



Cisco Enterprise Lab

#Devnet #DNA

Akshaya Kumar Sankaran
Systems Engineer, CCIE #28296
Dec 5 2018

Rev: January 2018

Before we proceed...

- Open Lab guide in your browser - <http://cs.co/denverlab18>
- Make sure you completed Steps 1 & 2 (on your desk)
 - Step 1 Connect using Anyconnect VPN Pod
 - Step 2 RDP in to CentOS machine
 - Check your desk for credentials

Note: If you don't have Anyconnect/RDP the links are in Lab guide

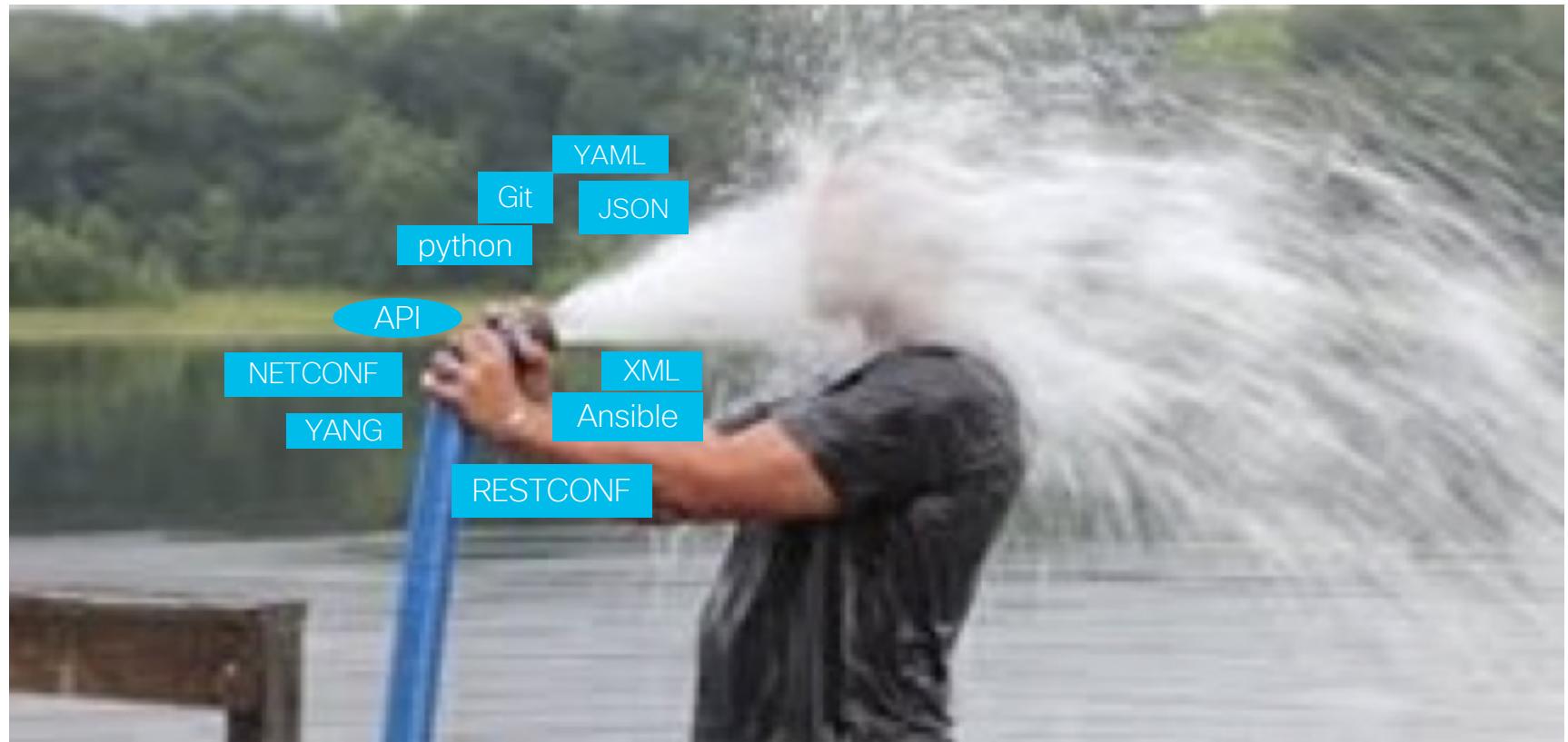
Let's get started ...

Why we are here

*“Civilization advances
by extending the number of important
operations which we can
perform without thinking about them.”*

Alfred North Whitehead, Mathematician and Philosopher

But one problem .. Too many things coming at us



How to be a network engineer in programmable age

Phase1

- Python,
- REST APIs,
JSON/XML,
- Git, GitHub

As Needed

- IOS XE
- Network Controller

Phase2

- Linux,
- Ansible,
- Docker,
- NETCONF/RESTCONF/YANG

As Needed

- IOT Networking,
- Cloud Networking

Phase3

- Linux Networking,
- Container Networking,
- NFV

As Needed

- NFV,
- “DevOps”

(Recap)

Connecting to your Lab Pod & CentOS RDP

- 1) Open <http://cs.co/denverlab18>
- 2) Use Cisco AnyConnect or OpenConnect – VPN to log into your pod using info from proctor
 - Anyconnect Host:
<https://dcloud-sjc-anyconnect.cisco.com>
- 3) RDP to CentOS (please install RDP client in your machine if not available already)
 - * Address: 198.18.134.48
 - * Credentials: developer / C1sco12345

The Lab Environment

Per Attendee dCloud Pod

DEVBOX

198.18.134.48
developer / C1sco12345

CSR1

198.18.134.11
admin /
C1sco12345

NFVIS

198.18.134.46
admin / C1sco12345_

CSR2

198.18.134.12
admin / C1sco12345

Access Options:

- VPN into Pod with AnyConnect or OpenConnect
- RDP into DevBox

Shared DevNet Always On Sandboxes

DNA Center

<http://cs.co/9007DV8rl>
devnetuser / Cisco123!

Meraki

<http://cs.co/9001DV8Vc>
devnetmeraki@cisco.com /
ilovemeraki

CMX

<http://cs.co/9005DV8T9>
learning@cisco.com /
learning

Access Options:

- Public web portal and API access

Lets Run some Program(1/3 contd..)

> Open Terminal 'centos'

```
[root@centos]# cd code
```

```
[root@centos code]# source venv/bin/activate
```

```
(venv) [root@centos code]# pip list
```

```
<output>...
```

```
(venv) [root@centos code]# cd DenverCC
```



Ready to Run the scripts (2/3)

API Test

Open terminal & Run this command

```
curl https://api.icndb.com/jokes/random
```

Ready to Run the scripts (3/3)

Lab Guide – 2a – 2c

```
(venv) [root@centos DenverCC]# python api_netmiko.py
```

```
(venv) [root@centos DenverCC]# python api_requests.py
```

```
(venv) [root@centos DenverCC]# python api_netconf.py
```

Next, Lets Review What Happened..

Chapter 1 : Device Automation

Getting Structured Data from the system...

Configuration vs. Operational data

Configuration data tells the device what to do. It is data that you see in a “show run”.

```
# sh run int mgmt0  
  
interface mgmt0  
    vrf member management  
    ip address 172.26.244.162/24
```

Operational data tells us how a device is operating, from show commands other than “show run”.

```
# sh int mgmt0  
mgmt0 is up  
admin state is up  
Internet Address is 172.26.244.162/24  
110380 input packets
```

We can **write** configuration data (think “conf t”), and we can **read** configuration data (think “show run”).

Operational data is **read-only**.
(think “show int”)

Note inconsistent “key” format!

```
switch1# sh int e1/10
Ethernet1/10 is up
Hardware: 1000/10000 Ethernet, address: 0005.73d0.9331 (bia 0005.73d0.9331)
Description: To UCS-11
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Switchport monitor is off
EtherType is 0x8100
Last link flapped 8week(s) 2day(s)
Last clearing of "show interface" counters 1d02h
30 seconds input rate 944 bits/sec, 118 bytes/sec, 0 packets/sec
30 seconds output rate 3110376 bits/sec, 388797 bytes/sec, 5221 packets/sec
```

CLI can be considered **unstructured** data. In this output there are “tags” indicating what data is here, they are not consistent, standardized, or easy to parse.

Data Models

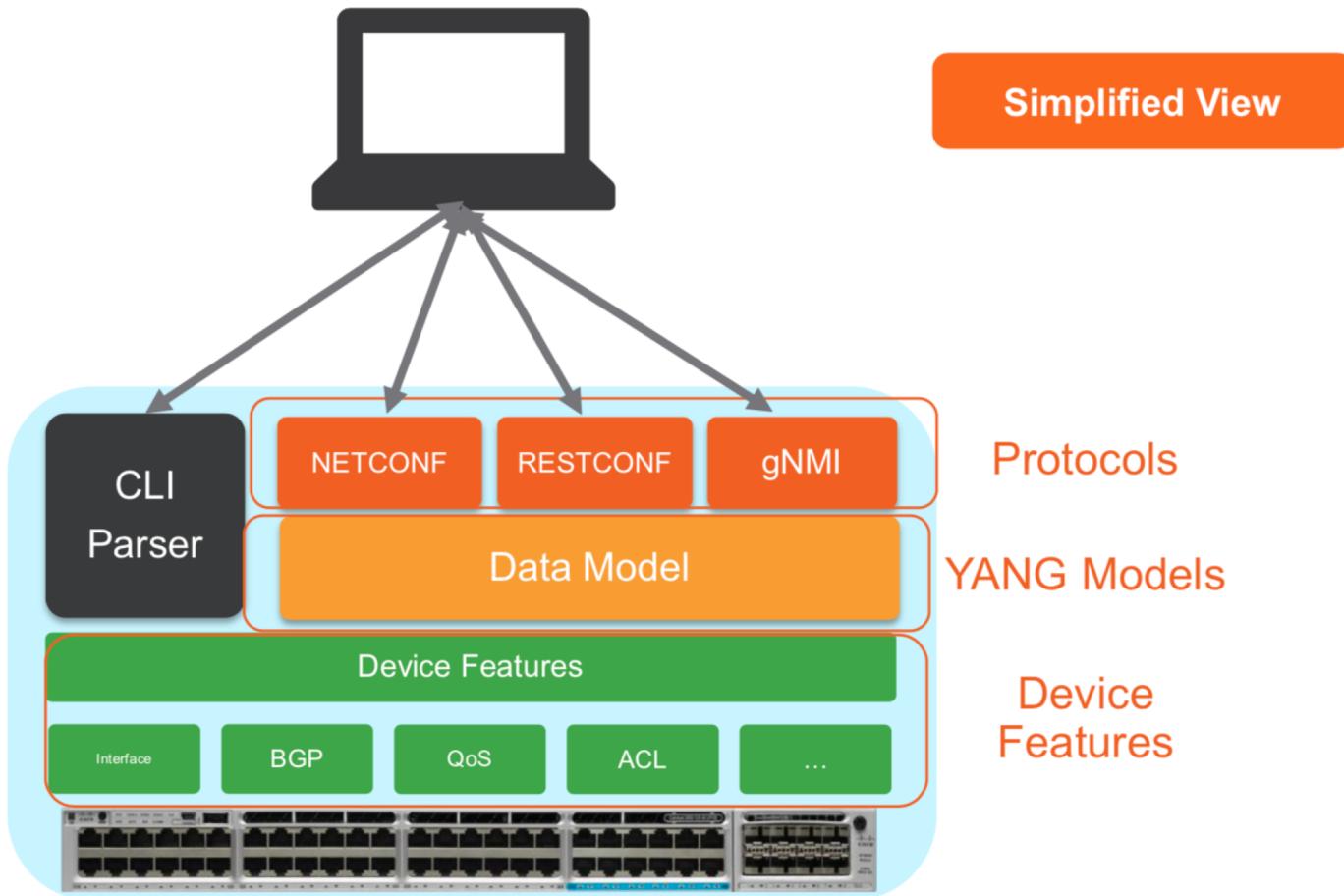
*“A Data-Model Explicitly and precisely **defines Data Structure, Syntax and Semantics**”*

Interface Model **definition**

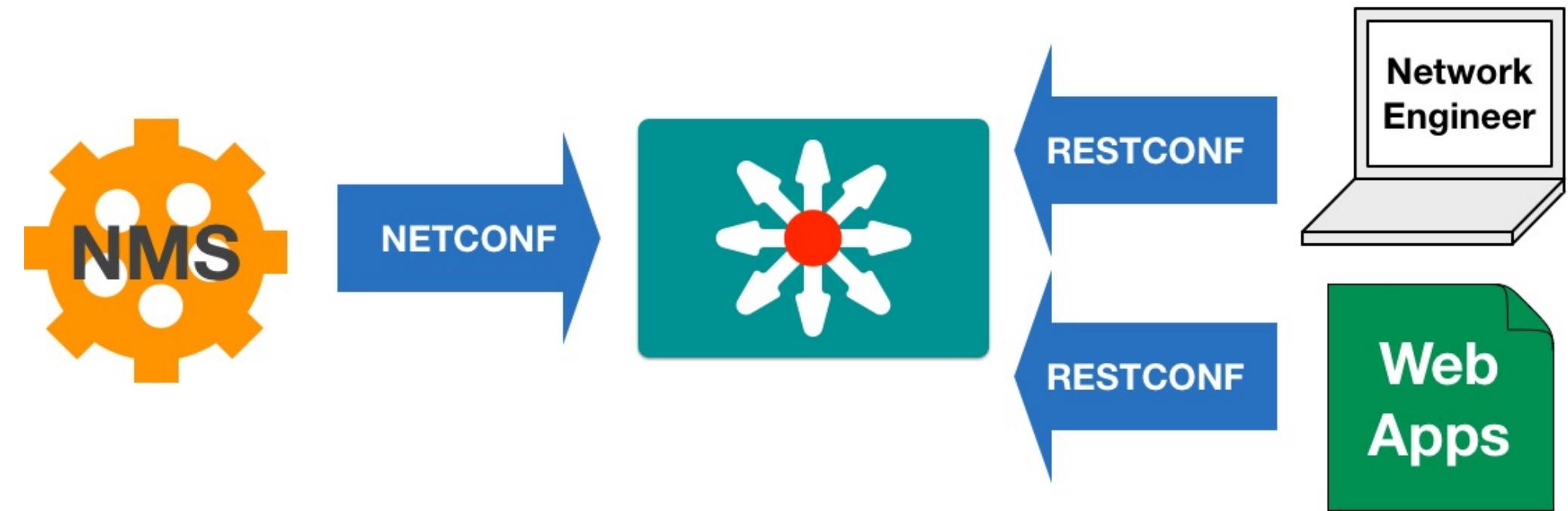
```
ietf-interfaces@2014-05-08.yang •  
/*  
 * Configuration data nodes  
 */  
container interfaces {  
    description  
        "Interface configuration parameters."  
    list interface {  
        key "name";  
        description  
        leaf name {  
            type string;  
        }  
        leaf description {  
            type string;  
        }  
        leaf type {  
            type identityref {  
                base interface-type;  
            }  
            mandatory true;  
        }  
        leaf enabled {  
            type boolean;  
            default "true";  
        }  
    }  
}
```

IOS-XE 16.x

Open Standards Based API's

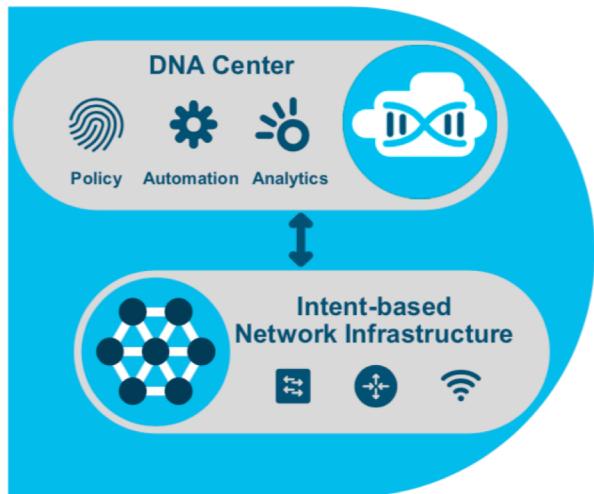


Standard Network Management

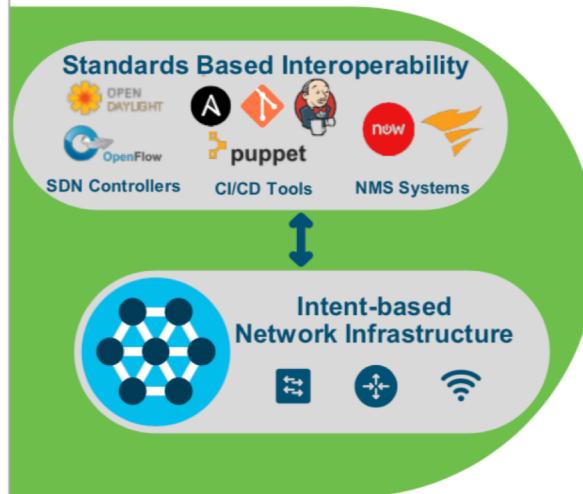


IOS XE 16.X API's enable...

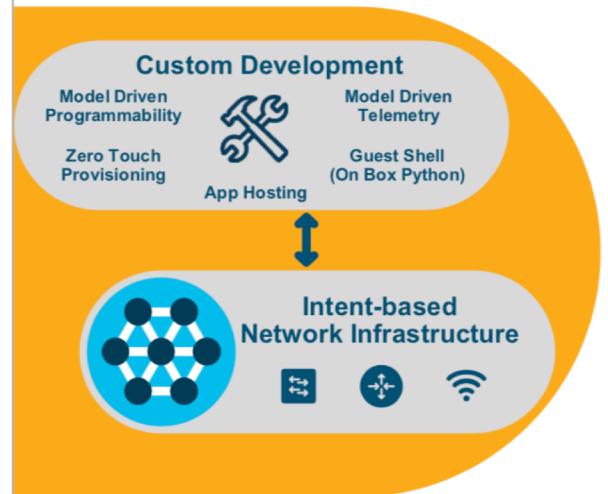
Cisco Solutions



3rd Party Integration

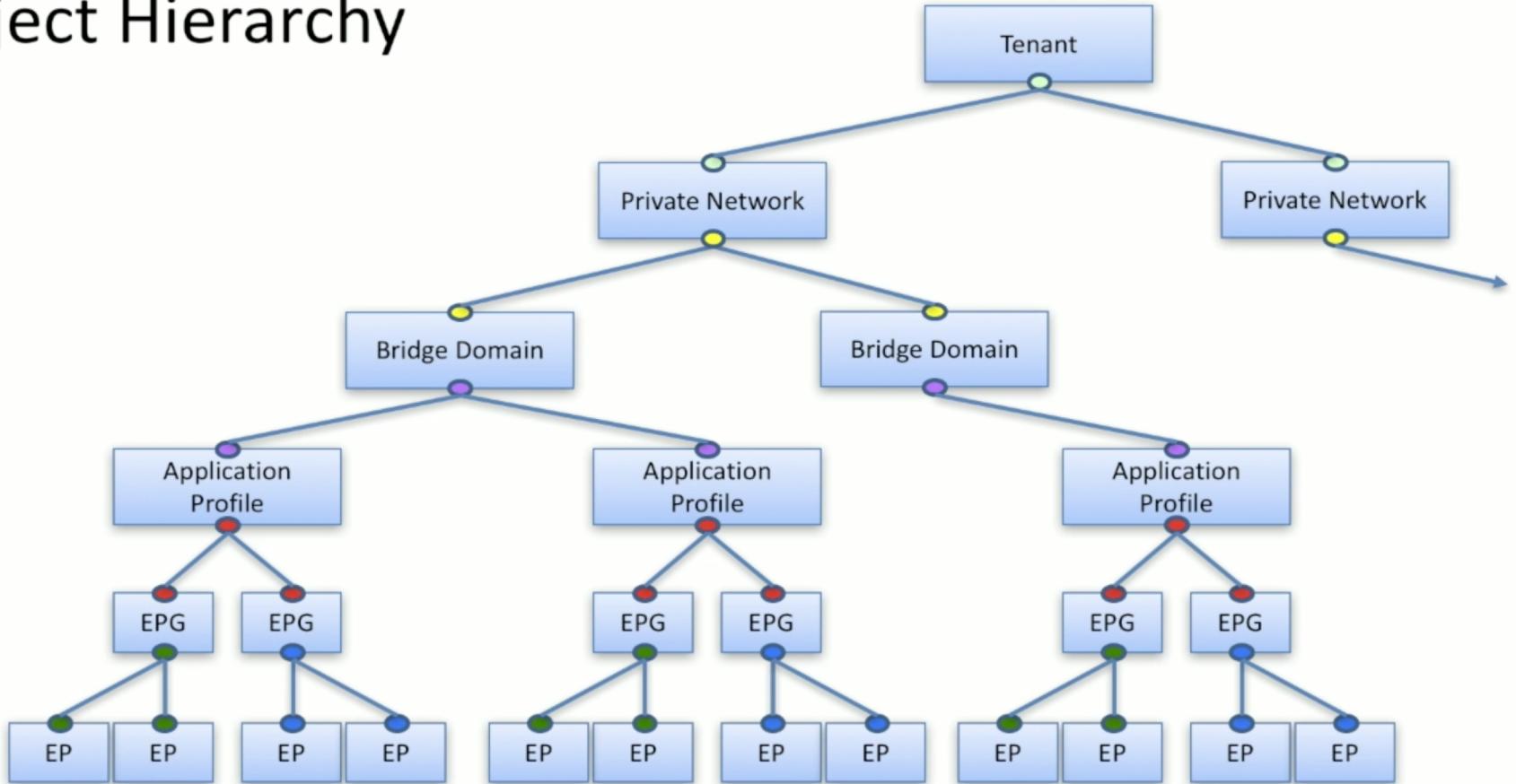


DIY Solutions



Another example : ACI Data Model

Object Hierarchy

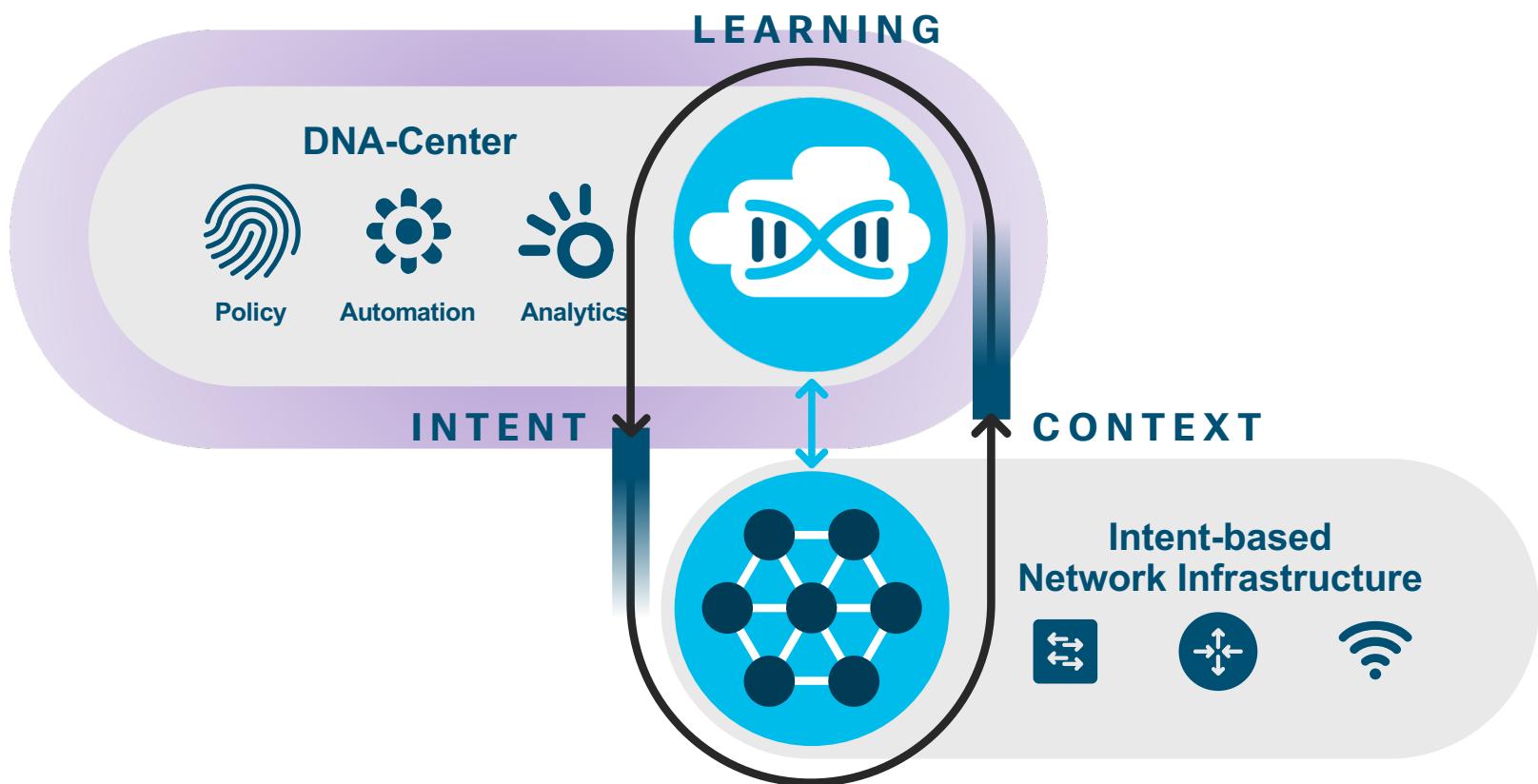


Hands-on with Postman

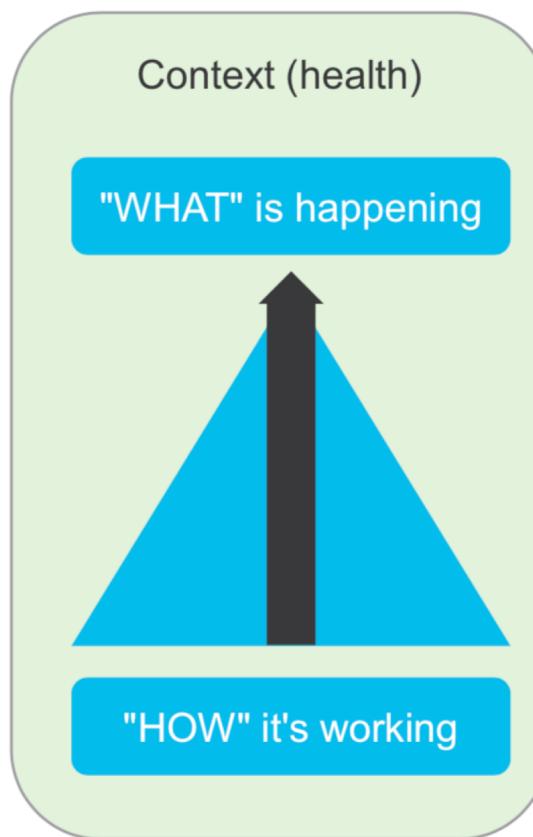
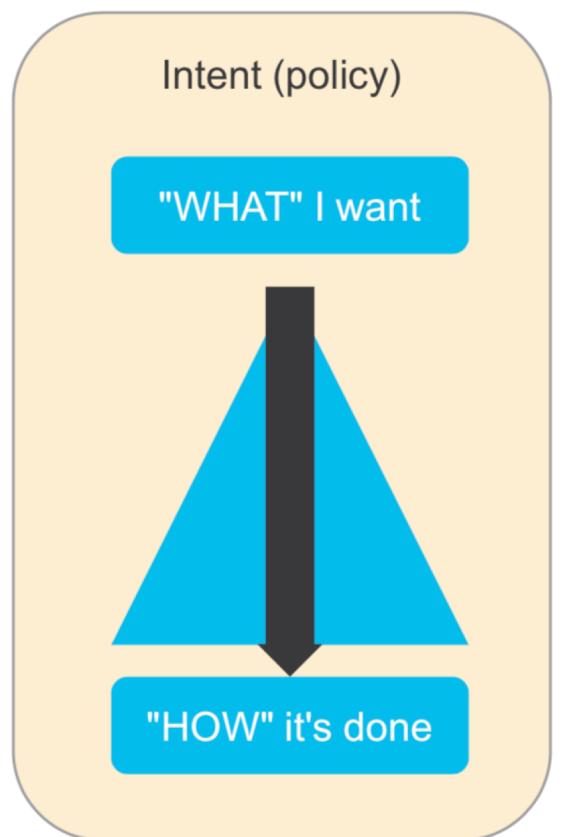
- 1) Try the device API
- 2) DNAC API

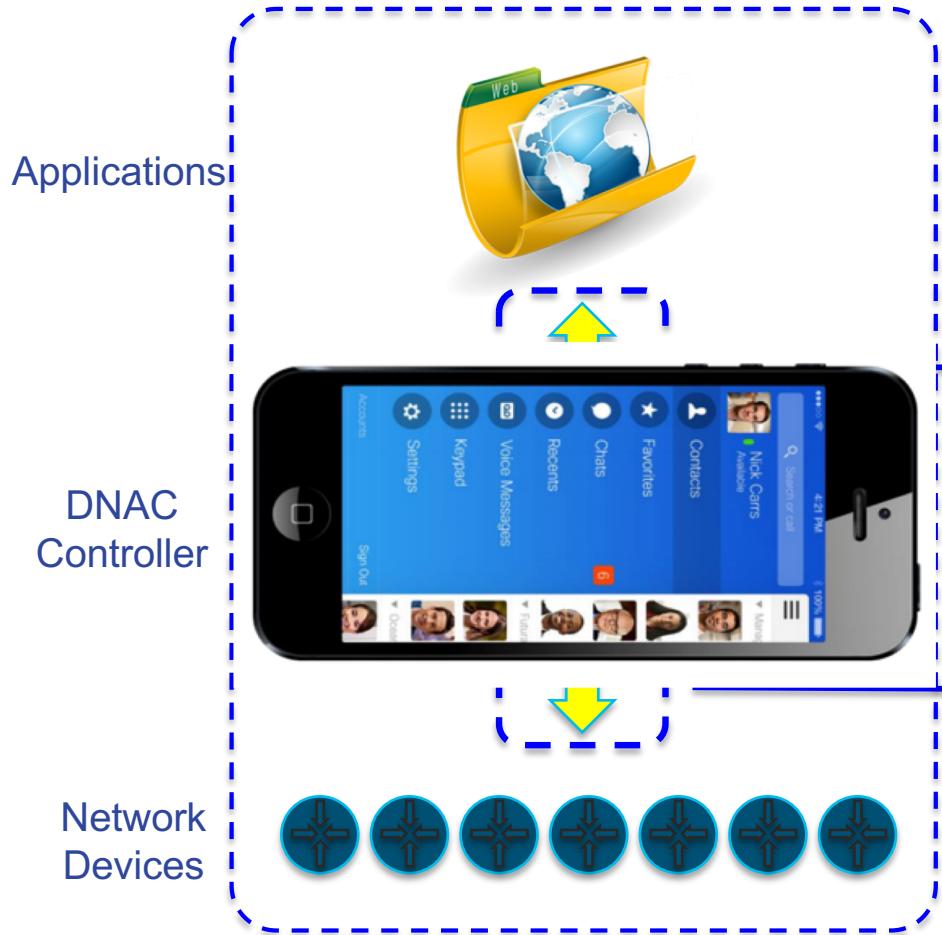
Chapter 2 : Network Orchestration

The Network. Intuitive.



DNA Center – Intent Based Applications



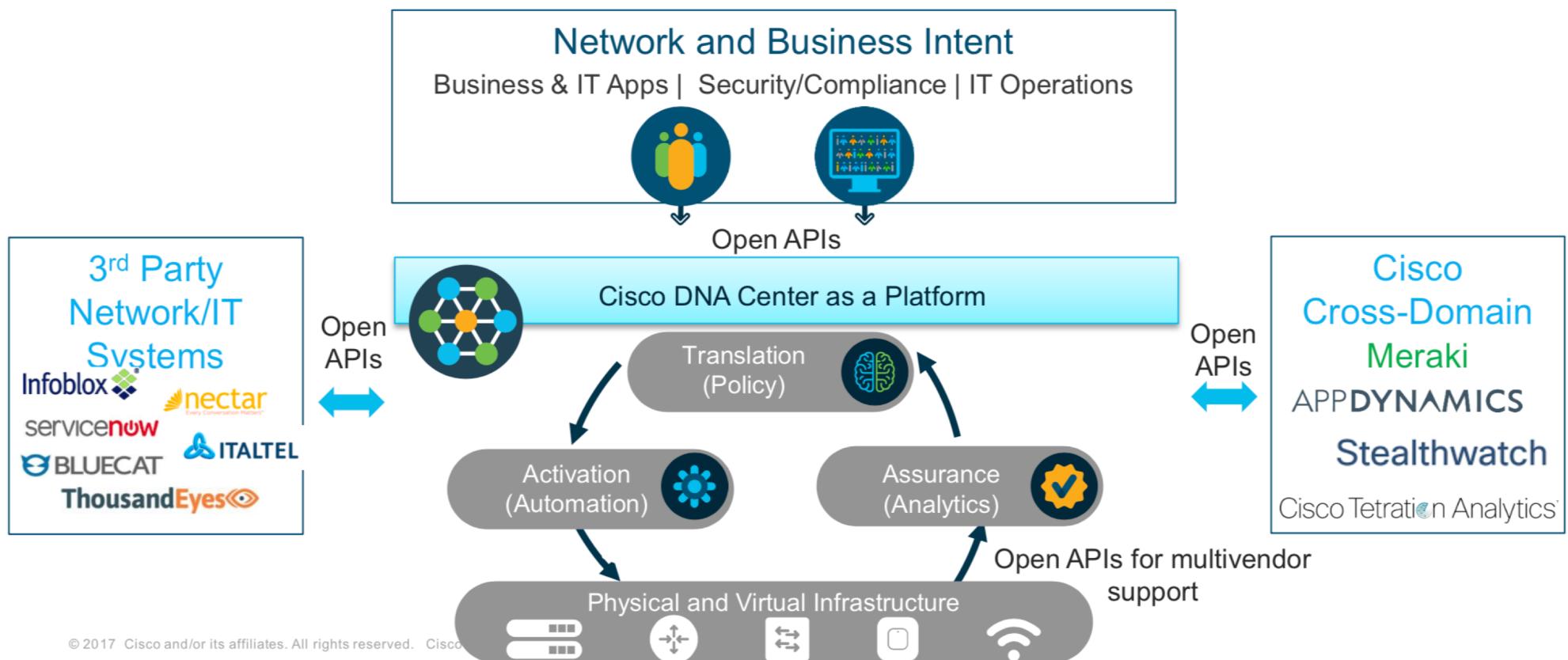


The Smartphone is the Platform for Intelligent Application Control

Allow Protocol/API choice while maintaining stack integrity

DNA Center as a Platform

Bridging the Gap Between Business and IT



Extra Credits

- 1) Try the Ansible lab Section 2d on the Lab Guide ...
- 2) Try the DNA Center Postman

Continue your journey with Devnet

<https://developer.cisco.com/startnow/>

