February 15 - 19, 2016 · Berlin, Germany

We're ready. Are you?

APIC-EM-WRiedel-BRKCRS-3011-20160213.pptx

APIC-EM
A scale out architecture for SDN in the Enterprise

Wolfgang Riedel

BRKCRS-3011

Wolfgang Riedel
Principal Engineer Engineering
ENG Product Management – Architecture
CCIE #13804, VCP #42559
wolfgang@cisco.com

Cisco live!

Who is Wolfgang Riedel ???

- **Personal:**
  - Location: Erlangen, Germany (between Munich – Frankfurt)
  - Other Interests: Alpine Snowboarding, High-End Audio, AS51871, Data Center, Real World LAB, High-performance sports cars, Geothermal DC cooling research project, …
- **Background:**
  - Joined CISCO January 2001
  - Before; self-employed as an in-depended consultant in the Networking and IT space for more then fifteen years.
    - ✓ SE – RS Germany (2001 – 2006) -> Campus with a DC attached
    - ✓ CSE – DC EMEA (2006 - 2008) -> DC with Campus attached
    - ✓ CE – Cisco CTO Office (2008 – 2011)
    - ✓ PE – ARND (2011 – 2013)
    - ✓ PE – CTO Team ENG (2013 - 2014)
    - ✓ PE – Architecture Team ENG (2014 – …)
  - HA Campus & DC Design, Routed Access, DC POD Design
  - CCIE RS, VCP 3/4/5 and pile of CPOC's
  - Worked with more then 250 customers within several projects over the last +15 years
  - Individual Contributor: Cat4k, Cat6k, N7k, ASR1k, FC, FCoE, DCB, UCS, N5k, N2k, N1k, PoE FEX, vPC, OTV, LISP (Pioneer Award), OF, SDN
- **Stuff I am currently working on:**
  - Network Transformation, Architecture (Mark, Matthias, Tim, Dave, Jason, Simone, I)
  - APIC-EM, DNS-AS, AVC, USP
  - TECSDN-3600 + BRKCRS-3011 + BRKSDN-3004

Cisco live!

Agenda

1. Introduction
2. World of Controllers and Technologies
3. Controllers
    1. OpenFlow – Stanford Clean Slate
    2. APIC-DC - Application Policy Infrastructure Controller for the Datacenter
    3. Virtual Managed Services
    4. OSC – Open SDN Controller
    5. ODL – Open Day Light Controller
    6. APIC-EM - Application Policy Infrastructure Controller for the Enterprise
    7. PI and APIC-EM
4. APIC-EM
    1. Policy Infrastructure
    2. Auto Scale Architecture
    3. Grapevine Cloud Deployment
    4. Use Cases
5. Demo
6. A Few Conclusions and Q&A, if we have time

1. Introduction

Industry trends in Networking

Cloud (2008)

OpenFlow Networking (Stanford clean slate) (2011)

Software Defined Networking (2012)

OPEN DAYLIGHT www.opendaylight.org

Open Daylight Project (2013)

DevOps, The API Driven Datacenter (2013)

Network Function Virtualization (2013)

Managing Networks through abstractions (2014)

Metadata Driven Networking (2016)

Cisco live!

Atomic Services (2018)

SDN – **S**till **D**on't k**N**ow – **S**tanford **D**efined **N**etworking

The Promise of OF/SDN had been "Decoupling Policy from Configuration"

Physical separation of control and data plane

Managing the network through abstractions

Whitebox routing and switching

Software Defined Networking

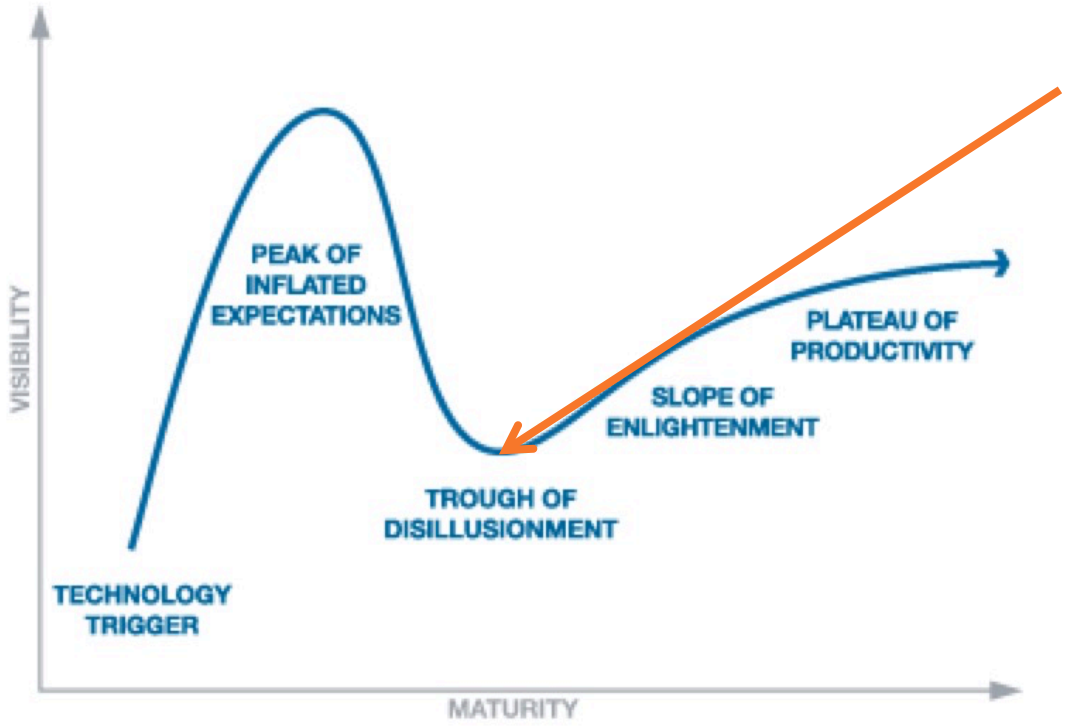
Packet forwarding on x86 compute

Running networks in agile DEV-OPS model

You can't just buy SDN.
It's an architecture which you have to embrace and life

SDN – Hype Cycle

Where we are with SDN 2016, five years later



- Technology Trigger
- Peak of Inflated Expectations
- **Trough of Disillusionment**
  - Interest wanes as experiments and implementations fail to deliver.
  - Producers of the technology shake out or fail.
  - Investments continue only if the surviving providers improve their products to the satisfaction of early adopters.
- Slope of Enlightenment
- Plateau of Productivity

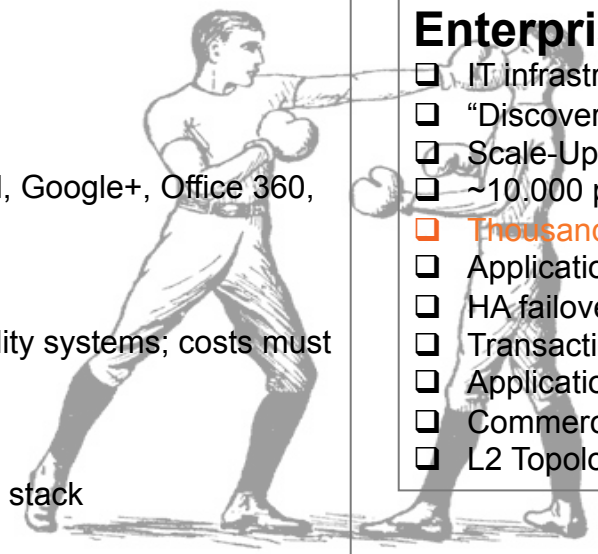
Gartner Hype Cycle

Today's DC Architectural Battle

System administration is over - we should stop doing it

Web Approach (MSDC)

- IT infrastructure core of its business
- Warehouse Datacenter
- Scale-Out Architecture
- ~100.000 of physical servers
- Single Application Optimization
- Small Number of Applications, like Gmail, Google+, Office 360, Xbox, Bing, …
- Application Designed for Failure
- Automate everything possible
- It's all about being super-cheap commodity systems; costs must grow in a "sub-linear" fashion
- Open Source
- Backbone Bandwidth Calendaring
- TDM style provisioning with custom TCP stack
- L3 Topology



Enterprise Approach (EPDC)

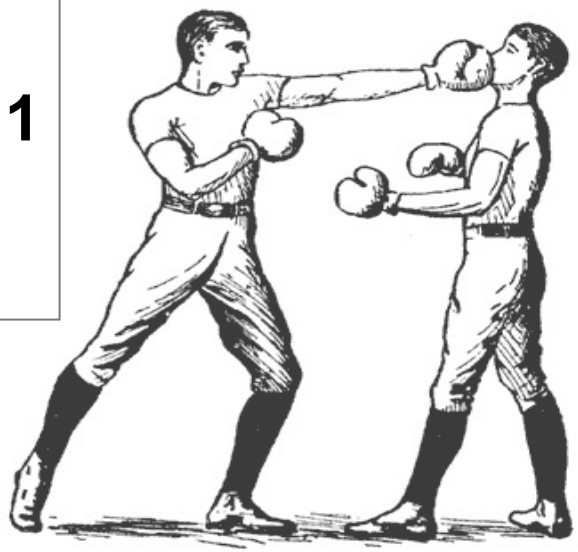
- IT infrastructure is an expense
- "Discovery" Datacenter
- Scale-Up Architecture
- ~10.000 physical servers
- Thousands of Applications
- Application trust boundaries
- HA failover model
- Transactional
- Application specific Infrastructure
- Commercial Of The Shelf
- L2 Topology

Today's DC Architectural Battle

Device to Admin Ratio

2009

Traditional IT: 50:1
Amazon: 200:1
Google: 10000:1

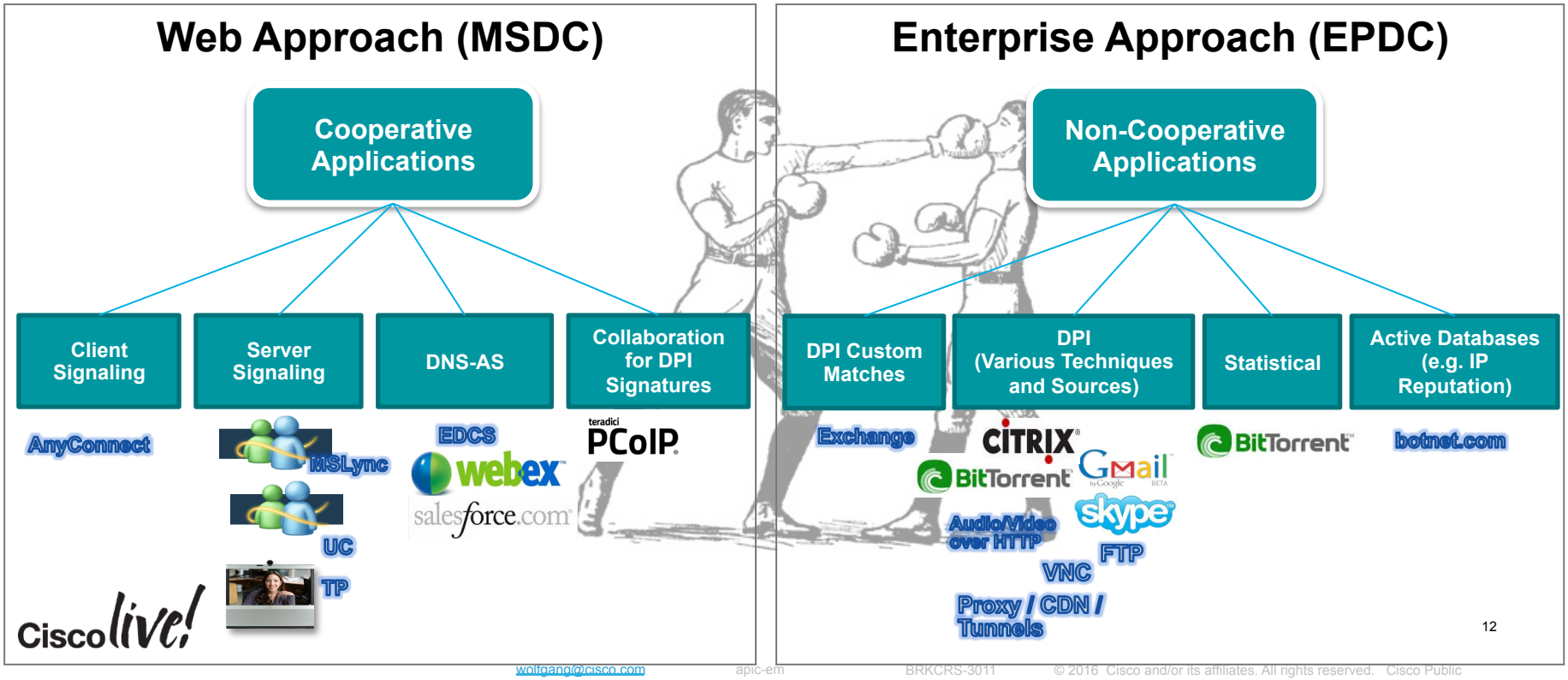


2013

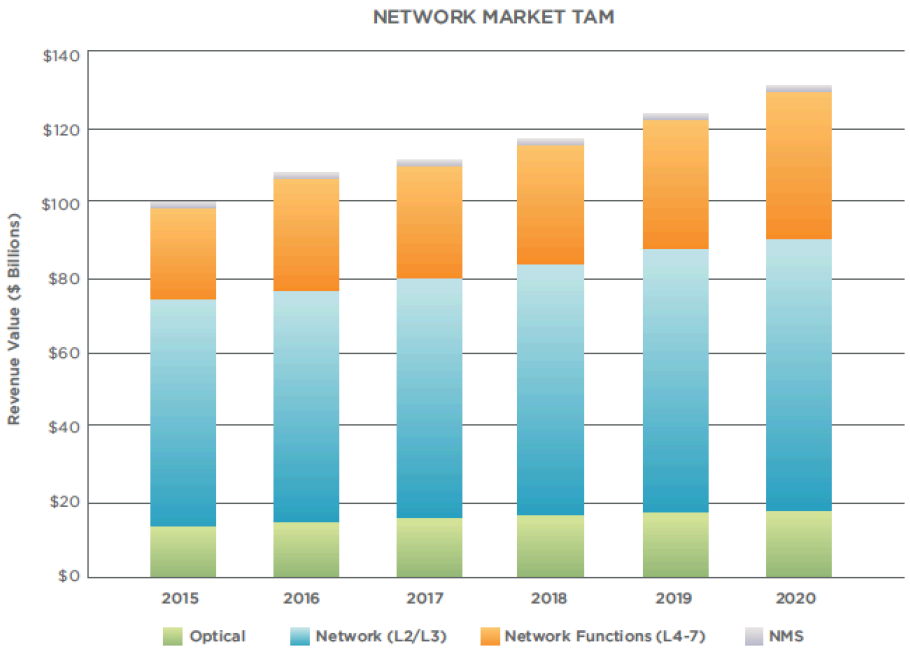
Traditional IT 50:1
Amazon 10000:1
Google: 30000:1

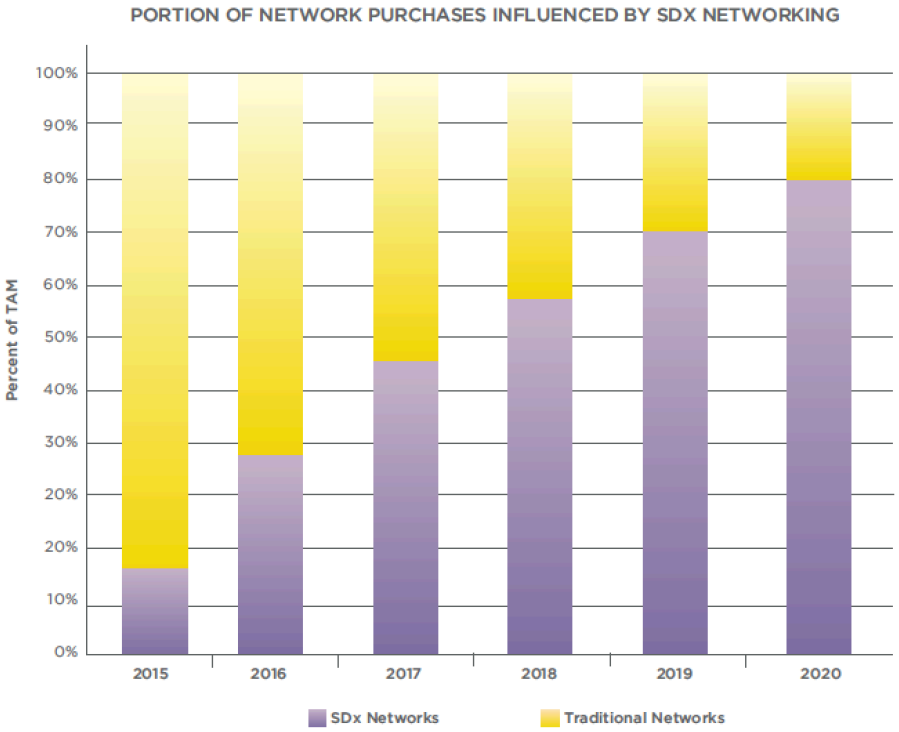
Today's DC Architectural Battle

It's all about the Application



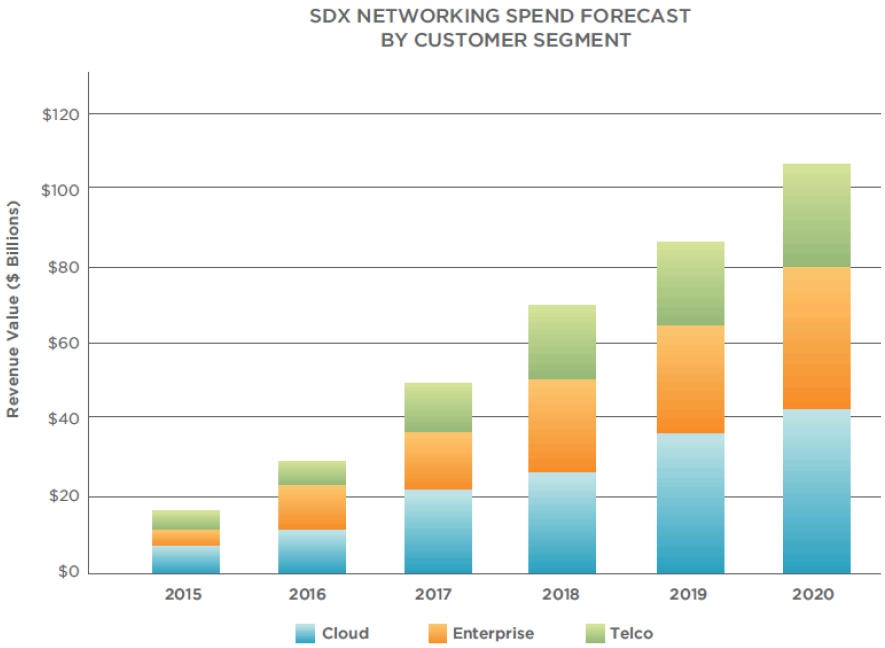
SDx Influences Network Purchases

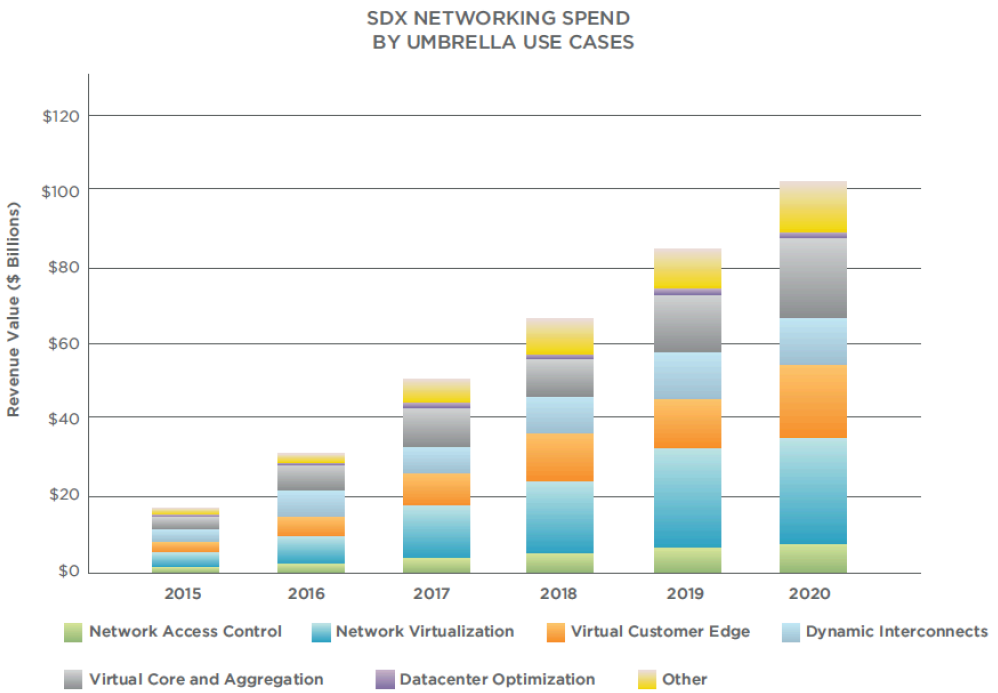




Source: SDx Central

SDx Spending by Customer and Use Cases





Source: SDx Central

Managing the network through abstractions

There are two approaches to Control Systems

IMPERATIVE CONTROL

Baggage handlers follow sequences of simple, basic instructions

DECLARATIVE CONTROL

Air traffic control tells where to take off from, but not *how* to fly the plane

Managing the network through abstractions

There are two approaches to Control Systems

IMPERATIVE CONTROL

DECLARATIVE CONTROL

It's 2016 and network admins still enjoy being "masters of complexity"

Cisco live!

APIC - EM Design Points

Validated with FAT & CAT

Abstraction and Automation of Manual Net… Operations

Reduce Network Complexity

Advanced Visualization (HTML5/Ja… …ct oriented interface)

Brownfield Support –
No Software / Hardware upgrade required

Start with small set of solvable problems …anagement , Zero Touch Deployment and IWAN as key applications with identifiable metrics (OPEX savings , ROI)

Low Risk adoption of SDN

Elastic Services Infrastructure ensures scaling as adoption grows

Enterprise Scale… …action network use

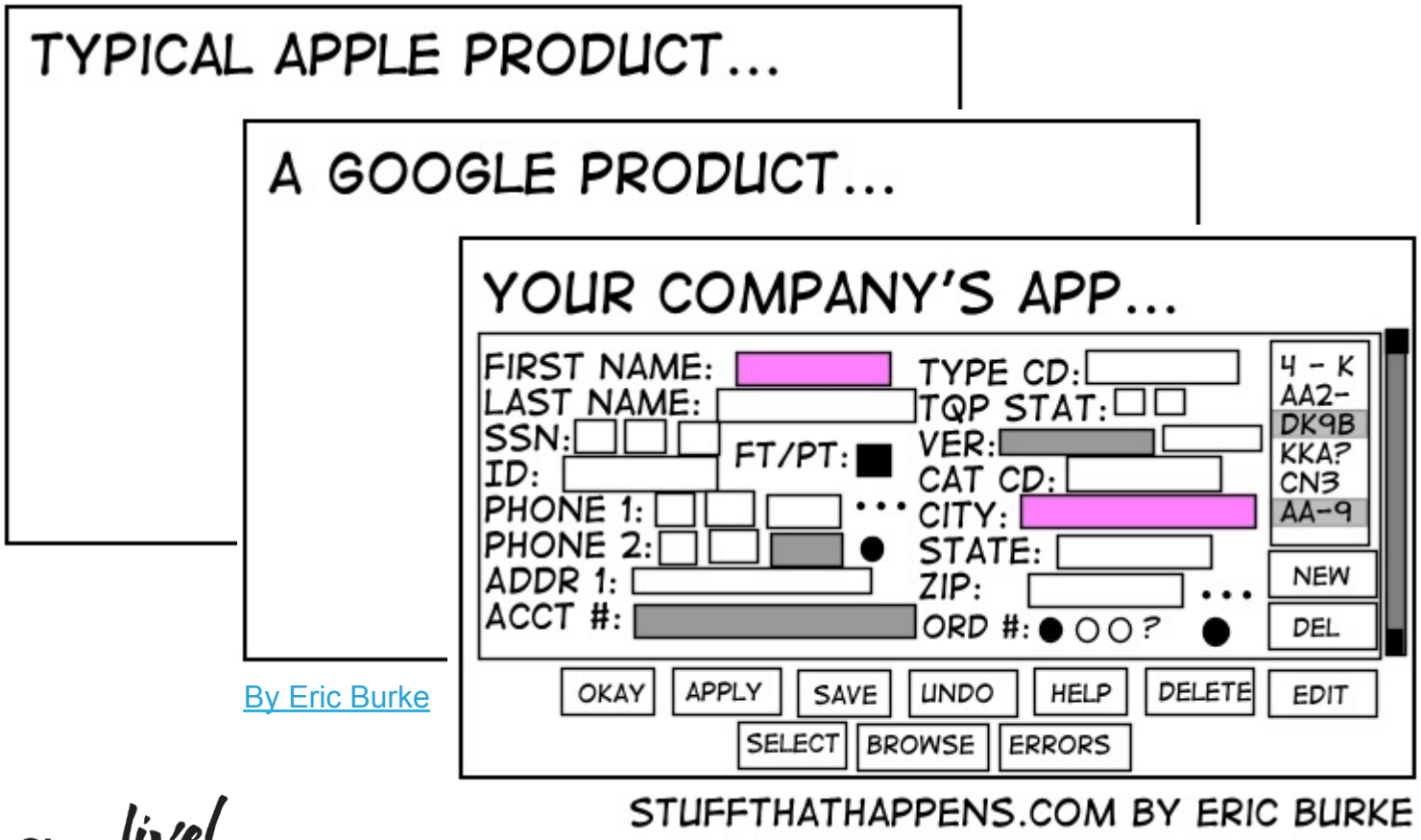
Auto-Translation of high level business intent into network control function

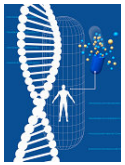
Pr… …th minimal to no programming requirement

Advanced analytics for real time network visibility and faster response time

Don't make it simpler by making it more complex!

Enterprise SDN customer asks in an iPhone world

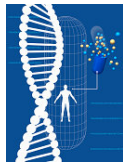




New Network Needs for the Digital Business

Information Era Network		Digital Ready Network
Closed and Hardware Centric		Open , Programmable, Software Driven
Manual Box-by-Box Management		Network Wide Policy Based Automation
Perimeter Based Reactive Security		Proactive Context-Based Security Everywhere
IT & Historical Analytics		Business & Real Time Analytics

Principles for the New Network Architecture

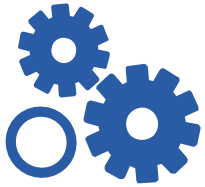


Open and Software-Driven

Cloud

Services & Apps Built for Cloud Consumption

On-Demand Scale
Faster IT Innovation



Controllers

Complete Controller-based Automation

Simplicity through Abstraction
Centralized Policy

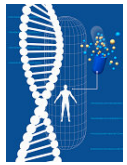
Virtualization

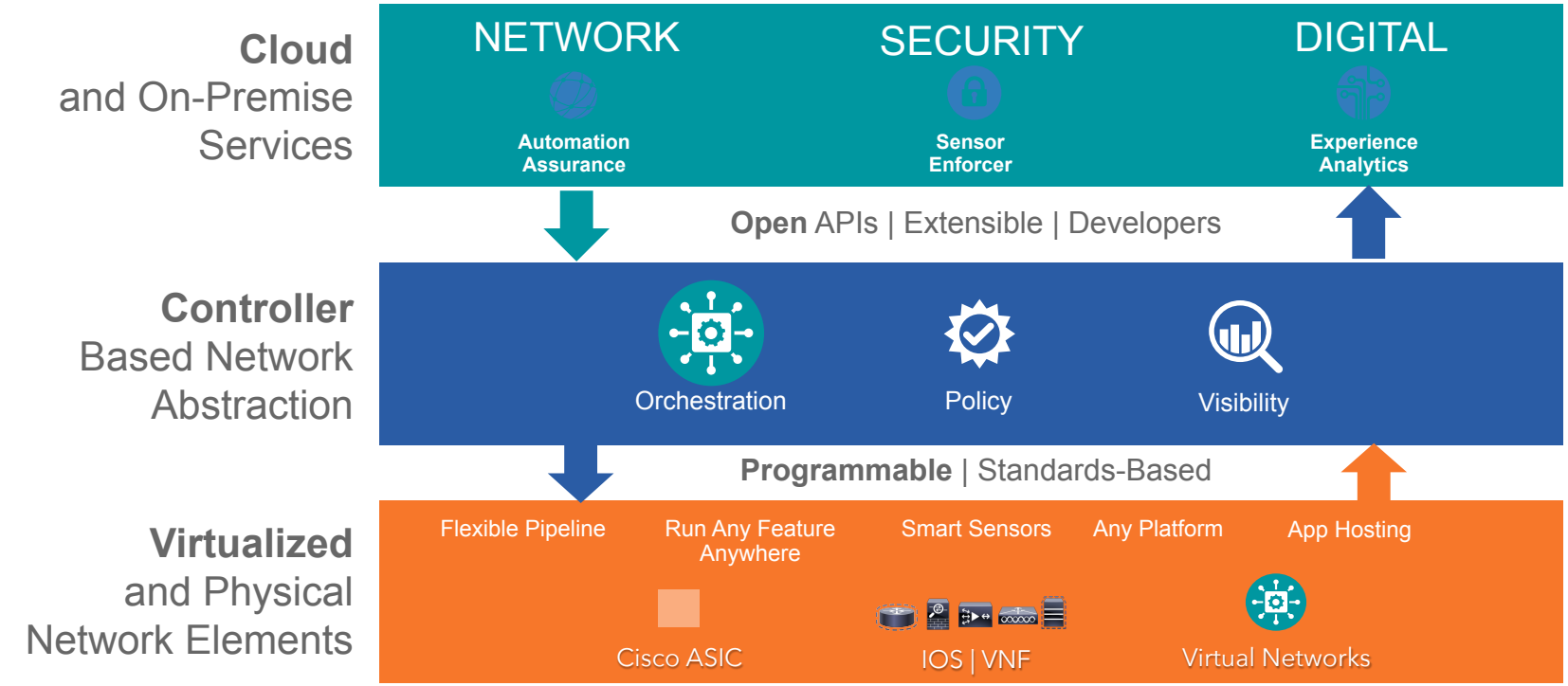
Virtualized Networks Functions and App Hosting

Freedom of Choice - Any Platform
Run Applications over the Network

Cisco live!

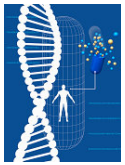
Digital Network Architecture (DNA)





Cisco live!

DNA Service Innovation



Available Today

APIC EM
Enterprise Network Controller

IWAN App
Automate Hybrid WAN Services

Path Trace App
Faster Troubleshooting

New Services

Available Now

Plug and Play
Automate Day 0 Deployment

Controller

March 2016

Enterprise NFV
Run Services Anywhere

Virtualization

March 2016

Easy QoS
Automate Collab Traffic Priority

Controller

Available Now

Lancope
Network as a Sensor

Controller

January 2016

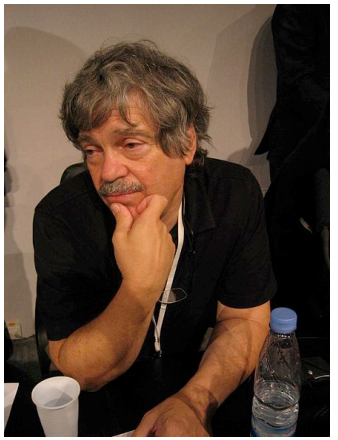
CMX Cloud
Location Service Subscription

Cloud

Available on DNA-Ready Infrastructure through Cisco ONE Software

ISR 4000 | ASR 1000 | Catalyst 6800 | Catalyst 4000-E | Catalyst 3850 | Catalyst 3650 | Aironet 802.11ac

“People who are really serious about software should make their own hardware.”

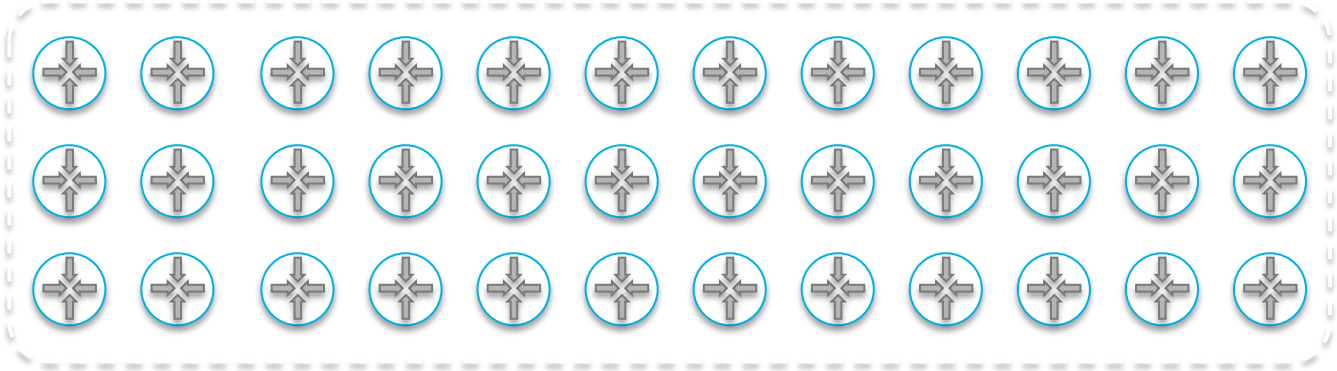


Alan Kay, 1982

Cisco live!

1.1 Analogies

Distributed Networking has worked

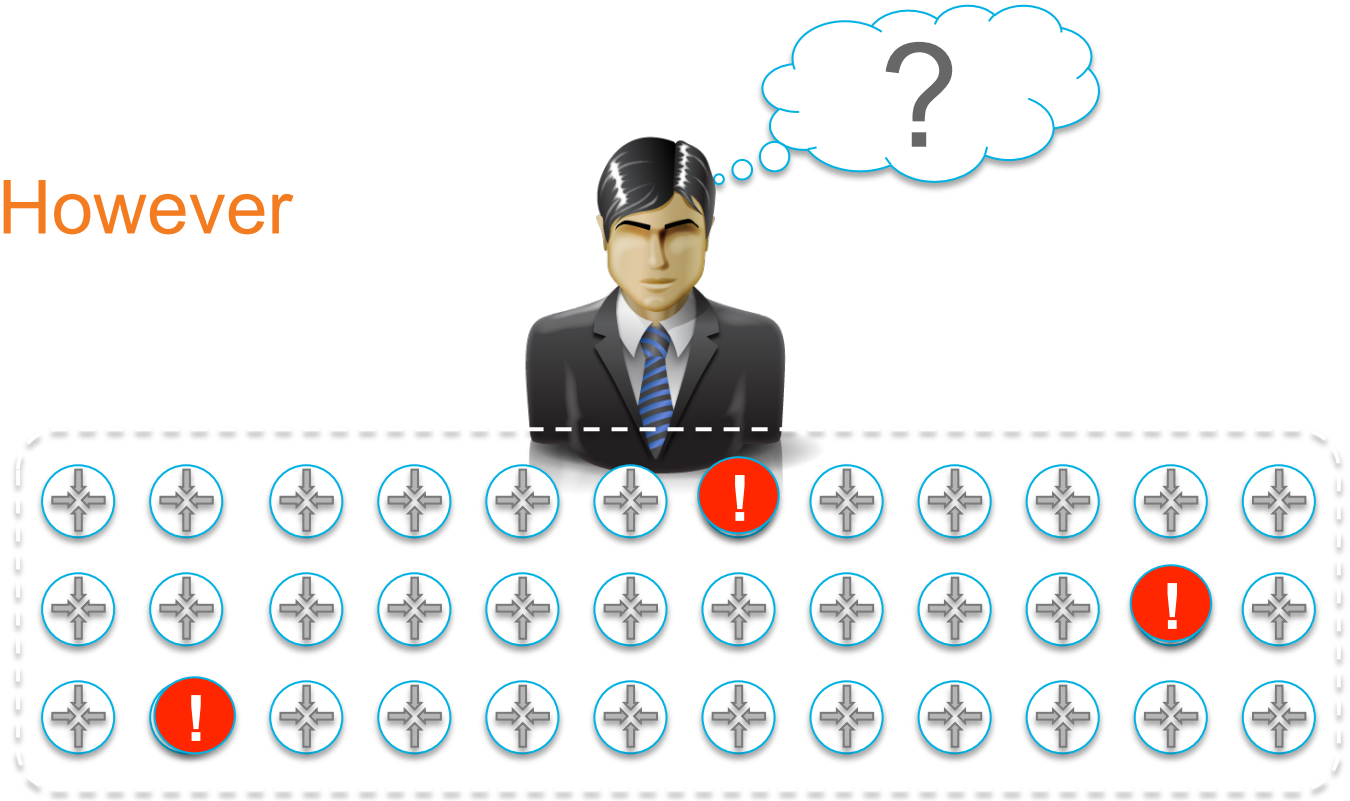


Resiliency/Scale has been proven

Cisco live!

Distributed Networking has worked

However

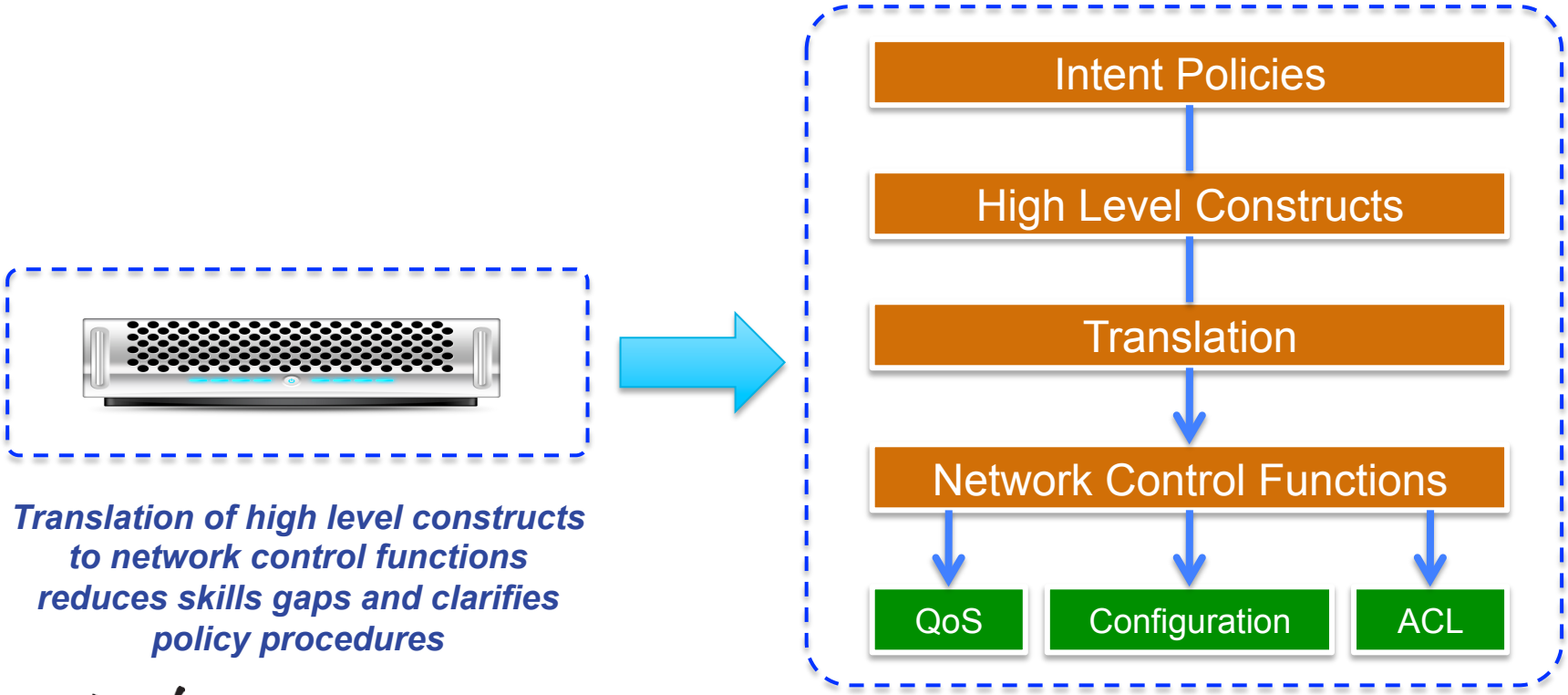


Distributed Networking adds complexity to manage/comprehend

Admin still makes network behavior decisions

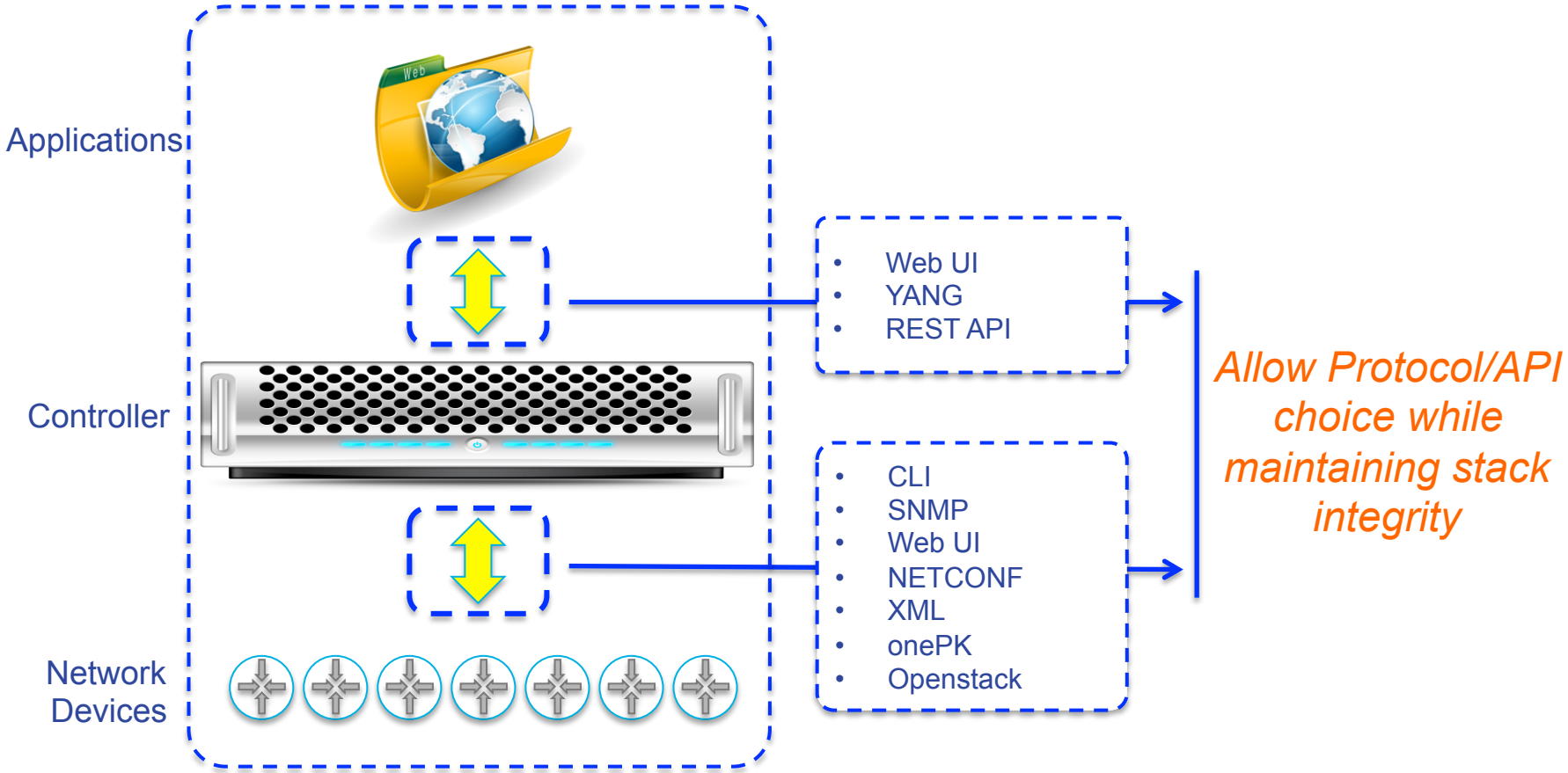
But uses controller
to mask complexity

Cisco Intent Policy Management

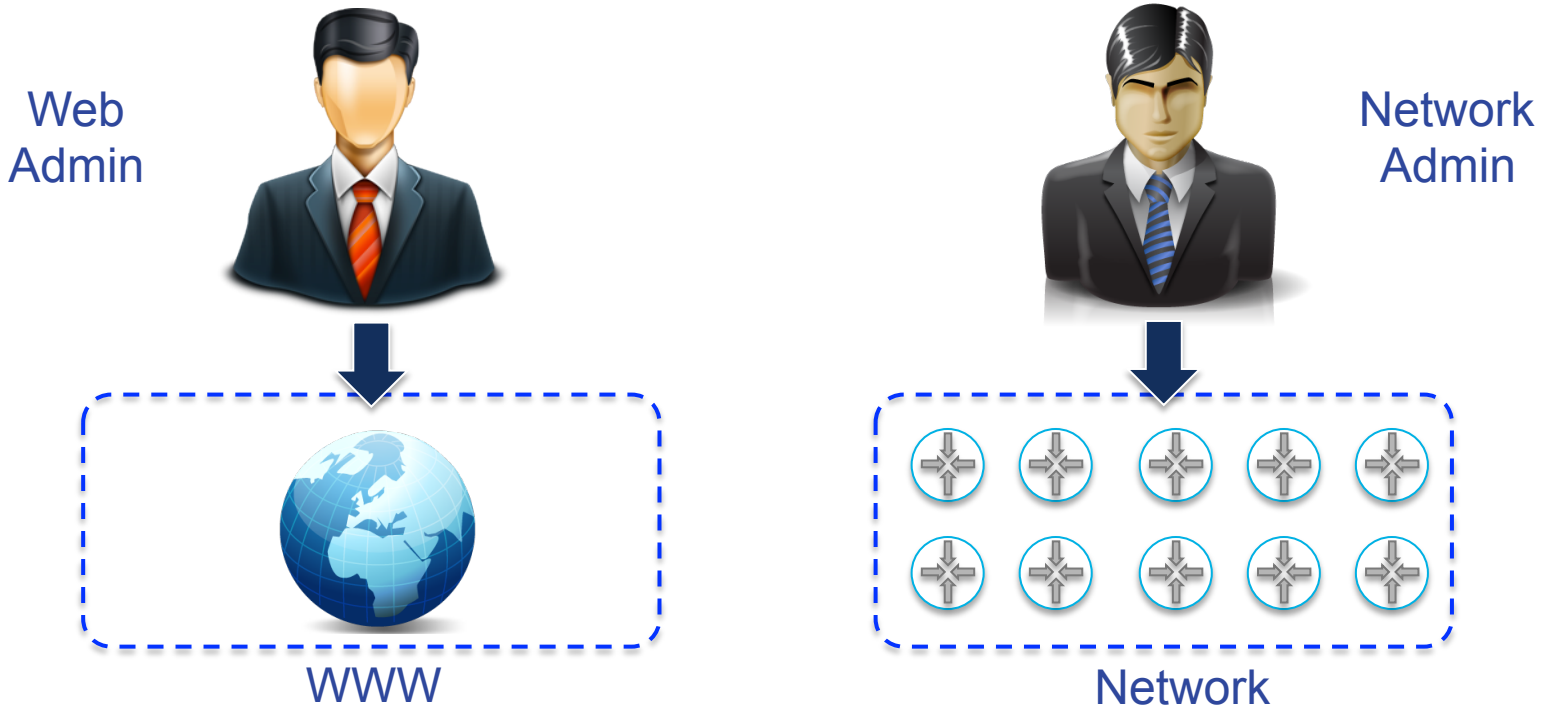


Translation of high level constructs to network control functions reduces skills gaps and clarifies policy procedures

Cisco live!

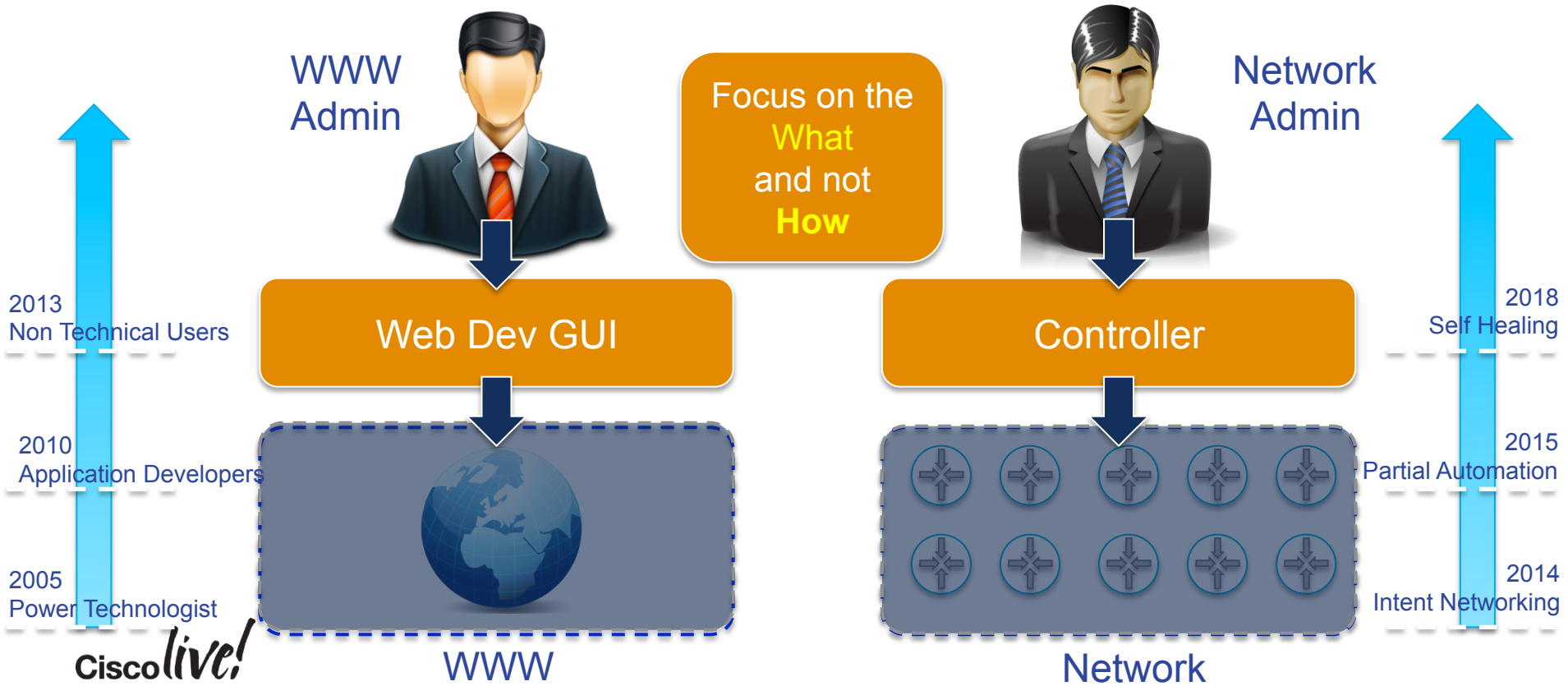


Both at one time had direct admin control



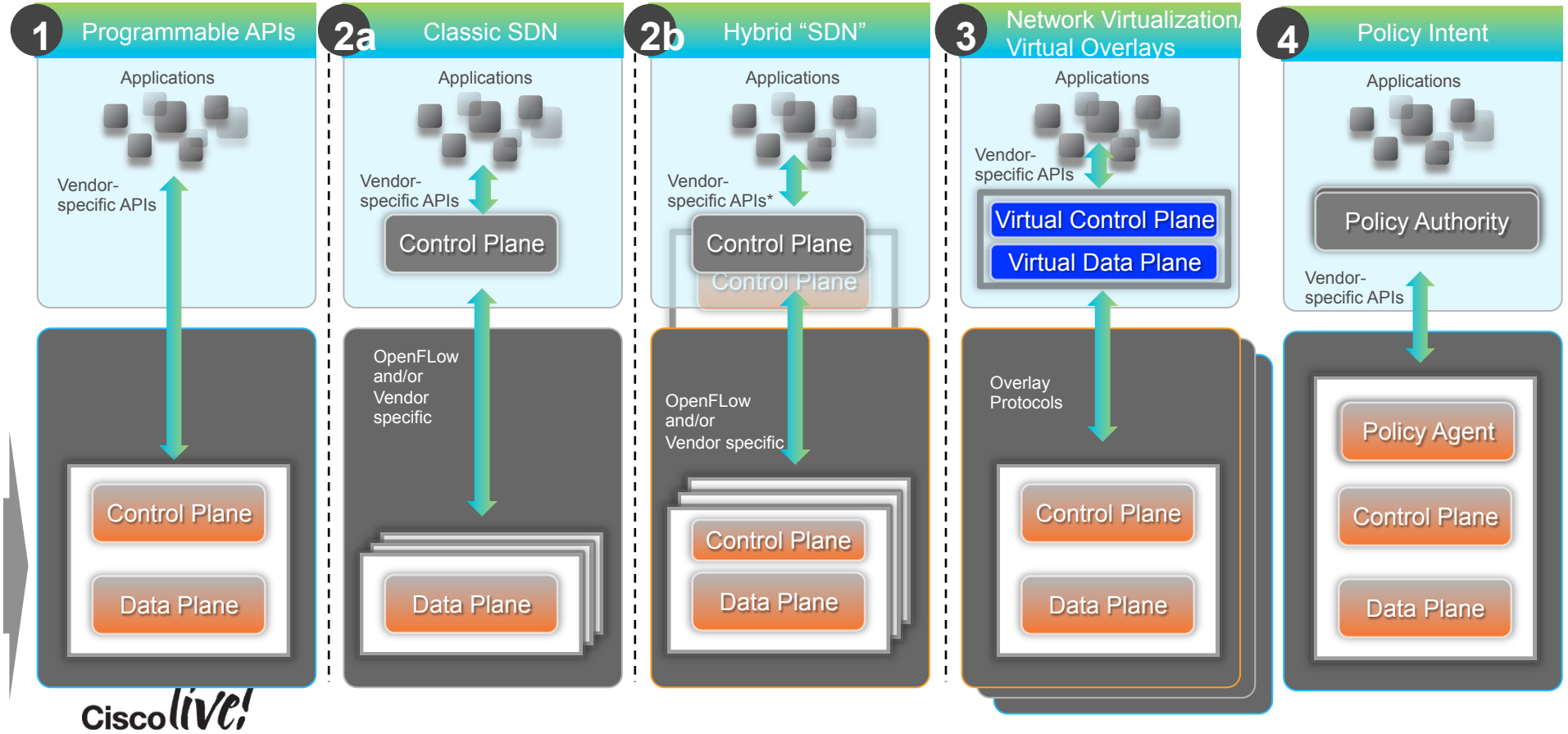
Direction to abstract complexity

Network Management should follow Web Development



2. World of Controllers and Technologies

Network Programmability Models

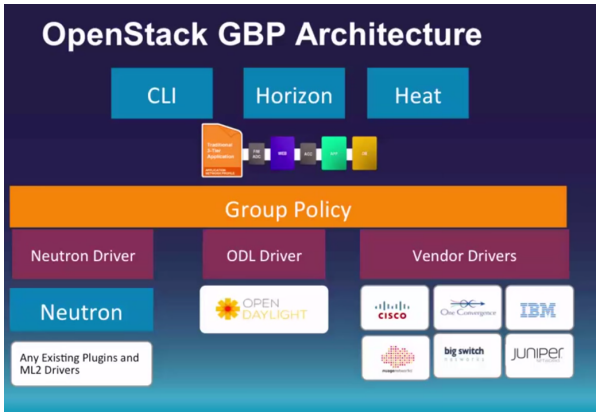


SDN Controllers – Types

There's nothing like the SDN controller

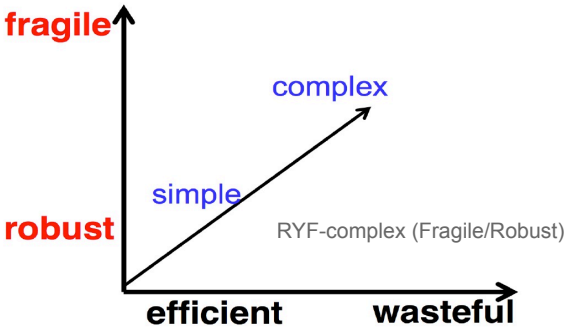
- SDN Config-Pusher
  - Orchestration (robot micromanaging manual to-do's)
  - NCM (Network Configuration Management)
  - Customers may see or edit any part of the config
  - ✓ Prime Infrastructure, Action Packed, Solarwinds
  - ✓ Puppet, Chef
  - ✓ Openstack
  - ✓ Netconf
- SDN Policy-Compiler
  - Customer is never exposed to nor has access to nor influence over direct snippets of configuration elements.
  - They express their intent only – like in a programming language – and the conversion to machine language is invisible.
  - ✓ Cisco APIC-EM
- SDN Policy-Enabler
  - ✓ Cisco APIC-DC
- SDN Overlay Controller
  - ✓ VMWare: VCS, VCD, NSX
  - ✓ VSM (N1kv), EVP, VTS
  - ✓ Windows Server, Microsoft System Center)
- SDN Open Flow Controller
  - ✓ Primary for research

Group Based Policy / NIC



SDN Controllers – Types

Start with the End in Mind - the RYF-complex (Fragile/Robust)



RYF-complex (Fragile/Robust)

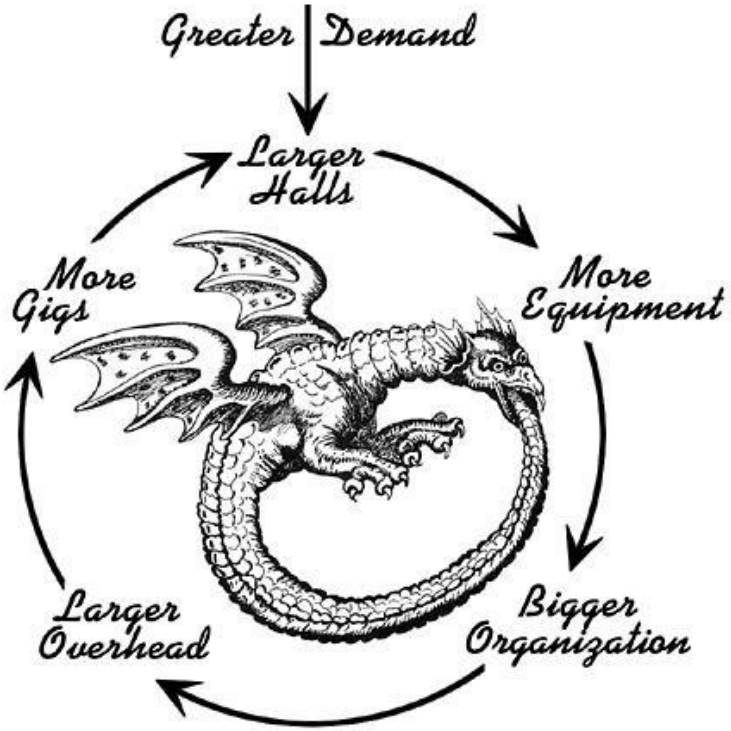
Five dimensions of robustness in complex systems
(1) Reliability
(2) Efficiency
(3) Scalability
(4) Modularity
(5) Evolvability

See J. Doyle, et. al.,
"Robustness and the Internet: Theoretical Foundations"

Alderson and Doyle identify four kinds of constraints on system robustness:
(1) Component-level
(2) System-level
(3) Protocols
(4) Emergent constraints

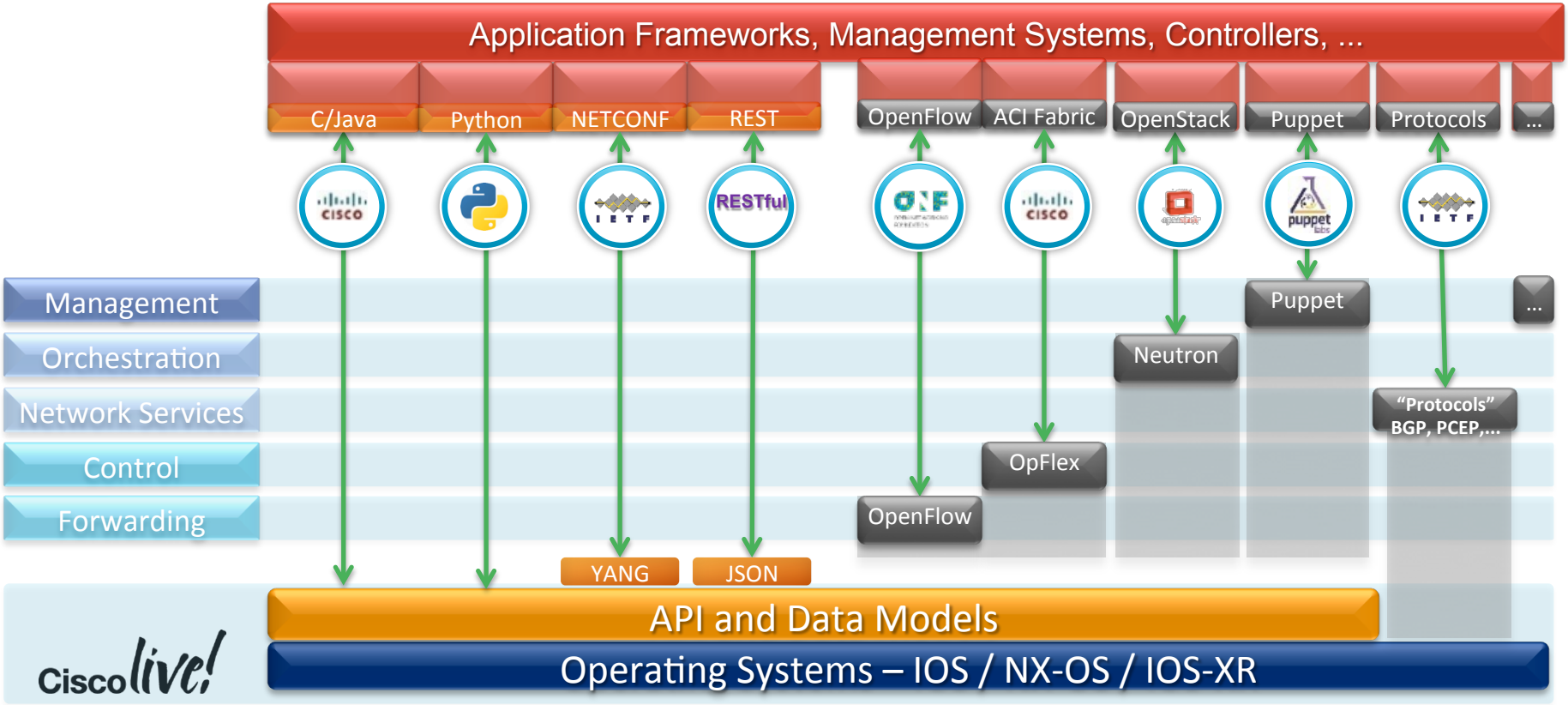
Complex systems science as conflicting constraints
John C. Doyle, HOT and SF networks

Grateful Dead Sources
How the Dragon Urobouros (Giga Exponentia) Makes Us Go Round and Round

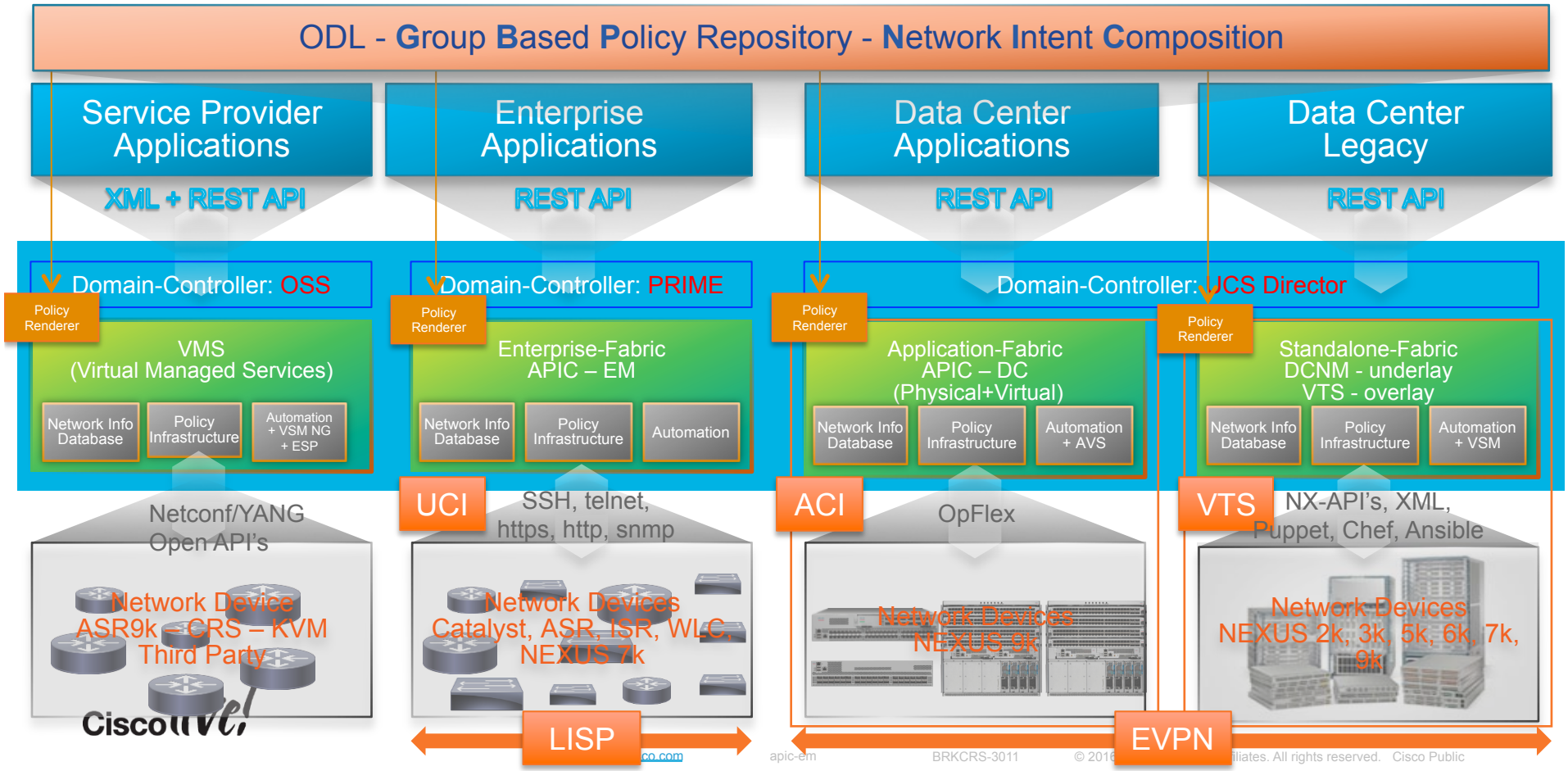


Device Programmability Options

No Single Answer!

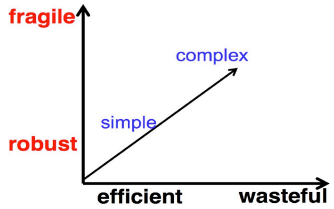


Cross Domain Controller Architecture



SDN Controller – Overview

OK that looks really ugly but wait a minute…



- Four wheels
- Steering wheel
- Gas pedal
- Brake pedal

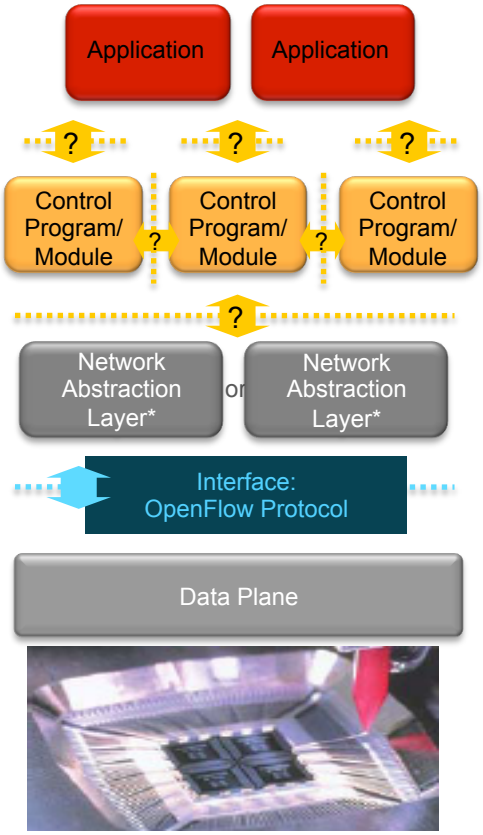
… all cars

But complete different use-cases

Cisco live!

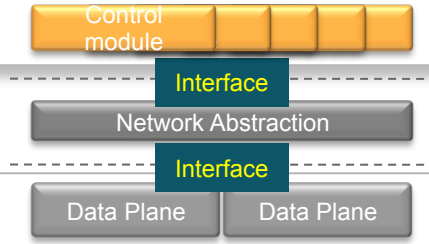
3. Controllers

3.1 OpenFlow - Stanford Clean Slate



SDN

- Mandates a separation of control and data plane and an interface to it
- Higher level abstraction to store state information
- Classical networking relies on distributed state but *local* decision making
- SDN Networks is all about distributed state but *central* decision making
- Enabler to deal with the networks in a different way as we do today



OF

- Separation of control plane and data plane on packet switched networks
- Allows for optimized placement of these components
- Low-level interaction set of changing the state of the network data plane via an switch API (similar to Broadcom, Fulcrum, Cisco ASIC API's)
- A Communications protocol which allows the SDN controller to manipulate flow tables with simple primitives of match/action operations
- Pro-active mode pretty much like CEF, Re-active mode similar to flow based switching
- Asynchronous reporting of statistics
- *OpenFlow* allows up to 12-*tuple* wildcard and/or exact match

Cisco live!

SDN Controllers – Types

There's nothing like "a OpenFlow controller"

SDN Open Flow Controller

- ✓ **NOX** (C++/Python) NOX was the first OpenFlow controller.
- ✓ **POX** (Python) Pox as a general SDN controller that supports OpenFlow. It has a high-level SDN API including a queriable topology graph and support for virtualization.
- ✓ **Jaxon** (Java) Jaxon is a NOX-dependent Java-based OpenFlow Controller.
- ✓ **Trema** (C/Ruby) Trema is a full-stack framework for developing OpenFlow controllers in Ruby and C.
- ✓ **Beacon** (Java) Beacon is a Java-based controller that supports both event-based and threaded operation.
- ✓ **Floodlight** (Java) The Floodlight controller is Java-based OpenFlow Controller. It was forked from the Beacon controller, originally developed by David Erickson at Stanford.
- ✓ **Maestro** (Java) Maestro is an OpenFlow "operating system" for orchestrating network control applications.
- ✓ **NDDI - OESS** OESS is an application to configure and control OpenFlow Enabled switches through a very simple and user friendly User Interface.
- ✓ **Ryu** (Python) Ryu is an open-sourced Network Operating System (NOS) that supports OpenFlow.
- ✓ **NodeFlow** (JavaScript) NodeFlow is an OpenFlow controller written in pure JavaScript for Node.JS.
- ✓ **ovs-controller** (C) Trivial reference controller packaged with Open vSwitch.
- ✓ **RouteFlow** RouteFlow, is an open source project to provide virtualized IP routing services over OpenFlow enabled hardware. RouteFlow is composed by an OpenFlow Controller application, an independent RouteFlow Server, and a virtual network environment that reproduces the connectivity of a physical infrastructure and runs IP routing engines (e.g. Quagga).
- ✓ **Flowvisor** (Java) FlowVisor is a special purpose OpenFlow controller that acts as a transparent proxy between OpenFlow switches and multiple OpenFlow controllers.
- ✓ **SNAC** (C++) SNAC is an OpenFlow controller builton NOX, which uses a web-based policy manager to manage the network.
- ✓ **Resonance** Resonance is a Network Access Control application built using NOX and OpenFlow.
- ✓ **Oflops** (C) OFlops (OpenFLow Operations Per Second) is a standalone controller that benchmarks various aspects of an OpenFlow switch.
- ✓ **RouteFlow** RouteFlow, is an open source project to provide virtualized IP routing services over OpenFlow enabled hardware. RouteFlow is composed by an OpenFlow Controller application, an independent RouteFlow Server, and a virtual network environment that reproduces the connectivity of a physical infrastructure and runs IP routing engines (e.g. Quagga).
- ✓ **Flowvisor** (Java) FlowVisor is a special purpose OpenFlow controller that acts as a transparent proxy between OpenFlow switches and multiple OpenFlow controllers.
- ✓ XNC Cisco Extensible Network Controller (XNC) is the first commercial version of the OpenDaylight controller
- ✓ **ODL Linux-Foundation: community-driven, open source controller framework (Brocade, Cisco, Citrix, Ericsson, IBM, Juniper, Microsoft, RedHat)**

SDN Controllers – Types

comparison of open source controllers

Use-Cases \ Controllers	Trema	Nox/Pox	RYU	Floodlight	ODL	ONOS***
Network Virtualizaiton by Virtual Overlays	YES	YES	YES	PARTIAL	YES	NO
Hop-by-hop Network Virtualization	NO	NO	NO	YES	YES	YES
OpenStack Neutron Support	NO	NO	YES	YES	YES	NO
Legacy Network Interoperability	NO	NO	NO	NO	YES	PARTIAL
Service Insertion and Chaining	NO	NO	PARTIAL	NO	YES	PARTIAL
Network Monitoring	PARTIAL	PARTIAL	YES	YES	YES	YES
Policy Enforcement	NO	NO	NO	PARTIAL	YES	PARTIAL
Load Balancing	NO	NO	NO	NO	YES	NO
Traffic Engineering	PARTIAL	PARTIAL	PARTIAL	PARTIAL	YES	PARTIAL
Dynamic Network Taps	NO	NO	YES	YES	YES	NO
Multi-Layer Network Optimization	NO	NO	NO	NO	PARTIAL	PARTIAL
Transport Networks - NV, Traffic-Rerouting, Interconnecting DCs, etc.	NO	NO	PARTIAL	NO	PARTIAL	PARTIAL
Campus Networks	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	NO
Routing	YES	NO	YES	YES	YES	YES

Ciscolive!

[SDN Series Part Eight: Comparison Of Open Source SDN Controllers]

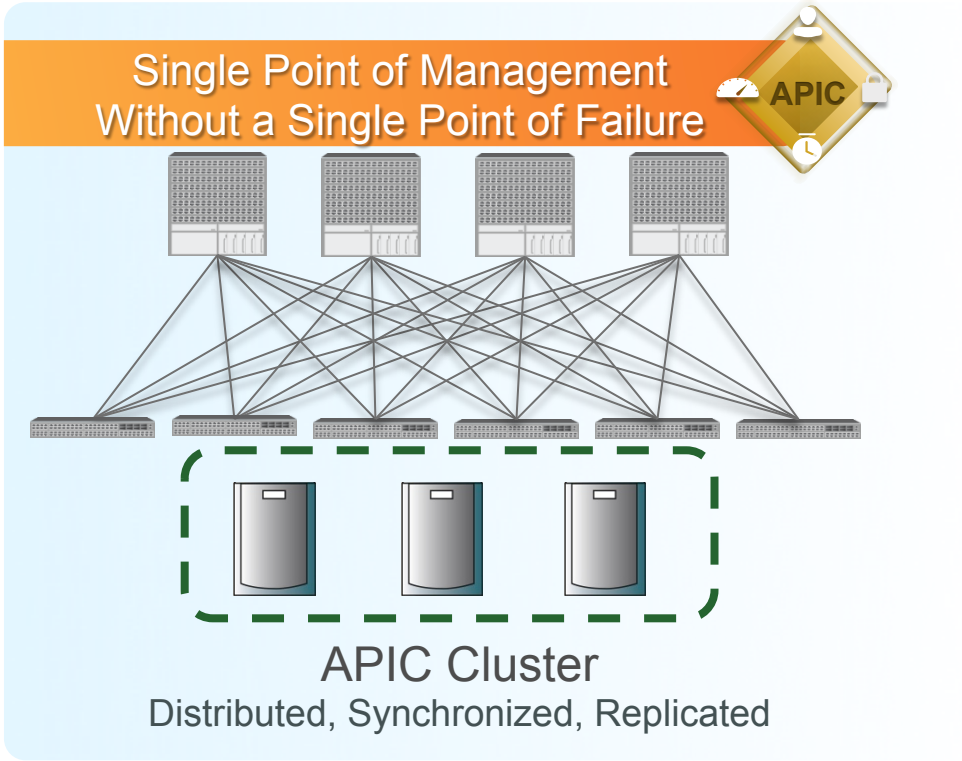
3.2 APIC-DC
(Application Policy Infrastructure Controller for the Datacenter)

Cisco ACI: Full Stack SDN in Data Center

Centralized Automation and Fabric Management

ACI = CONTROLLER + POLICY MODEL+ NEXUS 9k

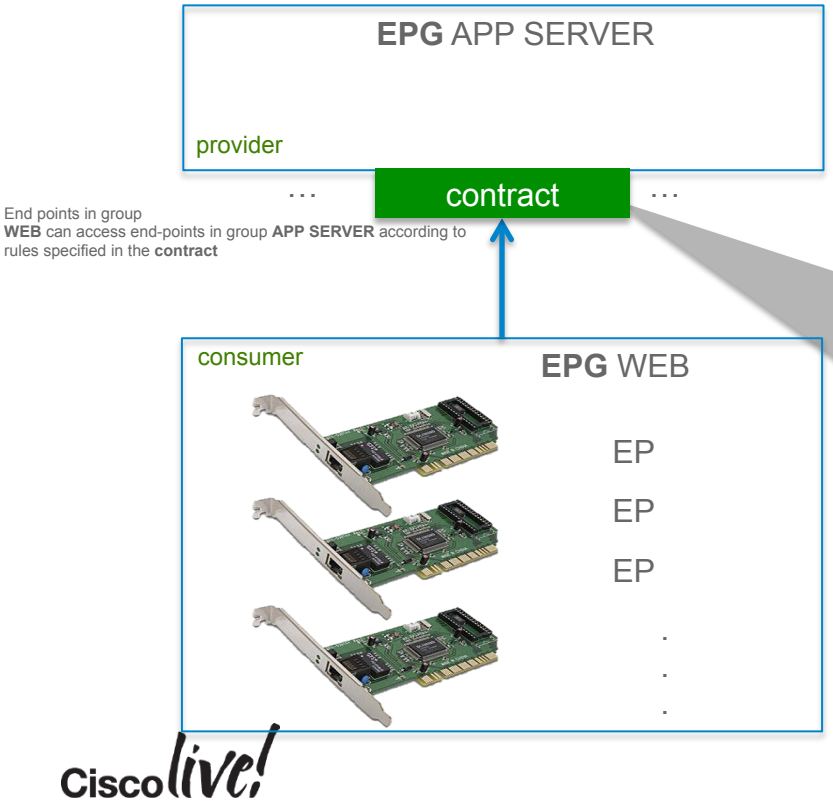
- Turnkey integrated solution with security, centralized management, compliance and scale
- L4-L7 Service Graph
- Automated application centric-policy model with embedded security
- Simplify provisioning, operating through relational object-model
- Fully programmable (REST API, Python bindings)
- Broad and deep ecosystem



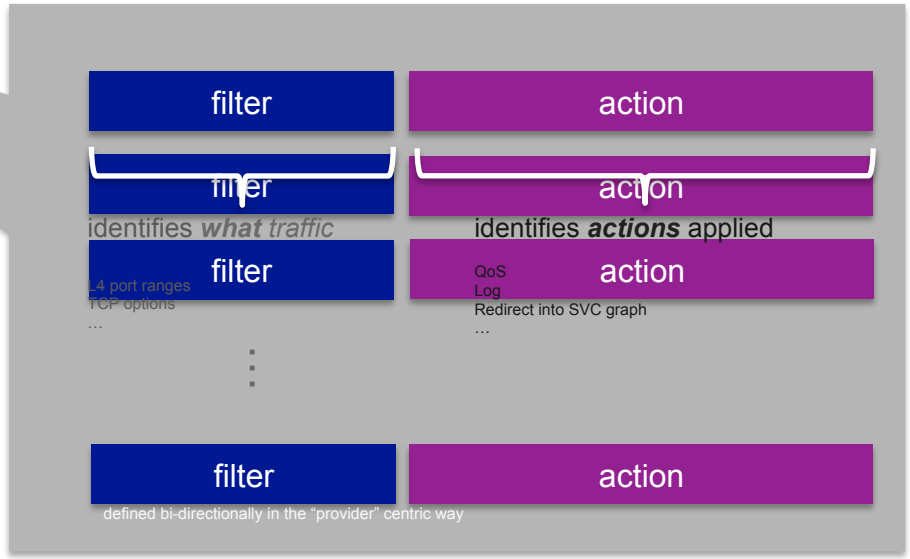
Cisco live!

What is APIC-DC

End Point ➜ End Points Groups ➜ Contracts



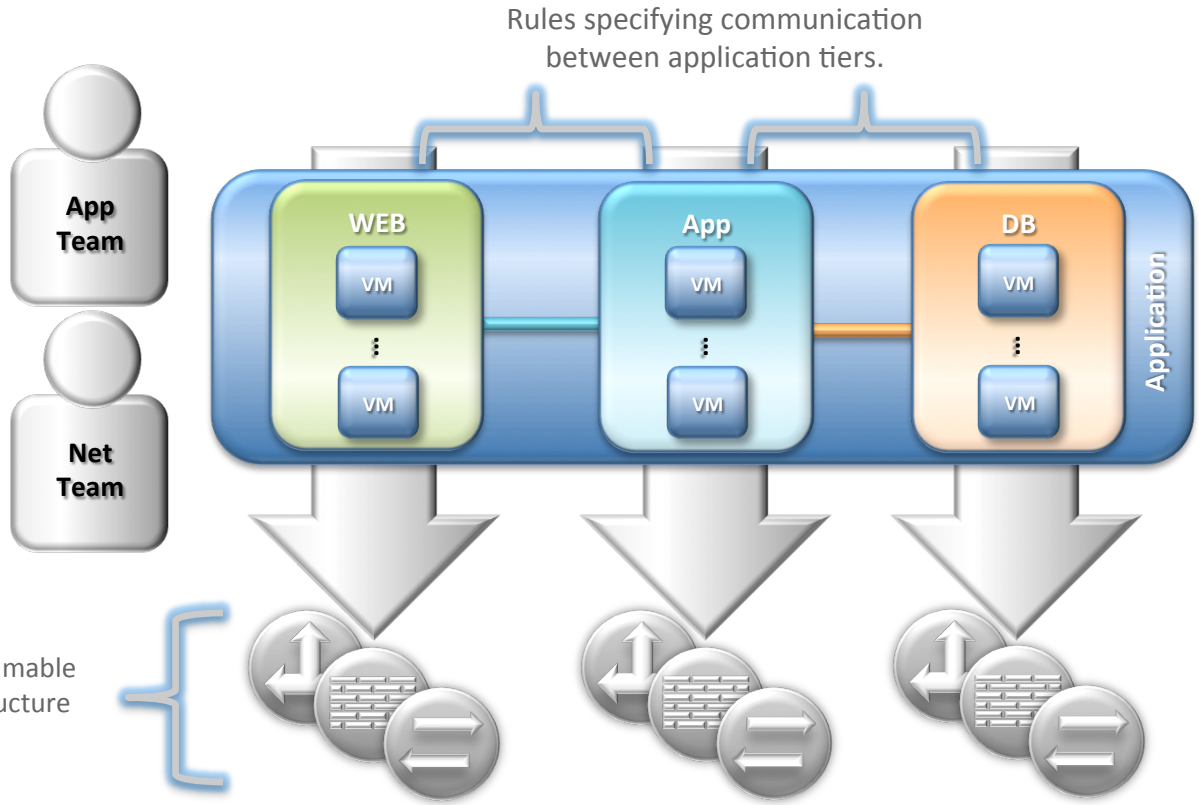
Contract specifies rules and policies on groups of physical or virtual end-points without understanding of specific identifiers and regardless of physical location.



Application Network Profiles

Applying Contracts between Application Tiers

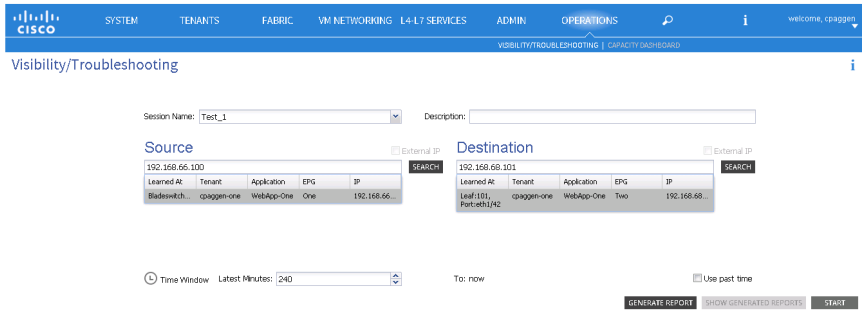
Application Network Profiles are a group of EPGs and the policies that define the communication between them.



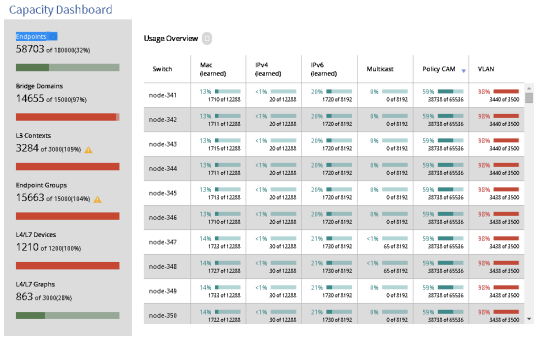
Cisco APIC Provides Full FCAPS

fault, configuration, accounting, performance, security

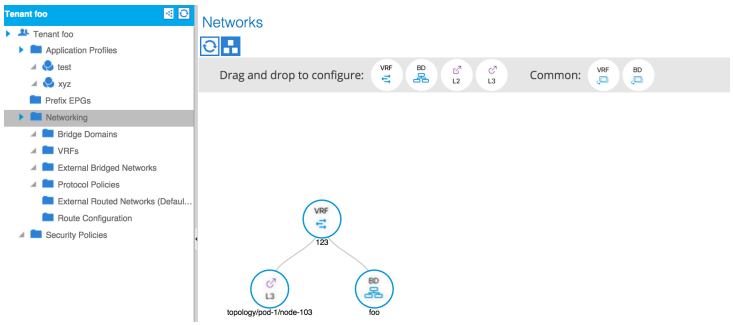
Troubleshooting Wizards



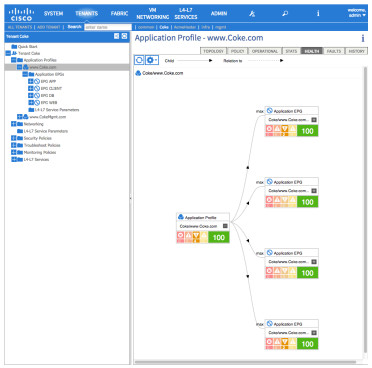
Capacity Dashboard



Drag and Drop Configuration

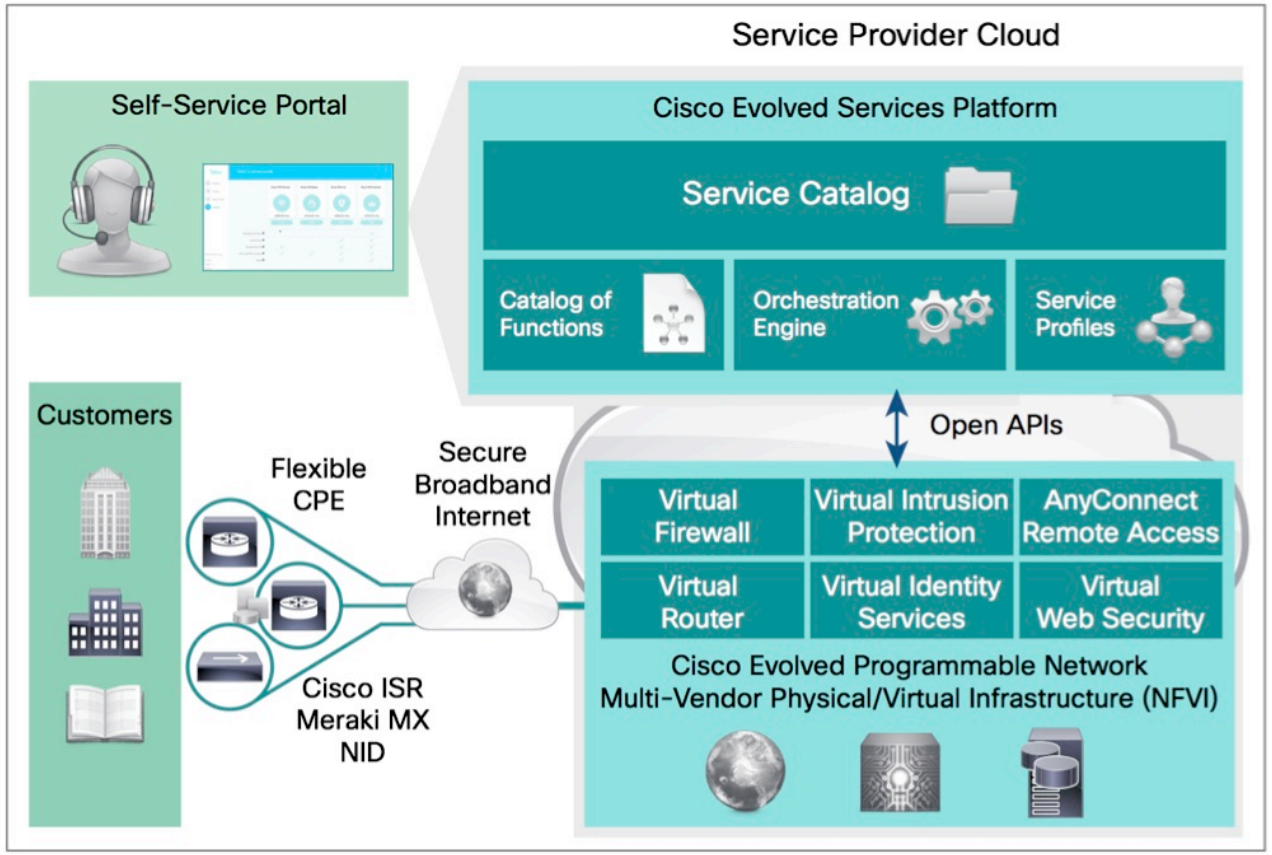


App Health Score

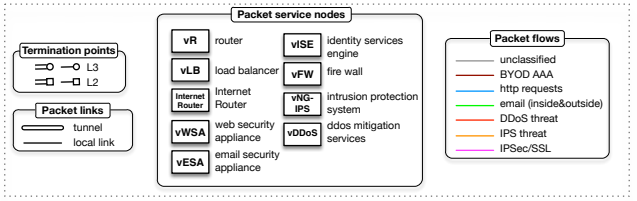


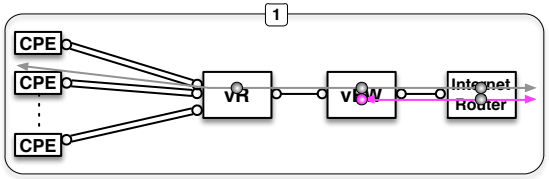
3.3 Virtual Managed Services
(aka Mozart – ESP – DSC - VMS)

Cisco Virtual Managed Services (VMS)



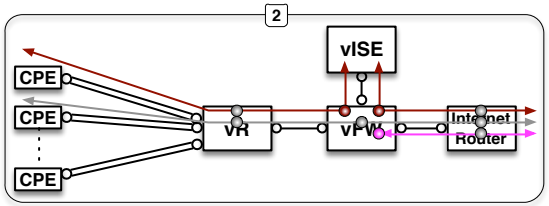
Flexible Service Chains





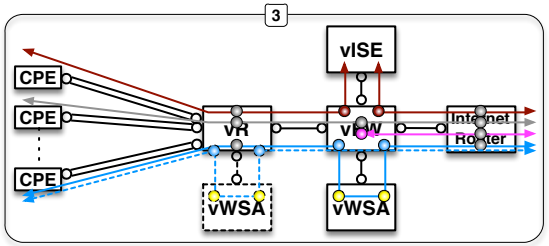
1

vIPVPN with FW and RA
- vFW with NAT and FW policy.
- vFW with IPSec/SSL Remote Access (RA) incl. remote end-host security posture verification.



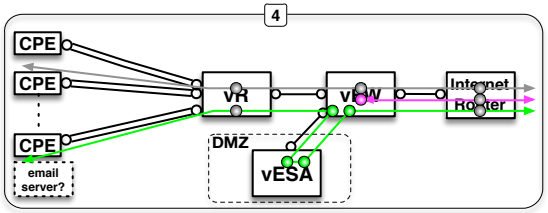
2

vIPVPN with BYOD, FW and RA
- vFW with NAT and FW policy.
- vFW with IPSec/SSL remote access incl. remote end-host security posture verification.
- vISE for BYOD svc auth (AAA, trust-sec label to IP binding)



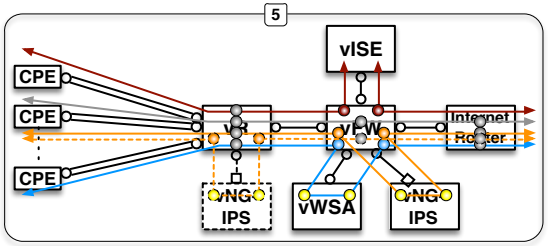
3

vIPVPN with BYOD, FW, RA, WebSec
- vFW with NAT and FW policy.
- vFW with IPSec/SSL remote access incl. remote end-host security posture verification.
- vISE for BYOD svc auth (AAA, trust-sec label to IP binding)
- vWSA for Enhanced Web Security



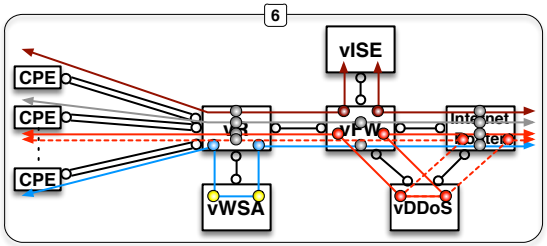
4

vIPVPN with BYOD, FW, RA, EmailSec
- vFW with NAT and FW policy.
- vFW with IPSec/SSL remote access incl. remote end-host security posture verification.
- vESA for Critical Information Protection (inbound and outbound Emails)



5

vIPVPN with BYOD, FW, RA, WebSec, ngIPS
- vFW with NAT and FW policy.
- vFW with IPSec/SSL remote access incl. remote end-host security posture verification.
- vISE for BYOD svc auth (AAA, trust-sec label to IP binding)
- vWSA for Enhanced Web Security
-vNG-IPS (SourceFire) for advanced threat protection and real-time contextual awareness



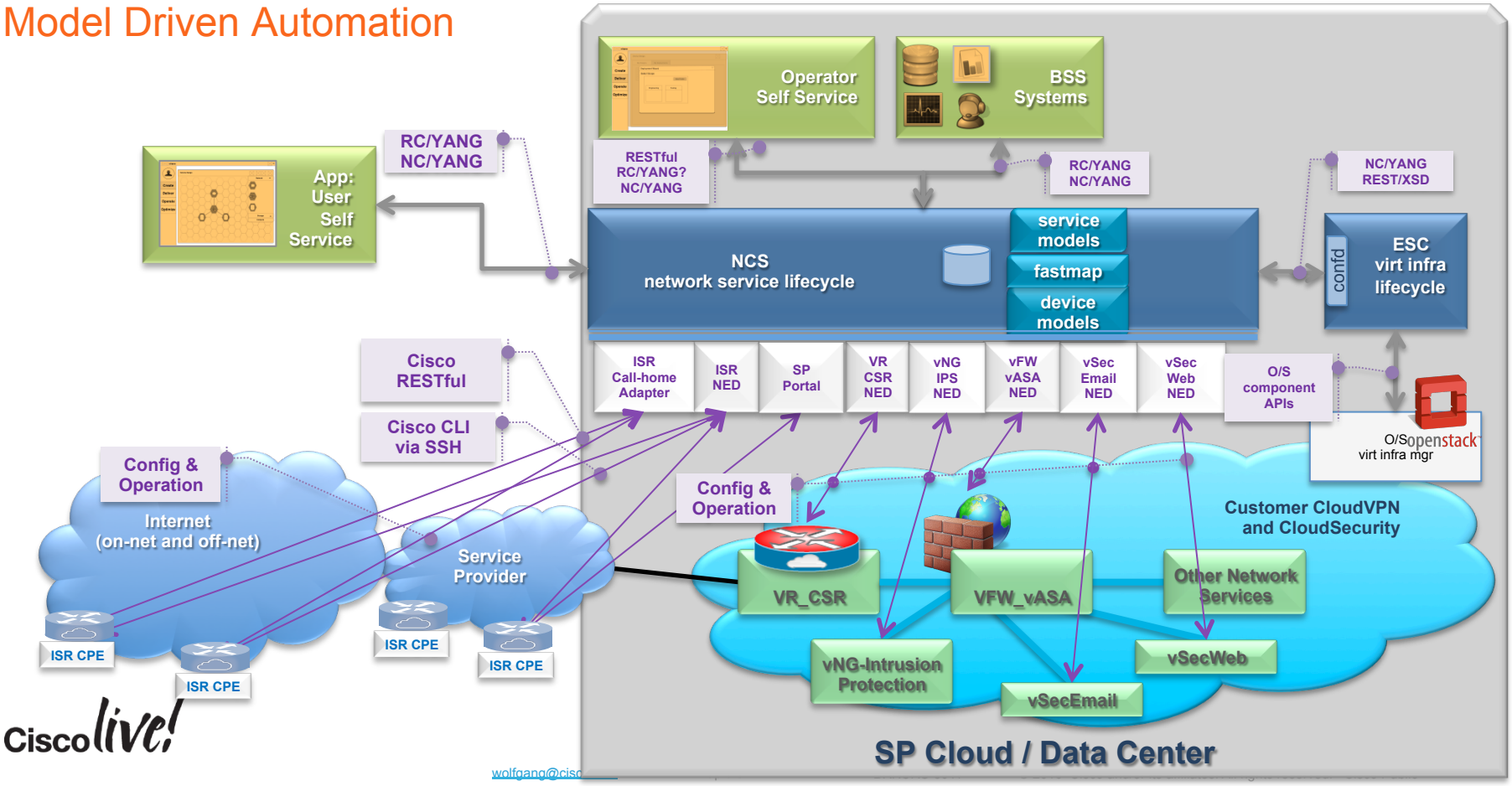
6

vIPVPN with BYOD, FW, RA, WebSec, DDoS
- vFW with NAT and FW policy.
- vFW with IPSec/SSL remote access incl. remote end-host security posture verification.
- vISE for BYOD svc auth (AAA, trust-sec label to IP binding)
- vWSA for Enhanced Web Security
- vDDoS (Radware DefensePro) for volumetric and application DDoS visibility and mitigation services

Cisco live!

Cisco VMS Service Delivery Workflow

Model Driven Automation

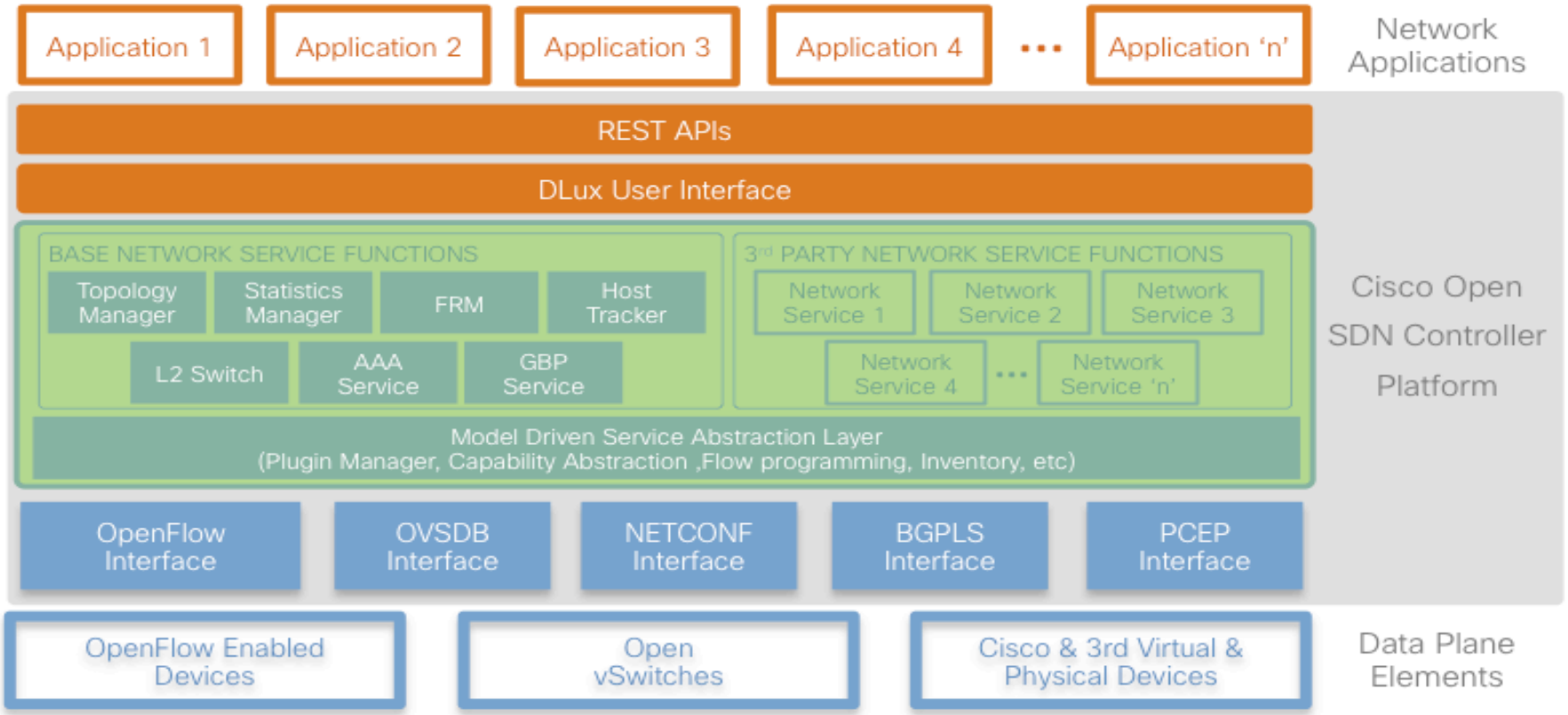


wolfgang@cisc

Cisco live!

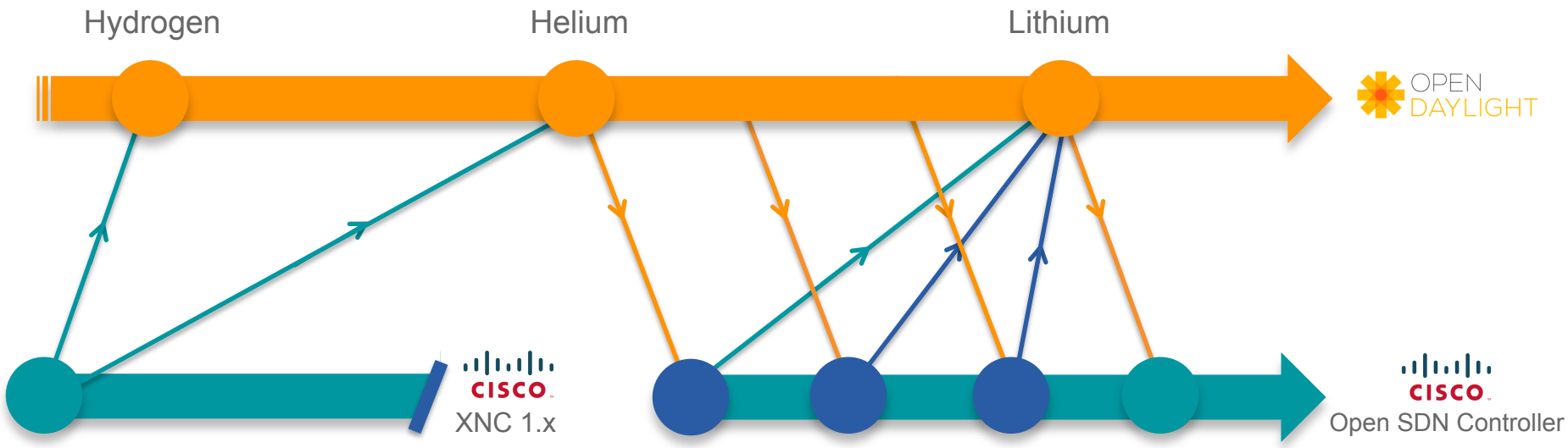
3.4 OSC - Open SDN Controller

CISCO - Open SDN Controller



CISCO - Open SDN Controller

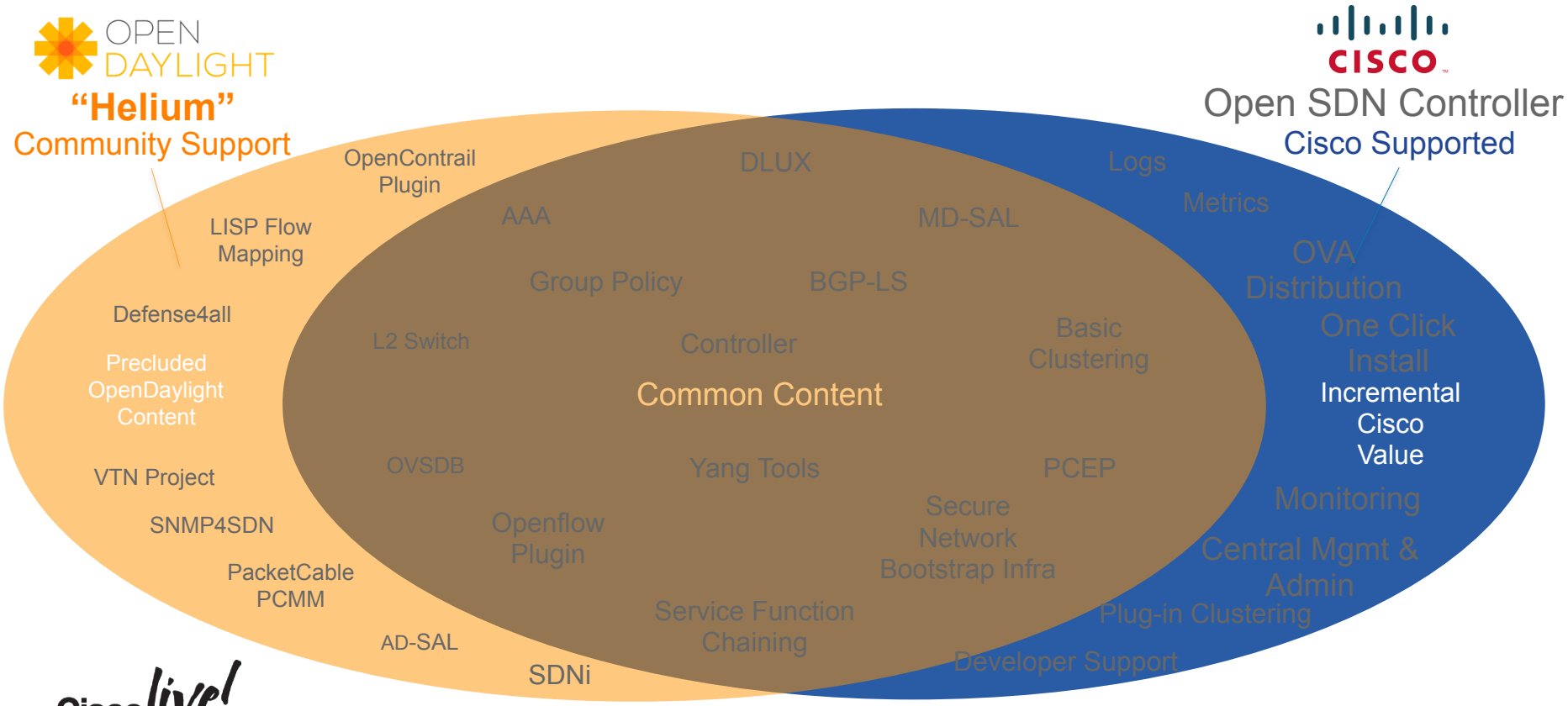
Re-bases XNC on OpenDaylight Helium Release



Ciscolive!

CISCO - Open SDN Controller

Open SDN Controller vs OpenDaylight Helium



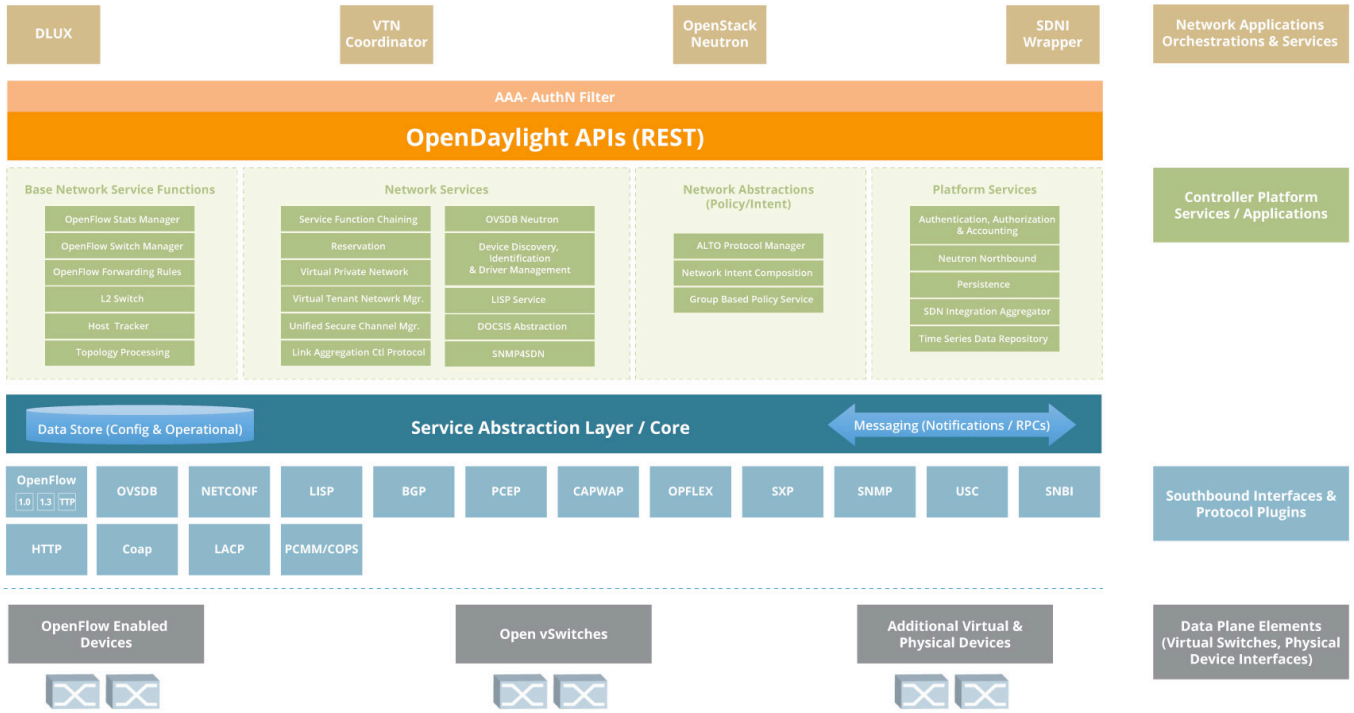
Cisco live!

3.5 ODL- Open Day Light Controller

OpenDaylight Platform

OPEN DAYLIGHT

Open Source SDN Platform
Lithium, 3rd Release, June 2015



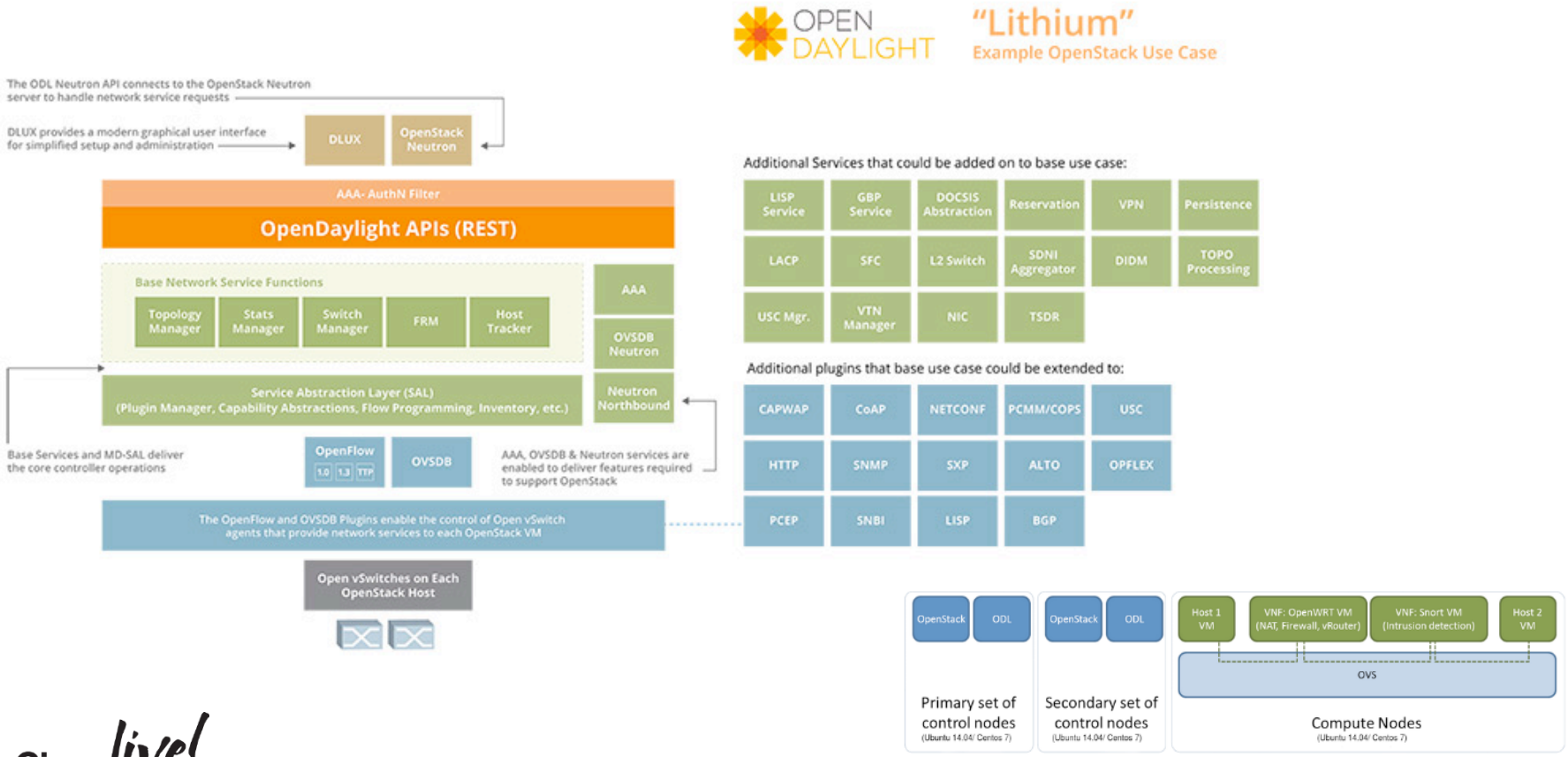
Hydrogen
- Released February 2014

Helium
- Released October 2014
- 1.87M+ lines of code
- 28 Projects
- 256 Contributors

Lithium
- Lithium-SR3 December 3, 2015

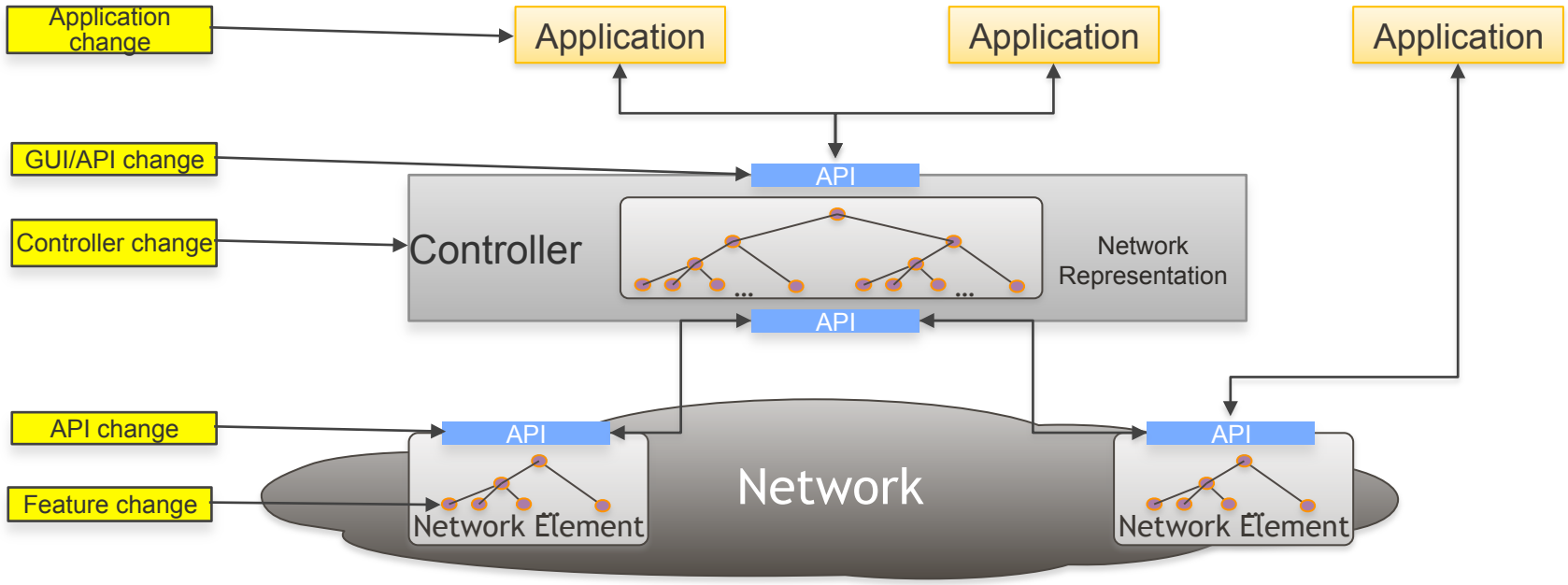
Cisco live!

OpenDaylight Platform



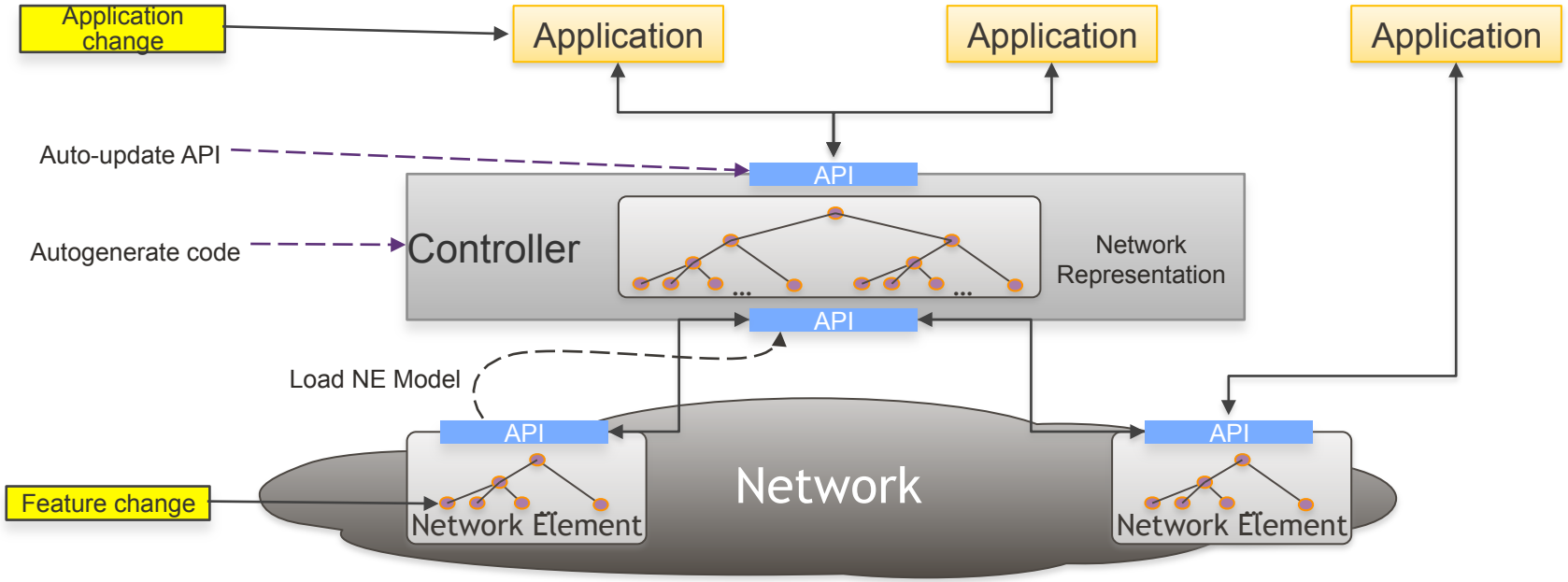
Network Application Life Cycle (Today)

Hop-by-Hop API-Driven Architecture



Network Application Life Cycle (Tomorrow)

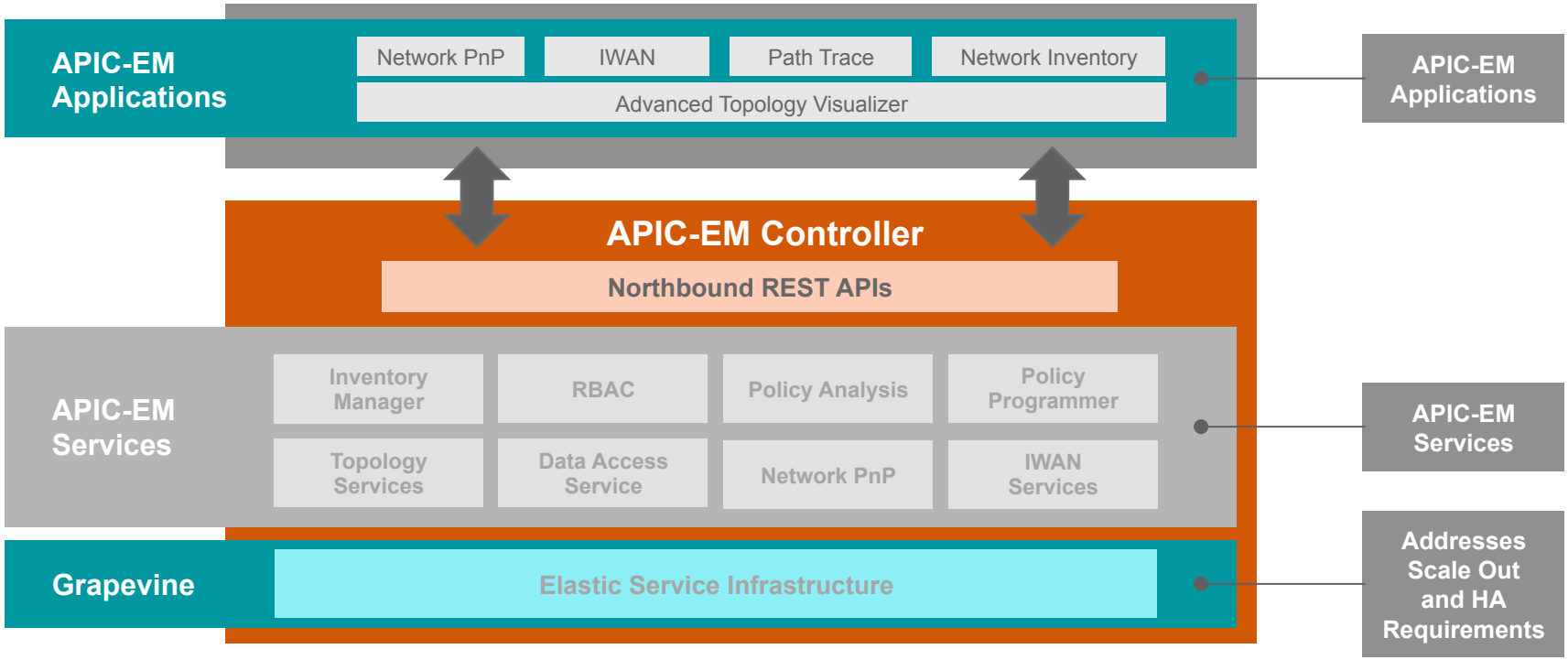
End-to-End Model-Driven Architecture



3.6 APIC-EM

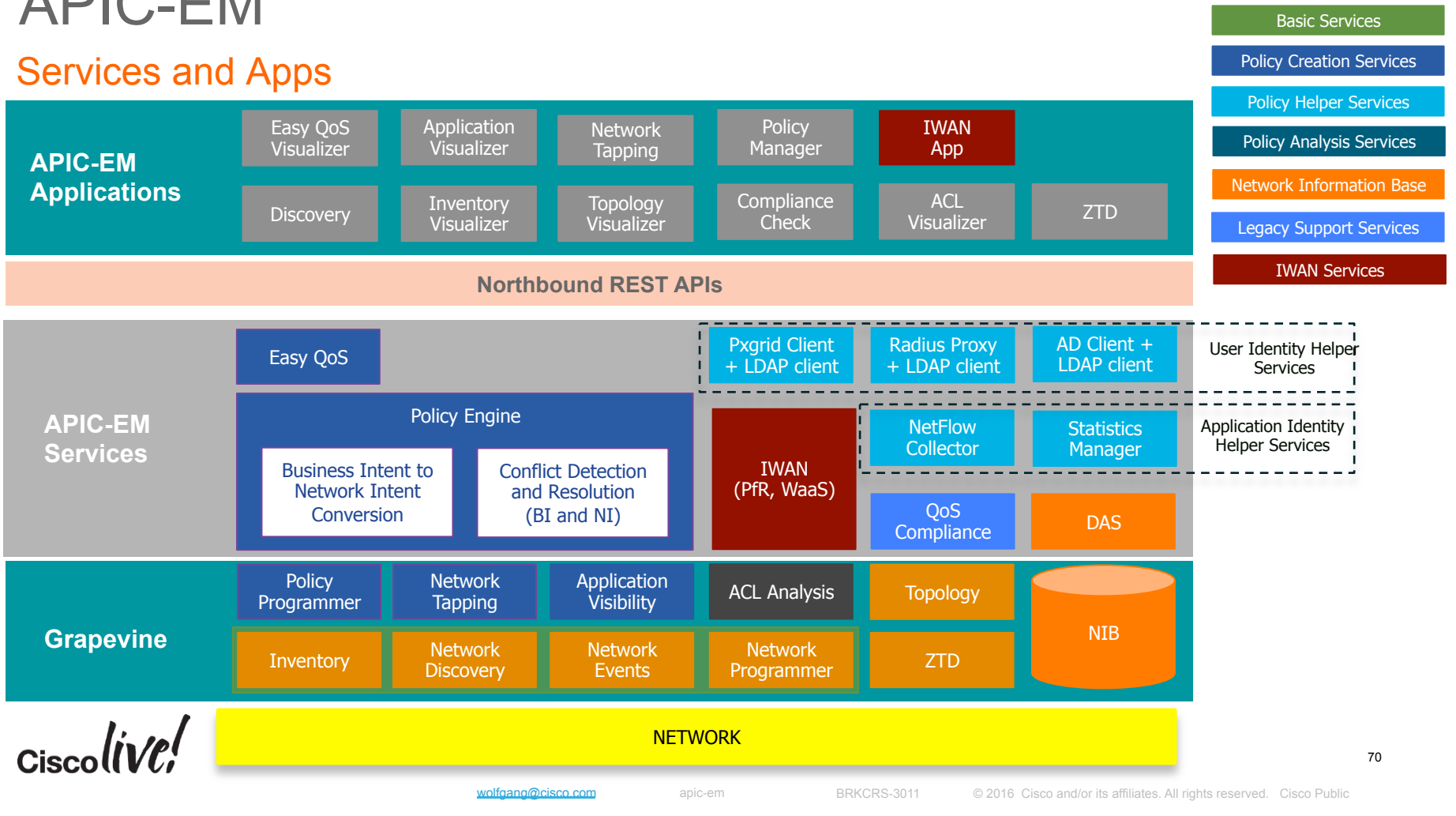
APIC-EM

Platform Architecture



APIC-EM

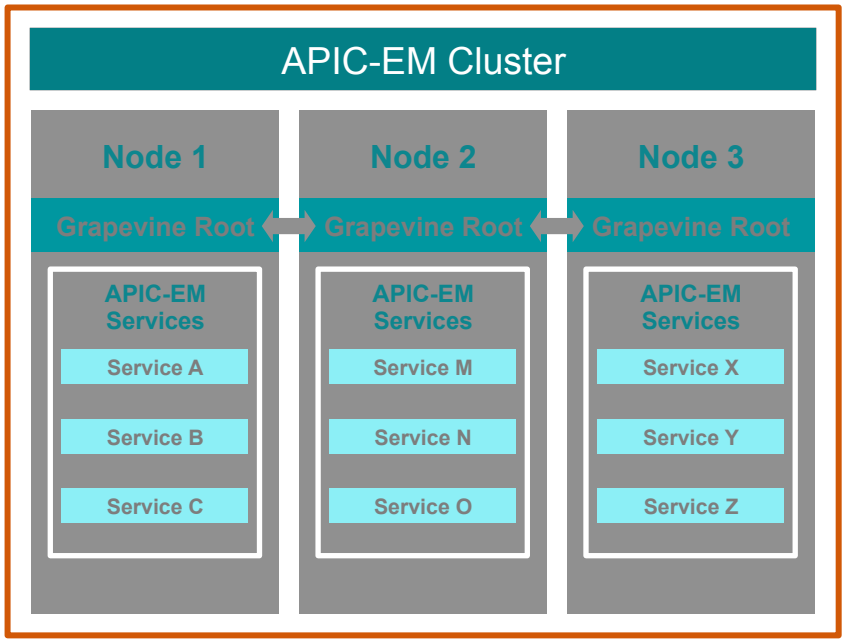
Services and Apps



APIC-EM

High-Availability (HA) Design

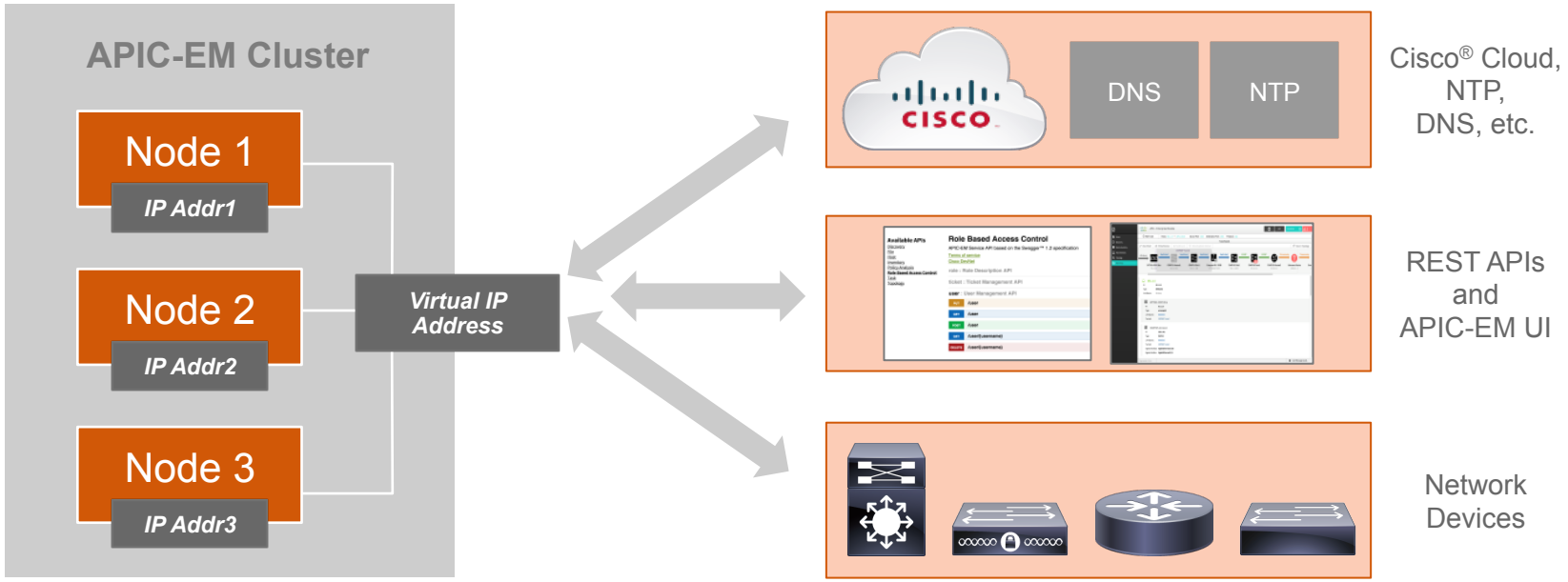
- Multiple instances of the GV root across different physical hosts and operating in **Active-Active** mode for optimal performance, load-sharing, and high availability
- Data persistence layer that has instances spread across different physical nodes; provides support for HA and scale
- Non-HA deployment (single/dual hosts):
  - Supports SW failure (APIC-EM services)
  - No support for HW (host) failure
- HA deployment (3 hosts):
  - Supports SW failure (APIC-EM services)
  - Supports HW failure of single host



Cisco live!

APIC-EM

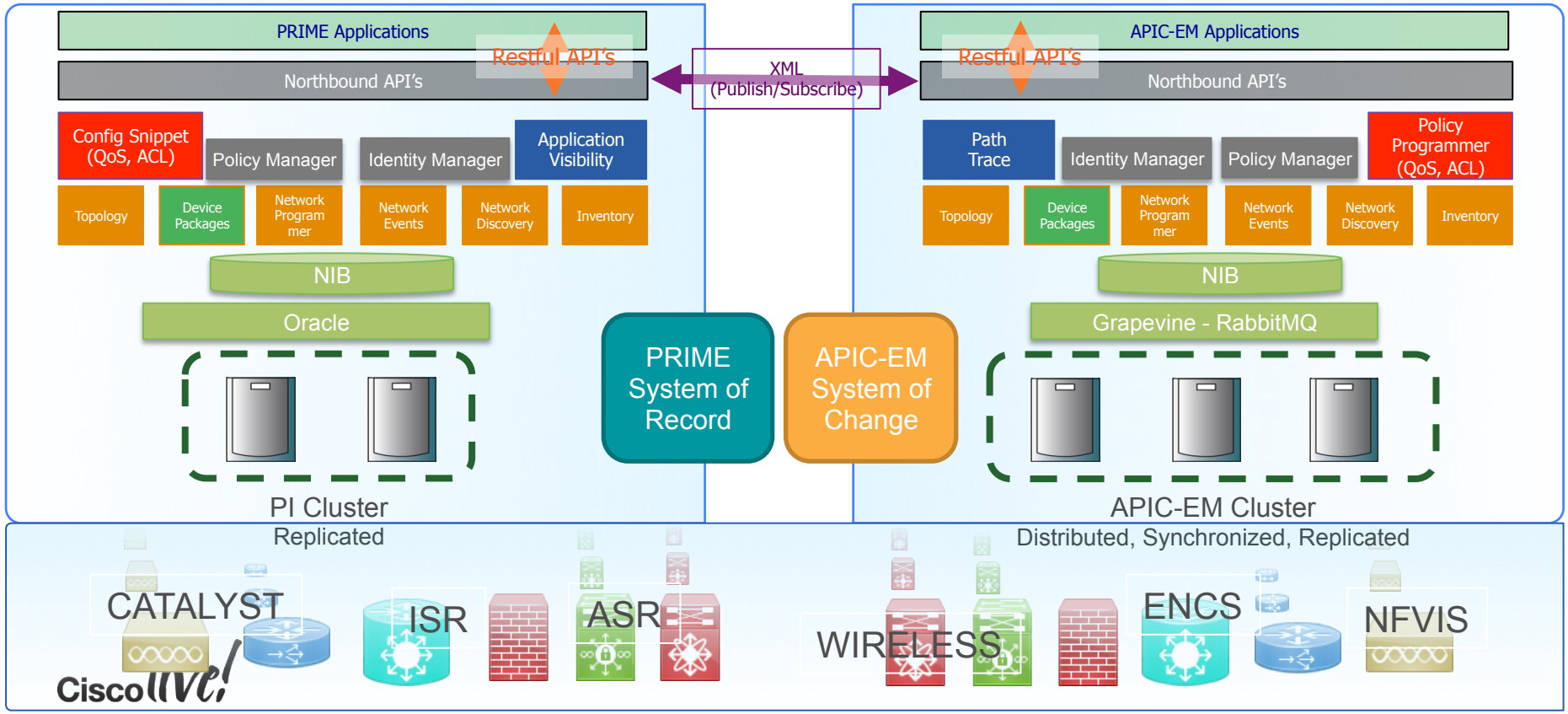
Multi-Host Deployment



3.7 PI and APIC-EM

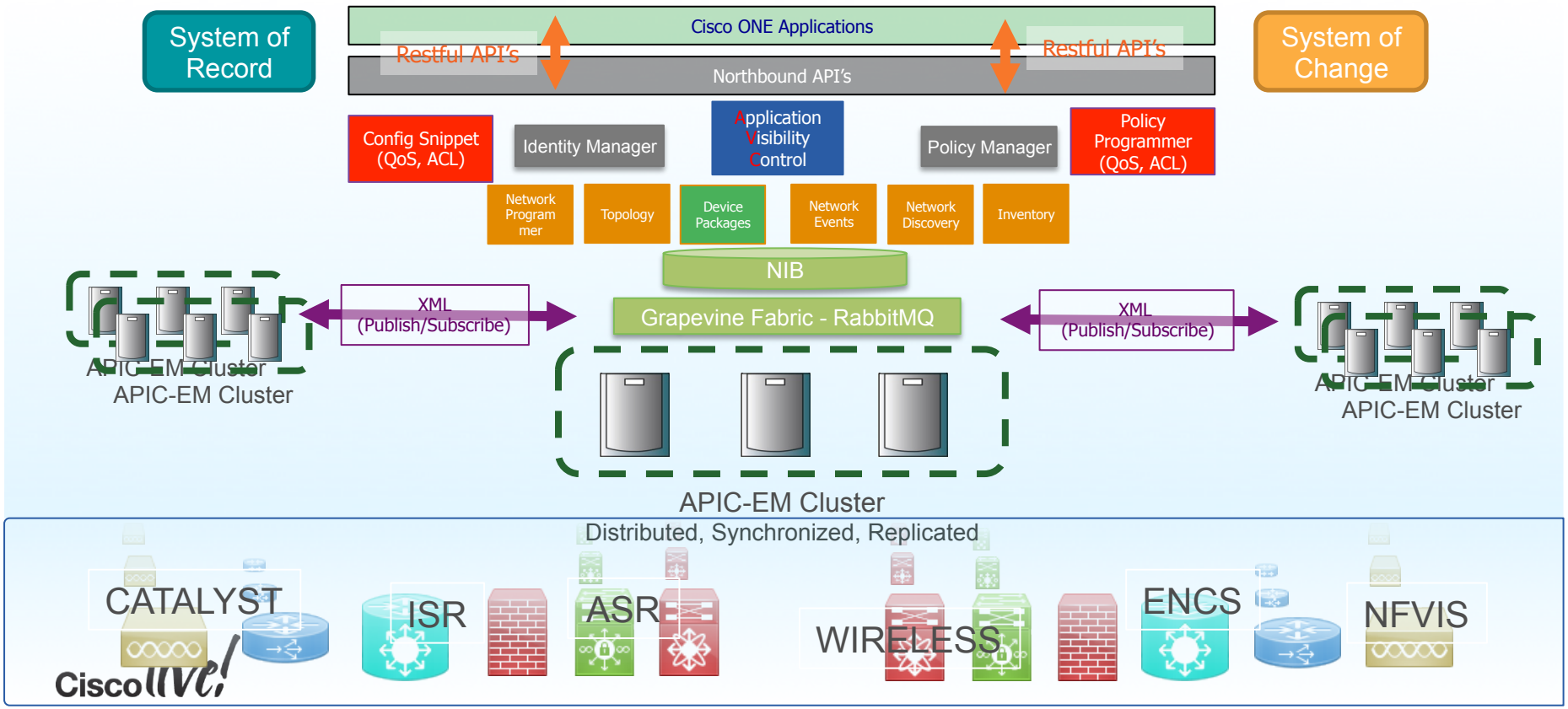
Prime Infrastructure and APIC-EM - Today

East-West API and Work Flow Integration



Prime Infrastructure and APIC-EM - Tomorrow

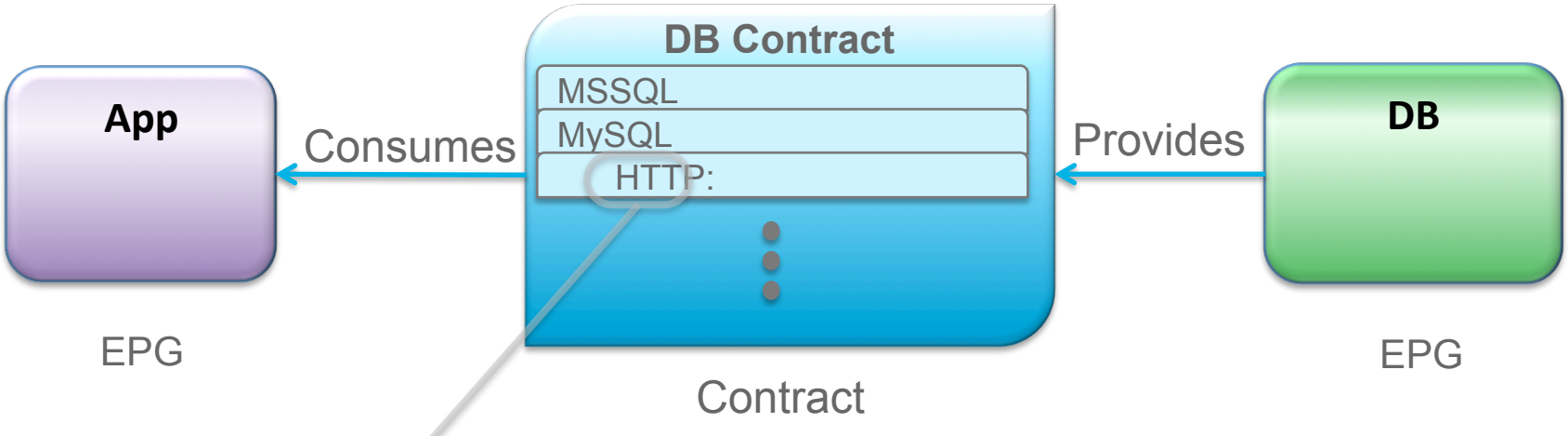
Wolfgang's view: Common Services with Common northbound App's and API's



4.1 APIC-EM - Policy Infrastructure

APIC-DC Policy Model

Recap: EPGs and Contract



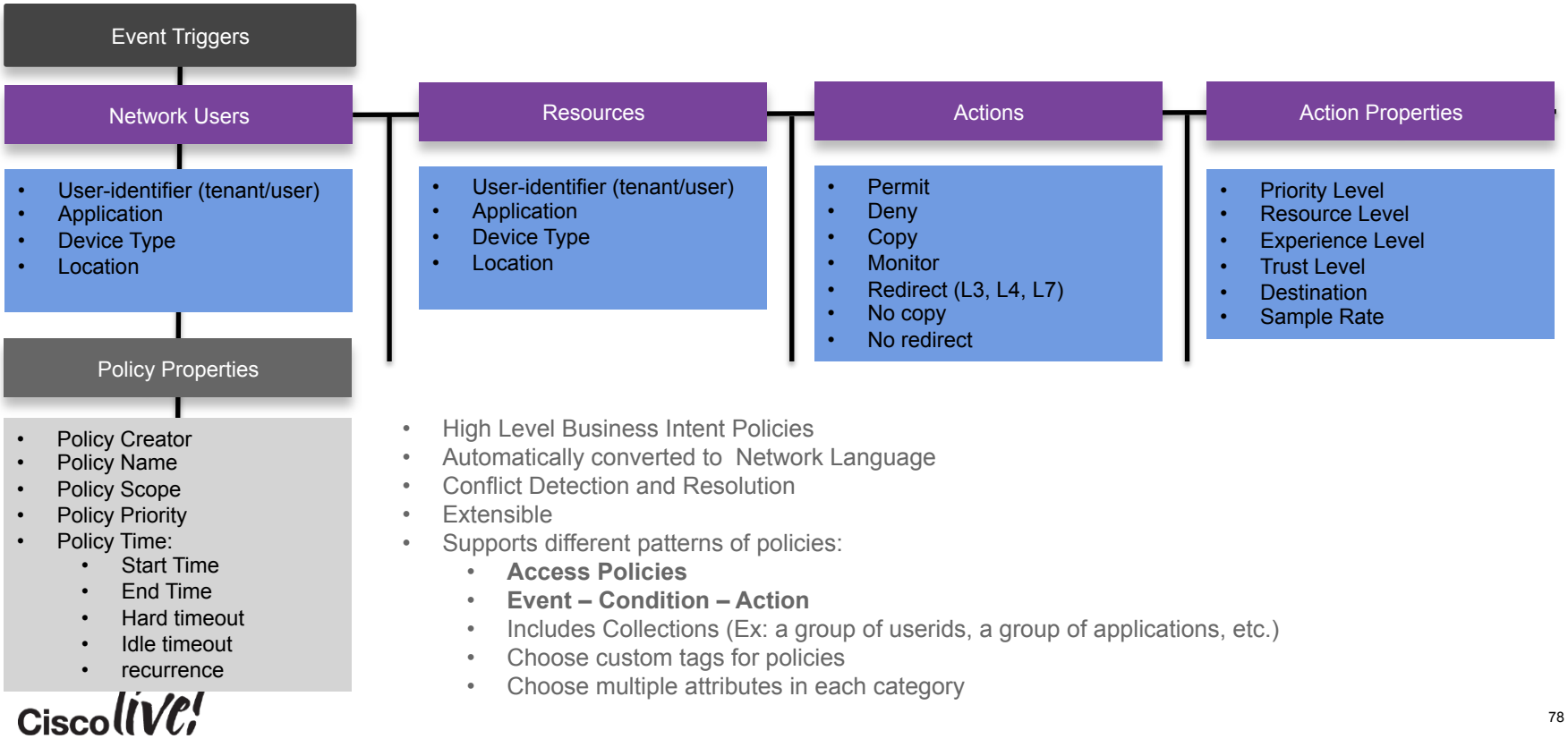
Filter

Named collection of L4 port ranges
- HTTP = [TCP],[80, 443]
- MSSQL = [UDP],[1433-1434]
- MySQL = [TCP],[3306, 25565]

ACI Model will be extended for APIC EM Utilization

Cisco live!

APIC-EM Policy Construct



- High Level Business Intent Policies
- Automatically converted to Network Language
- Conflict Detection and Resolution
- Extensible
- Supports different patterns of policies:
  - **Access Policies**
  - **Event – Condition – Action**
  - Includes Collections (Ex: a group of userids, a group of applications, etc.)
  - Choose custom tags for policies
  - Choose multiple attributes in each category

APIC-EM

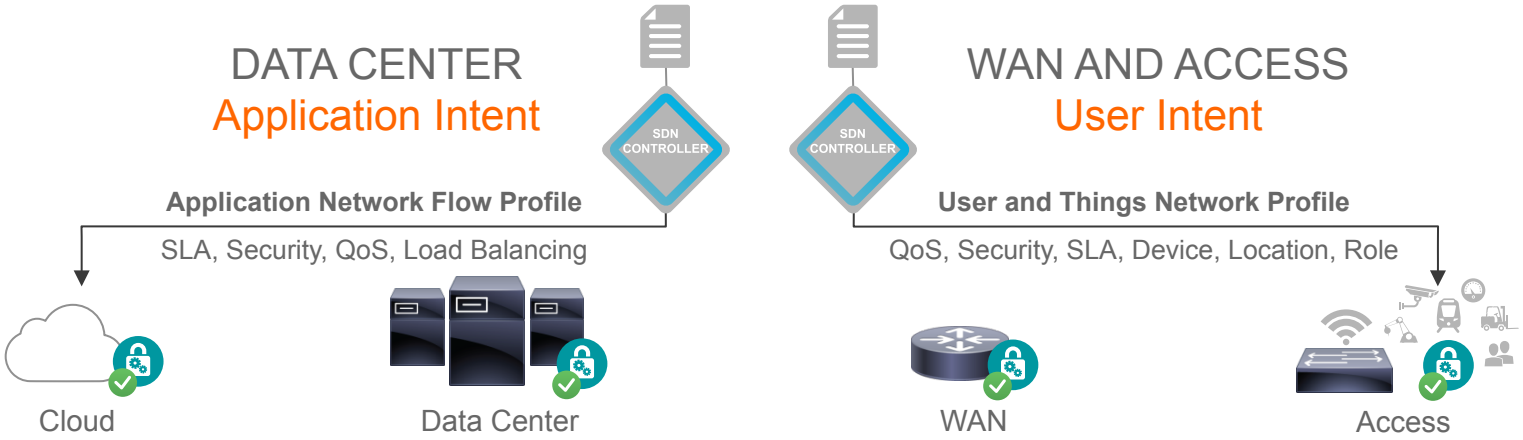
Extensions for Enterprise use cases

- Accommodation for Groups
  - ✓ Every EP is part of multiple groups in real-life
  - ✓ Groups are sometimes overlapping
  - ✓ Groups could be defined from multiple context-attributes
- Finer grain access
  - ✓ involves combination of consumer EP attributes and producer EP
  - ✓ implies overlapping rules. Resolution TBD
- Contract extensions
  - ✓ Need to extend contracts to include DPI-based application/groups.
  - ✓ Need rich set of actions such as Permit, Monitor, Permit with Warning, etc.
  - ✓ Actions include additional rule profiles such as: IPS-profile, File-filter-profile, QOS-profile etc.
- Question about implicit deny:
  - ✓ explicit ‘permit’ action
  - ✓ explicit ‘deny’ action

APIC-EM

Common Policy Model from Branch to Data Center

POLICY (Common Namespace for Business Intent)

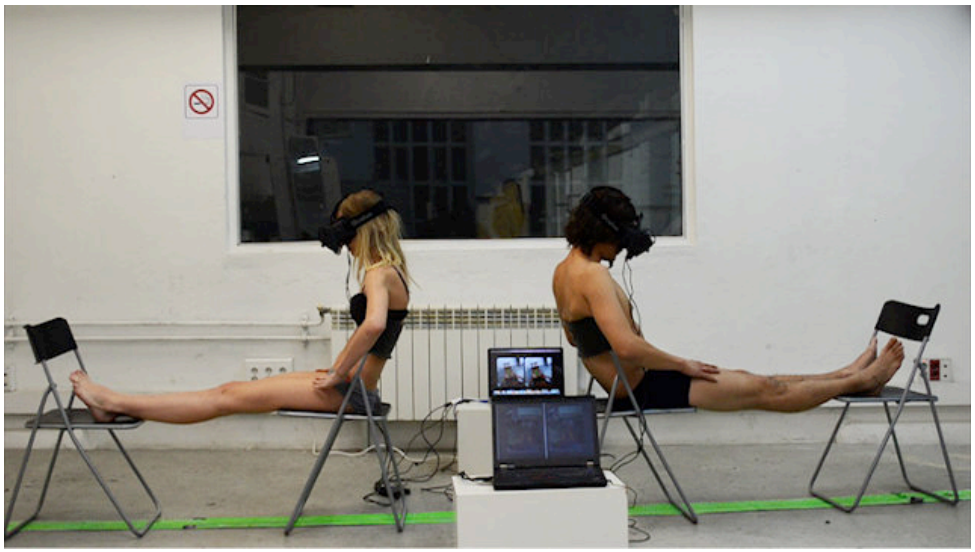


CISCO® ADVANTAGE

BROWNFIELD AND GREENFIELD	END TO END	POLICY FRAMEWORK: FOCUS ON APPLICATION AND USER ENABLEMENT

Common End-Points

End 2 End Communication, do we talk?



Hey, I meant from a policy Intent point of view!

4.2 APIC-EM – Auto Scale Architecture

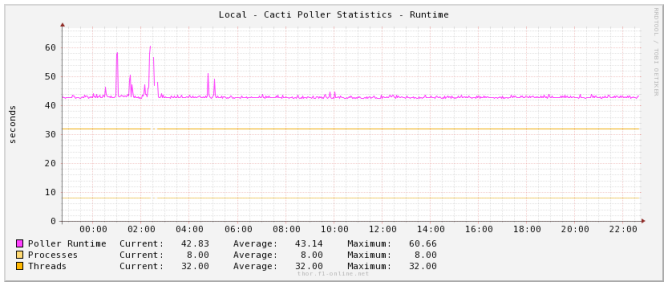
APIC-EM Grapevine

Why do we need a "Platform for Service Elasticity”?

- **Distributed service behavior is both unpredictable and dissimilar**
- A "one size fits all" approach to service scaling and management lacks the comprehension to manage both, the autonomic and bespoke requirements of a service ecosystem.
- Service groups can be managed by monitoring the container (the virtual machine)
- Events as common as log overflows, memory leaks, and runaway processes will quickly fool any system lacking both service introspection and strong policy into generating all of the classic distributed system failure conditions: storms, flaps, unmanaged contention, and deadlocks.
- Services themselves require support for:
  - **specialized policies for scaling in both directions**
  - inter-instance communication for building quorum and consensus on scale events
  - unified security for access and authorization
  - unified model and data views for elements managed by multiple services

Remember Cacti – Spine – Poller issues?

```
output: Time:42.6984 Method:spine Processes:8 Threads:32 Hosts:79
HostsPerProcess:10 DataSources:8985 RRDsProcessed:2616
Poller[0] Maximum runtime of 58 seconds exceeded. Exiting.
```



Cisco live!

APIC-EM Grapevine

What is Grapevine?

- **Grapevine** and **APIC-EM** are **de-coupled** from a technical perspective.
  Grapevine is the horizontal scale *platform* on which *services* such as those for APIC-EM run
- **Cisco groups** wanted to create a new solution XYZ (that was completely unrelated to APIC-EM) that needed scale, HA, rolling-upgrades, service life-cycle management, etc... could **use Grapevine** (as long as they adhere to the Grapevine service design requirements) without needing to deploy/use any of the APIC-EM services

- Is a PaaS (**Platform as a Service**) with an associated SDK
- SDN developers can use to write their "services" (similar to a Google AppEngine or VMware Cloud Foundry model).
- Is a simplified refinement of the PaaS model provided by both Amazon and Google for their cloud services. While you can run any program you like on their IaaS, using the PaaS requires adherence to a framework.
- The major difference is that Grapevine **introspects at the service level and autoscales at the VM level** rather than breaking scaled resources down to the level of compute, block storage, network, etc.
- It is important to note that Grapevine **controls elasticity at the granularity of "services"** rather than at the more coarse-grained, **virtual machine** granularity.
- You can run Grapevine on bare-metal as also within VM's or in a mix of physical and virtual machines
- Advantages of controlling elasticity at the service granularity are:

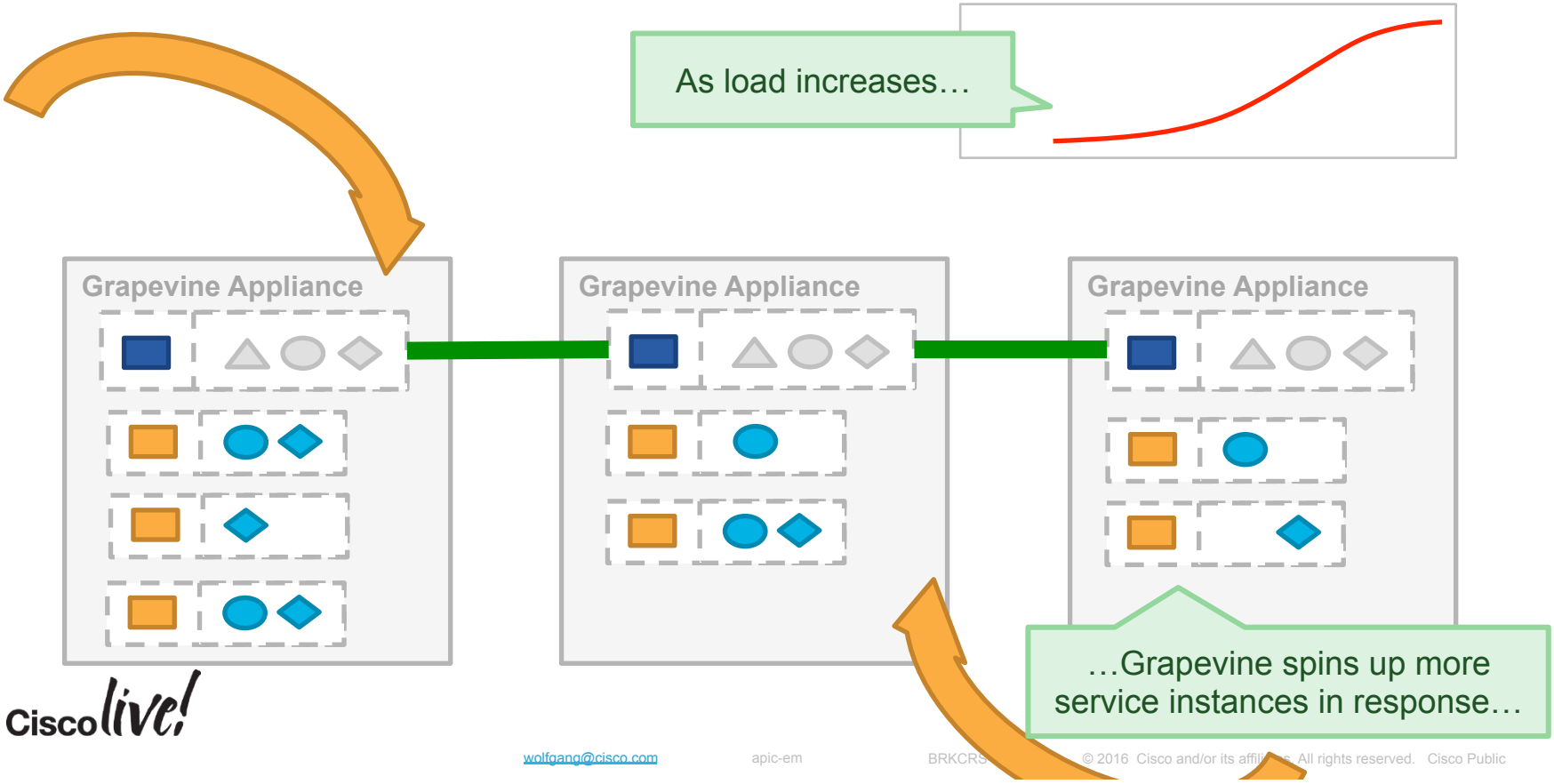
- ✓ Avoids **VM boot up / shutdown time** in the
- ✓ Better determine whether or not a service is **indeed healthy and is working as expected** vs just knowing whether or not a VM is running or not
- ✓ **Better utilize** a VM's capacity by running instances of different services within the same VM instance
- ✓ **Perform service-specific monitoring** to better determine whether an instance is "under heavy load"

Cisco live!

APIC-EM Grapevine

Automatic Scaling

Grapevine
Platform for Service Elasticity



Cisco live!

APIC-EM Grapevine

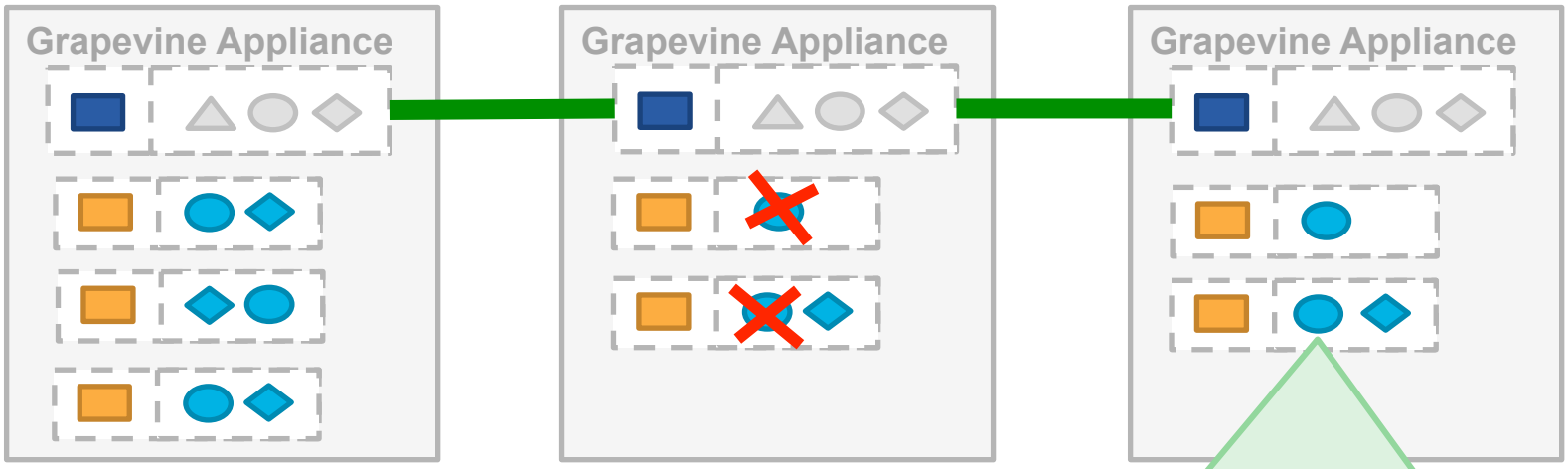
Grapevine, the 20,000 foot view

- With Grapevine you would define "service bundles".
- Each "service bundle" deployed runs as a separate process.
- Can deploy a single instance of these services or multiple instances of these services, on the same server or across multiple virtual as also physical servers.
- You can add, remove, start, stop, update these services at runtime without downtime
- Services can be written in pretty much any programming language (Java, C/C++, Go, Python, Ruby, Perl, Tcl, Bash, etc) and would communicate with each other via remote able APIs based on HTTP, AMQP, Thrift, etc.
- Given this, you can easily deploy services like OSGi within Grapevine
- Grapevine will monitor the load of these services
- Grapevine will provide scale for these services
    - In the presence of increased load, Grapevine will "grow" multiple instances of the services to provide horizontal scale.
    - In the presence of decreased load, Grapevine will "harvest" service instances
- Grapevine will provide HA for these services. In the presence of software/hardware failures Grapevine will grow replacement service instances to take over the workload of those instances that have failed
- Grapevine will provide "rolling upgrades" for these services.
    - You can deploy new services, or updates to existing services to the cloud.
    - Grapevine would periodically poll the cloud for updates and would download and deploy them onto the Grapevine cluster when they're available with minimal to no downtime.

APIC-EM Grapevine

High Availability

Grapevine
Platform for Service Elasticity

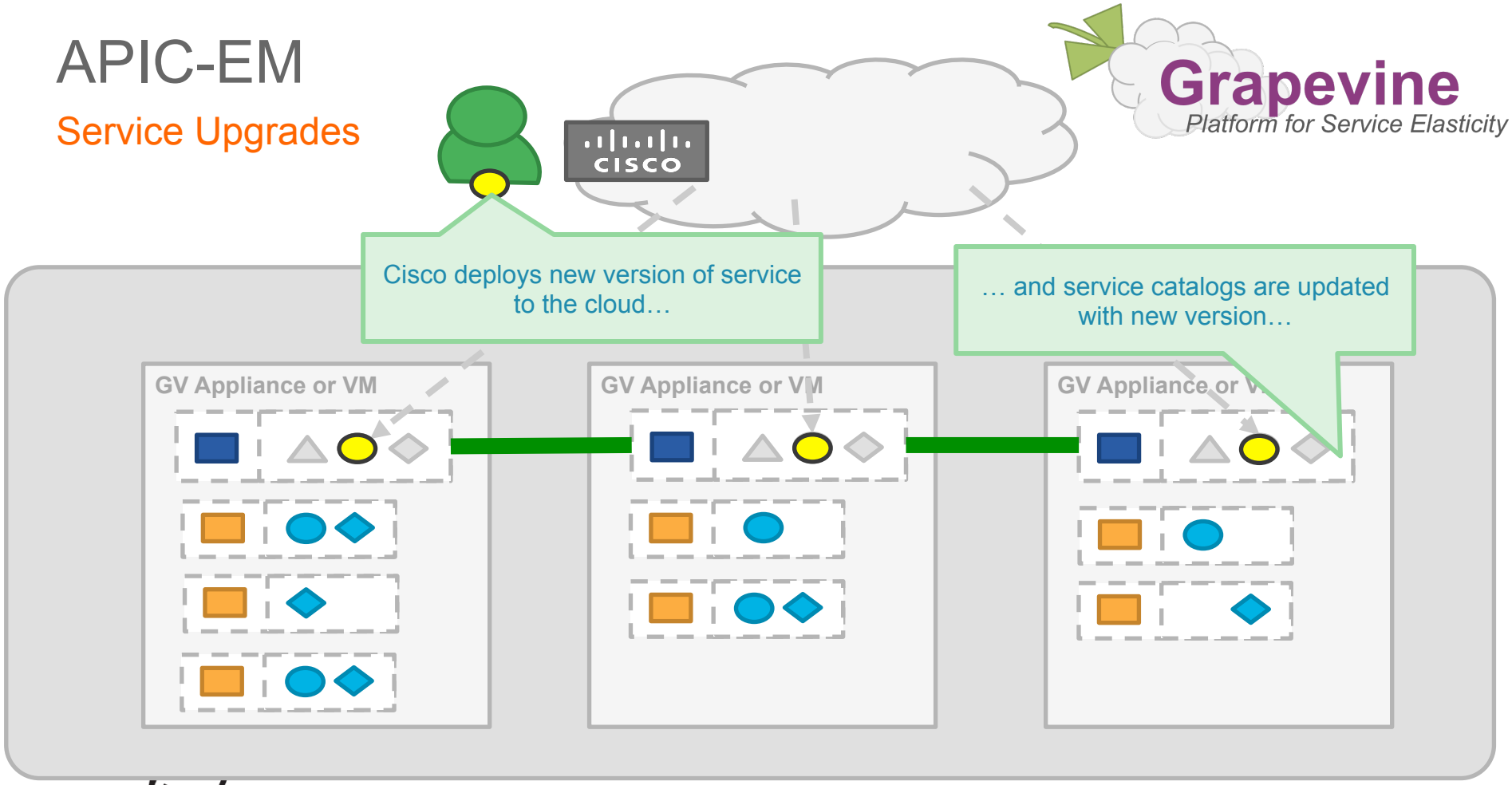


When a service fails, Grapevine starts a replacement instance, ensuring service's "min instance count" requirements are maintained…

Cisco live!

APIC-EM

Service Upgrades

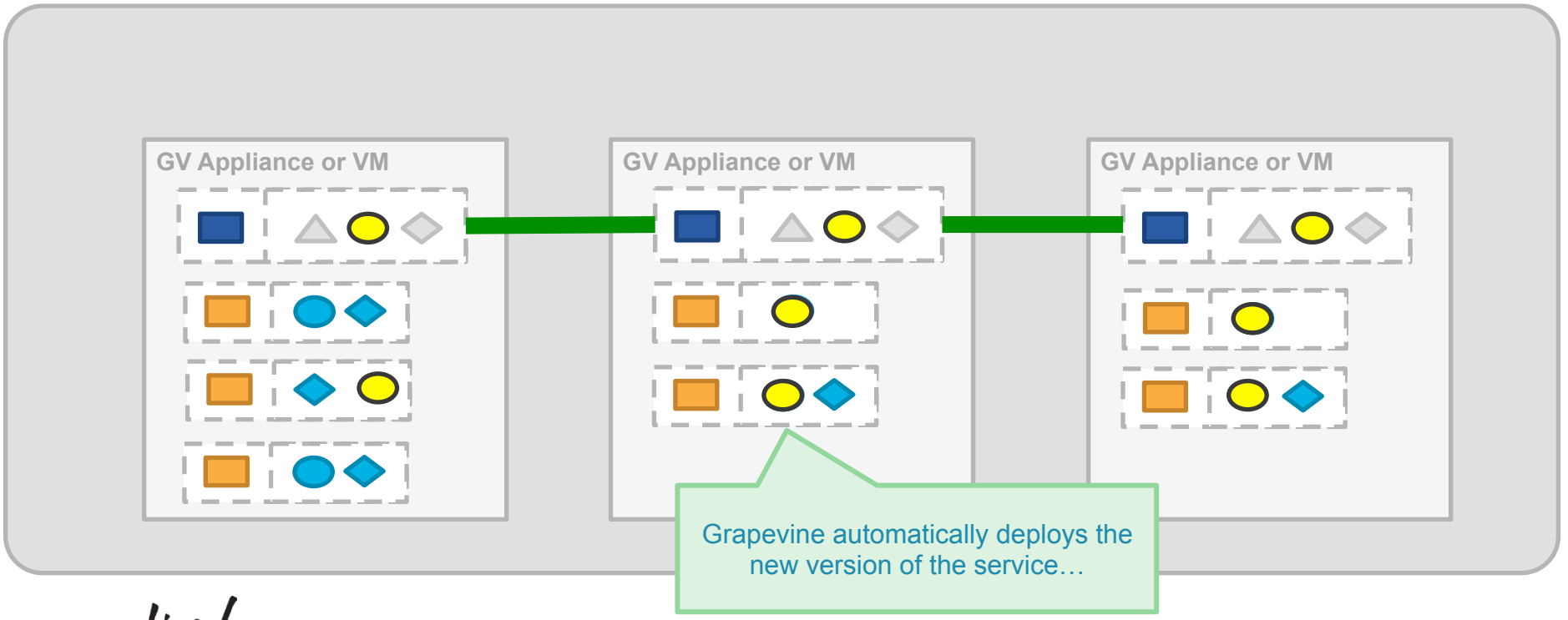


Cisco live!

APIC-EM

Service Upgrades

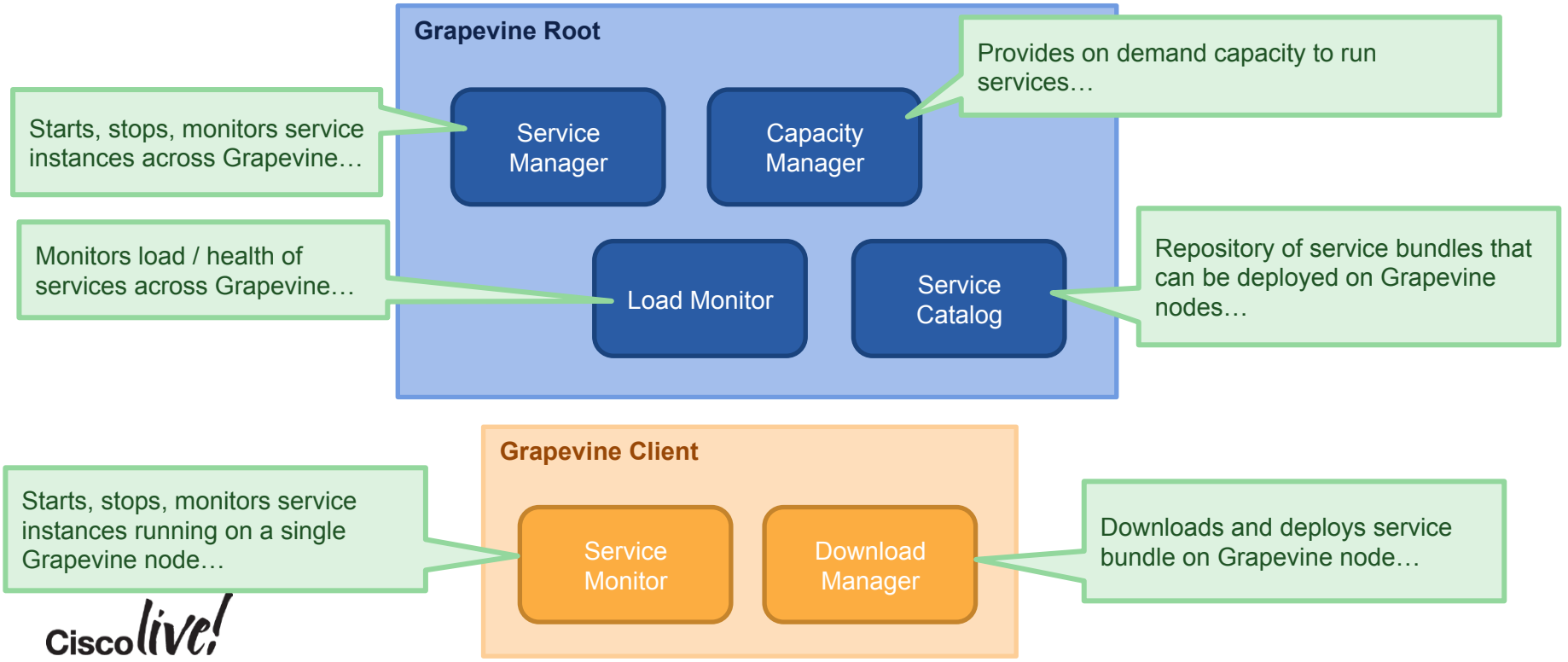
Grapevine
Platform for Service Elasticity



Cisco live!

APIC-EM Grapevine

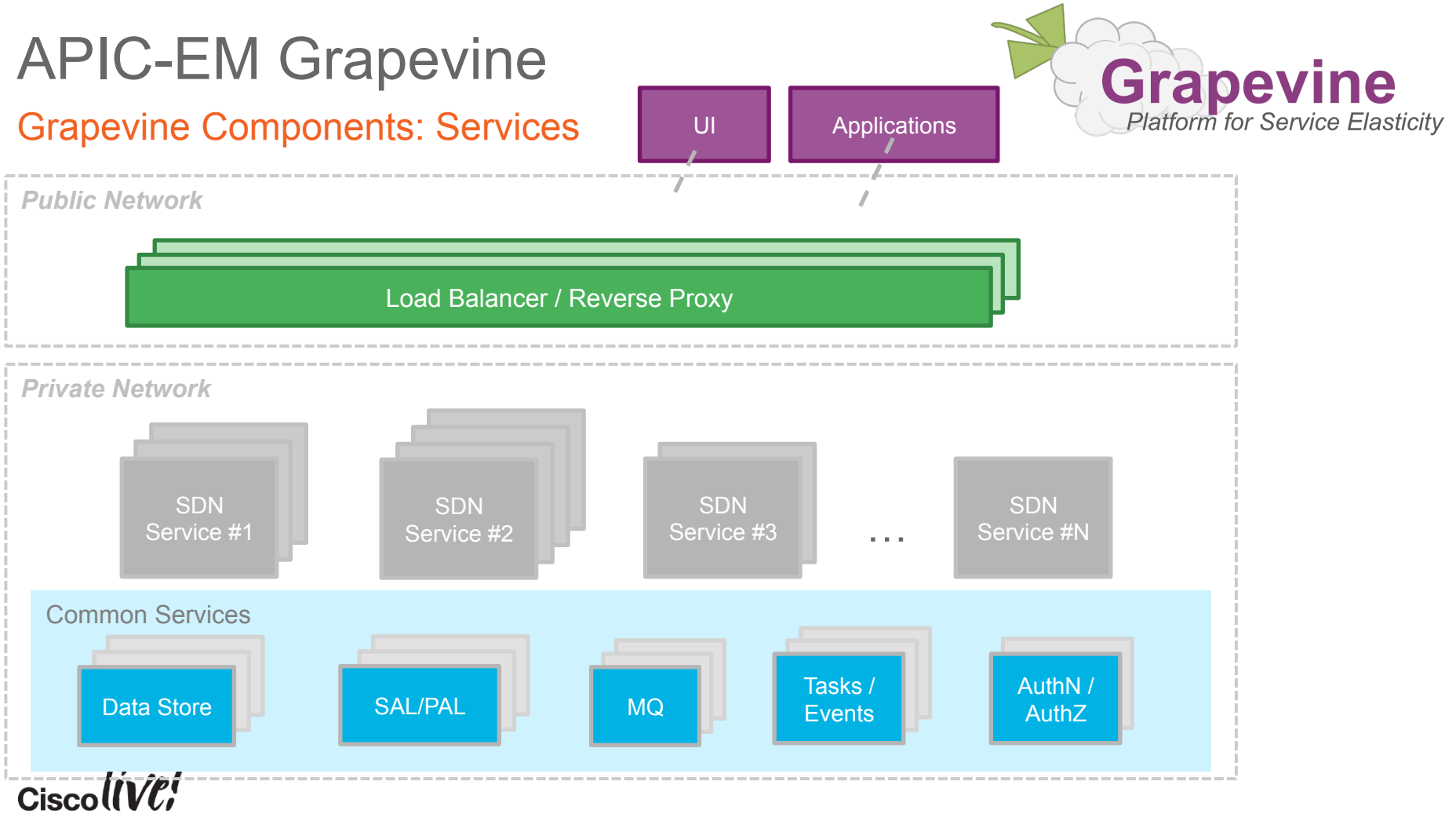
Grapevine Components: Grapevine



Cisco live!

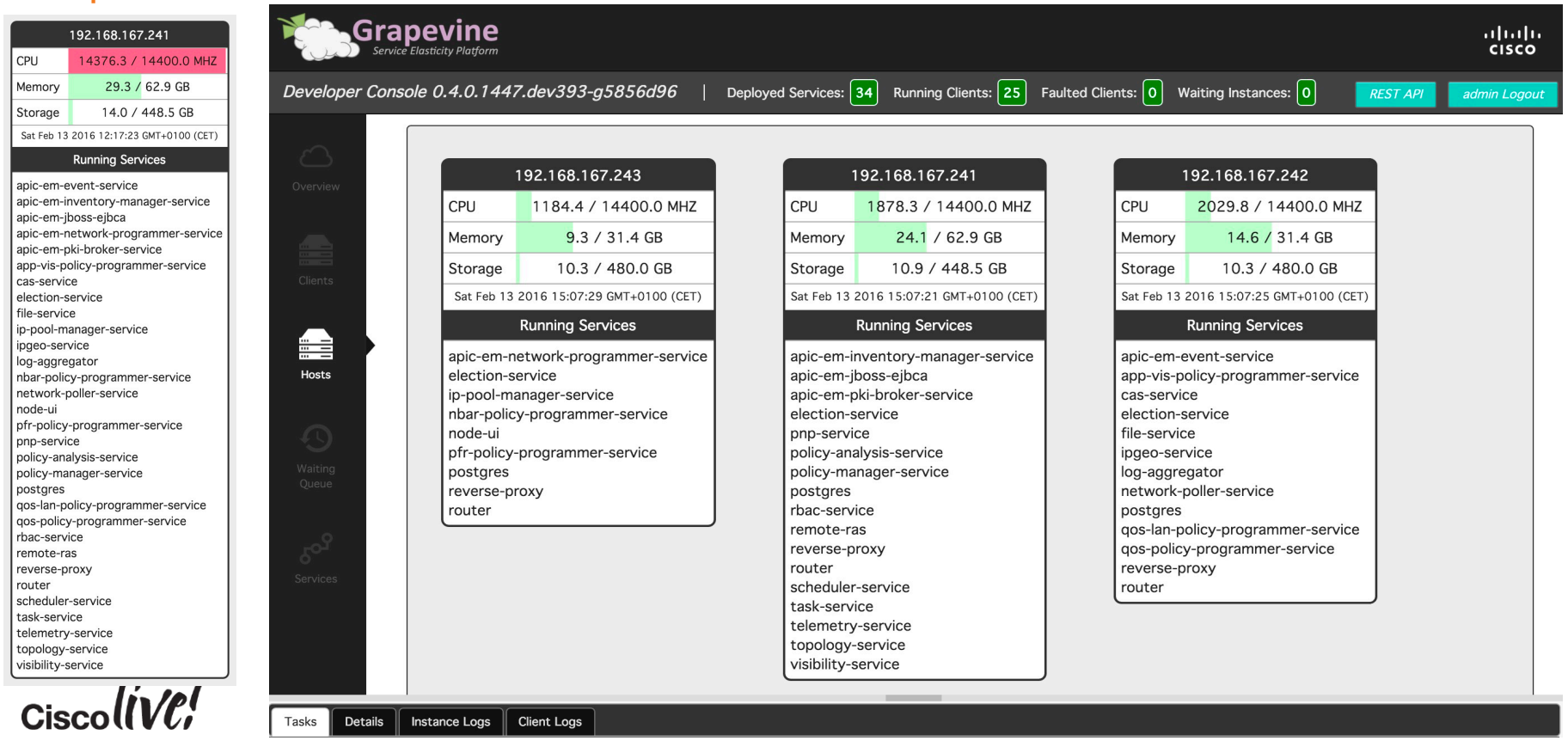
APIC-EM Grapevine

Grapevine Components: Services



APIC-EM Grapevine Services

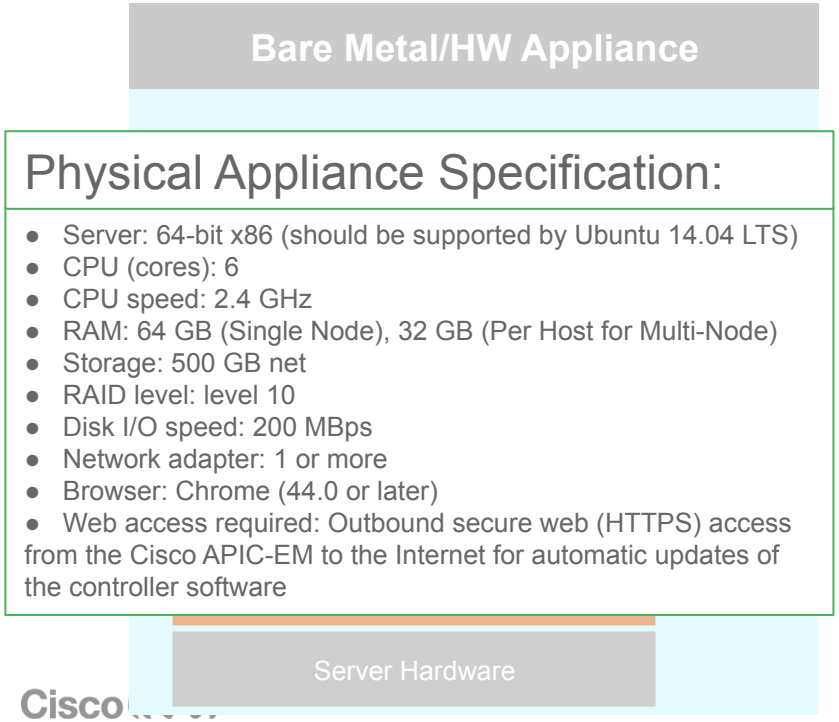
Grapevine Services Console

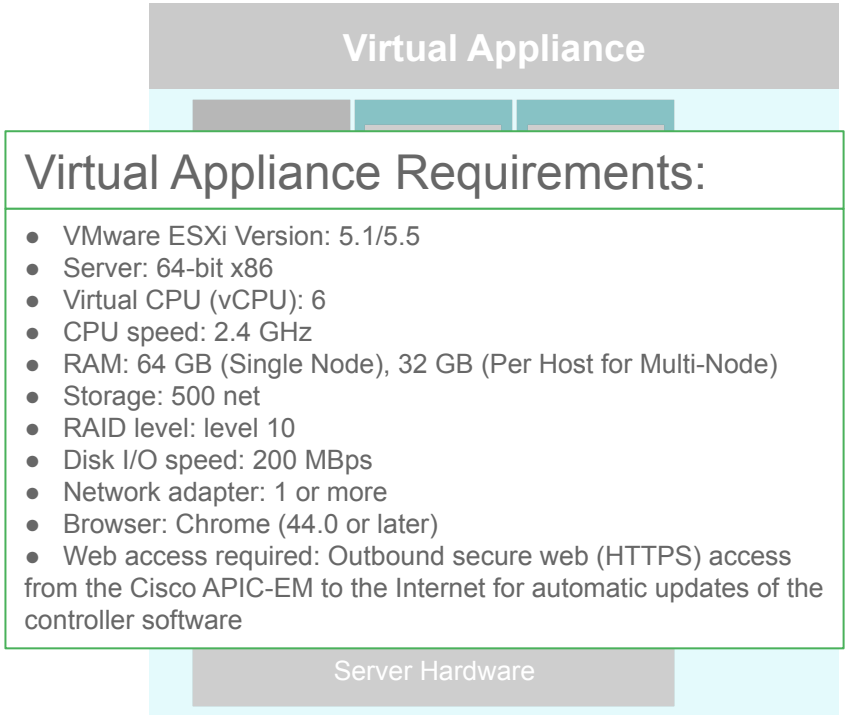


APIC-EM Grapevine

Deployment Considerations - System Requirements

The APIC-EM platform and its hosted applications can run as a virtual appliance when installed on a hypervisor or a bare-metal server. It is also available as a hardware appliance. System resources to run these two different form factors follow.



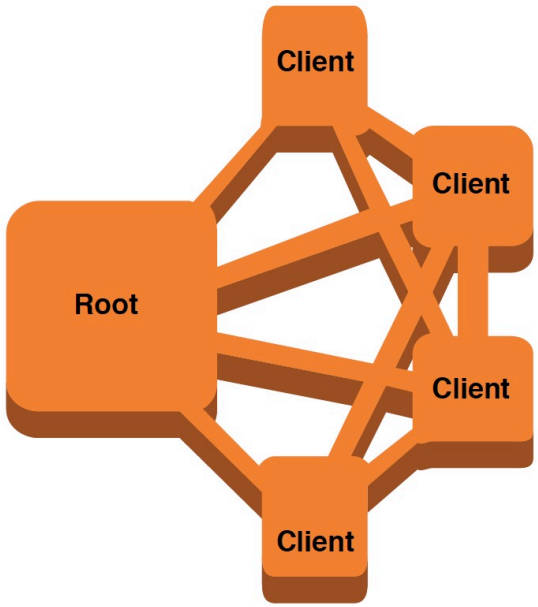


4.3 APIC-EM – Grapevine Cloud Deployment

APIC-EM Grapevine

GV Deployment: Elastic Service Management Framework

Grapevine
Platform for Service Elasticity



Mandatory Requirements:

- Easy to adopt
- Low cost of operation
- Cloud-like user experience

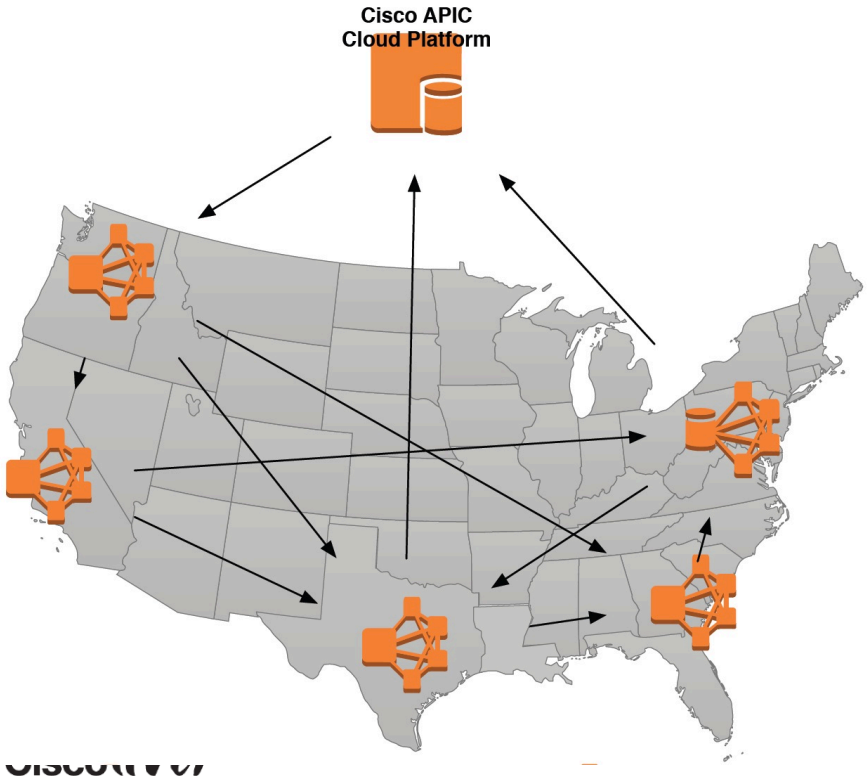
Goals:

- Manages mix of physical and virtual machines
- Common solution for physical and virtual
- Balances service instances between containers
- Services set elasticity policies
- Admin sets service priority policy
- Provides introspection of physical capacity
- Provides intelligent service routing to ensure optimal utilization
- Scales automatically into any provided resource
- No operational overhead to user
- Provides high-scale common services - data, queue, security, etc

Cisco live!

APIC-EM Grapevine

GV Deployment: Platform Wide-Geo Deployment



LAN-Local Grapevine Network Control

"Admin Role" Grapevine Policy Generation

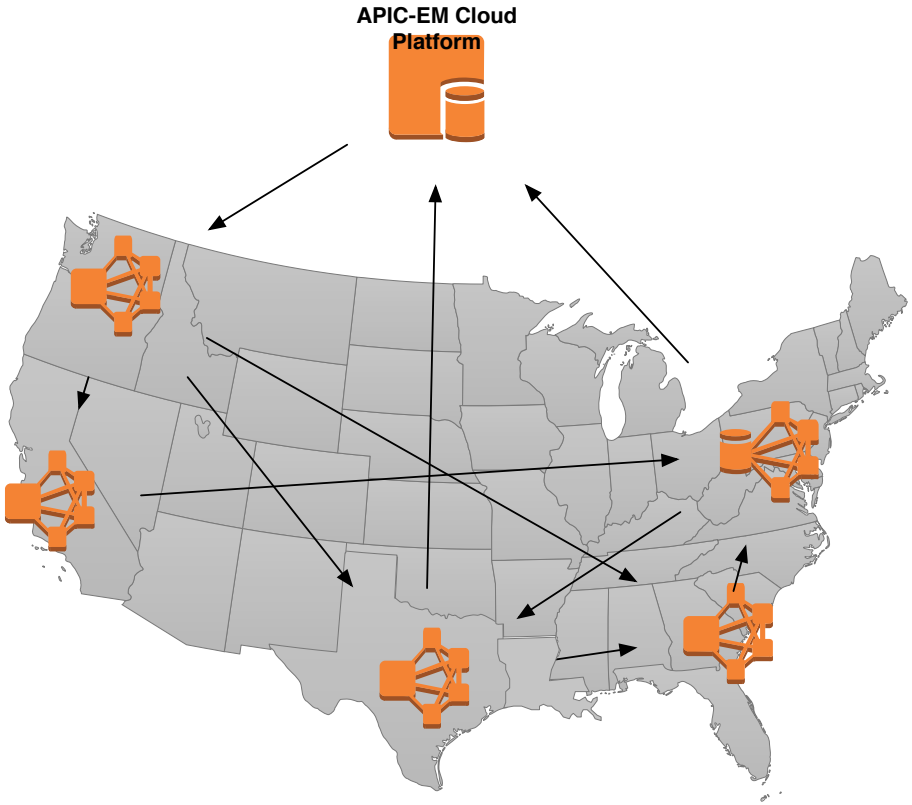
Metadata, Policy and Reporting Replication

Cloud Platform: Global Reporting, Backup, DR, Conflict and Split-Brain Resolution

APIC-EM

Cloud Connect Support Model

- Modern software uses cloud today
- Controller releases will be incremental (no big releases)
- Partially opt-in and fully auditable
- Core value is seamless, “never-touch-it” upgrade
- Data secured in Cisco cloud
- Single, global reporting system for your networks
- Config, state, and policy backup
- Split-brain resolution
- Push notification to mobile devices



Cisco live!

4.4 APIC-EM - Use Cases

APIC-EM Applications

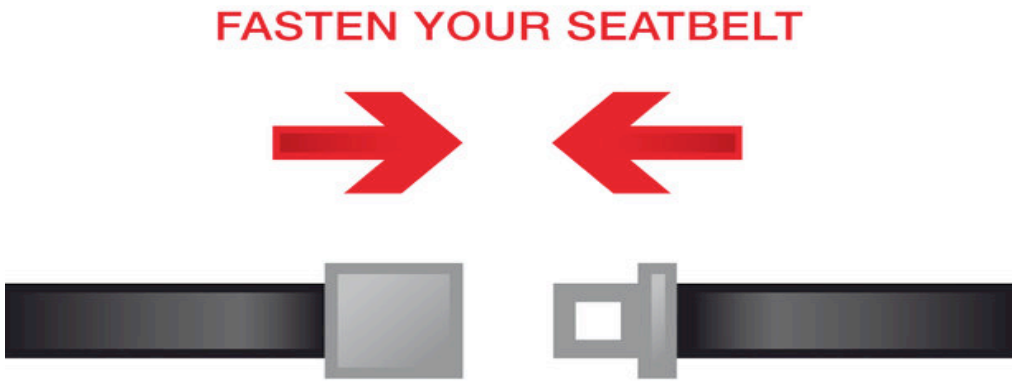
Things we have on our radar….

- Use Case: Path Trace
  One Click Host to Host connection analysis
- Use Case: Traffic Prioritization
  One Click QoS Policy Enforcement (Easy QoS)
- Use Case: Granular Control
  Per User Per Application Access Policy Enforcement
- Use Case: Next Generation Security Management
  Sourcefire and APIC-EM
- Use Case: DDoS Protection:
  Per User Network Traffic Redirection
- Use Case: Traffic Monitoring
  Per User Per Application Network Traffic Tapping
- Use Case: IWAN - Smart Routing
  Automated Provisioning of Routing Paths
- Use Case: Zero Touch Deployment (ZTD)
  Automated Provisioning and Deployment

APIC-EM – Application Slide burst

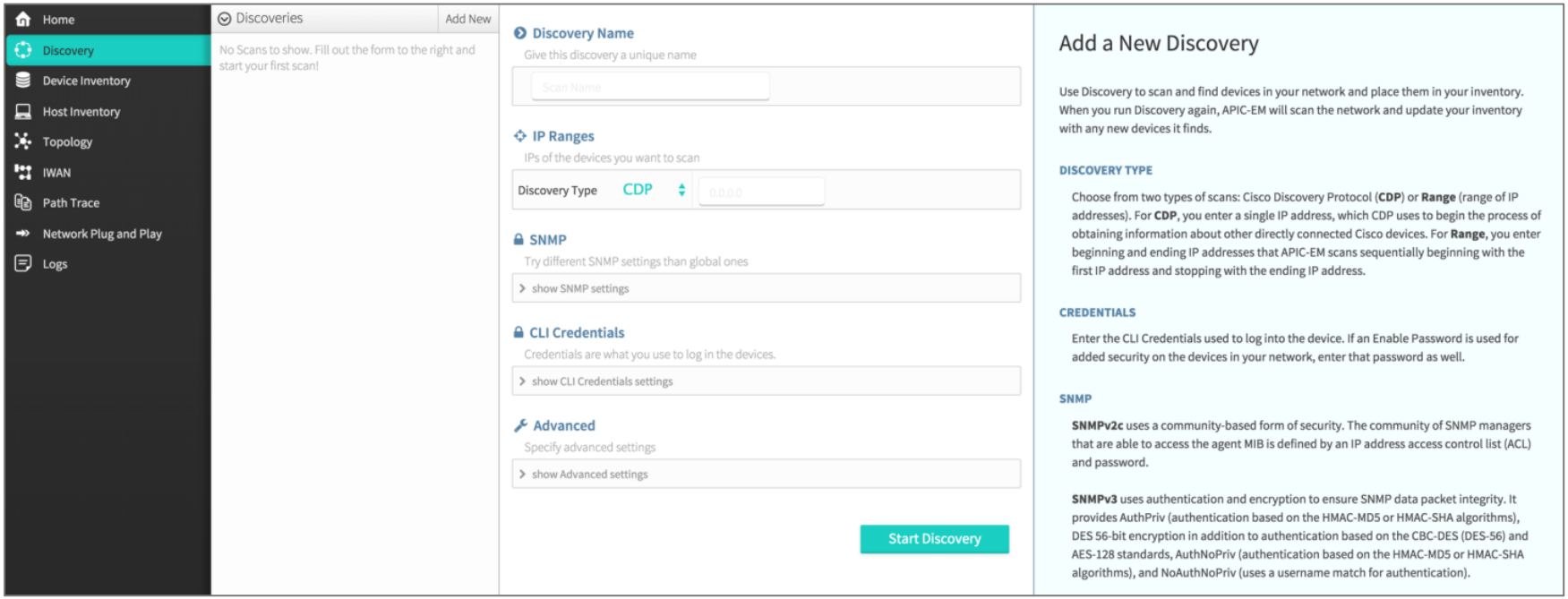
Fasten your seatbelts

For Your Reference

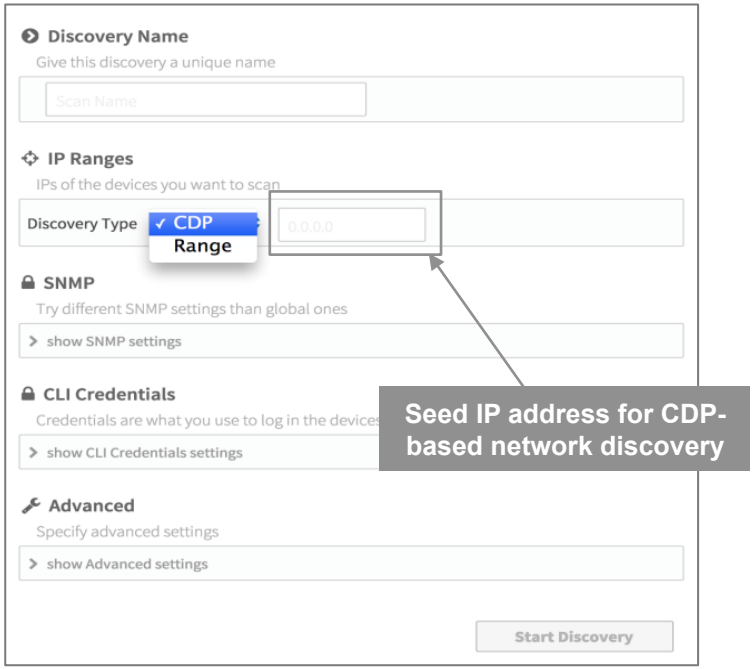


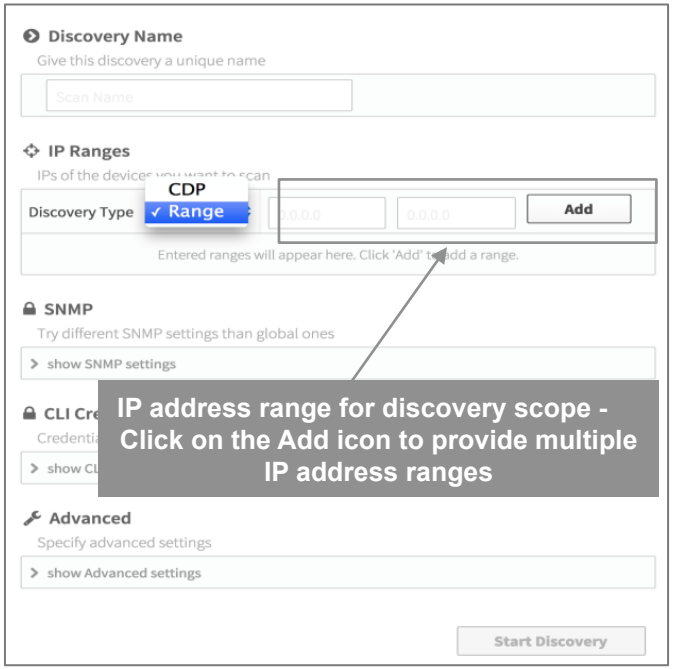
Cisco live!

Controller Application - Network Discovery

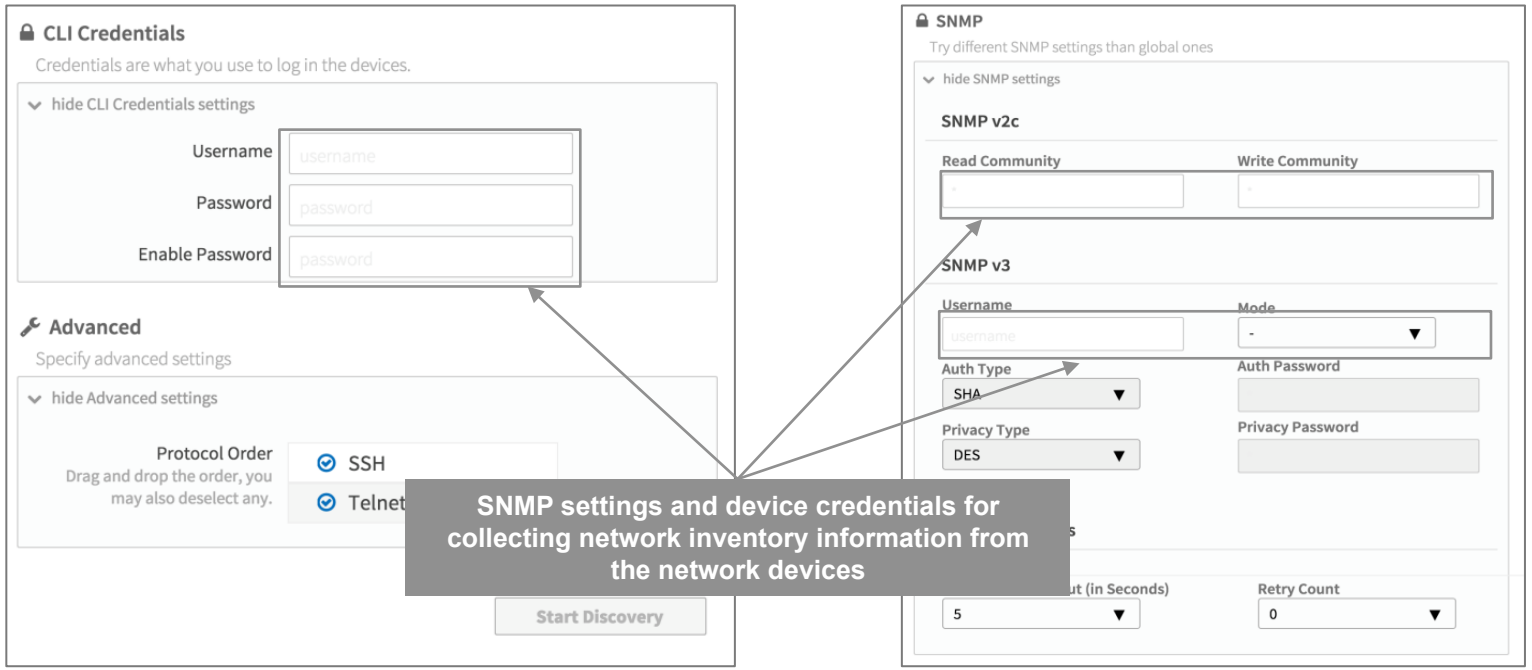


Network Discovery - Input Parameters

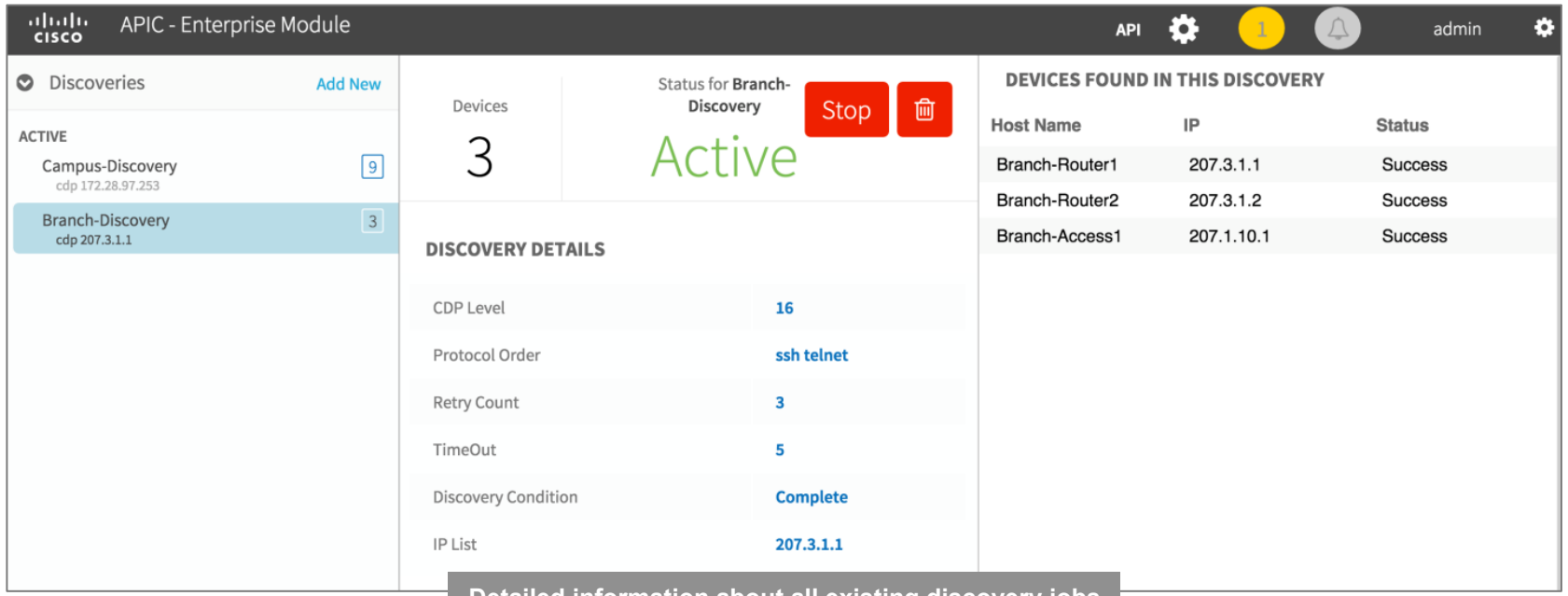




Network Discovery - Input Parameters

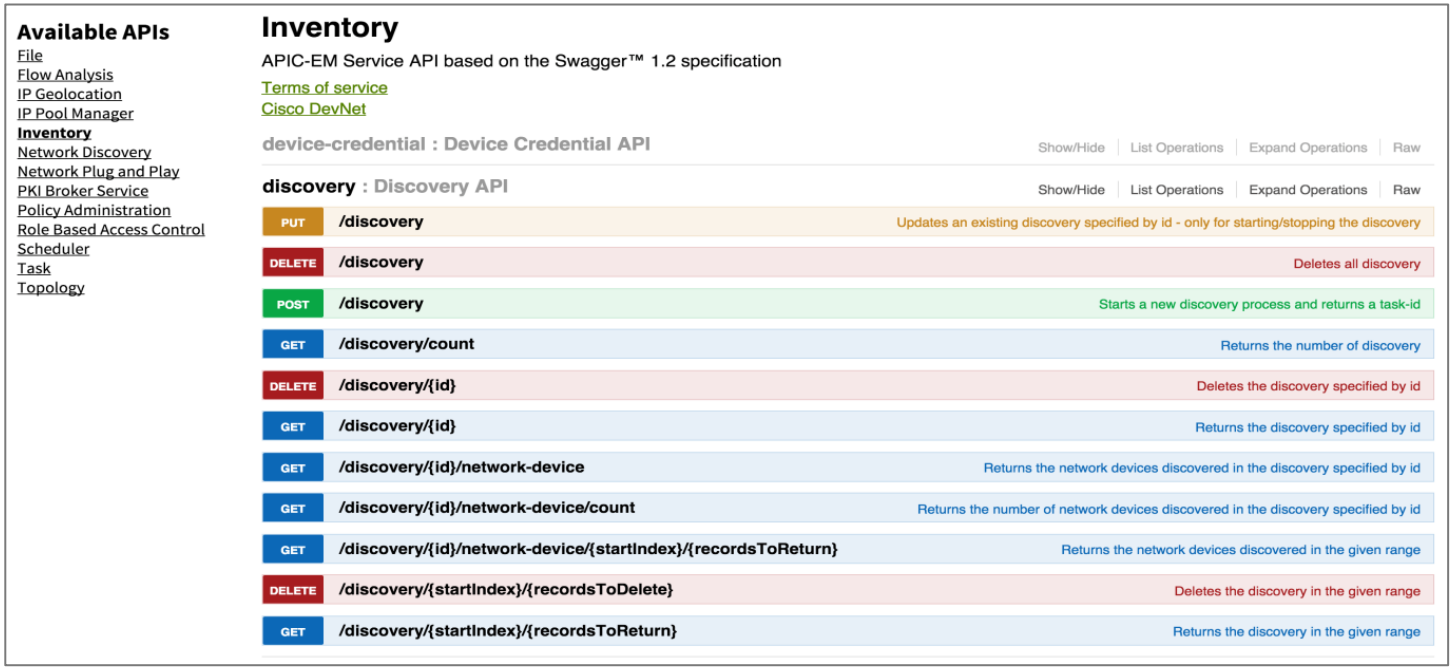


Network Discovery - Discovery Status



Detailed information about all existing discovery jobs

Network Discvery - Northbound REST APIs



Available APIs

File
Flow Analysis
IP Geolocation
IP Pool Manager
Inventory
Network Discovery
Network Plug and Play
PKI Broker Service
Policy Administration
Role Based Access Control
Scheduler
Task
Topology

Inventory

APIC-EM Service API based on the Swagger™ 1.2 specification

Terms of service
Cisco DevNet

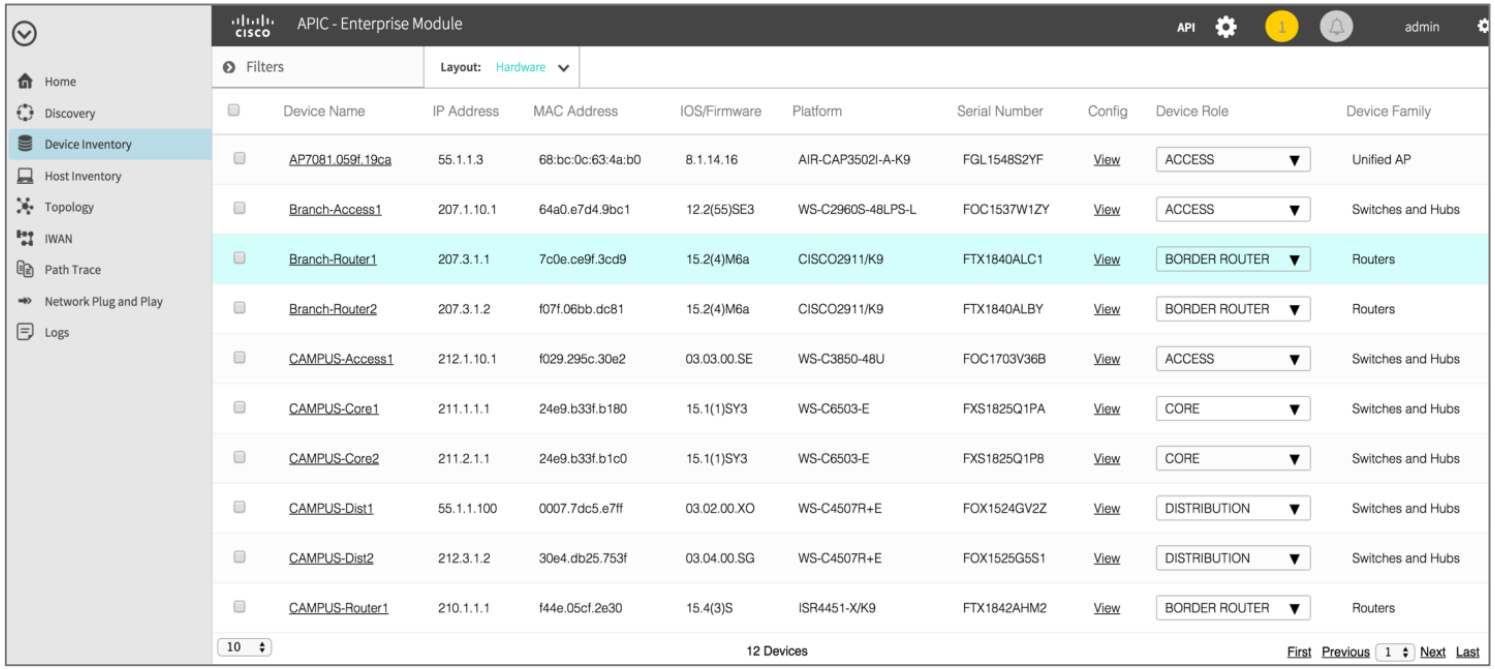
device-credential : Device Credential API — Show/Hide | List Operations | Expand Operations | Raw

discovery : Discovery API — Show/Hide | List Operations | Expand Operations | Raw

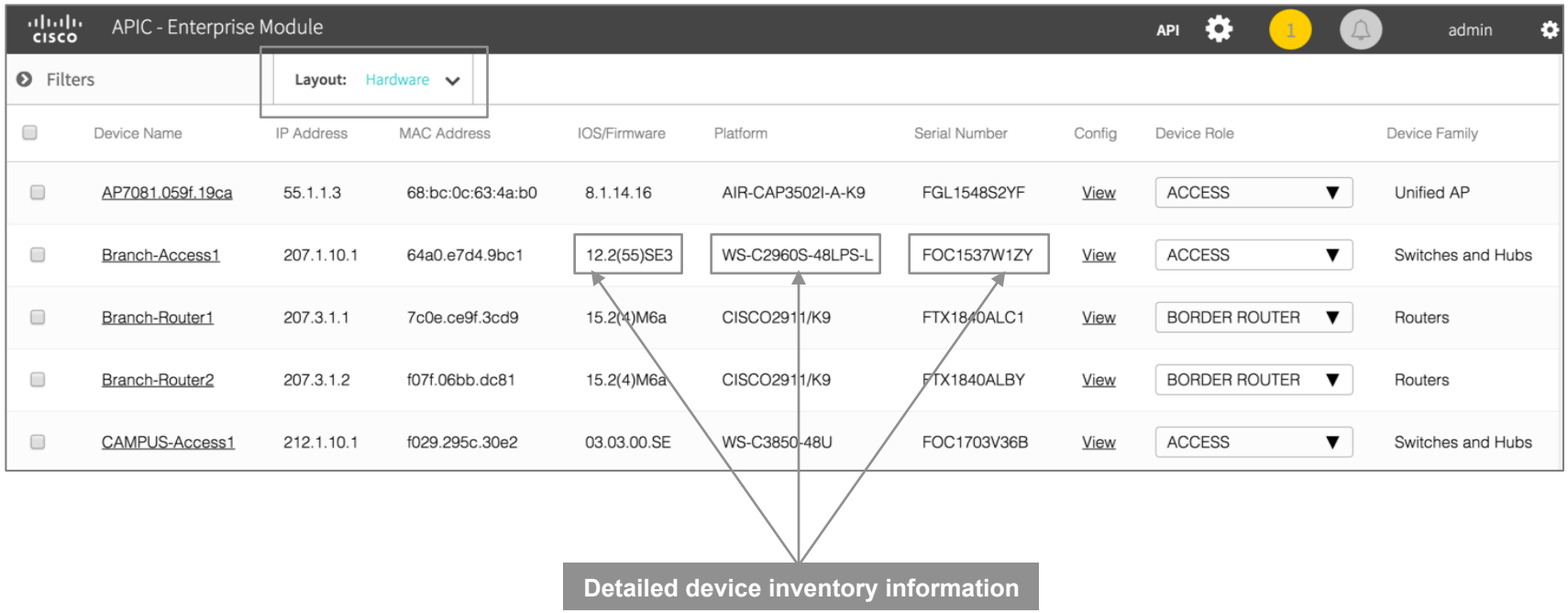
PUT	/discovery	Updates an existing discovery specified by id - only for starting/stopping the discovery
DELETE	/discovery	Deletes all discovery
POST	/discovery	Starts a new discovery process and returns a task-id
GET	/discovery/count	Returns the number of discovery
DELETE	/discovery/{id}	Deletes the discovery specified by id
GET	/discovery/{id}	Returns the discovery specified by id
GET	/discovery/{id}/network-device	Returns the network devices discovered in the discovery specified by id
GET	/discovery/{id}/network-device/count	Returns the number of network devices discovered in the discovery specified by id
GET	/discovery/{id}/network-device/{startIndex}/{recordsToReturn}	Returns the network devices discovered in the given range
DELETE	/discovery/{startIndex}/{recordsToDelete}	Deletes the discovery in the given range
GET	/discovery/{startIndex}/{recordsToReturn}	Returns the discovery in the given range

Cisco live!

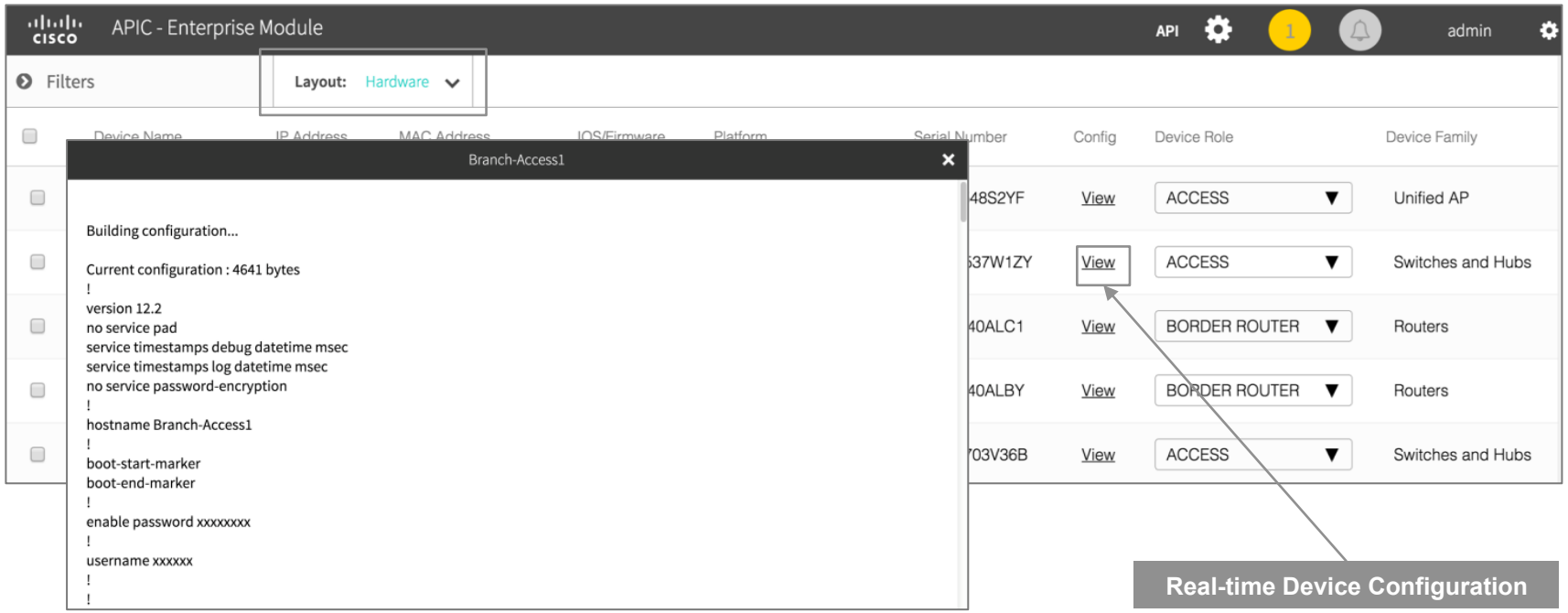
Controller Applications - Device Inventory



Device Inventory - Hardware Layout



Device Inventory - Hardware Layout



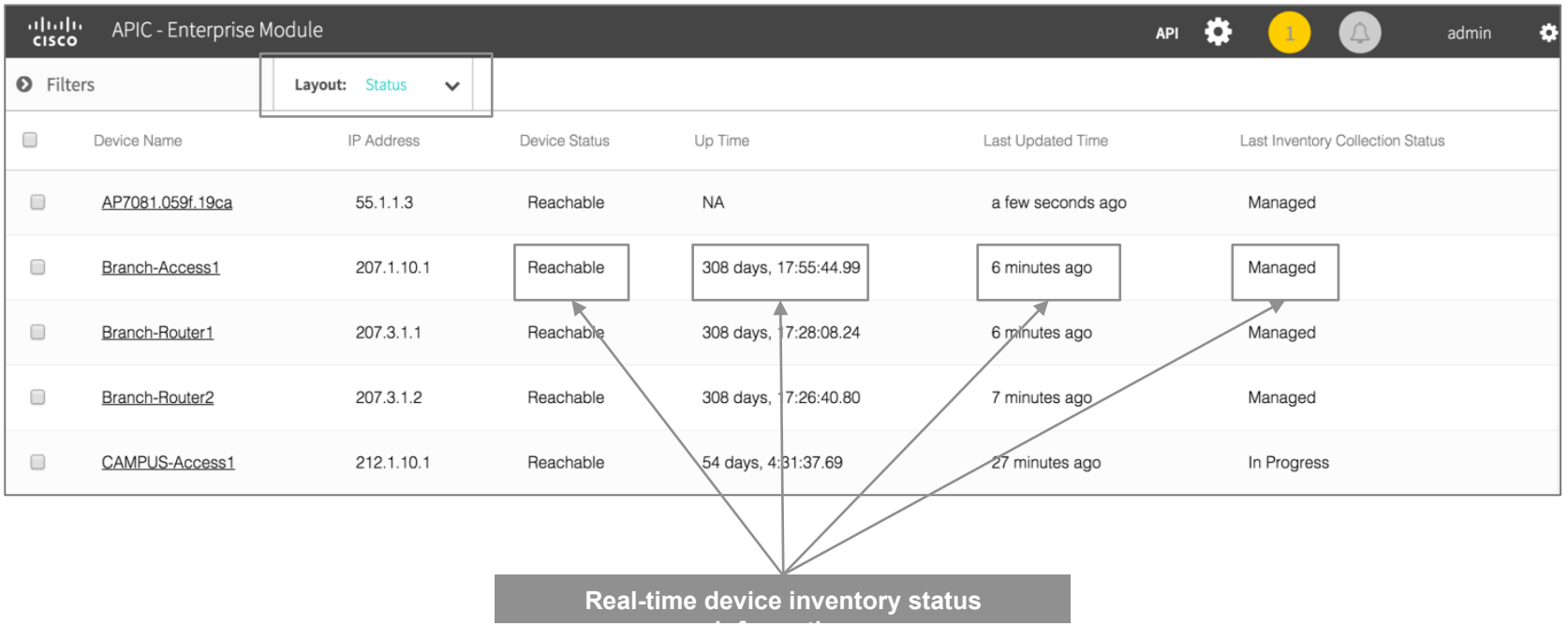
Device Inventory - Tagging Layout

Sophisticated and automated devices are given a role assignment based on intelligent matching against pre-set templates and attributes

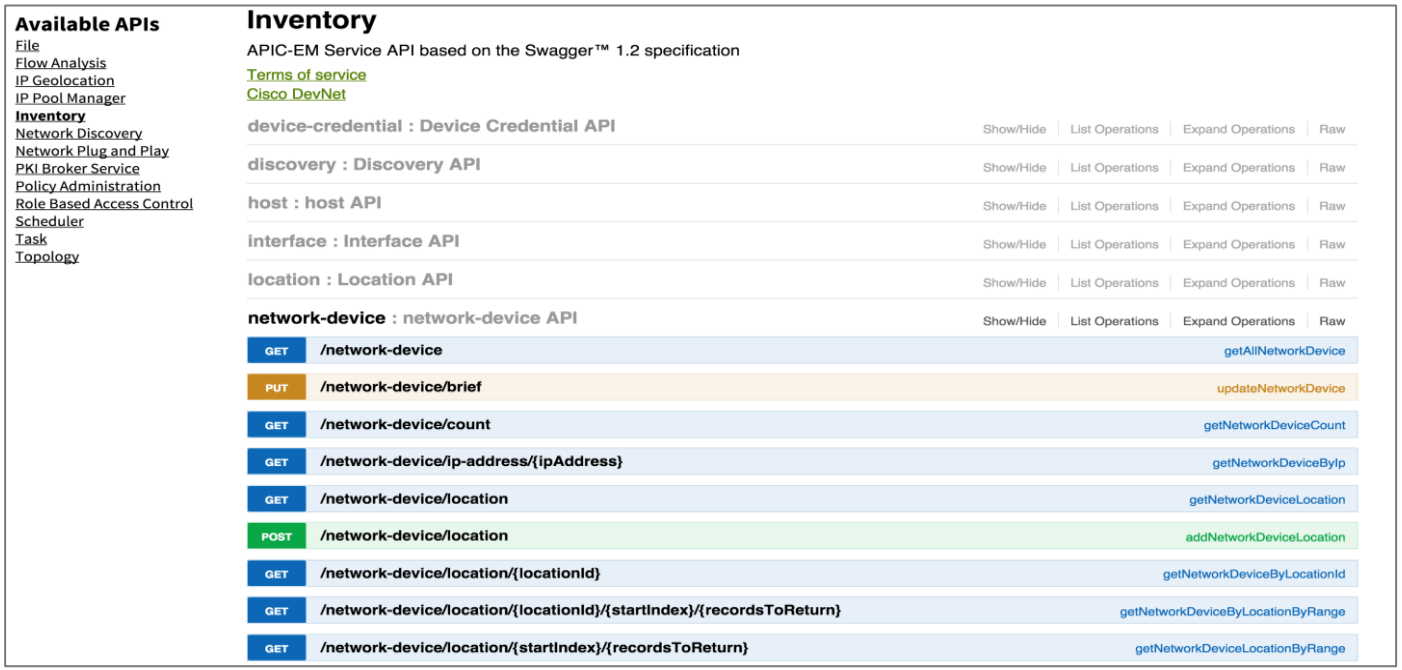
Geo-site (location) and custom tags for complete flexibility in grouping and classification of devices based on business logic (for example, lines of business, service mix, etc.)

Cisco live!

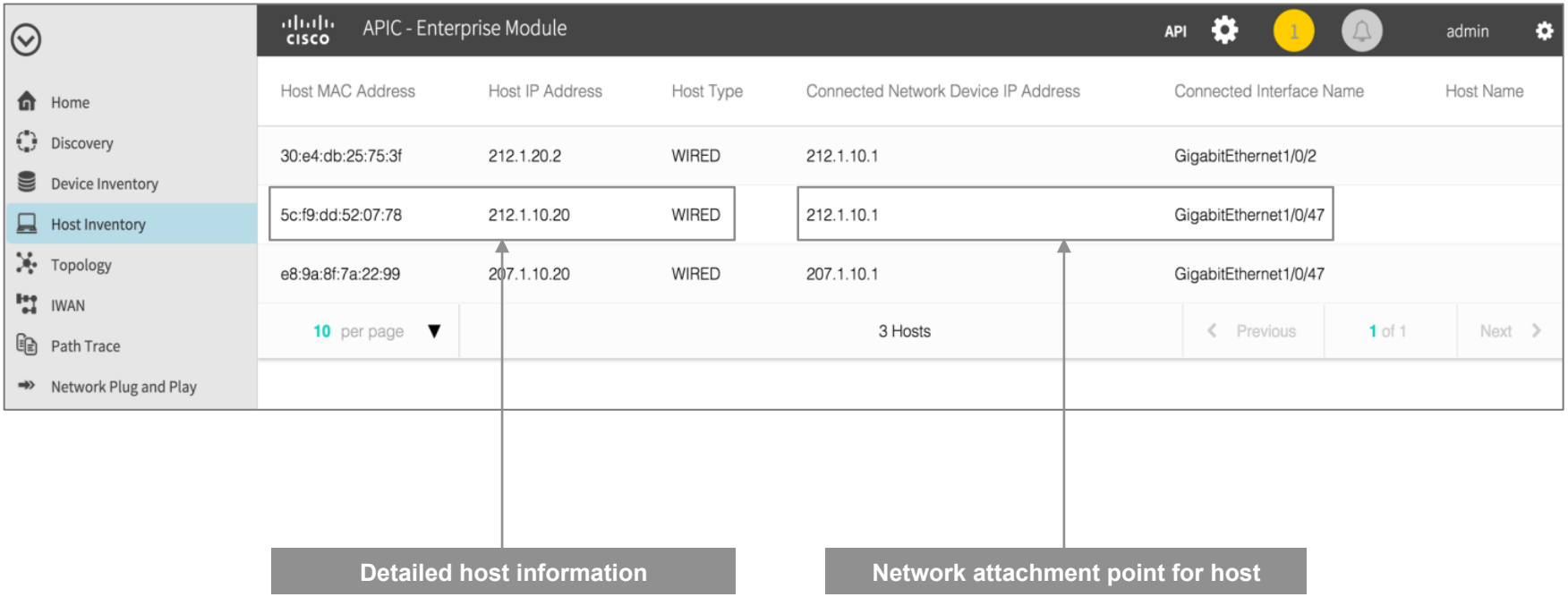
Device Inventory - Status Layout



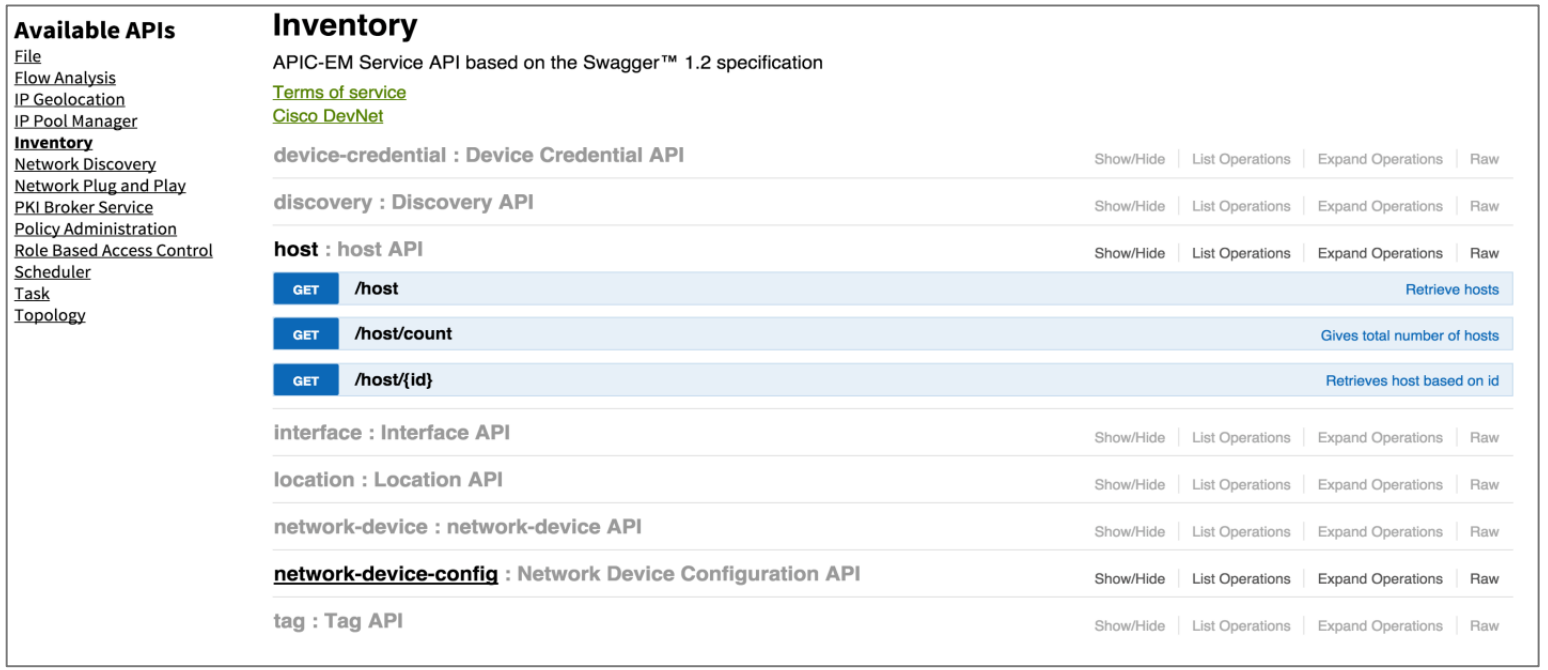
Device Inventory - Northbound REST APIs



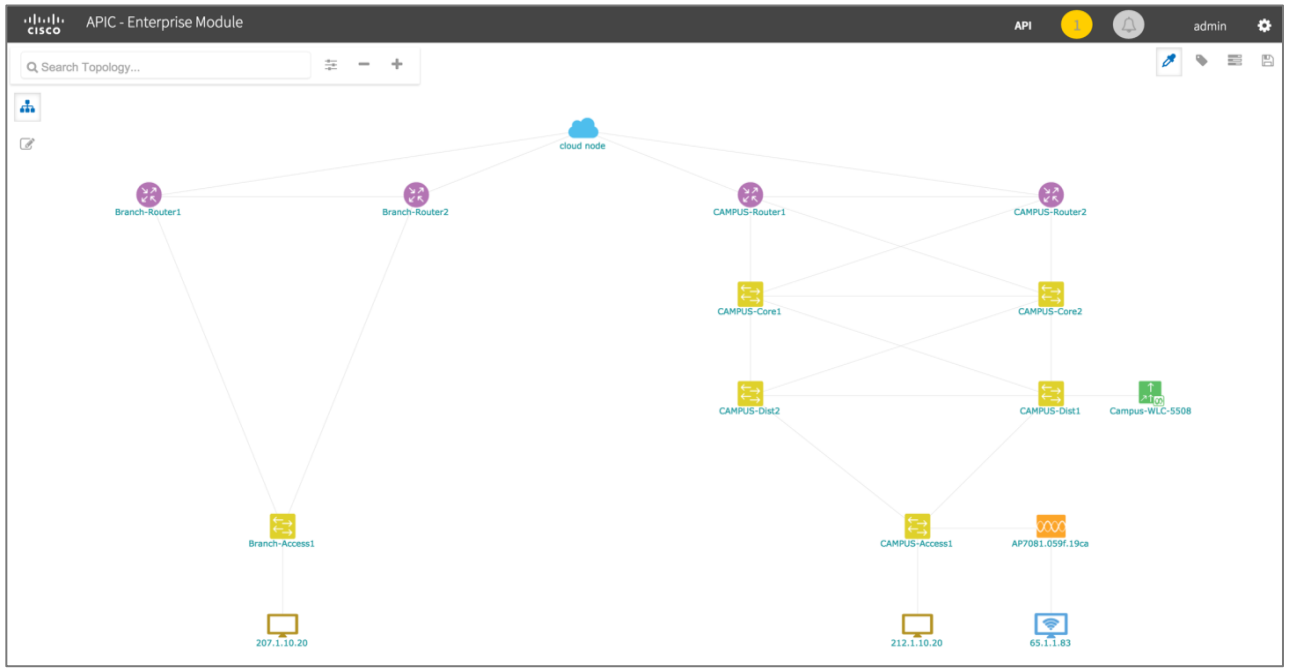
Controller Applications - Host Inventory



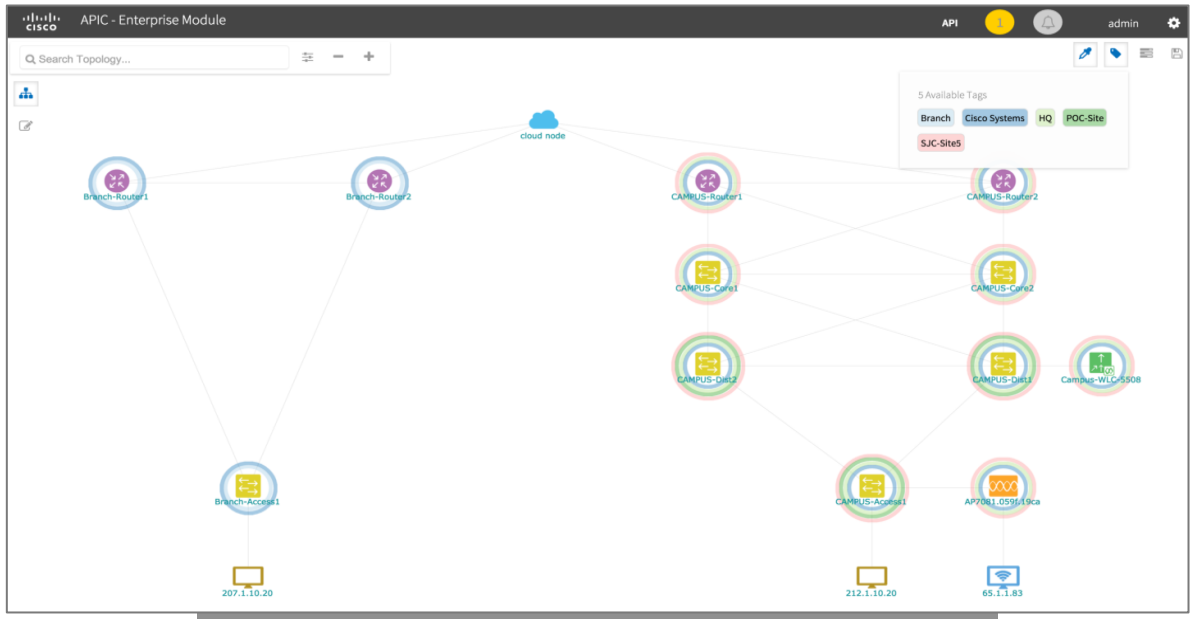
Host Inventory - Northbound REST APIs



Controller Applications - Topology Visualizer

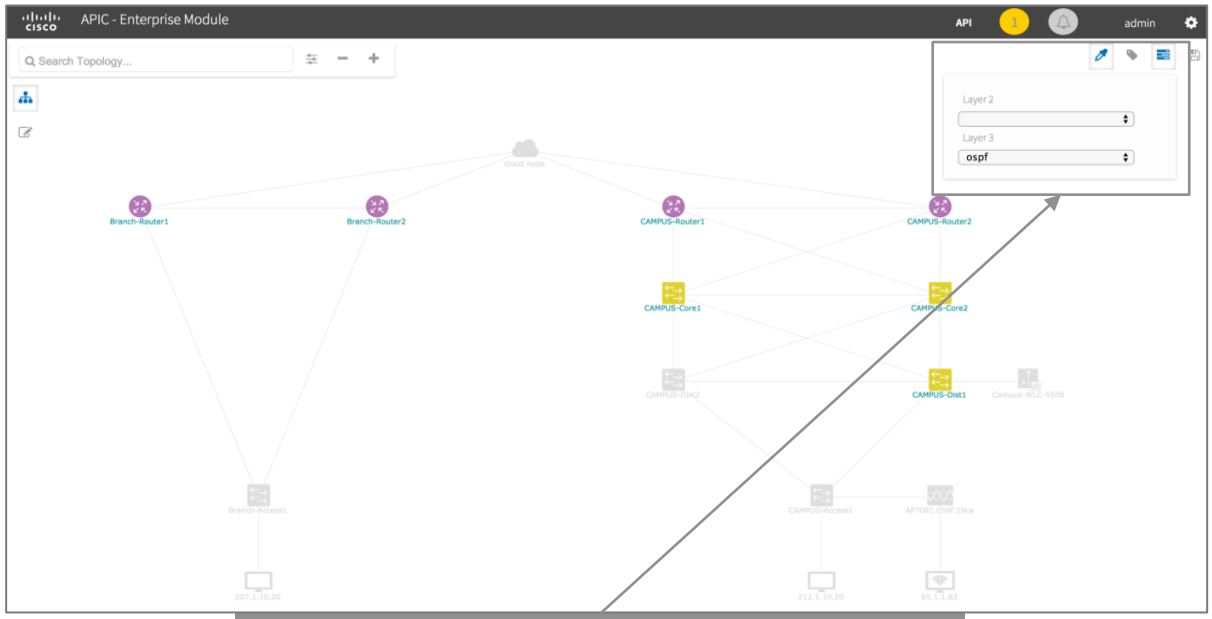


Topology Visualizer - TAG View



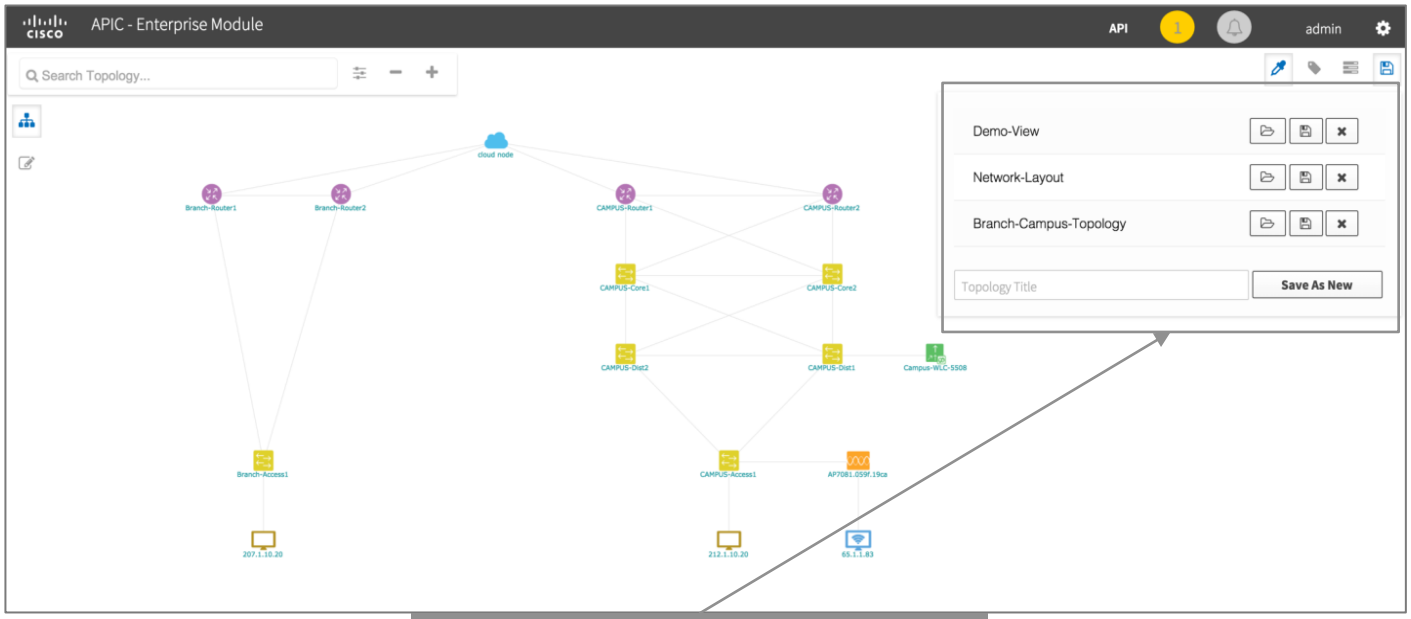
Ability to visualize device TAGs in the topology view

Topology Visualizer - L2/L3 Topology View



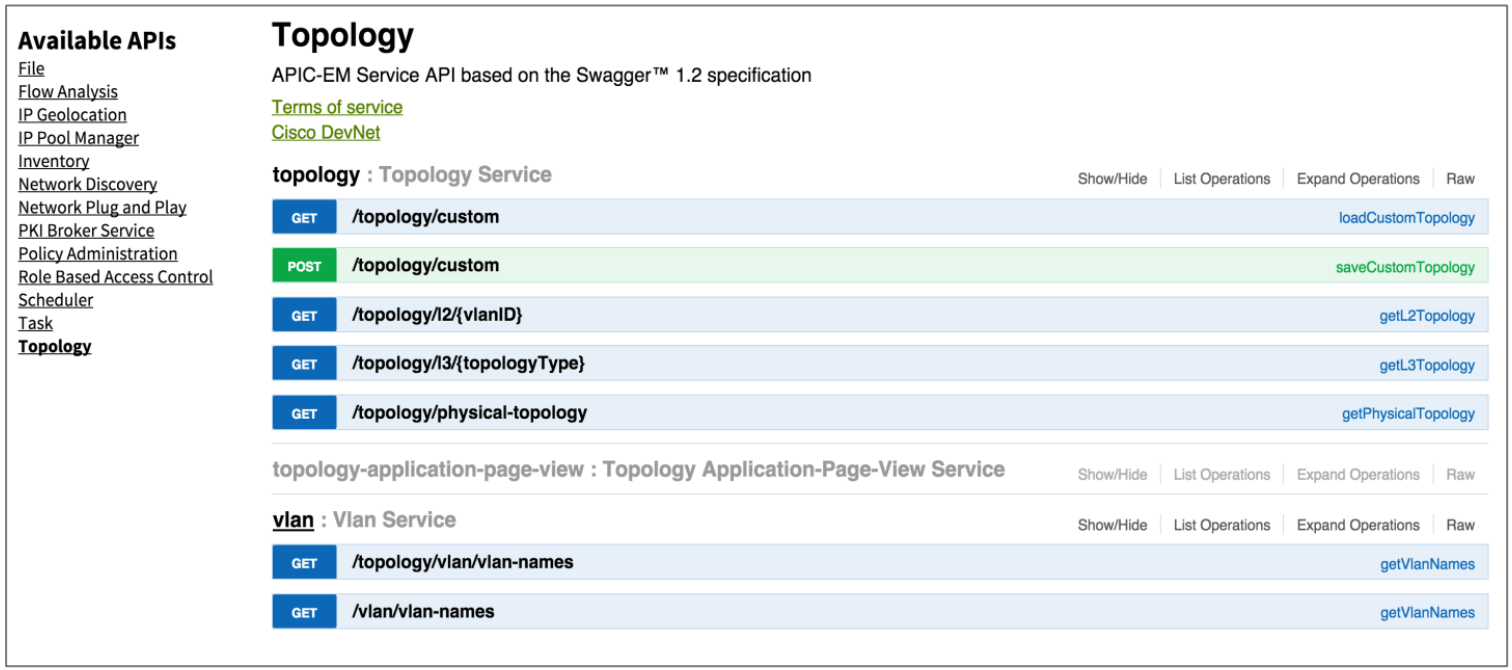
Ability to visualize Layer 2/Layer 3 topologies

Topology Visualizer - Saved Layouts



Ability to save multiple topology layouts

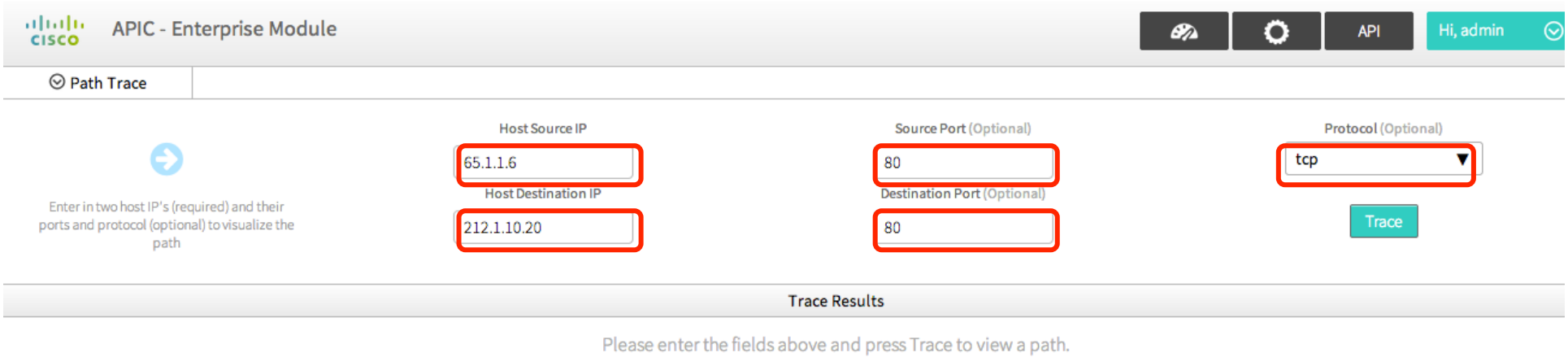
Topology Visualizer - Northbound REST APIs



Path Trace

5-Tuple Input

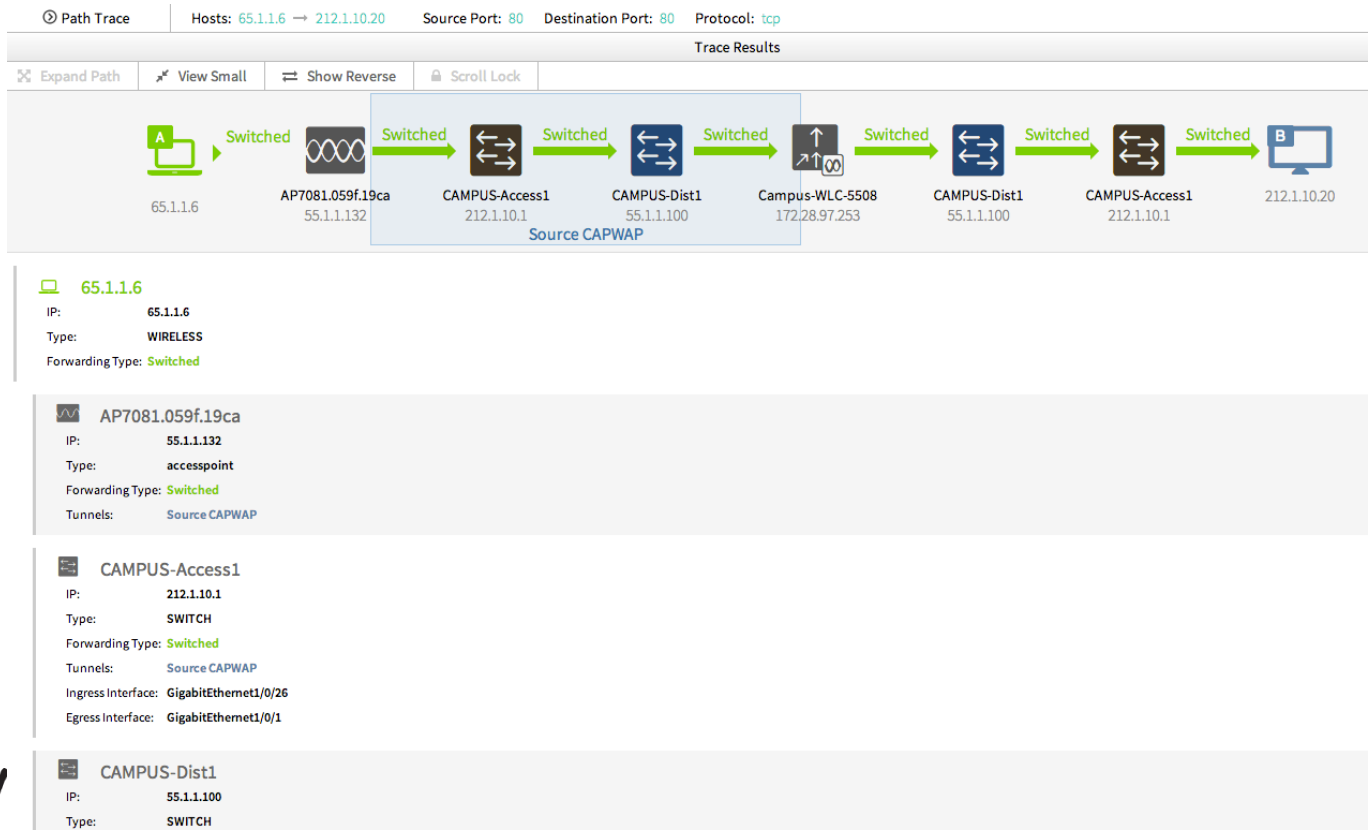
For Your Reference



Path Trace

Results

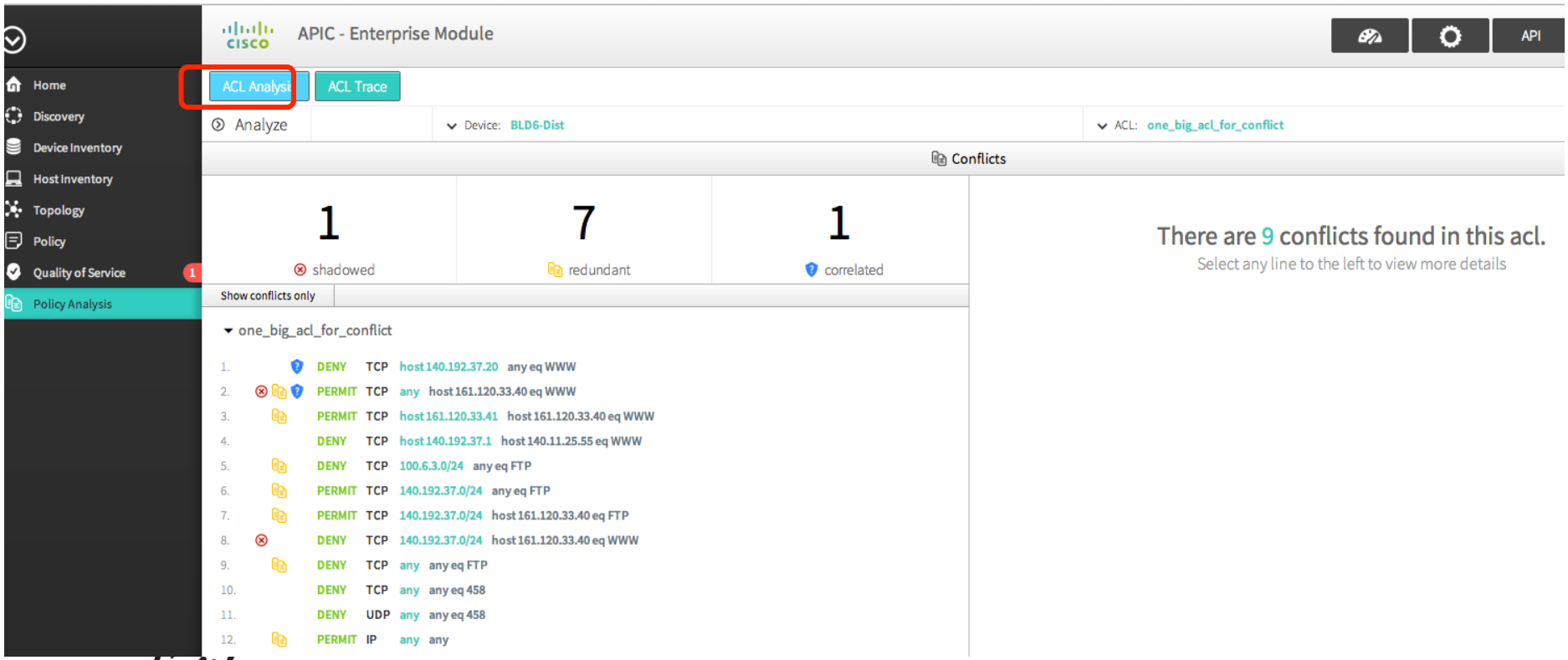
For Your Reference



Policy Analysis

ACL Analysis

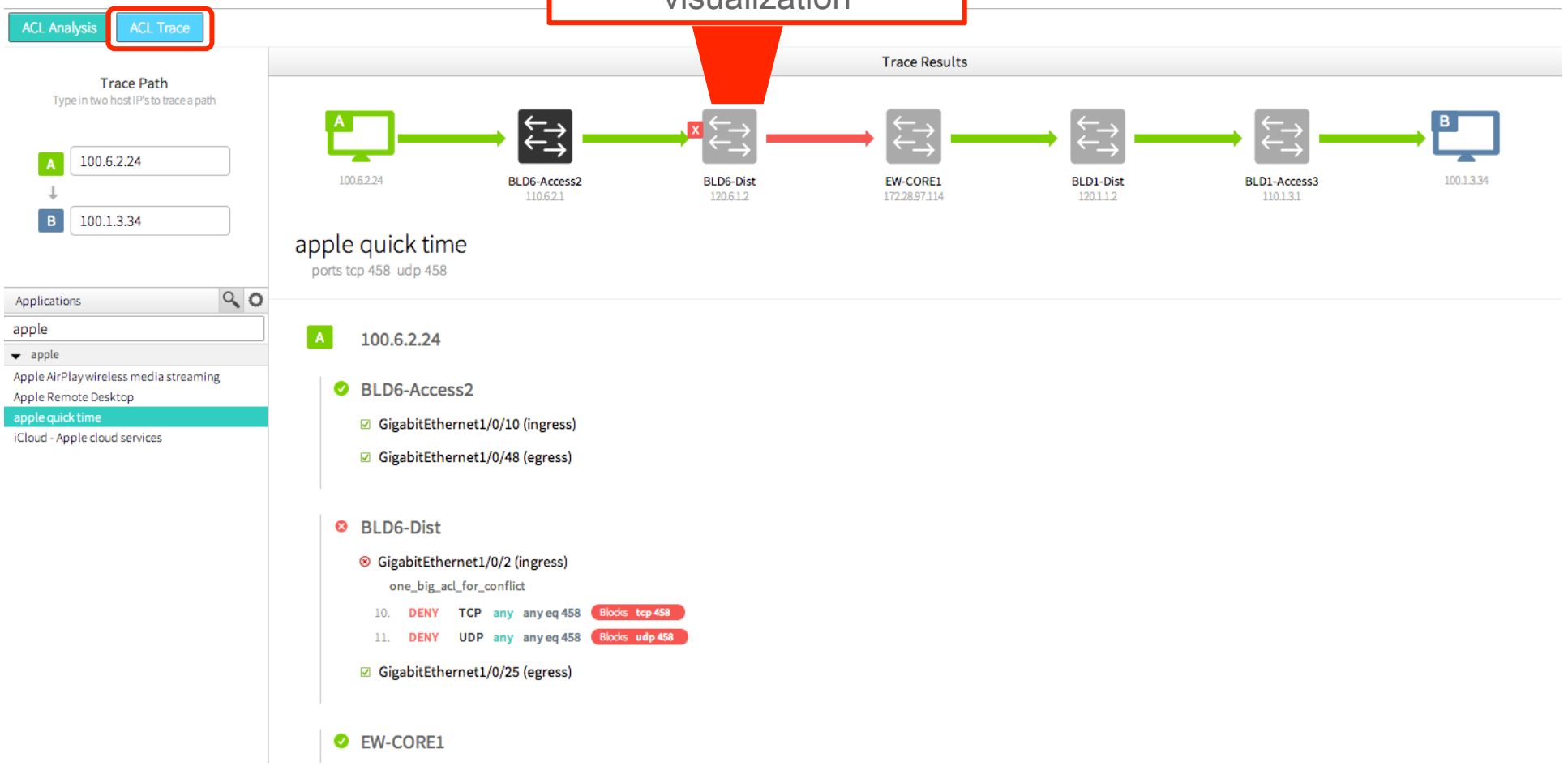
For Your Reference



Policy Analysis
ACL Trace

Boxes greyed out once traffic is blocked for easy visualization

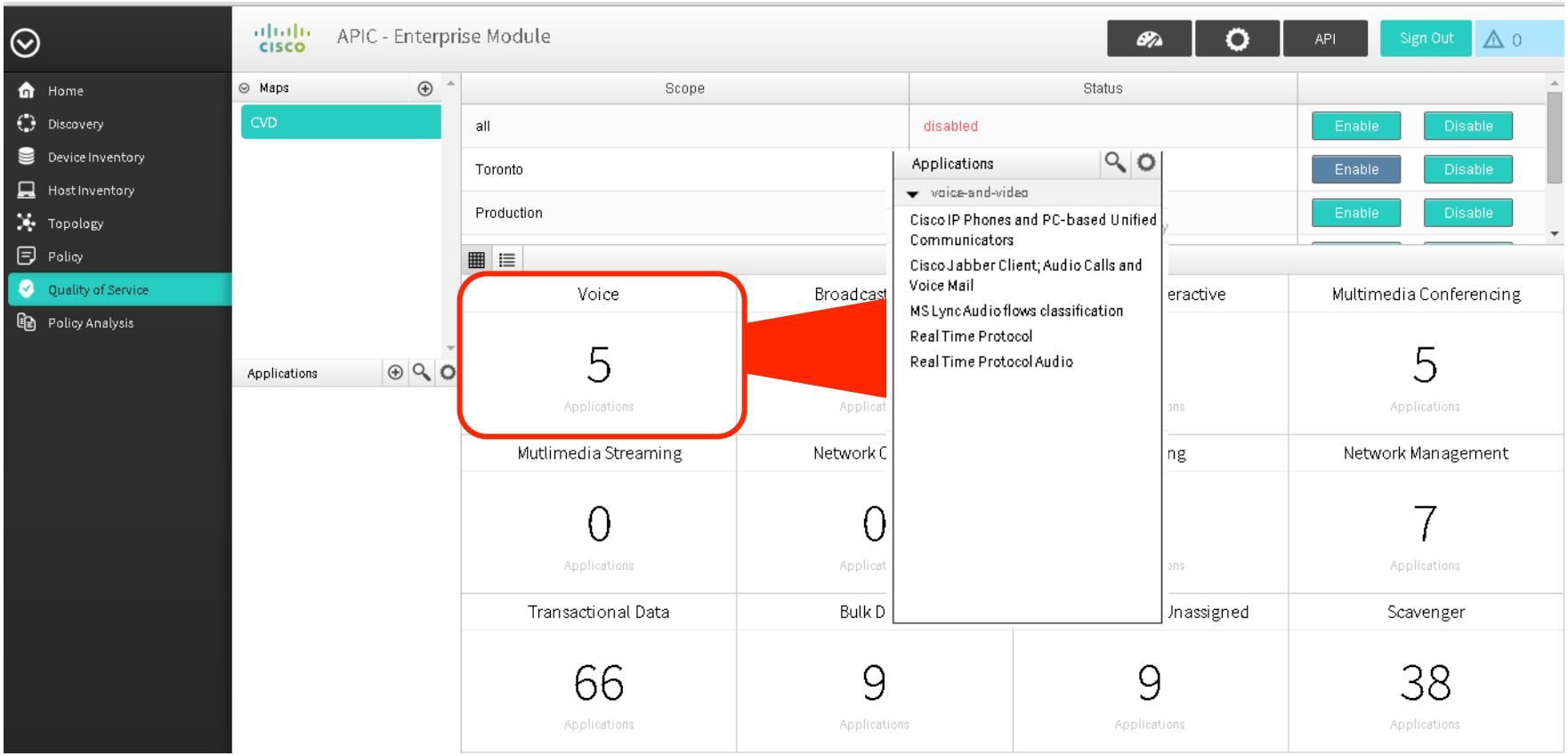
For Your Reference



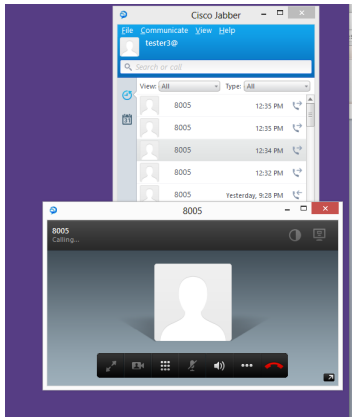
Easy QoS

Easy customization of policies

For Your Reference



Dynamic QoS Classification



For Your Reference

APIC - Enterprise Module

Settings

Applications

- Home
- Discovery
- Device Inventory
- Host Inventory
- Topology
- Policies
- Quality of Service
- ACL Analysis
- Zero Touch Deployment
- Collaboration Services

Create Policy | Delete All Policies

Current Policies

Name	Scope	Users: Users	Users: Application	Resources: Users	Resources: Application	Actions	Priority Level	Destination	Status	Actions
createPolicy:: UCPolicy172.28.97.52:32013		172.28.97.52	0,0,UDP	172.28.97.51	0,0,UDP	PERMIT	32	N/A	Active	
createPolicy:: UCPolicy172.28.97.51:32016		172.28.97.51	0,0,UDP	172.28.97.52	0,0,UDP	PERMIT	32	N/A	Active	

Displaying Two of Two Policies

Delete All | Refresh

Cisco live!

5. Demo

- **W**orld **O**f **S**olutions
- Whisper suite sessions

Demo

6. Conclusion and Open Discussion


The World is
CHANGING


And it's
CHANGING FAST
Market
Transitions
Technology
Transitions
Economic
Transitions
In a "share economy" world of "real time" and "co-innovation",
the relationship between supplier and customer is blurring.


“The biggest risk is not taking any risk…
In a world that changing really quickly, the only strategy that is guaranteed to fail is not taking risks.”
Zuck’s

SDN Hard Problems

Some musings on SDN for EN

- **Technology**
  - Separation of Control and Data Planes:
    - ✓ Control Plane Scalability and Resilience
    - ✓ State Management: Logically Centralized?
    - ✓ State Distribution Trade-offs in SDN
      - ➢ Control State Consistency ./. Application Optimality
      - ➢ Application Complexity ./. Robustness to Inconsistency
    - ✓ Combinatorial state explosion: Feasibility, CAP theorem, …
    - ✓ Control Plane Performance:
      Ω = RTT(switch2packet) + pps(switch) + pps(controller)
  - Hybrid Switch Implications
  - Flow Setup Scalability and Performance
  - Topology Discovery and response times
  - CPUs ./. TCAMs = overlay ./. underlay = state ./. Speed
  - Abstractions
    - ✓ Sweet spot: Leverage ideas from distributed systems, programming languages, and other areas to bridge the gap between the centralized controller abstraction and the distributed/hierarchical reality
    - ✓ "northbound" + "southbound" abstractions
    - ✓ Forwarding targets – ASICs and TCAMs
  - Policy Controller
  - Reasoning Systems, Big Data
  - "network as a computer", network compilers…
  - OpenFlow, A Retrospective on Evolving SDN ⇒ MPLS
  - OpenStack

- **Sociology**
  - OF/SDN approach challenges much of our central dogma
  - Remember QoS trust boundaries
  - Not the least of which are
    - ✓ Circuits vs. Hop-by-hop forwarding
    - ✓ Centralized ./. Distributed control planes ./. "flow-based'
  - Operational Models
    - ✓ Operational change is quite substantial (ITIL & ITSM)
    - ✓ How to you build/operate/debug these networks?
    - ✓ Who is in charge of creating a 12-tuple?
    - ✓ How to Combine Compute, Storage, Networking and App teams
    - ✓ How to translate business intent into policies
    - ✓ Convolution of policy and configuration
  - A solution looking for a problem
  - Controller – Agent – Troubleshooting ./. Single BU
  - Have we been unwilling or unable to abstract complexity.
  - Believe network teams do NOT have the skills and experience to implement and manage SDN
  - Influence shift from from NetOps ⇒ DevOps
  - Is it really about NetOps or more about DEV ./. OPS ?

- **Economics**
  - Well…all of the above
  - RYF-complex (Fragile/Robust)
  - Product "de-siloing"
  - Does it really become "cheaper"???

Lunch and Learn

LALCRS-0006 - APIC-EM - Thursday 18 February 13:00 – 14:15

During lunch on Tuesday, Wednesday and Thursday, you can join Cisco subject matter experts and your peers in these casual conversations about topics of interest to you.

The Lunch and Learn tables are located in the Catering Area in Hall 4.1.

For a full list of topics on each day, go to:

Cisco live!

http://cs.co/berlin-lal

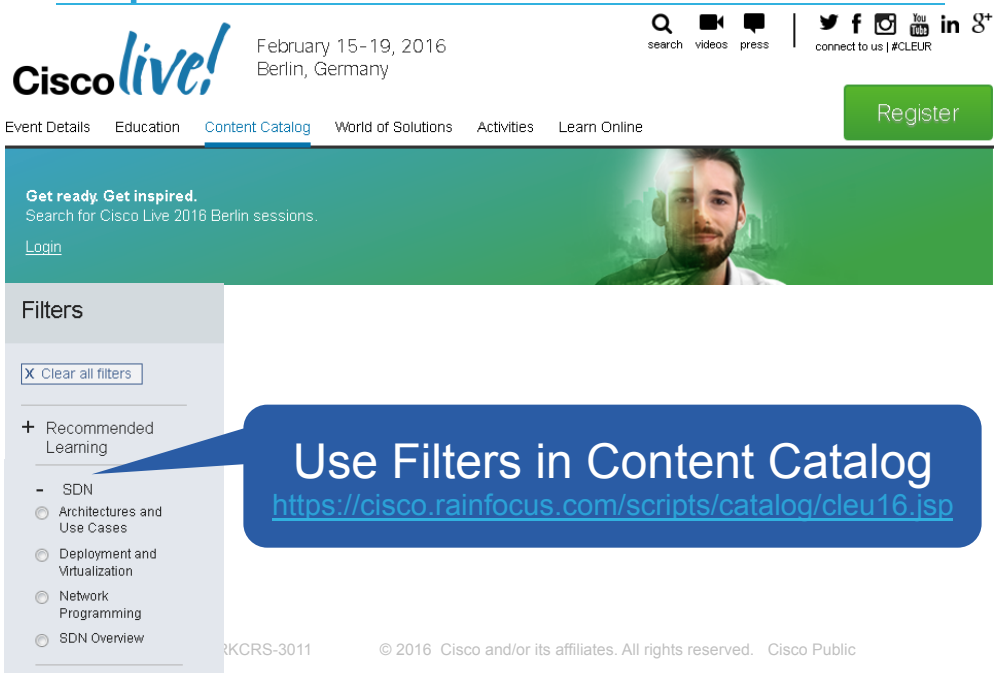
SDN @ CiscoLive

- Recommended Learning Path on SDN
- 60+ Sessions
  - Technical Seminars
  - Breakout Sessions
  - Hands-on Labs
  - Panel Discussion
- DevNet Zone
- Demos, MTE, Lunch&Learn, Whisper Suites,and more ....

Ciscolive!

wolfgang@cisco.com

http://www.ciscolive.com/emea/



Use Filters in Content Catalog
https://cisco.rainfocus.com/scripts/catalog/cleu16.jsp

Enterprise SDN @ CiscoLive

Monday	Advanced APIC Enterprise Module: SDN Controller for the Campus and Branch - TECSDN-3600
Monday	Enterprise SDN: Architectures and Key Concepts - TECSDN-2602
Monday	Enterprise SDN: Advanced Network Programming - Hands-On Lab TECSDN-3602
Tuesday	APIC-EM: Controller Workflow and Use Cases - BRKARC-3004
Tuesday	IWAN management via APIC-EM (SDN Controller) - BRKSDN-2099
Tuesday	CCIE Skill Transformation to SDN Kungfu Master - BRKSDN-4005
Wednesday	SDN Enabled QoS-A Deep Dive - BRKSDN-2046
Wednesday	Hitchhiker's Guide to Device APIs - BRKSDN-1119
Wednesday	Containers on routers and switches: Run your apps and tools natively on Cisco boxes - BRKSDN-2116
Wednesday	Playing With Your Traffic: Exploring Software-Defined Packet Control - BRKSDN-3014
Wednesday	Cisco Application Policy Infrastructure Controller Enterprise Module (APIC-EM) – Hands on Lab - LTRSDN-1914
Thursday	APIC-EM: The evolution from traditional management to SDN-led, policy-based automation - BRKNMS-2031
Thursday	Cisco Open SDN Controller Hands-on Lab - LTRSDN-1913
Thursday	Deploying Cisco IOS Autonomic Networking Infrastructure - BRKSDN-2047
Thursday	DNS-AS: Done with SDN and Tired of Dealing with Snowflake Network Complexity? Change the Game with a Simple TXT String! - BRKSDN-3004
Friday	Solutions Enablement by Cisco Open SDN Controller - BRKSDN-1020

Cisco live!

More SDN Sessions in the Recommended Learning Path

Thank you

Some more fun stuff to watch…

- Fundamentals of Cisco APIC-EM
  https://www.youtube.com/watch?v=17lDRT9tuWY
- Metadata-Defined Data Center, Mike Dvorkin, Cisco Systems
  http://techfieldday.com/appearance/introducing-the-next-generation-sddc-leaders-1
- Developing OpenDaylight Apps with MD-SAL
  https://www.youtube.com/watch?v=uBnDJNsd6Qo
- Application Centric Infrastructure (ACI) Overview
  http://www.youtube.com/watch?v=VZWwjNAiUpI
- APIC EM Demo, Apr 2014 - VT Recording
  http://videosharing.cisco.com/p.jsp?i=10394
- CCO:
  http://www.cisco.com/c/en/us/products/cloud-systems-management/application-policy-infrastructure-controller-enterprise-module/index.html?wcmmode=preview