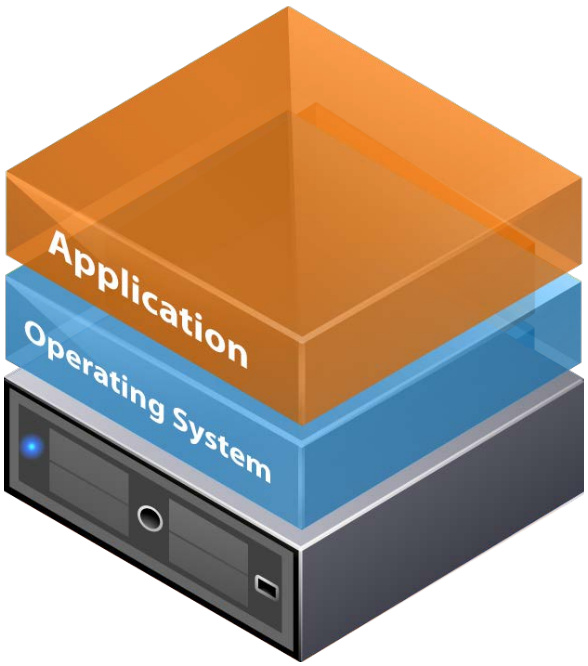
Network Virtualization

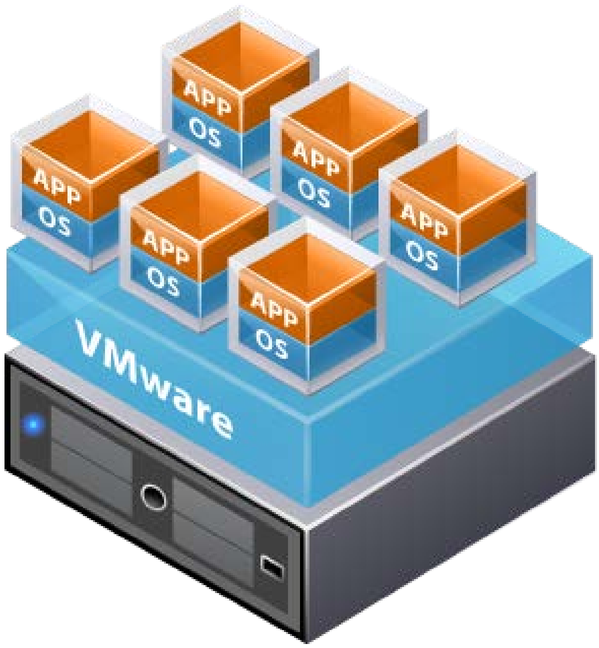
Agenda

- VMware Virtualization Overview
- Access Layer Virtualization
  - Virtual NIC
  - Virtual Standard Switch
  - Uplink
  - Distributed Virtual Switch
- Data Center Network Virtualization
  - vShield
  - VXLAN

vmware

Basic concept :Virtualization





Traditional Architecture

Virtual Architecture

VMware ESXi: 3rd Generation Hypervisor Architecture

VMware GSX (VMware Server)

- Installs as an application
- Runs on a host OS
- Depends on OS for resource management

VMware ESX

- Installs "bare metal"
- Complete HW management
- Relies on a Linux OS (Service Console) for running agents and scripting

VMware ESXi

- Installs "bare metal"
- Complete HW management
- Management tasks are moved outside of the hypervisor (3rd party integration via APIs and CIM; scripting via vRCLI)

2001 — **2003** — **2007**

vmware

Key Properties of Virtualization

Partitioning

- Run multiple operating systems on one physical machine
- Divide system resources between virtual machines

Isolation

- Fault and security isolation at the hardware level
- Advanced resource controls preserve performance

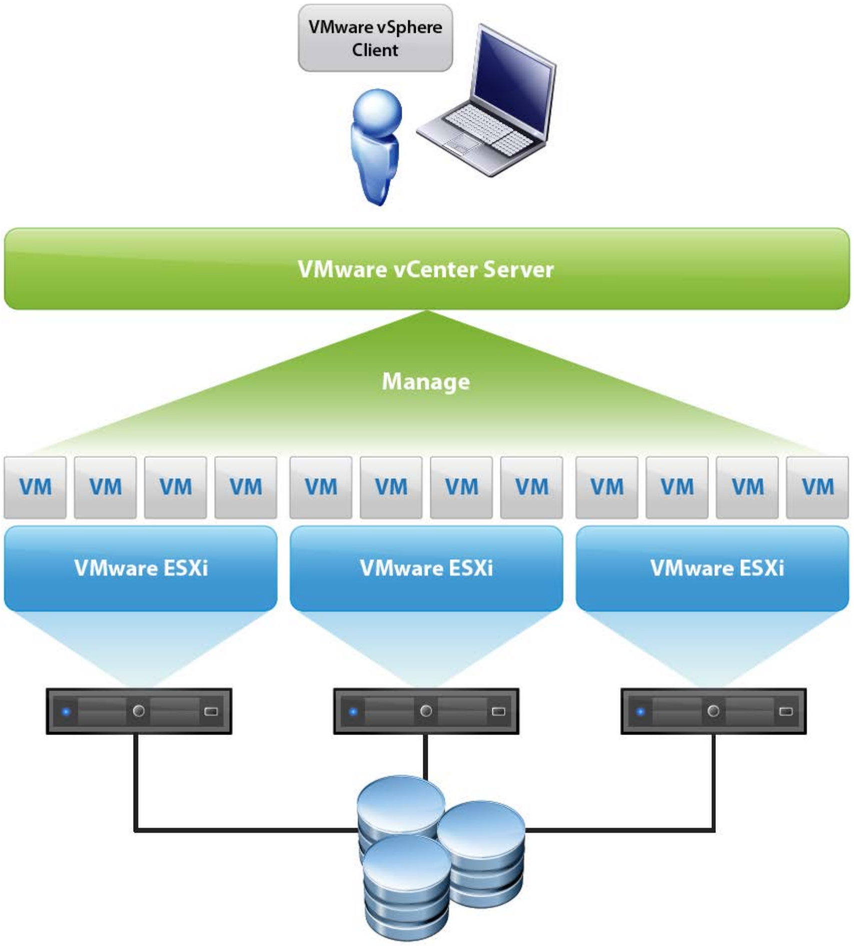
Encapsulation

- Entire state of the virtual machine can be saved to files
- Move and copy virtual machines as easily as moving and copying files

Hardware Independence

- Provision or migrate any virtual machine to any similar or different physical server

VMware vSphere Deployment Architecture



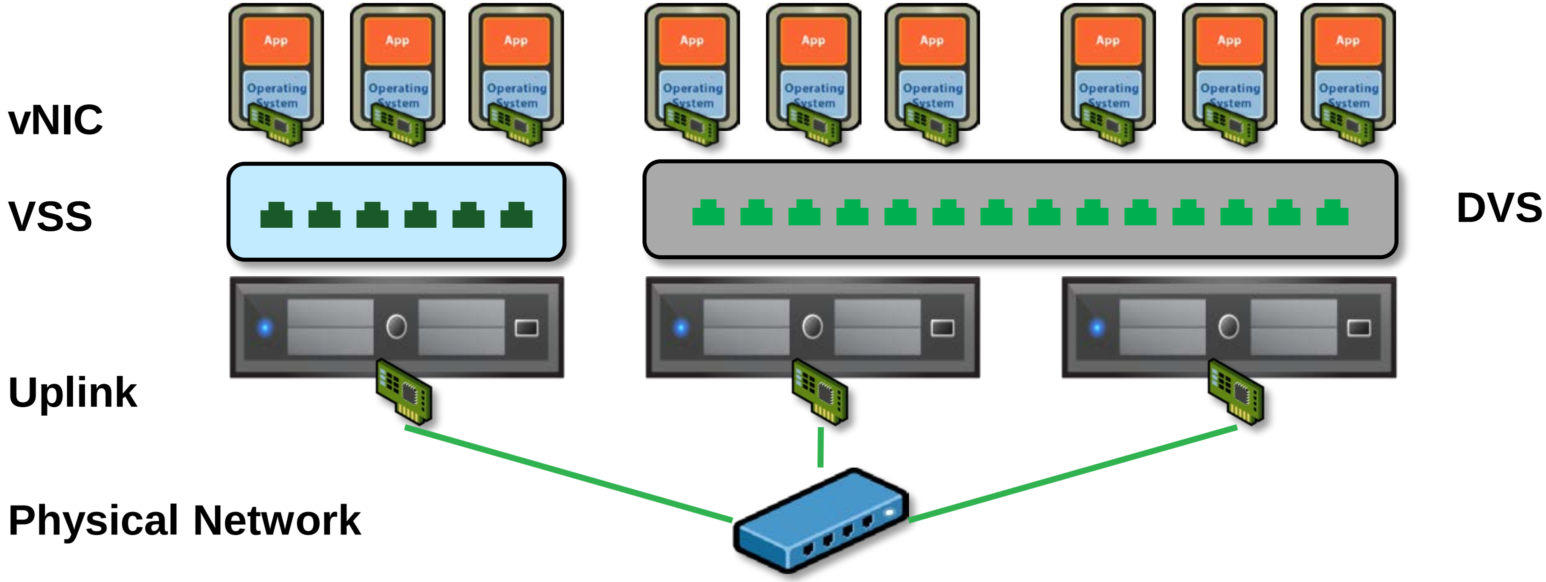
- Deploy ESXi on each host
- Add vCenter Server to Centrally manage ESXi hosts
- Upgrade license file to vSphere

Agenda

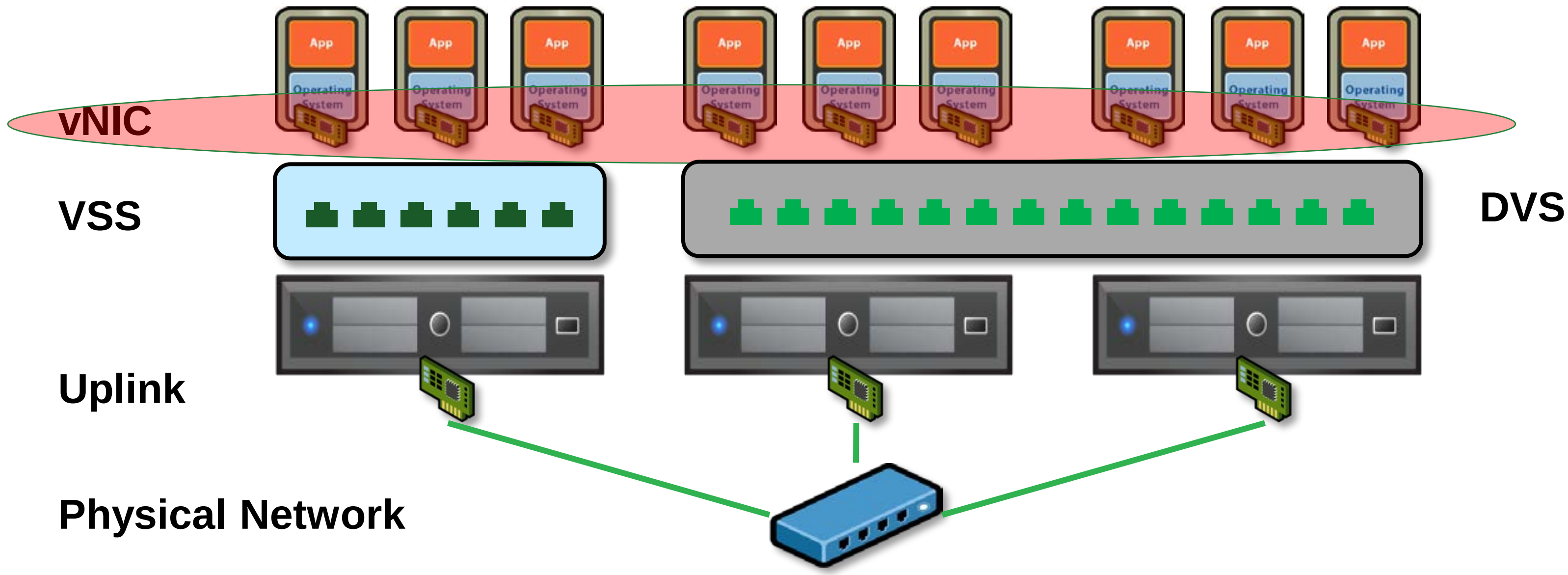
- Virtualization
- Access Layer Virtualization
  - Virtual NIC
  - Virtual Standard Switch
  - Uplink
  - Distributed Virtual Switch
- Data Center Network Virtualization
  - vShield
  - VXLAN

Access Layer Virtualization

- Traditional access layer switch is moved into hypervisor
  - Virtual NICs (vNIC) are added to VMs
  - Virtual Standard Switch (VSS) is introduced between VMs and physical network
  - Uplink layer is added to connect VSS to upstream physical switches
  - Distributed Virtual Switch (DVS) is added to support distributed configuration



Access Layer Virtualization - Virtual NIC



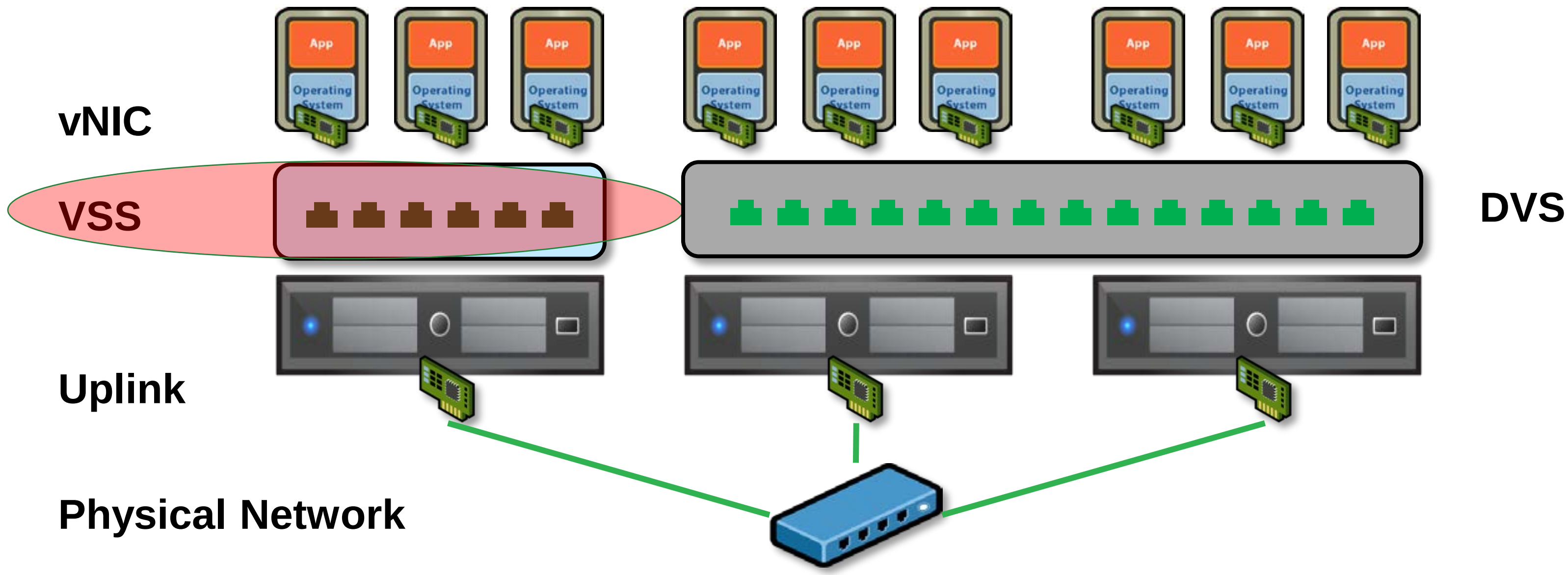
Virtual NICs

- Emulated layer 2 device used to connect to the vSwitch
  - Each virtual NIC has a MAC address of its own does address based filtering
- No need for implementation of a PHY (Physical Layer)
  - No auto-negotiation
  - Speed/Duplex/Link are irrelevant
    - Ignore speed/duplex reported in the guest OS
  - Actual speed of operation depends on the CPU cycles available and speed of the uplinks.
- Different types of Virtual NICs
  - Virtual adapter for VMs
    - VLance, E1000, vmxnet2/vmxnet3(vmware)
  - Vswif for Service console(not in ESXi)
  - Vmknic for VMKernel

Virtual NIC Hardware Offload

- Delay process some hardware offloading capabilities to physical NICs or process them with software if physical NICs do not support them at uplink layer
  - VLAN
  - TSO
  - LRO
  - Checksum offload

Access Layer Virtualization – Virtual Standard Switch (VSS)



Virtual Standard Switch (VSS)

- Software implementation of an Ethernet switch
- How is it similar to a physical switch?
  - Does MAC address based forwarding
  - Provides standard VLAN segmentation
  - Configurable
- How is it different?
  - Does not need to learn MAC addresses
    - It knows the MAC addresses of the virtual NICs connecting to it
  - Single tier topology
    - No need to participate in Spanning Tree Protocol
  - Overall fewer bells and whistles, but provides some unique features

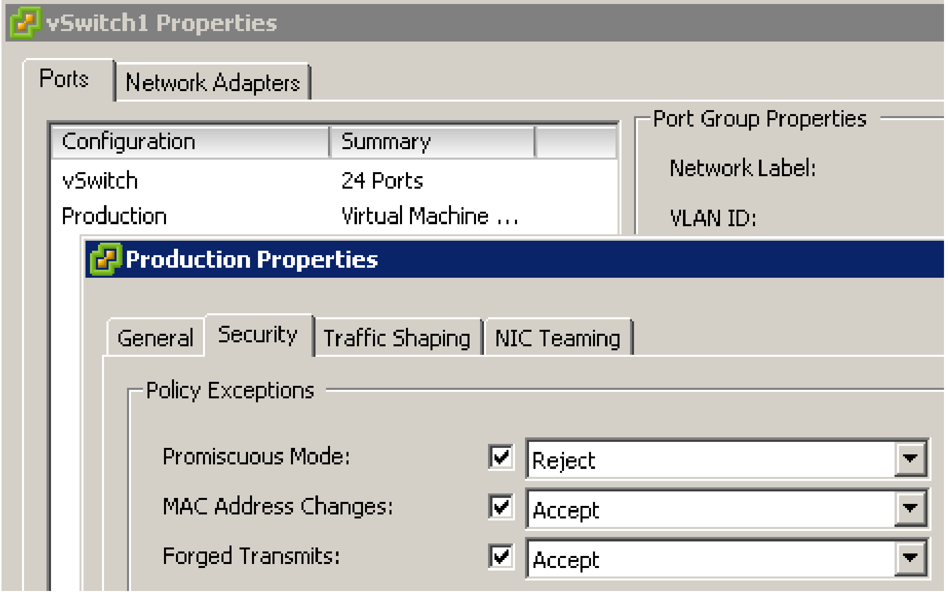
Portgroups

- Portgroups are configuration templates for ports on the vSwitch
  - Efficient way to specify the type of network connectivity needed by a VM
- Portgroups specify
  - VLAN Configuration
  - Teaming policy
  - Layer 2 security policies
  - Traffic shaping parameters
- Portgroups are not VLANs
  - Portgroups ***do not*** segment the vSwitch into separate broadcast domains unless they have different VLAN Ids

vmware

Implications of L2 Security Policies

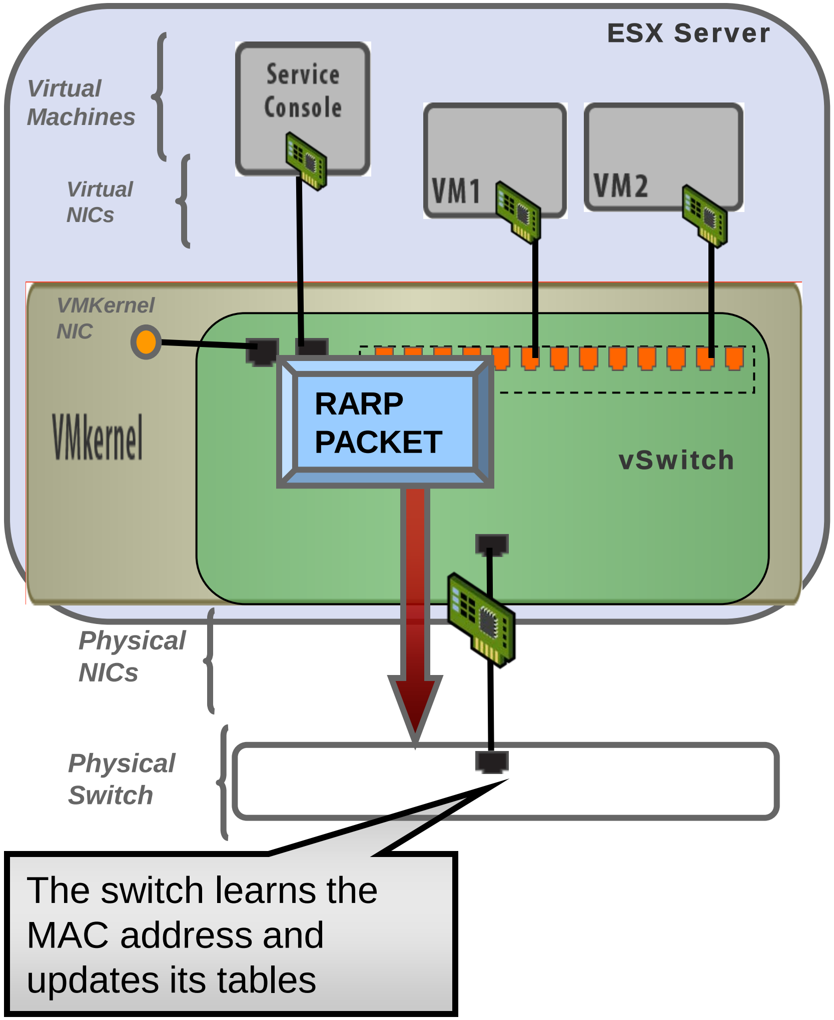
- Promiscuous Mode
  - If allowed, guest receives all frames on the vSwitch
  - Some applications need promiscuous mode
    - Network sniffers
    - Intrusion detection systems
- MAC Address Change
  - If allowed, malicious guests can spoof MAC addresses



- Forged Transmits
  - If allowed, malicious guests can cause MAC Flooding and/or spoofing
- Security settings should reflect application requirements
  - Some applications might need to forge or change MAC addresses
    - E.g.: Microsoft NLB in unicast mode works by forging MAC addresses.

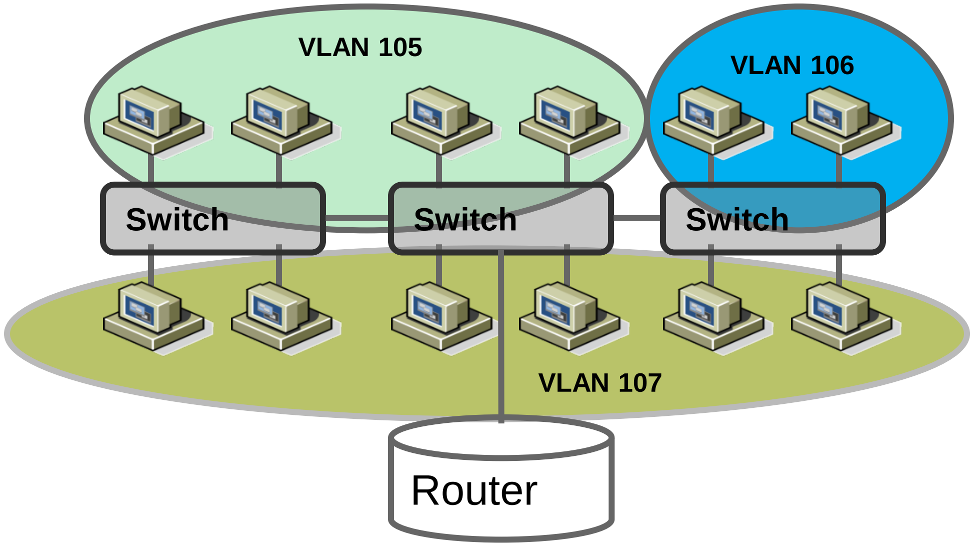
Notify Switch

- Client MAC address is notified to the switch via RARP packet
- Allows the physical switch to learn the MAC address of the client immediately
- Why RARP?
  - L2 broadcast reaches all switches
  - L3 information not required
- Switch notified whenever
  - New client comes into existence
  - MAC address changes
  - Teaming status changes
- Settings should reflect application requirements

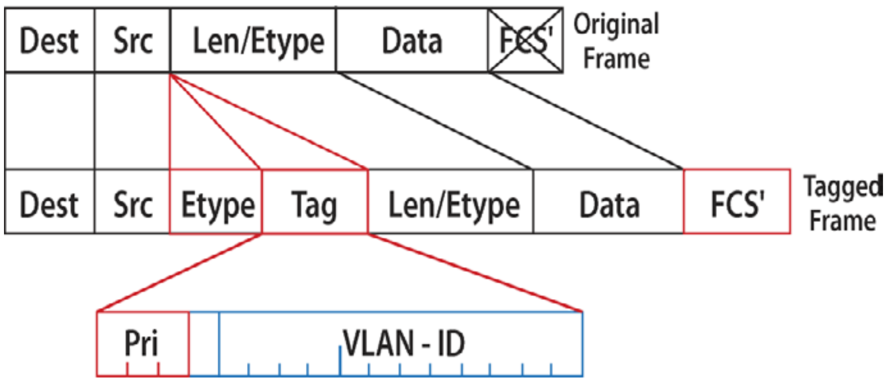


VLAN

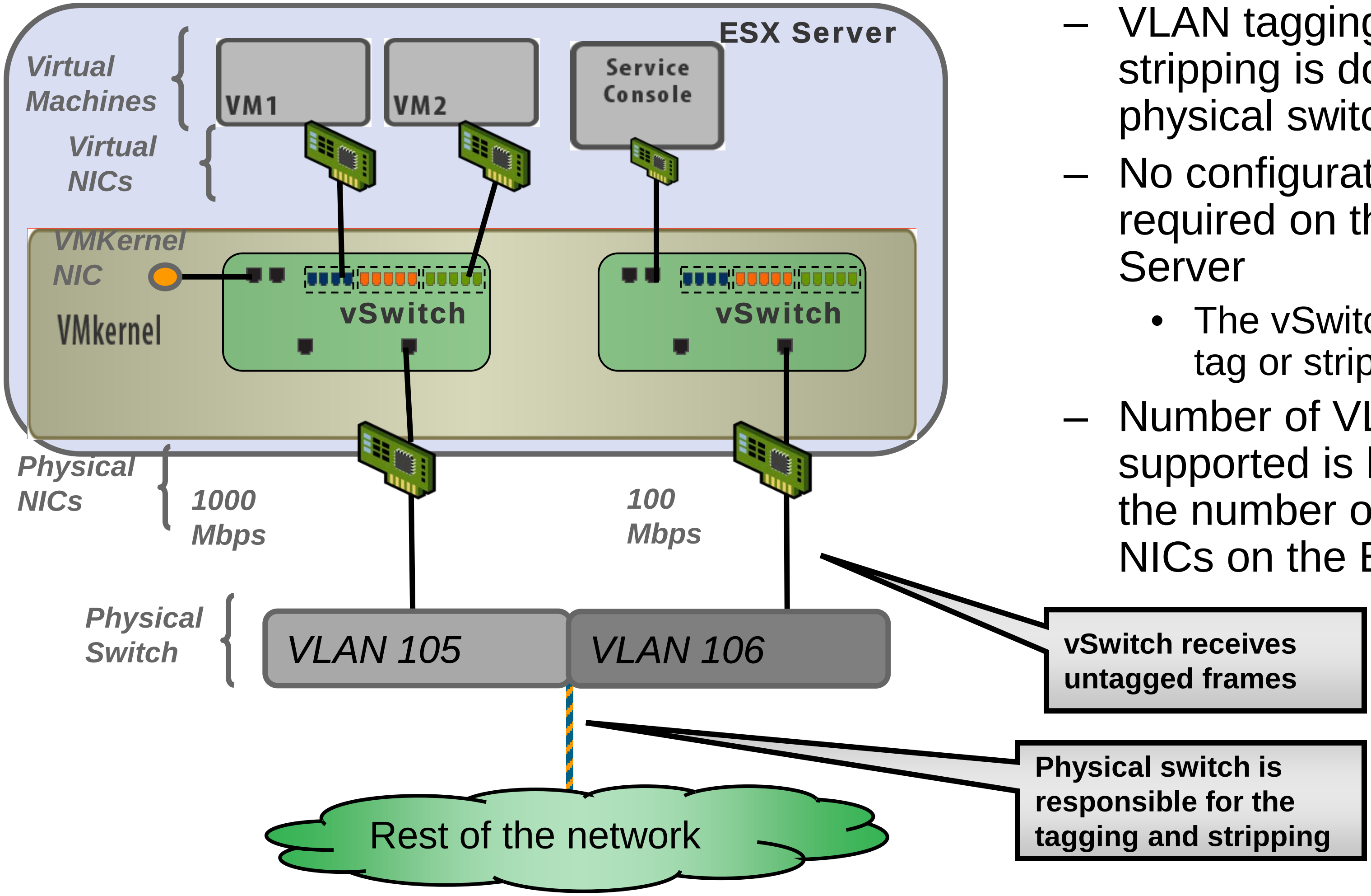
- Carves out distinct layer 2 broadcast domains
- VSS supports IEEE 802.1Q format
  - 4 byte VLAN tag inserted in the frame
- Three types of VLAN configurations
  - External Switch Tagging
  - Virtual Switch Tagging
  - Virtual Guest Tagging



802.1Q Frame Format:

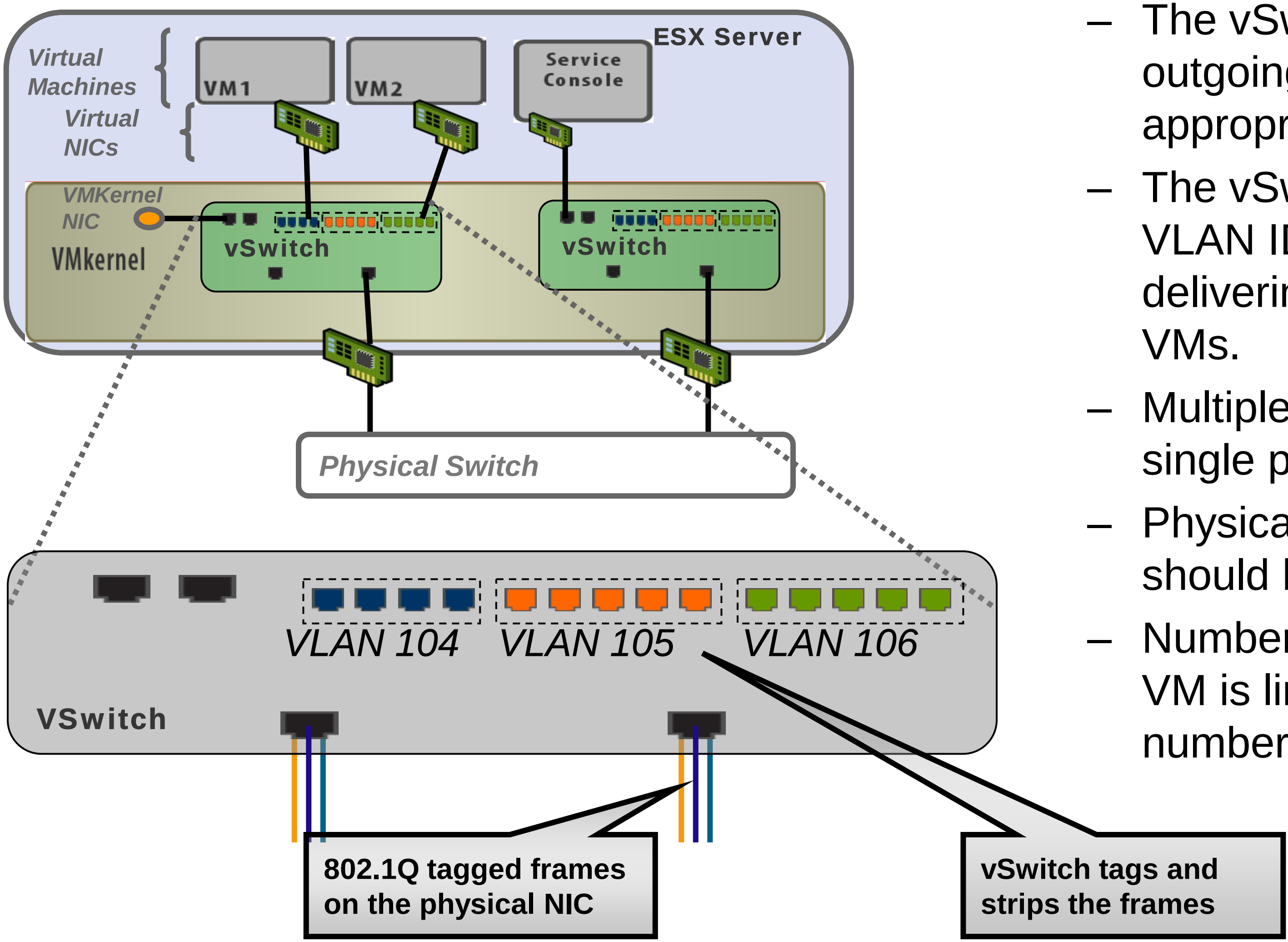


External Switch Tagging



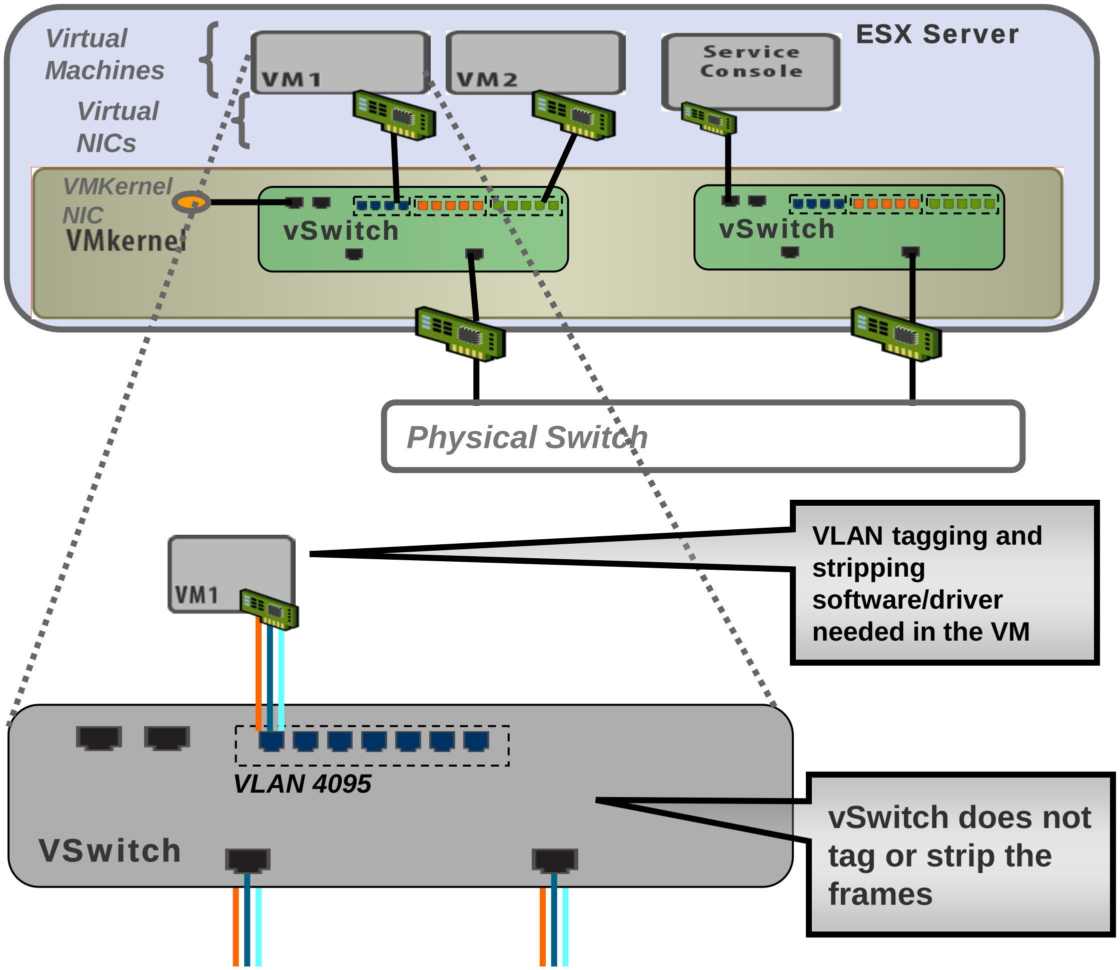
- VLAN tagging and stripping is done by the physical switch
- No configuration required on the ESX Server
  - The vSwitch does not tag or strip the frames
- Number of VLANs supported is limited to the number of physical NICs on the ESX server

Virtual Switch Tagging



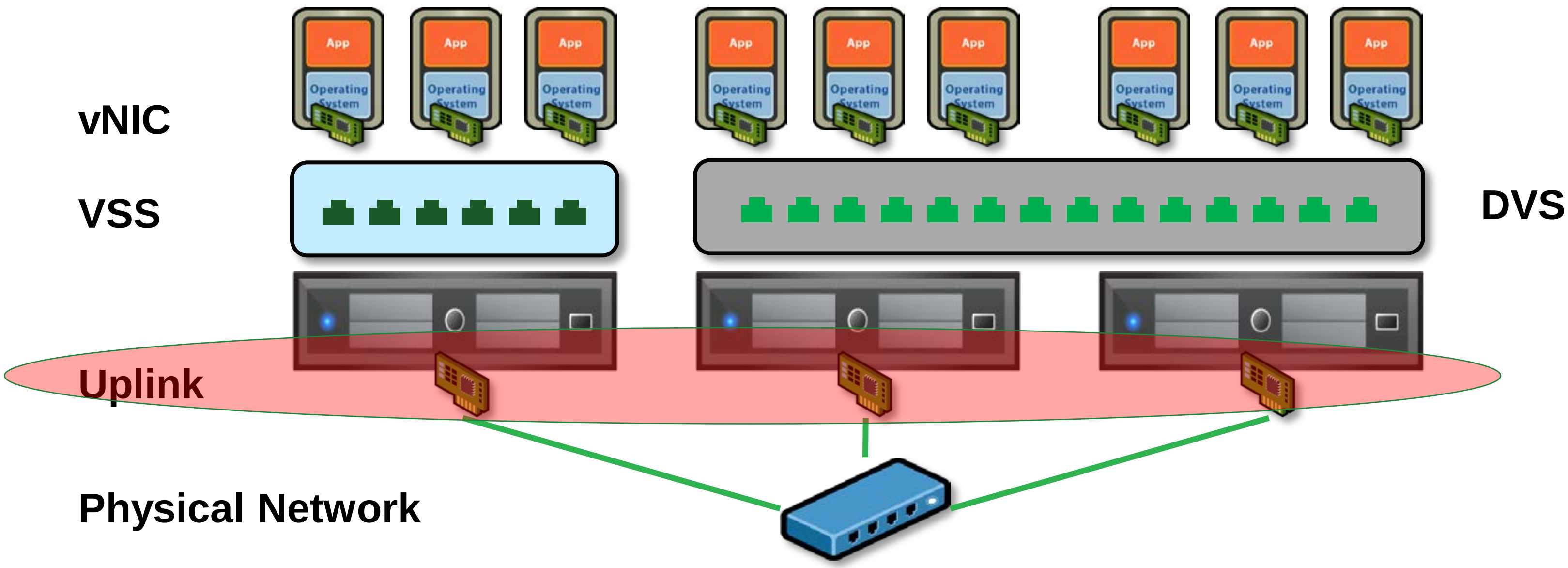
- The vSwitch tags outgoing frames with the appropriate VLAN ID.
- The vSwitch strips any VLAN IDs before delivering frames to VMs.
- Multiple VLAN IDs on single physical NIC.
- Physical switch port should be a trunk port.
- Number of VLANs per VM is limited to the number of vNICs.

Virtual Guest Tagging



- Portgroup VLAN ID is set to 4095
- Tagging and stripping of VLAN IDs happens in the guest VM
- In VGT mode guest can send/receive any VLAN tagged frame
- Number of VLANs per guest is not limited to the number of vNICs

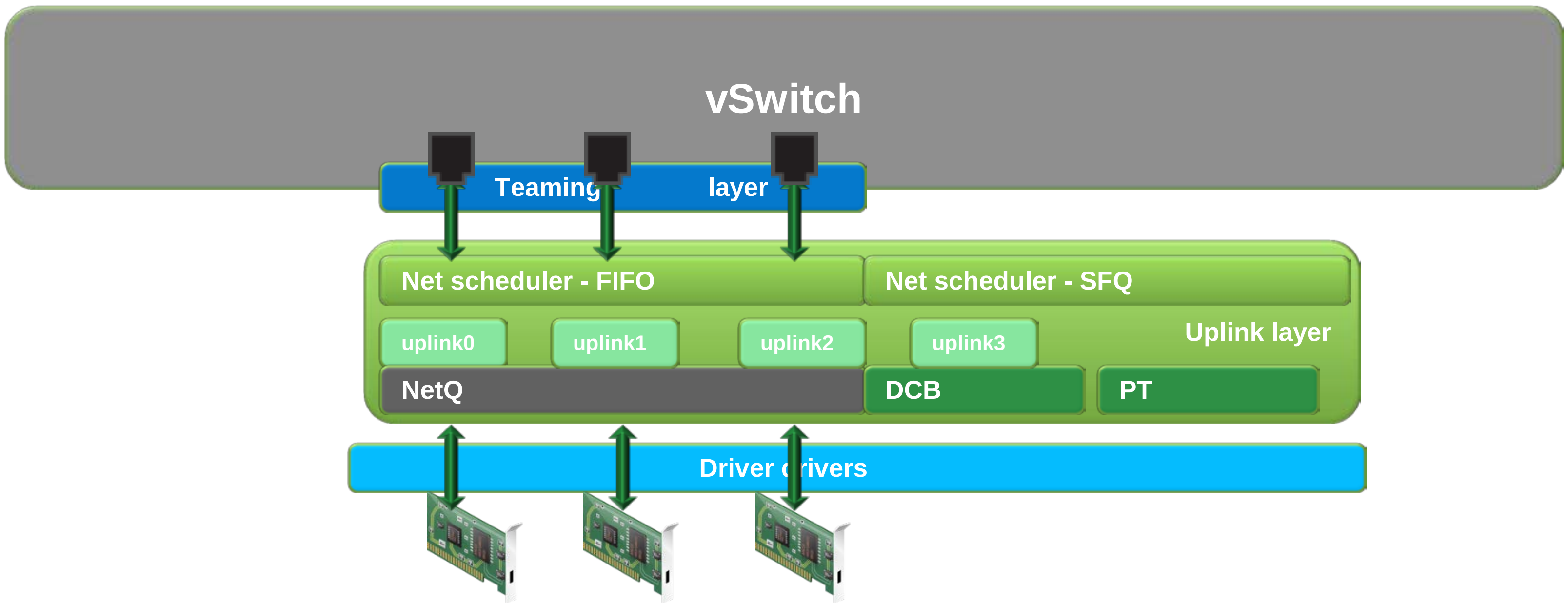
Access Layer Virtualization – Uplink



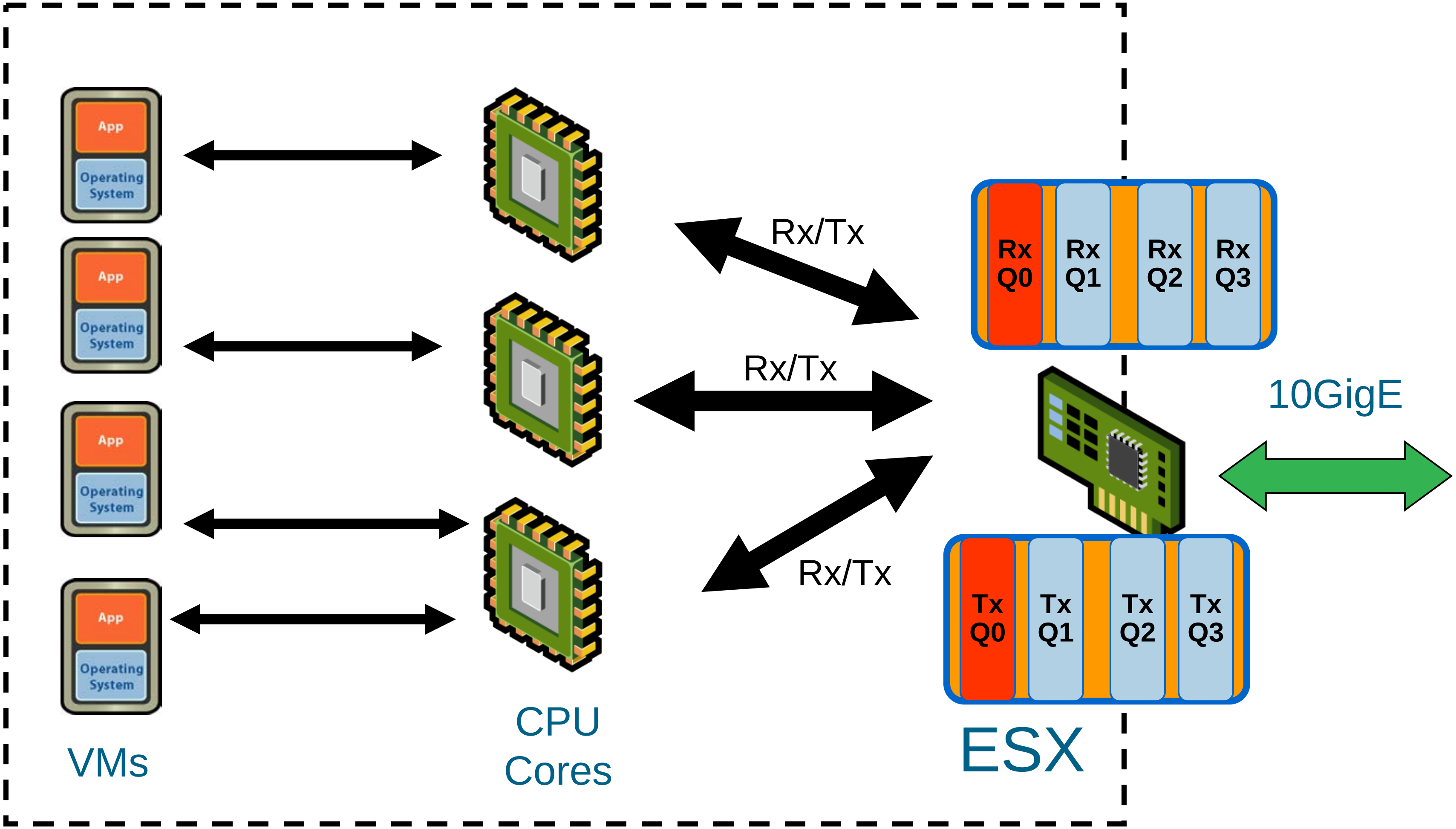
Uplink Layer

Uplinks :

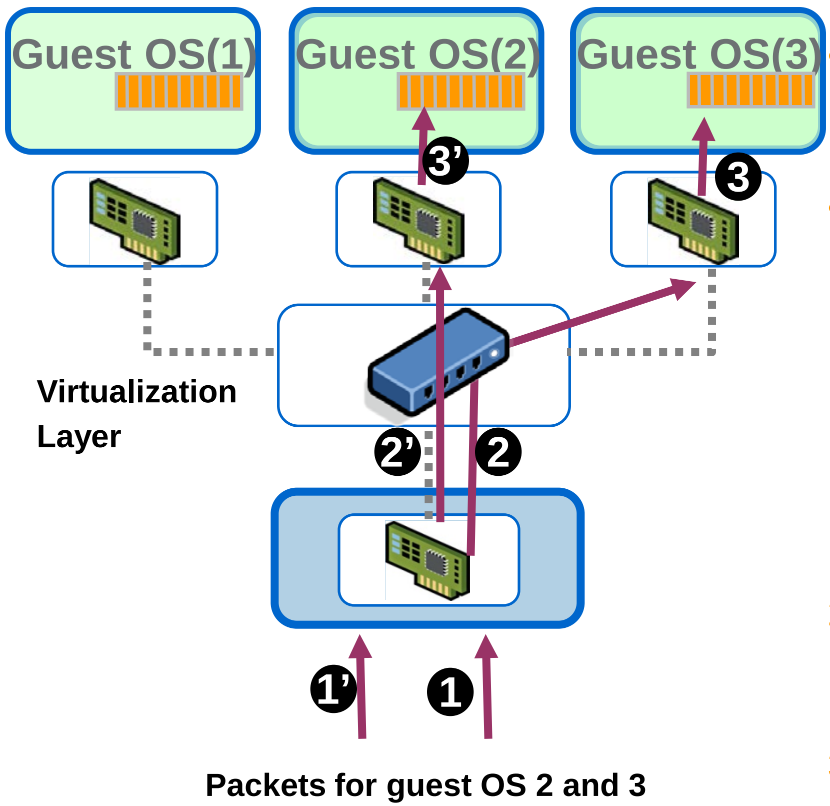
Physical Ethernet adapters serve as bridges between virtual and physical networks. In VMware Infrastructure, they are called uplinks, and the virtual ports connected to them are called uplink ports. A single host may have a maximum of uplinks, which may be on one switch or distributed among a number of Switches.



NetQ – (Multiple Queue/Ring)



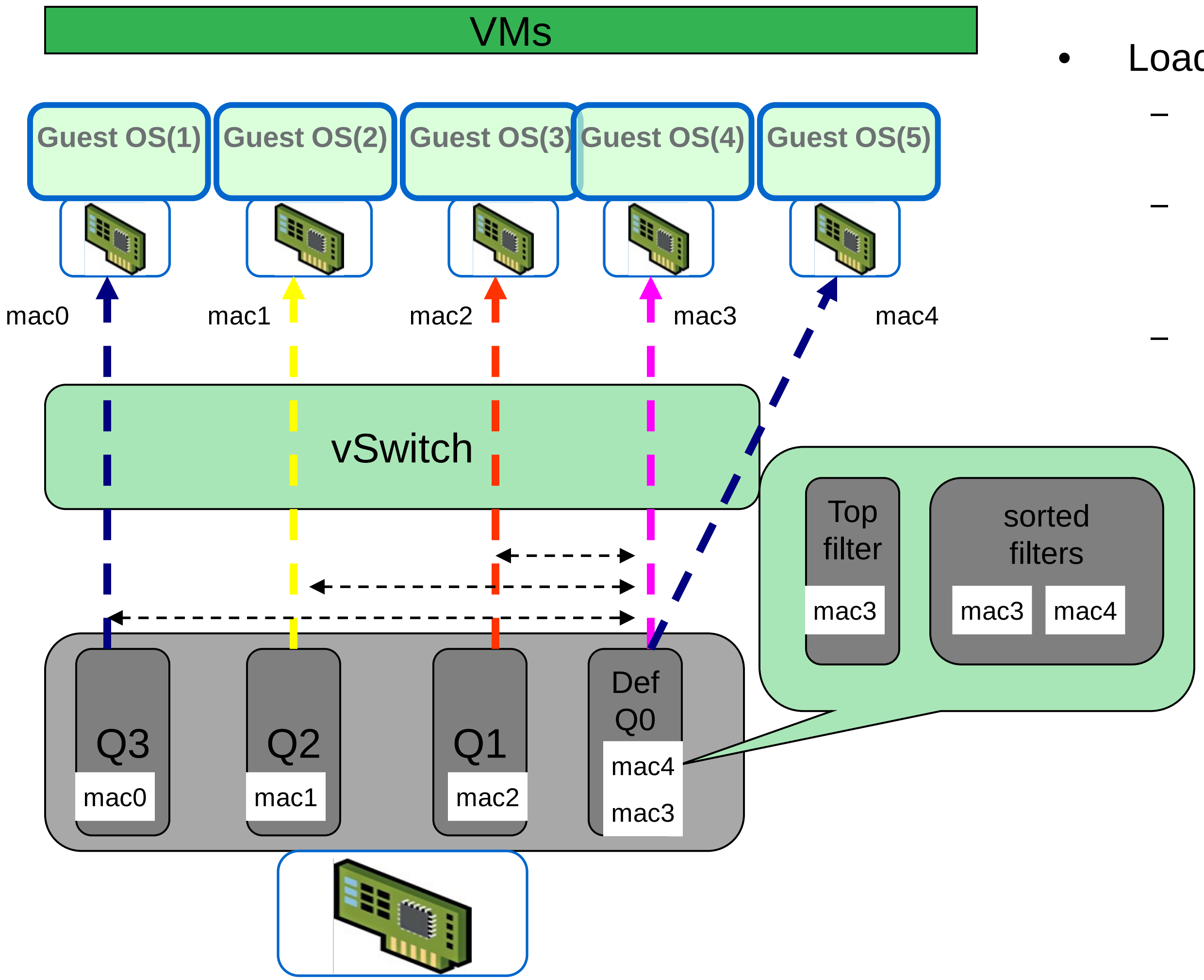
Netqueue - Rx



- Program the queue with Guest MAC address
- Assign a unique MSI-X interrupt

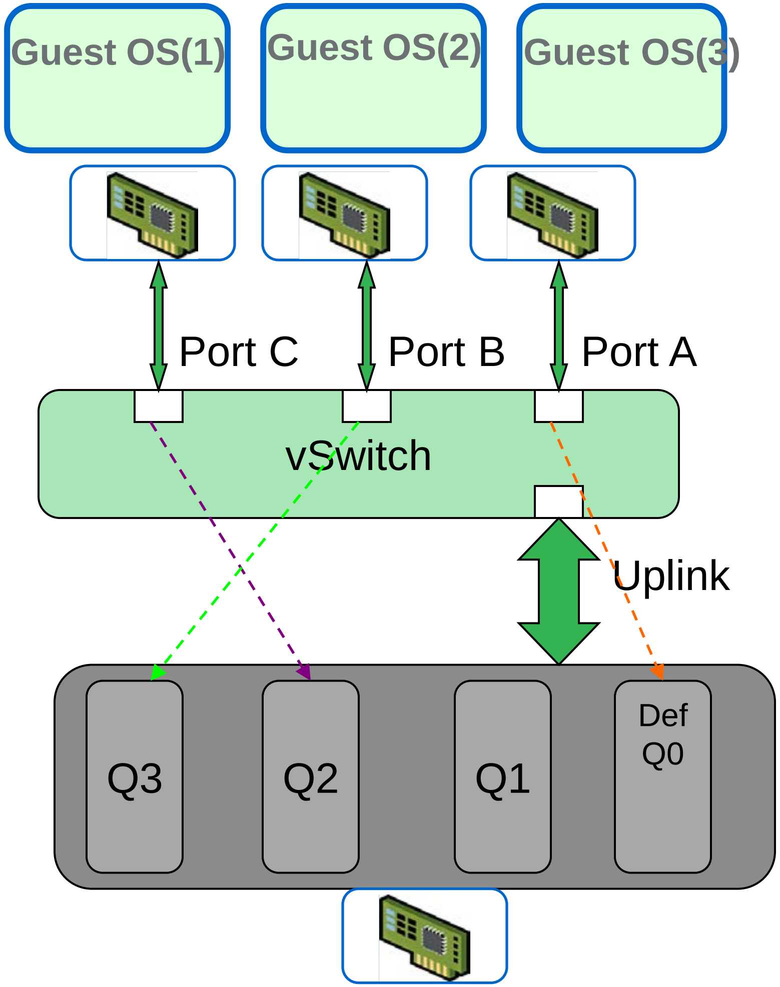
1. NIC classifies the packet based on MAC address
   - DMA to memory for the queue
2. Vmkernel delivers the packet to virtual device
3. Virtual device posts virtual interrupt to the guest OS

Netqueue – Rx – Load Balance



- Load balancing
  - Limited queues (default queue for rest of the VMs)
  - Reprogram the queues based on the load on the VMs every 5s by default
  - Queues: VMs = 1: N (N is multiple filter per queue)

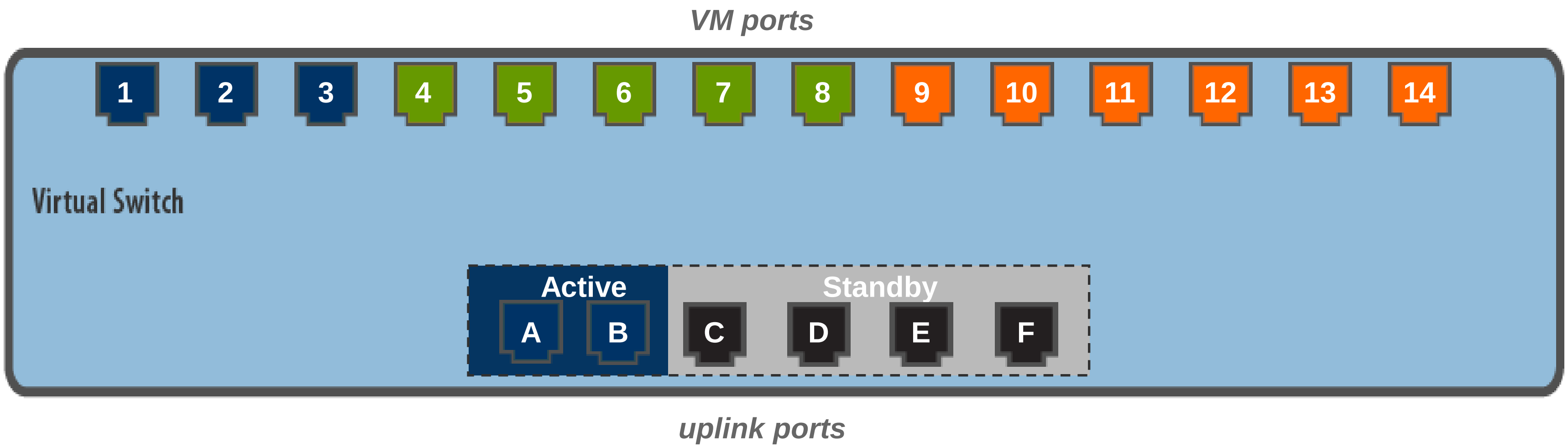
Netqueue - Tx



- Tx queue number is based on Teaming policy.
- Tx load balancer runs every 5s.
- every 5s * 40(DecayCounter), reset.
- Broadcast packets only send in default queue.

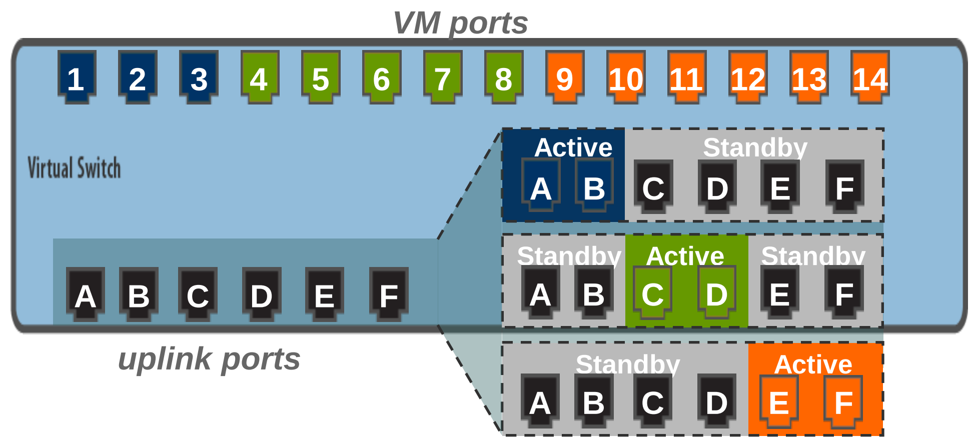
Teaming

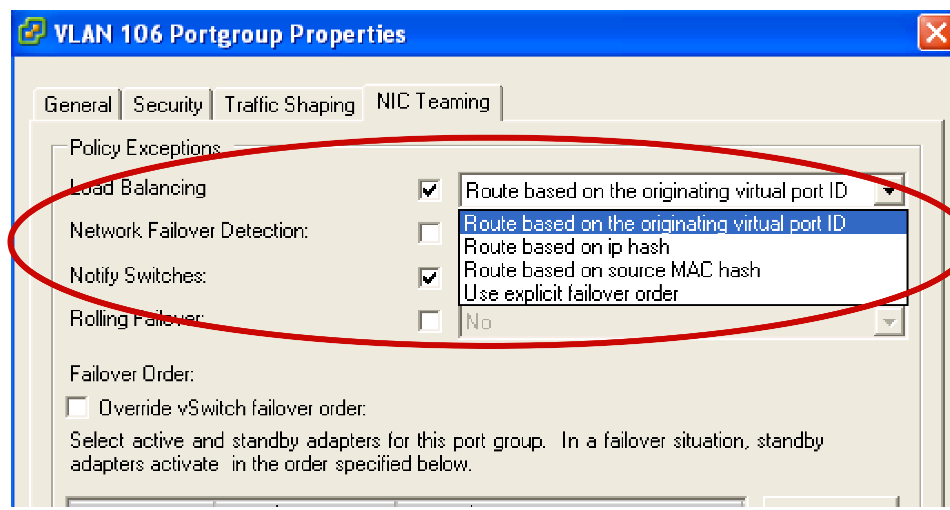
- Allows for multiple active NICs to be used in a teaming configuration.
- User can choose the policy for distribution of traffic across the NICs.
- Standby uplinks replace active uplinks when active uplinks fail to meet user specified criteria



Portgroup Based Teaming Configuration

- Teaming policy attributes can vary by portgroups on a single vSwitch
- Load balancing policies
  - Originating Port ID based
  - IP hash based
  - Source MAC address based
  - Explicit failover order
  - Load balance based

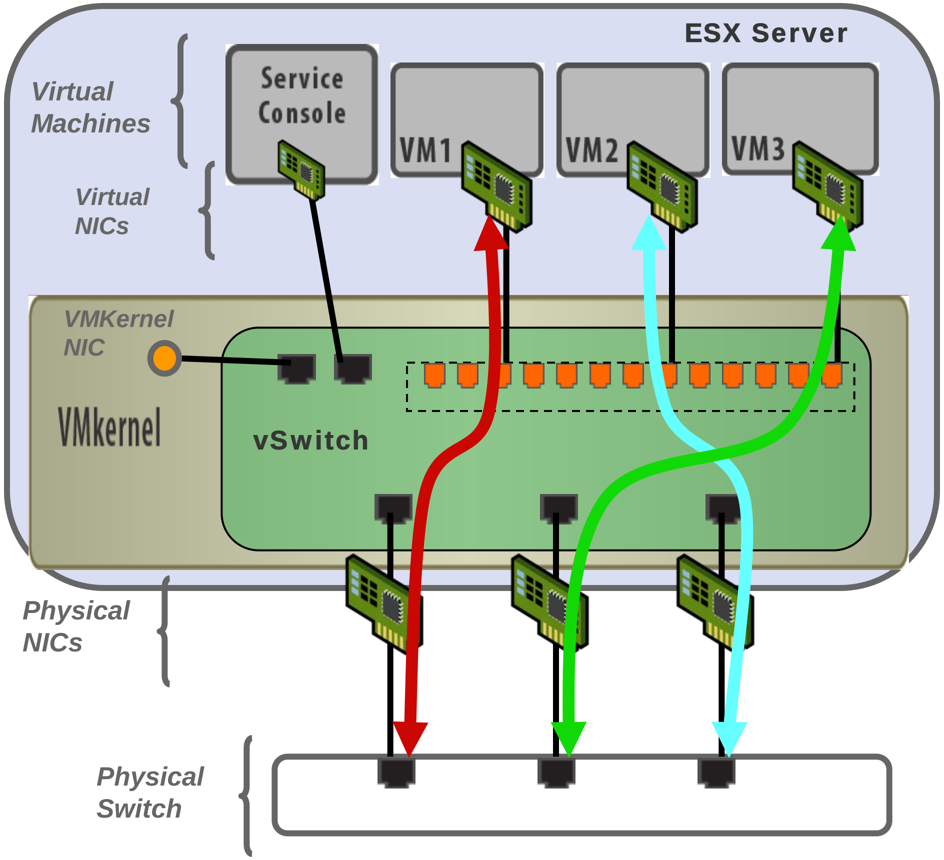




Teaming Based on Port ID or MAC Hash

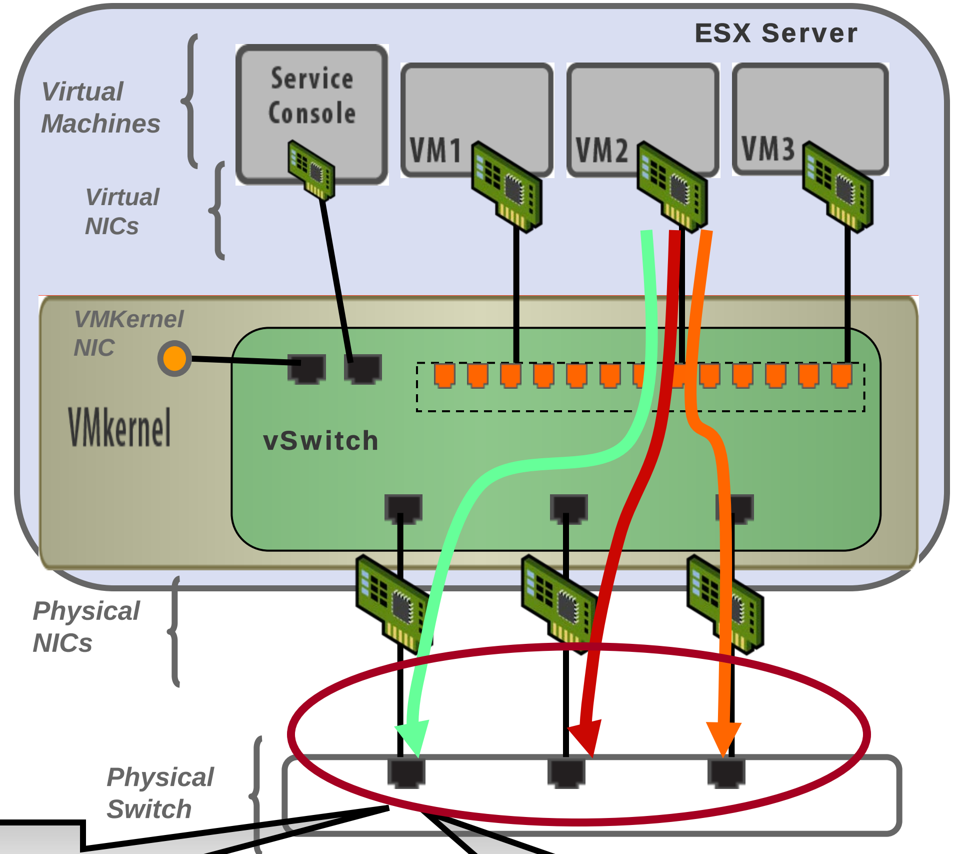
– An uplink is chosen based on

- Sender's vSwitch Port ID or LSB of the source MAC
  – Load balancing on a per vNIC basis
  – Allows teaming across physical switches in the same broadcast domain
  – Does not require the physical switches to be aware of the teaming
  – The physical switch learns the MAC/ switch port association
    - Inbound traffic is received on the same NIC



Teaming Based on IP Hash

- Uplink chosen based on
  - Source *and* Destination IP Address
    - Load balancing on a per connection basis
    - Requires physical switch to be aware of the teaming
    - Does *not* allow teaming across physical switches
    - Inbound traffic can be received on any one of the uplinks
    - Static link aggregation on physical switch.



Need to enable Link Aggregation on the physical switch ports

The switch sees VM2's MAC address on all three ports

Load Based Teaming - Basics

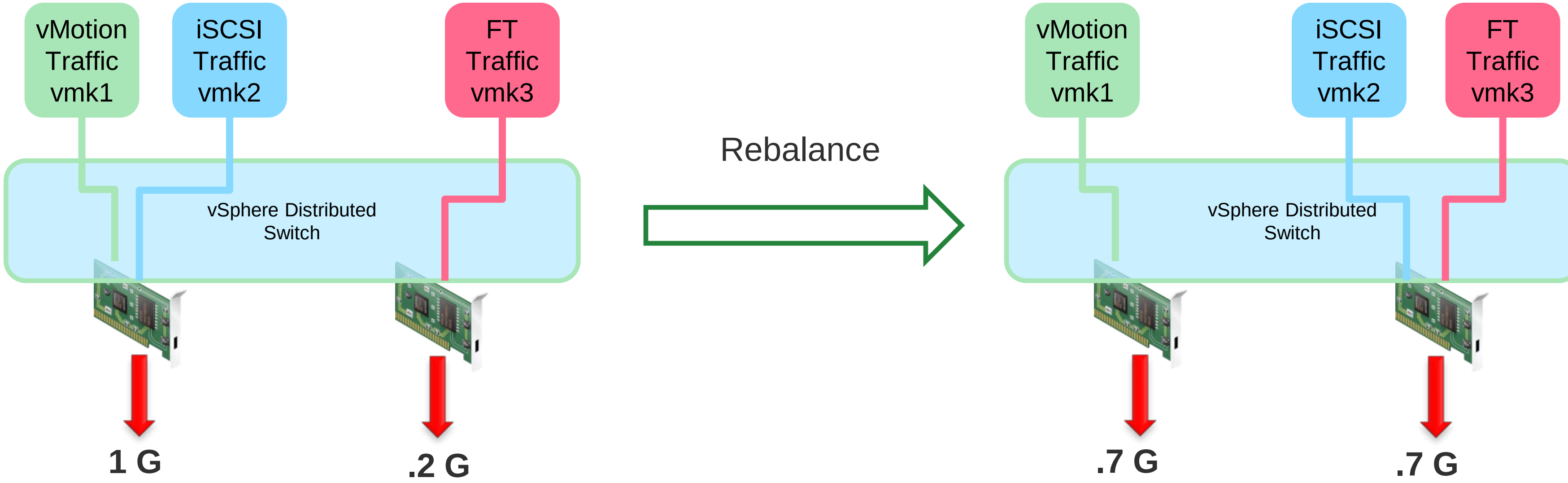
- Teaming Algorithm based on physical NIC load
- Avoids congestion on one physical NIC
- Algorithm
  - Every 30 Sec physical NIC load check is performed
  - If greater than 75% mean utilization on Tx or Rx is detected, LBT is invoked
  - Based on the utilization number of other NICs in the team and VNIC BW decision is made to move the traffic.
  - Works with mismatched port speeds as well.

vmware

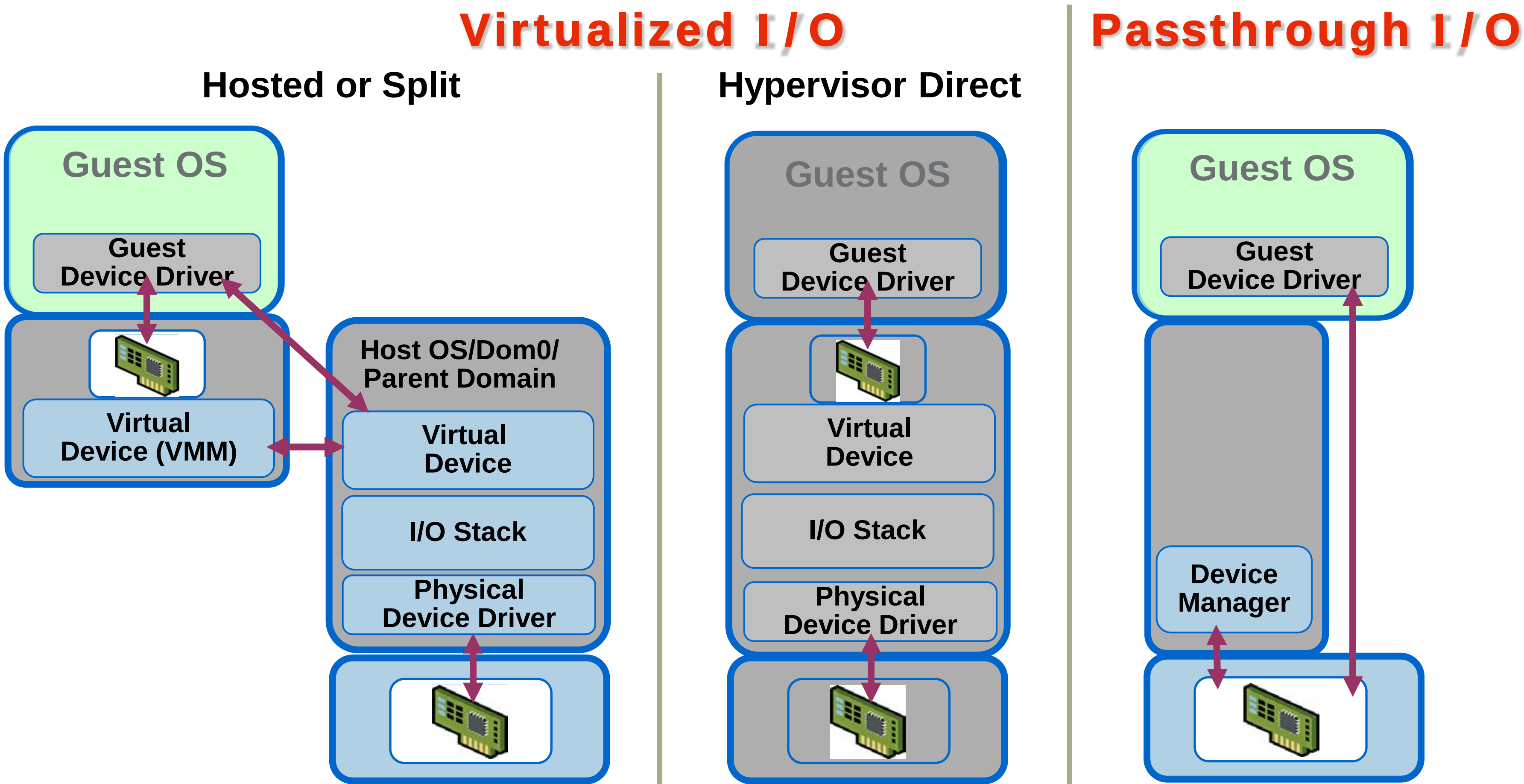
Load Based Teaming - Example

- Consider a two - one gig interfaces in a team configuration

Network Traffic	Bandwidth Requirement
iSCSI traffic	500Mbps
vMotion traffic	700Mbps
FT traffic	200 Mbps



Passthrough – Another Option



VMware Workstation, VMware GSX Server

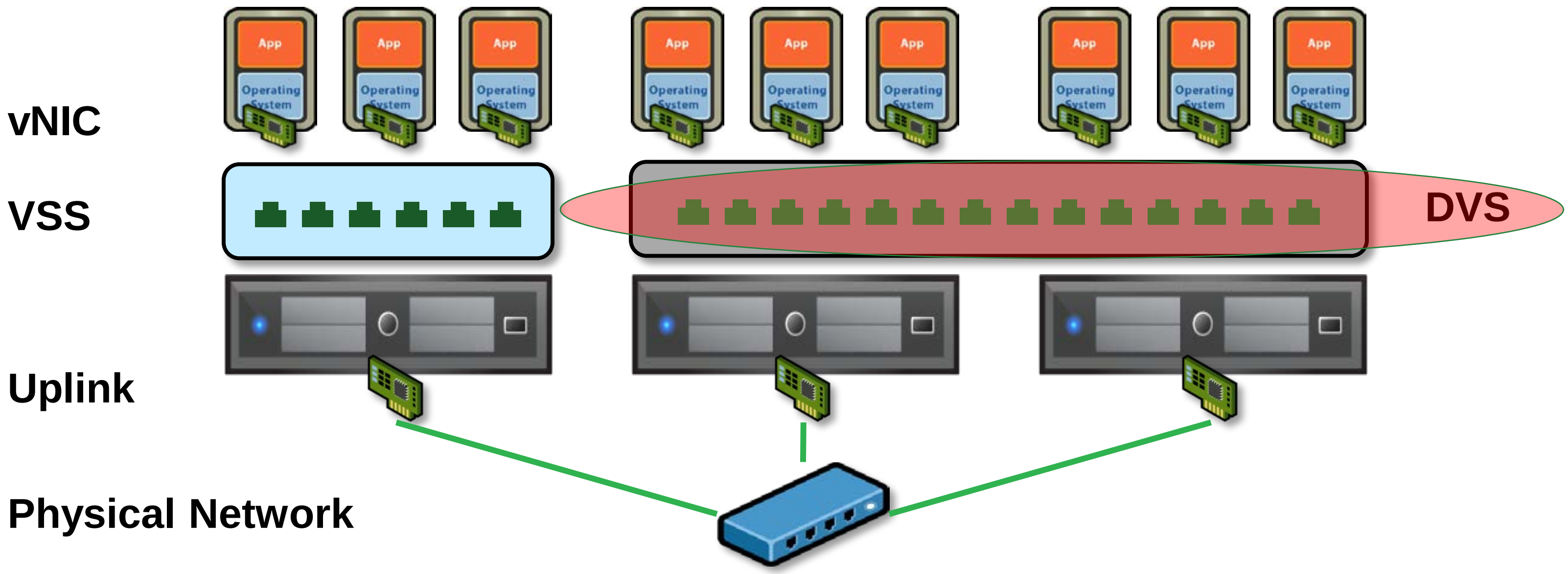
VMware ESX Server (storage and network)

Another Option

Passthrough – Use cases

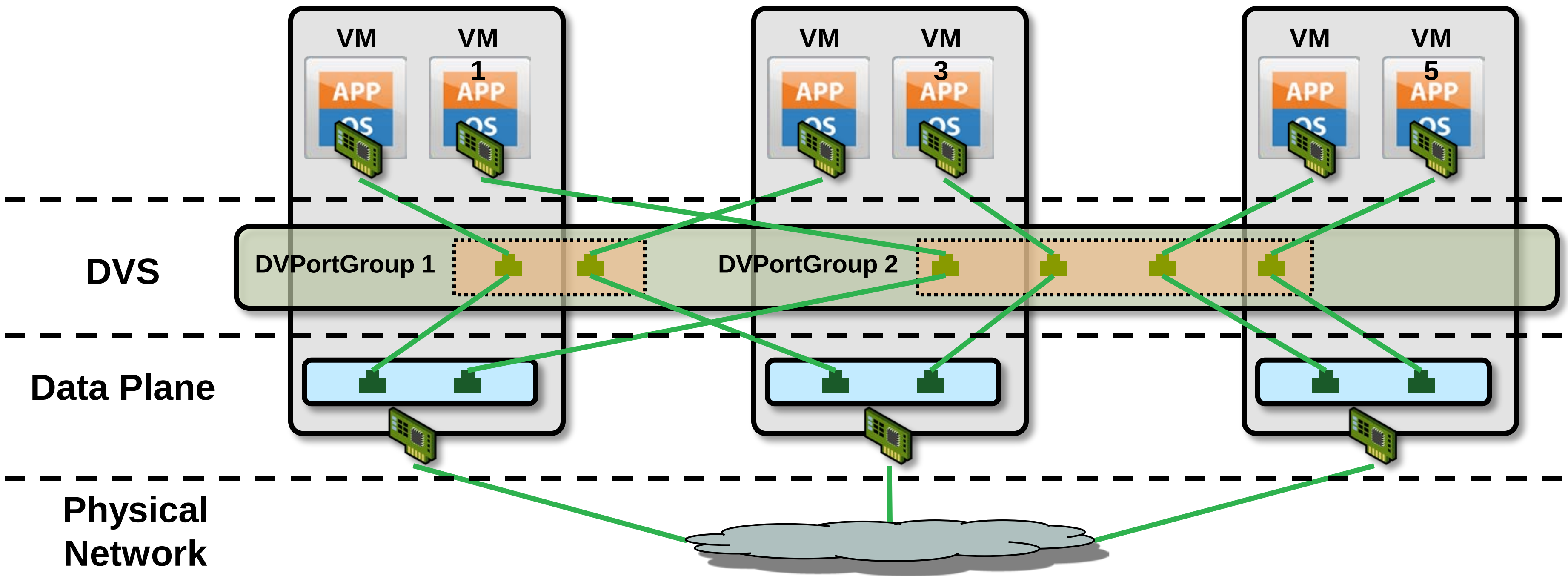
- Use case: Appliance VMs or special purpose VMs
  - Unsupported devices – Graphics, TOE
- Get the management benefits of VMs
- Get high performance by avoiding emulated I/O

Access Layer Virtualization – Distributed Virtual Switch (DVS)



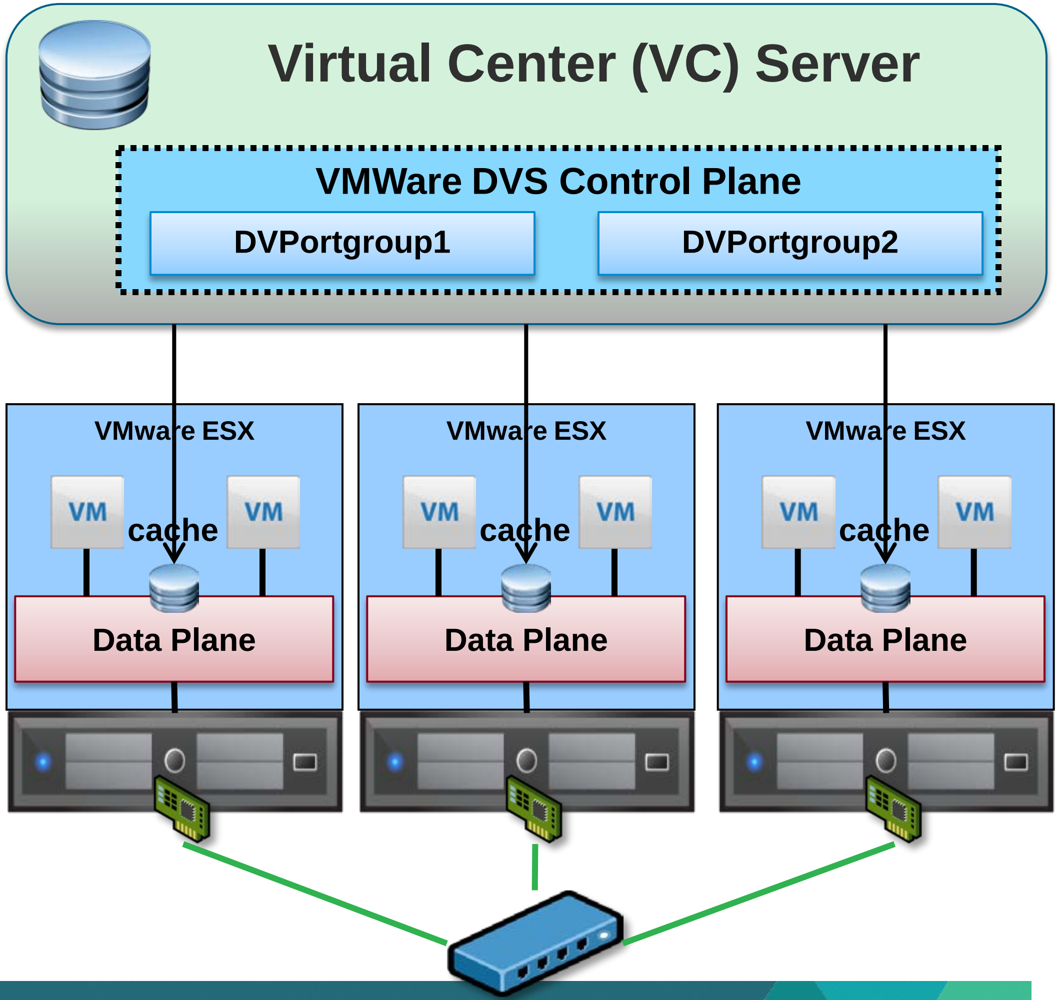
Distributed Virtual Switch (DVS)

- DVS is a distributed management layer across a set of ESX hosts
  - A high level virtual switch is established above vSwitch
    - VMs are connected to DVS's port (DVPort)
    - Each DVPort is associated with a data plane port
    - DVPort's configuration and life cycle are managed by a centralized node



DVS Architecture

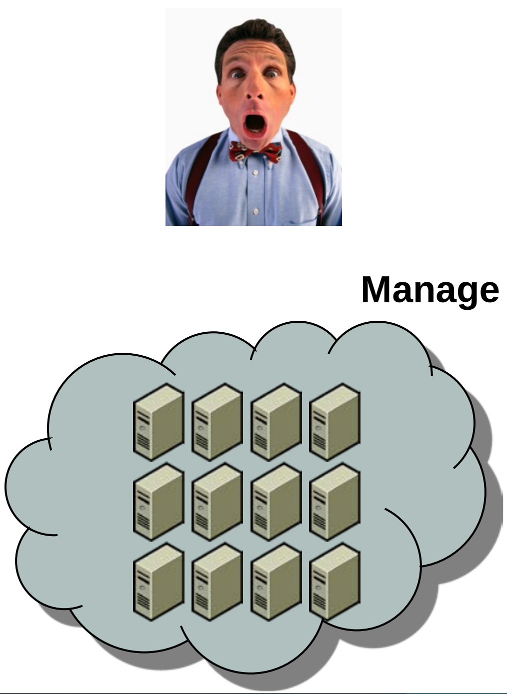
- Components
  - VC server
    - Store DVS configuration
    - Responsible for pushing configuration to ESX hosts
  - Data plane
    - Data plane is created when a host is added to a DVS
    - Configuration from VC is cached
    - VM traffic flows through data plane on the local host

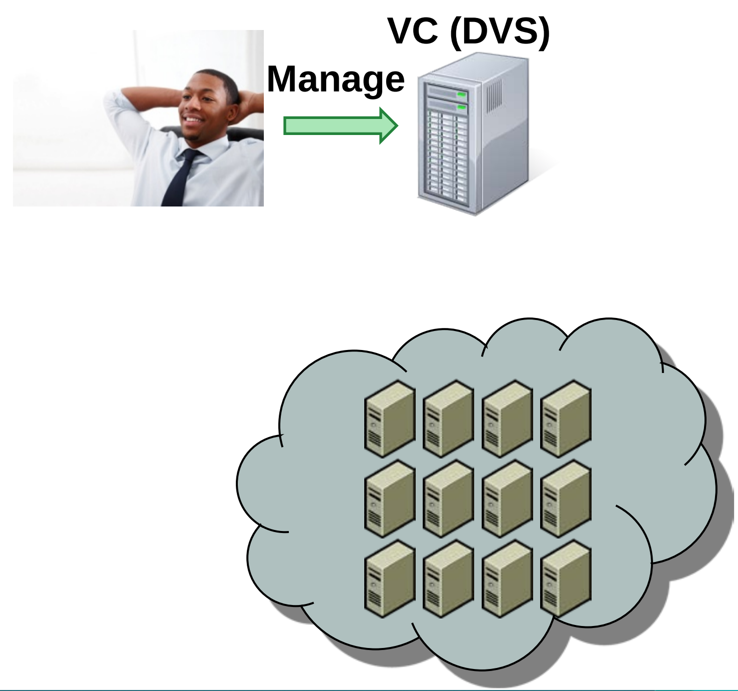


DVS vs VSS: Configuration

- VSS
  - An administrator has to configure hosts one by one

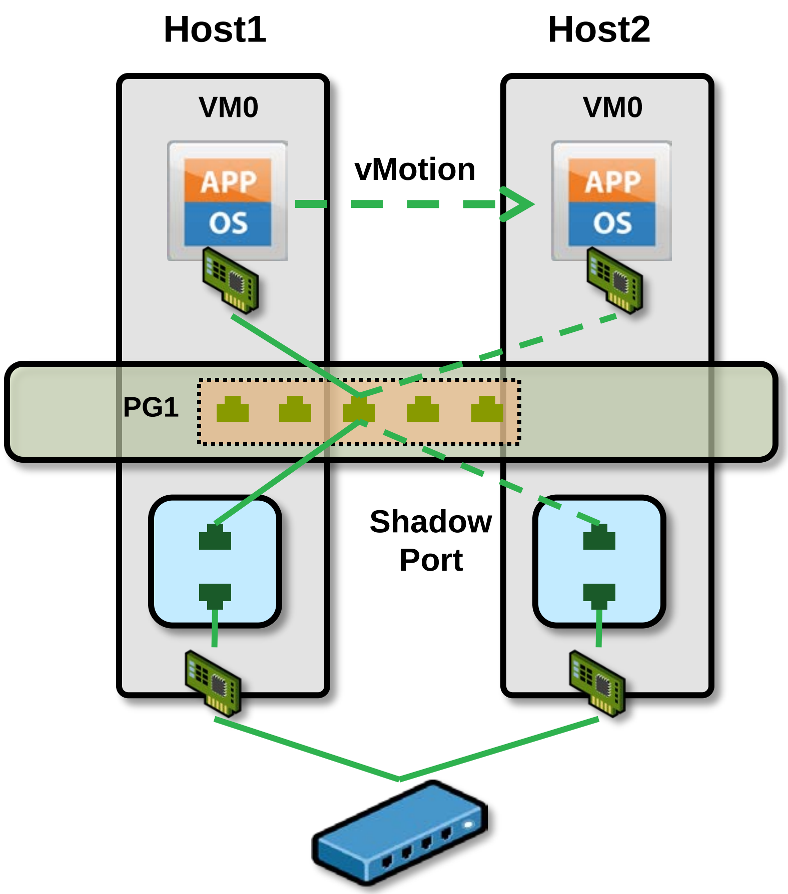
- DVS
  - – A central node dispatches configuration to each host on behave of the administrator





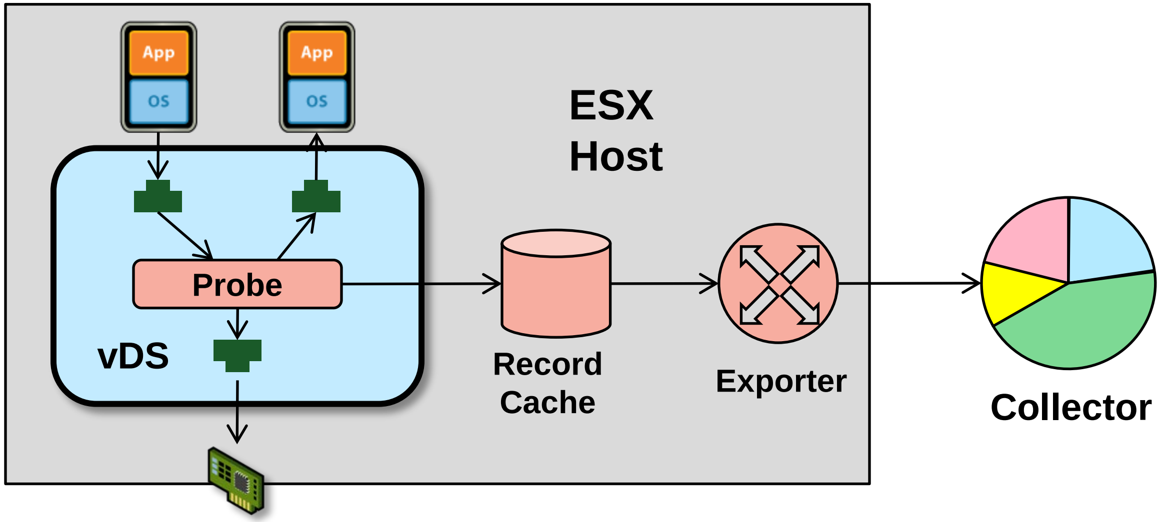
DVS vs VSS: vMotion Support

- vMotion support
  - VSS
    - Before vMotion, an administrator has to manually create a port on the destination host with the same configuration
  - DVS
    - During vMotion, the port configuration, as well as other port status, is migrated to the destination host automatically along with the VM



vDS Features: NetFlow

- NetFlow is a technology used to collect network traffic information
  - Flow
    - A sequence of packets with the same properties, such as source/destination IP, source/destination port, etc
    - The collected information includes packet number, total bytes, etc
- Components
  - Probe: monitor traffic and update flow information to record cache
  - Record cache: keep and age flow information
  - Exporter: export expired records to collector
  - Collector: summarize records and show the result to users
- NetFlow information can be used for troubleshooting, auditing, etc

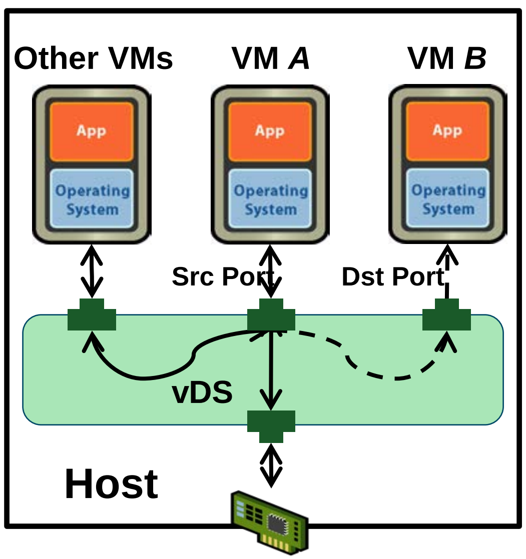


vmware

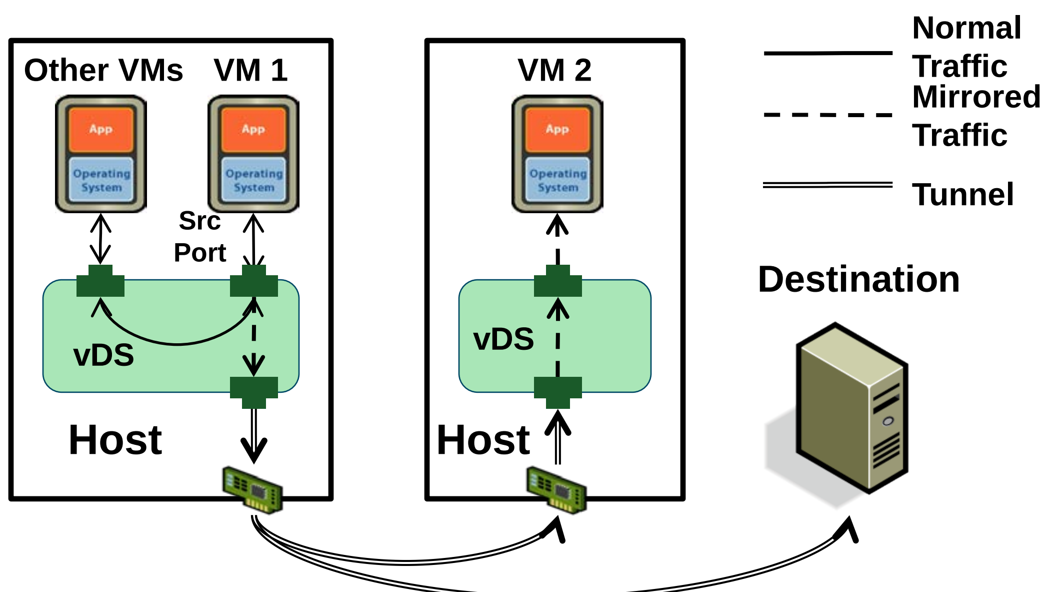
vDS Features: DVMirror

- DVMirror
  - VMware's port mirroring implementation
  - Used for troubleshooting, traffic monitoring, etc

- Local Mirror
  - Source and destination are on the same host

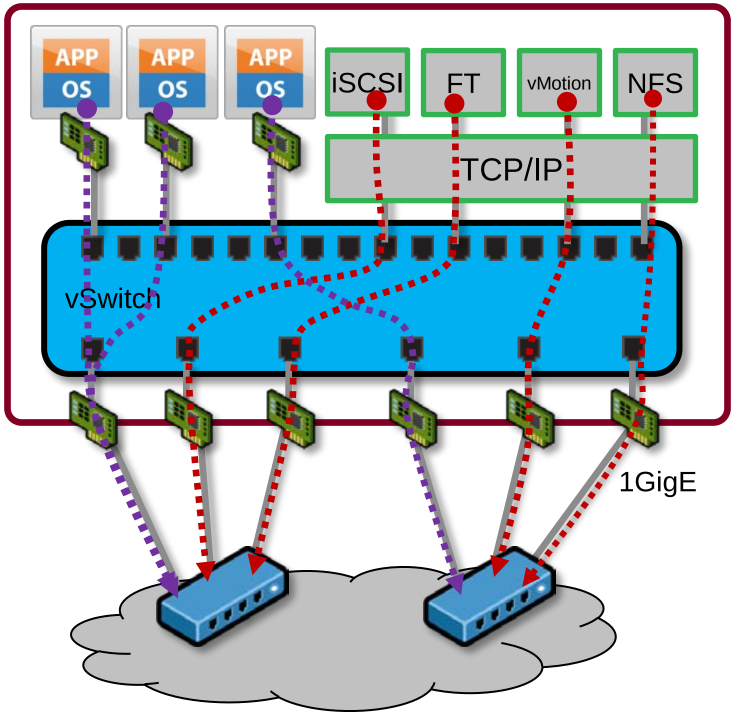


- Remote Mirror
  - Destination is on another host, or a physical box
  - Mirrored traffic is transmitted via a tunnel



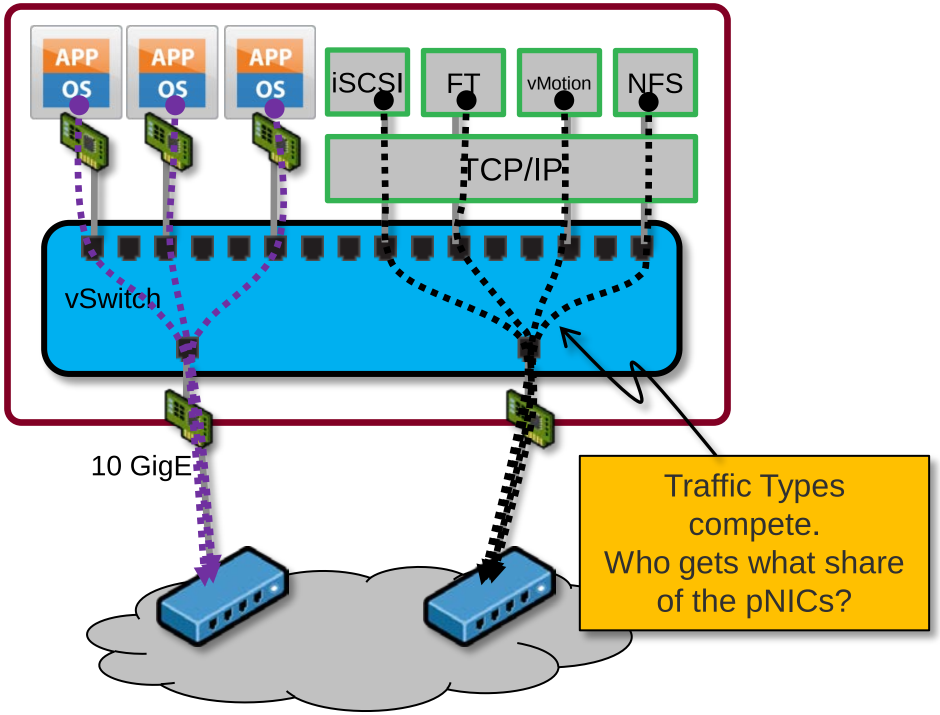
vDS Features: Network I/O Control

1GigE pNICs



- NICs dedicated for some traffic types e.g. vMotion, IP Storage
  - Bandwidth assured by dedicated physical NICs

10 GigE pNICs



- **Traffic typically converged to two 10 GigE NICs**
- **Some traffic types could dominate others.**
- **Hence need Traffic Management**

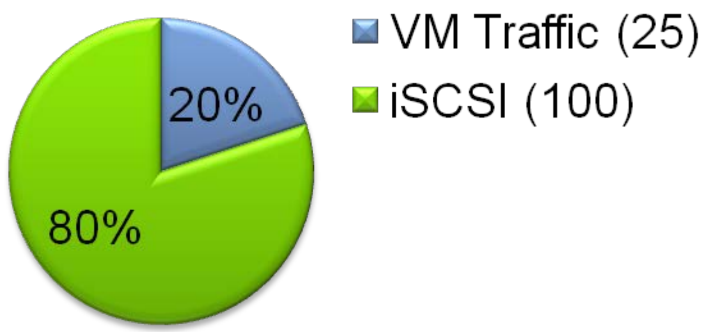
vmware

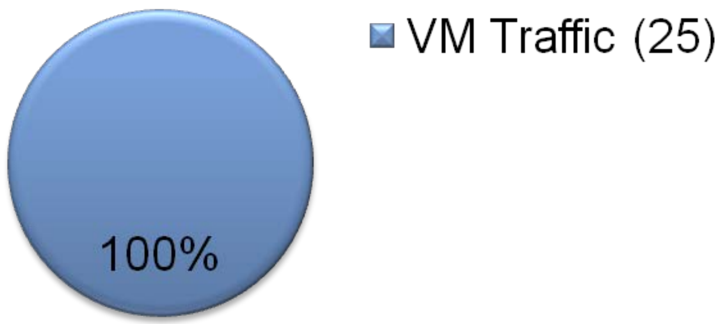
Network I/O Control - Parameters

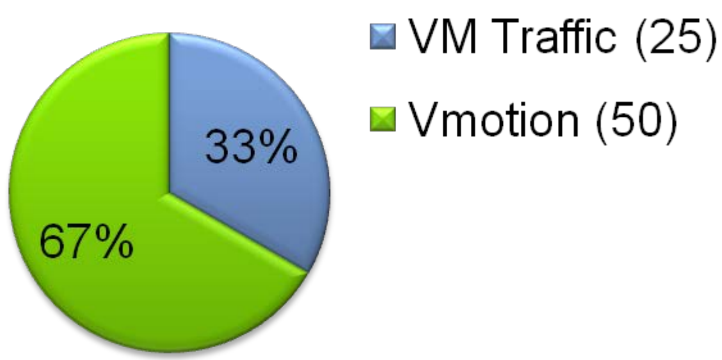
- Limits and Shares
  - **Limits** specify the **absolute maximum** bandwidth for a traffic type
  - Specified in Mbps
    - Traffic will never exceed its specified limit
  - **Shares** specify the **relative importance** of an egress traffic on a **vmnic** i.e. guaranteed minimum
    - Specified in abstract units, from 1-100
    - Presets for *Low* (25 shares), *Normal* (50 shares), *High* (100 shares), plus *Custom*
    - Bandwidth divided between traffic types based on their relative shares
  - Controls apply to output from ESXi host
  - Shares apply to a given vmnic or uplink
  - Limits apply across the team

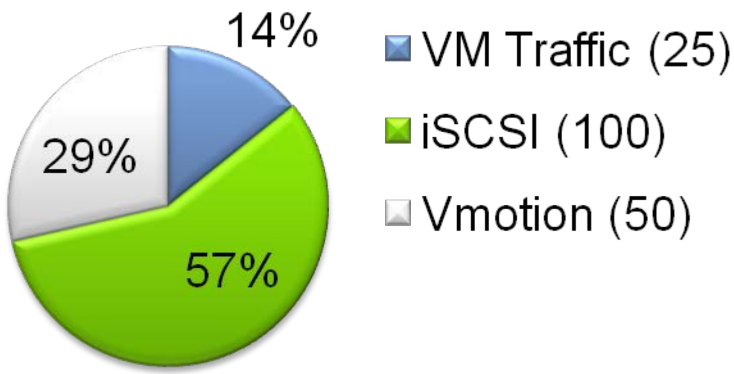
Network I/O Control - Example

•Shares Example: VM=25; Vmotion=50; iSCSI=100







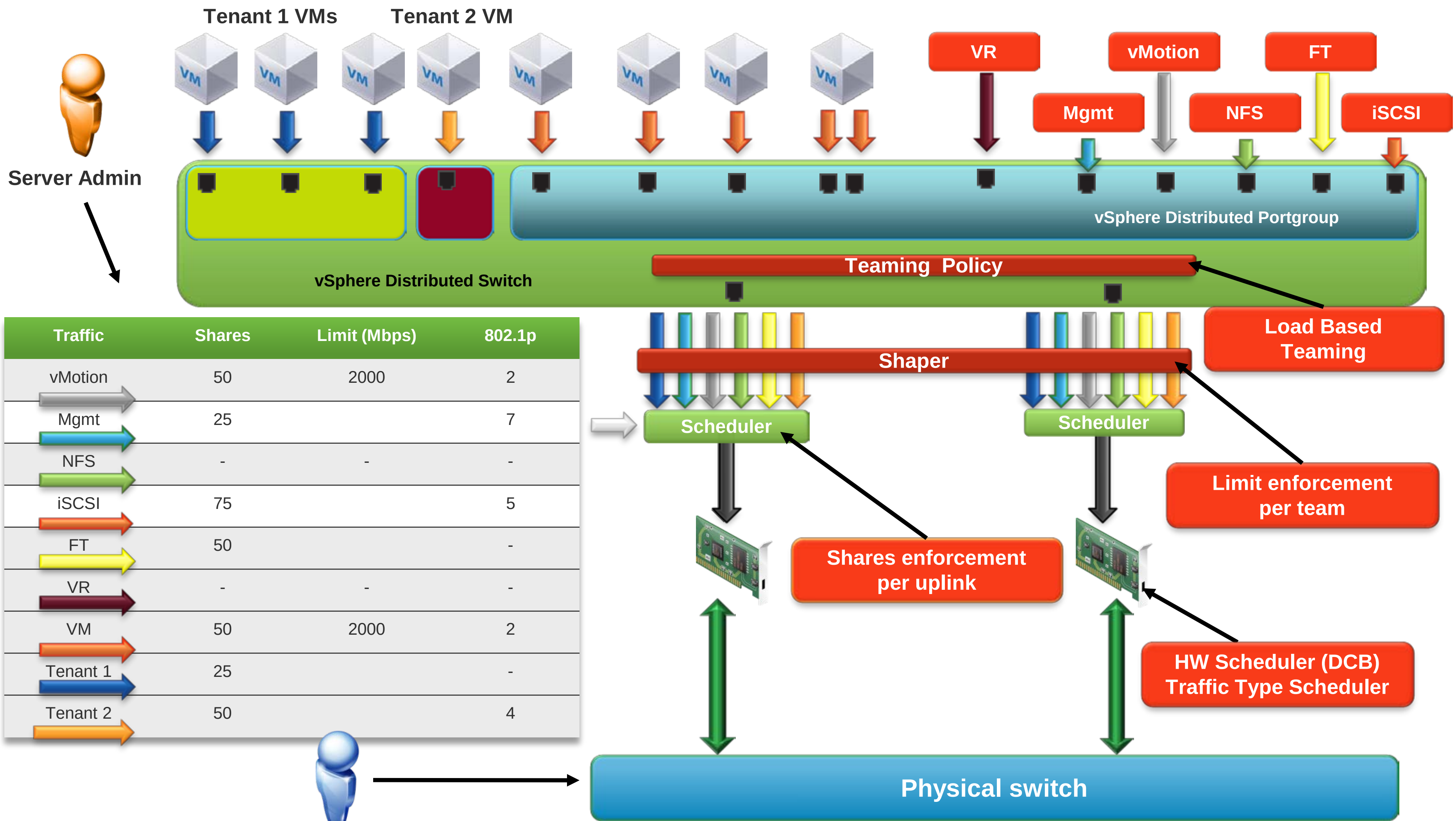


Network I/O Control - Benefits

- Network I/O Control provides
  - Isolation
    - One flow should not dominate others
  - Flexible Partitioning
    - Unused bandwidth is automatically distributed to other traffic type
  - Guarantee Service Levels when flows compete
  - User Defined Network Resource Pools
  - QoS Tagging to provide End to End service guarantees

- Supports 7 different traffic types
  - Management, iSCSI, vMotion, FT, NFS, VM, VR

Network I/O Control - Multi-Tenant + CNA Offload



Traffic	Shares	Limit (Mbps)	802.1p
vMotion	50	2000	2
Mgmt	25		7
NFS	-	-	-
iSCSI	75		5
FT	50		-
VR	-	-	-
VM	50	2000	2
Tenant 1	25		-
Tenant 2	50		4

Network Admin

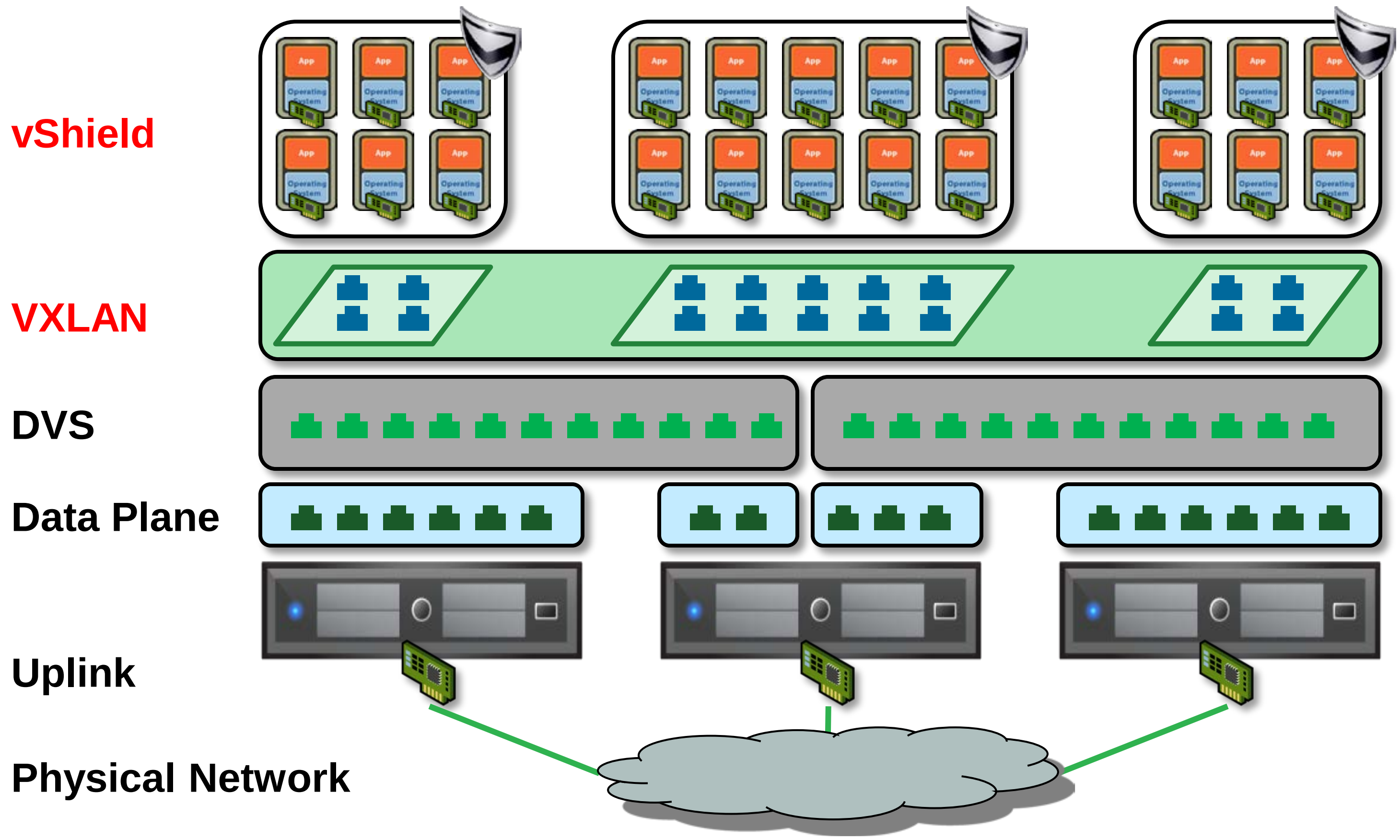
Agenda

- Virtualization
- Access Layer Virtualization
  - Virtual NIC
  - Virtual Standard Switch
  - Uplink
  - Distributed Virtual Switch
- Data Center Network Virtualization
  - vShield
  - VXLAN

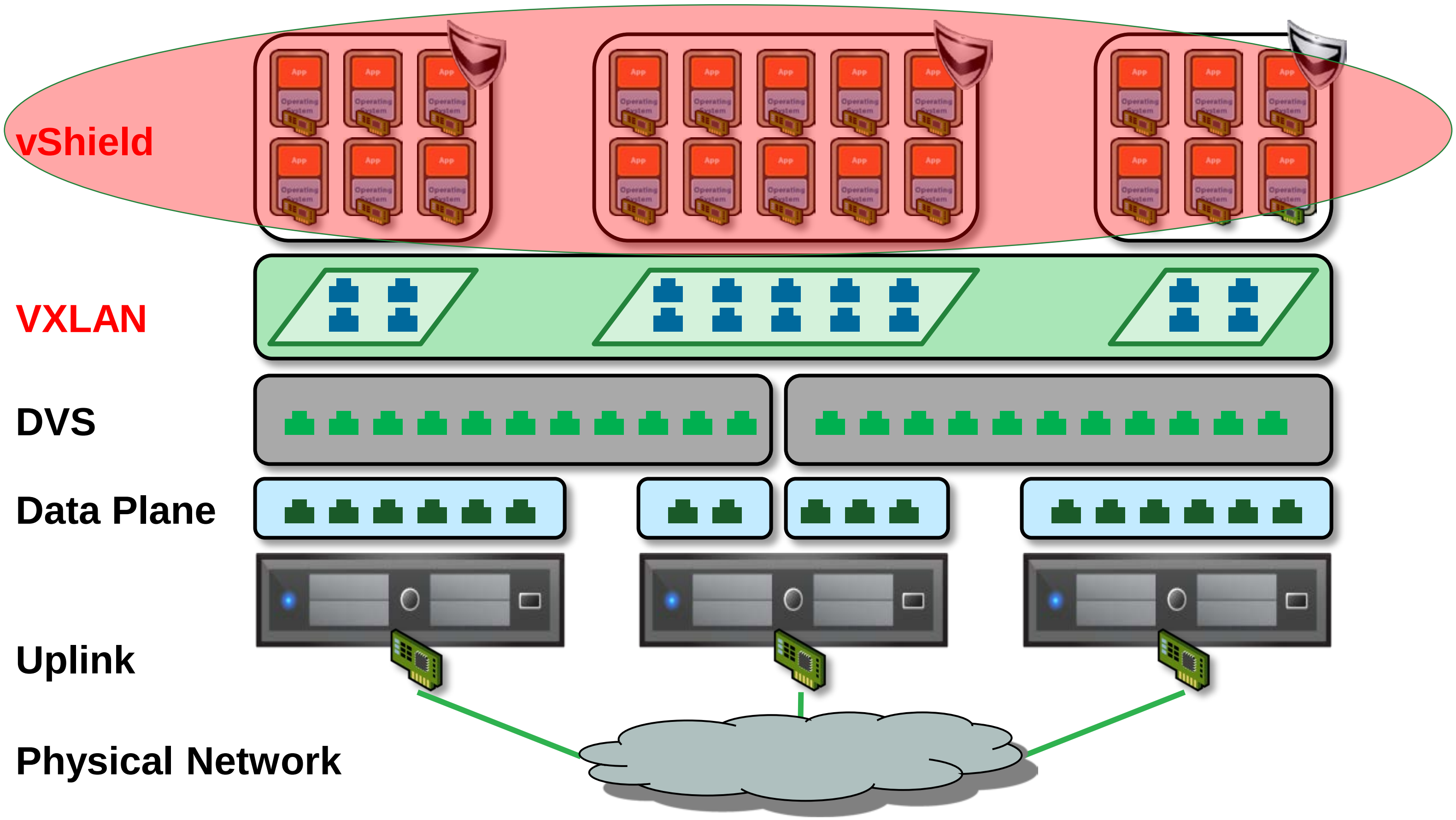
vmware

Data Center Network Virtualization

- VXLAN: distributed virtual L2
- vShield: private network management

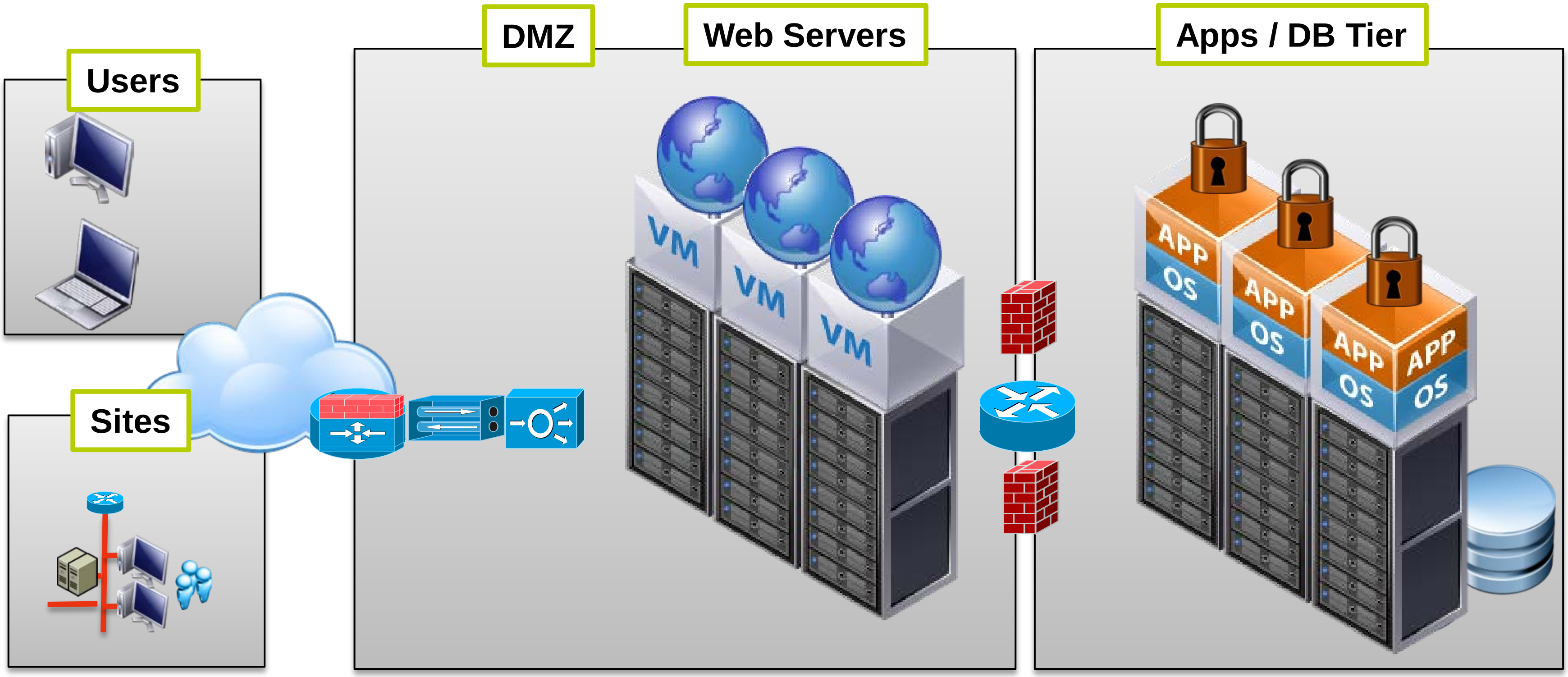


vShield

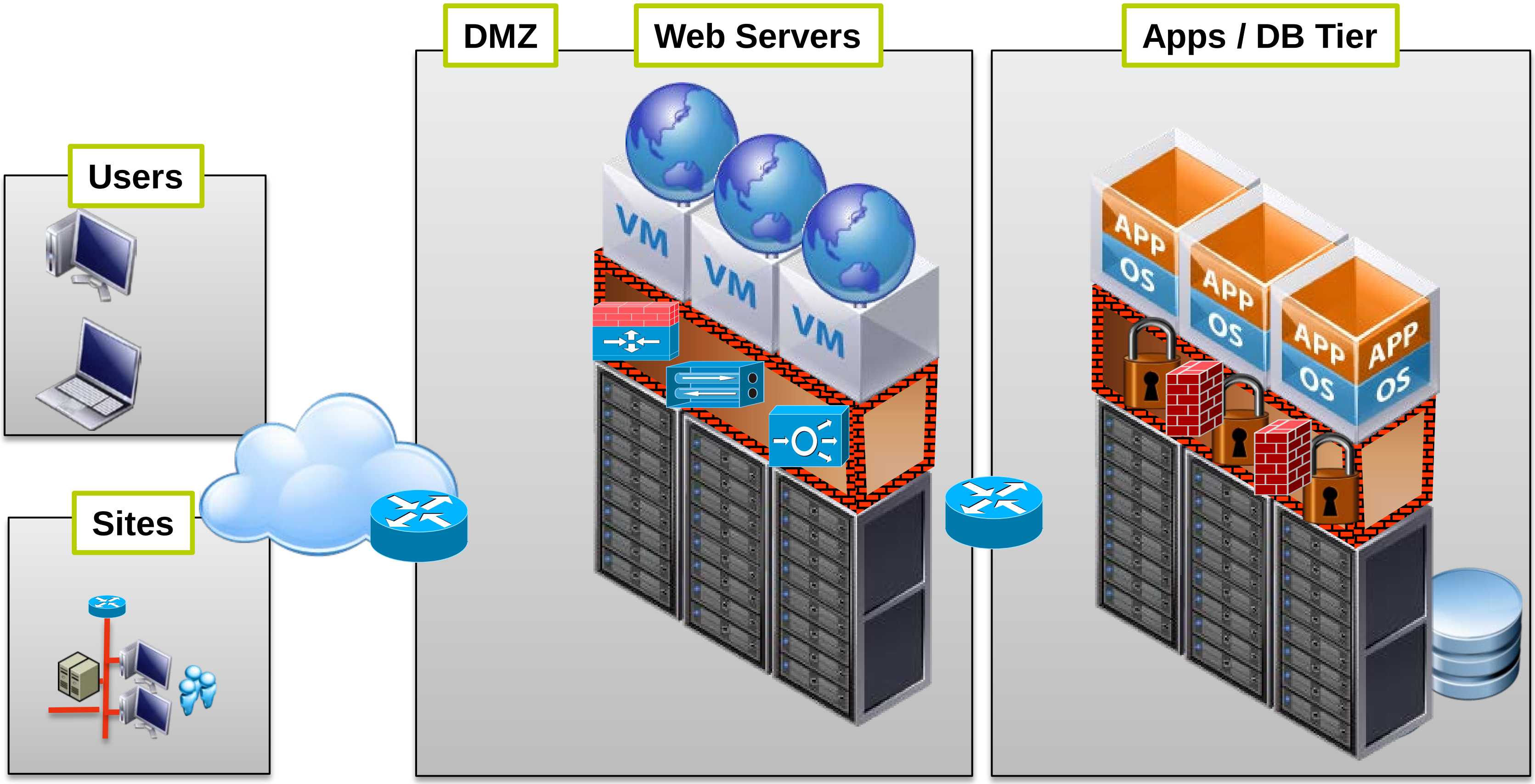


Security Requirements

- Server: antivirus, data integrity, ...
- Region: firewall, ...
- Edge: firewall, tunnel, WAN optimization, ...



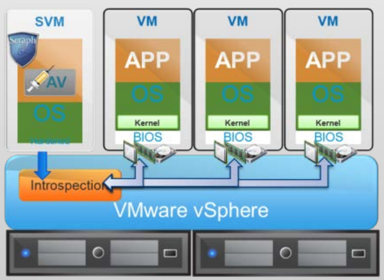
Consolidate Security Functions into Hypervisor



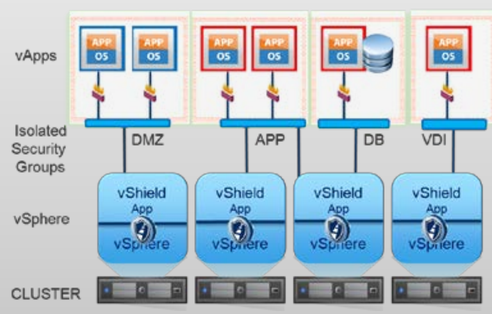
vShield: Multilayer Security

From inside the Guest to the Edge of the Cloud

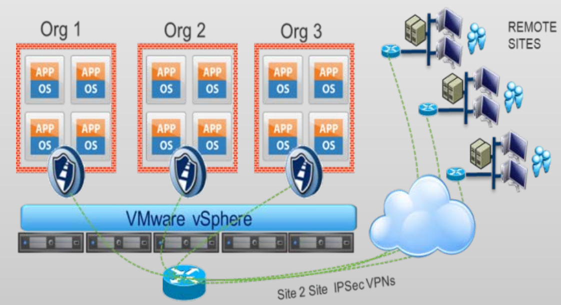
Inside VM	VM <-> network	vCloud / VDC edge



vShield Endpoint



vShield App

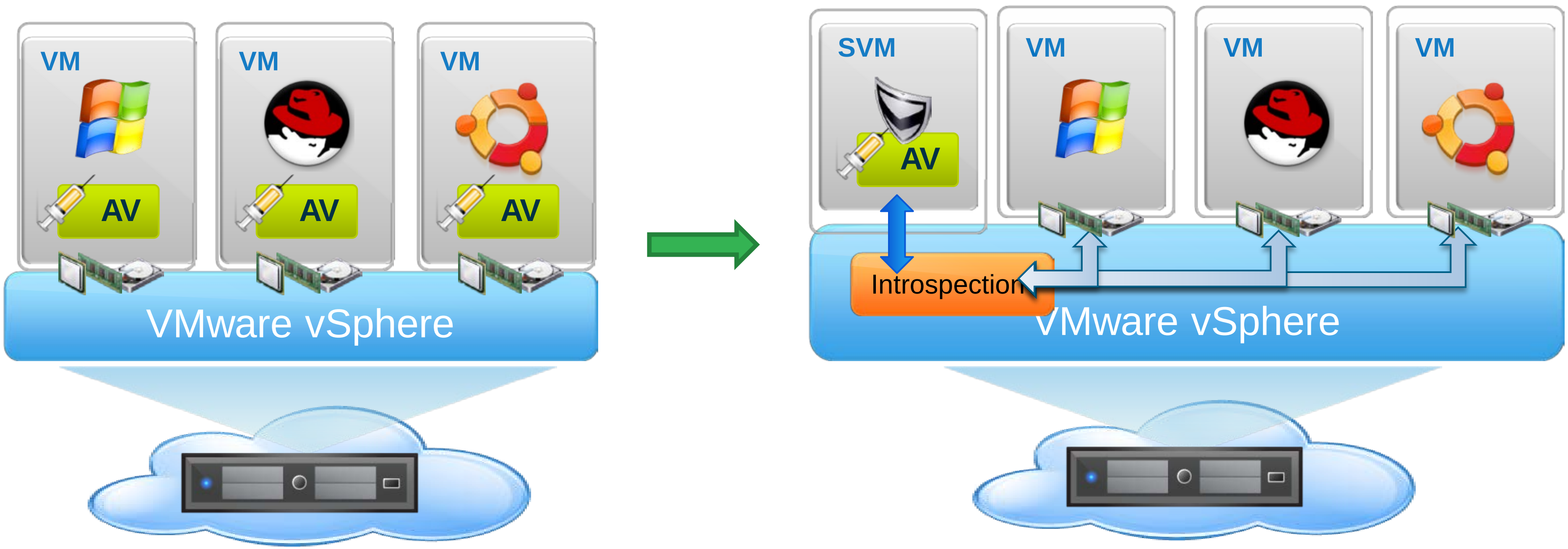


vShield Edge

- Endpoint: A/V offload with Partners; DLP with RSA
- App: DVFilter based distributed virtual firewall
- Edge: firewall, NAT, DHCP, IPSec VPN, load balancer

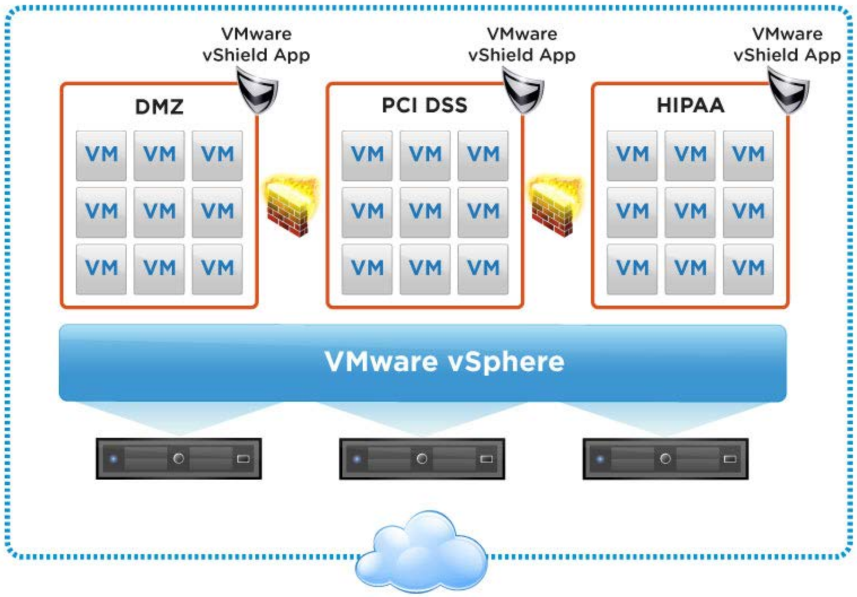
vShield Endpoint

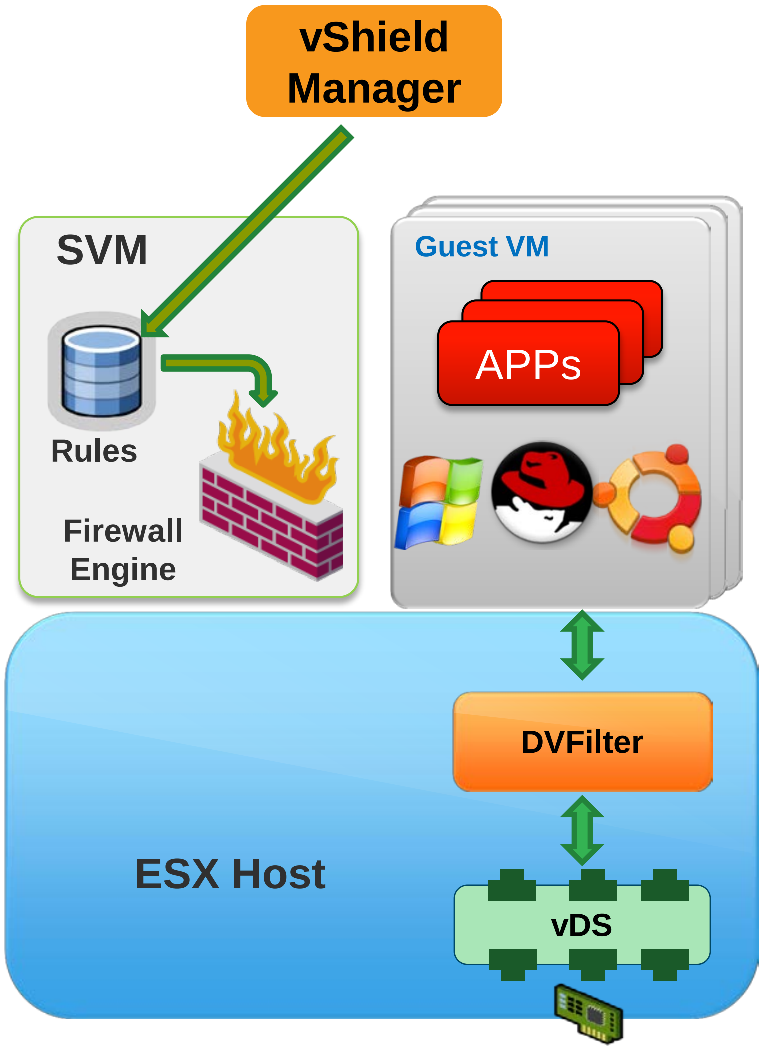
- Security as service
- Protection
  - Tamper-resistance – AV engine not directly accessible by malware
- Efficiency
  - No redundancy of AV code, virus definitions and updates



vShield App

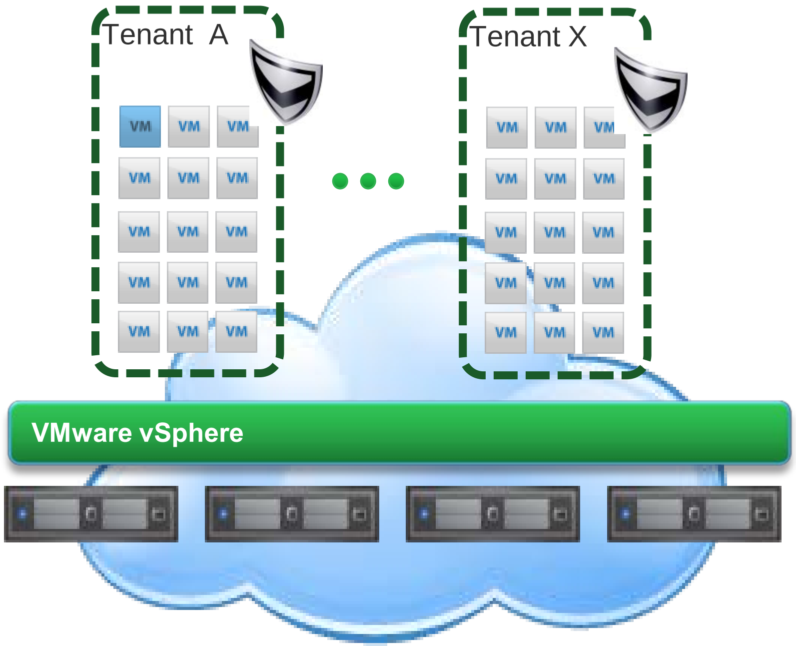
- vShield App protects among regions
  - Firewall rules are pushed from VSM
  - Traffic is filtered by firewall engine





vShield Edge

- vShield Edge: a virtual gateway sitting on the edge of a tenant's virtual network
  - Provide network services: NAT, DHCP, etc
  - Secure the Edge of the Virtual Data Center

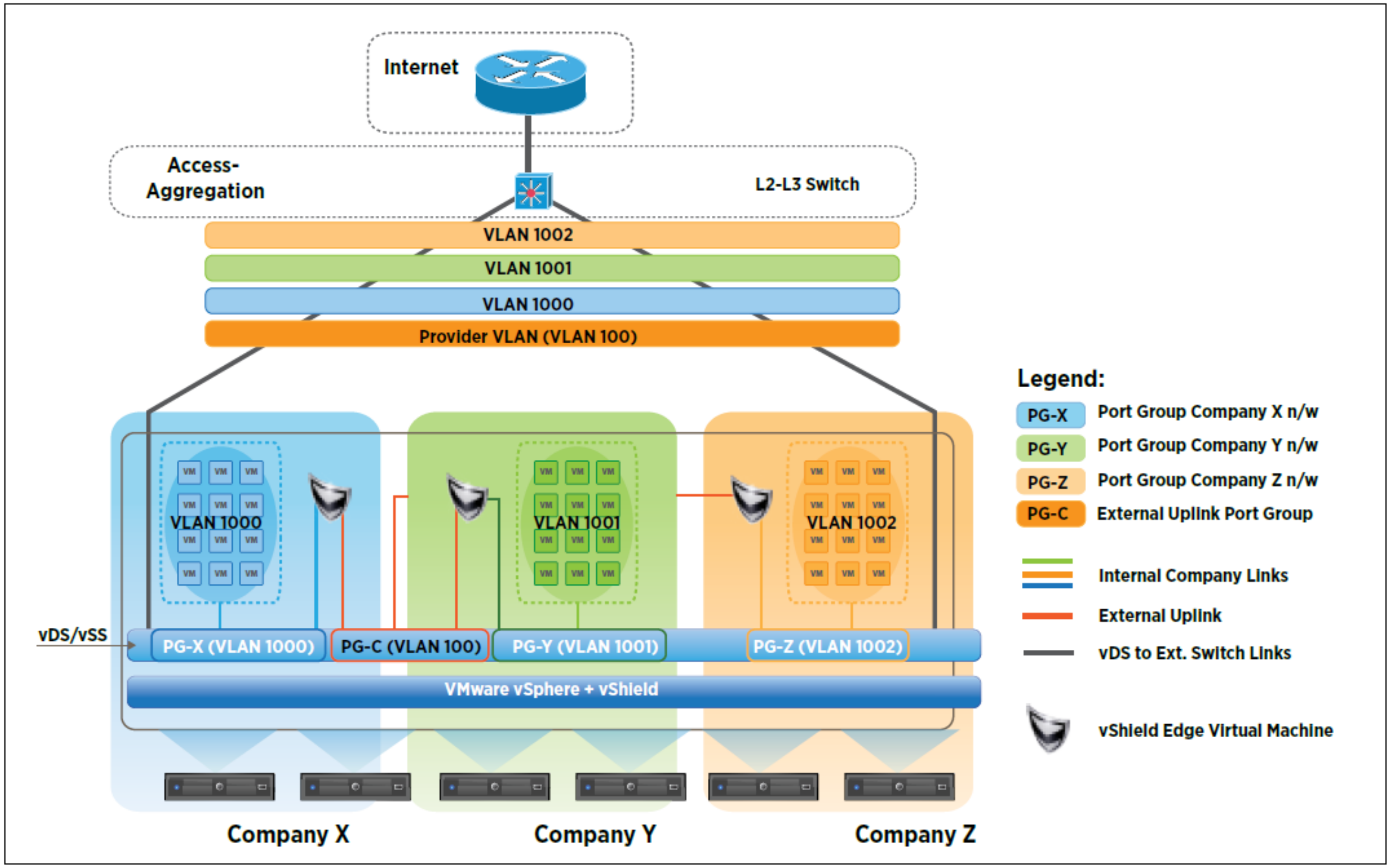


Features

- Multiple edge security services in one appliance
  - Stateful inspection firewall
  - Network Address Translation (NAT)
  - Dynamic Host Configuration Protocol (DHCP)
  - Site to site VPN (IPsec)
  - Web Load Balancer
- Policy management through UI or REST APIs
- Logging and auditing based on industry standard syslog format

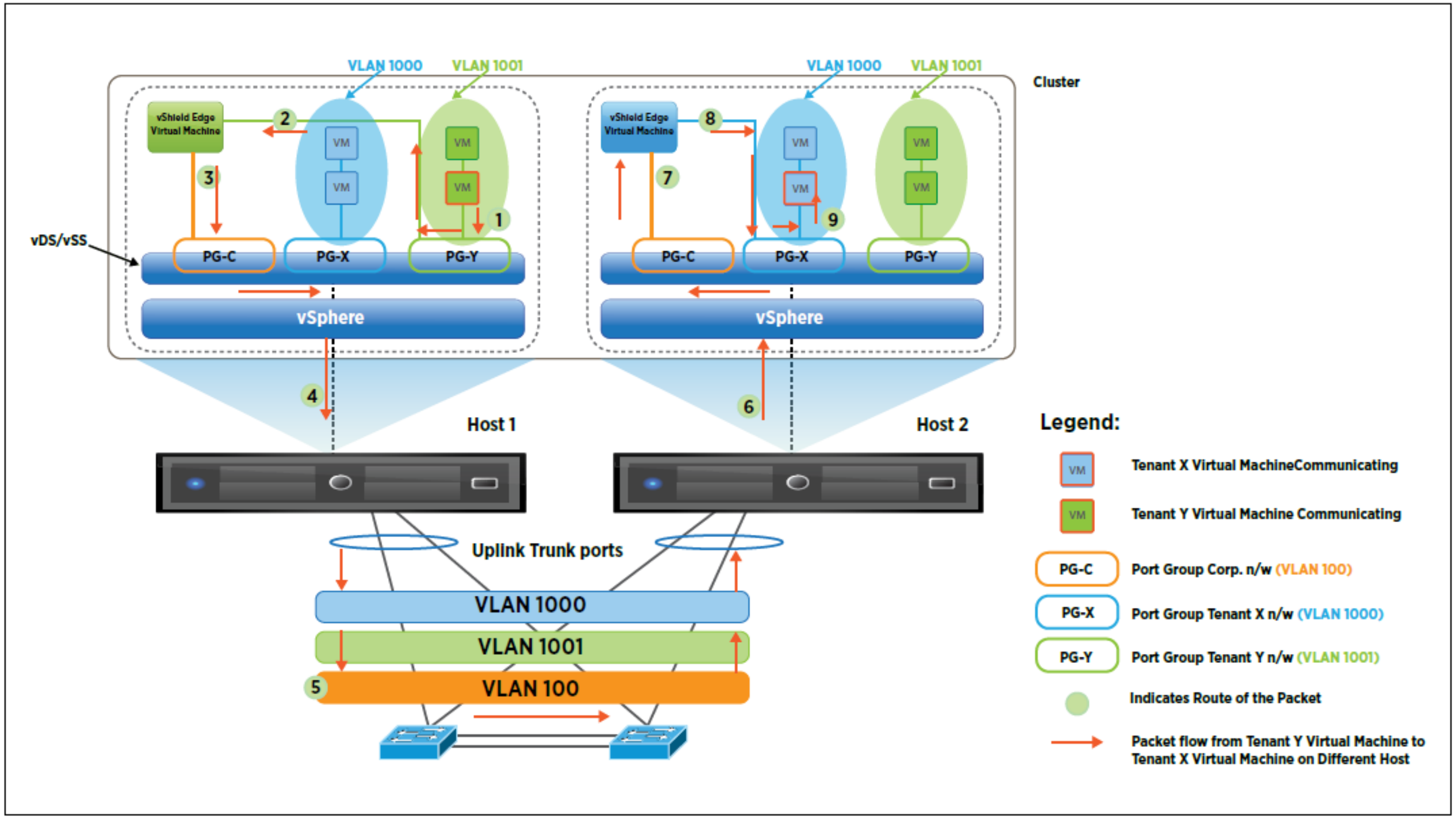
vShield Edge Deployment

- Isolation via VLAN
  - Each edge has two interface, located in different VLANs

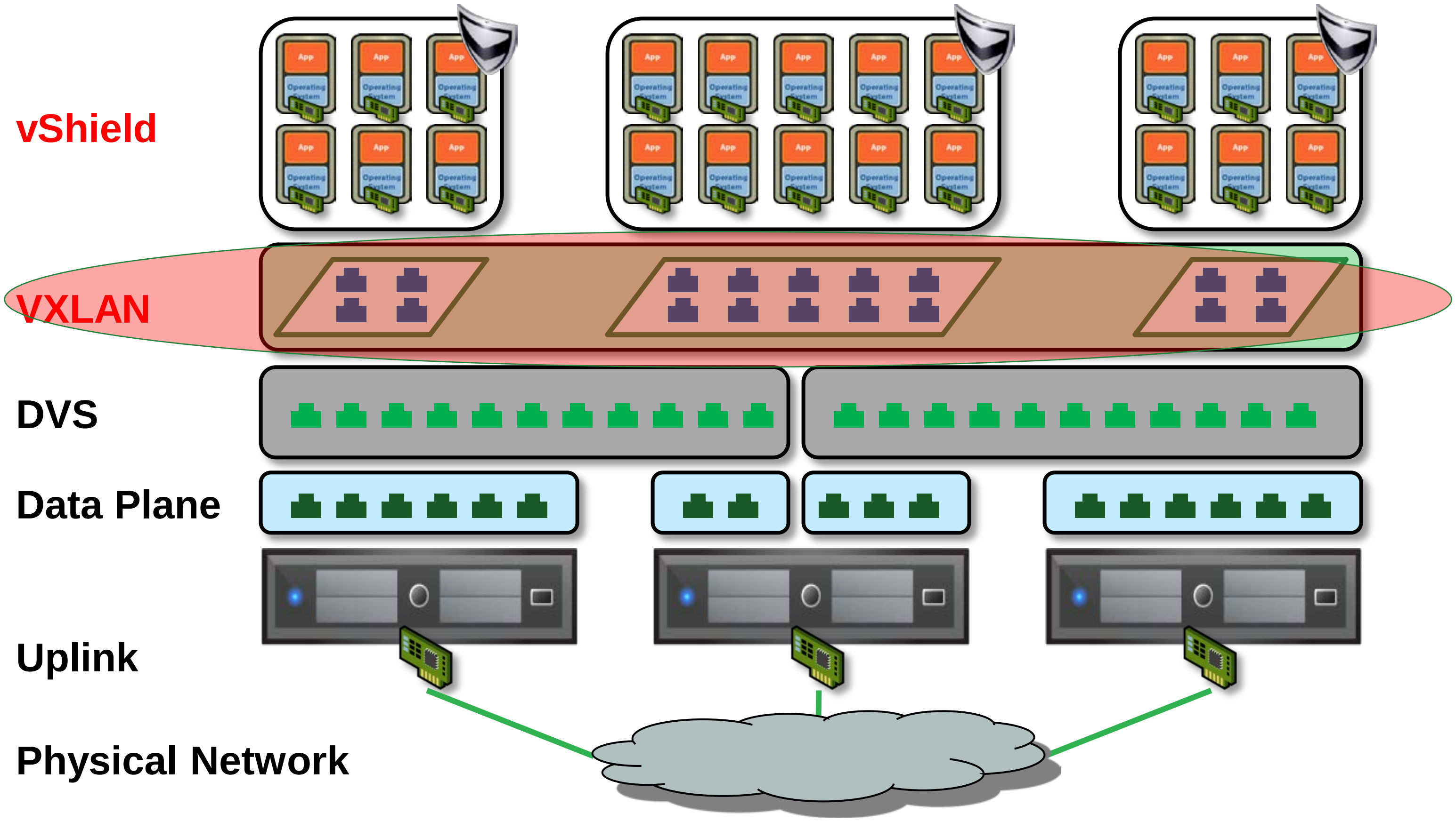


vShield Edge Traffic Flow

- Tenant Y's VM on Host1 sends packets to tenant X's VM on Host 2



VXLAN



Whv VXLAN?

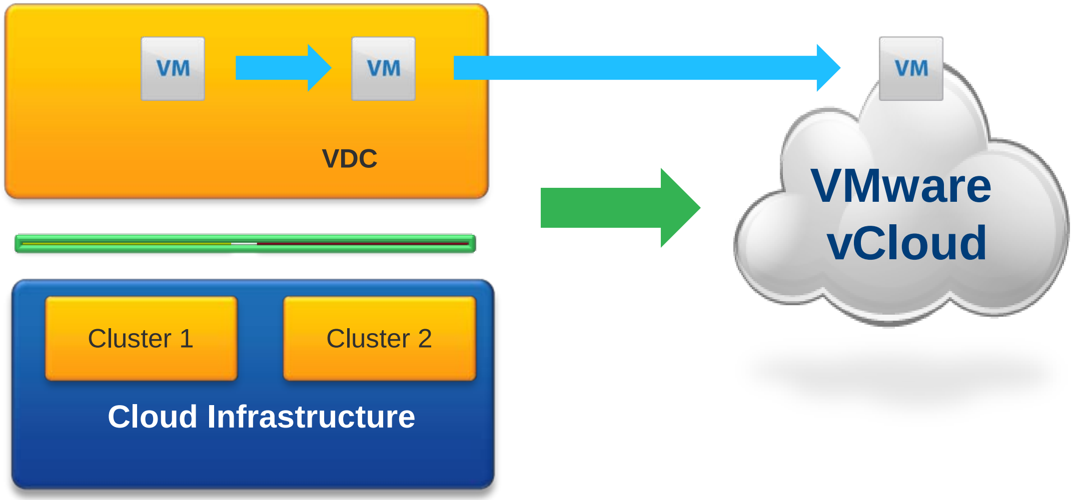
Drivers

- Need cross cluster mobility
- Enable provisioning workload where compute is available. Avoid operational heaviness of VLAN’s
- Provision large number of tenants (>4K limits of VLAN’s, avoid STP)
- Enable stateful movement of workloads (vMotion Anywhere) and failover scenarios with SRM

Untether the workload from the physical network

vmware

VXLAN: Enabling Elastic Compute



Overview

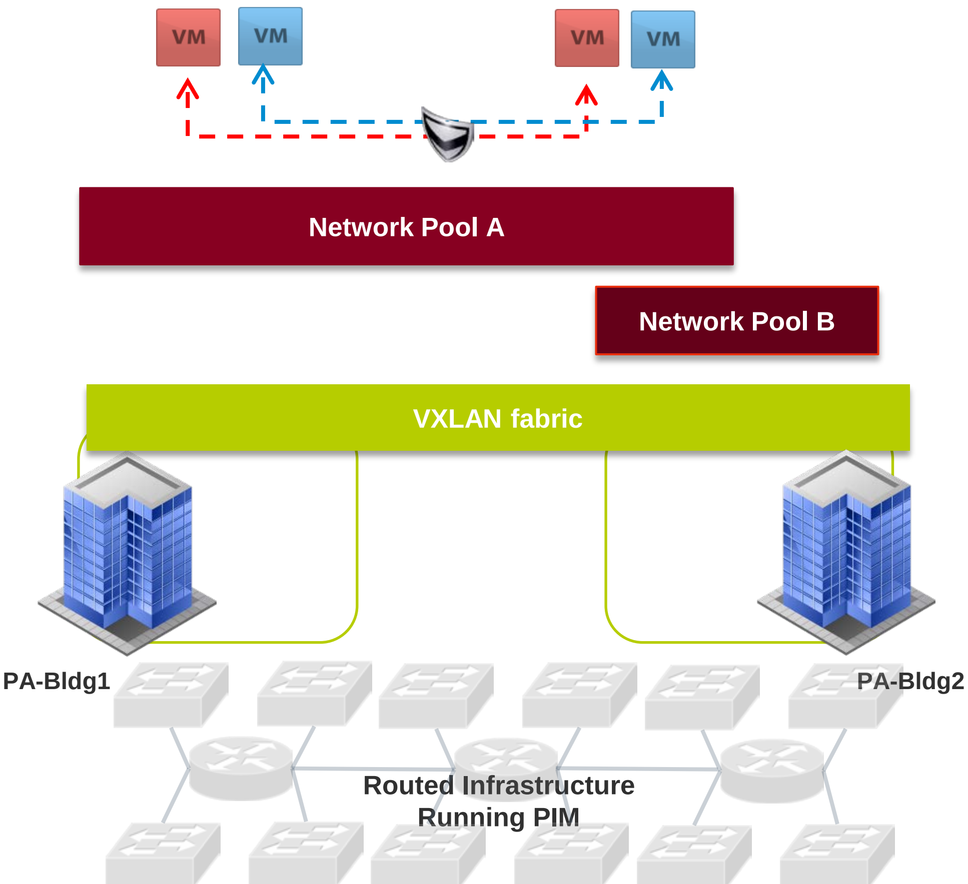
VXLAN allows mobility across subnet boundaries

Foundation for elastic portable VDC's

Benefits

- Cross cluster mobility within or across datacenters
- On demand networks without physical network re-configuration
- Massive scale for multi-tenant environments

VXLAN at High Level

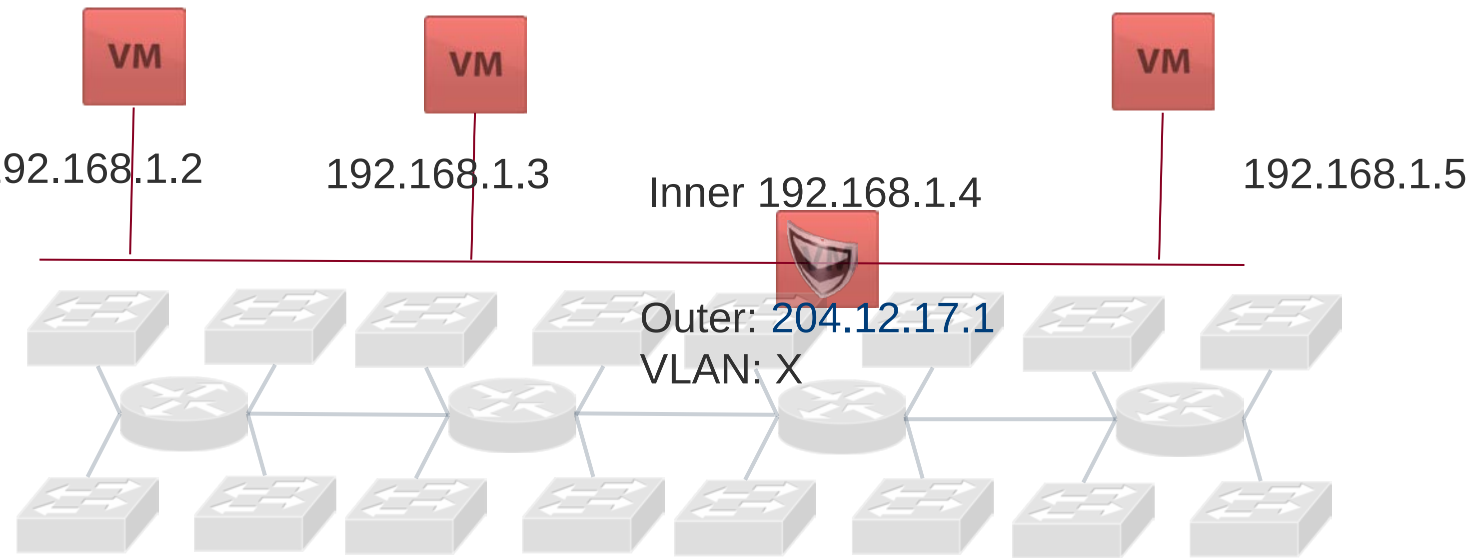


Build VXLAN wires and gateway on a network pool

Build network scopes based on compute containers

Build VXLAN fabric – Select your compute fabric, VDS, transport VLAN and multi-cast pool

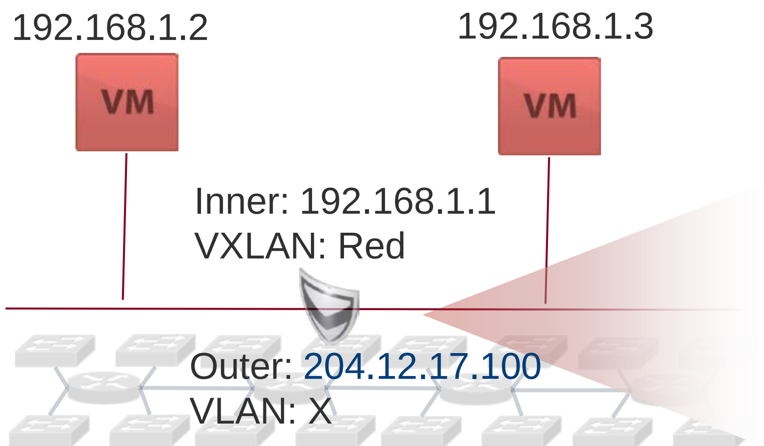
Logical view of VXLAN



Key Properties

- **Works with any switching fabric without change even across WAN**
- **Maintain visibility and control for network admins**
- **API to authoritatively program the logical network**

VXLAN - Details

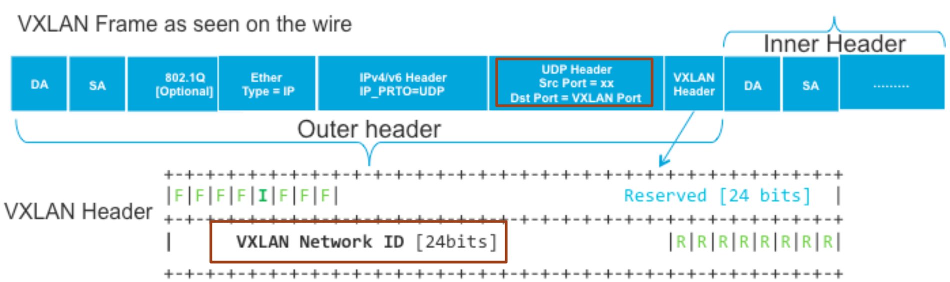


VXLAN Gateway

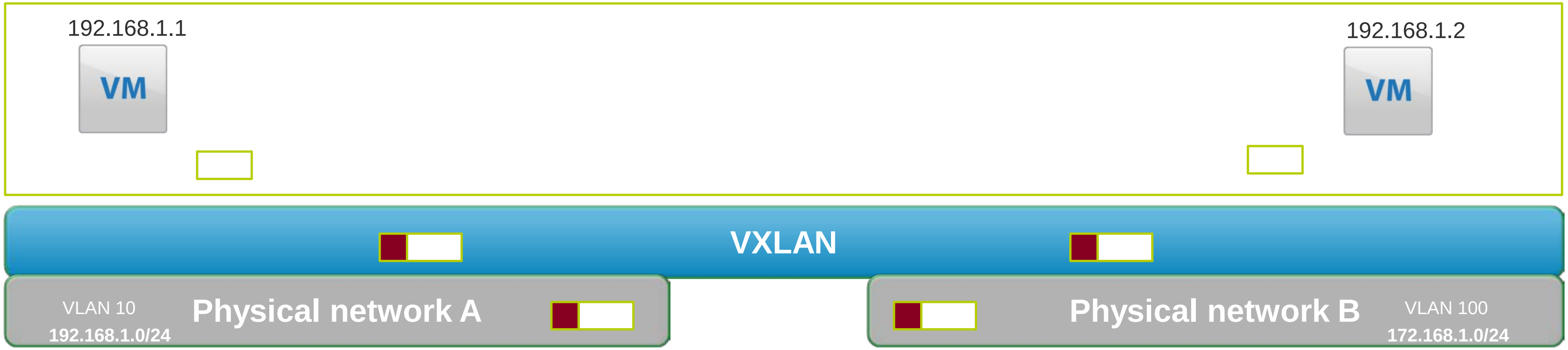
- Connect with legacy VLAN envs
- Inter VXLAN routing
- Provides Services

Frame format

- VXLAN Network ID (VNI) is 24 bits up to 16M networks
- Leverage ECMP by using UDP for encapsulation
- Uses Multicast to replicate for broadcast/unknown forwarding - leverages PIM and IGMP pruning for traffic management



VXLAN - How



Key Properties

- **Works with any switching fabric without change even across WAN**
- **Maintain visibility and control for network admins**
- **API to authoritatively program the logical network**

Questions?