探索SDN(Software Defined Network) 如何跨越網路與應用的藩籬

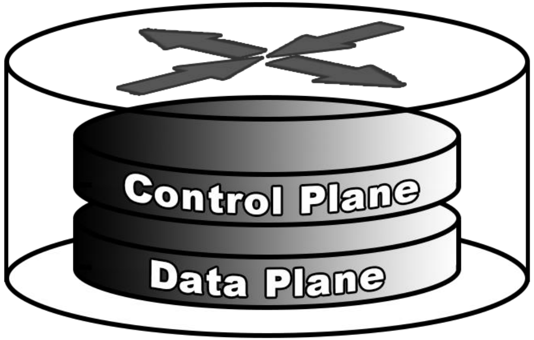
林瑝錦(jerrylin@cisco.com)

Cisco Systems

2013/Aug/30

什麼是SDN

The Network Paradigm As We Know It



Processing Plane	Where it runs	How fast these processes run	Type of processes performed
Control Plane	Switch CPU	In the order of thousands of packets per second	Routing protocols (i.e. OSPF, IS-IS, BGP), Spanning Tree, SYSLOG, AAA (Authentication Authorization Accounting), NDE (Netflow Data Export), CLI (Command Line interface), SNMP
Data Plane	Dedicated Hardware ASIC's	Millions or Billions of packets per second	Layer 2 switching, Layer 3 (IPv4 | IPv6) switching, MPLS forwarding, VRF Forwarding, QOS (Quality of Service) Marking, Classification, Policing, Netflow flow collection, Security Access Control Lists

Control and Data Plane resides within Physical Device

"…In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications…"

https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf

"…open standard that enables researchers
to run experimental protocols in campus networks. Provides standard hook for researchers to run experiments, without exposing internal working of vendor devices……"

http://www.openflow.org/wp/learnmore/

“A way to optimize link utilization in my network enhanced, application driven routing”

“An open solution for customized flow forwarding control in and between Data Centers”

“A platform for developing new control planes”

“An open solution for VM mobility in the Data-Center”

“A solution to automated network configuration and control”

“Develop solutions at software speeds: I don’t want to work with my network vendor or go through lengthy standardization.”

“A way to reduce the CAPEX of my network and leverage commodity switches”

“A means to get assured quality of experience for my cloud service offerings”

“A solution to build a very large scale layer-2 network”

“A means to do traffic engineering without MPLS”

“A solution to build virtual topologies with optimum multicast forwarding behavior”

Diverse Drivers
Common Concepts
Different Execution Paths

“A way to scale my firewalls and load balancers”

“A means to scale my fixed/mobile gateways and optimize their placement”

“A way to optimize broadcast TV delivery by optimizing cache placement and cache selection”

“A way to build my own security/encryption solution”

“A way to distribute policy/intent, e.g. for DDoS prevention, in the network”

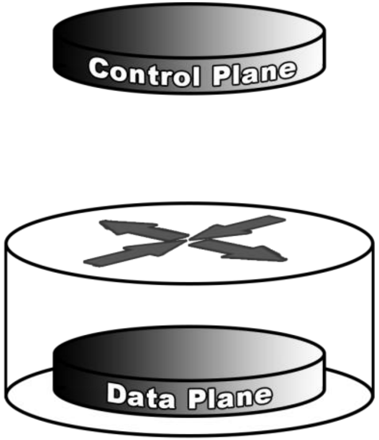
“A way to configure my entire network as a whole rather than individual devices”

“A solution to get a global view of the network – topology and state”

Simplified Operations – Enhanced Agility – New Business Opportunities

What is SDN?

(per Wikipedia definition)

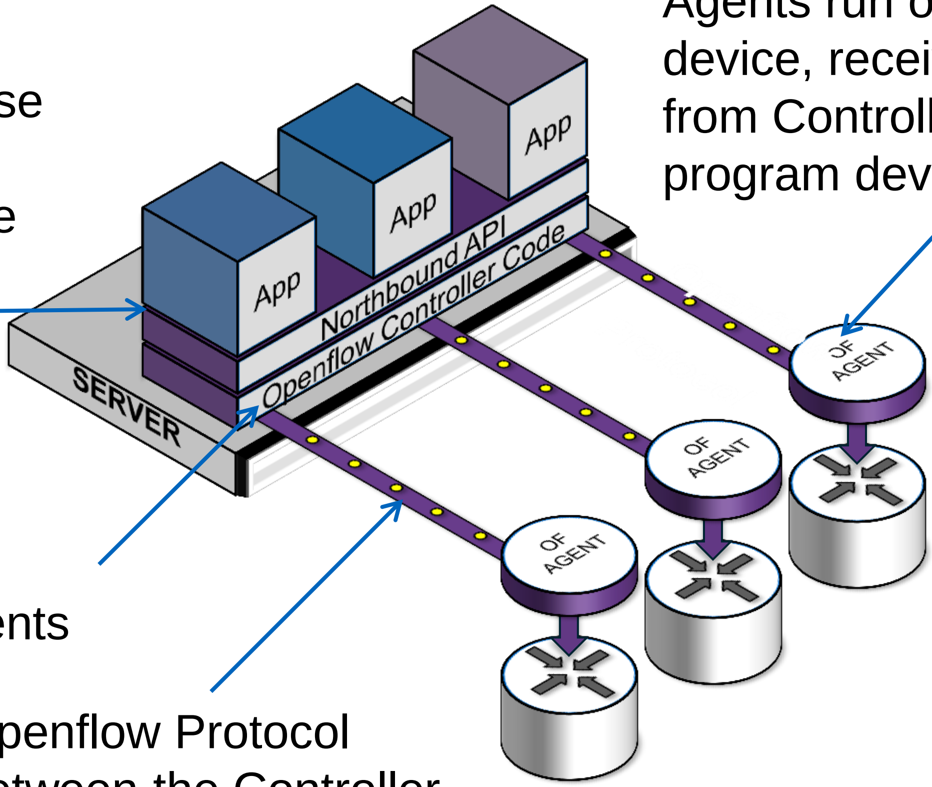


Software defined networking (SDN) is an approach to building computer networks that separates and abstracts elements of these systems

What is Openflow?

“Network enabled” applications can make use of Northbound API to request services from the network

Agents run on the network device, receive instructions from Controller and program device tables



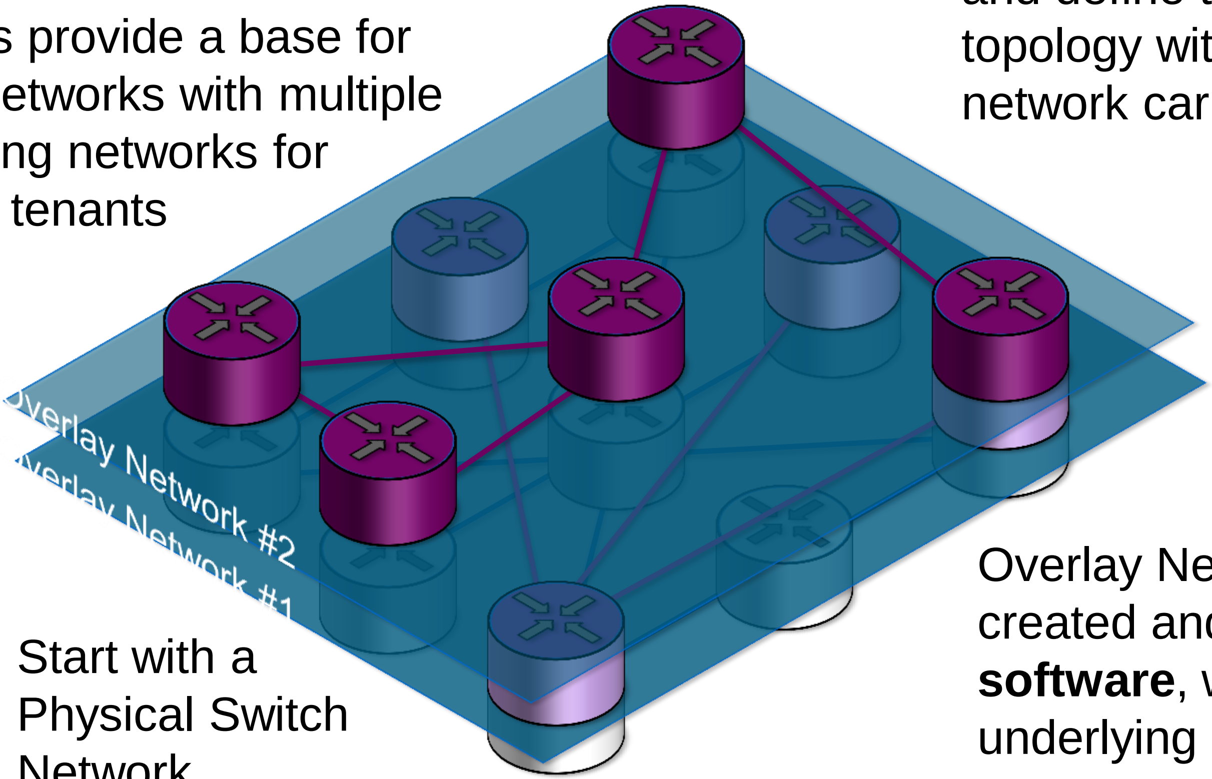
Central Administration and Operations point for Network Elements

Openflow Protocol between the Controller and the Agents.

Virtual Overlay Networks

Overlays provide a base for logical networks with multiple co-existing networks for different tenants

Logical "switch" devices overlay the physical network and define their own topology with the physical network carrying the data



Start with a Physical Switch Network

Overlay Networks can be created and torn down, with **software**, without changing underlying physical network

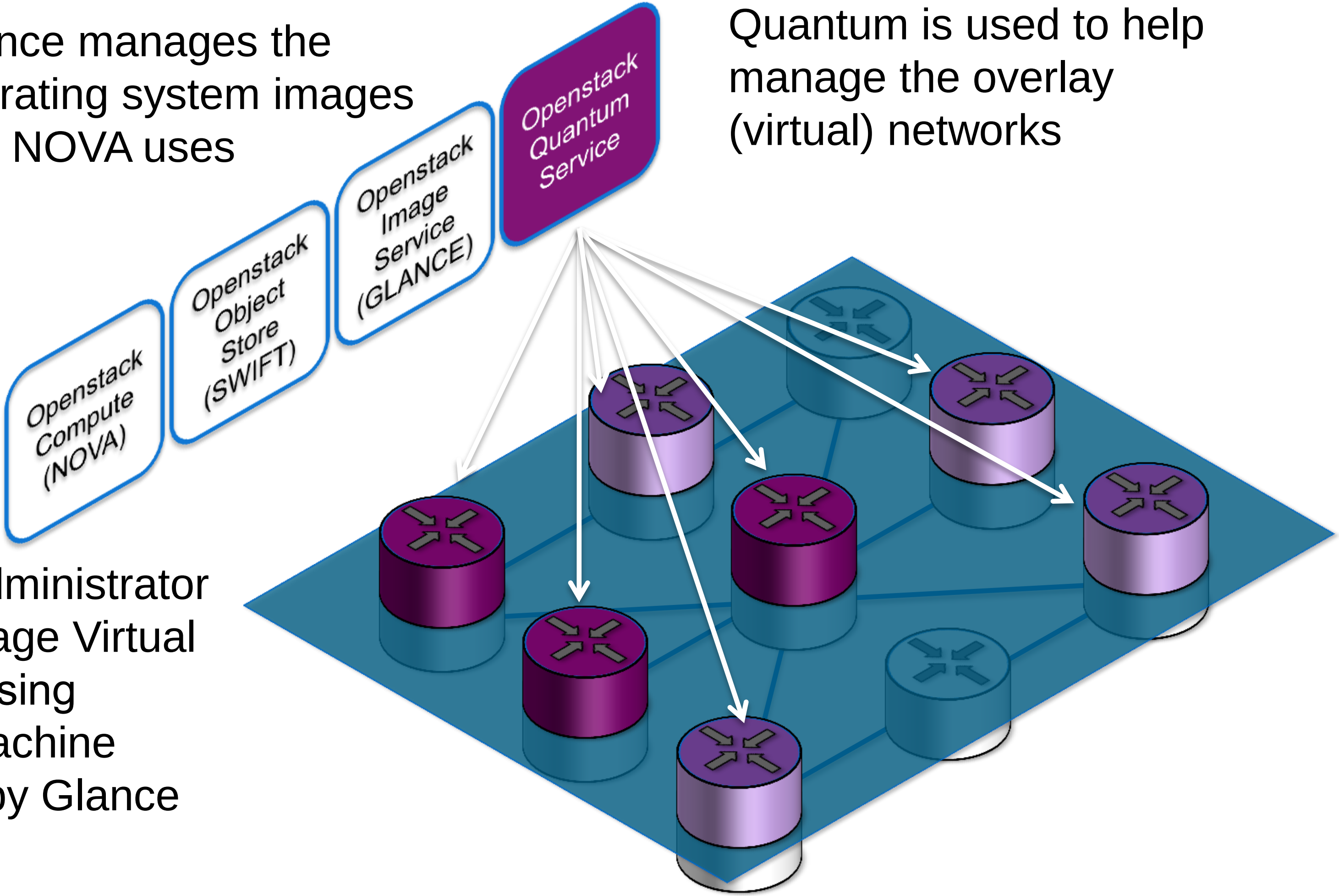
Openstack is an IAAS (Infrastructure As A Service) cloud computing project

It is also referred to as a Cloud Operating System

"…provides a means to control (administer) compute, storage, network and virtualization technologies…"

Glance manages the operating system images that NOVA uses

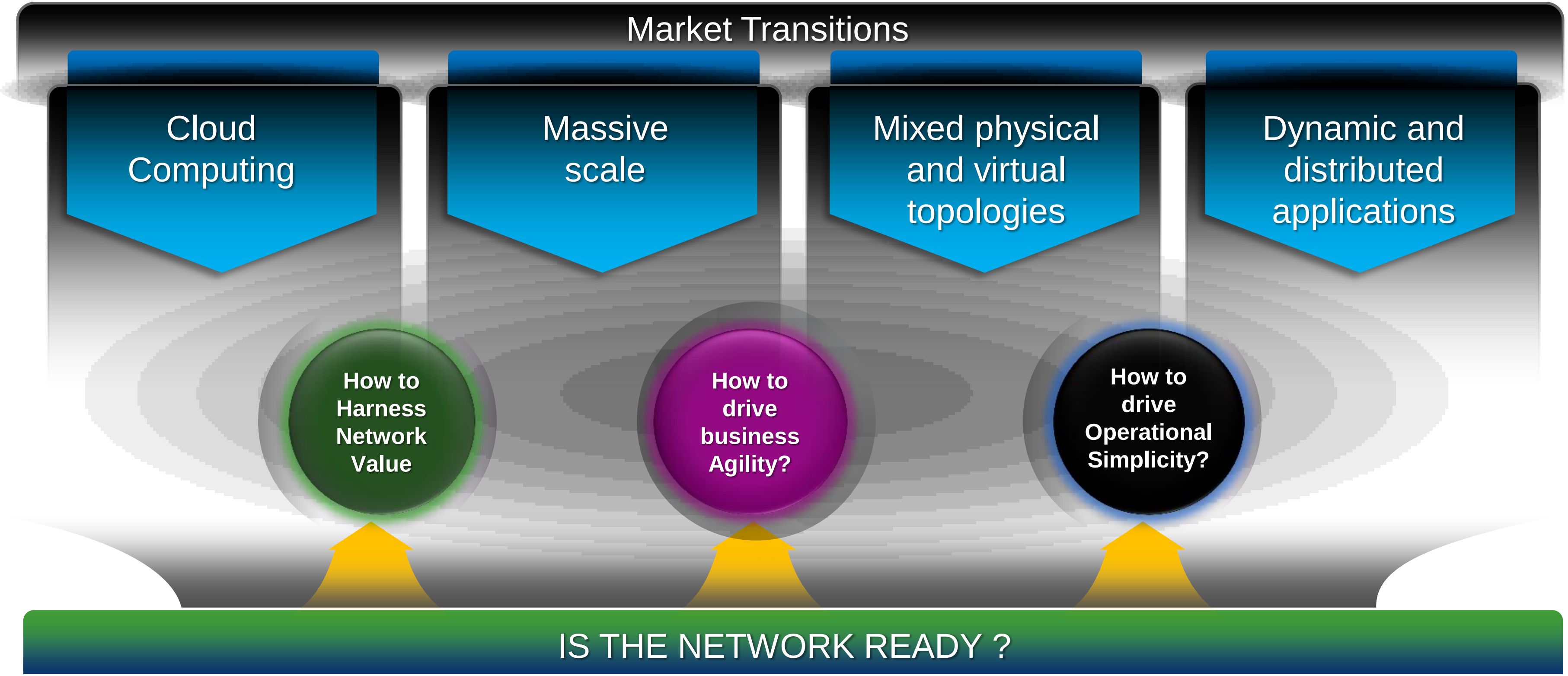
Quantum is used to help manage the overlay (virtual) networks


Openstack Compute (NOVA)
Openstack Object Store (SWIFT)
Openstack Image Service (GLANCE)
Openstack Quantum Service

Nova allows the administrator to create and manage Virtual Machines (VM’s) using various (stored) machine images managed by Glance

What SDN is turning out to be

Market Transitions Driving Greater Demands on the Network



Market Transitions

Cloud Computing

Massive scale

Mixed physical and virtual topologies

Dynamic and distributed applications

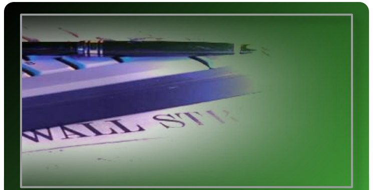
How to Harness Network Value

How to drive business Agility?

How to drive Operational Simplicity?

IS THE NETWORK READY ?

Customer Insights: Network Programmability



Research/ Academia

Experimental OpenFlow/SDN components for production networks

Network "Slicing"

Massively Scalable Data Center

Customize with Programmatic APIs to provide deep insight into network traffic

Network Flow Management

Cloud

Automated provisioning and programmable overlay, OpenStack

Scalable Multi-Tenancy

Service Providers

Policy-based control and analytics to optimize and monetize service delivery

Agile Service Delivery

Enterprise

Virtual workloads, VDI, Orchestration of security profiles

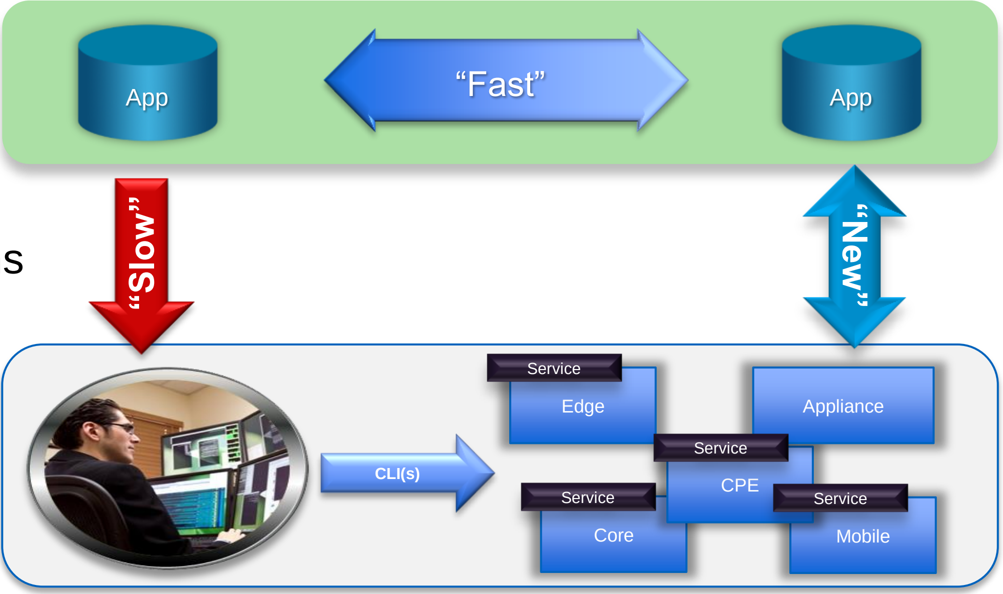
Private Cloud Automation

Diverse Programmability Requirements Across Segments
Most Requirements are for Automation & Programmability

Towards Programmatic Interfaces to the Network

Approaching Today's Application Developer Dilemma

- Many Network Applications today:
  - OTT – for speed and agility
  - Avoid network interaction – complex and slow innovation
- New Model for Network Applications
  - Keep speed and agility
  - Full-duplex interaction with the network across multiple planes – extract, control, leverage network state

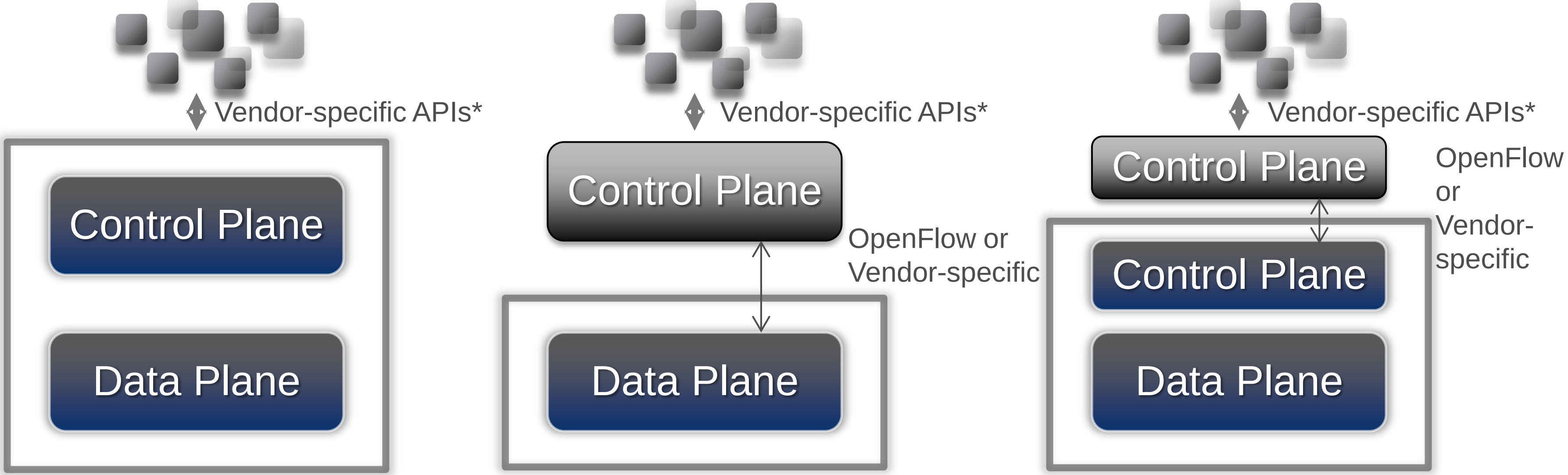


A New Programming Paradigm Is Needed

Network Programmability Models

Physical or Virtual

Current switch/router | “SDN” Approach | Hybrid Model?



Vendor-specific APIs*

Control Plane

Data Plane

Vendor-specific APIs*

Control Plane

OpenFlow or Vendor-specific

Data Plane

Vendor-specific APIs*

Control Plane

OpenFlow or Vendor-specific

Control Plane

Data Plane

Resilient, Scalable, Secure, Rich Features, Evolutionary, Investment Protection

Simpler (fewer nodes to manage)
Centralized Topology View

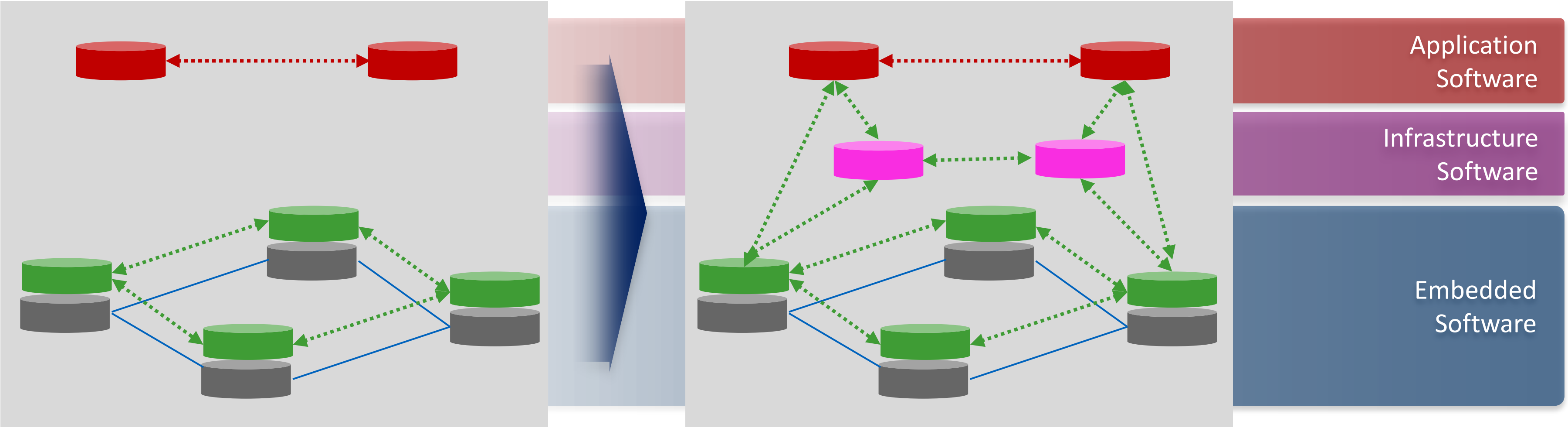
Combined Benefits

* Standards based over time

Openstack & Network Overlays apply to all models (physical / virtual)

Towards an Open Network Environment

Evolve the Control- and Management Plane Architecture

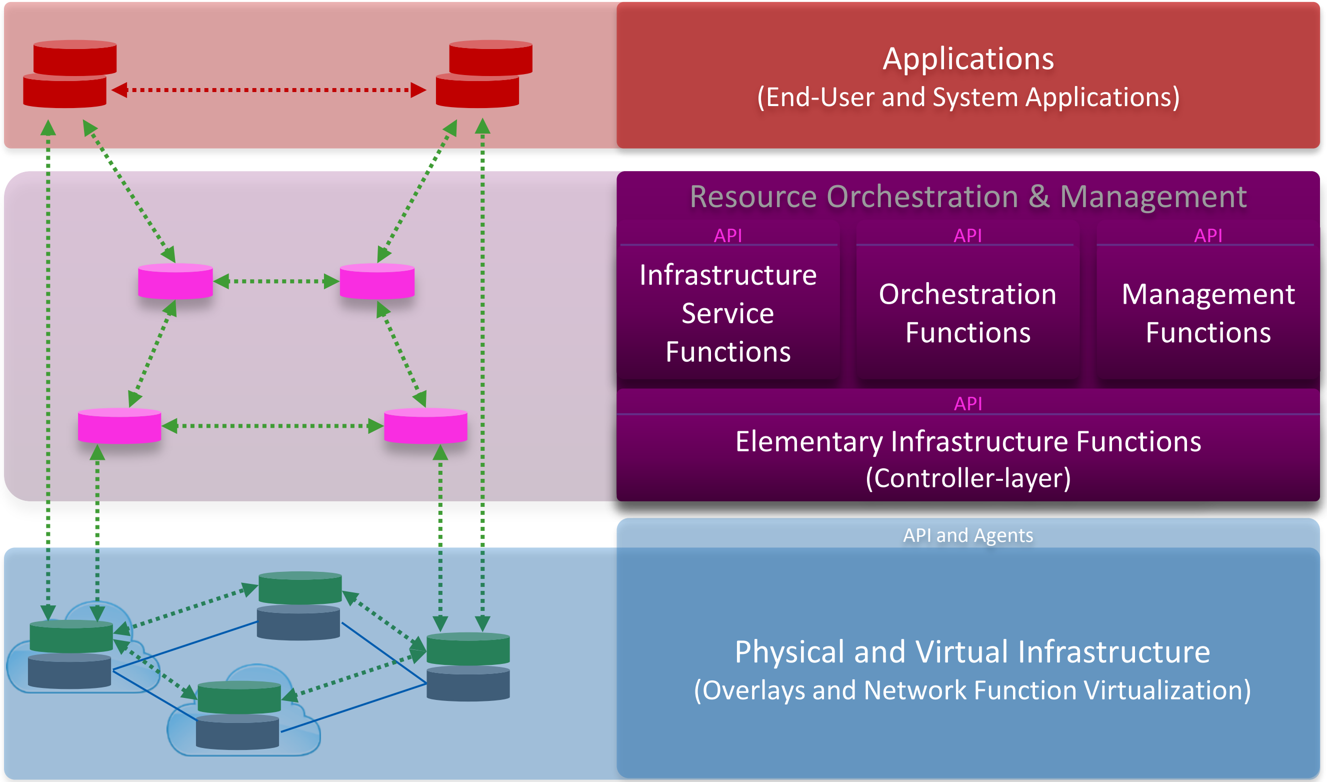


Fully Distributed Control Plane:
Optimized for reliability

Hybrid Control plane:
Distributed control combined with logically centralized control for optimized behavior
(e.g. reliability and performance)

Open Network Environment

The Next Step: Infrastructure Software Platform



Applications
(End-User and System Applications)

Resource Orchestration & Management

API

Infrastructure Service Functions

API

Orchestration Functions

API

Management Functions

API

Elementary Infrastructure Functions
(Controller-layer)

API and Agents

Physical and Virtual Infrastructure
(Overlays and Network Function Virtualization)

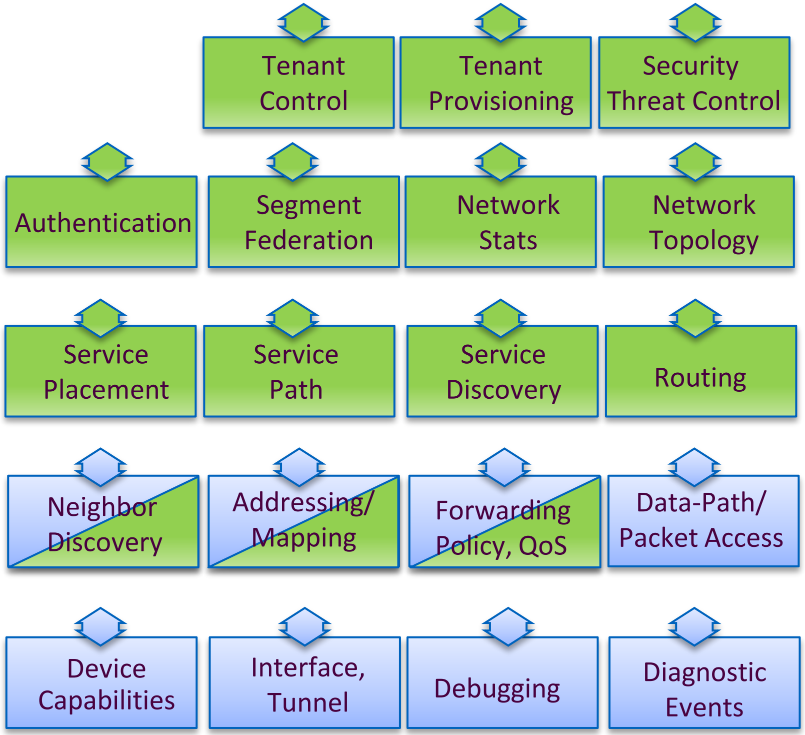
Application Software

Infrastructure Software

Embedded Software

Approaching abstractions for Networking

- Abstractions allow the definition of associated APIs
  - Enable API platform kit across all platforms, to integrate with development environments
  - Accelerate development of network applications: Completely integrated stack from device to network
  - Multiple deployment modes (local and remote (blade/server) based APIs)
  - Multiple Language Support (C, Java, Python…)
  - Integrate with customer development to deliver enhanced routing, forwarding..



Tenant Control | Tenant Provisioning | Security Threat Control

Authentication | Segment Federation | Network Stats | Network Topology

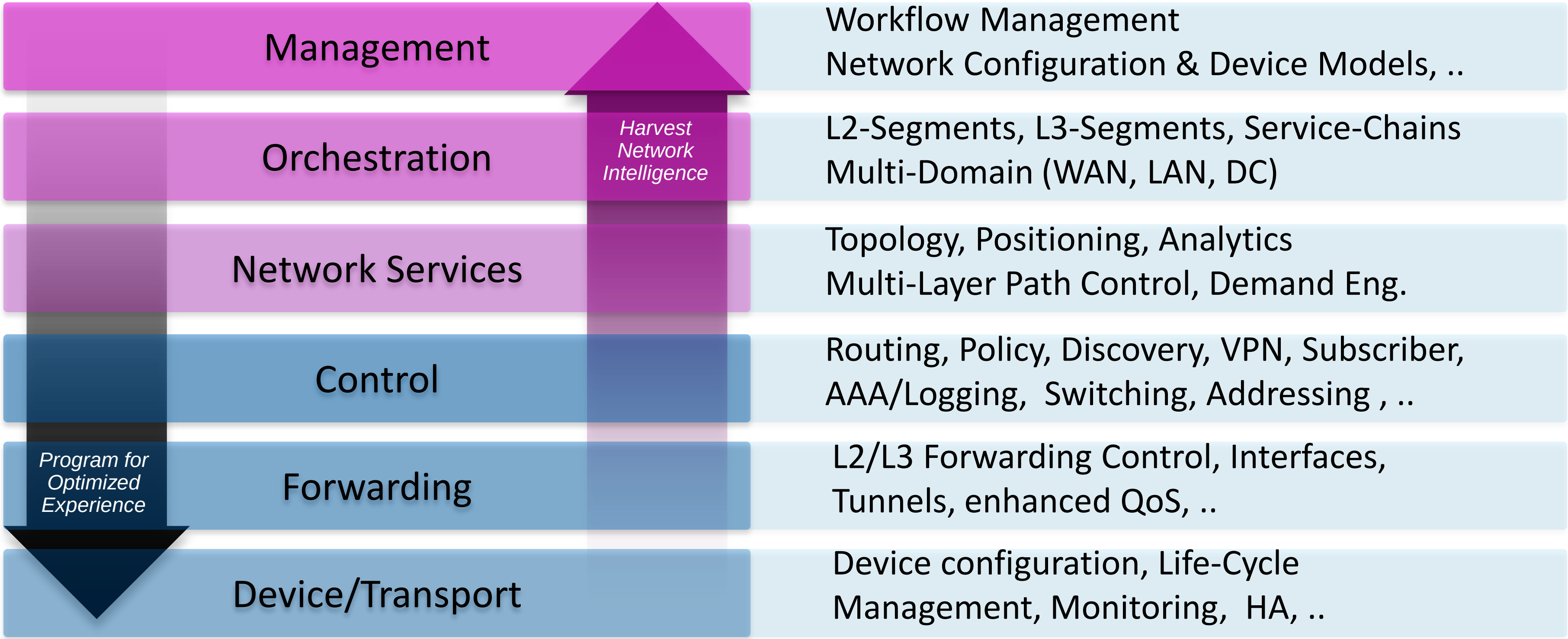
Service Placement | Service Path | Service Discovery | Routing

Neighbor Discovery | Addressing/ Mapping | Forwarding Policy, QoS | Data-Path/ Packet Access

Device Capabilities | Interface, Tunnel | Debugging | Diagnostic Events

🟦 Device focused abstraction 🟩 Service/Network focused abstractions

Full-Duplex, Multi-Layer/Multi-Plane APIs



Harvest Network Intelligence

Program for Optimized Experience

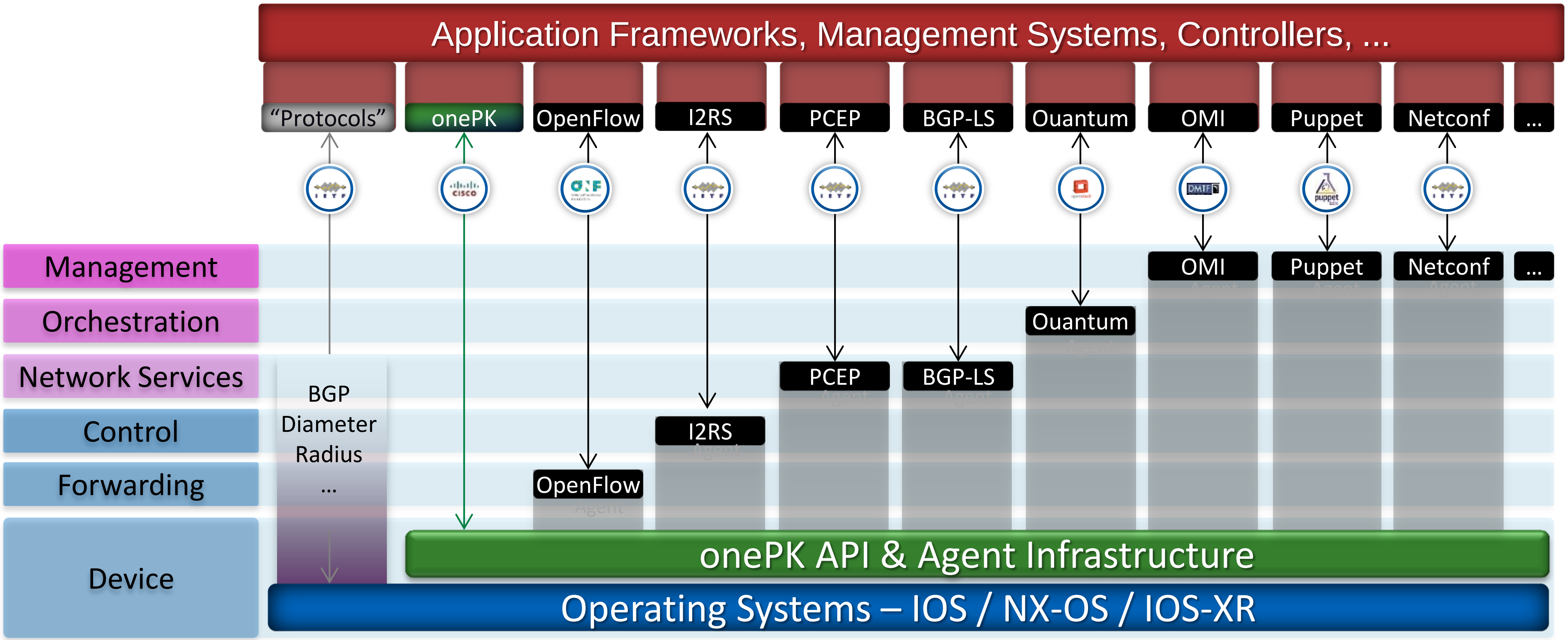
Management	Workflow Management Network Configuration & Device Models, ..
Orchestration	L2-Segments, L3-Segments, Service-Chains Multi-Domain (WAN, LAN, DC)
Network Services	Topology, Positioning, Analytics Multi-Layer Path Control, Demand Eng.
Control	Routing, Policy, Discovery, VPN, Subscriber, AAA/Logging, Switching, Addressing , ..
Forwarding	L2/L3 Forwarding Control, Interfaces, Tunnels, enhanced QoS, ..
Device/Transport	Device configuration, Life-Cycle Management, Monitoring, HA, ..

Full-Duplex, Multi-Layer/Multi-Plane APIs

Industry Examples

Management	Workflow Management Network Configuration & Device Models, ..	DMTF	Network Models - Interfaces (OMI)
Orchestration	L2-Segments, L3-Segments, Service-Chains Multi-Domain (WAN, LAN, DC)	openstack	OpenStack, Quantum API
Network Services	Topology, Positioning, Analytics Multi-Layer Path Control, Demand Eng.	IETF	Positioning (ALTO) Path Control (PCE)
Control	Routing, Policy, Discovery, VPN, Subscriber, AAA/Logging, Switching, Addressing , ..	IETF	Interface to the Routing System (I2RS)
Forwarding	L2/L3 Forwarding Control, Interfaces, Tunnels, enhanced QoS, ..	ONF OPEN NETWORKING FOUNDATION	OpenFlow Protocol
Device/Transport	Device configuration, Life-Cycle Management, Monitoring, HA, ..	ETSI	Network Function Virtualization (NfV)

Programmatic Network Access
Agents as Flexible Integration Vehicles



Application Frameworks, Management Systems, Controllers, ...

"Protocols"

onePK

OpenFlow

I2RS

PCEP

BGP-LS

Quantum

OMI

Puppet

Netconf

...

Management

Orchestration

Network Services

Control

Forwarding

Device

BGP
Diameter
Radius
...

OMI Agent

Puppet Agent

Netconf Agent

...

Quantum Agent

PCEP Agent

BGP-LS Agent

I2RS Agent

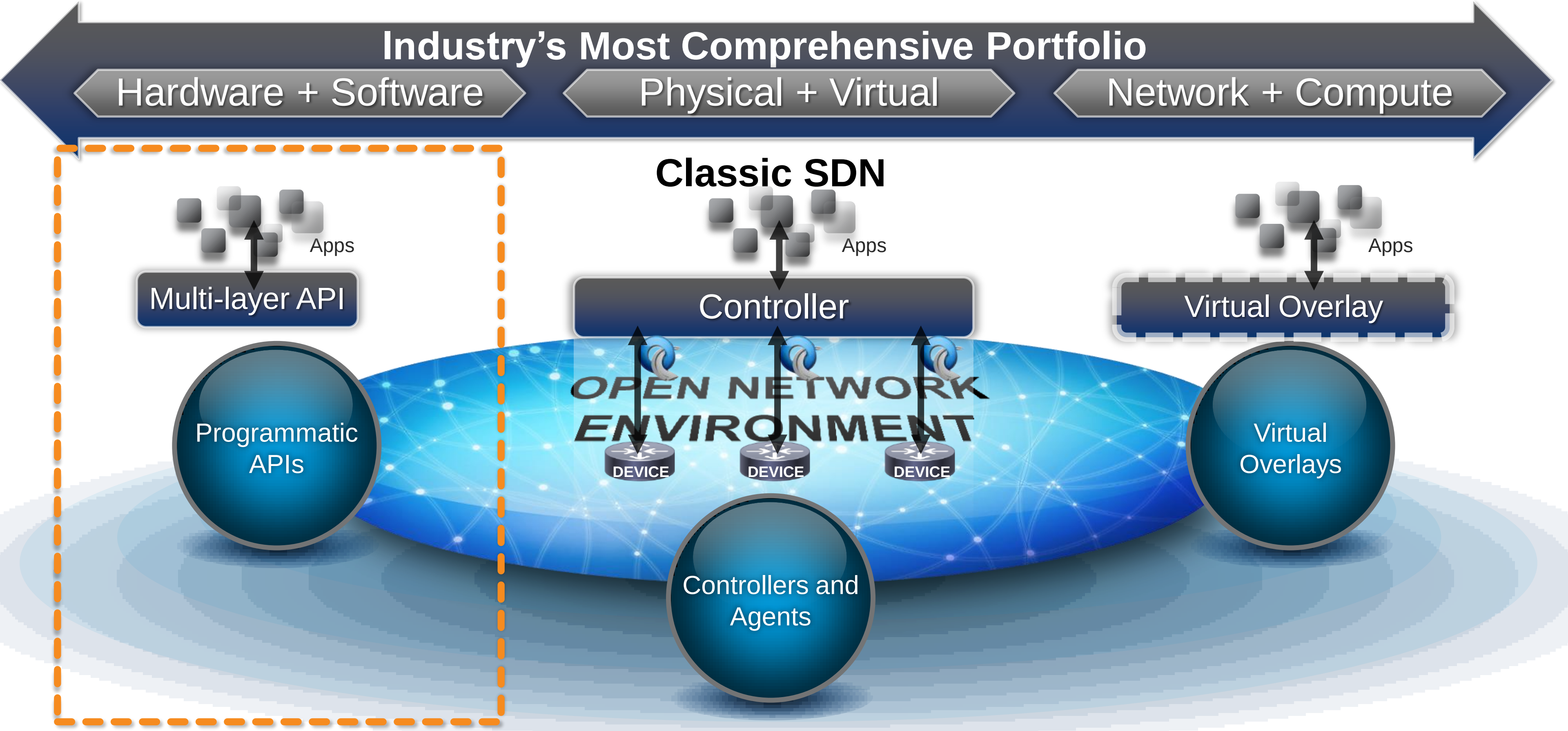
OpenFlow Agent

onePK API & Agent Infrastructure

Operating Systems – IOS / NX-OS / IOS-XR

Cisco Open Network Environment (ONE)

Cisco Open Network Environment



Industry's Most Comprehensive Portfolio

Hardware + Software

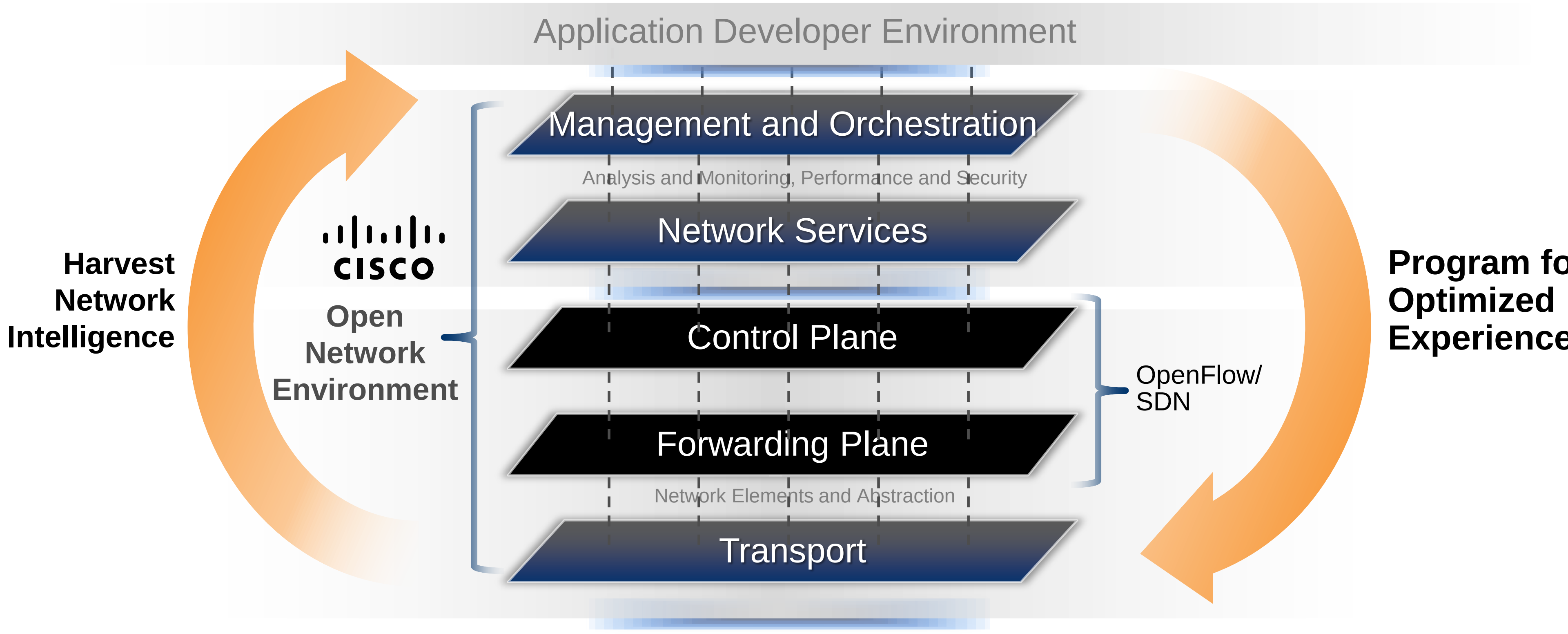
Physical + Virtual

Network + Compute

Classic SDN

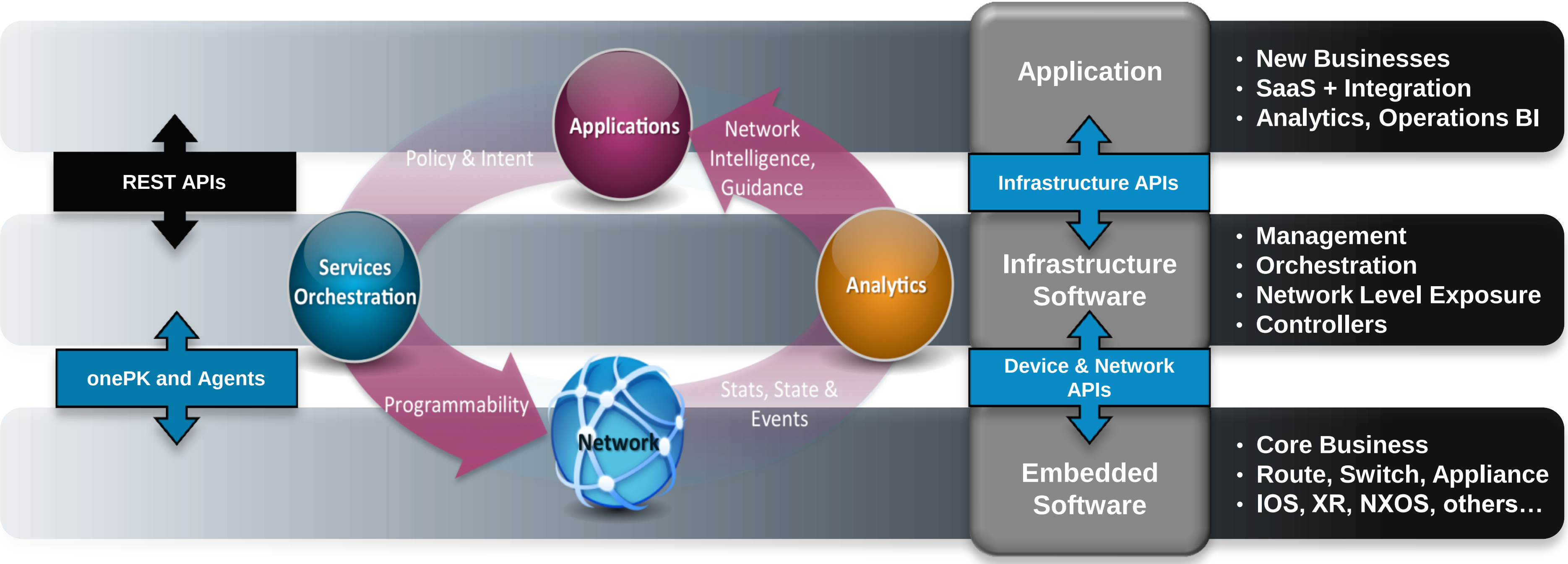
Cisco’s Differentiation: Multi-layered Programmability

Flexibility in Deriving Abstractions


Application Developer Environment
Management and Orchestration
Analysis and Monitoring, Performance and Security
Network Services
Control Plane
Forwarding Plane
Network Elements and Abstraction
Transport
Harvest Network Intelligence
cisco
Open Network Environment
OpenFlow/ SDN
Program fo
Optimized
Experience

Faster, Smarter, Simpler

Business Applications Enabled by Cisco ONE



- New Businesses
- SaaS + Integration
- Analytics, Operations BI

- Management
- Orchestration
- Network Level Exposure
- Controllers

- Core Business
- Route, Switch, Appliance
- IOS, XR, NXOS, others…

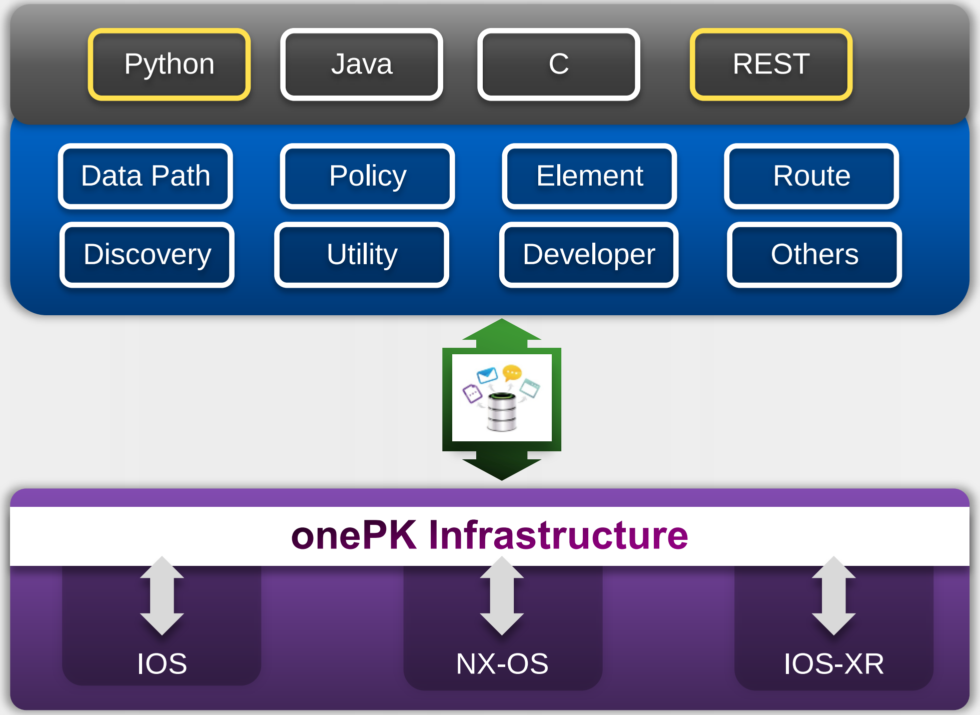
Introducing One Platform Kit (onePK)

DEVELOPER ENVIRONMENT
- Language of choice
- Programmatic interfaces
- Rich data delivery via APIs

COMPREHENSIVE SERVICE SETS
- Better apps
- New services
- Monetization opportunity

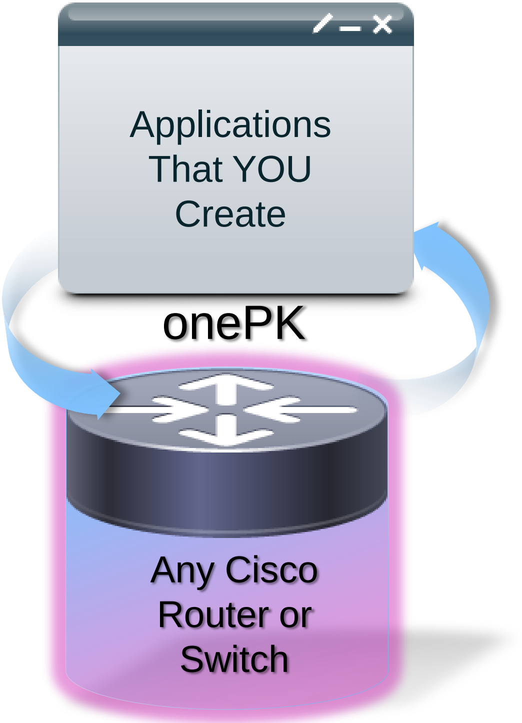
DEPLOY
- On a server blade
- On an external server
- Directly on the device

CONSISTENT PLATFORM SUPPORT
- IOS
- NX-OS
- IOS-XR



APIs make Abstractions available to Programmers

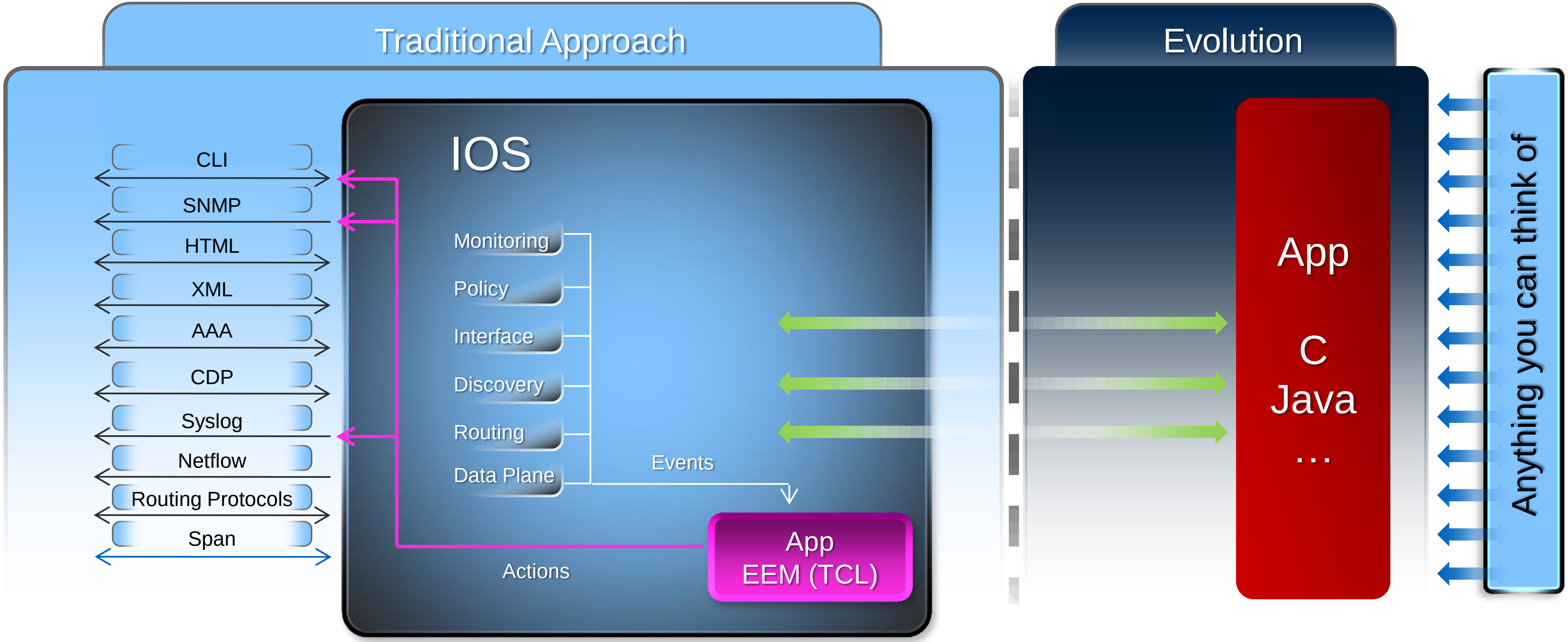
Example: Cisco's onePK (one Programming Kit) – Get your build on!



Flexible development environment to:

- Innovate
- Extend
- Automate
- Customize
- Enhance
- Modify

Evolving How We Interact With The Network Operating System

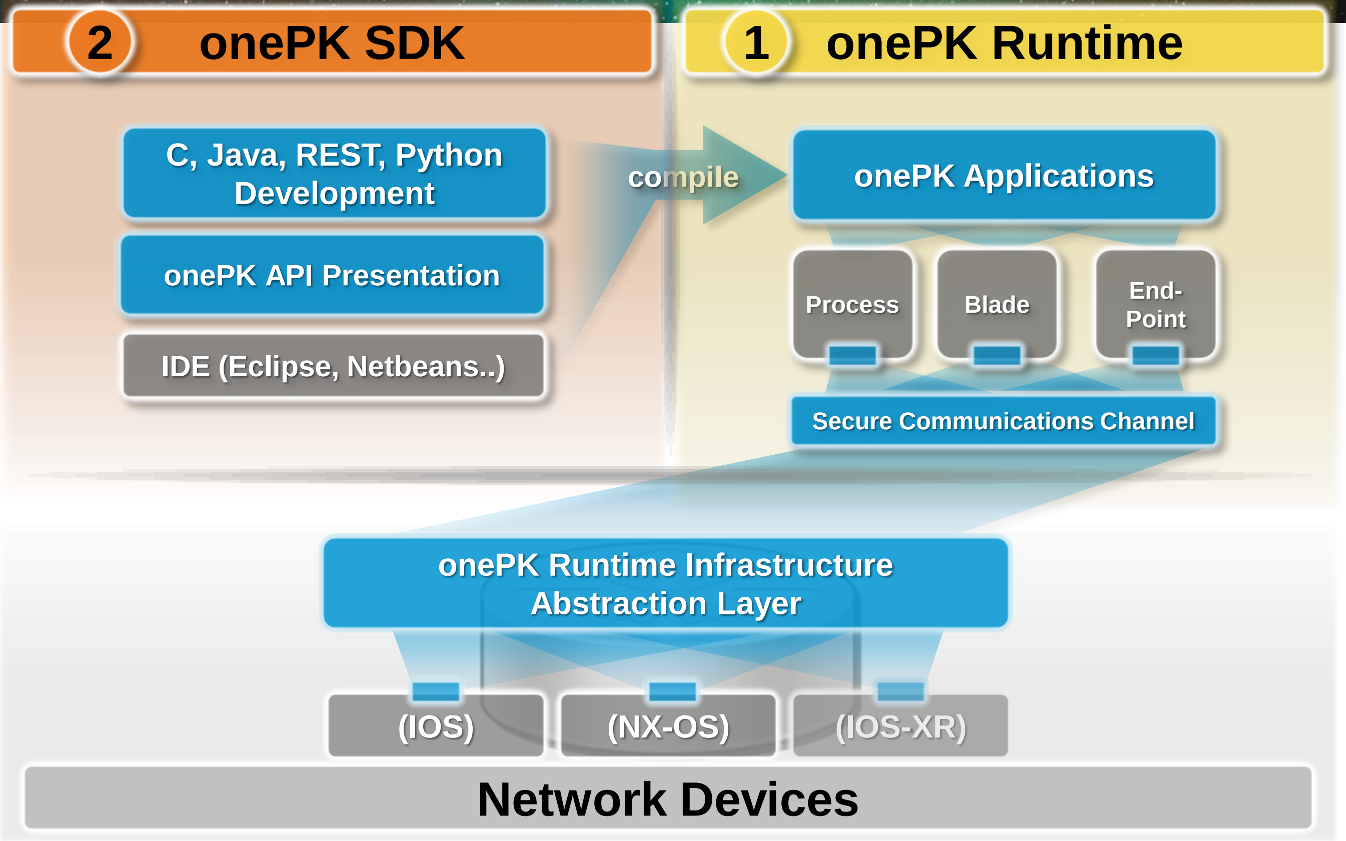


onePK Architecture Overview: What is onePK?

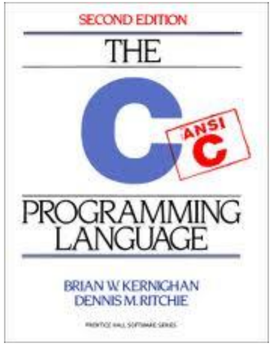
1 onePK represents an abstraction layer and unifying API that resides within all of Cisco's network software systems (IOS, IOS-XR, NX-OS).

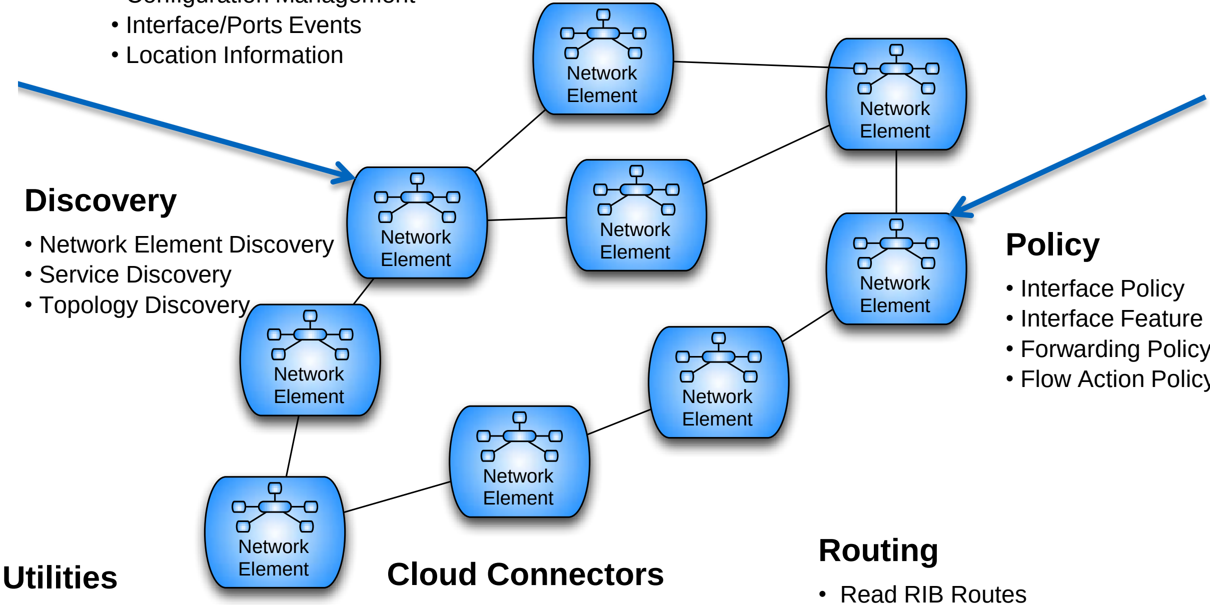
2 onePK SDK is an easy-to-use toolkit for development, automation and rapid service.

onePK is a key element within Cisco's announced Open Network Environment SDN strategy.


2 onePK SDK
C, Java, REST, Python Development
onePK API Presentation
IDE (Eclipse, Netbeans..)
compile
1 onePK Runtime
onePK Applications
Process
Blade
End-Point
Secure Communications Channel
onePK Runtime Infrastructure Abstraction Layer
(IOS)
(NX-OS)
(IOS-XR)
Network Devices

Languages and Service Sets





Element

- Element Capabilities
- Configuration Management
- Interface/Ports Events
- Location Information

Developer

- Debug Capabilities
- Tracing Interfaces
- Management Extensions

Data Path

- Packet/Flow Classifiers
- Copy/Punt/Inject
- Statistics

Discovery

- Network Element Discovery
- Service Discovery
- Topology Discovery

Policy

- Interface Policy
- Interface Feature Pc
- Forwarding Policy
- Flow Action Policy

Utilities

- Syslog Events and Queries
- AAA Interface
- Path Trace

Cloud Connectors

- Identity
- QoS
- Provisioning

Routing

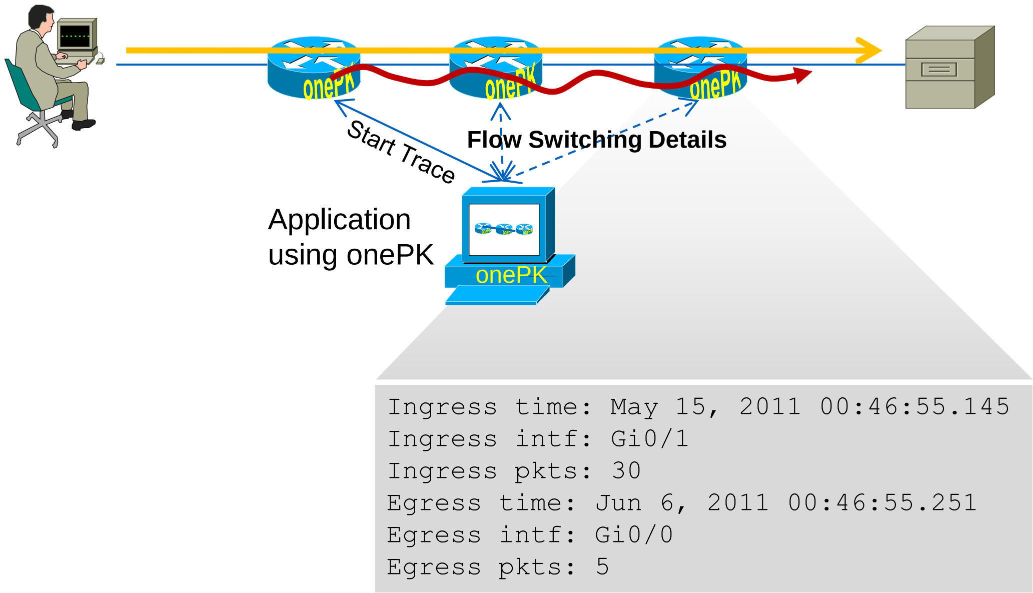
- Read RIB Routes
- Add/Delete Application Routes
- RIB Events (Route up/down)

Web Services for the Real World
RESTful Web Services
O'REILLY
Leonard Richardson & Sam Ruby

APIs at work – Element APIs

Example: Statistics, Diagnostics & Troubleshooting

- Objective:
  - Provide operators/ administrators/ support engineers with details about how packets flow through the network.
  - Reveal network issues
- Approach
  - NMS application leverages onePK APIs to show path of flow, timestamp, ingress/egress interfaces, interface packet counts



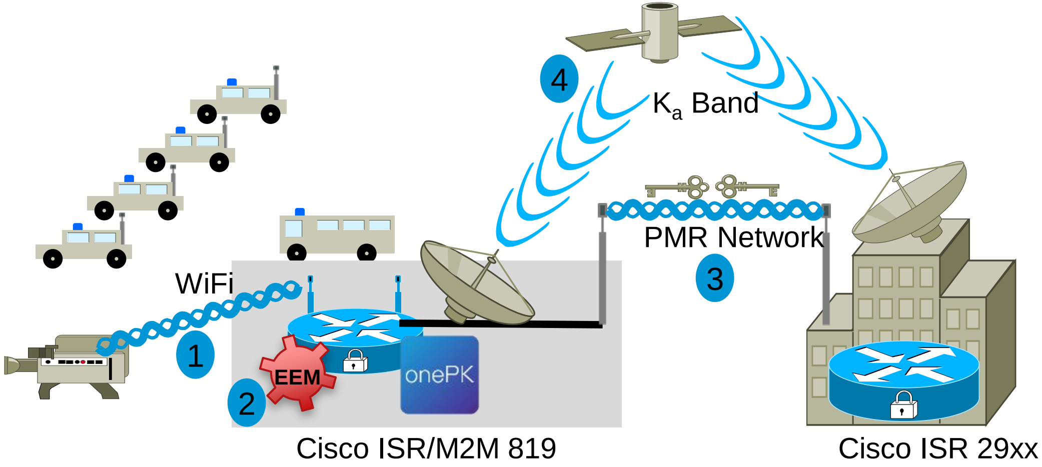
```
Ingress time: May 15, 2011 00:46:55.145
Ingress intf: Gi0/1
Ingress pkts: 30
Egress time: Jun 6, 2011 00:46:55.251
Egress intf: Gi0/0
Egress pkts: 5
```

Example: Emergency Response Network

Problem: How to deliver secure, trusted, robust, cost-effective broadband connectivity to mobile emergency response units?

Solution: Use Network Programming based on Cisco onePK and Cisco IOS Embedded Event Manager to integrate low-cost, high-bandwidth options with accredited legacy radio connectivity

Design: Pramacom (the key customers: Ministry of Interior of Czech Republic and Ministry of Interior of Slovak Republic)

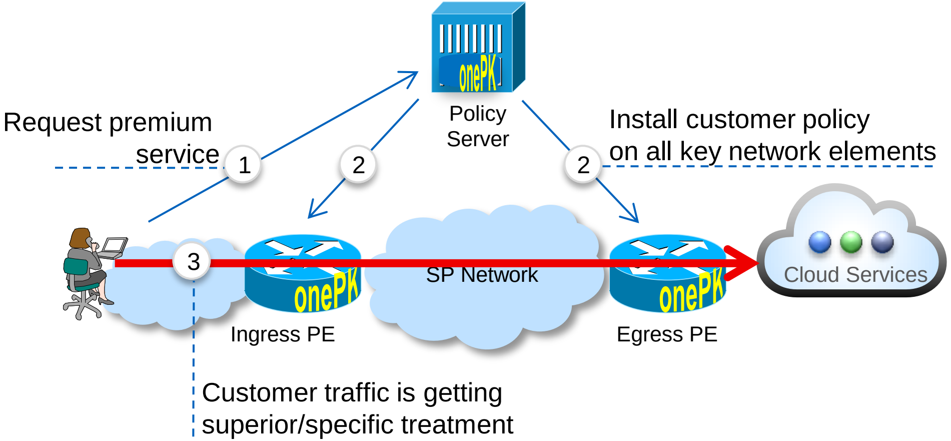


1. Connect high-bandwidth forward clients via WiFi
2. Use Cisco IOS EEM for onboard system integration and adaptation
3. Use Cisco onePK to redirect IKE key exchange out-of-band via legacy radio
4. Secure IPSec tunnel via cost-effective high bandwidth K_a Band
5. Reliable, secure emergency response network saving ~4M€ operating cost annually

APIs at work – Place in the Network APIs

Example: Dynamic Bandwidth/QoS Allocation

- Business Problem
  - Enable superior experience for subscribers which access a particular cloud service
- Solution
  - Install customer policy (QoS, access control,..) using onePK on key networking elements, e.g. Provider Edge (PE) routers
  - Similarities to broadband “Bandwidth on Demand” use cases
    - Broadband: Policy controlled on Subscriber-Gateway (BRAS/BNG, GGSN/PGW, ..) only
    - Common API like onePK enables control points on all key networking devices



Request premium service

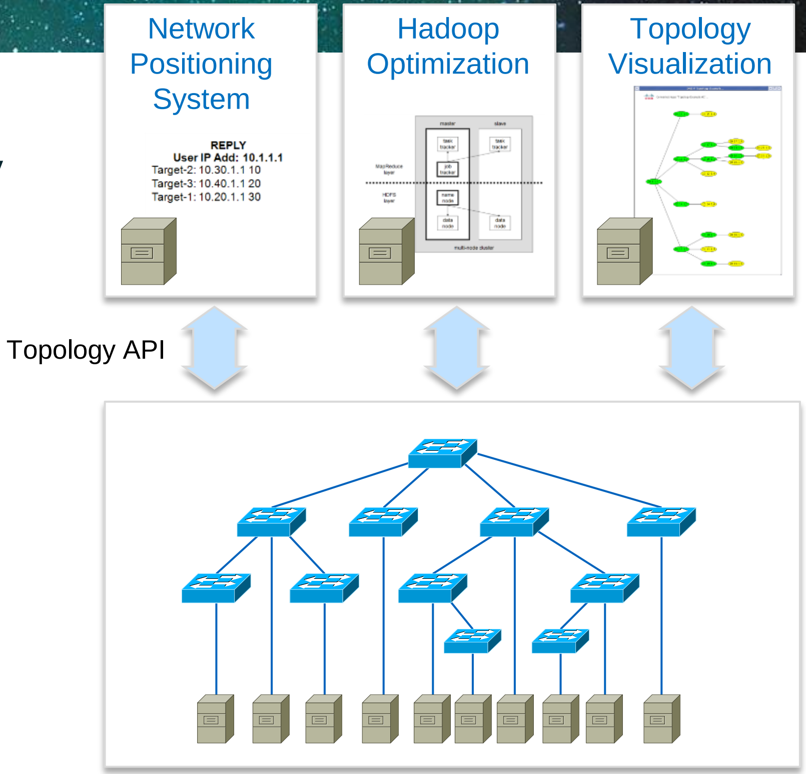
Install customer policy on all key network elements

Customer traffic is getting superior/specific treatment

APIs at work – Area APIs

Examples: Topology graph

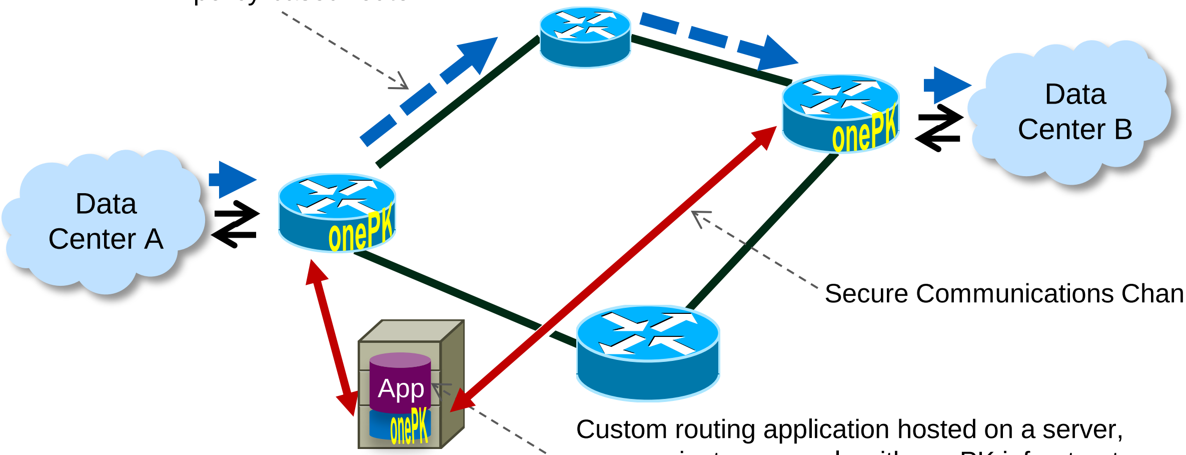
- Business Problem
  - Several problems require a view of the network topology (area, domain, or whole network)
  - Examples:
    - Locate optimal service out of a given list
    - Optimize Load Placement
    - Visualize the active Network Topology
- Solution
  - Topology API to expose network topology to applications, such as
    - NPS (for service selection)
    - Hadoop (for optimal job placement)
    - NMS (for topology visualization)



APIs at work – Area APIs

Example: Custom Routing

- Business Problem
  - Network operator needs to direct traffic using unique or external decision criteria; e.g route long lived elephant flows, like backup traffic differently
- Solution
  - Custom route application built and deployed using onePK, communicating directly with the forwarding plane.
  - Unique data forwarding algorithm highly optimized for the network operator’s application.



Select packets take a custom policy-based route

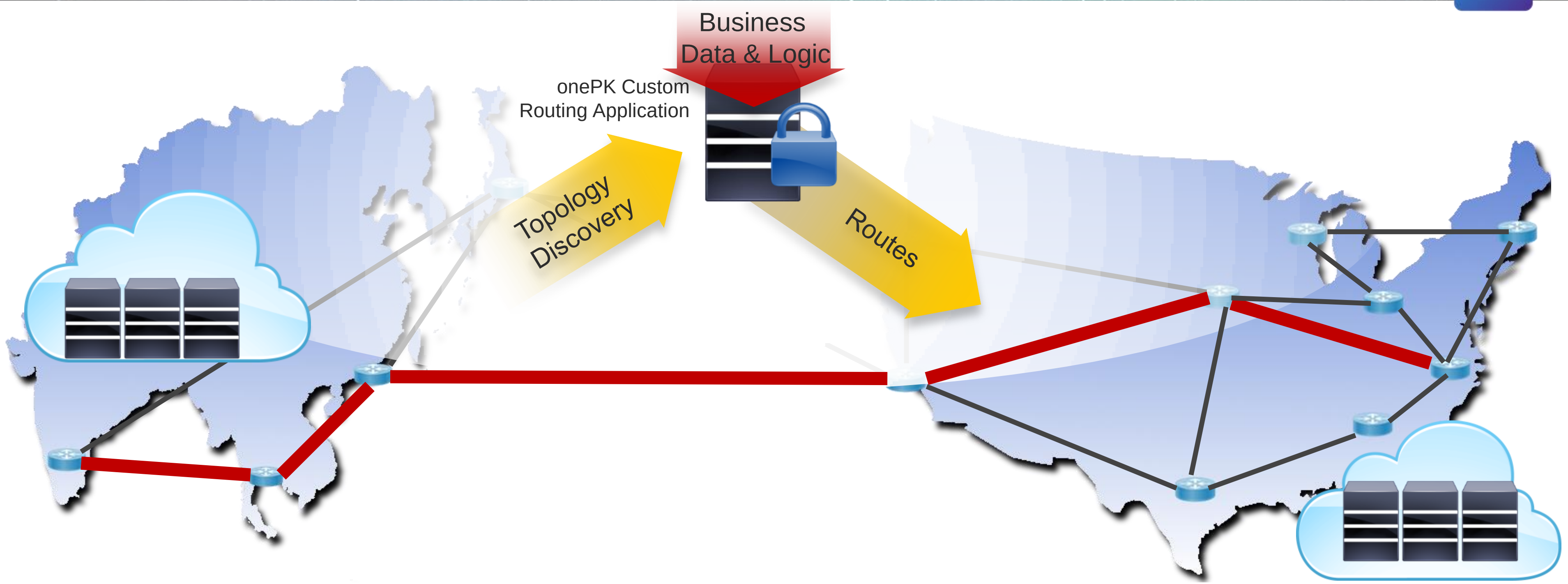
Secure Communications Channel

Custom routing application hosted on a server, communicates securely with onePK infrastructure to route specific packets according to a custom policy.

API presentation layer

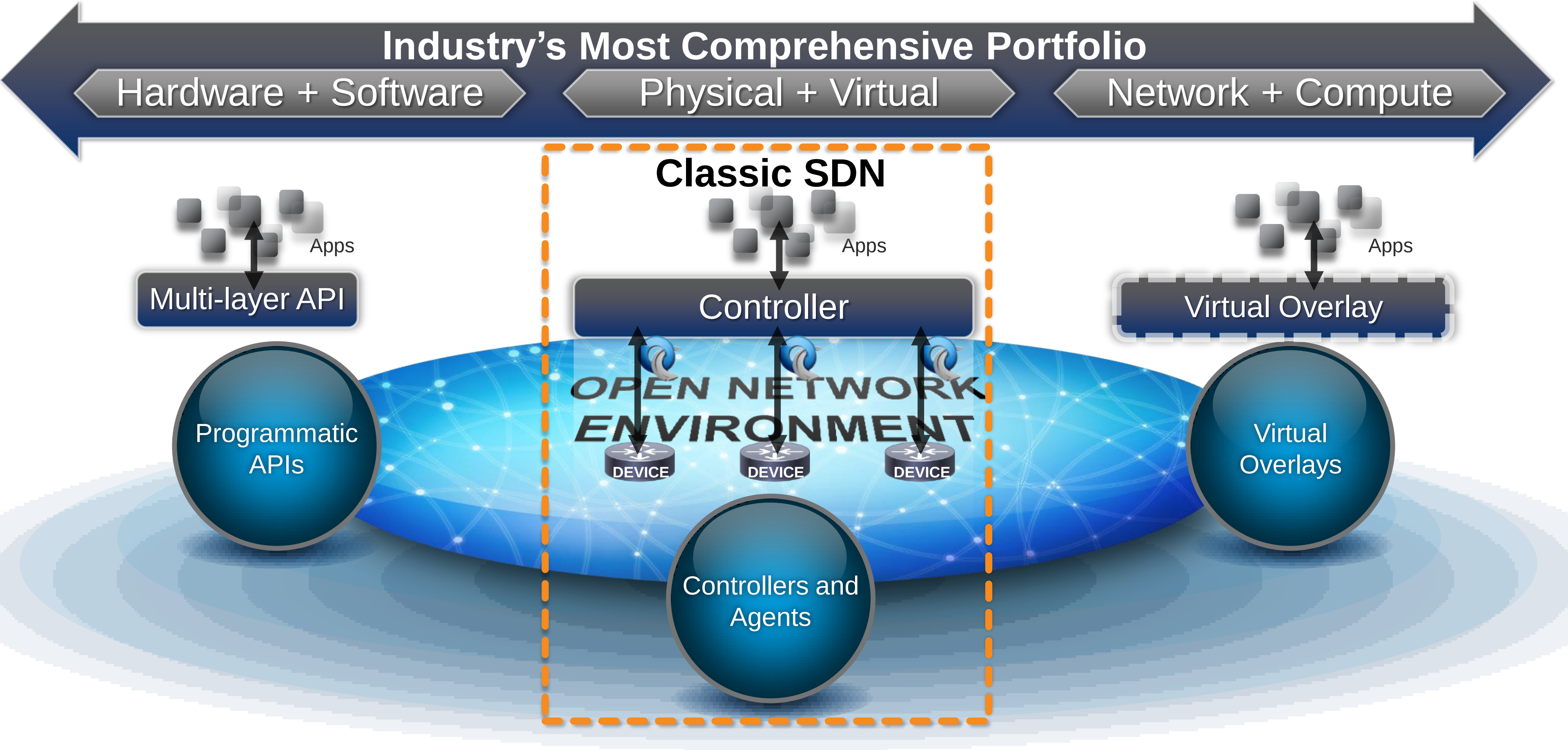
Example: Custom Routing

Data Center Traffic Forwarding Based on a Custom Algorithm



Unique Data Forwarding Algorithm Highly Optimized
for the Network Operator’s Application

Cisco Open Network Environment



Industry's Most Comprehensive Portfolio

Hardware + Software

Physical + Virtual

Network + Compute

Classic SDN

Apps

Multi-layer API

Apps

Controller

Apps

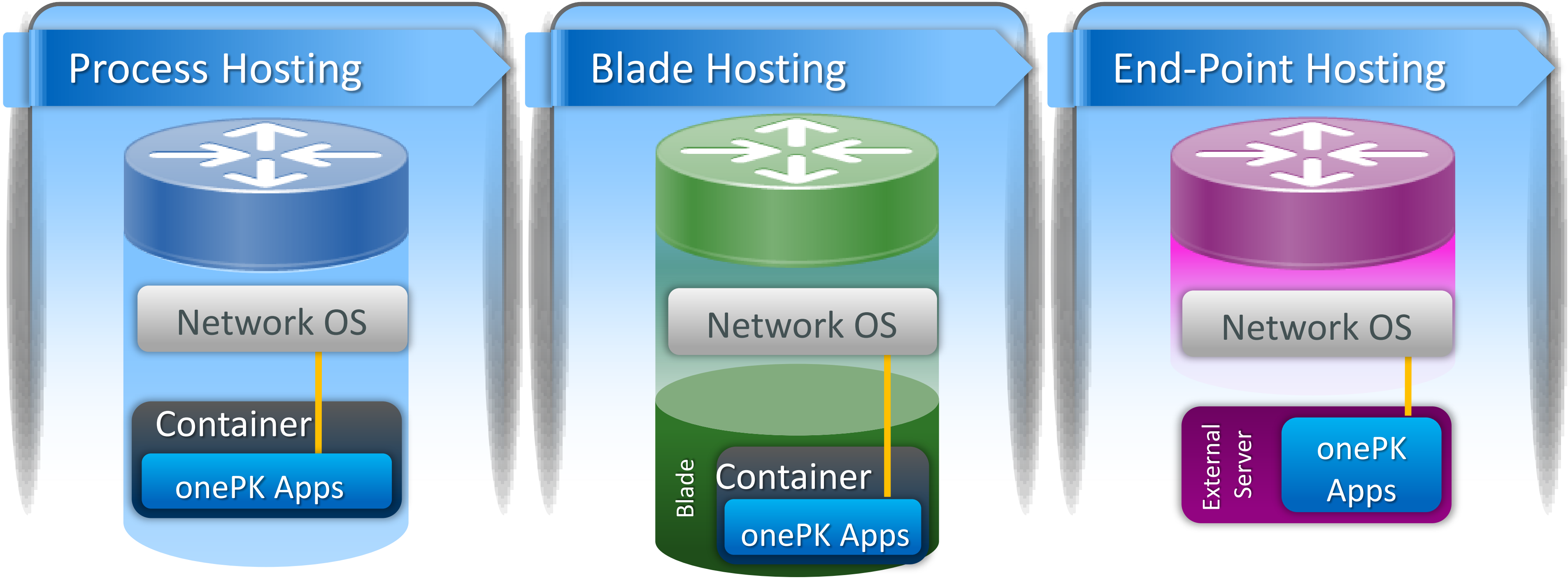
Virtual Overlay

Programmatic APIs

Controllers and Agents

Virtual Overlays

onePK Application Hosting Options

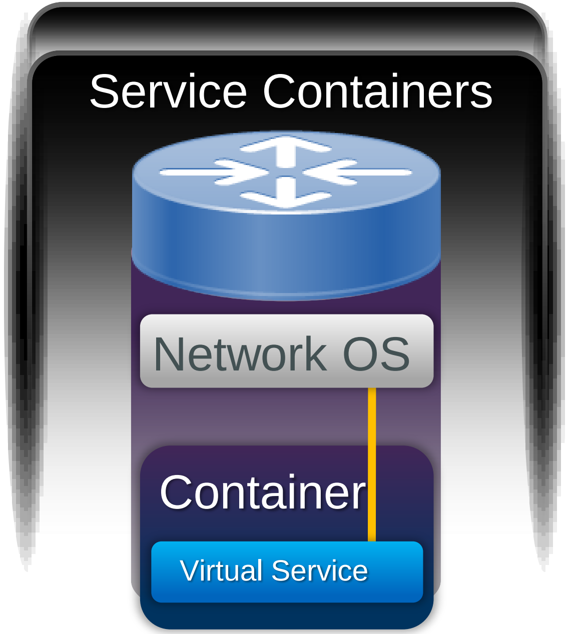


Write Once, Run Anywhere

What is a Cisco Service Container?

Service Containers use virtualization technology to provide a hosting environment on Cisco routers/switches for applications which may be developed and released independent of platform release cycles.

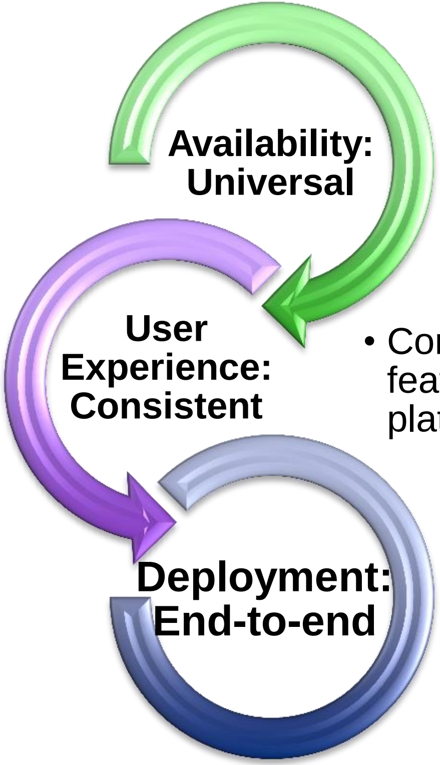
- Virtualized environment on a Cisco device.
- Use Case Cisco Virtual Services:
  - Work/Appliance Consolidation
  - Example: ISR4451X-WAAS
- Use Case Cisco Agents:
  - Integral Router Features with decoupled release cycles
  - Example: RESTFul API
- Use Case Third Party Services (onePK applications):
  - Process Hosted onePK Applications



OpenFlow Agent on Cisco devices

Development Approach

- ✓ Implements the standard OpenFlow switch model
- ✓ Speaks the ‘standard’ OpenFlow protocol
- ✓ Native dedicated CLI for provisioning & troubleshooting
- ✓ Leverages onePK API & unique capabilities of Cisco architecture
- ✓ Supported on all relevant Cisco NOS*s & platforms*

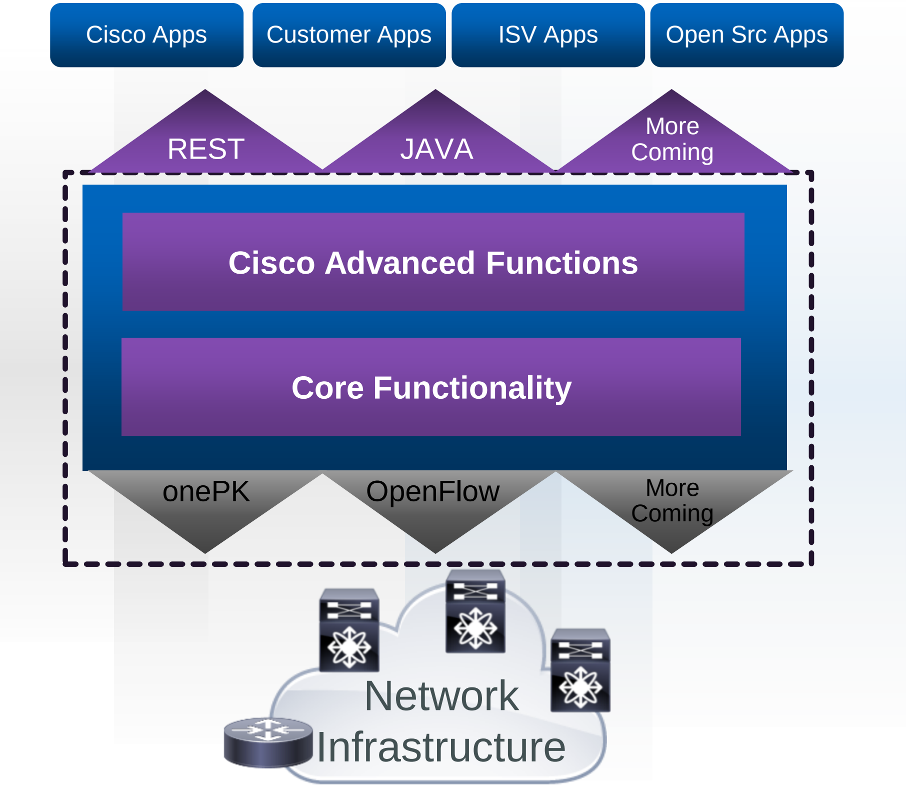


- Available* on NX-OS, IOS-XR, IOS and IOS-XE
- Common code, base features and CLI across platforms
- Targeting multiple network segments such as Data center, SP, Campus and Enterprise, with appropriate features for each

*** Please check roadmap for details on supported platforms & timelines**

Cisco XNC Controller

Industry’s Most Extensible Controller Architecture



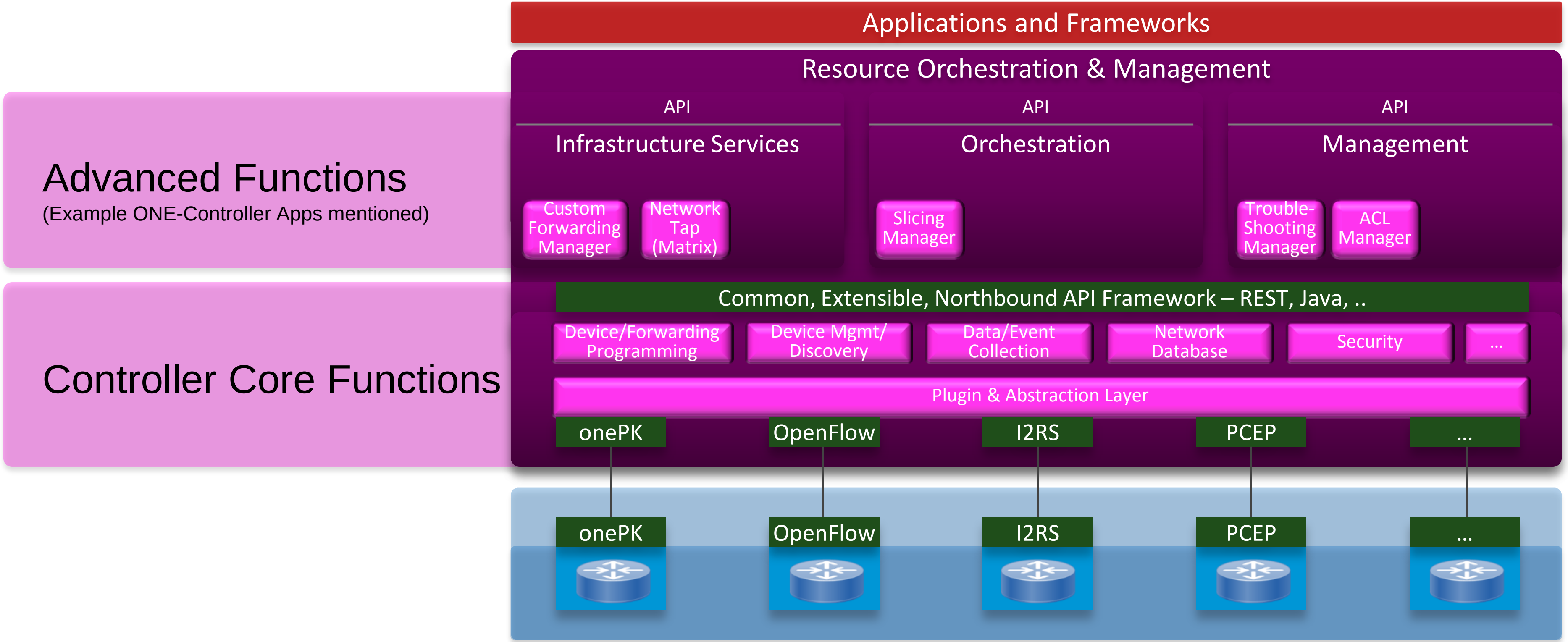
Multiple Published APIs for Popular Languages and Software, e.g., OpenStack

Modular Architecture Allows Rapid Adoption of Evolving Controller Functionality While Minimizing Operational Disruption

Extensible Protocol Support Ensures Continuous Adoption of Emerging Standards

XNC

Architecture Outline



Applications and Frameworks

Resource Orchestration & Management

Advanced Functions
(Example ONE-Controller Apps mentioned)

API
Infrastructure Services
Custom Forwarding Manager
Network Tap (Matrix)

API
Orchestration
Slicing Manager

API
Management
Trouble-Shooting Manager
ACL Manager

Controller Core Functions

Common, Extensible, Northbound API Framework – REST, Java, ..

Device/Forwarding Programming
Device Mgmt/ Discovery
Data/Event Collection
Network Database
Security
...

Plugin & Abstraction Layer

onePK
OpenFlow
I2RS
PCEP
...

onePK
OpenFlow
I2RS
PCEP
...

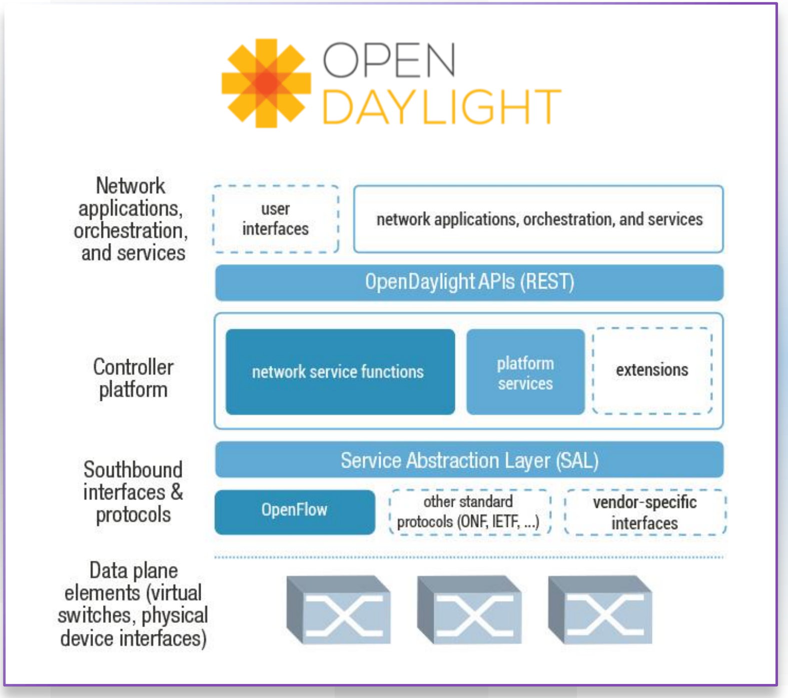
OpenDaylight

An Open Source Project Under the Linux Foundation With the Mutual Goal of Furthering the Adoption and Innovation of Software Defined Networking (SDN) Through the Creation of a Common Market-Supported Framework

Goal Is to Drive Innovation and Accelerate Adoption of SDN

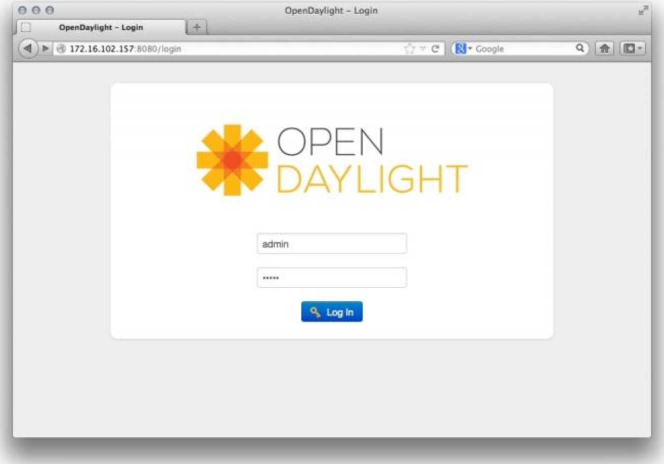
Cisco Is a Founding Platinum Member—Contributed Controller Code and App Framework

Cisco XNC Is Built on Top of OpenDaylight


OPEN DAYLIGHT
Network applications, orchestration, and services
user interfaces
network applications, orchestration, and services
OpenDaylight APIs (REST)
Controller platform
network service functions
platform services
extensions
Service Abstraction Layer (SAL)
Southbound interfaces & protocols
OpenFlow
other standard protocols (ONF, IETF, ...)
vendor-specific interfaces
Data plane elements (virtual switches, physical device interfaces)

Goals and Cisco's Contribution

- Code: To create a robust, extensible, open source code base that covers the major common components required to build an SDN solution.
- Acceptance: To get broad industry acceptance amongst vendors and users.
- Community: To have a thriving and growing technical community contributing to the code base, using the code in commercial products, and adding value above, below and around.
- Current Cisco Contribution
  - Cisco contributes a Controller and Service Abstraction Layer that ensures the modularity and extensibility of the Controller.
  - An OpenFlow 1.0 plugin will be provided on the South bound side, and Northbound API interfaces (OSGi and RESTful) will be provided for application development



OpenDaylight – Login

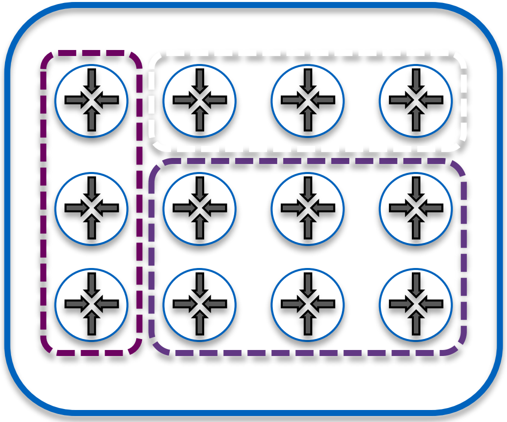
172.16.102.157:8080/login

admin

•••••

XNC Use Cases

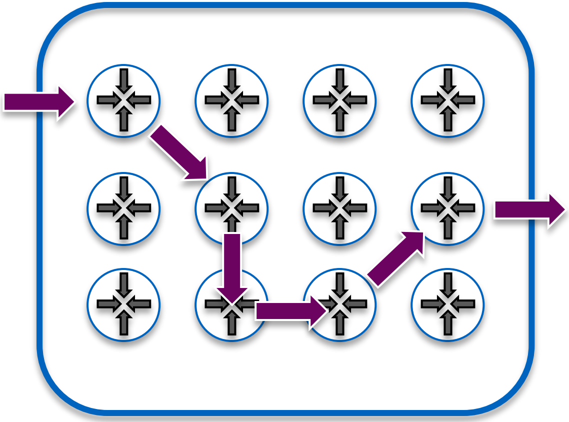
Overview



Network Segmentation

(a.k.a. Campus Slicing)

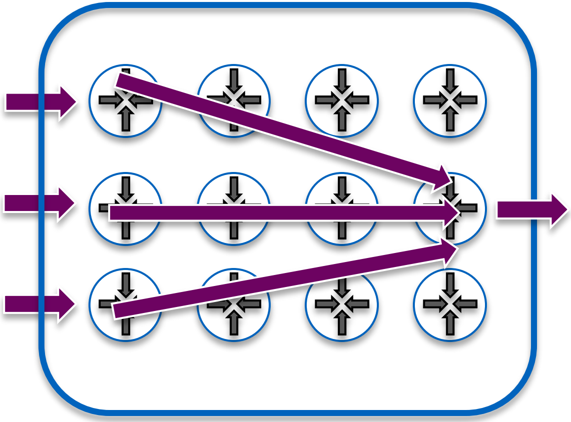
Ability to logically partition the network



Topology Independent Forwarding

(Traffic Steering)

Per Flow Control for how the traffic gets from A to B



Network Tapping

(a.k.a. Matrix use case)

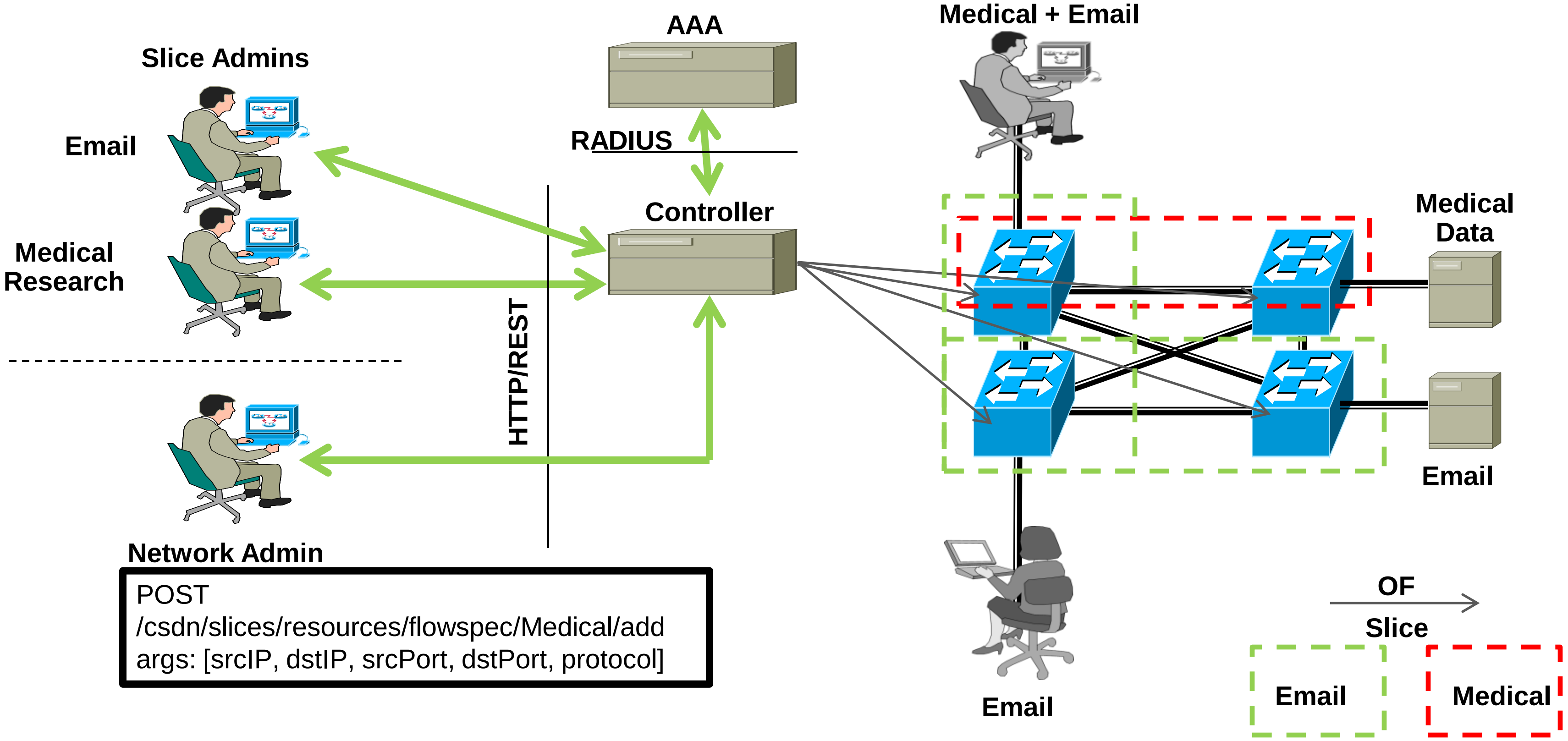
Using off the shelf switches to forward monitor traffic (SPAN, RSPAN, etc) from production network to

XNC Use Cases

Network Segmentation

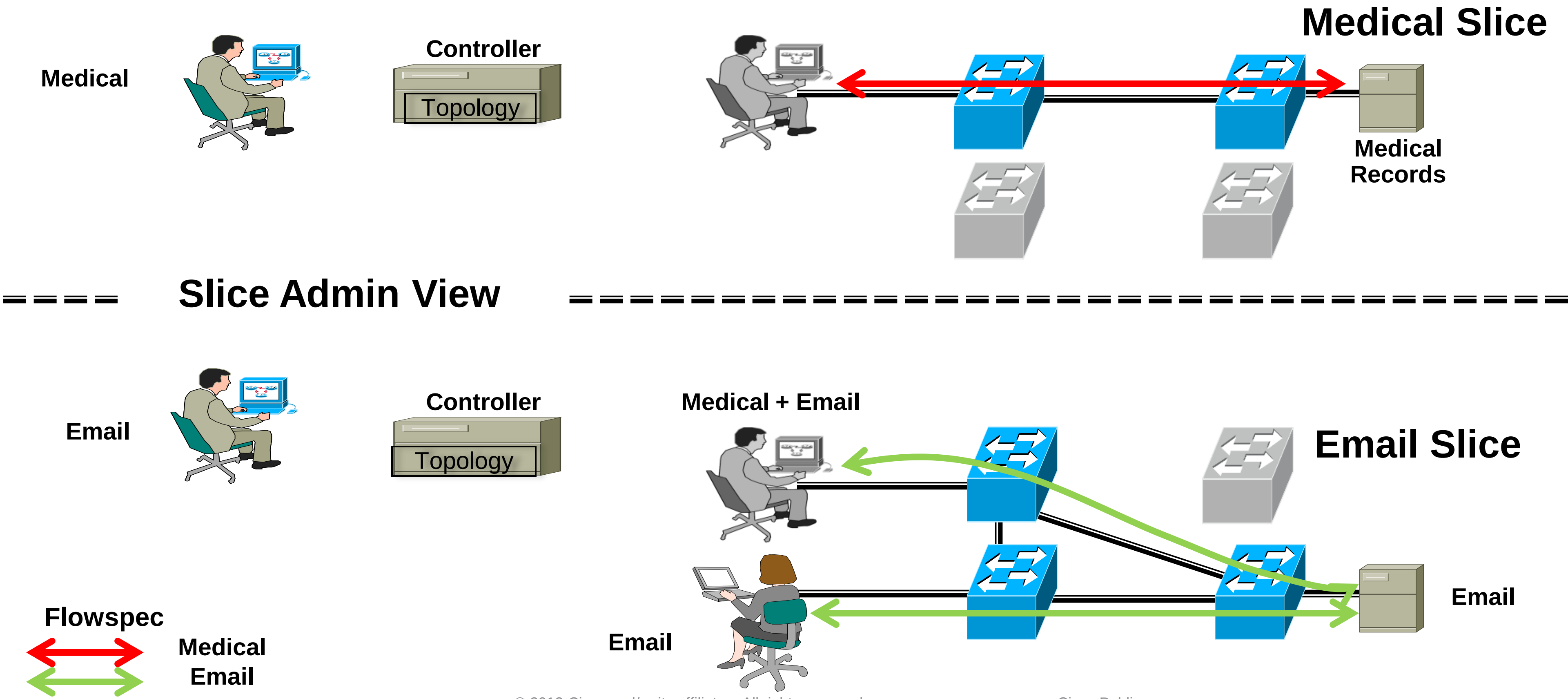
- Allows administrator to “slice” the network into logical partitions based on:
  - Physical devices
  - Interfaces
  - Traffic Characteristics (Protocol, port, etc.)
- Primarily requested by universities and research institutions to partition portions of the network for testing

Network Segmentation



```
POST
/csdn/slices/resources/flowspec/Medical/add
args: [srcIP, dstIP, srcPort, dstPort, protocol]
```

Network Segmentation by Traffic Type



Medical Slice

Medical

Controller

Topology

Medical Records

Slice Admin View

Email

Controller

Topology

Medical + Email

Email Slice

Email

Email

Flowspec

Medical

Email

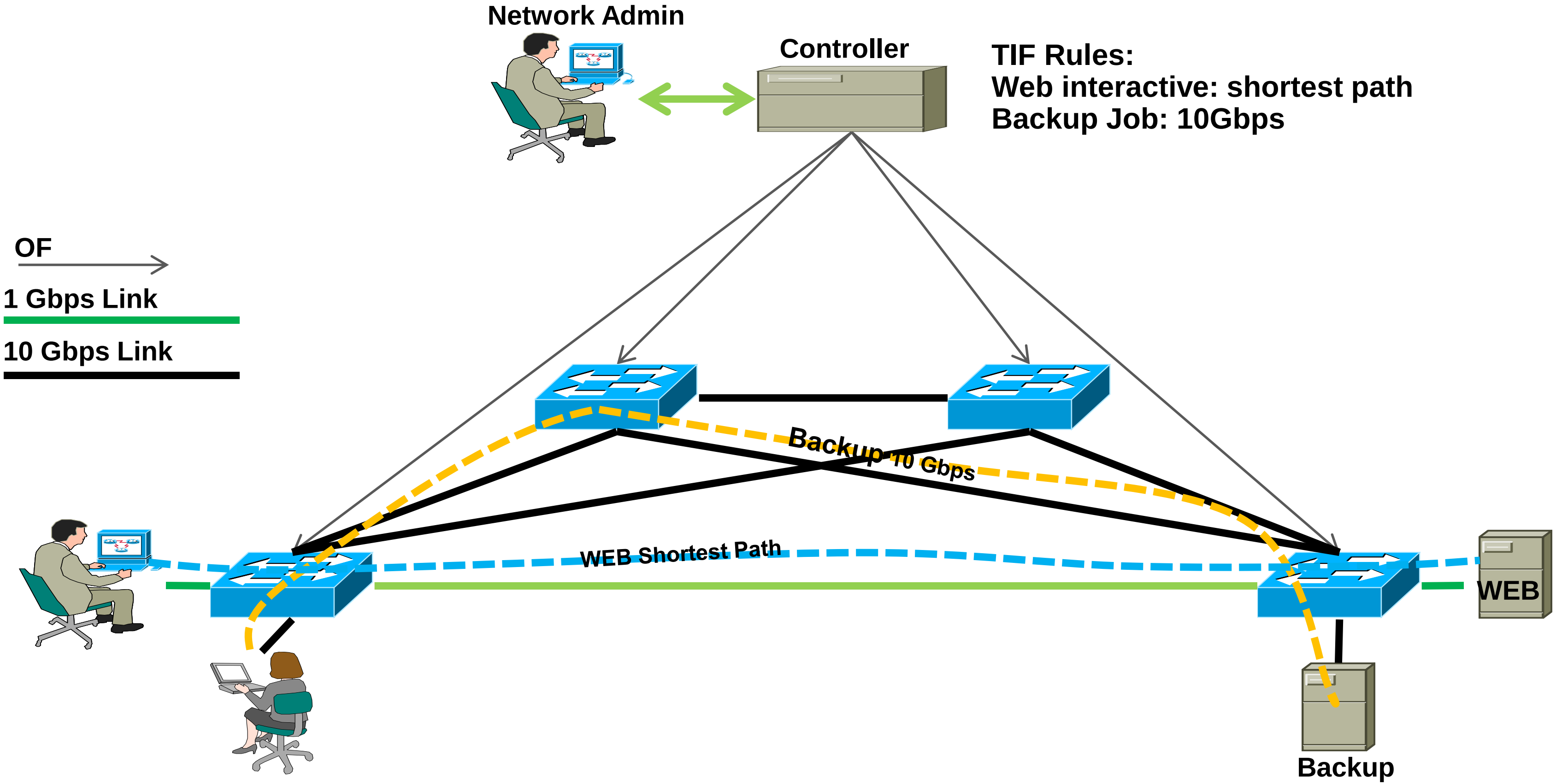
XNC Use Cases

Topology Independent Forwarding (TIF)

- Topology Independent Forwarding (TIF) allows the administrator to configure a path for specifics flows based on:
  - Source/Destination IP Address
  - Protocol
  - Source/Destination Port
- Traffic forwarding is configurable based on a number of factors, including:
  - Link Cost
  - Link Bandwidth
  - String Regular Expression

Topology Independent Forwarding

TIF Rules:
Web interactive: shortest path
Backup Job: 10Gbps

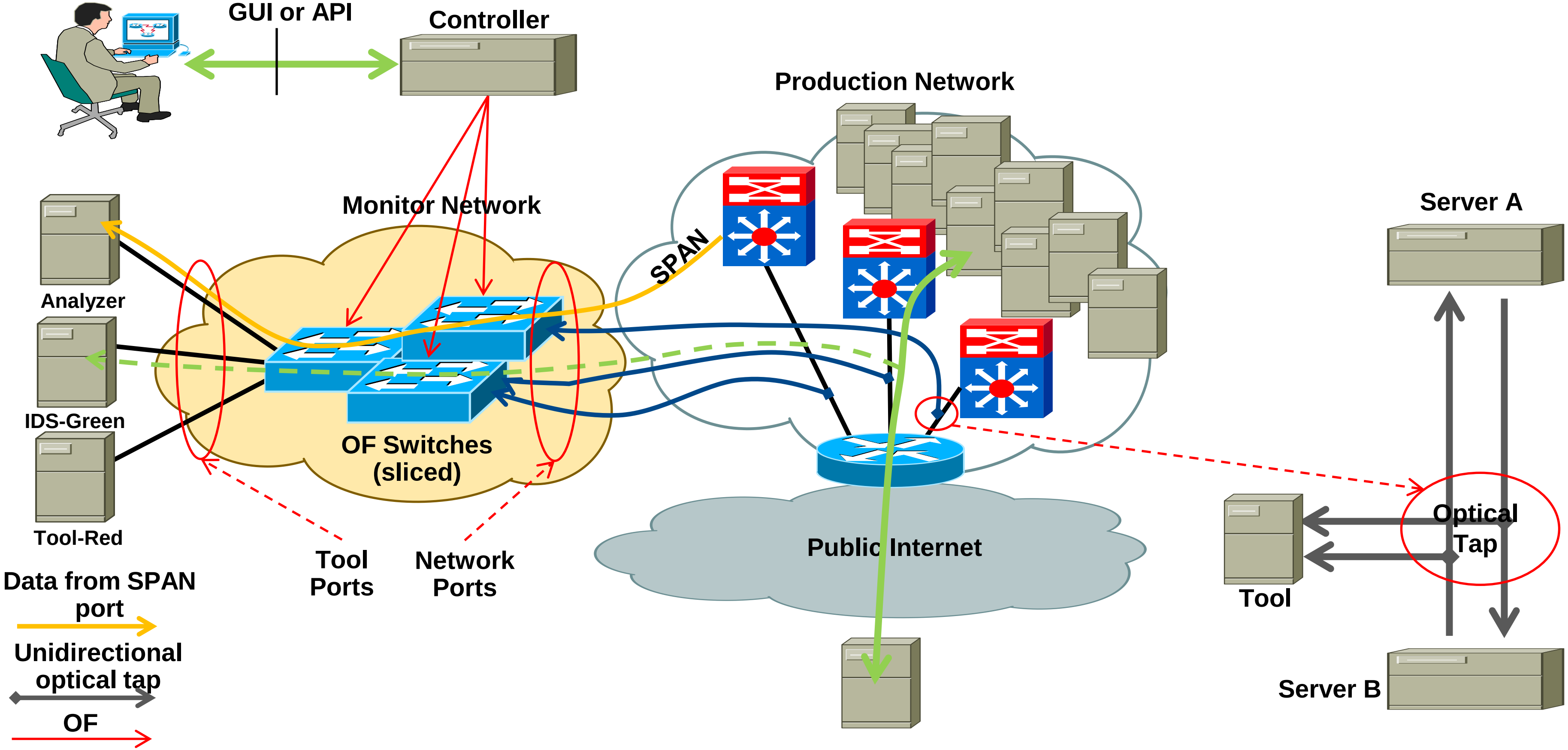


XNC Use Cases

Network Tapping

- Ability to forward traffic from multiple devices to a central tapping point
- Central tapping point can be one or more Nexus 3000 switches
- XNC Monitor Manager application used to:
  - Dynamic Manage Topology
  - Direct Traffic to Monitor Devices
- Solution Advantages:
  - Cost effective alternative to dedicated hardware tapping devices
  - Overcomes concurrent SPAN session limitations
  - Safe way to introduce SDN technology into an environment

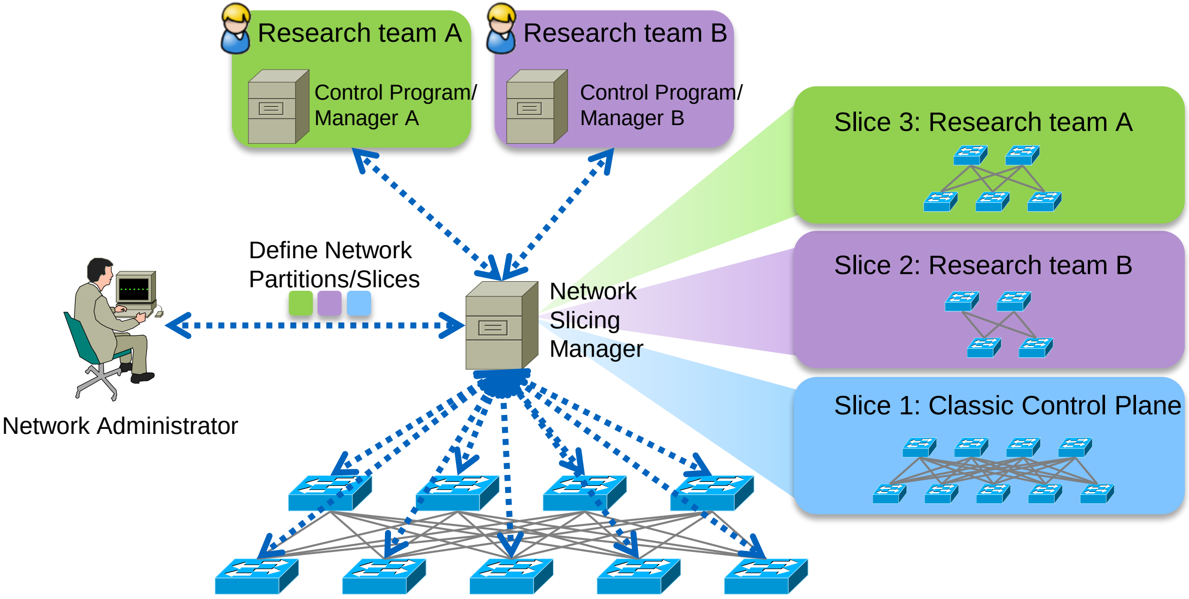
Network Tapping



Orchestration & Virtualization: Network Partitioning

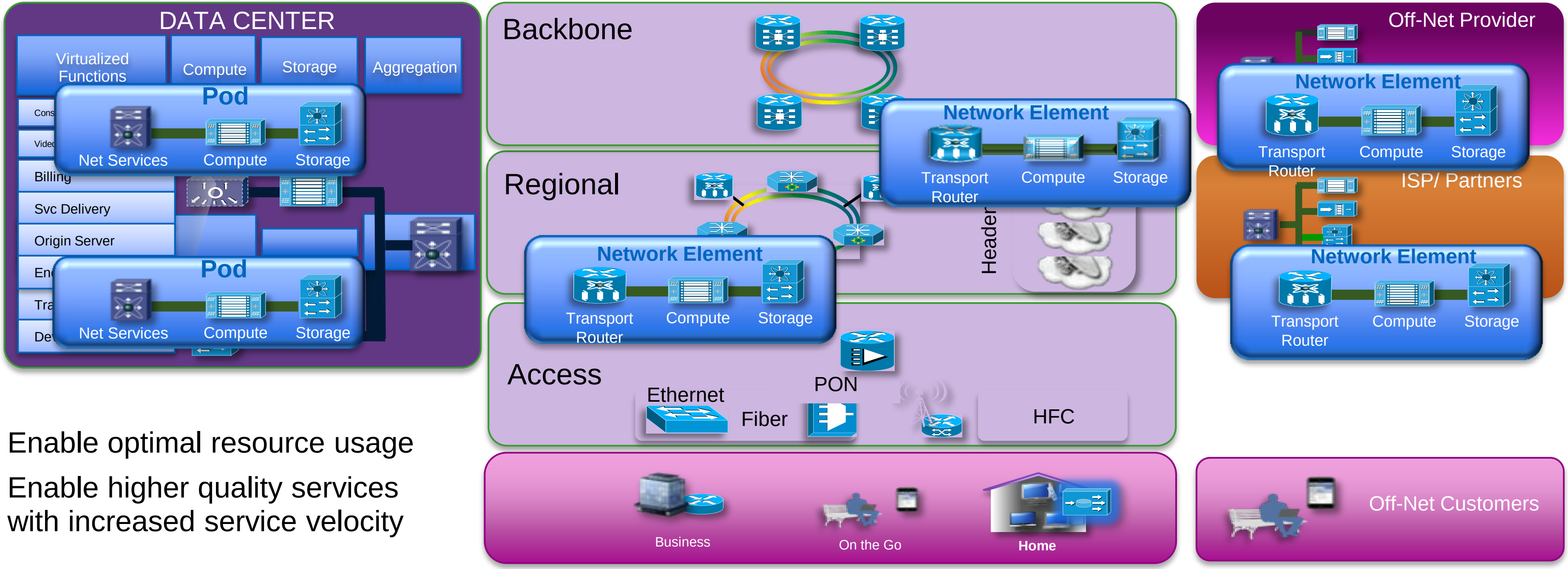
Example: Network Slicing for Research Environments

- Business Problem
  - University desires to “slice” the network into multiple partitions:
    - Production network – classic control plane
    - Several research networks – experimentation with new control algorithms, programs etc.
- Solution
  - Network Slicing Manager partitions the network based on e.g. ports or VLANs
    - Provides northbound interfaces, incl. OpenFlow (Flowvisor-like)
    - Effects of a particular control function of a partition/slice limited to that partition/slice



Orchestration

Content, Applications, Resources Where You Need Them



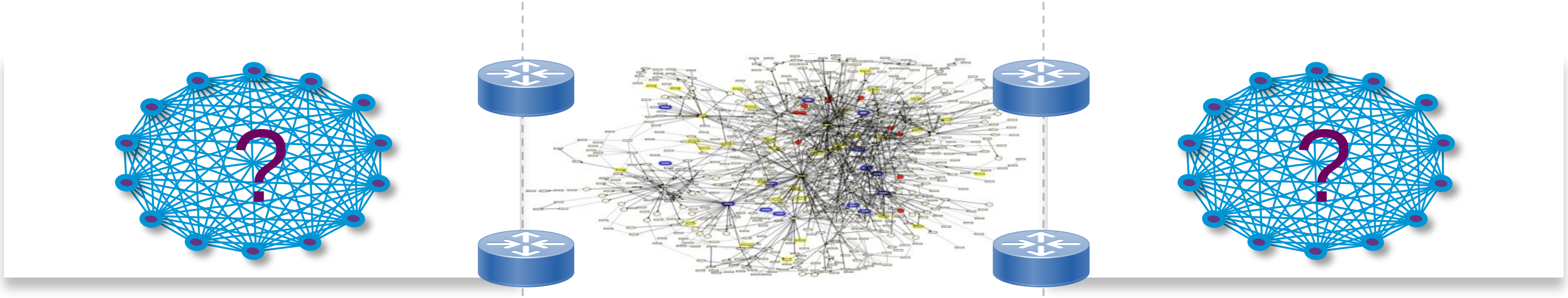
Enable optimal resource usage

Enable higher quality services with increased service velocity

Services hosted in Central Data-Centers and Data-Centers in the PoP

Physical & Virtual – Networks & Services

WAN != LAN – TOR != PE



Un-Constrained Bandwidth
Regular Topology

Constrained Bandwidth
Un-Constrained Topology:
→ Granular Control

Un-Constrained Bandwidth
Regular Topology

Administrative
Boundary:
Policy-Security

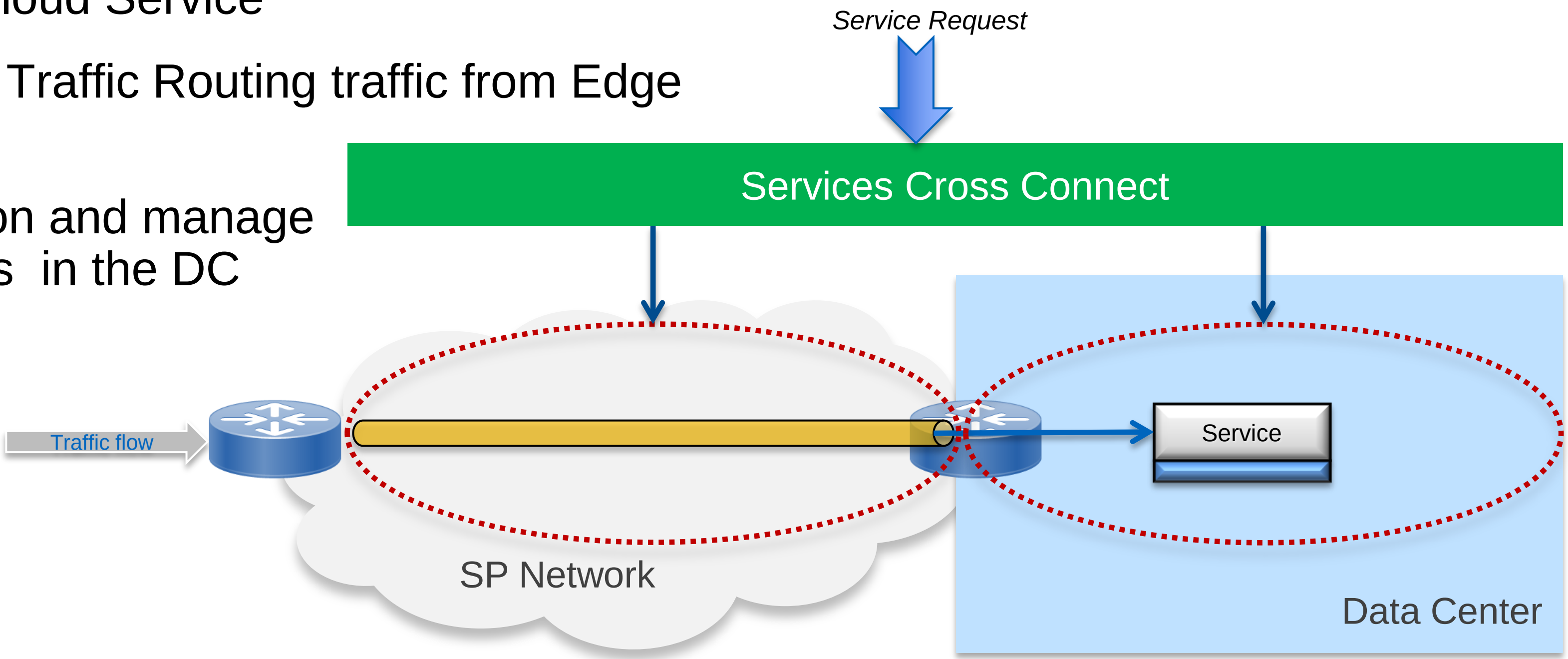
Administrative
Boundary:
Policy-Security

Different solutions for different domains: DC != WAN, TOR != PE

Orchestration

Service Cross-Connect – Network-Ramp to Cloud Services

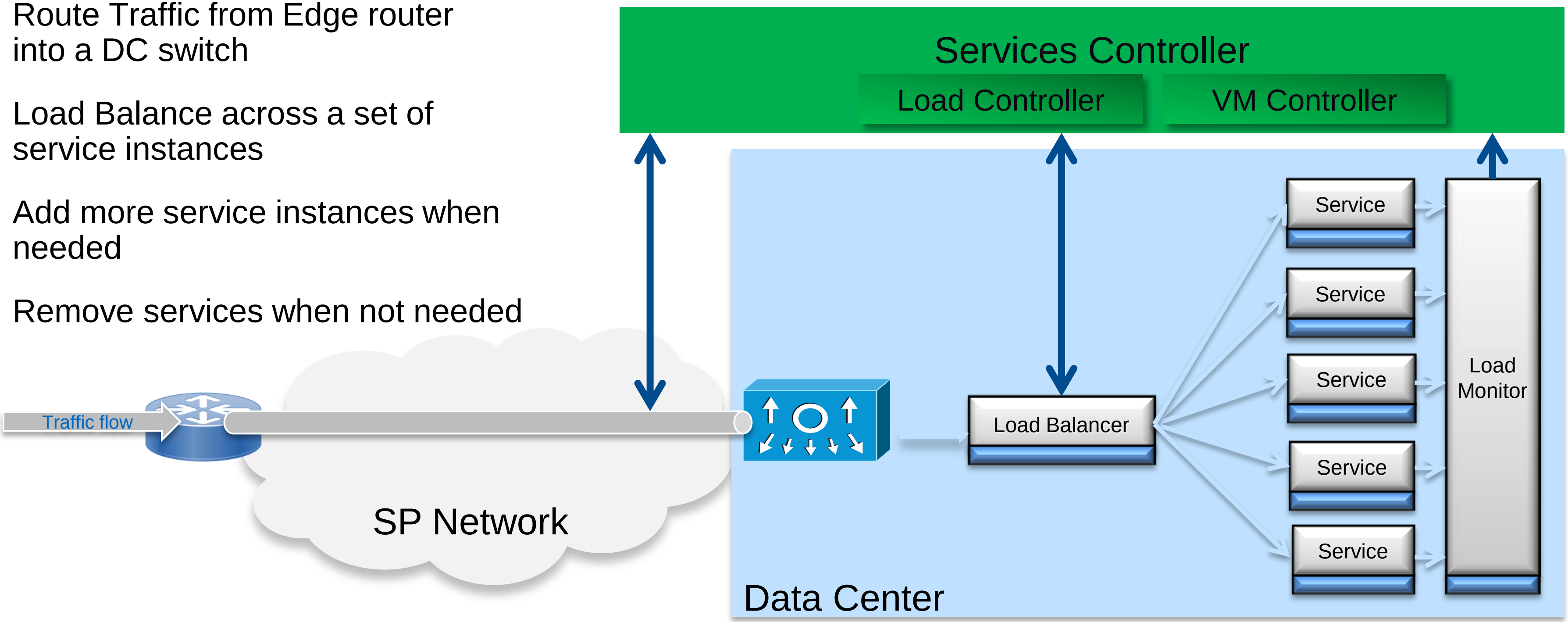
- Take request to provide services to a given Cloud Service
- Control Traffic Routing traffic from Edge to DC
- Provision and manage services in the DC



Orchestration

Elastic DC Services

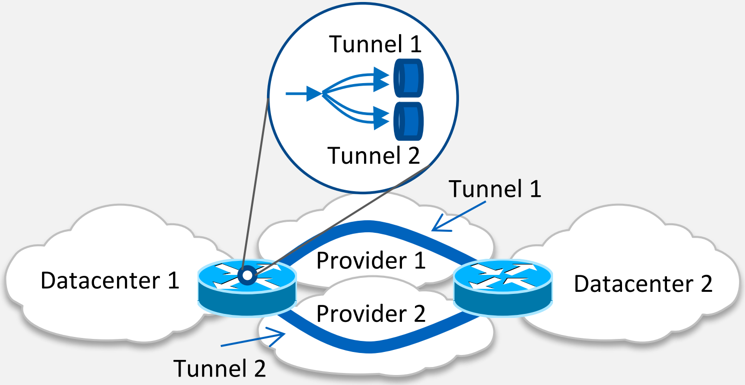
- Route Traffic from Edge router into a DC switch
- Load Balance across a set of service instances
- Add more service instances when needed
- Remove services when not needed

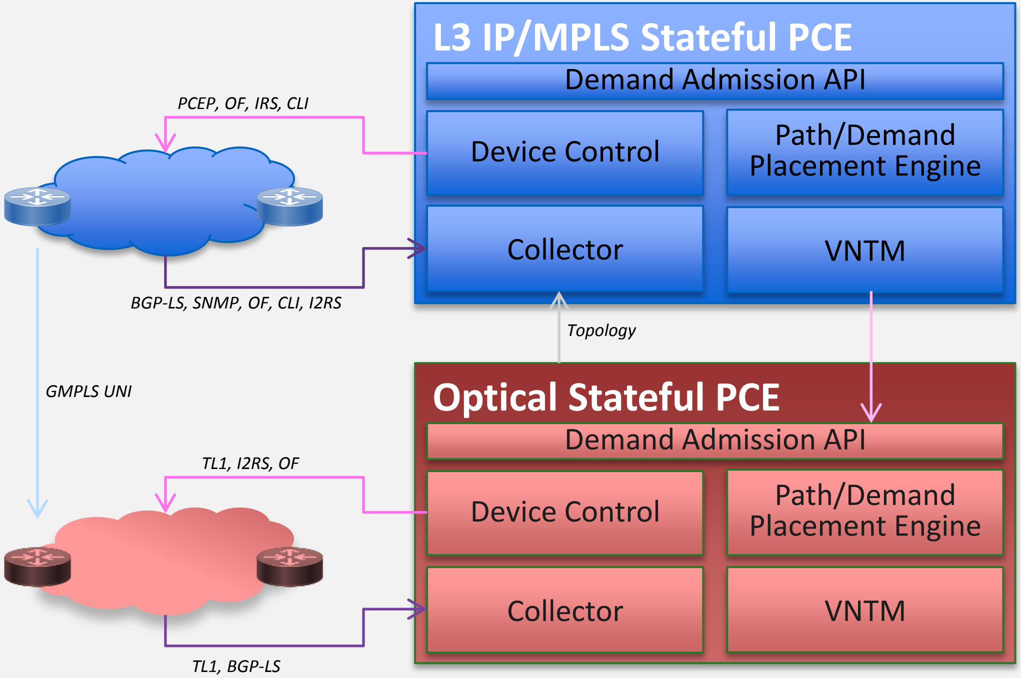


Orchestration & Path Computation

Deployments typically combine Device-APIs, device delivered Network-APIs, and controller delivered Network APIs for a particular solution

Example: Data-Center Interconnect across two providers with granular traffic forwarding control

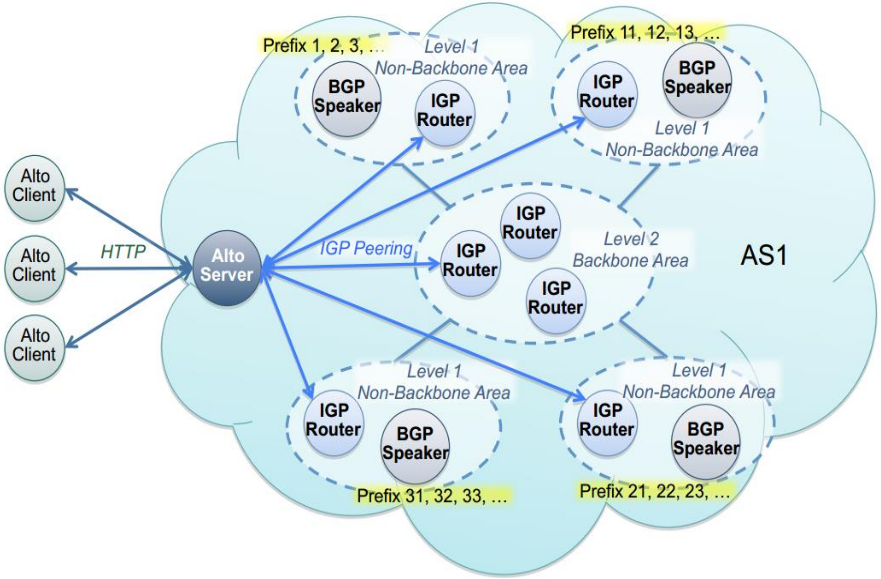




Example: Topology Exposure: Multi-Area IGP

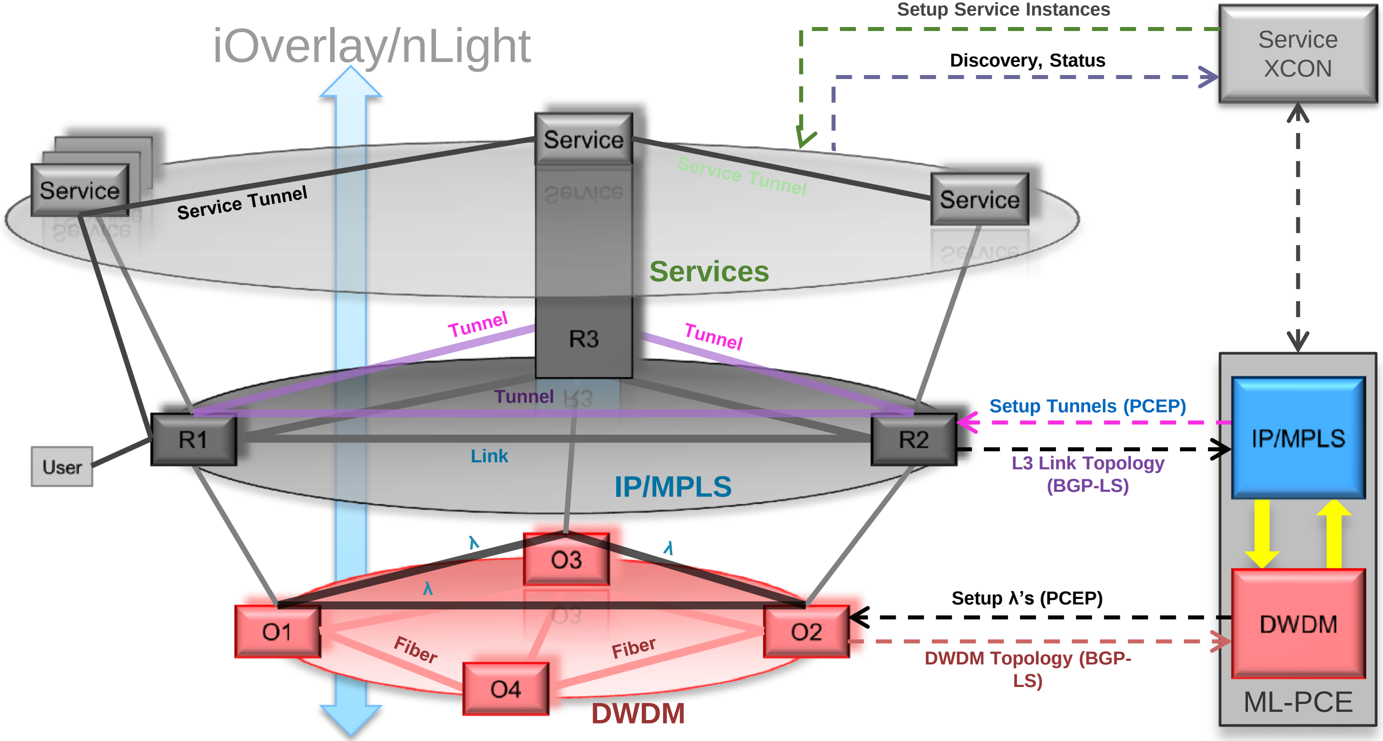
ALTO server exposes multi-area IGP topology

- ALTO server needs to know all areas topology
  - Manually crafting of “IGP peering” topology is tedious and error prone
- Approach:
  - Advertize Link-State Information in BGP
  - draft-gredler-bgp-te

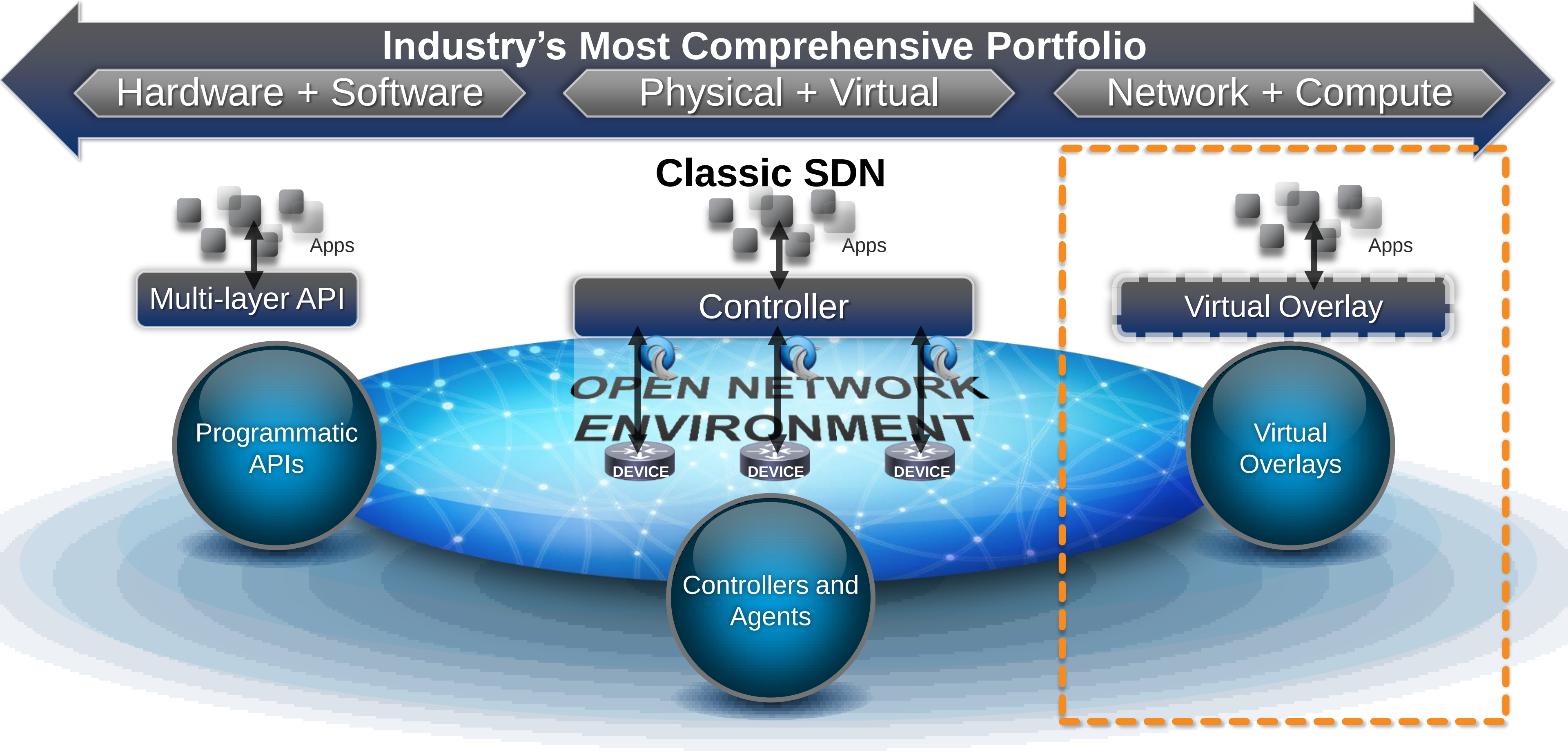


Orchestration

Multi-Layer PCE with iOverlay/nLight



Cisco Open Network Environment



Network Abstractions support Virtualization

Blurring the lines between physical and virtual entities – networks and services

Common Abstractions and common APIs across physical and virtual network elements

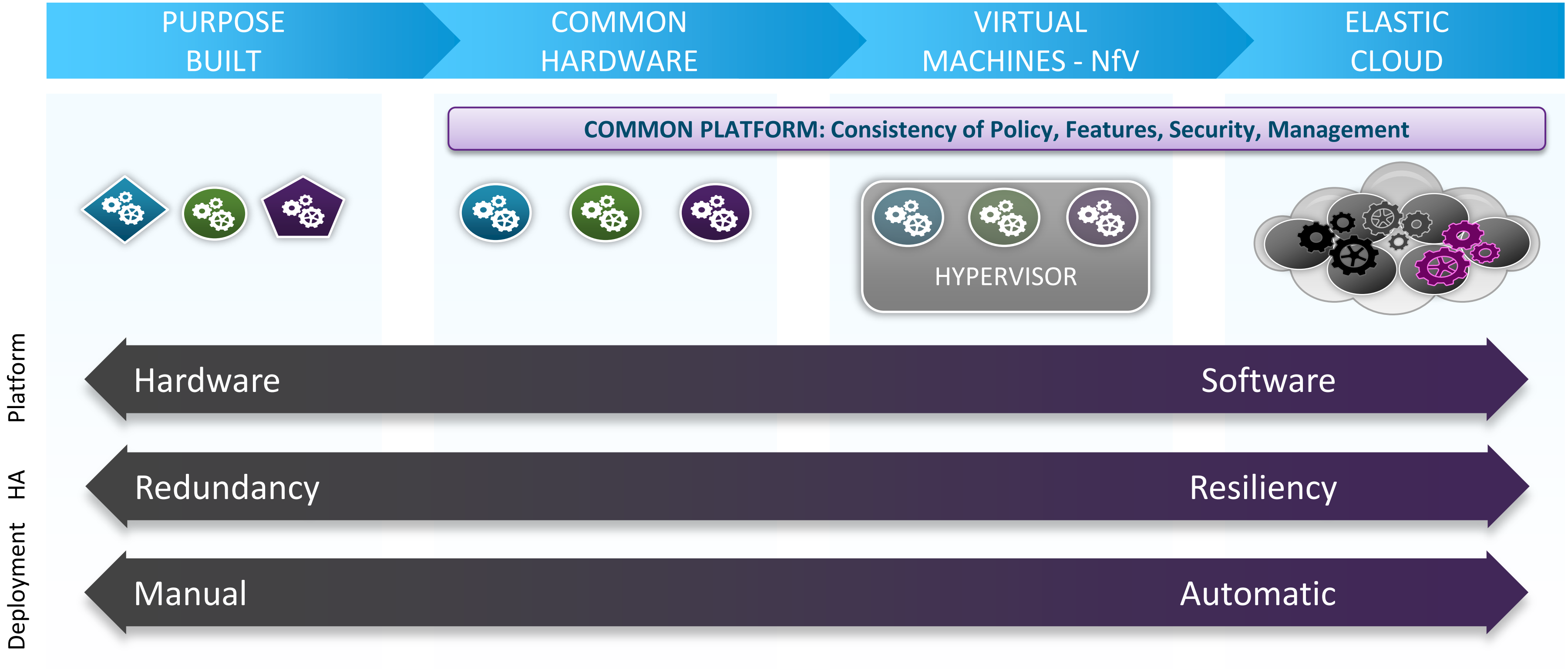
- Virtual Overlay Networks
  - custom endpoint addressing (e.g. for simple endpoint mobility)
  - custom topologies/segmentation
  - custom service chains
    - Example: vPath

Map ‘n Encap approaches to allow for flexible overlays and “identity” and “location” addresses:

- *L2-transport*: FabricPath, 802.1ah
- *IP-transport*: VXLAN, OTV, (L2-)LISP (all use the same frame format)
- *MPLS-transport*: (PBB-)VPLS, (PBB-)EVPN

- Virtual Service Nodes/Appliances/Gateways *Network Function Virtualization (NfV)*
  - VSG, vWAAS, CSR1000v, ASA 1000v, ...

Physical, Virtual, Cloud Evolution



PURPOSE BUILT

COMMON HARDWARE

VIRTUAL MACHINES - NfV

ELASTIC CLOUD

COMMON PLATFORM: Consistency of Policy, Features, Security, Management

Platform: Hardware → Software

HA: Redundancy → Resiliency

Deployment: Manual → Automatic

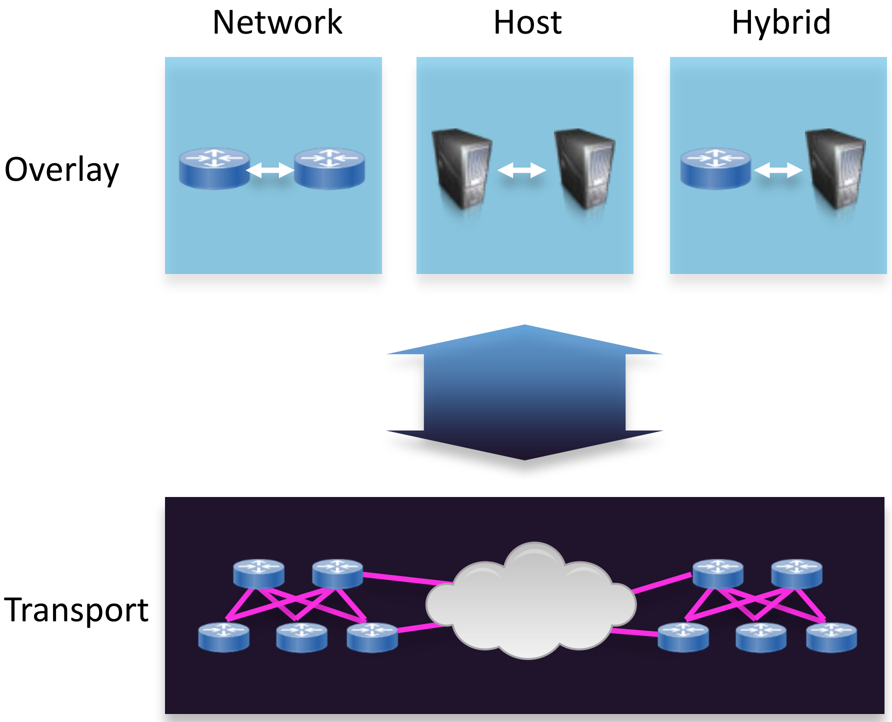
Evolve: Engineering, Operations, Architecture

Physical and Virtualized Network Functions

Examples

Nexus/Catalyst *vSwitch (Nexus 1000v)*	ASR/ISR/CRS *vRouter (CSR1000v)*	Identity/Policy - ISE *vISE*	Firewall - ASA *vFW (ASA 1000v)*
WAAS *vWAAS*	Email Security - ESA *vESA*	Wireless LAN Controller *vWLC*	*Security Gateway - VSG*
Video Cache *vVideoCache*	Web Security - WSA *vWSA*	Network Analysis - NAM *vNAM*	IOS/XR RR *vRouteReflector*

Overlay and Transport Networks



Instance Scale
VM Mobility & LAN Extension
Agile Operations
Hypervisor-agnostic (ESX, HyperV, KVM, Xen,..)
Network / Host / Hybrid
NfV – Service Chains

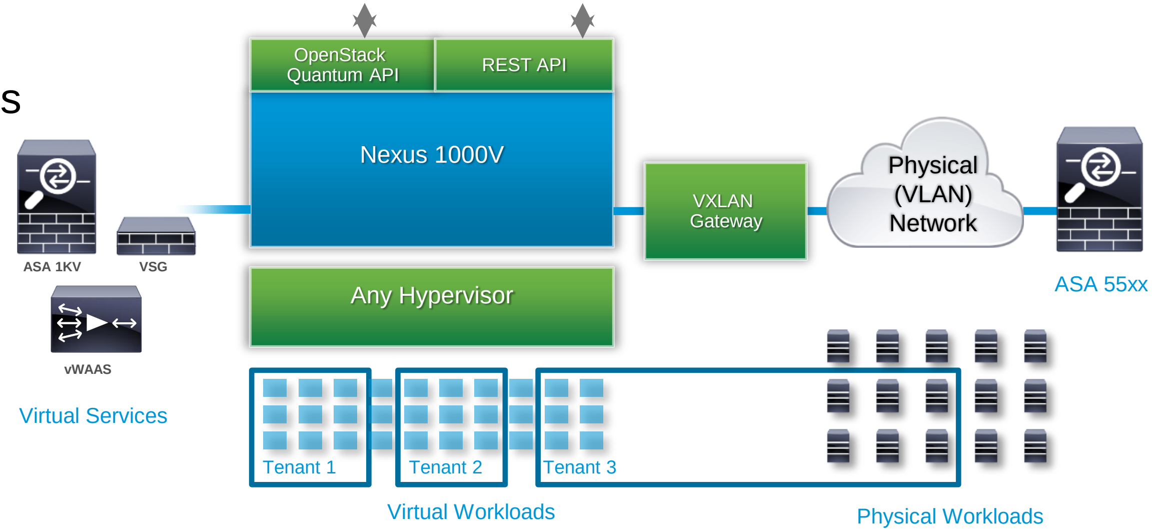
Service Placement / Topology
Multi-Segment Integration (DC-WAN)
OAM – Correlate Overlay and Transport
Traffic Forwarding Control (Flow-Steering, Multicast)

Speeds & Feeds (e.g. low latency forwarding)
Fast Convergence (50ms), Segment Routing
Statistics / Events (e.g. latency measurement)
Buffering / Scheduling / QoS
System resiliency

Virtual Overlay Networks

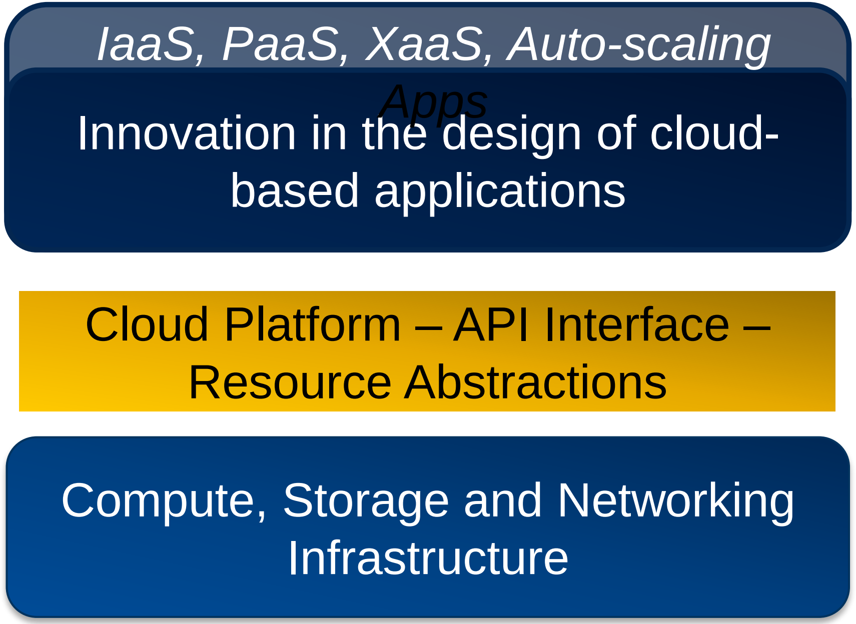
- Example: Virtual Overlay Networks and Services with Nexus 1000V

- Large scale L2 domains: Tens of thousands of virtual ports
- Common APIs
  - Incl. OpenStack Quantum API's for orchestration
- Scalable DC segmentation and addressing
  - VXLAN
- Virtual service appliances and service chaining/traffic steering
  - VSG (cloud-ready security), vWAAS (application acceleration), vPATH
- Multi-hypervisor platform support: ESX, Hyper-V, OpenSource Hypervisors
- Physical and Virtual: VXLAN to VLAN Gateway



Network Service becomes a first class citizen in cloud computing and automation

- Enable full automation of Infrastructure Provisioning and Control – including the Network
  - Cloud Automation: Automation of Compute, Network, Storage resources
- Apply to automate all types of networks: physical devices, virtual devices, overlay/non-overlay networks
  - Orthogonal to whether SDN concepts are leveraged



IaaS, PaaS, XaaS, Auto-scaling

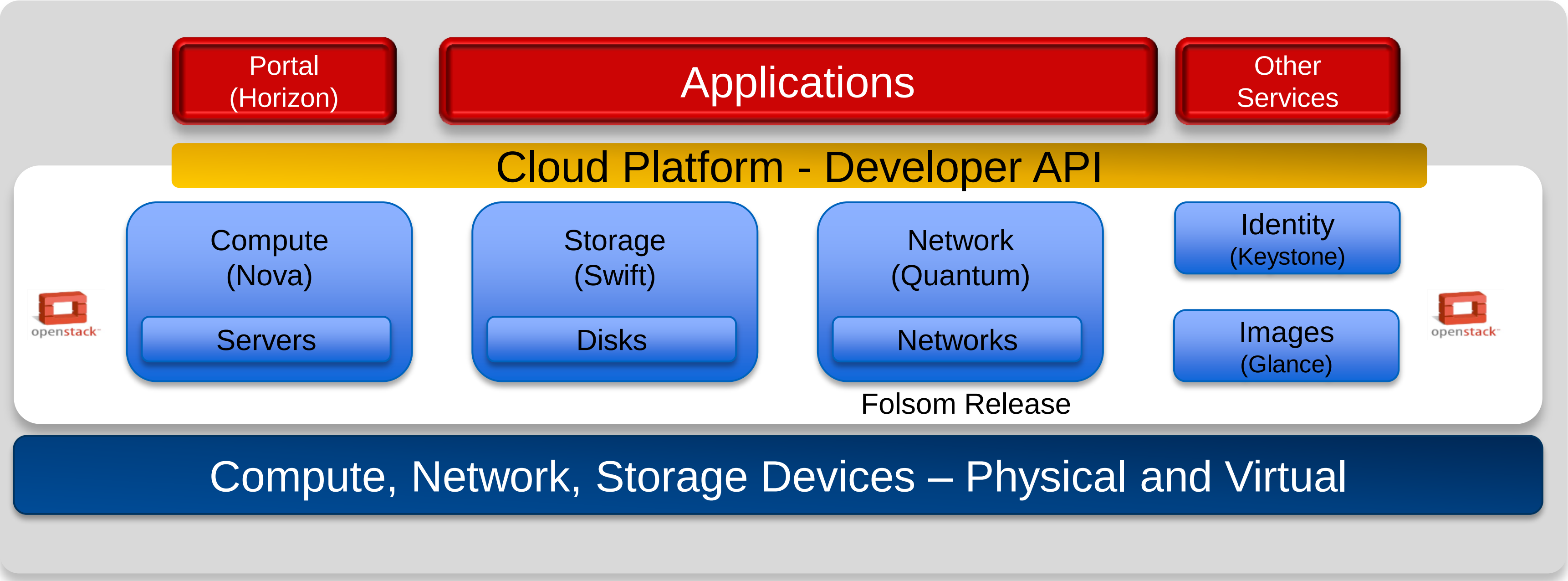
Apps

Innovation in the design of cloud-based applications

Cloud Platform – API Interface – Resource Abstractions

Compute, Storage and Networking Infrastructure

Network Service becomes a first class citizen



Portal (Horizon)

Applications

Other Services

Cloud Platform - Developer API

Compute (Nova)

Servers

Storage (Swift)

Disks

Network (Quantum)

Networks

Identity (Keystone)

Images (Glance)

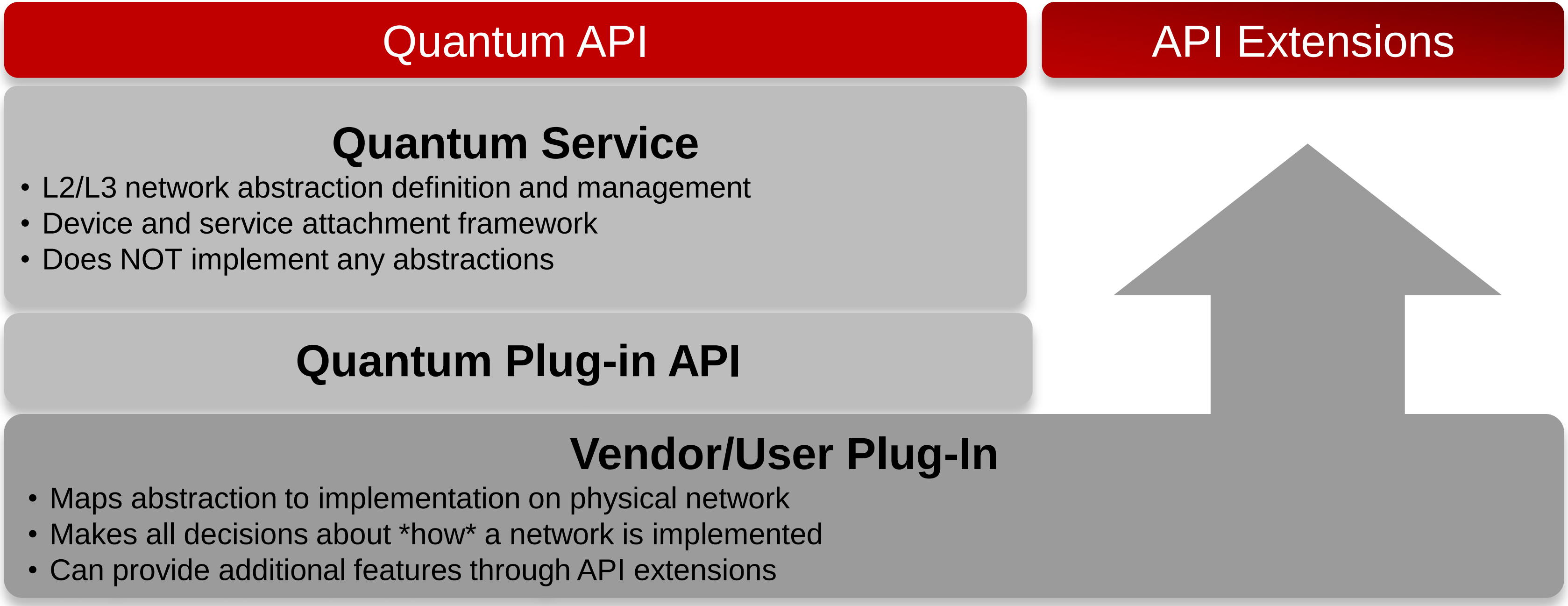
Folsom Release

Compute, Network, Storage Devices – Physical and Virtual

Openstack is for infrastructure automation – orthogonal to whether SDN concepts are applied

Quantum Architecture

Extensible allowing vendor specific capabilities



Quantum API

API Extensions

Quantum Service

- L2/L3 network abstraction definition and management
- Device and service attachment framework
- Does NOT implement any abstractions

Quantum Plug-in API

Vendor/User Plug-In

- Maps abstraction to implementation on physical network
- Makes all decisions about *how* a network is implemented
- Can provide additional features through API extensions

Nexus – Initial Support of OpenStack Quantum

- Nexus 1000
  - Based on Grizzly release
  - Red Hat and Ubuntu - KVM
  - 512 servers per VSM and scaling to future with federations
  - VLAN - 4096, VXLAN – 16000 segments, 32000 ports, 300+ veths/vem
  - Enhanced VXLAN – No multicast requirement in a VSM and in future across VSMs
  - VSM on any hypervisor or Nexus1010
  - CSR as the tenant router – integrated into OpenStack (VXLAN aware)
  - NAT is supported/overlapping IP support
- Nexus 3000 and Higher
  - http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps11541/data_sheet_c78-727737.html

Top 5 Takeaways: Cisco Open Network Environment

1 Cisco Open Network Environment > SDN > OpenFlow

2 Industry broadest approach to network programmability

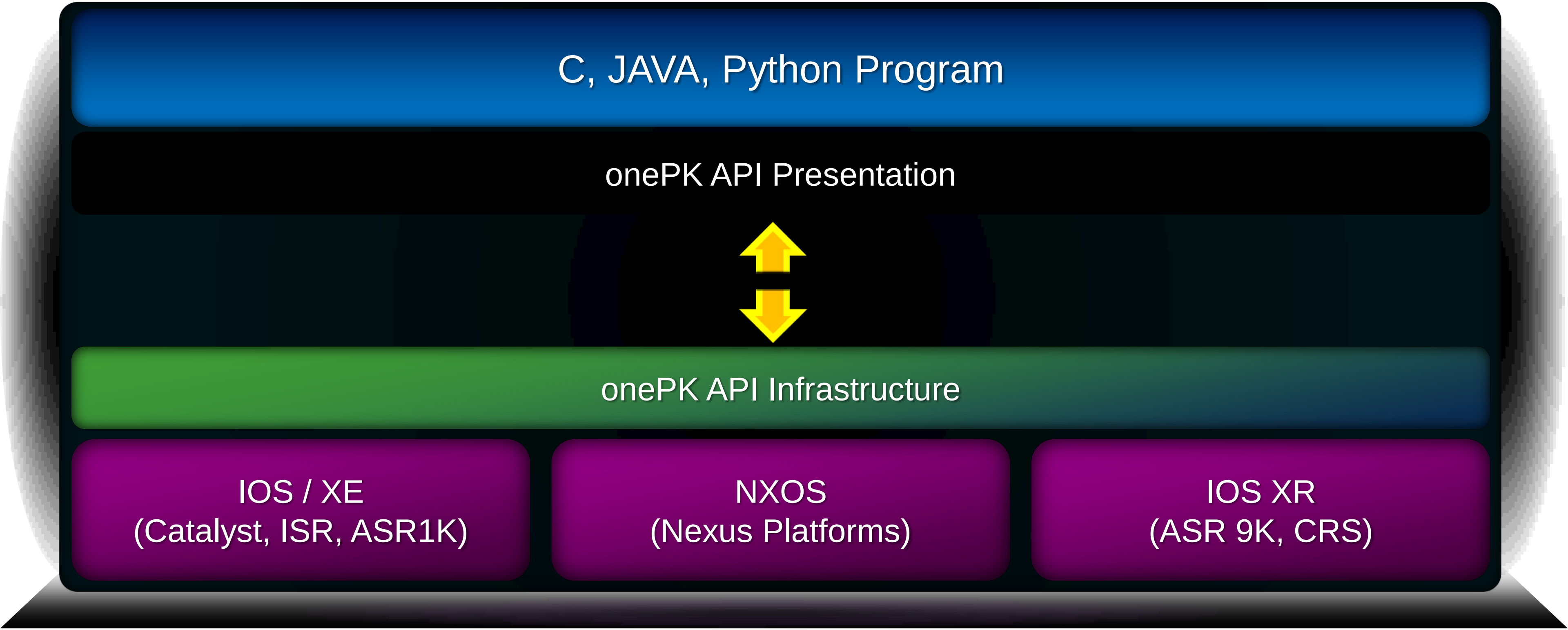
3 Open Standards: Consistency across physical and virtual environments

4 Multi-hypervisor, multi-protocol, multi-layer

5 Applicable to Enterprise, Service Provider and Cloud Environments

SDN與實際應用的結合

onePK Architecture



C, JAVA, Python Program

onePK API Presentation

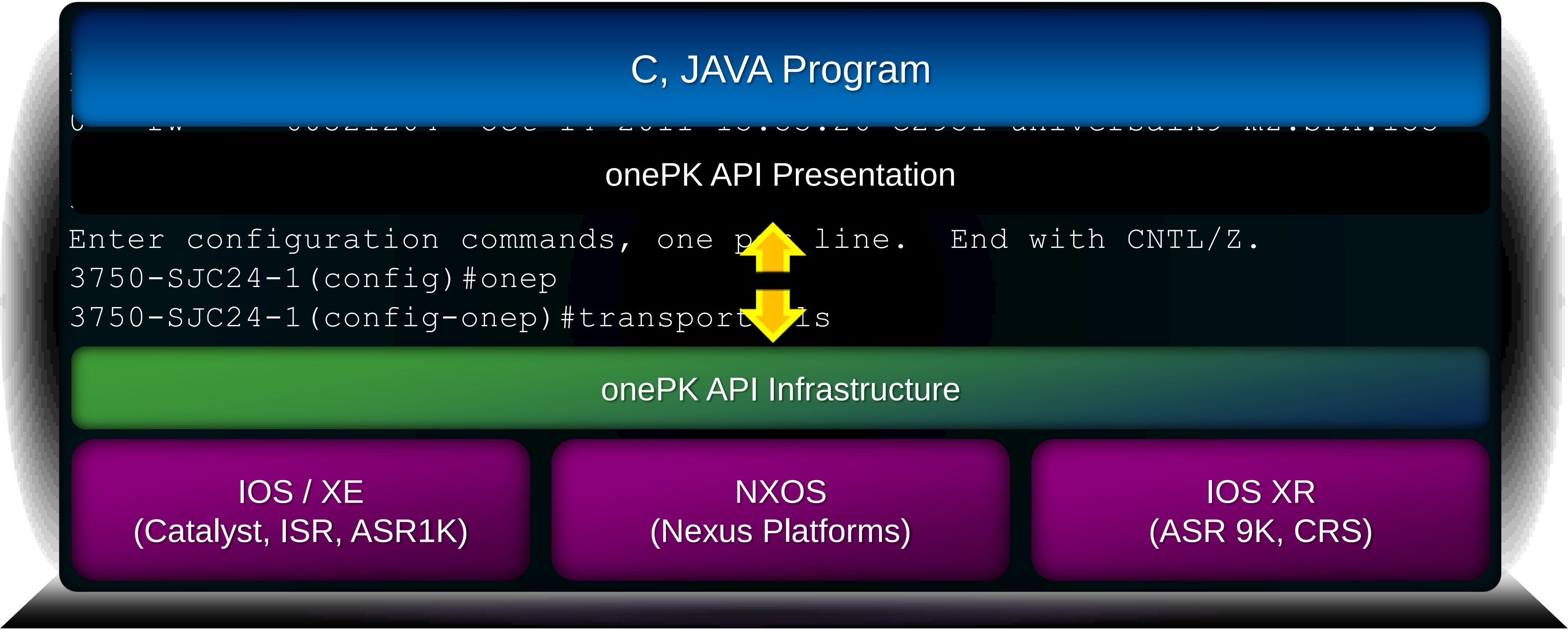
onePK API Infrastructure

IOS / XE
(Catalyst, ISR, ASR1K)

NXOS
(Nexus Platforms)

IOS XR
(ASR 9K, CRS)

What Does the API Infrastructure “Look Like” ?



C, JAVA Program

onePK API Presentation

```
Enter configuration commands, one per line.  End with CNTL/Z.
3750-SJC24-1(config)#onep
3750-SJC24-1(config-onep)#transport tls
```

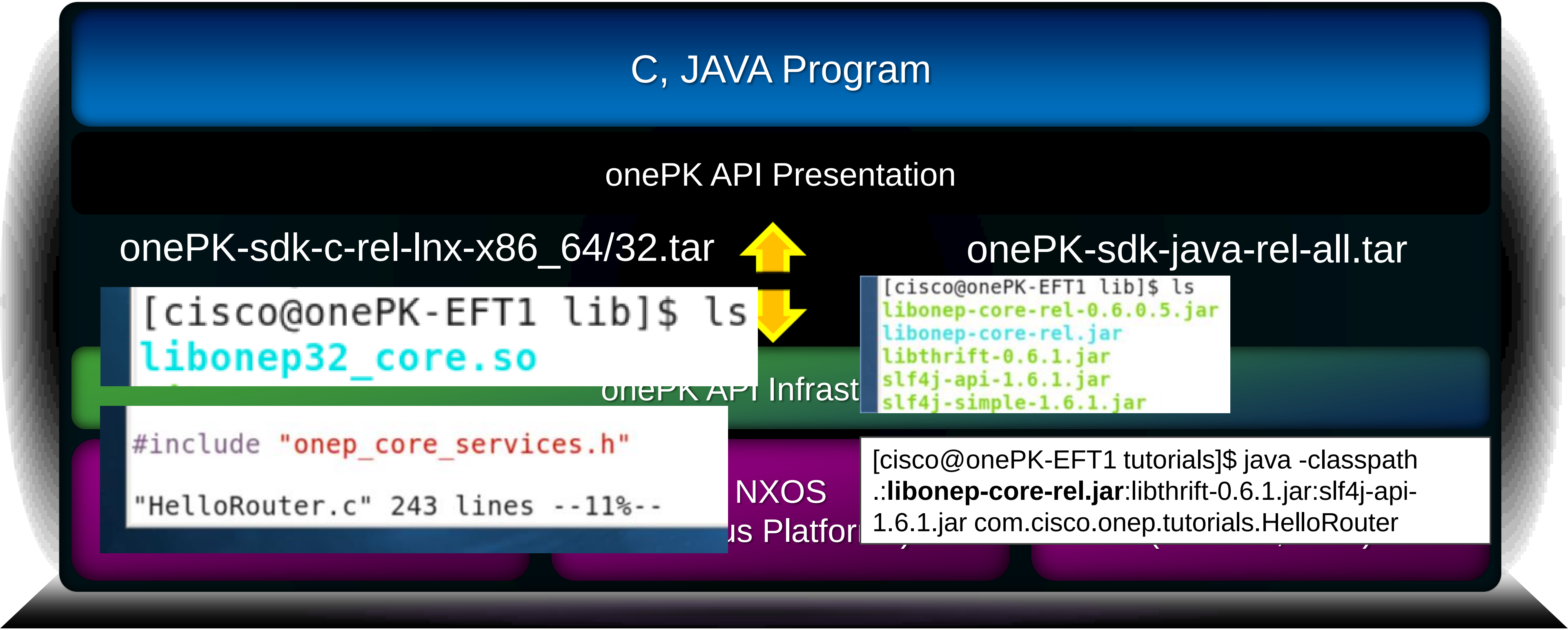
onePK API Infrastructure

IOS / XE
(Catalyst, ISR, ASR1K)

NXOS
(Nexus Platforms)

IOS XR
(ASR 9K, CRS)

What Does the API Presentation Layer “Look Like”?



C, JAVA Program

onePK API Presentation

onePK-sdk-c-rel-lnx-x86_64/32.tar

```
[cisco@onePK-EFT1 lib]$ ls
libonep32_core.so
```

```
#include "onep_core_services.h"

"HelloRouter.c" 243 lines --11%--
```

onePK-sdk-java-rel-all.tar

```
[cisco@onePK-EFT1 lib]$ ls
libonep-core-rel-0.6.0.5.jar
libonep-core-rel.jar
libthrift-0.6.1.jar
slf4j-api-1.6.1.jar
slf4j-simple-1.6.1.jar
```

```
[cisco@onePK-EFT1 tutorials]$ java -classpath
.:libonep-core-rel.jar:libthrift-0.6.1.jar:slf4j-api-
1.6.1.jar com.cisco.onep.tutorials.HelloRouter
```

onePK API Infrast

NXOS

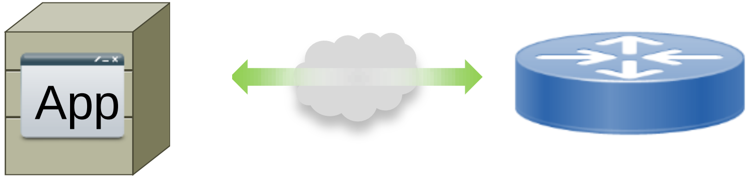
us Platfor

onePK APIs are Grouped in Service Sets

Base Service Set	Description
Data Path	Provides packet delivery service to application: Copy, Punt, Inject
Policy	Provides filtering (NBAR, ACL), classification (Class-maps, Policy-maps), actions (Marking, Policing, Queuing, Copy, Punt) and applying policies to interfaces on network elements
Routing	Read RIB routes, add/remove routes, receive RIB notifications
Element	Get element properties, CPU/memory statistics, network interfaces, element and interface events
Discovery	L2 topology and local service discovery
Utility	Syslog events notification, Path tracing capabilities (ingress/egress and interface stats, next-hop info, etc.)
Developer	Debug capability, CLI extension which allows application to extend/integrate application’s CLIs with network element

Where Do onePK Applications Run?

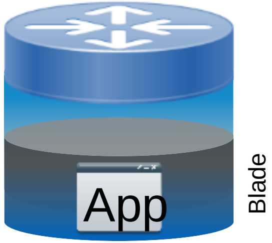
Choose the Hosting Model that Suits Your Platform and Your Application



On An External Server

- Plentiful memory/compute
- Higher latency and delay
- Supported on by all platforms

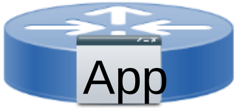
"End-Node"



On A Hardware Blade

- Dedicated memory/compute
- Low latency and delay
- Requires modular hardware blade

"Blade"

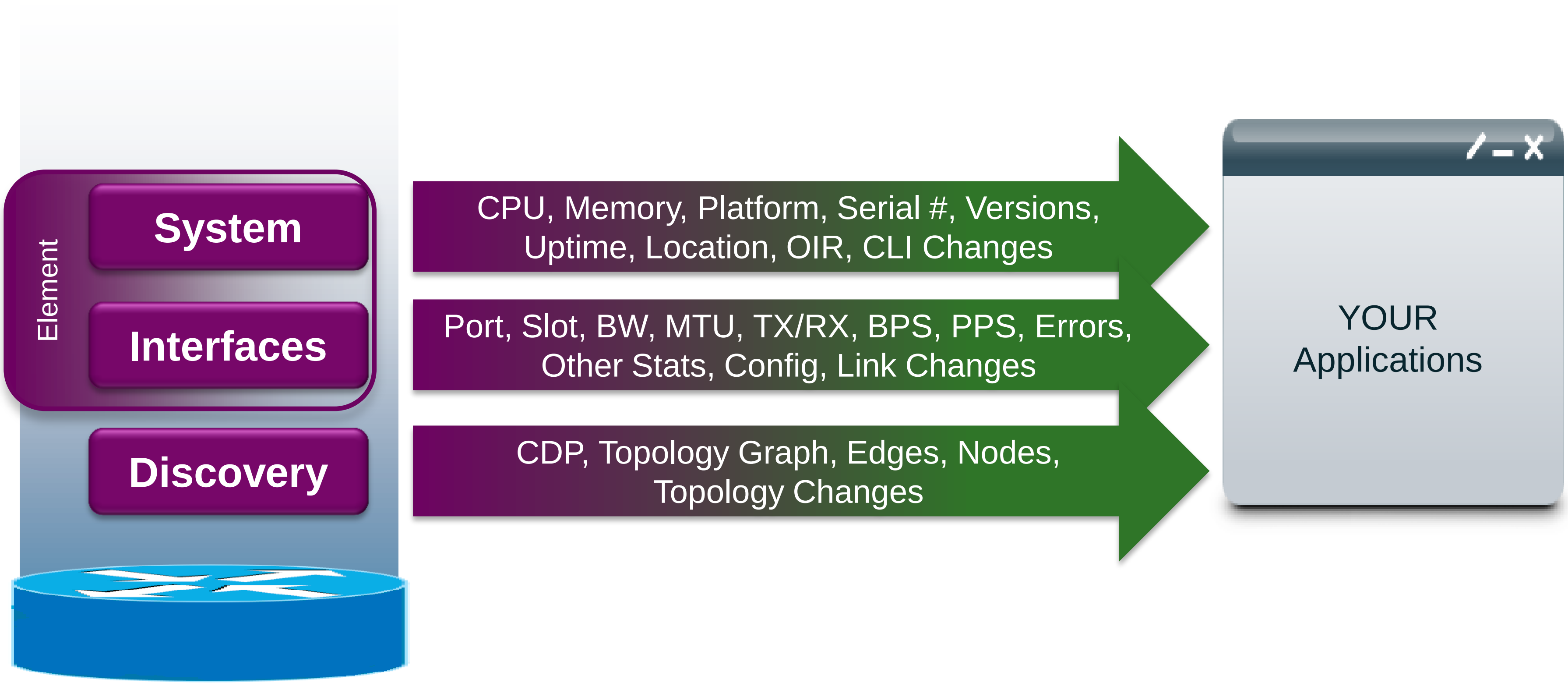


On the Router

- Shared memory/compute
- Very low latency and delay
- Requires modular software architecture

"Process"

onePK Service Sets – Element Properties – 1/2

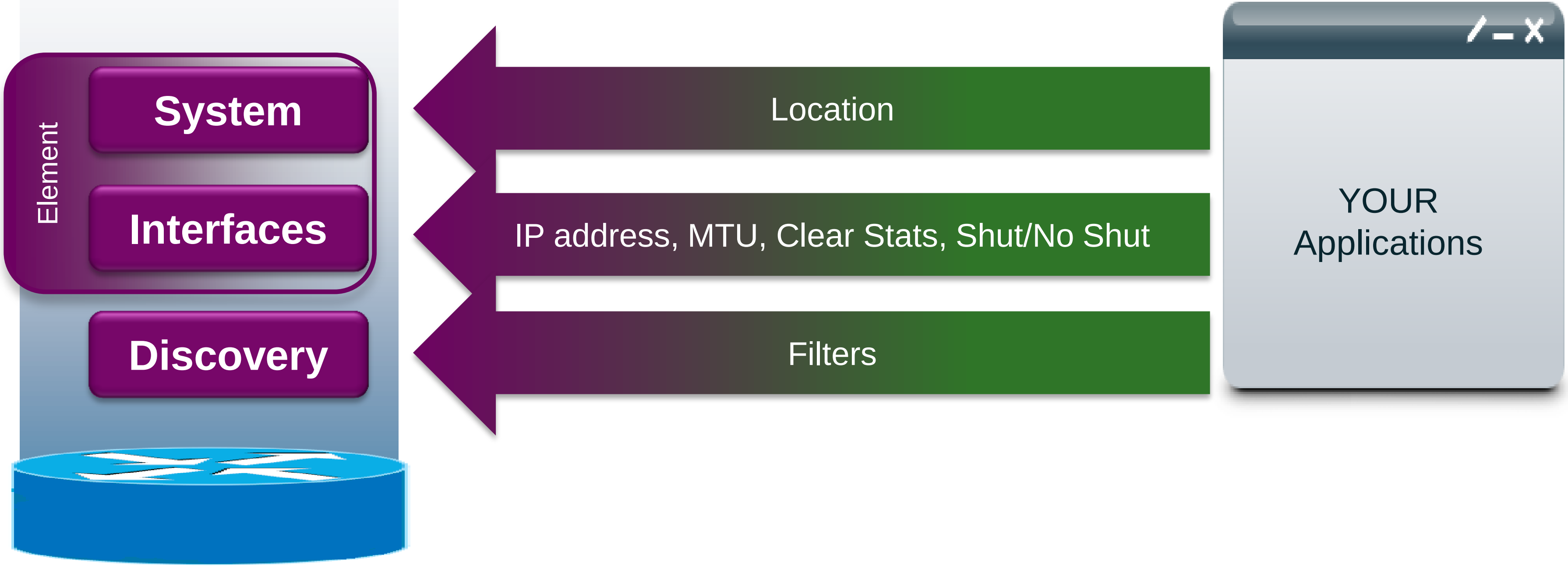


CPU, Memory, Platform, Serial #, Versions, Uptime, Location, OIR, CLI Changes

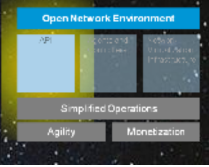
Port, Slot, BW, MTU, TX/RX, BPS, PPS, Errors, Other Stats, Config, Link Changes

CDP, Topology Graph, Edges, Nodes, Topology Changes

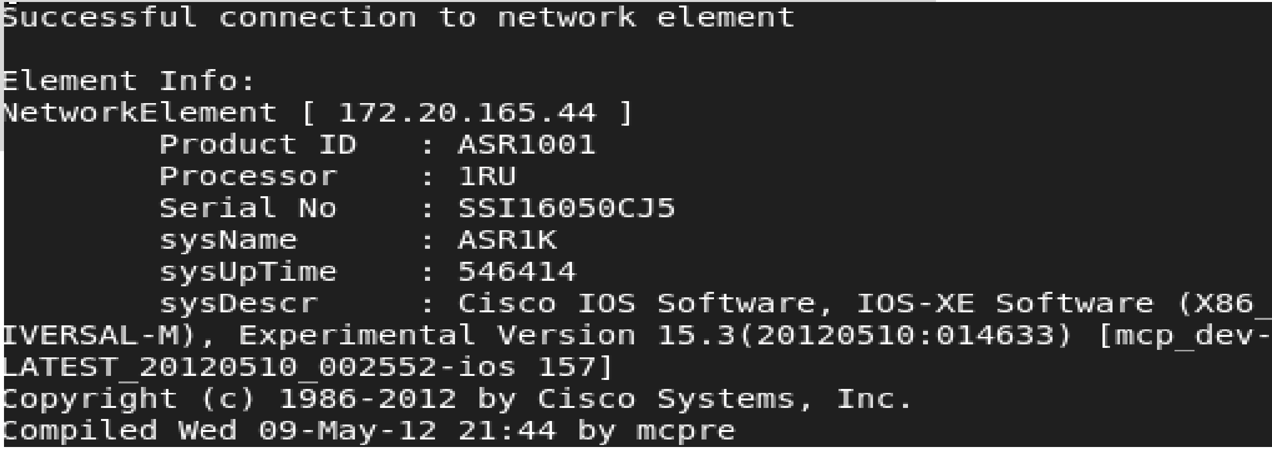
onePK Service Sets – Element Properties – 2/2



Connect and Get Properties via onePK (C)

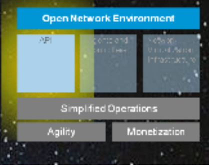


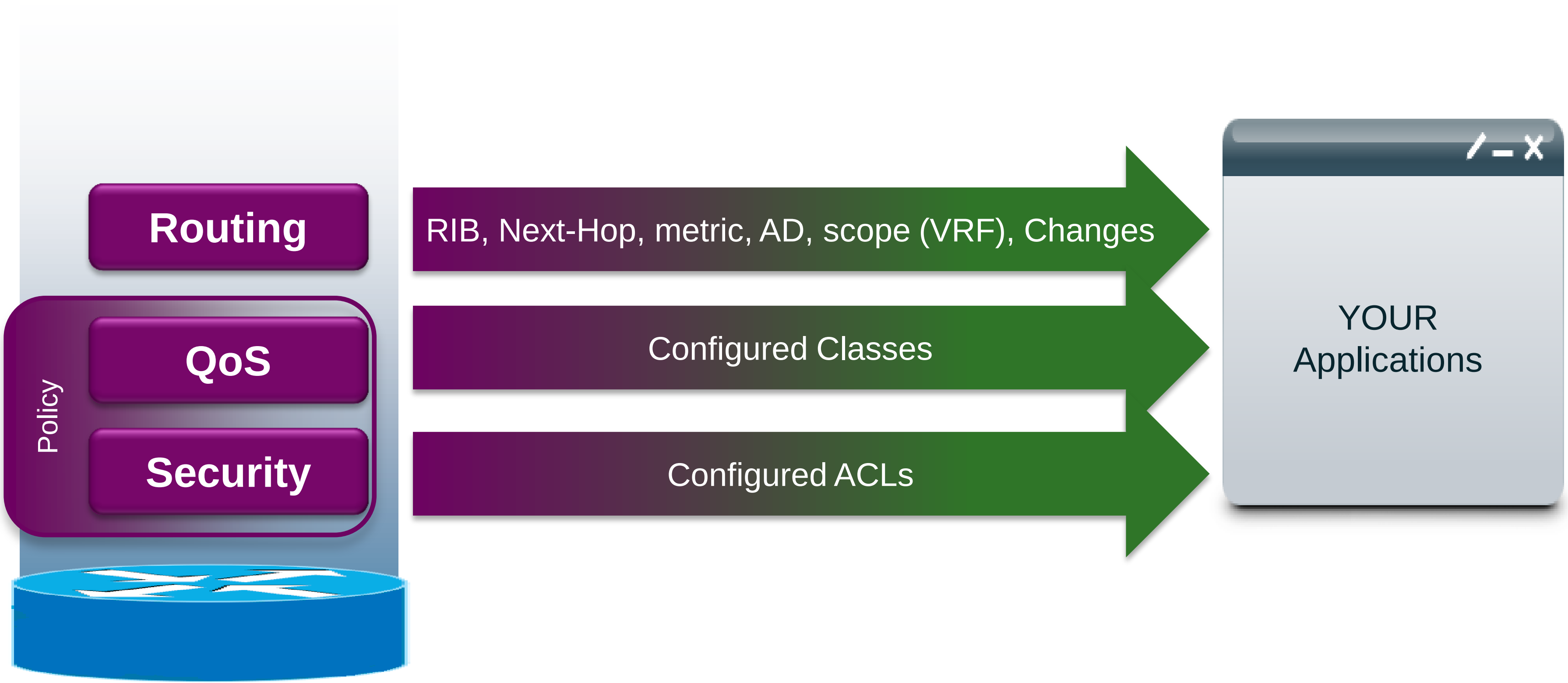
```
char *str = NULL;
onep_element_connect(elemA, user, pwd, NULL, &sh);
onep_element_get_property(elemA, &property);
if (property) {
        onep_element_to_string(elemA, &str);
            if (str) {
                fprintf(stderr, "\nElement Info: %s\n", str);
                free(str);
            }
}
```



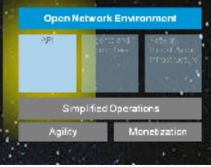
```
Successful connection to network element

Element Info:
NetworkElement [ 172.20.165.44 ]
        Product ID    : ASR1001
        Processor     : 1RU
        Serial No     : SSI16050CJ5
        sysName       : ASR1K
        sysUpTime     : 546414
        sysDescr      : Cisco IOS Software, IOS-XE Software (X86_
IVERSAL-M), Experimental Version 15.3(20120510:014633) [mcp_dev-
LATEST_20120510_002552-ios 157]
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Wed 09-May-12 21:44 by mcpre
```

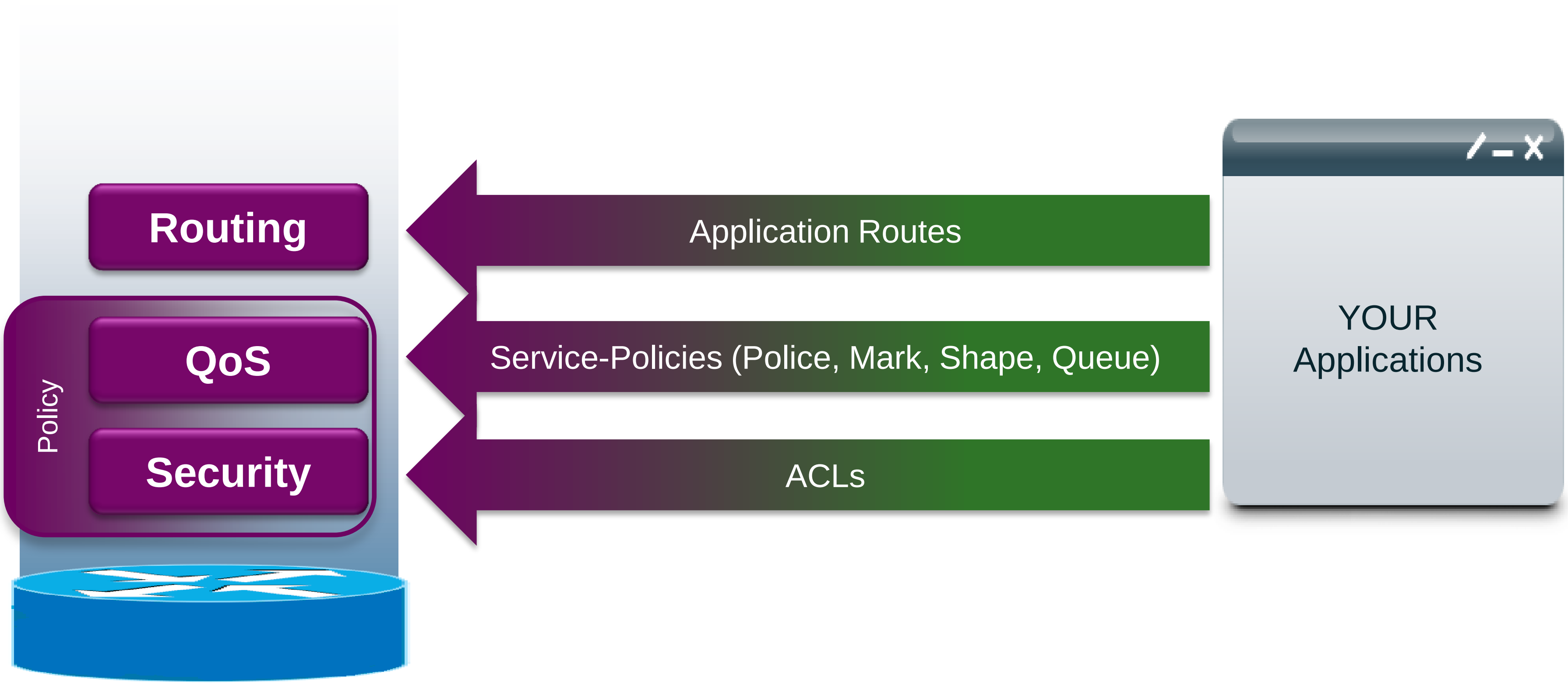
onePK Service Sets – Policy and Routing – 1/2



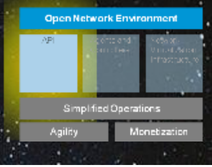


onePK Service Sets – Policy and Routing – 2/2

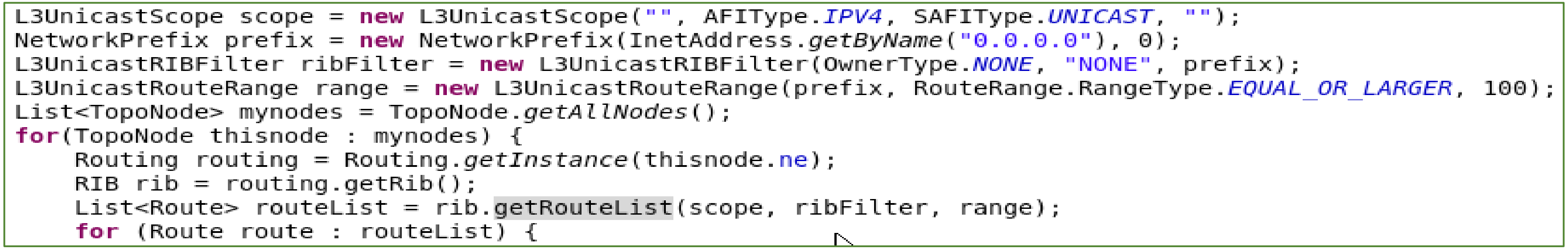




Example: Get and Set Routes via onePK (Java)

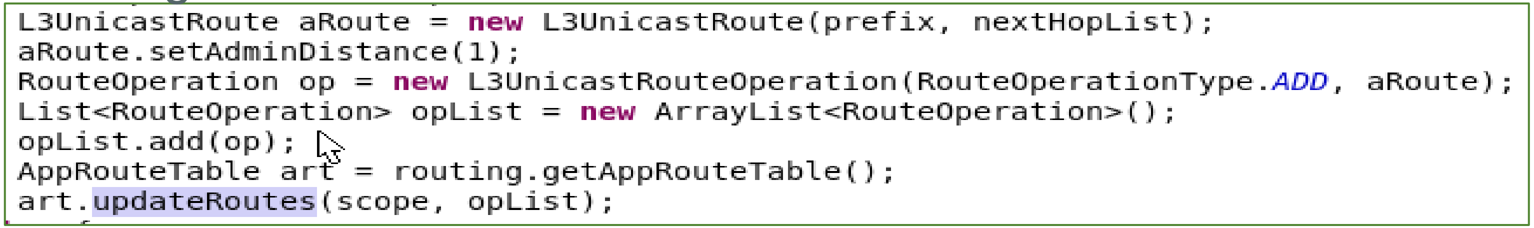


- Getting Routes



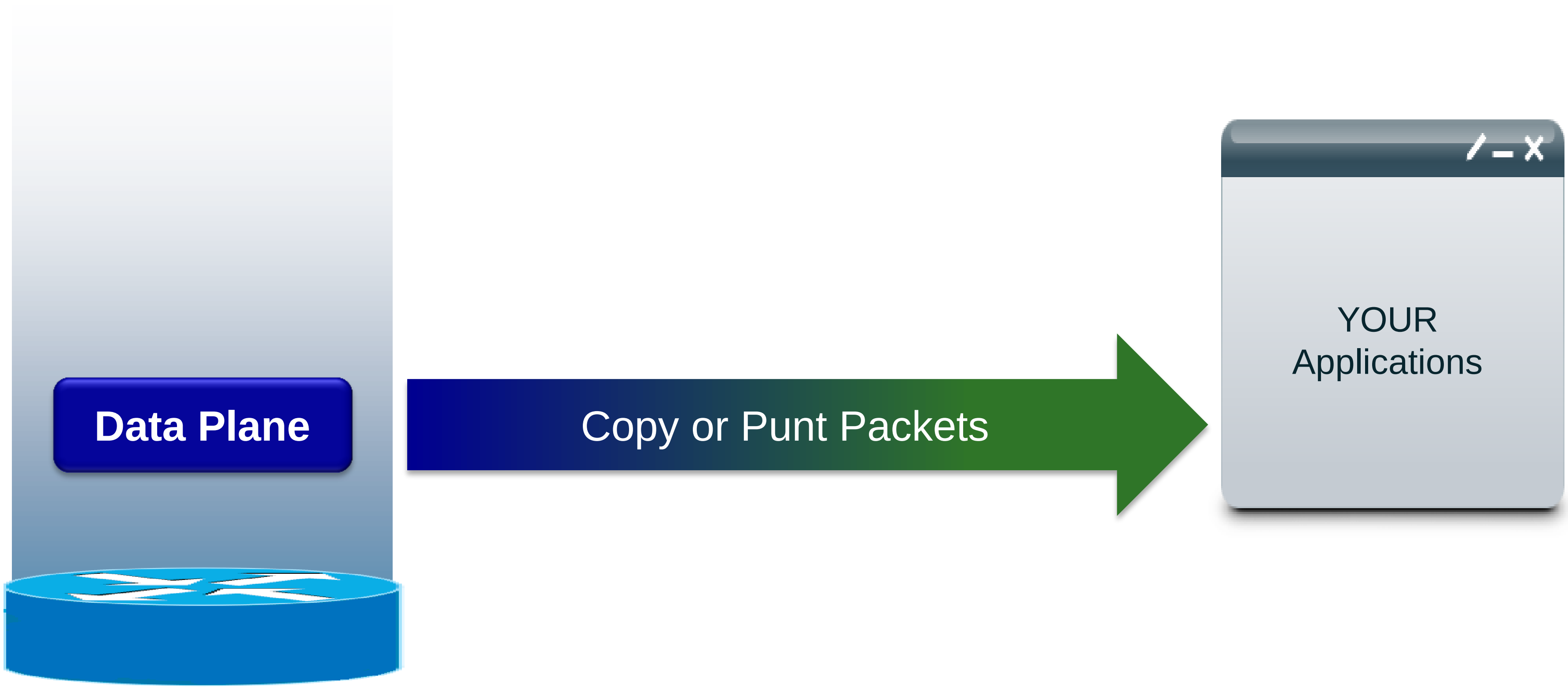
```
L3UnicastScope scope = new L3UnicastScope("", AFIType.IPV4, SAFIType.UNICAST, "");
NetworkPrefix prefix = new NetworkPrefix(InetAddress.getByName("0.0.0.0"), 0);
L3UnicastRIBFilter ribFilter = new L3UnicastRIBFilter(OwnerType.NONE, "NONE", prefix);
L3UnicastRouteRange range = new L3UnicastRouteRange(prefix, RouteRange.RangeType.EQUAL_OR_LARGER, 100);
List<TopoNode> mynodes = TopoNode.getAllNodes();
for(TopoNode thisnode : mynodes) {
    Routing routing = Routing.getInstance(thisnode.ne);
    RIB rib = routing.getRib();
    List<Route> routeList = rib.getRouteList(scope, ribFilter, range);
    for (Route route : routeList) {
```

- Setting Routes

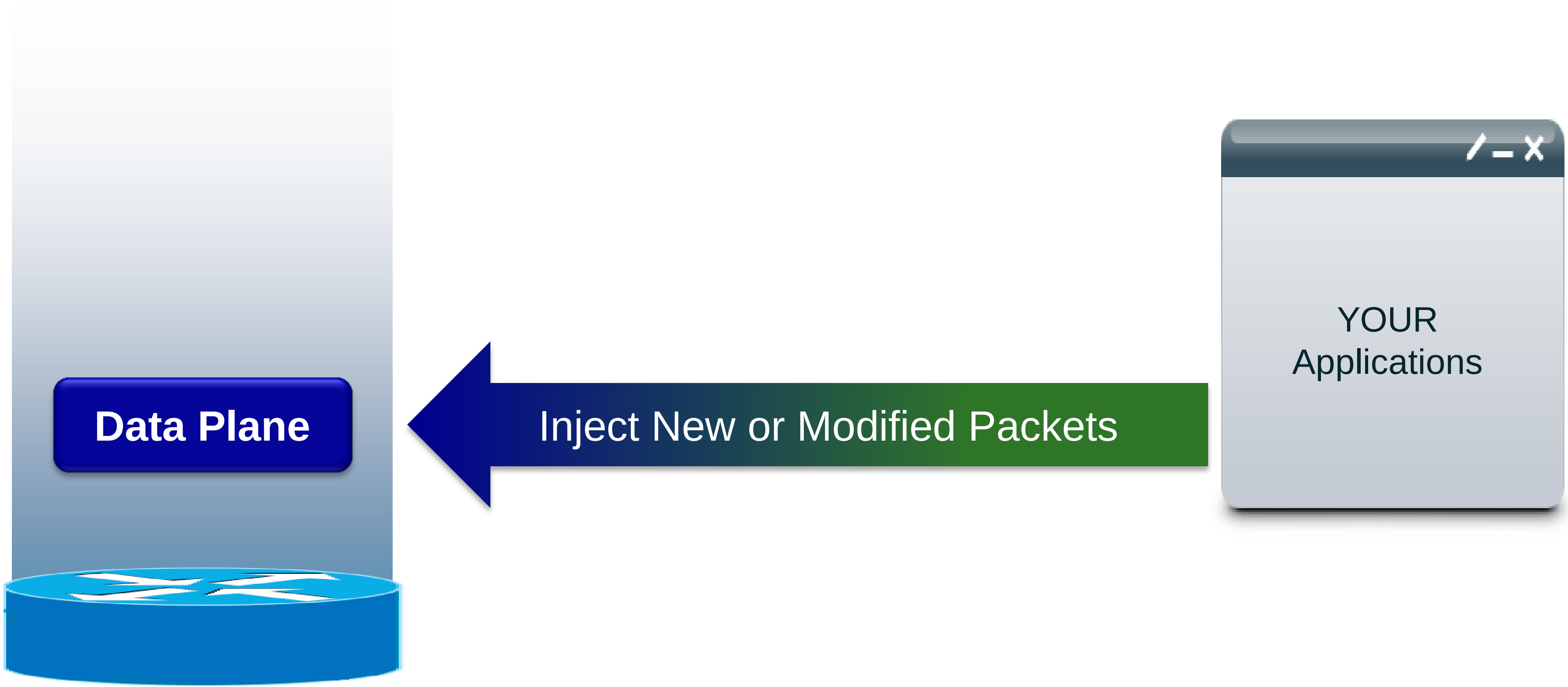


```
L3UnicastRoute aRoute = new L3UnicastRoute(prefix, nextHopList);
aRoute.setAdminDistance(1);
RouteOperation op = new L3UnicastRouteOperation(RouteOperationType.ADD, aRoute);
List<RouteOperation> opList = new ArrayList<RouteOperation>();
opList.add(op);
AppRouteTable art = routing.getAppRouteTable();
art.updateRoutes(scope, opList);
```

onePK Service Sets – Data Path – 1/2



onePK Service Sets – Data Path – 2/2

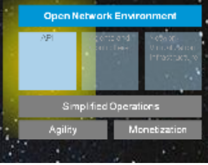


Data Plane

Inject New or Modified Packets

YOUR Applications

Example: Punt and Inject Packets via onePK (C)



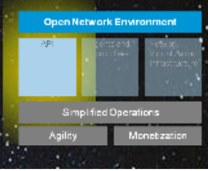
```
TRY(rc, onep_dpss_register_for_packets(
                 ne1,
                 dpss,
                 targ_left,
                 interesting_class,
                 ONEP_DPSS_ACTION_PUNT,
                 encrypt_callback,
                 (void *)intf_left,
                 &reg_handle), "Register for packets");
```

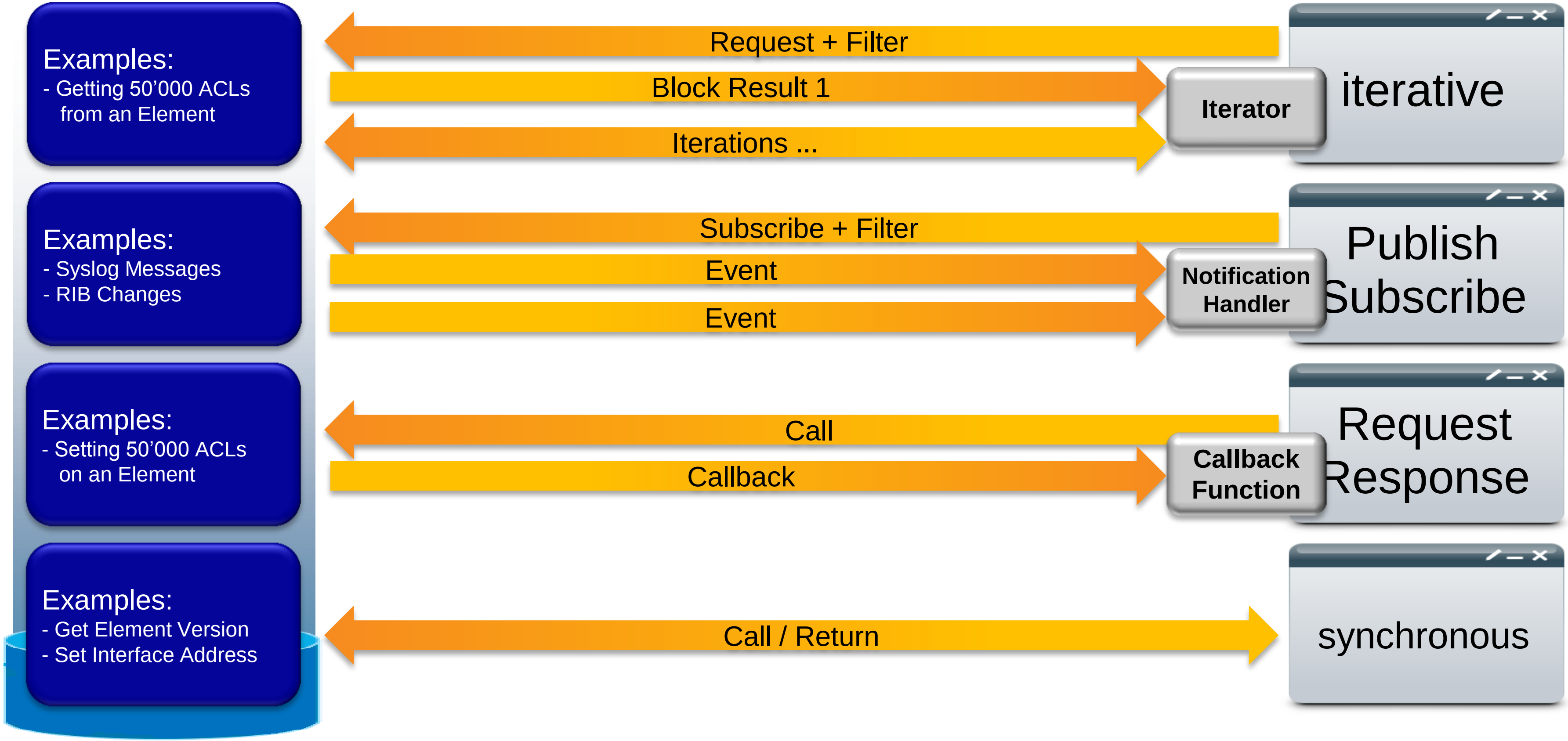
Defines traffic of interest

Action to take on interesting traffic

Where traffic goes next

onePK Agent ←→ Application Interactions





Examples:
- Getting 50’000 ACLs from an Element

Request + Filter

Block Result 1

Iterations ...

Iterator

iterative

Examples:
- Syslog Messages
- RIB Changes

Subscribe + Filter

Event

Event

Notification Handler

Publish Subscribe

Examples:
- Setting 50’000 ACLs on an Element

Call

Callback

Callback Function

Request Response

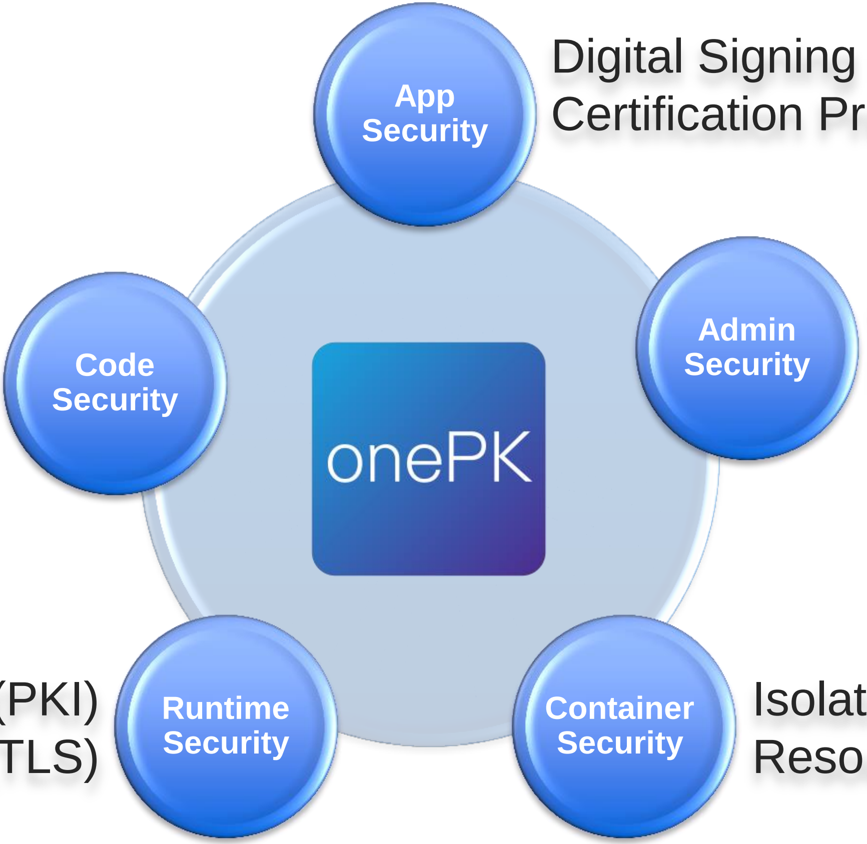
Examples:
- Get Element Version
- Set Interface Address

Call / Return

synchronous

Yes, it is secure

Security Five Ways



Digital Signing
Certification Process

CLI Control
Resource Allocation

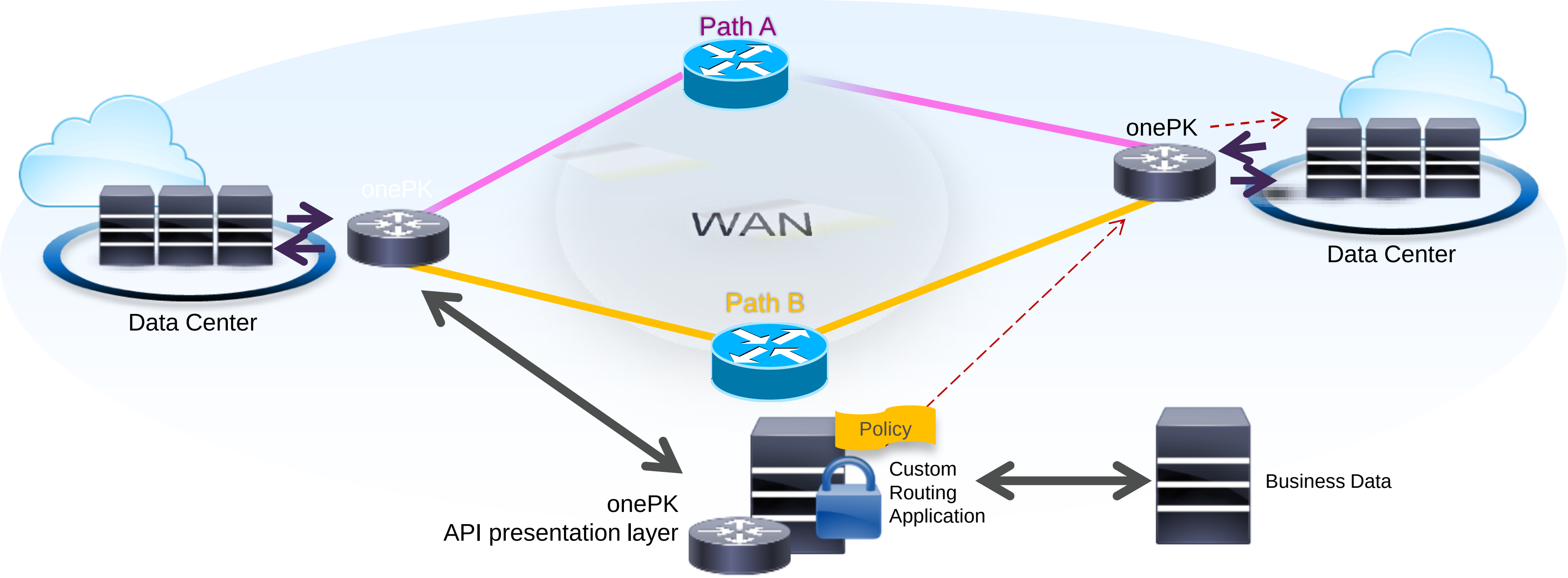
Isolation
Resource Consumption

AAA (PKI)
Encryption (TLS)

Code Isolation
Strong Typing

Example: Routing for Dollars / CO_2 / Tulips /…

Real-World Example



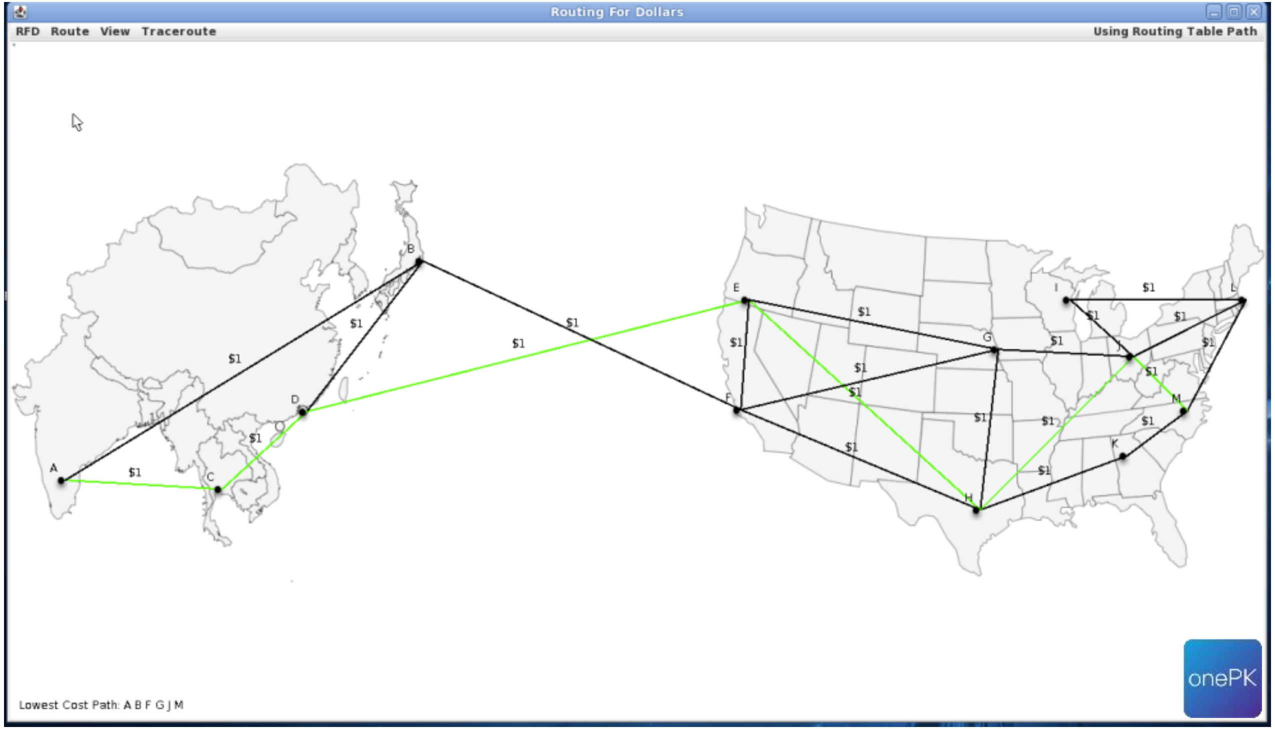
Network Extrinsic Metrics Influencing the Routing Topology

Real-World Example

Example: Routing for Dollars / CO_2 / Tulips /…

Setup

- EIGRP
- Routing Topology
- No External Metrics
- No External Algorithm



Using Routing Table Path

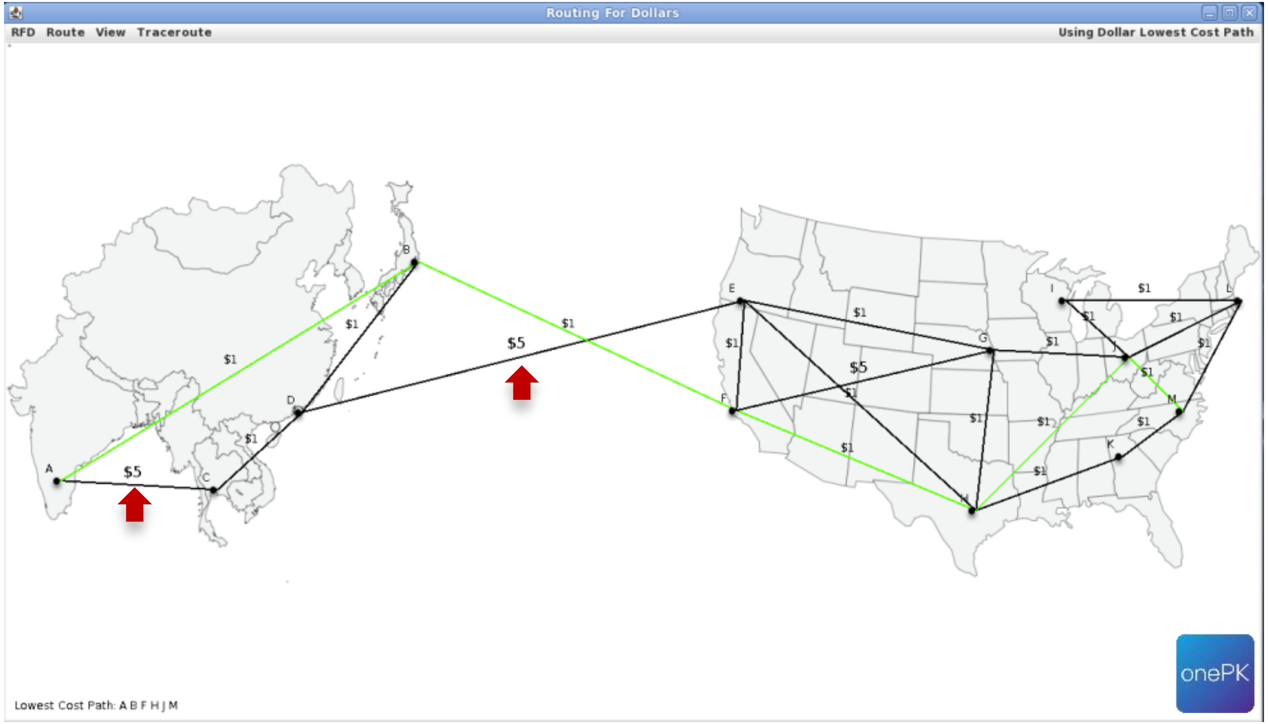
Lowest Cost Path: A B F G J M

Real-World Example

Example: Routing for Dollars / CO_2 / Tulips /…

Application Routes

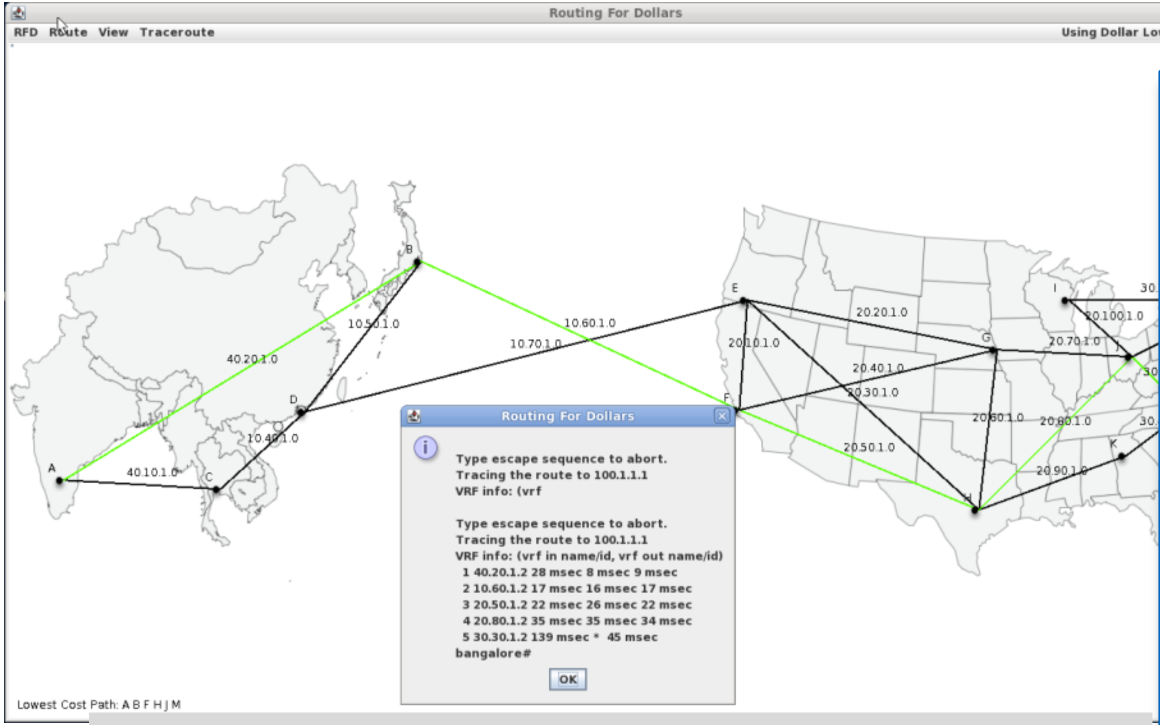
- EIGRP
- onePK
- External Metrics
- External Algorithm



Using Dollar Lowest Cost Path

Lowest Cost Path: A B F H J M

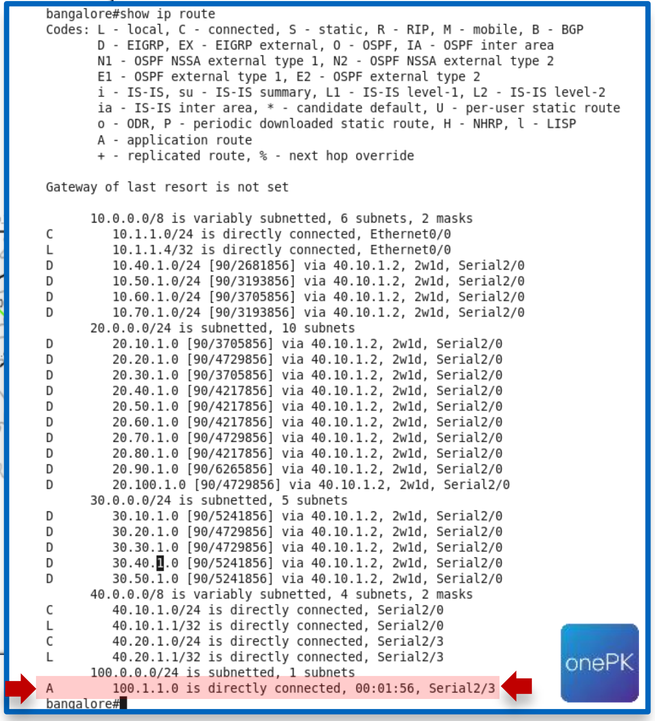
Example: Routing for Dollars / CO_2 / Tulips /…



```
Type escape sequence to abort.
Tracing the route to 100.1.1.1
VRF info: (vrf

Type escape sequence to abort.
Tracing the route to 100.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 40.20.1.2 28 msec 8 msec 9 msec
  2 10.60.1.2 17 msec 16 msec 17 msec
  3 20.50.1.2 22 msec 26 msec 22 msec
  4 20.80.1.2 35 msec 35 msec 34 msec
  5 30.30.1.2 139 msec *  45 msec
bangalore#
```

Lowest Cost Path: A B F H J M

```
router ospf 1
redistribute application  <app name> ...
```



```
bangalore#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       A - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.1.1.0/24 is directly connected, Ethernet0/0
L        10.1.1.4/32 is directly connected, Ethernet0/0
D        10.40.1.0/24 [90/2681856] via 40.10.1.2, 2w1d, Serial2/0
D        10.50.1.0/24 [90/3193856] via 40.10.1.2, 2w1d, Serial2/0
D        10.60.1.0/24 [90/3705856] via 40.10.1.2, 2w1d, Serial2/0
D        10.70.1.0/24 [90/3193856] via 40.10.1.2, 2w1d, Serial2/0
      20.0.0.0/24 is subnetted, 10 subnets
D        20.10.1.0 [90/3705856] via 40.10.1.2, 2w1d, Serial2/0
D        20.20.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
D        20.30.1.0 [90/3705856] via 40.10.1.2, 2w1d, Serial2/0
D        20.40.1.0 [90/4217856] via 40.10.1.2, 2w1d, Serial2/0
D        20.50.1.0 [90/4217856] via 40.10.1.2, 2w1d, Serial2/0
D        20.60.1.0 [90/4217856] via 40.10.1.2, 2w1d, Serial2/0
D        20.70.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
D        20.80.1.0 [90/4217856] via 40.10.1.2, 2w1d, Serial2/0
D        20.90.1.0 [90/6265856] via 40.10.1.2, 2w1d, Serial2/0
D        20.100.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
      30.0.0.0/24 is subnetted, 5 subnets
D        30.10.1.0 [90/5241856] via 40.10.1.2, 2w1d, Serial2/0
D        30.20.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
D        30.30.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
D        30.40.1.0 [90/5241856] via 40.10.1.2, 2w1d, Serial2/0
D        30.50.1.0 [90/5241856] via 40.10.1.2, 2w1d, Serial2/0
      40.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        40.10.1.0/24 is directly connected, Serial2/0
L        40.10.1.1/32 is directly connected, Serial2/0
C        40.20.1.0/24 is directly connected, Serial2/3
L        40.20.1.1/32 is directly connected, Serial2/3
      100.0.0.0/24 is subnetted, 1 subnets
A        100.1.1.0 is directly connected, 00:01:56, Serial2/3
bangalore#
```

onePK

Example: Routing for Dollars / CO_2 / Tulips /…

Real-World Example

Statistics and Metrics

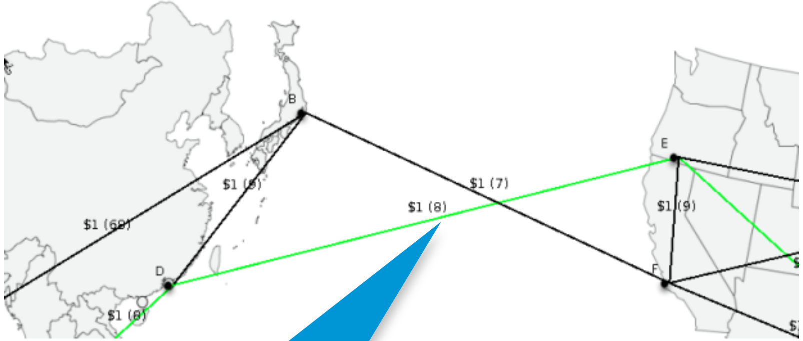
- Code Metrics
  - Total lines of code: 4700 (JAVA)
  - 40% SWING GUI
  - 20% Dijkstra’s algorithm, lowest cost path determination
  - 25% Housekeeping: Node and link database
  - 15% Calls to onePK infrastructure + error checking
- Code increase to add “Latency based routing” on top of “Routing for Dollars”
  - 100 lines of code
- Modular code base written in Java has allowed us to port this to mobility client.

Framework makes it easy to modify code and change business logic.

Modular java code makes it easy to deploy on multiple clients.

Real-World Example

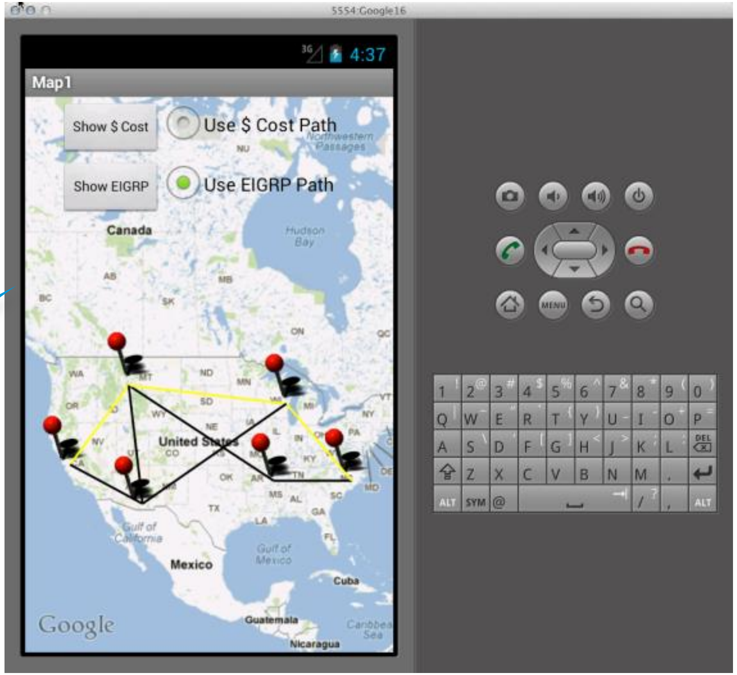
Custom Routing: Modifications



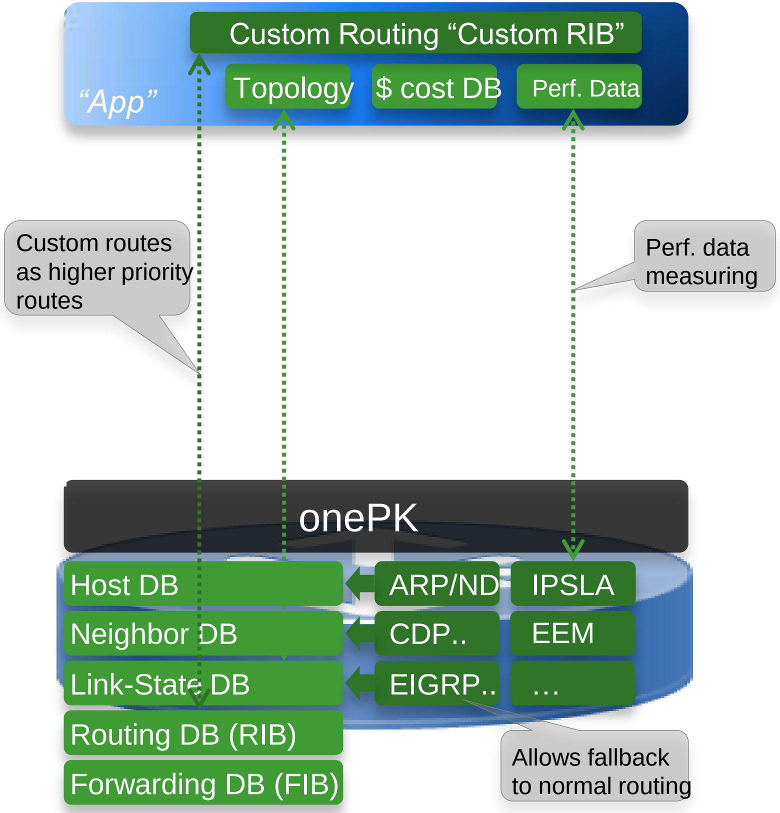
Path determination based on lowest latency

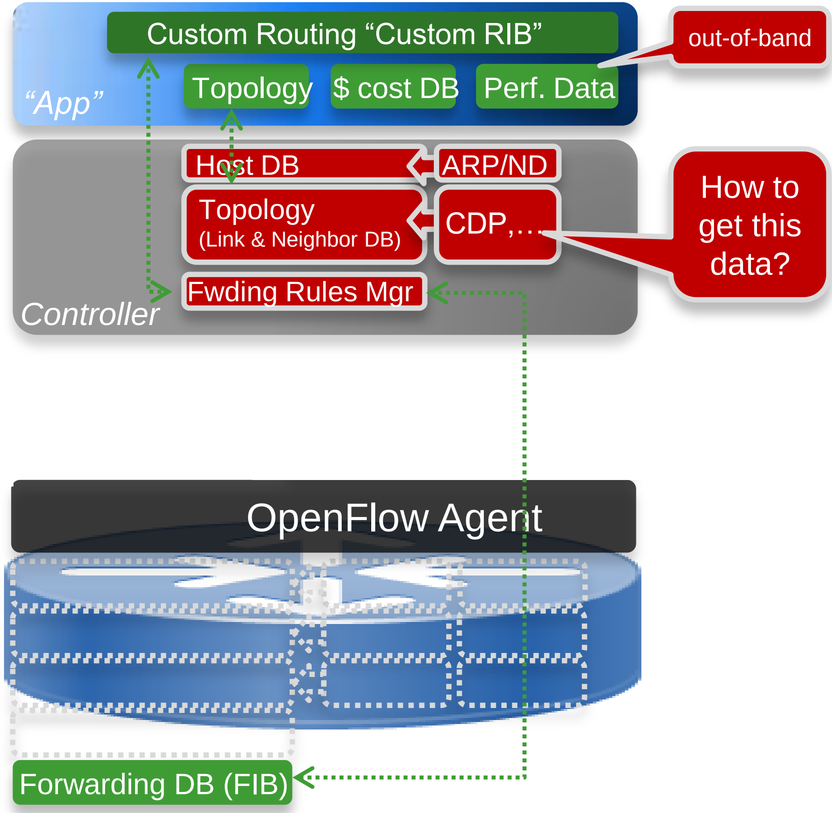
Latency information fed into app through IPSLA

Port to mobility client



Custom Routing: Implementation Variants





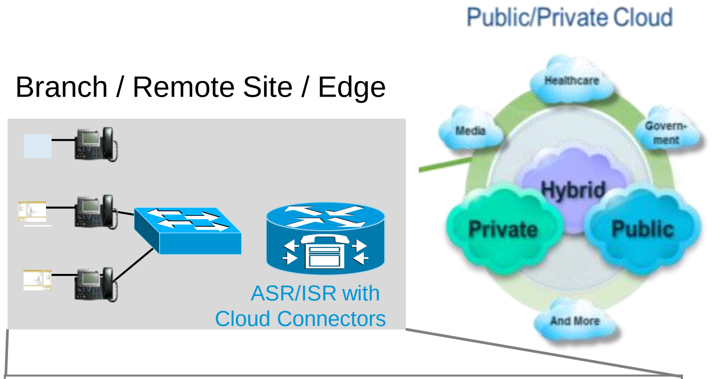
Example: Cloud Connectors

Cloud Connectors Provide

- Network-Awareness to Cloud Services
- Cloud Service-Awareness to Network
- Improved Quality and Experience
- Simplified Deployment and Operations

Cloud Connector Anatomy

- Deployed into Branch on ASR/ISR
- Native (in Network OS) or Hosted (on SRE, UCS-E Blad
- Abstractions on top of Network OS



Some Examples

Available Now

- Scansafe Connector
- HCS Connector
- Webex Cloud Connect Audio

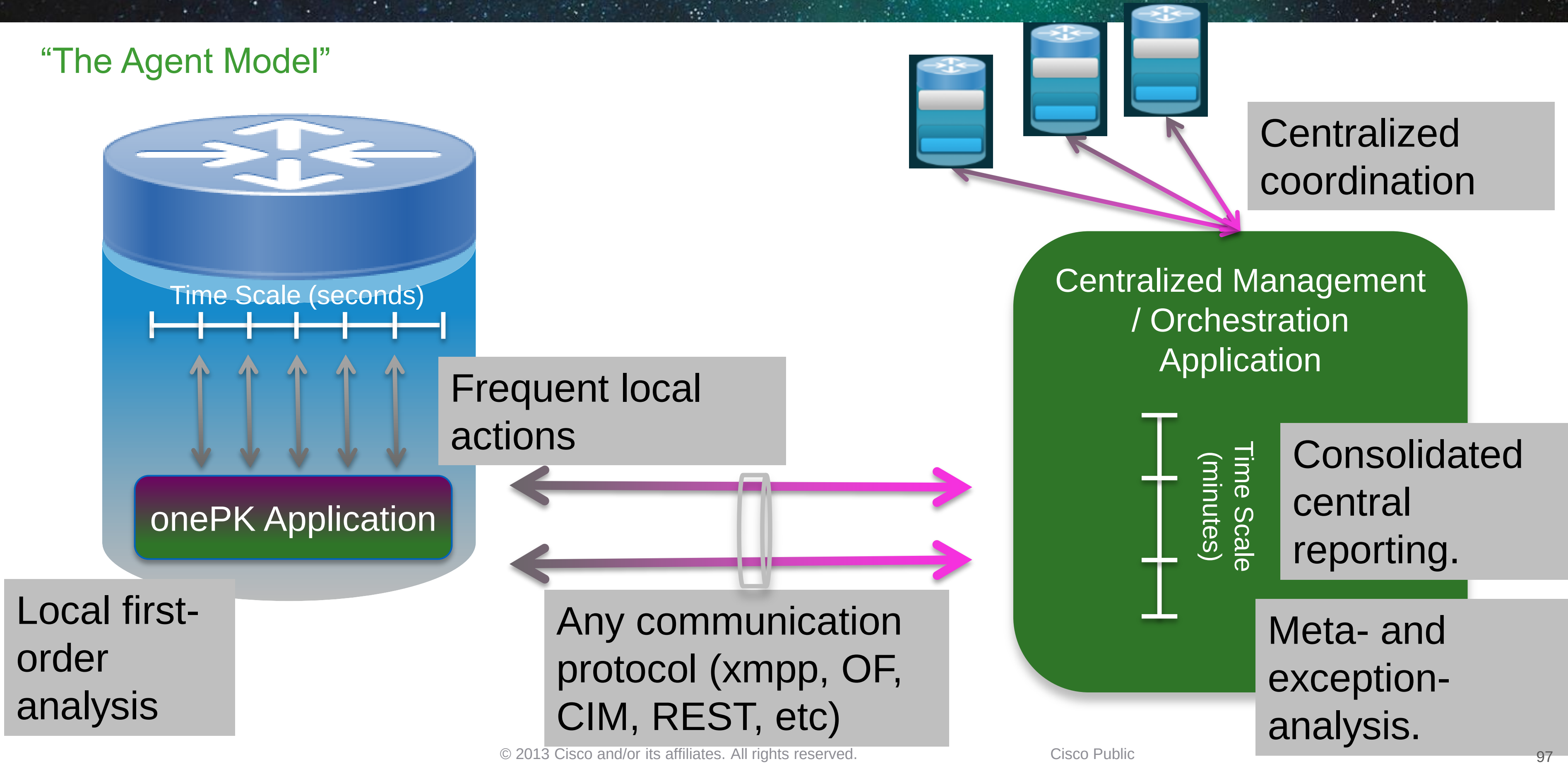
Future

- Backup/Storage Connector
- Identity Services Connector
- Securelogix / UC Services Connector
- VXI Connector

20+ Cloud Connectors available from marketplace.cisco.com !!

Properties Use Case: Network Be Nimble…

“The Agent Model”



Frequent local actions

Local first-order analysis

Any communication protocol (xmpp, OF, CIM, REST, etc)

Centralized coordination

Consolidated central reporting.

Meta- and exception-analysis.

Feedback Loop Applications

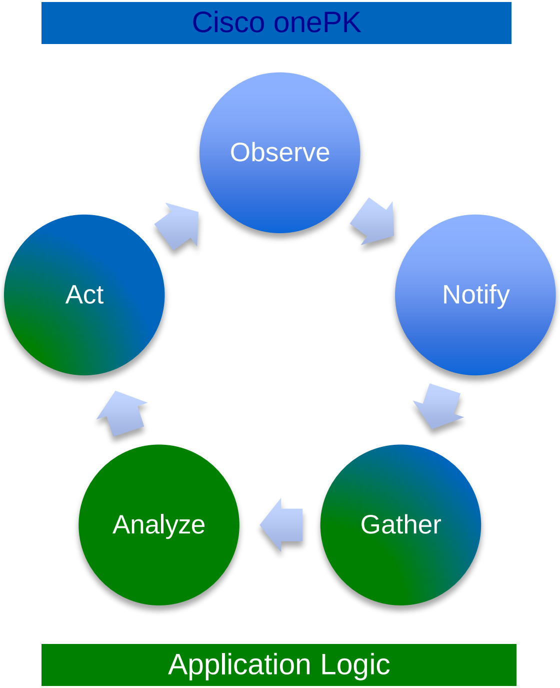
Integration of application and the network

Application domain tasks

- Gather, Analyze, Receive Requests
- Makes a decision, pushes back to Network Element

Network domain tasks

- Act, Observe, Notify
- Application can delegate rules to network to enable the network to take local decisions

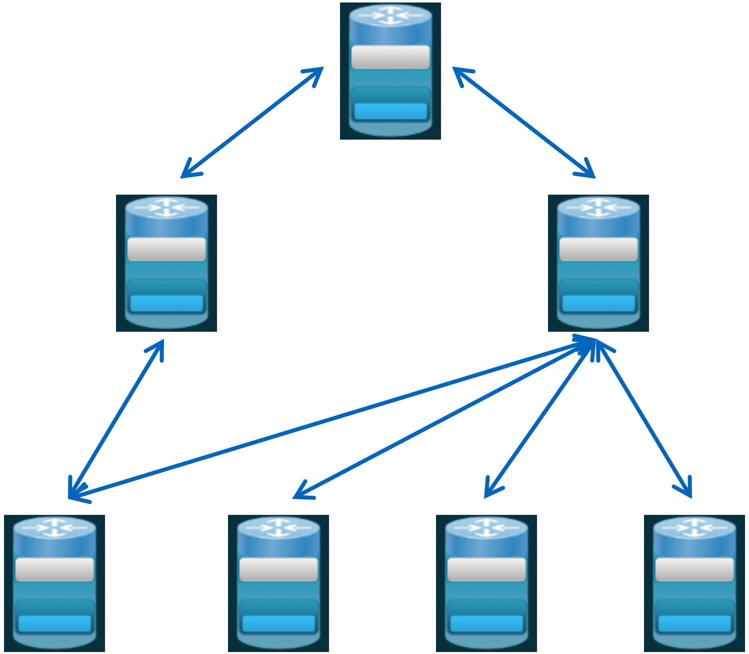


Peer to Peer Applications

Applications can reside within network elements communicating with each other.

Decentralized control

Example: Locally designed routing protocols
Self correcting applications



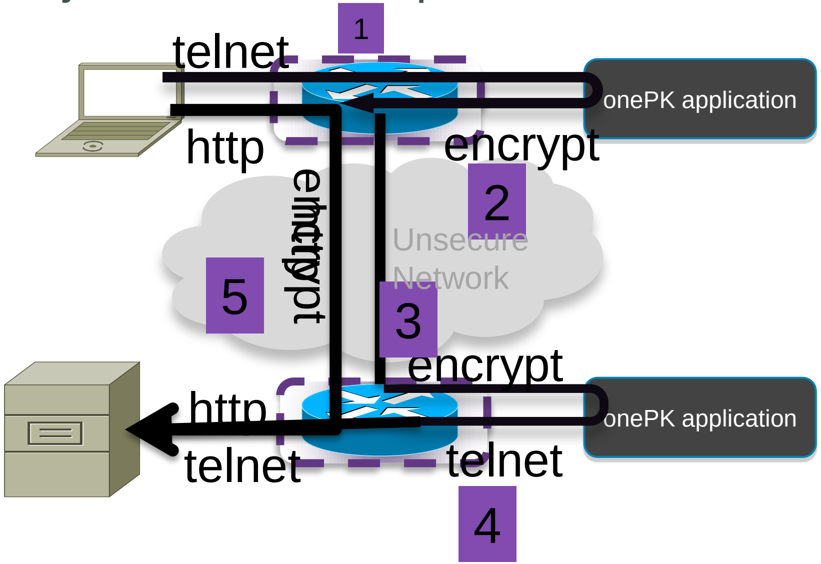
Real-World Example

Custom Application Traffic Flow Handling – 1/2

Problem: We need to custom encrypt packets of a specific application traffic flow

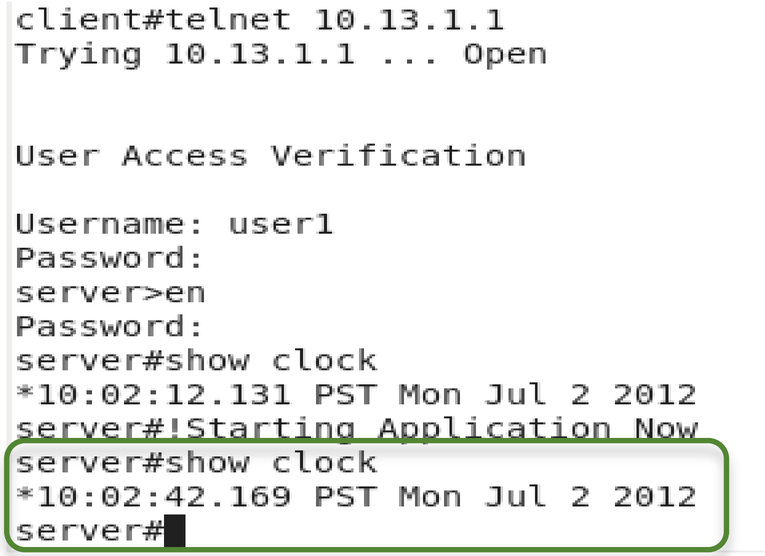
Solution: Use onePK to punt, encrypt and reinject the relevant packets

1. Policy APIs on ingress router are set to punt telnet and syslog to app
2. App encrypts punted traffic and re-injects into data path.
3. Policy APIs on egress router punt telnet and syslog to app
4. App decrypts punted traffic and re-injects into data path.
5. Traffic that does not match policy passes through unencrypted.



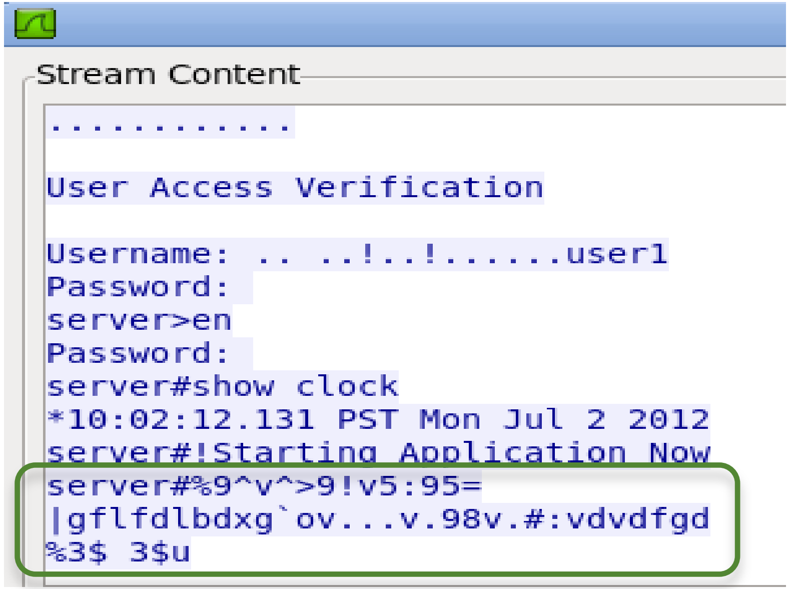
Custom Application Traffic Flow Handling – 2/2

What Client Sees



```
client#telnet 10.13.1.1
Trying 10.13.1.1 ... Open


User Access Verification

Username: user1
Password:
server>en
Password:
server#show clock
*10:02:12.131 PST Mon Jul 2 2012
server#!Starting Application Now
server#show clock
*10:02:42.169 PST Mon Jul 2 2012
server#
```

What Wireshark Sees



Stream Content

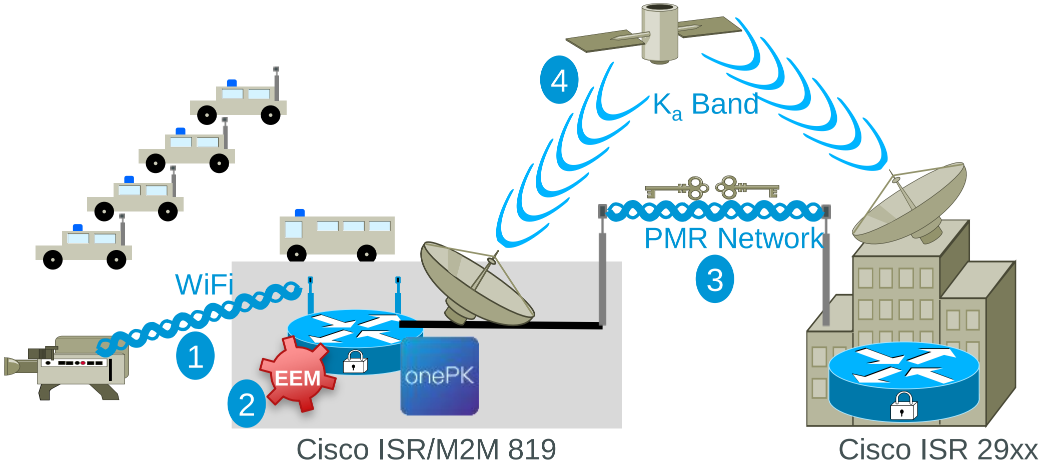
```
............

User Access Verification

Username: .. ..!..!......user1
Password:
server>en
Password:
server#show clock
*10:02:12.131 PST Mon Jul 2 2012
server#!Starting Application Now
server#%9^v^>9!v5:95=
|gflfdlbdxg`ov...v.98v.#:vdvdfgd
%3$ 3$u
```

Real-World Example

Emergency Response Network

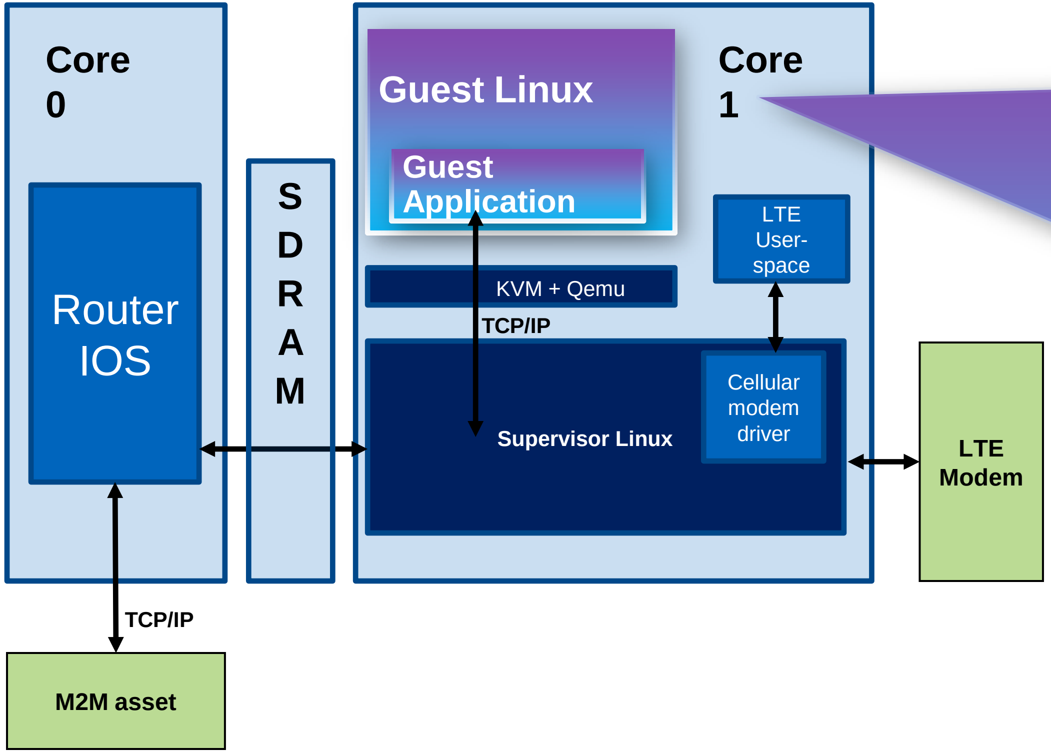
Problem: How to deliver secure, trusted, robust, cost-effective broadband connectivity to mobile emergency response units?

Solution: Use Network Programming based on Cisco onePK and Cisco IOS Embedded Event Manager to integrate low-cost, high-bandwidth options with accredited legacy radio connectivity:



1. Connect high-bandwidth forward clients via WiFi
2. Use Cisco IOS EEM for onboard system integration and adaptation
3. Use Cisco onePK to redirect IKE key exchange out-of-band via legacy radio
4. Secure IPSec tunnel via cost-effective high bandwidth K_a Band
5. Reliable, secure emergency response network saving ~4M€ operating cost annually

819 Guest Linux and Guest Application Hosting



YOUR Application running aboard a Cisco 819 M2M Router

Guest Linux

Memory Footprint (incl Guest App):	**< 256 MB**
Bare bone Kernel 3.0.6	

Guest Application

Memory Footprint:	**< 64 MB**

Field Trial Summer 2013

Enrollment open now ....

Puppet and Chef: Open Source Configuration Agents

- Objective:

  Configure/Provision DC switches using Puppet or Chef

- Motivation:

  Puppet or Chef widely used for compute node software configuration management

  Allows for configuration AT SCALE

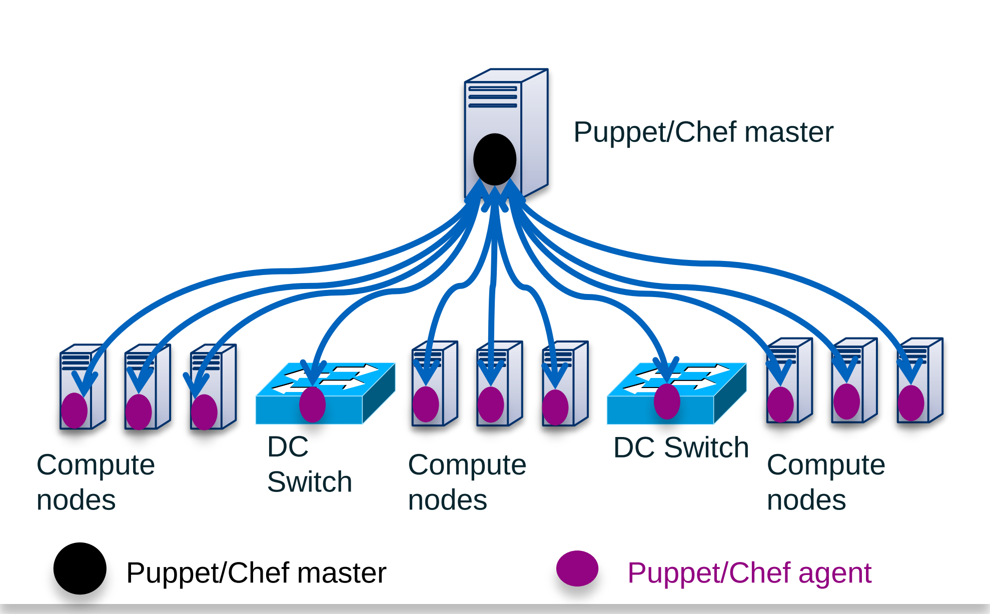
  Extend the same toolset to manage network

- Technical Requirements:

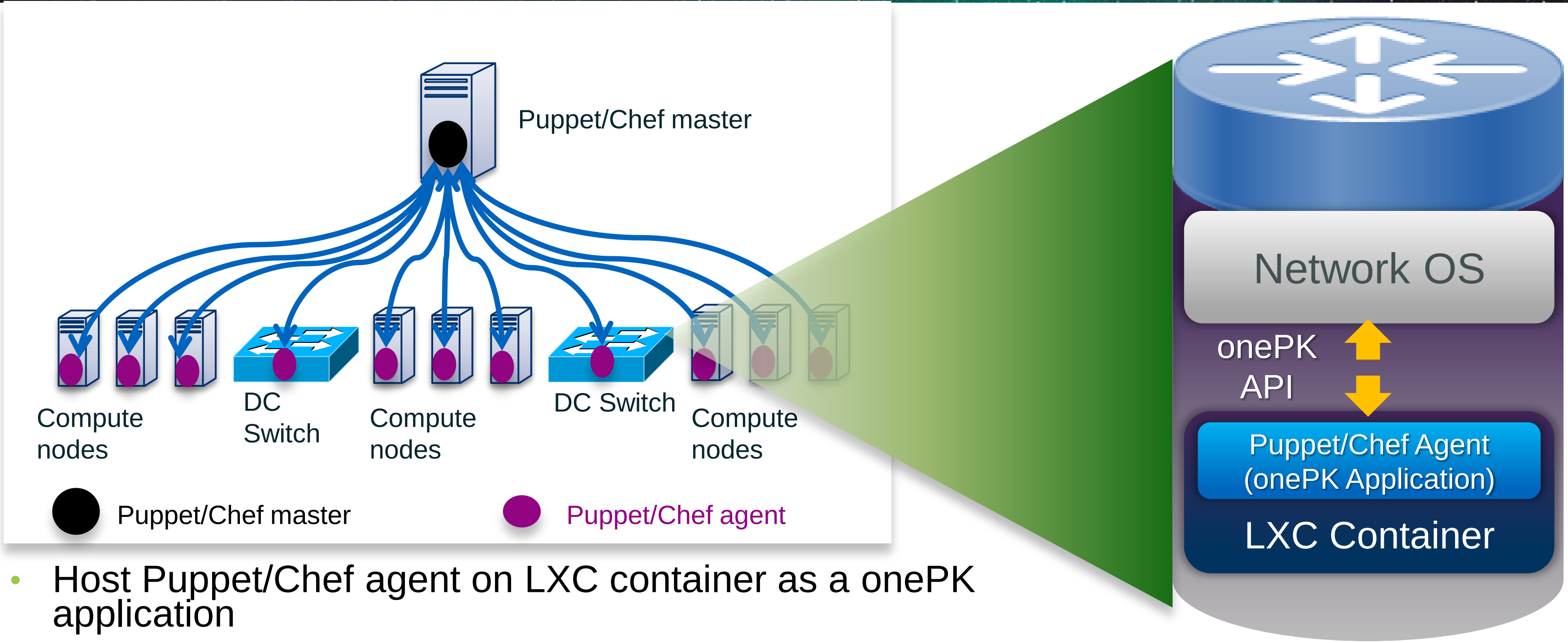
  Host Open Source Puppet/Chef agent on Nexus switches

  Create plug-ins for Puppet/Chef models (manifests/recipes)

  onePK API to inject config change



Puppet/Chef Agent Implementation



- Host Puppet/Chef agent on LXC container as a onePK application
- Use onePK configuration API to implement configuration tasks
- Future extensions – Image and software upgrade management

cisco TM