CISCO

You make **possible**

Programmability and Telemetry

in the next generation wireless stack

Jeremy Cohoe
@jeremycohoe

BRKEWN-2050

cisco Live!

Barcelona | January 27-31, 2020

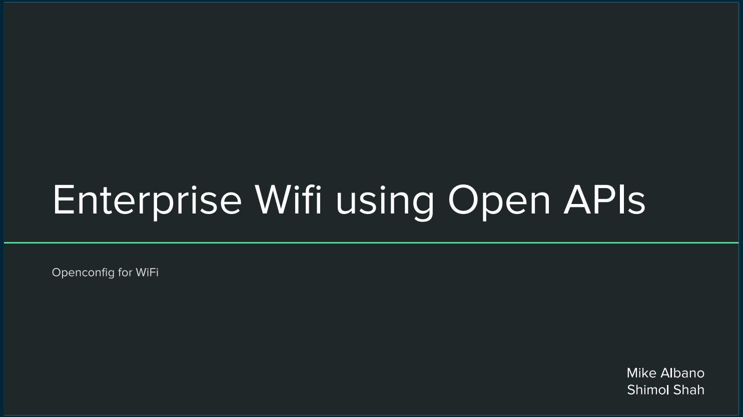
Slides @ http://cs.co/BRKEWN-2050-CLEUR20

Programmability and Telemetry

in the next generation wireless stack

Jeremy Cohoe
@jeremycohoe

BRKEWN-2050



Enterprise Wifi using Open APIs

Openconfig for WiFi

Mike Albano
Shimol Shah

cisco Live!

Barcelona | January 27-31, 2020

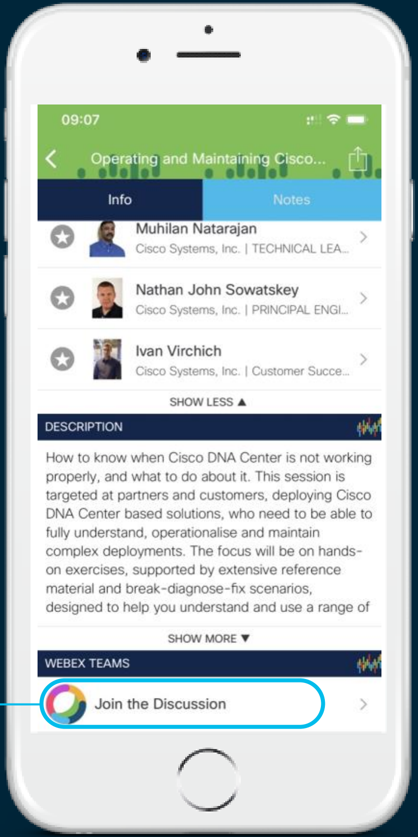
Cisco Webex Teams

Questions?

Use Cisco Webex Teams to chat with the speaker after the session

How

1. Find this session in the Cisco Events Mobile App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space



09:07

Operating and Maintaining Cisco...

Muhilan Natarajan
Cisco Systems, Inc. | TECHNICAL LEA...

Nathan John Sowatskey
Cisco Systems, Inc. | PRINCIPAL ENGI...

Ivan Virchich
Cisco Systems, Inc. | Customer Succe...

DESCRIPTION

How to know when Cisco DNA Center is not working properly, and what to do about it. This session is targeted at partners and customers, deploying Cisco DNA Center based solutions, who need to be able to fully understand, operationalise and maintain complex deployments. The focus will be on hands-on exercises, supported by extensive reference material and break-diagnose-fix scenarios, designed to help you understand and use a range of

WEBEX TEAMS

Wireless streaming telemetry is one way to receive intelligence from the wireless controller about the health and status of your wireless network: clients, access points, and the network and system that connect everything together. Ranging from small to enterprise needs, the YANG model-based telemetry data is easy to consume and provides a better understanding of what's actually happing within your wireless infrastructure, all from the wireless controllers perspective which includes RF metrics and other data otherwise not easily accessible. Kibana, part of the Elastic stack, is used to display charts and graphs of key metrics to provide valuable insights that can be used to better operate and maintain the wireless network

Mike & Shimol (@google.com) will discuss their use case utilizing the gNMI API using the OpenConfig YANG data models for network programmability and telemetry

Agenda

1. Intro to Catalyst 9800 WLC
2. Catalyst 9100 AP and EWC
3. Programmable Interfaces
4. YANG
5. Telemetry Interfaces
6. OpenConfig
7. Demo: OpenConfig
8. Tooling and gRPC Telemetry
9. YangSuite, YangExplorer, pyang

About Jeremy

jcohoe@cisco.com

FYI

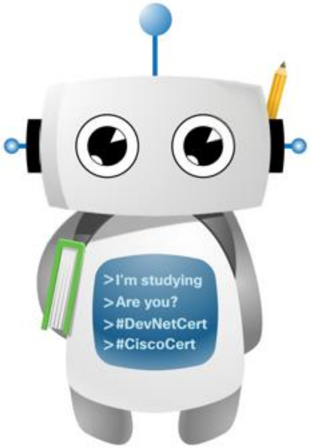
- From Vancouver, BC, Canada
- Amateur Radio Operator, VA7NSA
- Canadian Forces Army – Signals Operator – 4 yrs
- UBC – Wireless Infrastructure – 7 yrs
  - 8k AP, 60k+ concurrent, 200k+ client MAC's
- Cisco – Enterprise Networks – < 3 yrs
  - Programmability and Automation TME

Learn more about the new DevNet Certifications and how you can prepare now!

	Associate Level	Specialist Level	Professional Level	Expert Level
Engineering	CISCO CERTIFIED CCNA	CISCO CERTIFIED SPECIALIST	CISCO CERTIFIED CCNP	CISCO CERTIFIED CCIE
Software	CISCO CERTIFIED DEVNET Associate	CISCO CERTIFIED DEVNET SPECIALIST	CISCO CERTIFIED DEVNET Professional	CISCO CERTIFIED DEVNET Expert Future Offering

Start Here | Upcoming Cisco DevNet Certifications

- Start at Meet DevNet
  - DEVNET-2864: Getting ready for Cisco DevNet Certifications
    Offered daily at 9am, 1pm & 4pm at Meet DevNet
- Attend a brownbag session
  - DEVNET-4099: DevNet Certifications: Bringing software practices & software skills to networking
    Offered daily 12:15-12:45 in the DevNet Zone Theater
- Visit the Learning@Cisco booth
- Scan this code to sign up for the latest updates or go to
  http://cs.co/20eur02



Catalyst Wireless

Introducing Cisco's Next Generation Wireless Stack

Cisco DNA Center

Translate business intent into network policy and capture actionable insights

Cisco DNA Spaces

NEW

Digitize people, spaces and things

Cisco Catalyst 9800
Wireless Controllers

Cisco Catalyst 9100
Access Points

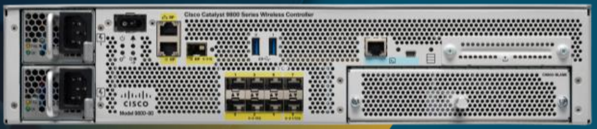
Resilient | Secure | Intelligent

With Innovations in Performance, Security and Analytics

Catalyst 9800 Series Wireless Controllers

DNA Center

Translate business intent into network policy and capture actionable insights with DNA Center



Catalyst 9800-80

Catalyst 9800-40

Catalyst 9800-L

cisco ENCS
vmware
aws
KVM

Catalyst 9800 for Cloud

Catalyst 9800 embedded wireless

for Cat 9k Switch

Aironet and Catalyst Access Points

Works with Cisco Aironet 802.11ac Wave 1 and Wave 2 and 802.11ax C9100 Access Points

ENCS

KVM vmware ESXi

Private Cloud/Virtual

Physical

aws Google Cloud

Public Cloud

Scales up to

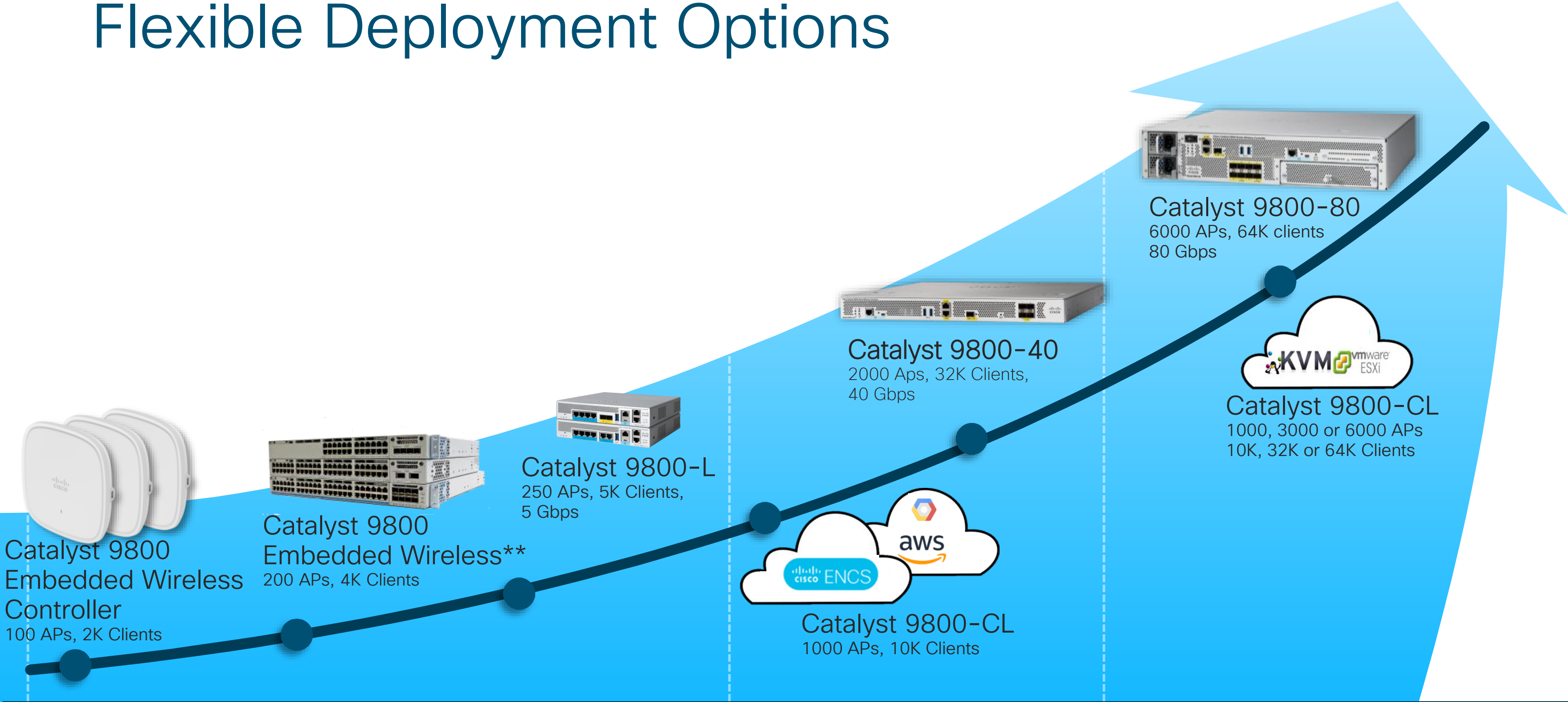
6K APs
64K Clients

Powered by IOS XE

Open & Programmable

Catalyst 9800 – Next Generation Controller

Flexible Deployment Options



Catalyst 9800 Embedded Wireless Controller
100 APs, 2K Clients

Catalyst 9800 Embedded Wireless**
200 APs, 4K Clients

Catalyst 9800-L
250 APs, 5K Clients,
5 Gbps

Catalyst 9800-CL
1000 APs, 10K Clients

Catalyst 9800-40
2000 Aps, 32K Clients,
40 Gbps

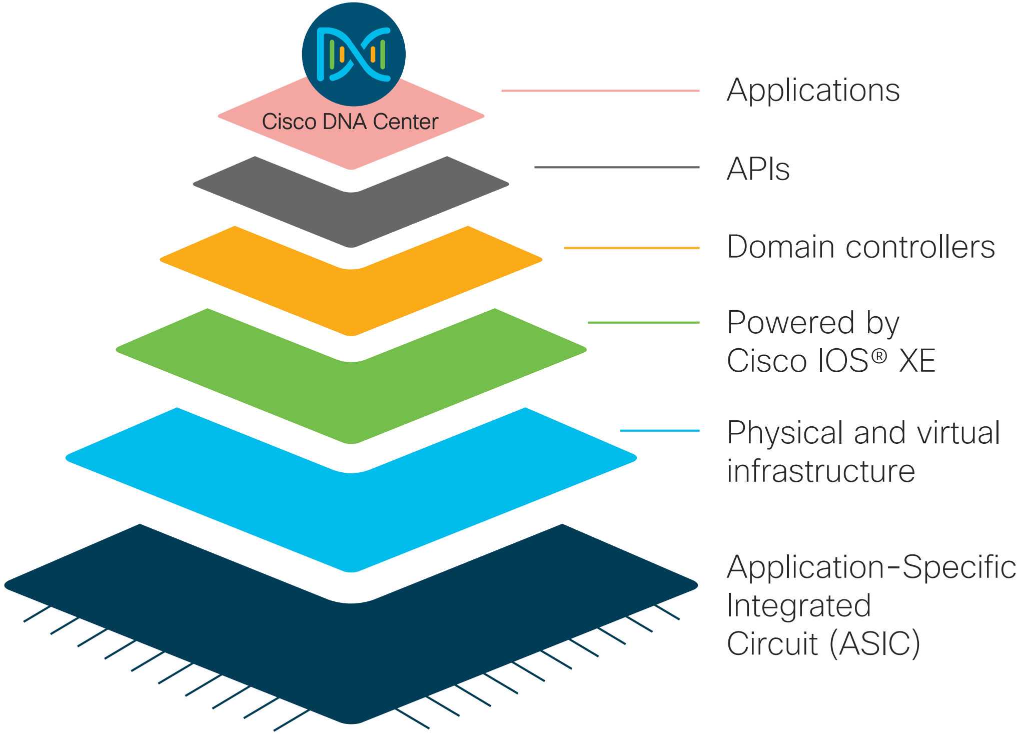
Catalyst 9800-80
6000 APs, 64K clients
80 Gbps

Catalyst 9800-CL
1000, 3000 or 6000 APs
10K, 32K or 64K Clients

Up to 100 APs	Up to 250 APs	Up to 1000 APs	Up to 3000 APs	Up to 6000 APs

Distributed Branch & Small Campus	Medium Campus	Large Campus

Following the principles of intent-based networking


Cisco DNA Center
Applications
APIs
Domain controllers
Powered by Cisco IOS® XE
Physical and virtual infrastructure
Application-Specific Integrated Circuit (ASIC)

- Bridging intent-based networking portfolio
- Open, programmable architecture
- Built-in security, streaming telemetry, and rich analytics

New Cisco Catalyst 9100 Series Access Points

Ideal for small to medium deployments

Mission critical

Best in Class

Powered by Cisco RF ASIC

9115AX

- 4x4 + 4x4
- MU-MIMO, OFDMA
- Spectrum Intelligence
- 1 x 2.5 mGig
- TWT

9117AX

- 8x8 + 4x4
- MU-MIMO, OFDMA (only DL)
- Spectrum intelligence
- 1 x 5 mGig
- TWT
- Integrated Antenna only

9120AX

- 4x4 + 4x4
- Cisco RF ASIC for Next gen CleanAir
- Dual 5GHz, HDX
- RF Layer 1 detail
- IoT ready (Zigbee)
- Application Hosting
- 1 x 2.5 mGig
- TWT

9130AX

- 8x8 + 4x4 or 4x4 + 4x4 + 4x4
- Tri-radio (Dual 5GHz + 2.4GHz), HDX
- Cisco RF ASIC for Next gen CleanAir
- RF Layer 1 details, Application Hosting
- Decrypted data packet iCAP
- IoT ready (Zigbee)
- 8 port Smart Antennas
- 1 x 5 mGig
- TWT

Cisco DNA Assurance with iCAP

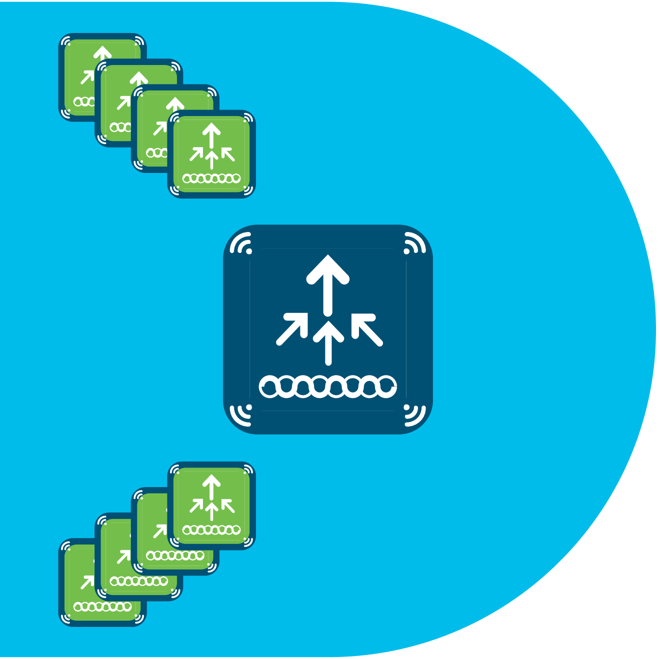
Bluetooth 5

USB

Integrated or external antenna SKUs

EWC on Cisco Catalyst access points

Ready for enterprise deployments



Runs 9800 Series Cisco IOS® XE wireless controller on Cisco Catalyst access points	Modern OS, scalable, open and programmable, supports telemetry
Supports advanced enterprise feature set	HA, SMU, adaptive wireless IPS (aWIPS), Cisco Umbrella™, NetFlow, ICAP
Flexible management options	Use mobile app, WebUI, and Cisco DNA Center to deploy, manage, and monitor
Investment protection	Migrate access points to controller for more than 100 access points

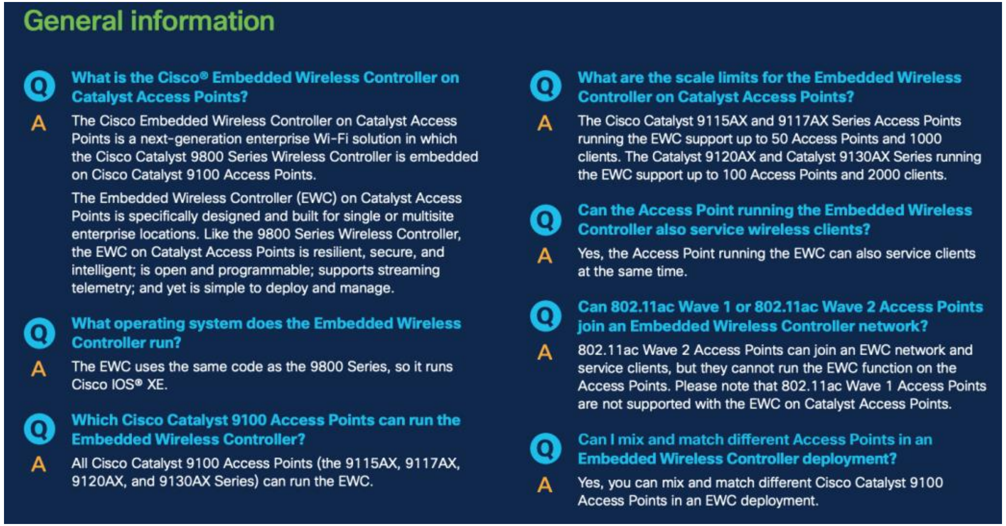
EWC is the Catalyst “Mobility Express”

Cisco Embedded Wireless Controller on Catalyst Access Points White Paper

What are the key difference between Mobility Express and EWC?

The key differences are – EWC is IOS-XE based. It supports advanced enterprise feature set like SMU, APDP, APSP, Intelligent Capture which Mobility Express does not support.

EWC has High Availability with active-standby redundancy with less than 10seconds of downtime. EWC also enables customers to use them for Site Surveys.



General information

Q **What is the Cisco® Embedded Wireless Controller on Catalyst Access Points?**

A The Cisco Embedded Wireless Controller on Catalyst Access Points is a next-generation enterprise Wi-Fi solution in which the Cisco Catalyst 9800 Series Wireless Controller is embedded on Cisco Catalyst 9100 Access Points.

The Embedded Wireless Controller (EWC) on Catalyst Access Points is specifically designed and built for single or multisite enterprise locations. Like the 9800 Series Wireless Controller, the EWC on Catalyst Access Points is resilient, secure, and intelligent; is open and programmable; supports streaming telemetry; and yet is simple to deploy and manage.

Q **What operating system does the Embedded Wireless Controller run?**

A The EWC uses the same code as the 9800 Series, so it runs Cisco IOS® XE.

Q **Which Cisco Catalyst 9100 Access Points can run the Embedded Wireless Controller?**

A All Cisco Catalyst 9100 Access Points (the 9115AX, 9117AX, 9120AX, and 9130AX Series) can run the EWC.

Q **What are the scale limits for the Embedded Wireless Controller on Catalyst Access Points?**

A The Cisco Catalyst 9115AX and 9117AX Series Access Points running the EWC support up to 50 Access Points and 1000 clients. The Catalyst 9120AX and Catalyst 9130AX Series running the EWC support up to 100 Access Points and 2000 clients.

Q **Can the Access Point running the Embedded Wireless Controller also service wireless clients?**

A Yes, the Access Point running the EWC can also service clients at the same time.

Q **Can 802.11ac Wave 1 or 802.11ac Wave 2 Access Points join an Embedded Wireless Controller network?**

A 802.11ac Wave 2 Access Points can join an EWC network and service clients, but they cannot run the EWC function on the Access Points. Please note that 802.11ac Wave 1 Access Points are not supported with the EWC on Catalyst Access Points.

Q **Can I mix and match different Access Points in an Embedded Wireless Controller deployment?**

A Yes, you can mix and match different Cisco Catalyst 9100 Access Points in an EWC deployment.

https://www.cisco.com/c/dam/en/us/products/collateral/wireless/catalyst-9800-series-wireless-controllers/q-and-a-c67-743152.pdf

http://cs.co/ewcwhitepaper

New Catalyst aesthetically redesigned APs

New Pininfarina design (Smaller in size and lighter in weight)

Compact design without Compromise

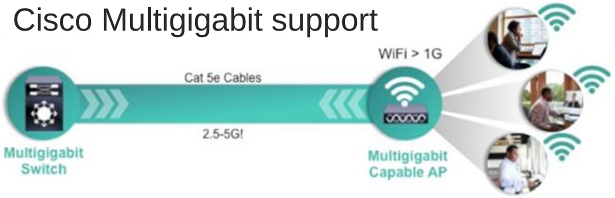
Designed to operate using 802.3at (30W)

802.3af (15.4W) supported

New external antennas that match the design of the Access Point

NEW Single insertion connector for faster installs using 8x8 DART connector

Cisco Multigigabit support

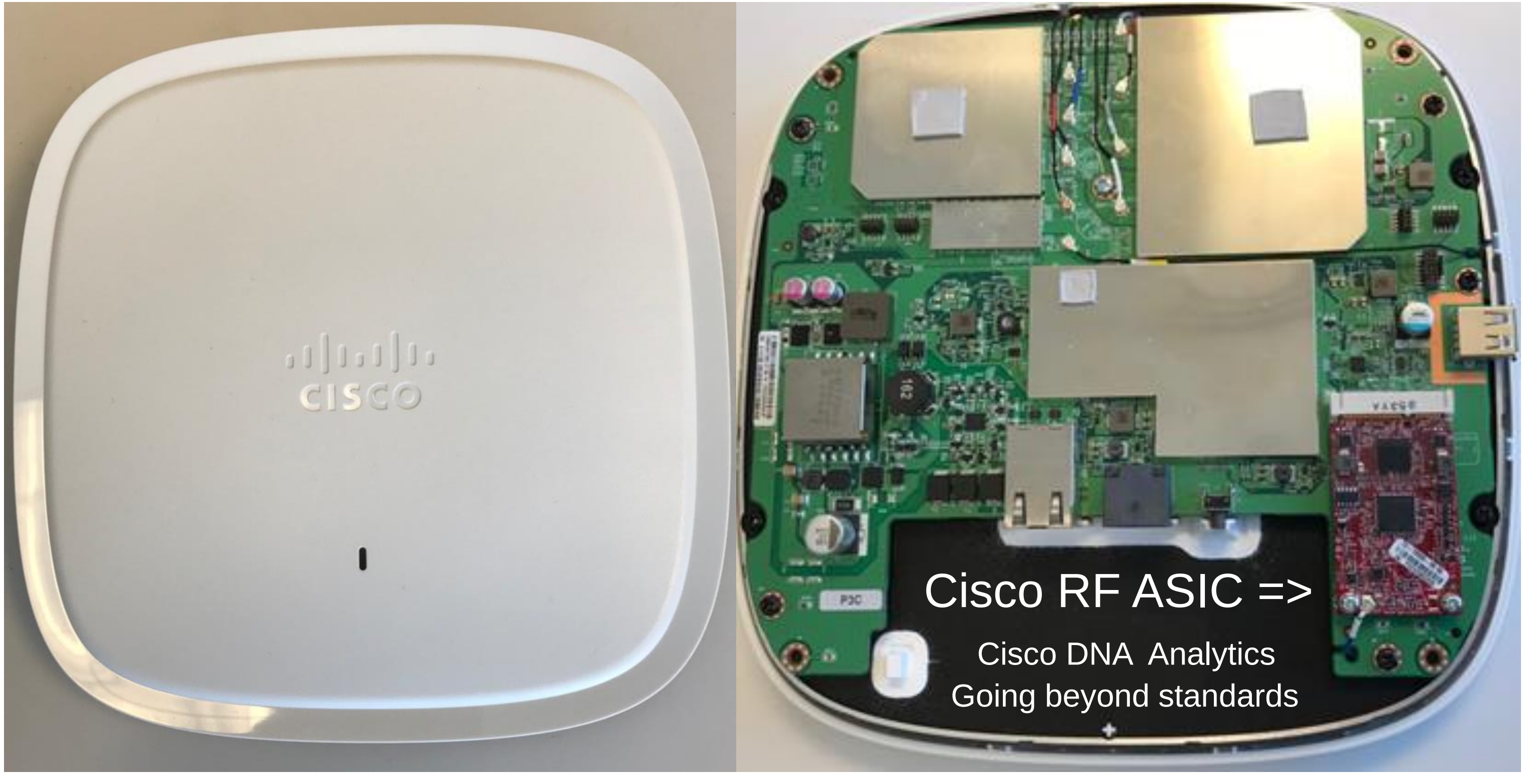


Backward compatibility modes to support 802.3af (15.4W) modes when needed.

Easy to deploy with Aironet series mounting brackets

Catalyst 9120/9130 has Cisco RF ASIC

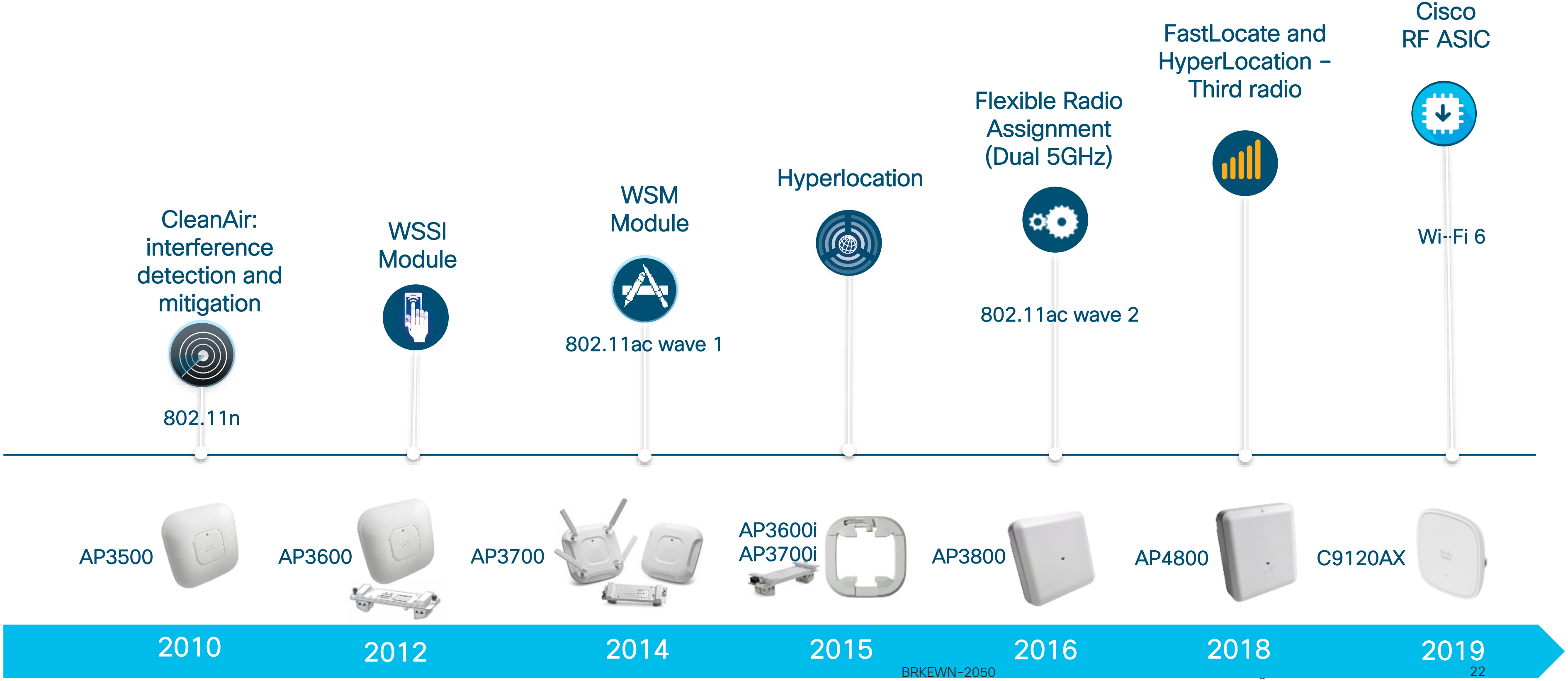
Cisco RF ASIC



Cisco RF ASIC =>

Cisco DNA Analytics
Going beyond standards

Cisco RF ASIC – A long Rich History of Innovation



CleanAir: interference detection and mitigation

802.11n

AP3500

2010

WSSI Module

AP3600

2012

WSM Module

802.11ac wave 1

AP3700

2014

Hyperlocation

AP3600i
AP3700i

2015

Flexible Radio Assignment (Dual 5GHz)

802.11ac wave 2

AP3800

2016

FastLocate and HyperLocation – Third radio

AP4800

2018

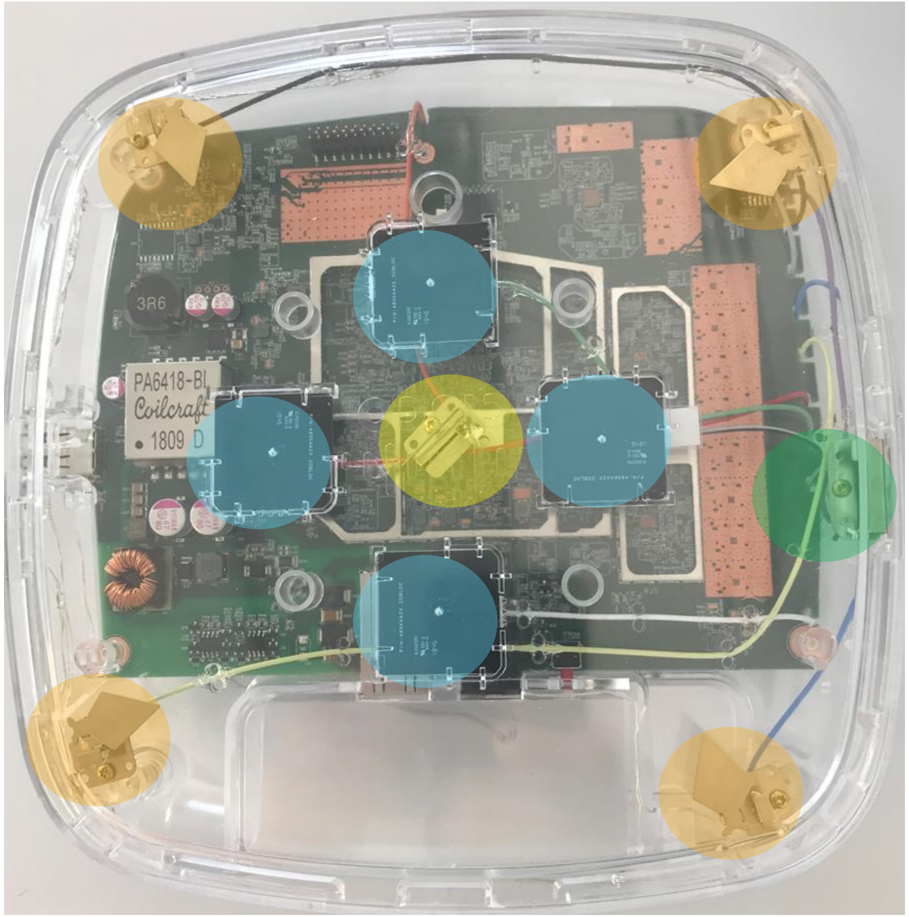
Cisco RF ASIC

Wi--Fi 6

C9120AX

2019

Catalyst 9130AXI 8x8 Antenna System



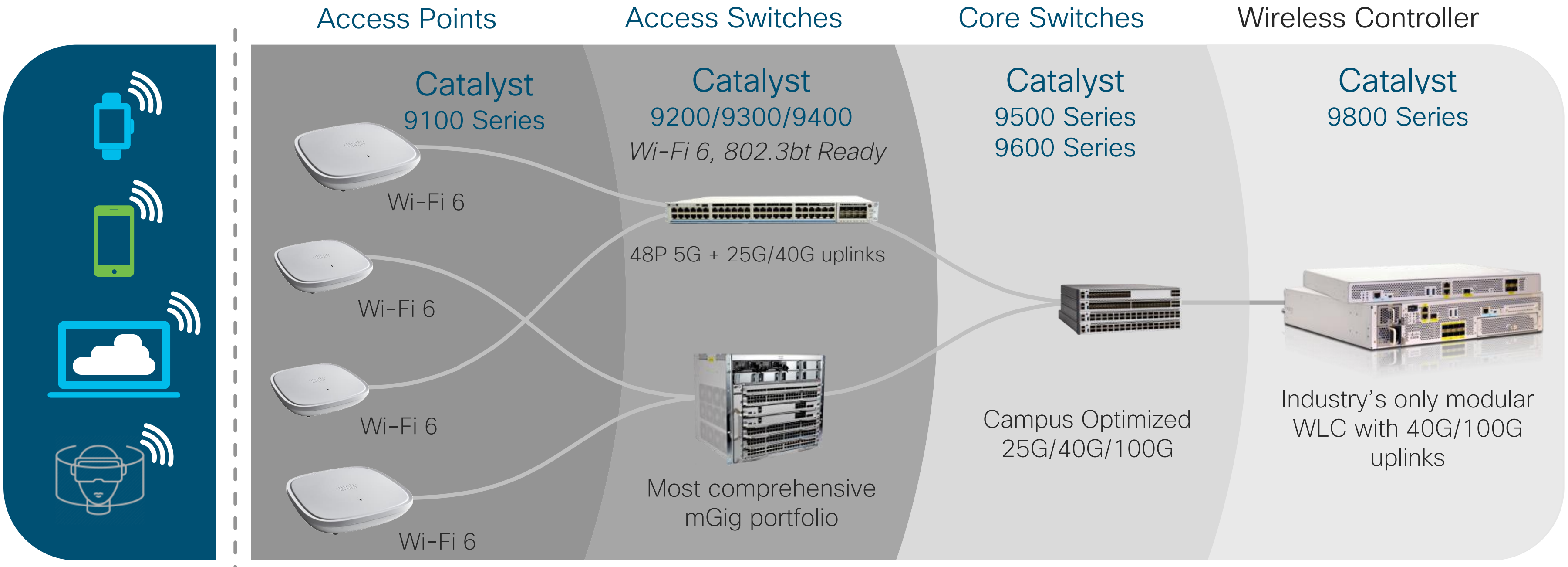
🟠 (4) Dual Band “Macro” antennas
2.4 GHz @ 4 dBi
5.0 GHz @ 5 dBi

🔵 (4) 5 GHz “Micro” antennas
5 GHz @ 5 dBi

🟢 (1) IOT Antenna
2.4 GHz @ 2.5 dBi

🟡 (1) RF ASIC Antenna
2.4 GHz @ 4.5 dBi
5.0 GHz @ 5 dBi

End-to-end Wi-Fi 6 leadership enabling next-generation mobility



Access Points

Catalyst 9100 Series

Wi-Fi 6

Wi-Fi 6

Wi-Fi 6

Wi-Fi 6

Access Switches

Catalyst 9200/9300/9400

Wi-Fi 6, 802.3bt Ready

48P 5G + 25G/40G uplinks

Most comprehensive mGig portfolio

Core Switches

Catalyst 9500 Series 9600 Series

Campus Optimized 25G/40G/100G

Wireless Controller

Catalyst 9800 Series

Industry's only modular WLC with 40G/100G uplinks

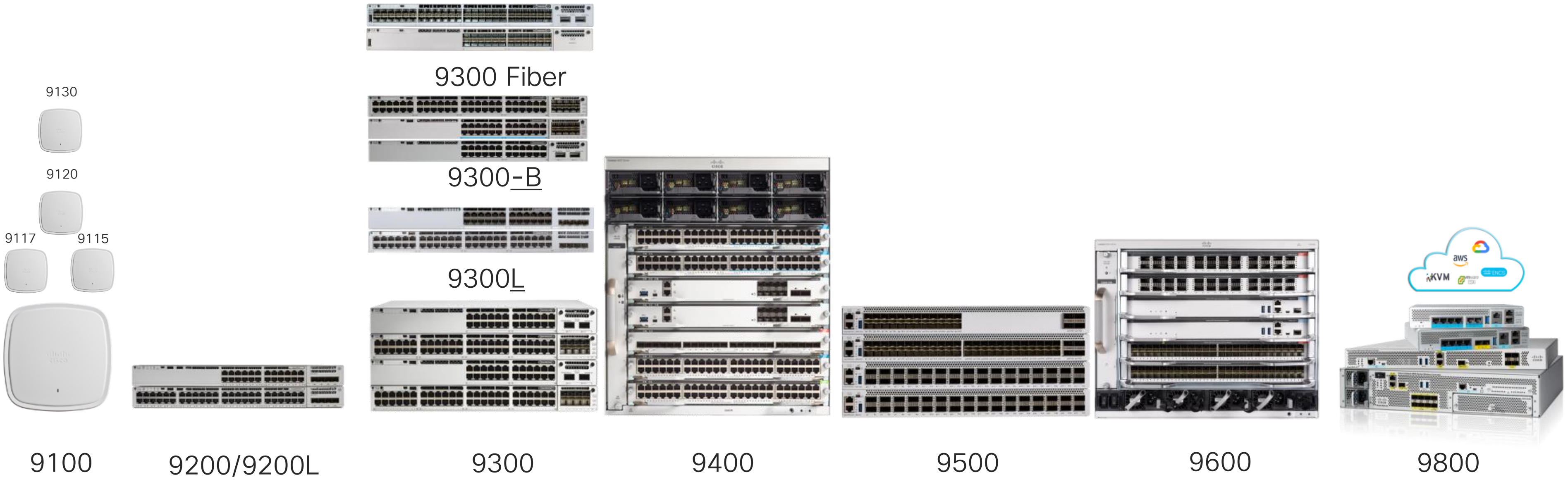
← The Full Experience End to End →

Built for intent-based networking

Automation

Security

Analytics



The Catalyst 9K Family

Cisco Recommended Releases and Interop

Catalyst 9800 and 3504/5520/8540 AireOS Wireless Controllers

Access Points	IOS-XE	AireOS	DNA-C	Prime	CMX	ISE
C9115AX, C9117AX , C9120AX	16.12.1s	8.10	1.3.1.2	3.7	10.6.2	2.3 2.4 2.6
C9130AX	16.12.1s with AP DP	N/A	1.3.1.2	3.7	10.6.2	2.3 2.4 2.6
C9120AX-E C9130AX	16.12.2	8.10	1.3.2	3.7	10.6.2	2.3 2.4 2.6
Wave 2 APs	16.12.1s	8.5MR5	1.3.1.2	3.7	10.6.2	2.3 2.4 2.6

Catalyst 9800-SW**
200 APs, 4K Clients

aws
ENCS
KVM vmware ESXi

Catalyst 9800-CL***
1000 APs, 10K Clients

aws
KVM vmware ESXi

Catalyst 9800-CL
3000 APs, 32K Clients

aws
KVM vmware ESXi

Catalyst 9800-CL
6000 APs, 64K Clients^

100 APs | 250 APs | 1000 APs | 2000 APs | 3000 APs | 6000 APs

Catalyst 9100
100 APs, 2K Clients

Catalyst 9800-L
250 APs, 5K Clients, 5 Gbps

Catalyst 9800-40
2000 APs, 32K Clients, 40 Gbps

Catalyst 9800-80
6000 APs, 64K Clients, 80 Gbps

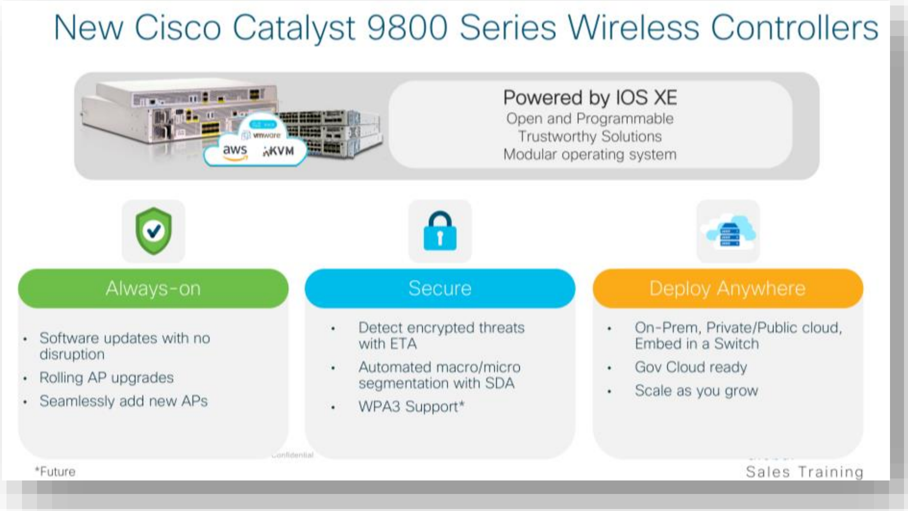
One IOS XE based Software – Deploy & Scale the way you want

cisco Live!

Programmable Interfaces

Cisco IOS XE: Network OS for the Enterprise

Switching | Wireless | Routing | IOT



New Cisco Catalyst 9800 Series Wireless Controllers

vmware
aws
KVM

Powered by IOS XE
Open and Programmable
Trustworthy Solutions
Modular operating system

Always-on

- Software updates with no disruption
- Rolling AP upgrades
- Seamlessly add new APs

Secure

- Detect encrypted threats with ETA
- Automated macro/micro segmentation with SDA
- WPA3 Support*

Deploy Anywhere

- On-Prem, Private/Public cloud, Embed in a Switch
- Gov Cloud ready
- Scale as you grow

*Future

Confidential

Sales Training

IOS XE
16.12.2s
17.1.1

Programmability Overview

AWS/GCP Run File

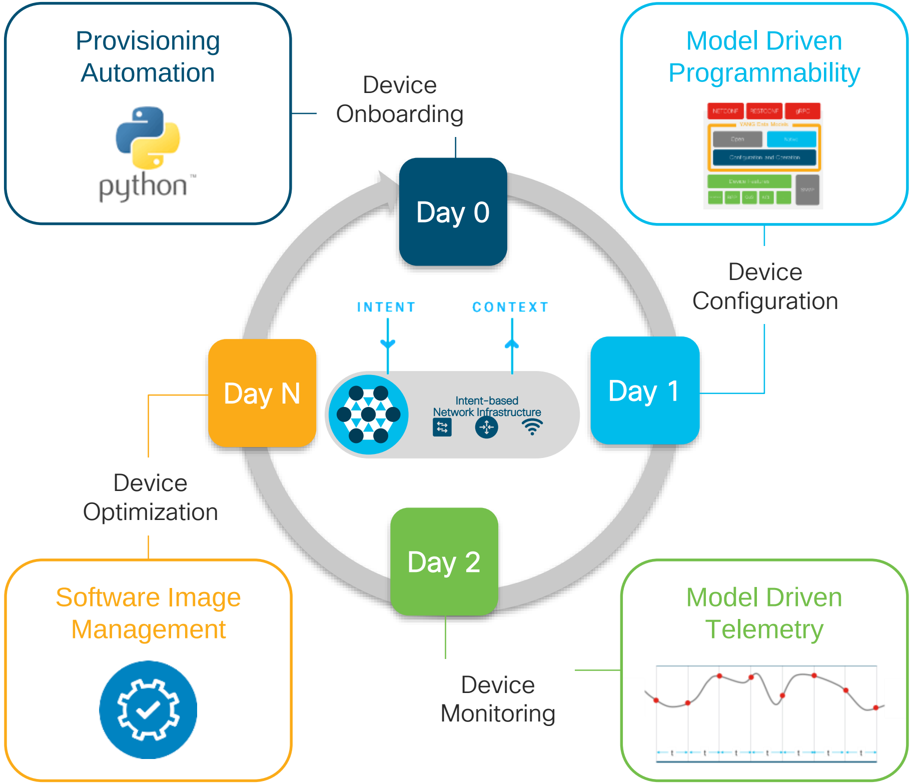
ESXi/KVM Script

Zero Touch (ZTP)

NETwork CONFiguration Protocol (NETCONF)

RESTCONF

YANG Data Models



Guest Shell Linux

On-Box Python API

NETCONF

gRPC

gNMI

IOS XE Programmability and Telemetry “Stack”

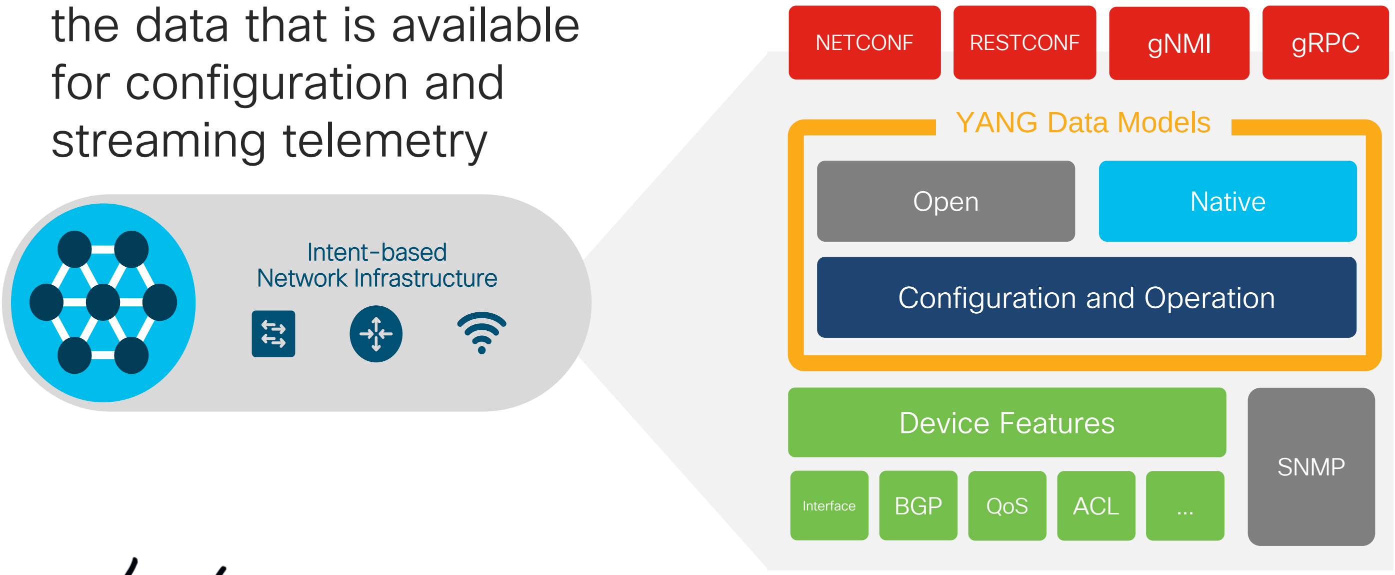
CLI

SNMP

WebUI

The NETCONF, RETCONF, gNMI and gRPC are **programmatic** interfaces that provide **<u>additional</u>** methods for interfacing with the device

YANG data models define the data that is available for configuration and streaming telemetry



IETF

OPENCONFIG

IEEE

CISCO

NETCONF Interface

NETCONF RESTCONF gNMI gRPC

"NETCONF is a protocol defined by the IETF to install, manipulate, and delete the configuration of network devices"

V 1.0

- RFC 4741 Base NETCONF Protocol
- RFC 4742 NETCONF over SSH

V 1.1

- RFC 6241 Base NETCONF Protocol
- RFC 6242 NETCONF over SSH

Extensions

- RFC 5277 Notifications
- RFC 5717 Partial Locking
- RFC 6243 With defaults
- RFC 6020 YANG

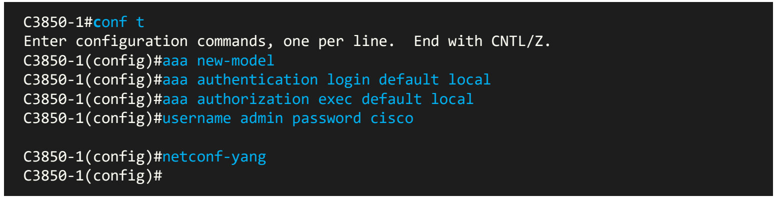
2006 2010 2011

https://tools.ietf.org/html/rfc6241

- **Transactional**
  - Either all configuration is applied or nothing
  - Avoids inconsistent state
  - Both at Single Device and Network-wide level
- **Error** Management
  - OK or error code
- **Capability Exchange**

```
ssh -p 830 admin@127.0.0.1 -s netconf
```

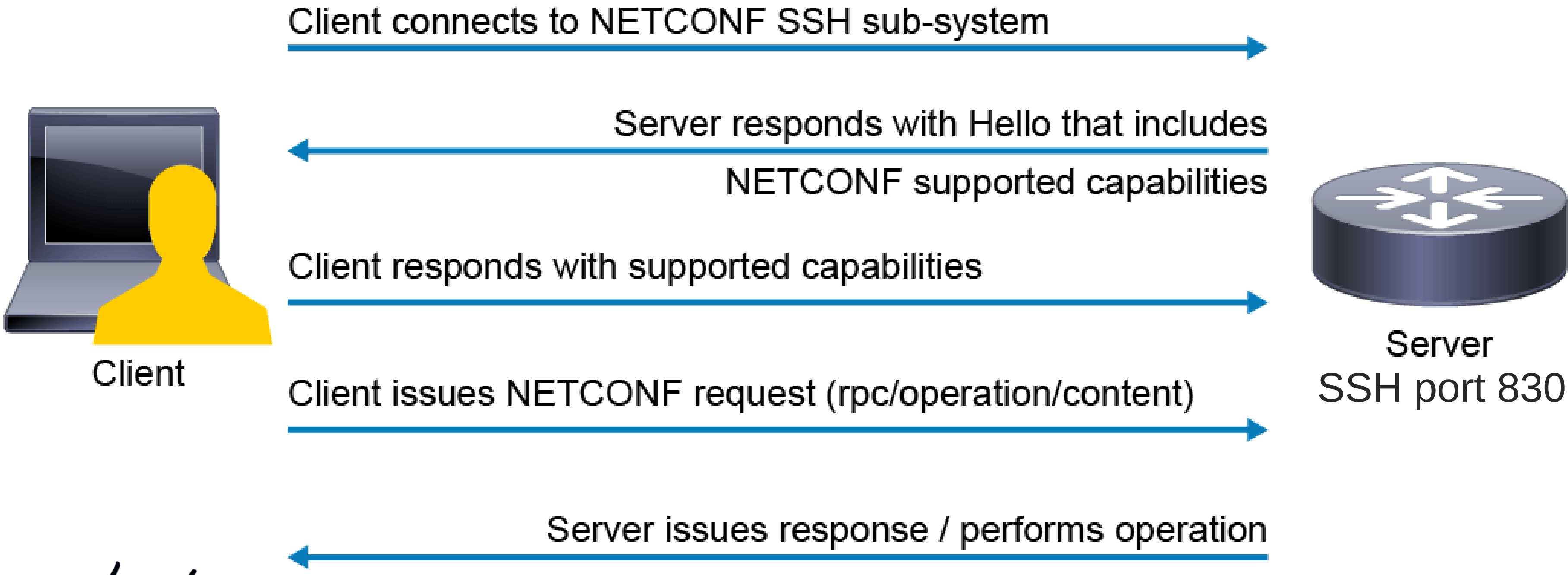
- **Models Download** from a Device



```
C3850-1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
C3850-1(config)#aaa new-model
C3850-1(config)#aaa authentication login default local
C3850-1(config)#aaa authorization exec default local
C3850-1(config)#username admin password cisco

C3850-1(config)#netconf-yang
C3850-1(config)#
```

NETCONF Transport

- NETCONF over SSH



Client connects to NETCONF SSH sub-system

Server responds with Hello that includes NETCONF supported capabilities

Client responds with supported capabilities

Client issues NETCONF request (rpc/operation/content)

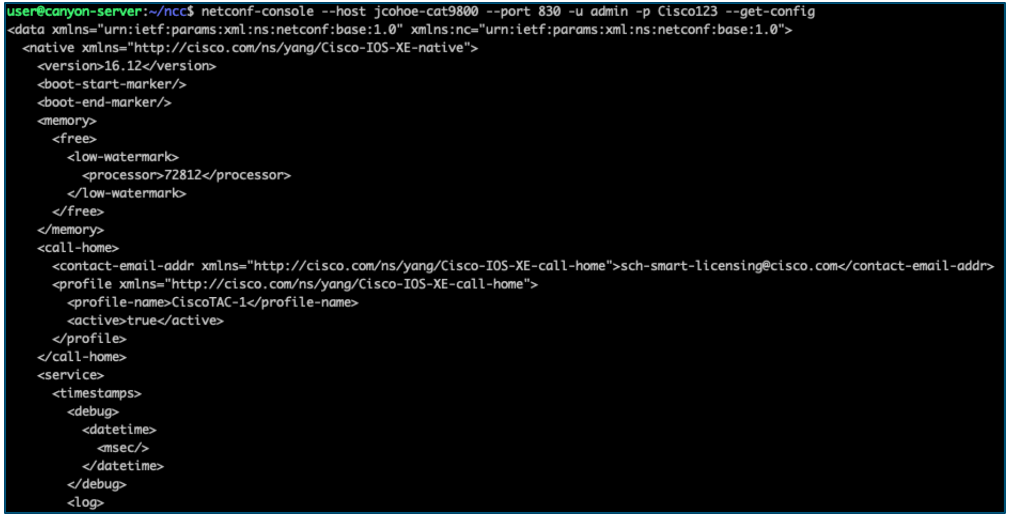
Server issues response / performs operation

Client

Server
SSH port 830

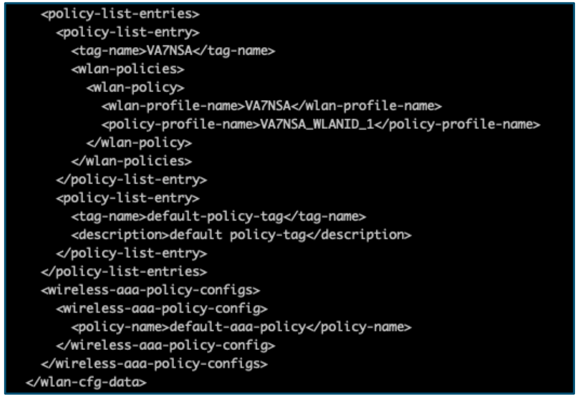
NETCONF Get Running Config

```
$ netconf-console --host ewc --port 830 -u admin -p Cisco123 --get-config
      --x wlan-cfg-data/wlan-cfg-entries
```



```
user@canyon-server:~/ncc$ netconf-console --host jcohoe-cat9800 --port 830 -u admin -p Cisco123 --get-config
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
    <version>16.12</version>
    <boot-start-marker/>
    <boot-end-marker/>
    <memory>
      <free>
        <low-watermark>
          <processor>72812</processor>
        </low-watermark>
      </free>
    </memory>
    <call-home>
      <contact-email-addr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-call-home">sch-smart-licensing@cisco.com</contact-email-addr>
      <profile xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-call-home">
        <profile-name>CiscoTAC-1</profile-name>
        <active>true</active>
      </profile>
    </call-home>
    <service>
      <timestamps>
        <debug>
          <datetime>
            <msec/>
          </datetime>
        </debug>
        <log>
```

This is an easy to way to backup and restore the config - instead of CLI



```
    <policy-list-entries>
      <policy-list-entry>
        <tag-name>VA7NSA</tag-name>
        <wlan-policies>
          <wlan-policy>
            <wlan-profile-name>VA7NSA</wlan-profile-name>
            <policy-profile-name>VA7NSA_WLANID_1</policy-profile-name>
          </wlan-policy>
        </wlan-policies>
      </policy-list-entry>
      <policy-list-entry>
        <tag-name>default-policy-tag</tag-name>
        <description>default policy-tag</description>
      </policy-list-entry>
    </policy-list-entries>
    <wireless-aaa-policy-configs>
      <wireless-aaa-policy-config>
        <policy-name>default-aaa-policy</policy-name>
      </wireless-aaa-policy-config>
    </wireless-aaa-policy-configs>
  </wlan-cfg-data>
```

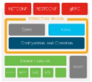
CLI show run is about 500 lines of commands
When retreived over NETCONF the XML meta-data is about 3000 lines

Now supporting SSH Keys for NETCONF Authentication

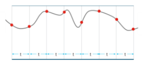
YANG Data Models

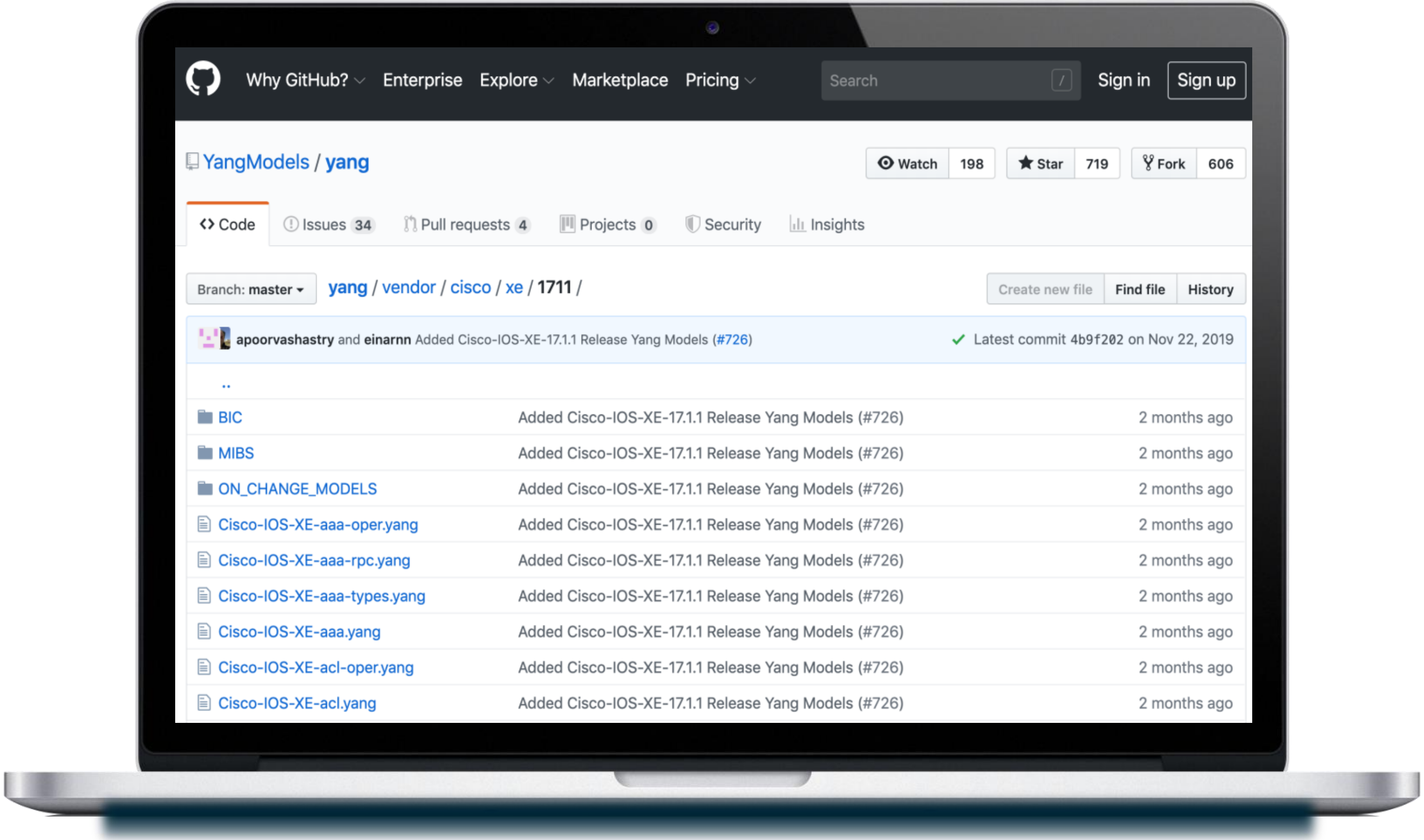
IOS XE YANG Models on Github

Model Driven Programmability



Model Driven Telemetry





YangModels / yang

yang / vendor / cisco / xe / 1711 /

apoorvashastry and einarnn Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726) ✓ Latest commit 4b9f202 on Nov 22, 2019

..		
BIC	Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726)	2 months ago
MIBS	Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726)	2 months ago
ON_CHANGE_MODELS	Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726)	2 months ago
Cisco-IOS-XE-aaa-oper.yang	Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726)	2 months ago
Cisco-IOS-XE-aaa-rpc.yang	Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726)	2 months ago
Cisco-IOS-XE-aaa-types.yang	Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726)	2 months ago
Cisco-IOS-XE-aaa.yang	Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726)	2 months ago
Cisco-IOS-XE-acl-oper.yang	Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726)	2 months ago
Cisco-IOS-XE-acl.yang	Added Cisco-IOS-XE-17.1.1 Release Yang Models (#726)	2 months ago

https://github.com/YangModels/yang/tree/master/vendor/cisco/xe

Open | Native

Configuration and Operation

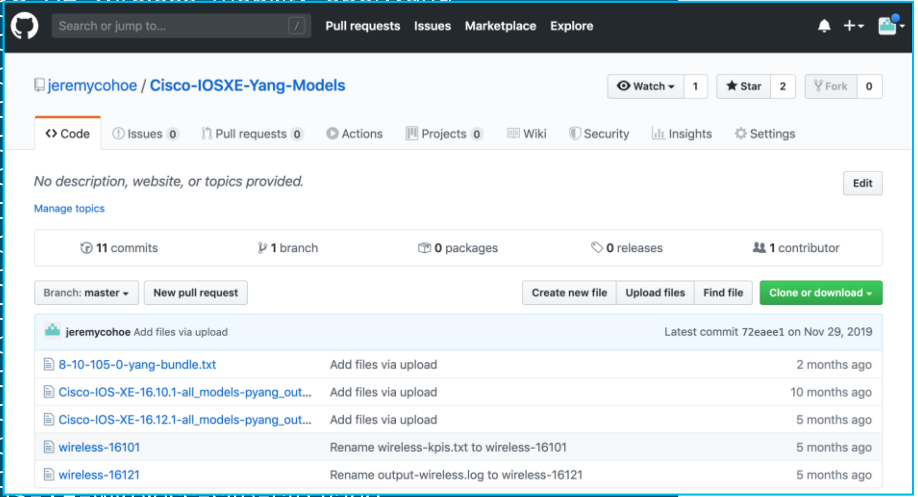
Wireless YANG Models

Of the 215 Cisco Native models 42 models are wireless:

pyang tree output with KPI's:
https://github.com/jeremycohoe/Cisco-IOSXE-Yang-Models

Cisco-IOS-XE-wireless-access-point-oper.yang
Cisco-IOS-XE-wireless-ap-cfg.yang
Cisco-IOS-XE-wireless-ap-types.yang
Cisco-IOS-XE-wireless-apf-cfg.yang
Cisco-IOS-XE-wireless-client-oper.yang
Cisco-IOS-XE-wireless-client-types.yang
Cisco-IOS-XE-wireless-cts-sxp-cfg.yang
Cisco-IOS-XE-wireless-cts-sxp-oper.yang
Cisco-IOS-XE-wireless-dot11-cfg.yang
Cisco-IOS-XE-wireless-enum-types.yang
Cisco-IOS-XE-wireless-events-oper.yang
Cisco-IOS-XE-wireless-fabric-cfg.yang
Cisco-IOS-XE-wireless-flex-cfg.yang
Cisco-IOS-XE-wireless-fqdn-cfg.yang
Cisco-IOS-XE-wireless-fqdn-oper.yang
Cisco-IOS-XE-wireless-general-cfg.yang
Cisco-IOS-XE-wireless-hyperlocation-oper.yang
Cisco-IOS-XE-wireless-lisp-agent-oper.yang
Cisco-IOS-XE-wireless-location-cfg.yang
Cisco-IOS-XE-wireless-location-oper.yang
Cisco-IOS-XE-wireless-mcast-oper.yang

Cisco-IOS-XE-wireless-mesh-cfg.yang
Cisco-IOS-XE-wireless-mesh-oper.yang
Cisco-IOS-XE-wireless-mobility-cfg.yang
Cisco-IOS-XE-wireless-mobility-oper.yang
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IO
Cisco-IOS-XE-wireless-site-cfg.yang
Cisco-IOS-XE-wireless-types.yang
Cisco-IOS-XE-wireless-wlan-cfg.yang



jeremycohoe / Cisco-IOSXE-Yang-Models

No description, website, or topics provided.

11 commits | 1 branch | 0 packages | 0 releases | 1 contributor

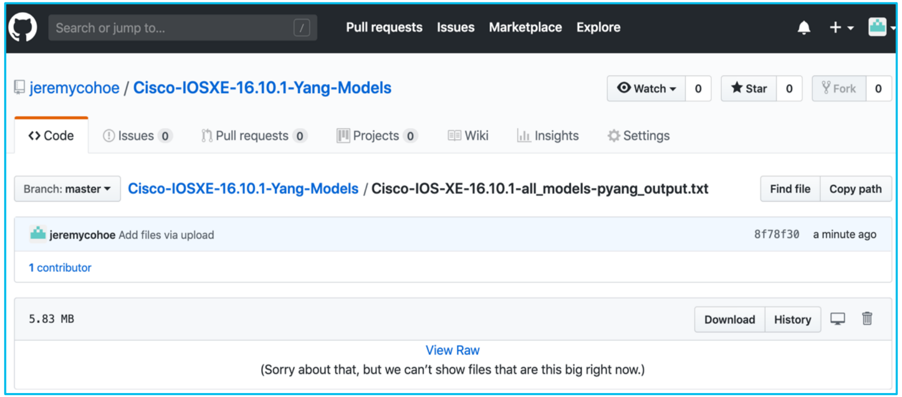
jeremycohoe Add files via upload — Latest commit 72eaee1 on Nov 29, 2019

8-10-105-0-yang-bundle.txt	Add files via upload	2 months ago
Cisco-IOS-XE-16.10.1-all_models-pyang_out...	Add files via upload	10 months ago
Cisco-IOS-XE-16.12.1-all_models-pyang_out...	Add files via upload	5 months ago
wireless-16101	Rename wireless-kpis.txt to wireless-16101	5 months ago
wireless-16121	Rename output-wireless.log to wireless-16121	5 months ago

YANG Models KPI Details

pyang KPI tree output:
https://github.com/jeremycohoe/Cisco-IOSXE-Yang-Models

There are 130,000+ KPI’s available from the 350+ YANG models
Here is an (easy) way to search all models and all KPI’s



jeremycohoe / Cisco-IOSXE-16.10.1-Yang-Models

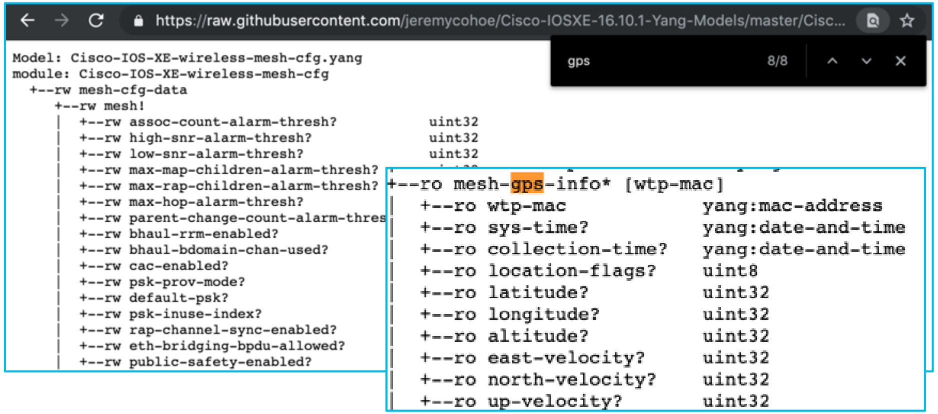
Cisco-IOSXE-16.10.1-Yang-Models / Cisco-IOS-XE-16.10.1-all_models-pyang_output.txt

jeremycohoe Add files via upload 8f78f30 a minute ago

1 contributor

5.83 MB

(Sorry about that, but we can’t show files that are this big right now.)



https://raw.githubusercontent.com/jeremycohoe/Cisco-IOSXE-16.10.1-Yang-Models/master/Cisc...

```
Model: Cisco-IOS-XE-wireless-mesh-cfg.yang
module: Cisco-IOS-XE-wireless-mesh-cfg
  +--rw mesh-cfg-data
     +--rw mesh!
     |  +--rw assoc-count-alarm-thresh?          uint32
     |  +--rw high-snr-alarm-thresh?             uint32
     |  +--rw low-snr-alarm-thresh?              uint32
     |  +--rw max-map-children-alarm-thresh?
     |  +--rw max-rap-children-alarm-thresh?
     |  +--rw max-hop-alarm-thresh?
     |  +--rw parent-change-count-alarm-thres
     |  +--rw bhaul-rrm-enabled?
     |  +--rw bhaul-bdomain-chan-used?
     |  +--rw cac-enabled?
     |  +--rw psk-prov-mode?
     |  +--rw default-psk?
     |  +--rw psk-inuse-index?
     |  +--rw rap-channel-sync-enabled?
     |  +--rw eth-bridging-bpdu-allowed?
     |  +--rw public-safety-enabled?
```

```
+--ro mesh-gps-info* [wtp-mac]
|  +--ro wtp-mac               yang:mac-address
|  +--ro sys-time?             yang:date-and-time
|  +--ro collection-time?      yang:date-and-time
|  +--ro location-flags?       uint8
|  +--ro latitude?             uint32
|  +--ro longitude?            uint32
|  +--ro altitude?             uint32
|  +--ro east-velocity?        uint32
|  +--ro north-velocity?       uint32
|  +--ro up-velocity?          uint32
```

Use case: “Is the GPS data from the outdoor wireless AP available from YANG?”

Config vs Operational YANG data models

https://github.com/YangModels/yang
https://github.com/openconfig

Config data

- What the device is told to do
- It's the way you express intent

Examples:
switch> show run interface Loopback0
switch(config)# interface Loopback0

Cisco-IOS-XE-Wireless: Config models

ap
apf
cts-sxp
dot11
fabric
flex
fqdn
general
location
mesh
mobility
mstream
rf
rfid
rogue
rrm
security
site
wlan

Operational data

- What the device is actually doing
- It's what you see from most show commands

Examples:
switch> show interface Loopback0
'*snmpget*' results

Cisco-IOS-XE-Wireless: Oper models

access-point
client
fqdn
lisp-agent
mcast
mesh
mobility
nmsp
rf-profile
rfid
rogue
rrm

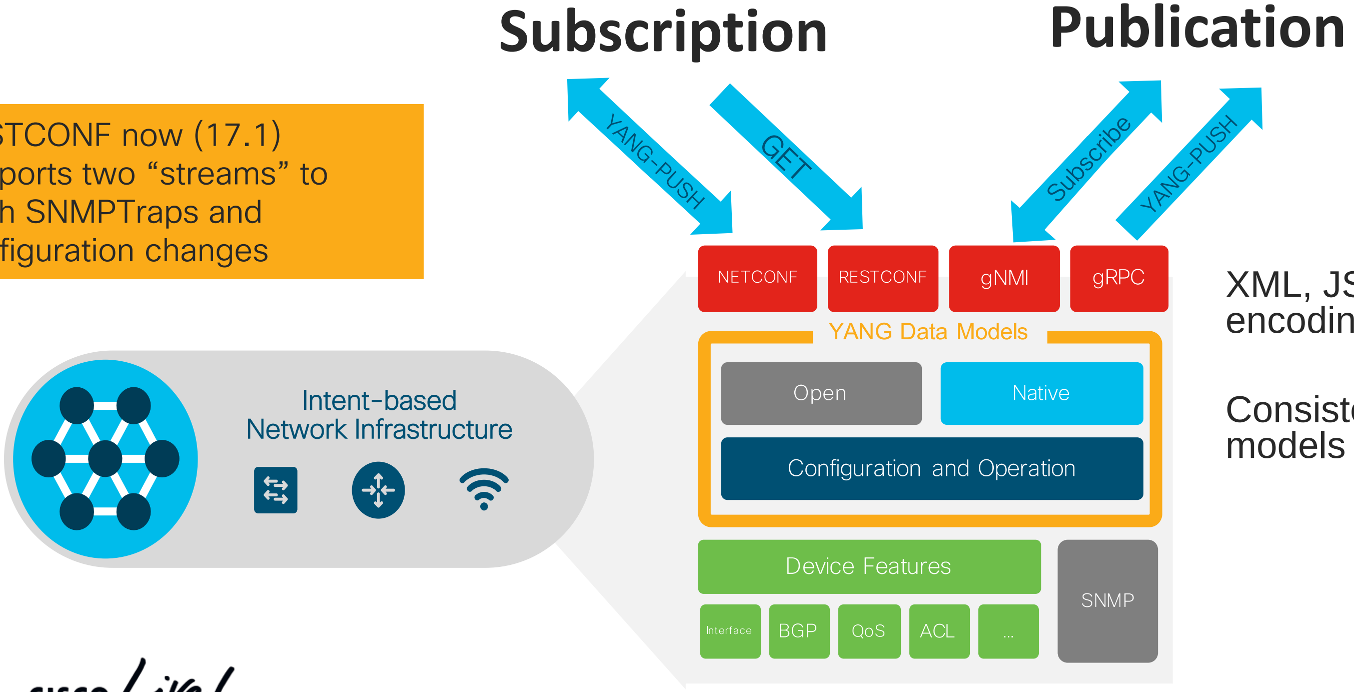
Telemetry Interfaces

Model Driven Telemetry

Model Driven Telemetry

Dial In: Collector establishes a connection to the device then subscribes to telemetry
Dial Out: Telemetry is pushed from the device to the collector based off configuration

RESTCONF now (17.1) supports two “streams” to push SNMPTraps and configuration changes



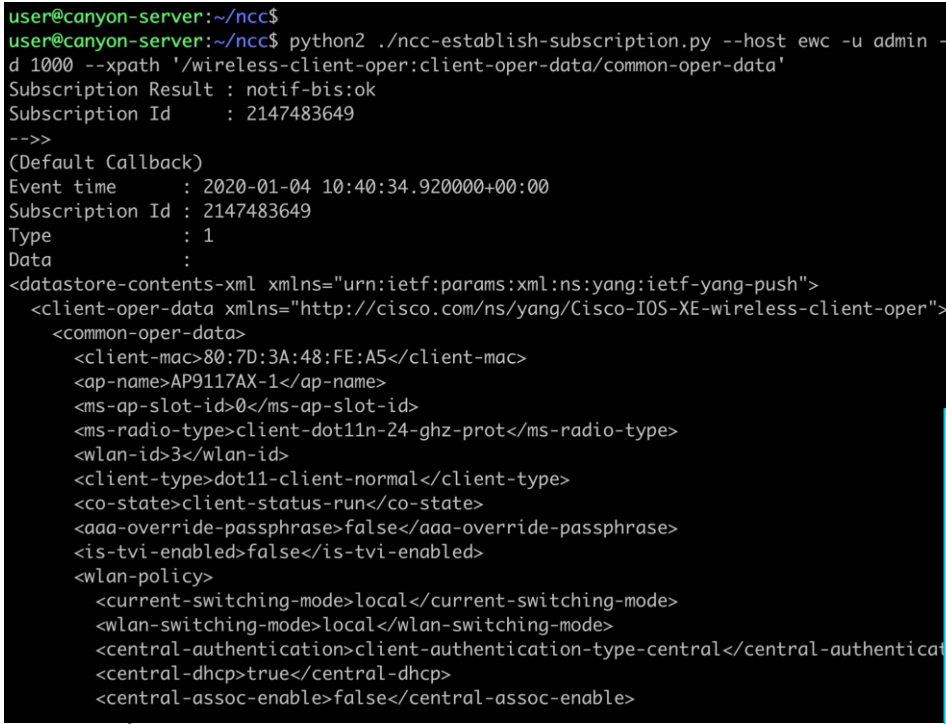
XML, JSON or kvGPB encoding

Consistent YANG data models between interfaces

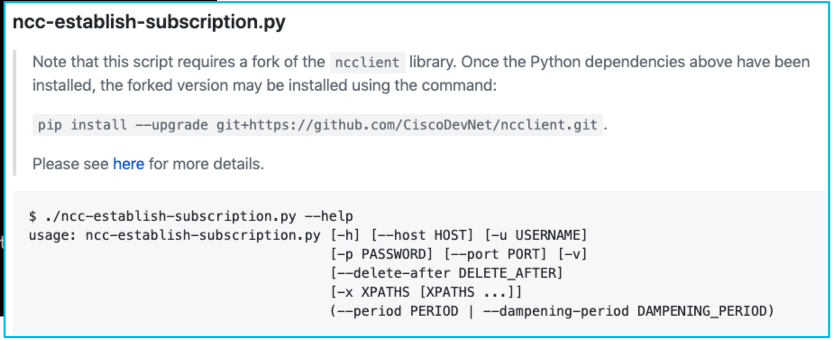
NETCONF + NCC Establish Subscription

https://github.com/CiscoDevNet/ncc

$ python2 ./ncc-establish-subscription.py --host ewc -u admin -p Cisco123 --period 1000 --xpath '/weless-client-oper:client-oper-data/common-oper-data'



```
user@canyon-server:~/ncc$
user@canyon-server:~/ncc$ python2 ./ncc-establish-subscription.py --host ewc -u admin -p Cisco123 --perio
d 1000 --xpath '/wireless-client-oper:client-oper-data/common-oper-data'
Subscription Result : notif-bis:ok
Subscription Id     : 2147483649
-->>
(Default Callback)
Event time       : 2020-01-04 10:40:34.920000+00:00
Subscription Id : 2147483649
Type             : 1
Data             :
<datastore-contents-xml xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <client-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-client-oper">
    <common-oper-data>
      <client-mac>80:7D:3A:48:FE:A5</client-mac>
      <ap-name>AP9117AX-1</ap-name>
      <ms-ap-slot-id>0</ms-ap-slot-id>
      <ms-radio-type>client-dot11n-24-ghz-prot</ms-radio-type>
      <wlan-id>3</wlan-id>
      <client-type>dot11-client-normal</client-type>
      <co-state>client-status-run</co-state>
      <aaa-override-passphrase>false</aaa-override-passphrase>
      <is-tvi-enabled>false</is-tvi-enabled>
      <wlan-policy>
        <current-switching-mode>local</current-switching-mode>
        <wlan-switching-mode>local</wlan-switching-mode>
        <central-authentication>client-authentication-type-central</central-authenticat
        <central-dhcp>true</central-dhcp>
        <central-assoc-enable>false</central-assoc-enable>
```



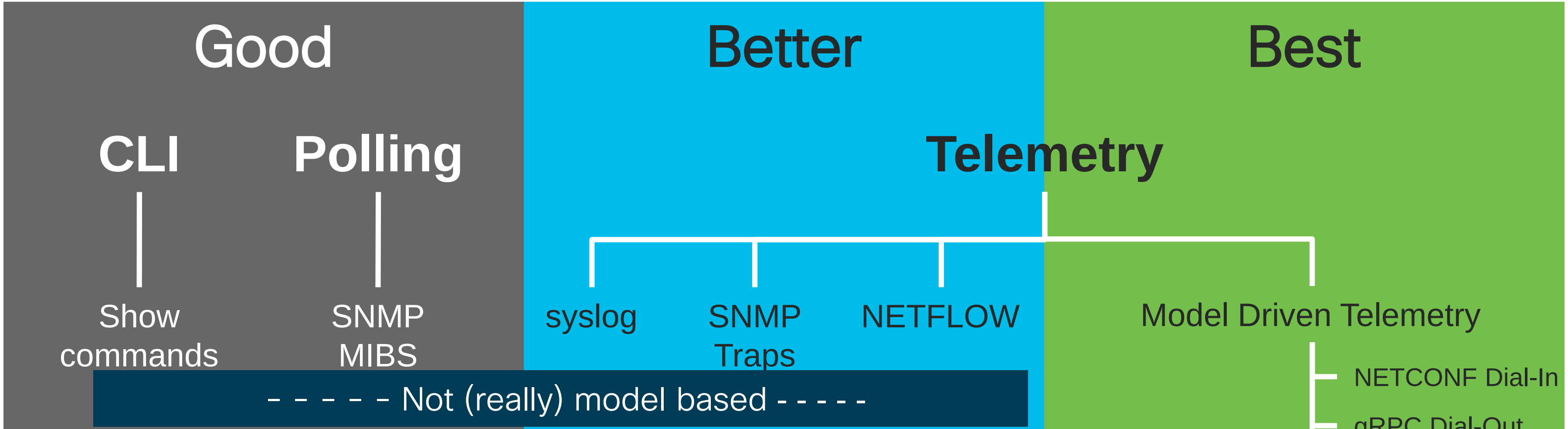
ncc-establish-subscription.py

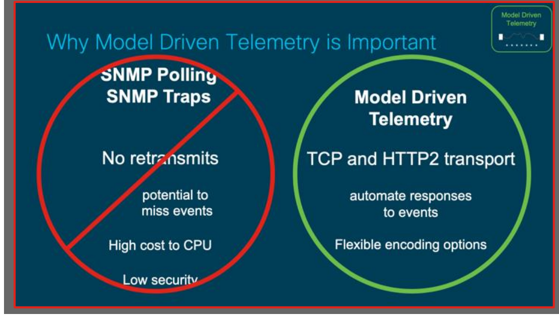
> Note that this script requires a fork of the `ncclient` library. Once the Python dependencies above have been installed, the forked version may be installed using the command:
>
> `pip install --upgrade git+https://github.com/CiscoDevNet/ncclient.git`.
>
> Please see here for more details.

```
$ ./ncc-establish-subscription.py --help
usage: ncc-establish-subscription.py [-h] [--host HOST] [-u USERNAME]
                                     [-p PASSWORD] [--port PORT] [-v]
                                     [--delete-after DELETE_AFTER]
                                     [-x XPATHS [XPATHS ...]]
                                     (--period PERIOD | --dampening-period DAMPENING_PERIOD)
```

IOS XE Operational Data Providers

Model Driven Telemetry





Why Model Driven Telemetry is Important

Model Driven Telemetry

SNMP Polling
SNMP Traps

No retransmits

potential to miss events

High cost to CPU

Low security

Model Driven Telemetry

TCP and HTTP2 transport

automate responses to events

Flexible encoding options

Everything defined in YANG

Consistent Encoding

Reliable Transport

IOS XE Model Driven Telemetry

Cisco IOS XE
17.1

CLI

...or with...

YANG

ANSIBLE

gNMI Dial-In
NETCONF Dial-In

gRPC Dial-Out

Receiver
Decodes to text

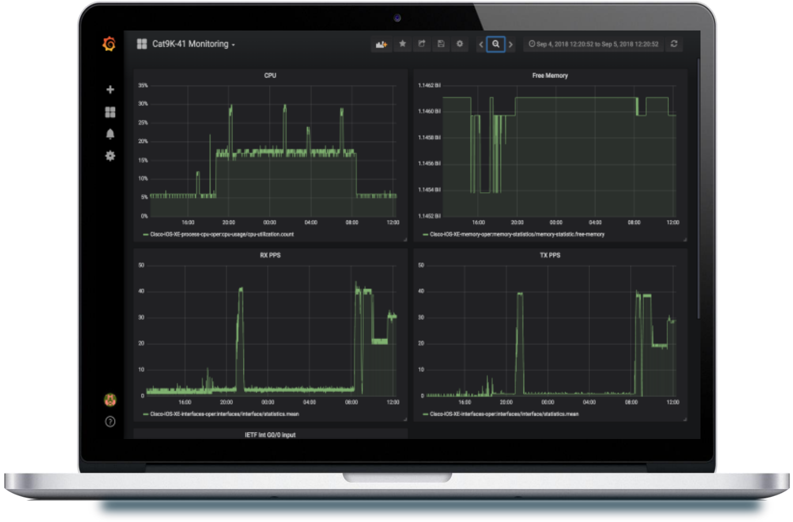
Telegraf

Storage
Time Series Database

InfluxDB

Monitoring
and Visualizations

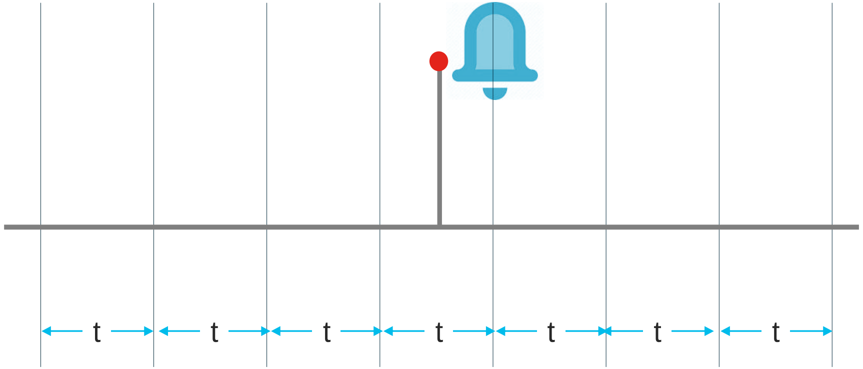
Grafana


Cat9K-41 Monitoring
Sep 4, 2018 12:20:52 to Sep 5, 2018 12:20:52
CPU
Free Memory
Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization.count
Cisco-IOS-XE-memory-oper:memory-statistics/memory-statistic.free-memory
RX PPS
TX PPS
Cisco-IOS-XE-interfaces-oper:interfaces/interface/statistics.mean
IETF Int G0/0 input

https://hub.docker.com/r/jeremycohoe/tig_mdt https://github.com/jeremycohoe/cisco-ios-xe-mdt
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/171/b_171_programmability_cg.html

Notification Types

On-Change

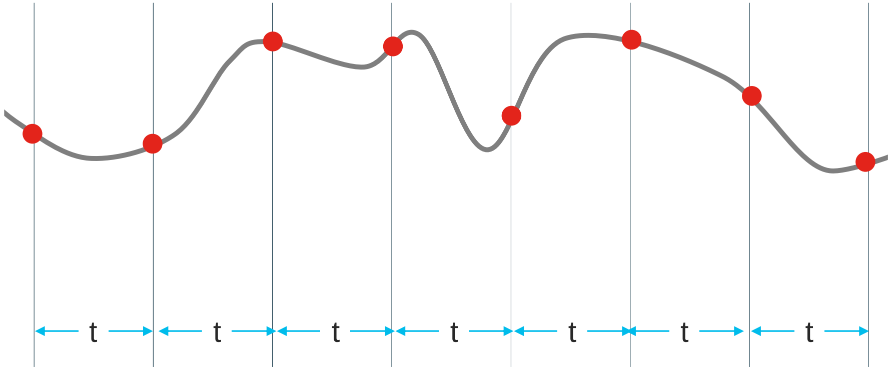


NETCONF Base Notifications (yang-push)

Event Notifications (failed login, etc)

Feature Model “On-Change” Notifications

Periodic



Minimum publication interval: 100 microseconds

Operational or Configurational data model

Feature Model “Periodic” Notifications

gRPC (Dial-Out/Configured) Telemetry Subscriptions

via CLI

```
telemetry ietf subscription 101

encoding encode-kvgpb

filter xpath /memory-ios-xe-oper:memory-statistics/memory-statistic

stream yang-push

update-policy periodic 6000

source-vrf Mgmt-intf

receiver ip address 10.10.1.45 57555

protocol grpc-tcp
```

via YANG Data Model

```
"mdt-config-data": {
   "mdt-subscription":[ {
        "subscription-id": "101",
        "base": {
            "stream": "yang-push",
            "encoding": "encode-kvgpb",
            "period": "6000",
            "xpath": "/memory-ios-xe-oper:memory-
                statistics/memory-statistic"
            }
        "mdt-receivers": {
            "address": ”10.10.1.45"
            "port": "57555" }
        }
   ]
}
```

Catalyst 9800 Wireless LAN Controller

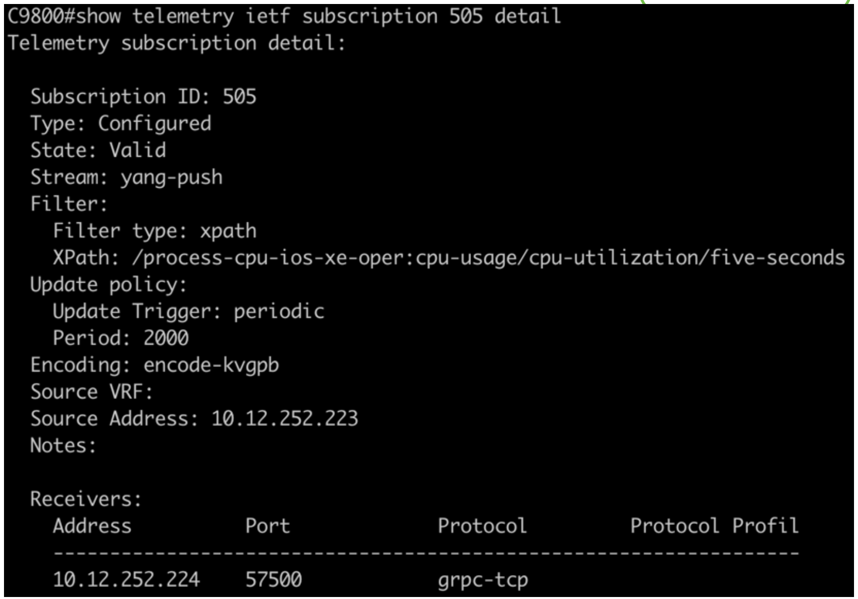
Model Driven Telemetry

sh run | sec tel

sh telemetry ietf subscription all

sh telemetry ietf subscription 101 receiver

sh telemetry ietf subscription 101 detail



```
C9800#show telemetry ietf subscription 505 detail
Telemetry subscription detail:

  Subscription ID: 505
  Type: Configured
  State: Valid
  Stream: yang-push
  Filter:
    Filter type: xpath
    XPath: /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
  Update policy:
    Update Trigger: periodic
    Period: 2000
  Encoding: encode-kvgpb
  Source VRF:
  Source Address: 10.12.252.223
  Notes:

  Receivers:
    Address             Port                Protocol            Protocol Profil
    ------------------------------------------------------------------------------
    10.12.252.224       57500               grpc-tcp
```

```
telemetry ietf subscription 501
 encoding encode-kvgpb
 filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
 source-address 10.60.0.19
 source-vrf Mgmt-vrf
 stream yang-push
 update-policy periodic 500
 receiver ip address 10.12.252.224 57500 protocol grpc-tcp
```

Grafana

Grafana is the HTM5 UI, it connects to and accesses data from the InfluxDB

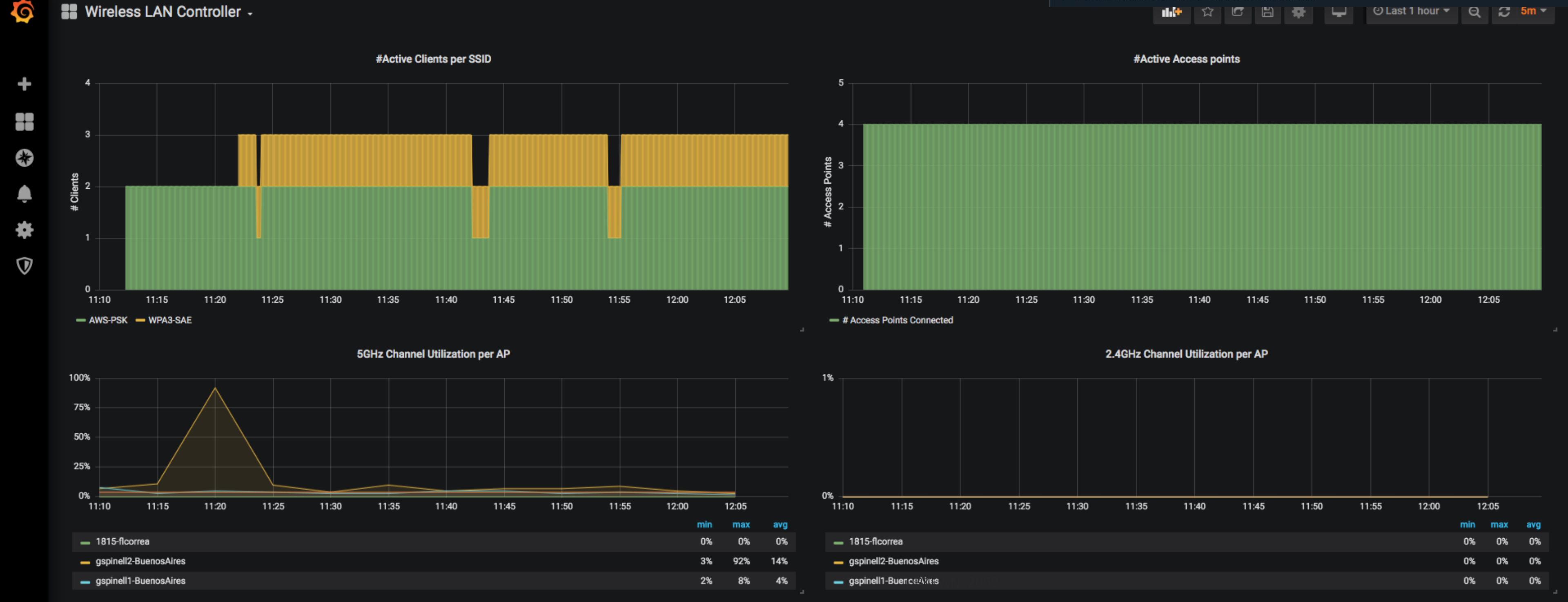
Follow

@correaflavio

Flavio Correa
@correaflavio

Solutions Architect, CCIE Wireless #38913, Innovation enthusiast, Network Programability (DevNet) champion, like road cycling and drums.

Sao Paulo, Brazil · Joined April 2009



5GHz Channel Utilization per AP

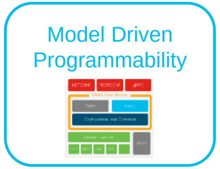
	min	max	avg
1815-flcorrea	0%	0%	0%
gspinell2-BuenosAires	3%	92%	14%
gspinell1-BuenosAires	2%	8%	4%

2.4GHz Channel Utilization per AP

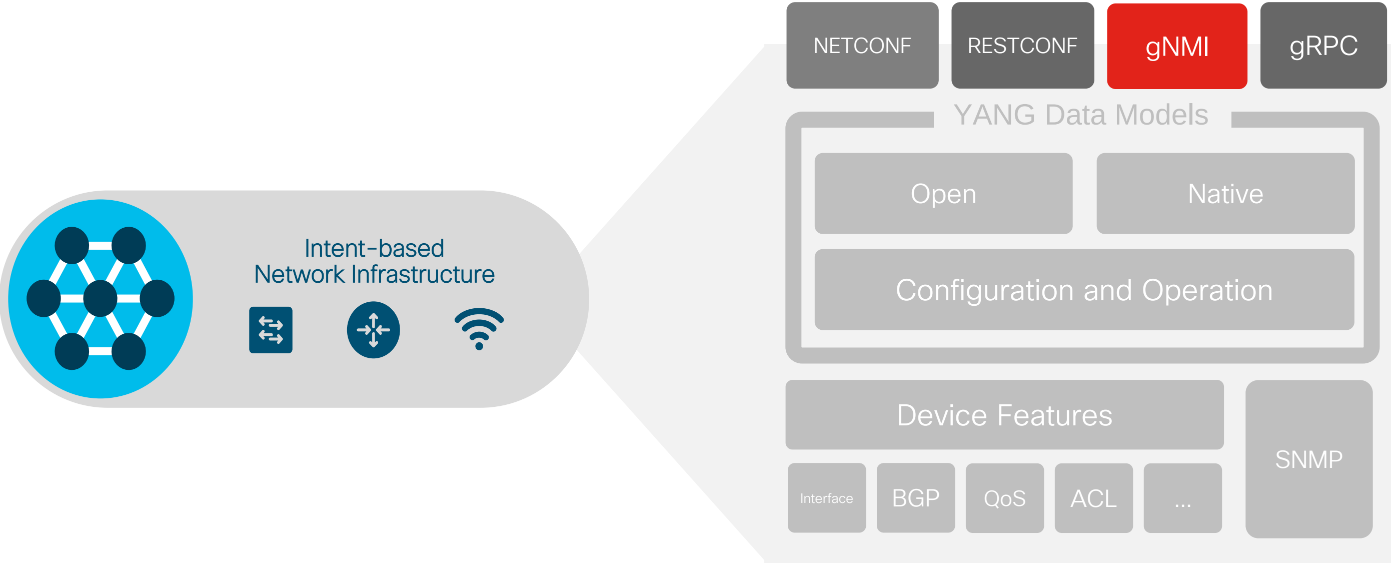
	min	max	avg
1815-flcorrea	0%	0%	0%
gspinell2-BuenosAires	0%	0%	0%
gspinell1-BuenosAires	0%	0%	0%

Model Driven Programmability and Telemetry

IOS XE Open Interface "Stack"

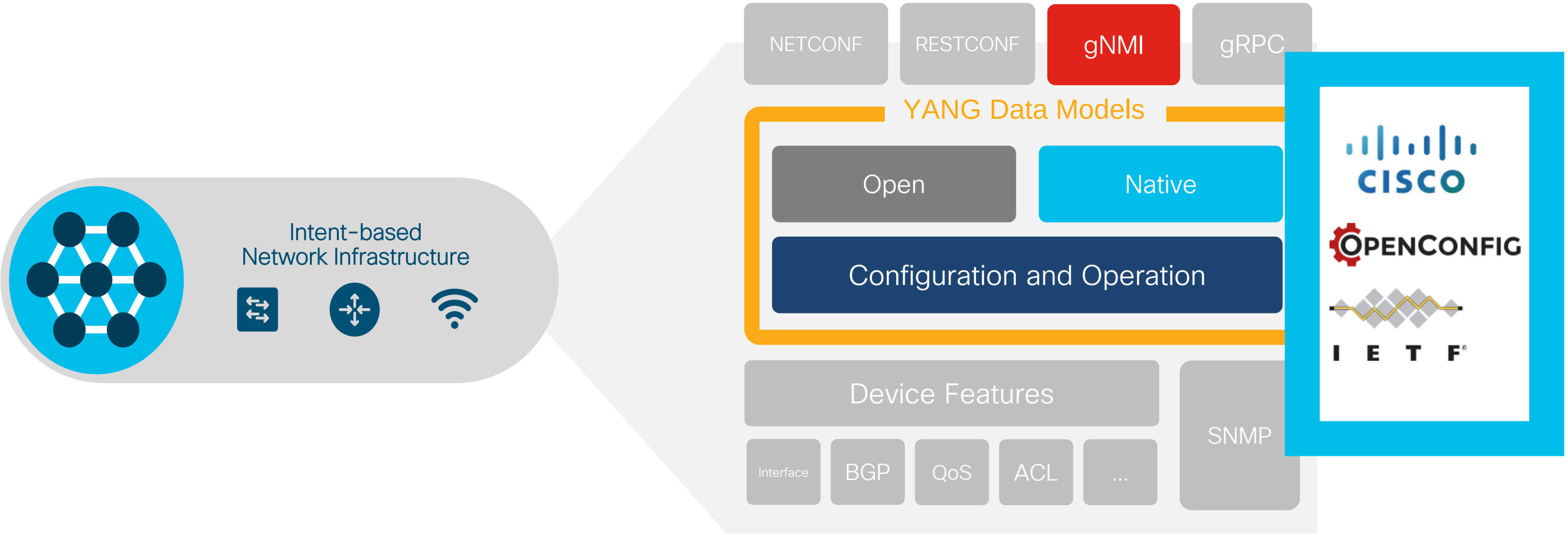

Model Driven Programmability

Model Driven Telemetry



Model Driven Programmability and Telemetry

IOS XE Open Interface “Stack”



Intent-based Network Infrastructure

NETCONF

RESTCONF

gNMI

gRPC

YANG Data Models

Open

Native

Configuration and Operation

Device Features

Interface

BGP

QoS

ACL

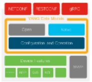
...

SNMP

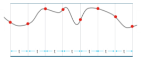
Model Driven Programmability and Telemetry

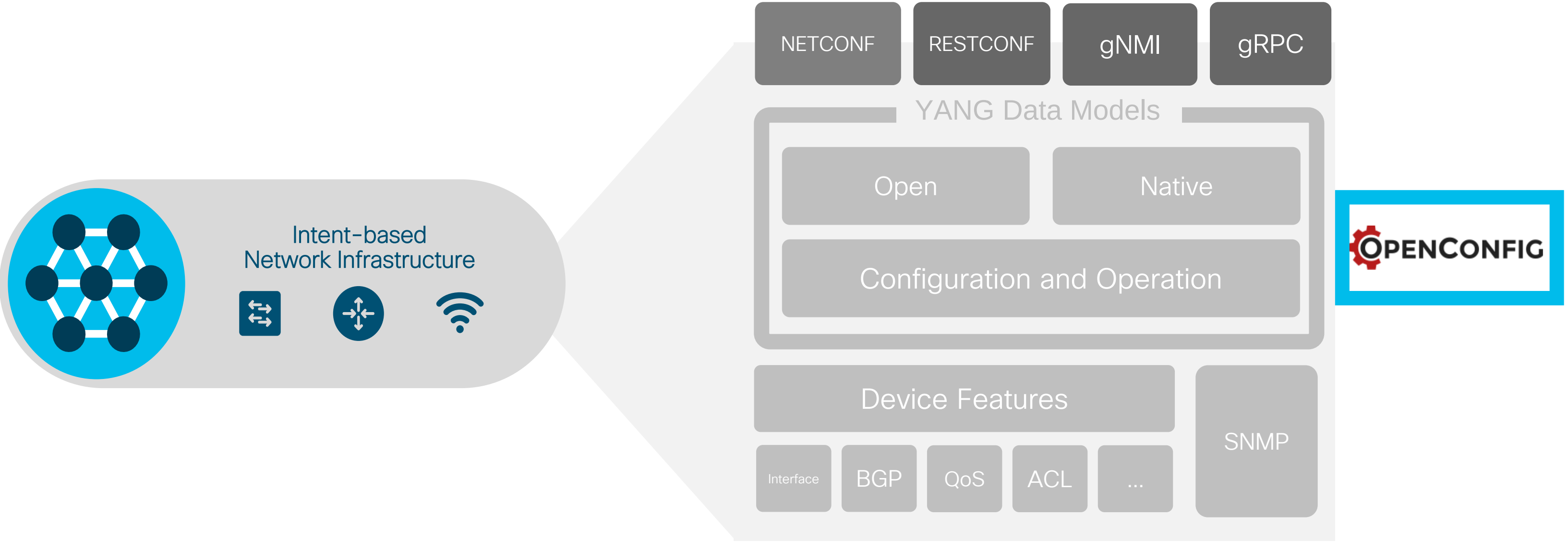
IOS XE Open Interface “Stack”

Model Driven Programmability



Model Driven Telemetry





https://github.com/YangModels/yang/blob/master/vendor/cisco/xe/16121/openconfig-ap-manager.yang
https://github.com/YangModels/yang/blob/master/vendor/cisco/xe/16121/openconfig-access-points.yang

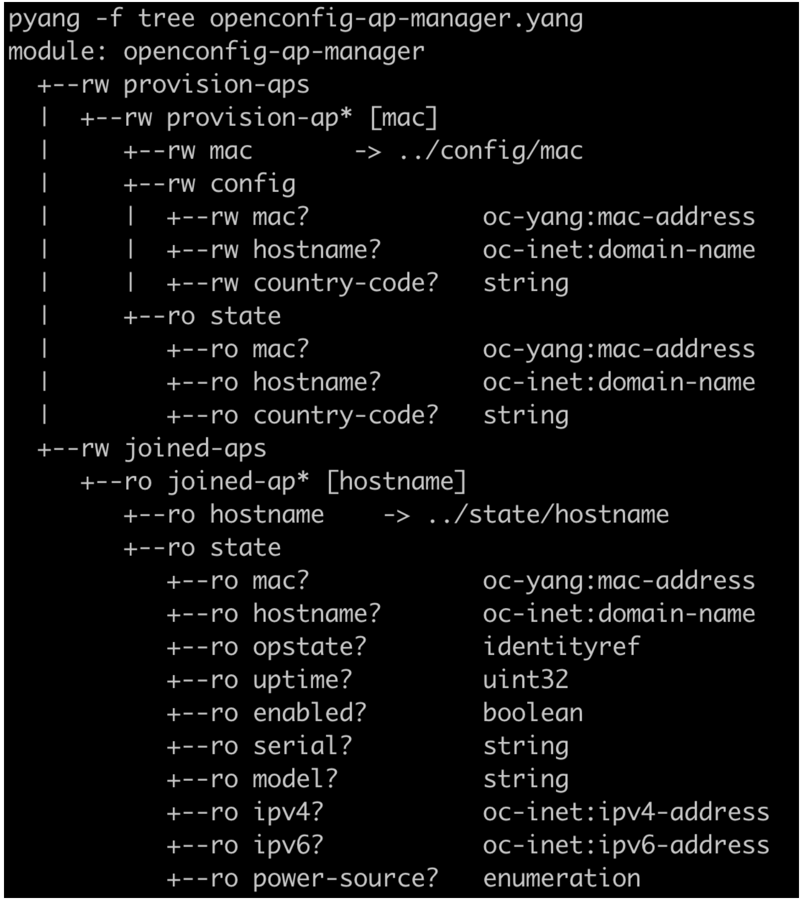
OpenConfig YANG

AP Manager and Access Points

AP Manger YANG is used to provision the AP:

It creates the link between the AP hostname and the MAC address of the AP

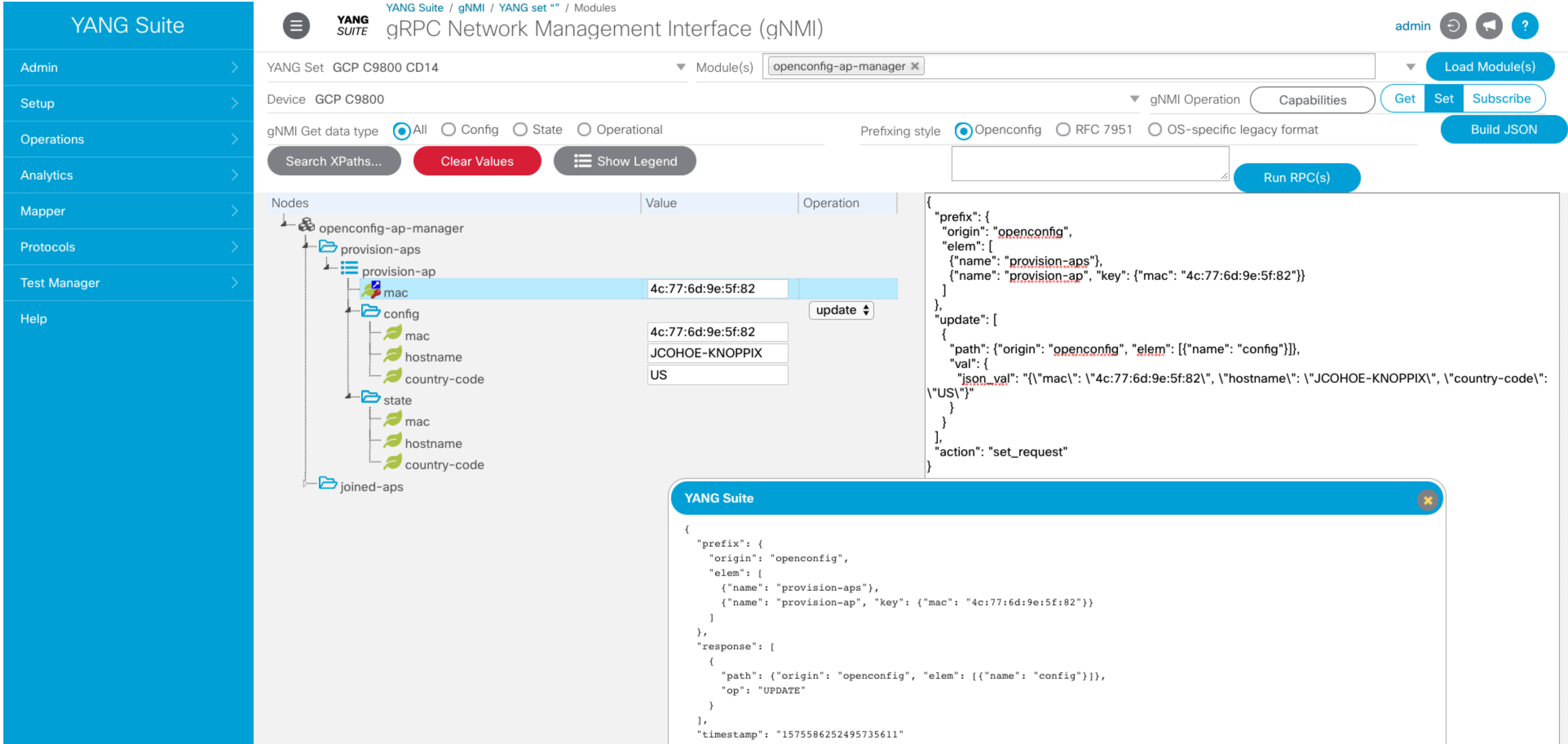
The gNMI API has an AP hostname centric model – the MAC address is never used again, instead it uses the hostname !



```
pyang -f tree openconfig-ap-manager.yang
module: openconfig-ap-manager
  +--rw provision-aps
  |  +--rw provision-ap* [mac]
  |     +--rw mac       -> ../config/mac
  |     +--rw config
  |     |  +--rw mac?            oc-yang:mac-address
  |     |  +--rw hostname?       oc-inet:domain-name
  |     |  +--rw country-code?   string
  |     +--ro state
  |        +--ro mac?            oc-yang:mac-address
  |        +--ro hostname?       oc-inet:domain-name
  |        +--ro country-code?   string
  +--rw joined-aps
     +--ro joined-ap* [hostname]
        +--ro hostname    -> ../state/hostname
        +--ro state
           +--ro mac?            oc-yang:mac-address
           +--ro hostname?       oc-inet:domain-name
           +--ro opstate?        identityref
           +--ro uptime?         uint32
           +--ro enabled?        boolean
           +--ro serial?         string
           +--ro model?          string
           +--ro ipv4?           oc-inet:ipv4-address
           +--ro ipv6?           oc-inet:ipv6-address
           +--ro power-source?   enumeration
```

https://github.com/YangModels/yang/blob/master/vendor/cisco/xe/16121/openconfig-ap-manager.yang
https://github.com/YangModels/yang/blob/master/vendor/cisco/xe/16121/openconfig-access-points.yang

gNMI Access Point Provisioning



YANG Set GCP C9800 CD14

Module(s) openconfig-ap-manager ✕

Device GCP C9800

gNMI Get data type ● All ○ Config ○ State ○ Operational

Prefixing style ● Openconfig ○ RFC 7951 ○ OS-specific legacy format

Nodes	Value	Operation
openconfig-ap-manager		
provision-aps		
provision-ap		
mac	4c:77:6d:9e:5f:82	
config		update
mac	4c:77:6d:9e:5f:82	
hostname	JCOHOE-KNOPPIX	
country-code	US	
state		
mac		
hostname		
country-code		
joined-aps		

```
{
  "prefix": {
    "origin": "openconfig",
    "elem": [
      {"name": "provision-aps"},
      {"name": "provision-ap", "key": {"mac": "4c:77:6d:9e:5f:82"}}
    ]
  },
  "update": [
    {
      "path": {"origin": "openconfig", "elem": [{"name": "config"}]},
      "val": {
        "json_val": "{\"mac\": \"4c:77:6d:9e:5f:82\", \"hostname\": \"JCOHOE-KNOPPIX\", \"country-code\": 
\"US\"}"
      }
    }
  ],
  "action": "set_request"
}
```

```
{
  "prefix": {
    "origin": "openconfig",
    "elem": [
      {"name": "provision-aps"},
      {"name": "provision-ap", "key": {"mac": "4c:77:6d:9e:5f:82"}}
    ]
  },
  "response": [
    {
      "path": {"origin": "openconfig", "elem": [{"name": "config"}]},
      "op": "UPDATE"
    }
  ],
  "timestamp": "1575586252495735611"
```

Cisco + Google

gNMI + Openconfig

Co-Development

Enterprise Wifi using Open APIs

Openconfig for WiFi

Mike Albano
Shimol Shah

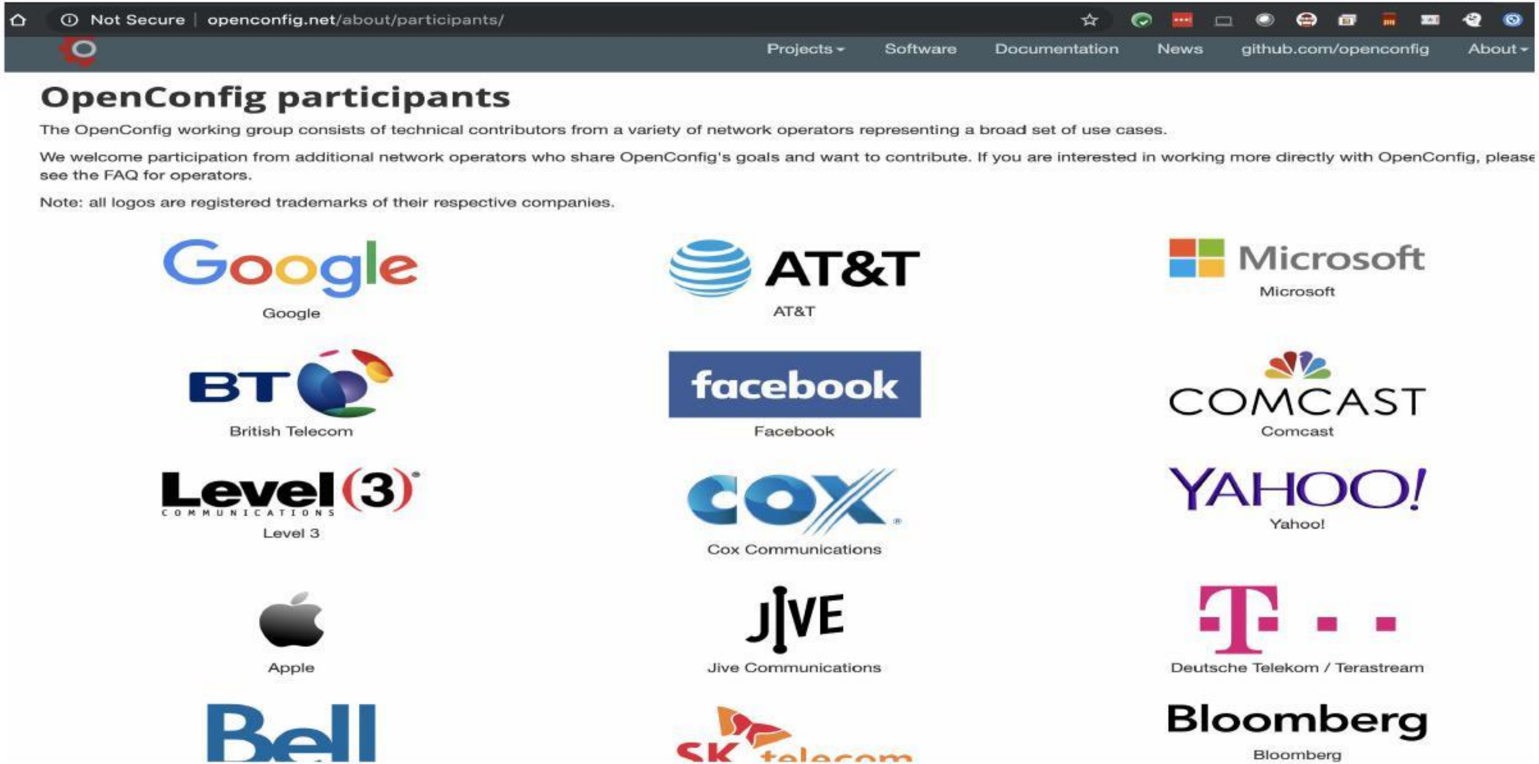
What is OpenConfig

- A set of Vendor-neutral data models for interacting with the network; authored by Network engineers.

- Informal, structured like open source: https://github.com/openconfig/public/

- OpenConfig is the Schema. gNMI is the Transport. gNOI is what's left.

Current participants: www.openconfig.net/about/participants/



OpenConfig participants

The OpenConfig working group consists of technical contributors from a variety of network operators representing a broad set of use cases.

We welcome participation from additional network operators who share OpenConfig's goals and want to contribute. If you are interested in working more directly with OpenConfig, please see the FAQ for operators.

Note: all logos are registered trademarks of their respective companies.

Google

AT&T

Microsoft

British Telecom

Facebook

Comcast

Level 3

Cox Communications

Yahoo!

Apple

Jive Communications

Deutsche Telekom / Terastream

Bloomberg

OPENCONFIG

Why? What problems did it solve?

1. We needed **Telemetry** (radio-data) and we needed it fast. *[Ops Impact]*

2. We wanted to move away from translation layers. (We tried them. Difficult & error-prone). *[Tools Impact]*

3. We need **programmatic access** and structured APIs for **everything**. *[Deploy & Ops Impact]*

How does it solve them?

1. **Telemetry**

Streaming Telemetry is a big part of Openconfig

2. **Translation Layers**

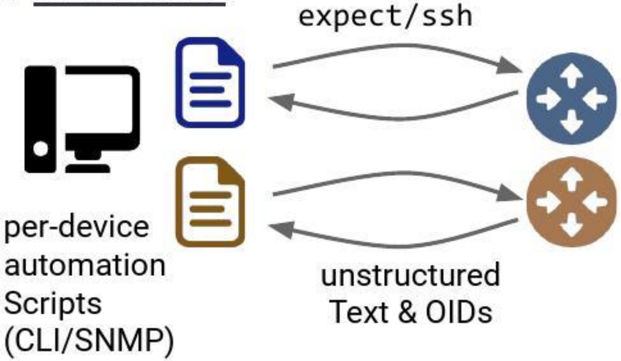
Everyone adheres to 1 Schema. See:
https://github.com/openconfig/public/tree/master/release/models/wifi

3. **Programmatic Access**

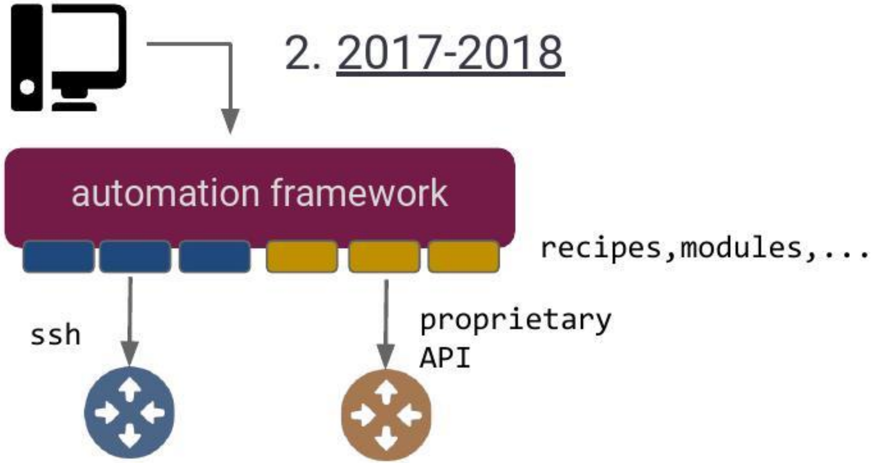
Entirely API driven, through gNMI/gRPC. 0 native access.

Evolution of network element interaction...

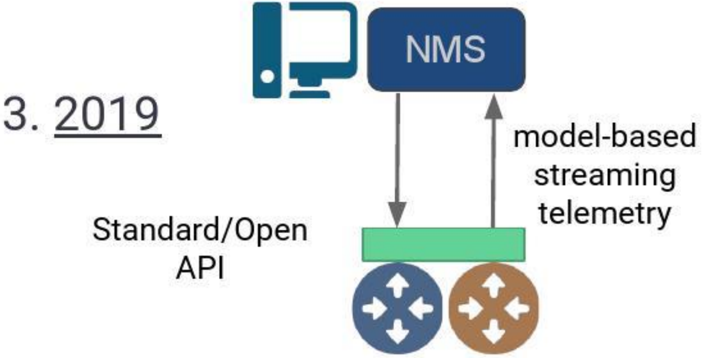
1. Pre 2017



2. 2017-2018



3. 2019



Pre Openconfig Network

- Physical Controllers -- like everywhere.
  - Multiple physical management points. Configure them, Operate them, LCM them
  - No programmatic access
- Centralized Data plane (lots of tunnels in network to solve mgmt-plane)
  - Large failure domains (WLC goes down -- so did a lot of APs)
- Configuration Management
  - CLI access needed to make changes
- Too much human input (CLI based)
  - To push config
  - To operate
- Non granular Telemetry
  - SNMP based
  - Not fast enough

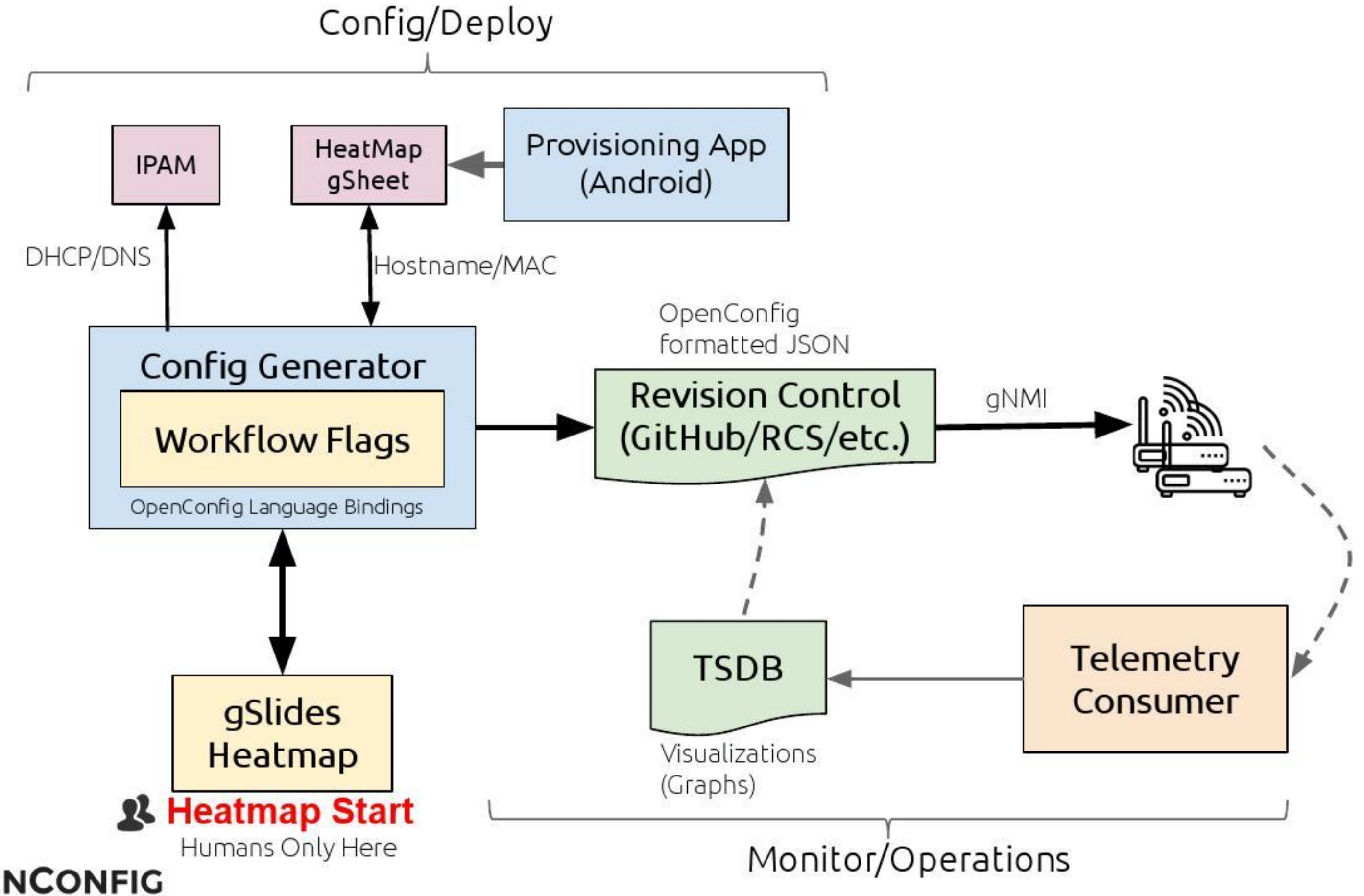
Openconfig Network

- Controllers; only where required
  - Data-plane out-of-scope.
  - Programmatic access.
- Configuration Management
  - Standard APIs used to configure & monitor
- No Human Input for config/operate
  - To push config
  - To operate
- Granular Telemetry
  - Publisher/Subscriber (pub/sub); not polling
  - Fast, encrypted by default

How Do We Use It

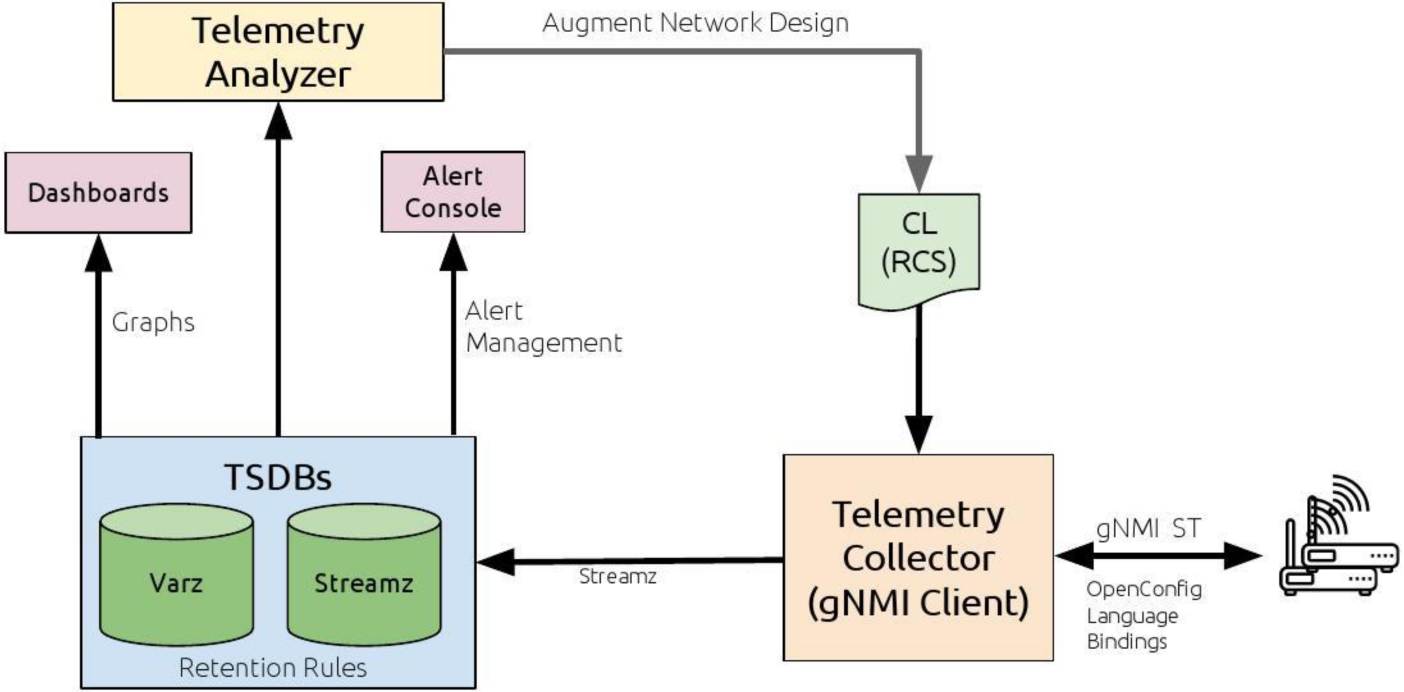
- Intent is populated by automation (inferred from design rules in heatmap)
- No humans interact directly (no CLI or direct access) with network elements
- Network Admin modifies design rules (heatmap) to trigger configuration changes
- Network Operator only uses vendor-independent UI (eg TSDB Visualizations). No CLI or direct access to operate the network
- Network Operator does not know what vendor's network element in use.

E2E Automatred Toolchain Example



OPENCONFIG

Respond to Conditions



OPENCONFIG

Demo

Questions

Reach out to albanom@google.com & shimol@google.com

gNMI

gNMI Call Architecture

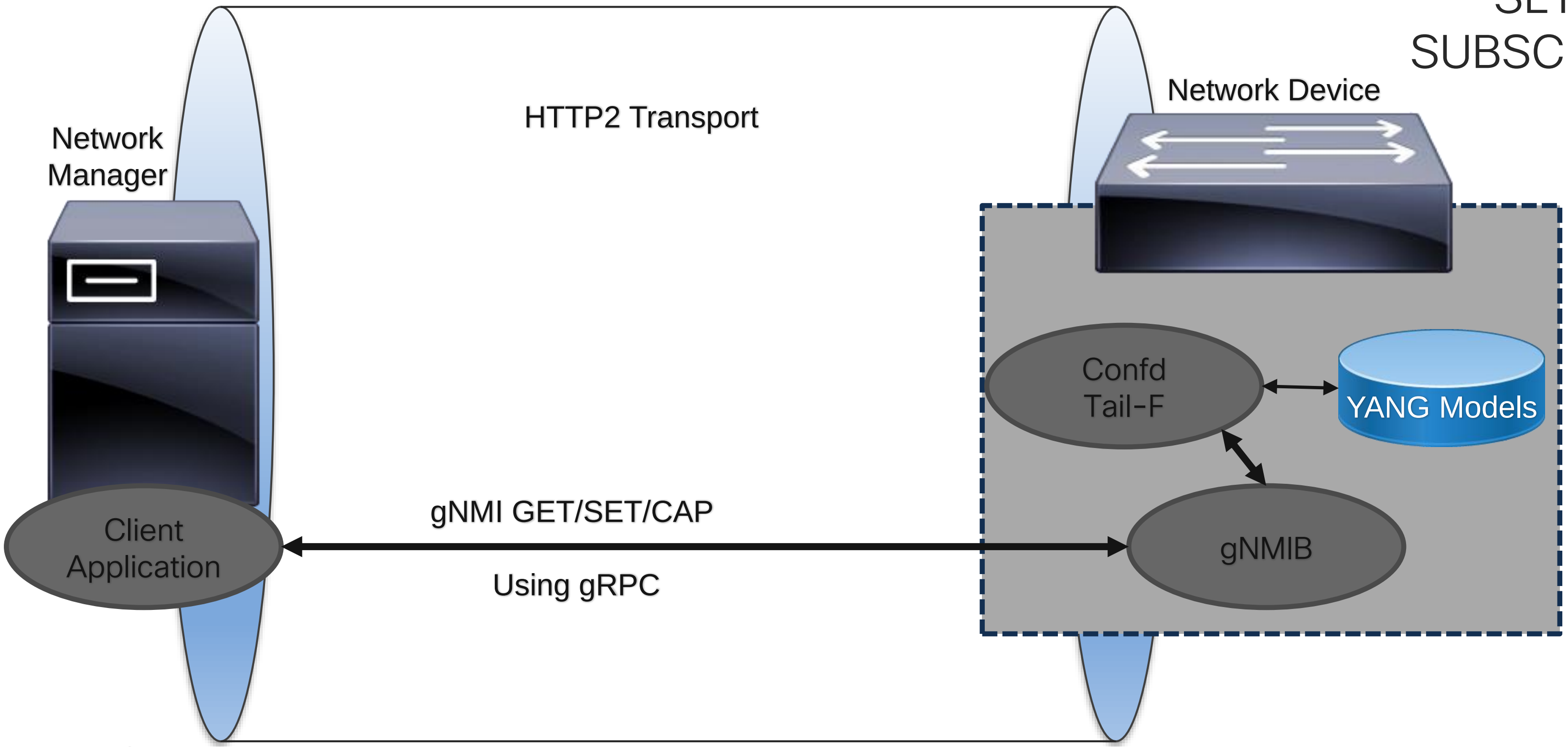
gNMI Operations

CAP

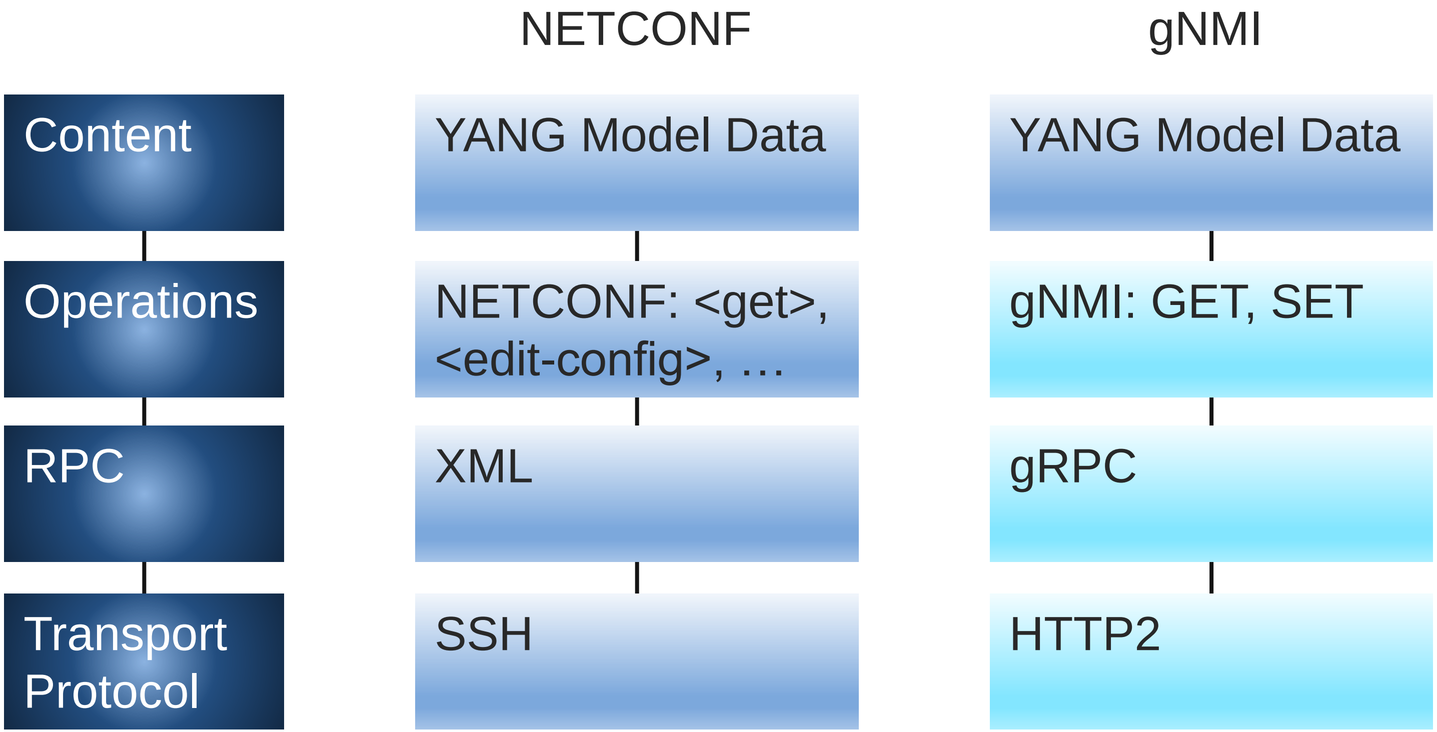
GET

SET

SUBSCRIBE



gNMI vs Netconf Layering

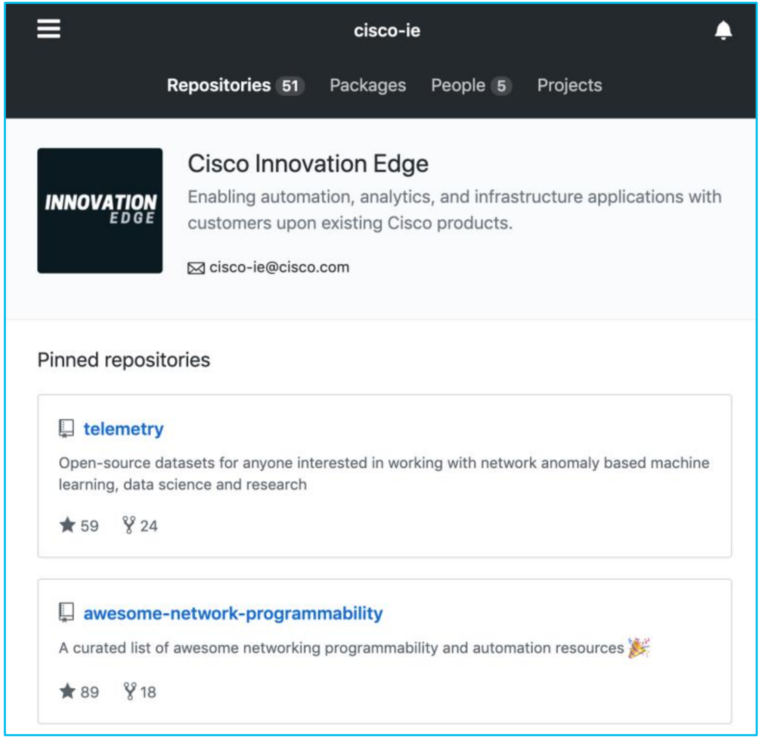


	NETCONF	gNMI
Content	YANG Model Data	YANG Model Data
Operations	NETCONF: <get>, <edit-config>, …	gNMI: GET, SET
RPC	XML	gRPC
Transport Protocol	SSH	HTTP2

gnmi_cli

- OpenConfig Developed
  - https://github.com/openconfig/gnmi
- We have a fork with documentation
  - https://github.com/cisco-ie/gnmi
- This repository is both a gNMI library implementation as well as a CLI wrapper
- gnmi_cli is a simple wrapper of the library!
- https://github.com/openconfig/gnmi/blob/master/proto/gnmi/gnmi.proto
- Syntax

```
./gnmi_cli -address <IP> [-insecure] [-with_user_pass]
        -proto <SubscribeRequest>
        [-get -proto <GetRequest>]
        [-set -proto <SetRequest>]
        [-capabilities [-proto <CapabilityRequest>]]
```



Cisco Innovation Edge

Enabling automation, analytics, and infrastructure applications with customers upon existing Cisco products.

cisco-ie@cisco.com

Pinned repositories

telemetry

Open-source datasets for anyone interested in working with network anomaly based machine learning, data science and research

★ 59 24

awesome-network-programmability

A curated list of awesome networking programmability and automation resources 🎉

★ 89 18

Enabling gNMI (Secure modes)

- C9800#gnmi-yang
- C9800#gnmi-yang secure-server
- C9800#gnmi-yang secure-trustpoint <<trustpoint name>>
- ! Trustpoint is already configured
- ! Configure secure-client-auth (optional):
- C9800#gnmi-yang secure-client-auth
- !By default secure gNMI listens on port 50051.
- !Set the secure listen port (optional):
- C9800#gnmi-yang secure-port #####

```
C9800#show gnmi-yang state
State            Status
--------------------------------
Enabled          Up
```

See IOS XE Programmability Configuration Guide for more details
gNMI default port ratified 2019

gnmi_cli TLS

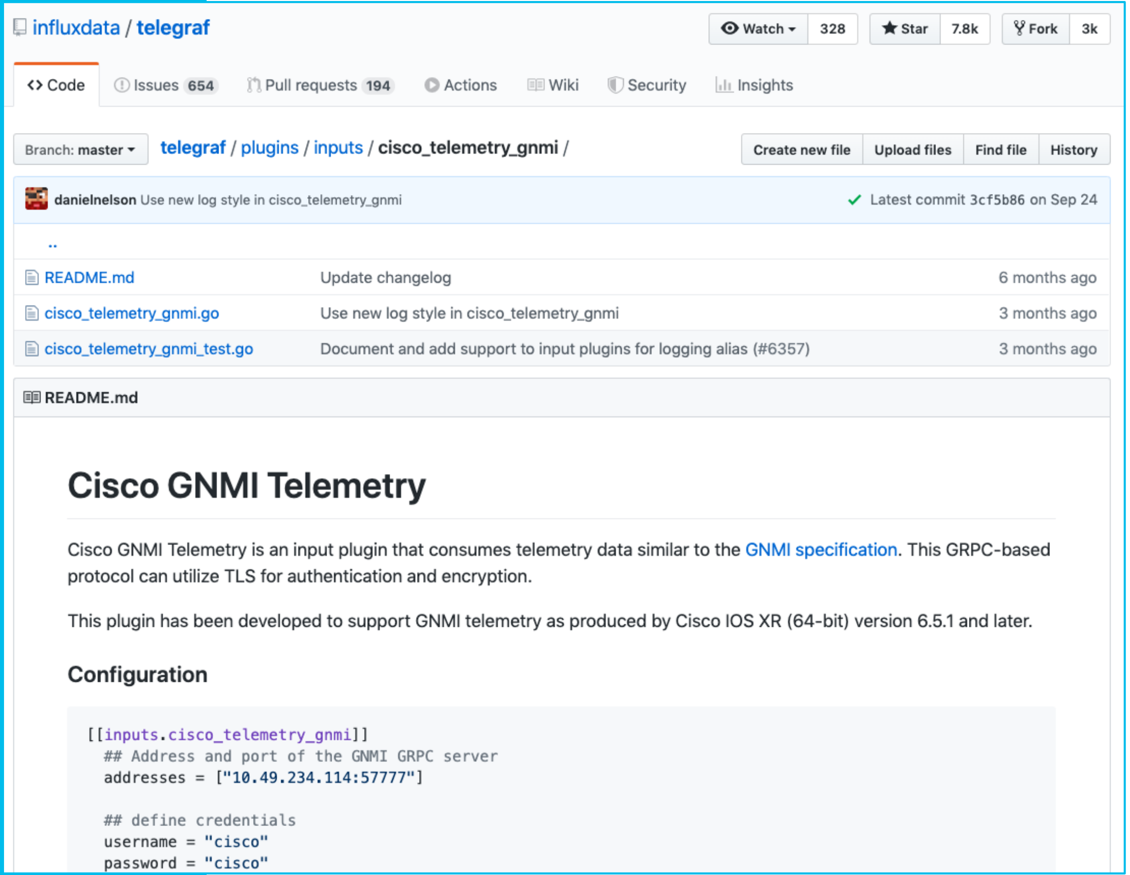
- TLS MUST be enabled for gnmi_cli to work correctly.
- We can easily remove this requirement from the code itself, but gNMI REQUIRES TLS per spec and if gRPC is configured without TLS gNMI requests will time out.

```
grpc
 port 57400
 ~~no-tls~~
 address-family ipv4
 tls-cipher service-layer
 !
!
```

```
./gnmi_cli -insecure ...
```

-insecure does NOT indicate lack of TLS. It indicates that the host TLS identity will not be verified, thus is insecure, but the connection is still encrypted. Encrypted !== Secure

gNMI Telemetry with Telegraf



telegraf / plugins / inputs / cisco_telemetry_gnmi /

danielnelson Use new log style in cisco_telemetry_gnmi ✓ Latest commit 3cf5b86 on Sep 24

..

README.md	Update changelog	6 months ago
cisco_telemetry_gnmi.go	Use new log style in cisco_telemetry_gnmi	3 months ago
cisco_telemetry_gnmi_test.go	Document and add support to input plugins for logging alias (#6357)	3 months ago

README.md

Cisco GNMI Telemetry

Cisco GNMI Telemetry is an input plugin that consumes telemetry data similar to the GNMI specification. This GRPC-based protocol can utilize TLS for authentication and encryption.

This plugin has been developed to support GNMI telemetry as produced by Cisco IOS XR (64-bit) version 6.5.1 and later.

Configuration

```
[[inputs.cisco_telemetry_gnmi]]
  ## Address and port of the GNMI GRPC server
  addresses = ["10.49.234.114:57777"]

  ## define credentials
  username = "cisco"
  password = "cisco"
```

https://github.com/influxdata/telegraf/tree/master/plugins/inputs/cisco_telemetry_gnmi

Model Driven Telemetry

CLI

... or with ...

NETCONF

ANSIBLE

Catalyst 9800 WLC
IOS XE 16.12.2s

gNMI Dial-In
NETCONF Dial-In

gRPC Dial-Out

Receiver
Decodes to text

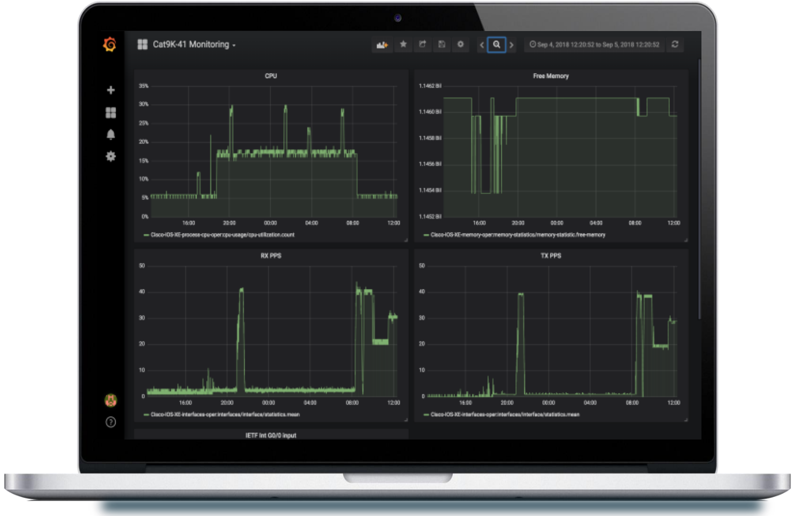
Telegraf

Collector
Time Series Database

InfluxDB

Monitoring
and Visualizations

Grafana


Cat9K-41 Monitoring
Sep 4, 2018 12:20:52 to Sep 5, 2018 12:20:52
CPU
35%
30%
25%
20%
15%
10%
5%
0%
16:00
20:00
00:00
04:00
08:00
12:00
Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization.count
Free Memory
1.1462 Bil
1.1460 Bil
1.1458 Bil
1.1456 Bil
1.1454 Bil
1.1452 Bil
Cisco-IOS-XE-memory-oper:memory-statistics/memory-statistic.free-memory
RX PPS
50
40
30
20
10
0
Cisco-IOS-XE-interfaces-oper:interfaces/interface/statistics.mean
TX PPS
Cisco-IOS-XE-interfaces-oper:interfaces/interface/statistics.mean
IETF Int G0/0 input

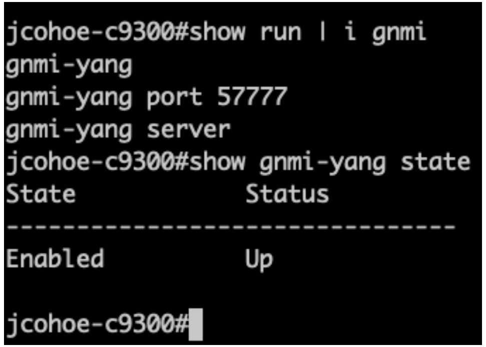
https://hub.docker.com/r/jeremycohoe/tig_mdt https://github.com/jeremycohoe/cisco-ios-xe-mdt
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/171/b_171_programmability_cg.html

Telegraf + gNMI

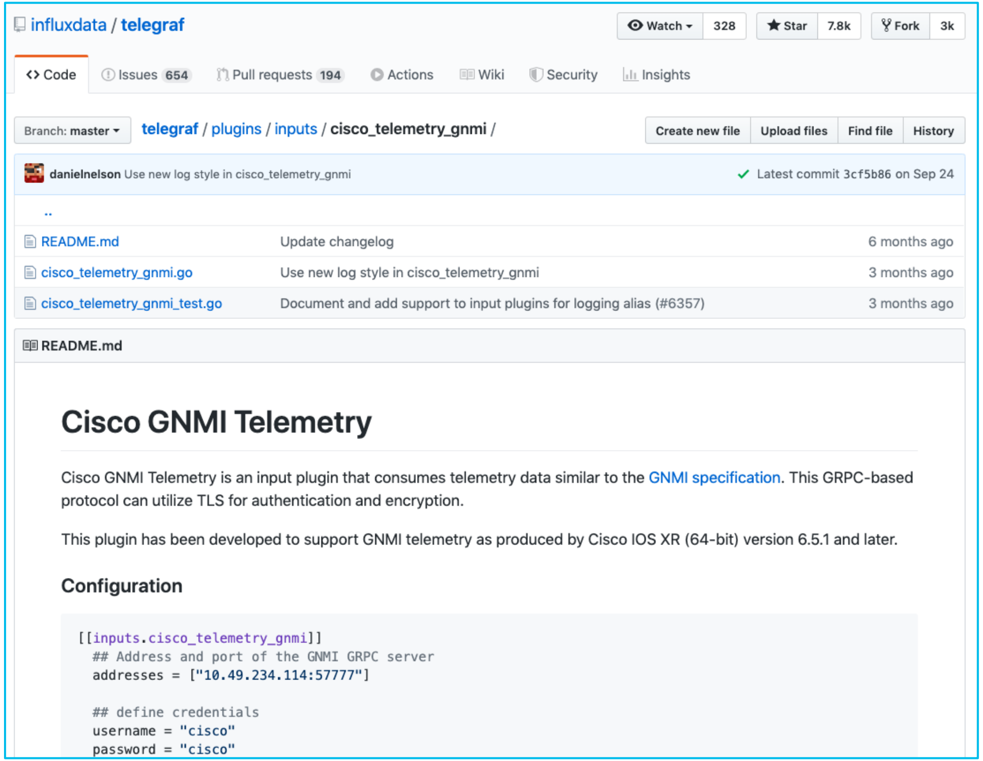
The telegraf plugin “cisco_telemetry_gnmi” can be used to “dial in” to IOS XE 16.12+ to collect Model Driven Telemetry

Example telegraf.conf Configuration:

```
[[inputs.cisco_telemetry_gnmi]]
addresses = ["jcohoe-cat9300.cisco.com:57777"]
username = "cisco"
password = "cisco"
encoding = "json_ietf"
redial = "10s"
[[inputs.cisco_telemetry_gnmi.subscription]]
name = "ifcounters"
origin = "openconfig"
path = "/interfaces/interface[name=TenGigabitEthernet1/0/1]/"
subscription_mode = "sample"
sample_interval = "10s"
```



```
jcohoe-c9300#show run | i gnmi
gnmi-yang
gnmi-yang port 57777
gnmi-yang server
jcohoe-c9300#show gnmi-yang state
State                Status
--------------------------------
Enabled              Up

jcohoe-c9300#
```



telegraf / plugins / inputs / cisco_telemetry_gnmi /

danielnelson Use new log style in cisco_telemetry_gnmi ✓ Latest commit 3cf5b86 on Sep 24

..		
README.md	Update changelog	6 months ago
cisco_telemetry_gnmi.go	Use new log style in cisco_telemetry_gnmi	3 months ago
cisco_telemetry_gnmi_test.go	Document and add support to input plugins for logging alias (#6357)	3 months ago

README.md

Cisco GNMI Telemetry

Cisco GNMI Telemetry is an input plugin that consumes telemetry data similar to the GNMI specification. This GRPC-based protocol can utilize TLS for authentication and encryption.

This plugin has been developed to support GNMI telemetry as produced by Cisco IOS XR (64-bit) version 6.5.1 and later.

Configuration

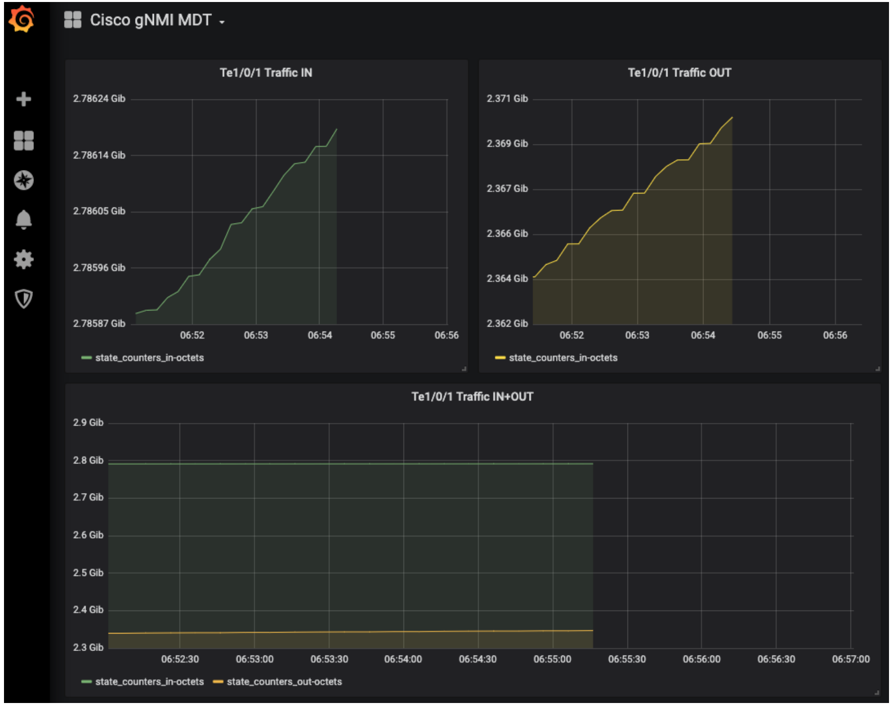
```
[[inputs.cisco_telemetry_gnmi]]
  ## Address and port of the GNMI GRPC server
  addresses = ["10.49.234.114:57777"]

  ## define credentials
  username = "cisco"
  password = "cisco"
```

https://github.com/influxdata/telegraf/tree/master/plugins/inputs/cisco_telemetry_gnmi

Programmability Configuration Guide: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1612/b_1612_programmability_cg/grpc_network_management_interface.html

gNMI Telemetry Example

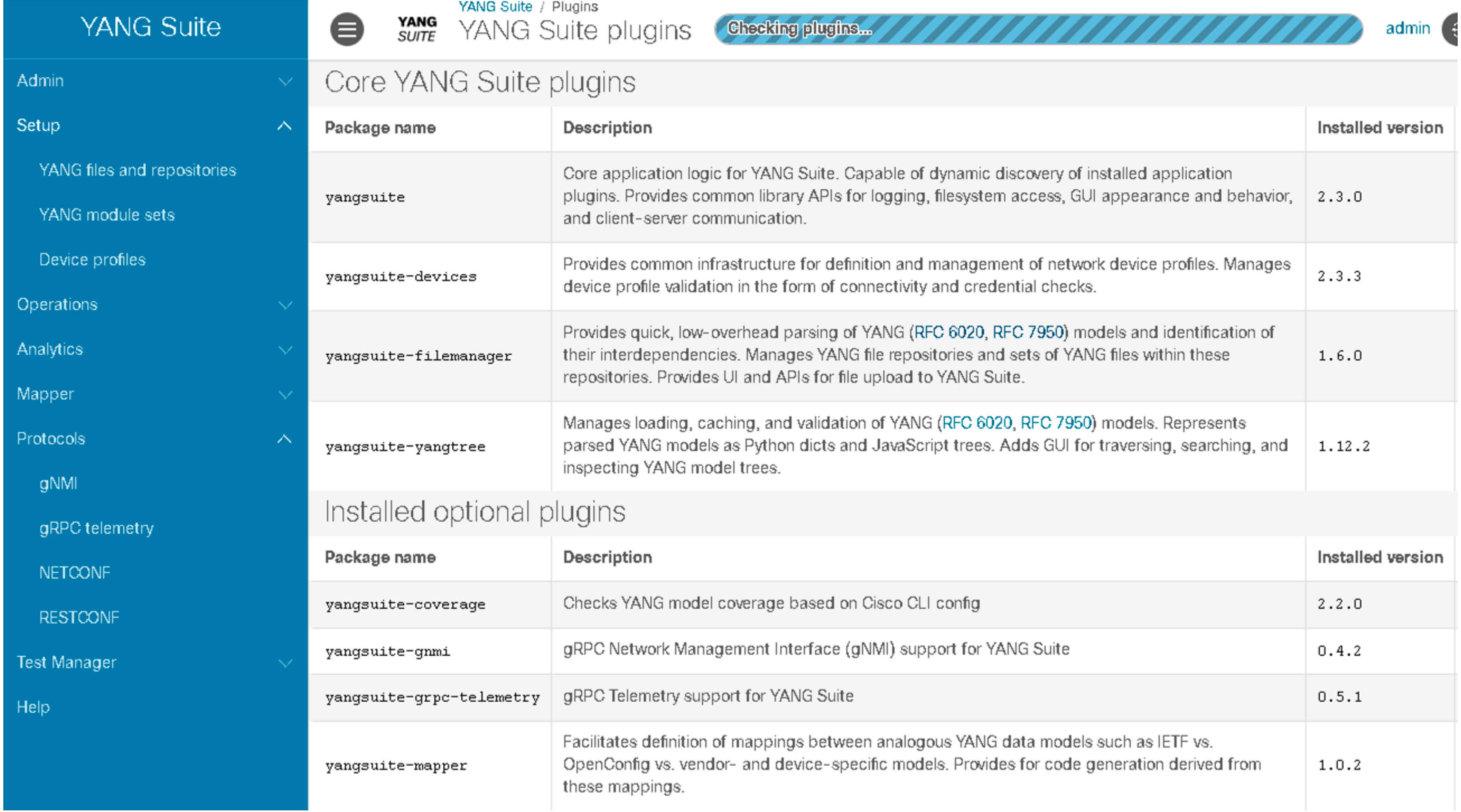
Telegraf data visualized with InfluxDB in Grafana



Raw data received by Telegraf

```
ifcounters,host=jcohoe-ubuntu,name=TenGigabitEthernet1/0/1,source=jcohoe-
cat9300.cisco.com subinterfaces_subinterface_0_state_admin-
status="UP",subinterfaces_subinterface_0_state_counters_out-
discards="0",subinterfaces_subinterface_0_openconfig-if-
ip:ipv4_state_counters_in-forwarded-pkts="0",openconfig-if-
ethernet:ethernet_state_hw-mac-address="70:1f:53:9b:0f:81",openconfig-if-
ethernet:ethernet_state_counters_in-mac-control-frames="0",state_counters_in-
fcs-errors="0",subinterfaces_subinterface_0_openconfig-if-
ip:ipv4_state_counters_out-discarded-pkts="0",openconfig-if-
ethernet:ethernet_state_counters_in-oversize-frames="0",state_counters_in-
octets="2991324767",state_openconfig-platform-port:hardware-
port="TenGigabitEthernet1/0/1",subinterfaces_subinterface_0_state_oper-
status="UP",subinterfaces_subinterface_0_state_counters_in-
discards="0",openconfig-if-ethernet:ethernet_state_mac-
address="70:1f:53:9b:0f:81",config_type="ethernetCsmacd",state_counters_last-
clear="1573596199000951000",subinterfaces_subinterface_0_state_counters_in
-multicast-pkts="128346",subinterfaces_subinterface_0_state_counters_out-
unicast-pkts="15247858",subinterfaces_subinterface_0_openconfig-if-
ip:ipv4_proxy-
arp_state_mode="DISABLE",subinterfaces_subinterface_0_openconfig-if-
ip:ipv4_state_counters_out-forwarded-
pkts="0",subinterfaces_subinterface_0_openconfig-if-ip:ipv6_state_counters_in-
forwarded-pkts="0",subinterfaces_subinterface_0_openconfig-if-
ip:ipv6_state_counters_in-forwarded-octets="0",openconfig-if-
ethernet:ethernet_state_auto-negotiate=false,openconfig-if-
ethernet:ethernet_state_negotiated-port-
speed="SPEED_1GB",subinterfaces_subinterface_0_state_counters_in-unicast-
pkts="4884070",openconfig-if-ethernet:ethernet_state_counters_out-mac-
control-
frames="0",config_description="UPLINK",subinterfaces_subinterface_0_state_ena
bled=true,subinterfaces_subinterface_0_openconfig-if-
ip:ipv4_state_counters_in-pkts="0",subinterfaces_subinterface_0_openconfig-if-
ip:ipv4_state_counters_out-forwarded-
```

YangSuite
YangExplorer (circa 2015 - 2020)



YANG Suite plugins

Core YANG Suite plugins

Package name	Description	Installed version
`yangsuite`	Core application logic for YANG Suite. Capable of dynamic discovery of installed application plugins. Provides common library APIs for logging, filesystem access, GUI appearance and behavior, and client-server communication.	2.3.0
`yangsuite-devices`	Provides common infrastructure for definition and management of network device profiles. Manages device profile validation in the form of connectivity and credential checks.	2.3.3
`yangsuite-filemanager`	Provides quick, low-overhead parsing of YANG (RFC 6020, RFC 7950) models and identification of their interdependencies. Manages YANG file repositories and sets of YANG files within these repositories. Provides UI and APIs for file upload to YANG Suite.	1.6.0
`yangsuite-yangtree`	Manages loading, caching, and validation of YANG (RFC 6020, RFC 7950) models. Represents parsed YANG models as Python dicts and JavaScript trees. Adds GUI for traversing, searching, and inspecting YANG model trees.	1.12.2

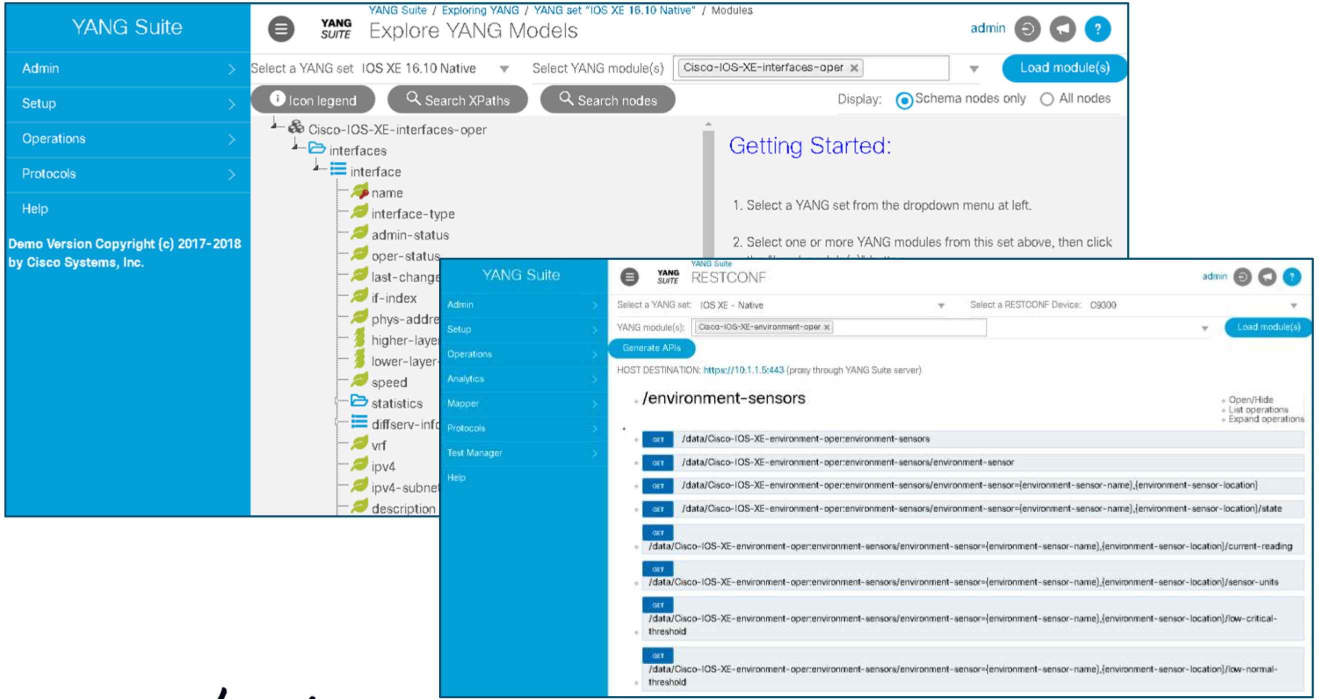
Installed optional plugins

Package name	Description	Installed version
`yangsuite-coverage`	Checks YANG model coverage based on Cisco CLI config	2.2.0
`yangsuite-gnmi`	gRPC Network Management Interface (gNMI) support for YANG Suite	0.4.2
`yangsuite-grpc-telemetry`	gRPC Telemetry support for YANG Suite	0.5.1
`yangsuite-mapper`	Facilitates definition of mappings between analogous YANG data models such as IETF vs. OpenConfig vs. vendor- and device-specific models. Provides for code generation derived from these mappings.	1.0.2

Yang Suite

Demo in World of Solutions

- Internal only at this time… Set to release publicly ~2020
- HTML5 based, unlike YangExplorer which is Flash (EOL 2020)
- gNMI, gRPC, RESTCONF plugins available to generate SwaggerUI, telemetry receiver, etc



YANG Suite / Exploring YANG / YANG set "IOS XE 16.10 Native" / Modules

Explore YANG Models

Select a YANG set: IOS XE 16.10 Native

Select YANG module(s): Cisco-IOS-XE-interfaces-oper ×

Display: ● Schema nodes only ○ All nodes

- Cisco-IOS-XE-interfaces-oper
  - interfaces
    - interface
      - name
      - interface-type
      - admin-status
      - oper-status
      - last-change
      - if-index
      - phys-addre
      - higher-laye
      - lower-layer
      - speed
      - statistics
      - diffserv-info
      - vrf
      - ipv4
      - ipv4-subnet
      - description

Getting Started:

1. Select a YANG set from the dropdown menu at left.

2. Select one or more YANG modules from this set above, then click

Demo Version Copyright (c) 2017-2018 by Cisco Systems, Inc.

YANG Suite

RESTCONF

Select a YANG set: IOS XE - Native

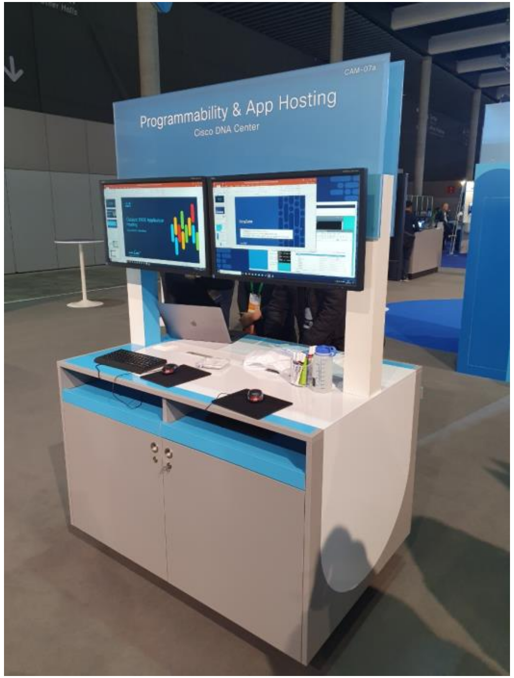
Select a RESTCONF Device: C9300

YANG module(s): Cisco-IOS-XE-environment-oper ×

HOST DESTINATION: https://10.1.1.5:443 (proxy through YANG Suite server)

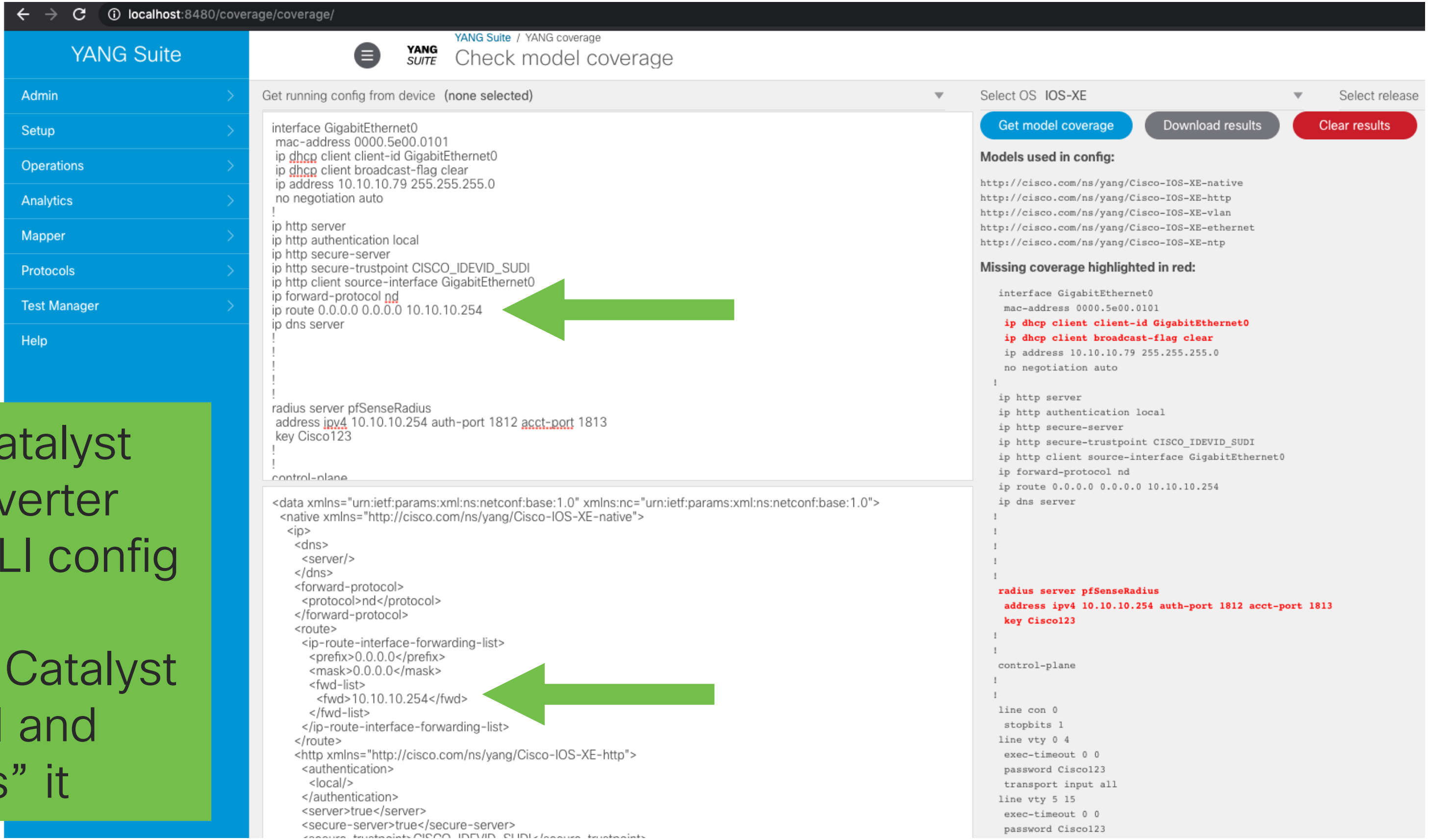
/environment-sensors

```
GET /data/Cisco-IOS-XE-environment-oper:environment-sensors
GET /data/Cisco-IOS-XE-environment-oper:environment-sensors/environment-sensor
GET /data/Cisco-IOS-XE-environment-oper:environment-sensors/environment-sensor={environment-sensor-name},{environment-sensor-location}
GET /data/Cisco-IOS-XE-environment-oper:environment-sensors/environment-sensor={environment-sensor-name},{environment-sensor-location}/state
GET /data/Cisco-IOS-XE-environment-oper:environment-sensors/environment-sensor={environment-sensor-name},{environment-sensor-location}/current-reading
GET /data/Cisco-IOS-XE-environment-oper:environment-sensors/environment-sensor={environment-sensor-name},{environment-sensor-location}/sensor-units
GET /data/Cisco-IOS-XE-environment-oper:environment-sensors/environment-sensor={environment-sensor-name},{environment-sensor-location}/low-critical-threshold
GET /data/Cisco-IOS-XE-environment-oper:environment-sensors/environment-sensor={environment-sensor-name},{environment-sensor-location}/low-normal-threshold
```



cisco Live!

Yang Suite: IOS XE CLI to XML YANG



Get running config from device (none selected)

Select OS IOS-XE

Select release

```
interface GigabitEthernet0
 mac-address 0000.5e00.0101
 ip dhcp client client-id GigabitEthernet0
 ip dhcp client broadcast-flag clear
 ip address 10.10.10.79 255.255.255.0
 no negotiation auto
!
ip http server
ip http authentication local
ip http secure-server
ip http secure-trustpoint CISCO_IDEVID_SUDI
ip http client source-interface GigabitEthernet0
ip forward-protocol nd
ip route 0.0.0.0 0.0.0.0 10.10.10.254
ip dns server
!
!
!
!
!
radius server pfSenseRadius
 address ipv4 10.10.10.254 auth-port 1812 acct-port 1813
 key Cisco123
!
!
control-plane
```

```
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
 <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <ip>
   <dns>
    <server/>
   </dns>
   <forward-protocol>
    <protocol>nd</protocol>
   </forward-protocol>
   <route>
    <ip-route-interface-forwarding-list>
     <prefix>0.0.0.0</prefix>
     <mask>0.0.0.0</mask>
     <fwd-list>
      <fwd>10.10.10.254</fwd>
     </fwd-list>
    </ip-route-interface-forwarding-list>
   </route>
   <http xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-http">
    <authentication>
     <local/>
    </authentication>
    <server>true</server>
    <secure-server>true</secure-server>
    <secure-trustpoint>CISCO_IDEVID_SUDI</secure-trustpoint>
```

Models used in config:

```
http://cisco.com/ns/yang/Cisco-IOS-XE-native
http://cisco.com/ns/yang/Cisco-IOS-XE-http
http://cisco.com/ns/yang/Cisco-IOS-XE-vlan
http://cisco.com/ns/yang/Cisco-IOS-XE-ethernet
http://cisco.com/ns/yang/Cisco-IOS-XE-ntp
```

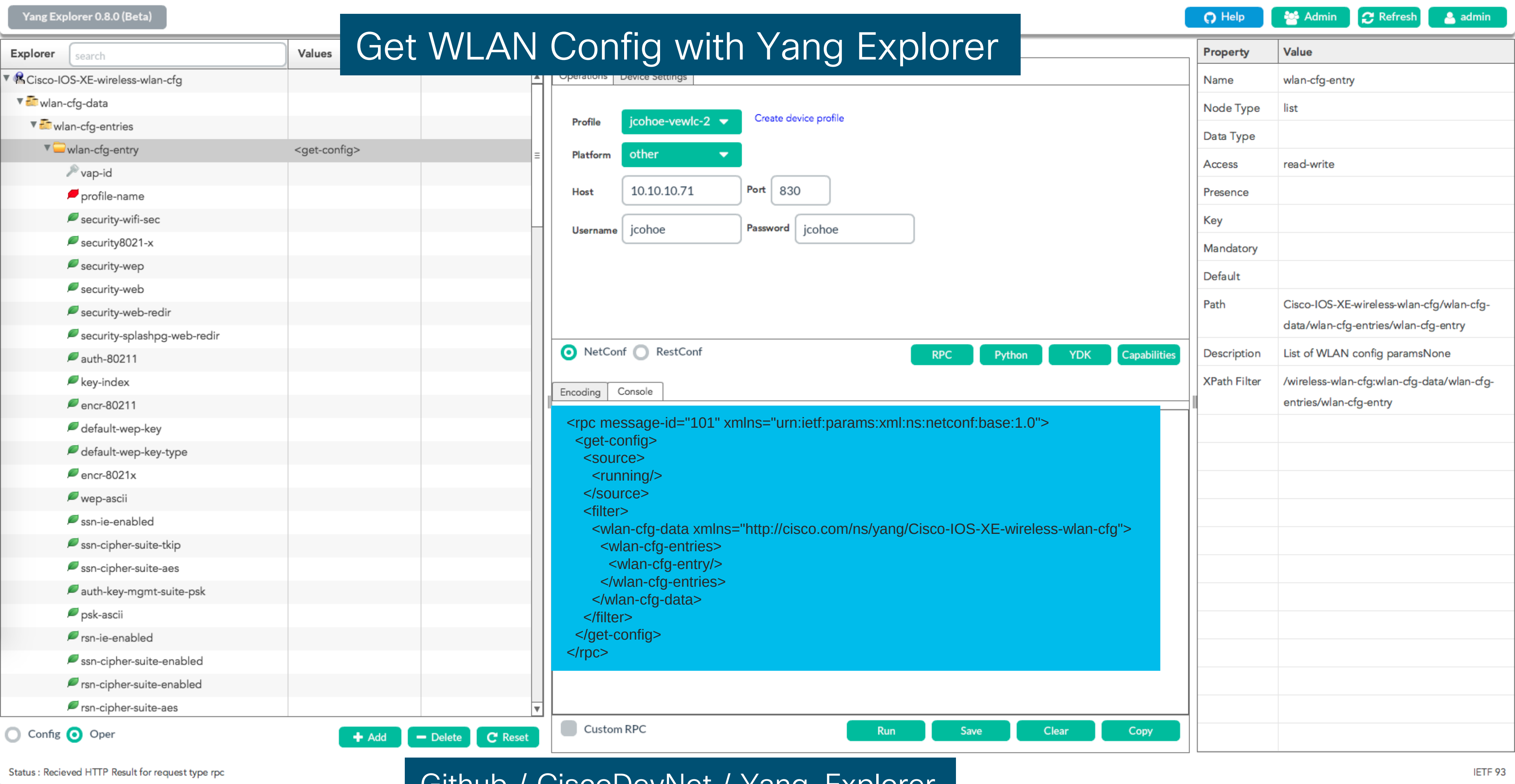
Missing coverage highlighted in red:

```
  interface GigabitEthernet0
   mac-address 0000.5e00.0101
   ip dhcp client client-id GigabitEthernet0
   ip dhcp client broadcast-flag clear
   ip address 10.10.10.79 255.255.255.0
   no negotiation auto
 !
  ip http server
  ip http authentication local
  ip http secure-server
  ip http secure-trustpoint CISCO_IDEVID_SUDI
  ip http client source-interface GigabitEthernet0
  ip forward-protocol nd
  ip route 0.0.0.0 0.0.0.0 10.10.10.254
  ip dns server
 !
 !
 !
 !
 !
  radius server pfSenseRadius
   address ipv4 10.10.10.254 auth-port 1812 acct-port 1813
   key Cisco123
 !
 !
  control-plane
 !
 !
  line con 0
   stopbits 1
  line vty 0 4
   exec-timeout 0 0
   password Cisco123
   transport input all
  line vty 5 15
   exec-timeout 0 0
   password Cisco123
```

AireOS to Catalyst Config Converter available for CLI config

This takes the Catalyst IOS XE CLI and "yangifies" it

Get WLAN Config with Yang Explorer



Profile: jcohoe-vewlc-2

Platform: other

Host: 10.10.10.71 Port: 830

Username: jcohoe Password: jcohoe

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <wlan-cfg-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-wlan-cfg">
        <wlan-cfg-entries>
          <wlan-cfg-entry/>
        </wlan-cfg-entries>
      </wlan-cfg-data>
    </filter>
  </get-config>
</rpc>
```

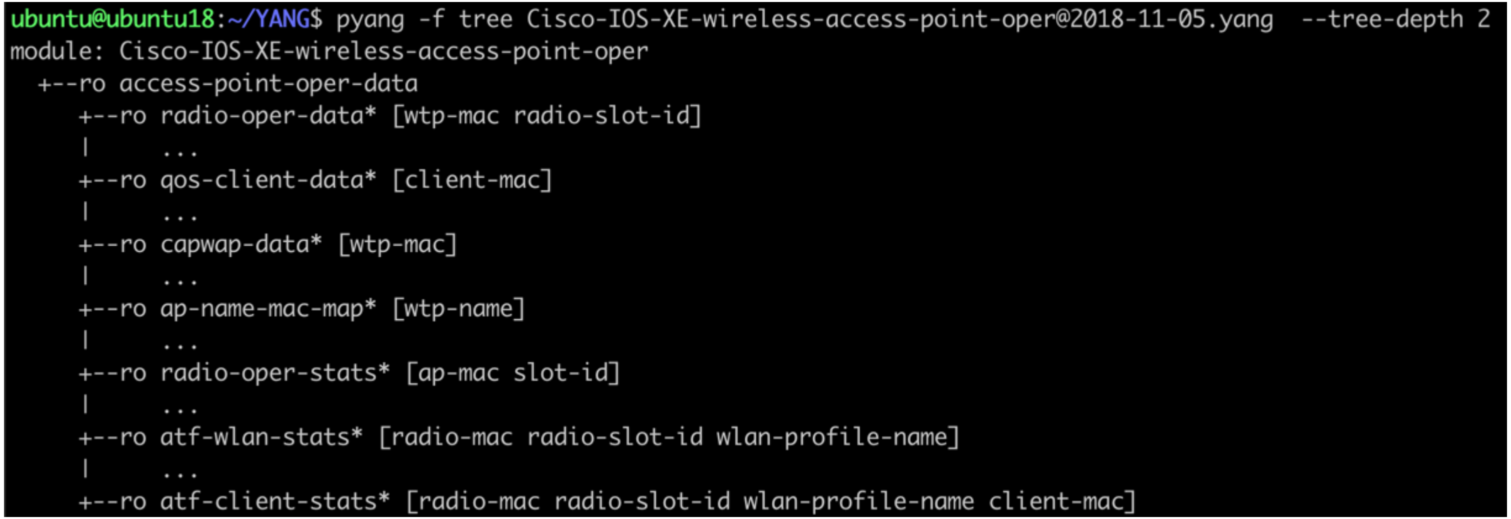
Property	Value
Name	wlan-cfg-entry
Node Type	list
Data Type	
Access	read-write
Presence	
Key	
Mandatory	
Default	
Path	Cisco-IOS-XE-wireless-wlan-cfg/wlan-cfg-data/wlan-cfg-entries/wlan-cfg-entry
Description	List of WLAN config paramsNone
XPath Filter	/wireless-wlan-cfg:wlan-cfg-data/wlan-cfg-entries/wlan-cfg-entry

Status : Recieved HTTP Result for request type rpc

Github / CiscoDevNet / Yang-Explorer

pyang

Use pyang to visualize the structure and data



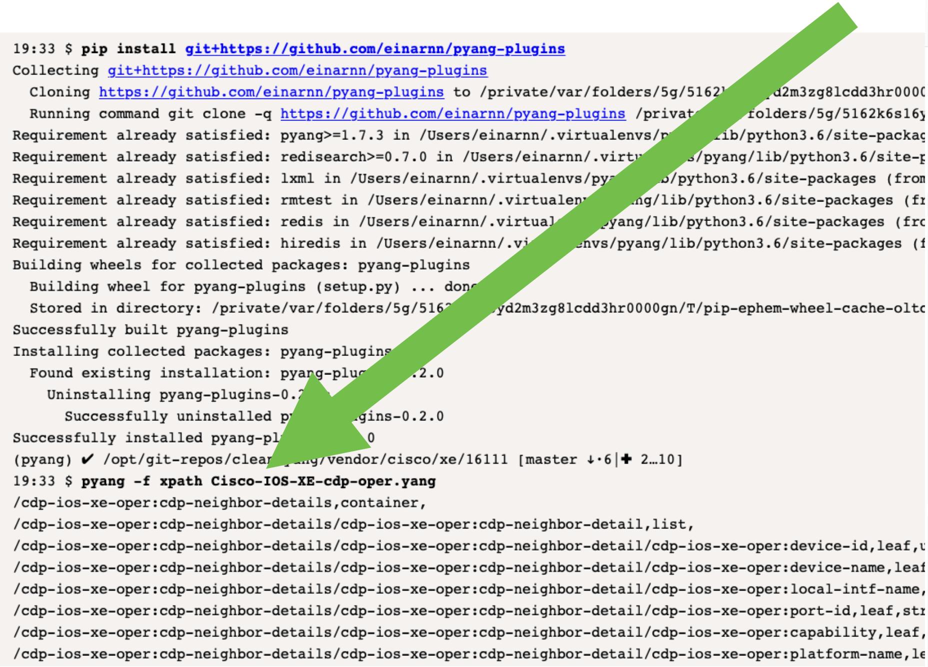
```
ubuntu@ubuntu18:~/YANG$ pyang -f tree Cisco-IOS-XE-wireless-access-point-oper@2018-11-05.yang  --tree-depth 2
module: Cisco-IOS-XE-wireless-access-point-oper
  +--ro access-point-oper-data
     +--ro radio-oper-data* [wtp-mac radio-slot-id]
     |     ...
     +--ro qos-client-data* [client-mac]
     |     ...
     +--ro capwap-data* [wtp-mac]
     |     ...
     +--ro ap-name-mac-map* [wtp-name]
     |     ...
     +--ro radio-oper-stats* [ap-mac slot-id]
     |     ...
     +--ro atf-wlan-stats* [radio-mac radio-slot-id wlan-profile-name]
     |     ...
     +--ro atf-client-stats* [radio-mac radio-slot-id wlan-profile-name client-mac]
```

Install with pip, easy_install, or directly from the Github repository:

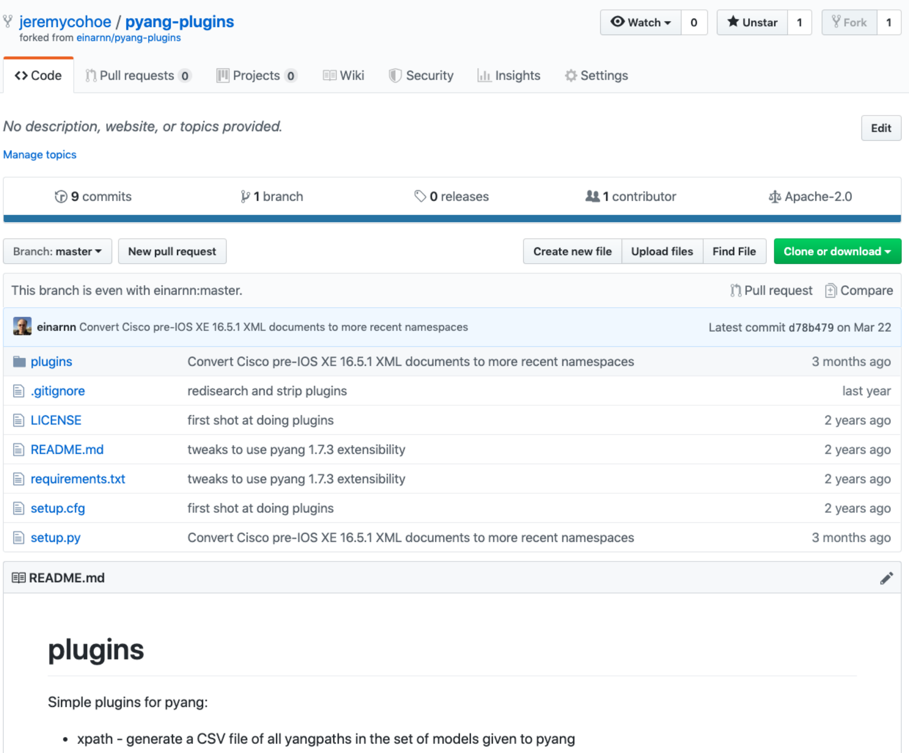
```
$ pip install pyang
$ sudo easy_install pyang
$ git clone https://github.com/mbj4668/pyang
```

The pyang tool can be used to visualize the structure and elements within the YANG model

Generate xpath with pyang



```
19:33 $ pip install git+https://github.com/einarnn/pyang-plugins
Collecting git+https://github.com/einarnn/pyang-plugins
  Cloning https://github.com/einarnn/pyang-plugins to /private/var/folders/5g/5162[illegible]d2m3zg8lcdd3hr000
  Running command git clone -q https://github.com/einarnn/pyang-plugins /privat[illegible]folders/5g/5162k6s16y
Requirement already satisfied: pyang>=1.7.3 in /Users/einarnn/.virtualenvs/p[illegible]ib/python3.6/site-packag
Requirement already satisfied: redisearch>=0.7.0 in /Users/einarnn/.virtu[illegible]s/pyang/lib/python3.6/site-p
Requirement already satisfied: lxml in /Users/einarnn/.virtualenvs/py[illegible]b/python3.6/site-packages (from
Requirement already satisfied: rmtest in /Users/einarnn/.virtualen[illegible]ng/lib/python3.6/site-packages (fr
Requirement already satisfied: redis in /Users/einarnn/.virtual[illegible]pyang/lib/python3.6/site-packages (fro
Requirement already satisfied: hiredis in /Users/einarnn/.vi[illegible]envs/pyang/lib/python3.6/site-packages (f
Building wheels for collected packages: pyang-plugins
  Building wheel for pyang-plugins (setup.py) ... don
  Stored in directory: /private/var/folders/5g/516[illegible]yd2m3zg8lcdd3hr0000gn/T/pip-ephem-wheel-cache-oltc
Successfully built pyang-plugins
Installing collected packages: pyang-plugins
  Found existing installation: pyang-plu[illegible].2.0
    Uninstalling pyang-plugins-0.2[illegible]
      Successfully uninstalled p[illegible]gins-0.2.0
Successfully installed pyang-pl[illegible]0
(pyang) ✔ /opt/git-repos/clear[illegible]/vendor/cisco/xe/16111 [master ↓·6|✚ 2…10]
19:33 $ pyang -f xpath Cisco-IOS-XE-cdp-oper.yang
/cdp-ios-xe-oper:cdp-neighbor-details,container,
/cdp-ios-xe-oper:cdp-neighbor-details/cdp-ios-xe-oper:cdp-neighbor-detail,list,
/cdp-ios-xe-oper:cdp-neighbor-details/cdp-ios-xe-oper:cdp-neighbor-detail/cdp-ios-xe-oper:device-id,leaf,u
/cdp-ios-xe-oper:cdp-neighbor-details/cdp-ios-xe-oper:cdp-neighbor-detail/cdp-ios-xe-oper:device-name,leaf
/cdp-ios-xe-oper:cdp-neighbor-details/cdp-ios-xe-oper:cdp-neighbor-detail/cdp-ios-xe-oper:local-intf-name,
/cdp-ios-xe-oper:cdp-neighbor-details/cdp-ios-xe-oper:cdp-neighbor-detail/cdp-ios-xe-oper:port-id,leaf,str
/cdp-ios-xe-oper:cdp-neighbor-details/cdp-ios-xe-oper:cdp-neighbor-detail/cdp-ios-xe-oper:capability,leaf,
/cdp-ios-xe-oper:cdp-neighbor-details/cdp-ios-xe-oper:cdp-neighbor-detail/cdp-ios-xe-oper:platform-name,le
```



jeremycohoe / pyang-plugins
forked from einarnn/pyang-plugins

No description, website, or topics provided.

9 commits · 1 branch · 0 releases · 1 contributor · Apache-2.0

This branch is even with einarnn:master.

einarnn Convert Cisco pre-IOS XE 16.5.1 XML documents to more recent namespaces — Latest commit d78b479 on Mar 22

plugins	Convert Cisco pre-IOS XE 16.5.1 XML documents to more recent namespaces	3 months ago
.gitignore	redisearch and strip plugins	last year
LICENSE	first shot at doing plugins	2 years ago
README.md	tweaks to use pyang 1.7.3 extensibility	2 years ago
requirements.txt	tweaks to use pyang 1.7.3 extensibility	2 years ago
setup.cfg	first shot at doing plugins	2 years ago
setup.py	Convert Cisco pre-IOS XE 16.5.1 XML documents to more recent namespaces	3 months ago

README.md

plugins

Simple plugins for pyang:

- xpath - generate a CSV file of all yangpaths in the set of models given to pyang

Zero Touch Provisioning (ZTP)

Time check !

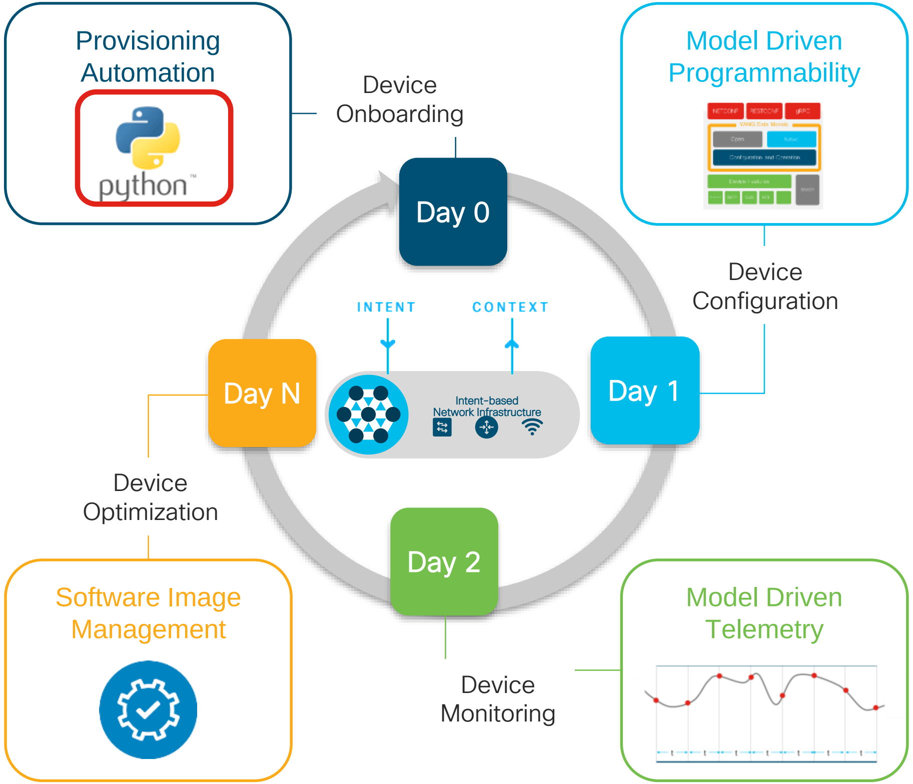
Bonus Round ? !

IOS XE Delivers ...

Pre-boot Execution Environment

Zero Touch Provisioning

Plug and Play



Provisioning Automation

Device Onboarding

Model Driven Programmability

Device Configuration

Day 0

Day 1

Day 2

Day N

INTENT

CONTEXT

Intent-based Network Infrastructure

Device Optimization

Software Image Management

Device Monitoring

Model Driven Telemetry

Network Configuration Protocols

YANG Data Models

Guest Shell (On-Box Python)

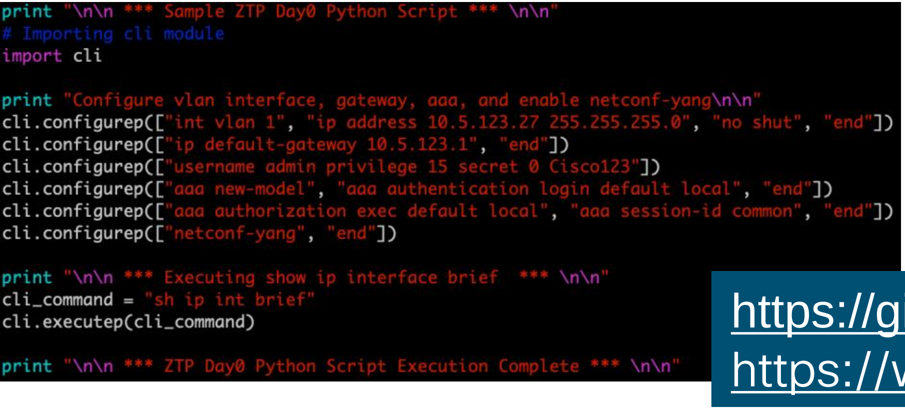
Application Hosting

Telemetry

Python Modules - API

3 Python modules are available that are the API between Guest Shell and the IOS XE device:

- cli.cli, cli.clip
- cli.execute, cli.executep
- cli.configure, cli.configurep



```
print "\n\n *** Sample ZTP Day0 Python Script *** \n\n"
# Importing cli module
import cli

print "Configure vlan interface, gateway, aaa, and enable netconf-yang\n\n"
cli.configurep(["int vlan 1", "ip address 10.5.123.27 255.255.255.0", "no shut", "end"])
cli.configurep(["ip default-gateway 10.5.123.1", "end"])
cli.configurep(["username admin privilege 15 secret 0 Cisco123"])
cli.configurep(["aaa new-model", "aaa authentication login default local", "end"])
cli.configurep(["aaa authorization exec default local", "aaa session-id common", "end"])
cli.configurep(["netconf-yang", "end"])

print "\n\n *** Executing show ip interface brief  *** \n\n"
cli_command = "sh ip int brief"
cli.executep(cli_command)

print "\n\n *** ZTP Day0 Python Script Execution Complete *** \n\n"
```

1. cli.cli(command) —This function takes an IOS command as an argument, runs the command through the IOS parser, and returns the resulting text.

2. cli.execute(command) —This function executes a single EXEC command and returns the output; however, does not print the resulting text. No semicolons or newlines are allowed as part of this command. Use a Python list with a for-loop to execute this function more than once.

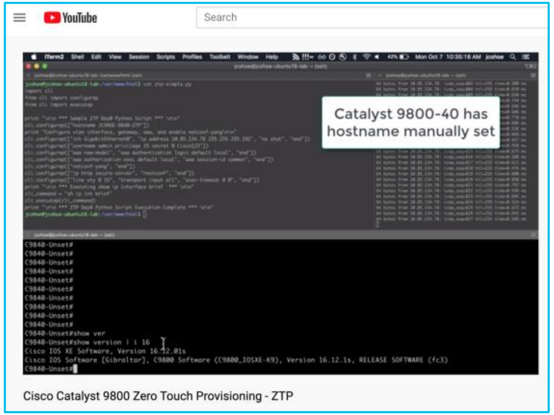
3. cli.configure(command) —This function configures the device with the configuration available in commands. It returns a list of named tuples that contains the command and its result

4, 5, 6: cli.{cli, execute, configure}p (command) — This function works exactly the same as the other functions, except that it prints the resulting text to *stdout* rather than returning it .

https://github.com/jeremycohoe/catalyst9840-ztp

https://www.youtube.com/watch?v=qVkXd1nWGVY

ZTP



```
jcohoe@jcohoe-ubuntu18-lab:/var/www/html$ cat ztp-simple.py
import cli
from cli import configurep
from cli import executep

print "\n\n *** Sample ZTP Day0 Python Script *** \n\n"
cli.configurep(["hostname JCOHOE-9800-ZTP"])
print "Configure vlan interface, gateway, aaa, and enable netconf-yang\n\n"
cli.configurep(["int GigabitEthernet0", "ip address 10.85.134.78 255.255.255.192", "no shut", "end"])
cli.configurep(["username admin privilege 15 secret 0 Cisco123"])
cli.configurep(["aaa new-model", "aaa authentication login default local", "end"])
cli.configurep(["aaa authorization exec default local", "aaa session-id common", "end"])
cli.configurep(["netconf-yang", "end"])
cli.configurep(["ip http secure-server", "restconf", "end"])
cli.configurep(["line vty 0 15", "transport input all", "exec-timeout 0 0", "end"])
print "\n\n *** Executing show ip interface brief  *** \n\n"
cli_command = "sh ip int brief"
cli.executep(cli_command)
print "\n\n *** ZTP Day0 Python Script Execution Complete *** \n\n"
jcohoe@jcohoe-ubuntu18-lab:/var/www/html$
```

```
C9840-Unset#
C9840-Unset#
C9840-Unset#
C9840-Unset#
C9840-Unset#
C9840-Unset#
C9840-Unset#
C9840-Unset#
C9840-Unset#
C9840-Unset#
C9840-Unset#show ver
C9840-Unset#show version | i 16
Cisco IOS XE Software, Version 16.12.01s
Cisco IOS Software [Gibraltar], C9800 Software (C9800_IOSXE-K9), Version 16.12.1s, RELEASE SOFTWARE (fc3)
C9840-Unset#
```

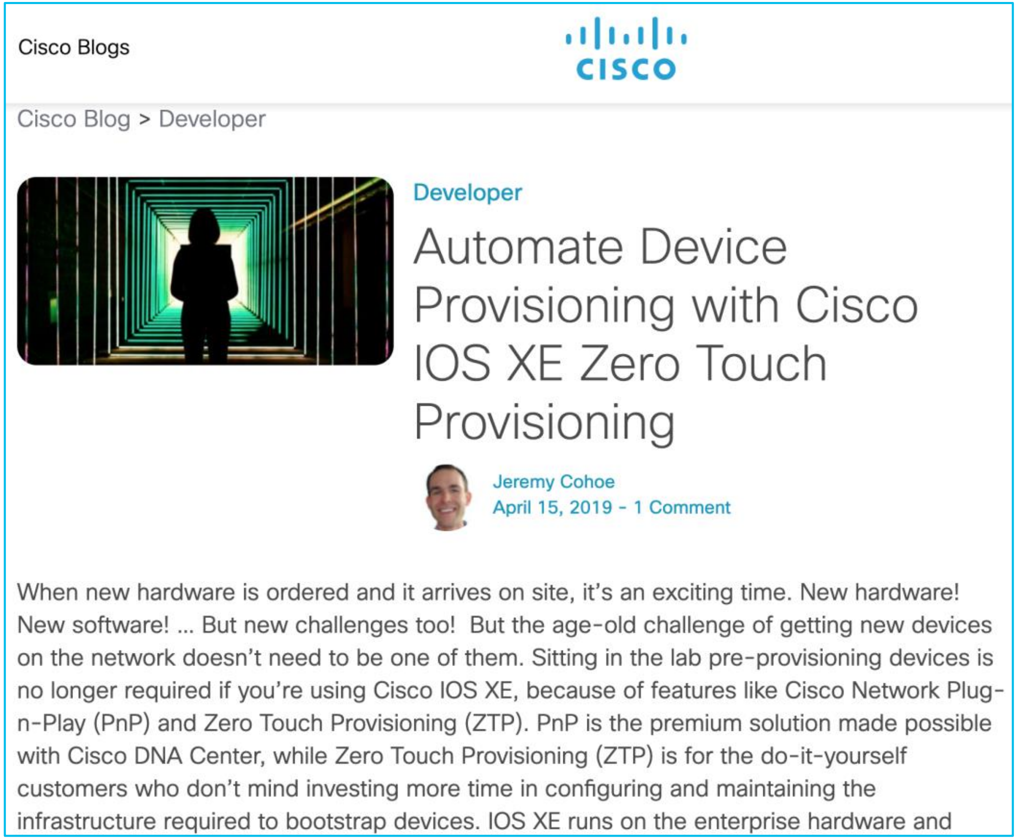
Catalyst 9800-40 has hostname manually set

Cisco Catalyst 9800 Zero Touch Provisioning - ZTP

1. When an IOS XE device boots and no configuration is present, the device will issue a DHCP request on the management port and on the front panel port.

2. If the DHCP response contains option 67 then ZTP is initiated and the device will retrieve and execute the python script from within the Guestshell

3. Guestshell is started and networking is automatically configured



Cisco Blogs

Cisco Blog > Developer

Developer

Automate Device Provisioning with Cisco IOS XE Zero Touch Provisioning

Jeremy Cohoe
April 15, 2019 - 1 Comment

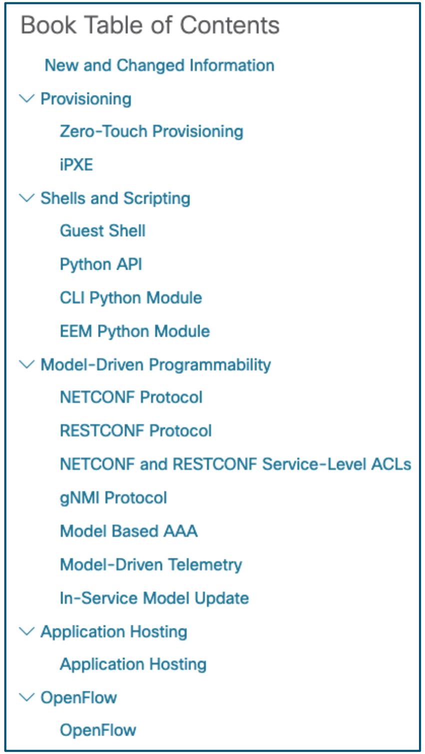
When new hardware is ordered and it arrives on site, it's an exciting time. New hardware! New software! ... But new challenges too! But the age-old challenge of getting new devices on the network doesn't need to be one of them. Sitting in the lab pre-provisioning devices is no longer required if you're using Cisco IOS XE, because of features like Cisco Network Plug-n-Play (PnP) and Zero Touch Provisioning (ZTP). PnP is the premium solution made possible with Cisco DNA Center, while Zero Touch Provisioning (ZTP) is for the do-it-yourself customers who don't mind investing more time in configuring and maintaining the infrastructure required to bootstrap devices. IOS XE runs on the enterprise hardware and

https://www.youtube.com/watch?v=EAXnftG6odg
https://blogs.cisco.com/developer/device-provisioning-with-ios-xe-zero-touch-provisioning

Closing

Programmability Configuration Guide

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1612/b_1612_programmability_cg.html



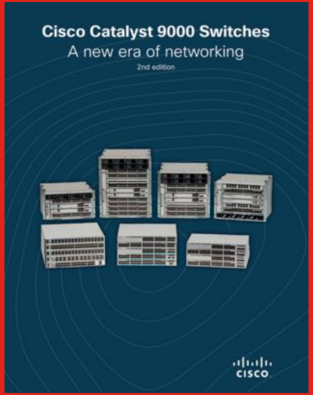
Book Table of Contents

- New and Changed Information
- Provisioning
  - Zero-Touch Provisioning
  - iPXE
- Shells and Scripting
  - Guest Shell
  - Python API
  - CLI Python Module
  - EEM Python Module
- Model-Driven Programmability
  - NETCONF Protocol
  - RESTCONF Protocol
  - NETCONF and RESTCONF Service-Level ACLs
  - gNMI Protocol
  - Model Based AAA
  - Model-Driven Telemetry
  - In-Service Model Update
- Application Hosting
  - Application Hosting
- OpenFlow
  - OpenFlow

Programmability Configuration Guide, Cisco IOS XE Gibraltar 16.12.x

Enterprise Networks Booksprints

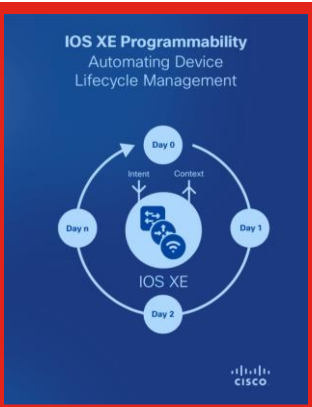
http://cs.co/cat9000book
http://cs.co/sdabook
http://cs.co/wirelessbook
http://cs.co/programmabilitybook
http://cs.co/assurancebook
http://cs.co/sdwanbook



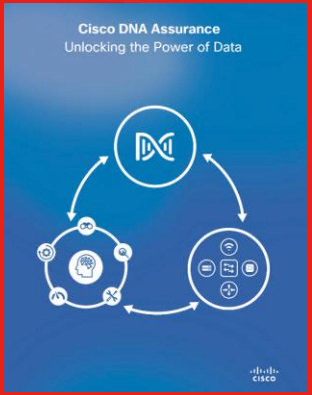
Cisco Catalyst 9000 Switches
A new era of networking
2nd edition

Cisco Software-Defined Access
Enabling intent-based networking
2nd edition

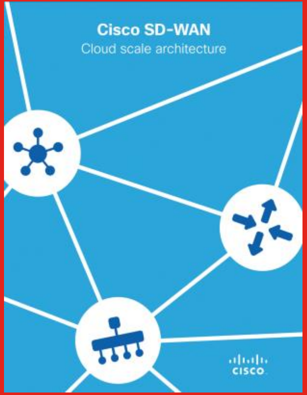
Cisco Enterprise Wireless
Intuitive Wi-Fi starts here
2nd edition



IOS XE Programmability
Automating Device
Lifecycle Management

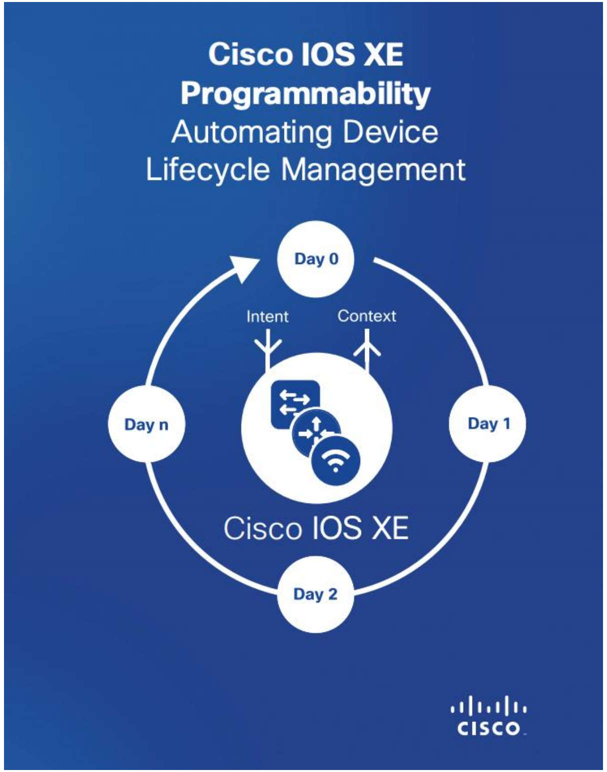


Cisco DNA Assurance
Unlocking the Power of Data



Cisco SD-WAN
Cloud scale architecture

Cisco IOS XE Programmability – Booksprint Book



Cisco IOS XE
Programmability
Automating Device
Lifecycle Management

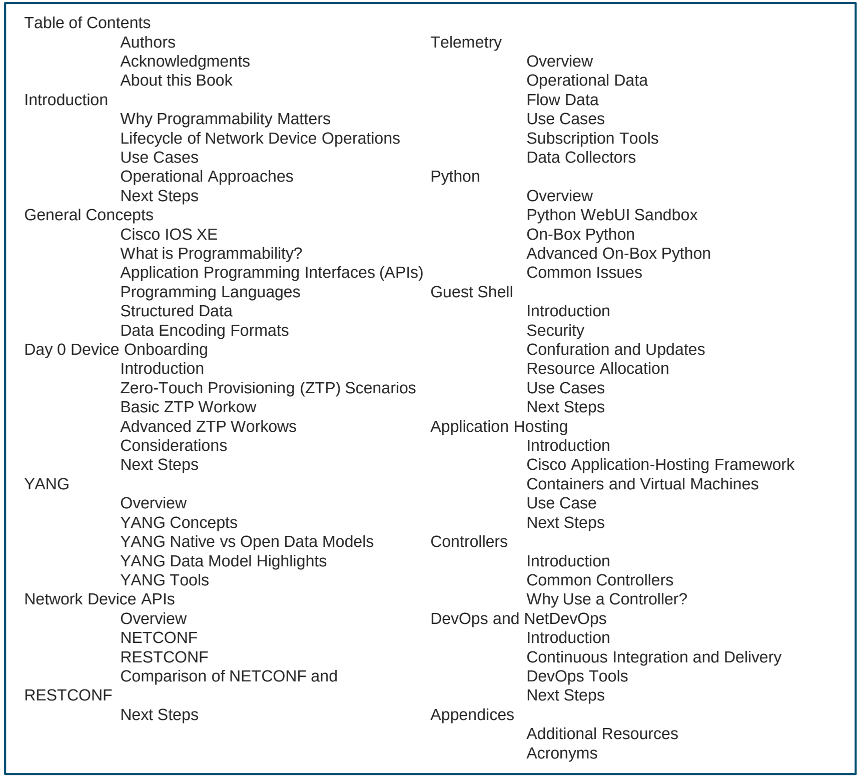
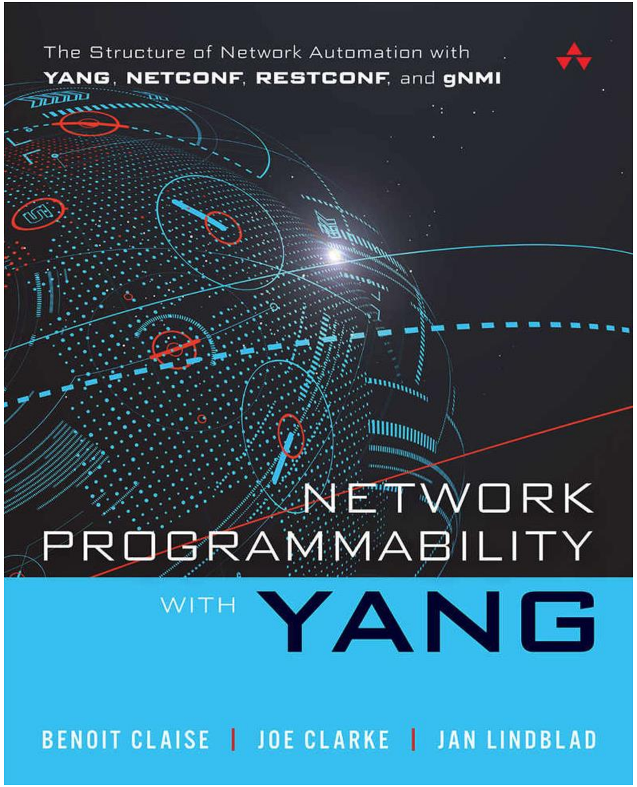


Table of Contents

https://www.cisco.com/c/dam/en/us/products/collateral/enterprise-networks/nb-06-ios-xe-prog-ebook-cte-en.pdf

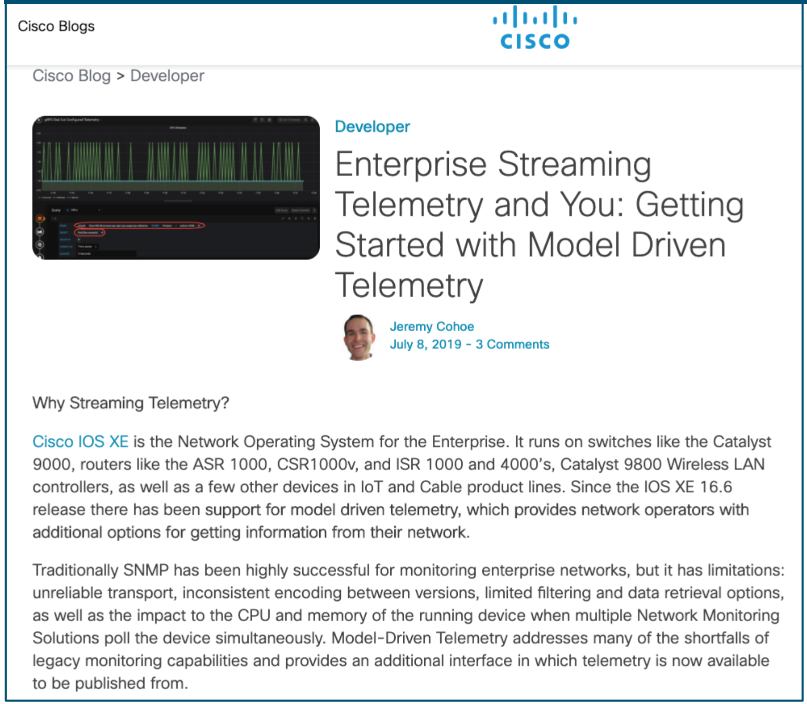
Network Programmability with YANG


The Structure of Network Automation with
YANG, NETCONF, RESTCONF, and gNMI
NETWORK
PROGRAMMABILITY
WITH YANG
BENOIT CLAISE | JOE CLARKE | JAN LINDBLAD

https://www.amazon.com/Network-Programmability-YANG-Modeling-driven-Management/dp/0135180392

Blogs on Cisco.com: Streaming Telemetry, ZTP

https://blogs.cisco.com/developer/device-provisioning-with-ios-xe-zero-touch-provisioning
https://blogs.cisco.com/developer/getting-started-with-model-driven-telemetry



Cisco Blogs

Cisco Blog > Developer

Developer

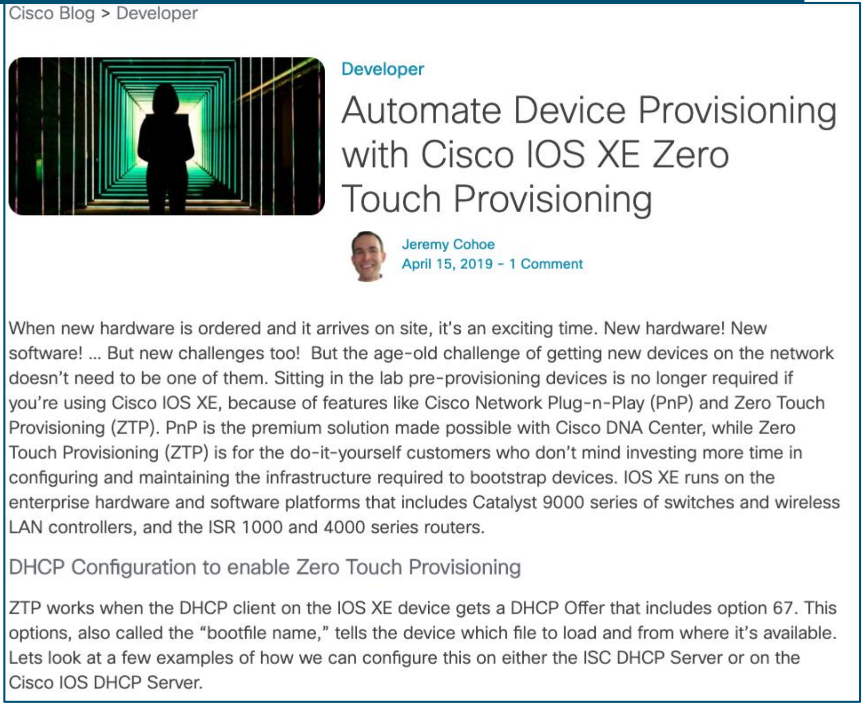
Enterprise Streaming Telemetry and You: Getting Started with Model Driven Telemetry

Jeremy Cohoe
July 8, 2019 - 3 Comments

Why Streaming Telemetry?

Cisco IOS XE is the Network Operating System for the Enterprise. It runs on switches like the Catalyst 9000, routers like the ASR 1000, CSR1000v, and ISR 1000 and 4000's, Catalyst 9800 Wireless LAN controllers, as well as a few other devices in IoT and Cable product lines. Since the IOS XE 16.6 release there has been support for model driven telemetry, which provides network operators with additional options for getting information from their network.

Traditionally SNMP has been highly successful for monitoring enterprise networks, but it has limitations: unreliable transport, inconsistent encoding between versions, limited filtering and data retrieval options, as well as the impact to the CPU and memory of the running device when multiple Network Monitoring Solutions poll the device simultaneously. Model-Driven Telemetry addresses many of the shortfalls of legacy monitoring capabilities and provides an additional interface in which telemetry is now available to be published from.



Cisco Blog > Developer

Developer

Automate Device Provisioning with Cisco IOS XE Zero Touch Provisioning

Jeremy Cohoe
April 15, 2019 - 1 Comment

When new hardware is ordered and it arrives on site, it's an exciting time. New hardware! New software! … But new challenges too! But the age-old challenge of getting new devices on the network doesn't need to be one of them. Sitting in the lab pre-provisioning devices is no longer required if you're using Cisco IOS XE, because of features like Cisco Network Plug-n-Play (PnP) and Zero Touch Provisioning (ZTP). PnP is the premium solution made possible with Cisco DNA Center, while Zero Touch Provisioning (ZTP) is for the do-it-yourself customers who don't mind investing more time in configuring and maintaining the infrastructure required to bootstrap devices. IOS XE runs on the enterprise hardware and software platforms that includes Catalyst 9000 series of switches and wireless LAN controllers, and the ISR 1000 and 4000 series routers.

DHCP Configuration to enable Zero Touch Provisioning

ZTP works when the DHCP client on the IOS XE device gets a DHCP Offer that includes option 67. This options, also called the "bootfile name," tells the device which file to load and from where it's available. Lets look at a few examples of how we can configure this on either the ISC DHCP Server or on the Cisco IOS DHCP Server.

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education

Demos in the Cisco Showcase

Walk-In Labs

Meet the Engineer 1:1 meetings

Related sessions

Thank you

cisco

You make **possible**