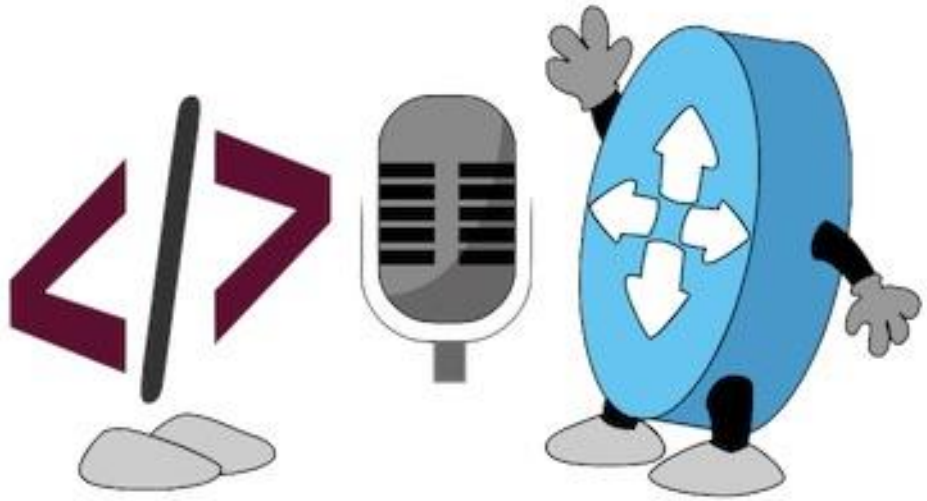
Demo Time!



Summing up

What did we talk about?

- A brief History of tail-f
- Introduction to NSO
- Using NSO as a single CLI and API to solve common configuration and automation challenges



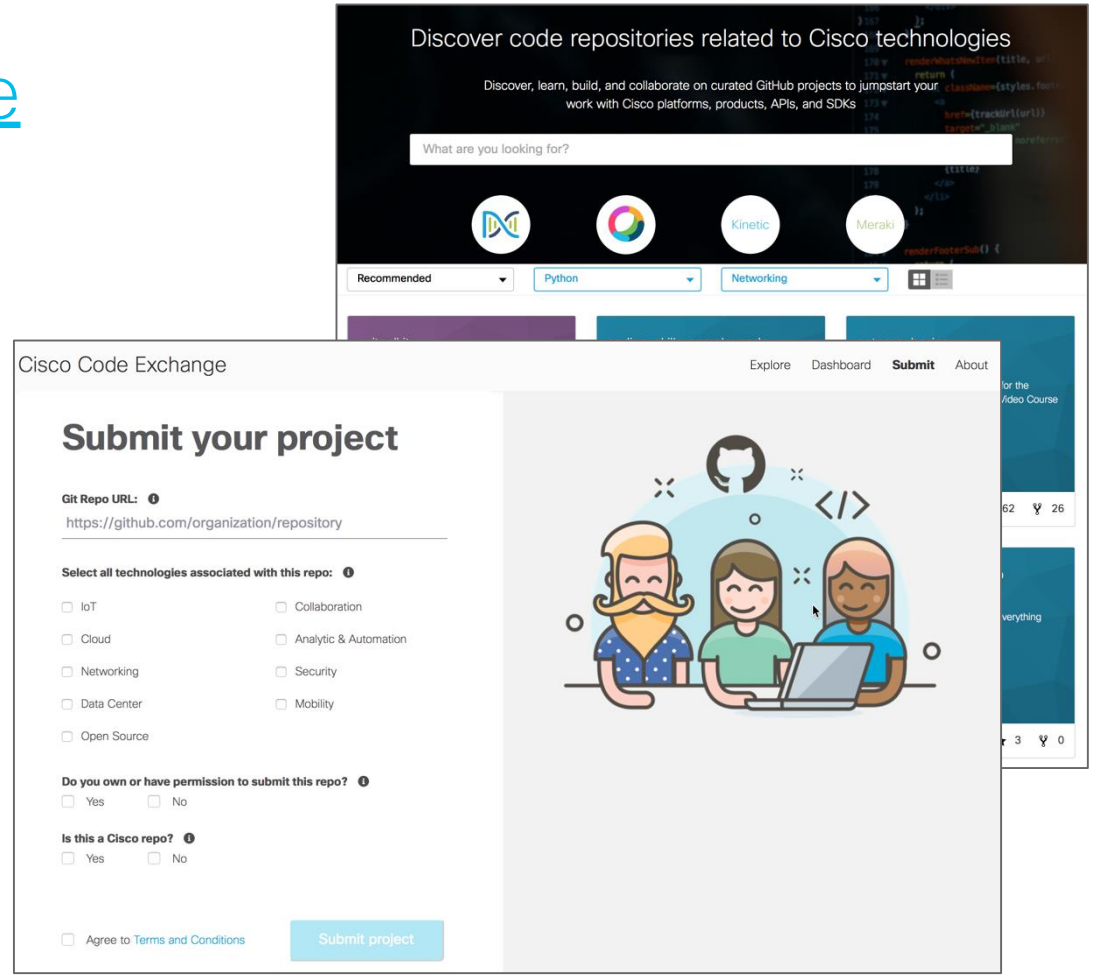
Webinar Resource List

- Try NSO Today!!!
  - https://developer.cisco.com/docs/nso/#!getting-nso
- Code for todays demos
  - https://github.com/NSO-developer/sample-policies
- DevNet Sandboxes
  - Multi-IOS Sandbox with VIRL and NSO! http://cs.co/sbx-multi
  - Accompanying code samples http://cs.co/code-sbx-multi
- Learning Labs
  - https://learninglabs.cisco.com/modules/nso-basics
  - https://learninglabs.cisco.com/modules/nso

NetDevOps Live! Code Exchange Challenge

developer.cisco.com/codeexchange

Build your own network policies using NSO and netsim.



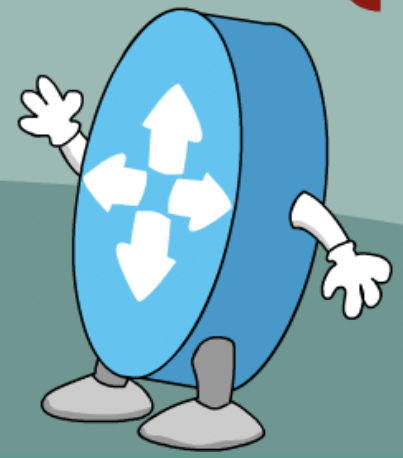
Example: Add NTP server configuration policy.

Looking for more about NetDevOps?

- NetDevOps on DevNet
  developer.cisco.com/netdevops
- NetDevOps Live!
  developer.cisco.com/netdevops/live
- NetDevOps Blogs
  blogs.cisco.com/tag/netdevops
- Network Programmability Basics Video Course
  developer.cisco.com/video/net-prog-basics/

NETDEVOPS {LIVE!}

cisco DEVNET



https://developer.cisco.com/netdevops/live

@netdevopslive