



i i i i i i i i

You make **possible**

A decorative graphic of vertical bars in various colors (blue, green, orange, red) is positioned on both the left and right sides of the text. The text itself consists of the word "You make" followed by the word "possible" in a large, bold, blue font. The letter "i" in "possible" is repeated nine times, each in a different color: blue, green, blue, orange, red, orange, blue, green, blue.



The CCIE in an SDN World

Jeff McLaughlin, CCIE #14023
Enterprise Technical Marketing
BRKCRT-3075

Cisco live!
June 9-13, 2019 • San Diego, CA

#CLUS



Cisco Webex Teams

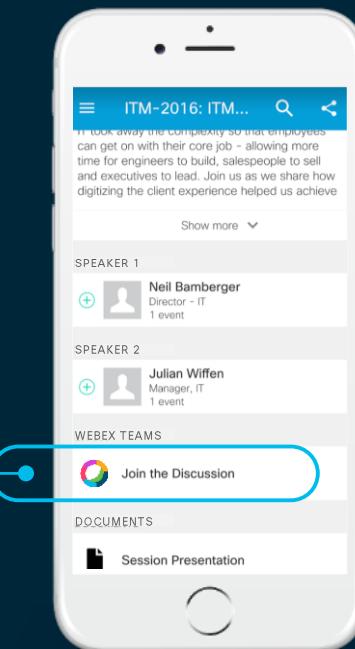
Questions?

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

How

- 1 Find this session in the Cisco Live 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

Webex Teams will be moderated by the speaker until June 16, 2019.



Complete your online session evaluation



- 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 water bottle.
- All surveys can be taken in the Cisco Live Mobile App or by logging in to the Session Catalog on ciscolive.cisco.com/us.

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

Your Host

Director of Technical Marketing, Enterprise

- We guide Cisco engineering on what to build
- We explain to people what engineering builds
- Focused on programmability and automation

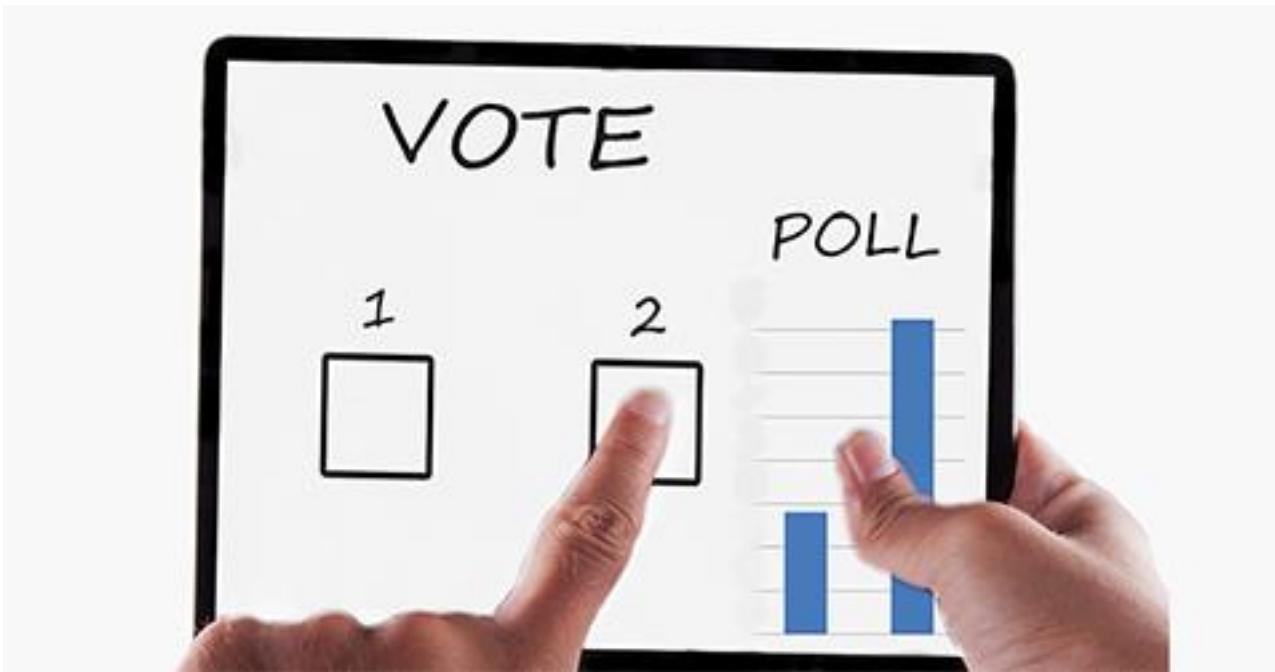


Past Lives

- Director of IT network architecture for Juniper
- Routing Protocols engineer in High-Touch TAC
- Gold partner SE and post-sales deployment engineer

Certification Program

- CCIE in R/S and Security; JNCIE in Service Provider (expired)
- Subject Matter Expert for CCIE program
- NOT an official spokesman for the certification program





ccie still worth it



[Is the CCIE Still Worth It? - CBT Nuggets](#)

<https://www.cbtnuggets.com> › Blog home › Everything CBTN ▾

Aug 24, 2018 - Is the CCIE Still Worth It? This question about whether the CCIE is **worth it** has been swirling for years with no clear answer. CCIE commands a high salary, but so do many certs. ... Whether CCIE is **worth it** or not comes down to whether it's **worth it** to you.

[What is your opinion in CCIE vs SDN? Is CCIE still worth it to ...](#)

[https://www.quora.com/What-is-your-opinion-in-CCIE-vs-SDN-Is-CCIE-still-worth-it... ▾](https://www.quora.com/What-is-your-opinion-in-CCIE-vs-SDN-Is-CCIE-still-worth-it...)

Nov 29, 2016 - The answer is no CCIE is no longer **worth it** as it used to be. SDN with Cyber Security skill is the way to go. However CCNP is **still worth it**.

[Quitting My CCIE Status. Time to Move On. - EtherealMind](#)

<https://etherealmind.com> › Cisco › CCIE ▾

Mar 14, 2018 - 14th March 2018 By Greg Ferro Filed Under: **CCIE**. Its been 17 years since ... The foundations of the last decade are **still** there and educational material is widely available. Its very ... My time is **worth** more than the test delivers.

Why the concern?

Networking Industry



**CCIE/CLI
(Old World)**

**SDN/APIs
(New World)**

This...



...replaced by this:



This...



...replaced by this:



Network engineers...



...replaced by this?

Cisco DNA Center DESIGN POLICY PROVISION ASSURANCE PLATFORM

Devices Fabric

SD-Access Fabrics and Transits

Choose a Fabric or Transit below to manage, or add a new item by clicking "Add Fabric or Transit".

Fabrics i

i Default_LAN_Fabric LAN	i SanJose_Fabric LAN
--	--



The Headlines...

Cisco live!

#CLUS

BRKCRT-3075

© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Public

13

API Is The New CLI For Cisco Systems



Moor Insights and Strategy Contributor [\(i\)](#)

Straight talk from Moor Insights & Strategy tech industry analysts

POST WRITTEN BY

Will Townsend

[Will Townsend](#) is a Moor Insights & Strategy senior analyst covering wireless telecommunications and enterprise networking





Does SDN Mean IT Will Be Able To Get Rid of Network People?

Network pros need SDN training, not CCIE status

Network pros will need to look further than traditional vendor certifications if they want to build and manage SDN and programmable infrastructure.

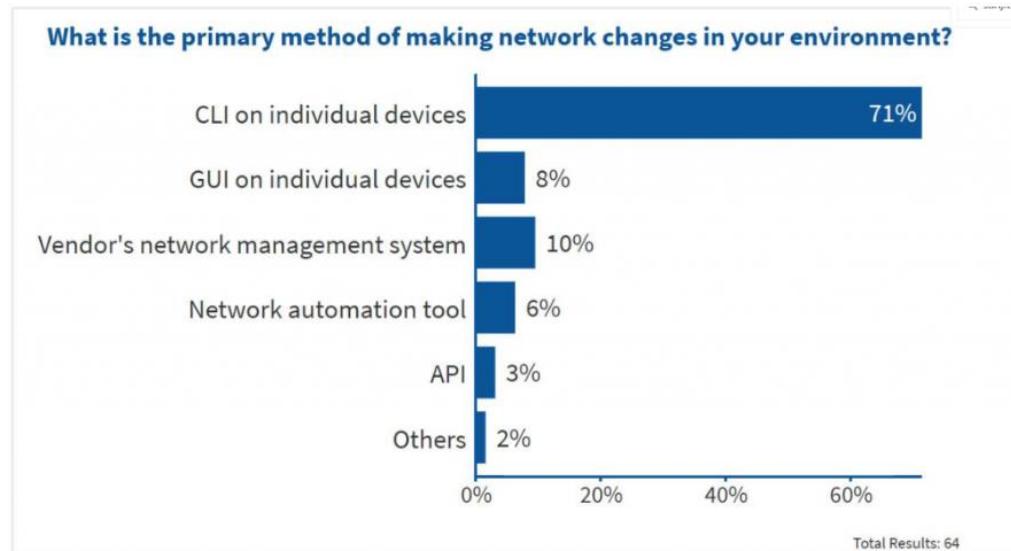
_____ , who is part of the Silicon Valley SDN research vanguard, believes CCIE status is already worthless -- he doesn't even include his own CCIE status on his presentations anymore.

"There are a bunch of certified individuals who don't know how to do crap," said _____ .
"What I want to know when I am hiring is: Can you write code? ... Are you smart?"

07 Oct 2014

Checking in on the Death of the CLI

by Andrew Lerner | January 4, 2018 | Submit a Comment



Agenda

- The CCIE: A brief history, and how far we've come
- Are APIs the new CLI?
- What is SDN and why do we need it?
- SD-Access: SDN in the Campus
- The CCIE in an SDN World
- The future of certifications and what you should do

CCIE History



You make networking **possible**



1993

CCIE certification first Introduced

Topics from the original blueprint...



AppleTalk

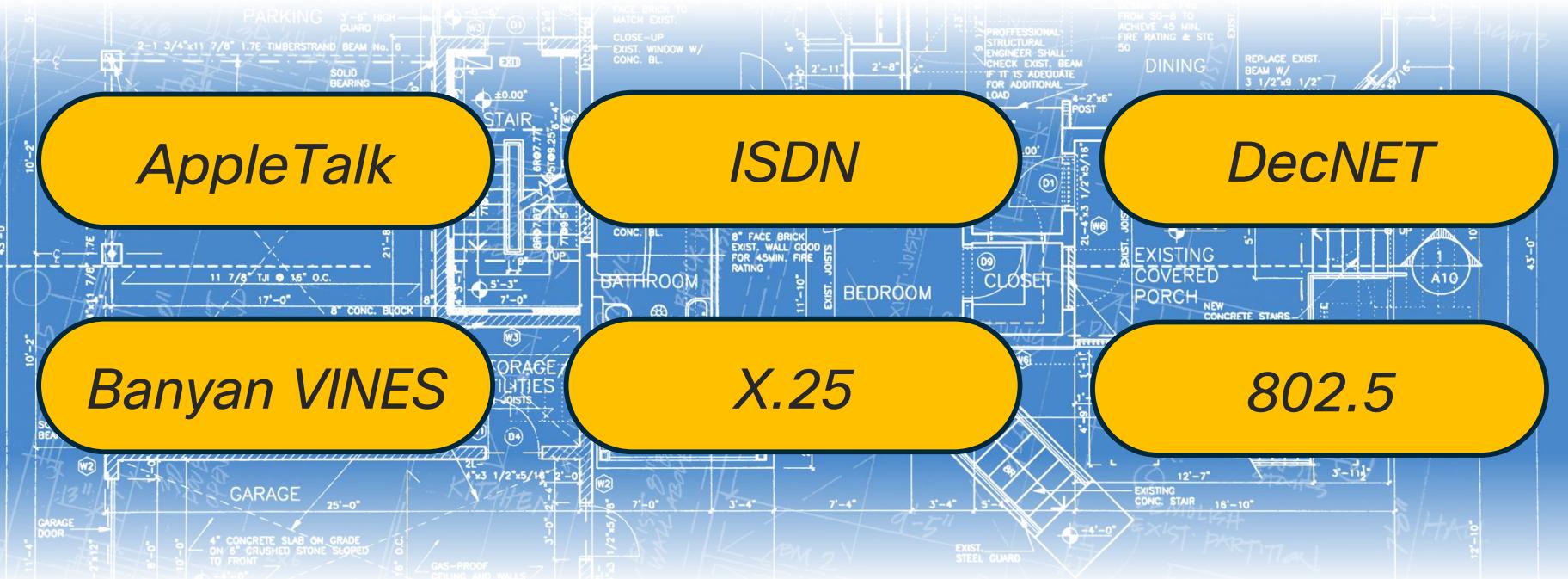
ISDN

DecNET

Banyan VINES

X.25

802.5



The Original CCIEs



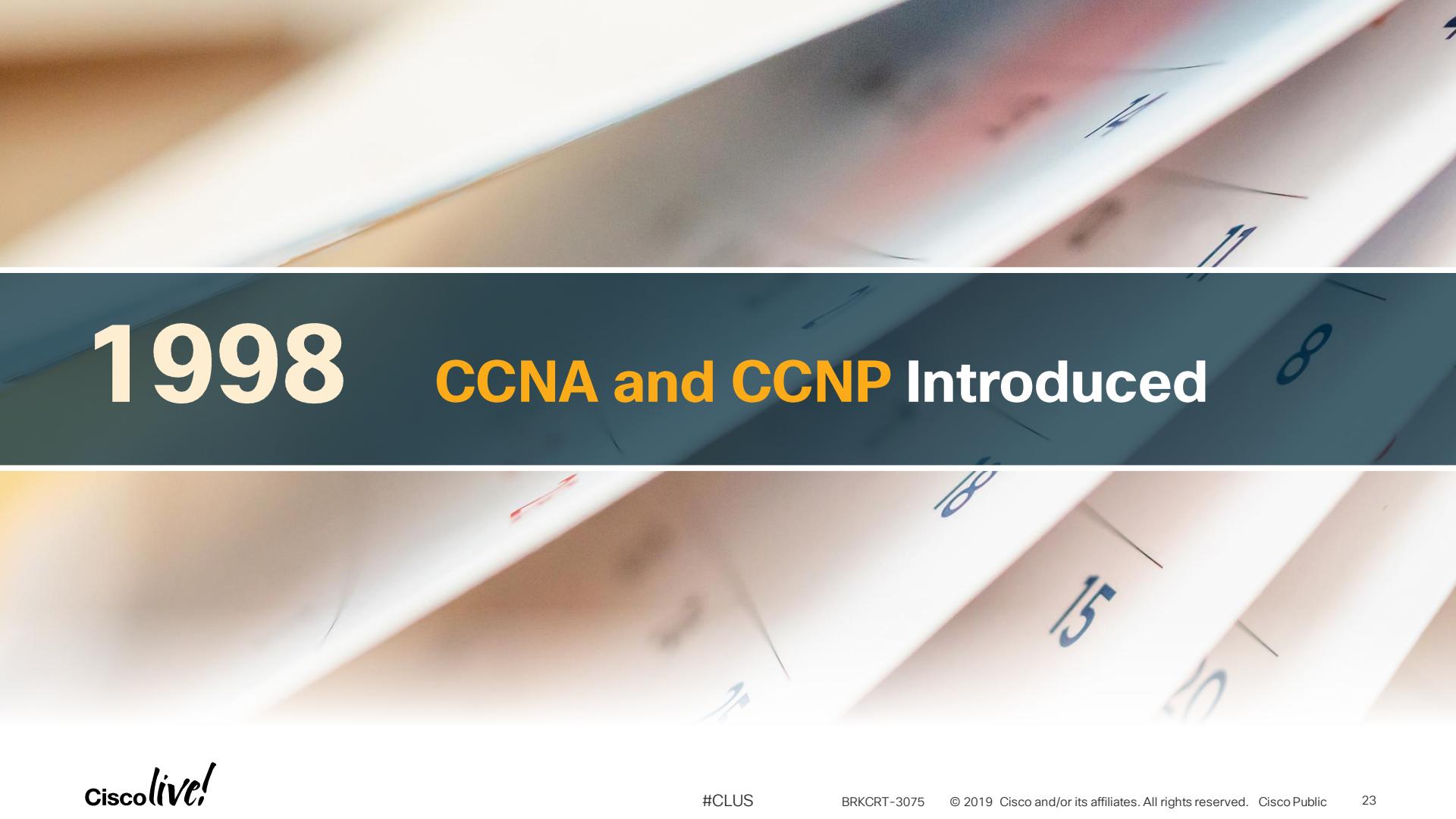
1024



1025

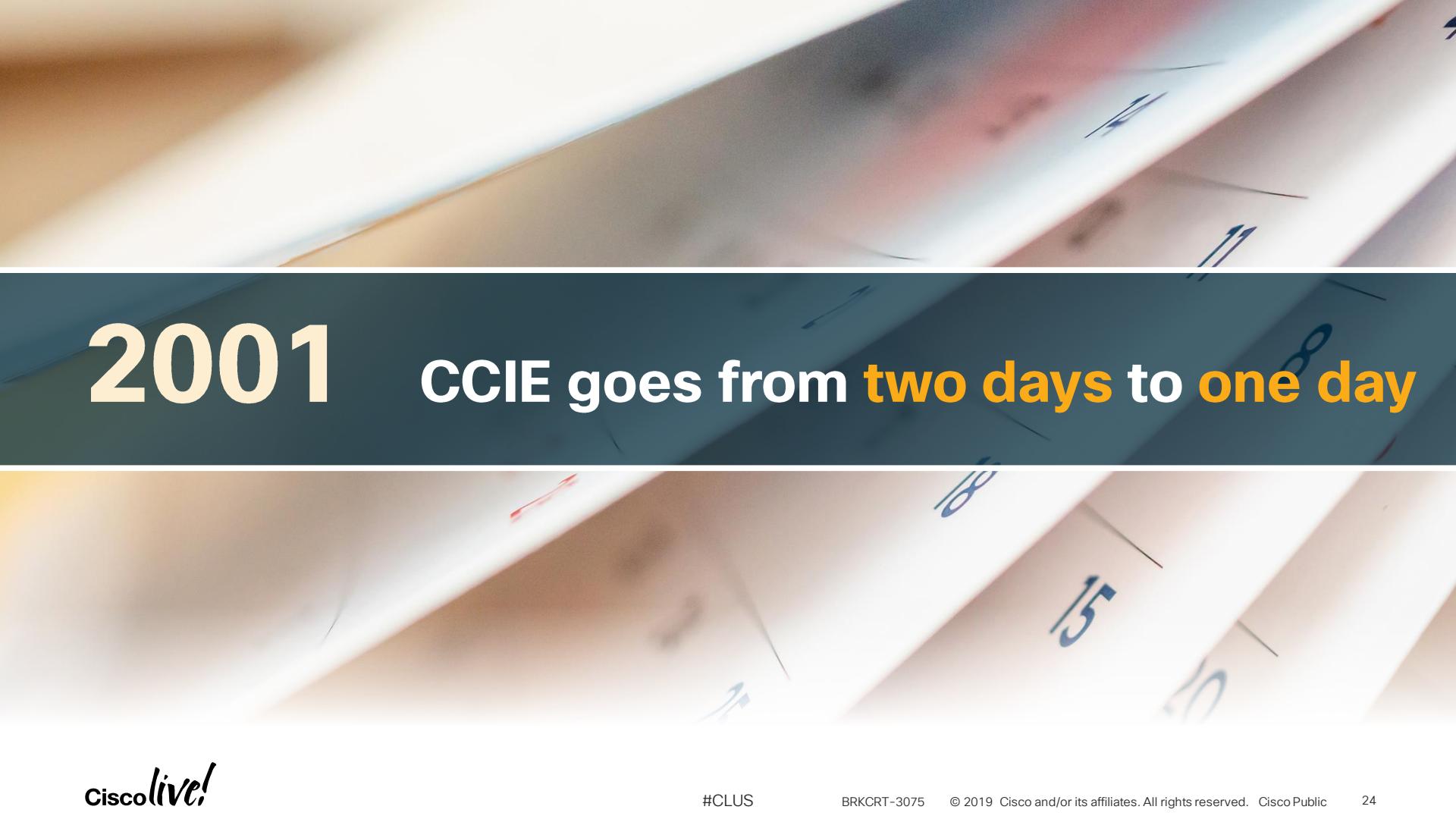


1026



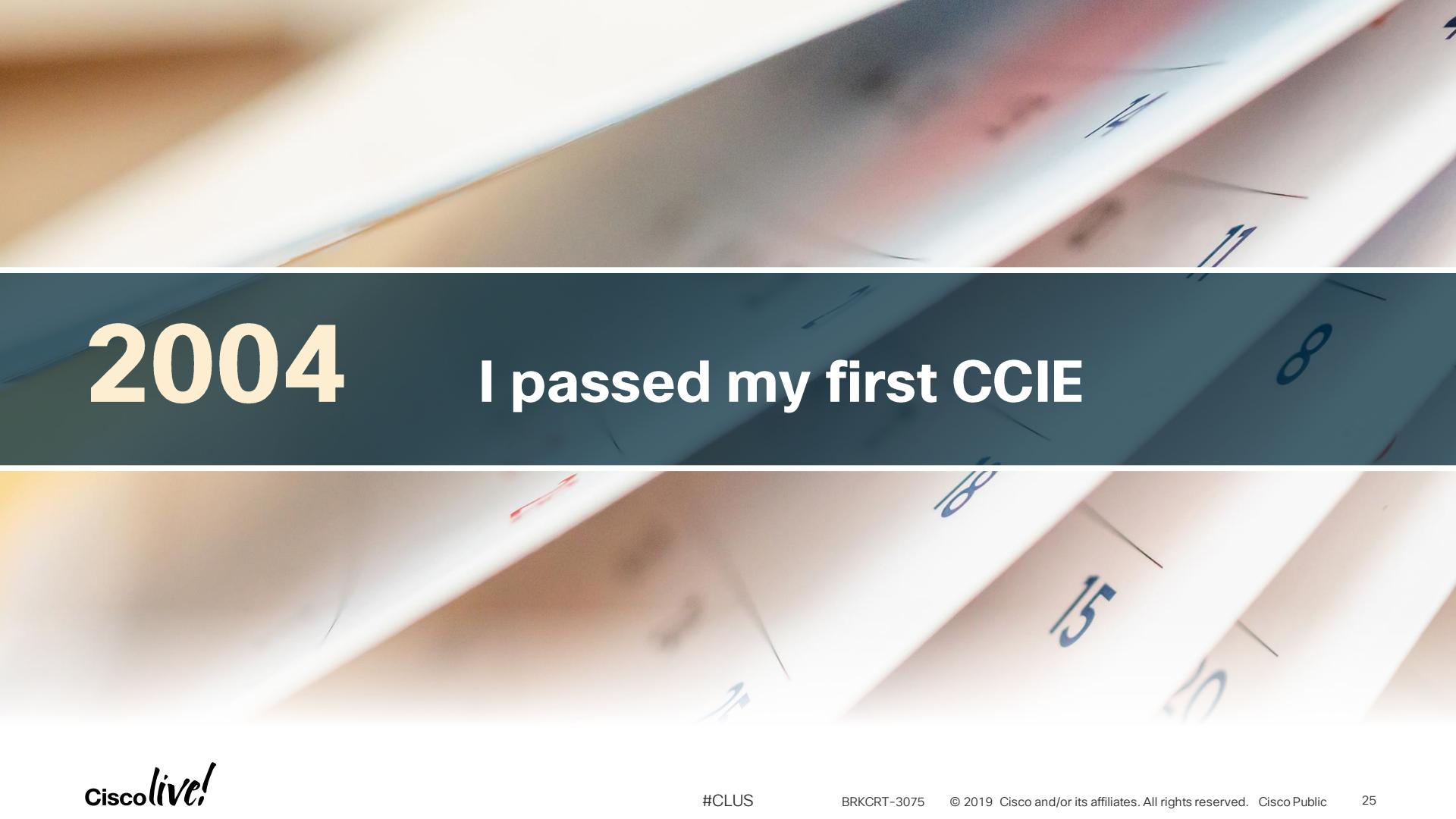
1998

CCNA and CCNP Introduced



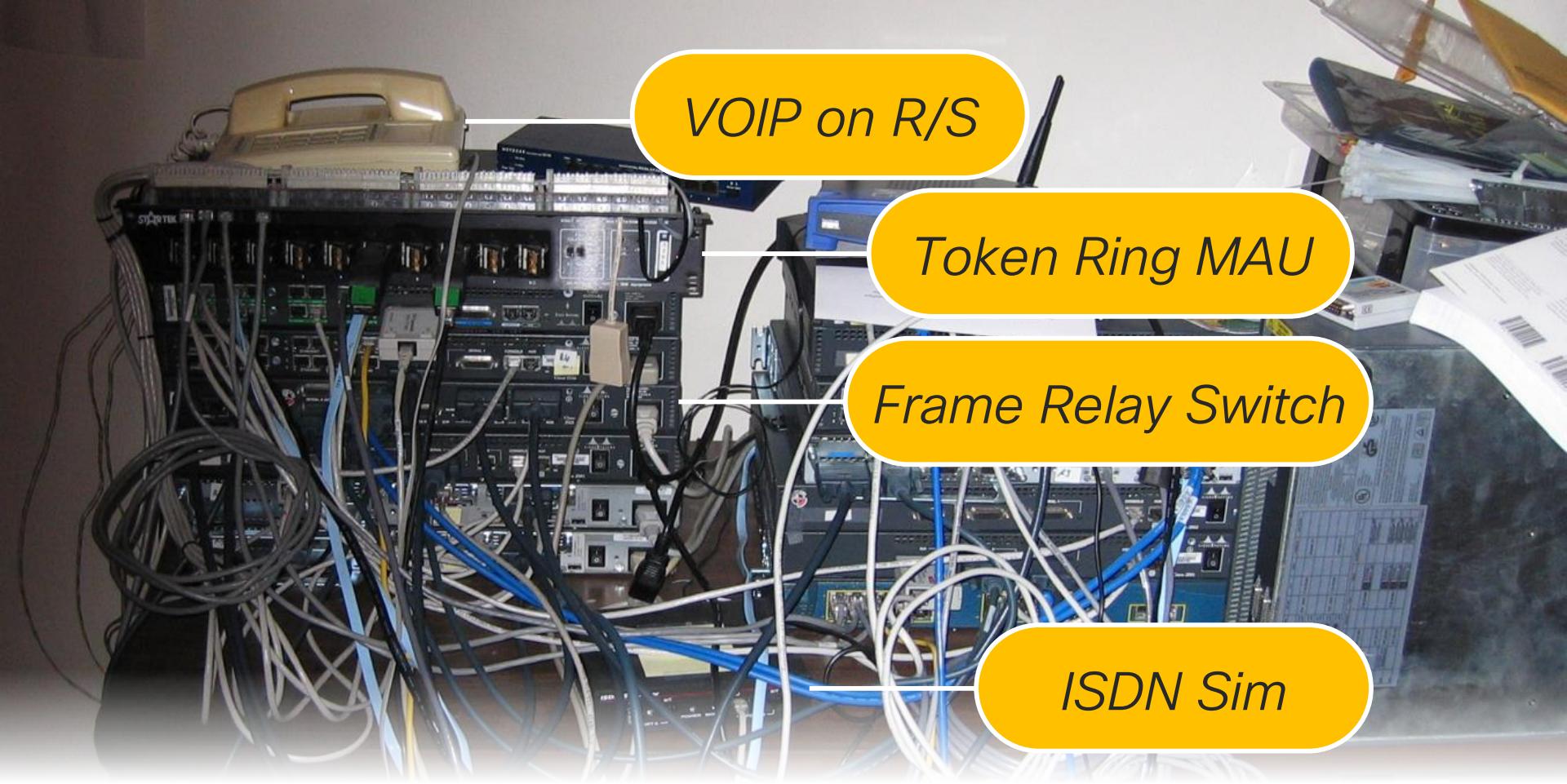
2001

CCIE goes from **two days** to **one day**



2004

I passed my first CCIE

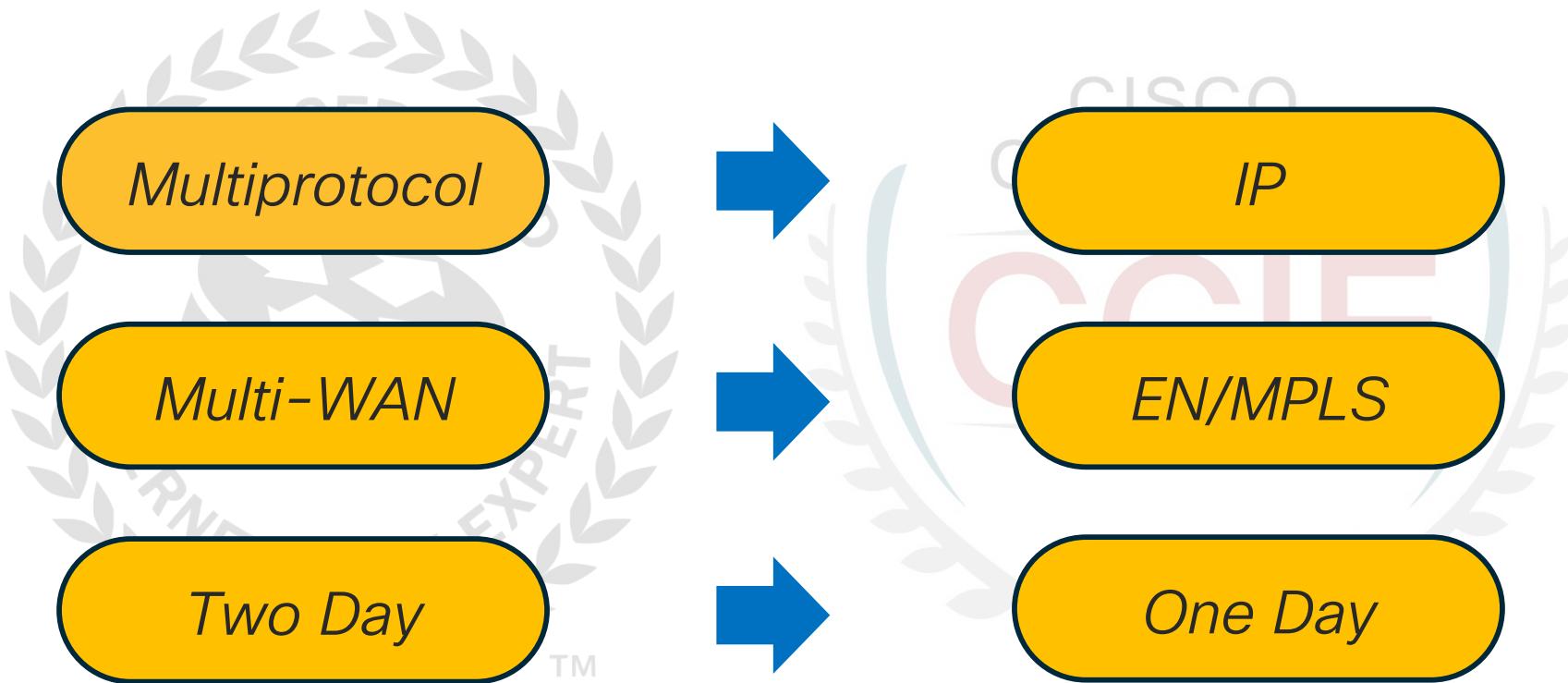




2019

SDN comes to CCIE

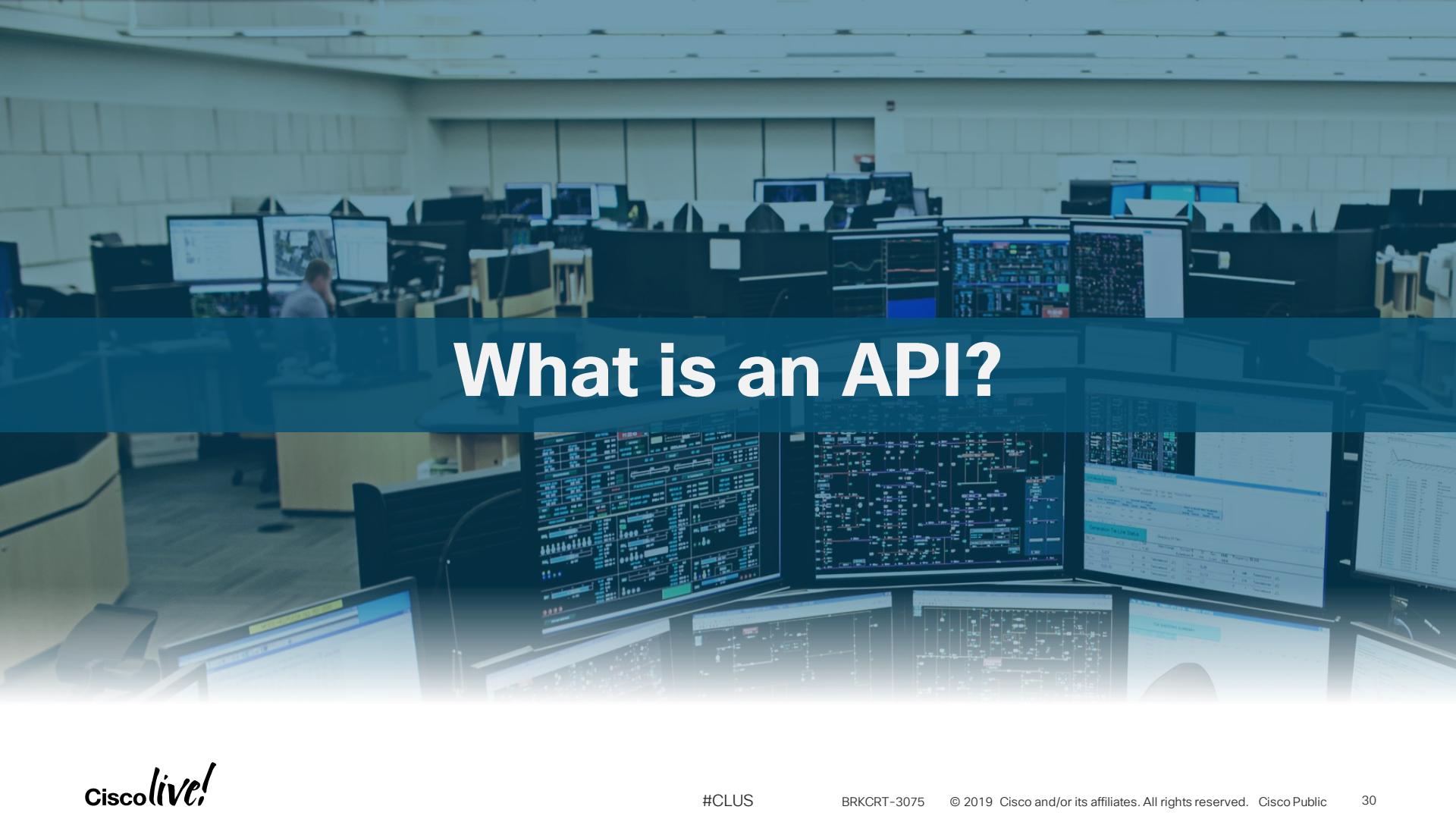
CCIE Evolution...or Adaptation?



CCIE Threats #1: Are APIs the new CLI?

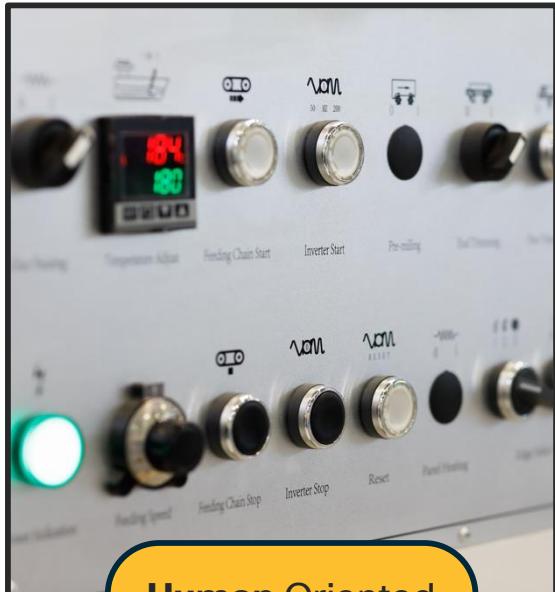


You make networking **possible**

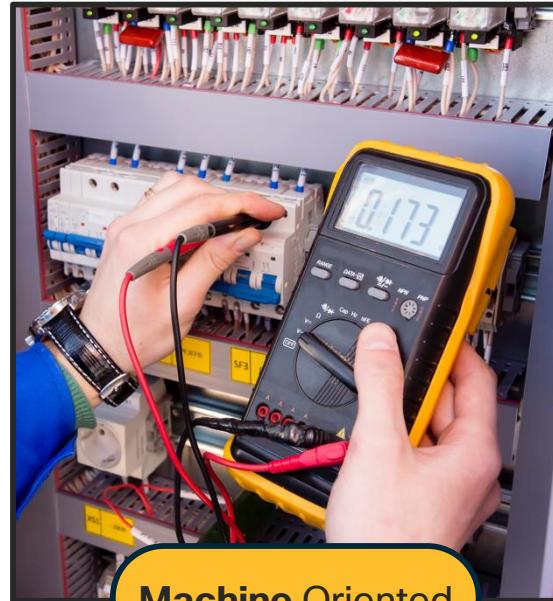


What is an API?

Two kinds of interfaces



Human Oriented



Machine Oriented



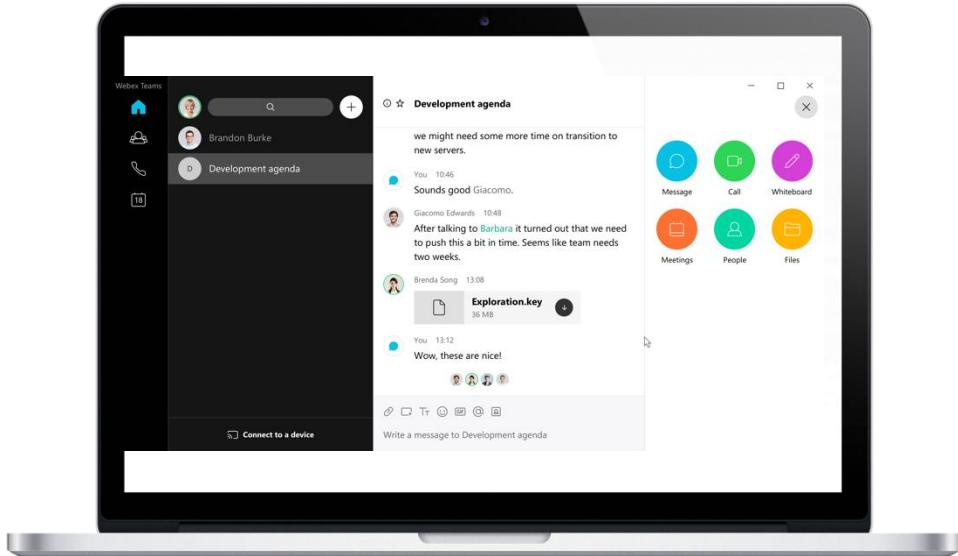
Machines using human-oriented interfaces is **inefficient!**

API = Application Programming Interface

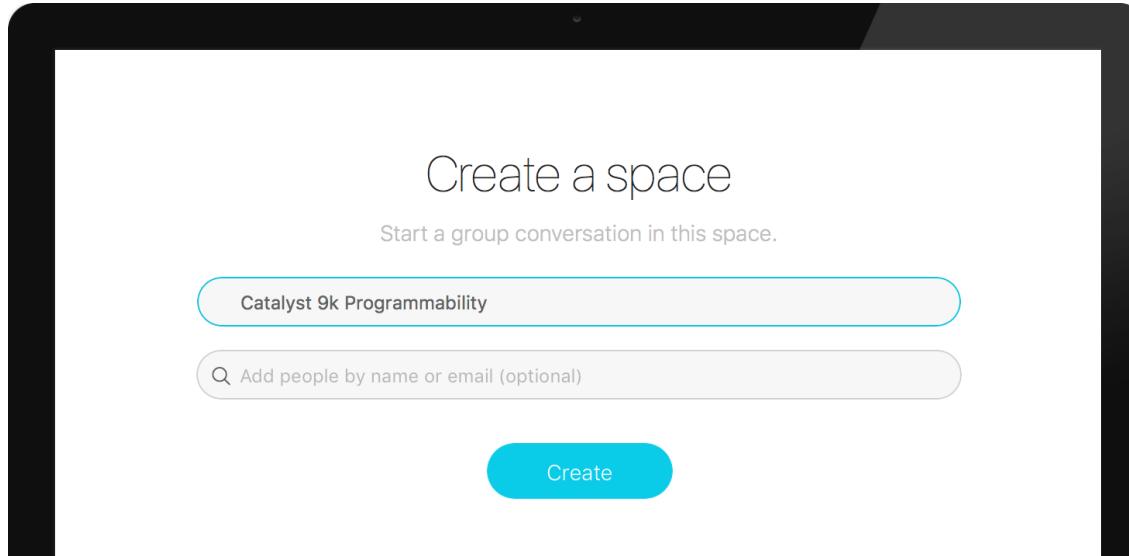


Designed for programmers to access *through code*

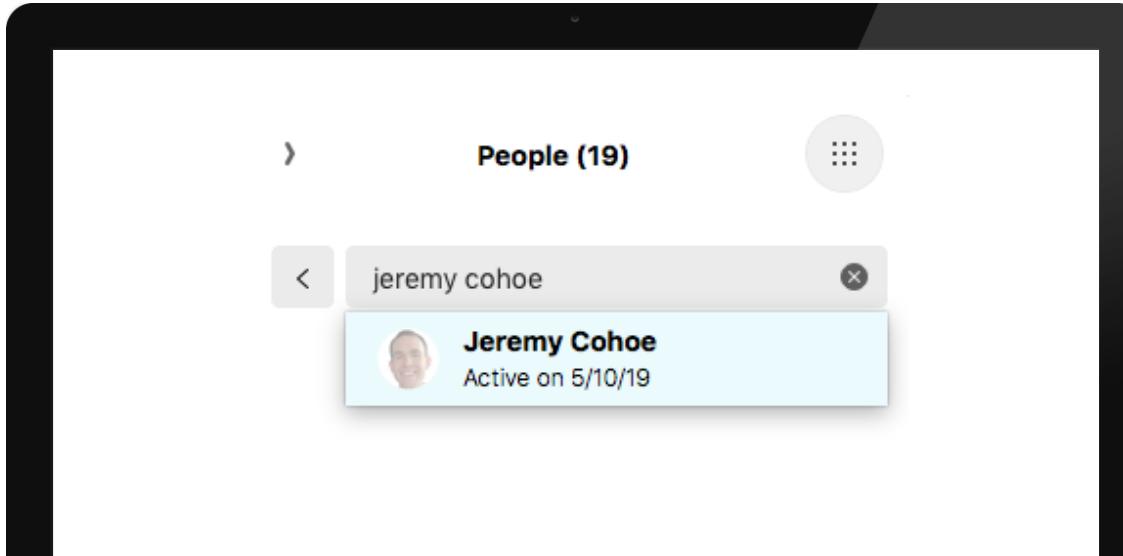
For example, a REST API...



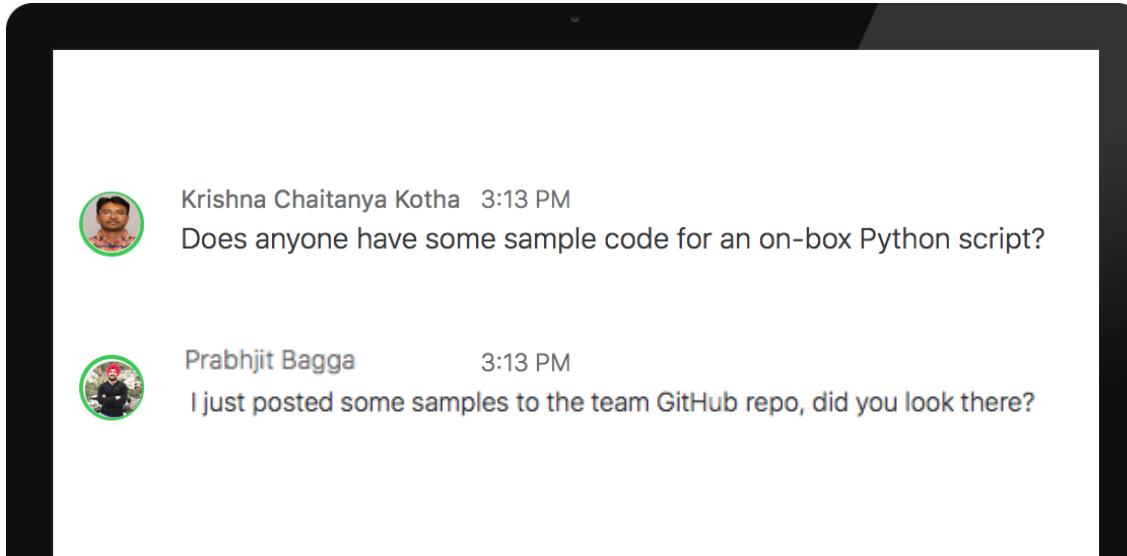
What can you do with the Webex Teams GUI?



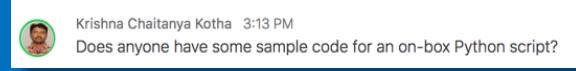
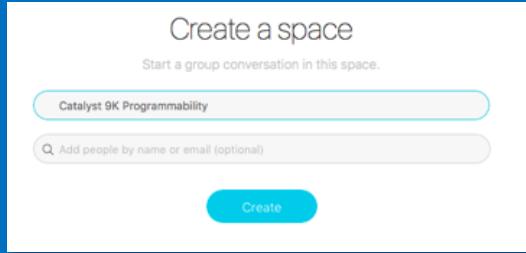
Create a room/space



Add a user to a room



Read a message



APIs: script instead of click

```
9 def create_room(room_name, token):
10    requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
11
12    headers = {'Authorization': 'Bearer ' + token,
13               'Content-Type': 'application/json'}
14    body = json.dumps({'title': room_name})
15
16    resp = requests.post('https://api.ciscospark.com/v1/rooms',
17                         verify=False, headers=headers, data=body)
18
19    print resp
20
21
```

```
def add_user_to_room(room_id, user, token):
    """
    Adds a user to a room.
    """

    requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

    headers = {'Authorization': 'Bearer ' + token,
               'Content-Type': 'application/json'}
    body = json.dumps({'room_id': room_id, 'personId': user})

    resp = requests.post('https://api.ciscospark.com/v1/memberships',
                         verify=False, headers=headers, data=body)

    print resp
```

```
54 def list_messages(room_id, token):
55     requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
56
57     headers = {'Authorization': 'Bearer ' + token,
58                'Content-Type': 'application/json'}
59
60     resp = requests.get('https://api.ciscospark.com/v1/messages?roomId={}'.format(room_id),
61                         verify=False, headers=headers)
62
63     return resp.text
```

APIs: script instead of click

APIs expose the following to coders:



Actions (Create a space, read a message)



Data (Name of the space, text of the message)

REST API calls use **HTTP methods** like GET, PUT and POST.

Create a Room



Creates a room. The authenticated user is automatically added as a member of the room. See the Memberships API to learn how to add more people to the room.

POST <https://api.ciscospark.com/v1/rooms>

```
resp = requests.post('https://api.ciscospark.com/v1/rooms',
verify=False,headers=headers,data=body)
```

Two types of API



Northbound: Exposed from controller to scripts

Controller



Southbound: Exposed from devices

Network Device

Two types of API



Northbound: Typically REST

Controller



Southbound: NETCONF, RESTCONF, SNMP

Network Device

Let's look at **Southbound APIs
(And save controllers for later...)**

```

// interface * / ip ospf authentication
container authentication {
    description
        "Enable authentication";
    presence "Enable authentication";
    container key-chain {
        description
            "Use a key-chain for cryptographic authentication k
        leaf name {
}
leaf message-digest {
    description
        "Use message-digest authentication";
    type empty;
}
leaf null {
    description
        "Use no authentication";
    type empty;
}
}
}

```

```

grouping config-interface-ip-igmp-grouping {
    container igmp {
        description
            "IGMP interface commands";
    container static-group {
        description
            "IGMP static multicast group";
        list groups {
            key "name";
            leaf name {
                type inet:ipv4-address;
                type enumeration {
                    enum "*";
                }
            }
        }
    }
    leaf-list source {
        type union {
            type inet:ipv4-address;
            type enumeration {
                enum "ssm-map";
            }
        }
    }
}

```

In IOS XE 16.3, Cisco introduced support for YANG models



Machine Oriented



```

interface AppGigabitEthernet1/0/1
!
interface Vlan1
  no ip address
  shutdown
!
router ospf 1 vrf Mgmt-vrf
  network 172.26.244.0 0.0.0.255 area 0
!
router bgp 100
  bgp log-neighbor-changes
  neighbor 38.38.38.38 remote-as 200
  neighbor 38.38.38.38 update-source Lo
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet1/0/1
ip route vrf Mgmt-vrf 0.0.0.0 0.0.0.0
ip route vrf Mgmt-vrf 38.38.38.38 255.
!
ip scp
  
```

CLI

```

list interface {
  key "name";
  unique "type location";

  leaf name {
    type string;
    reference
      "RFC 2863: The Interfaces Group MIB - ifName";
  }

  leaf description {
    type string;
  }

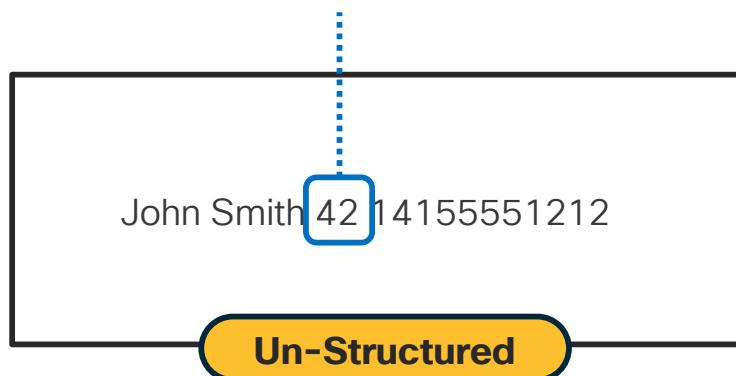
  ...
  container statistics {
    config false;
    leaf discontinuity-time {
      type yang:date-and-time;
    }
  }

  leaf in-octets {
    type yang:counter64;
    reference
      "RFC 2863: The Interfaces Group MIB - inOctets";
  }
}
  
```

YANG

Structured vs. Unstructured Data

What is this value?



Keys **Values**

Name:	John Smith
Age:	42
Phone:	+1-415-555-1212

A rectangular box containing a table with three rows. The first row has "Name:" and "John Smith". The second row has "Age:" and "42". The third row has "Phone:" and "+1-415-555-1212". Above the box is a diagram showing two vertical dashed blue lines connecting the question "What is this value?" to the "42" in the unstructured data and the "Age:" key in the structured data. Below the box is a yellow button labeled "Structured".

Hierarchical Structured Data (XML-like)

The diagram illustrates hierarchical structured data using curly braces {} to group elements. On the left, two labels identify the groups: "First User" and "Second User". To the right of each label is a brace that spans multiple lines of XML-like code. The first user's data is grouped under a brace, and the second user's data is grouped under another brace.

```
<user1>
  <name>John Smith</name>
  <age>42</age>
  <phone>+1-415-555-1212</phone>
</user1>

<user2>
  <name>Sarah Kim</name>
  <age>27</age>
  <phone>+1-718-555-1212</phone>
</user2>
```

ΙΠΠΟΝΙΚΩΔΕΤΩΚΑΛΛΙΟΥ
ΠΑΤΡΙΚΑΙΔΟΣΛΗΝΕΧΟΝΤΙ
ΜΕΓΑΛΗΝΚΑΙΔΥΝΑΜΙΝΑΠΟ
ΠΛΟΥΤΟΥΚΑΙΓΕΝΟΥΣΕΝΕ
ΤΡΙΤΕΚΟΝΔΑΥΛΟΝΟΥΧΥΠ
ΟΡΓΗΣΗΔΙΑΦΟΡΑΣΤΙΝΟΣ
ΠΡΟΔΗΘΕΙΣΑΛΛΕΠΙΓΕΛΛ
ΤΙΣΥΝΘΕΜΕΝΟΣΠΡΟΣΤΟΥΣ
ΕΤΑΙΡΟΥΣΠΕΡΙΒΟΝΤΟΥΔΕ
ΤΗΣΑΣΕΛΓΕΙΑΣΕΝΤΗΠΟΛΕΙ
ΓΕΝΟΜΕΝΗΣΚΑΙΣΥΝΑΓΑ
ΝΑΚΤΟΥΝΤΩΝΩΣΠΕΡΕΙΚ
ΟΣΑΠΑΝΤΩΝΑΜΗΜΕΡΑ
ΠΑΡΗΝΟΔΛΚΙΒΙΔΗΣΕΠΗ
ΤΗΝΟΙΚΙΑΝΤΟΥΙΠΠΟΝΙΚΟΥ
ΚΑΙΤΗΝΟΥΡΑΝΚΟΦΑΣΕΙС
ΗΛΘΕΠΡΟΣΑΥΤΟΝΚΑΙΘΕΙΣ
ΤΟΙΜΑΤΩΝΠΑΡΕΔΙΔΟΥΤΩ
ΣΩΜΑΜΑΣΤΙΓΟΥΝΚΑΙΚΟ
ΛΑΖΕΙΝΚΕΛΑΓΥΩΝΔΑΕ



Ancient unstructured data

COULDYOUREADITIFWEWROTELIKETHISWIT
HNOPUNCTUATIONITISHARDTOFIGUREOUT
WHEREONEWORDORSENTENCEBEGINSORE
NDSANDITISNOTEASYTOSEPARATEOUTALL
THEELEMENTSOFTHETEXTTHANKGOODNES
SSOMEONECAMEUPWITHABETTERWAY

```
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
```

Structured

Or

Un-Structured

?

```
<ipv4>
  <addresses>
    <address>
      <config>
        <ip>172.26.194.212</ip>
        <prefix-length>24</prefix-length>
      </config>
    </address>
  </addresses>
</ipv4>
```

Configuration

```
<tcam-detail>
  <asic-no>1</asic-no>
  <hash-entries-max>
    24576
  </hash-entries-max>
  <tcam-entries-max>8192</tcam-entries-max>
  <hash-entries-used>2</hash-entries-used>
  <tcam-entries-used>19</tcam-entries-used>
</tcam-detail>
```

Operational

Needed: Standardized way to represent device data

```
<neighbor>
  <id>38.38.38.38</id>
  <remote-as>200</remote-as>
  <update-source>
    <Loopback>0</Loopback>
  </update-source>
</neighbor>
```

XML

```
"neighbor": [
  {
    "id": "38.38.38.38",
    "remote-as": 200,
    "update-source": {
      "Loopback": 0
    }
}
```

JSON

NETCONF/RESTCONF

RESTCONF

XML/JSON: Formatting but not content



Sends

```
<interface>Gigabit 1/0</interface>  
<ifaddr>10.0.0.1/24</ifaddr>
```



Error!

Expecting:

```
<interface>  
  <name>Gigabit 1/0</name>  
  <address>10.0.0.1/24</address>  
</interface>
```

YANG Data Models

```
list neighbor {  
    leaf id {  
        type inet:ip-address;  
    }  
}
```



1.1.1.1



```
<neighbor>  
  <id>1.1.1.1</id>  
</neighbor>
```



YANG Data Models

```
list neighbor {  
    leaf id {  
        type inet:ip-address;  
    }  
}
```



```
1.1.1.1
```



```
"neighbor": [  
    {  
        "id": "38.38.38.38"  
    }  
]
```



YANG Configuration Model Example*

```
container ip {
    list vrf {
        description
            "Configure an IP VPN Routing/Forwarding
             instance";

        leaf name {
            type string;
        }

        leaf rd {
            description
                "Specify Route Distinguisher";
            type rd-type;
        }
    }
} * Note: YANG model simplified for clarity
```

YANG

```
<ip>
  <vrf>
    <name>vrf_red</name>
    <rd>65000:1</rd>
  </vrf>
  <vrf>
    <name>vrf_green</name>
    <rd>65000:2</rd>
  </vrf>
</ip>
```

XML

```
ip vrf vrf_red
  rd 65001:1
!
ip vrf vrf_green
  rd 65001:2
!
```

CLI

Where are YANG models?



Models installed on device automatically with IOS XE.



Also can be downloaded from GitHub.

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

The screenshot shows the GitHub repository page for 'YangModels/yang'. The repository has 113 watchers, 211 stars, and 138 forks. It contains 11 issues, 1 pull request, 0 projects, and 0 wiki pages. The code tab is selected. A specific commit by user 'pgohite' titled 'Cisco IOS XE 16.3.2 Release Yang Models' is highlighted, dated Nov 28, 2016. The commit message includes the URL 'https://github.com/YangModels/yang/commit/55bd294'. The commit details show files like MIBS, README.md, cat3k-netconf-capability.xml, check-models.sh, cisco-acl-oper.yang, and cisco-bfd-state.yang, all updated 2 months ago.

File	Description	Last Updated
MIBS	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
README.md	Cisco IOS XE 16.3.2 Release Yang Models	a month ago
cat3k-netconf-capability.xml	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
check-models.sh	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-acl-oper.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago
cisco-bfd-state.yang	Cisco IOS XE 16.3.2 Release Yang Models	2 months ago

Who defines the YANG models?

Vendors

- Only work on specific vendor devices
- Greater feature coverage
- Can be OS-unique (IOS-XE, XR, etc.)



Standards Bodies

- Multi-vendor support
- More limited feature coverage
- Allow vendor-specific extensions



Actually an "industry forum"

Important Point!

Cisco's data models and IETF/OpenConfig data models are just two ways of doing the same thing.

```
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"> ← IETF-defined model
  <interface>
    <name>GigabitEthernet 1/0/24</name>
    <description>Configured by NETCONF!</description>
  </interface>
</interfaces>
```

Both of these do exactly the same thing!

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native"> ← Cisco-defined “native” model
  <interface>
    <GigabitEthernet>
      <name>1/0/24</name>
      <description>Configured by NETCONF!</description>
    </GigabitEthernet>
  </interface>
</native>
```

Important Point!

Cisco's data models and IETF/OpenConfig data models are just two ways of doing the same thing.

```
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"> ← IETF-defined model
  <interface>
    <name>GigabitEthernet 1/0/24</name>
    <description>Configured by NETCONF!</description>
  </interface>
</interfaces>
```

```
switch# show run interface g1/0/24
interface GigabitEthernet 1/0/24
  description Configured by NETCONF!
```

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native"> ← Cisco-defined “native” model
  <interface>
    <GigabitEthernet>
      <name>1/0/24</name>
      <description>Configured by NETCONF!</description>
    </GigabitEthernet>
  </interface>
</native>
```

NETCONF



RESTCONF



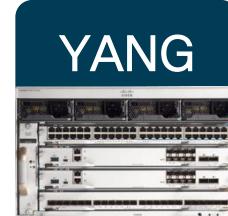
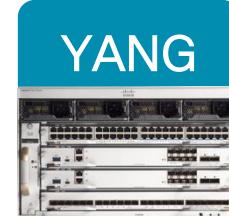
GNMI



Transport



Encoding



So will this replace CLI?

```
<router>
  <ospf>
    <id>1</id>
    <redistribute>
      <bgp>
        <as-number>100</as-number>
        <redist-options>
          <metric>10</metric>
        </redist-options>
      </bgp>
    </redistribute>
  <ospf>
</router>
```

```
router ospf 1
 redistribute bgp 100 metric 10
```

Note Well:

```
<router>
  <ospf>
    <id>1</id>
    <redistribute>
      <bgp>
        <as-number>100</as-number>
        <redist-options>
          <metric>10</metric>
        </redist-options>
      </bgp>
    </redistribute>
  <ospf>
</router>
```

```
router ospf 1
 redistribute bgp 100 metric 10
```

Note Well:
Similar syntax

```
<router>
  <ospf>
    <id>1</id>
    <redistribute>
      <bgp>
        <as-number>100</as-number>
        <redist-options>
          <metric>10</metric>
        </redist-options>
      </bgp>
    </redistribute>
  <ospf>
</router>
```

```
router ospf 1
 redistribute bgp 100 metric 10
```

Note Well:
Similar syntax
Same parameters

Assume the most dramatic scenario:

In 2019, NETCONF/RESTCONF **completely replaces** CLI...

Everything you learned for your CCIE would still be totally relevant.

How would you build a script if you didn't know what you were automating?

Assume the most realistic scenario:

In 2019, CLI continues to be the **primary means** of configuration/operation for most networks...

Everything you learned for your CCIE would still be totally relevant.

How would you build a script if you didn't know what you were automating?

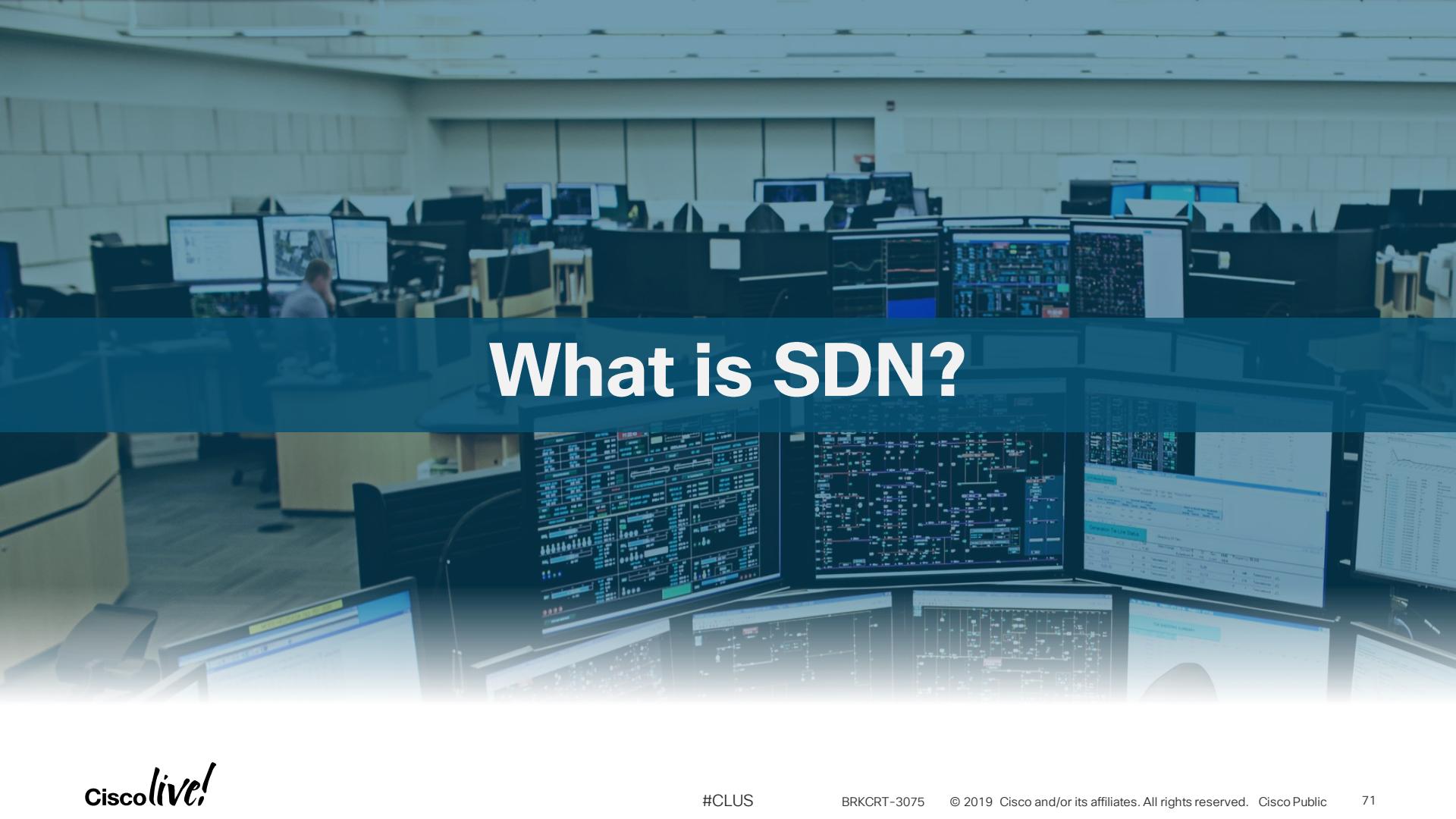
So what should I do?

- ↳ **Study *basic* scripting**
- ↳ **Study device APIs (YANG/NC/RC)**
- ↳ **Add this tool to your toolbox**

Is SDN the new CLI?



You make networking **possible**



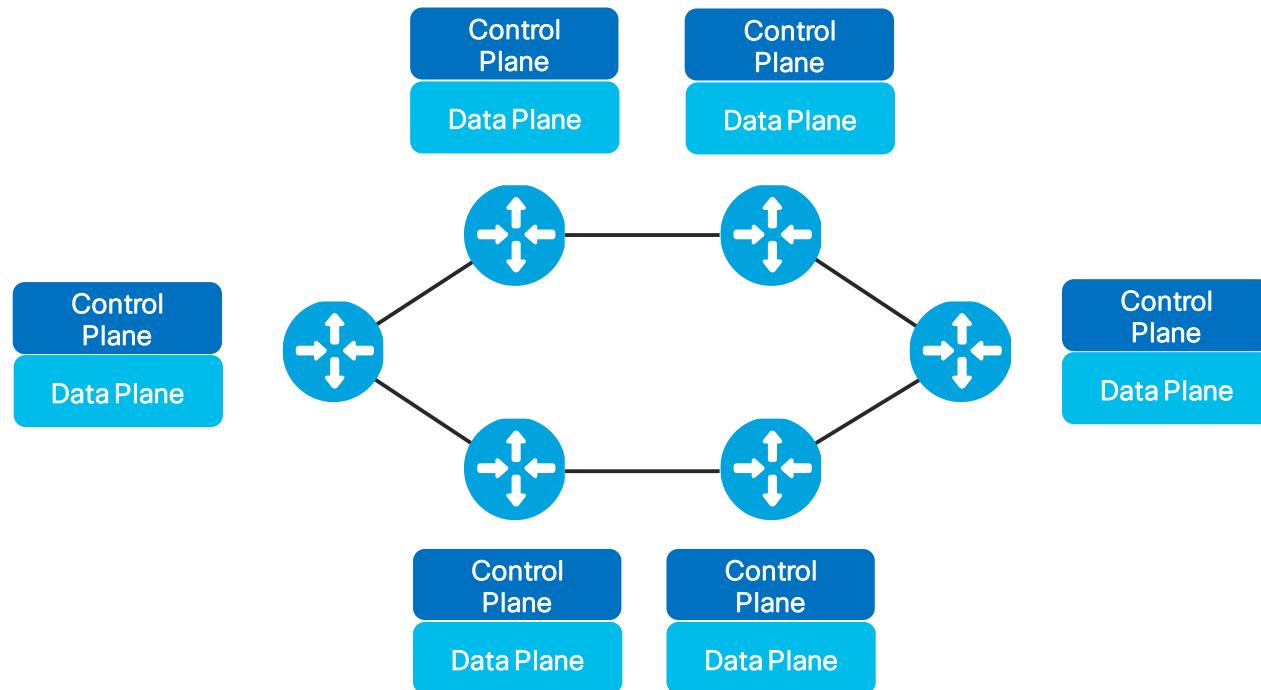
What is SDN?

Before the SDN Hype...

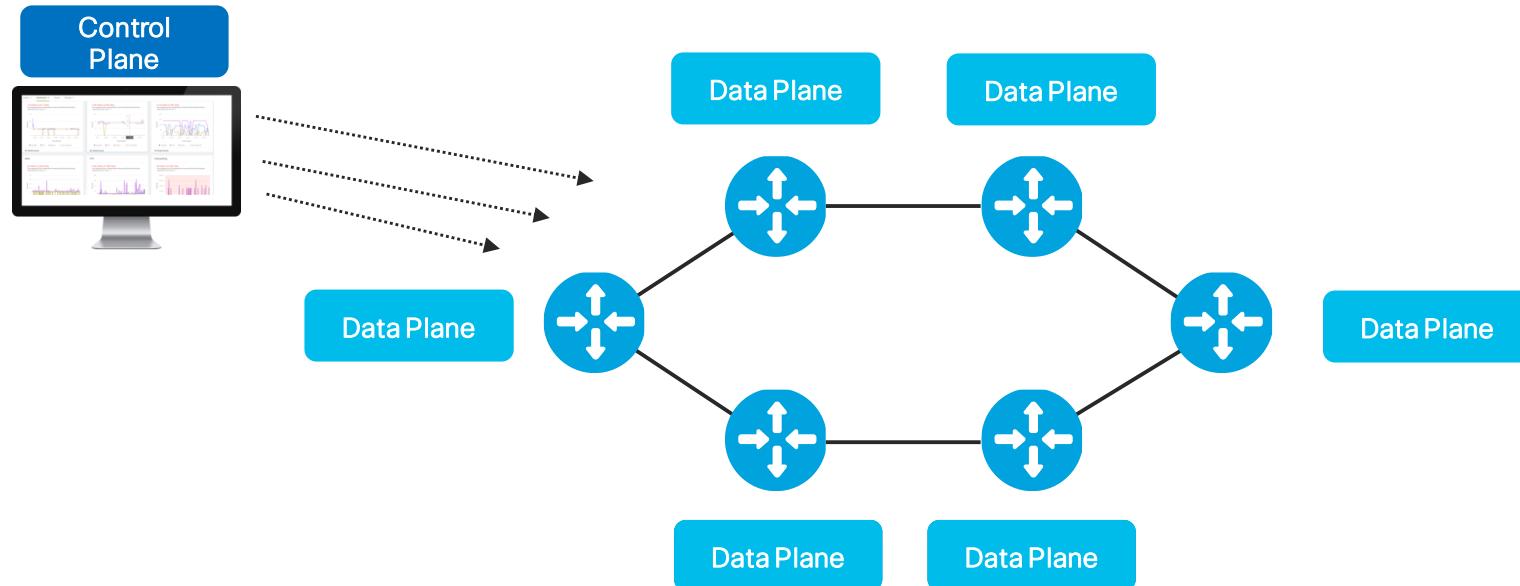


Networks were software-defined!

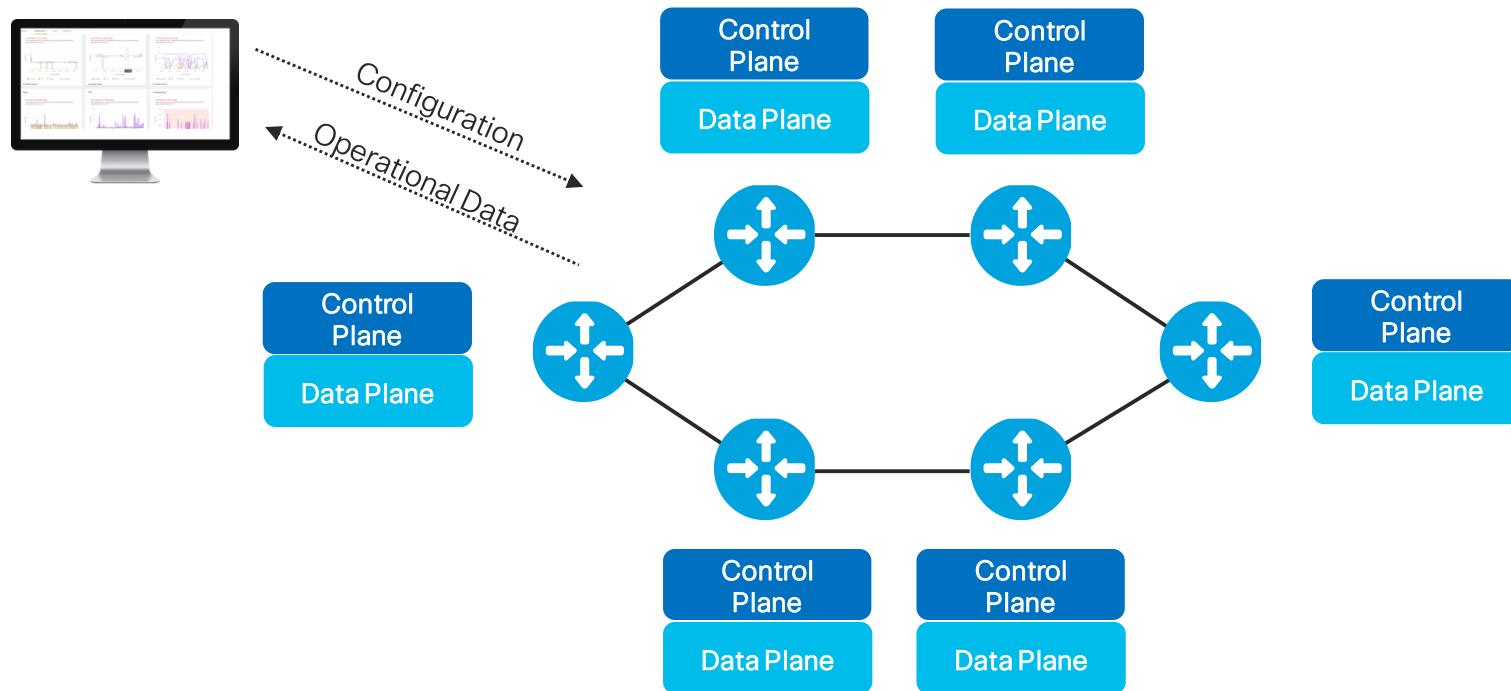
Traditional Networking...



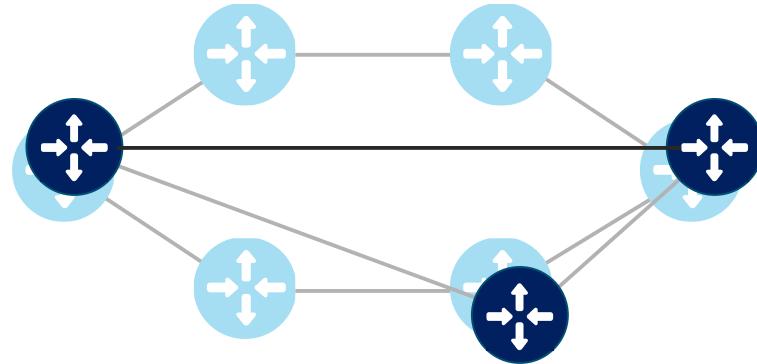
SDN: The original idea...



SDN: The reality...



SDN now includes overlay networks (more on that later)



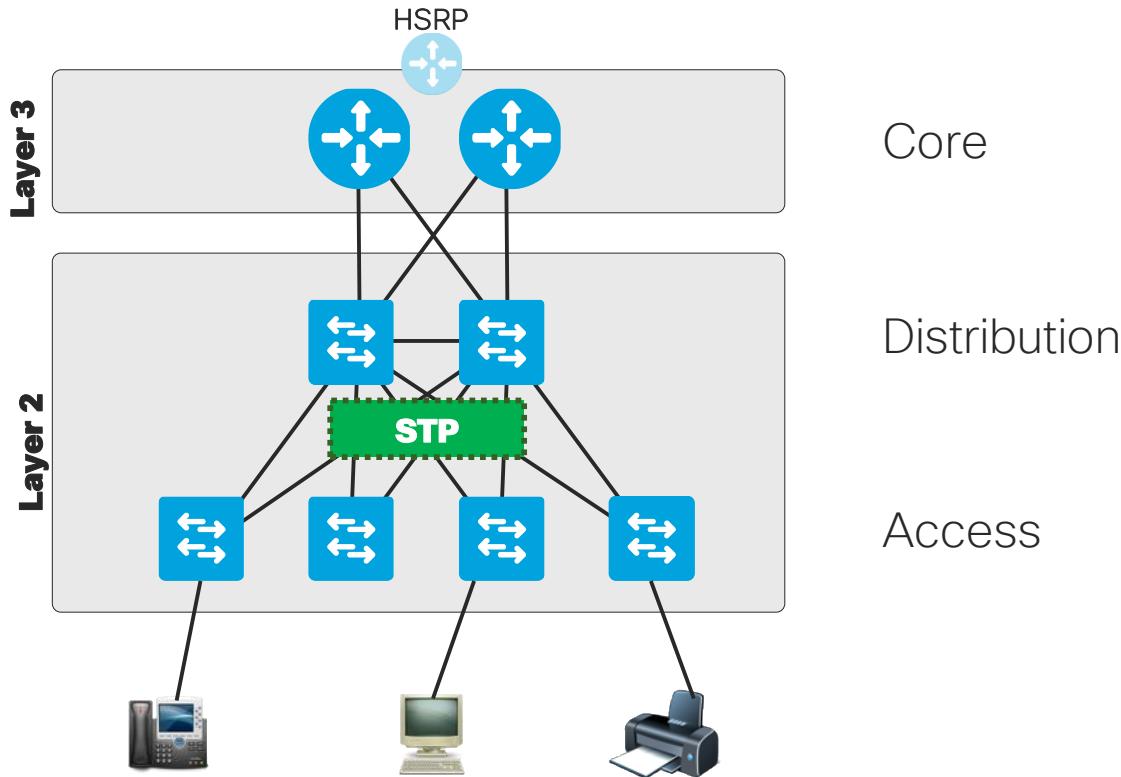
SDN in the campus



You make networking **possible**

Traditional Campus Networks

Default Gateways →



Traditional Network Challenges

Slow Convergence

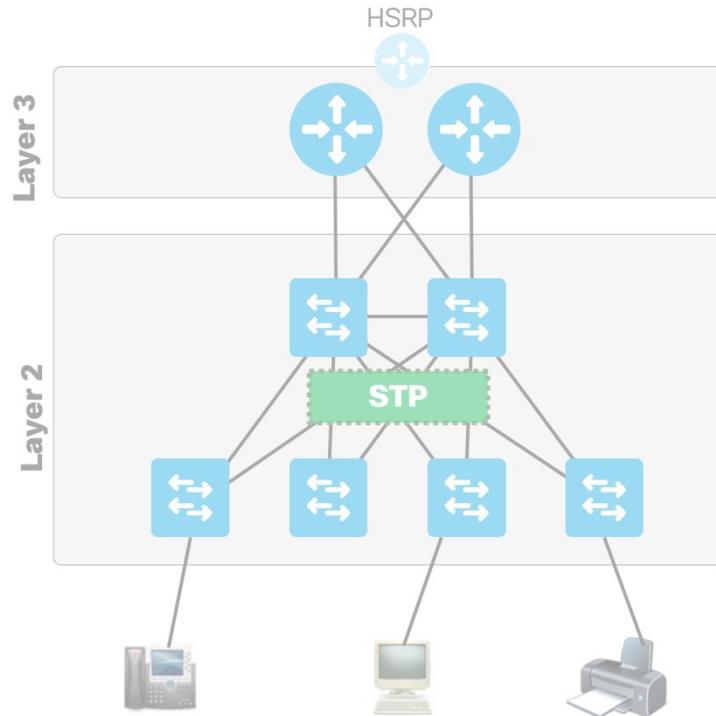
Inefficient Bandwidth Use

Sub-Optimal Routing

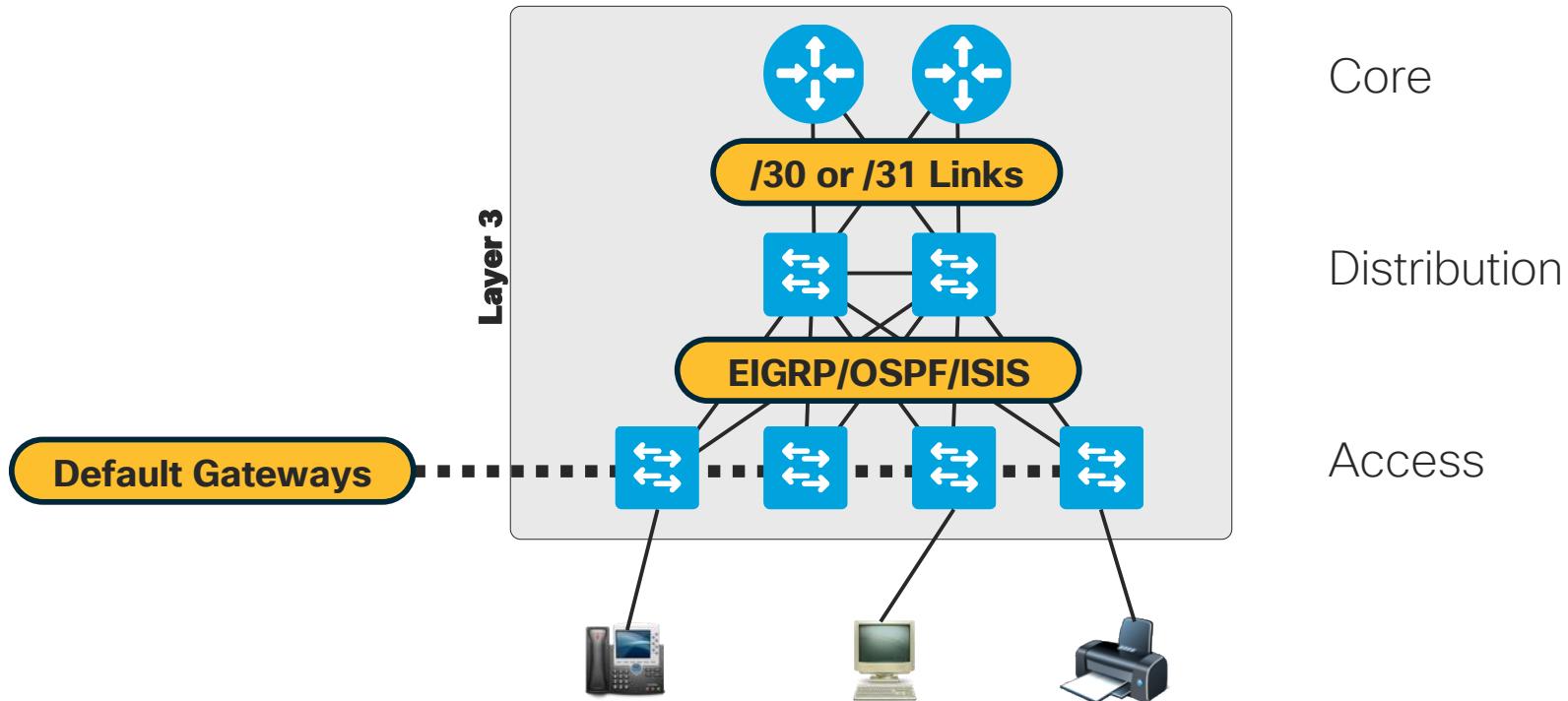
Attempts to solve

STP Improvements

Box Aggregation



Layer 3 Routed Network

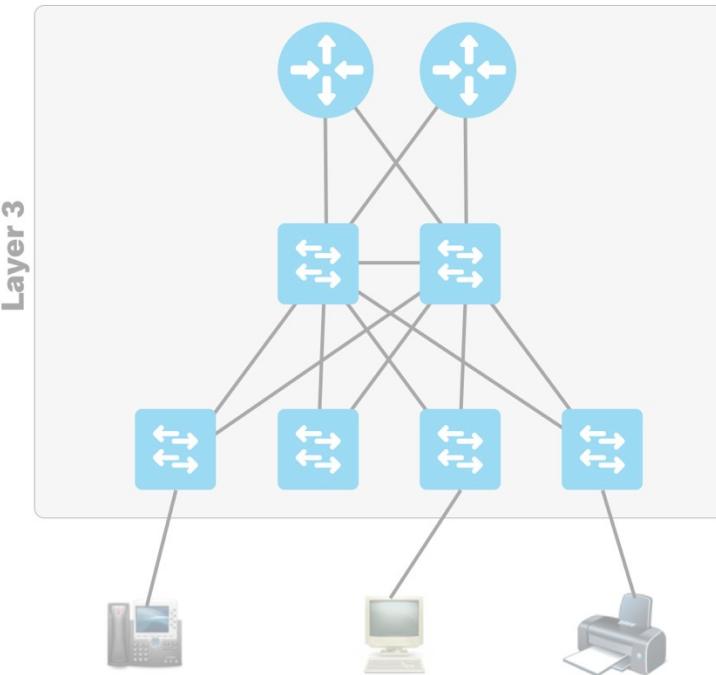


Layer 3 Routed Network Challenges

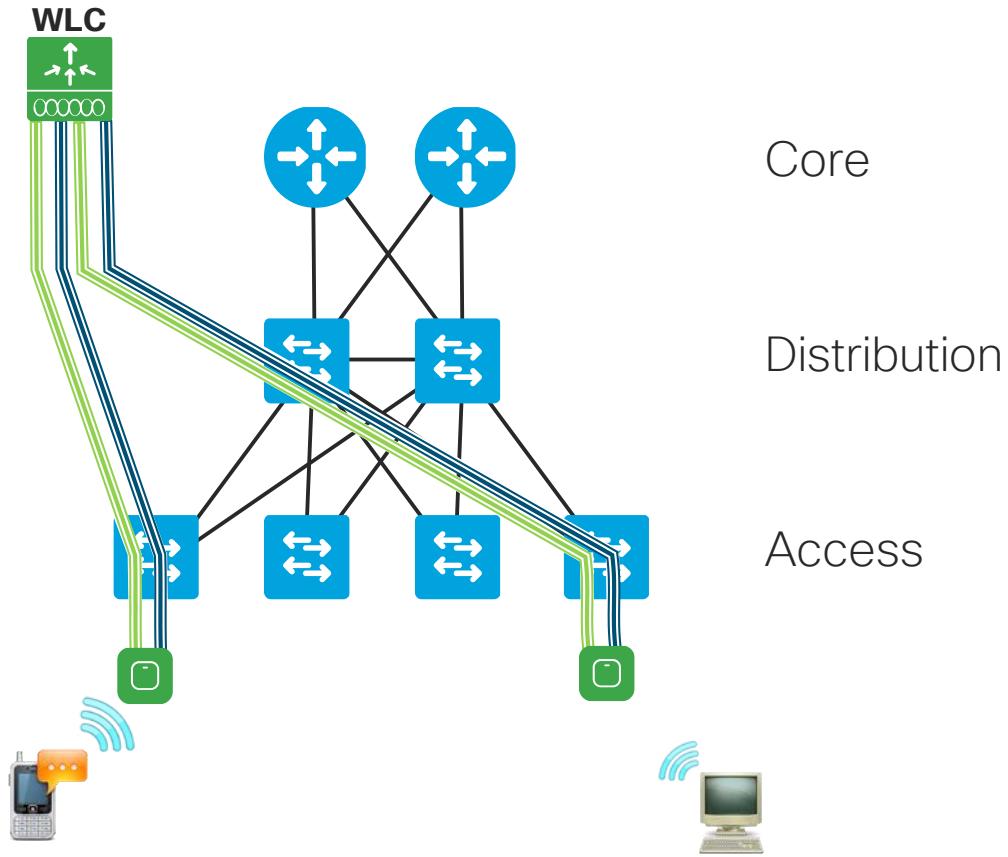
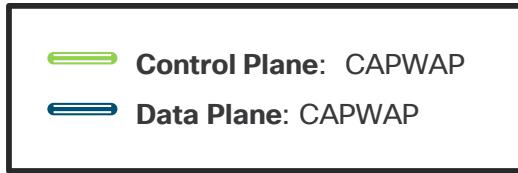
No VLAN Stretch

Gateway Redundancy

Device Mobility



Wireless

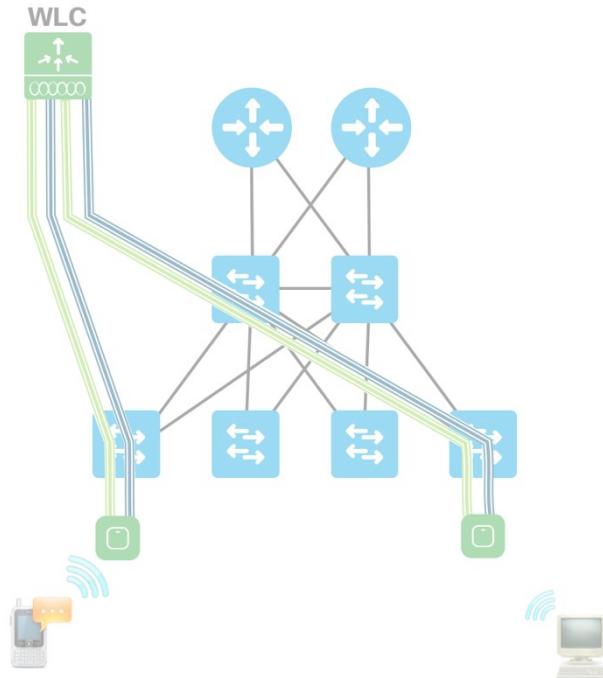


Wireless Challenges

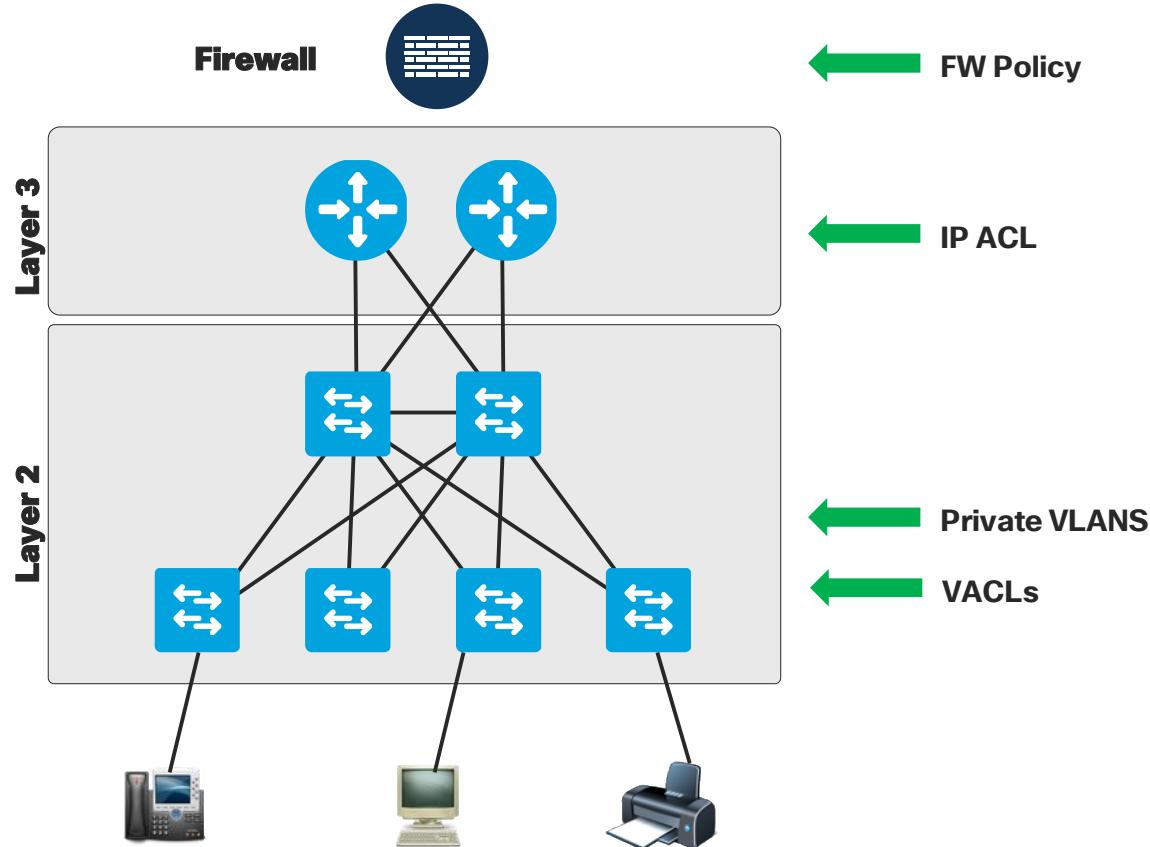
Traffic Hairpinning

Different Management

Device Mobility



Security



Challenges for Security

IP Based

Out of Control ACE's

No Microsegmentation

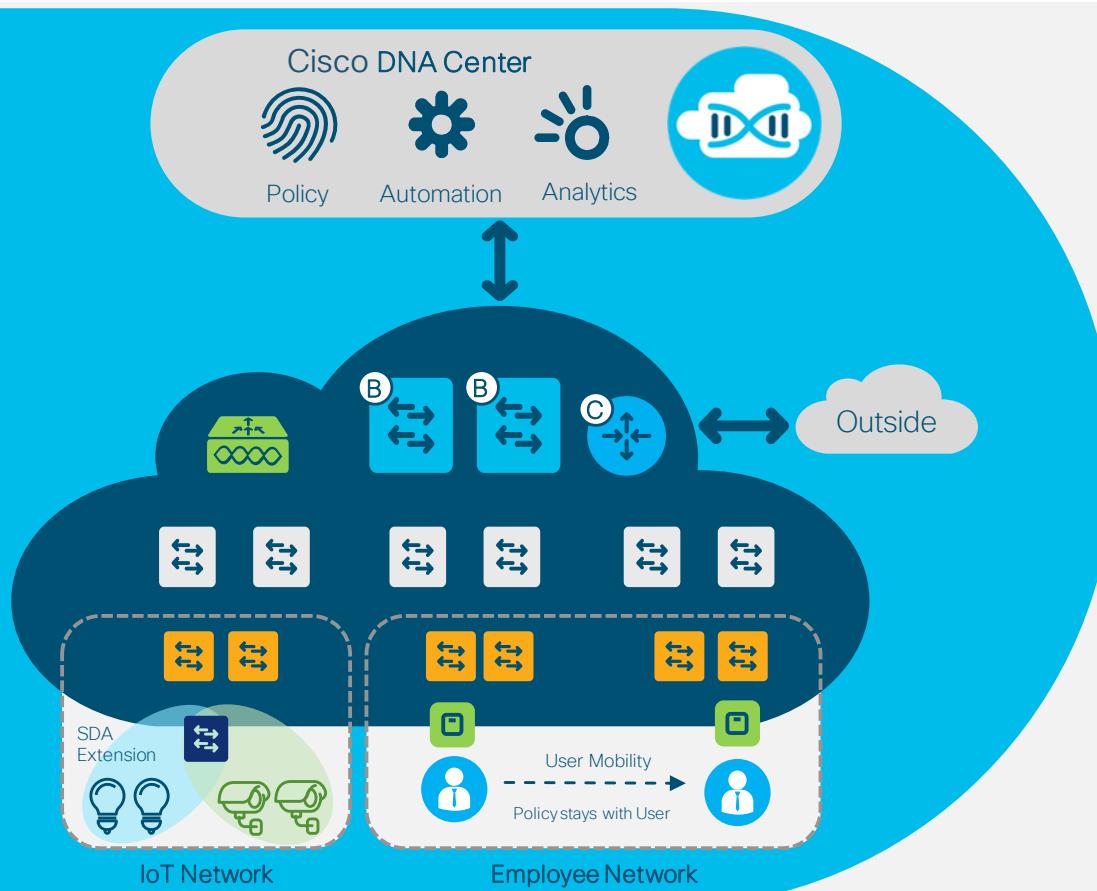
```
access-list 102 permit icmp 186.246.40.245 0.255.255.255 eq 3508 191.139.67.54 0.0.1.255 eq 1479
access-list 102 permit ip 209.111.254.187 0.0.1.255 gt 4640 93.99.173.34 255.255.255.255 gt 28
access-list 102 permit ip 184.232.88.41 0.0.31.255 lt 2247 186.33.104.31 255.255.255.255 lt 4481
access-list 102 deny ip 106.79.247.50 0.0.31.255 gt 1441 96.62.207.209 0.0.0.255 gt 631
access-list 102 permit ip 39.136.60.170 0.0.1.255 eq 4647 96.129.185.116 255.255.255.255 lt 3663
access-list 102 permit tcp 30.175.189.93 0.0.31.255 gt 228 48.33.30.91 0.0.0.255 gt 1388
access-list 102 permit ip 167.100.52.185 0.0.1.255 lt 4379 254.202.200.26 255.255.255.255 gt 4652
access-list 102 permit udp 172.16.184.148 0.255.255.255 gt 4163 124.38.159.247 0.0.0.127 lt 3851
access-list 102 deny icmp 206.107.73.252 0.255.255.255 lt 2465 171.213.183.230 0.0.31.255 gt 1392
access-list 102 permit ip 96.174.38.79 0.255.255.255 eq 1917 1.156.181.180 0.0.31.255 eq 1861
access-list 102 deny icmp 236.123.67.53 0.0.31.255 gt 1181 31.115.75.19 0.0.1.255 gt 2794
access-list 102 deny udp 14.45.208.20 0.0.0.255 lt 419 161.24.159.166 0.0.0.255 lt 2748
access-list 102 permit udp 252.40.175.155 0.0.31.255 lt 4548 87.112.10.20 0.0.1.255 gt 356
access-list 102 deny tcp 124.102.192.59 0.0.0.255 eq 2169 153.233.253.100 0.255.255.255 gt 327
access-list 102 permit icmp 68.14.62.179 255.255.255.255 lt 2985 235.228.242.243 255.255.255.255 lt 2286
access-list 102 deny tcp 91.198.213.34 0.0.0.255 eq 1274 206.136.32.135 0.255.255.255 eq 4191
access-list 102 deny udp 76.150.135.234 255.255.255.255 lt 3573 15.233.106.211 255.255.255.255 eq 3721
access-list 102 permit tcp 126.97.113.32 0.0.1.255 eq 4644 2.216.105.40 0.0.31.255 eq 3716
access-list 102 permit icmp 147.31.93.130 0.0.0.255 gt 968 154.44.194.206 255.255.255.255 eq 4533
access-list 102 deny tcp 154.57.128.91 0.0.0.255 lt 1290 106.233.205.111 0.0.31.255 gt 539
access-list 102 deny ip 9.148.176.48 0.0.1.255 eq 1310 64.61.88.73 0.0.1.255 lt 4570
access-list 102 deny ip 124.236.172.134 255.255.255.255 gt 859 56.81.14.184 255.55.255.255 gt 2754
access-list 102 deny icmp 227.161.68.159 0.0.31.255 lt 3228 78.113.205.236 255.55.255.255 lt 486
access-list 102 deny udp 167.160.188.162 0.0.0.255 gt 4230 248.11.187.246 0.255.255.255 eq 2165
access-list 102 deny udp 32.124.217.1 255.255.255.255 lt 907 11.38.130.82 0.0.31.255 gt 428
access-list 102 permit ip 64.98.77.248 0.0.0.127 eq 639 122.201.132.164 0.0.31.255 gt 1511
access-list 102 deny ip 247.54.117.116 0.0.0.255 gt 4437 126.69.159.104 0.0.1.255 gt 1015
```

Software-Defined Access



You make networking **possible**

Cisco DNA & SD-Access



Automated Network Fabric

Single Fabric for Wired & Wireless
with simple Automation



Identity-Based Policy & Segmentation

Decouples Security & QoS
from VLAN and IP Address



Insights & Telemetry

Analytics and Insights into
User and Application behavior



SD-Access Components

Control Plane

Data Plane

Policy Plane

Mgmt Plane



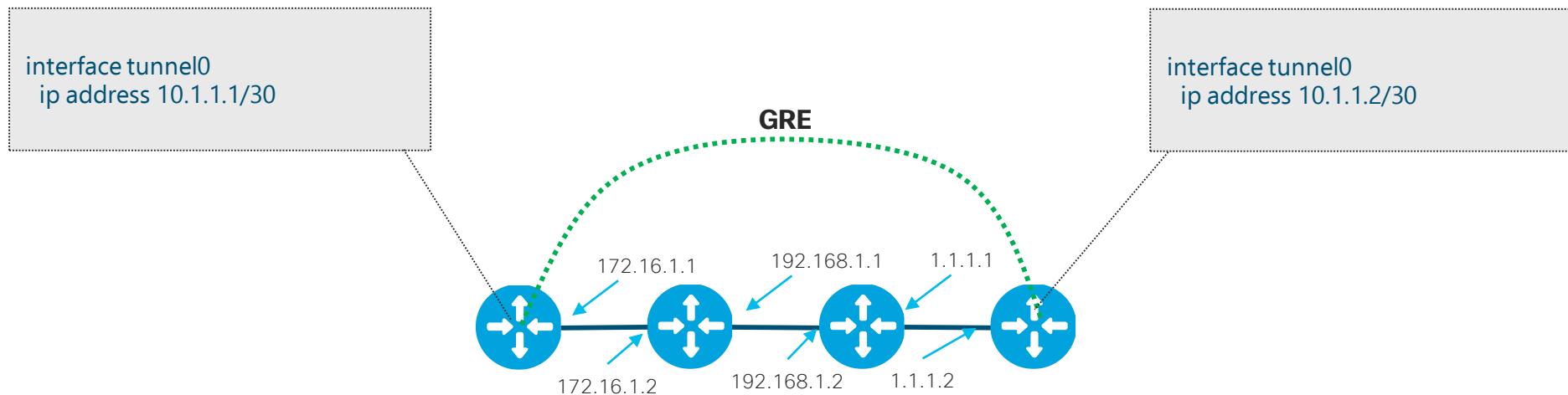
LISP

VXLAN

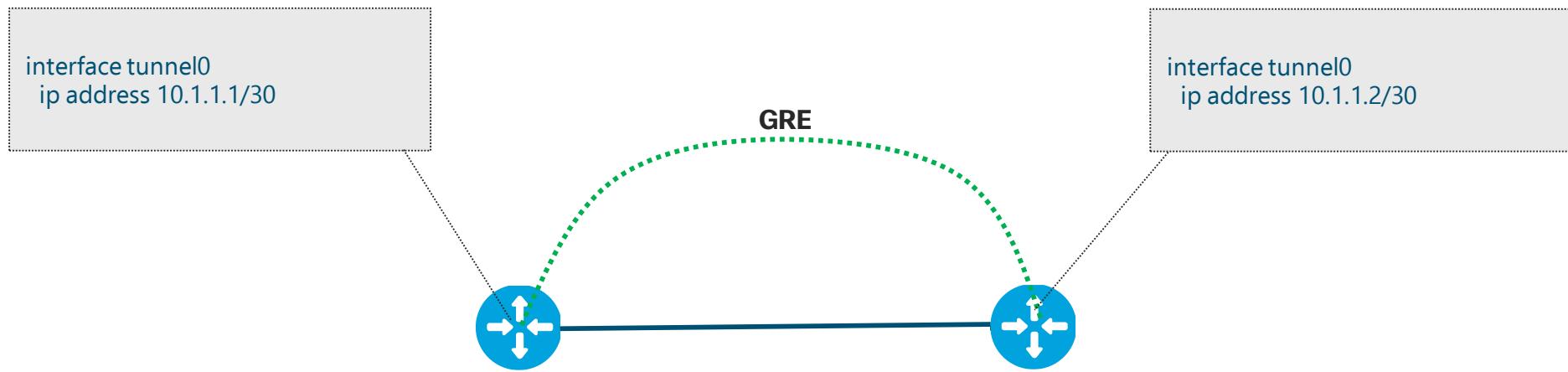
SGT

Cisco DNAC

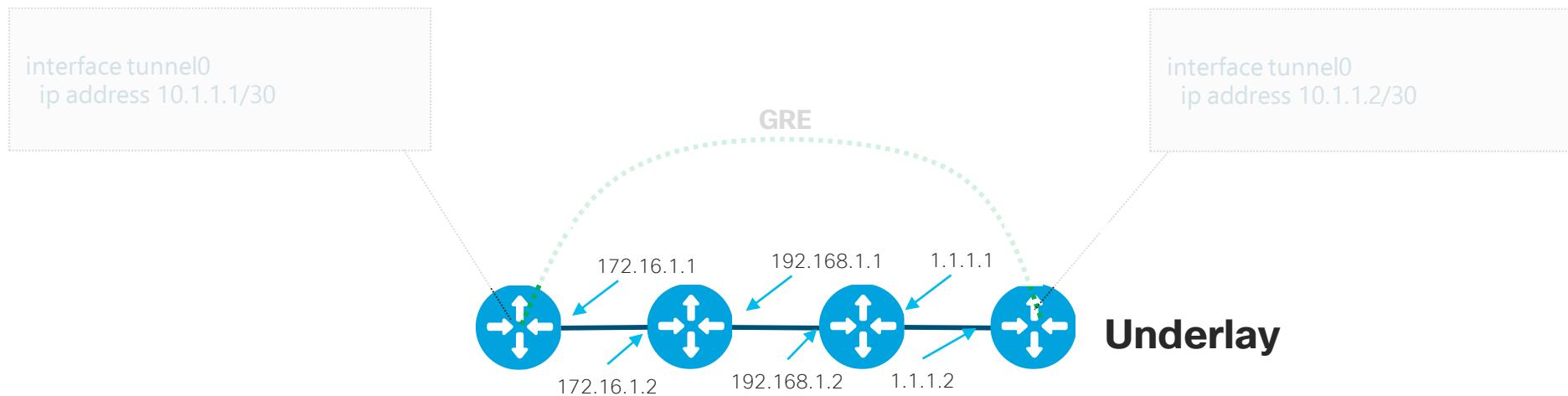
Overlays and Underlays



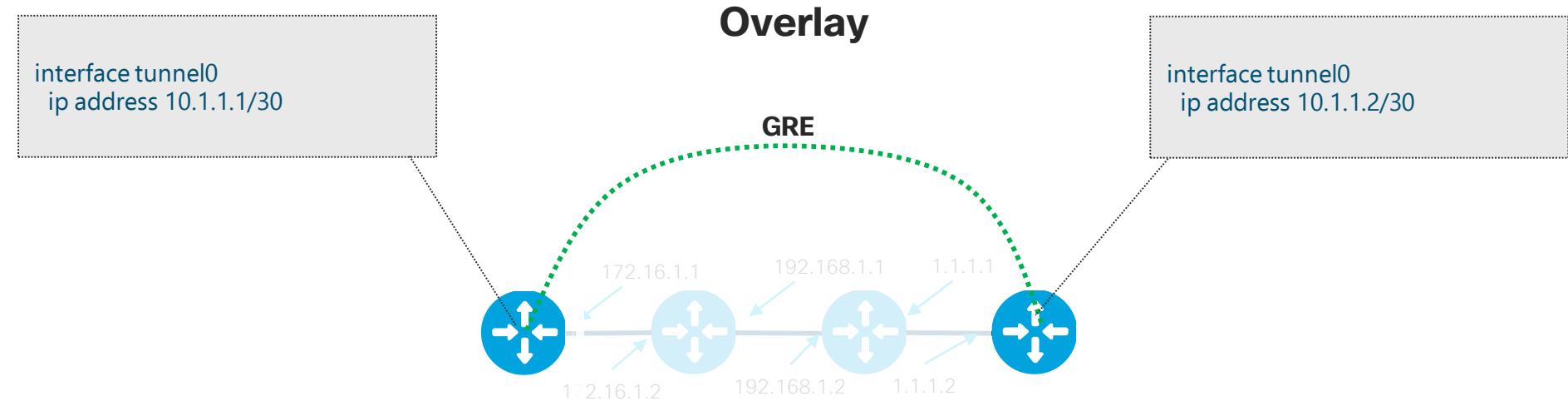
Overlays and Underlays



Overlays and Underlays

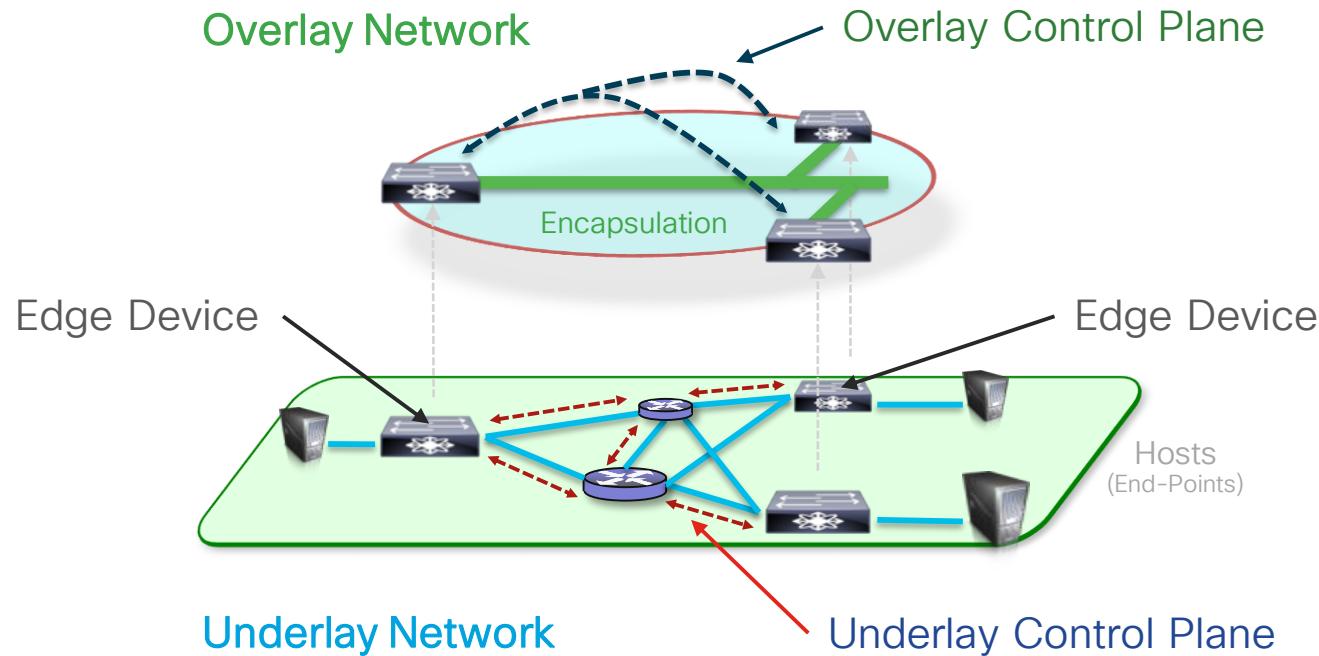


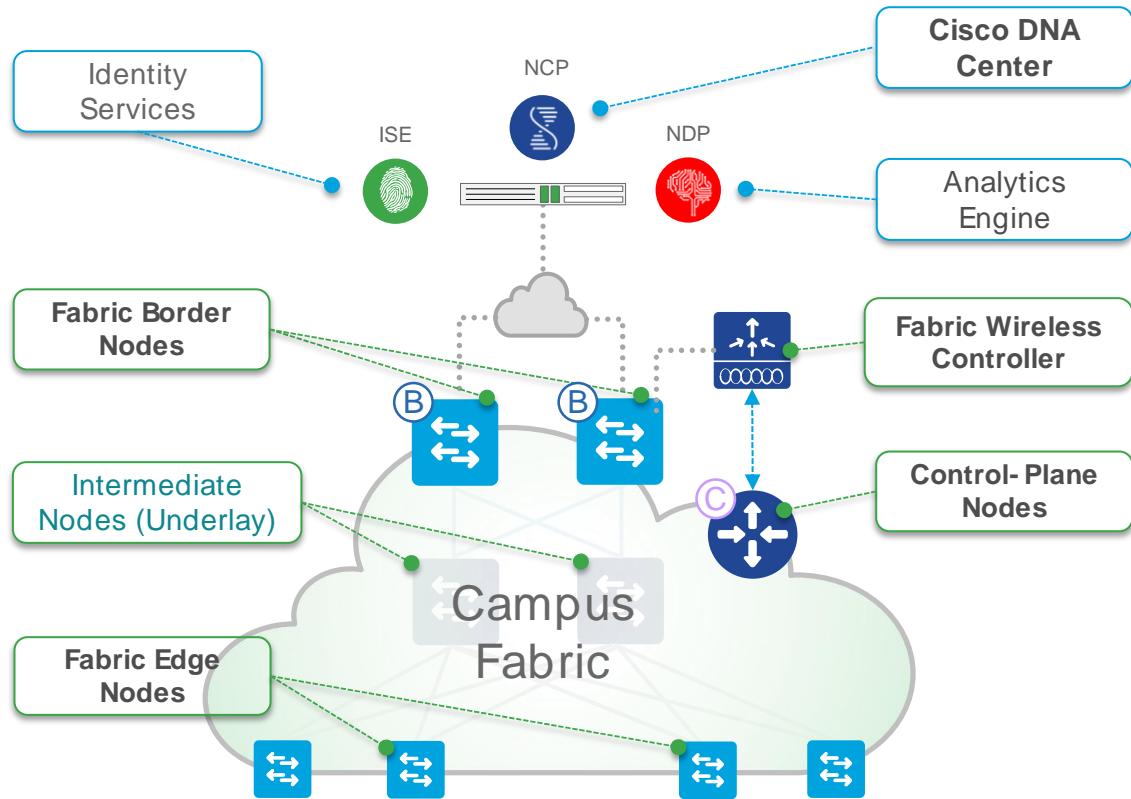
Overlays and Underlays



SD-Access

Fabric Terminology





- **Cisco DNA Center** – Enterprise SDN Controller provides GUI management and abstraction via Apps that share context
- **Identity Services** – External ID System(s) (e.g. ISE) are leveraged for dynamic Endpoint to Group mapping and Policy definition
- **Analytics Engine** – External Data Collector(s) (e.g. NDP) are leveraged to analyze Endpoint to App flows and monitor fabric status
- **Control-Plane Nodes** – Map System that manages Endpoint to Device relationships
- **Fabric Border Nodes** – A Fabric device (e.g. Core) that connects External L3 network(s) to the SDA Fabric
- **Fabric Edge Nodes** – A Fabric device (e.g. Access or Distribution) that connects Wired Endpoints to the SDA Fabric
- **Fabric Wireless Controller** – A Fabric device (WLC) that connects Wireless Endpoints to the SDA Fabric

Policy and Segmentation



You make networking **possible**



VN = Employees

Groups
Assigned to
Virtual Networks

Finance



HR



Marketing



VN = Employees

Finance



Marketing



Multiple Virtual Networks

VN = Contractors

Manufacturing



Accounting



VN = Employees

Finance



Marketing



VNs cannot talk by default

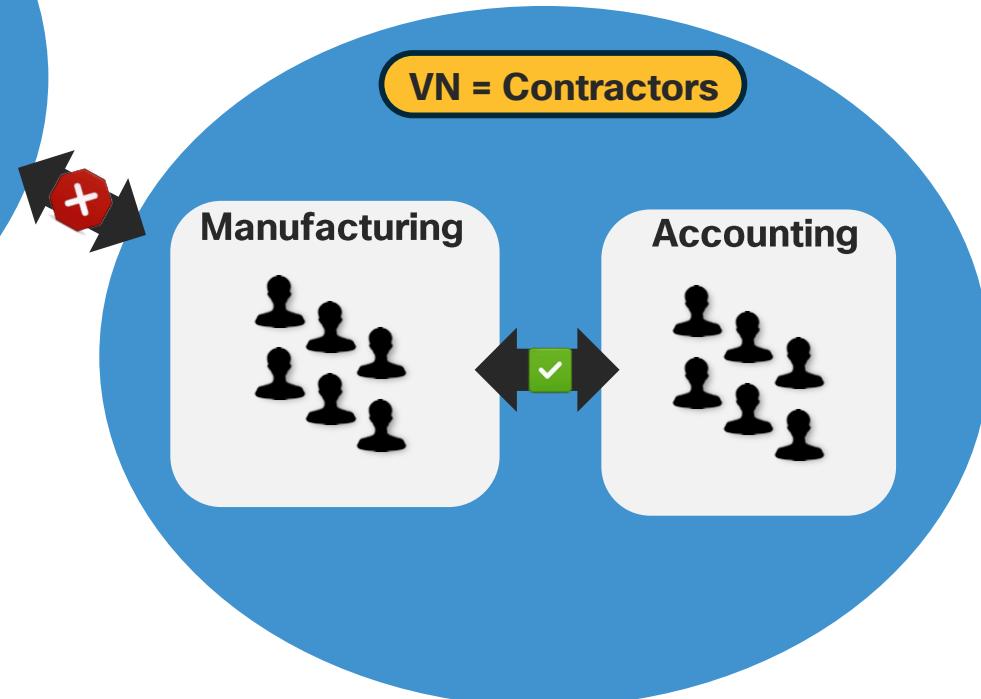
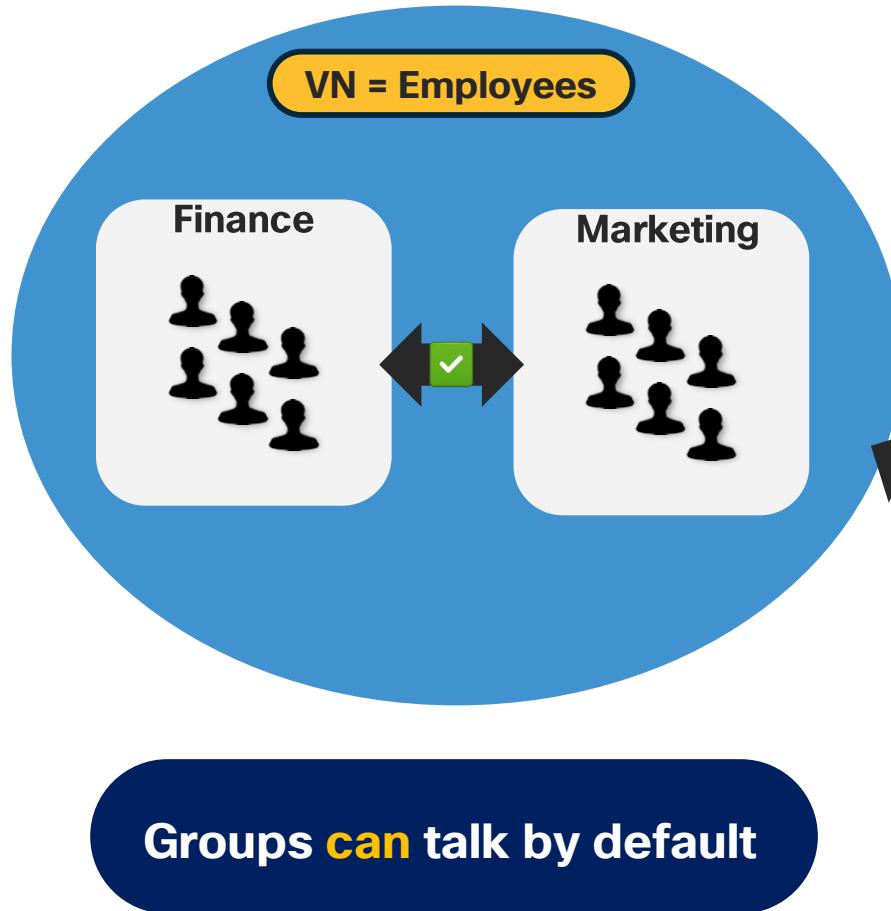
VN = Contractors

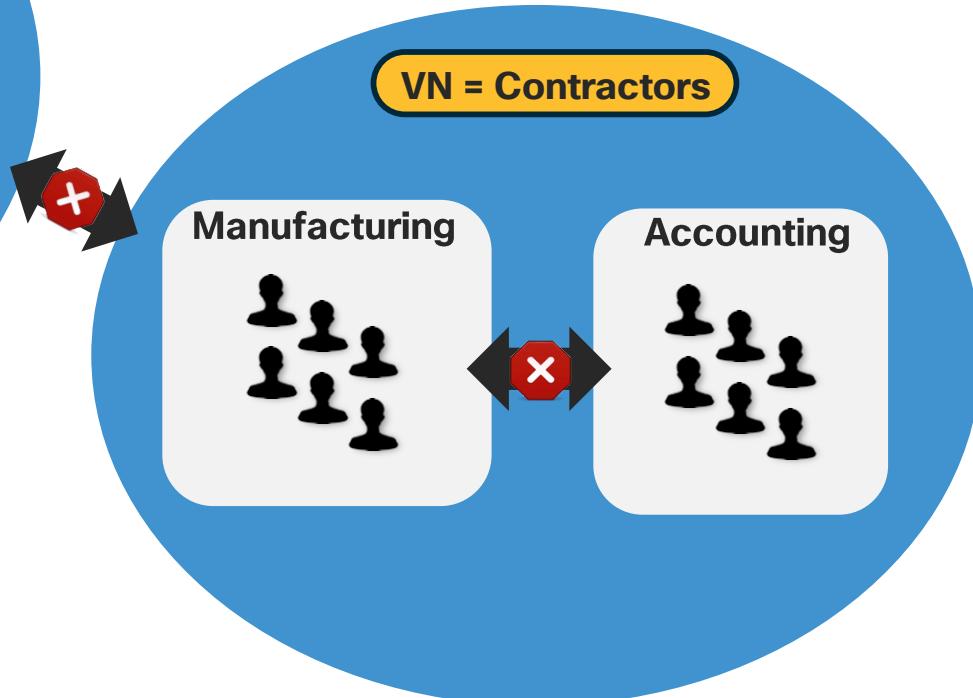
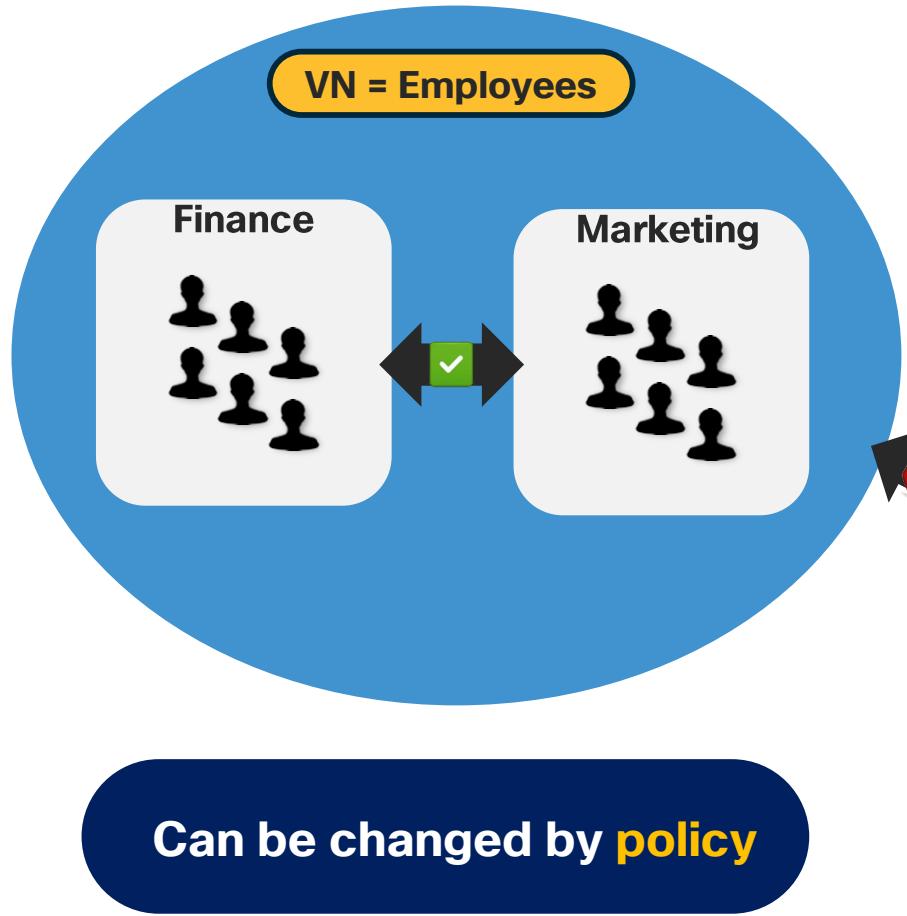
Manufacturing



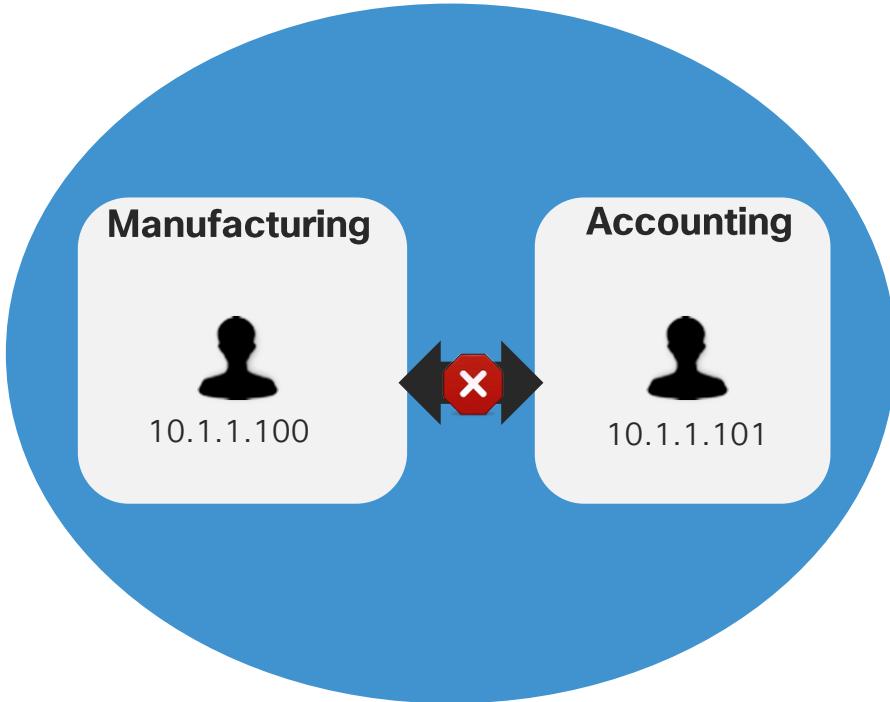
Accounting



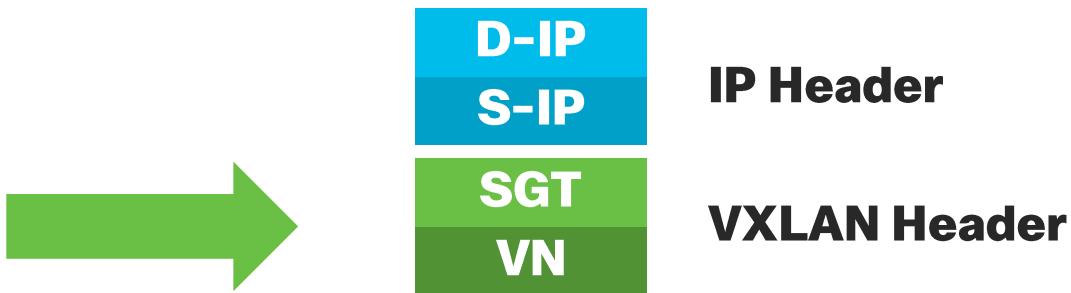


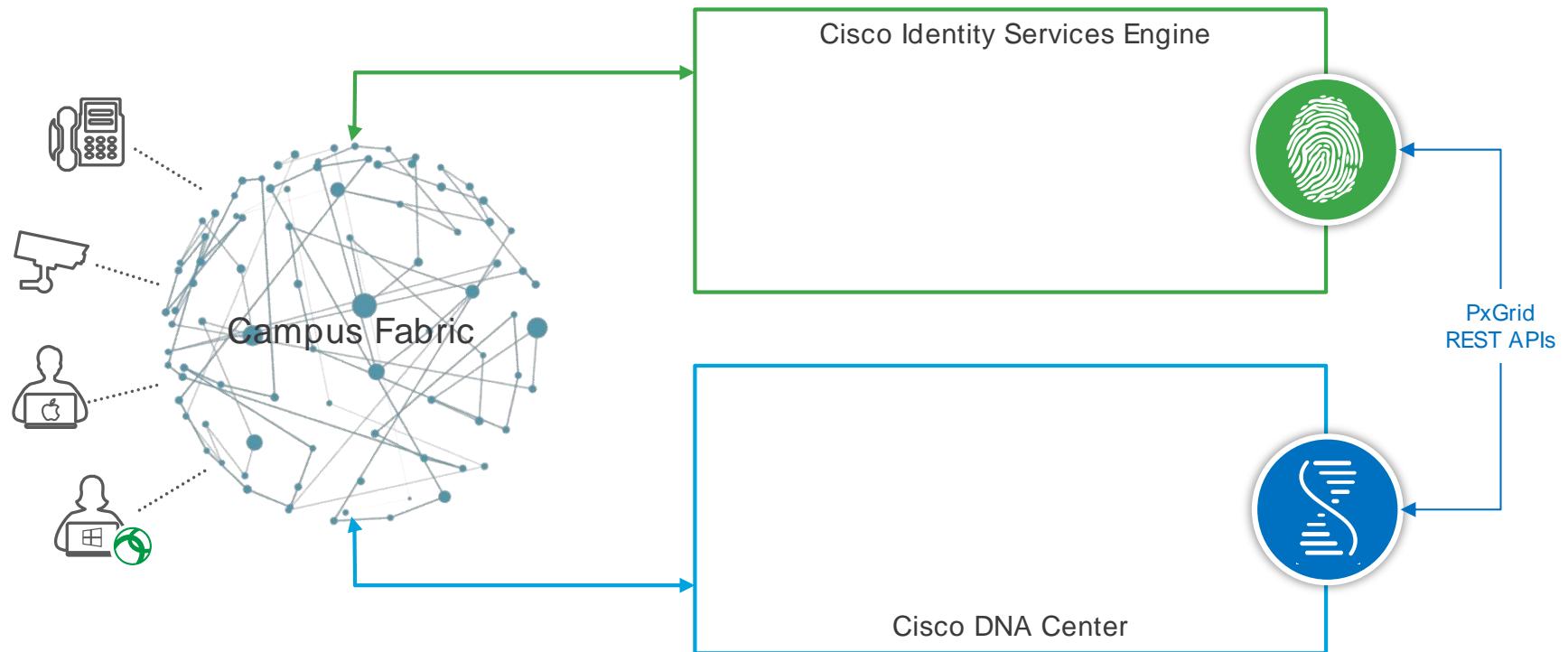


Users in **same subnet** can be isolated



Policy is tied to VN and group, not IP address...



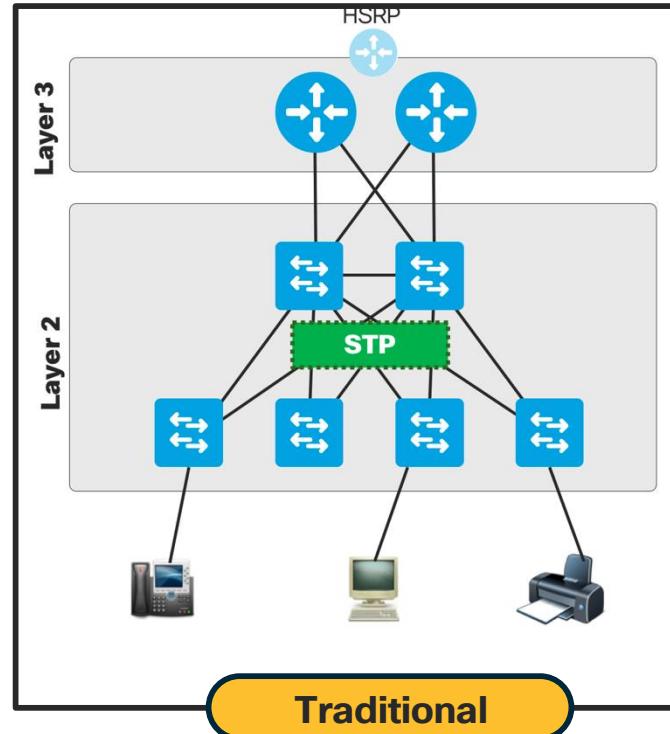
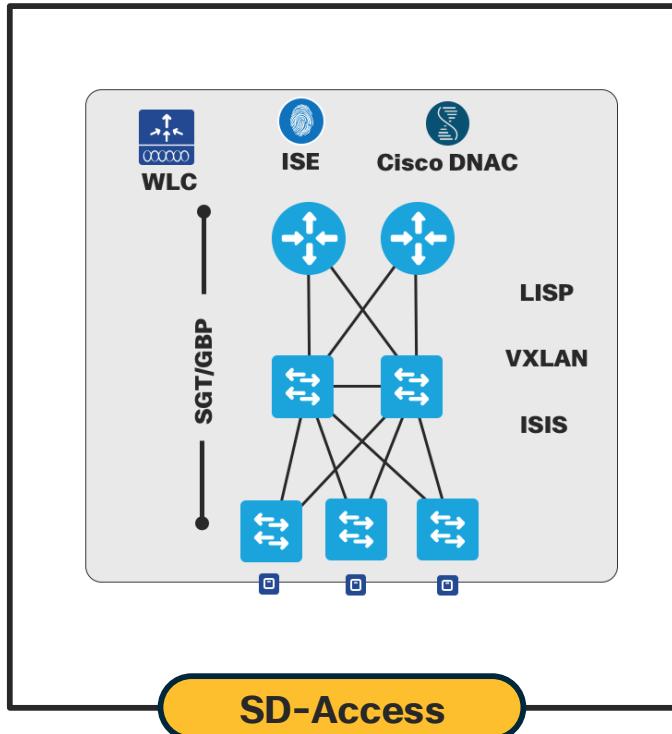




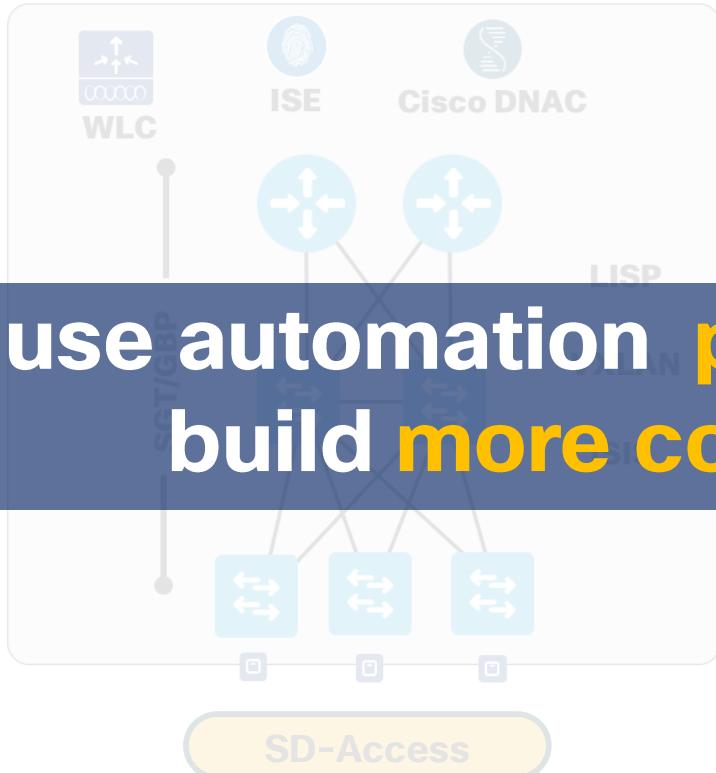
So what about the CCIE?



Which is less complex?



We use automation precisely to enable us to build more complex systems!



How many of these are familiar?

Management: ISE, Cisco DNAC

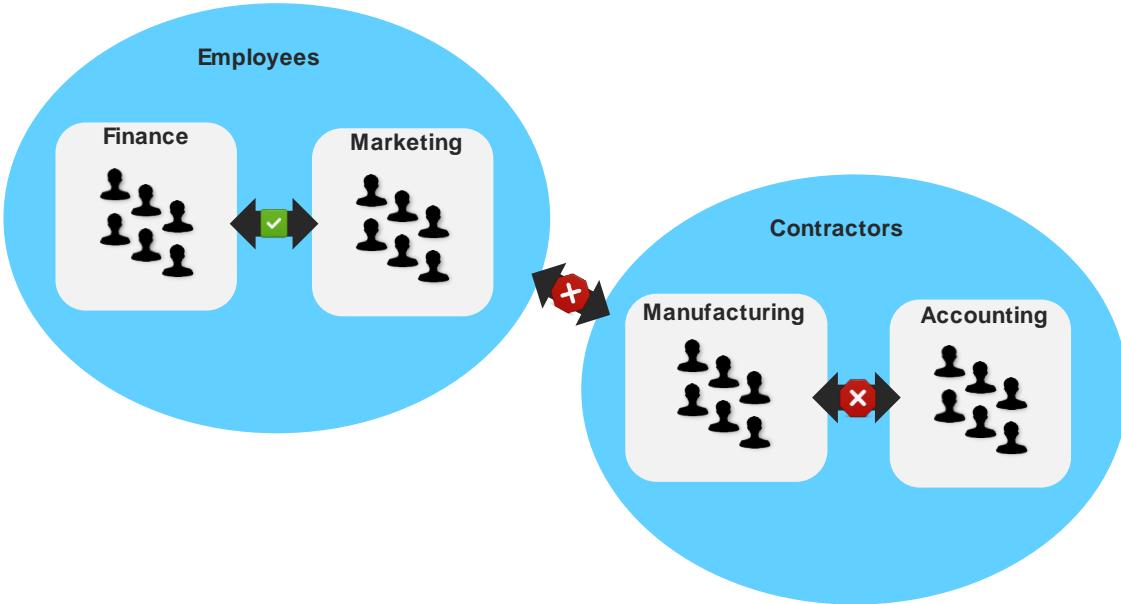
Policy Plane: VN/SGT

Control Plane: LISP

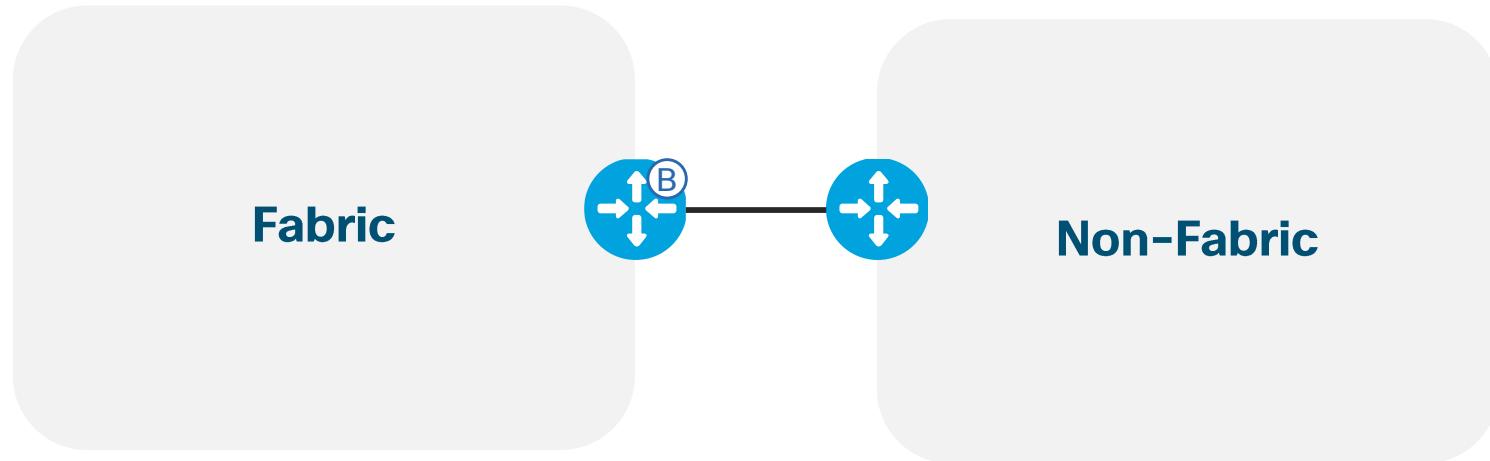
Routing Protocols: ISIS, BGP

IP, VXLAN

Ethernet



Complex policy constructs need skilled engineers to design.

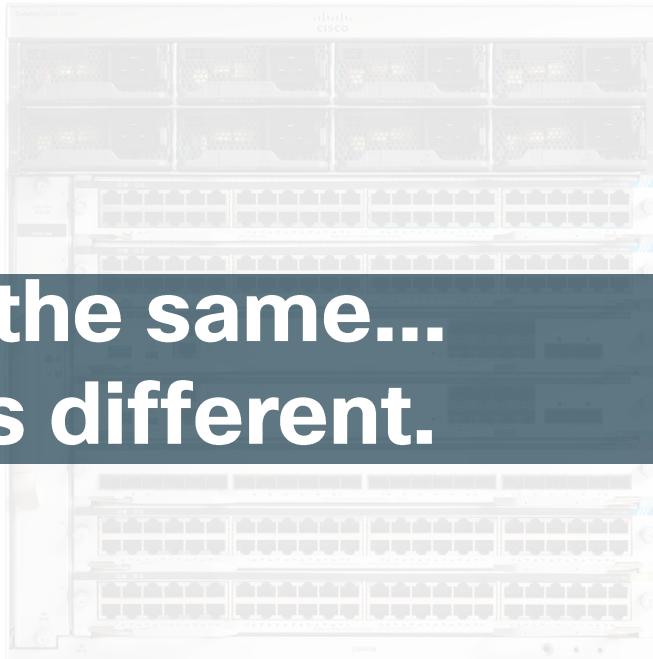


Border hand-off and fusion routers require protocol expertise

The CCIE in an SDN world



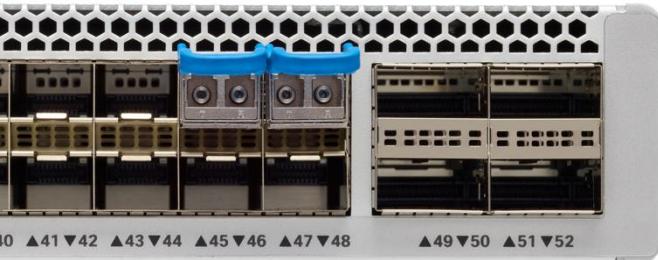
You make networking **possible**



**What we do is the same...
How we do it is different.**







Focus moves from box-level

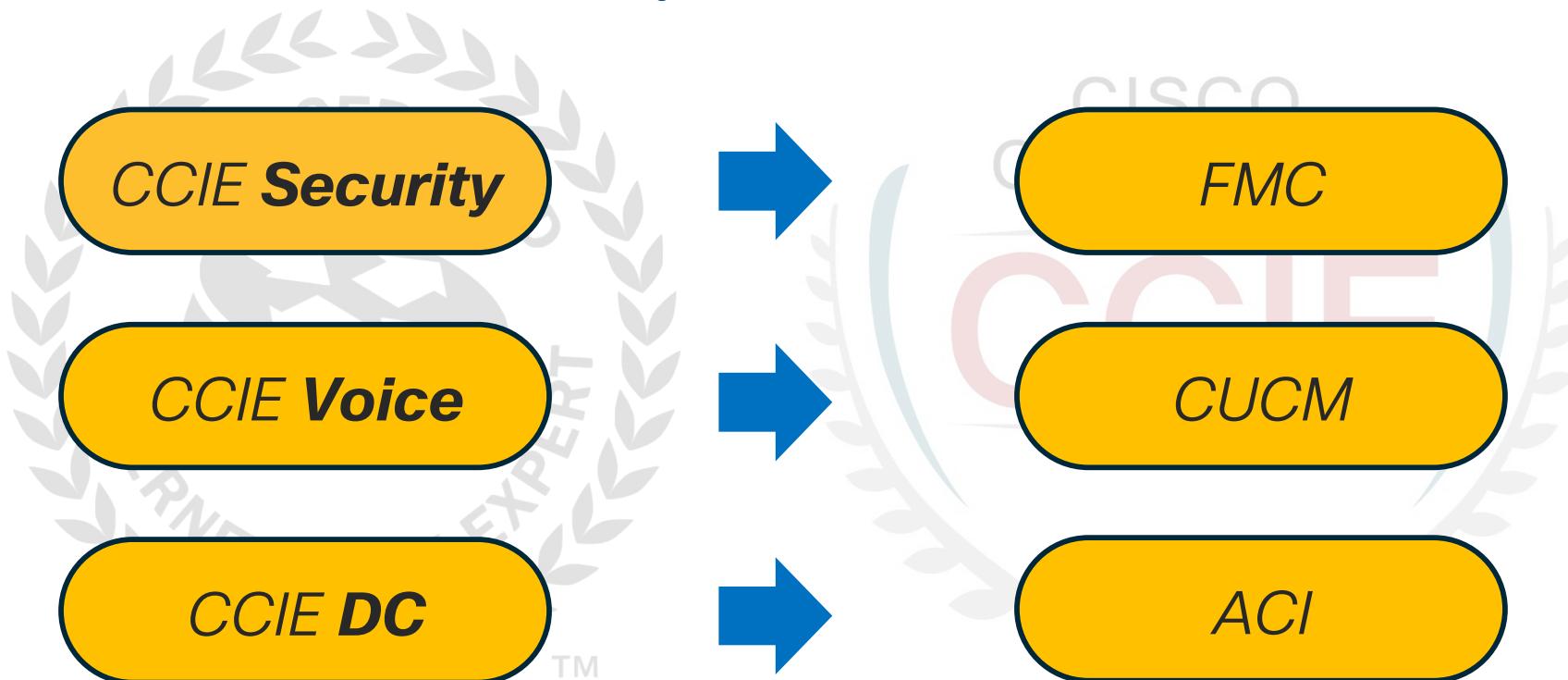
*How do I configure BGP?
How do I redistribute and tag routes?*



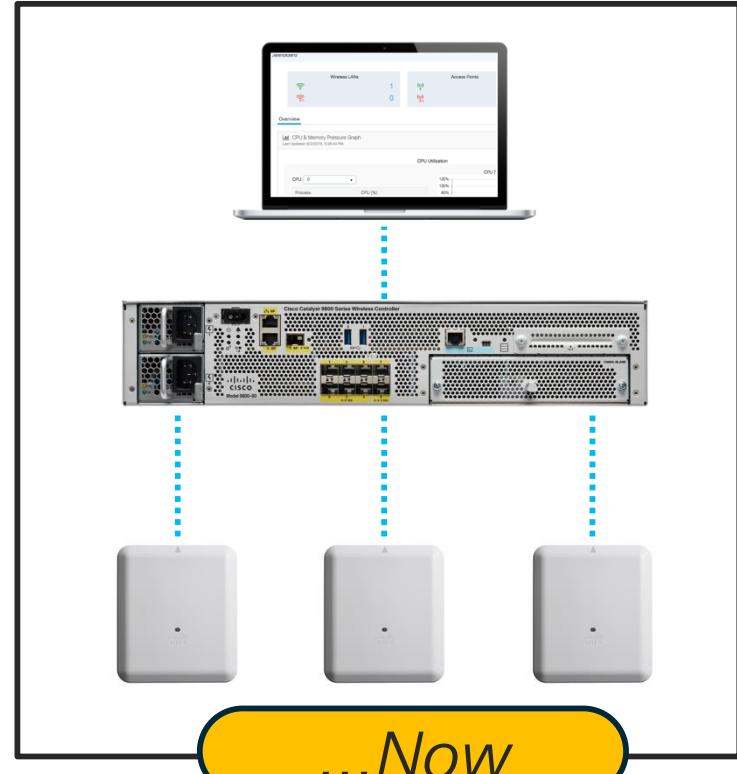
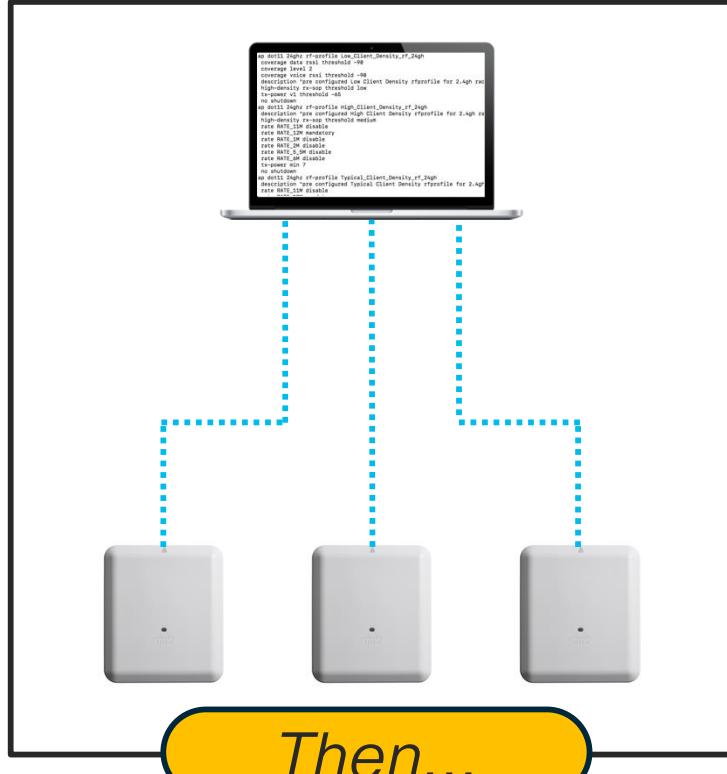
...to the solution level.

*How do I design policy?
How do I connect fabric to non-fabric?
How do I do cross-domain?
How do components integrate?*

The CCIE has already survived automation...



APs used to be manually configured!



How the CCIE will change



You make networking **possible**

Existing Exams No change until Feb. 2020

CCIE Evolving Technologies 1.1 Blueprint

This domain, worth 10 percent overall, ensures that all CCIE/CCDE candidates have a clear understanding of important cloud, network programmability, and IoT concepts.

A.1 Cloud

- A.1.b Describe cloud infrastructure and operations
- A.1.b.i Compute virtualization (containers and virtual machines)
- A.1.b.ii Connectivity (virtual switches, SD-WAN and SD-Access)
- A.1.b.iii Virtualization functions (NFVi, VNF, and L4/L1)
- A.1.b.iv Automation and orchestration tools (cloud center, Cisco DNA Centre, and Kubernetes)

WRITTEN

CCIE Evolving Technologies 1.1 Blueprint

This domain, worth 10 percent overall, ensures that all CCIE/CCDE candidates have a clear understanding of important cloud, network programmability, and IoT concepts.

A.2 Network Programmability

- A.2.a Describe architectural and operational considerations for a programmable network
 - A.2.a.i Data models and structures (YANG, JSON and XML)
 - A.2.a.ii Device programmability (gRPC, NETCONF and RESTCONF)
 - A.2.a.iii Controller based network design (policy driven configuration and northbound/southbound APIs)
 - A.2.a.iv Configuration management tools (agent and agent-less) and version control systems (Git and SVN)

Coming soon...



60%
"Classic" Networking

25%
Software Defined

15%
Programmability

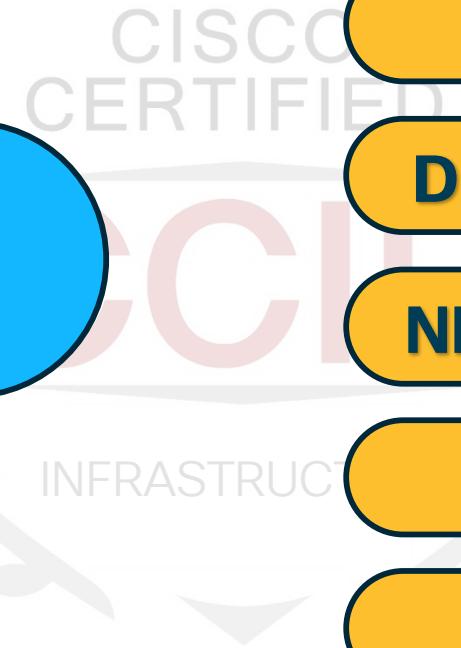


25%

Software Defined

SD-Access

SD-WAN



15%
Programmability

XML/JSON

Scripting

DNAC/vManage APIs

NETCONF/RESTCONF

gRPC Telemetry

Ansible

So what next?



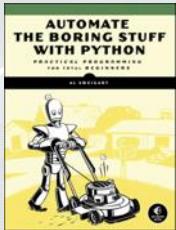
You make networking **possible**

The new blueprint is **intimidating**...



...but CCIE blueprints are **always** intimidating!

Programmability: Learn



Automate the Boring Stuff with Python, Al Sweigart

Great introduction to Python focused on automation. (Not specifically network automation.) Covers Python 3.0 only. Assumes zero knowledge. Read Excel docs, generate PDFs, etc. Highly recommended.



DevNet

Videos, learning labs, code exchange

Programmability: Lab

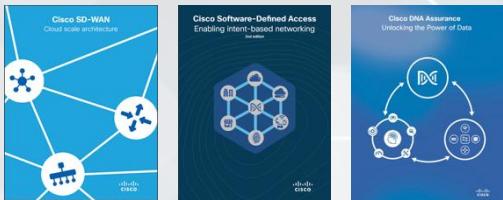


DevNet: Sandboxes and Learning Labs



Your own laptop: YangExplorer, Python, Ansible, GitHub, NETCONF/RESTCONF

SDN: Learn



A screenshot of the Cisco Community website under the 'Networking' category. It includes navigation links for Technology & Support, For Partners, Customer Connection, Events, and Member. Below this are sections for 'Steve's News', 'CCP Deep Dive', 'Advertise Demo', and 'Wi-Fi 6'. The main content area shows posts for 'Cisco Digital Network Architecture (DNA)', 'IPv6', and 'Optical Networking', each with a post count, reply count, and timestamp.

Category	Post Count	Reply Count	Latest Post
Cisco Digital Network Architecture (DNA)	246	730	06-07-2019 10:29 AM
IPv6	818	2667	06-07-2019 11:57 AM
Optical Networking	2069	5509	06-05-2019 02:06 PM

EN Books

Produced by Cisco technical marketing team. Hard copy and free PDF. Links on slide to follow.

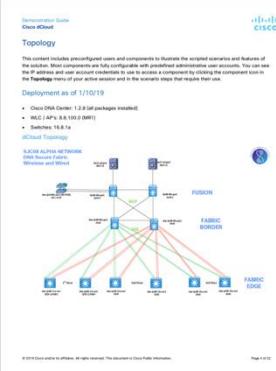


Cisco Communities

<http://communities.cisco.com>

Technical articles. Questions answered by experts.

SDN: Lab



dCloud

<http://dcloud.cisco.com>

Introductory level labs available for numerous SDx/IBN technologies.



Learning Partners

We will be working with learning partners to ensure SDx/IBN lab coverage.

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



CISCO

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



CISCO

Cisco DNA Assurance
Unlocking the Power of Data



CISCO

Our books...

<http://cs.co/cat9kbook>

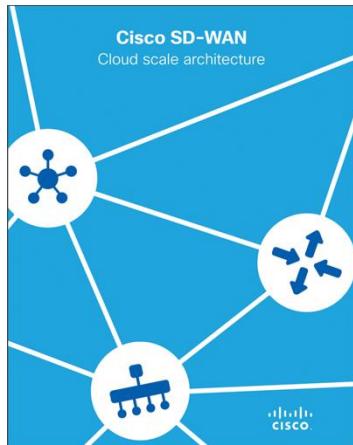
<http://cs.co/sdabook>

<http://cs.co/programmabilitybook>

<http://cs.co/wirelessbook>

<http://cs.co/assurancebook>

<http://cs.co/sdwanbook>



CISCO

IOS XE Programmability
Automating Device Lifecycle Management



CISCO

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



CISCO

Cisco live!

#CLUS

BRKCRT-3075

© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Public

133

So...Is it worth it?



You make networking **possible**

We still need **pilots** to fly planes...





We still need **YOU**
to build
networks...



**Continue to strive
for the best, and...**



A collage of networking equipment and cables. The top half shows a rack unit with various modules, a blue wireless router with two antennas, and a yellow telephone. The bottom half is dominated by a large pile of tangled network cables in black, blue, and white, with server racks visible in the background.

Happy labbing!



More on the changes

BRKCRT-3100: The making of tomorrow's expert: CCIE

Erik Vangrunderbeek, Yusuf Bhaiji

Tuesday, June 11, 08:30 AM - 10:00 AM

PSOCRT-1007: How are Certifications and Chameleons Alike? They Adapt to the Environment

Yusuf Bhaiji, Antonella Corno

Wednesday, June 12, 11:00 AM - 12:00 PM

Complete your online session evaluation



- 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 water bottle.
- All surveys can be taken in the Cisco Live Mobile App or by logging in to the Session Catalog on ciscolive.cisco.com/us.

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

Continue your education



Demos in the
Cisco campus



Walk-in labs



Meet the engineer
1:1 meetings



Related sessions



Thank you





A horizontal sequence of nine stylized lowercase 'i' characters, each composed of a colored dot at the top and a vertical bar below. The colors alternate and include blue, green, orange, and red.

You make **possible**