Ethernet Virtual Bridging Automation Use Cases

Renato Recio, Sivakumar Krishnasamy, and Rakesh Sharma

Presentation Abstract

- In today’s Data Center Networks, a lot of network state1 is manually created, applied and maintained.
    - This complexity increases with server virtualization,
      where a single switch port can connect 100s to 1000s of Virtual Machines (VMs) and
      those Virtual Machines can migrate across servers.

- This presentation describes several use cases for managing network state today, including the advantages/disadvantages of each approach.

- It then describe how protocol contributions to the IEEE 802.1Qbg Edge Virtual Bridging (EVB) working group can be used to enable new, more automated network state management use cases.

[1] Note: Network state includes VLAN Identifiers, Access controls (e.g. IP address filters, MAC address filters), Quality of Service controls (e.g. rate limits) and/or security flow rules associated with a (physical or virtual) port.

Agenda

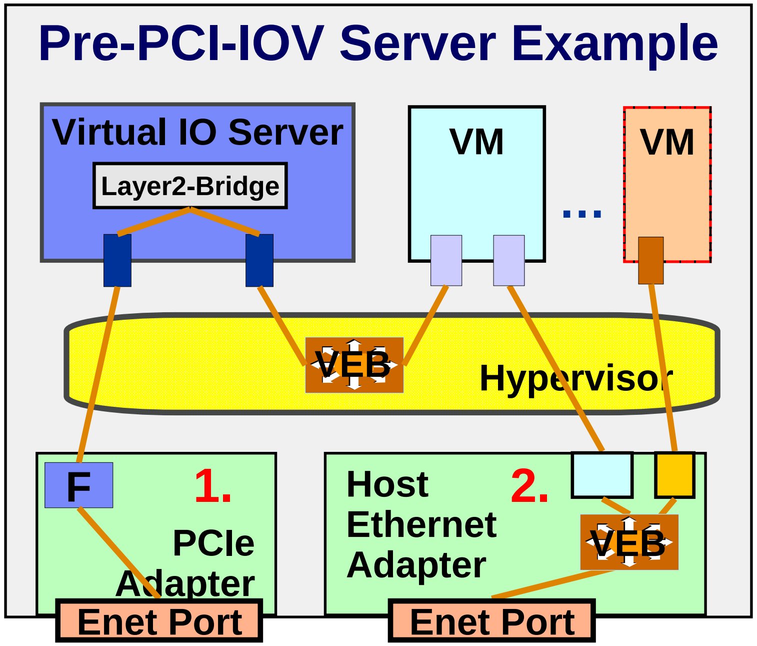
- Approaches used today to manage network state associated with Ethernet Virtual Bridges
  - Homogeneous tenant network state
  - Post VM migration of heterogeneous tenant network state
  - Open access for heterogeneous tenant network state

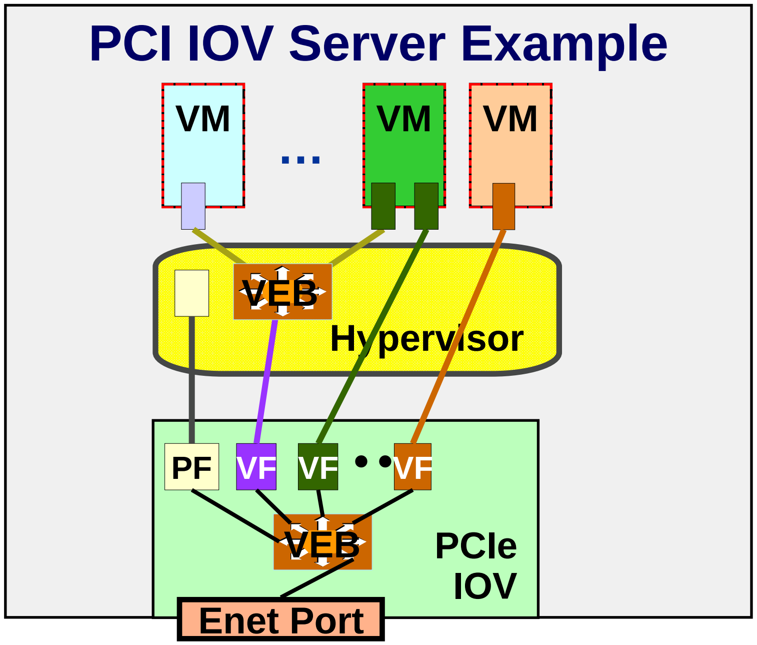
- Ethernet Virtual Bridging (EVB) Principles of Operation
  - Defining network states and capabilities
  - Ethernet Virtual Bridging Protocols

- Ethernet Virtual Bridging Use Cases
  - VSI Manager Use Cases
  - VSI Type Database Use Cases
  - Push Based VSI Type Use Cases
  - Pull Based VSI Type Use Cases
  - Secure VSI Type Control Use Cases
  - Cross-Data Center VSI Instance Migration Use Cases

- Summary

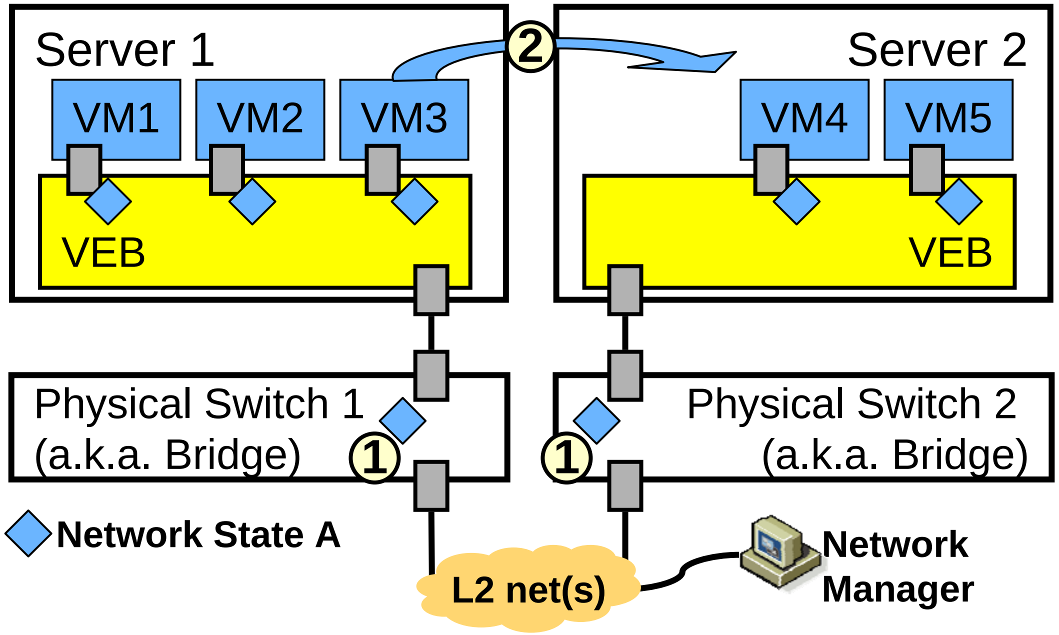
Virtual Ethernet Bridging (VEB)





- Virtual Ethernet Bridging (VEB), a.k.a. Virtual Switching (vSwitch), has been around for decades (e.g. zVM, PowerVM) and can be implemented in:
  - – Software/firmware running in a special Virtual Machine (VM) or embedded in the Hypervisor, which adds code overhead to every IO operation.
  - – In IO adapter hardware (e.g. Power's Host Ethernet Adapter or a PCIe IOV adapter, in which case the IO adapter is directly shared by VMs, enabling Hypervisor bypass.
- In today's VEB model, when a VM migrates from one server to another, all the network state resident in the VEB migrates with VM.
  - – However, the migration of external network's state is more problematic (*more next*).

Homogeneous tenant network state



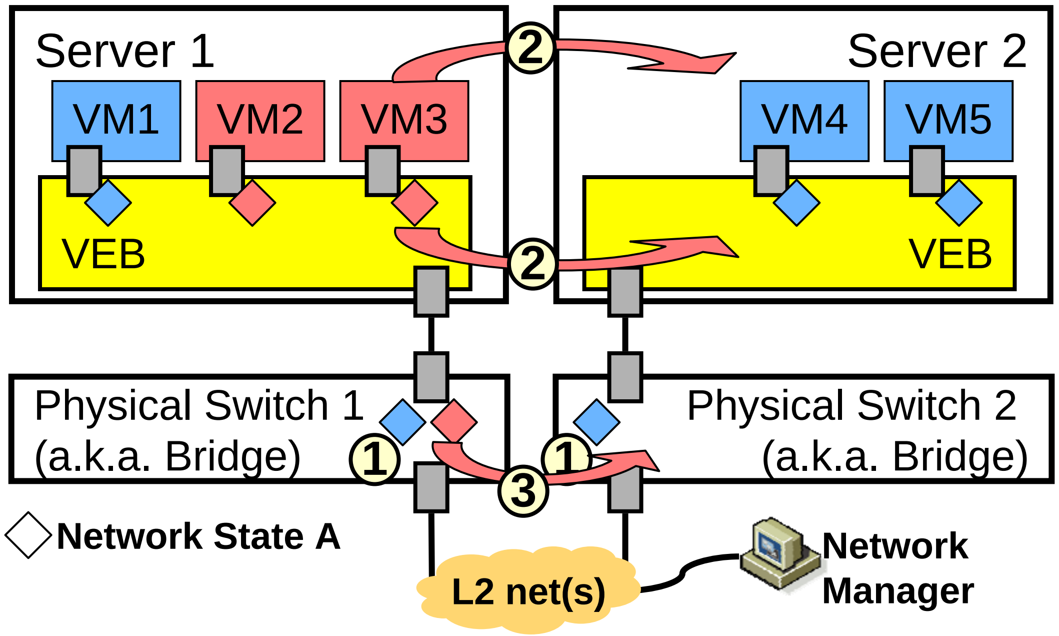
- In this use case,
  - All servers run the same application type (tenant).
    - For example, all Virtual Machines run a Web Serving Application.
  - All VMs have the same network state.

- A network manager is used to configure the same network state (1) on each access switch (***Physical Switch 1 and 2) port*** used by the servers (***Server 1 and 2)*** to access the network.
  - Note: All VMs are associated to the same network state (1) in the Physical Switch.

- Under this use case, when a VM migrates (2) below, no network state needs to migrate, because the same network state is resident in all the access switches.

- Problem: If servers hosting a tenant get over utilized,
VMs cannot migrate to servers hosting a different tenant.

Post VM migration of heterogeneous tenant network state

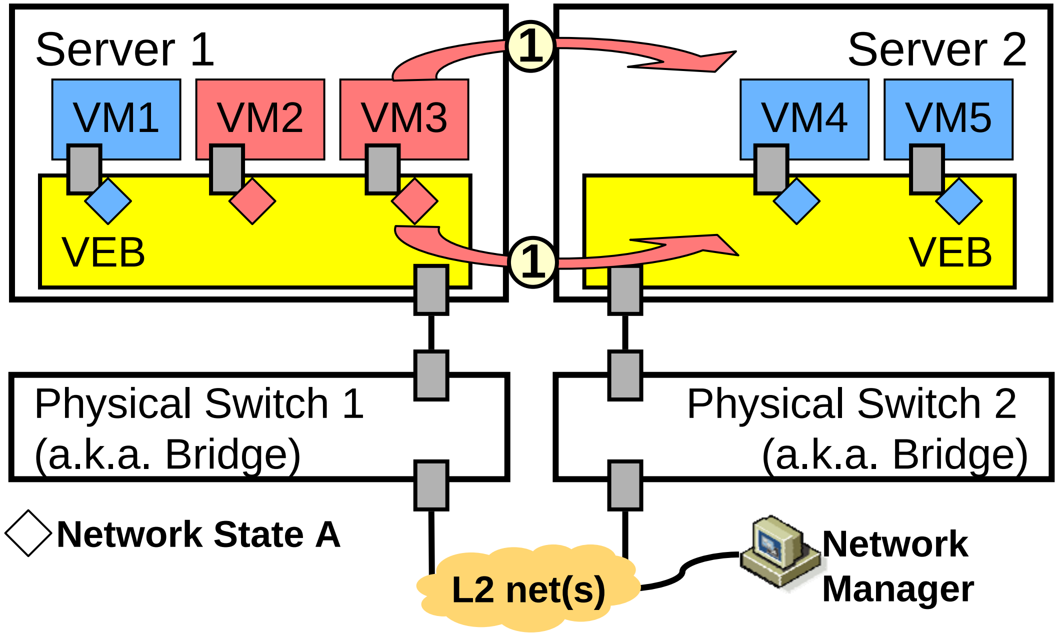


- In this use case:
  - Servers run different application types (tenants) and network SLA moves after the VM.
    - For example, the Virtual Machines (VMs) may be running: Web Serving, e-mail, Database…
    - Each tenant type has the same network state.
    - Different tenant types usually have different network state.

- A network manager is used to configure network state (1) on each access switch port (***Physical Switch 1 and 2)*** used by the servers (***Server 1 and 2)*** to access the network.

- When the Hypervisor migrates a Virtual Machine (VM) migration,
  the Hypervisor migrates all of the network state associated with the VM.
  - When the Physical Switch sees Ethernet frames from the migrated VM (VM3),
    the switch (***Physical Switch 2)*** applies the network state associated VM3's MAC Address.
    Problem: switches cannot distinguish between migrated and re-incarnated VMs.
  - Alternatively, an API can be used between the Hypervisor and the network manager to provide signal the change. Problem: Requires maintaining a per Hypervisor mechanism (which may be proprietary.

Open access for heterogeneous tenant network state

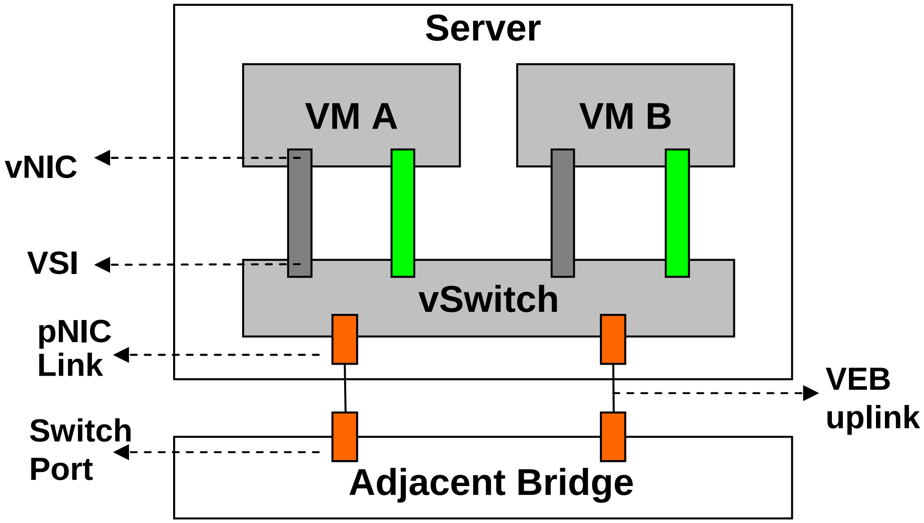


- In this use case:
  - Servers run different application types (tenants).
  - Only Virtual Ethernet Bridges (VEBs), a.k.a. Virtual Switches (vSwitches) have network state.
  - Ports between external switches and the server are configured as trunk ports.

- A network manager is used to configure each access switch port (***Physical Switch 1 and 2)*** as a trunk port.

- The issue with this approach is all physical servers must be in the same security domain, which has similar limitations as the “Homogeneous tenant network state” option.
  - For example, a physical server cannot be managed by tenant A
    in the same layer-2 LAN as a physical server that is managed by tenant B.

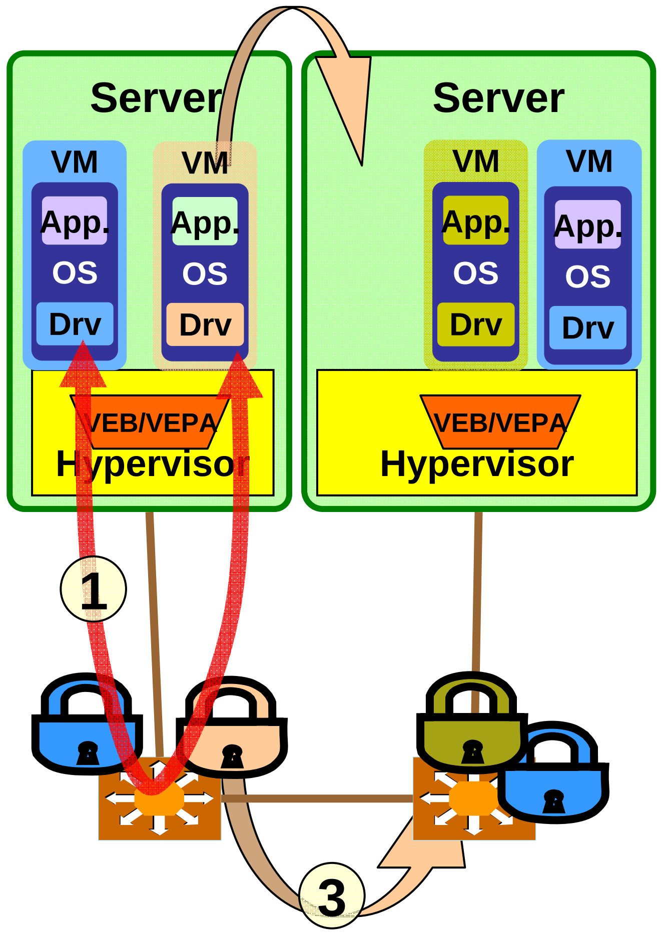
Defining network states and capabilities



- vNIC (virtual Network Interface Controller) - A device that includes a non-forwarding station.
- VSI (Virtual Stations Interface) - A point-to-point Ethernet LAN that connects a VEB or VEPA port to a vNIC's port.
- VSI Type - Network state (e.g. access and traffic controls) associated with a VSI.

- The above figure depicts an example of a Server connected to an Adjacent Bridge.
  - In this example, the Server has two Virtual Machines (VMs), each VM has two virtual NICs (vNIC).
  - Each vNIC connects to the Hypervisor virtual switch (vSwitch) through a Virtual Station Interface (VSI).
- The vSwitch can be configured in a Virtual Ethernet Bridge (VEB) mode, which case the vSwitch performs all VM-VM layer-2 packet forwarding internally.
  - It can also be configured as a Virtual Ethernet Port Aggregator (VEPA), in wich case the Adjacent Bridge is used for layer-2 VM-VM packet forwarding.
  - A single Hypervisor instance may contain one or more such vSwitches.
- Notes:
  - all VEB and VEPA semantics are described in the version 0, Edge Virtual Bridge Proposal to the IEEE 802.1Qbg PAR.
  - Similar to CEE Authors version 0 Data Center Bridging capability eXchange (DCBX) protocol proposals, vendors may implement the version 0, EVB proposal.

Proposed Ethernet Virtual Bridging Protocols

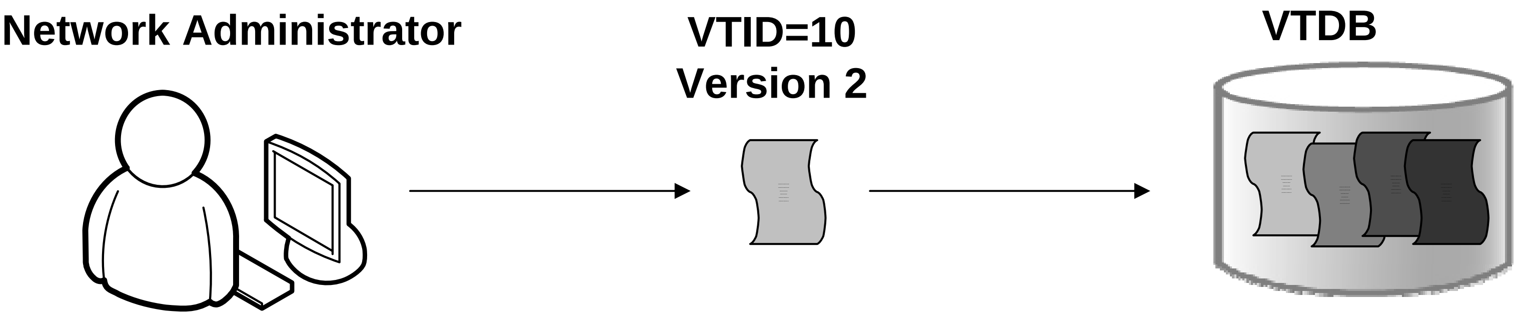


A VSI Type consists of network state associated with the VM, (e.g. Access, QoS & Security Controls). Note: VDP also associates a VLAN ID with the VM.

There are 3 proposed protocols that are used to associate and de-associate a VSI with network state:

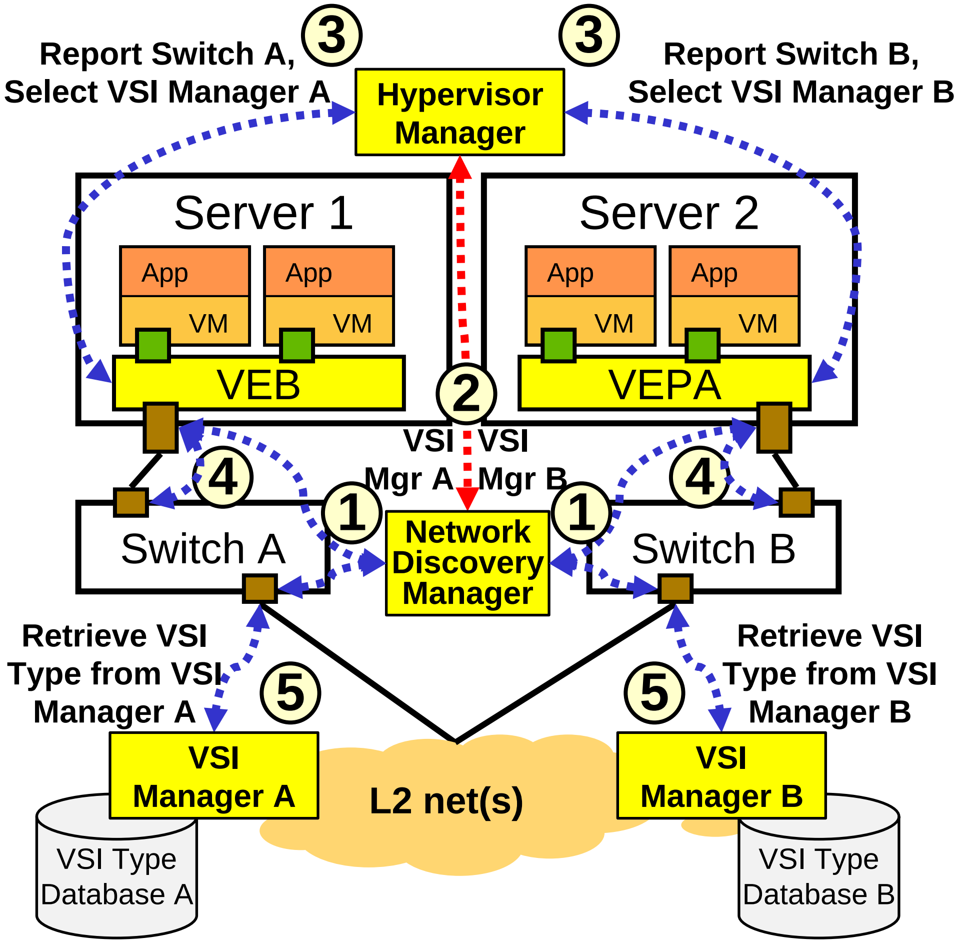
1. Ethernet Virtual Bridging (EVB) Protocol
   - Discovers if adjacent bridge supports VDP and where VM-VM packet forwarding is performed:
     - Within the server, Virtual Ethernet Bridge (VEB); or
     - In external switch, uses a Virtual Ethernet Port Aggregator, VEPA, at the server.
2. Edge Control Protocol (ECP);
   - An acknowledgement based control plane protocol.
     - The acknowledgements are used by the receiver to signal to the sender that an ECP Upper Level Protocol (ULP) buffer is available for the reception an ECP Data Unit.
     - The use of acknowledgements enables the sender to transmit discovery and configuration operations more frequently than possible with timer approaches, such as LLDP.
3. Virtual Station Interface (VSI) Discovery Protocol (VDP)
   - Used to pre-associate, associate and de-associate VM MAC Addresses to a VSI Type [(previously referred to a port profile and VDP as Automated Migration of Port Profile (AMPP)].
   - Enables port profiles (a.k.a. Virtual Station Interface Type) to dynamically migrate with a VM when that VM migrates.

Defining network states and capabilities



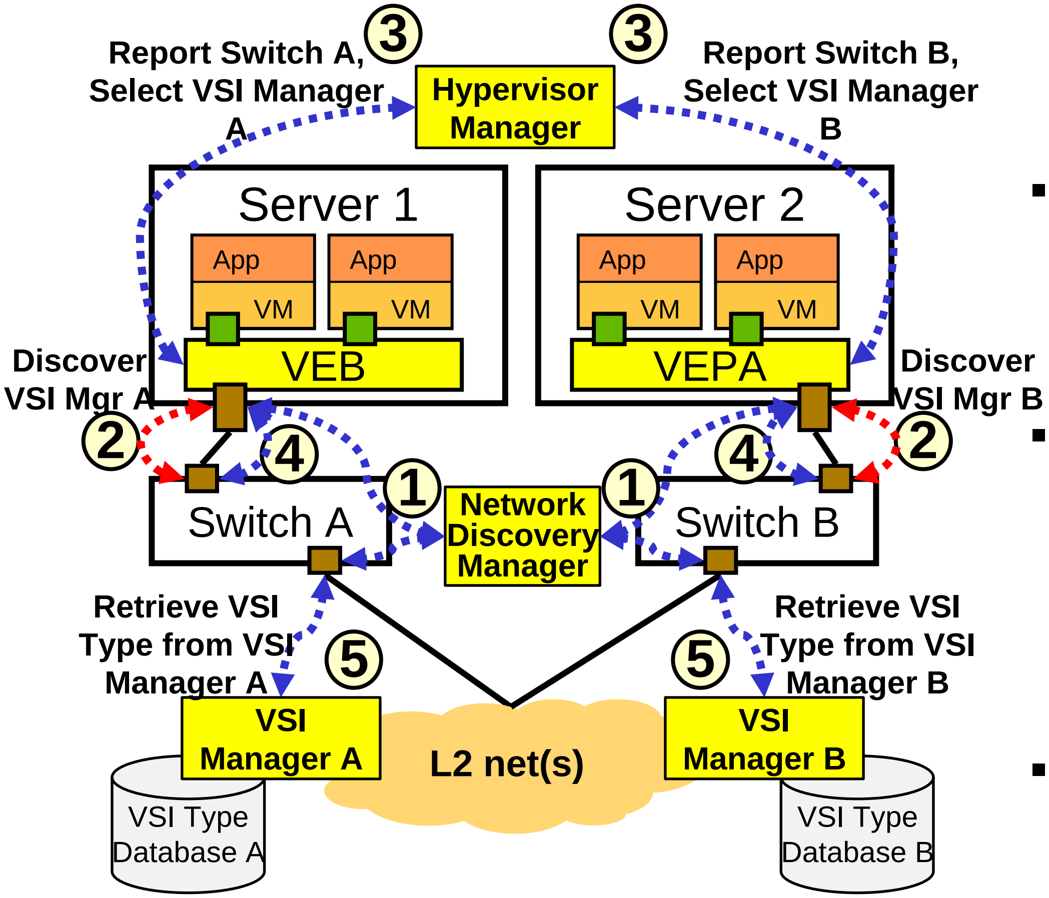
- The above figure depicts an example management model for the VSI Type Database (VTDB).
  - – A network administrator uses a VSI Management user interface to create, change and destroy VSI Types in the VSI Manager's VTDB.
  - – The VSI Manager is expected to be a part of edge switch's Network Change and Configuration Manager (NCCM).
  - – A network administrator can create single VSI Type in the VTDB and use the VSI Type Version field to further refine the VSI Type's controls (*more later*).
- The VSI Manager can be implemented as a vendor specific tool or as a holistic tool, which spans multiple vendors.
  - – If a holistic tool is used, a single VSI Manager Identifier can be used for all switch types.
  - – If the VSI Manager is implemented using a vendor specific tool, the virtualization infrastructure must know which VSI Manager Identifier to use in the VDP exchange.
    - In this case, the virtualization infrastructure can determine which VSI Manager Identifier to use through a management or control plane mechanism (*more next*).

VSI Manager Selection via a Management Plane Mechanism



- In step 1 a Network Discovery Manager is used to determine the topology of the physical network, including what switch each NIC is connected to.
- In step 2, a management interface is used to inform the Hypervisor manager what VSI Manager Identifier must be used in conjunction with each physical NIC discovered by the Network Discovery Manager.
- In step 3, the Hypervisor manager informs the Hypervisor’s VEB or VEPA control plane which VSI Manager must be used for each vNIC.
- In step 4, when the Hypervisor’s VEB or VEPA control plane performs the VDP exchange, it then uses the VSI Manager Identifier passed from the Hypervisor.
- In step 5 the switch uses the VSI Manager Identifier passed through the VDP exchange to retrieve the VSI Type information.

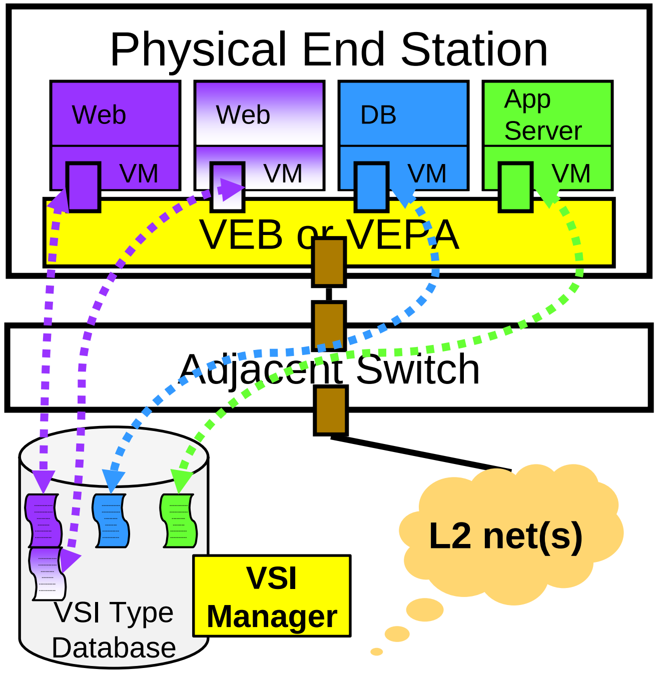
VSI Manager Selection via a Control Plane Mechanism

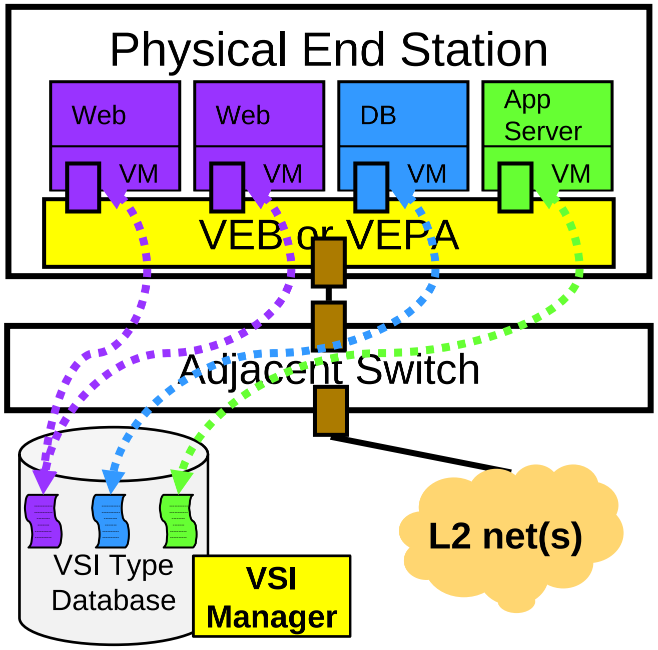


- In step 1 a Network Discovery and Configuration Manager is used to determine the physical network topology & configure the VSI Manager Identifier and location on each switch used by servers to access the network.
- In step 2, a control plane protocol (e.g. EVB) is used by the server's virtualization infrastructure to discover the VSI Manager Identifier that must be used by VDP.
- In step 3, for each NIC, the Hypervisor's VEB or VEPA control plane reports to the Hypervisor manager the VSI Manager Identifiers it has discovered.
  - The Hypervisor Manager returns the rest of the VSI state to use for VDP.
- In step 4, the Hypervisor's VEB or VEPA control plane performs VDP.
- In step 5, the switch uses the VSI Manager Identifier passed in the VDP exchange to retrieve the VSI Type information.

VSI Type Database Use Cases

- As described in the version 0, Edge Virtual Bridge Proposal to the IEEE 802.1Qbg PAR,
  - –The VSI Type Database (VTDB) can use any combination of the VSI Type Identifier, VSI Type Version and VSI Instance Identifier to retrieve a VSI Type from the database.
  - –Enables a very flexible set of database schema use cases (*see paper for additional examples not covered here*).





- A simple VSI schema is to just use the VSI Type field as an index into the VTDB.
  - Each VSI Type is associated with a VM type.
  - In the example above, each VM types (Web Server, Application Server and Database Server) has a specific, corresponding VSI Type.

- The VSI Type also define base set of access, traffic, and security controls; and the VSI Type Version to define deltas from that base.
  - For example, two types of web server application may be needed, where the only difference between the two is in the traffic rate limiting control.
  - In this case, the same VSI Type can be used for both VM Types and the VSI Version defines the delta, rate limiting control.

Push vs Pull VSI Models

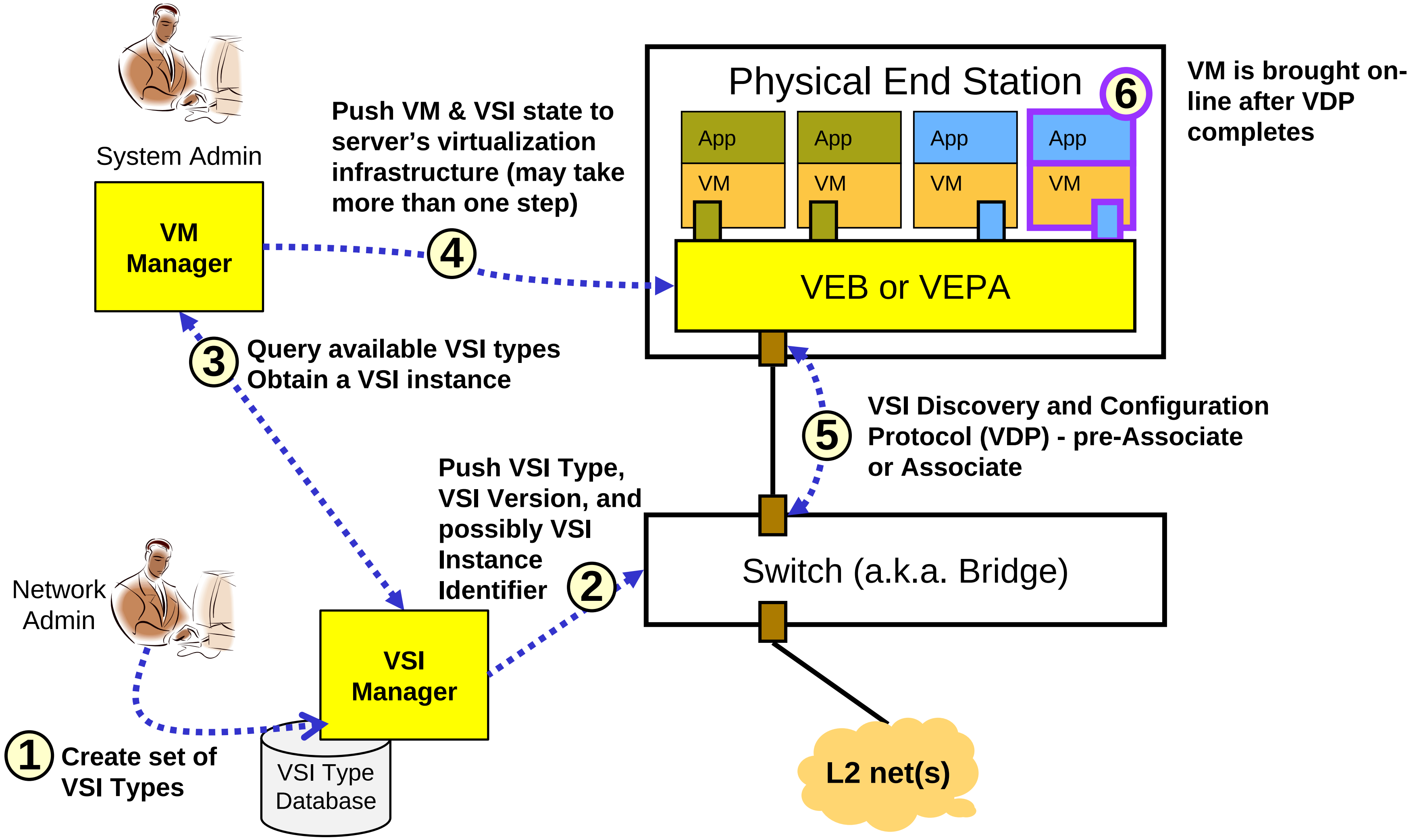
- Two models can be used to load VSI Type state into physical switches:
  - A push model, where all the VSI Types are loaded into network switches during network discovery/configuration.
  - A pull model, as defined in the version 0, Edge Virtual Bridge Proposal to the IEEE 802.1Qbg PAR.

- The push model works well when the state associated with the full set of VSI Types is less than the state each physical switch can maintain.
  - It also provides faster pre-association and association, because all state is resident in the physical switch (vs retrieved from the VSI Manager).

- The pull model works well when the state associated with the full set of VSI Types is greater than the state each physical switch can maintain.

High Level VDP Push Model Use Case

VM Creation

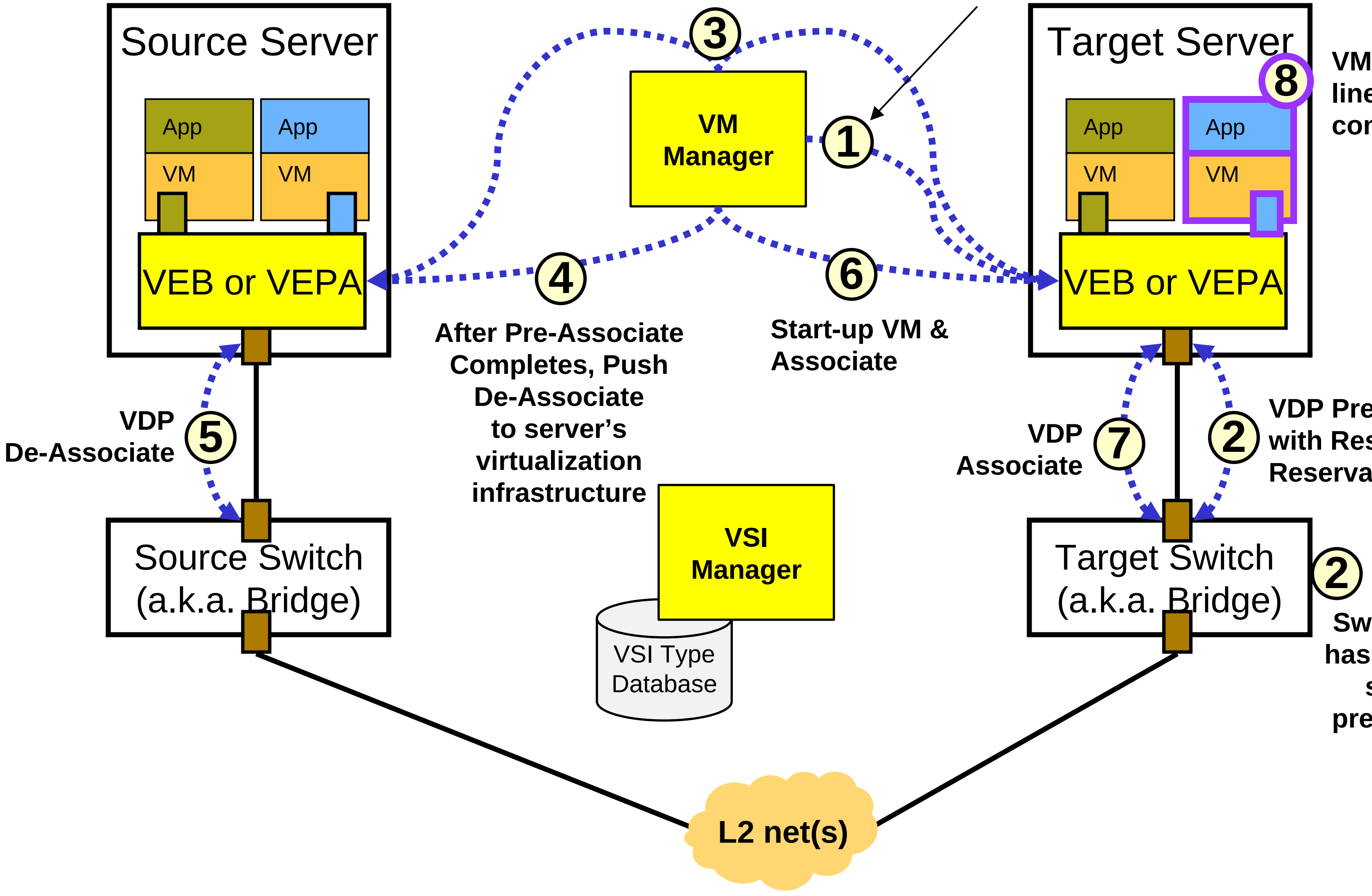


System Admin
VM Manager
Push VM & VSI state to server's virtualization infrastructure (may take more than one step)
4
Physical End Station
6
App
App
App
App
VM
VM
VM
VM
VEB or VEPA
VM is brought on-line after VDP completes
3
Query available VSI types
Obtain a VSI instance
5
VSI Discovery and Configuration Protocol (VDP) - pre-Associate or Associate
Push VSI Type, VSI Version, and possibly VSI Instance Identifier
2
Switch (a.k.a. Bridge)
Network Admin
VSI Manager
1
Create set of VSI Types
VSI Type Database
L2 net(s)

High Level VDP Push Model Use Case

VM Migration

Push Associate or Pre-Associate with resource reservation to target server's virtualization infrastructure.



VM is brought on-line after VDP completes

After Pre-Associate Completes, Push De-Associate to server's virtualization infrastructure

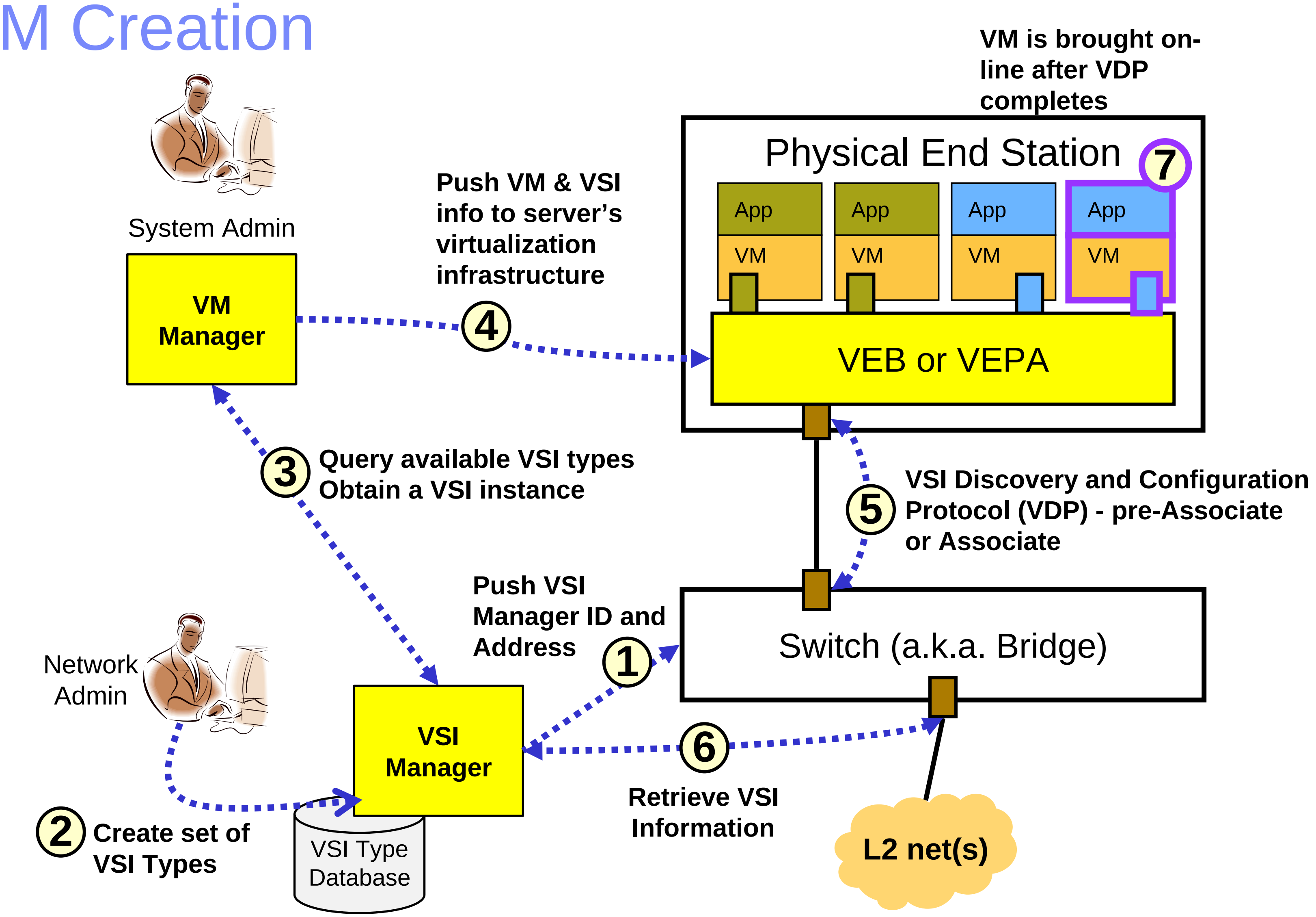
Start-up VM & Associate

VDP Pre-Associate with Resource Reservation

Switch already has all VSI Type state (see previous slide)

High Level VDP Pull Model Use Case

VM Creation


VM is brought on-line after VDP completes

Push VM & VSI info to server's virtualization infrastructure

Query available VSI types
Obtain a VSI instance

VSI Discovery and Configuration Protocol (VDP) - pre-Associate or Associate

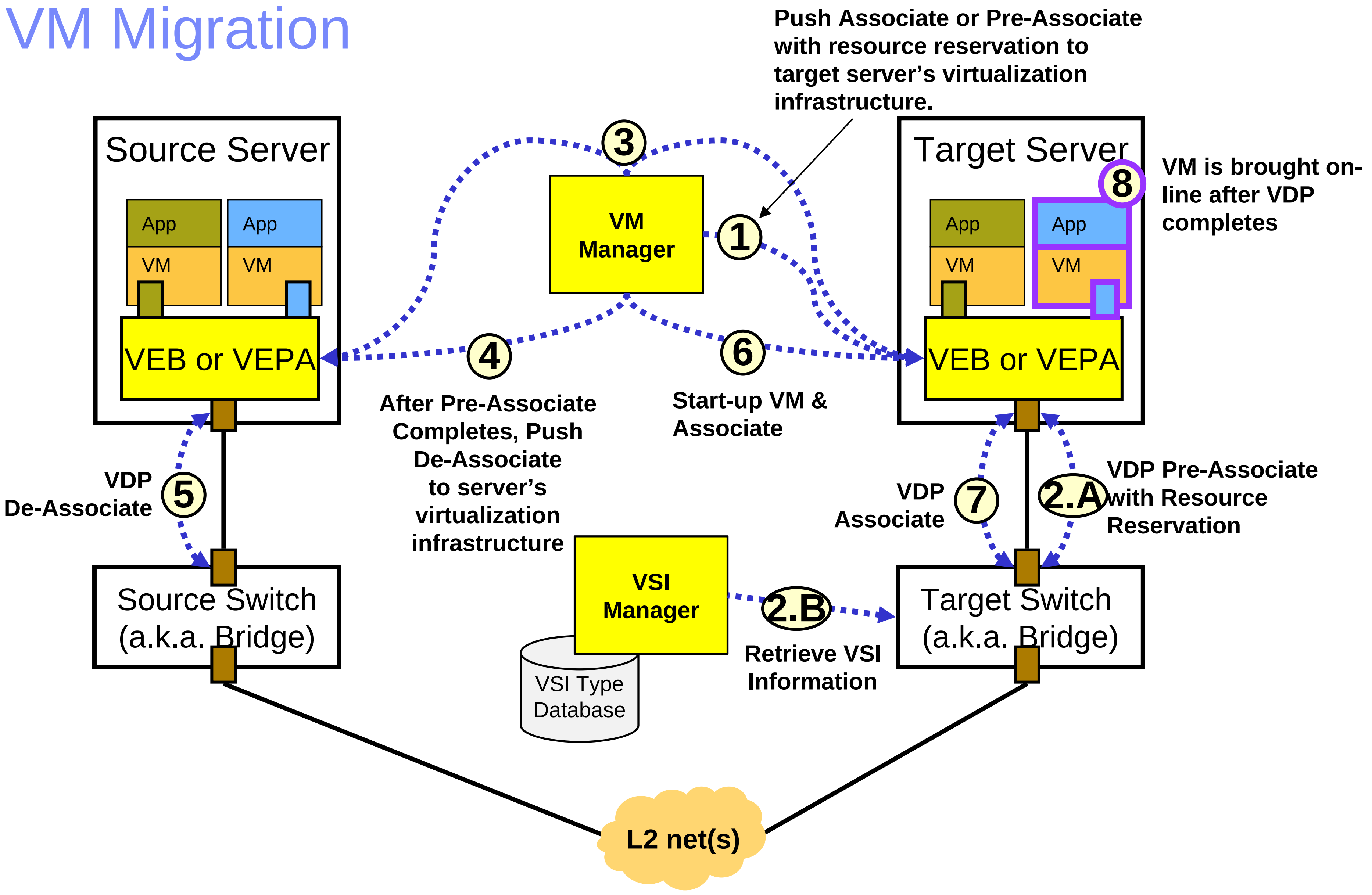
Push VSI Manager ID and Address

Create set of VSI Types

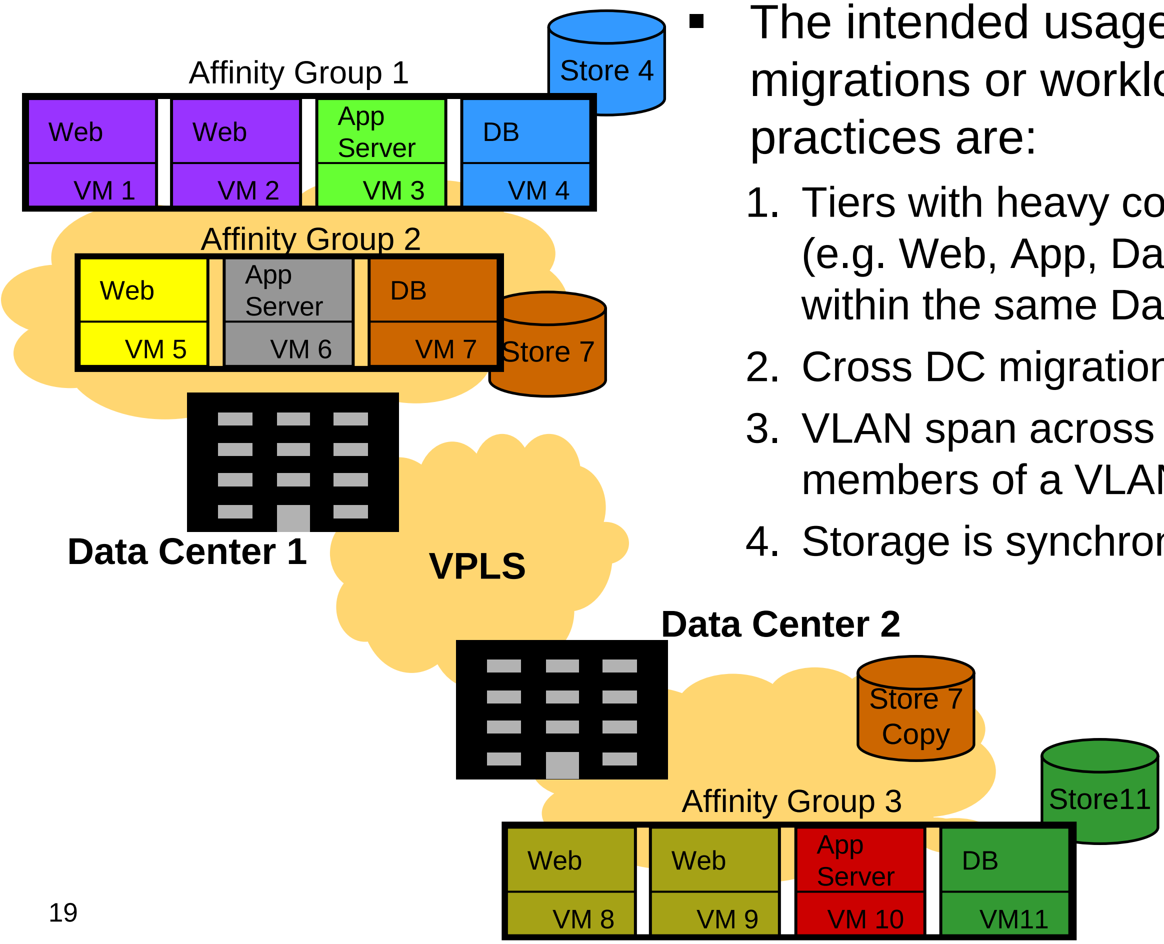
Retrieve VSI Information

High Level VDP Pull Model Use Case

VM Migration


Push Associate or Pre-Associate with resource reservation to target server's virtualization infrastructure.
Source Server
App
App
VM
VM
VEB or VEPA
3
VM Manager
1
4
6
Target Server
8
VM is brought on-line after VDP completes
App
App
VM
VM
VEB or VEPA
After Pre-Associate Completes, Push De-Associate to server's virtualization infrastructure
Start-up VM & Associate
VDP De-Associate
5
VDP Associate
7
2.A
VDP Pre-Associate with Resource Reservation
Source Switch (a.k.a. Bridge)
VSI Manager
2.B
Target Switch (a.k.a. Bridge)
VSI Type Database
Retrieve VSI Information
L2 net(s)

Cross-Data Center VSI Instance Migration Use Cases



- The version 0, Edge Virtual Bridge Proposal to the IEEE 802.1Qbg PAR can be used in conjunction with layer-2 extension technologies, such as Virtual Private LAN Service (VPLS) over a Multi-Protocol Label Switching service provider infrastructure.
- The intended usage model is planned workload migrations or workload redundancy, where best practices are:
  1. Tiers with heavy communication patterns (e.g. Web, App, Database) are kept within the same Data Center (DC) vs across DCs.
  2. Cross DC migration keeps tiers together.
  3. VLAN span across Data Centers, but based on 1-2, all members of a VLAN are in the same DC.
  4. Storage is synchronously copied across DC sites.

Summary

- The version 0, Edge Virtual Bridge Proposal to the IEEE 802.1Qbg PAR use cases described in this paper enable the automation of network state configuration, such as port access, traffic and security flow controls.

- These use cases also enable VM migration that includes the automated migration of all the network state associated with a VM
  - Versus just the migration of internal network state.

- The version 0, Edge Virtual Bridge Proposal enables dynamic multi-tenant environments, with per tenant type network controls that automatically migrate in conjunction with tenant migration.

- Unlike other alternatives, the version 0, Edge Virtual Bridge Proposal to the IEEE 802.1Qbg PAR provides a proposal for standardizing these mechanisms.