

NetAcad goes DevNet

Ayca Ruppert

Programmability Lead Germany

Cisco

@cisco_ayruppert



DEVNET
Create





Ayca Ruppert

System Engineer – DevNet Germany

Ayca has been a System Engineer for over 15 years. Starting 2005 in Enterprise Networking, moving to a partner changed her core technology to Collaboration. After 12 years she's now moving on to software and starts her coding carrier. She is a passionate Geek and loves connecting the dots. Her strong analyzing skills support projects to find a path to possible!



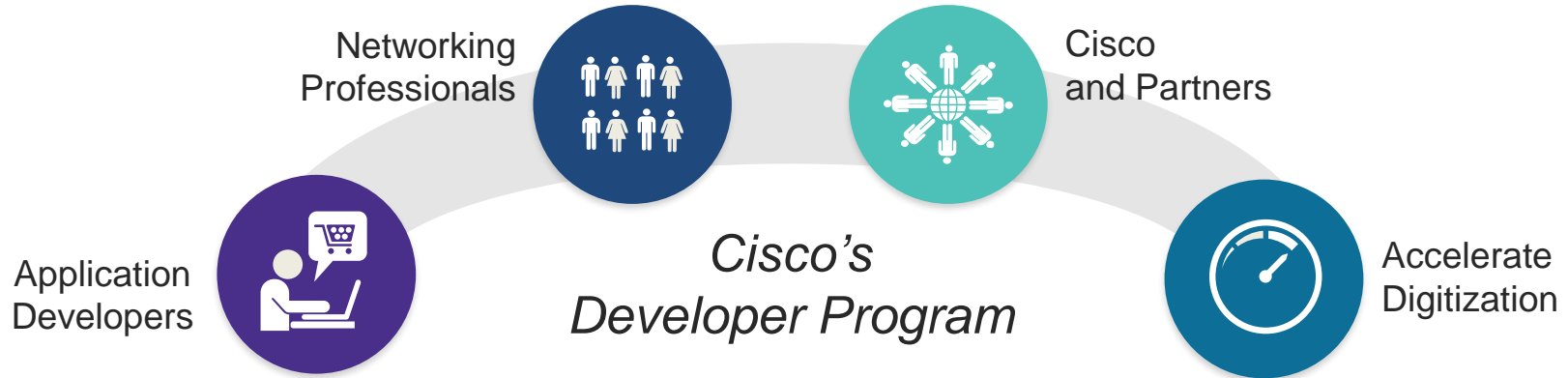
Agenda

- Was ist DevNet?
- Vorstellung DevNet Associate
- DeepDive into Model Driven Programmability



Was ist DevNet?

Cisco Developer Network – Cisco DevNet



Open APIs for Cisco Platforms

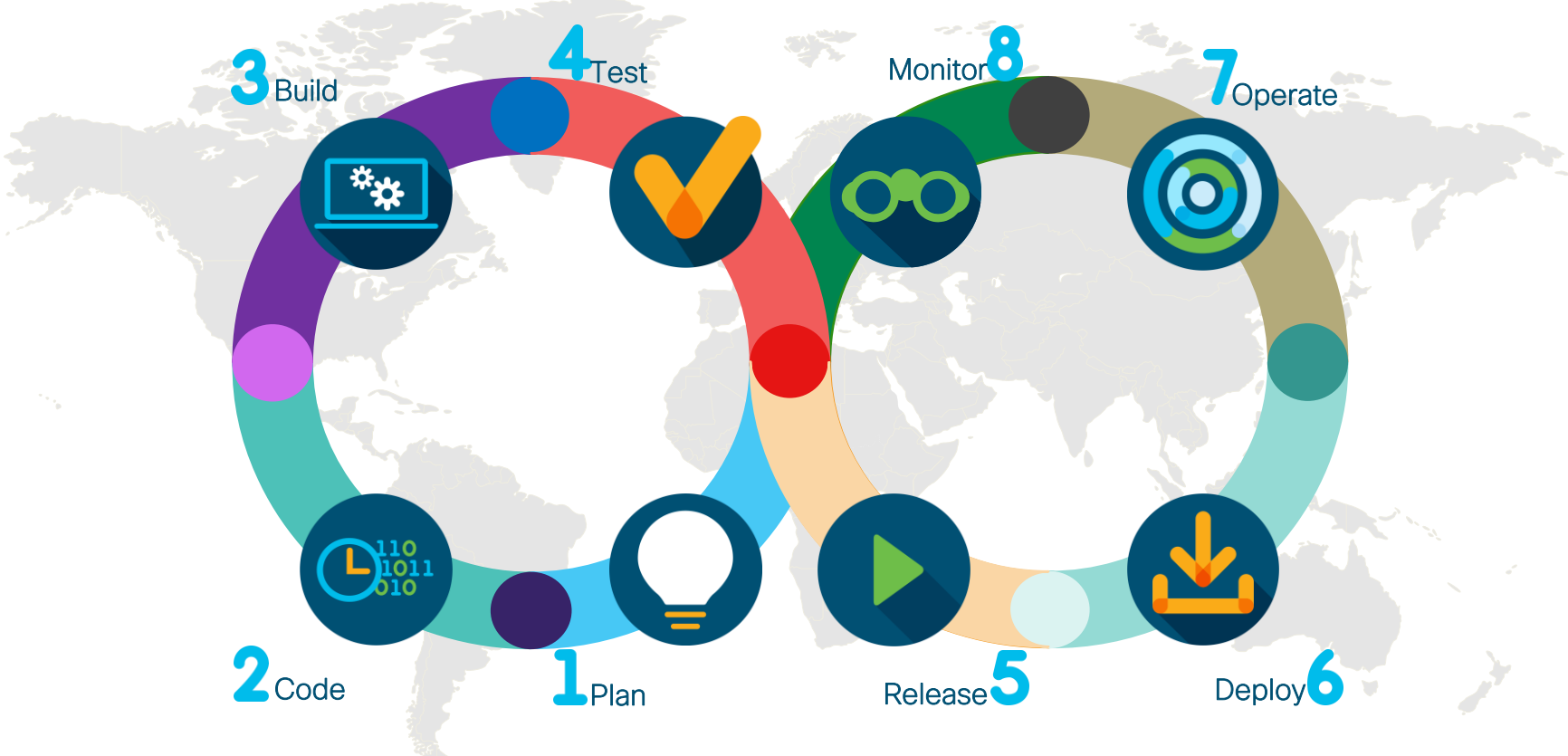


developer.cisco.com

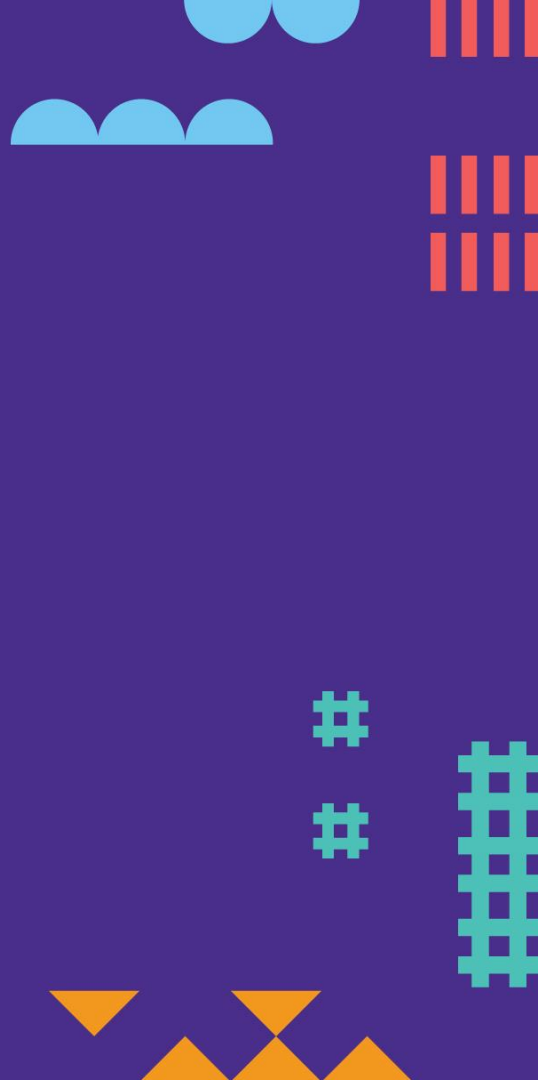


Was gibt es für Module?

DevOps und Geek-Stuff



DevNet Associate Certification



Cisco's expanded certification suite

	Associate Level	Specialist Level	Professional Level	Expert Level
Engineering				
Software				

Cisco's expanded certification suite



DevNet Certified Associate

Skilled in software development, network programmability, automation.

Aware of Cisco hardware, technologies, and solutions and network fundamentals.

Cisco Certified Network Associate

Skilled in how to operate and maintain Cisco hardware in a small enterprise network.

Aware of programmability and network automation capabilities.



Complementary balance and role alignment

DevNet Associate Curriculum

1. Software Development & Design
 - JSON / XML / YAML, Git, Waterfall, Agile etc.
2. Understanding and Using APIs
 - HTTP Header / Body / Request / Response, REST APIs
3. Cisco Platforms & Development
 - DNA-Center / Webex Teams / Meraki, Cisco SDKs etc.
4. Application Deployment & Security
 - Virtual Machines / Containers, XSS / SQL injection etc.
5. Infrastructure & Automation
 - Ansible / NSO / Puppet, NETCONF + YANG, RESTCONF
6. Network Fundamentals
 - CCNA : VLANs, MAC / IP Addresses, Routers / Switches, Protocols etc.

Cisco's expanded certification suite

DevNet Associate Course from NetAcad



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

Features	<ul style="list-style-type: none">• Online Curriculum with Formative and Summative Assessments• Hands-On Labs running Locally using Software tools• Introduction of a Project-Based Learning Framework
Target Audience	<ul style="list-style-type: none">• Vocational Training Center• College• University
Prerequisites	<ul style="list-style-type: none">• Writing code in any Object-Oriented Programming language (Python, C#, Java, etc.)• Fundamental skills of networking, equivalent of CCNA ITN
Course Delivery	Instructor-Led
Equipment	<ul style="list-style-type: none">• 2 Virtual Machines:<ul style="list-style-type: none">• Student's Lab VM bundled with all software tools• Cisco CSR1000v• Packet Tracer for Network Automation
Estimated Time to Complete	70h
Learning Domains	<ol style="list-style-type: none">1. Software Development and Design2. Understanding and Using APIs3. Cisco Platforms and Development4. Application Deployment and Security5. Infrastructure and Automation6. Network Fundamentals

DevNet Tools & Resources

developer.cisco.com

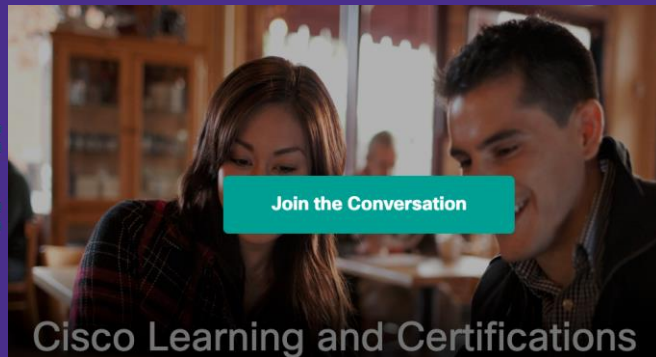
1. Video Courses
2. Learning Paths
3. Sandboxes
4. Sample Code (Github)
5. Communities

Cisco DNA Center Programmability

This learning track explores the programmability capabilities of Cisco DNA Center. You'll see how to leverage the APIs to explore what makes up, and who is on your network, gain a better understanding of how the network is operating with Cisco DNA Assurance, manage the software lifecycle, and so much more!

🔗 Cisco DNA Center, DevNet Express, NETCONF, YANG, RESTCONF, Networking, SDN

2 Modules	8 Labs	3 Hours
--------------	-----------	------------



Application Hosting and the Network

The switches and routers in your network can do much more than just pass packets these days. Edge or Fog computing is the next big thing, and in this module learn how Linux, Python and Containers can all be hosted at the edge.

Play module

- 1 Cloud to Fog: Why Host Apps in the Network
- 2 Linux at the Edge: Introduction to Guest Shell



DevNet Tools & Resources

DevNet Associate Fundamentals Course

Cisco network management

This token is then assigned to a header parameter named 'X-Auth-Token'.

Authenticate using Python

Package imports

There are some Python packages that we'll use to help us out. The 'sys' module provides system-specific functions such as `sys.exit()`. The 'json' module will help parse json structured requests and results. The 'requests' module provides a set of classes supporting HTTP requests, including the means to provide a username and password as credentials directly such that they are correctly encoded and included in the POST request header. Learn more about the 'requests' module, here:

<https://requests.readthedocs.io/en/master/>.

There are empty files that we'll add to in the `~/src/dna-center` directory, so get started with:

```
cd ~/src/dna-center
```

Then click on the file `~/src/dna-center/dna-center-authenticate.py`, to open it for edit, and use the copy widget to copy the following Python code into it.

Search ...

- src
 - aci
 - ansible
 - coding-basics
 - deployment
 - dna-center
 - dna-center-authenticate.py
 - get_network_devices_list.py
 - get_network_devices.py
 - meraki
 - nso
 - parsing
 - sample-app
 - sd-wan
 - security
 - unittest
 - webex-teams
- main.py
- .bash_profile

dna-center-authenticate.py

```
26 username (str): dnac user name
27 password (str): password
28
29 Return:
30 -----
31 str: dnac access token
32 ***
33
34 # Suppress credential warning for this exercise
35 requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
36
37 # Authentication API full URI
38 post_uri = "https://" + dnacip + "/dna/system/api/v1/auth/token"
39 print ("\nAuthenticate: POST %s"%(post_uri))
40
41 try:
42     # verify - set to False to tell requests to NOT verify server's TLS cert
43     # In production code this should be left to default to 'True'
44     r = requests.post(post_uri, auth=(username, password), verify=False)
45     return r.json()["Token"]
46 except:
47     # Something wrong, cannot get access token
48     print ("Status: %s"%r.status_code)
49     print ("Response: %s"%r.text)
50     sys.exit ()
51
52 # Authenticate to the Cisco DNA Center
53 # and obtain an authentication token
54 token = get_X_auth_token(dnacip, username, password)
55 print ("returned Authentication Token: ", (token))
56
```


Terminal

```
developer:src > cd ~/src/dna-center
developer:dna-center >
developer:dna-center > |
```

DevNet Tools & Resources

digital-learning.cisco.com

- Cisco Learning Platform
- Offers multiple DevNet courses
 - virtual labs to gain hands-on experience
- Courses about all other Cisco topics

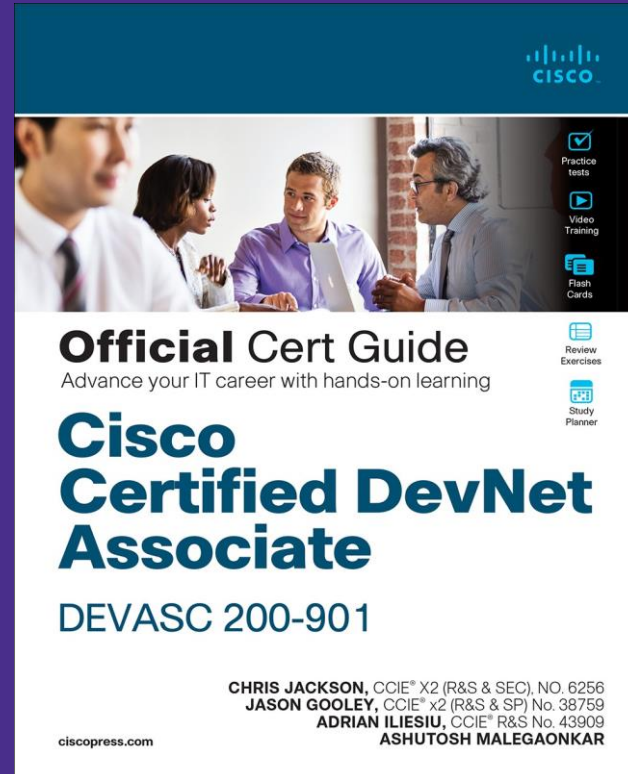
	Implementing Automation for Cisco Enterprise Solutions (ENAUJ) v1.0 Delivery Type: Cisco Training on Demand Courses Certification: CCNP, DevNet, Professional	★★★★★ 🕒 24hr 0min
	Developing Applications Using Cisco Core Platforms and APIs (DEVCOR) v1.0 Delivery Type: Cisco Training on Demand Courses Certification: DevNet, Professional	★★★★★ 🕒 40hr 0min
	Implementing Automation for Cisco Data Center Solutions (DCAUI) v1.0 Delivery Type: Cisco Training on Demand Courses Technology: Data Center Certification: CCNP, Data Center, DevNet, Professional	★★★★★ 🕒 24hr 0min
	Introducing Automation for Cisco Solutions (CSAU) v1.0 Delivery Type: Cisco Training on Demand Courses Certification: CCNP, Data Center	★★★★★ 🕒 16hr 0min
	Developing Applications and Automating Workflows using Cisco Core Platforms (DEVASC) v1.0 Delivery Type: Cisco Training on Demand Courses Certification: DevNet, Associate	★★★★★ 🕒 64hr 0min



DevNet Tools & Resources

Cisco Press “Official Cert Guide”

- Release: 20th July 2020
- Usually very detailed & In-depth



DevNet Tools & Resources



- Cisco Developer Website
<https://developer.cisco.com>
- Cisco Press Official Cert Guide
<https://www.ciscopress.com/store/cisco-certified-devnet-associate-devasc-200-901-official-9780136677338>
- Cisco Digital Learning Library
<https://digital-learning.cisco.com/>
- Cisco Developer Website: DevNet ASC Fundamentals Training
<https://developer.cisco.com/certification/fundamentals/>





By Susie Wee, SVP/GM Cisco DevNet & CX Ecosystem Success

Get Cisco certified and Cisco DevNet certified today with new online testing





Model Driven Programmability

Netconf

Restconf

gRPC





Configuration monitoring and compliance

NetOps

Gabi Zapodeanu

Technical Marketing Engineer

Business challenge

- The Challenge
 - 75% of network outages are due to user errors
 - Configuration drifting
- The Goal
 - Automated rollback of non-compliant changes
 - Alert on all network configuration changes
- The Solution
 - Integration between DNA Center, ServiceNow, Cisco IOS XE, and Webex Teams
- The Results
 - Non-compliant configuration changes are mitigated in minutes
 - Real Time view of any device configuration changes

Proposed Solution

NetOps App

```
#!/usr/bin/perl
# This script will monitor device configuration changes. It could be executed on demand as in this lab,
# periodically (every 30 minutes, for example) or continuously.
# It will collect the configuration file for each Net Center managed device, compare with the existing cached file,
# and detect if any changes.
# When changes detected, identify the last user that configured the device, compare with the existing cached file,
# and detect if any changes.
# Automatically create new JIRA non-compliance configurations, or save new configurations if approved in ServiceNow.
# Send Email alerts to devices using PuTTY.
# Configuration changes on this file:
# - No Access Control Lists changes
# - No logging changes
# - No duplicated IPV6 addresses
# ...

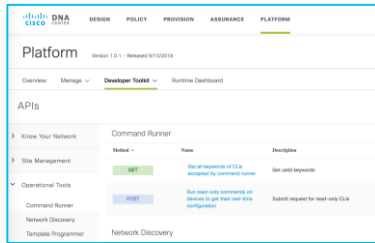
# get the DNA C such token
dna_c_token = dna_c_get_oauth_token($ENV{DNA_C_TOKEN})
print "token: ", dna_c_token, "\n"

temp_run_config = "temp_run_config.txt"

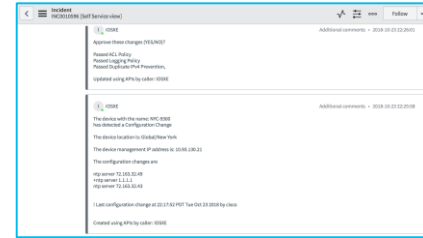
# get the DNA C managed devices list (excluded wireless, for one location)
my @devices_list = dna_c_get_managed_devices($ENV{DNA_C_TOKEN})
my @services_instances = {}
my @device = @DNA_C_managed_devices
if ($device{hostname} ne "hostname" and $device{hostname} ne "hostname") {
    print "Device: ", $device{hostname}, "\n"
    $services_instances{$device{hostname}} = $device{hostname}
}

my @services_instances = @services_instances{hostname} {}
```

DNA Center



ServiceNow

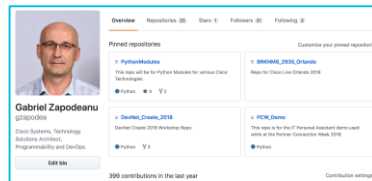


Guest Shell

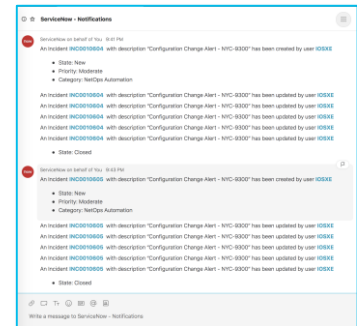


Open IOS XE, Guest Shell

GitHub



Webex Teams

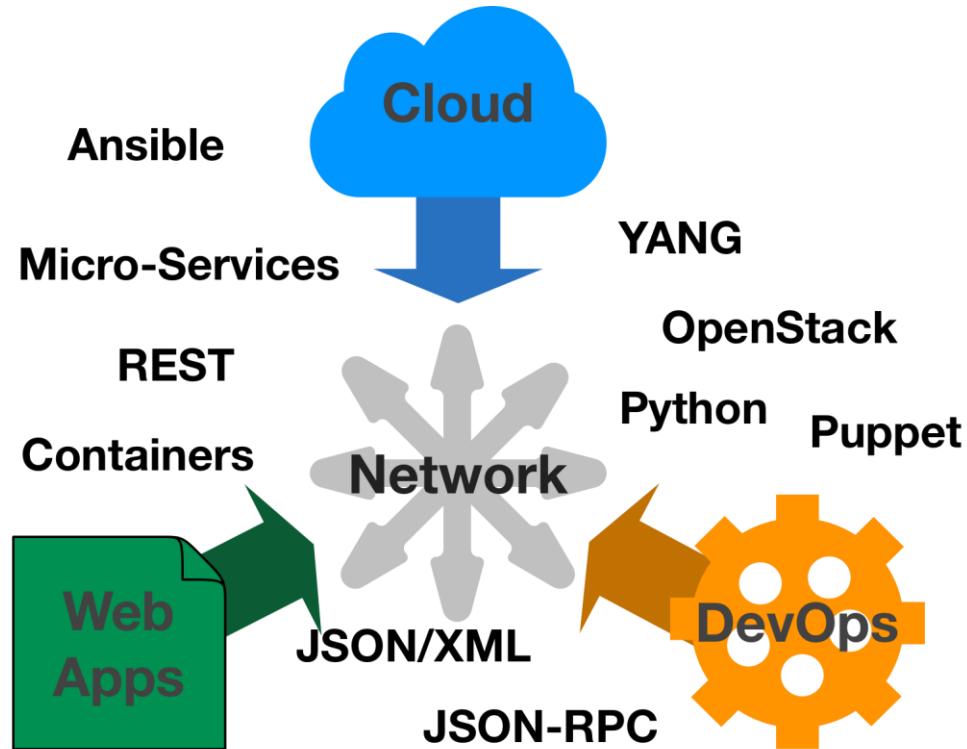


Further details



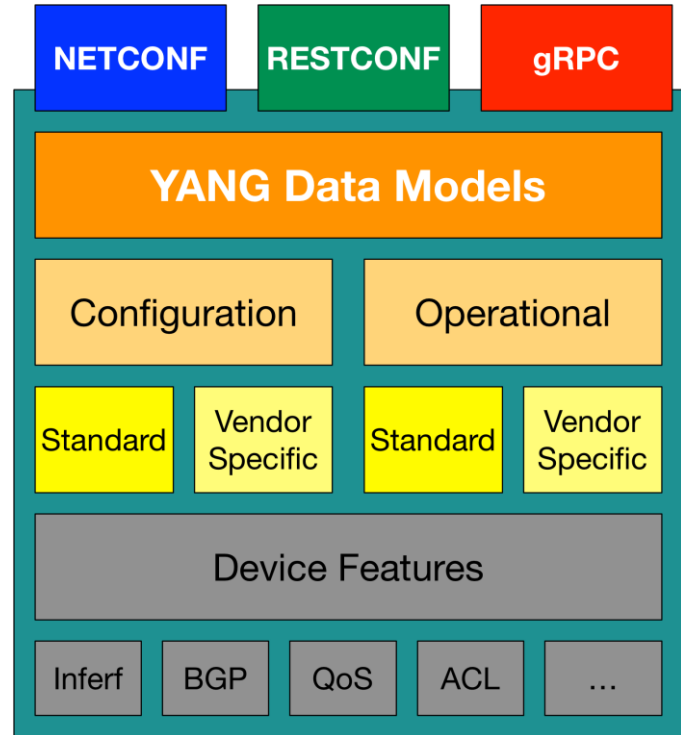
[Here](#)

The Network is No Longer Isolated



Model Driven Programmability

- NETCONF – 2006 – RFC 4741
(RFC 6241 in 2011)
- YANG – 2010 – RFC 6020
- RESTCONF – 2017 – RFC 8040
- gRPC – 2015 – OpenSource project by Google



Transport (Protocol) vs Data (Model)

TCP/IP Network Frame Format



- NETCONF
- RESTCONF
- gRPC

- YANG

Transport mechanisms

NETCONF

- SSH
- RPC
 - <get-config>
 - <edit-config>
 - <commit>
 - <lock>
- ...

RESTCONF

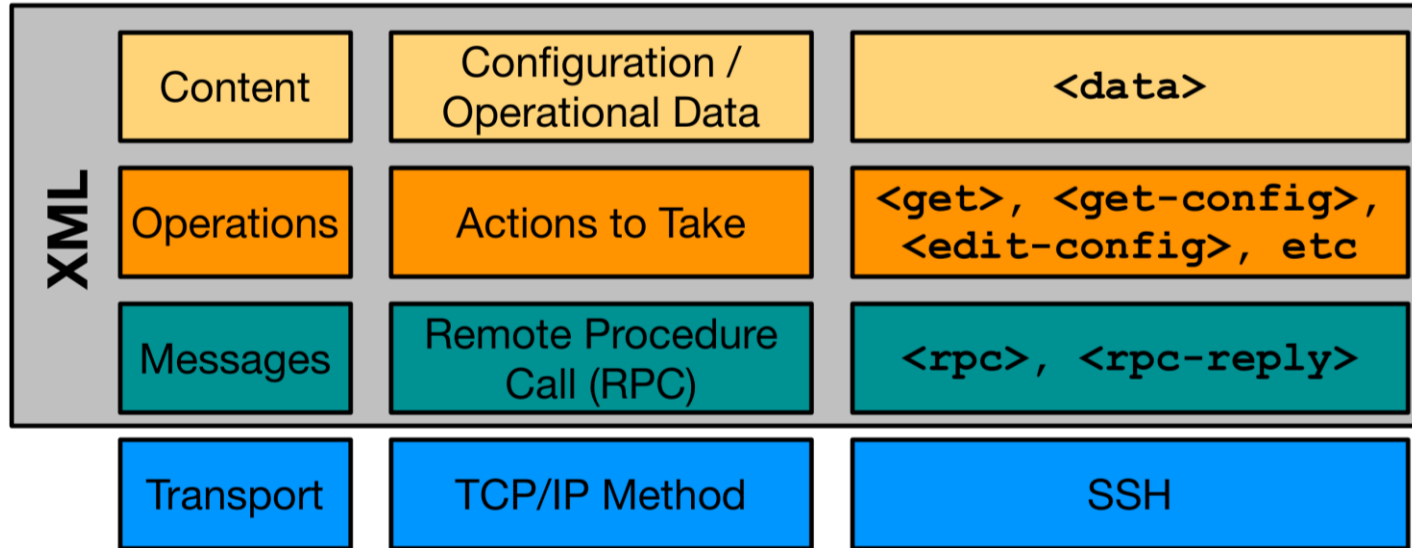
- HTTP
- Methods
 - GET
 - POST
 - DELETE
 - PUT
- ...

gRPC/gNMI

- HTTP/2
- RPC (user defined)
 - Capabilities
 - Get
 - Set
 - Subscribe

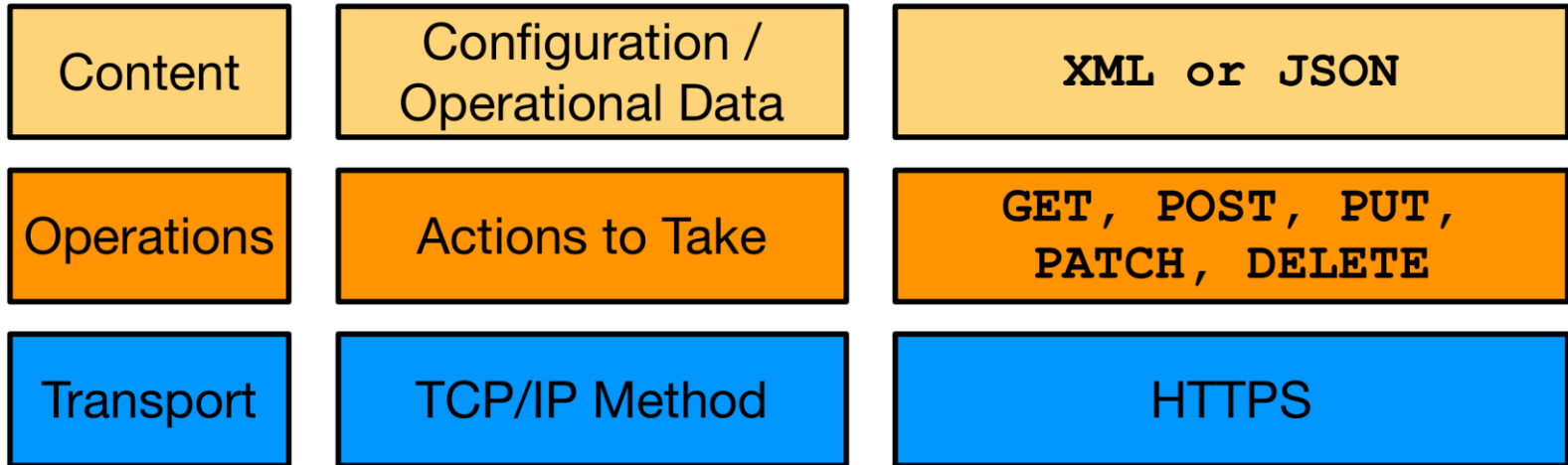


NETCONF Protocol Stack



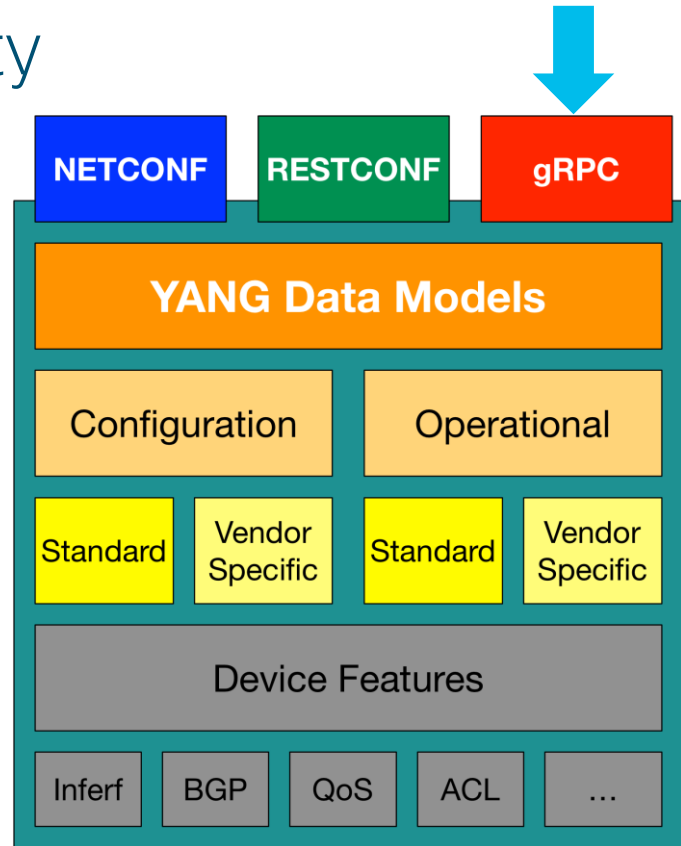
RESTCONF Protocol Stack & Transport

RESTCONF Protocol Stack



Model Driven Programmability

- NETCONF – 2006 – RFC 4741
(RFC 6241 in 2011)
- YANG – 2010 – RFC 6020
- RESTCONF – 2017 – RFC 8040
- gRPC – 2015 – OpenSource project by Google



Transport mechanisms

NETCONF

- SSH
- RPC
 - <get-config>
 - <edit-config>
 - <commit>
 - <lock>
- ...

RESTCONF

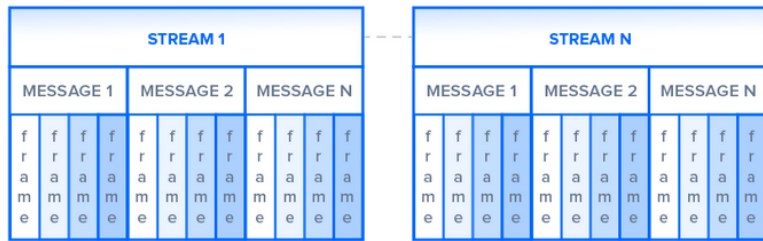
- HTTP
- Methods
 - GET
 - POST
 - DELETE
 - PUT
- ...

gRPC/gNMI

- HTTP/2
- RPC (user defined)
 - Capabilities
 - Get
 - Set
 - Subscribe

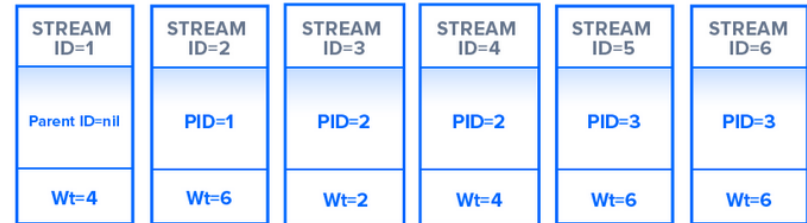
HTTP2: Something like Multiplexing and QOS...

Connection



Within this connection there are multiple *streams* of data. Each stream consists of multiple messages in the familiar request/response format. Finally, each of these messages split into smaller units called *frames*. The binary framing layer organizes messages into parallel streams of data.

Channel



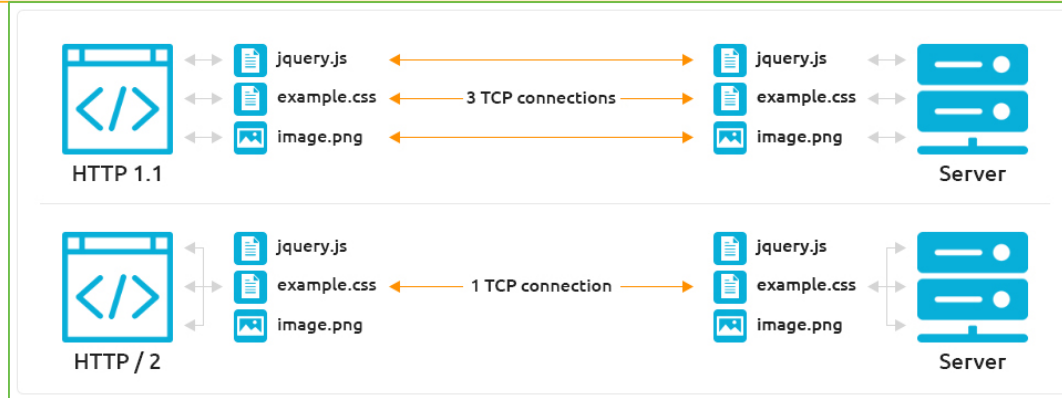
When a client sends concurrent requests to a server, it can prioritize the responses it is requesting by assigning a weight between 1 and 256 to each stream. The higher number indicates higher priority. In addition to this, the client also states each stream's dependency on another stream by specifying the ID of the stream on which it depends.

So, what I understood

Since multiplexing allows the client to construct multiple streams in parallel, these streams only need to make use of a single TCP connection. Having a single persistent connection per origin improves upon HTTP/1.1 by reducing the memory and processing footprint throughout the network.

→ This results in better network and bandwidth utilization and thus decreases the overall operational cost.

A single TCP connection also improves the performance of the HTTPS protocol, since the client and server can reuse the same secured session for multiple requests/responses.



<https://coolicehost.com/http2-protocol.html>

gRPC benchmark

Procession time

RPC	# of requests	# of clients	total time	per-request time
jsonrpc	300,000	1	8m7.270s	1.624ms
gRPC	300,000	1	36.715s	122.383μs
gRPC	300,000	100	7.167s	23.892μs

Memory allocation

RPC	jsonrpc	gRPC	delta
NsPerOp	487271046903	36716116701	-92.46%
AllocsPerOp	32747687	25221256	-22.98%
AllocedBytesPerOp	3182814152	1795122672	-43.60%

<https://blog.gopheracademy.com/advent-2015/etcd-distributed-key-value-store-with-grpc-http2/>

CiscoLive!

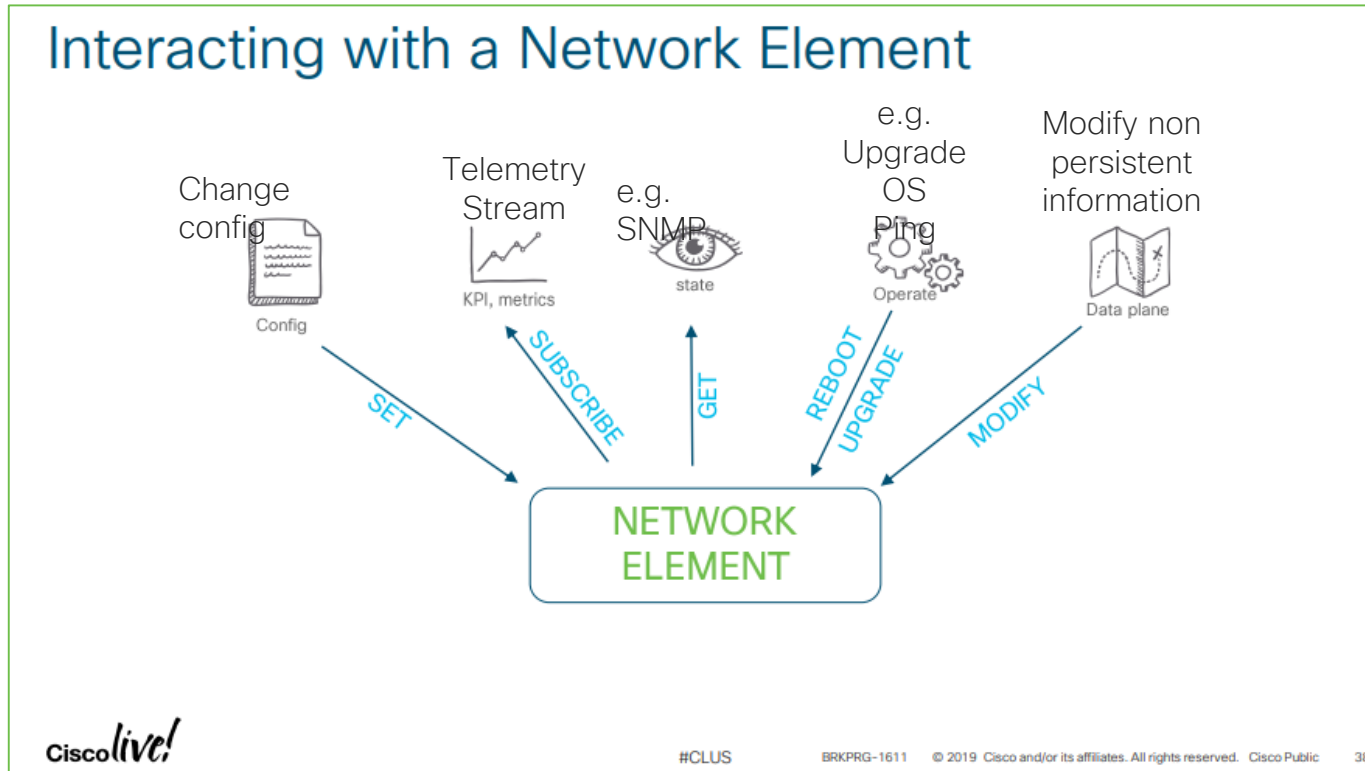
#CLUS

BRKPRG-1611

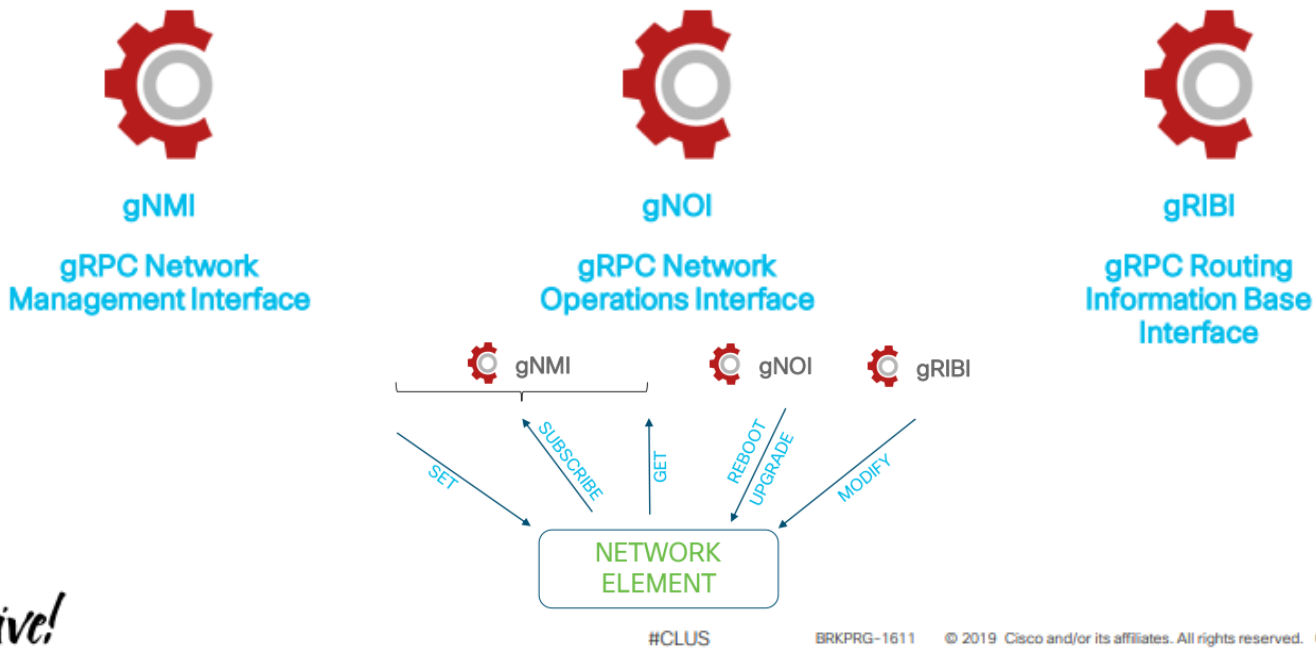
© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Public

36

What can we use gRPC for?



All those Interactions are represented on OpenConfig gRPC interfaces



CiscoLive!

gRPC service interface definitions

gNMI

```
service gNMI {  
  rpc Capabilities(CapabilityRequest) returns (CapabilityResponse);  
  rpc Get(GetRequest) returns (GetResponse);  
  rpc Set(SetRequest) returns (SetResponse);  
  rpc Subscribe(stream SubscribeRequest) returns (stream SubscribeResponse);  
}
```

gNOI

```
service System {  
  rpc Ping(PingRequest) returns (stream PingResponse) {}  
  rpc Traceroute(TracerouteRequest) returns (stream TracerouteResponse) {}  
  rpc Time(TimeRequest) returns (TimeResponse) {}  
  rpc SetPackage(stream SetPackageRequest) returns (SetPackageResponse) {}  
  ...  
}
```

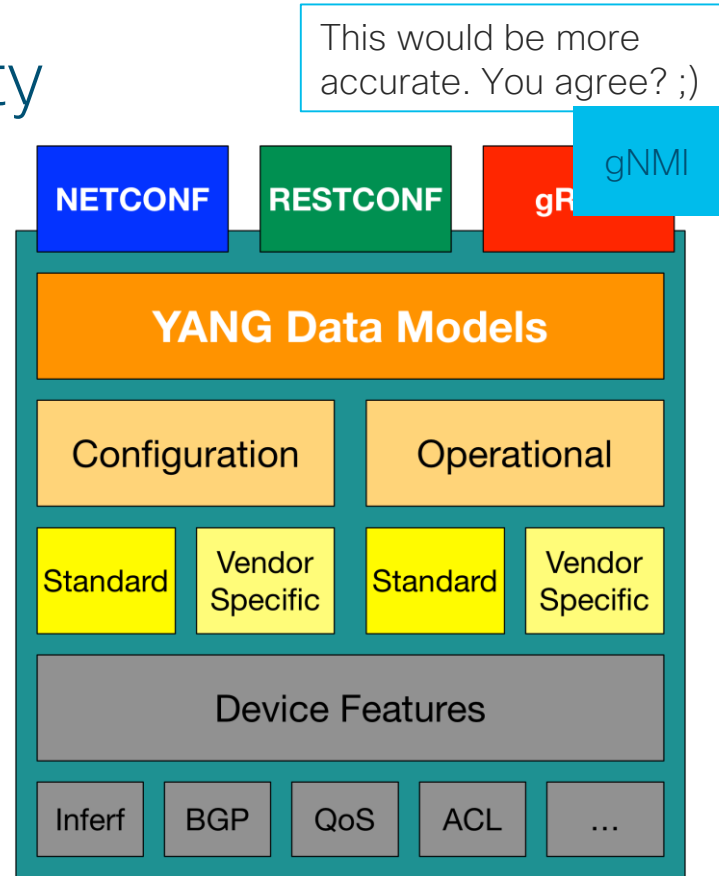
Cisco

```
service gRPCConfigOper {  
  rpc GetConfig(ConfigGetArgs) returns(stream  
  ConfigGetReply) {};  
  rpc MergeConfig(ConfigArgs)  
  returns(ConfigReply) {};  
  ...  
  rpc CreateSubs(CreateSubsArgs) returns(stream  
  CreateSubsReply) {};  
}  
  
service gRPCExec {  
  ...  
  rpc ActionJSON(ActionJSONArgs) returns(stream  
  ActionJSONReply) {};  
}
```

Goal is to make all services available across all platforms in one method over all vendors e.g. Cisco, HP etc.

Model Driven Programmability

- NETCONF – 2006 – RFC 4741
(RFC 6241 in 2011)
- YANG – 2010 – RFC 6020
- RESTCONF – 2017 – RFC 8040
- gRPC – 2015 – OpenSource project by Google





Q&A 😊





CISCO

DEVNET

