midokura

How Linux kernel enables MidoNet’s overlay networks for virtualized environments.

LinuxTag Berlin, May 2014

About Me: Pino de Candia

At Midokura since late 2010:

- Joined as a Software Engineer
- Managed the Network Agent team starting in 2012
- VP of Engineering since April 2013

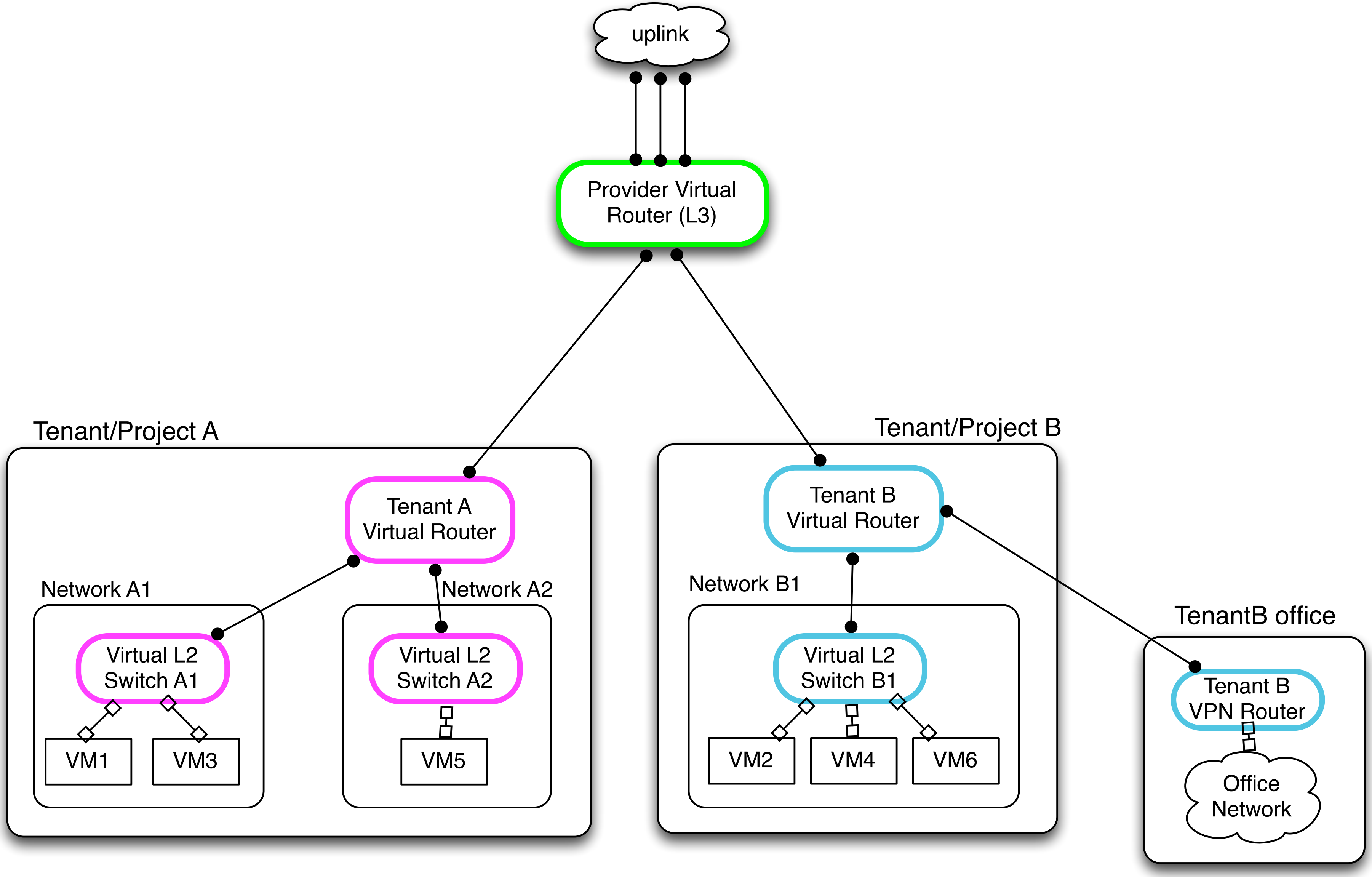
Prior to Midokura spent 5 years at Amazon:

- Helped build Dynamo, a NoSQL data store
- Managed an internal software team focused on caching technologies

Talk Agenda

- Network Virtualization Definition and Requirements
- How MidoNet implements Network Virtualization
- Advantages of the Network Overlay approach
- How Linux Kernel makes this possible

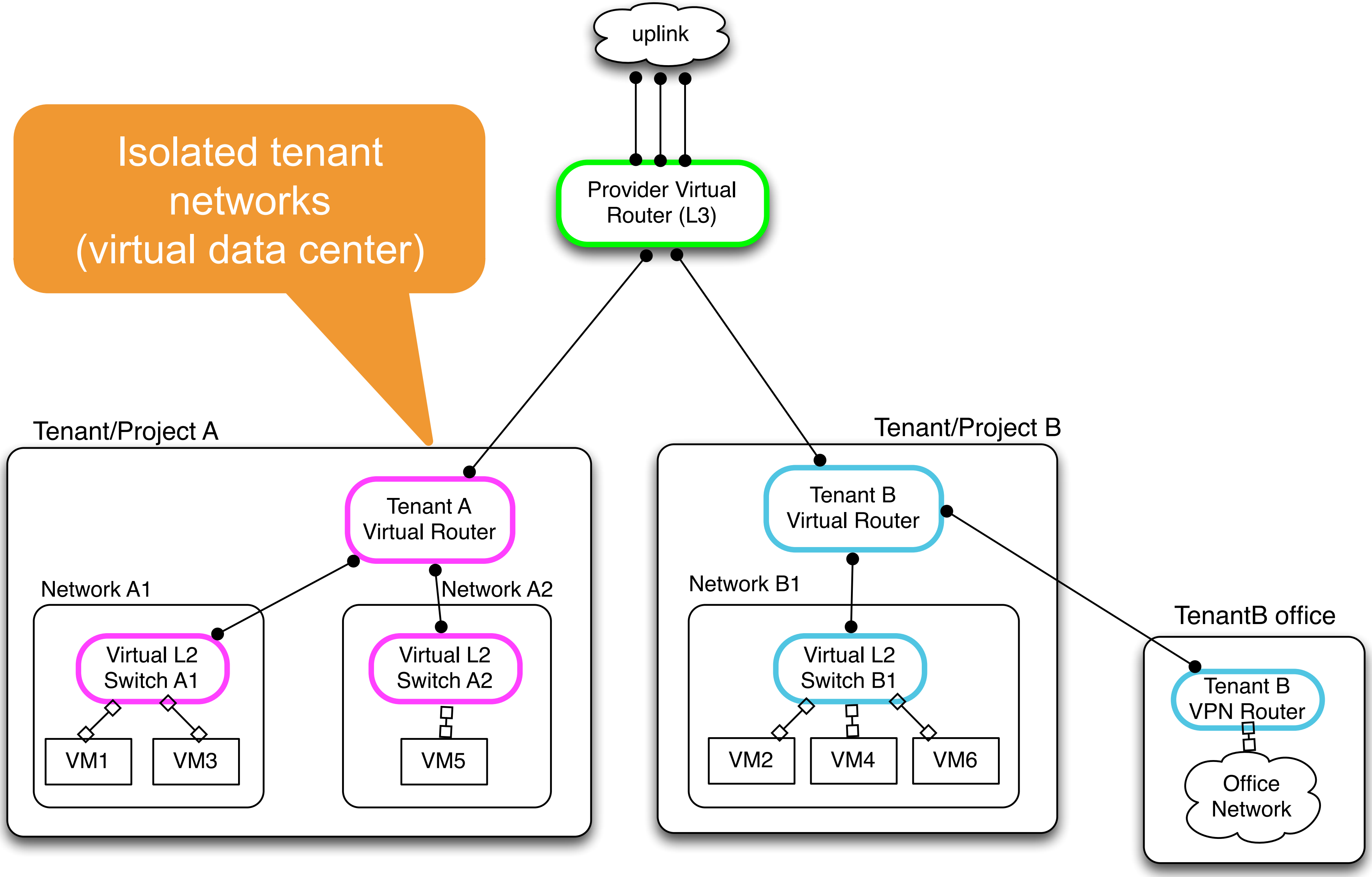
Requirements for NV



midokura

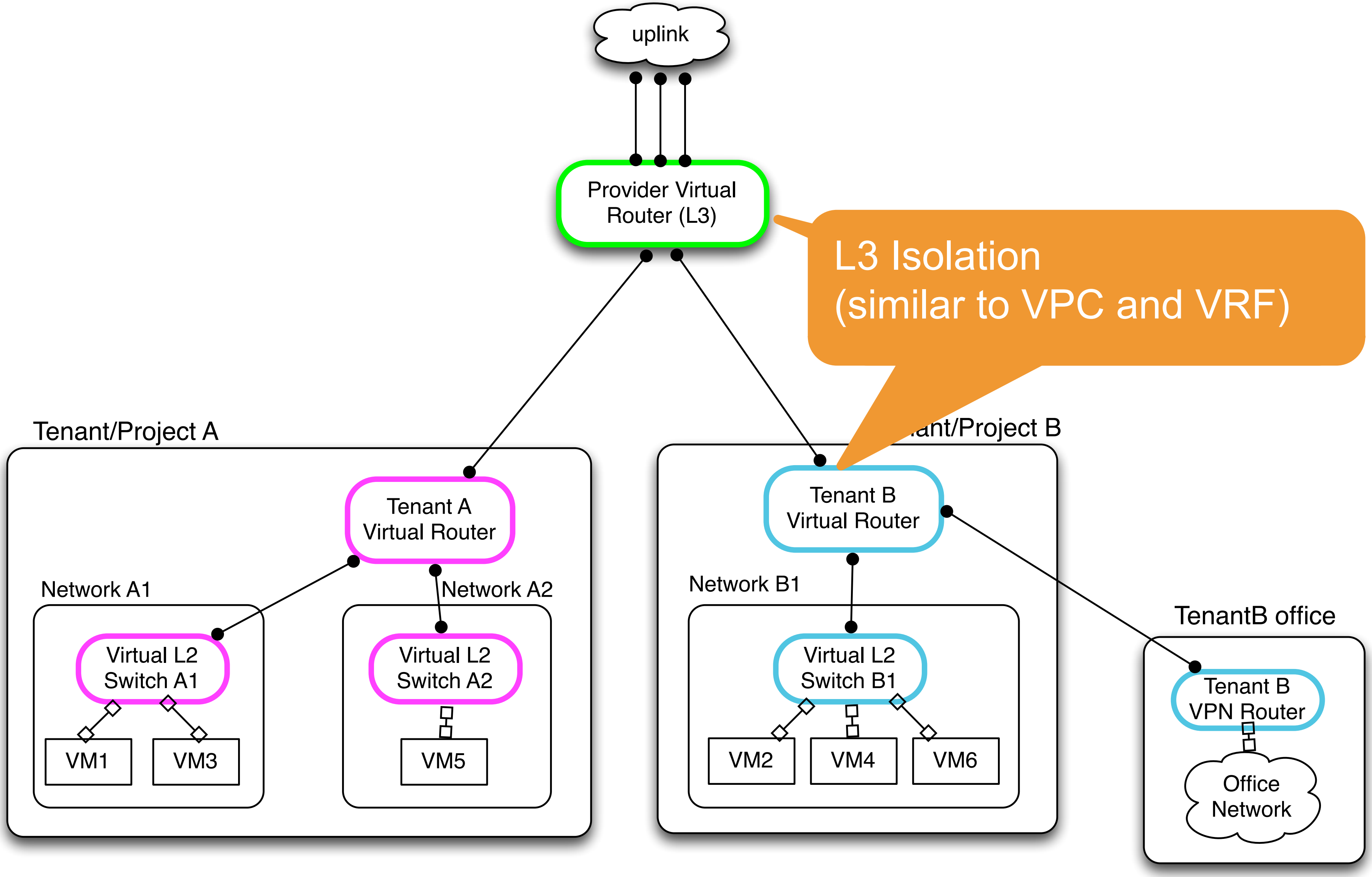
Requirements for NV

Isolated tenant networks (virtual data center)



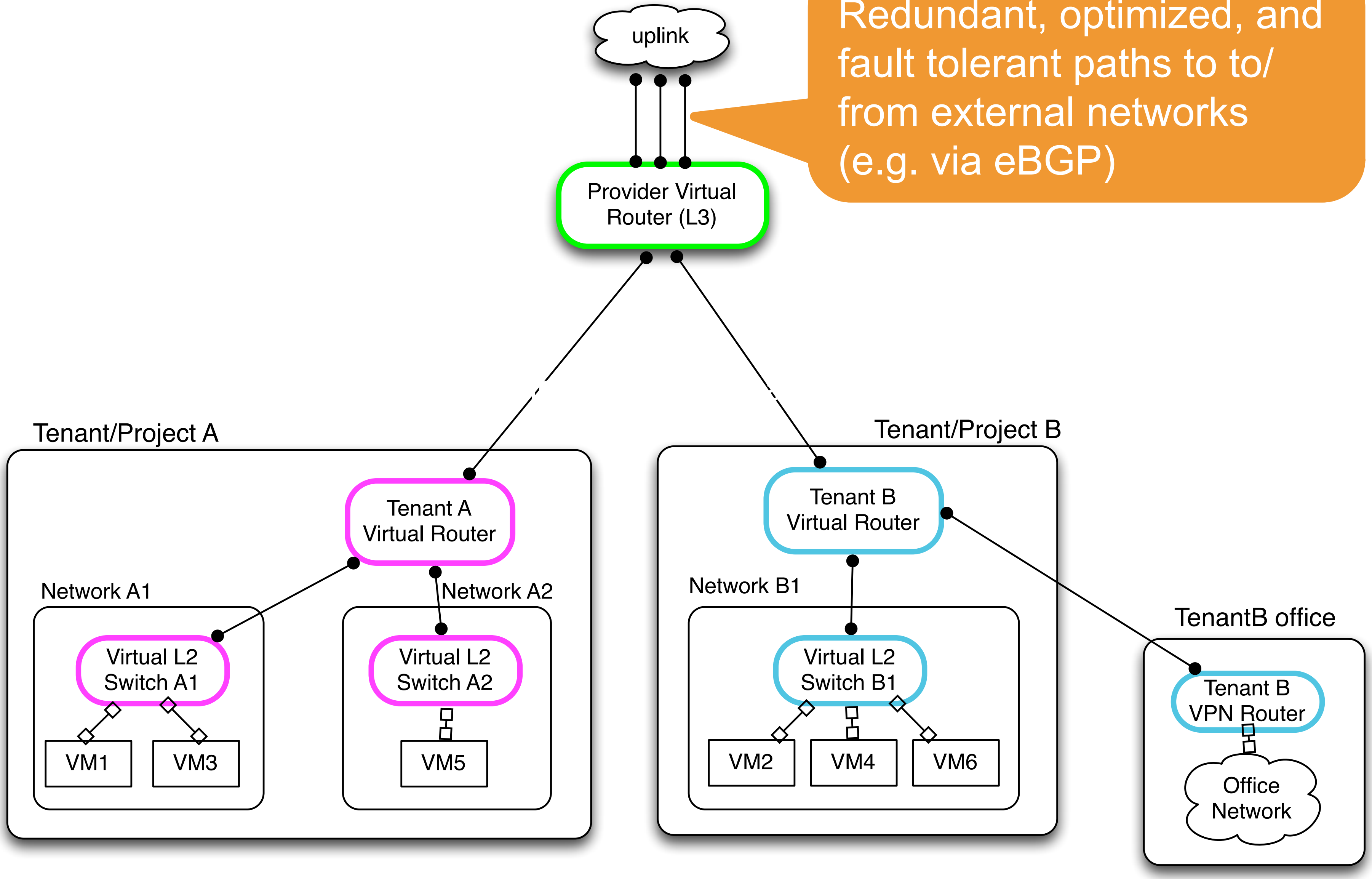
midokura

Requirements for NV



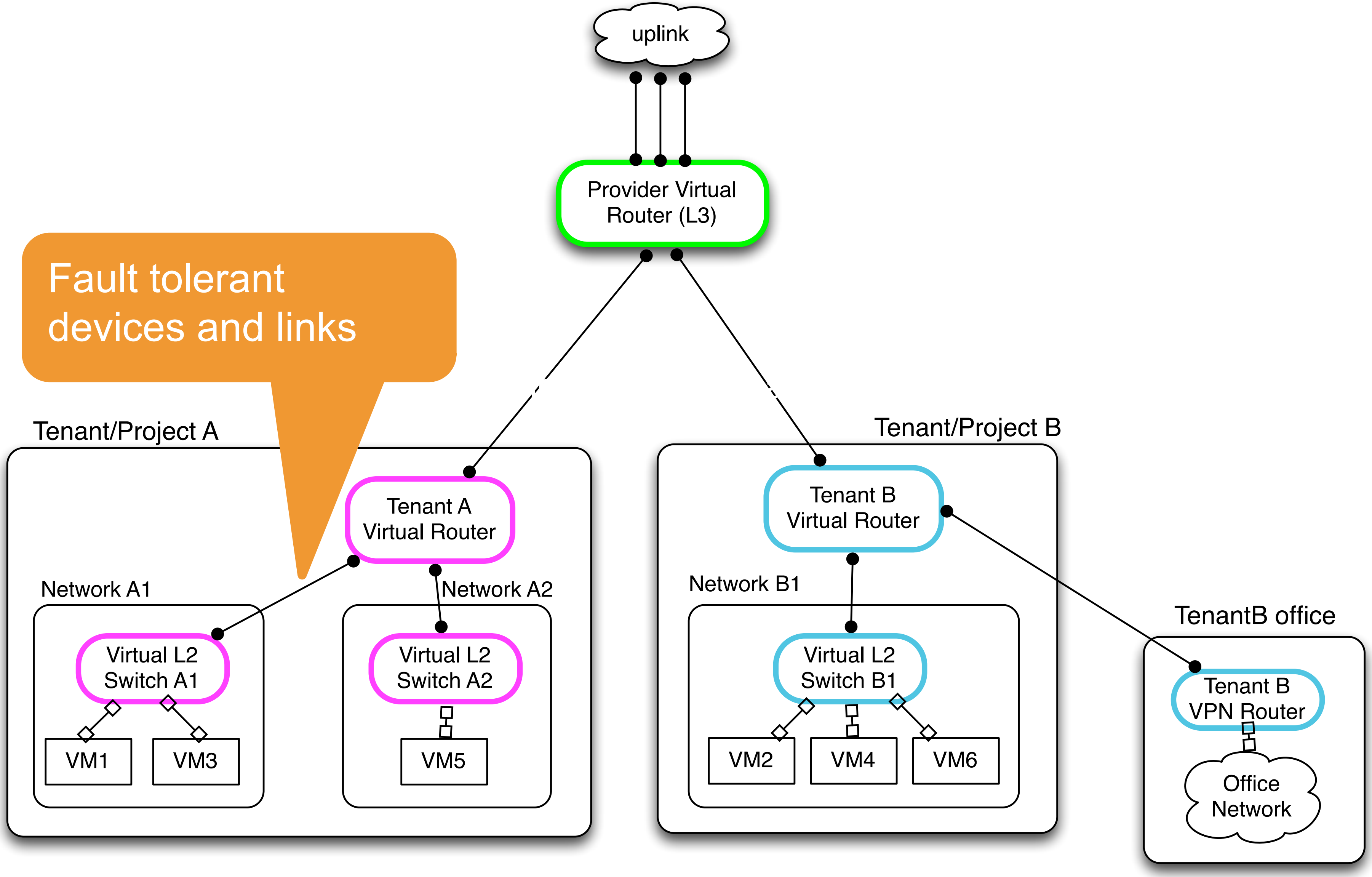
L3 Isolation
(similar to VPC and VRF)

Requirements for NV



Redundant, optimized, and fault tolerant paths to to/from external networks (e.g. via eBGP)

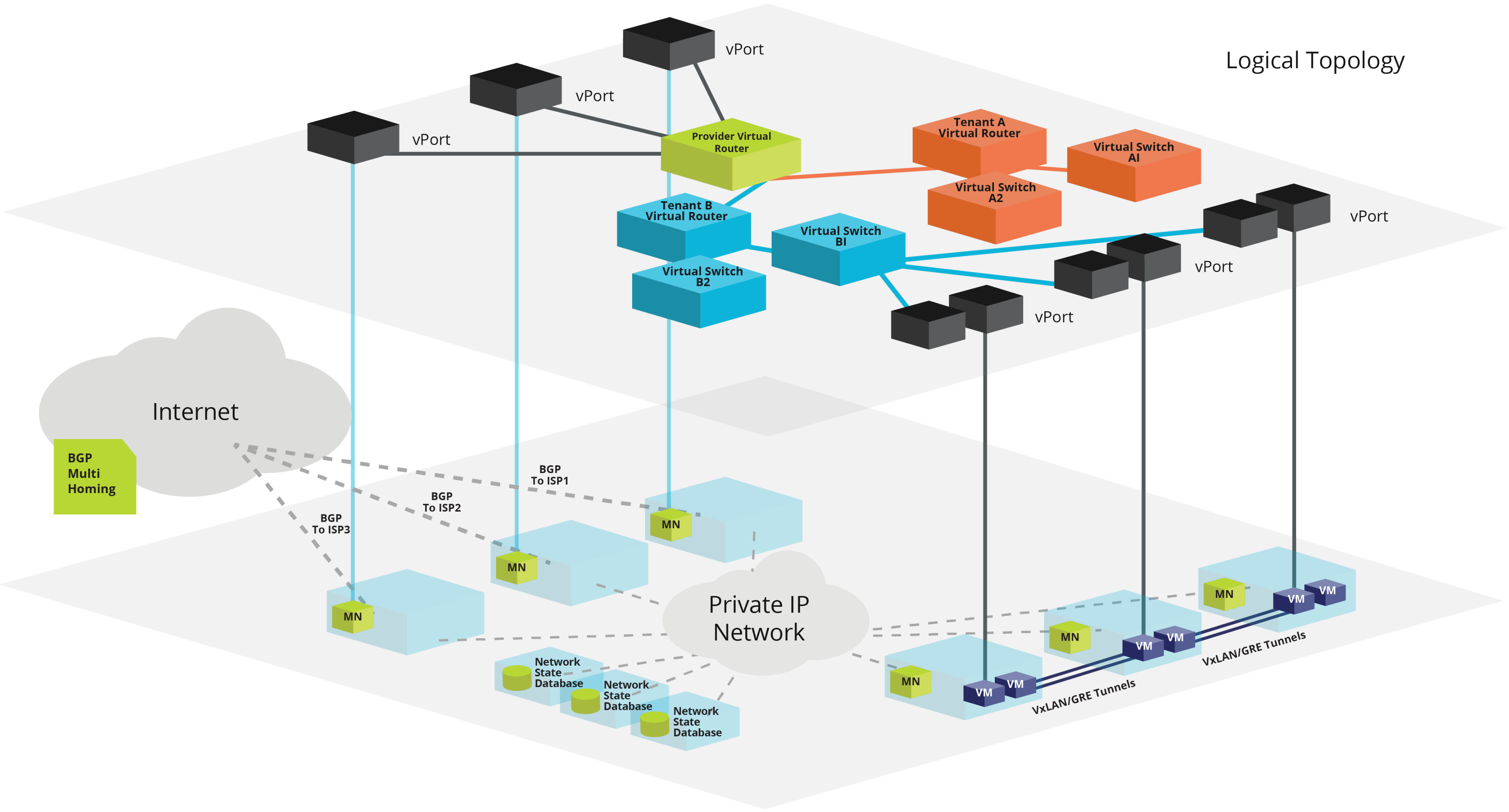
Requirements for NV



Fault tolerant devices and links

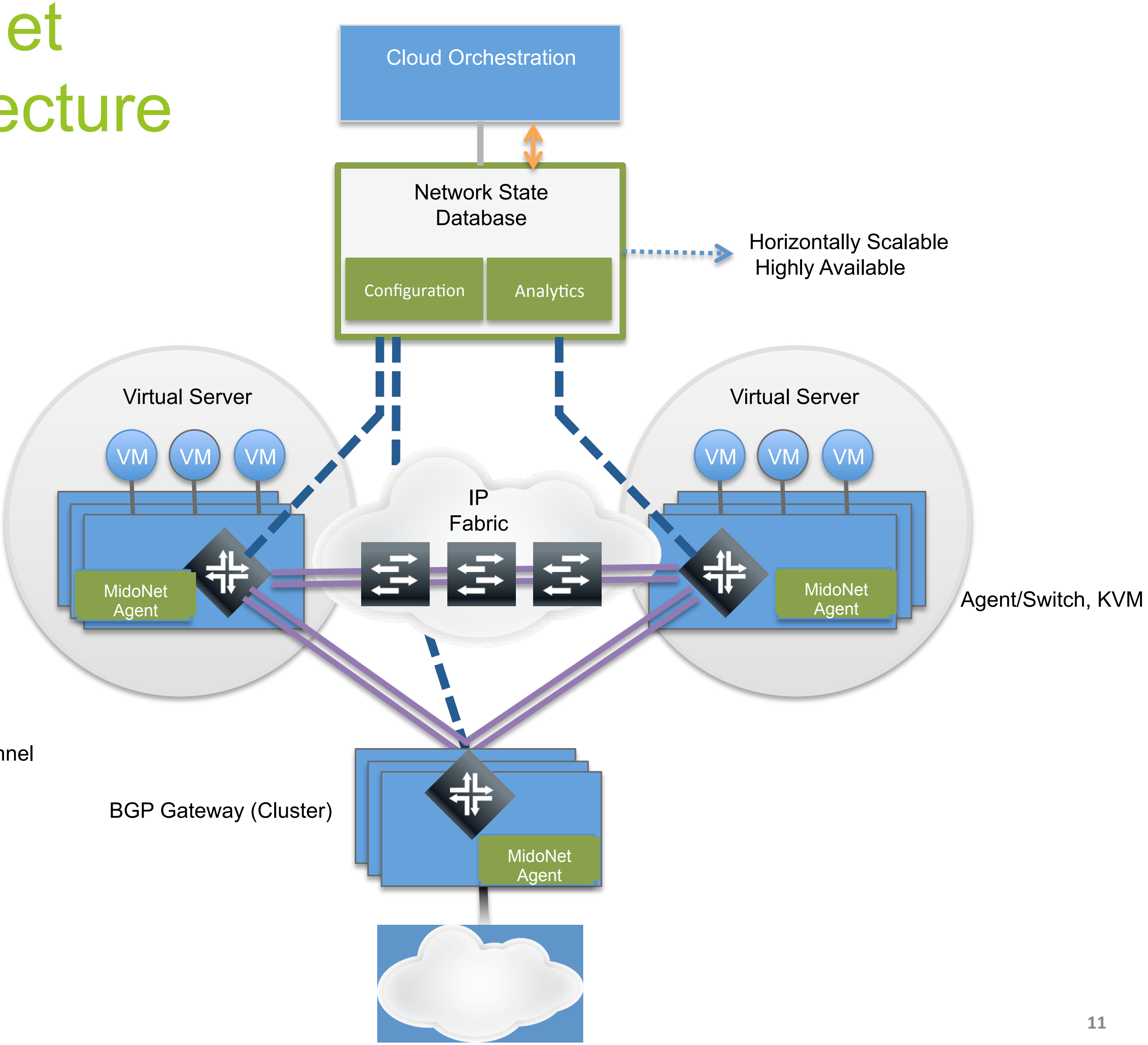
How MidoNet implements network virtualization using overlays

Logical Topology – Overlay Networks



midokura

MidoNet Architecture

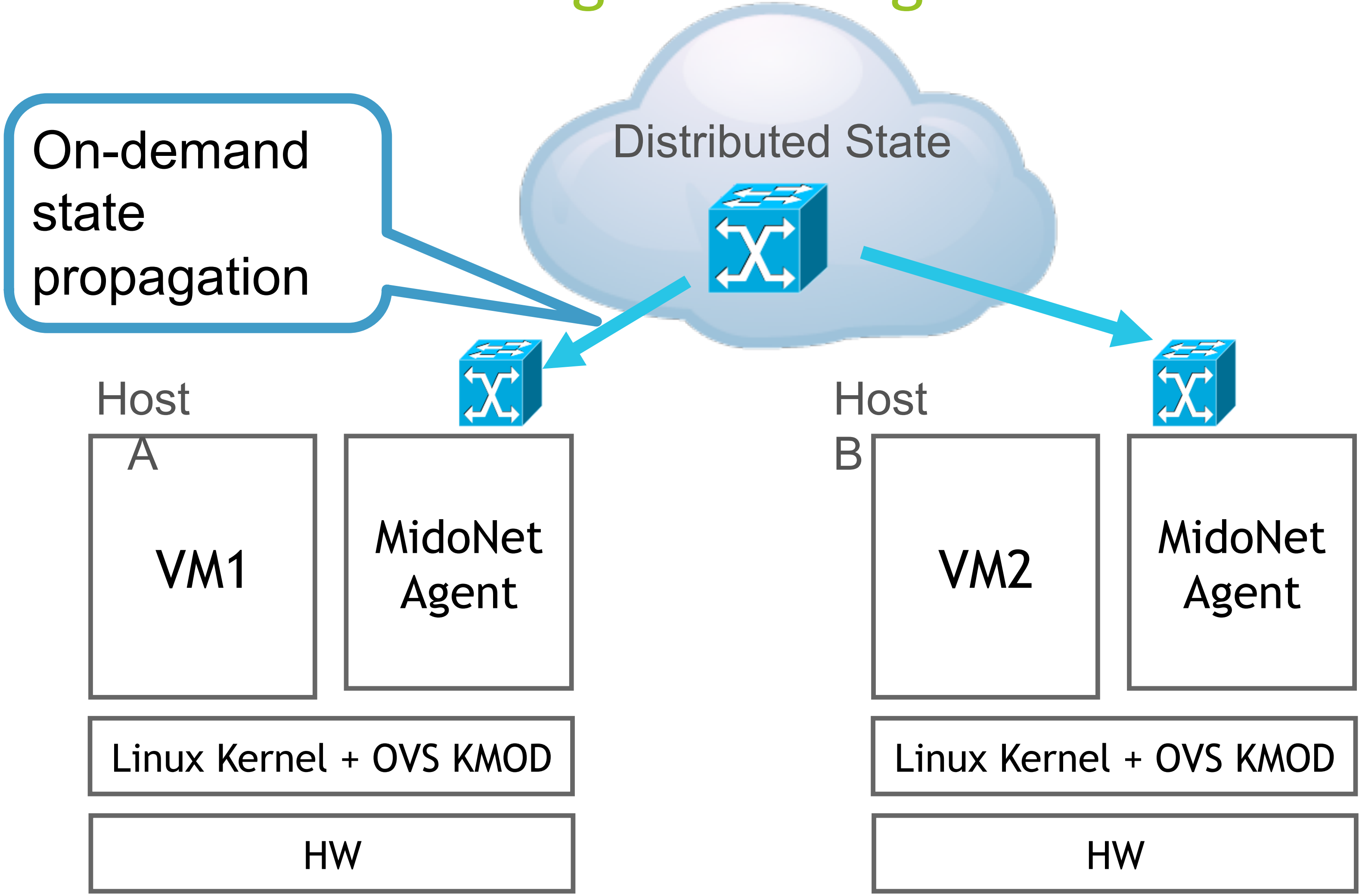


GRE/VXLAN tunnel

TCP traffic

NB APIs

Virtual Networking at the Edge



On-demand state propagation

Distributed State

Host A

VM1

MidoNet Agent

Linux Kernel + OVS KMOD

HW

Host B

VM2

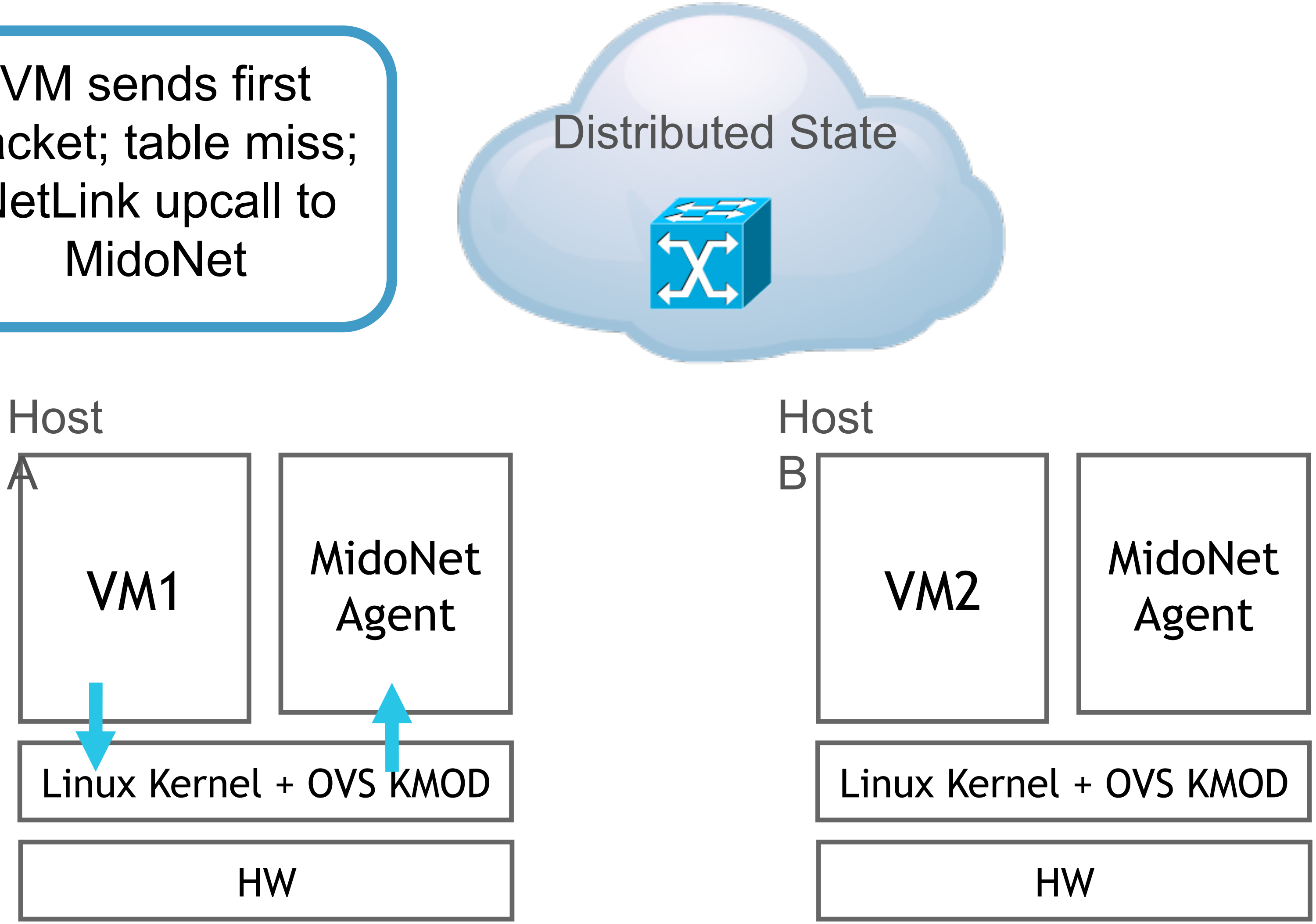
MidoNet Agent

Linux Kernel + OVS KMOD

HW

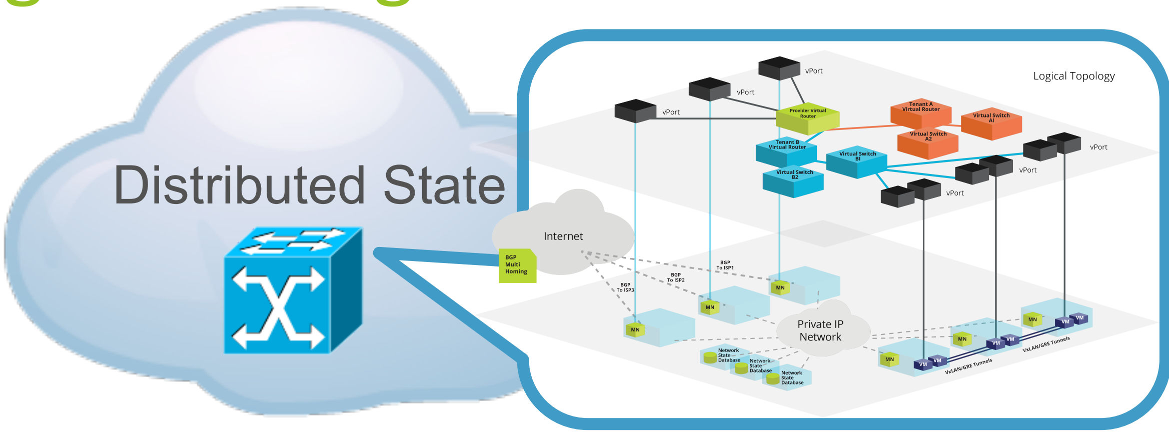
Virtual Networking at the Edge

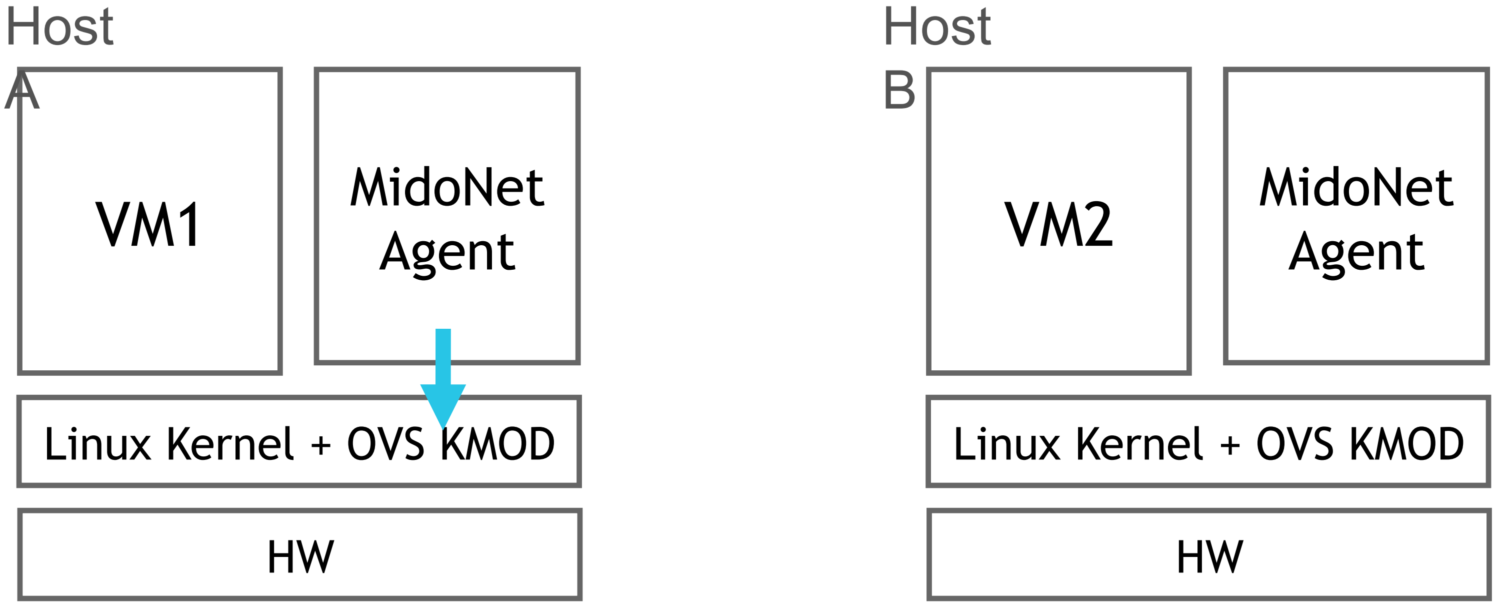
VM sends first packet; table miss; NetLink upcall to MidoNet



Virtual Networking at the Edge

MidoNet agent locally processes packet (virtual layer simulation); installs local flow (drop/mod/fwd)


Distributed State


Host A

VM1

MidoNet Agent

Linux Kernel + OVS KMOD

HW

Host B

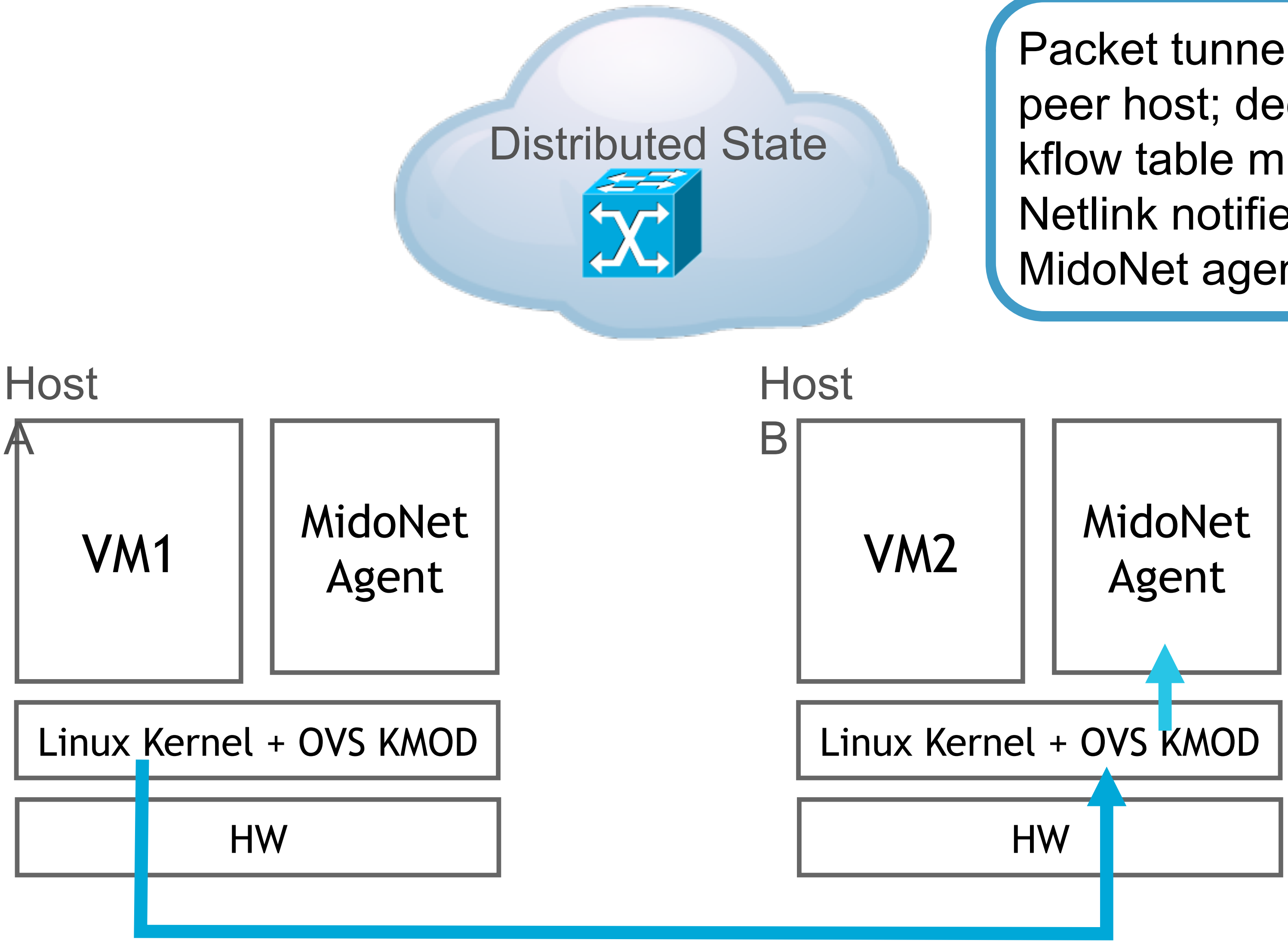
VM2

MidoNet Agent

Linux Kernel + OVS KMOD

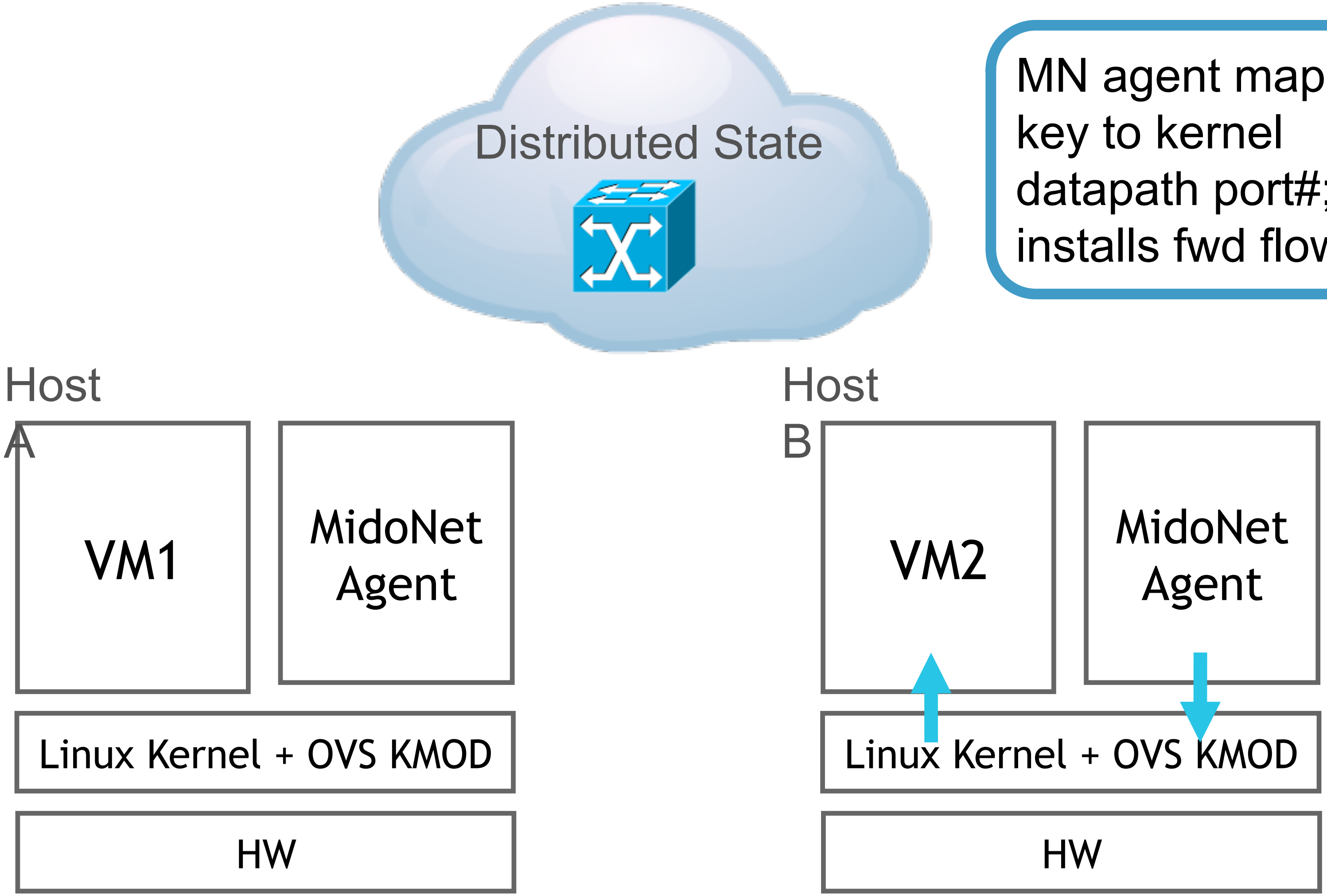
HW

Virtual Networking at the Edge



Packet tunneled to peer host; decap; kflow table miss; Netlink notifies peer MidoNet agent

Virtual Networking at the Edge



Distributed State

Host A

VM1

MidoNet Agent

Linux Kernel + OVS KMOD

HW

Host B

VM2

MidoNet Agent

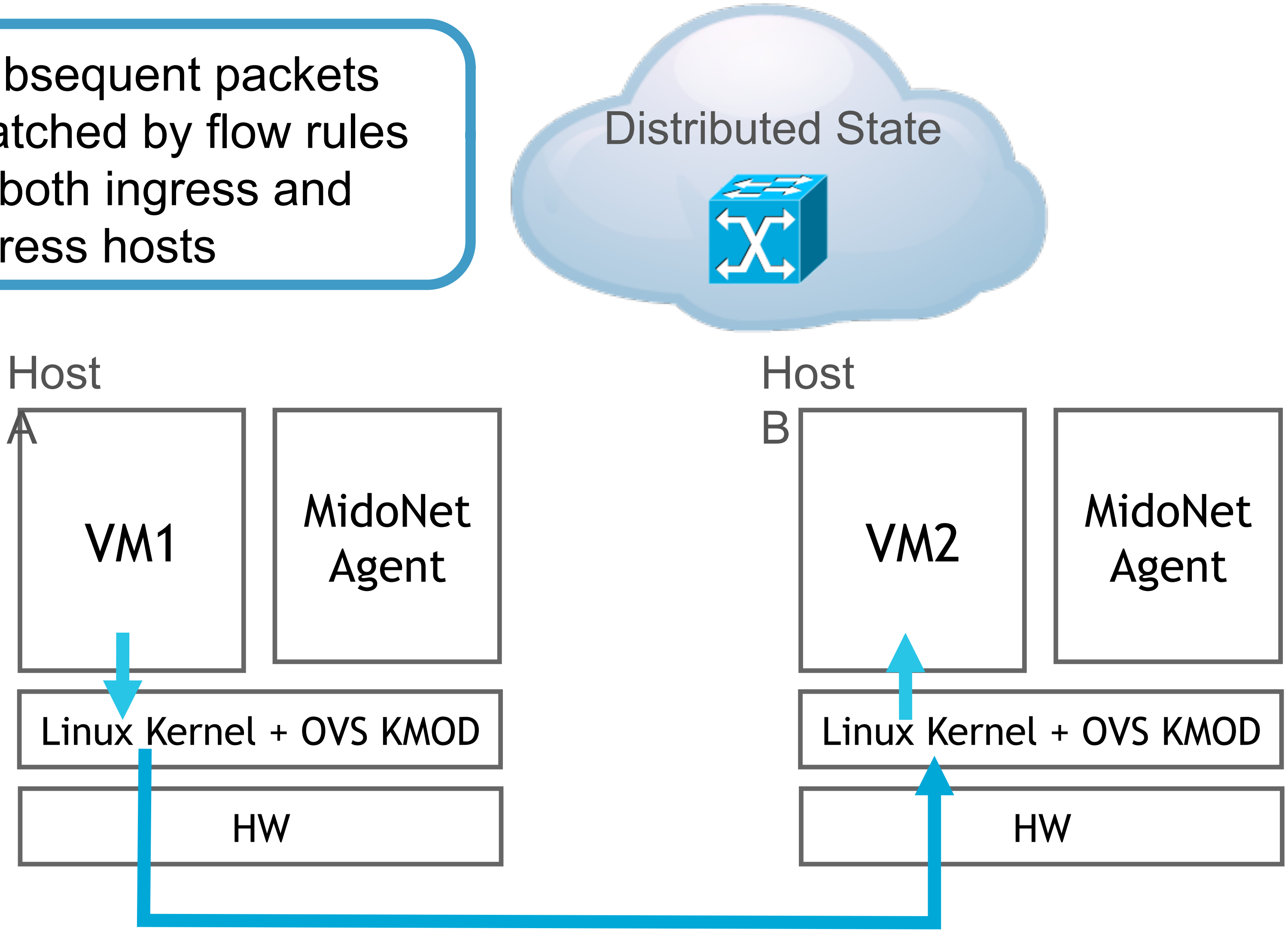
Linux Kernel + OVS KMOD

HW

MN agent maps tun-key to kernel datapath port#; installs fwd flow rule

Virtual Networking at the Edge

Subsequent packets matched by flow rules at both ingress and egress hosts



Distributed State

Host A

VM1

MidoNet Agent

Linux Kernel + OVS KMOD

HW

Host B

VM2

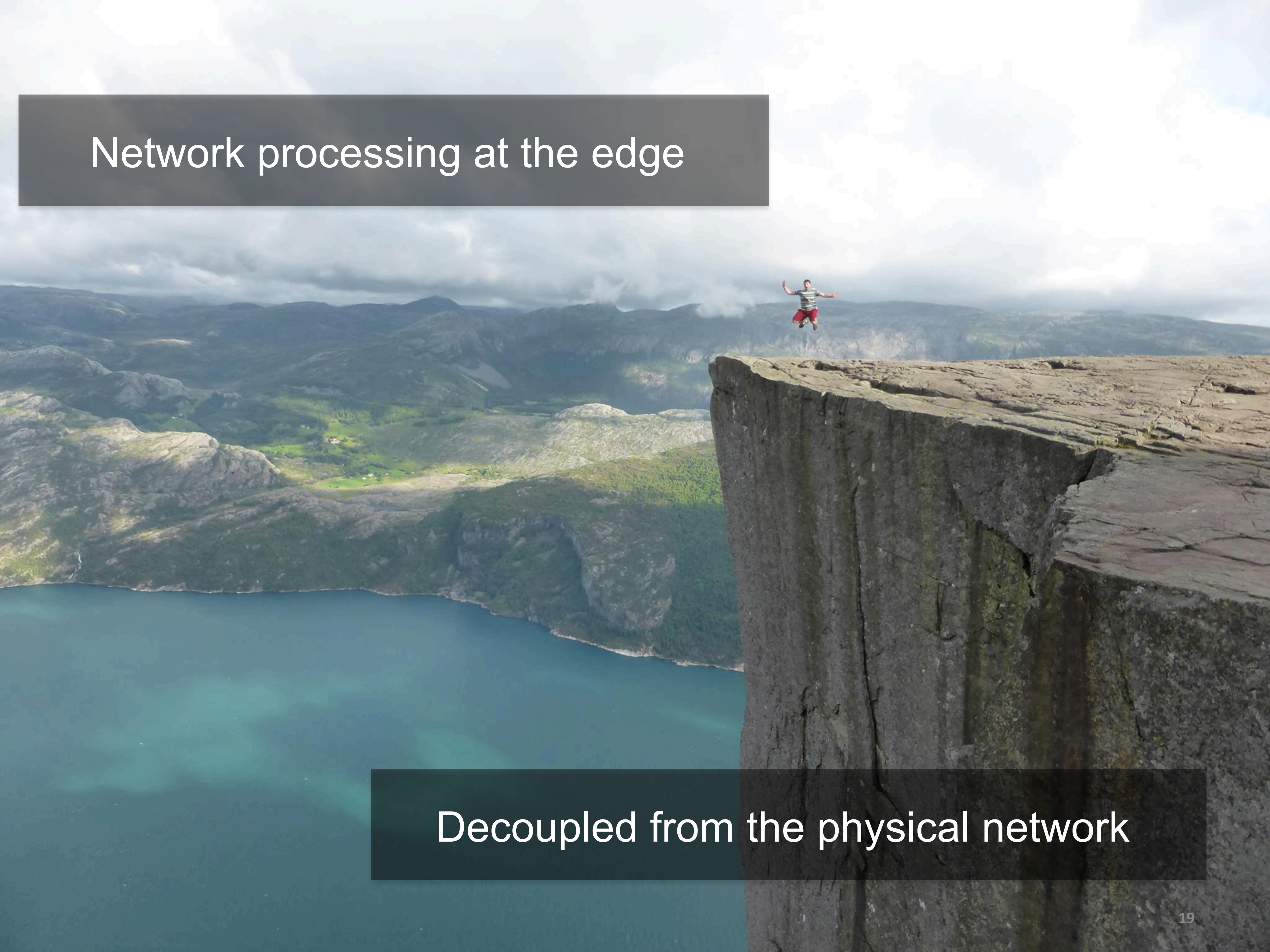
MidoNet Agent

Linux Kernel + OVS KMOD

HW

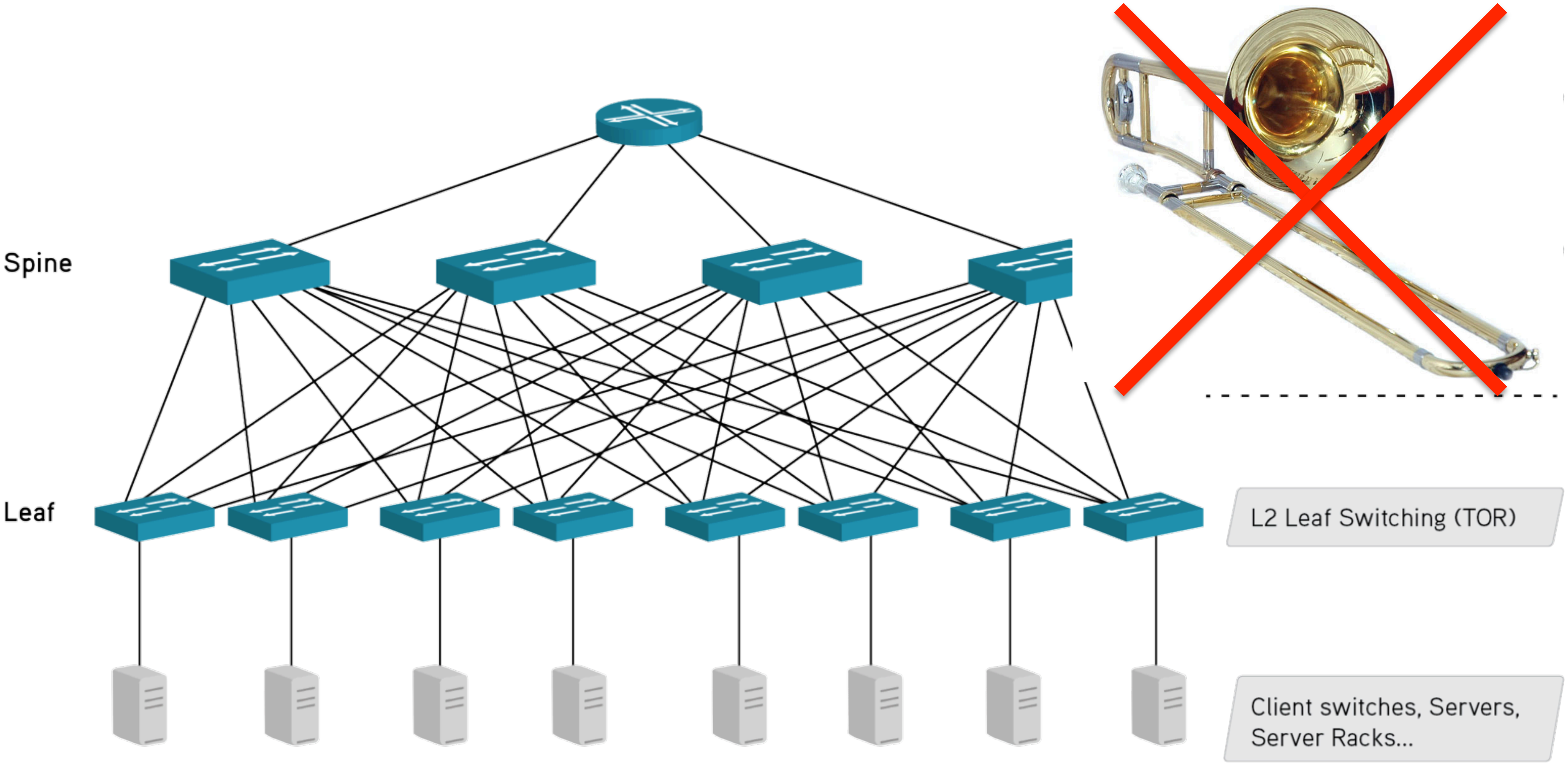
Advantages of the Network Overlay approach

Network processing at the edge



Decoupled from the physical network

Edge Processing avoids traffic “trombones”



Spine

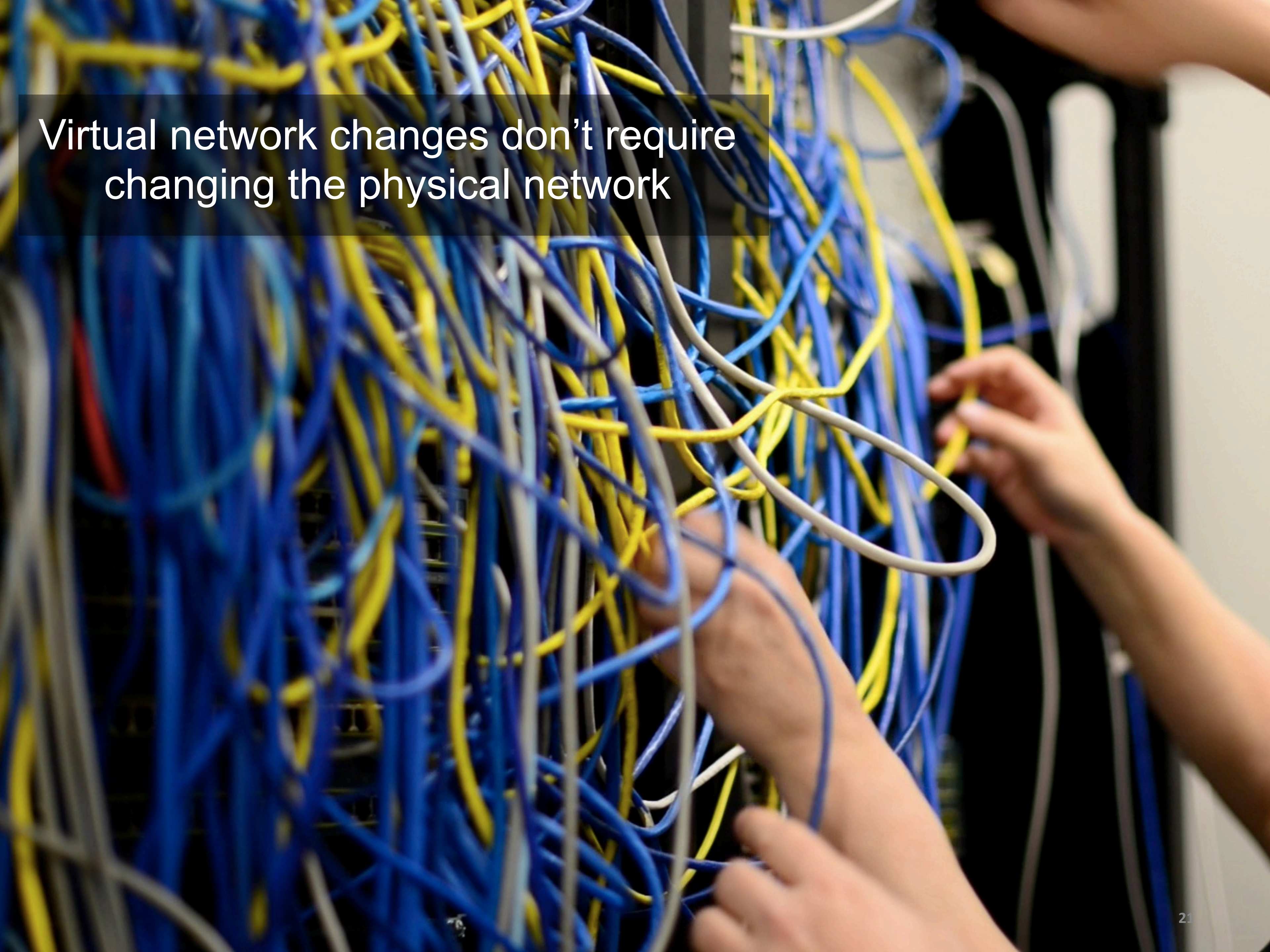
Leaf

L2 Leaf Switching (TOR)

Client switches, Servers, Server Racks...

Image from: http://blogs.ixiacom.com

Virtual network changes don’t require changing the physical network



Summary of Overlay Advantages

- Update physical network without re-orchestrating apps.
- Virtual network software evolves at software time scales.
- The physical network gets simpler (standard, cheap, easy)
- Leaf-and-spine L3+ECMP is a good design for dc physical networks
- Services in software, at the edge, fault-tolerant
- The overlay is easier to debug or troubleshoot
- Less state in the core eases hardware requirements.
- Rapid creation and modification of virtual networks.

What Kernel features support Network Virtualization?

Related kernel features

Flow-programmable datapath (Open vSwitch kmod upstream)

Tunneling options (GRE, VXLAN, STT?)

Rich set of software network interfaces

Network Namespaces

Guest/host paravirtual network drivers + QEMU

Kernel by-pass support

Flow-programmable datapath - OVS

Open vSwitch datapath – and don't forget Netlink channel

Perform arbitrary network computation once and cache the result in the kernel.

Previously limited to microflows (microflows), now have megaflow support for wildcard matching in the kernel.

MidoNet simulates a packet passing through many devices and compute the outcome once, then install that as a flow in the datapath.

We can still gather per-flow metrics and then map them back to per-device-per-packet metrics.

Tunneling Options

- GRE
- VXLAN
- Previously also CAPWAP

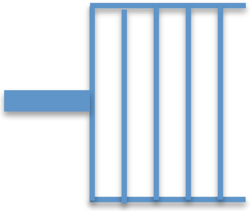
VXLAN allows entropy in the UDP source port, which can be leverage for ECMP path selection. Works well with the spine-and-leaf fabric.

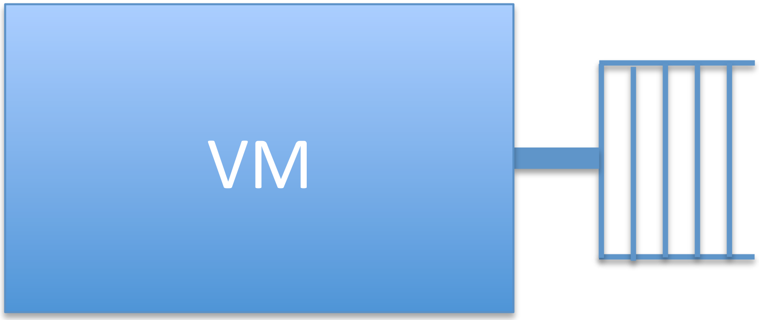
Presumes using the kernel’s network stack, but network cards starting to support VXLAN offload. Still, may need to bypass the kernel altogether.

Virtual Network Interfaces - Tap

A software (simulated) link layer (Ethernet) network device.

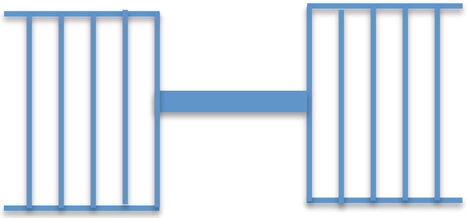
Provides a character-device that a user-space process can open to exchange wholly constructed L2 packets with the kernel.



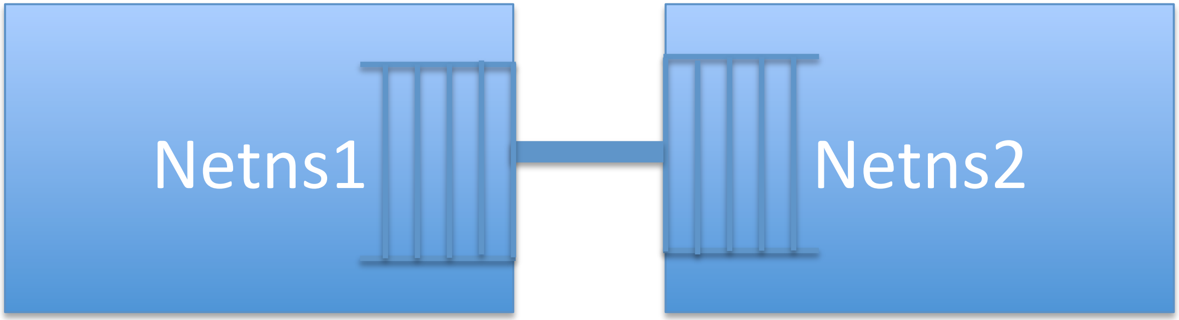


Software Interfaces – Veth Pairs

Two software Ethernet devices connected back to back.

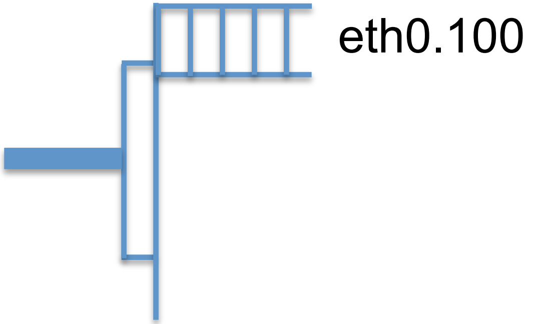


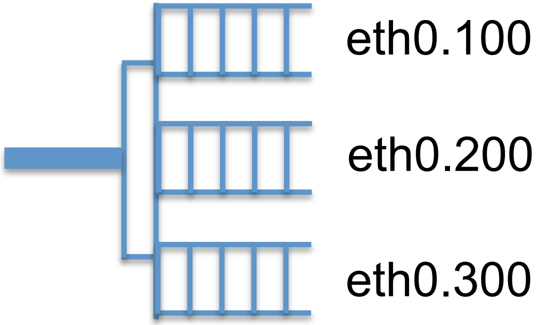
Can be used to interconnect 2 Network Namespaces.

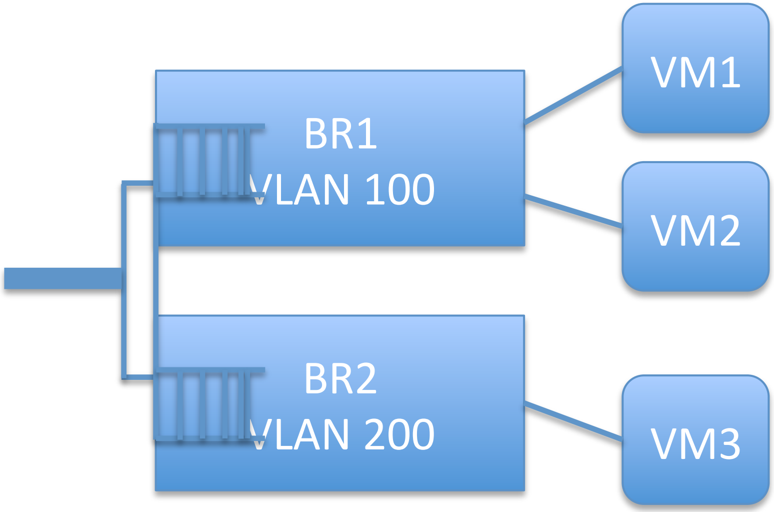


Software Interfaces – vlan

Create network interfaces that use untagged frames from an interface that uses VLAN tagged frames.

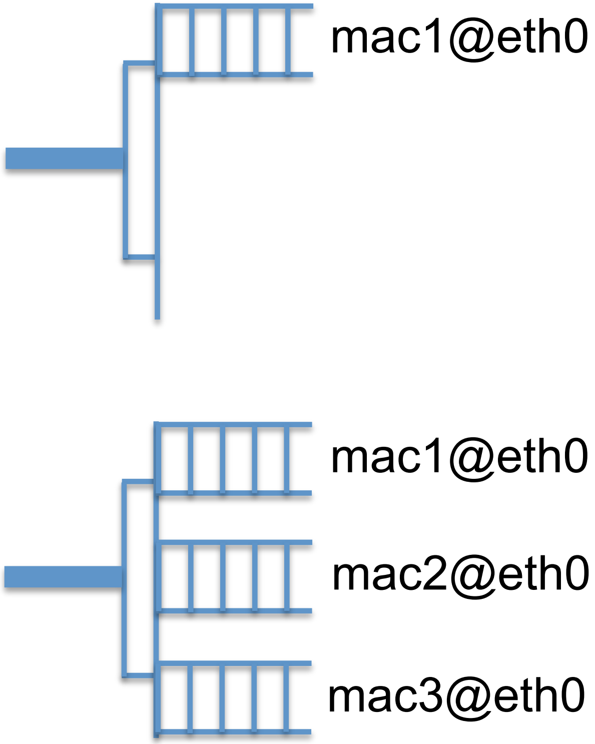






Software Interfaces – macvlan

Give multiple MAC addresses to a single Ethernet interface and view each as a separate virtual Ethernet interface.

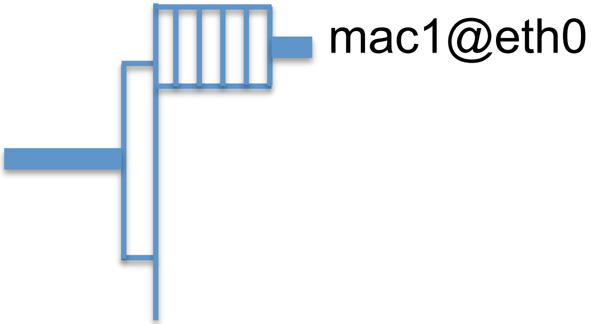


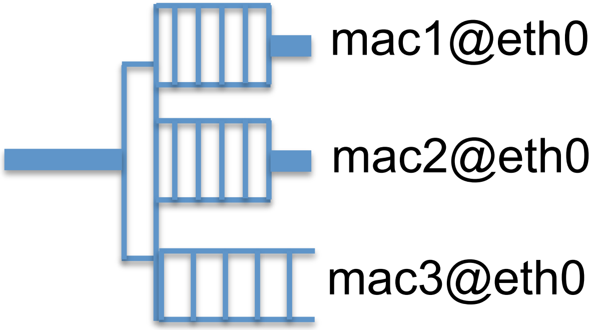
Software Interfaces – macvtap

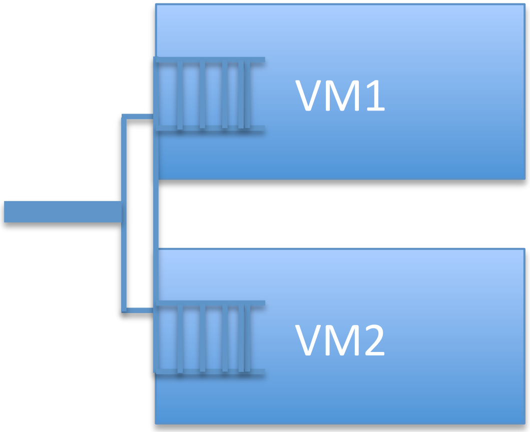
Hybrid macvlan and tap.

Allow multiple VMs direct access to a NIC.

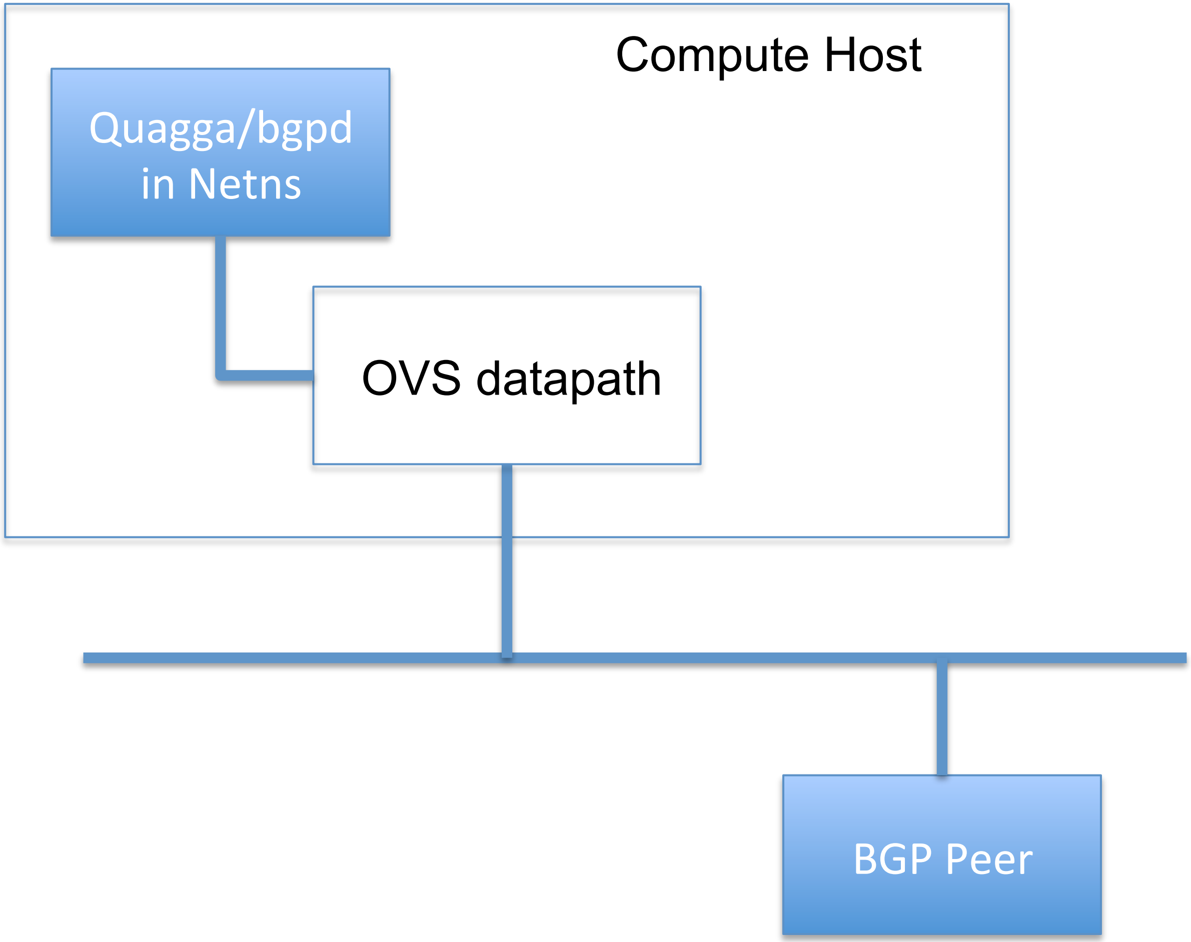
Can still give the host access to the NIC by using macvlan.



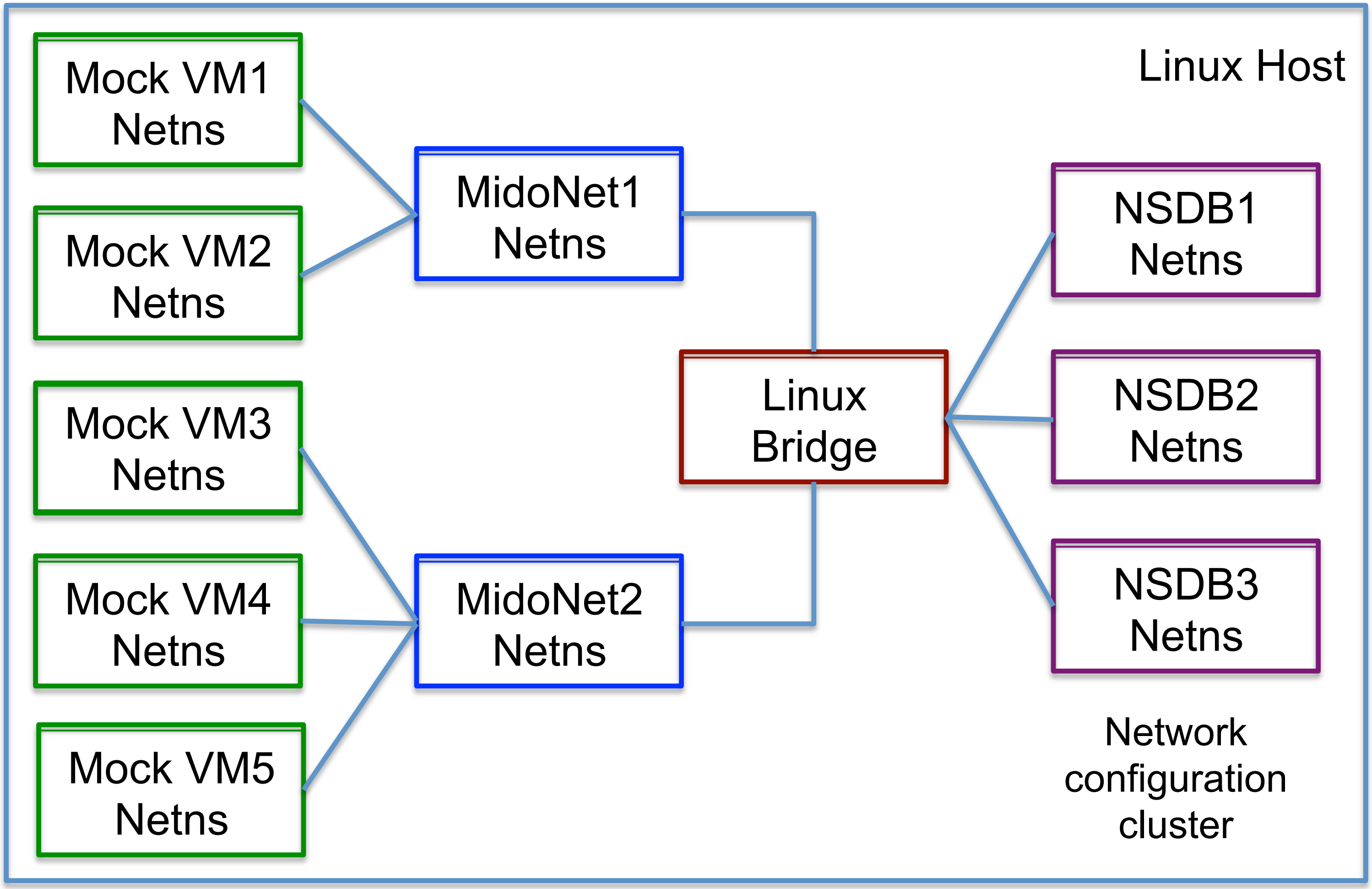




Network Namespaces



Network Namespaces



Networking Drivers

- Earliest approach: unmodified Guest OS, in-kernel device emulation.
- Then: Virtio drivers in the Guest allowed faster packet transfer by reducing system calls.
- QEMU is a user-space process that emulates resources (used by KVM, Xen and others) and implements the Virtio backend.
- Then: Kernel’s vhost-net driver allows by-passing QEMU.
- The bottleneck shifts to the interrupt processing. Need kernel by-pass.

Intel DPDK (also SnabbSwitch & others)

Data **P**lane **D**evelopment **K**it
www.dpdk.org

- By-pass the kernel – interrupt-driven networking is slow
- Run-to-completion processing of packets
- Pin network-processing threads to VMs
- Use non-locking, cache-aligned, shared memory data structures
- Better with guest network drivers – but still Virtio.

Network Virtualization Overlays Today

midokura

vmware®

JUNIPer® NETWORKS

Alcatel·Lucent

PLUMgrid

Thank you and Q&A

Pino de Candia
pino@midokura.com