TOMORROW starts here.
APIC Enterprise Module
SDN in the Enterprise

TECSDN-3600

Wolfgang Riedel – Cisco Principal Engineer Engineering
Lila Rousseaux – Cisco Consulting Systems Engineer
Adam Radford – Cisco Distinguished Systems Engineer
Markus Harbeck – Cisco Consulting Systems Engineer
To Whom Are You Listening?
for the next 570 minutes

Lila Rousseaux
CONSULTING SYSTEMS ENGINEER.SALES
TORONTO
CANADA

Wolfgang Riedel
PRINCIPAL ENGINEER.ENGINEERING
ERLANGEN
GERMANY

Adam Radford
DISTINGUISHED SYSTEMS ENGINEER.SALES
NORTH SYDNEY
AUSTRALIA

Markus Harbeck
CONSULTING SYSTEMS ENGINEER.SALES
ESCHBORN
GERMANY
Who is Lila Rousseaux

• Personal:
  – Location: Toronto, Canada (born in Argentina and moved in 2002)
  – Other Interests: my two wonderful kids, spinning and working out, rock climbing, reading

• Background:
  – Joined Cisco December 1999
  – Before:
    • SoftNet (Cisco Gold Partner)
    • Lucent Technologies
    • World Bank contractor for the RIU project (Red de Interconexión Universitaria)
    • University of Buenos Aires

  • For those of you trying to do the math … please note that I started working when I was less than 10 years old so I’m still 29 years old :)

• Current Role:
  – Consulting Systems Engineer for Routing and Switching Technologies
  – BU Advisor for R/S Technologies
  – Field Advisor for R/S SEVT (Systems Engineer Virtual Team)

• Other Cisco Live sessions:
  – LTRCRS-2004 - "VSS/Instant Access Lab"
Who is Adam Radford

• Personal:
  – Location: Sydney Australia
  – Other Interests: Triathlon – Swimming, cycling, running. Software based performance analysis of data for elite (world champion) athletes.
  – Married to a professional triathlete

• Background:
  – Joined Cisco September 1999
  – Before:
    • 10 years as software developer
    • First Web server (one of 5 in Australia) at University of New South Wales ran on my Sun Workstation under my desk

• Current Role:
  – Distinguished Systems Engineer
  – BU Advisor for R/S Technologies
  – DevNet Ambassador

• Other Cisco Live sessions:
  – DevNet-1007 - API Deep Dive: APIC EM Rest API
  – DevNet-1044 - Create a Hello World with APIC EM APIs
Who is Markus Harbeck ???

• Personal:
  – Location: Eschborn, Germany (near Frankfurt) but lives in Bavaria
  – Other Interests: My family, Horse back riding, motor cycling,
  – Other business interests: LISP, Multicast, MPLS, IPv6, SDN …

• Background:
  – Joined CISCO October 2010
  – Before; 12 years, operations, engineering, application engineering at Lufthansa Systems
  – Drives APIC-EM in EMEAR
  – LISP innovations – first customer project in Germany
  – Owner and Head of the Network Innovation Summit [http://cs.co/NIS2015](http://cs.co/NIS2015)

• Current Projects:
  – APIC-EM
  – Network Transformation
  – Network Automation
  – LISP
  – Other CL Sessions: [BRKRST-3045](http://example.com)

My Kids view on Network Design
Who is Wolfgang Riedel ???

• Personal:
  – Location: Erlangen, Germany (between Munich – Frankfurt)
  – Other Interests: Alpine Snowboarding, High-End Audio, AS5187, 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
    – 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 +14 years
    – Worked on Cat6k, Cat4k, N7k, ASR1k, FC, FCoE, DCB, UCS, N5k, N2k, N1k, PoE FEX, vPC, OTV, LISP (Pioneer Award), OF, ONE, SDN

• Current Projects:
  – ENG Architecture
  – Network Transformation
  – APIC-EM
  – DNS-AS
  – USP (Universal Service Platform)
  – CSFX
# Agenda - TECSDN-3600 (570 minutes)

<table>
<thead>
<tr>
<th>Agenda</th>
<th>Start</th>
<th>End</th>
<th>Duration</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>09:00 am</td>
<td>09:15 am</td>
<td>15 min</td>
<td>Wolfgang, Markus, Adam, Lila</td>
</tr>
<tr>
<td>World of Controllers</td>
<td>09:15 am</td>
<td>11:00 am</td>
<td>105 min</td>
<td>Wolfgang Riedel</td>
</tr>
<tr>
<td>Break</td>
<td>11:00 am</td>
<td>11:15 am</td>
<td>15 min</td>
<td>All</td>
</tr>
<tr>
<td>APIC-EM Architecture and Demo (Part 1)</td>
<td>11:15 am</td>
<td>01:15 pm</td>
<td>120 min</td>
<td>Lila Rousseaux &amp; Markus Harbeck</td>
</tr>
<tr>
<td>Lunch Break</td>
<td>01:15 pm</td>
<td>02:15 pm</td>
<td>60 min</td>
<td>All</td>
</tr>
<tr>
<td>APIC-EM Architecture and Demo (Part 2)</td>
<td>2:15 pm</td>
<td>2:45 pm</td>
<td>30 min</td>
<td>Lila Rousseaux &amp; Markus Harbeck</td>
</tr>
<tr>
<td>API and programmability (Part 1)</td>
<td>2:45 pm</td>
<td>04:15 pm</td>
<td>90 min</td>
<td>Adam Radford</td>
</tr>
<tr>
<td>Break</td>
<td>04:15 pm</td>
<td>04:30 pm</td>
<td>15 min</td>
<td>All</td>
</tr>
<tr>
<td>API and programmability (Part 2)</td>
<td>4:30 pm</td>
<td>05:00 pm</td>
<td>30 min</td>
<td>Adam Radford</td>
</tr>
<tr>
<td>Scale and High Availability</td>
<td>5:00 pm</td>
<td>06:15 pm</td>
<td>75 min</td>
<td>Markus Harbeck</td>
</tr>
<tr>
<td>Conclusion and Q&amp;A</td>
<td>6:15 pm</td>
<td>06:30 pm</td>
<td>15 min</td>
<td>Wolfgang, Markus, Adam, Lila</td>
</tr>
</tbody>
</table>
The World of Controllers
The APIC Story

TECSDN-3600 1/6

Wolfgang Riedel
Principal Engineer Engineering

ENG Product Management – Architecture
CCIE #13804, VCP #42559
wolfgang@cisco.com
Agenda

1. Introduction
2. SDN Controllers and Technologies
3. Application Policy Infrastructure Controllers (APIC-DC) for the Datacenter
4. Application Policy Infrastructure Controllers (APIC-EM) for the Enterprise
5. A Few Conclusions and Q&A if we have time

TECSDN-3600 (570 minutes)
The SDN Acronym Decoder Ring

✔ MSDC - Massive Scale Data Center
✔ EPDC – EnterPrise Data Center
✔ SDN – Software Defined Networking, Stanford Defined Networking, Still Don’t Know, …
✔ OF - Open Flow
✔ ODP, ODL - Linux Foundation OpenDaylight Project
✔ NfV – Network Function Virtualization, Services Chaining
✔ OS – OpenStack
✔ XNC – eXtensible Network Controller
✔ ONE – Open Networking Environment
✔ ACI – Application Centric Infrastructure
✔ onePK – Cisco ONE Platform Kit
✔ oneDK – Cisco ONE Developer Kit
✔ ESP, APIC–VTP - Cisco Evolved Services Platform for Service Providers
✔ Northbound-API – Application API (REST, WebSockets, OSGi)
✔ Southbound-API – Device APIs (onePK, OpenFlow, CLI, NetConf, …)
✔ SAL - Service Abstraction Layer
✔ MDSAL - Model Driven SAL (Netconf, OF-Config/OVSDB, OF x.y, PCEP, BGP-LS, …)
1. Introduction
Industry trends

OpenFlow Networking (Stanford clean slate) (2011)

Software Defined Networking (2012)

Open Daylight Project (2013)

Network Function Virtualization (2013)

Application Policy Infrastructure Controller (2014)
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
- Many smaller services, 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
**SDN – Still Don’t kNow – Stanford Defined Networking**

Many things to Many people

**Is the physical separation of control and data plane**

**Is managing the network through abstractions**

**Software Defined Networking**

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

- **Is the physical separation of control and data plane**
  - “An open solution for customized flow forwarding in the Data-Center”
  - “Is packet forwarding on general x86 compute”
  - “A way to define virtual networks with specific topologies for my multi-tenant Data-Center”
  - “A way to optimize multi-path traffic using new multi-path algorithms”

- **Is managing the network through abstractions**
  - “A platform for developing new control planes”
  - “An open solution for custom flow forwarding control in the Data-Center”
  - “A way to define virtual networks with specific topologies for my multi-tenant Data-Center”
  - “A way to optimize multi-path traffic using new multi-path algorithms”

- **Is whitebox routing and switching**
  - “A means to do traffic engineering without MPLS”
  - “A means to do traffic engineering without MPLS”
  - “A way to build my own security/encryption solution, avoiding RSA”
  - “A solution to get a global view of the network – topology and state”

- **Is running our network in a agile DEV-OPS model**
  - “A way to distribute policy/intent, e.g. for DDoS prevention, in the network”
  - “A way to configure my entire network as an entire device, not as individual devices”
  - “A way to build my own security/encryption solution, avoiding RSA”
  - “A solution to get a global view of the network – topology and state”

**SDN – Still Don’t kNow – Stanford Defined Networking**

Many things to Many people
# APIC - EM Design Points

## Validated with FAT & CAT

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstraction and Automation of Manual Network Operations</td>
<td></td>
</tr>
<tr>
<td>Reduce Network Complexity</td>
<td>Low Risk adoption of SDN</td>
</tr>
<tr>
<td>Brownfield Support – No Software / Hardware upgrade required</td>
<td>Brownfield Support – No Software / Hardware upgrade required</td>
</tr>
<tr>
<td>Start with small set of solvable problems</td>
<td>Start with small set of solvable problems</td>
</tr>
<tr>
<td>Elastic Services Infrastructure ensures scaling as adoption grows</td>
<td>Elastic Services Infrastructure ensures scaling as adoption grows</td>
</tr>
<tr>
<td>Advanced Visualization (HTML5/JavaScript) with object oriented interface</td>
<td>Advanced Visualization (HTML5/JavaScript) with object oriented interface</td>
</tr>
<tr>
<td>Advanced Visualization (HTML5/JavaScript) with object oriented interface</td>
<td>Advanced Visualization (HTML5/JavaScript) with object oriented interface</td>
</tr>
<tr>
<td>Auto-Translation of high level business intent into network control function</td>
<td>Auto-Translation of high level business intent into network control function</td>
</tr>
<tr>
<td>Advanced analytics for real time network visibility and faster response time</td>
<td>Advanced analytics for real time network visibility and faster response time</td>
</tr>
</tbody>
</table>

Don’t make it simpler by making it more complex!
APIC - Design Points
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
Declarative Over Imperative

Google - Containers At Scale

• Imperative:
  – **Imperative programming**: telling the "machine" how to do something, and as a result what you want to happen will happen.
  – Hardware implementation of almost all computers is imperative; nearly all computer hardware is designed to execute machine code, which is native to the computer, written in the imperative style.
  – Imperative management leads to “Snowflake” Servers, By-product of N different commands from M different users
  – "Start this container on that server"
  – OnePK, Address Device extensibility use-cases e.g. data-path APIs
  – OpenFlow, centralized controller (usually a clustered set of controllers) that act as the “brains” for your SDN

• Declarative:
  – **Declarative programming**: telling the "machine" what you would like to happen, and let the computer figure out how to do it.
  – Expresses the logic of a computation without describing its control flow
  – "Run 100 copies of this container with a target of <= 2 tasks down at any time"
  – OpFlex, distributing the intelligence out to the network fabric. The ‘controller’ is “declaring” application needs, then relays that need to the network fabric using the OpFlex protocol. The end devices then determine the proper method to meet the application requirements.

Pros:
  – Repeatable
  – "Set it and forget it"
  – Eventually consistent
  – Easily updatable
  – Stateless

Cons:
  – Tracing action/reaction can be difficult.
  – "I made a change, is it done?"
  – In a imperative world someone needs to build abstractions
Enterprise SDN customer asks

TYPICAL APPLE PRODUCT...

A GOOGLE PRODUCT...

YOUR COMPANY’S APP...

STUFFTHATHAPPENS.COM BY ERIC BURKE
“People who are really serious about software should make their own hardware.”

Alan Kay, 1982
Network Element Architecture

- Highly Scalable Platforms
- Simplified Management
- Consistent Network Services
- Highly resilient
- Physical, Virtual, and Stackable platforms
- Virtual Containers for network services
1.1 Analogies
Distributed Networking has worked

Resiliency/Scale has been proven
Distributed Networking has worked

However

Distributed Networking adds complexity to manage/comprehend
Admin still makes network behavior decisions

But uses controller to mask complexity
Flexible “Programmable” Interfaces

Allow Protocol/API choice while maintaining stack integrity

• Web UI
• YANG
• REST API

• CLI
• SNMP
• Web UI
• NETCONF
• XML
• onePK
• Openstack
Both at one time had direct admin control
Direction to abstract complexity
Network Management should follow Web Development

Focus on the What and not the How

Web Dev GUI

Controller

WWW

Network

WWW Admin

Network Admin

2005
Power Technologist

2010
Application Developers

2013
Non Technical Users

2014
Intent Networking

2015
Partial Automation

2018
Self Healing

2010
Application Developers

2013
Non Technical Users

2005
Power Technologist

2014
Intent Networking

2015
Partial Automation

2018
Self Healing
What I am doing next?

So this begs the Question?
2. Overview SDN controllers and techniques
Network Programmability Models

1. **Programmable APIs**
   - Applications
   - Vendor-specific APIs
   - Control Plane
   - Data Plane
   - OpenFlow and/or Vendor specific

2a. **Classic SDN**
   - Applications
   - Vendor-specific APIs
   - Control Plane
   - Data Plane
   - OpenFlow and/or Vendor specific

2b. **Hybrid “SDN”**
   - Applications
   - Vendor-specific APIs
   - Control Plane
   - Data Plane
   - OpenFlow and/or Vendor specific

3. **Network Virtualization Virtual Overlays**
   - Applications
   - Vendor-specific APIs
   - Control Plane
   - Data Plane
   - Virtual Control Plane
   - Virtual Data Plane
   - Overlay Protocols

4. **Policy Intent**
   - Applications
   - Vendor-specific APIs
   - Policy Authority
   - Policy Controller
   - Policy Plane
   - Policy Agent
   - Data Plane
   - Control Plane

TECSDN-3600 © 2015 Cisco and/or its affiliates. All rights reserved.
Cisco Public
SDN Controllers – Types
There’s nothing like “a 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
  - Cisco: VSM (N1kv), EVP
  - IBM: DOVE; MS: (Windows Server, Microsoft System Center)

- **SDN Open Flow Controller**
  - next slide

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

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
SDN Controllers – Types
There’s nothing like “a SDN 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 built on 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 Controller – Overview
CISCO Domain Controllers
Cisco Controller Architecture

Ueber Controller (peering point DB, policy repository)

Service Provider Applications
- XML + REST API

Enterprise Applications
- REST API

Data Center Applications
- REST API

Data Center Legacy
- REST API

Domain-Controller: ONS
- DSC (Dynamic Service Composer)
  - Network Info Database
  - Policy Infrastructure
  - Automation + VSM

Domain-Controller: PRIME
- APIC – EM (Enterprise Module)
  - Network Info Database
  - Policy Infrastructure
  - Automation

Domain-Controller: UCS Director
- APIC – DC (Datacenter Module)
  - Network Info Database
  - Policy Infrastructure
  - Automation + AVS

Domain-Controller: ONS
- CPOM
  - Network Info Database
  - Policy Infrastructure
  - Automation + VSM

- Network Devices
  - ASR9k – CRS – KVM
  - Third Party

- Network Devices
  - Catalyst, ASR, ISR, WLC
  - NEXUS 7k

- Network Devices
  - NEXUS 2k, 3k, 5k, 6k, 7k

- ACI
  - OpFlex
  - DFA
  - xml

- SSH, telnet, https, http, snmp, onePK

- netconf

- Network Info Database
  - Policy Infrastructure
  - Automation + VSM NG + ESP

- Network Devices
  - Catalyst, ASR, ISR, WLC
  - NEXUS 7k

- Network Devices
  - NEXUS 2k, 3k, 5k, 6k, 7k

- Network Info Database
  - Policy Infrastructure
  - Automation + AVS

- Network Info Database
  - Policy Infrastructure
  - Automation + VSM

TECSDN-3600 © 2015 Cisco and/or its affiliates. All rights reserved.
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
2.1 Prime Infrastructure and APIC EM
Prime Infrastructure Vision

2014-15
Infrastructure Management
Platform Capabilities

2015-16
Service Management
Service Awareness / End to End

2016-17
Dynamic Service Assurance
Cisco ONE Applications
Cisco Prime and Cisco APIC Enterprise Module

Operational Automation
- Policy and Service Definition
- Automated Assurance Provisioning
- Visualization, Trending and Analytics

Network Intelligence
- Device Layer Abstraction
- Network Control
- Policy Enforcement & Network Change
System of Record vs. System of Change

Prime Infrastructure
- Policy definition
- Historical reporting on events, performance and configuration
- Troubleshooting workflows
- Capacity Trending
- Predictive Analytics

APIC - EM
- Policy enforcement
- Discovery (for change)
- Topology (for change)
- PnP
- Network state monitoring
- Device abstraction
- Network Control
2.3 Dynamic Service Composer - Architecture
(aka Mozart – ESP – DSC)
Dynamic Services Composer – Elements
“SW suite for automated NFV service lifecycle management & orchestration”

- Prime Service Catalog (PSC)
- Prime Order Fulfillment (POF)
- Network Services Orchestrator (NSO) [Based on Tail-F NCS]
- Elastic Services Controller (ESC)
- Virtual Topology System (VTS)

- Optional products in DSC
- Core products in DSC
- DC Overlay SDN system, consisting of a controller called the Virtual Topology Controller (VTC) and a Forwarder called the Virtual Topology Forwarder (VTF)

All components are modular and can be bought standalone or as an integrated SW suite.
2.4 CISCO - Open SDN Controller
CISCO Open SDN Controller - Java Application
XNC as a controller product is planned for EOS

Advanced Feature Set vs. Opensource
Production Network Requirements
Import Topology from Inventory or other sources

Network Applications
- Cisco Sourced
- Customers
- 3rd Parties

Cisco Open SDN Controller

Northbound APIs
- OSGI
- RESTful

Advanced Components
- Authentication
- Troubleshooting

Flow Manager
- Controller Applications
- Monitor Manager
- Slice Manager
- Topology Independent Forwarding (TIF)

Infrastructure (Core)
- Dijkstra SPF
- L3 Interface
- Physical and Logical Topology Manager
- Service Abstraction Layer (SAL)

Forwarding Rules Manager
- ARP Handler
- Device Manager

Southbound APIs
- OpenFlow

Network Devices

Java Bundle

Java Provides Dynamic Component Linking
Published APIs Are Expandable
Abstraction for Future SB Protocols
Dynamic Protocol Plugins

Production Network Requirements
Advanced Feature Set vs. Opensource
Import Topology from Inventory or other sources

Nortbound APIs
- OSGI
- RESTful

Advanced Components
- Authentication
- Troubleshooting

Flow Manager
- Controller Applications
- Monitor Manager
- Slice Manager
- Topology Independent Forwarding (TIF)

Infrastructure (Core)
- Dijkstra SPF
- L3 Interface
- Physical and Logical Topology Manager
- Service Abstraction Layer (SAL)

Forwarding Rules Manager
- ARP Handler
- Device Manager

Southbound APIs
- OpenFlow

Network Devices
2.5 ODL Architecture
Open Daylight Controller - Java Application

Network Applications

OpenDaylight Controller

Northbound APIs

OSGI

RESTful

GUI

Basic Operation Infrastructure

Dijkstra SPF

Forwarding Rules Manager

Host Tracker

Physical and Logical Topology Manager

ARP Handler

Device Manager

Service Abstraction Layer: API-Driven (AD-SAL) / Model-Driven (MD-SAL)

Southbound APIs

OpenFlow

OpenFlow

NETWORK DEVICES
ODL - Hydrogen Implementation
AD-SAL / MD-SAL
Network Application Life Cycle (Today)
Hop-by-Hop API-Driven Architecture
Network Application Life Cycle (Tomorrow)
End-to-End Model-Driven Architecture
Model-Driven SAL

Applications

Northbound APIs (Generated & Handcrafted)

Network Service Plugin  Platform Service Plugin  Transformer/Adapter  Internal Plugin

Java & REST SAL APIs (Generated)

Abstraction Layer

Network

Java SAL APIs (Generated)

Netconf  OF-Config/OVSDB  OF x.y  PCEP  BGP-LS

Network Elements
APIC-EM
Service Architecture Detail

- **Grapevine Root**
  - Service Manager
  - Capacity Manager
  - Load Monitor
  - Service Catalog

- **Grapevine Client**
  - Service Monitor
  - Download Manager

- **Inventory**
  - GV Lib

- **Topology**
  - GV Lib

- **Policy Manager**
  - GV Lib

- **Identity Manager**
  - GV Lib

- **Message Bus /MQ**

- **RPC**

- **Data Store**
  - GV Logs, Audits, Configs, Images, NE & Service Data

- **Network Element**
  - CLI Plugin
  - OnePK Plugin
  - OF Plugin
  - Network Elements

- **ONEPK**

- **ODL/MD-SAL**

- **DAS**

Cisco Confidential
3. **Application Policy Infrastructure Controller** for the Datacenter (APIC-DC)
What is UCS
Service policies deployed on stateless computing hardware

SIM Card
Identity for a Phone

Service Profile
Identity for a Server

UCS Service Profile
Unified Device Management

- Network Policy
- Storage Policy
- Server Policy
What is ACI
Centralized Automation and Fabric Management

- OPEN RESTFUL APIS
- CENTRALIZED POLICY MODEL
- OPEN SOURCE

CONTROLLER

POLICY MODEL

NEXUS 9500 and 9300
What is APIC-DC
End Points

→ Things that connect to the fabric and use it to interface with other things
→ A compute, storage or service instance attaching to a fabric

ACI Fabric

NIC
vNIC
end-points [EP]
What is APIC-DC
End Points (EPs)

→ Things that connect to the fabric and use it to interface with other things
→ A compute, storage or service instance attaching to a fabric

A collection of end-points with identical network behavior form a …

… end-point group [ EPG ]

All EPs share common properties
→ Connectivity
→ Security/Access control
→ QoS
→ Services
→ …
What is APIC-DC
End Point Groups (EPGs)

EPG WEB

- EP
- EP
- EP

EPG APP SERVER

- policies

Allows to specify rules and policies on groups of physical or virtual end-points without understanding of specific identifiers and regardless of physical location.

Can flexibly map into

- application tier of multi-tier app
- segmentation construct (ala VLAN)
- a security construct
- ESX port group
- …

... end-point group [ EPG ]

All EPs share common properties

- Connectivity
- Security/Access control
- QoS
- Services
- …
What is APIC-DC
End Point 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.

End points in group WEB can access end-points in group APP SERVER according to rules specified in the contract.

EPG WEB

provider

contract

consumer

EPG APP SERVER

Filter identifies what traffic
Filter defines port ranges and options
OoS
Log
Redirect into SVC graph

Filter defines bi-directionally in the "provider" centric way

Action

Action identifies actions applied

Action
Application Network Profiles
Applying Contracts

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

Rules specifying communication between application tiers.

Programmable Infrastructure
ACI - Logical Network Provisioning of Stateless Hardware

Does your Enterprise Topology look like this? Would you like to?

Outside (Tenant VRF) — Web — App — DB

QoS — Service — QoS

Filter — QoS — Filter

Application Policy Infrastructure Controller

Non-Blocking Penalty Free Overlay

ACI Fabric

VM

Tenant VRF
What is APIC-DC
User-Driven, Policy-Based IT Infrastructure

Application / Workload Orchestration and Scheduler

Unified Information Model and API

Policy Controller Compute  |  Policy Controller Network Fabric  |  Policy Controller Storage

Application Graph (EP, EPG, graph edges)

Endpoint Group (EPG)

Application Profile  =  Compute Service Profile  +  Network Profile  +  Storage Service Profile

Designed from Its Foundation to Be Application-Centric
4.1 APIC-EM - Policy Infrastructure
Consumes

DB Contract
- MSSQL
- MySQL
- HTTP:

Provides

Contract

ACI Model will be extended for APIC EM Utilization

Filter
Named collection of L4 port ranges
- HTTP = [TCP], [80, 443]
- MSSQL = [UDP], [1433-1434]
- MySQL = [TCP], [3306, 25565]
**APIC-EM Policy Construct**

**Event Triggers**
- User-identifier (tenant/user)
- Application
- Device Type
- Location

**Network Users**
- User-identifier (tenant/user)
  - Application
  - Device Type
  - Location

**Resources**
- User-identifier (tenant/user)
  - Application
  - Device Type
  - Location

**Actions**
- Permit
- Deny
- Copy
- Monitor
- Redirect (L3, L4, L7)
- No copy
- No redirect

**Action Properties**
- Priority Level
- Resource Level
- Experience Level
- Trust Level
- Destination
- Sample Rate

---

**Policy Properties**
- Policy Creator
- Policy Name
- Policy Scope
- Policy Priority
- Policy Time:
  - Start Time
  - End Time
  - Hard timeout
  - Idle timeout
  - recurrence

**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 DC + ENT
Our Vision for a common policy Intent framework

Common Namespace
for Business Intent

API
DC - Controller

Application Intent

User Intent

API
ENT - Controller
APIC-EM
Modifications 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**

**Model extensions**

- Context Parameters are required to represent enterprise use case: Circumstances

- It has been extended to add hierarchies to model enterprise use cases:
  - EPGs can contain EPGs
  - Contracts can contain Contracts
  - Circumstances can contain Circumstances
Groups and Circumstances

- functional group
  - consumable
    - group
      - User group
      - resource group
        - resource ep
        - ep group
      - ep group
      - attached ep
    - ep
      - ep
    - circumstance
      - relator

APIC-DC
APIC-EM extensions
Common End-Points
APIC End 2 End Communication, do we talk?

Hey, I meant from a policy Intent point of view!
4.2 Grapevine – Auto Scale Architecture for APIC-EM
APIC-EM Grapevine

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

• In the real world, **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.

• Additionally, 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?
APIC-EM Grapevine

Grapevine, the 20,000 foot view

- Grapevine is a PaaS (Platform as a Service) solution.
- With Grapevine you would define "service bundles". Each “service bundle” deployed on Grapevine runs as a separate process.
- Grapevine can deploy a single instance of these services or multiple instances of these services, on the same server or across multiple 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 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.
- Grapevine and APIC-EM are decoupled from a technical perspective.
  - Grapevine is the 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.

But wait a second for Marcus session :)}
APIC-EM Grapevine

Grapevine Components: Services

Public Network

Load Balancer / Reverse Proxy

Private Network

SDN Service #1
SDN Service #2
SDN Service #3
...
SDN Service #N

Common Services

Data Store
SAL/PAL
MQ
Tasks / Events
AuthN / AuthZ
APIC-EM Grapevine
GV Deployment: Elastic Service Management Framework

• Mandatory Requirements:
  – Easy to adopt
  – Low cost of operation
  – Cloud-like user experience

• Goals:
  – Manages mix of physical / virtual machines
  – 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
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
4.3 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: 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

But wait a second for Lila’s session ;-)
7. Conclusion and Open Discussion
The World is CHANGING
And it’s **CHANGING FAST**

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

### Technology

- **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
  - Influence shift from from NetOps ⇒ DevOps
  - Shift in buying centers
  - Have we been unwilling or unable to abstract complexity.
  - Believe network teams do NOT have the skills and experience to implement and manage SDN

- **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**

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

- **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**

- **Economics**
  - Well... all of the above
  - RYF-complex (Fragile/Robust)
  - Product “de-siloing”
  - Does it really become “cheaper”???
# Continue Your Education

## Network Transformation Demos @ World of Solutions

<table>
<thead>
<tr>
<th>Campus Area</th>
<th>Demo Description</th>
<th>Staffers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EN</strong></td>
<td>Lifecycle Management of Wired and Wireless Networks</td>
<td>Mark B, Sowmya S, Sada H</td>
</tr>
<tr>
<td></td>
<td>Software Defined WAN with Prime Infrastructure and APIC-EM</td>
<td>Sumanth K, Madhavan A, Philip H</td>
</tr>
<tr>
<td></td>
<td>IWAN Application Experience (leverages Prime Infrastructure)</td>
<td>Joe A, Patrick C</td>
</tr>
<tr>
<td></td>
<td>Next Gen Workspace - Simplify campus network deployments and operations with Instant Access and the Virtual Switching System, which provide IPv4/v6 feature consistency and AVC support. (leverages Prime Infrastructure)</td>
<td>Divya Rao, Shawn Wargo</td>
</tr>
<tr>
<td></td>
<td>Converged Access with Services</td>
<td>Rahul K, Tarunesh A</td>
</tr>
<tr>
<td><strong>ACI</strong></td>
<td>SDN for Branch Service Automation (Prime Infrastructure &amp; Plug and Play: routing &amp; switching)</td>
<td>Arun T, Saurav P</td>
</tr>
<tr>
<td></td>
<td>SDN and Collaboration Solutions (APIC-EM MapCollab with EasyQOS)</td>
<td>Jon Snyder, Jim Coffman</td>
</tr>
<tr>
<td></td>
<td>SDN and Network Security (APIC EM &amp;SourFire, MACsec integration)</td>
<td>Gokul Nair, Blue Lang</td>
</tr>
<tr>
<td></td>
<td>SDN for Branch Service Automation (Prime Infrastructure &amp; Plug and Play: routing &amp; switching)</td>
<td>Yogesh Shetty, Prakash R, Ravi A,</td>
</tr>
<tr>
<td><strong>Data Center</strong></td>
<td>Operational Efficiency with ACI (Prime NAM)</td>
<td>Prem Chandran, Vittal K</td>
</tr>
</tbody>
</table>
## Continue Your Education
### Technical Breakout Sessions

<table>
<thead>
<tr>
<th>Session ID</th>
<th>Title</th>
<th>Session Length</th>
<th>Day</th>
<th>Time</th>
<th>Primary Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRKNMS-2443</td>
<td>One Management for Unified Access</td>
<td>120 Mins</td>
<td>Tues, Jan 27</td>
<td>2:15 pm</td>
<td>Tejas Shah</td>
</tr>
<tr>
<td>BRKNMS-2445</td>
<td>Improve Application Delivery with Cisco AVC in Data Center and Cloud</td>
<td>120 Mins</td>
<td>Friday, Jan 30</td>
<td>11:30 am</td>
<td>Prem Chandran, Vittal Krishnamurthy</td>
</tr>
<tr>
<td>BRKNMS-2701</td>
<td>How I Learned To Stop Worrying And Love Prime Infrastructure</td>
<td>90 Mins</td>
<td>Thurs, Jan 29</td>
<td>9 am</td>
<td>Joe Clarke, Jason Davis</td>
</tr>
<tr>
<td>BRKNMS-2695</td>
<td>Administration and monitoring of the Cisco DataCenter with Cisco Prime DCNM</td>
<td>90 Mins</td>
<td>Wed, Jan 28</td>
<td>2:30 pm</td>
<td>David Kirsch</td>
</tr>
<tr>
<td>BRKNMS-2845</td>
<td>AVC and Prime Infrastructure create Network Intelligence for Business Initiatives</td>
<td>90 Mins</td>
<td>Tues, Jan 27</td>
<td>11:15 am</td>
<td>Tony Hosseiny, Rashmi Ramesh</td>
</tr>
<tr>
<td>BRKEWN-2011</td>
<td>Managing an Enterprise WLAN with Cisco Prime Infrastructure</td>
<td>120 Mins</td>
<td>Thurs, Jan 29</td>
<td>9 am</td>
<td>Paul Lysander</td>
</tr>
<tr>
<td>COCCOL-2011</td>
<td>Inside Cisco IT: Zero Touch Deployment using Cisco Prime Infrastructure</td>
<td>90 Mins</td>
<td>Wed, Jan 28</td>
<td>11:30 am</td>
<td>Marc De Preter</td>
</tr>
<tr>
<td>BRKNMS-1036</td>
<td>IT Operations Management in the SDN Era – with Prime Infra and APIC Controllers</td>
<td>90 Mins</td>
<td>Thurs, Jan 29</td>
<td>11:30 am</td>
<td>Ronnie Ray</td>
</tr>
<tr>
<td>BRKNMS-2007</td>
<td>Data Center Management with Cisco Prime</td>
<td>90 Mins</td>
<td>Thurs, Jan 29</td>
<td>2:30 pm</td>
<td>Prakash R, Sowmya S</td>
</tr>
<tr>
<td>BRKCRS-3011</td>
<td>APIC-EM (Application Policy Infrastructure Controller - Enterprise Module) - SDN in the Enterprise</td>
<td>120 Mins</td>
<td>Thurs, Jan 29</td>
<td>9 am</td>
<td>Wolfgang Riedel</td>
</tr>
</tbody>
</table>
## Continue Your Education
### Technical Seminars/Labs

<table>
<thead>
<tr>
<th>Session ID</th>
<th>Title</th>
<th>Session Length</th>
<th>Day</th>
<th>Time</th>
<th>Primary Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>PN1LNMS-3000</td>
<td>Cisco Live Network and NOC: Panel Discussion</td>
<td>90 Mins</td>
<td></td>
<td></td>
<td>Mark McKillop</td>
</tr>
<tr>
<td>TECSDN-3600</td>
<td>APIC Enterprise Module – SDN in the Enterprise</td>
<td>9.5Hours</td>
<td>Mon, Jan 26</td>
<td>9 am</td>
<td>Marcus Harbeck, Lila R, Adam Radford, Wolfgang Riedel</td>
</tr>
<tr>
<td>7463</td>
<td>Network and Application monitoring with Prime Infrastructure 2.2 and NAM 6.1</td>
<td>4 Hours</td>
<td>Thurs/Fri, Jan 29/30</td>
<td>9 am</td>
<td>Gilles Clugnac, Stefan Haertlein</td>
</tr>
<tr>
<td>Table Topics</td>
<td>Whatever is on your mind</td>
<td>2 Hours</td>
<td>29.01.2015</td>
<td>12:30 – 14:30</td>
<td>Wolfgang Riedel</td>
</tr>
<tr>
<td>MTE</td>
<td>Meet the Engineer</td>
<td>TDB</td>
<td>TDB</td>
<td>9 am</td>
<td>Marcus Harbeck, Lila R, Adam Radford, Wolfgang Riedel</td>
</tr>
</tbody>
</table>
## Continue Your Education
### DevNet Sessions

<table>
<thead>
<tr>
<th>Session ID</th>
<th>Title</th>
<th>Session Length</th>
<th>Day</th>
<th>Time</th>
<th>Primary Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>DevNet-1007</td>
<td>API Deep Dive: APIC EM Rest API</td>
<td>60 Mins</td>
<td>Tues, Jan 27</td>
<td>12:30 pm</td>
<td>Adam Radford</td>
</tr>
<tr>
<td>DevNet-1044</td>
<td>Create a Hello World with APIC EM APIs</td>
<td>60 Mins</td>
<td>Wed, Jan 28</td>
<td>11:30 am</td>
<td>Adam Radford</td>
</tr>
<tr>
<td>DevNet-1022</td>
<td>Let's discuss: Cisco's Controllers – Why, What, How, When</td>
<td>60 Mins</td>
<td>Wed, Jan 28</td>
<td>2:30 pm</td>
<td>Christine Bakan, Phil Casini, Mike Cohen</td>
</tr>
</tbody>
</table>
Some more fun stuff to watch…

• Fundamentals of Cisco APIC-EM
  – [https://www.youtube.com/watch?v=17IDRT9tuWY](https://www.youtube.com/watch?v=17IDRT9tuWY)

• Metadata-Defined Data Center, Mike Dvorkin, Cisco Systems

• Developing OpenDaylight Apps with MD-SAL
  – [https://www.youtube.com/watch?v=uBnDJNsd6Qo](https://www.youtube.com/watch?v=uBnDJNsd6Qo)

• Application Centric Infrastructure (ACI) Overview
  – [http://www.youtube.com/watch?v=VZWwjNAiUpI](http://www.youtube.com/watch?v=VZWwjNAiUpI)

• APIC EM Demo, Apr 2014 - VT Recording
Participate in the “My Favorite Speaker” Contest
Promote Your Favorite Speaker and You Could be a Winner

• Promote your favorite speaker through Twitter and you could win $200 of Cisco Press products (@CiscoPress)

• Send a tweet and include
  – Your favorite speaker’s Twitter handle @AS51871
  – Two hashtags: #CLUS #@AS51871

• You can submit an entry for more than one of your “favorite” speakers

• Don’t forget to follow @CiscoLive and @CiscoPress

• View the official rules at http://bit.ly/CLUSwin
Complete Your **Online Session Evaluation**

- Please complete your online session evaluations after each session. Complete 4 session evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt.

- All surveys can be completed via the Cisco Live Mobile App or the Communication Stations
Thank you.