Lecture 6: Data Center Network Virtualization

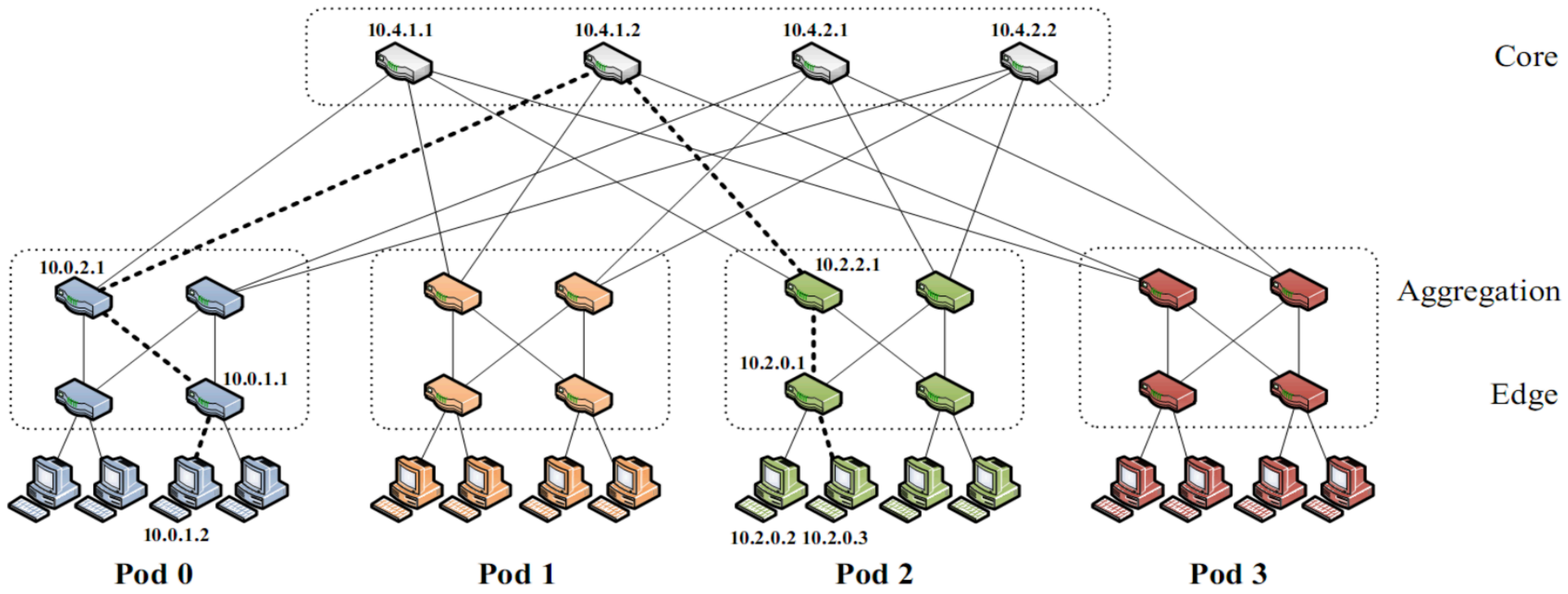
CS 234 / NetSys 210: Advanced Computer Networks

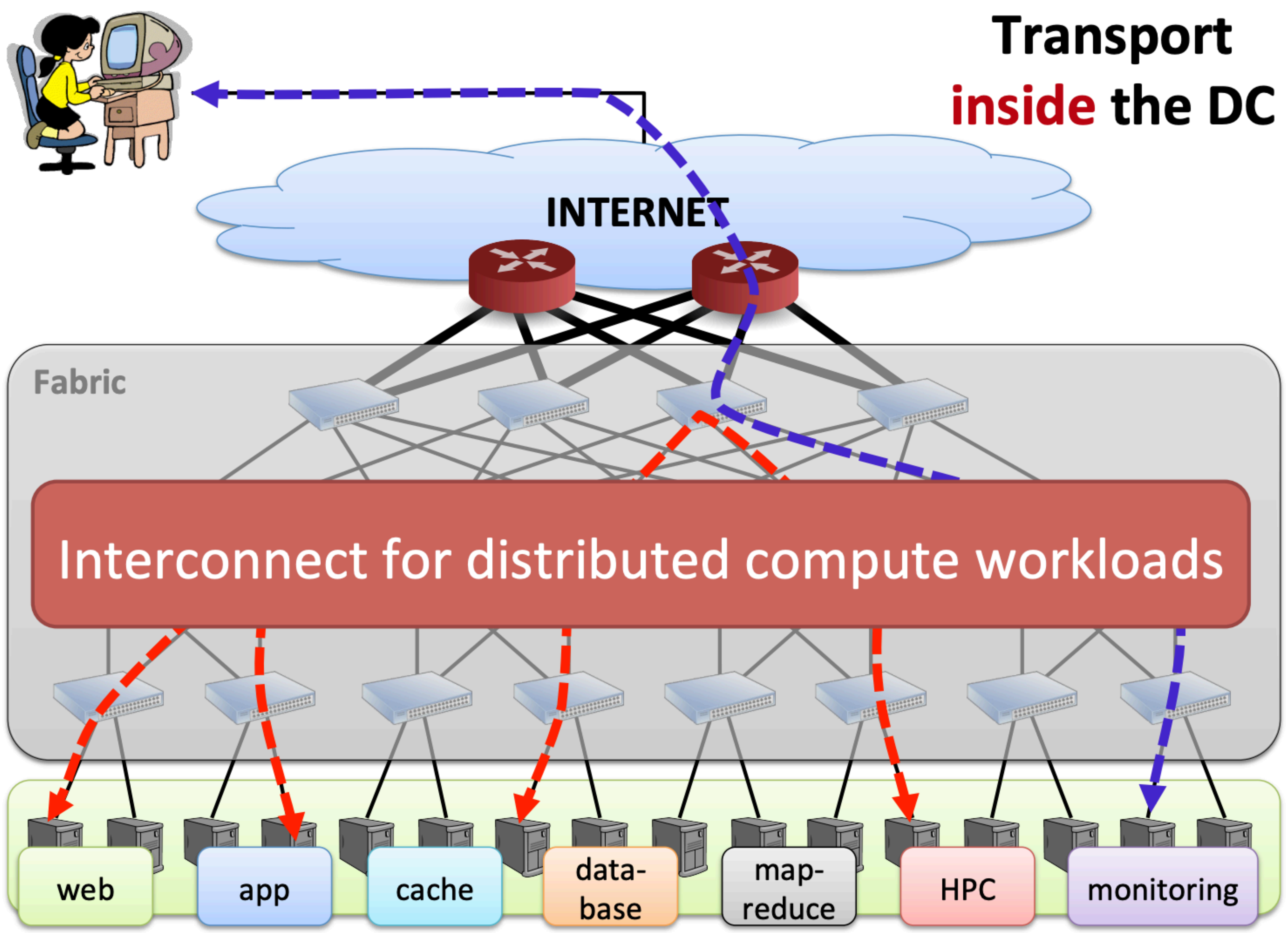
Sangeetha Abdu Jyothi

UCIrvine

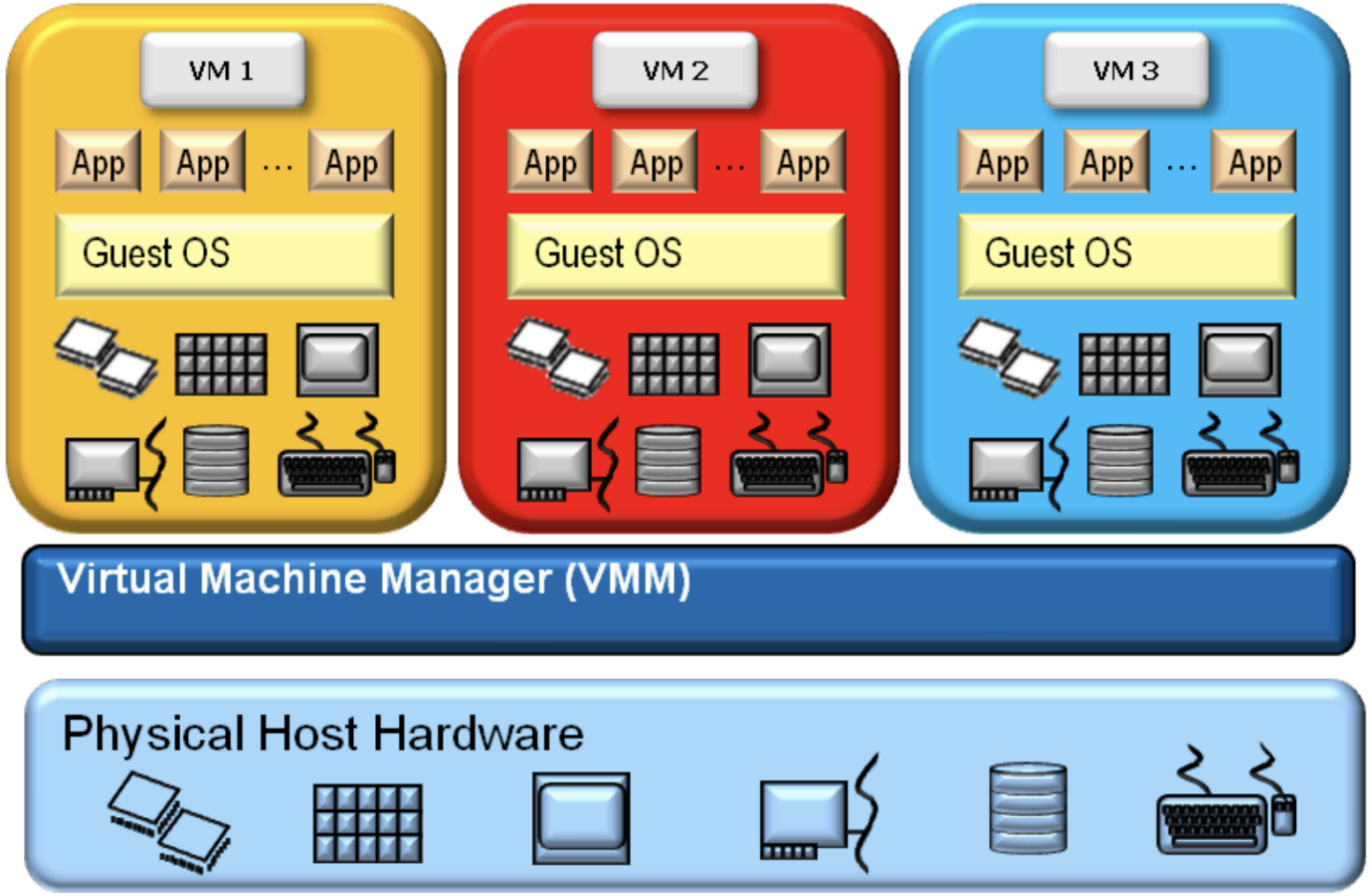
Parts of this lecture uses material from Teemu Koponen, ECE/CS598HPN at UIUC, Andromeda [NSDI'18]

Recap


10.4.1.1
10.4.1.2
10.4.2.1
10.4.2.2
Core
10.0.2.1
10.2.2.1
Aggregation
10.0.1.1
10.2.0.1
Edge
10.0.1.2
10.2.0.2 10.2.0.3
Pod 0
Pod 1
Pod 2
Pod 3


Transport
inside the DC
INTERNET
Fabric
Interconnect for distributed compute workloads
web
app
cache
data-
base
map-
reduce
HPC
monitoring

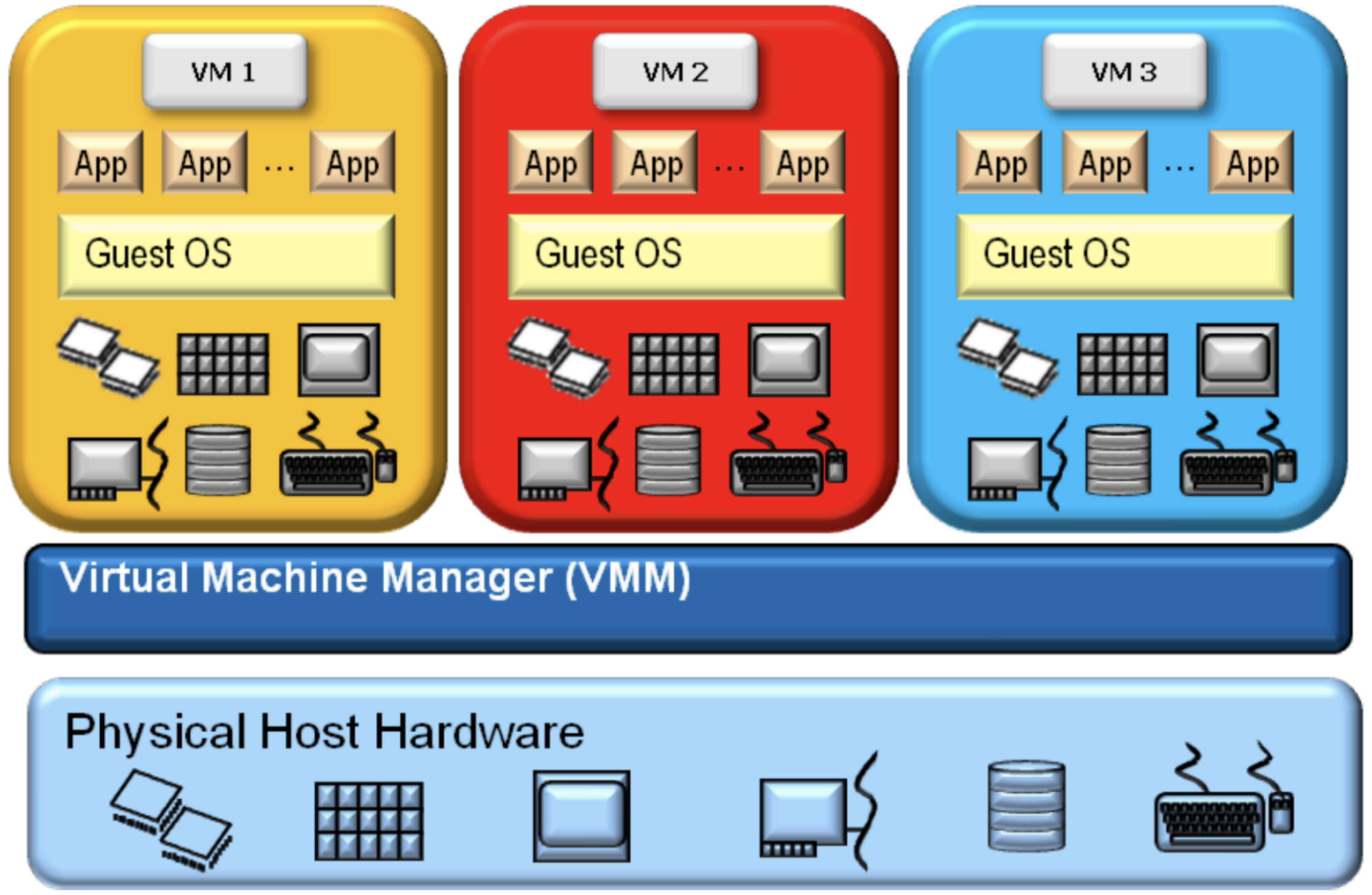
Multi-Tenant Data Centers



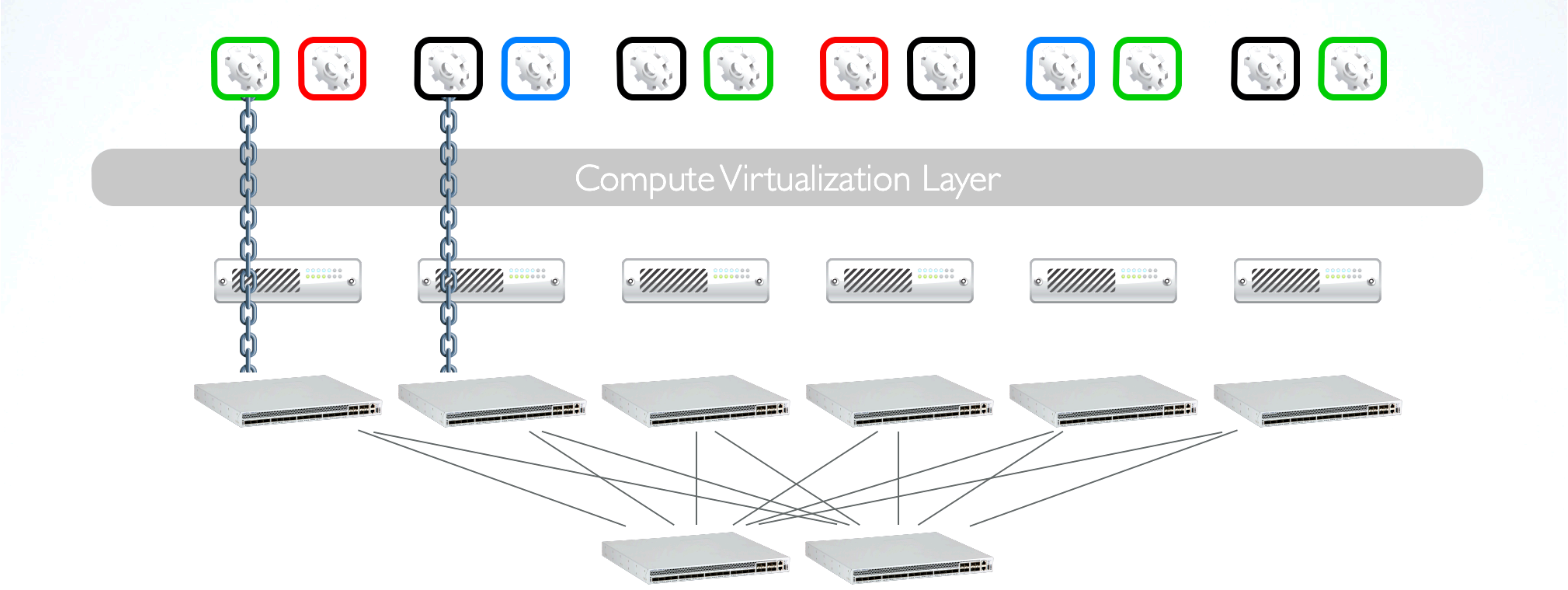
Networking between VMs: Conventional Approach

- Physical network treats each VM as a host directly attached to it

- The vSwitch in the hypervisor extends physical network



Issues with Conventional Approach

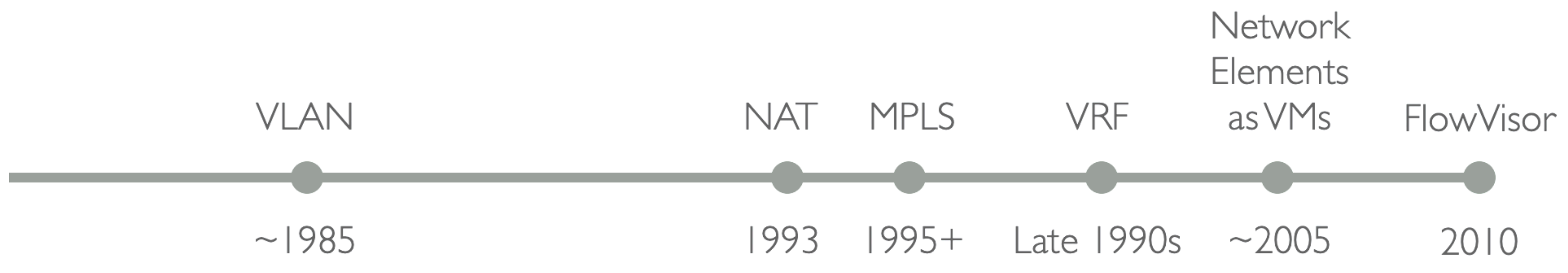


Result with the aforementioned primitives:

- Slow provisioning
- Mobility is limited
- Operationally intensive
- Limited VM placement
- Hardware dependent
- …

Prior Virtualization Techniques

- VLANs: virtualized L2 domains.
- VRF: virtualized L3 forwarding tables
- NAT: virtualized IP address space
- MPLS: virtualized paths.



VLAN	NAT	MPLS	VRF	Elements as VMs	FlowVisor
Subnet	IP address space	Path	L3 FIB	Elements	ASIC

Point solutions that virtualize singular aspects.
Need for a more holistic and global approach.

Network Virtualization

Allows creating virtual networks

(each with independent service models,

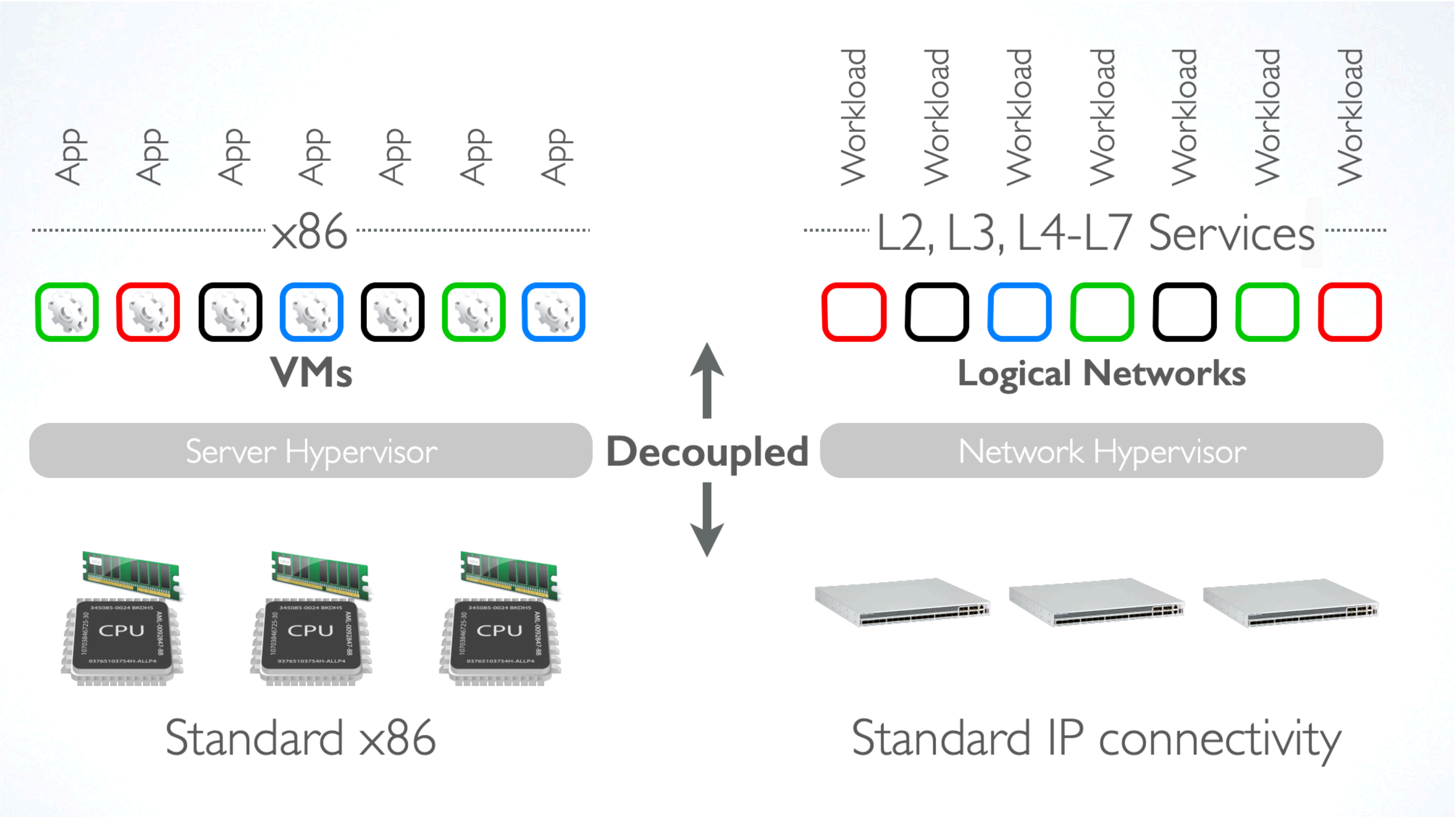
topologies, and addressing schemes)

over the same physical network.

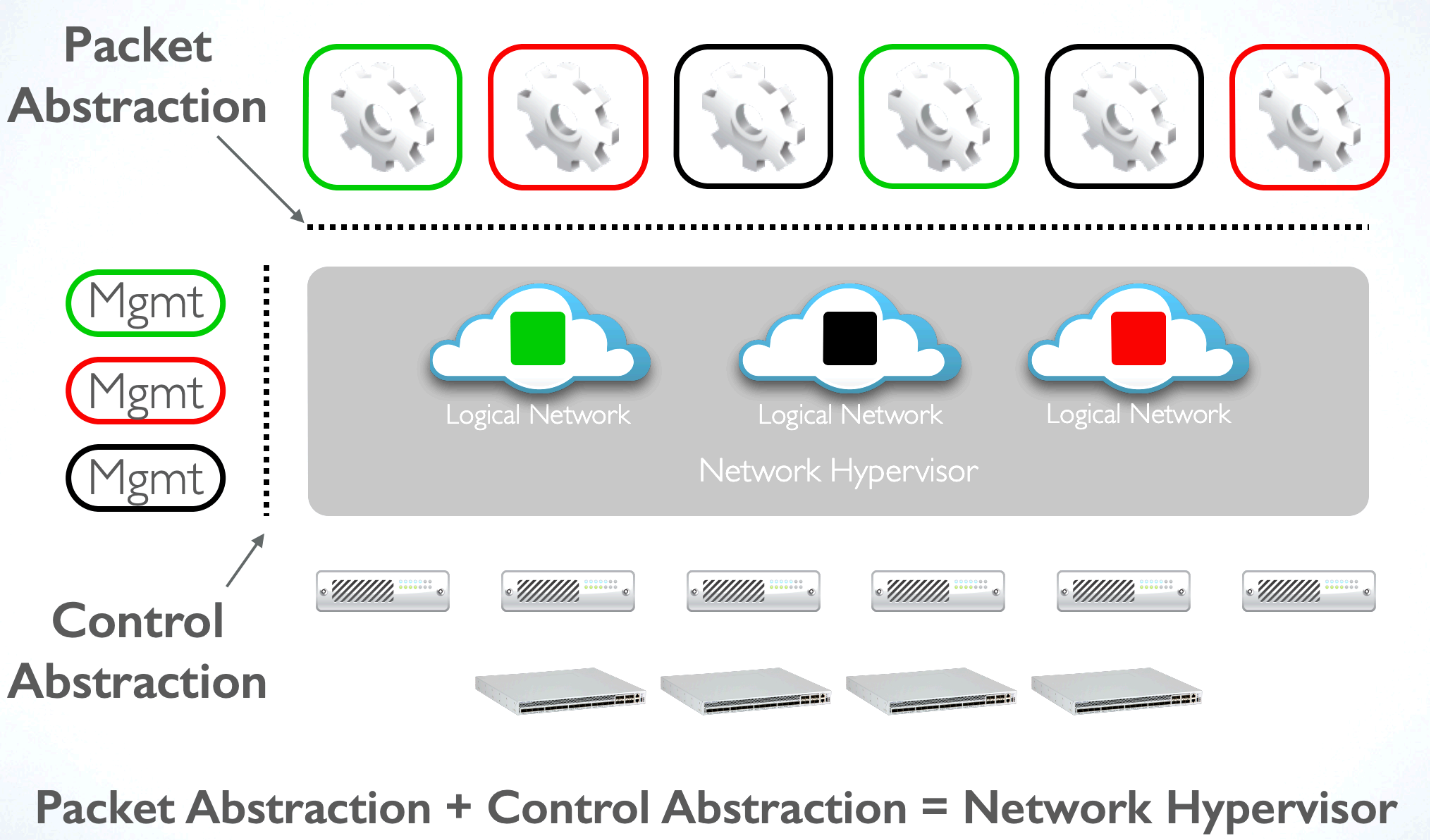
These virtual networks are

created, configured, and managed

via global abstractions.



Network Hypervisor



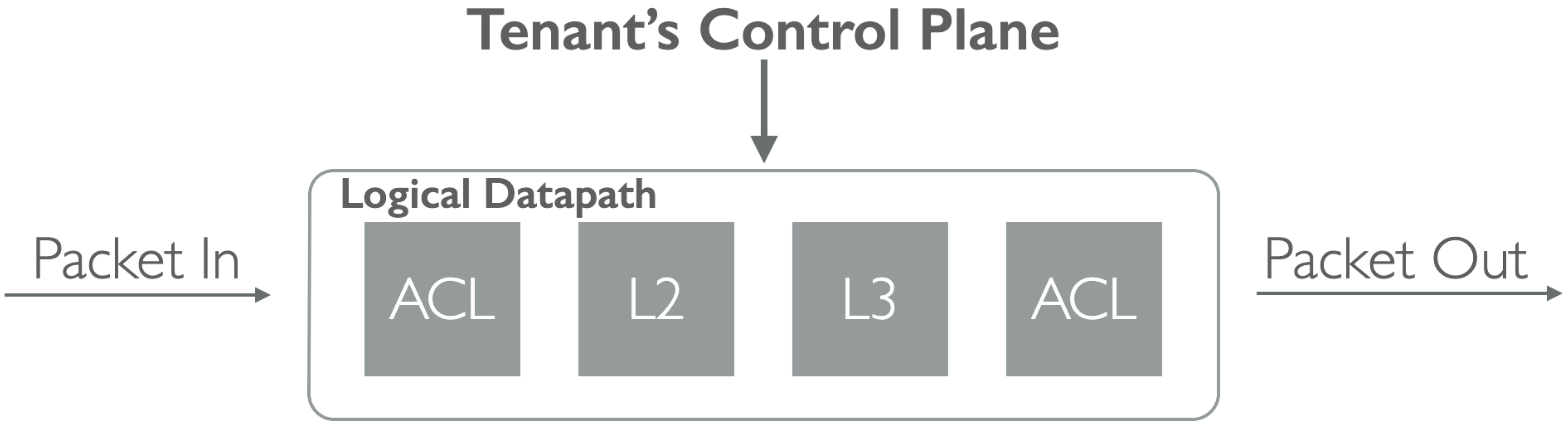
What are the Abstractions?

Packet abstraction

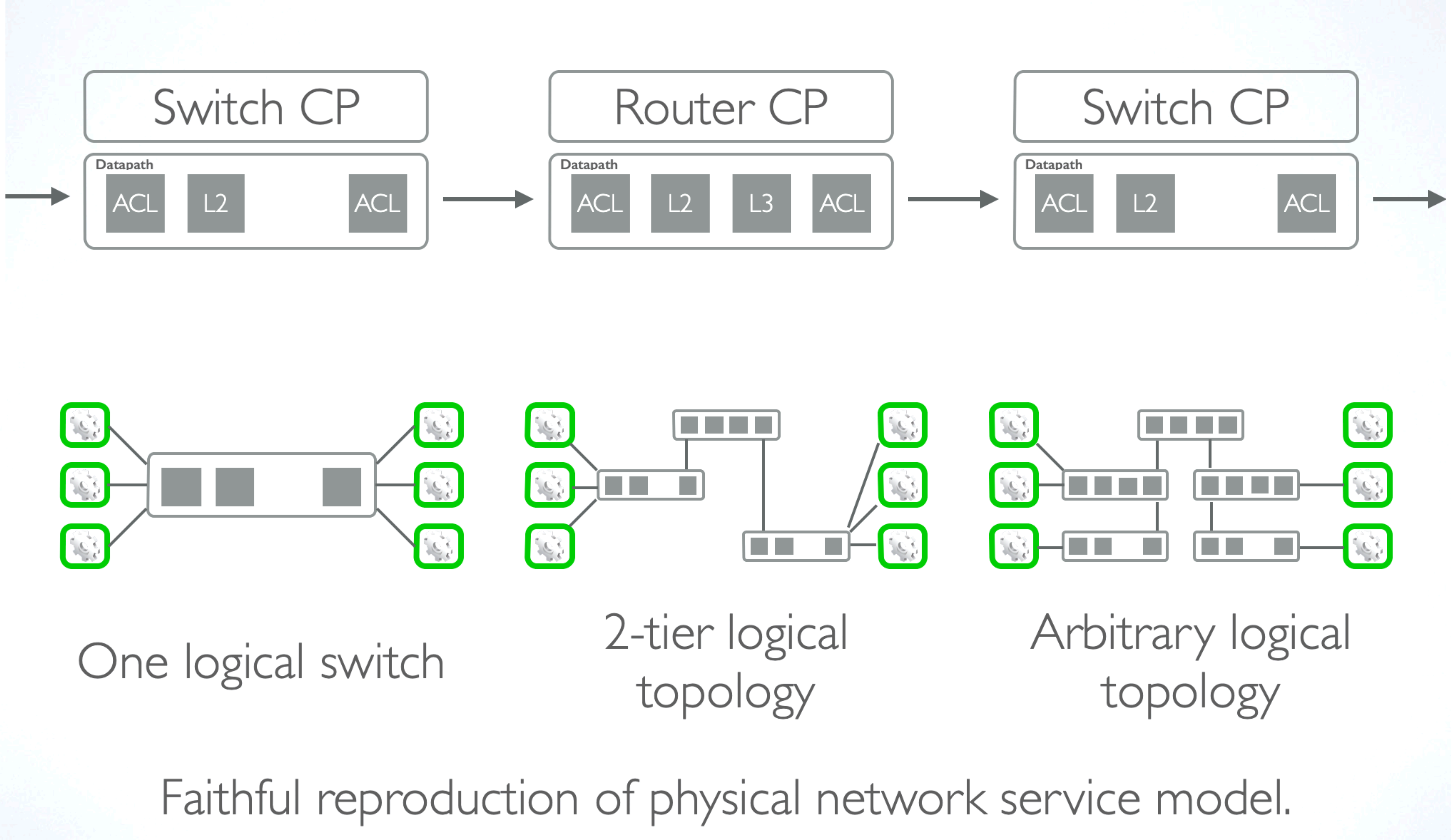
- Compliance with standard TCP/IP stack is a necessity:
  - L2, L3 semantics (unicast, ARP, …)

Control abstraction

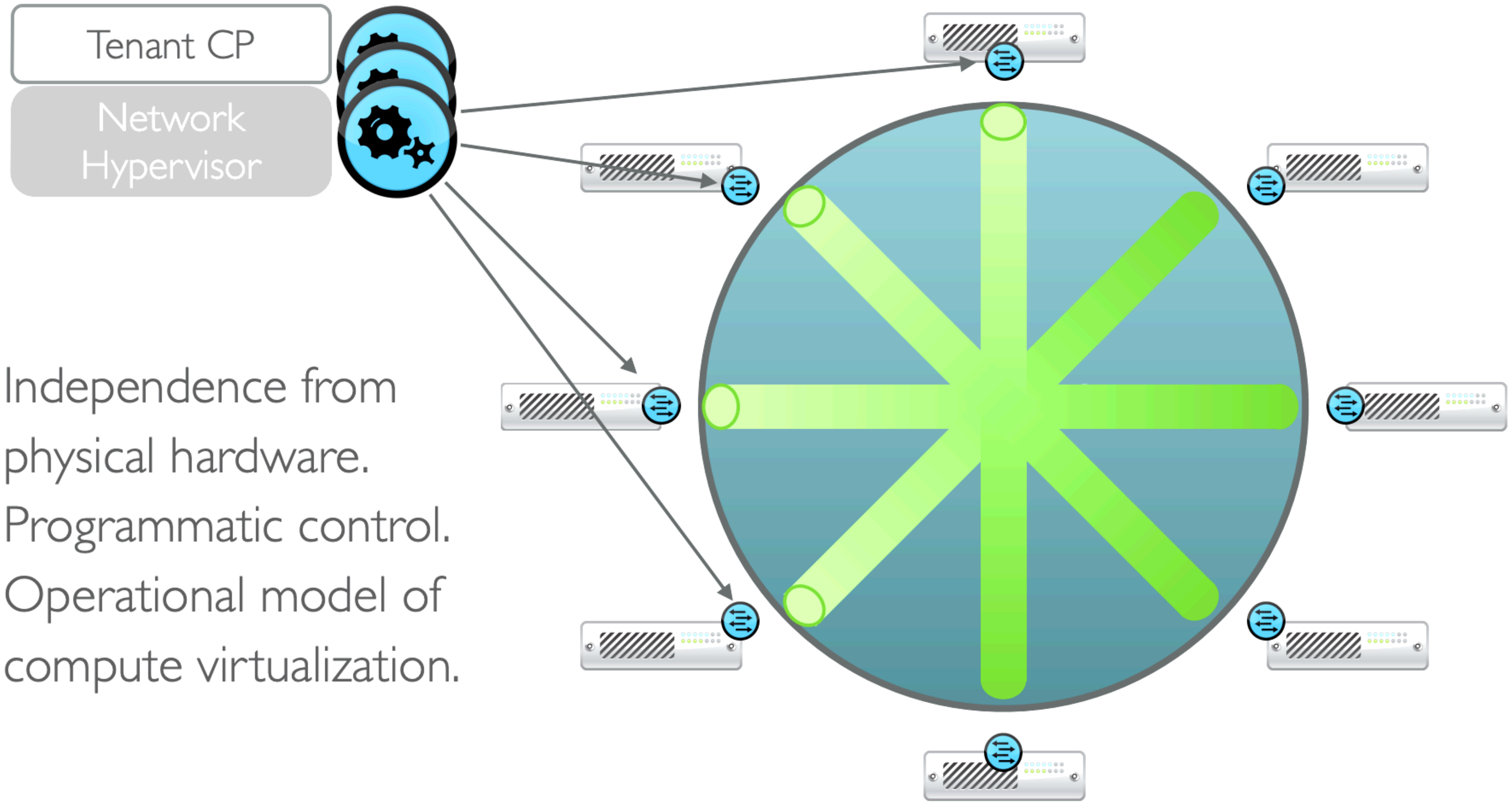
- Networking has no single high level control interface.
- There's a low-level one though!



Generality of Datapath



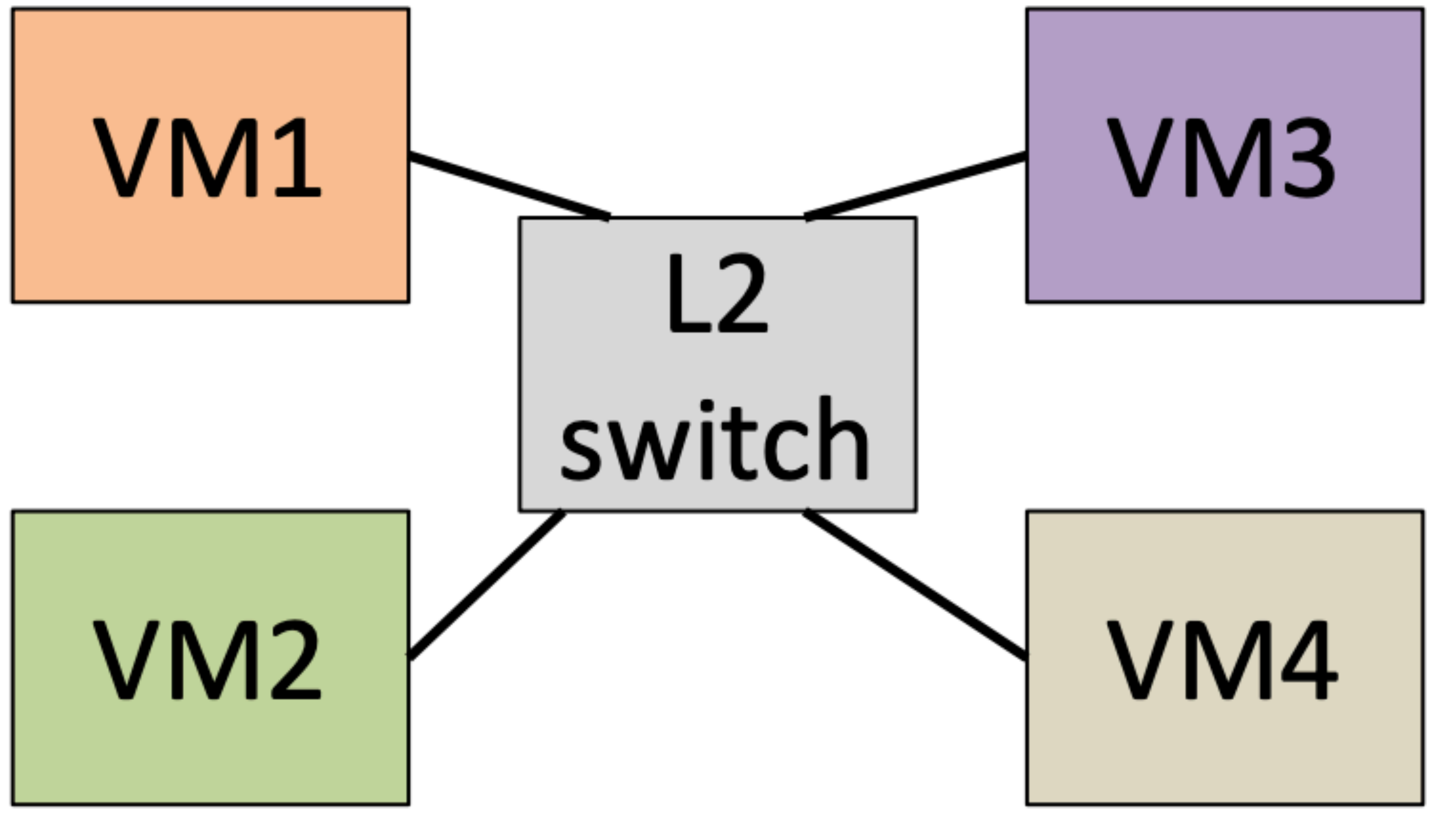
Where to Implement?



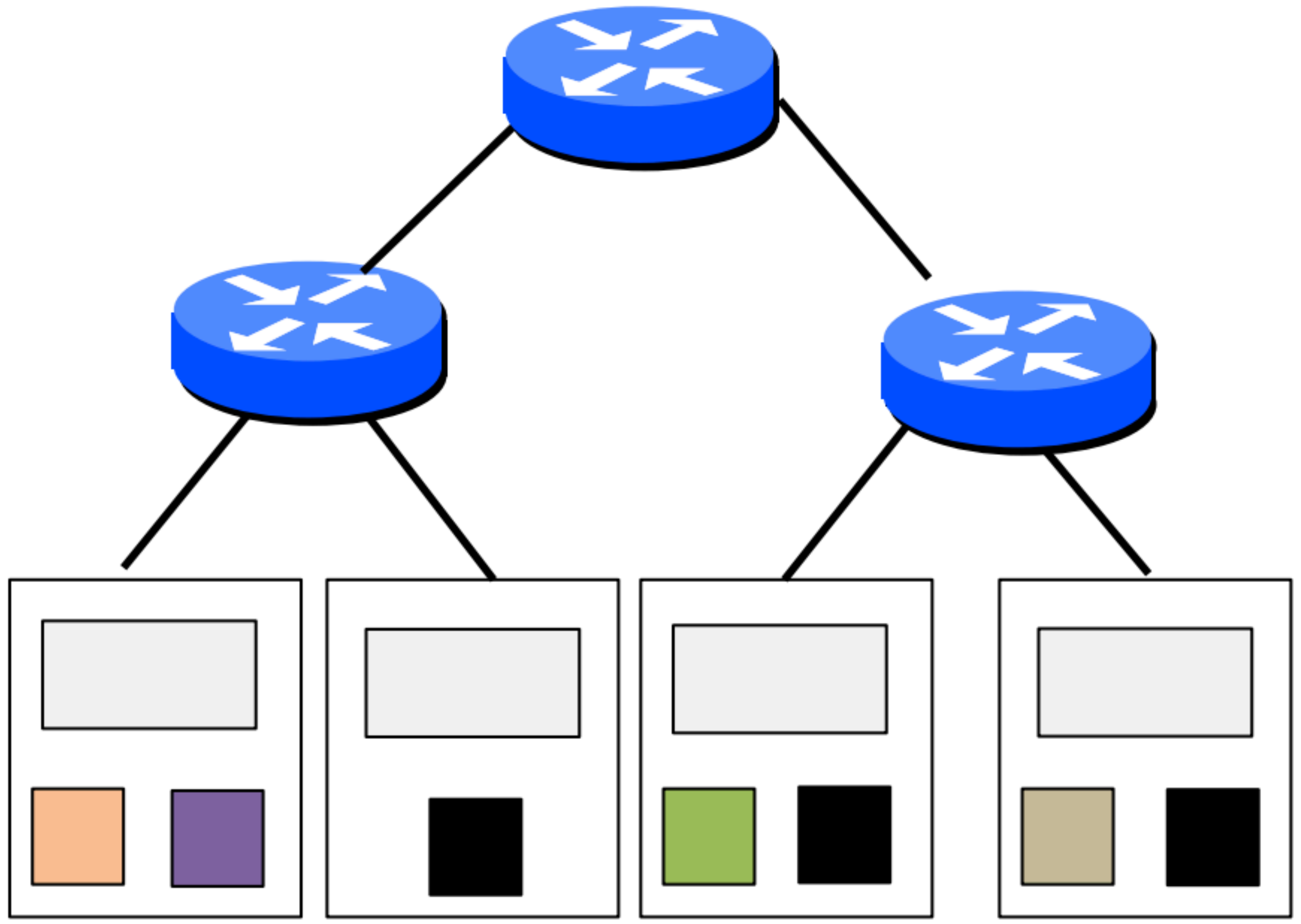
- Independence from physical hardware.
- Programmatic control.
- Operational model of compute virtualization.

No extra x86 hops: just the source and destination hypervisor!

Network Virtualization

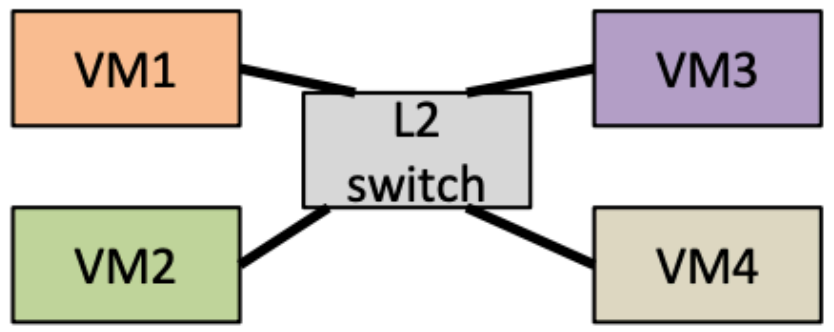


Abstraction



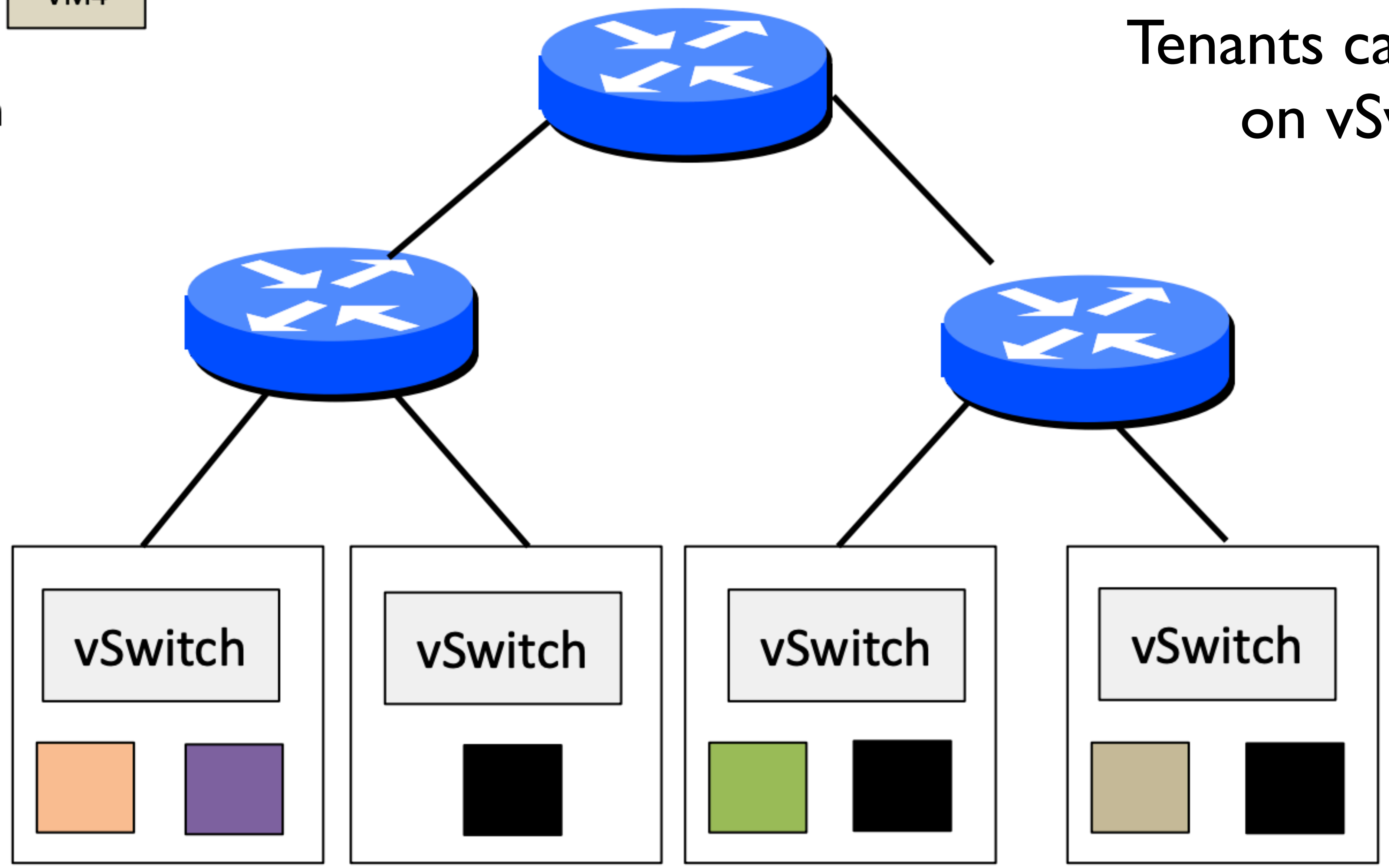
Physical Topology

vSwitches provide the Abstraction



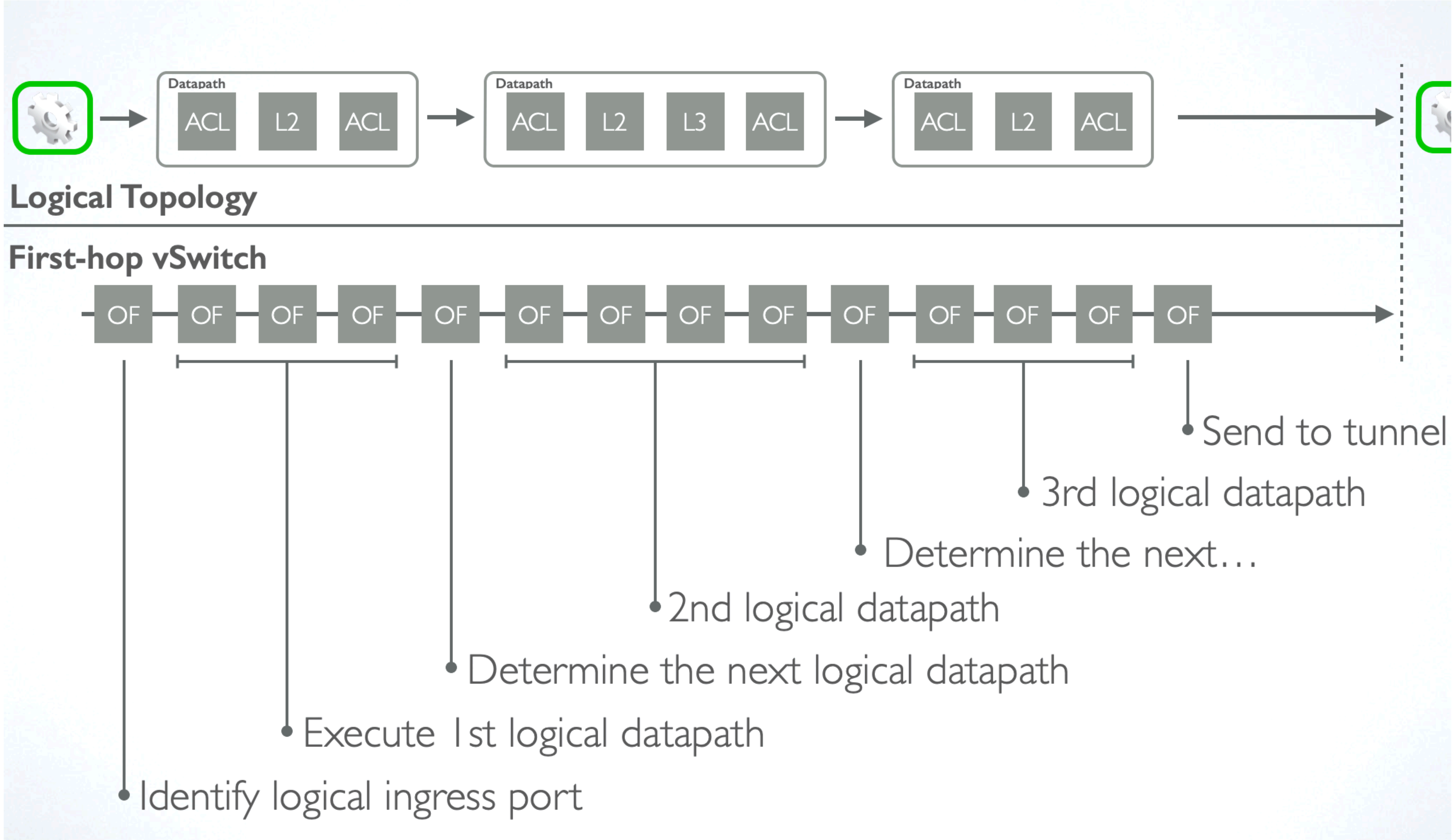
Abstraction

Tenants can configure logical datapaths on vSwitches using OpenFlow

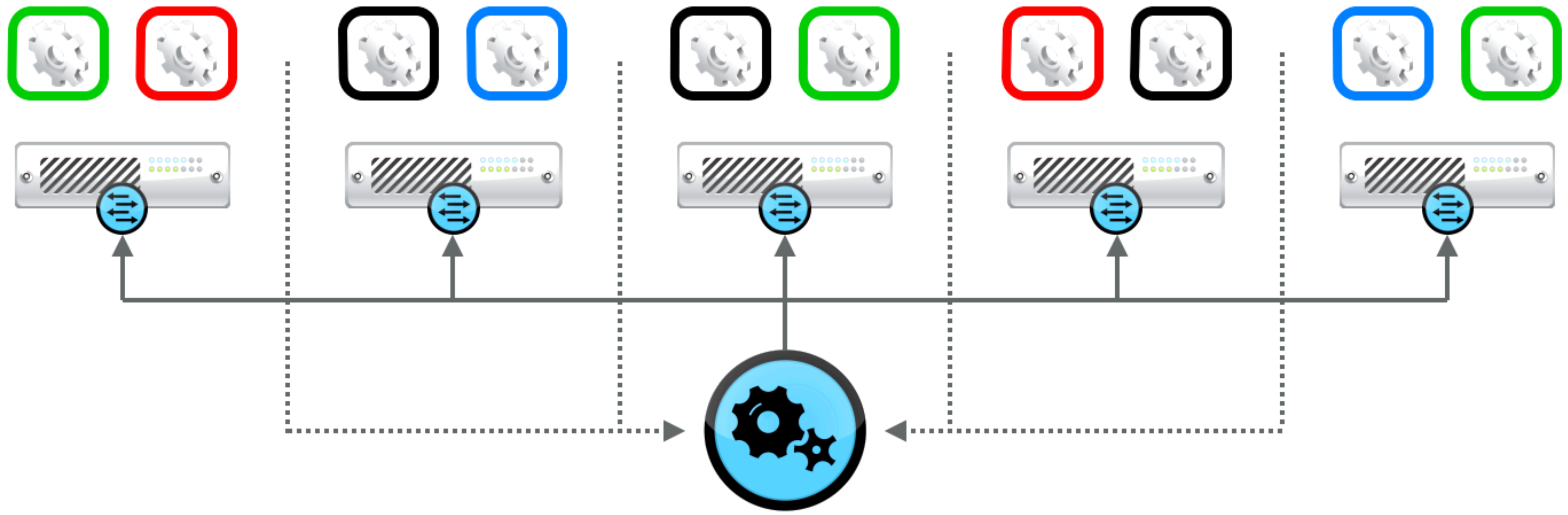


Physical Topology

Inside the vSwitch



Computational Challenge



1. Controllers learn the location of VMs.

2. Controllers proactively compute & push all forwarding state required to connect each VM.

→ Forwarding State = **F**(configuration, VM locations)

Repeat above as logical configuration or physical configuration (VM placement) changes.

Challenge: How to compute $O(N^2)$ volume of low-level OpenFlow and OVSDB state, when inputs change all the time.

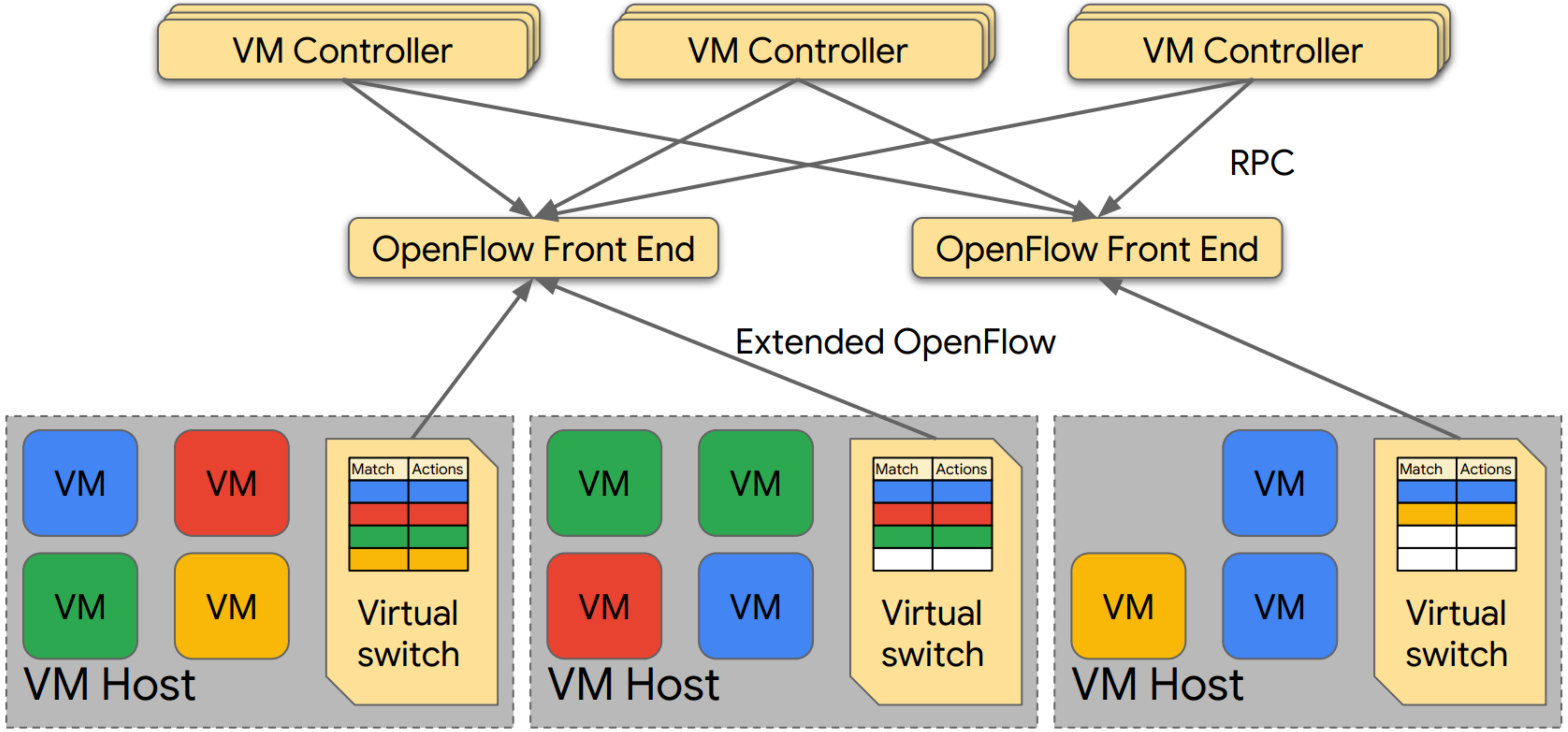
State Computation

- How to Scale Computation
    - Incremental computation and pushing for quick updates
- How to Guarantee Correctness
    - Avoid all handwritten finite state machines, machine generated instead
    - Datalog based declarative language to program the forwarding state
    - Shard the computation across controller cluster

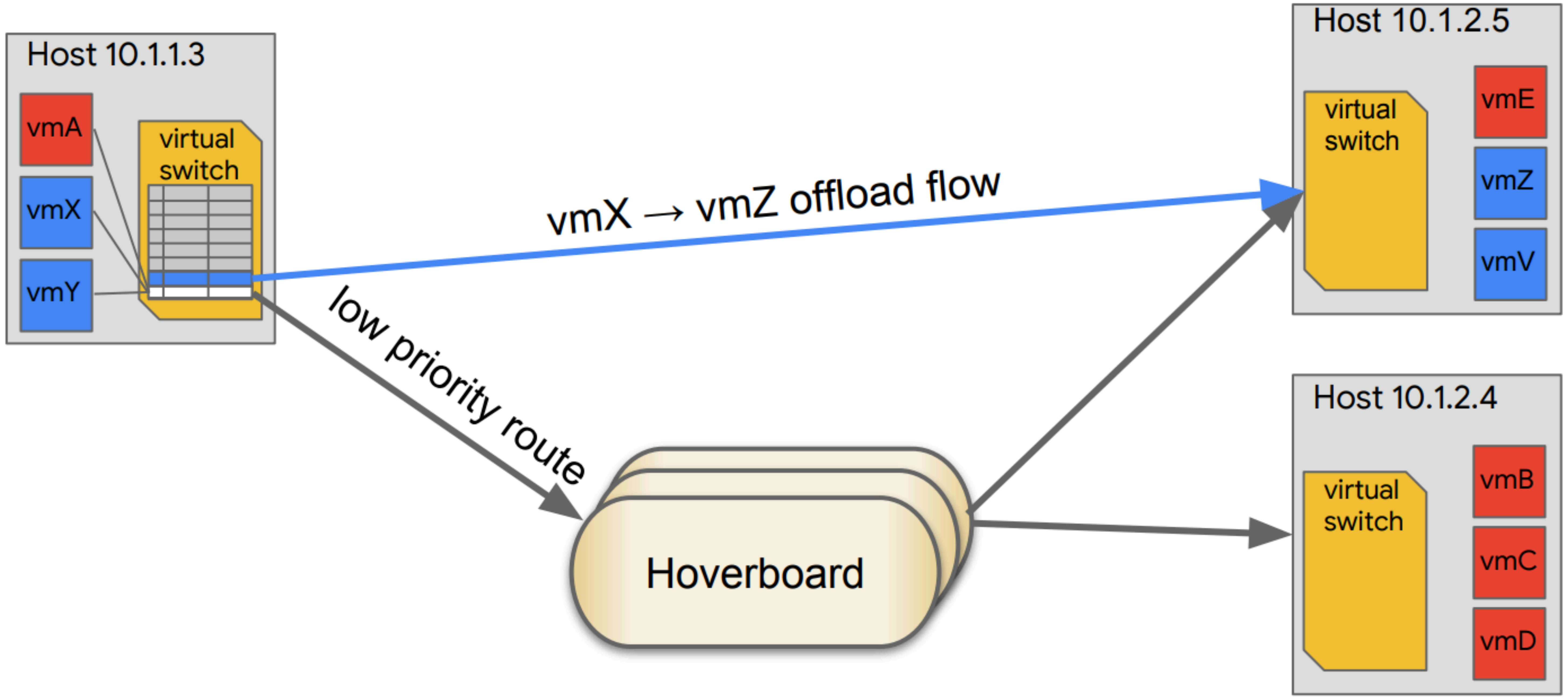
Andromeda [NSDI'18]

- Google's Network Virtualization Platform
- Goals
  - Performance and Isolation
    - High throughput and low latency, regardless of the actions of other tenants
  - Velocity
    - Quickly develop and deploy new features and performance improvements
  - Scalability
    - Large networks, many tenants, rapid provisioning

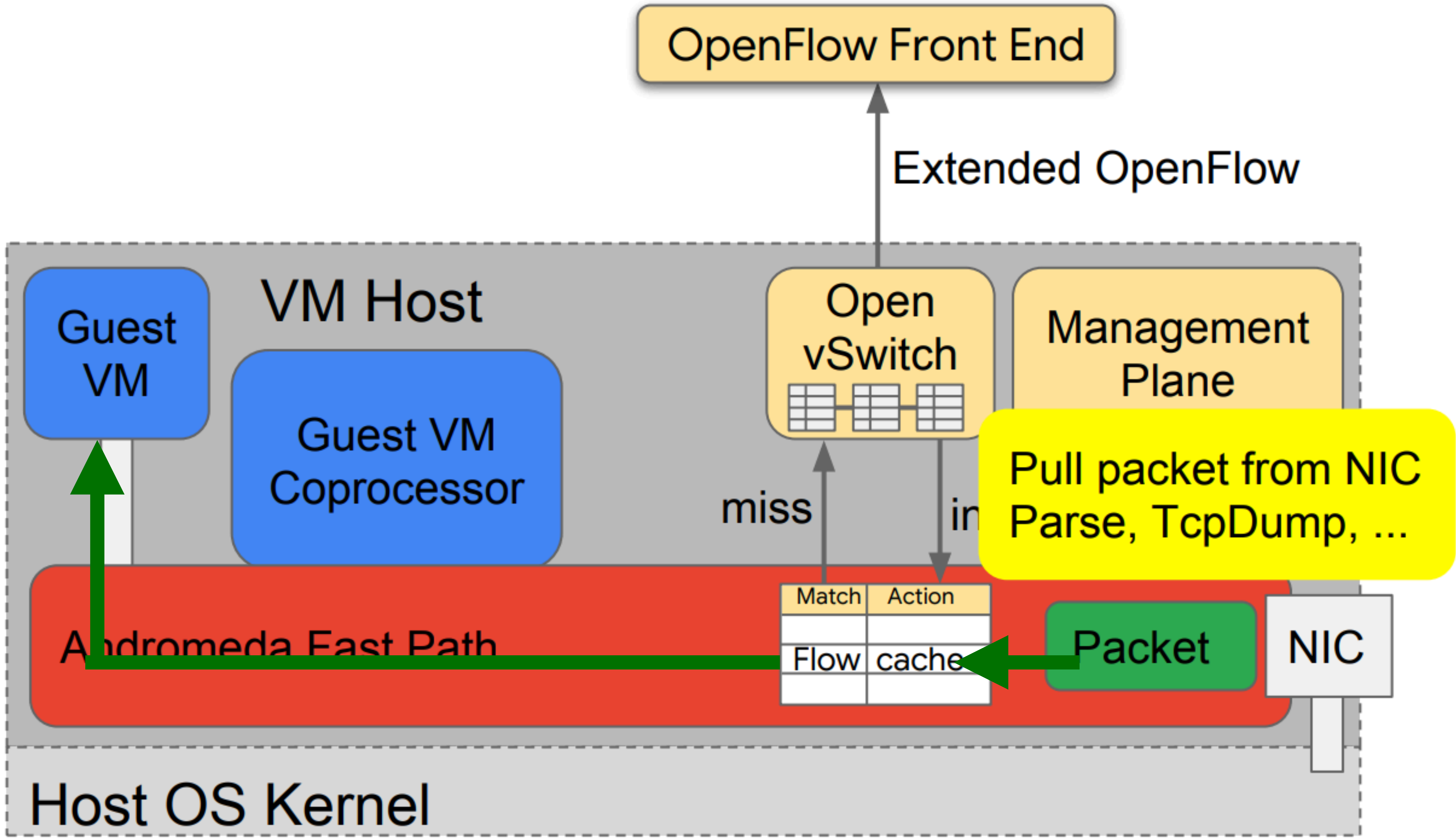
Andromeda Architecture



Hoverboard Offloading

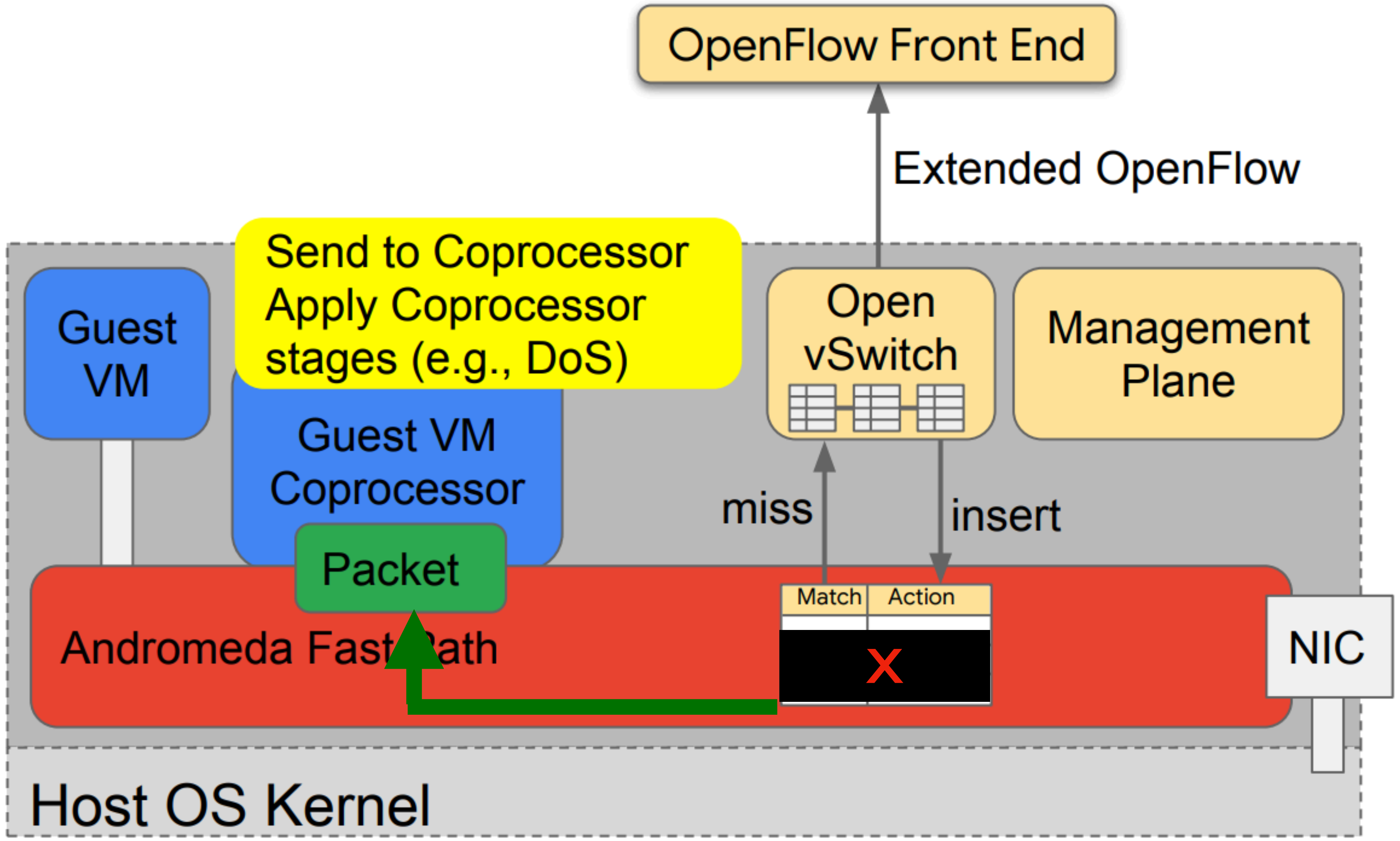


Data Plane: Fast Path



High performance traffic processed end-to-end on **Fast Path**

Data Plane: Coprocessor path



Coprocessors are per-VM threads CPU attributed to VM container

Coprocessors execute CPU-intensive packet ops such as DoS

Decouples feature growth from Fast Path speed

Network Virtualization: Other Possibilities

- New "out-of-band" header fields without breaking legacy TCP/IP stacks
- Enforcing security policies
- … ?

Thanks!