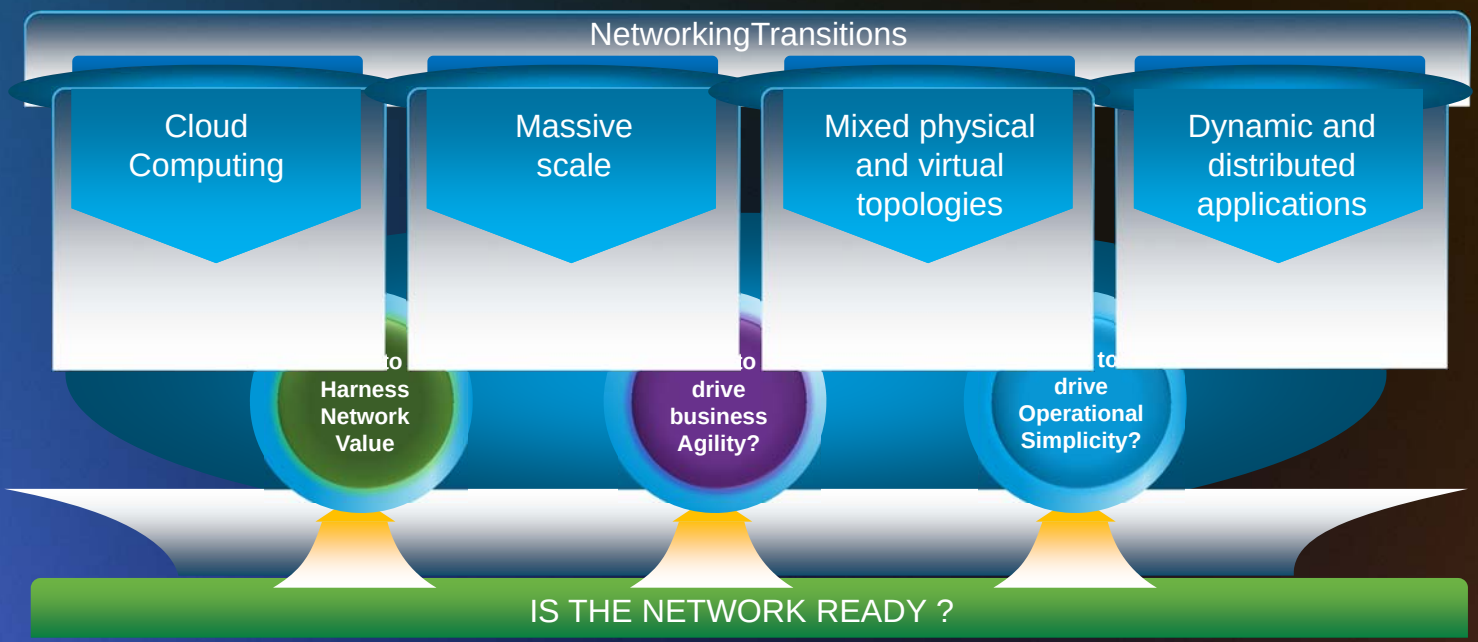
Software Defined Networks, onePK and a New World of Custom Network Applications"

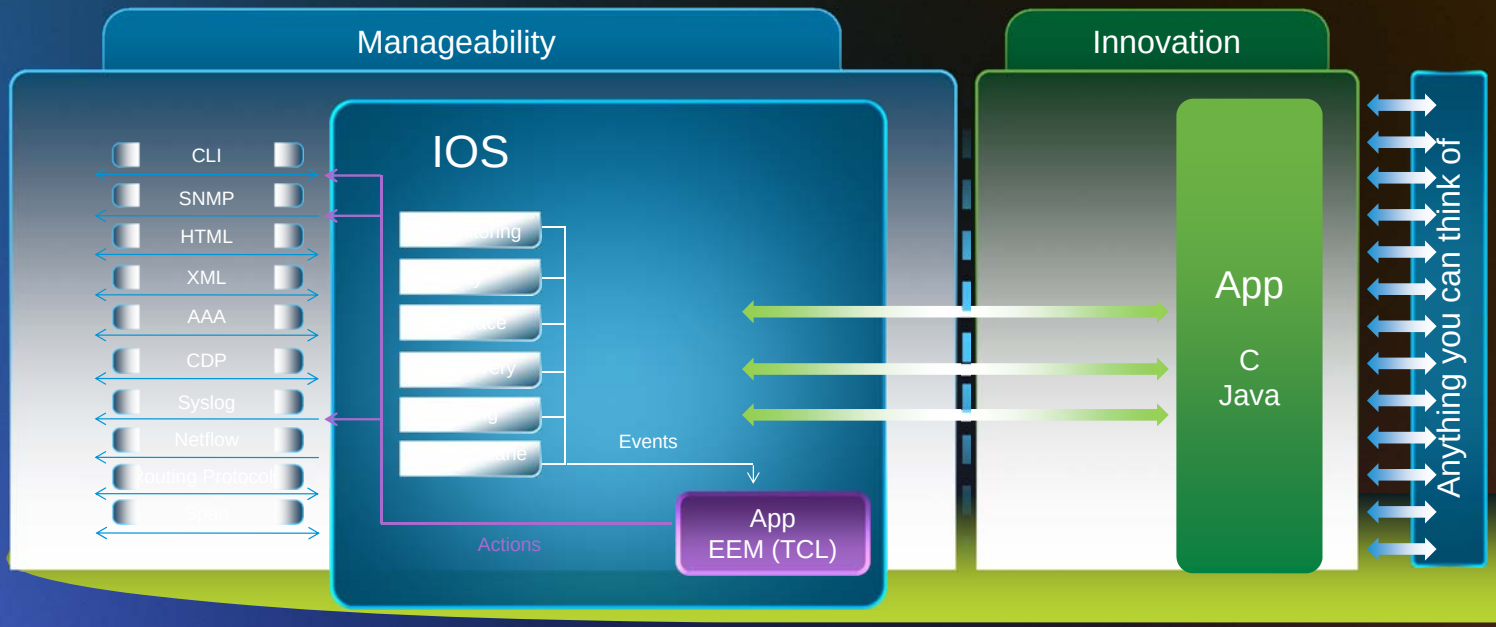
Richard Pruss
Network Operating Systems Technology Group, Cisco Systems Inc.

July 11, 2012

Network Transitions Driving Greater Demands



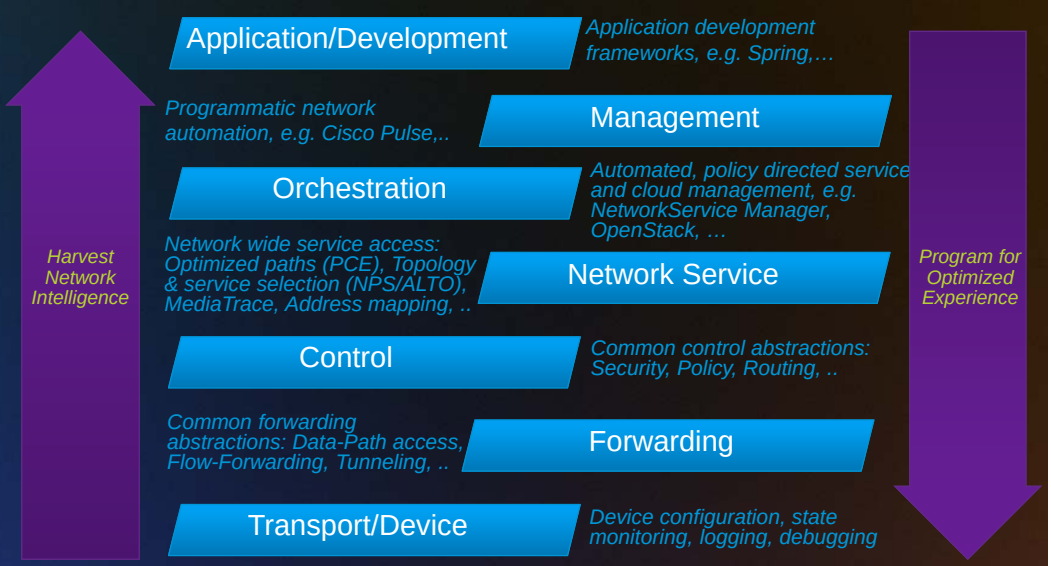
Evolving How We Interact With The Network Operating System



Programmatic Network Access – Multiple Layers

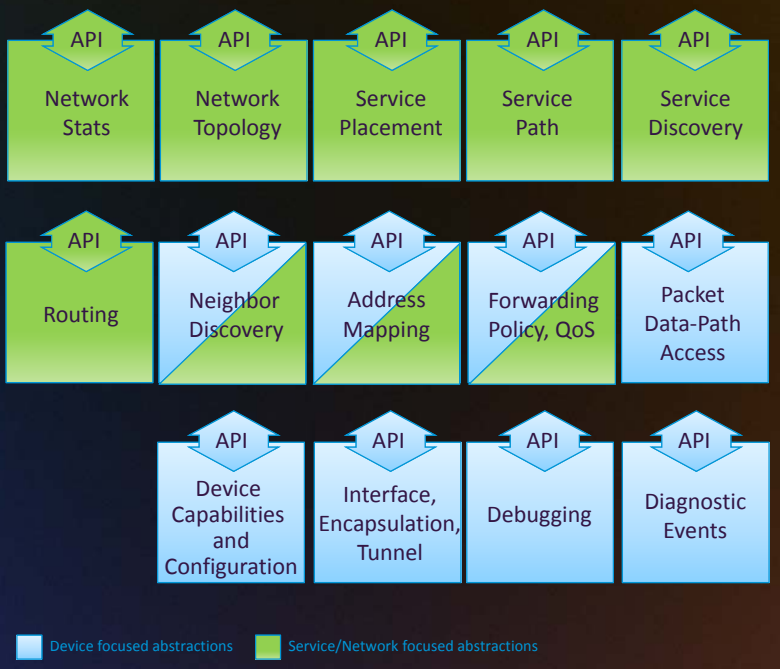
Full-Duplex access to the network at multiple layers and networking planes

- 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 APIs)
- Multiple Language Support (C, Java, …)
- Integrate with customer development environment to deliver enhanced functionality
- Reduced time to market by leveraging common platform for building services



Approaching abstractions for Networking

- Data-plane Abstractions – ISO Layering
  - Data plane abstractions key to Internet's success
- Abstractions for the other planes (control, services, management, orchestration,..)
  - … are missing
- Define network abstractions and associated APIs
  - Enable a holistic Network Programming model
  - Leverage and extend infrastructure at pace of the business
  - Deploy common applications across all devices
  - Extend/upgrade/add features without upgrading the network operating system



APIs make Abstractions available to Programmers

Not all Networking APIs are created the same: API classes and their scope

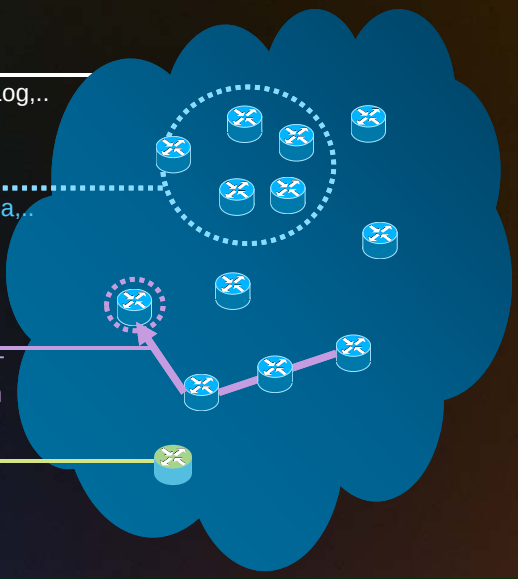
- Classify Networking APIs based on their *scope*
  - API Scopes: Location independent; Area; Particular place; Specific device
  - Alternate approaches like device/network/service APIs difficult to associate with use cases
  - Location where an API is hosted can differ from the scope of the API
- Different network planes could implement different flavors of APIs, based on associated abstractions

Utility
Example: Get Auth, Publish Log,..
Scope: Location independent

Area/Set
Example: Domain, OSPF-area,..
Scope: Group/Set/Area

Place in the Network
Example: Edge Session, NAT
Scope: Specific place/location

Element
Example: interface statistics
Scope: Specific element



Scope of an API and the location where the API is hosted are orthogonal

APIs make Abstractions available to Programmers

Example: Cisco's onePK (one Programming Kit)

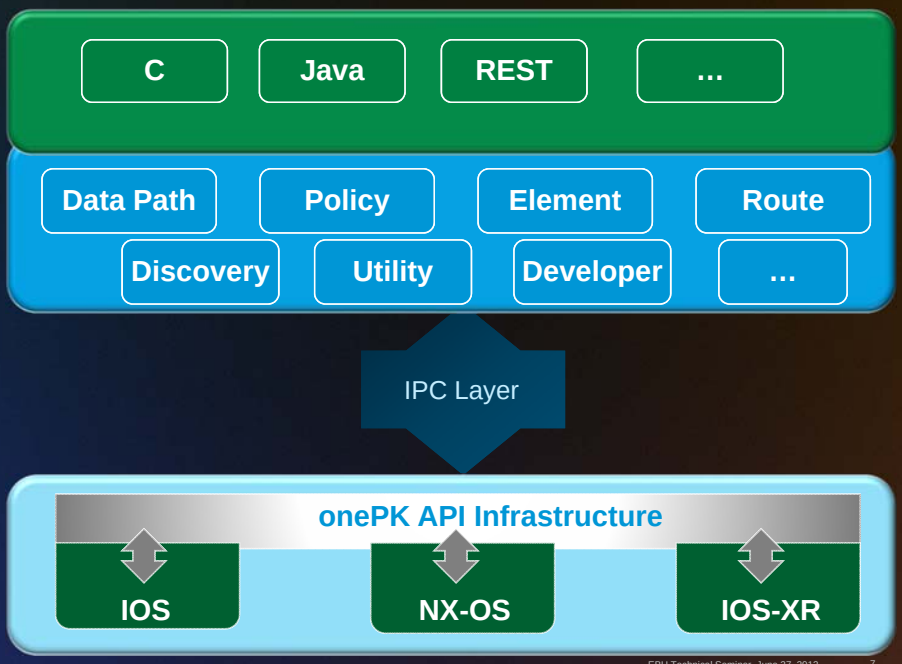
Developer Environment
- Language of Choice
- Programmatic Interfaces
- Comprehensive Data Delivery via APIs

Comprehensive Service Sets
Better Apps; New Services
Monetization Opportunity

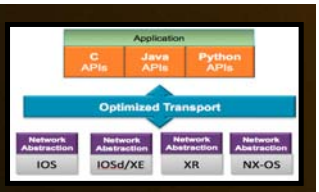
onePK

Deploy:
- On a Service Blade
- On an External Server
- Directly on the Device

Network Data
Control, Extend, Scale



The Layer Developers "See"

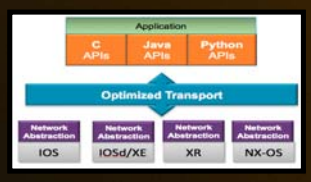


Presentation Layer

- Provides multiple language bindings so you can choose the language that suits your requirements
- Flexible and extensible
- Provided as a set of libraries in the Kit

```
[fedora~]$ ls /opt/cisco/onep-sdk/java/lib
libonep_core.jar  libthrift.jar  slf4j-api-1.6.1.jar  slf4j-simple-1.6.1.jar
[fedora~]$ ls /opt/cisco/onep-sdk/c/lib
libonep32_core.so  libonep32_datapath.so libonep32_pathtrace.so
```

Network Abstractions Enable Cross-OS Consistency



Invisible But Important

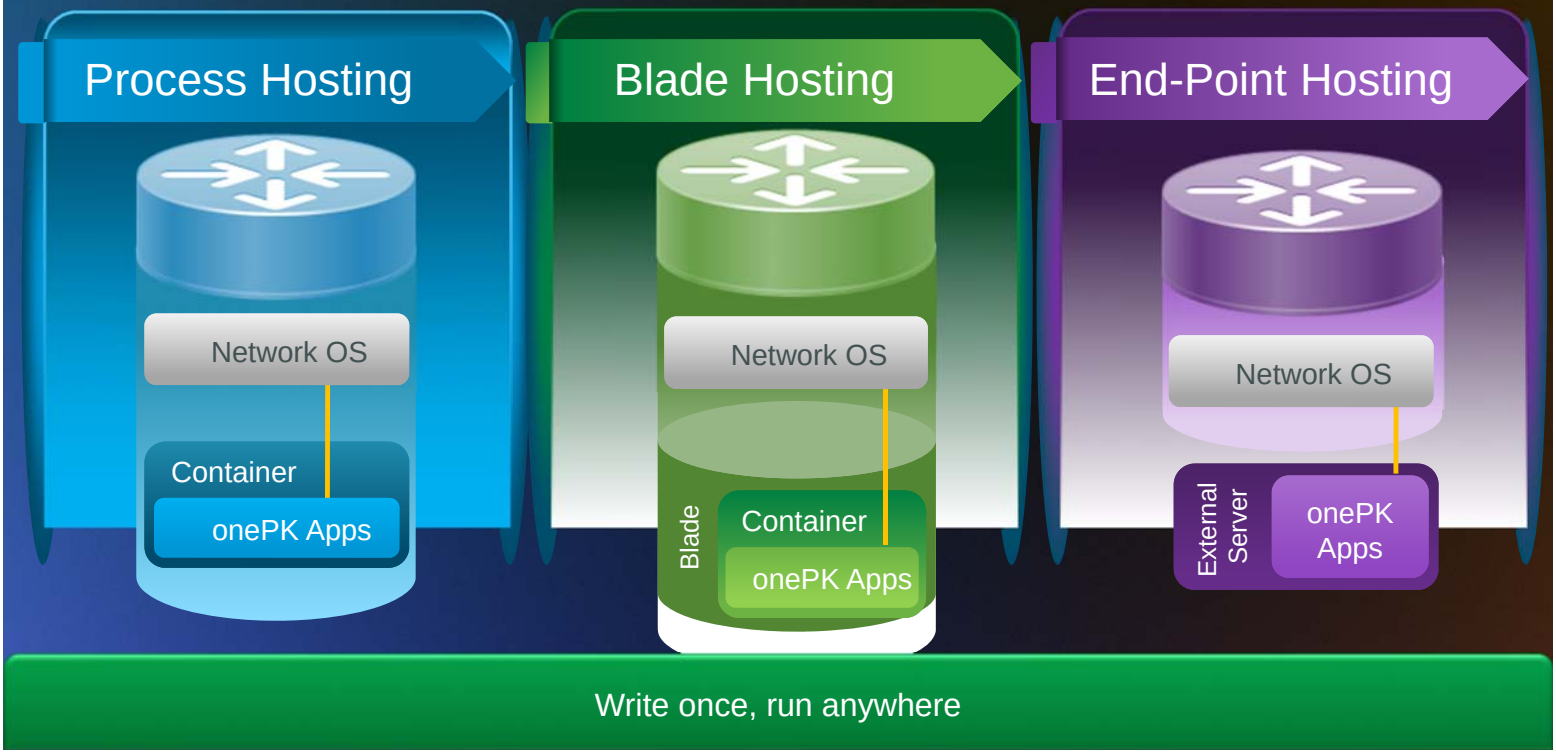
Network Abstraction Layer

Provides consistent interface to internal functions

Abstracts away underlying differences between OS's

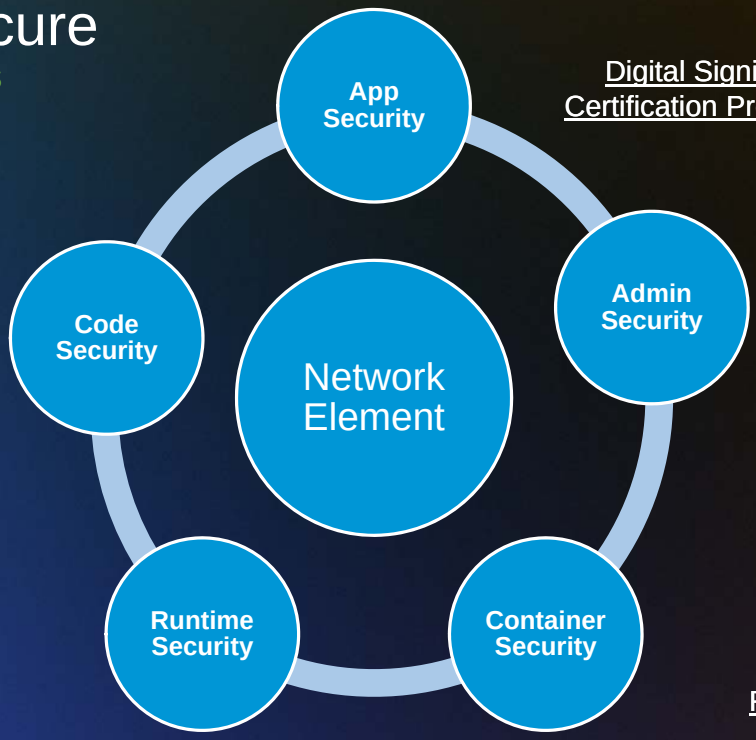
Network Abstraction	Network Abstraction	Network Abstraction	Network Abstraction
IOS	IOSd/XE	XR	NX-OS

APIs – Application Hosting Options



Yes, It's Secure

Security Five Ways



Digital Signing
Certification Process

CLI Control
Resource Allocation

Isolation
Resource Consumption

AAA
Encryption

Code Isolation
Strong Typing

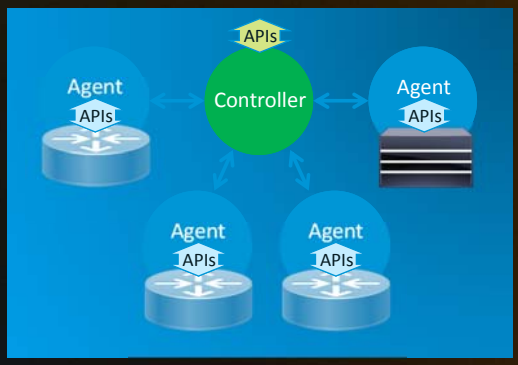
Base Service Sets Described

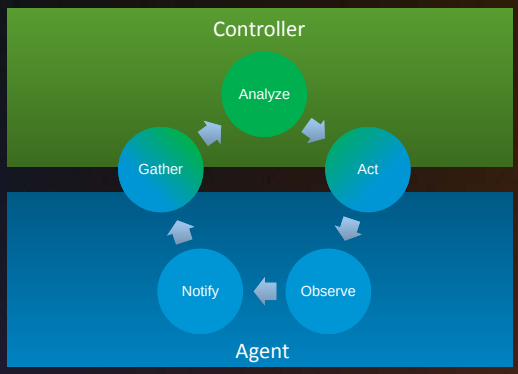
Service Set	Description
Data Path	Provides packet delivery service to application: Copy, Punt, Inject
Policy	Provides filtering (NBAR, ACL), classification (Class-maps, Policy-maps), actions (Marking, Policing, Queuing, Copy, Punt) and applying policies to interfaces on network elements
Routing	Read RIB routes, add/remove routes, receive RIB notifications
Element	Get element properties, CPU/memory statistics, network interfaces, element and interface events
Discovery	L3 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

Agents and Controllers

Consolidate State Across Multiple Network Elements

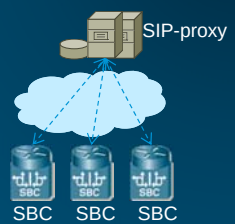
- Some network delivered functionality benefits from logically centralized coordination across multiple network devices
  - Functionality typically domain, task, or customer specific
  - Typically multiple Controller-Agent pairs are combined for a network solution
- Controller
  - Process on a device, interacting with a set of devices using a set of APIs or protocols
  - Offer a control interface/API
- Agent
  - Process or library on a device, leverages device APIs to deliver a task/domain specific function
- Controller-Agent Pairs offer APIs which integrate into the overall Network API suite



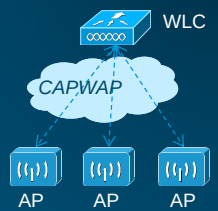


Agents and Controllers – Task Specific Sets

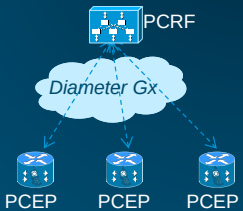
Session Border Control



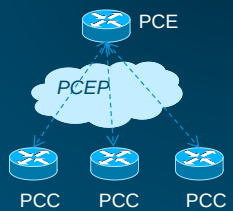
Wireless LAN Control



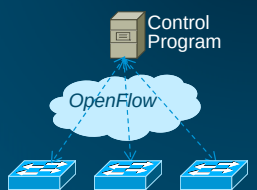
Policy Control



Path Computation



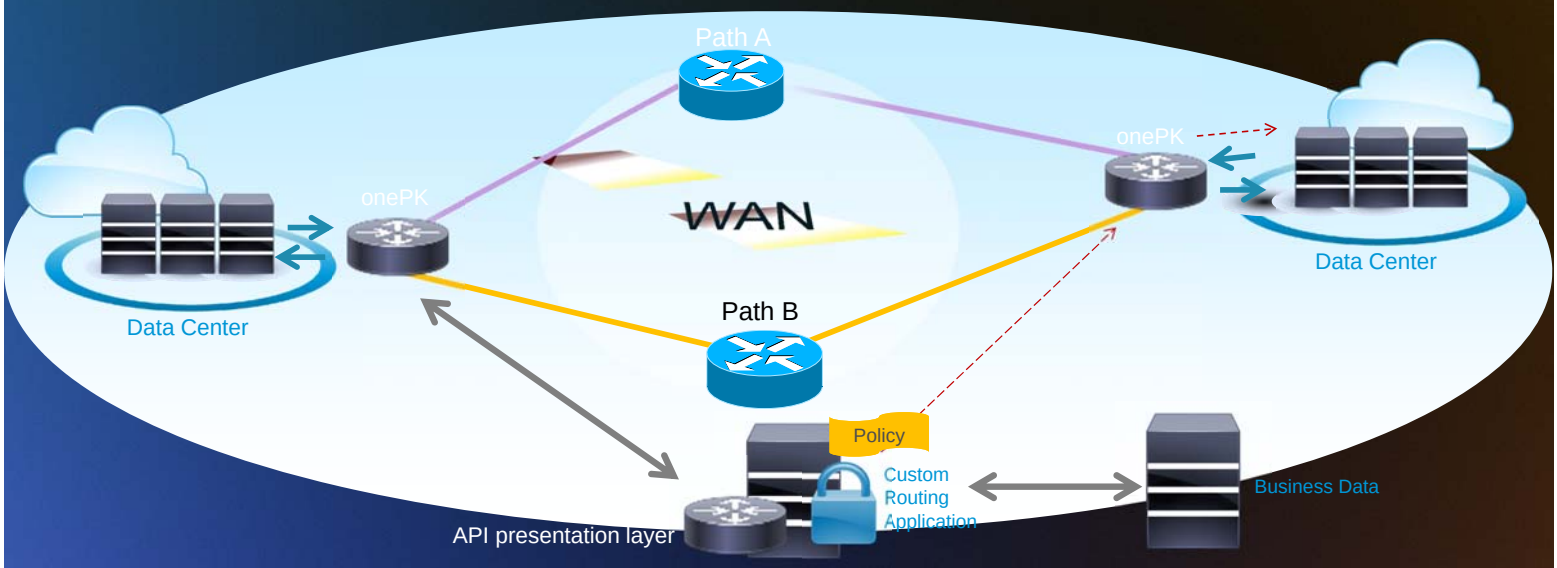
Flow Control with SDN



- Networking already leverages a great breath of Agents and Controllers
  - Current Agent-Controller pairs always serve a specific task (or set of tasks) in a specific domain
- System Design: Trade-off between Agent-Controller and Fully Distributed Control
  - Control loop requirements differ per function/service and deployment domain
  - "As loose as possible, as tight as needed"
  - Latency, Scalability, Robustness, Consistency, Availability

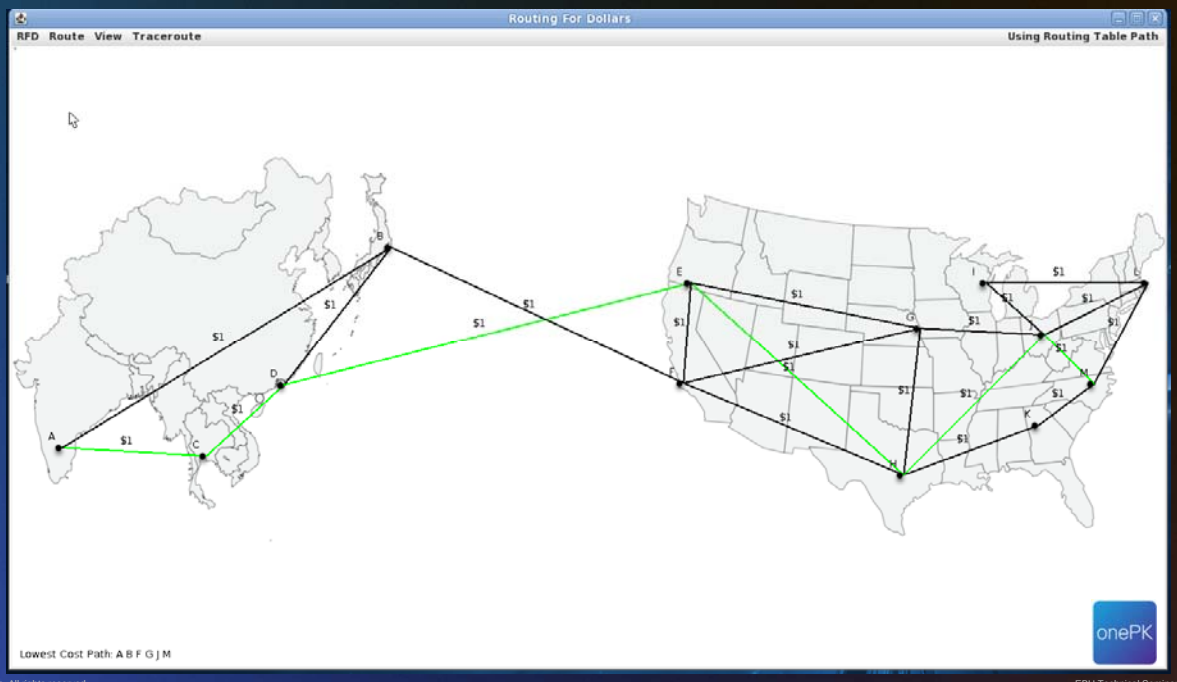
Example: Custom Routing

Traffic Forwarding Based on a Custom Algorithm collaborating with ntetrok



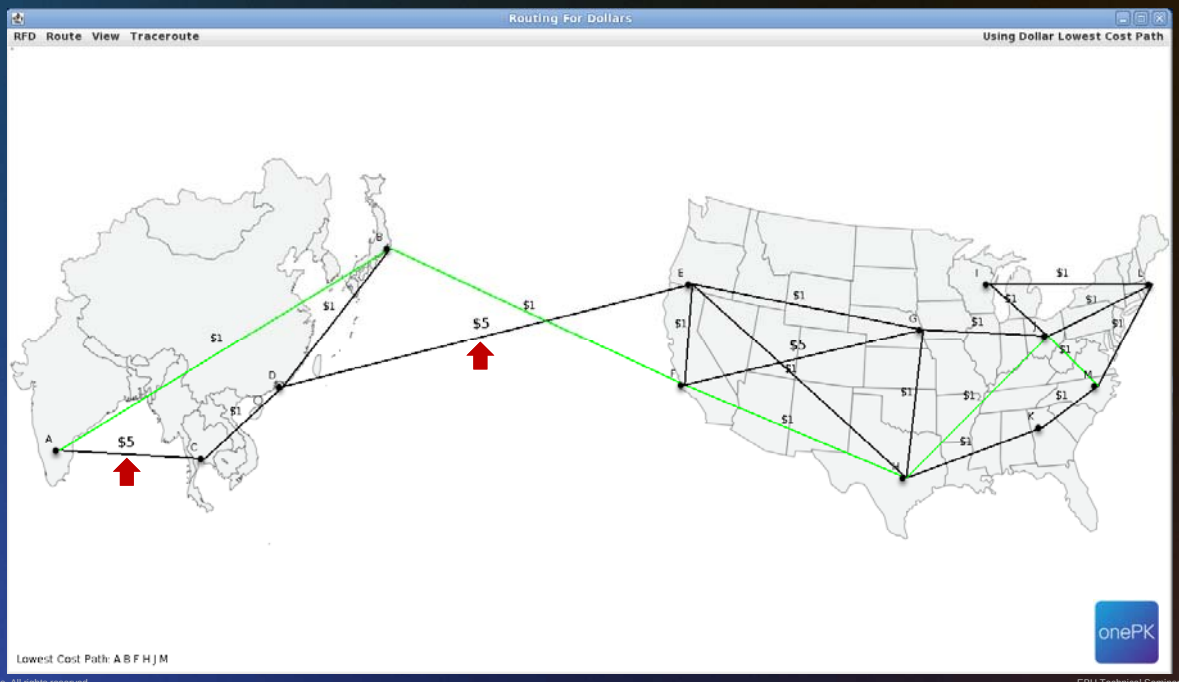
Custom Routing

Initial Setup: Default routing using EIGRP



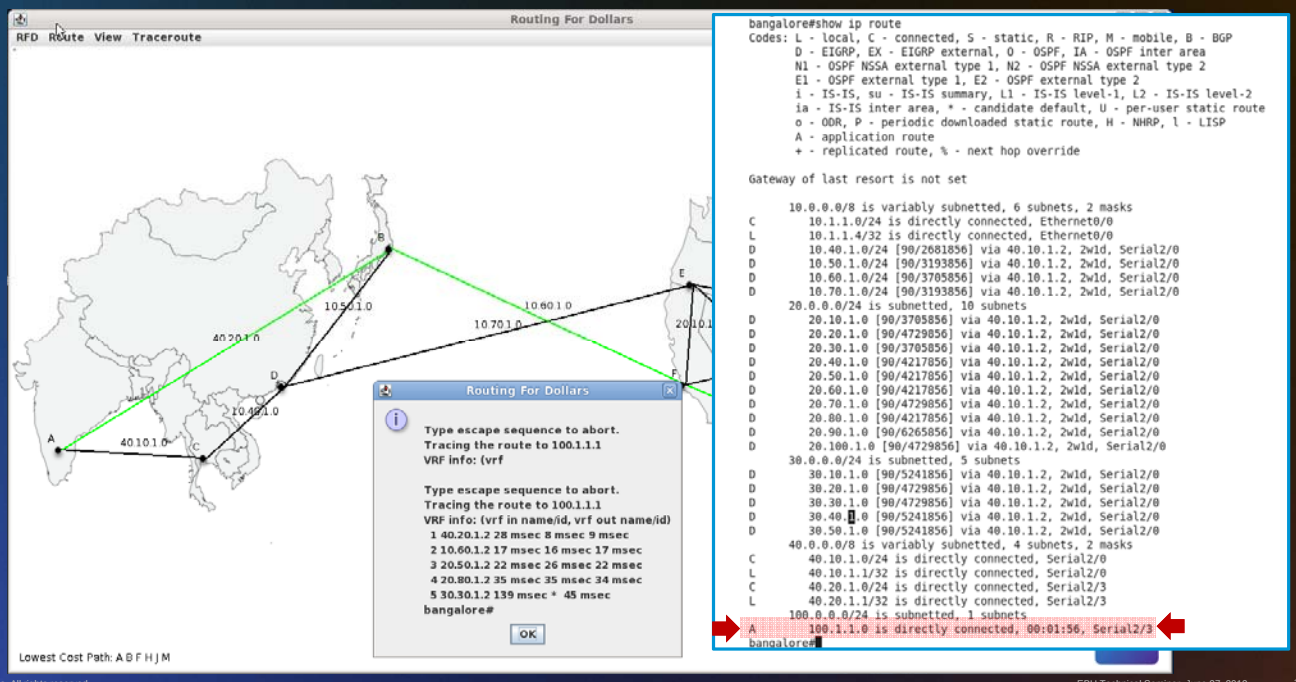
Custom Routing

Routing for Dollars: Application driven routes installed in network



Custom Routing

Tracing the application installed route – using the developer and element services



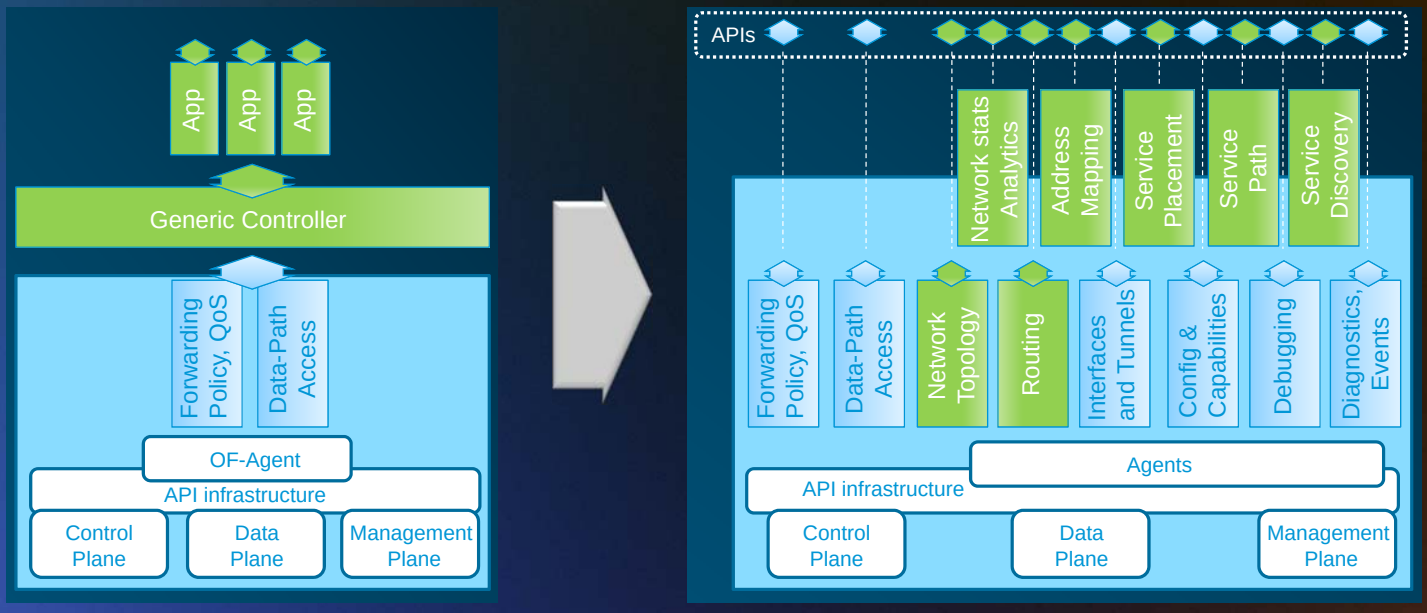
Open Network Environment

Towards a Broad Approach to Programmatic Access to the Network

- Evolutionary step for networking:
  Complement/evolve the Network Control Plane where needed
- Centered around delivering open, programmable environment for real-world use cases
  - No one-size-fits-all
  - APIs, Network Virtualization, Agents/Controllers
  - Joint evolution with industry and academia
- Technology-agnostic
  - Not predicated on a particular technology or standard
  - Draw from existing technologies and industry standards
- Delivered as incremental functionality
  - Many customers will use hybrid implementations
  - Build upon existing infrastructure with investment protection

Open Network Environment www.cisco.com/go/one onePK www.cisco.com/go/onepk; www.cisco.com/go/getyourbuildon

Evolve the early SDN Model…
… acknowledge the need for diverse abstractions



Thank you.