Software Defined Networking is an Architecture *Not* a Protocol

David Meyer
World Telecommunications Congress 2012
March 04 – 07, 2012
Miyazaki, Japan
dmm@cisco.com

Agenda

- Software Defined Networking (SDN)
  - A Bit of History and How we got here
  - OpenFlow and SDN
- Current SDN Thinking
- Where we're going: *The Programmable Network*
- Q&A

In The Beginning
(Martin, Nick, and a cast of many)

Ethane: Addressing the Protection Problem in Enterprise Networks

Martin Casado
Michael Freedman
Glen Gibb
Lew Glendenning
Dan Boneh
Nick McKeown
Scott Shenker
Gregory Watson

Presented By: Martin Casado
PhD Student in Computer Science,
Stanford University

casado@cs.stanford.edu
http://www.stanford.edu/~casado

A Little Later…OpenFlow
(again, with a cast of 2^10s)

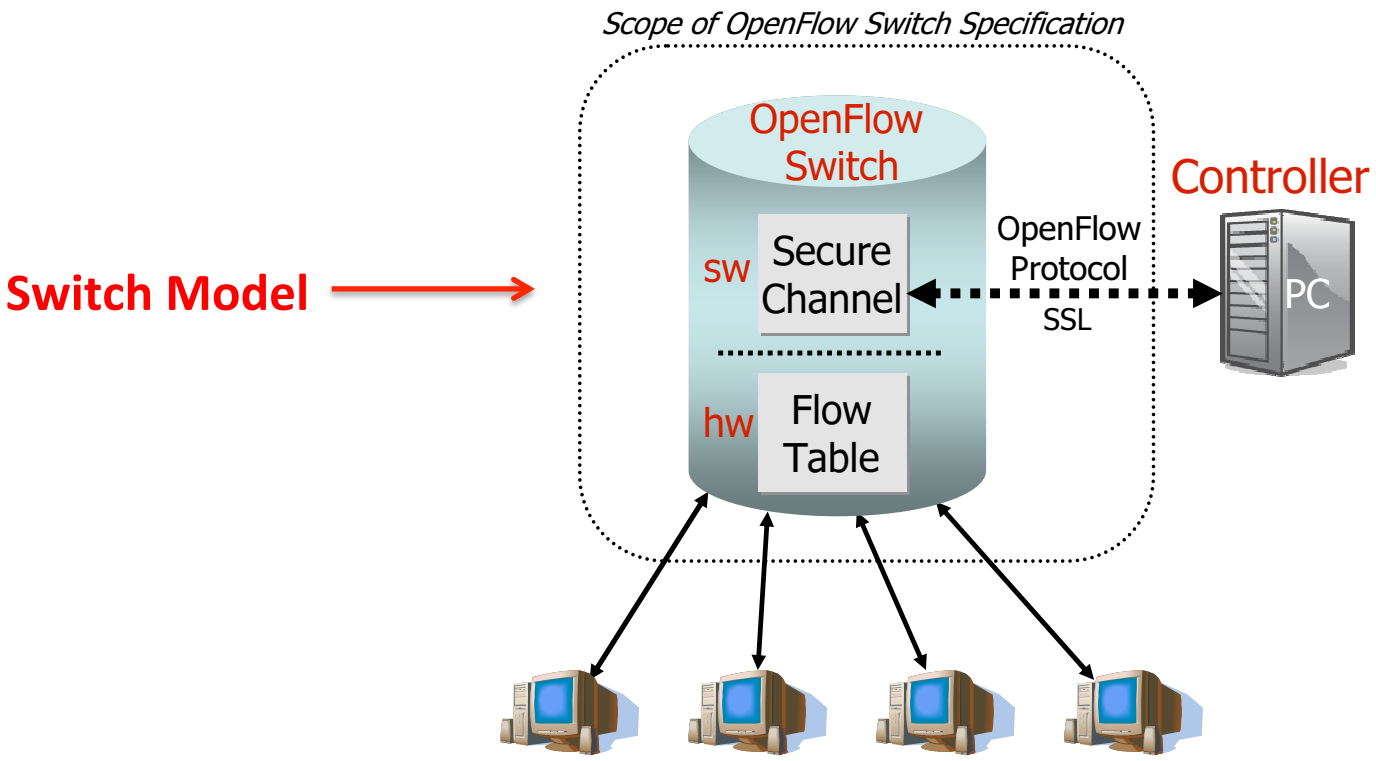


Figure 1: Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel.

OpenFlow Switch, v 1.0

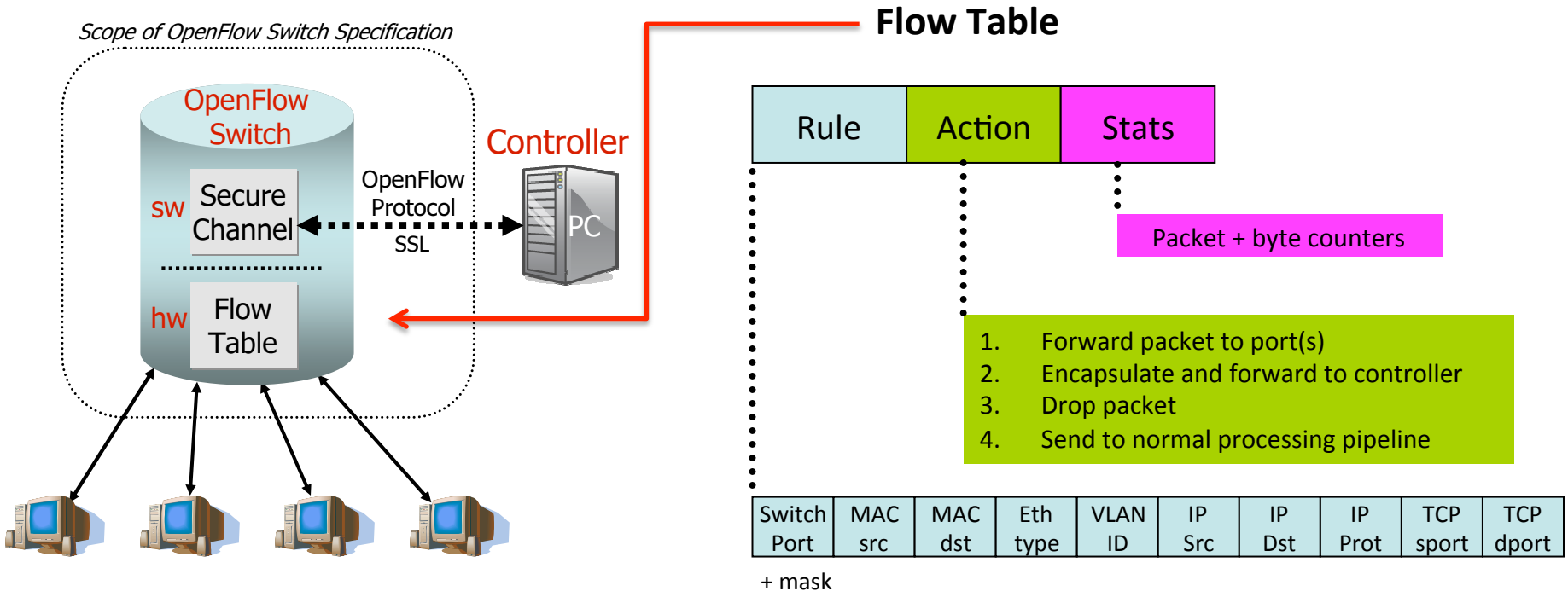


Figure 1: Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel.

This was one of the earliest SDN architectures to use OpenFlow (note flow based, separation of control and data planes)

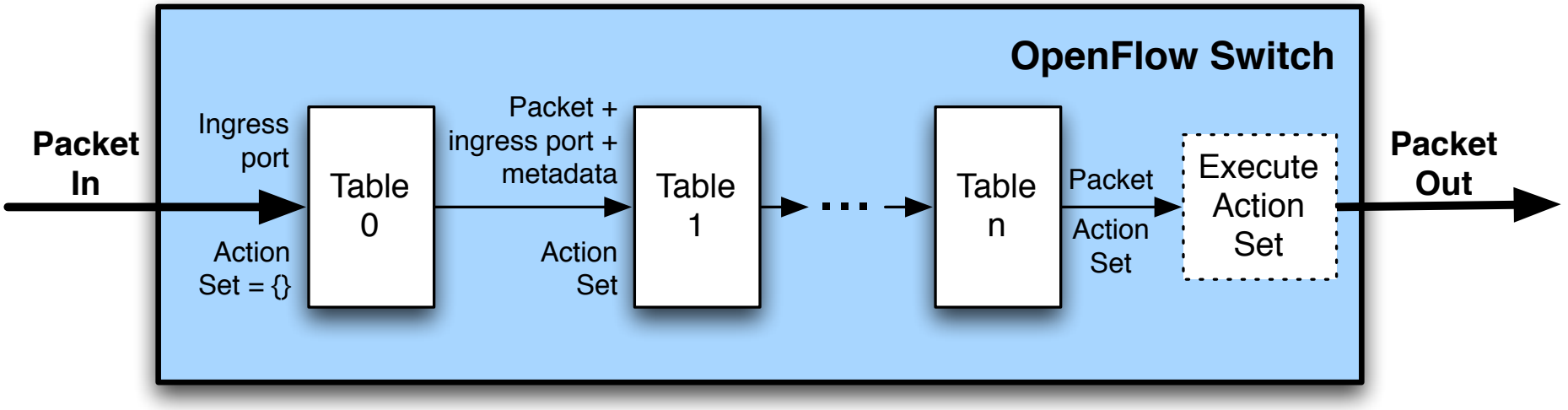
Header Fields for Matching (v.1.1)

Ingress Port	Metadata	Ether src	Ether dst	Ether type	VLAN id	VLAN priority	MPLS label	MPLS traffic class	IPv4 src	IPv4 dst	IPv4 proto / ARP opcode	IPv4 ToS bits	TCP/ UDP / SCTP src port ICMP Type	TCP/ UDP / SCTP dst port ICMP Code

Table 3: Fields from packets used to match against flow entries.

Note that by matching across the full header space, OpenFlow effectively "de-layers" the protocol stack

OpenFlow Version 1.X, X > 0



(a) Packets are matched against multiple tables in the pipeline

{Any,Multi}cast	(1.1)
ECMP	(1.1)
MPLS	(1.1, note push/pop, .1q)
IPv6	(1.2)

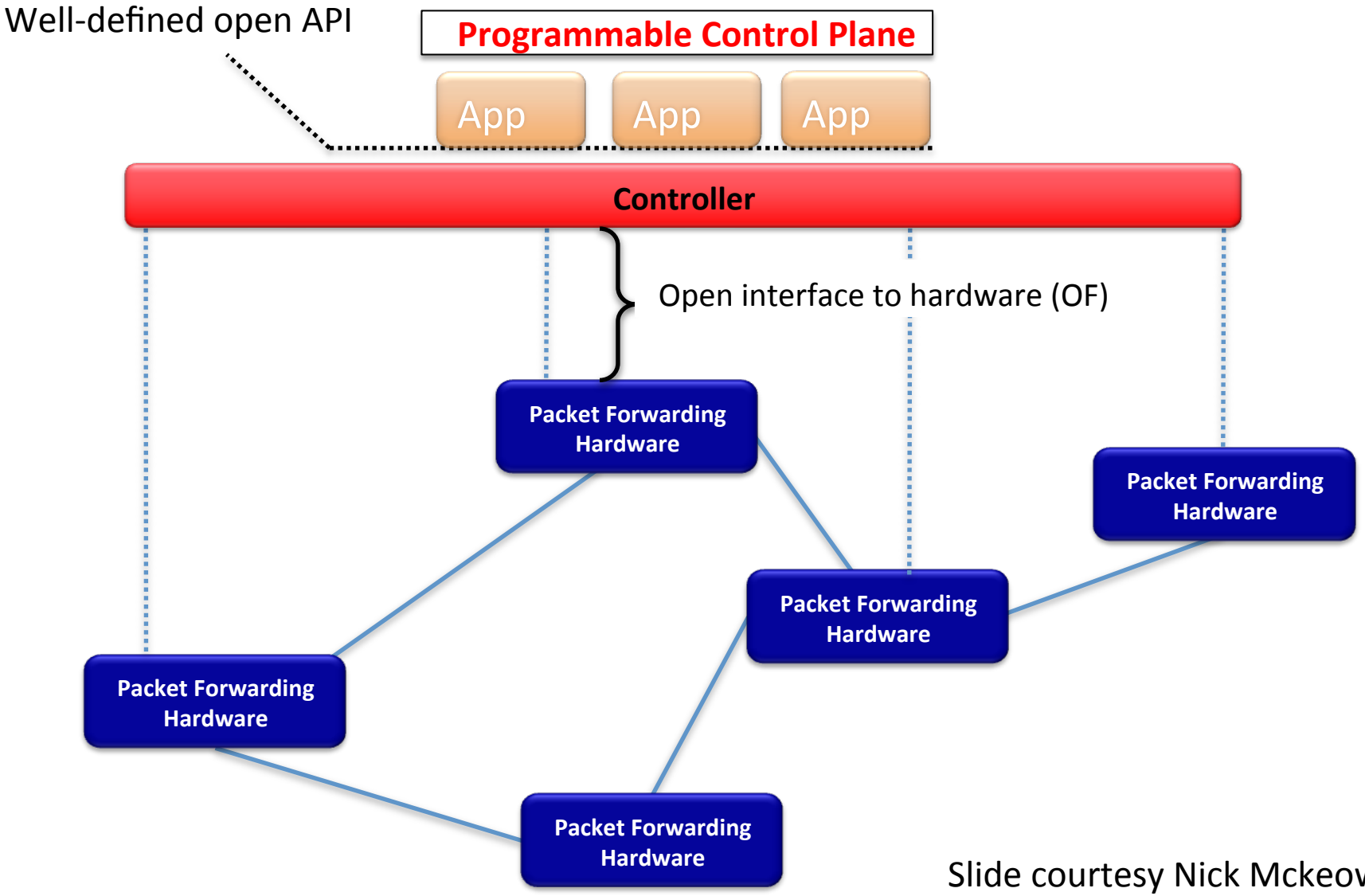
1.3 features being currently being considered
-- tunnels, tags {en,de}caps, meters,…
Configuration Protocol under co-development

So What Is OpenFlow Then?

- ***A Switch Model***
  - Match-Action Tables (MAT)
  - Per-flow counters
- ***An Application Layer Protocol***
  - Binary wire protocol, messages and state machine that allow programming of the MAT
- ***A Transport Protocol***
  - TLS, TCP, ..

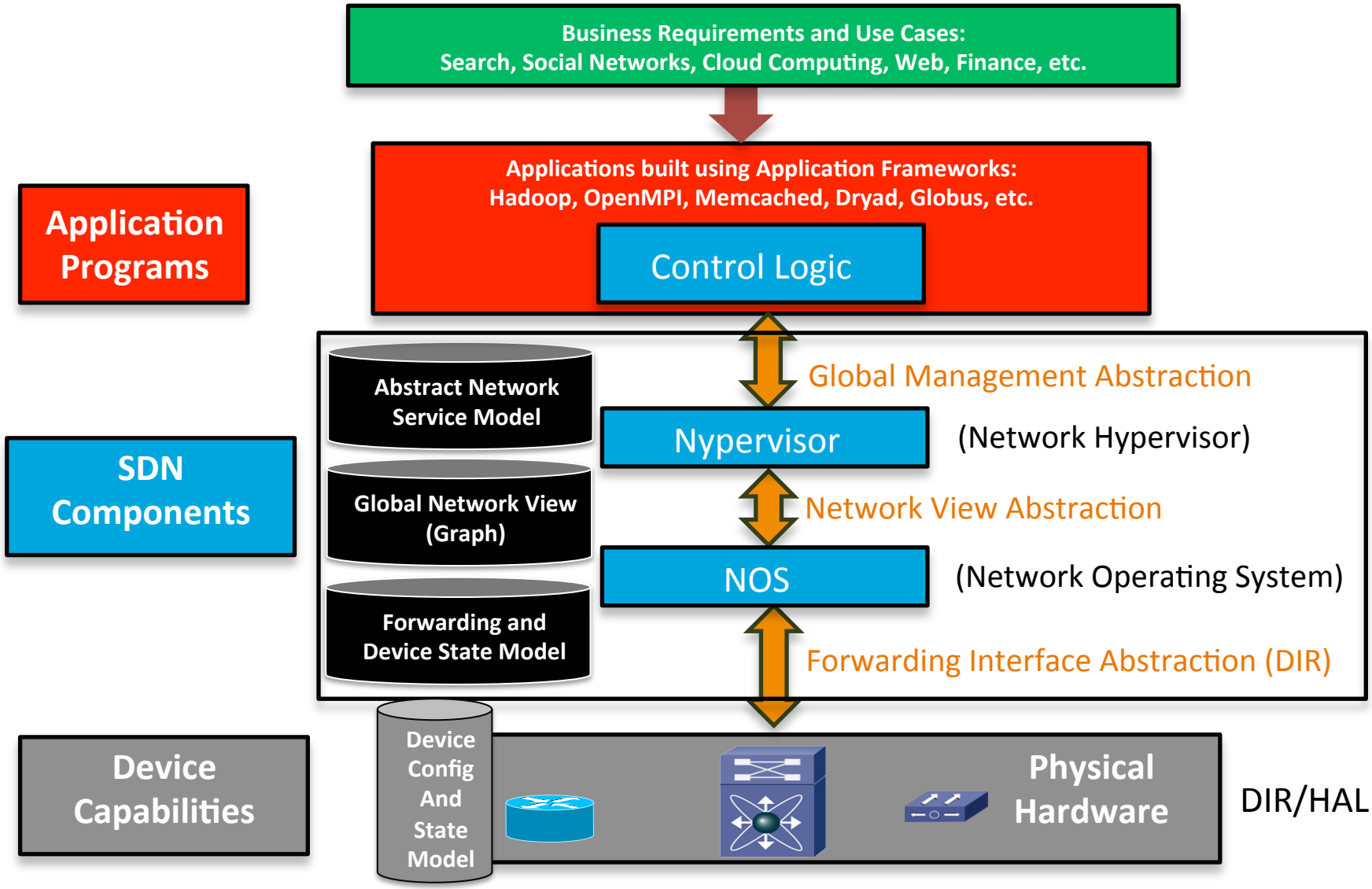
- Note that OF says nothing said about how a given forwarding target implements the switch model → ***OF is an Abstract Switch Model***

Early SDN Architecture

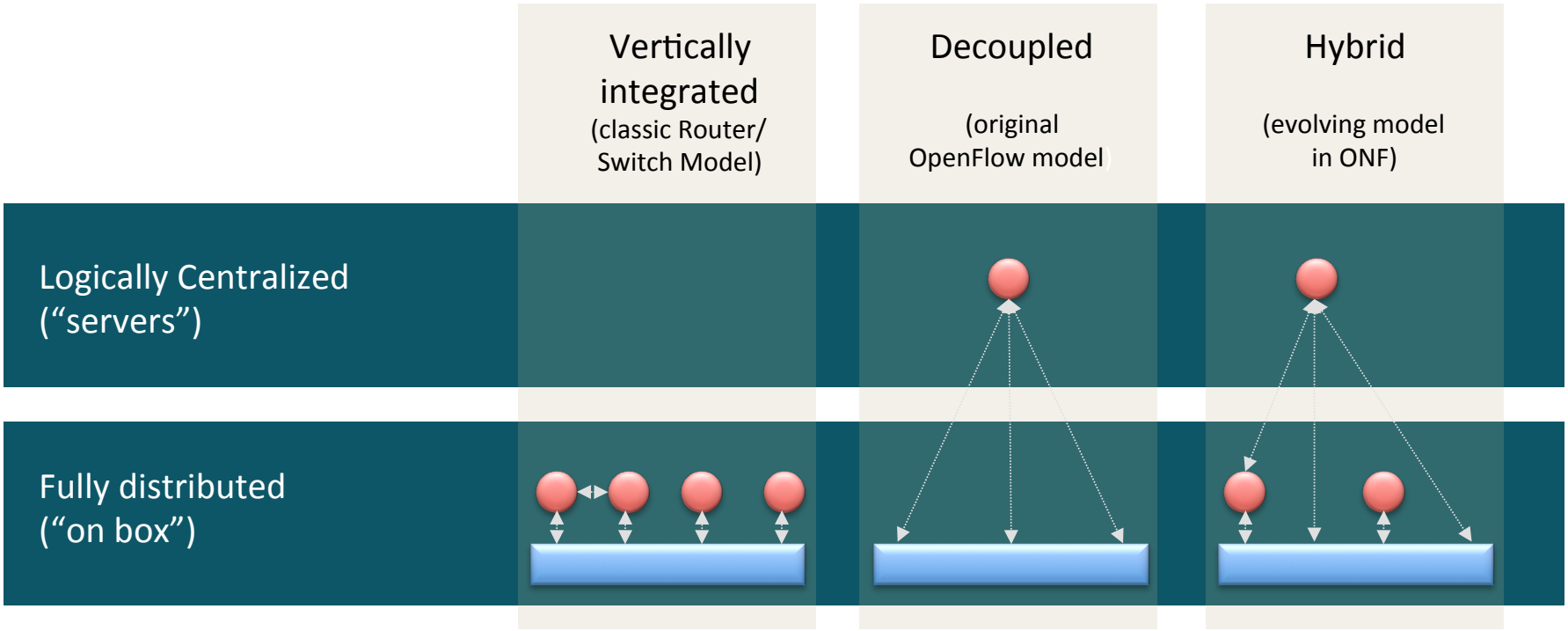


Slide courtesy Nick Mckeown

“Modern” OF/SDN Architecture



Cut Another Way: Control Plane Distribution Options



Data Path jointly controlled by standard on-box control plane and centralized off-box controller

Slide courtesy Frank Brockners

Legend: Data plane Control plane function

Nothing New Under The Sun…

- Much of the motivation for *this generation's foray into SDN* was grounded in the research community's desire to be able to experiment with new control paradigms.

- I call it "this generation's foray" because the basic idea, ***separation of control and data planes***, is not new. Examples include:
  - Ipsilon Flow Switching
    - Centralized flow based control, ATM link layer
    - GSMP (RFC 3292)
  - AT&T SDN
    - Centralized control and provisioning of SDH/TDM networks
  - A similar thing happened in TDM voice to VOIP transition
    - Softswitch → Controller
    - Media gateway → Switch
    - H.248 → Device interface
  - ForCES
    - Separation of control and data planes
    - RFC 3746 (and many others)

Great Diversity In Thinking/ Implementation

- **Research**
  - RYU
    - NTT
    - http://www.osrg.net/ryu
  - NOX
    - A first generation open source controller
    - http://noxrepo.org/doc/nox-ccr-final.pdf
  - POX
    - "NOX in python"
  - Trema
    - Ruby/C controller
    - http://trema.github.com/trema/
  - Floodlight
    - Java based OF controller
    - http://floodlight.openflowhub.org/
  - SNAC
    - Simple Network Access Control
    - http://www.openflow.org/wp/snac/
  - Flowvisor
    - A open source "slicing" controller
    - http://www.openflow.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf
  - Maestro/Beacon
    - Java controllers optimized for multi-core CPUs
    - http://www.cs.rice.edu/~eugeneng/papers/TR10-11.pdf
  - ONIX
    - A second generation infrastructure controller
    - https://www.usenix.org/events/osdi10/tech/full_papers/Koponen.pdf
- **Commercial**
  - Google, NEC, Broadcom, Bigswitch, Nicira, Pica8, …
    - Nicira and Bigswitch recently received new funding
    - Ericsson, Google and Nicira have implemented and deployed ONIX
    - Bigswitch and Pica8 claim that they will open source their controllers (beacon)

Another Way to Think About SDN

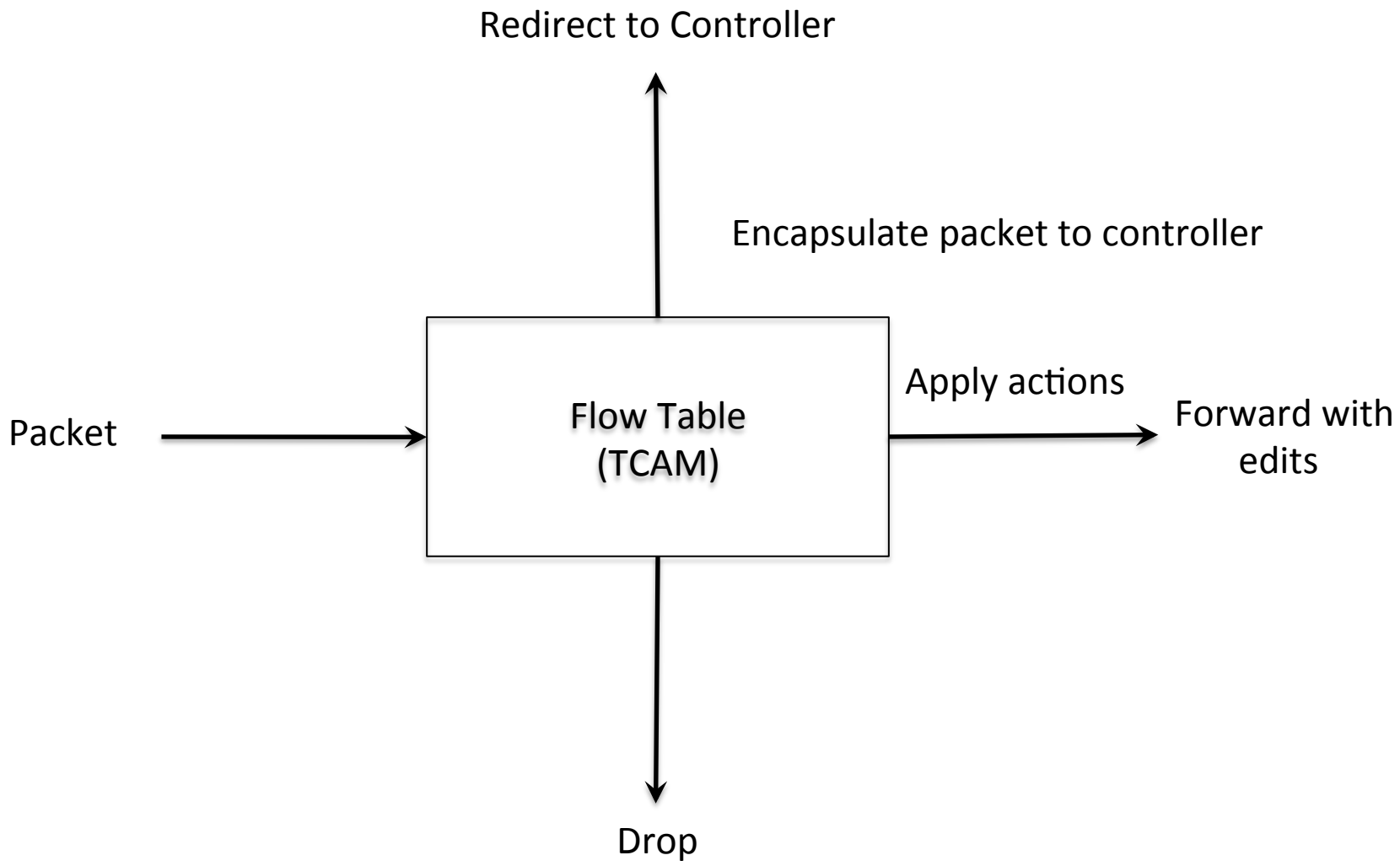
- **Architecture*:*** *Generalized Programmable Network (GPN)*
  - The term "SDN" already too overloaded,….
  - Encompass existing and future control planes
    - And associated features
    - "Hybrid Switch" modes
  - Standardize a diverse set of APIs in addition to OF
    - Such APIs talk to both existing control and data planes

- **Objective:** Enable tighter interaction between applications and the network
  - Inform network of desired behavior
  - Inform application of data intrinsically available in the network
    - Data mining, telemetry, NPS, …
  - Provide greater agility, flexibility, and feature velocity

- **Approach:** Define two broad classes of APIs
  - Network APIs
  - Network Element APIs

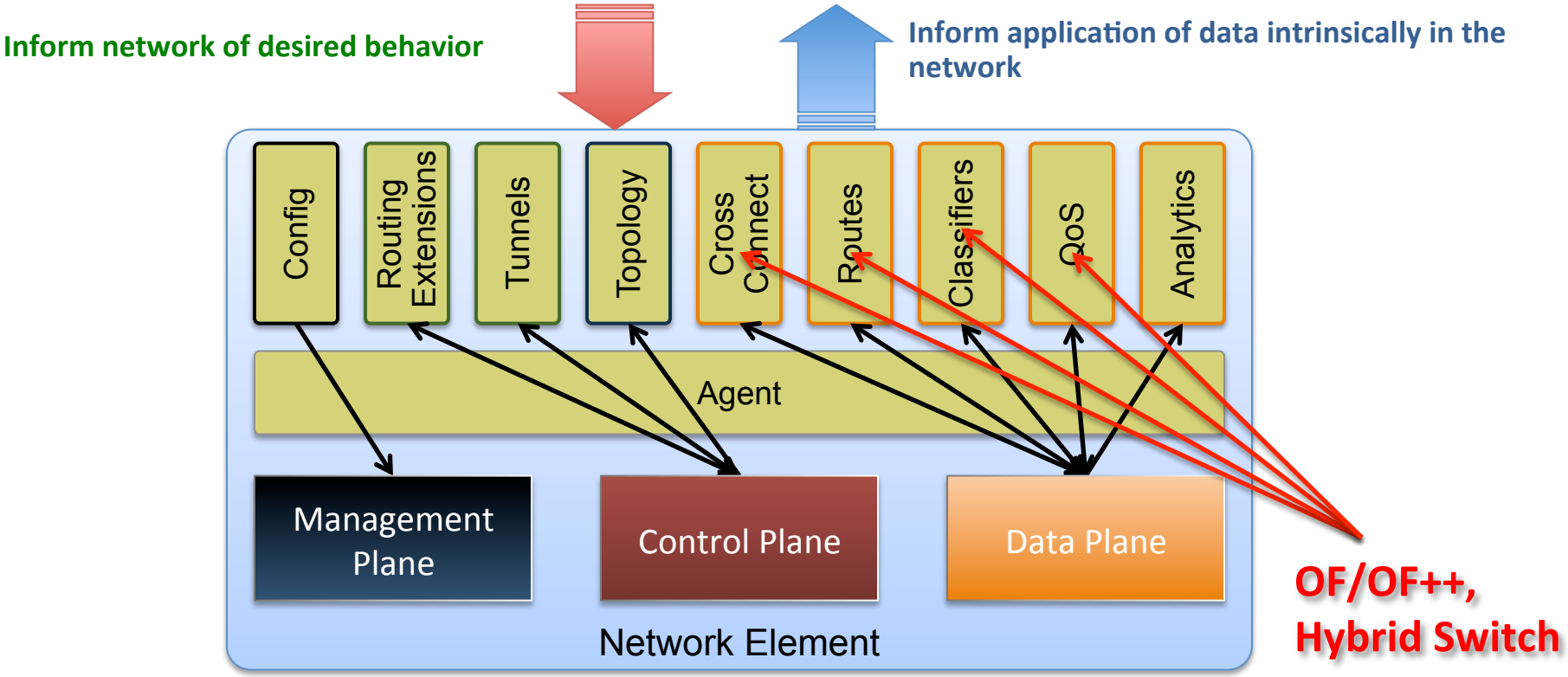
- Also need *a Generalized Switch Model*

Recall the OF 1.0 Switch Model



Need a richer switch model to deal with data and control planes and features

GPN Network Element (Agent Architecture/Hybrid Switch)



What is a Hybrid Switch?

- Abbreviated Hybrid Switch Problem Space
  - Make OF/SDN coexist with existing more general switch/network models
    - **Why is this hard again?**
- Proposed Models
  - *Ships in the Night* and *Integrated Mode*
  - *Integrated Mode*
    - Envision OF as one of many ***APIs*** we can use to build network probramability
- Hybrid Switch WG recently chartered by ONF
  - Jan Medved of Cisco is the Chair
  - Possibly related: "SDNP" activity in IETF
    - But note no "SDNP BOF" at upcoming Paris IETF

A Couple Of Hybrid Switch Use Cases

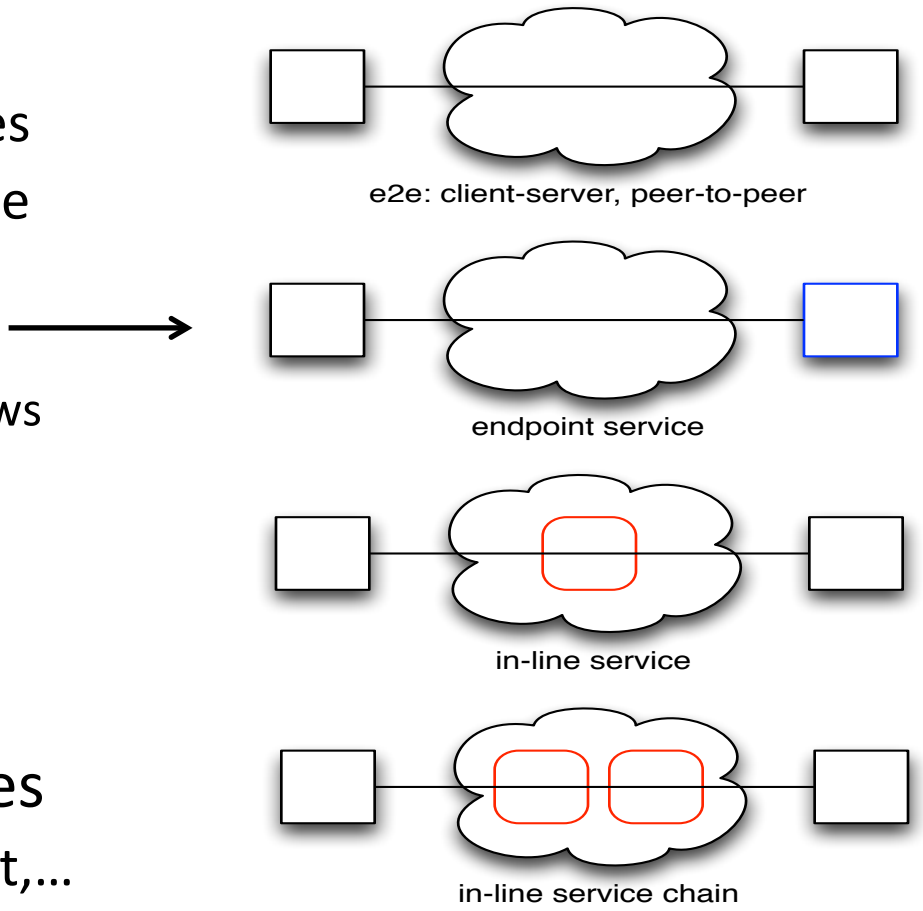
- ***Installing ephemeral routes*** in the RIB
  - Install routes in RIB subject to admin distance or …
  - Moral equivalent of static routes, but dynamic
  - May require changes to the OF protocol/model

- ***Edge classification***
  - Basically use the OF as an API used to install ***ephemeral*** classifiers ***at the edge***
  - Moral equivalent of … 'ip set next-hop <addr>' (PBR)
  - Use case: Service Engineered Paths
    - Program switch edge classifiers to select set of {MPLS, GRE, …} tunnels
    - Core remains the same

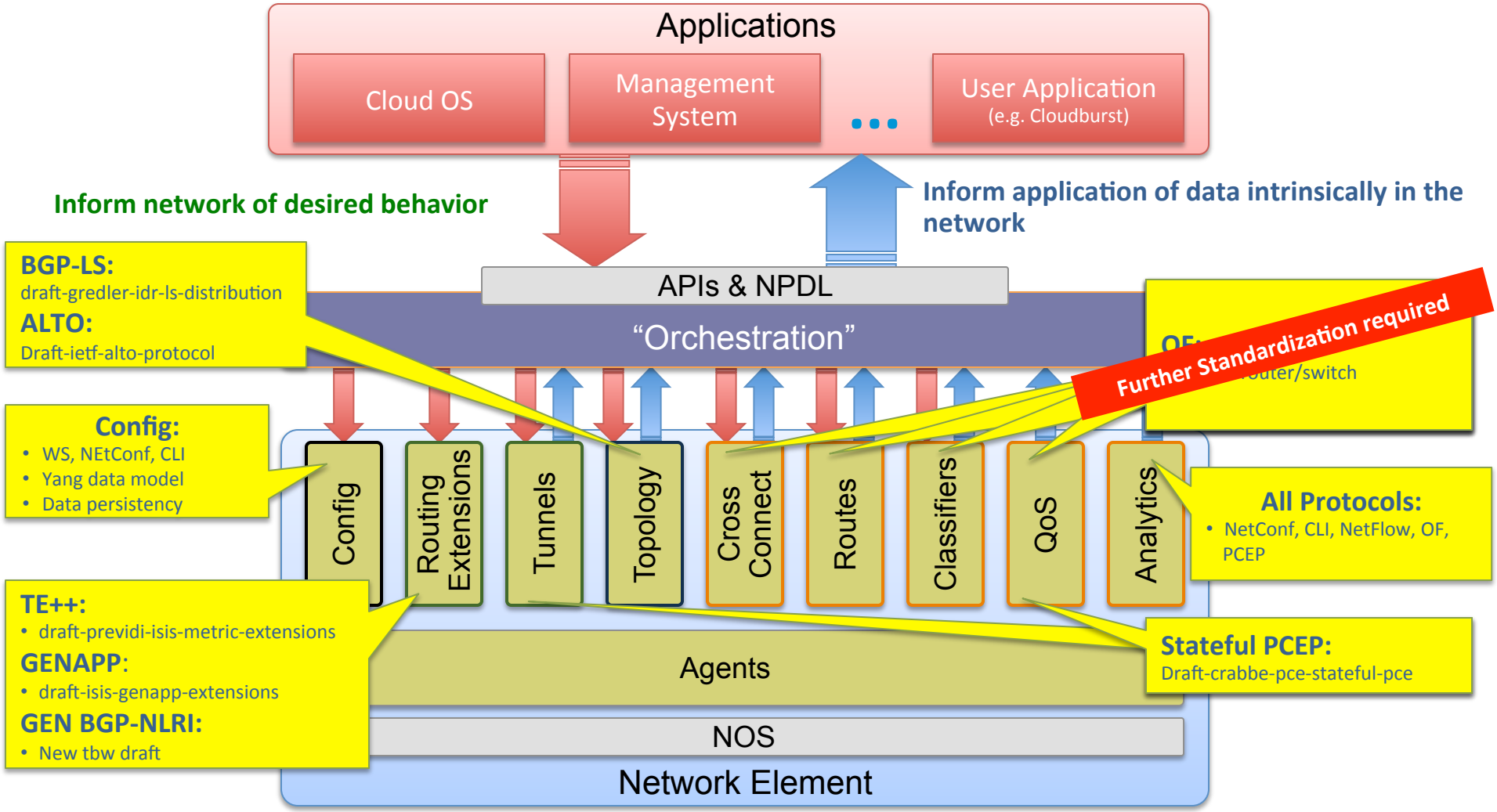
- ***Service Chaining***
  - Let's talk a bit more about kinds of service chains…

Programmable Service Chains

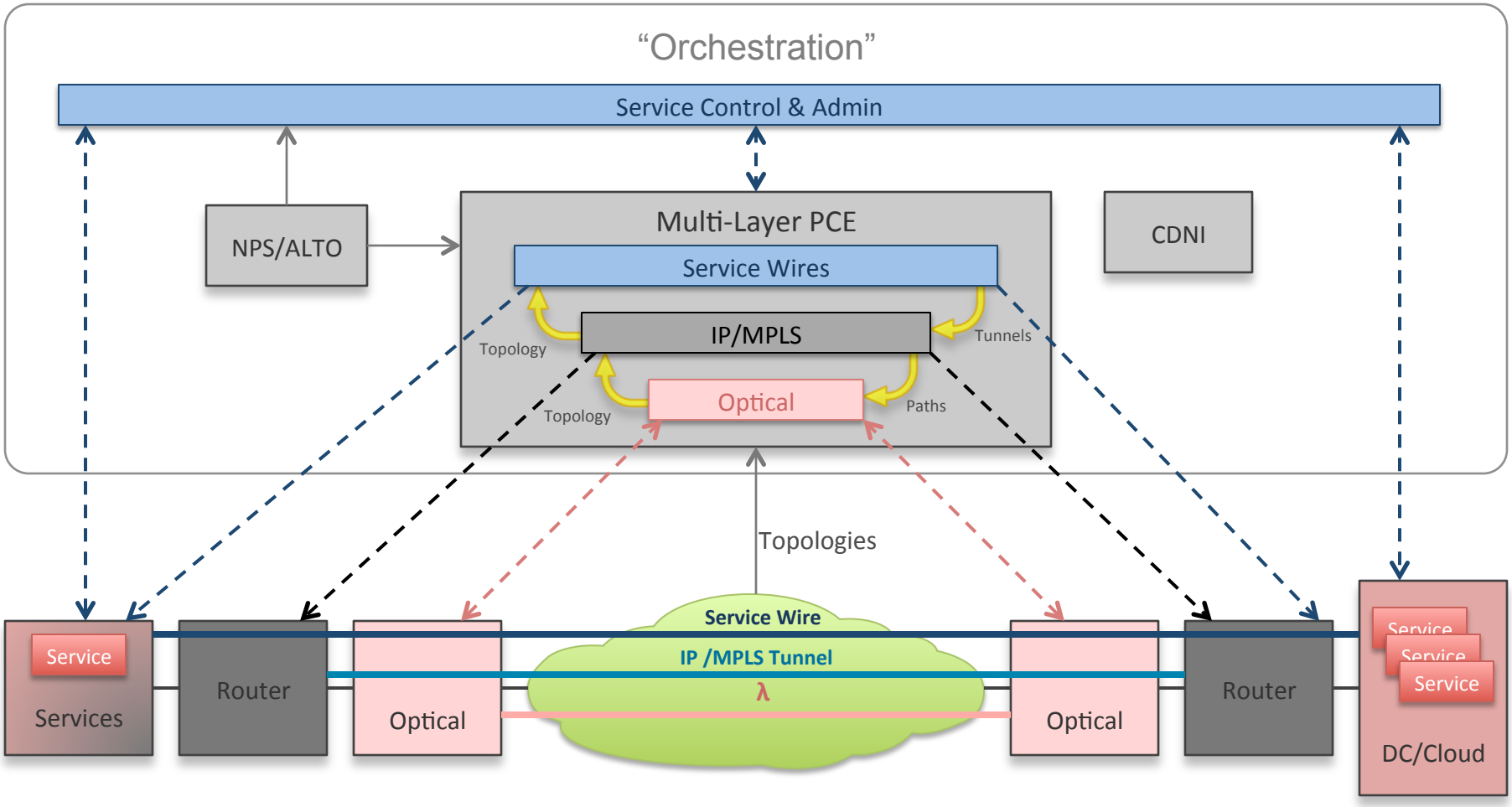
- Basic Use Cases
  - Endpoints vs. In-line services
  - Composite Services / Service Chaining
  - Flow Routing
    - Fine vs. Coarse Grained Flows
    - Filtering vs. Routing
    - Placement vs. Topology
    - Addressing vs. Flows
- Future/Unknown Use Cases
  - CDN, NDN, Optical xconnect,…



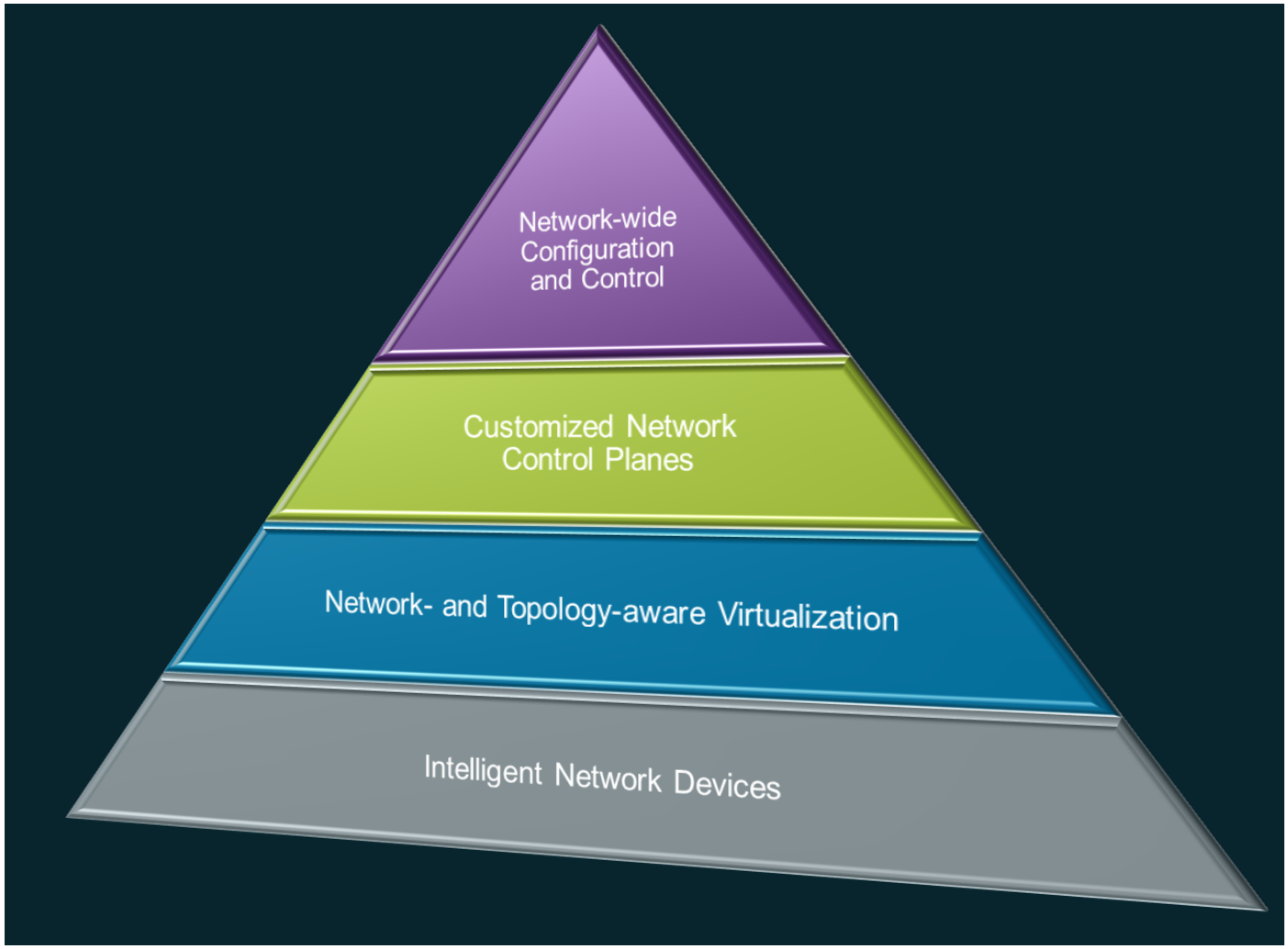
Programmable Network Architecture



Service Provider Network Model



So where this is all going?
The Programmable Network



"Orchestration, Configuration, Control"

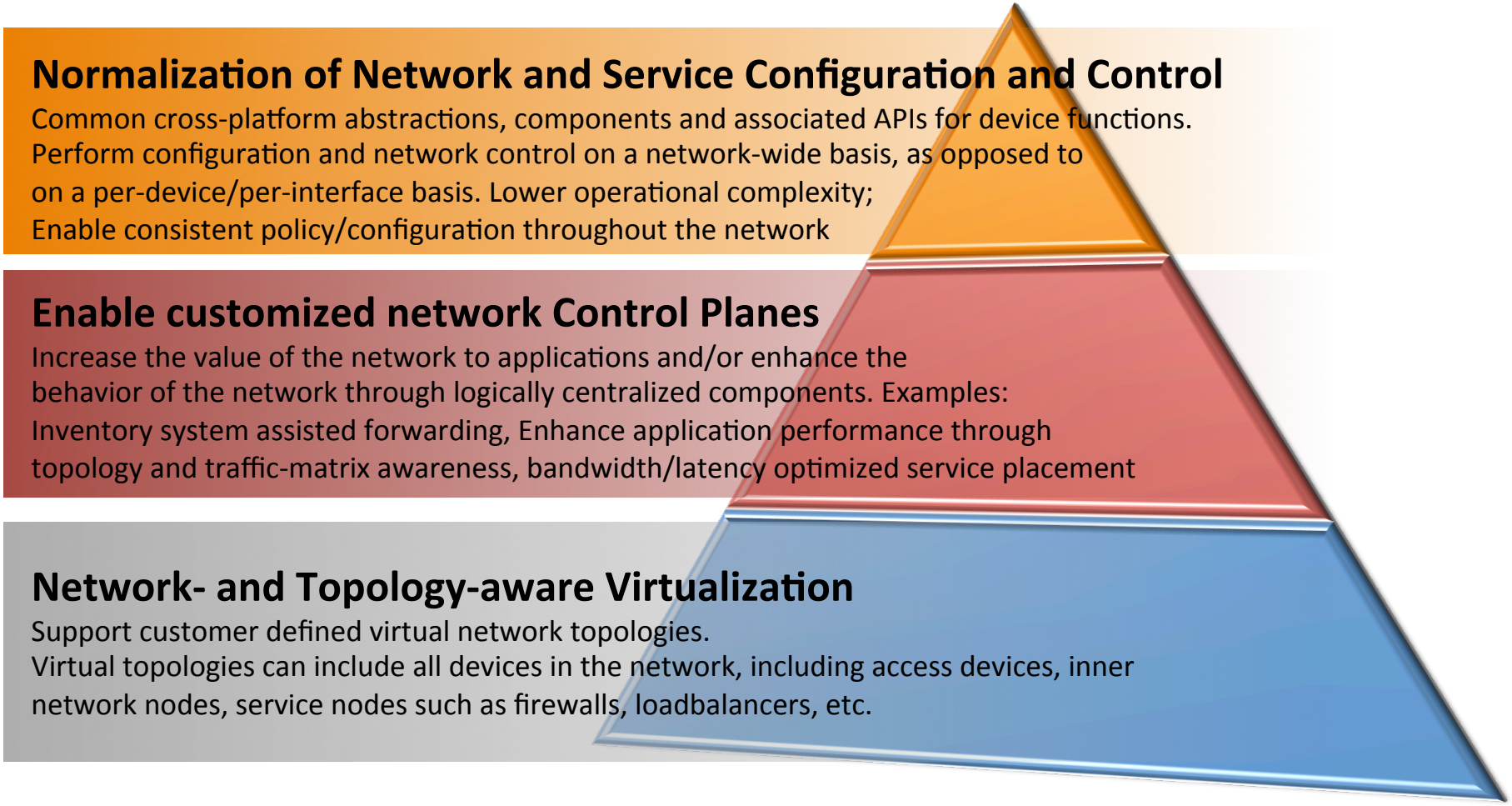
"Programming, Customization"

"Network Hypervisor"

"Efficient Forwarding"

Slide courtesy Frank Brockners

“Network Programmability” - Key Value Propositions



Note that the SDN value proposition differs from the “OpenFlow” value proposition. OpenFlow’s is focusing on **per-device** level programmability; i.e. creating a (potentially standard) interface to control the forwarding of flows in the device.

Slide courtesy Frank Brockners

Putting It All Together

- **Network Programmability (as well as SDN) is About Abstractions**
  - SDN is a bigger concept that OpenFlow
  - Network Programmability is a bigger concept that SDN

- **Network Services Abstraction**
  - Global Topology Network View (physical, logical, virtual)
  - Network Positioning, Telemetry, Data Mining
  - Service Chaining/Pooling
  - Service Orchestration/Provisioning

- **Distributed Network Control Abstraction**
  - Network Operating System (NOS)

- **Forwarding abstraction**
  - OpenFlow
  - ACL/PBR
  - "openflow-future (Google 2.0 proposal)
  - RIB/Routing Interfaces

- **Current thinking envisions OpenFlow/SDN working in concert with existing control planes**

Q&A

Thanks!