



DEVNET

How to be a Network Engineer in a Programmable Age

An evolution that goes beyond Infrastructure as Code and Automation

Hank Preston, Principal Engineer
NetDevOps Evangelist
ccie 38336 R/S

 @hfpreston
github.com/hpreston

Topics to Cover

- The Network Engineer of Old
- The Four Ages of Networking
- Cloud to the Rescue
- Enter NetDevOps
- Today's Network Engineer

The Network Engineer Evolves



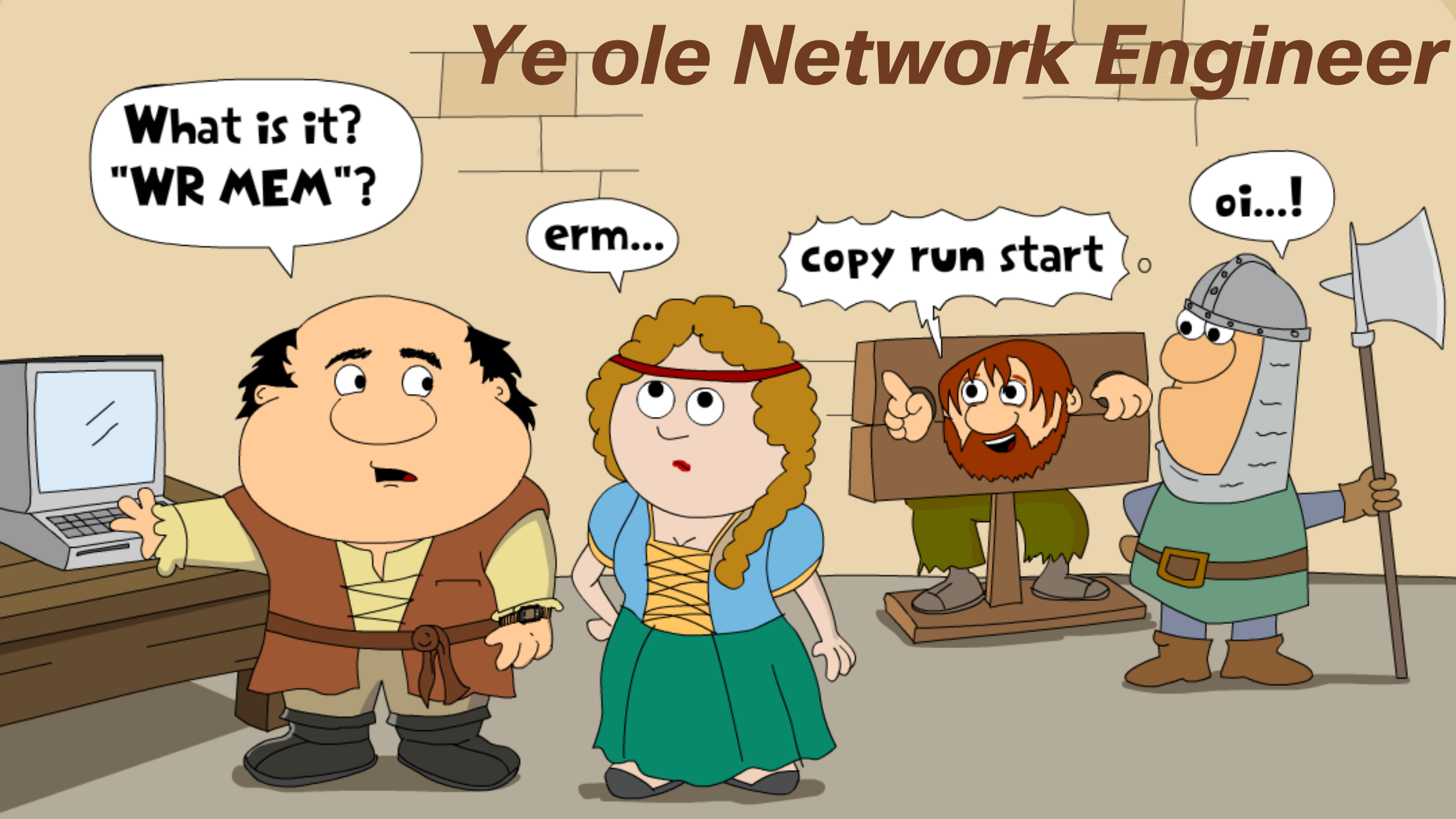
Ye ole Network Engineer

What is it?
"WR MEM"?

erm...

copy run start

oi...!



Meet Carl the Network Engineer



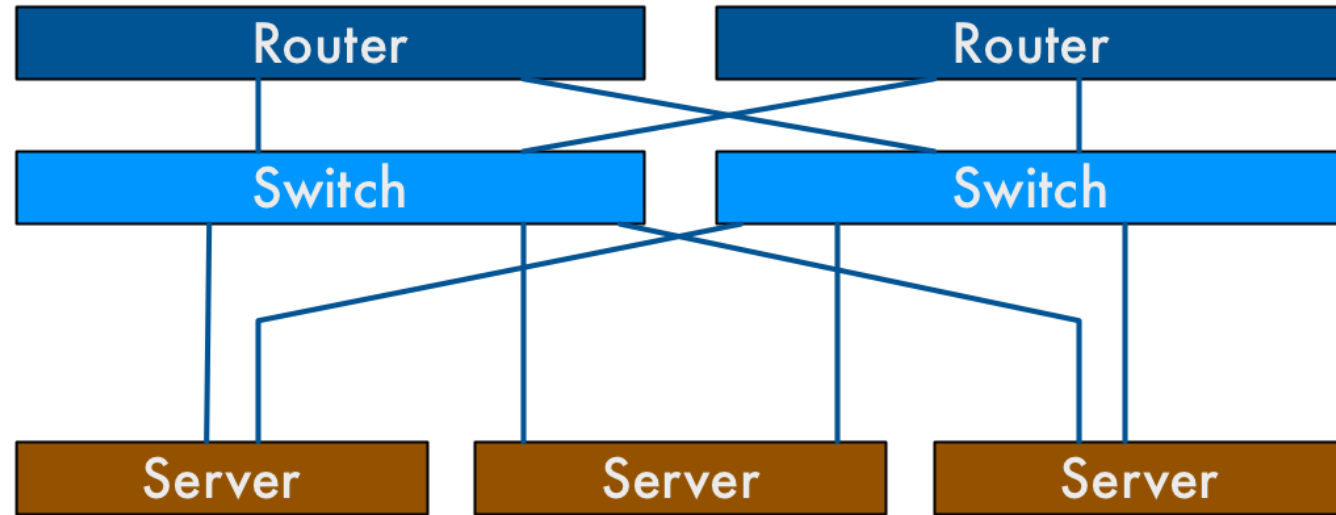
Network Skills

- Spanning-Tree
- Routing Protocols
- QoS
- VPN Design
- Spanning-Tree
- VOIP
- Fibre Channel
- Security Policy
- MPLS
- Did we mention Spanning-Tree?

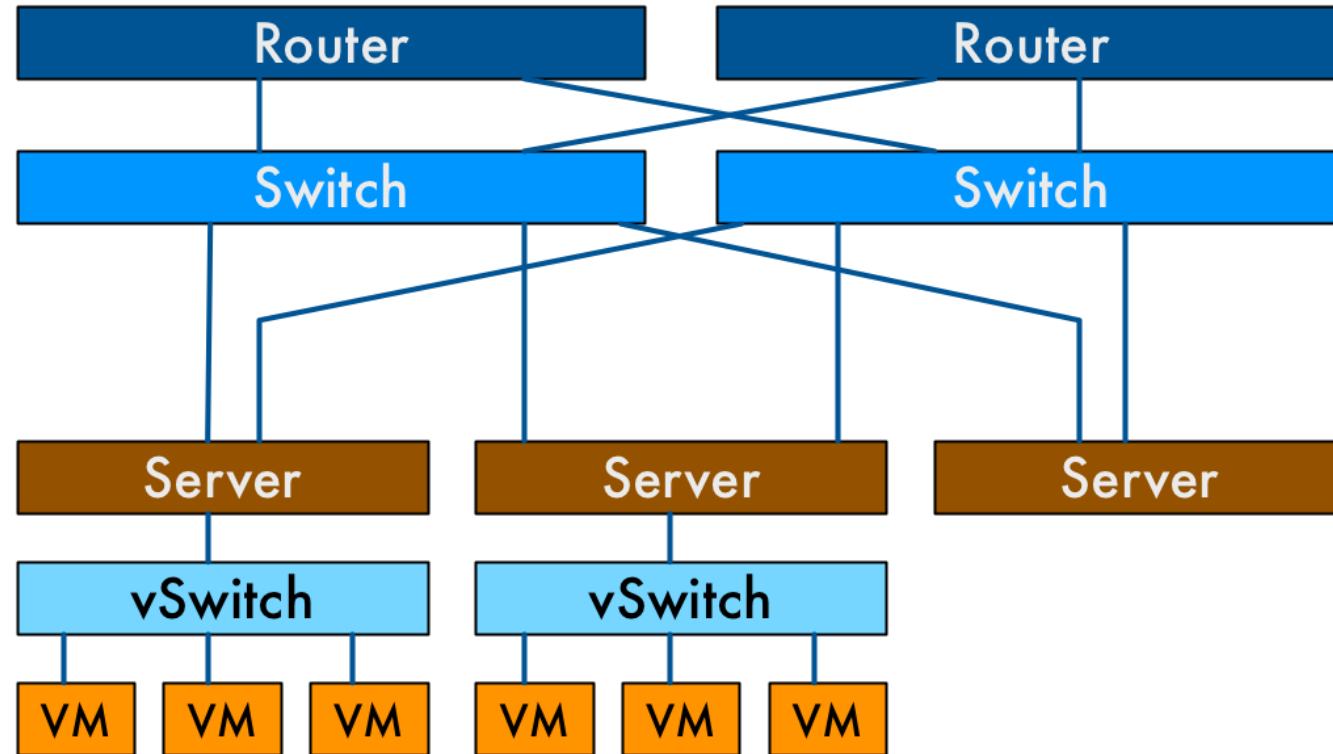
Programming Skills

- TCL
- EEM
- Expect Scripts

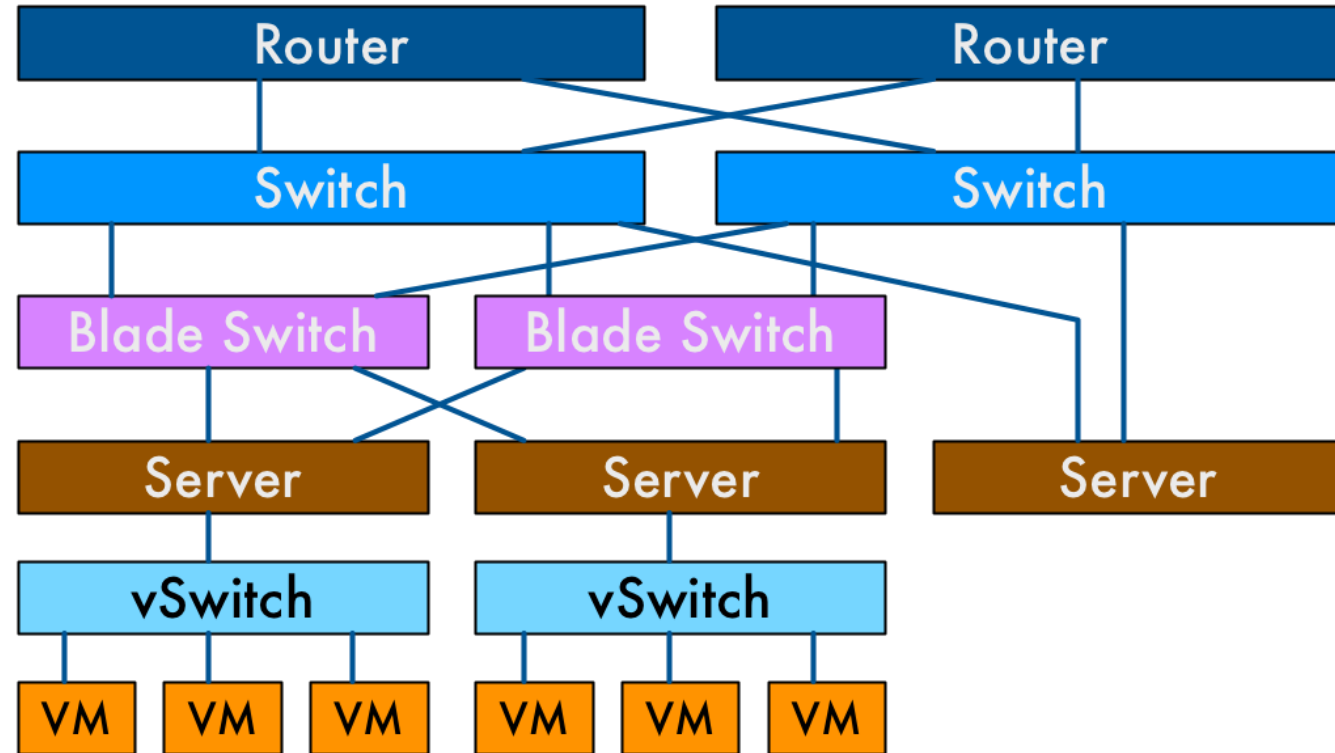
The Network...



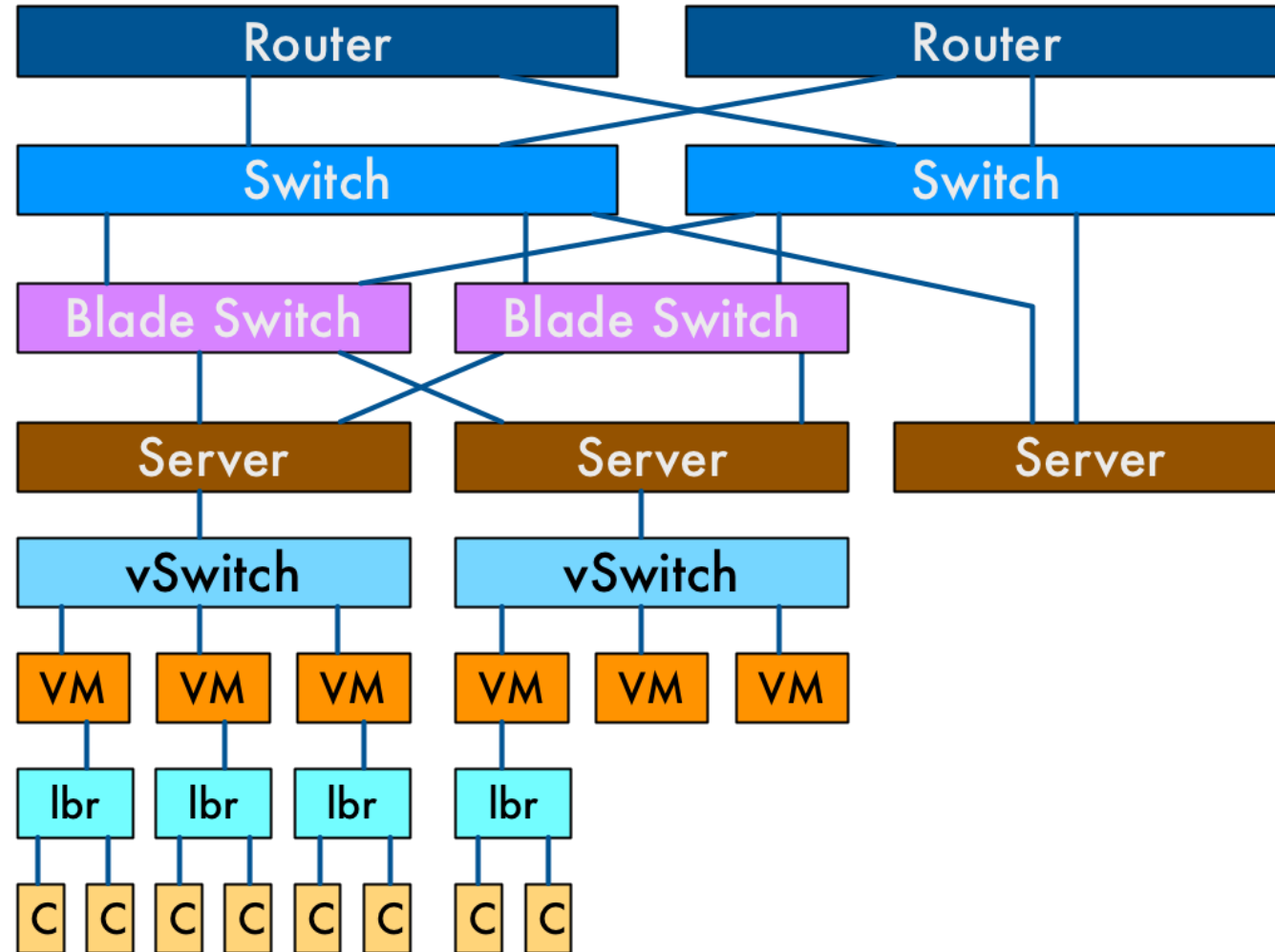
The Network...



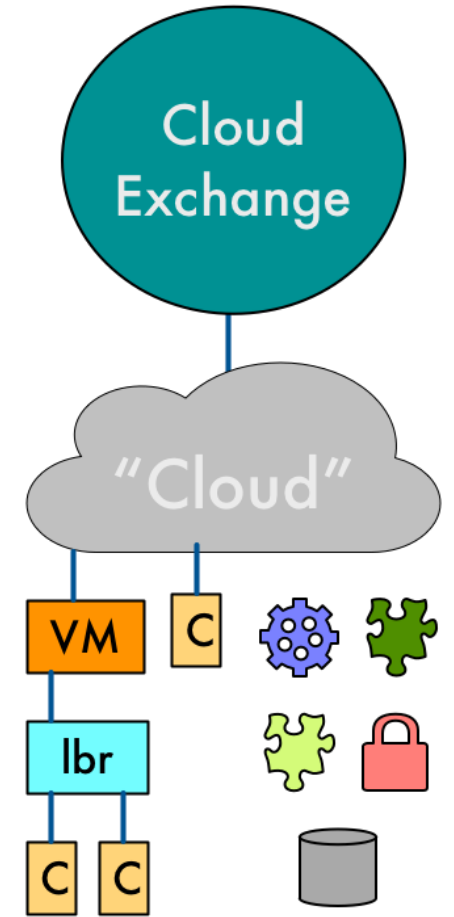
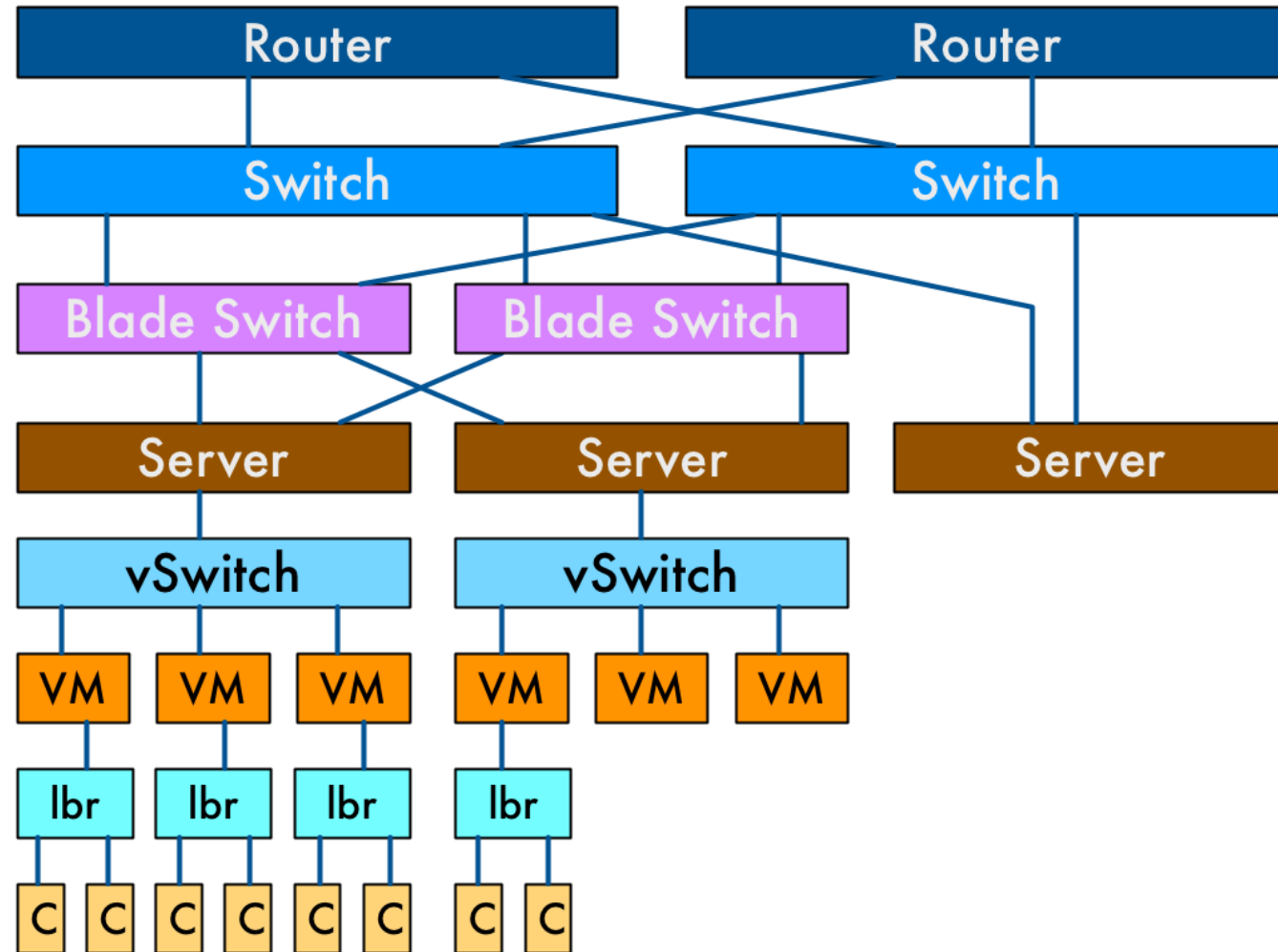
The Network...



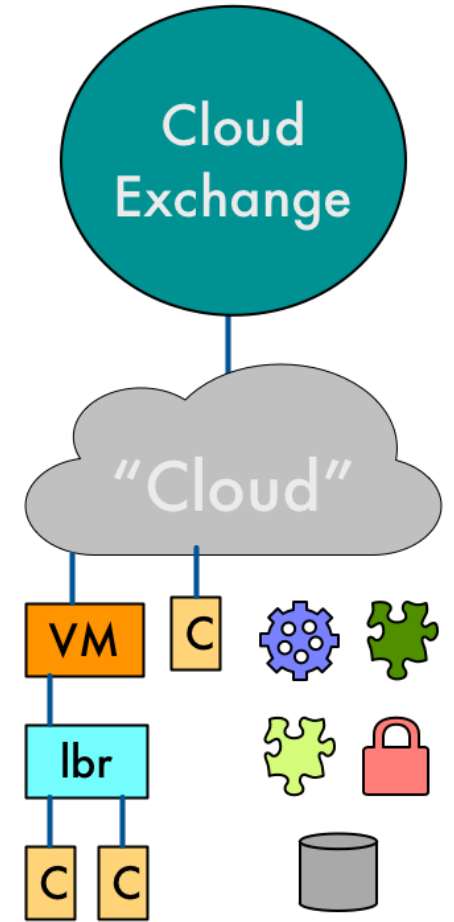
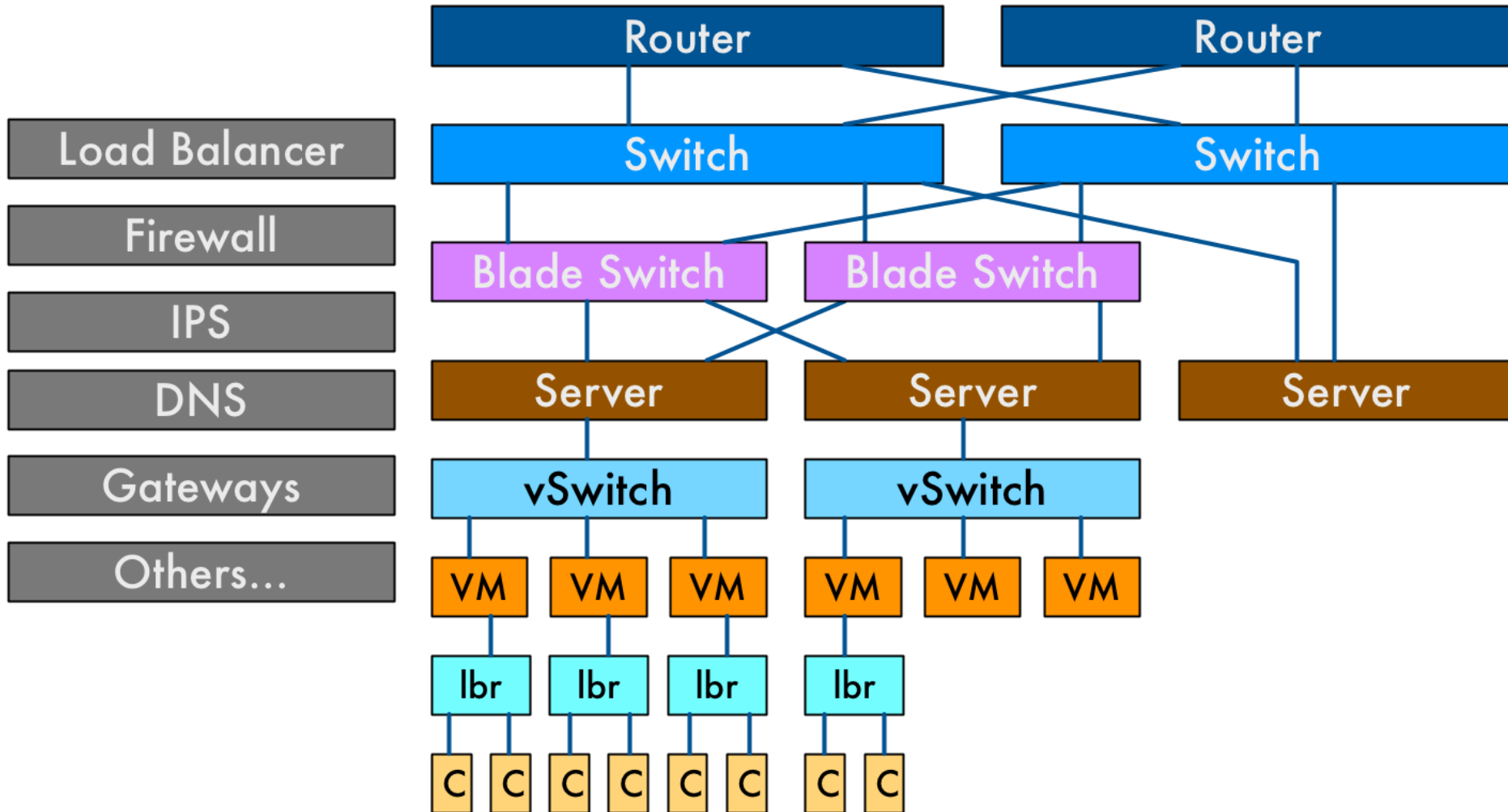
The Network...



The Network...



The Network...



The OSI Model of Networking...

Please don't ask about this...

L7: Application

L6: Presentation

L5: Session



L4: Transport

L3: Network

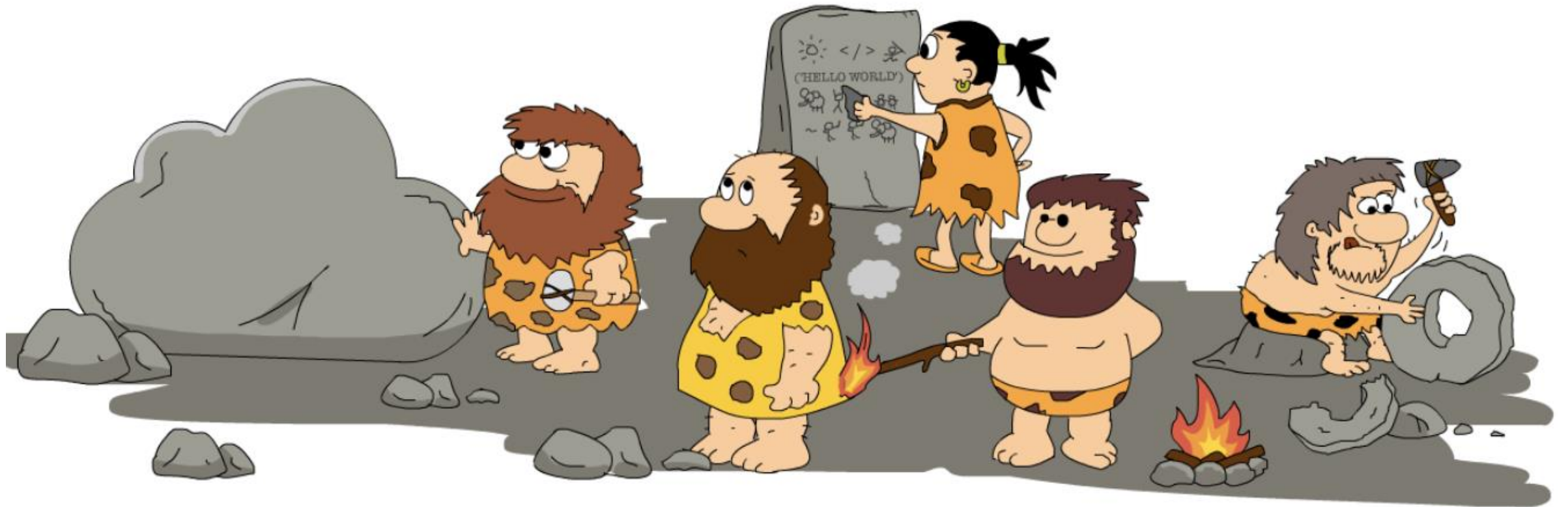
L2: Data Link

L1: Physical

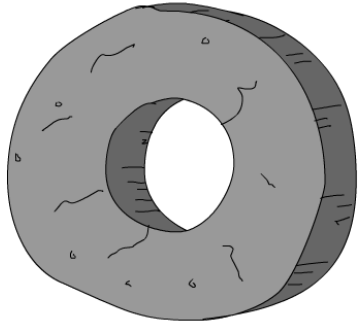
Oh Yeah... We Got this

Black Magic





Networking through the ages...

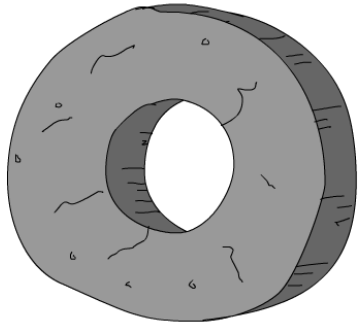


Stone Age

Spanning Tree

VLANs

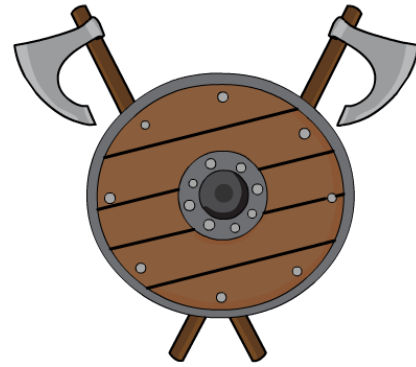
The Four Ages of Networking.....



Stone Age

Spanning Tree

VLANs



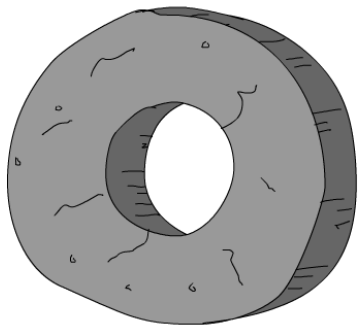
Bronze Age

Routing Protocols

WAN Design

IP-magedon

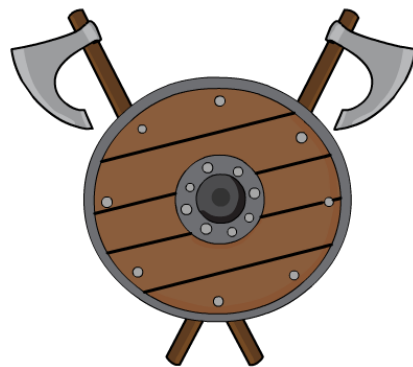
The Four Ages of Networking.....



Stone Age

Spanning Tree

VLANs



Bronze Age

Routing Protocols

WAN Design

IP-magedon



The Renaissance

SDN

OpenFlow

Controllers

Overlays

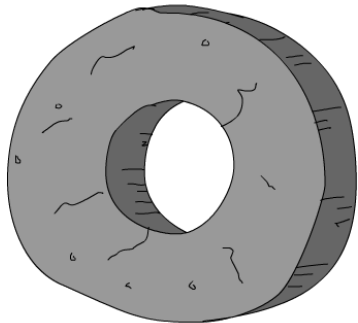
MP-BGP

VXLAN

Micro-Segmentation

White Box

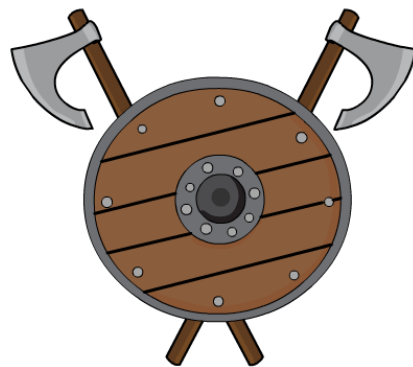
The Four Ages of Networking.....



Stone Age

Spanning Tree

VLANs



Bronze Age

Routing Protocols

WAN Design

IP-magedon



The Renaissance

SDN

OpenFlow

Controllers

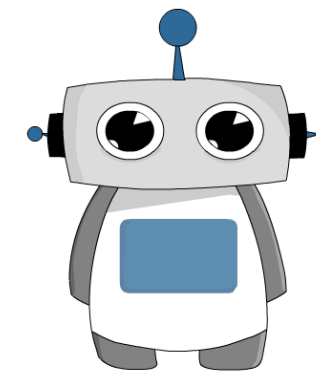
Overlays

MP-BGP

VXLAN

Micro-Segmentation

White Box



Programmable Age

Cloud

Python

REST / APIs

NETCONF / YANG

“Fabrics”

Network Function
Virtualization (NFV)

Containers

DevOps

NetDevOps!

The Four Ages of Networking.....

”Digitization” of the Enterprise



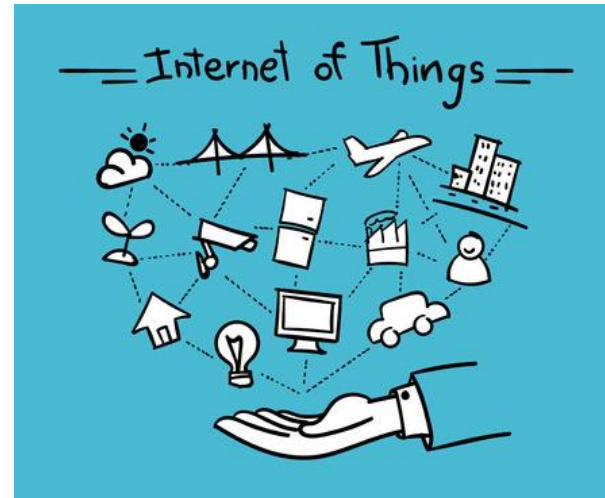
I want an agile bimodal hybrid cloud so we can develop containerised serverless trustless microservices applications to take us Digital to avoid disruption from any unicorns. Oh.. and I want DevOps...Two of those...

App Economy



User Expectations and
Agility

Internet of Things



If it isn't connected, don't
bother...

Tech Unicorns



Low barrier of entry for
disruptors

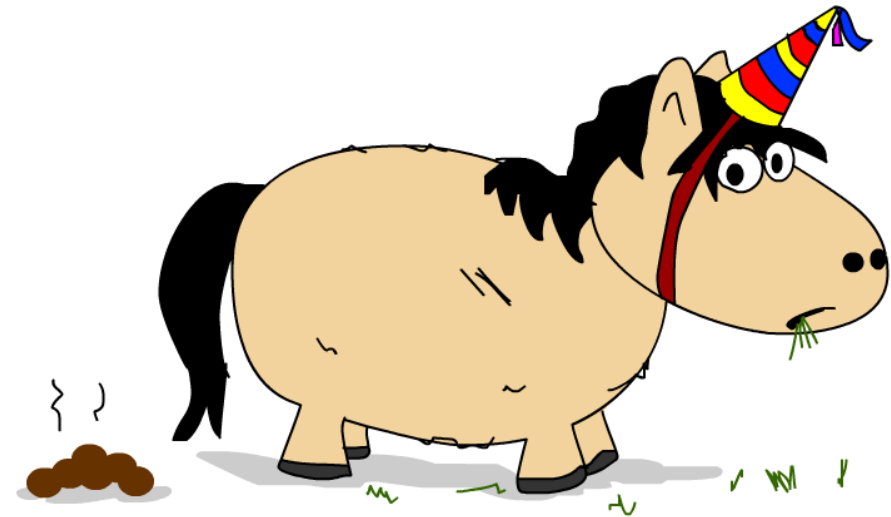
CLOUD



The Cloud You Plan to Build 😊



The Cloud You Plan to
Build 😊



The Cloud You End
Up With ☹️

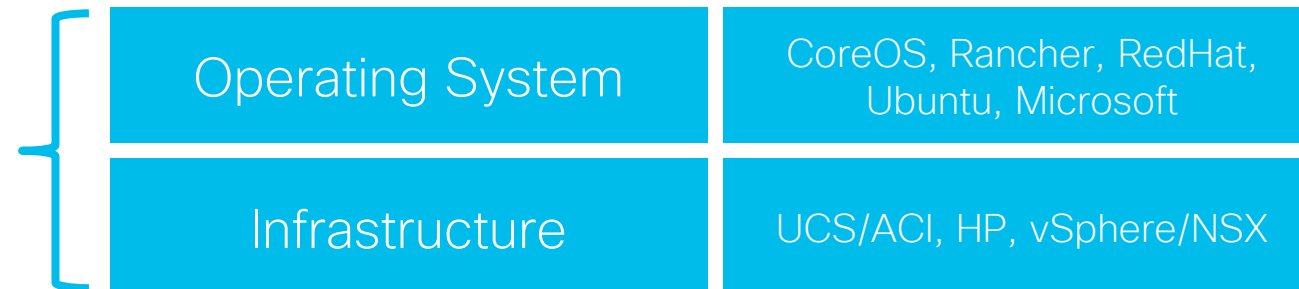
The New Infrastructure Stack



Users and
Developers



Architects and
Operators



The New Infrastructure Stack



Users and
Developers

Development Environment	Vagrant, Docker, Vim, Slack, Spark, Git
Delivery Pipeline	GitHub, BitBucket, Jenkins, Team City, Drone, Puppet, Ansible, Chef



Architects and
Operators

Cloud Management and Automation	UCS Director, vRealize, OpenStack, AWS, CloudCenter
Operating System	CoreOS, Rancher, RedHat, Ubuntu, Microsoft
Infrastructure	UCS/ACI, HP, vSphere/NSX

The New Infrastructure Stack



Users and Developers

Users and Developers	Development Environment	Vagrant, Docker, Vim, Slack, Spark, Git
	Delivery Pipeline	GitHub, BitBucket, Jenkins, Team City, Drone, Puppet, Ansible, Chef
	Scheduling and Placement	Docker/Swarm, Kubernetes, Mesosphere, Tectonic, Rancher, Rocket
	Container Layer	
	Applications and Middleware	HAProxy, Cassandra, RabbitMQ, Hadoop, Consul
Architects and Operators	Cloud Management and Automation	UCS Director, vRealize, OpenStack, AWS, CloudCenter
	Operating System	CoreOS, Rancher, RedHat, Ubuntu, Microsoft
	Infrastructure	UCS/ACI, HP, vSphere/NSX



Architects and Operators

The New Infrastructure Stack



Users and
Developers



DevOps Engineers



Architects and
Operators

Development Environment	Vagrant, Docker, Vim, Slack, Spark, Git
Delivery Pipeline	GitHub, BitBucket, Jenkins, Team City, Drone, Puppet, Ansible, Chef
Scheduling and Placement	Docker/Swarm, Kubernetes, Mesosphere, Tectonic, Rancher, Rocket
Container Layer	
Applications and Middleware	HAProxy, Cassandra, RabbitMQ, Hadoop, Consul
Cloud Management and Automation	UCS Director, vRealize, OpenStack, AWS, CloudCenter
Operating System	CoreOS, Rancher, RedHat, Ubuntu, Microsoft
Infrastructure	UCS/ACI, HP, vSphere/NSX

The New Infrastructure Stack



Users and Developers



DevOps Engineers



Architects and Operators

Users and Developers	Development Environment	Vagrant, Docker, Vim, Slack, Spark, Git	PaaS
	Delivery Pipeline	GitHub, BitBucket, Jenkins, Team City, Drone, Puppet, Ansible, Chef	
	Scheduling and Placement	Docker/Swarm, Kubernetes, Mesosphere, Tectonic, Rancher, Rocket	
Container Layer			
DevOps Engineers	Applications and Middleware	HAProxy, Cassandra, RabbitMQ, Hadoop, Consul	
	Cloud Management and Automation	UCS Director, vRealize, OpenStack, AWS, CloudCenter	
Architects and Operators	Operating System	CoreOS, Rancher, RedHat, Ubuntu, Microsoft	IaaS
	Infrastructure	UCS/ACI, HP, vSphere/NSX	

Network Stakeholders



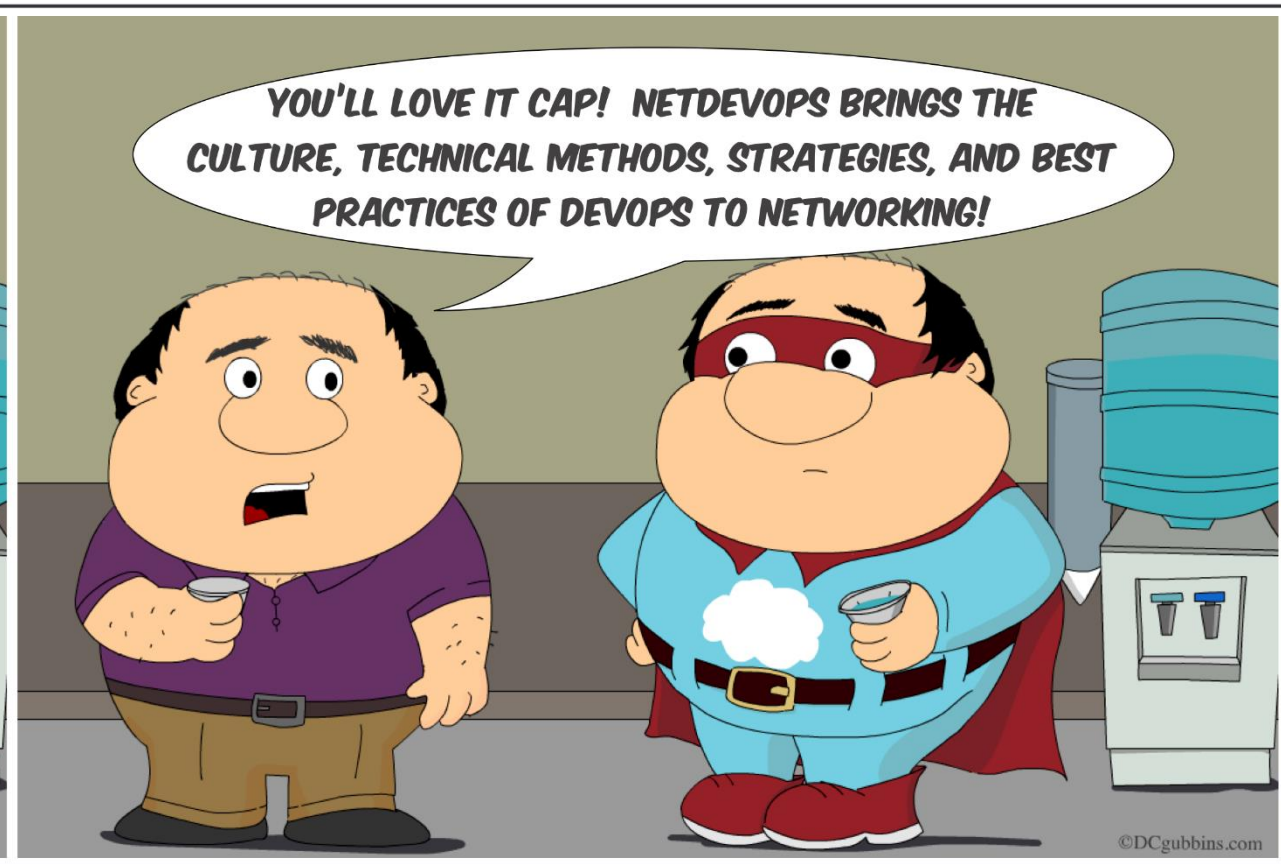
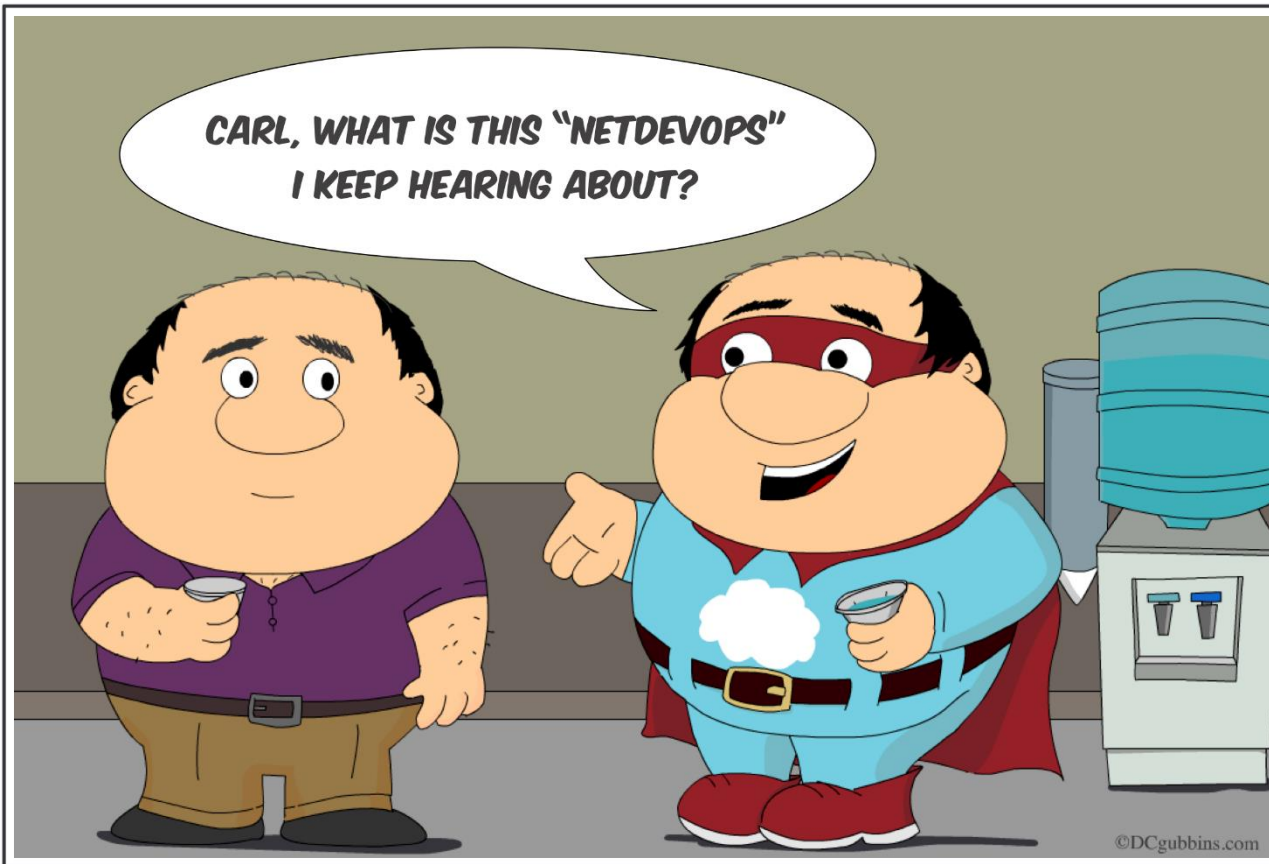
Network Builders

- Traditional networking teams
- Design, Build, and Maintain the Network
- Responsible for Care and Feeding



Network Consumers

- The users of the network
- Looking to consume network “services”
- The network is a “utility” – It should just work

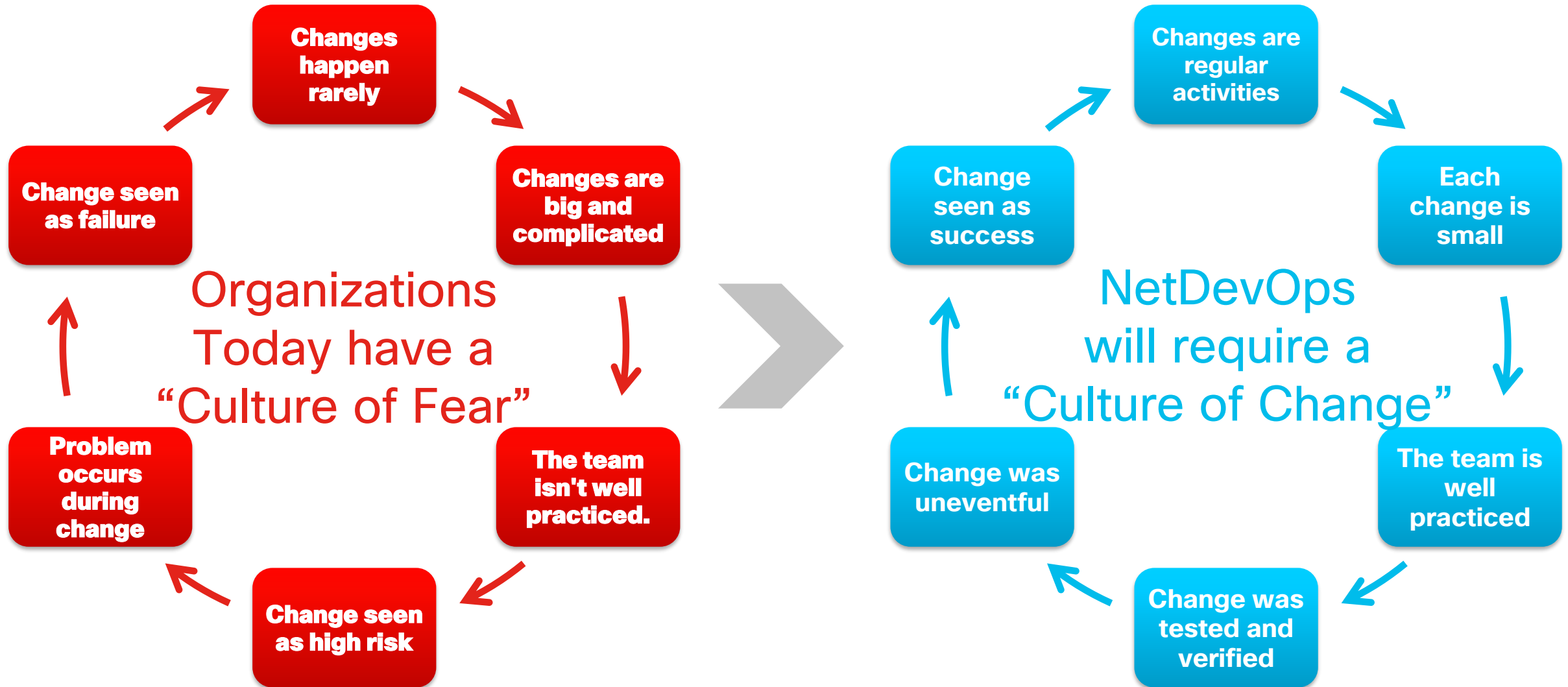


Enter NetDevOps!

Moving to a NetDevOps Culture and Mindset

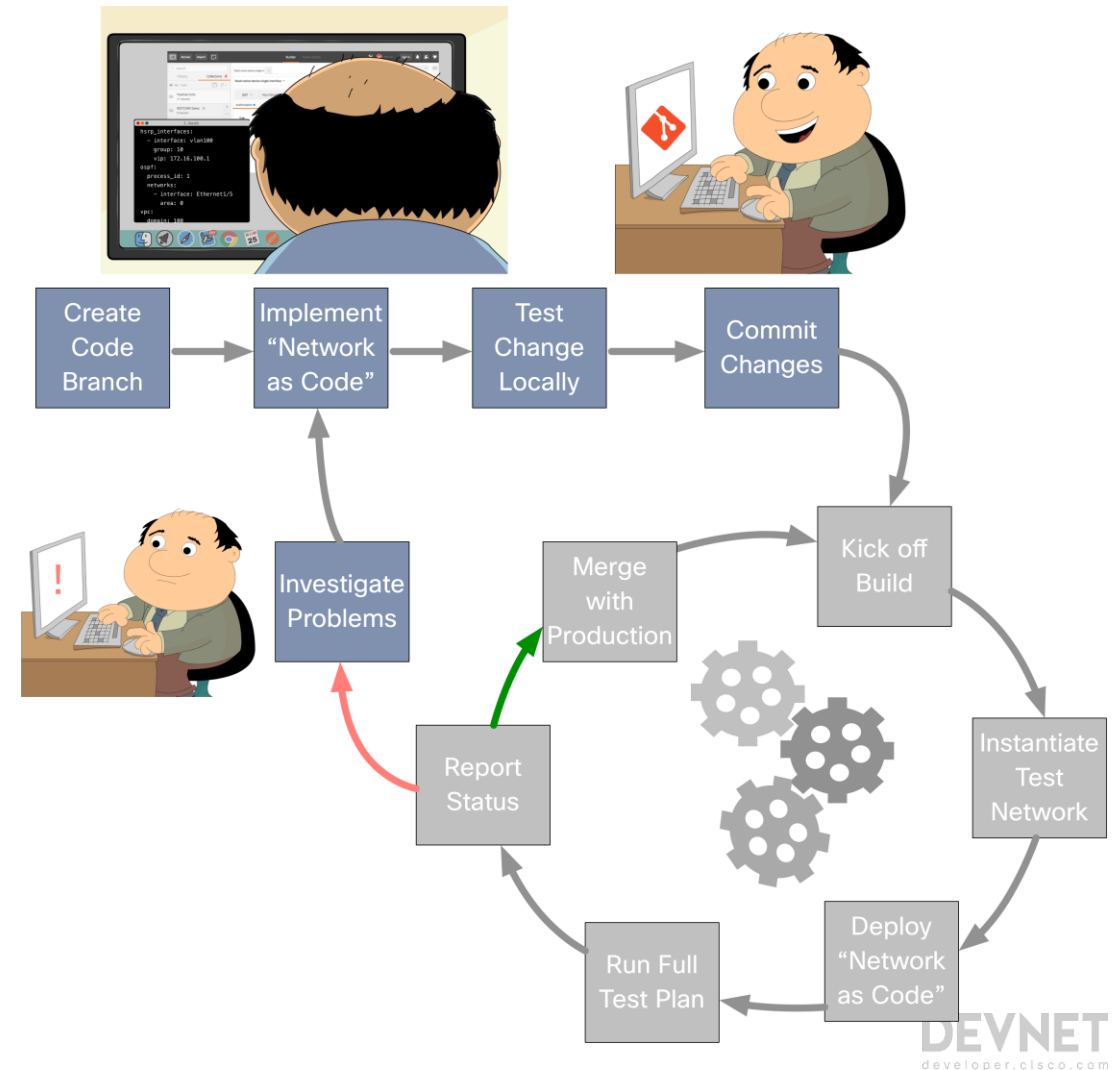


Moving to a NetDevOps Culture and Mindset



NetDevOps Pipeline: “Treating the Network as Code”

- Network Configuration stored in Source Control
- Changes are proposed in code “branches”
- CI/CD Build Servers deploy and test proposed configurations
- Successful configurations automatically deployed to “Production”



The NetDevOps Engineers Tool Bag

(Example tools, not comprehensive)

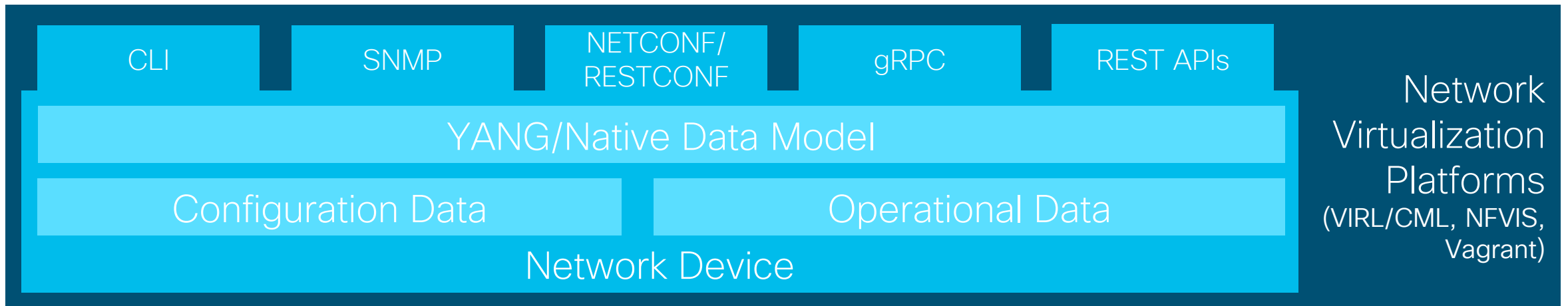
Distributed Source Control
(git, Subversion, Mercurial, GitHub, BitBucket, GitLab)

Build Server
(GitLab, Jenkins, Team City, Drone)

Configuration Management
(Ansible, Puppet, NSO, NAPALM, DIY)

Network Test Tooling
(pyATS, TRex, Robot, Behave)

Telemetry & Monitoring
(ELK, Grafana, Pipeline, UTM)



Development Environment
(Vagrant, NSO, VIRL/CML)

Test Environment
(VIRL/CML)

Production Environment

The NetDevOps Engineers Tool Bag

(Cisco Products and Projects)

Distributed Source Control

Build Server

Configuration Management

Network Service Orchestrator
(NSO)

Network Test Tooling

pyATS, TRex

Telemetry & Monitoring

Pipeline, UTR

CLI

SNMP

NETCONF/
RESTCONF

gRPC

REST APIs

Network Device

DNA Center Platform, APIC, Meraki, IOS XE, IOS XR, NX-OS,
Firepower, UCS

Network Virtualization
Platforms

NFVIS
VIRL/CML
NSO
VNFs

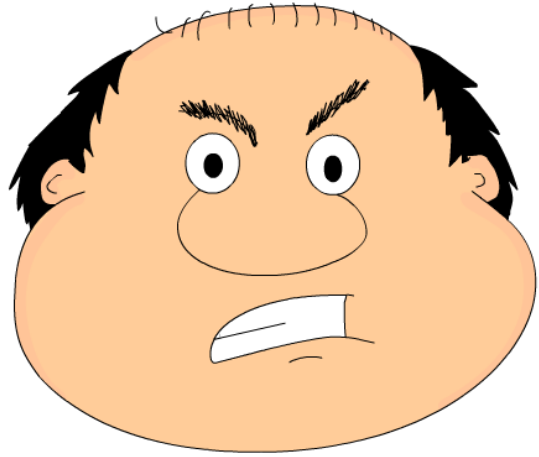
Development Environment

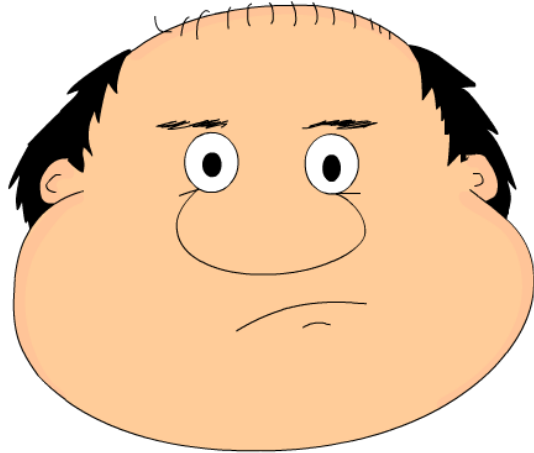
NSO, VIRL, VNFs

Test Environment

VIRL/CML

Production Environment



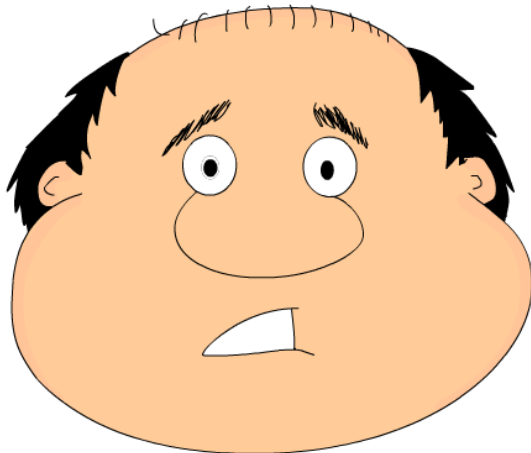


Denial

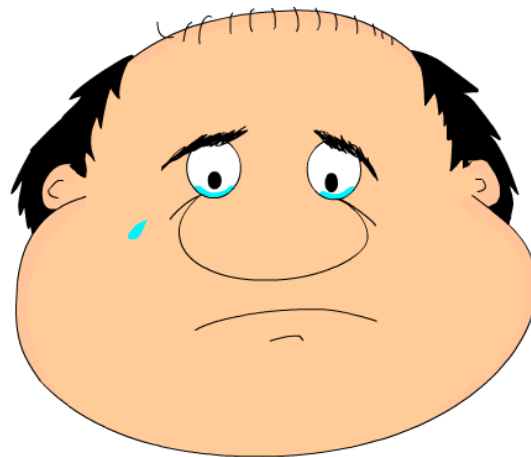


Anger

5 Stages of Grief



Bargaining



Depression



Acceptance



**IT'S
ALIVE!!!**

*Today's Network
Engineer*

Carl's 3 Step Approach to Network Programmability

Phase 1

- Python
- REST APIs
- JSON/XML
- git/GitHub

Phase 2

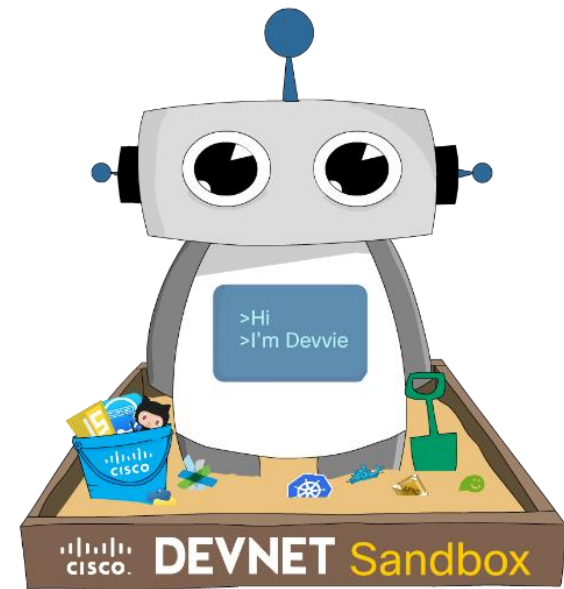
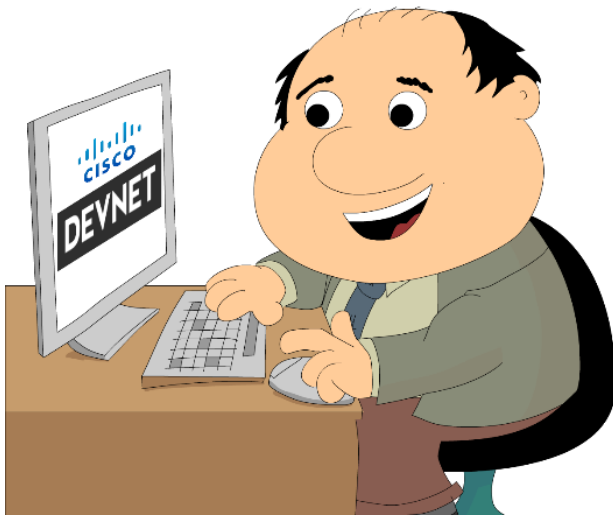
- Linux Skills
- Ansible
- Docker
- NETCONF/YANG

Phase 3

- Linux Networking
- Container Networking
- NFV

As Needed

- Network Controllers
- IOT Networking
- Cloud Networking
- "DevOps"



Carl has Embraced Programmability!



Network Skills

- Layer 2 & 3 Fundamentals
- Quality of Service
- Security and Segmentation
- Linux Networking
- Container Networking
- Cloud Networking
- IOT Networking
- Model Driven Programmability
- Network Function Virtualization

Platform Skills

- Linux Administration
- Container Fundamentals
- Micro Service Platforms
- Cloud Fundamentals

Programming Skills

- Data Formats (ex: JSON/YAML)
- Python and APIs (ex: REST)
- Source Control (ex: git)
- Configuration Management (ex: Ansible)



DEVNET

LEARN CODE INSPIRE CONNECT



DevNet Learning Labs

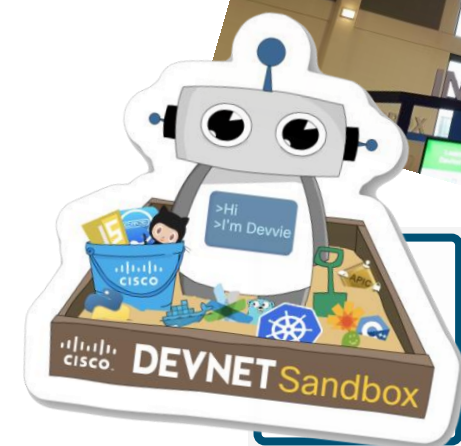
The DevNet Learning Labs will help you dive deeper into Cisco and Cisco Partner technologies, including Enterprise Networks, Data Center, Collaboration, Cloud, SDN, and IoT. If you're just getting started or need a programming REST refresher, the Learning Labs will help you get started with tutorials covering REST APIs, Python, JavaScript, and other programming technologies and concepts.

Springboards are now Tracks.

DevNet Tracks and Modules now make it easier for you to learn. Tracks and Modules are designed to guide you through learning labs that are conceptually related.

Get Started →

- Networking
- Mobility
- Coding
- Collaboration



DEVNET ZONE

- Cloud
- Networking
- Data Center
- Analytics & Automation
- Open Source
- Collaboration



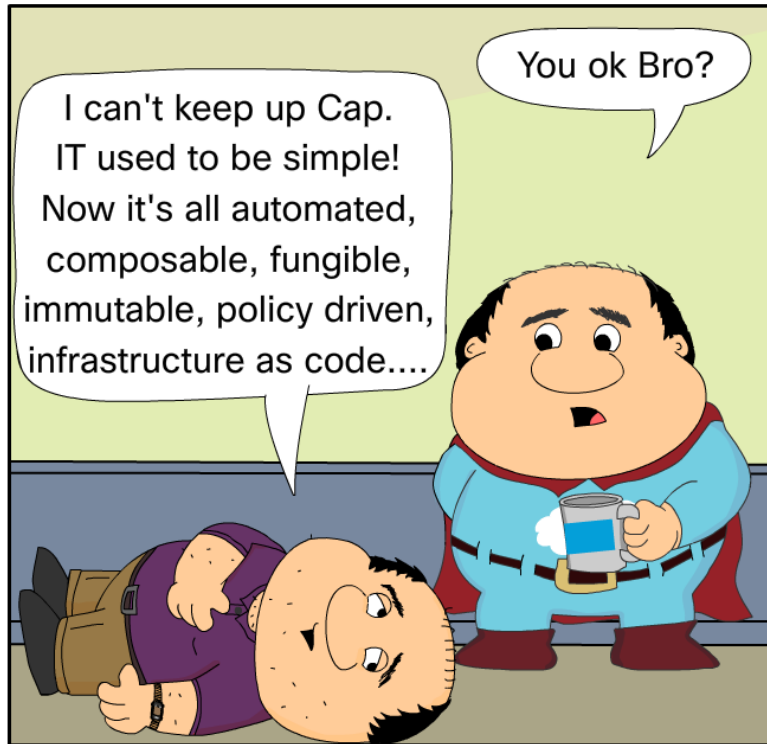
Introduction to Device Level Interfaces (ex: NETCONF/YANG)

Understand model-driven programmability and how NETCONF, YANG, and RESTCONF fit into the next generation of standard device-level interfaces.

1 Hour 15 Minutes

- Introduction to Standard Device Interfaces
- Introduction to YANG Data Modeling
- Introduction to the NETCONF Protocol
- Introduction to the RESTCONF Protocol





Summing up

Review

- We looked back on the history of the network and network engineering
- Traveled through the Four Ages of Networking
- Talked about the impact “Cloud” has had on IT
- Explored how NetDevOps will change Networking
- Considered the skills a network engineer needs today



What do do next?

- NetDevOps Readings
 - [Embrace NetDevOps, Say Goodbye to a "Culture of Fear"](#)
 - [NetDevOps Goes Beyond Infrastructure as Code](#)
 - [What does "Network as Code" Mean?](#)
 - [A Network Engineers Journey in Programmability](#)
 - [NetDevOps and the Rise of the Programmable Network](#)
- NetDevOps Learning Resources
 - [Network Programmability Basics Video Course](#)
 - [Network Programmability for Application Developers](#)
- NetDevOps Videos
 - [How to be a Network Engineer in a Programmable Age](#)
 - [Network as Code in Action](#)
 - [Benefits of Configuration Management](#)

Got more questions? Stay in touch!



Hank Preston

 hapresto@cisco.com

 [@hfpreston](https://twitter.com/hfpreston)

 <http://github.com/hpreston>



developer.cisco.com

 [@CiscoDevNet](https://twitter.com/CiscoDevNet)

 facebook.com/ciscocodevnet/

 <http://github.com/CiscoDevNet>



DEVNET
developer.cisco.com