



*TOMORROW  
starts here.*

Cisco *live!*



# Integration and Interoperation of Existing Nexus Networks into an ACI Architecture

BRKACI-2001

Mike Herbert – Principal Engineer INSBU

#clmel

Cisco *live!*

# Introducing: Application Centric Infrastructure (ACI)

Open + Secure



Apps + Infrastructure

SharePoint

Exchange

SAP

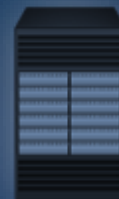
ORACLE



On-Premises + Cloud



Physical + Virtual + Containers



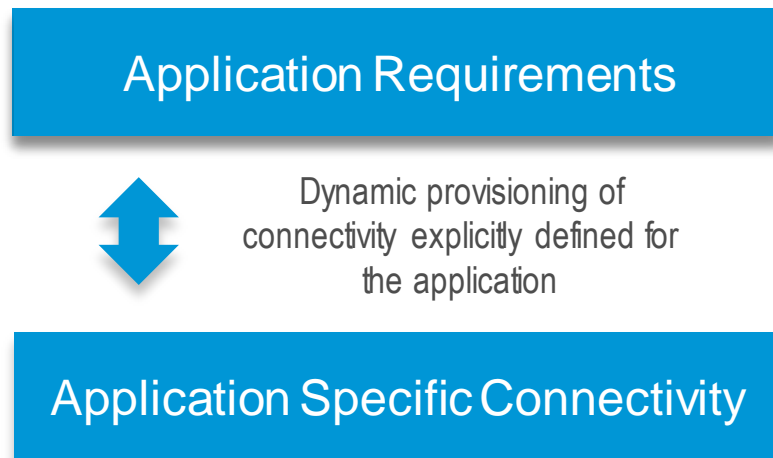
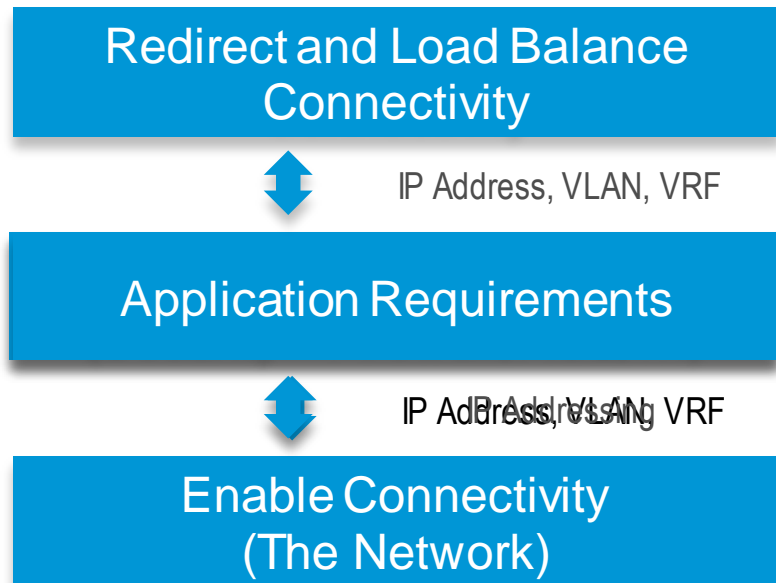
**Application Oriented Policy** = Operational Simplicity



# Why Networks Are Complex

## Overloaded Network Constructs

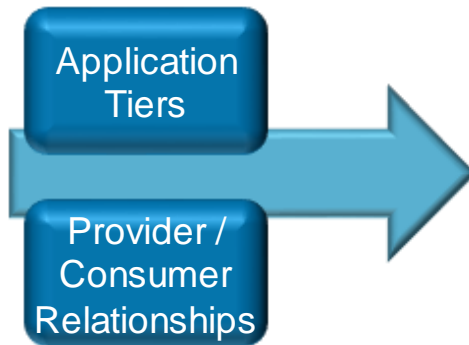
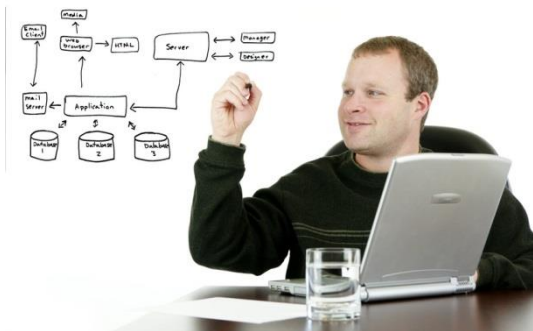
ACI directly maps the application connectivity requirements onto the network and services fabric



# Why Network Provisioning is Slow

## Application Language Barriers

Developers

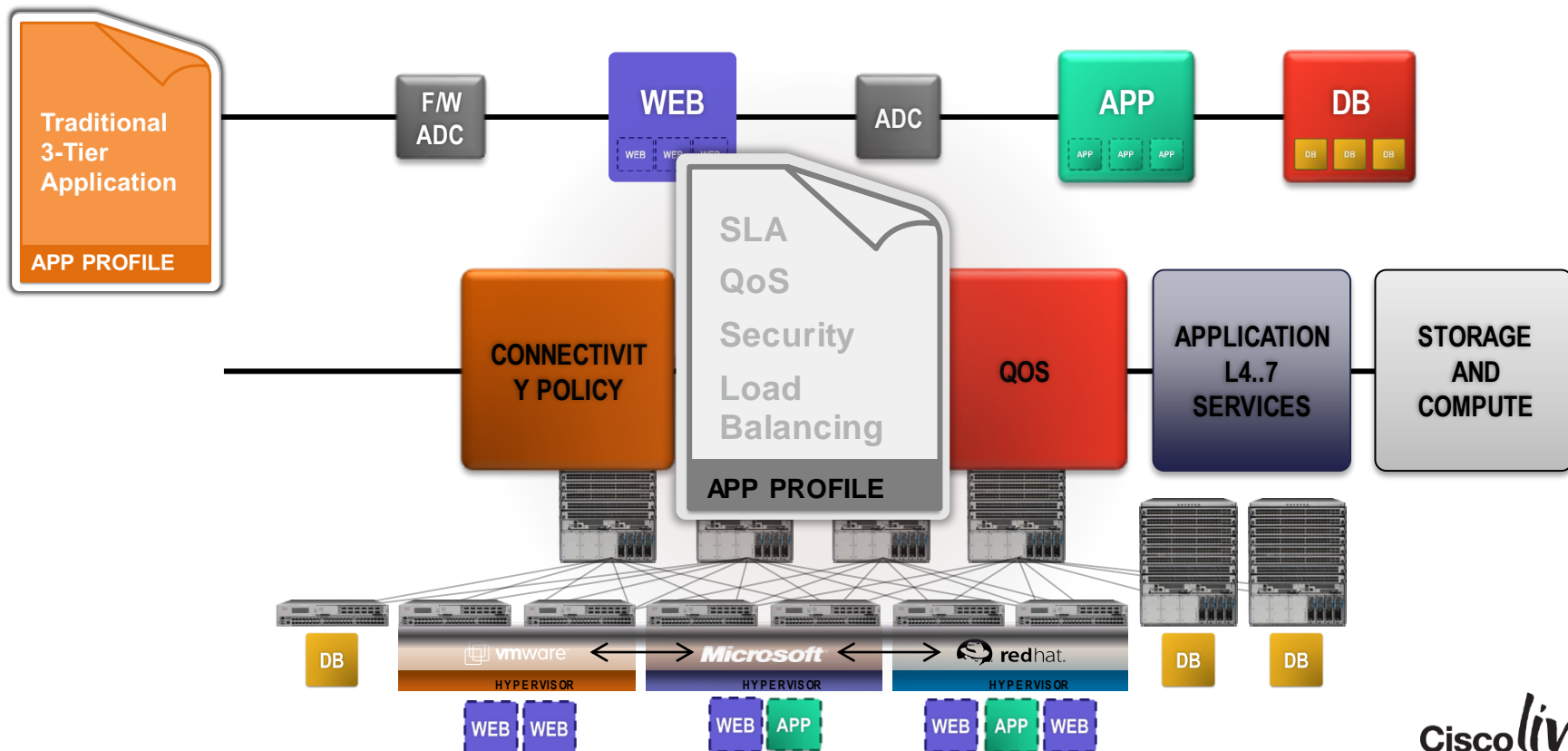


Infrastructure Teams



Developer and infrastructure teams must translate between disparate languages.

# Centralised Policy and Distributed Enforcement

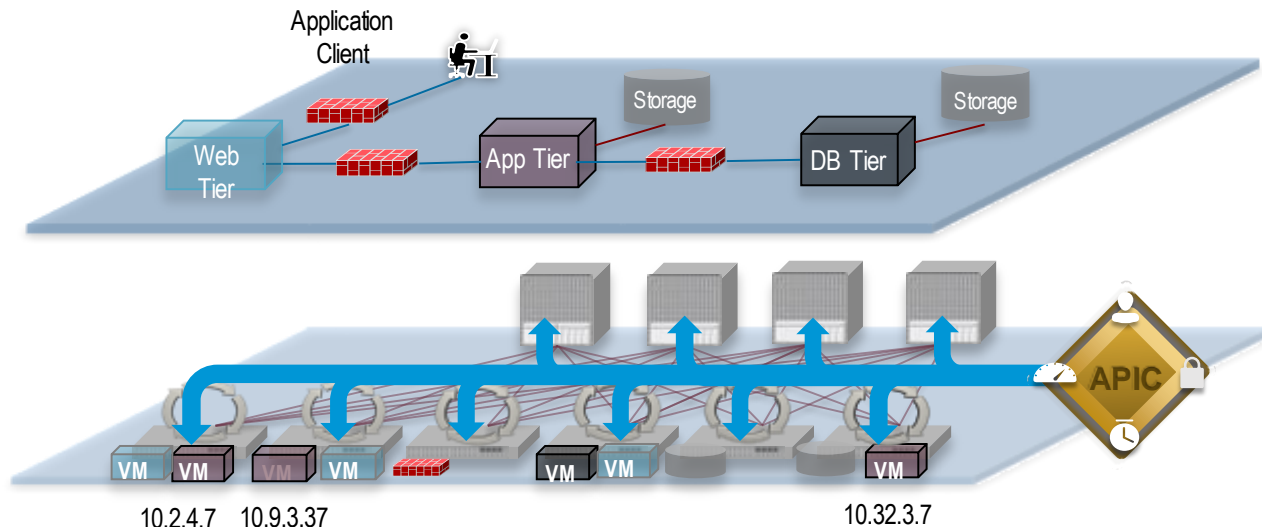


# ACI – 21<sup>st</sup> Century Distributed Systems in Action

**Application Policy Model:** Defines the application requirements (Application Network Profile)



**Policy Instantiation:** Each device dynamically instantiates the required changes based on the policies



- All forwarding in the fabric is managed via the Application Network Profile
  - IP addresses are fully portable *anywhere* within the fabric
  - Security & Forwarding are fully *decoupled* from any physical or virtual network attributes
  - Devices autonomously update the state of the network based on configured policy requirements

Cisco *live!*

# Two Big Questions

“Is ACI a Closed System?”

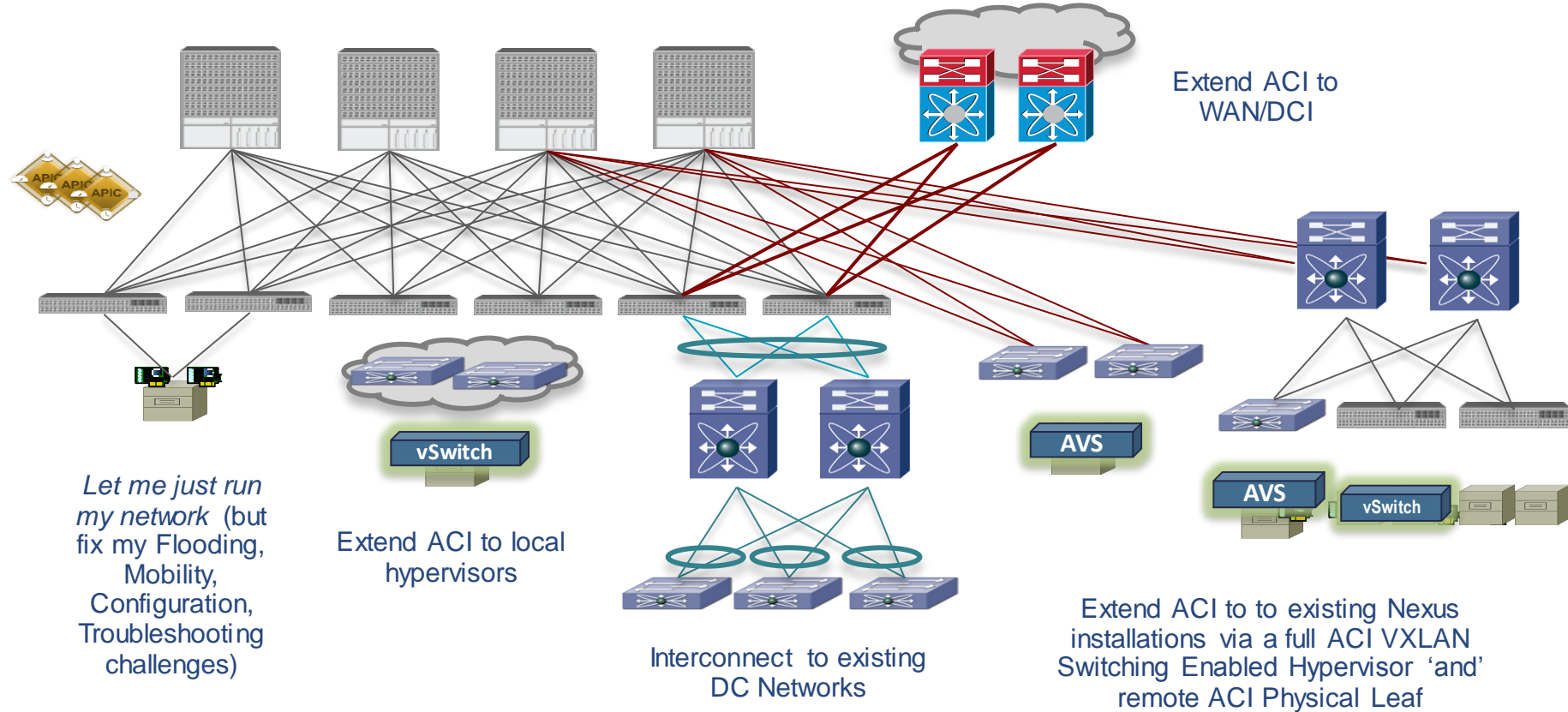
“Do I need to replace all of my existing infrastructure to begin leveraging ACI?”



**ABSOLUTELY NOT !!!**  
**Let's see WHY and HOW ...**



# Things We Would Like To Understand How To Do



A long-exposure photograph of a city street at night. The foreground is filled with vibrant, multi-colored light trails from moving vehicles, creating a sense of motion. In the background, a modern city skyline is visible with illuminated buildings and a pedestrian bridge crossing the street. The overall scene is a blend of urban architecture and dynamic light patterns.

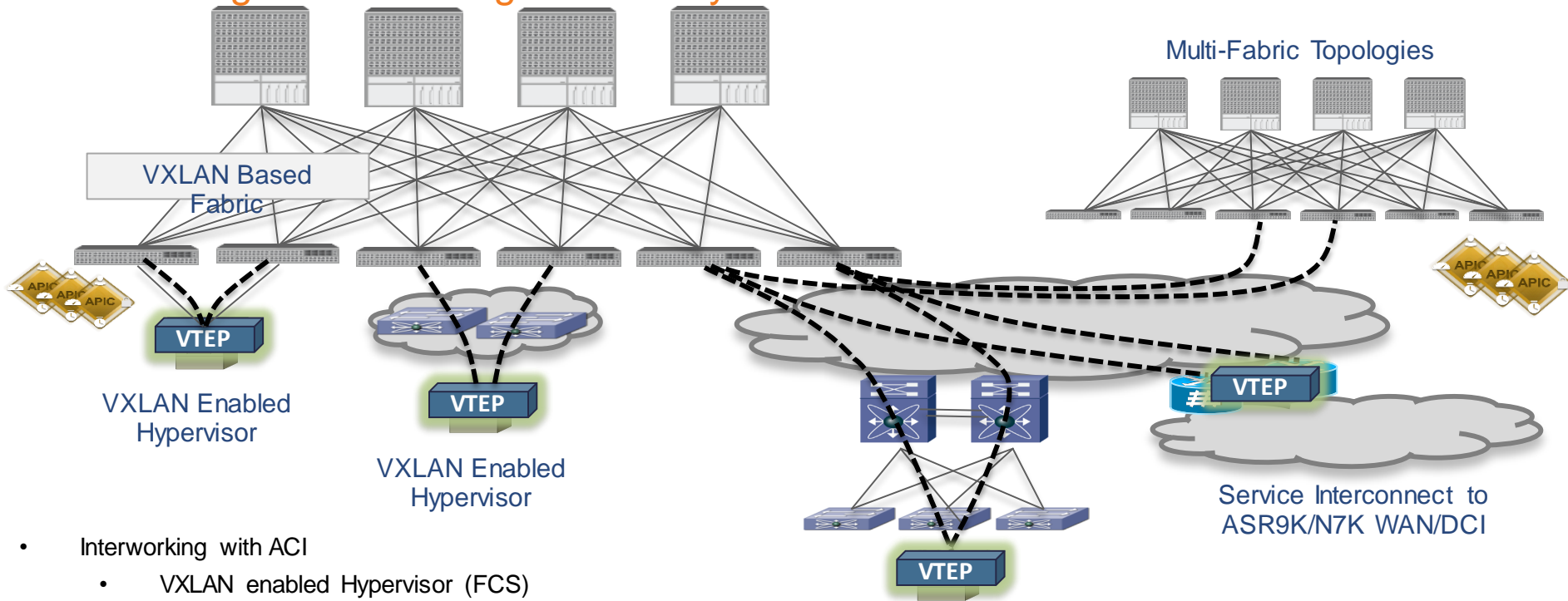
# ACI Policy Based Forwarding via an Integrated Overlay





# ACI Fabric – Integrated Overlay

## Connecting and Extending the Overlay



- Interworking with ACI
  - VXLAN enabled Hypervisor (FCS)
  - VXLAN Hardware VTEP (Nexus 9000 standalone, Nexus 3100/7000-F3, ASR9K, ...)
  - MP-BGP EVPN based control plane for external VTEP connectivity (post FCS)

## Decoupled Identity, Location & Policy

- [illegible]



# ACI Leverages VXLAN RFC Draft – Next IETF Meeting

NV03  
Internet-Draft  
Intended status: Standards Track  
Expires: April 30, 2015

M. Smith  
Cisco Systems, Inc.  
October 27, 2014

VXLAN Group Policy Option  
draft-smith-nv03-vxlan-group-policy-00

## Abstract

This document defines an extension to Virtual eXtensible Local Area Network (VXLAN) that allows an Endpoint Group Identifier to be carried for the purposes of policy application.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

Internet-Draft

VXLAN Group Policy Option

October 2014

## 1.2. Definition of Terms

EPG: Endpoint Group.  
VTEP: VXLAN Tunnel End Point  
VXLAN: Virtual EXtensible Local Area Network.

## 2. Approach

### 2.1. VXLAN Group Based Policy Extension

The VXLAN Group Based Policy Extension (VXLAN-GBP) header is defined as:

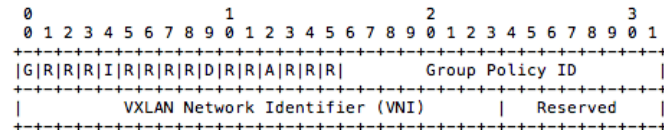


Figure 1: VXLAN-GBP Extension

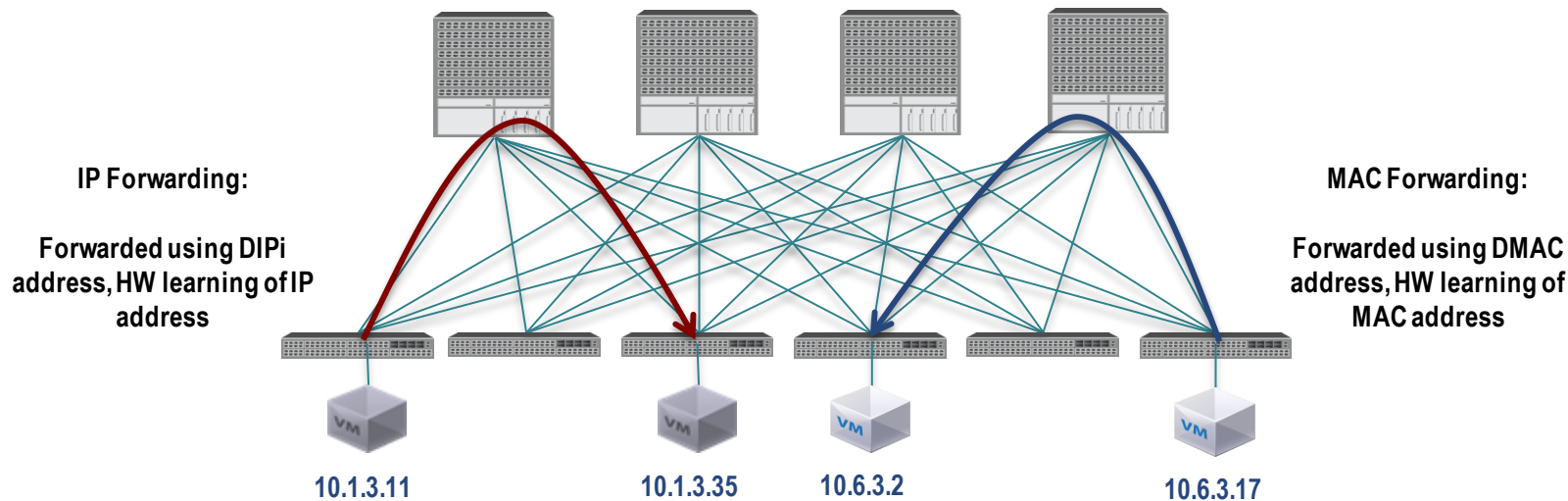
The following bits are defined in addition to the existing VXLAN fields:

G Bit: Bit 0 of the initial word is defined as the G (Group Based Policy Extension) bit.

G = 1 indicates that the Group Policy Identifier is being carried within the Group Policy ID field as defined in this document.

G = 0 indicates that the Group Policy Identifier is not being carried, and the G Bit MUST be set to 0 as specified in [TBD-VXLANxref].

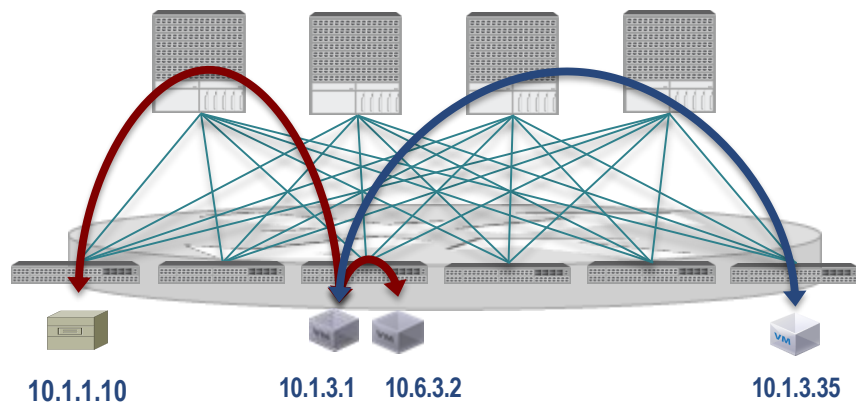
# Overlay Elements – Host Forwarding



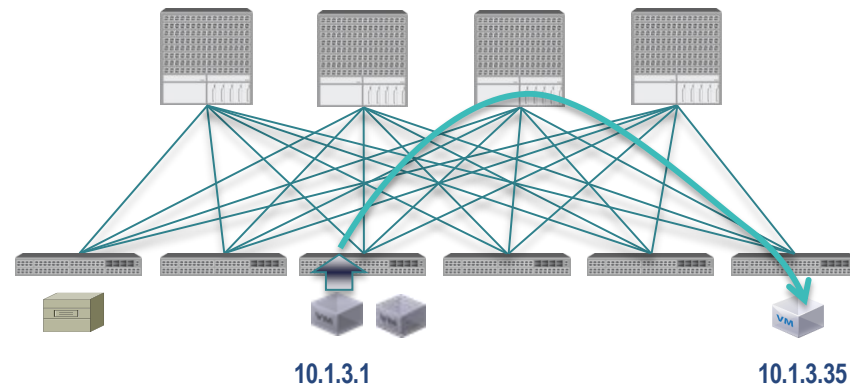
- Forward based on destination IP Address for intra and inter subnet (Default Mode)
  - Bridge semantics are preserved for intra subnet traffic (no TTL decrement, no MAC header rewrite, etc.)
  - Non-IP packets will be forwarded using MAC address. Fabric will learn MAC's for non-IP packets, IP address learning for all other packets
- Route if MAC is router-mac, otherwise bridge (standard L2/L3 behaviour)

# Location Independent Forwarding

## Layer 2 and Layer 3



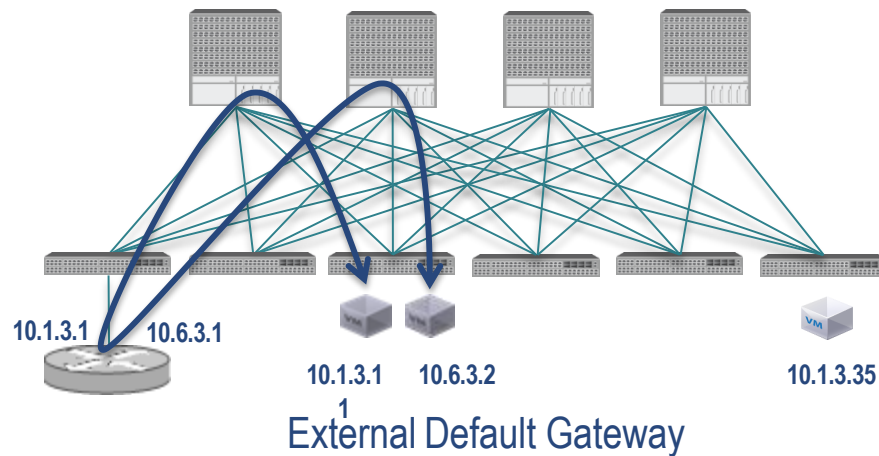
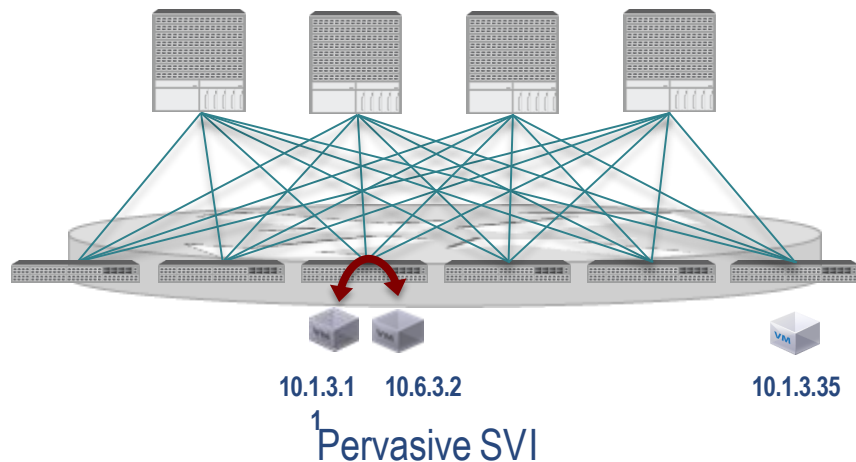
1  
Distributed Default Gateway



1  
Directed ARP Forwarding

- ACI Fabric supports full layer 2 and layer 3 forwarding semantics, no changes required to applications or end point IP stacks
- ACI Fabric provides optimal forwarding for layer 2 and layer 3
  - Fabric provides a pervasive SVI which allows for a distributed default gateway
  - Layer 2 and layer 3 traffic is directly forwarded to destination end point
- IP ARP/GARP packets are forwarded directly to target end point address contained within ARP/GARP header (elimination of flooding)

# Where is My Default Gateway?



- Default Gateway can reside internal or external to the Fabric
- Pervasive SVI provides a distributed default gateway (anycast gateway)
  - Subnet default gateway addresses are programmed in all Leaves with end points present for the specific Tenant IP subnet
  - Layer 2 and layer 3 traffic is directly forwarded to destination end point
- External Gateway is used when Fabric is configured to provide layer 2 transport only for a specific Tenant

# Overlay Elements - Mapping/Directory Proxy

Inline Hardware Mapping DB - 1,000,000+ hosts

Global Station Table contains a local cache of the fabric endpoints

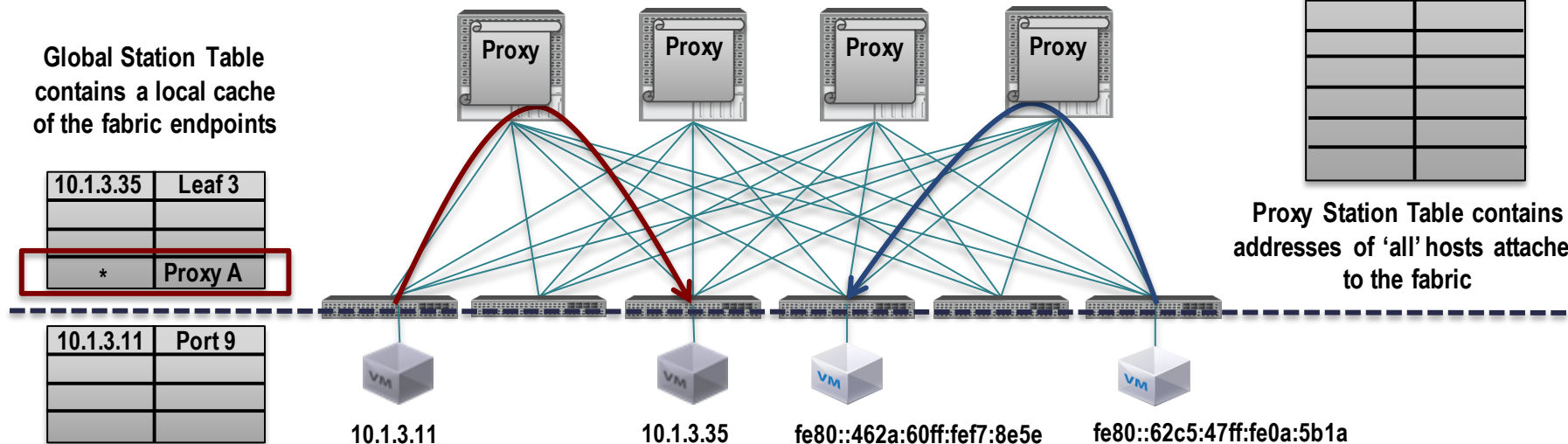
10.1.3.35	Leaf 3
*	Proxy A

10.1.3.11	Port 9

Local Station Table contains addresses of 'all' hosts attached directly to the iLeaf

10.1.3.35	Leaf 3
10.1.3.11	Leaf 1
fe80::8e5e	Leaf 4
fe80::5b1a	Leaf 6

Proxy Station Table contains addresses of 'all' hosts attached to the fabric



- The Forwarding Table on the Leaf Switch is divided between local (directly attached) and global entries
- The Leaf global table is a cached portion of the full global table
- If an endpoint is not found in the local cache the packet is forwarded to the 'default' forwarding table in the spine switches (1,000,000+ entries in the spine forwarding table)



# ACI

## Policy Defined Networking

Remove the problems  
of forcing the network  
to fit

Forwarding is defined by  
Policy EPG 'Web' can  
talk to EPG 'DB'  
independent of IP  
subnet, VLAN/VXLAN,  
VRF if Policy says it  
should

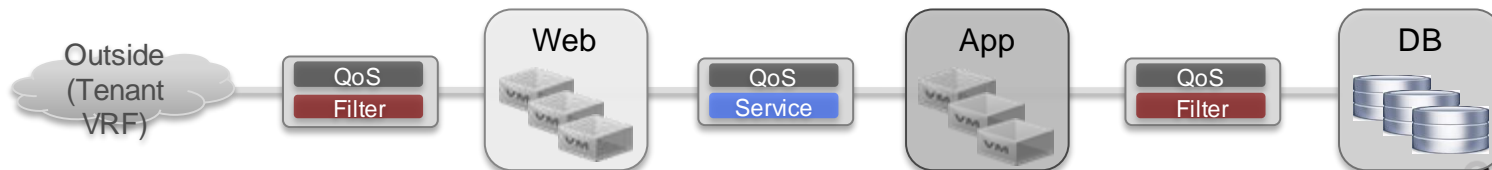
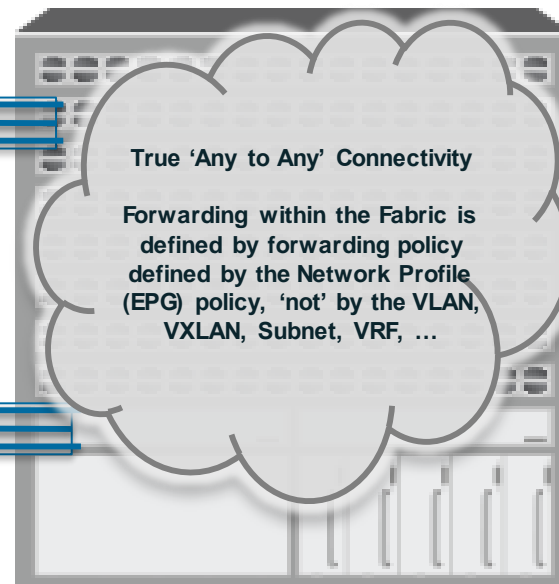
NVGRE  
VSID 5165

802.1Q  
VLAN 55

VXLAN  
VNID 8765

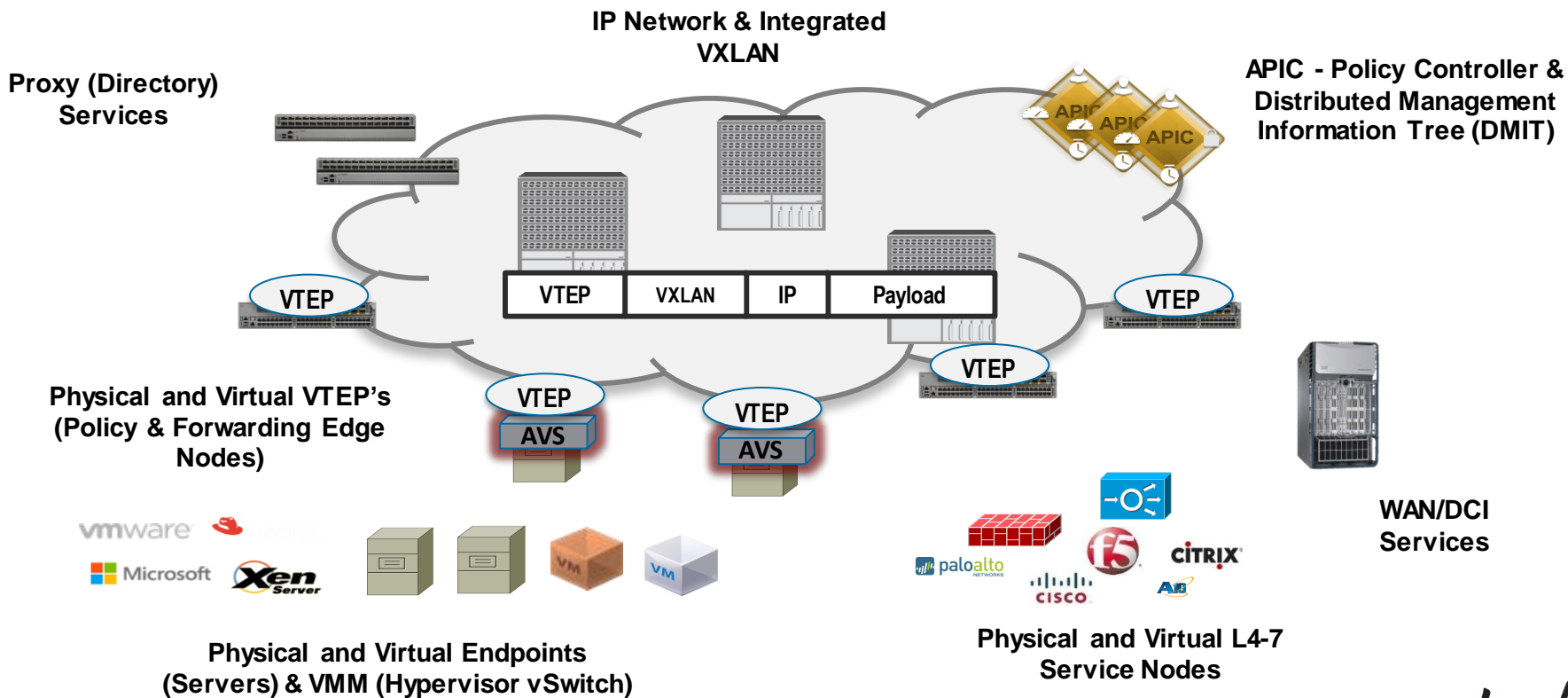
10.10.11.12  
VRF Shared  
10.10.11.12  
VRF Retail Bank

192.168.11.3  
VRF Storage



# ACI

## A Policy Based IP Network



A long-exposure photograph of a city street at night. The foreground is filled with vibrant, multi-colored light trails from moving vehicles, creating a sense of motion. In the background, a modern pedestrian bridge with blue lighting spans the street. Tall buildings with illuminated windows and storefronts line the street, and several flags are visible on poles to the left.

# Extending The Network

# Transitions Will and Need To Occur Independently

## Operations Evolution



## Policy Evolution

Policy Zone 'A'

Policy Zone 'B'

Policy Zone 'C'



## Component Evolution





# Extending ACI in to Current Data Centre's Three Stages

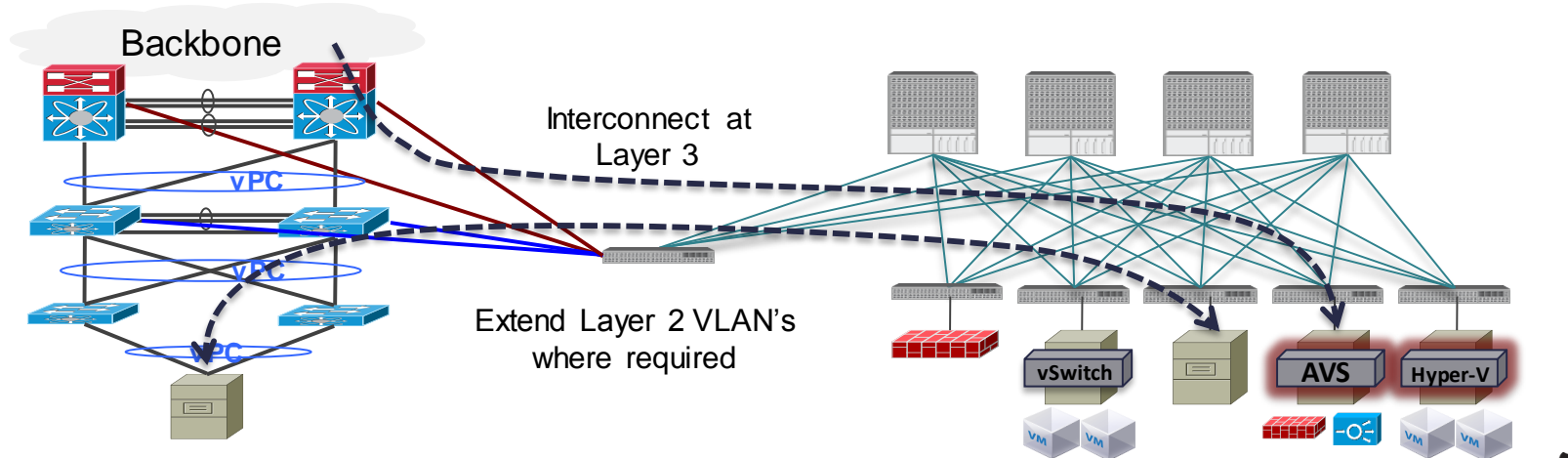
- Interconnect existing network POD's with new ACI POD's via standard Layer 2 extensions (VLAN or VXLAN) or via standard Layer 3 routing (OSPF, BGP)
- Leverage an ACI policy/services block attached to any existing Nexus or Catalyst network to provide L4-L7 services and policy automation for existing virtual and physical servers
- Extend the ACI forwarding and distributed policy capabilities to virtual and physical leaf switches connected to an existing Nexus/Catalyst IP based network



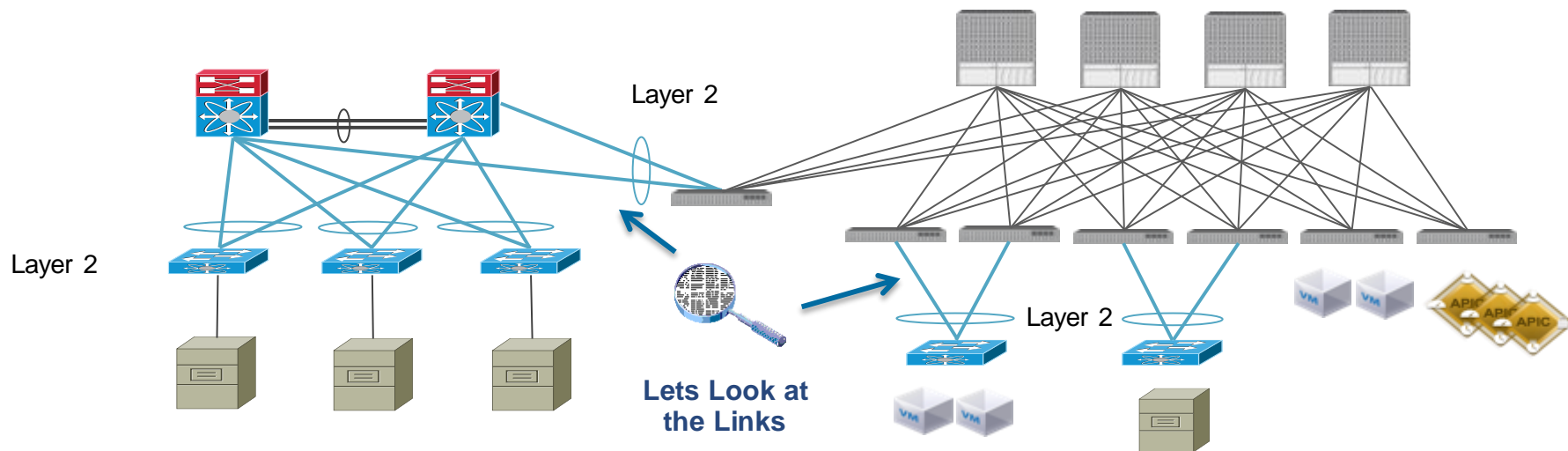
# Extending ACI in to Current Data Centre's

## Interconnect Existing to New ACI POD

- Layer 2 and Layer 3 interoperation between ACI Fabric and Existing Data Centre builds
- Layer 3 interconnect via standard routing interfaces, OSPF, Static, iBGP (FCS) MP-BGP, EIGRP, ISIS (Post FCS\_
- Layer 2 interconnect via standard STP or via VXLAN overlays



# Connecting/Extending ACI via Layer 2

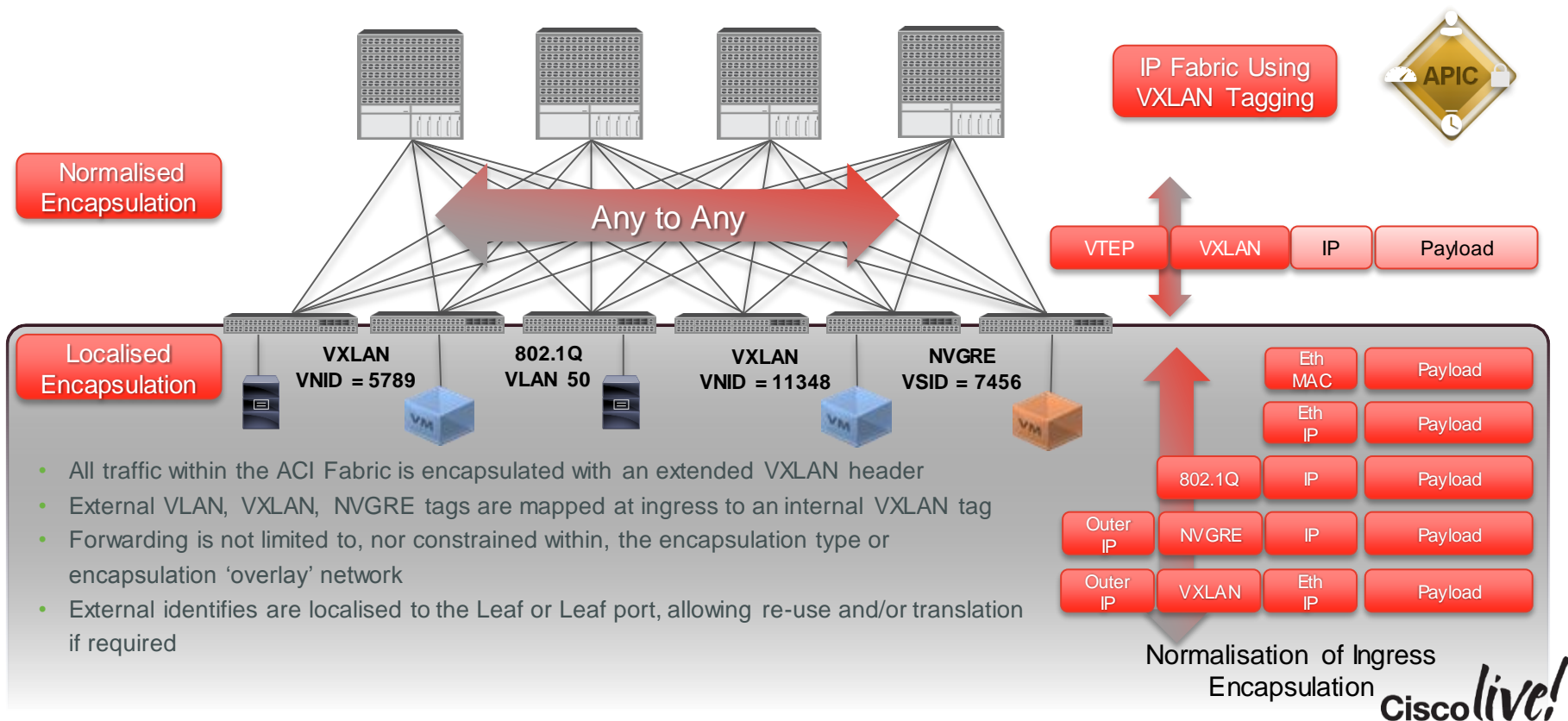


## Extend L2 domain beyond ACI fabric - 2 options

1. Manually assign a port to a VLAN which in turn mapped to an EPG. This **extend EPG beyond ACI fabric (EPG == VLAN)**
2. Create a L2 connection to outside network. **Extend bridge domain** beyond ACI fabric. Allow contract between EPG inside ACI and EPG outside of ACI

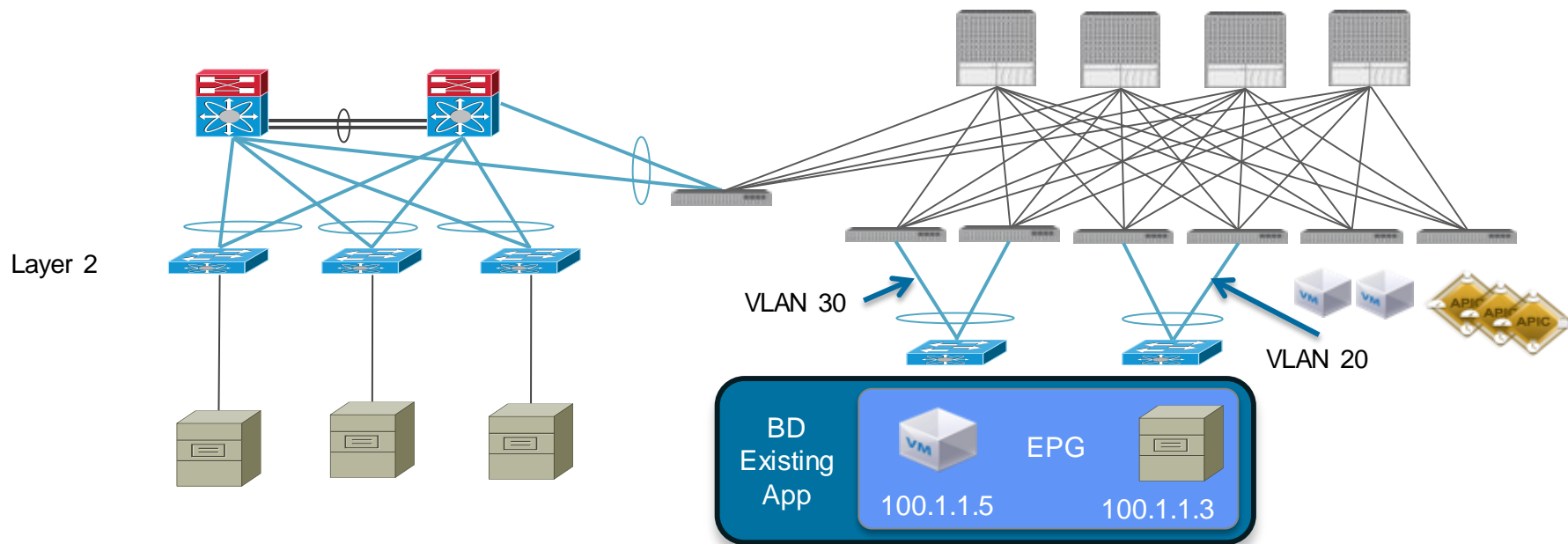
# ACI Fabric – Integrated Overlay

## Data Path - Encapsulation Normalisation



# Extend the EPG

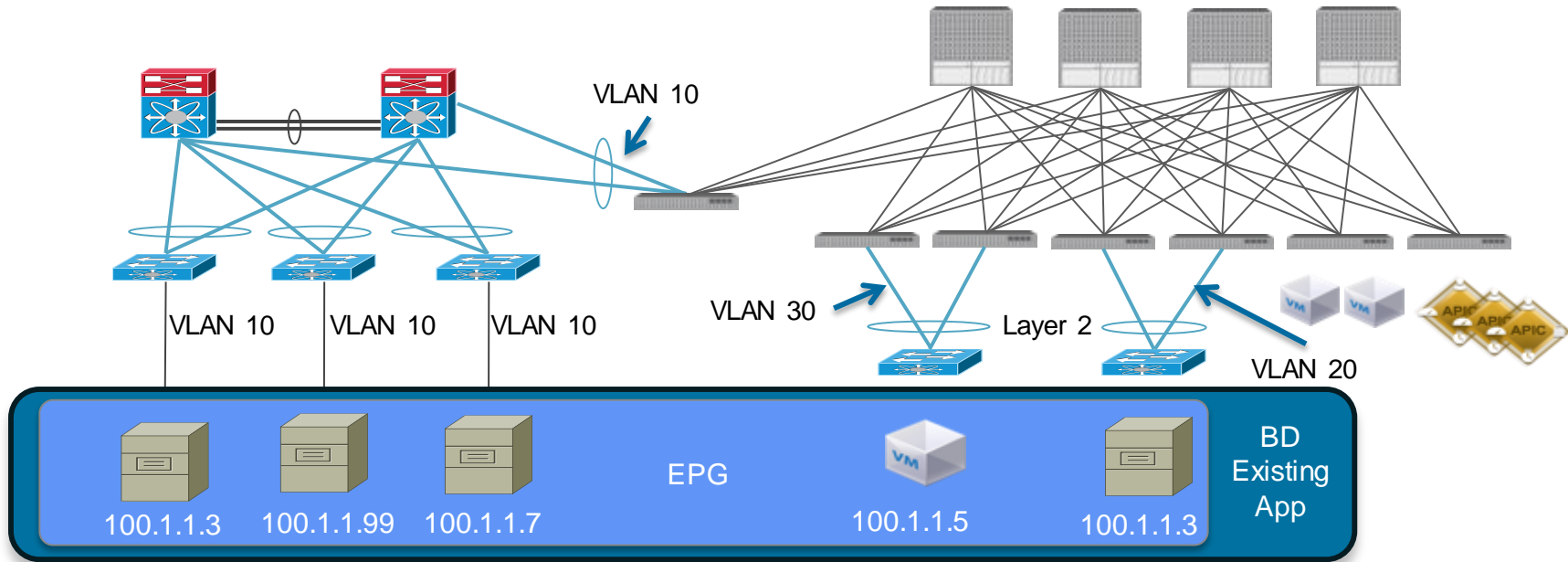
## Option 1



- VLAN's are localised to the leaf nodes
- The same subnet, bridge domain, EPG can be configured as a 'different' VLAN on each leaf switch
- In 1HCY15 VLAN's will be port local

# Extend the EPG

## Option 1



- Single Policy Group (one extended EPG)
- Leverage vPC for interconnect (diagram shows a single port-channel which is an option)
- BPDU should be enabled on the interconnect ports on the 'vPC' domain



# Assign Port to an EPG

- With VMM integration, port is assigned to EPG by APIC dynamically.
- In all other cases, such as connecting to switch, router, bare metal, port need to be assigned to EPG manually or use API
- Use “Static Binding” under EPG to assign port to EPG
- The example assigns traffic received on port eth1/32 with vlan tagging 100 to EPG VLAN 100

The screenshot displays the Cisco APIC GUI. On the left, a navigation tree shows the hierarchy: Quick Start, Tenant Tenant1, Application Profiles, Application1, Application EPGs, EPG DB, EPG WEB, Contracts, Static Bindings (Paths) (highlighted with a red box), Static Bindings (Leaves), Static EndPoint, and Subnode. The main panel shows the configuration for 'Node: Nodes-102-103'. It contains a table with columns: PATH, ENCAP, DEPLOYMENT IMMEDIACY, and MODE. The table has one row: 'Node-102-103/VPC1' with 'vlan-135' in the ENCAP column, 'Immediate' in the DEPLOYMENT IMMEDIACY column, and 'Tagged' in the MODE column. Below the table are 'UPDATE' and 'CANCEL' buttons. An 'ACTIONS' dropdown menu is visible in the top right corner of the main panel.

PATH	ENCAP	DEPLOYMENT IMMEDIACY	MODE
Node-102-103/VPC1	vlan-135	Immediate	Tagged

# Assign Port to EPG

## VLAN Tagging Mode

- **Tagged**. Trunk mode
- **Untagged**. Access mode. Port can only be in one EPG
- **802.1P Tag**. Native VLAN.
- No Tagged and Untagged(for different port) config for same EPG with current software
- Assign port eth1/1 with VLAN 100 tagged mode and port eth1/2 with VLAN 100 untagged mode to EPG WEB is not supported
- Use 802.1P Tag. Port eth1/1 vlan 100 tagged, eth1/2 vlan 100 802.1P Tag
- VLAN to EPG mapping is switch wide significant

Application Profiles

- Application1
  - Application EPGs
    - EPG DB
    - EPG WEB

Contracts

Static Bindings (Paths)

Static Bindings (Leaves)

Static EndPoint

Subnets

Domains (VMs and Bare-Metals)

Node: Nodes-102-103

PATH	ENCAP	DEPLOYMENT IMMEDIACY	MODE
Node-102-103/VPC1	vlan-135	Immediate	Tagged

UPDATE CANCEL

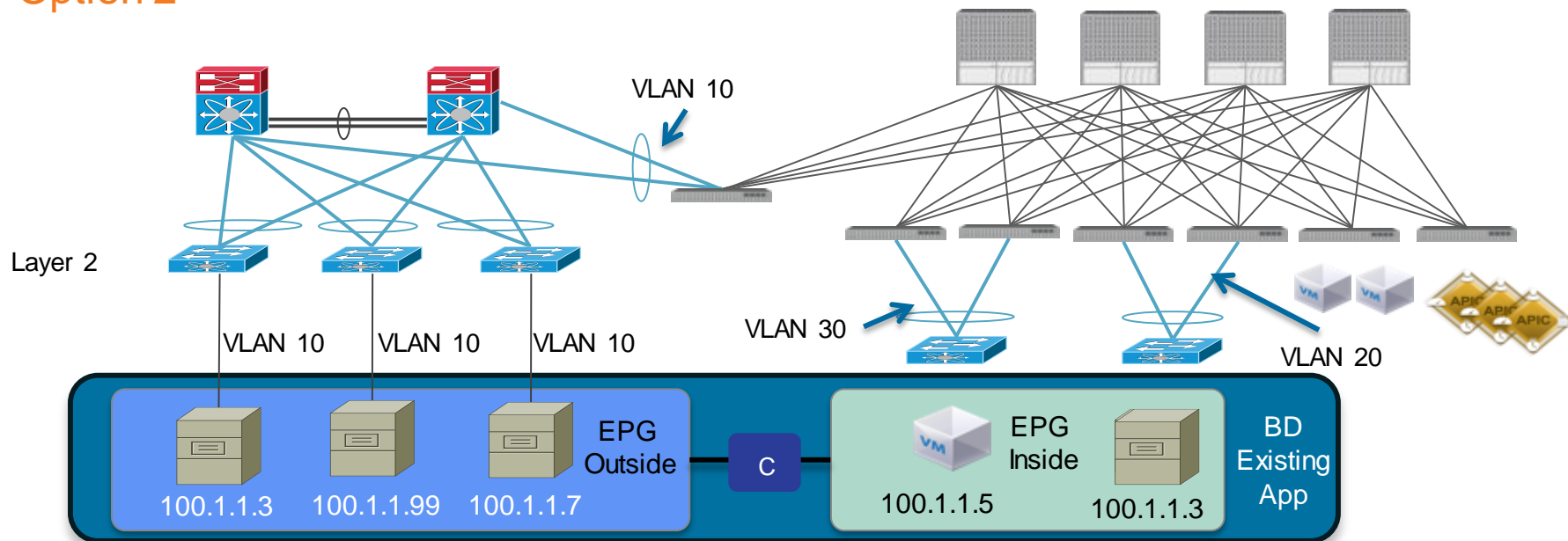
802.1P Tag

Tagged

Untagged

# Extend the Bridge Domain

## Option 2

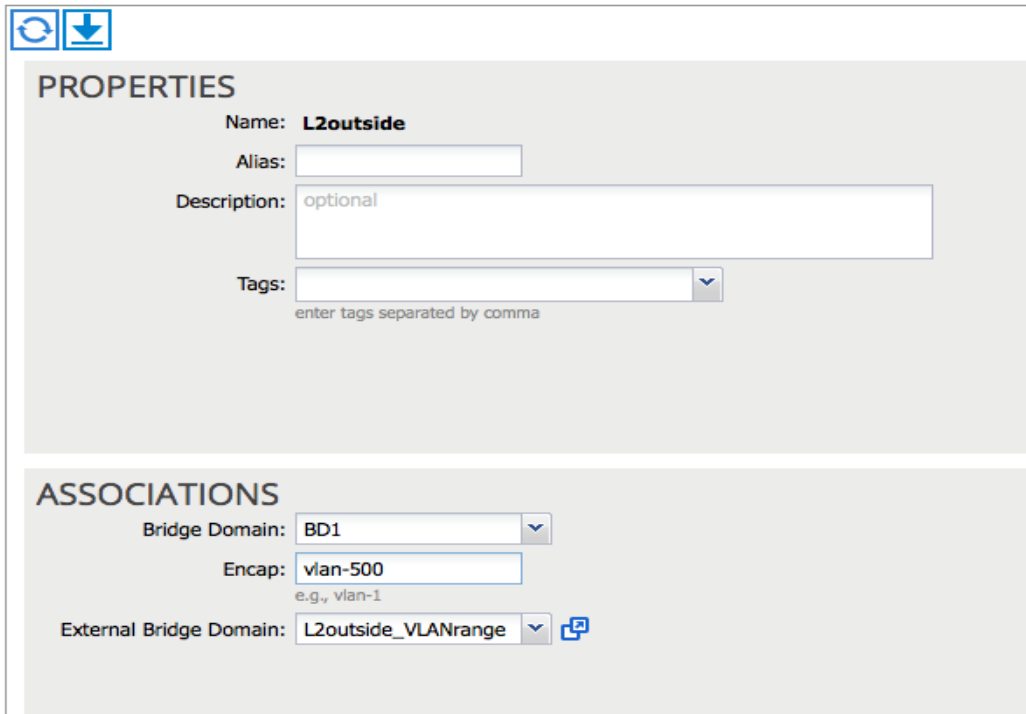


- External EPG (policy between the L2 outside EPG and internal EPG)
- Leverage vPC for interconnect (diagram shows a single port-channel which is an option)
- BPDU should be enabled on the interconnect ports on the 'vPC' domain
- L2 outside forces the same external VLAN << fewer operational errors

# L2 Outside Connection Configuration Example

## L2 Outside - L2outside

- Step 1. Create L2 Outside connection.
- Associate with BD.
- Specify VLAN ID to connect to outside L2 network
- External Bridge Domain is a way to specify the VLAN pool for outside connection.
- It is NOT a Bridge Domain.



The screenshot displays the configuration interface for an L2 Outside connection. At the top left, there are two icons: a circular arrow for refresh and a downward arrow for save. The interface is divided into two main sections: PROPERTIES and ASSOCIATIONS.

**PROPERTIES**

- Name:** L2outside
- Alias:** (empty text field)
- Description:** optional (text field)
- Tags:** (dropdown menu with a downward arrow and a small square icon to the right). Below the field is the text "enter tags separated by comma".

**ASSOCIATIONS**

- Bridge Domain:** BD1 (dropdown menu)
- Encap:** vlan-500 (text field). Below the field is the text "e.g., vlan-1".
- External Bridge Domain:** L2outside\_VLANrange (dropdown menu) with a small square icon to the right.

# L2 Outside Connection Configuration Example

- Step 2. Specify leaf node and interface providing L2 outside connection

## Logical Interface Profile - L2nodep



### PROPERTIES

Name: **L2nodep**

Description: optional

### INTERFACES



— PATH

TARGET DSCP

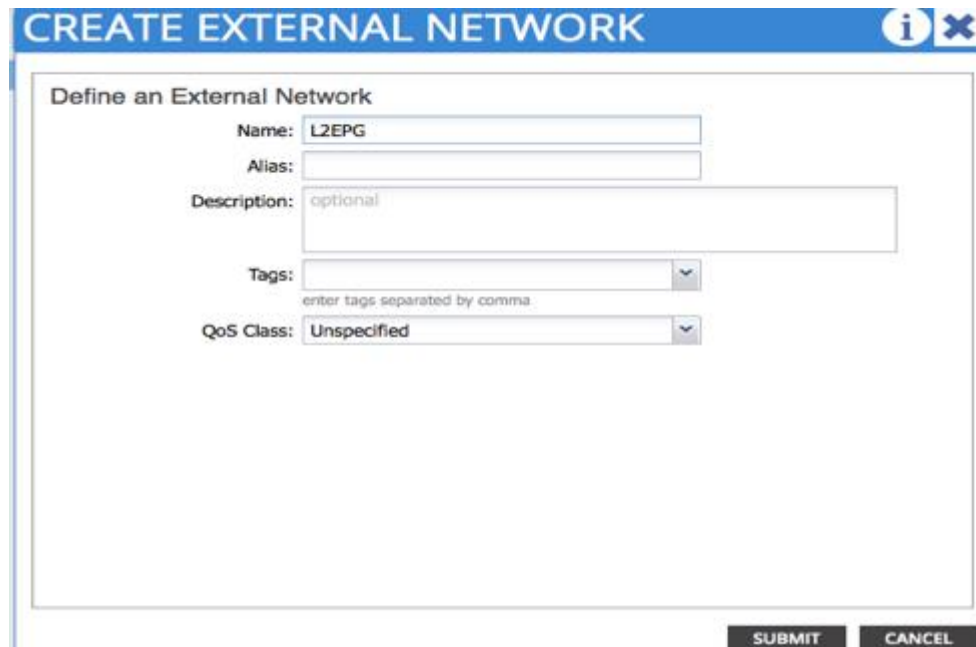
topology/pod-1/paths-101/pathep-[eth1/20]

Unspecified



# L2 Outside Connection Configuration Example

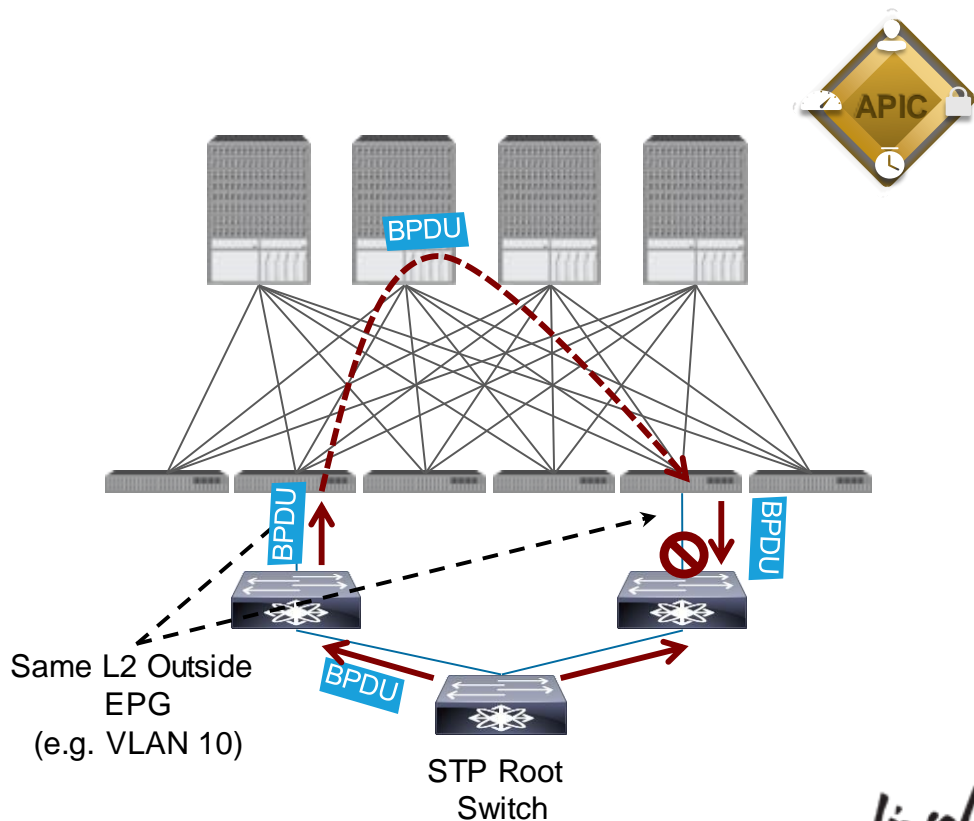
- Step 3. Create external EPG under L2 outside connection
- Step 4. Create contract between external EPG and internal EPG



The screenshot shows a web-based configuration interface titled "CREATE EXTERNAL NETWORK". Inside the main window, the heading "Define an External Network" is displayed. Below this heading, there are several input fields: "Name:" with the value "L2EPG", "Alias:" which is empty, and "Description:" with the value "optional". Below the description is a large text area. There is a "Tags:" field with a dropdown arrow and a hint "enter tags separated by comma". At the bottom, there is a "QoS Class:" field with a dropdown arrow and the value "Unspecified". At the bottom right of the window, there are two buttons: "SUBMIT" and "CANCEL".

# ACI Interaction with STP

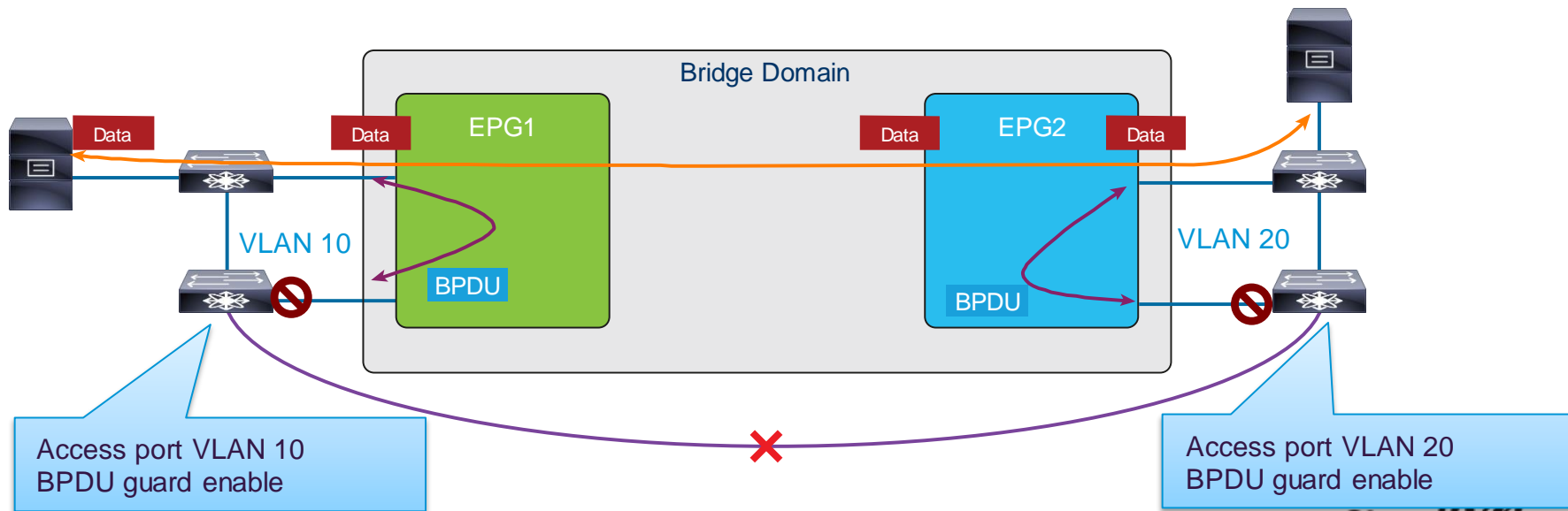
- No STP running within ACI fabric
- BPDU frames are flooded between ports configured to be members of the same external L2 Outside (EPG)
- No Explicit Configuration required
- Hardware forwarding, no interaction with CPU on leaf or spine switches for standard BPDU frames
- Protects CPU against any L2 flood that is occurring externally
- External switches break any potential loop upon receiving the flooded BPDU frame fabric
- BPDU filter and BPDU guard can be enabled with interface policy



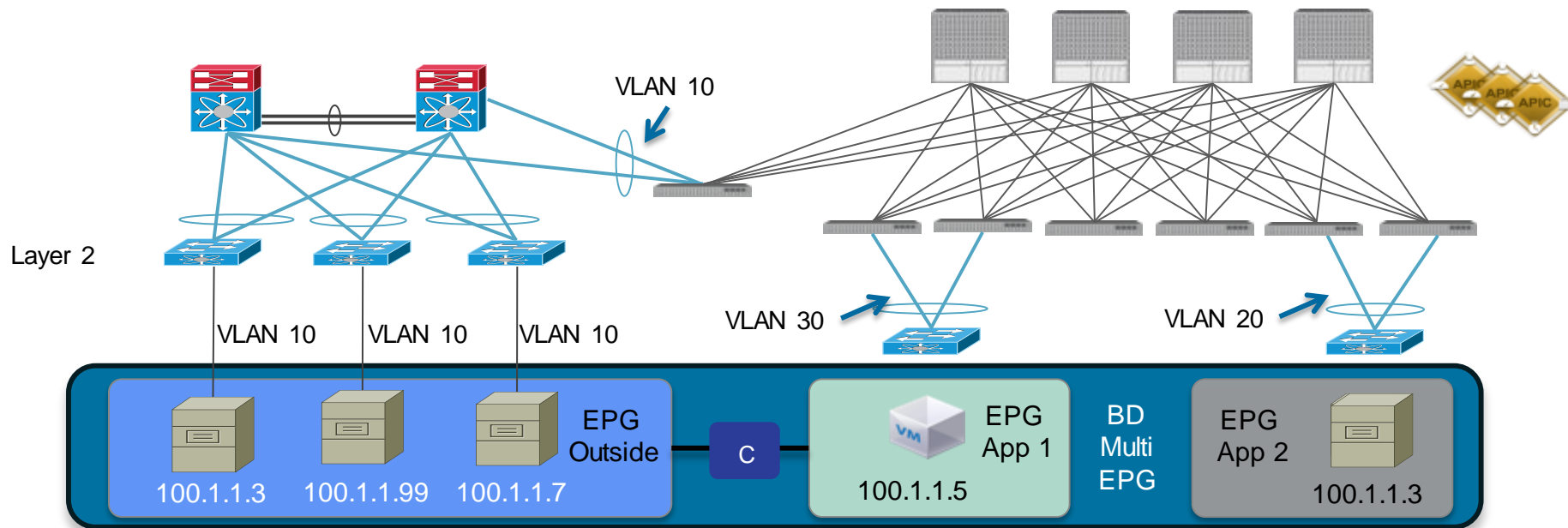
# ACI Interaction with STP

## VLAN Translation and STP

- ACI Fabric allows VLAN Translation (Switch and Port Local VLAN Significance)
- Make sure no external loop and the BPDU guard is enabled on access ports
- Data is flooded within BD and BPDU is flooded within EPG.



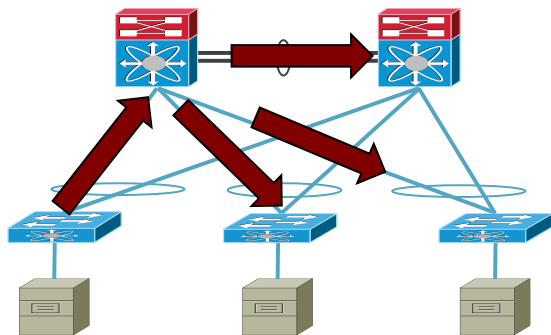
# Managing Flooding Within the BD



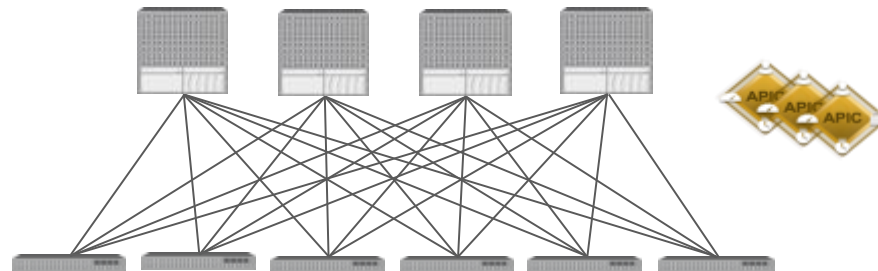
- In a classical network traffic is flooded with the Bridge Domain (within the VLAN)
- You have more control in an ACI Fabric but need to understand what behaviour you want



# Managing Flooding Within the Fabric



Flooding Domain is defined by the VLAN (Bridge Domain) Boundaries










Flooding Domain is defined by the configuration of the Bridge Domain and EPG Parameters (Enabled, Scoped, Disabled for different traffic types)

- In a classical network traffic is flooded with the Bridge Domain (within the VLAN)
- You have more control in an ACI Fabric but need to understand what behaviour you want

# Managing Flooding Within the Fabric

## ARP Unicast



 100

### PROPERTIES

Name: **default**

Description:

Unknown Unicast Traffic Class ID: **16387**

Segment: **15859678**

Multicast Address: **225.0.208.16**

Network:

Custom MAC Address:

L2 Unknown Unicast: ☐ Flood  
☒ Hardware Proxy

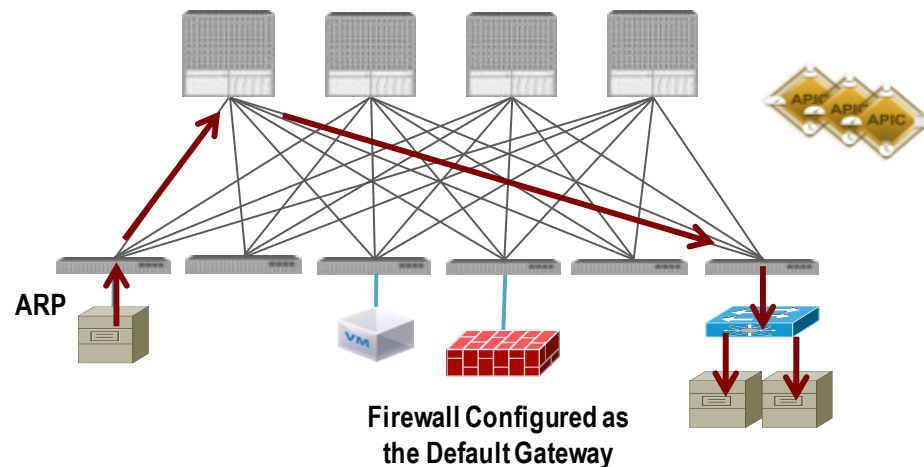
Unknown Multicast Flooding: ☒ Flood  
☐ Optimized Flood

ARP Flooding: ☐ **ARP Flooding Disabled (Default)**

Unicast Routing: ☒

IGMP Snoop Policy:

End Point Retention Policy:



- Disable ARP Flooding – ARP/GARP is forwarded as a unicast packet within the fabric based on the host forwarding DB
- On egress the ARP/GARP is forwarded as a flooded frame (supports hosts reachable via downstream L2 switches)

CiscoLive!

# Managing Flooding Within the Fabric

## ARP Flooding



### PROPERTIES

Name: **default**

Description: optional

Unknown Unicast Traffic Class ID: **16387**

Segment: **15859678**

Multicast Address: **225.0.208.16**

Network: select or type to pre-pi

Custom MAC Address: 00:22:BD:F8:19:FF

L2 Unknown Unicast:   
☐ Flood   
☒ Hardware Proxy

Unknown Multicast Flooding:   
☒ Flood   
☐ Optimized Flood

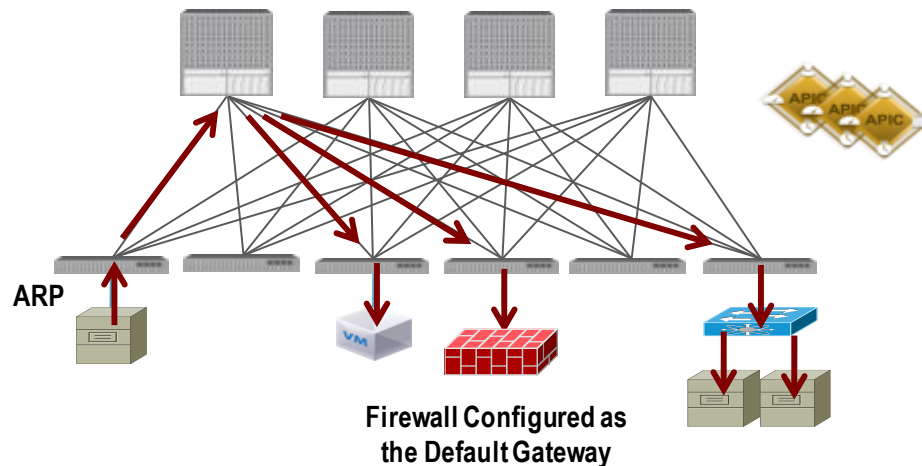
ARP Flooding: ☒

**ARP Flooding Enabled**

Unicast Routing: ☒

IGMP Snoop Policy: select or type to pre-pi

End Point Retention Policy: select or type to pre-pi



- Enabling ARP Flooding – ARP/GARP is flooded within the BD
- Commonly used when the default GW is external to the Fabric

CiscoLive!

# Managing Flooding Within the Fabric

## Unknown Unicast Proxy Lookup



**PROPERTIES**

Name: **default**

Description: optional

Unknown Unicast Traffic Class ID: **16387**

Segment: **15859678**

Multicast Address: **225.0.208.16**

Network: select or type to pre-**pi**

Custom MAC Address: 00:22:BD:F8:19:FF

L2 Unknown Unicast: ☐ Flood ☒ Hardware Proxy

Unknown Multicast Flooding: ☒ Flood ☐ Optimized Flood

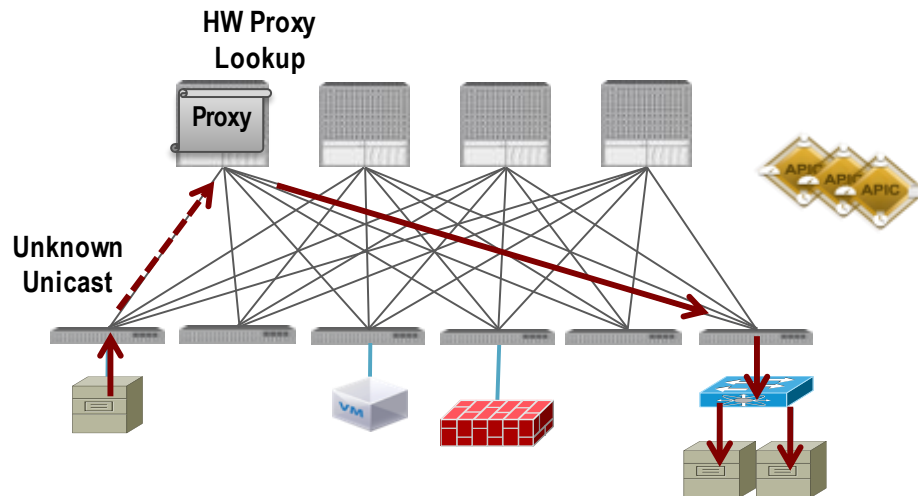
ARP Flooding: ☐

Unicast Routing: ☒

IGMP Snoop Policy: select or type to pre-**pi**

End Point Retention Policy: select or type to pre-**pi**

**Unknown Unicast Lookup via Proxy**



- Hosts (MAC, v4, v6) that are not known by a specific ingress leaf switch are forwarded to one of the proxies for lookup and inline rewrite of VTEP address
- If the host is not known by any leaf in the fabric it will be dropped at the proxy (allows honeypot for scanning attacks)

CiscoLive!

# Managing Flooding Within the Fabric

## Unknown Unicast Flooding



### PROPERTIES

Name: **default**

Description: optional

Unknown Unicast Traffic Class ID: **16387**

Segment: **15859678**

Multicast Address: **225.0.208.16**

Network: select or type to pre-pi

Custom MAC Address: 00:22:BD:F8:19:FF

L2 Unknown Unicast: ☒ Flood

☐ Hardware Proxy

Unknown Multicast Flooding: ☒ Flood

☐ Optimized Flood

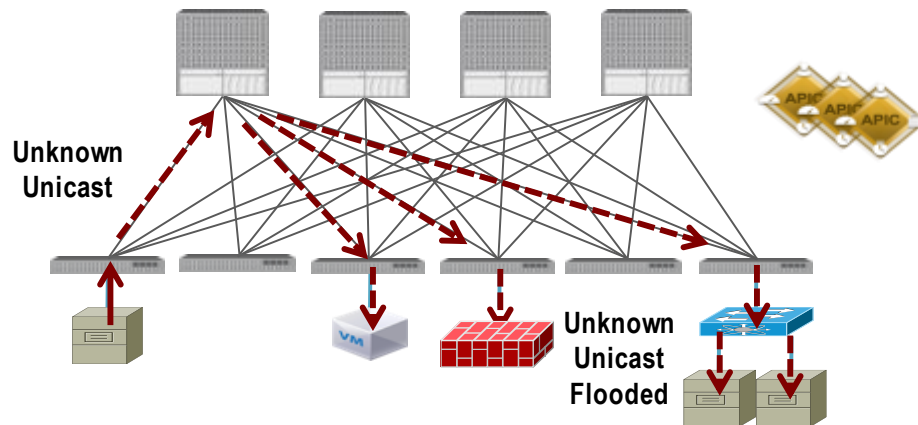
ARP Flooding: ☐

Unicast Routing: ☒

IGMP Snoop Policy: select or type to pre-pi

End Point Retention Policy: select or type to pre-pi

Unknown Unicast  
Flooded



- Hosts (MAC, v4, v6) that are not known by a specific ingress leaf switch are flooded to all ports within the bridge domain
- Silent hosts can be installed as static entries in the proxy (flooding not required for silent hosts)

CiscoLive!



# Managing Flooding Within the Fabric

## Unknown Multicast – Mode 1 (Flood)



### PROPERTIES

Name: **default**

Description: optional

Unknown Unicast Traffic Class ID: **16387**

Segment: **15859678**

Multicast Address: **225.0.208.16**

Network: select or type to pre-pi

Custom MAC Address: 00:22:BD:F8:19:FF

L2 Unknown Unicast: ☒ Flood

☐ Hardware Proxy

Unknown Multicast Flooding: ☒ Flood

☐ Optimized Flood

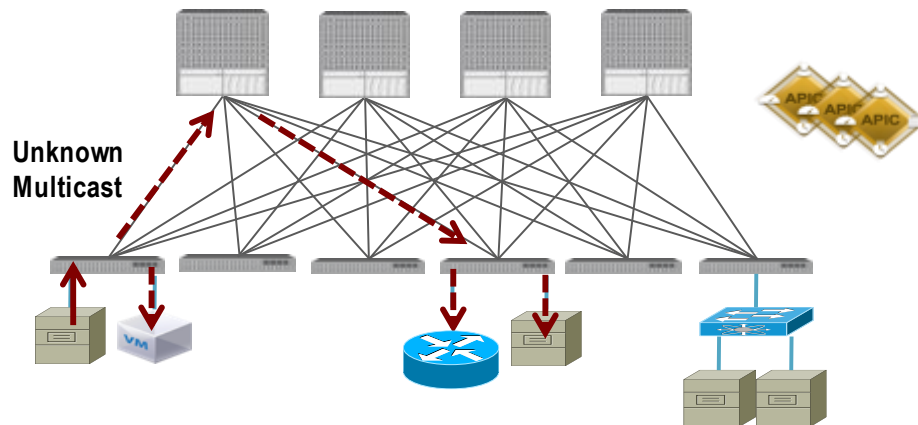
ARP Flooding: ☐

Unicast Routing: ☒

IGMP Snoop Policy: select or type to pre-pi

End Point Retention Policy: select or type to pre-pi

**Unknown Multicast  
Flooded**



- Unknown Multicast traffic is flooded locally to all ports in the BD on the same leaf the source server is attached to
- Unknown Multicast traffic is flooded to all ports in the BD on leaf nodes with a 'multicast router port'

CiscoLive!

# Managing Flooding Within the Fabric

## Unknown Multicast – Mode 2 (OMF 'or' Optimised Flood)



### PROPERTIES

Name: **default**

Description: optional

Unknown Unicast Traffic Class ID: **16387**

Segment: **15859678**

Multicast Address: **225.0.208.16**

Network: select or type to pre-pi

Custom MAC Address: 00:22:BD:F8:19:FF

L2 Unknown Unicast: ☐ Flood

☒ Hardware Proxy

Unknown Multicast Flooding: ☐ Flood

☒ Optimized Flood

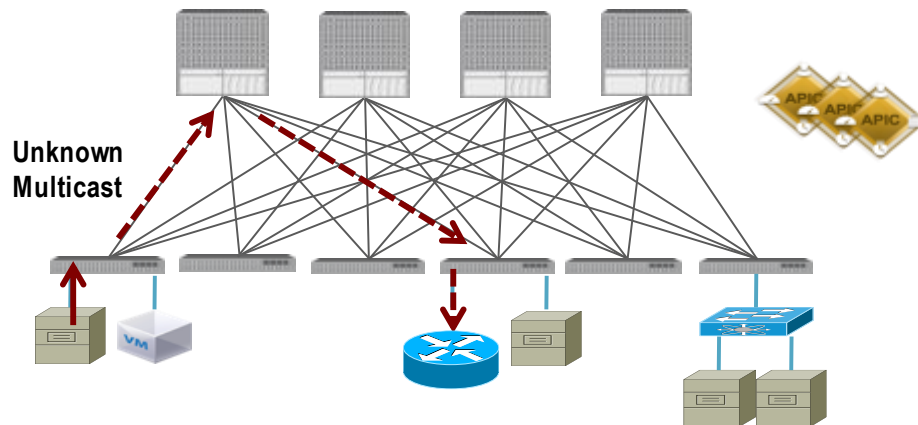
ARP Flooding: ☐

Unicast Routing: ☒

IGMP Snoop Policy: select or type to pre-pi

End Point Retention Policy: select or type to pre-pi

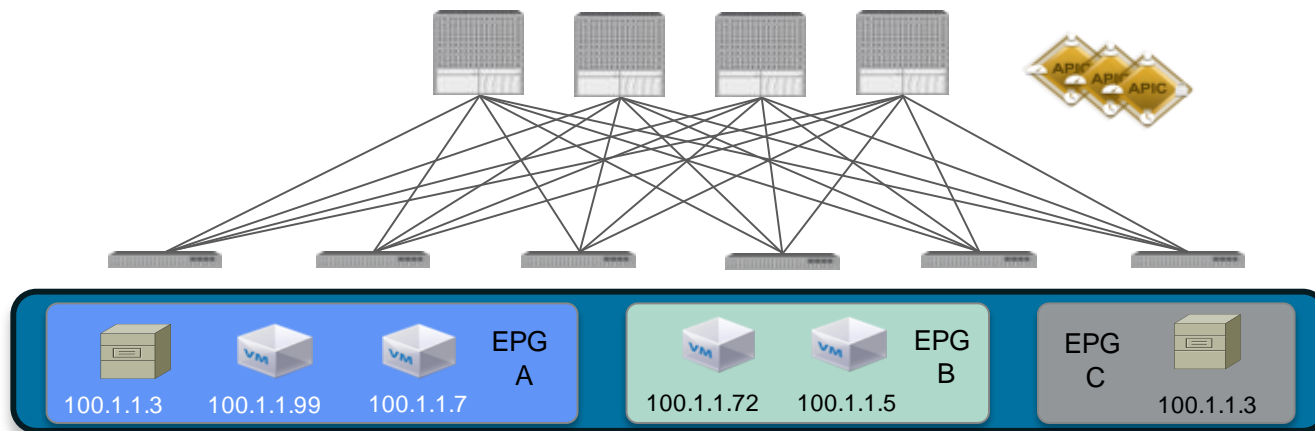
**Unknown Multicast  
Optimised Flooding**



- Unknown Multicast traffic is only flooded to 'multicast router ports' in this mode

# Managing Flooding Within the Fabric

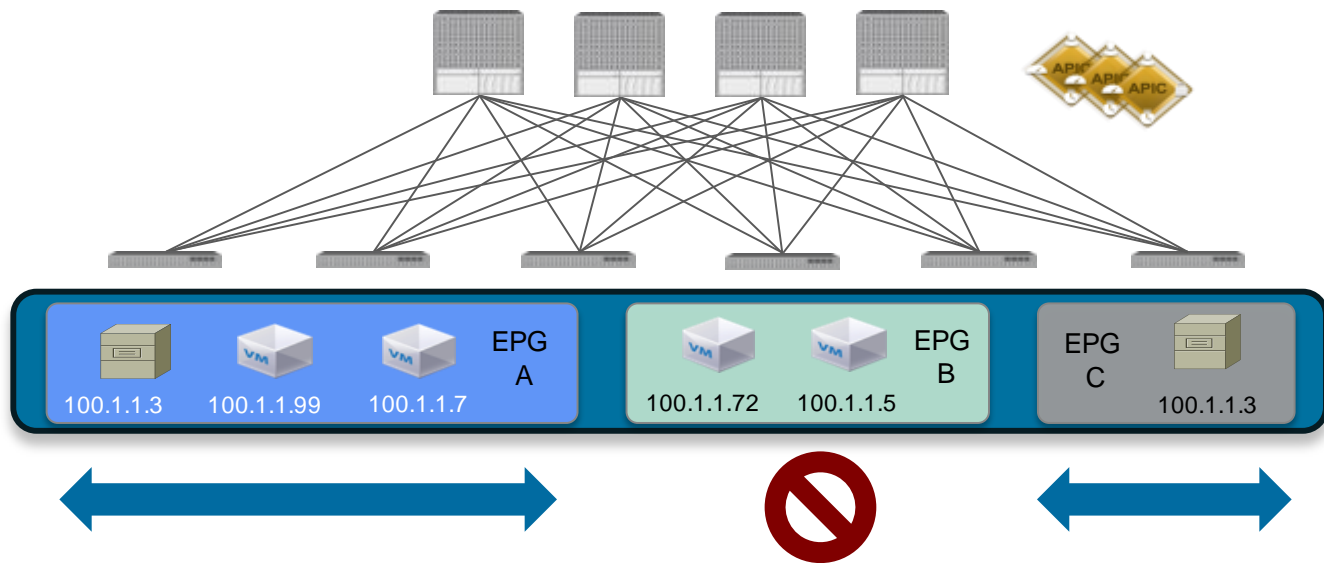
## Scoping Broadcasts to a micro segment



Traffic Type	11.0(x) Behaviour	11.1(x) Behaviour
ARP	Flood or Unicast	Flood or Unicast
Unknown Unicast	Flood or Leverage Proxy Lookup	Flood or Leverage Proxy Lookup
Unknown IP Multicast	Flood or OMF	Flood or OMF
L2 MCAST, BCAST, Link Local	Flood	Flood within the BD, Flood within the EPG, Disable Flooding within the BD/EPG

# Managing Flooding Within the Fabric

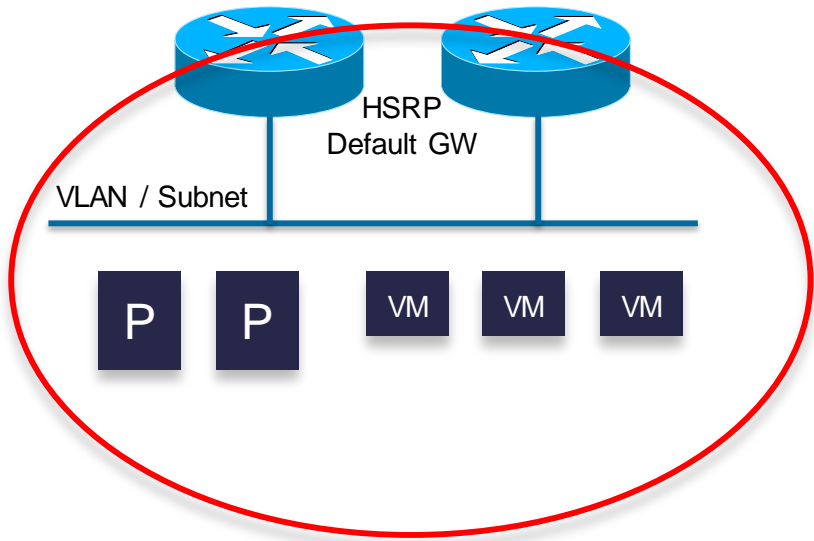
## Flooding scoped to the EPG



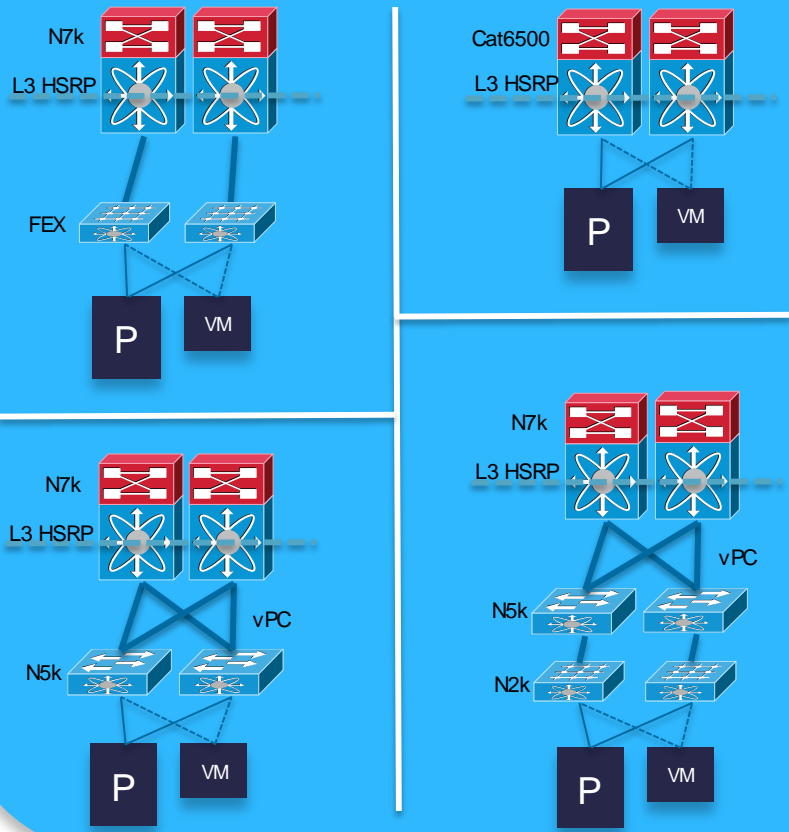
- Link Local, BCAST & L2 Multicast traffic can be managed on a micro-segment basis
- As an example:
  - EPG A & EPG C - Link Level traffic is flooded 'only' to the endpoints within the EPG
  - EPG B – Link Level traffic is dropped (not flooded)

# An Example of Interconnecting and Migrating

Logical Design



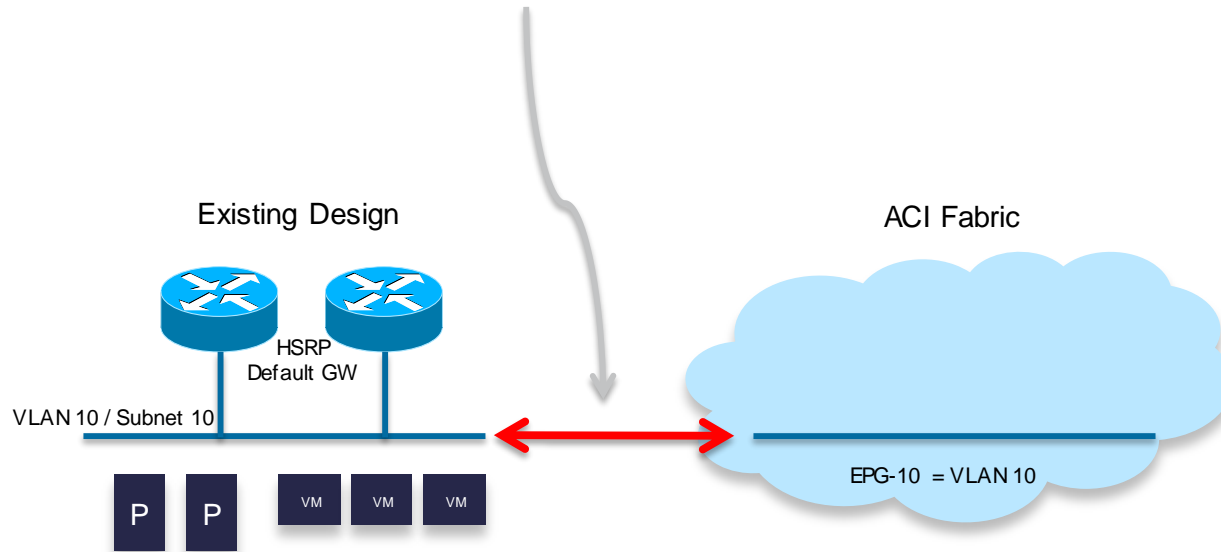
Many Different Physical Designs



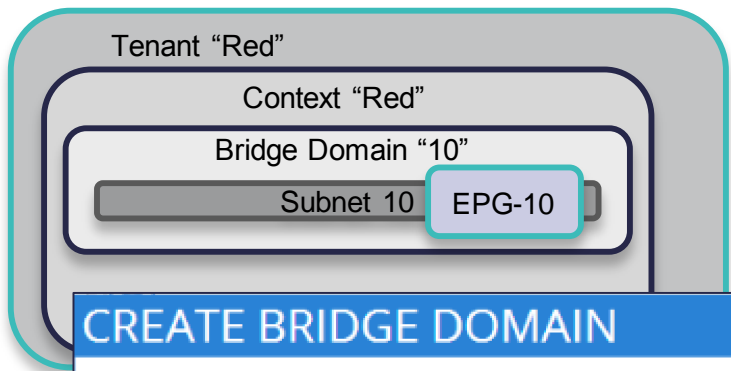


# Connect Fabric to Existing Network

- Functionally we are **expanding** the VLAN's into ACI.



# Configure ACI Bridge Domain Settings



Tenant "Red"

Context "Red"

Bridge Domain "10"

Subnet 10 EPG-10

## CREATE BRIDGE DOMAIN

Specify Bridge Domain for the Network

Name:

Description:

Network:

Forwarding:

L2 Unknown Unicast: ☒ Flood ☐ Hardware Proxy

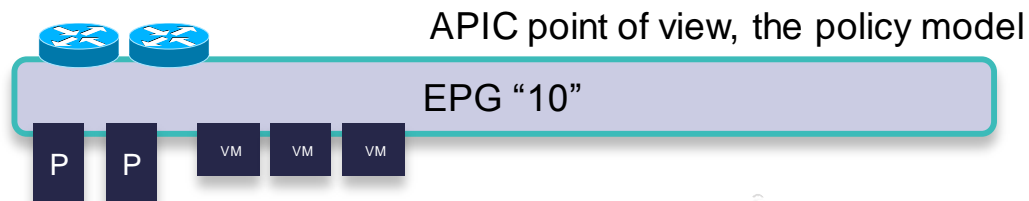
ARP Flooding: ☒ Enabled

Unicast Routing: ☐ Enabled

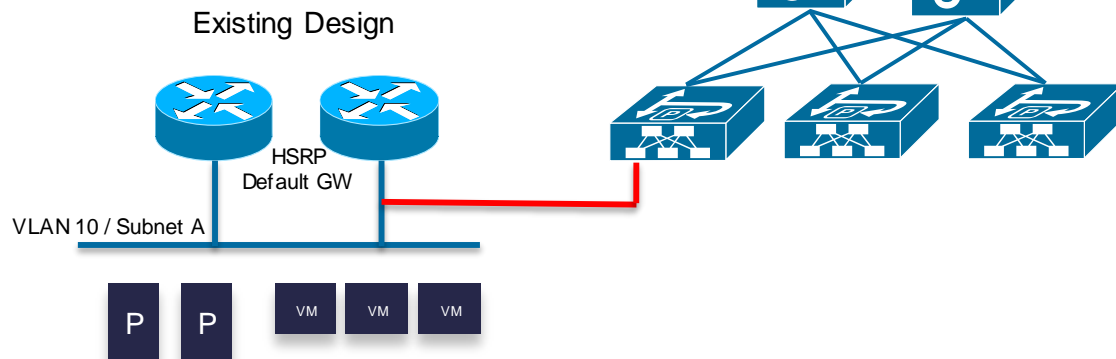
Config BD MAC Address: ☐

- Temporary Bridge Domain specific settings while we are using the HSRP gateways in the existing network.
- Select Forwarding to be "Custom" which allow
  - Enable Flooding of L2 unknown unicast
  - Enable ARP flooding
  - Disable Unicast routing

# Migrate Workloads



VM's will need to be connected to new Port Group under APIC control (AVS or DVS).



# Complete the Migration

Change BD settings back to normal for ACI mode

- Change BD settings back to default.
  - No Flooding
  - Unicast Routing enabled.

**CREATE BRIDGE DOMAIN**

Specify Bridge Domain for the Network

Name:

Description:

Network:

Forwarding:

L2 Unknown Unicast: ☐ Flood ☒ Hardware Proxy

ARP Flooding: ☐ Enabled

Unicast Routing: ☒ Enabled

Config BD MAC Address: ☐

# Migrating Default Gateway to the ACI Fabric



**PROPERTIES**

Name: **default**

Description: optional

Unknown Unicast Traffic Class ID: **16387**

Segment: **15859678**

Multicast Address: **225.0.208.16**

Network: select or type to pre-pi

Custom MAC Address: 00:22:BD:F8:19:FF

L2 Unknown Unicast: ☐ Flood ☒ Hardware Proxy

Unknown Multicast Flooding: ☒ Flood ☐ Optimized Flood

ARP Flooding: ☒

Unicast Routing: ☒

IGMP Snoop Policy: select or type to pre-pi

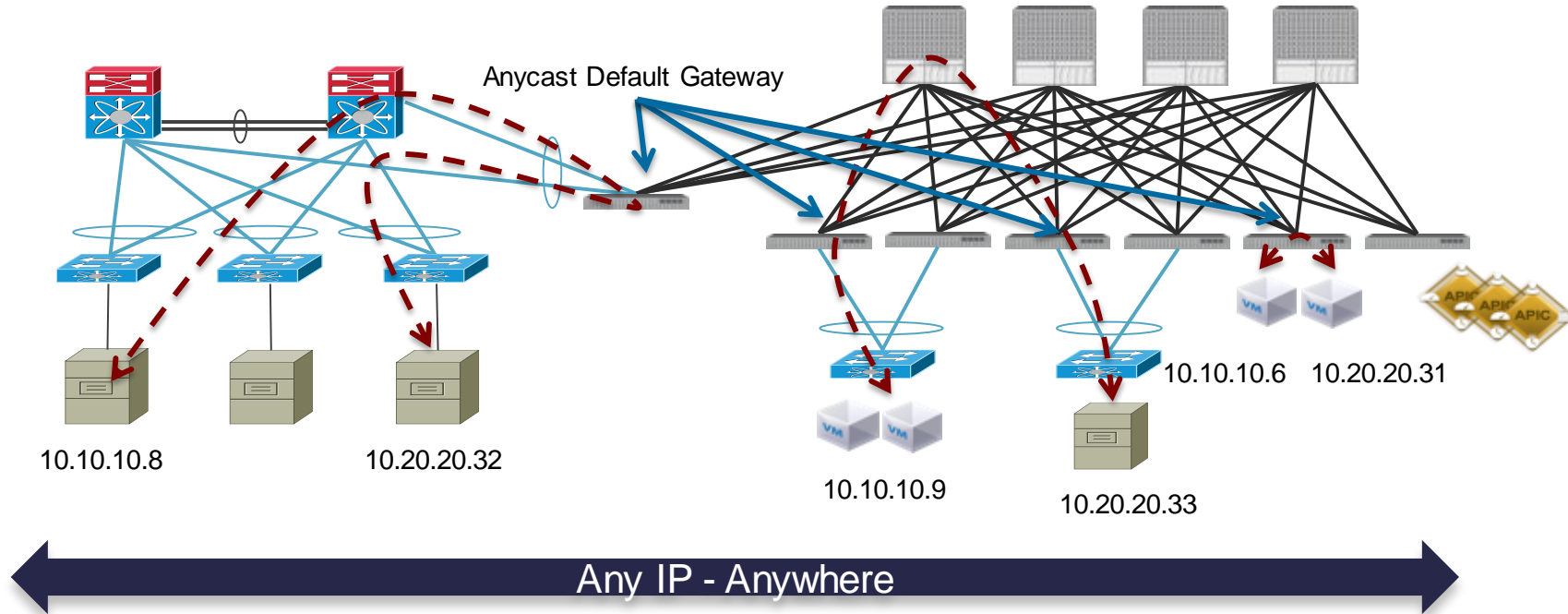
End Point Retention Policy: select or type to pre-pi

Change GW MAC address. By default, All fabric and all BD share same GW MAC

Enable Routing and ARP flooding

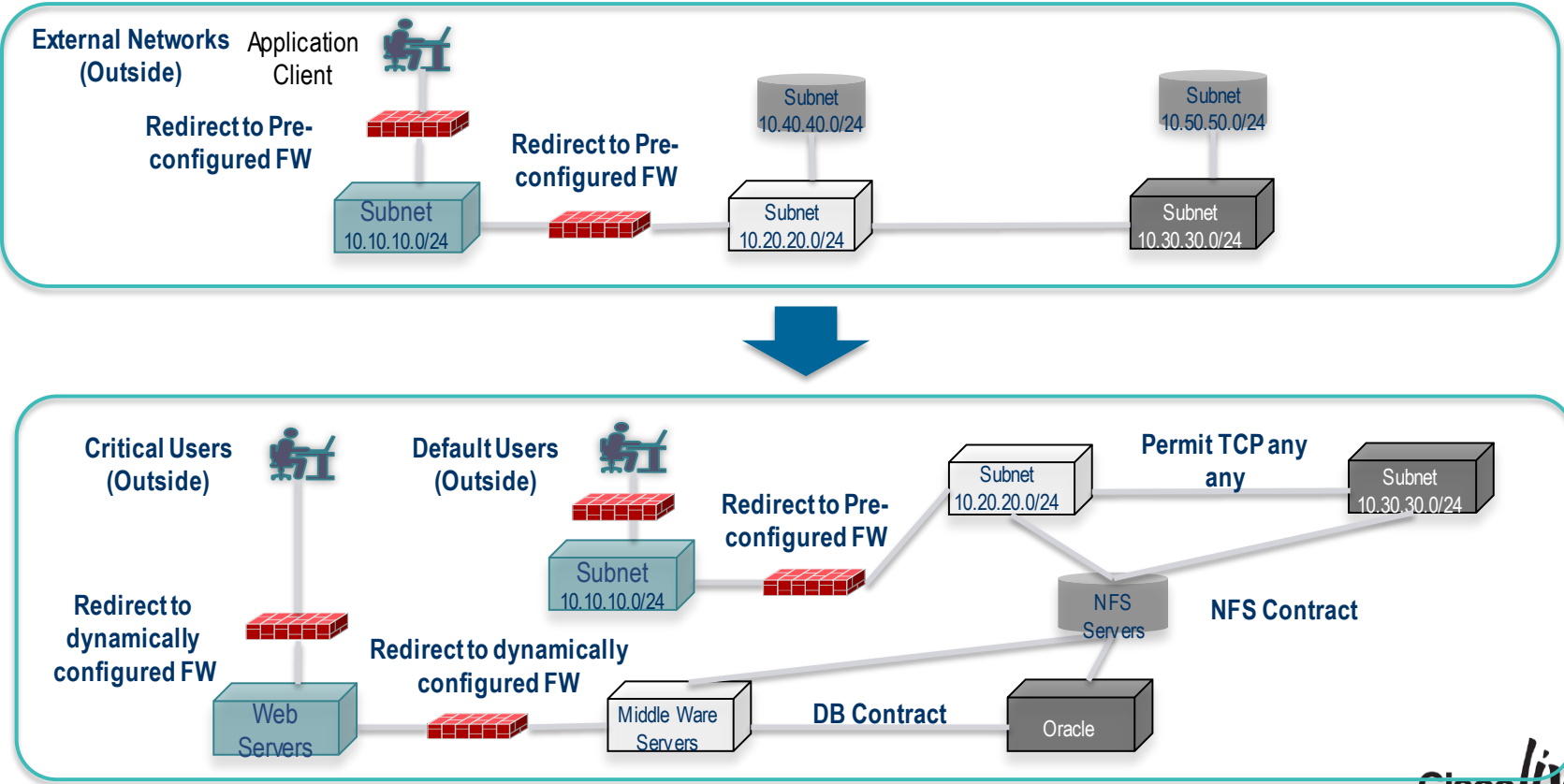
# Extension and Connecting

## It's a Network with Any VLAN Anywhere



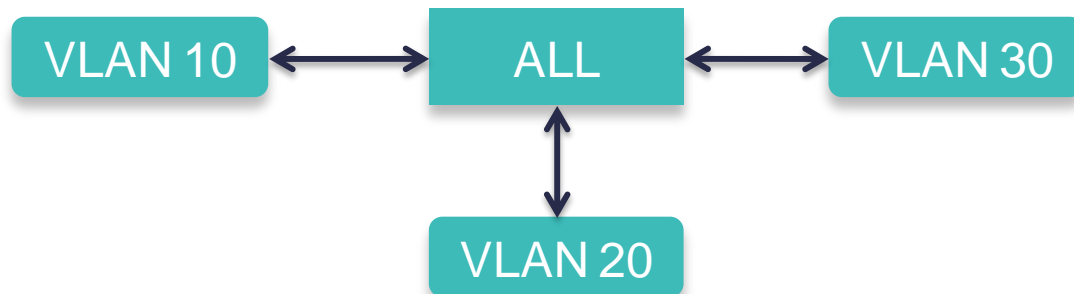


# Policy Can Be Added Gradually Starting With What You Have Today



# Simple Policy During Migration - Any-to-Any Configuration

	Contracts Provided	Filter	Contracts Provided	Contracts consumed	Filter
EPG "VLAN 10"	VLAN10	Default	ALL	ALL	Default
EPG "VLAN 20"	VLAN20	Default	ALL	ALL	
EPG "VLAN 30"	VLAN30	Default	ALL	ALL	



# I Want to Have a Very Open Configuration with VLAN10 Talking to Anything (1)

Create “Contract”  
ALL if it doesn’t exist  
yet

Use filter  
“common/default”

The screenshot displays the Cisco ISE configuration interface. On the left, a tree view shows the hierarchy: test > web-and-ordering > Application EPGs > L4-L7 Service Parameters > Networking > Security Policies > Contracts. Under 'Contracts', the 'any' contract is selected and highlighted. A blue arrow points from the text 'Create “Contract” ALL if it doesn’t exist yet' to the 'any' contract. Another blue arrow points from the text 'Use filter “common/default”' to the 'Filters' field in the 'PROPERTY' panel. The 'PROPERTY' panel on the right shows the configuration for the 'any' contract: Name: any, Description: optional, Reverse Filter Ports: checked, Apply Both Directions: false, and Filters: default. The 'Filters' field is expanded, showing a list with a plus icon, a minus icon, and the text 'NAME' and 'default'.

test

web-and-ordering

Application EPGs

L4-L7 Service Parameters

Networking

Security Policies

Contracts

any

VLAN10toOutside

VLAN10toVLAN20

VLAN30

Taboo Contracts

PROPERTY

Name: **any**

Description: optional

Reverse Filter Ports: ☒

Apply Both Directions: **false**

Filters: **default**

# I Want to Have a Very Open Configuration with VLAN10 Talking to Anything (2)

EPG VLAN 10  
provides and  
consumes “ALL”

Application Profiles

test

web-and-ordering

Application EPGs

EPG VLAN10

Contracts

Static Bindings (Paths)

Static Bindings (Leaves)



TENANT NAME

CONTRACT

Contract Type: Contract

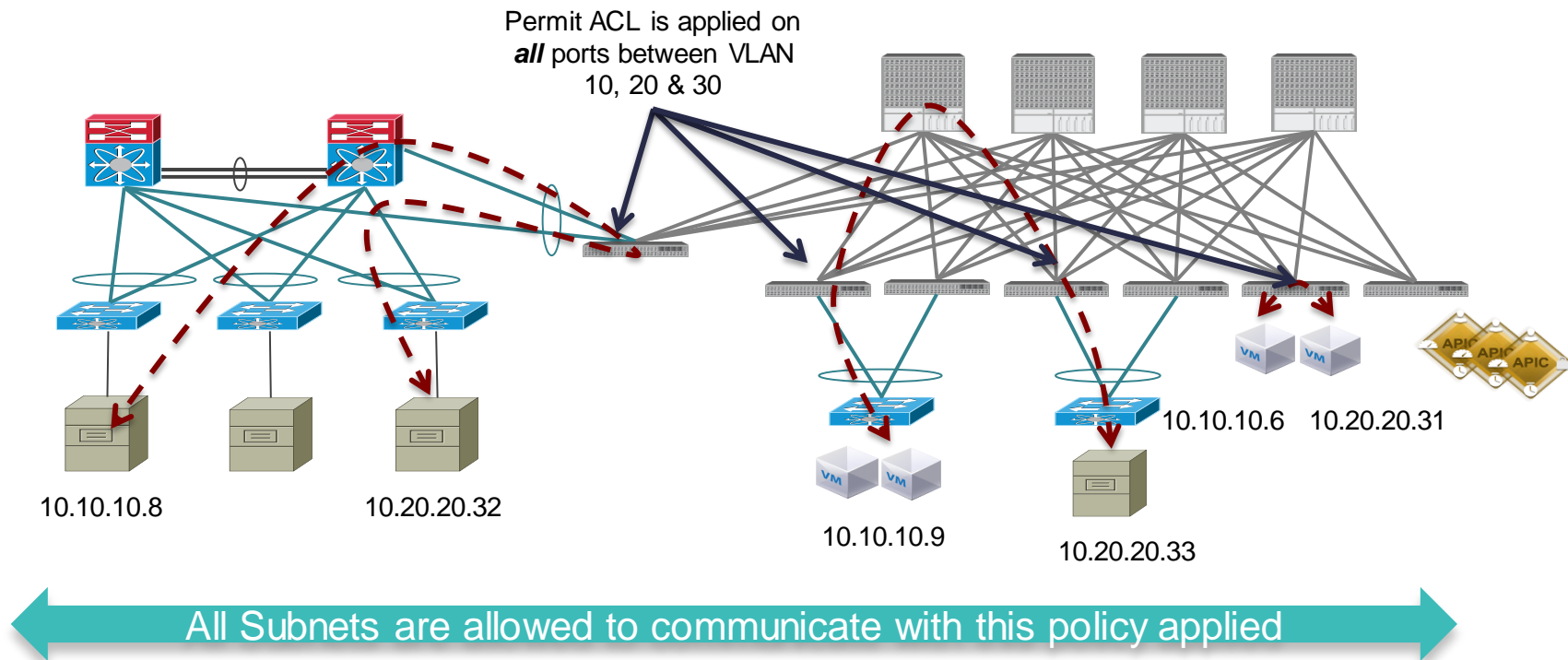
Customer1

ALL

Customer1

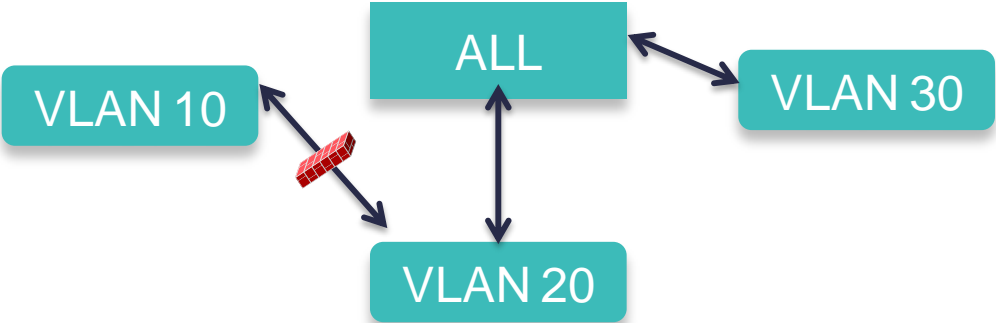
ALL

# Extension and Connecting Dynamic Distributed ACL's



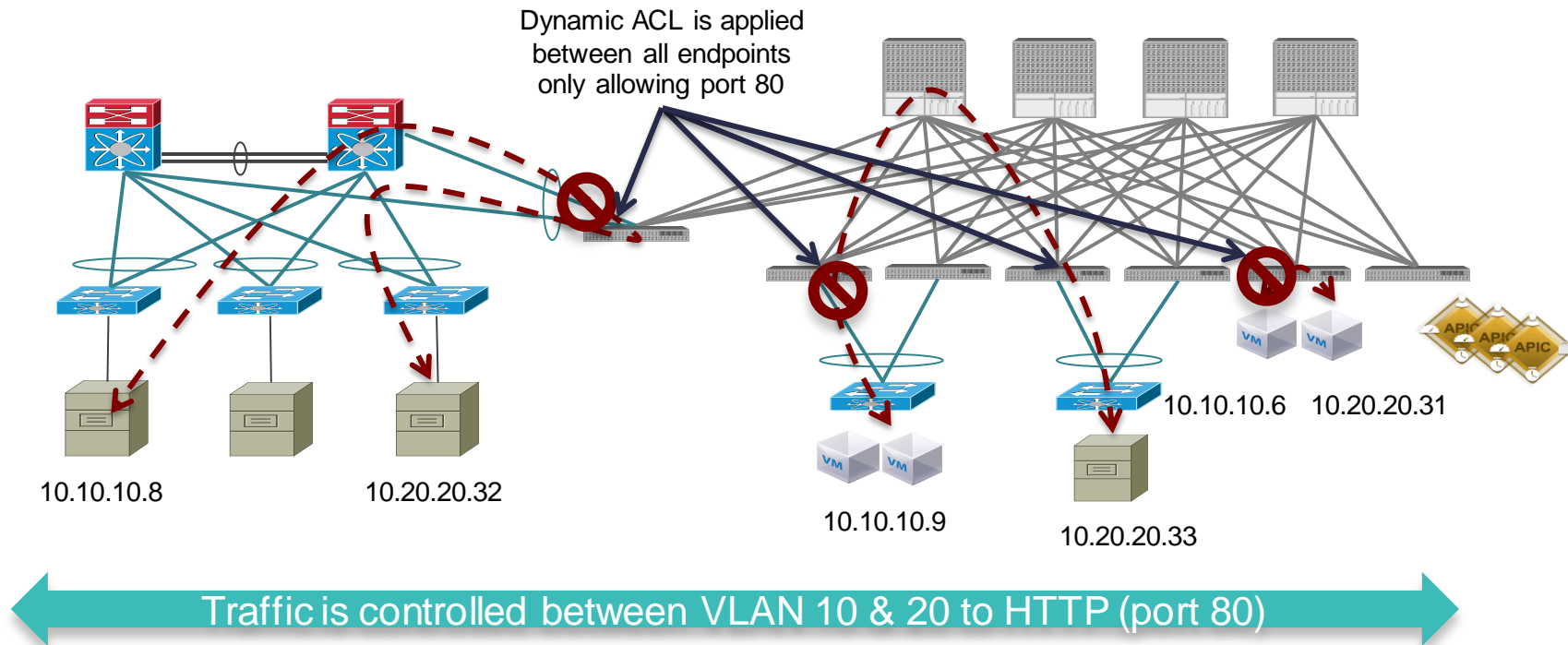
# Later If I want To Put an ACL Between VLAN 10 and 20

	Contracts Provided	Filter	Contracts Provided	Contracts consumed	Filter
EPG "VLAN 10"	VLAN10	Default		VLAN20	Port 80
EPG "VLAN 20"	VLAN20	Default	ALL	ALL	Default
EPG "VLAN 30"	VLAN30	Default	ALL	ALL	





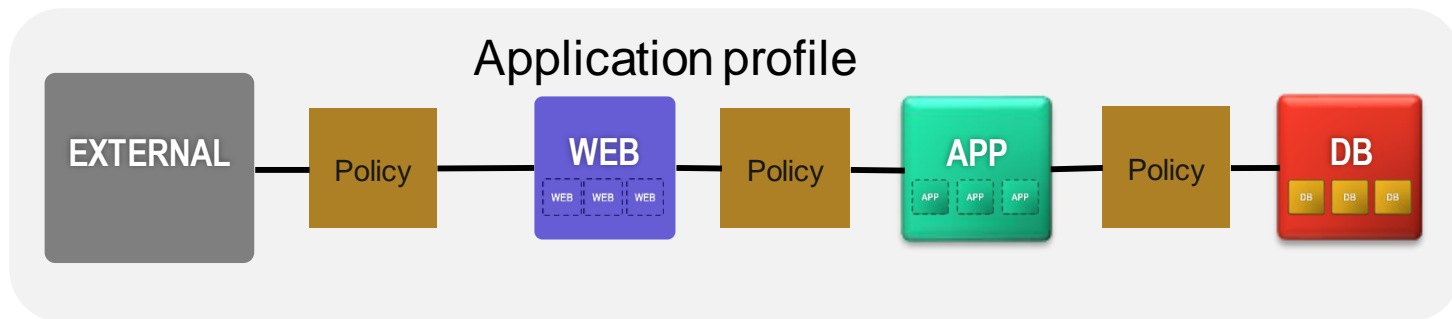
# Extension and Connecting Dynamic ACL's



A long-exposure photograph of a city street at night. The foreground is filled with vibrant, multi-colored light trails from moving vehicles, creating a sense of motion. In the background, a modern pedestrian bridge with blue lighting spans the street. Tall buildings with illuminated windows and storefronts line the street, and several flags are visible on the left side.

# Integrating and Extending L4-7 Services

# Automate Service Insertion Through APIC



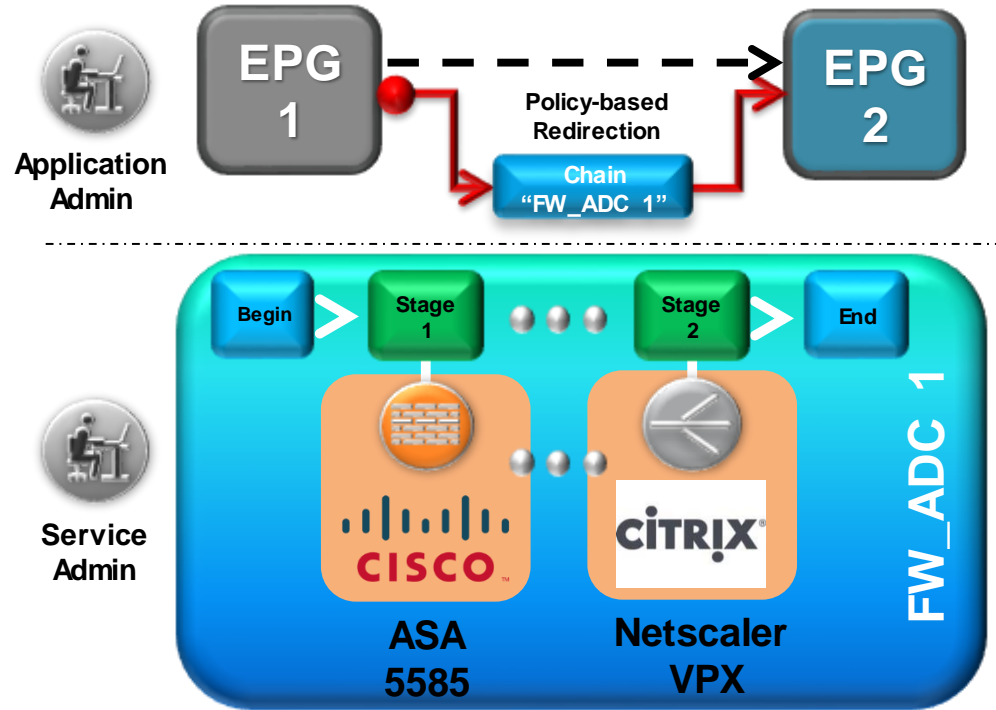
## APIC Policy Model

**Endpoint Group (EPG):** Collection of similar End Points identifying a particular Application Tier. Endpoint could represent VMs, VNICs , IP, DNS name etc

**Application Profile:** Collection of Endpoint Groups and the policies that define way Endpoint group communicate with each other

# ACI Service Insertion via Policy

- Automated and scalable L4-L7 service insertion
- Packet match on a redirection rule sends the packet into a services graph.
- Service Graph can be one or more service nodes pre-defined in a series.
- Service graph simplifies and scales service operations





# Service Graphs - Extensibility of the Data Path

## Insertion of NFV elements in the Data Path

L4 Filters can be applied to redirect subset of traffic via different paths

Split/Join chain based on pkt/flow/transaction Context (eg HTTP hdrs)

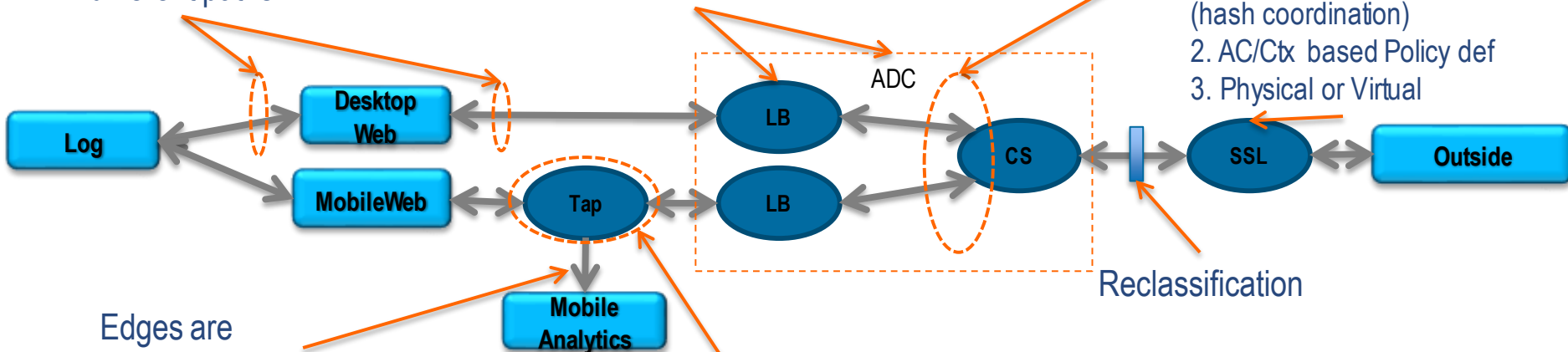
Logical Functions  
(location independence)

1. Scale in/out (hash coordination)
2. AC/Ctx based Policy def
3. Physical or Virtual

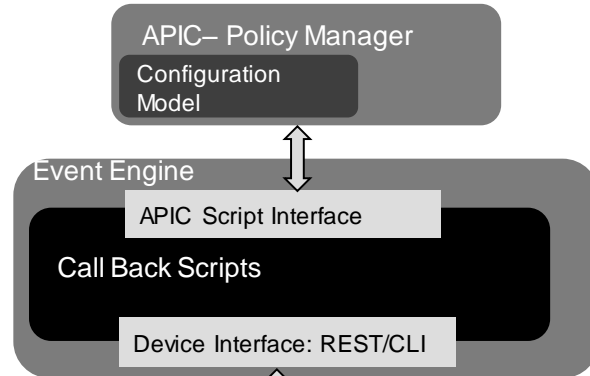
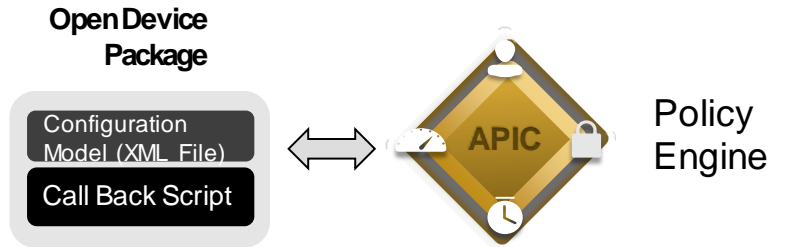
Reclassification

Attach a Mirror  
(HW mirroring)

Edges are  
direction aware



# Service Automation Through Device Package



APIC provides extendable policy model through Device Package

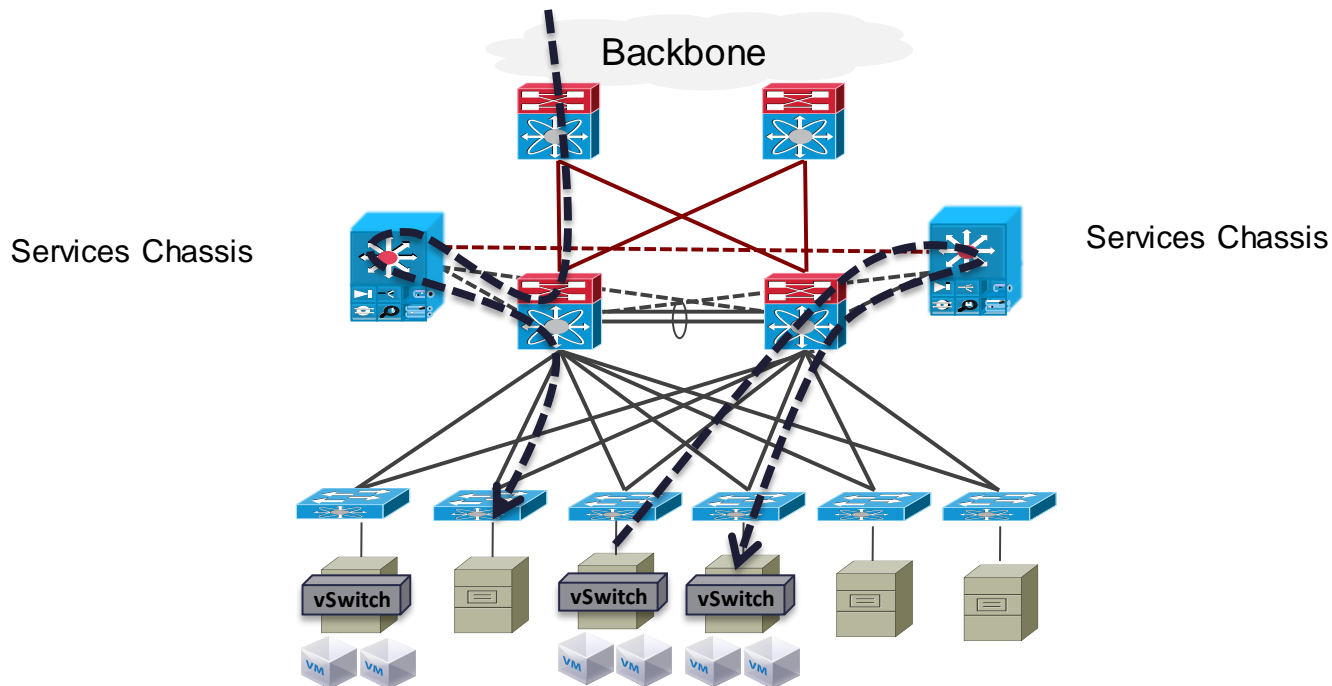
Device Package contains XML file defining Device Configuration

Provider Administrator can upload a Device Package

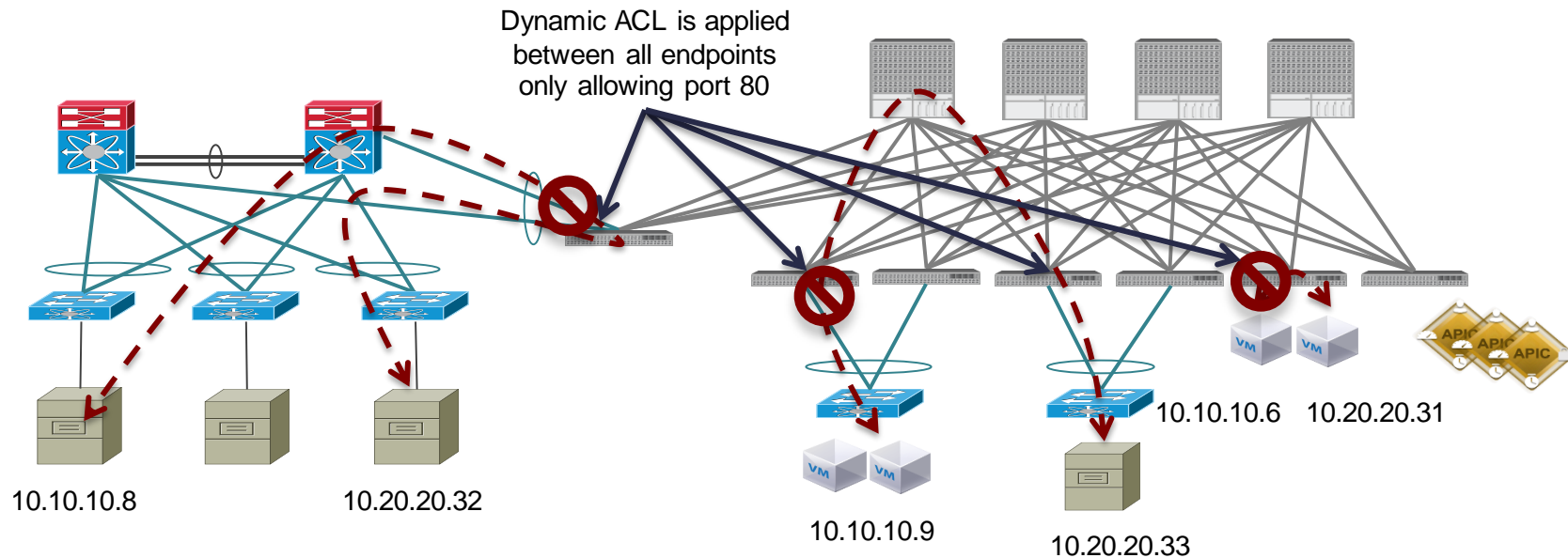
Device scripts translates APIC API callouts to device specific callouts



# Extending ACI in to Current Data Centre's Standard Architecture with Services

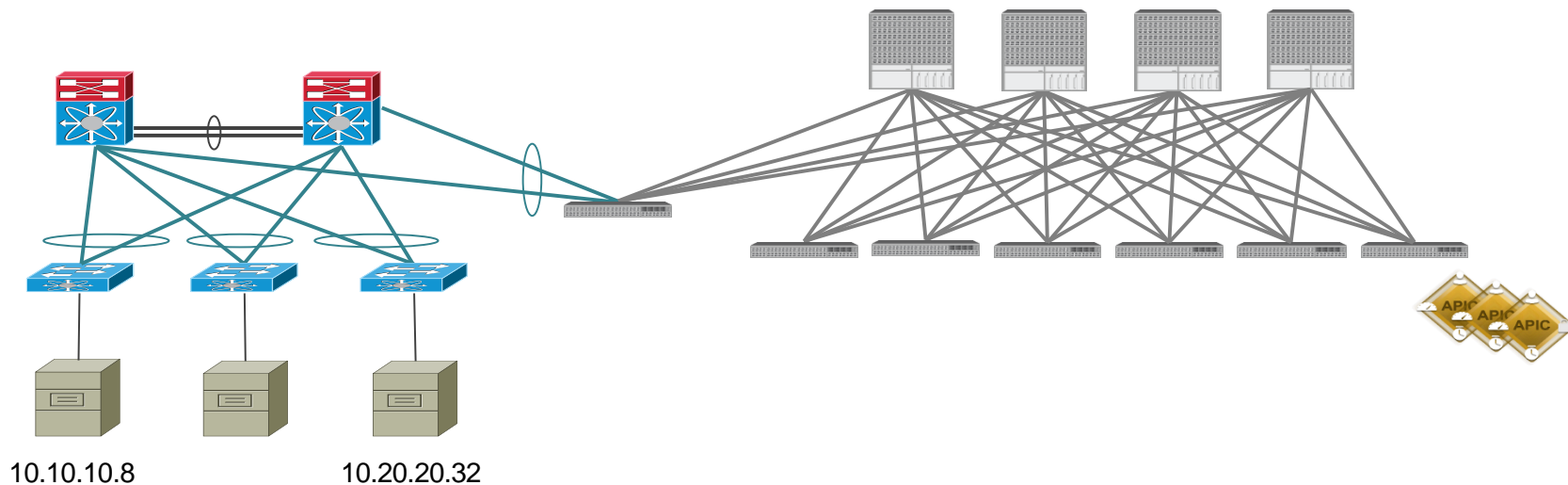


# Extension and Connecting Services Switch



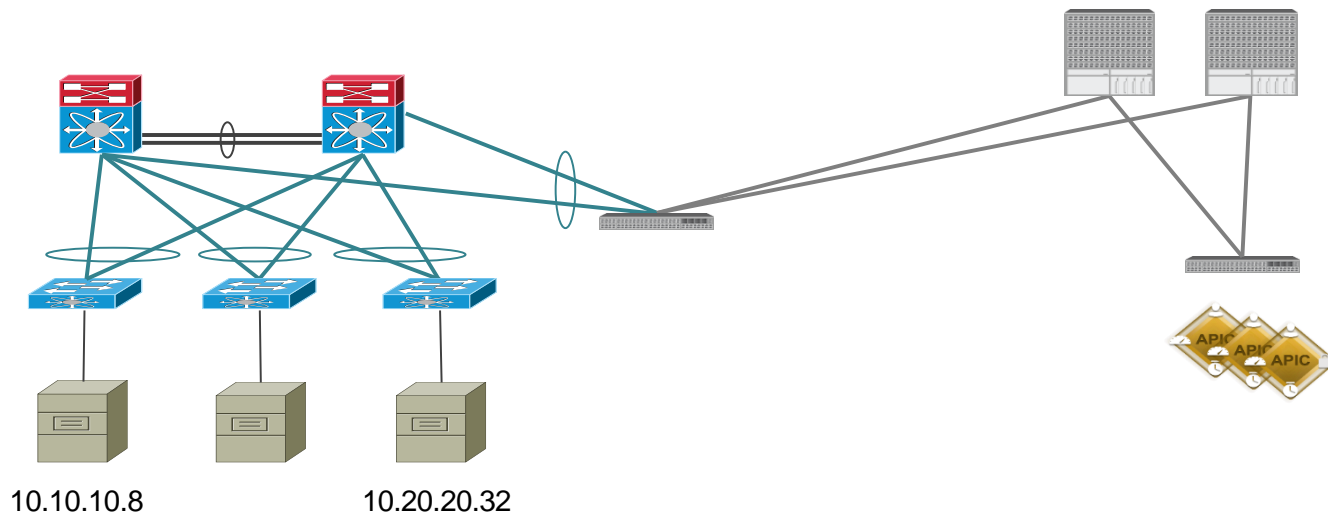
- Start with the picture we now understand

# Extension and Connecting Services Switch



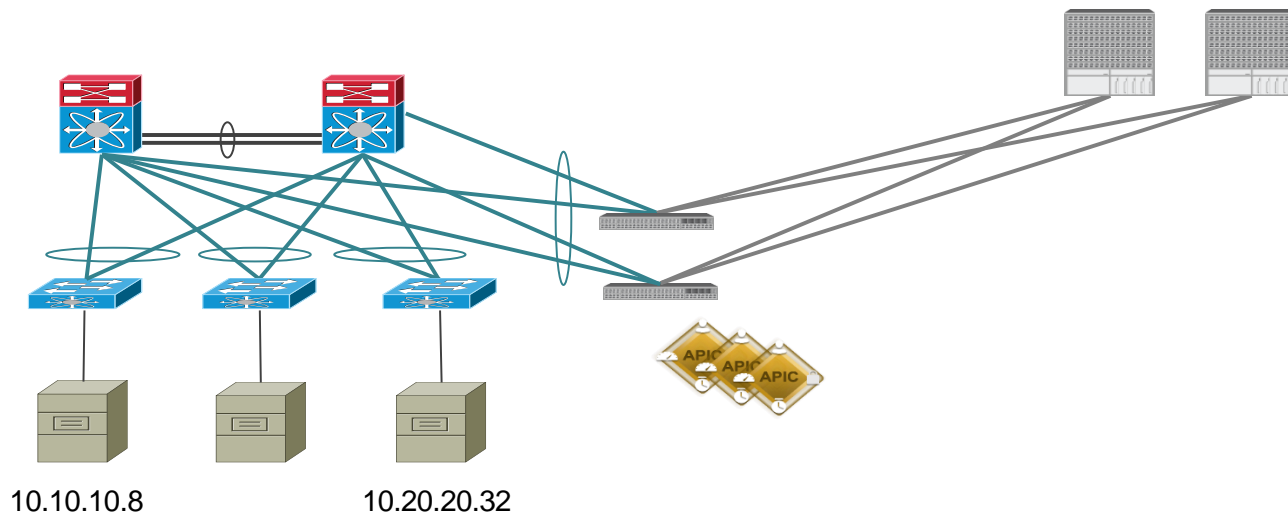
- Lets simplify things

# Extension and Connecting Services Switch



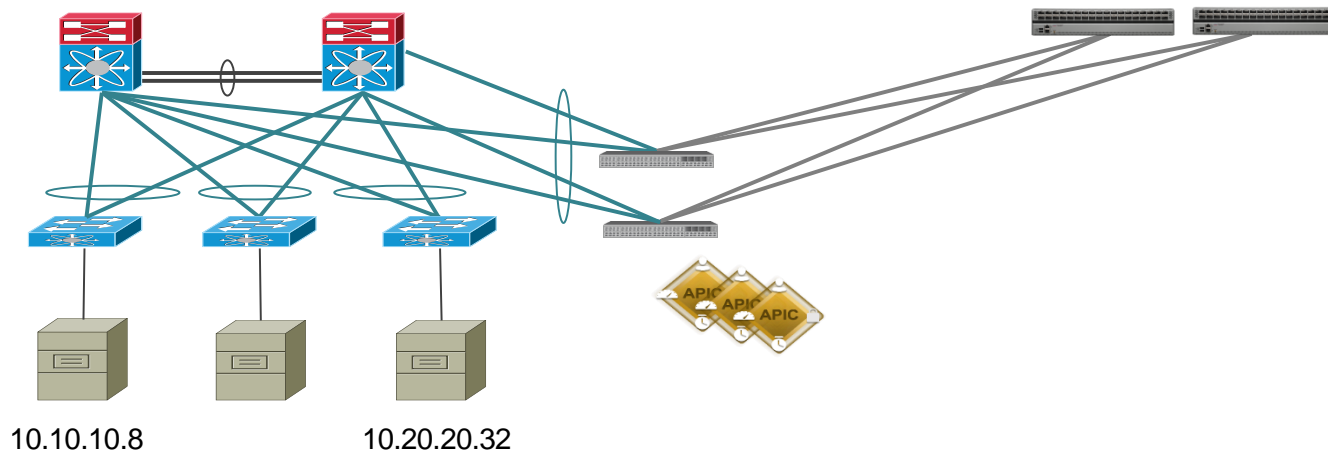
- Start with just a few switches

# Extension and Connecting Services Switch



- Let's connect them the way we should, redundantly (the previous slides showed a simple interconnect just as an example)

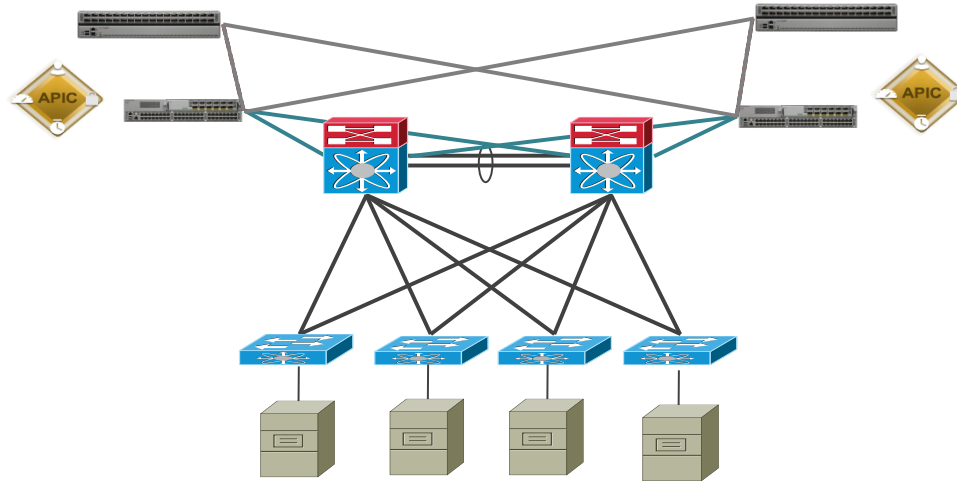
# Extension and Connecting Services Switch



- Let's use a small spine switch
- We are starting with a very small fabric

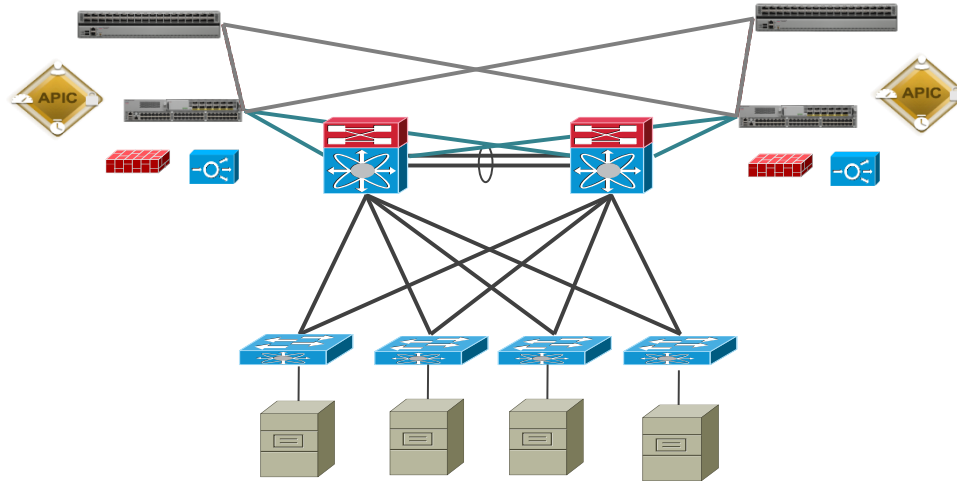


# Extension and Connecting Services Switch



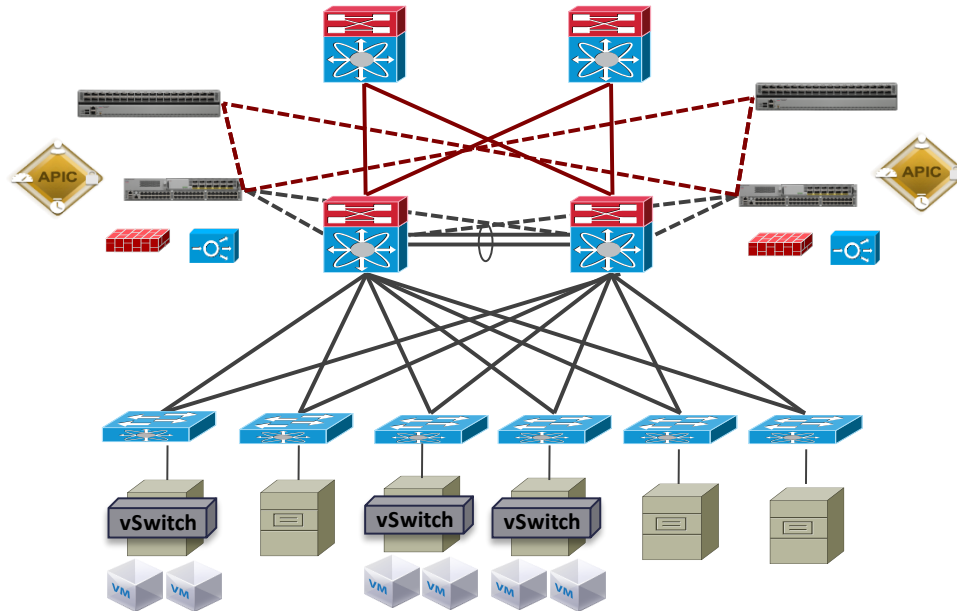
- Same picture as seen if the fabric was a services switch

# Extension and Connecting Services Switch



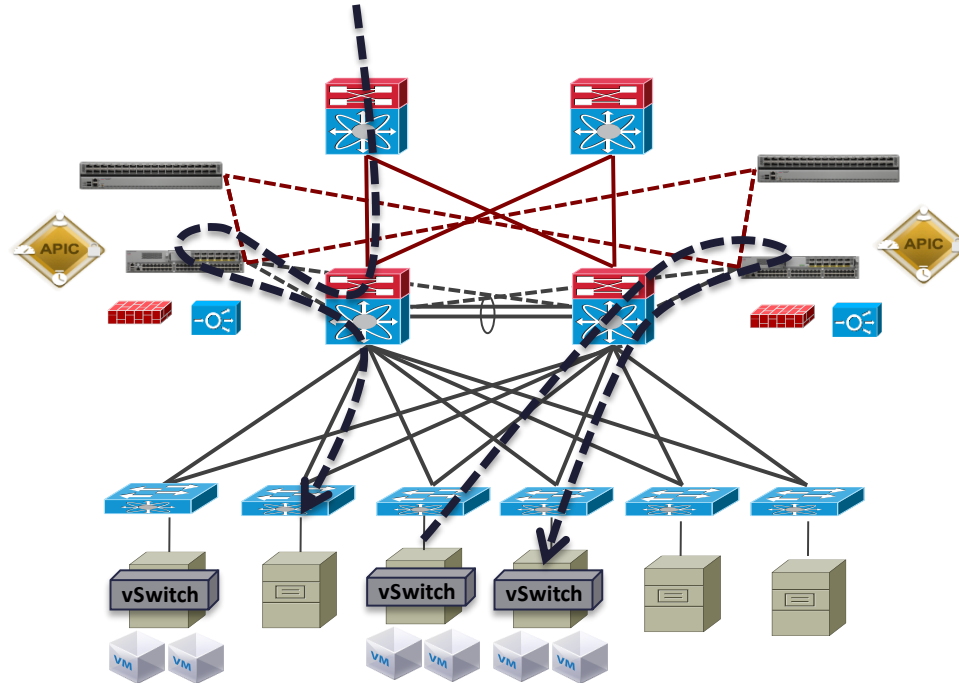
- Attach the L4-7 Services
- They can be physical and/or virtual

# Extension and Connecting Services Switch



- Add in the Core/WAN and the Servers and vSwitches

# Extension and Connecting Services Switch



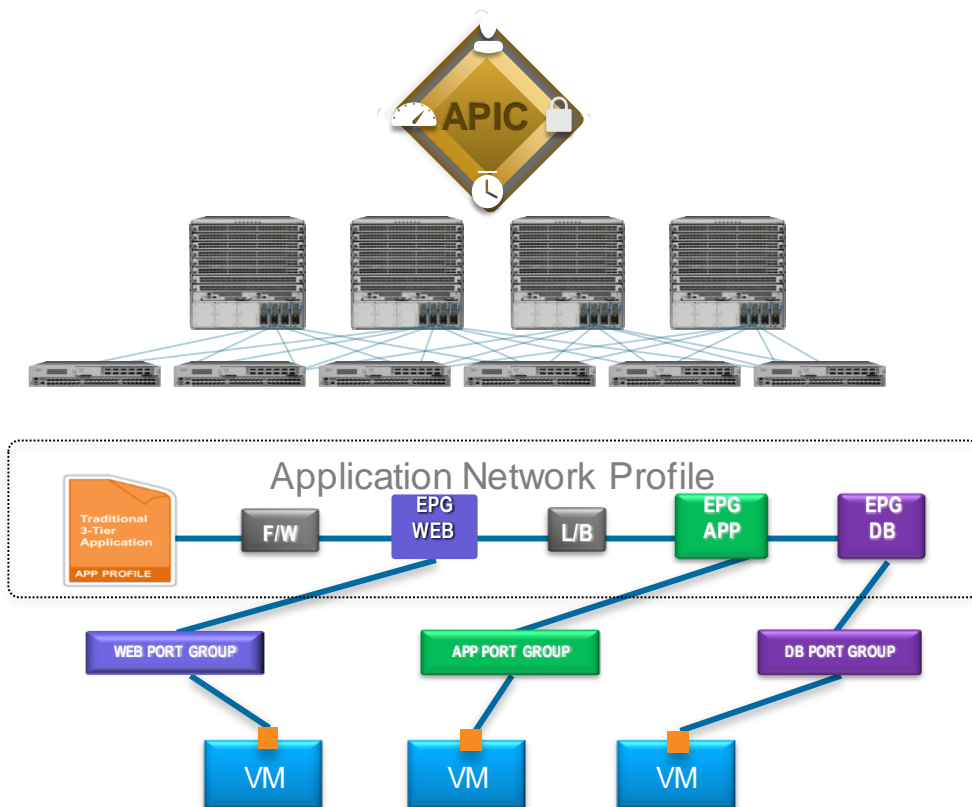
- Activate the services, leverage the services chaining and dynamic provisioning
- Leverage the fabric as the layer 3 gateway for all the other VLAN's

A long-exposure photograph of a city street at night. The foreground is filled with vibrant, multi-colored light trails from moving vehicles, creating a sense of motion. In the background, a pedestrian bridge spans the street, and various city buildings are illuminated with lights. The overall scene is a dynamic urban environment.

# Extending ACI - Application Virtual Switch



# Hypervisor Integration with ACI



- ACI Fabric implements policy on Virtual Networks by mapping Endpoints to EPGs
- Endpoints in a Virtualised environment are represented as the vNICs
- VMM applies network configuration by placement of vNICs into Port Groups or VM Networks
- EPGs are exposed to the VMM as a 1:1 mapping to Port Groups or VM Networks

# VMWare Integration

## Three Different Options

### Distributed Virtual Switch (DVS)



- Encapsulations: VLAN
- Installation: Native
- VM discovery: LLDP
- Software/Licenses: vCenter with Enterprise+ License

### vCenter + vShield



- Encapsulations: VLAN, VXLAN
- Installation: Native
- VM discovery: LLDP
- Software/Licenses: vCenter with Enterprise+ License, vShield Manager with vShield License

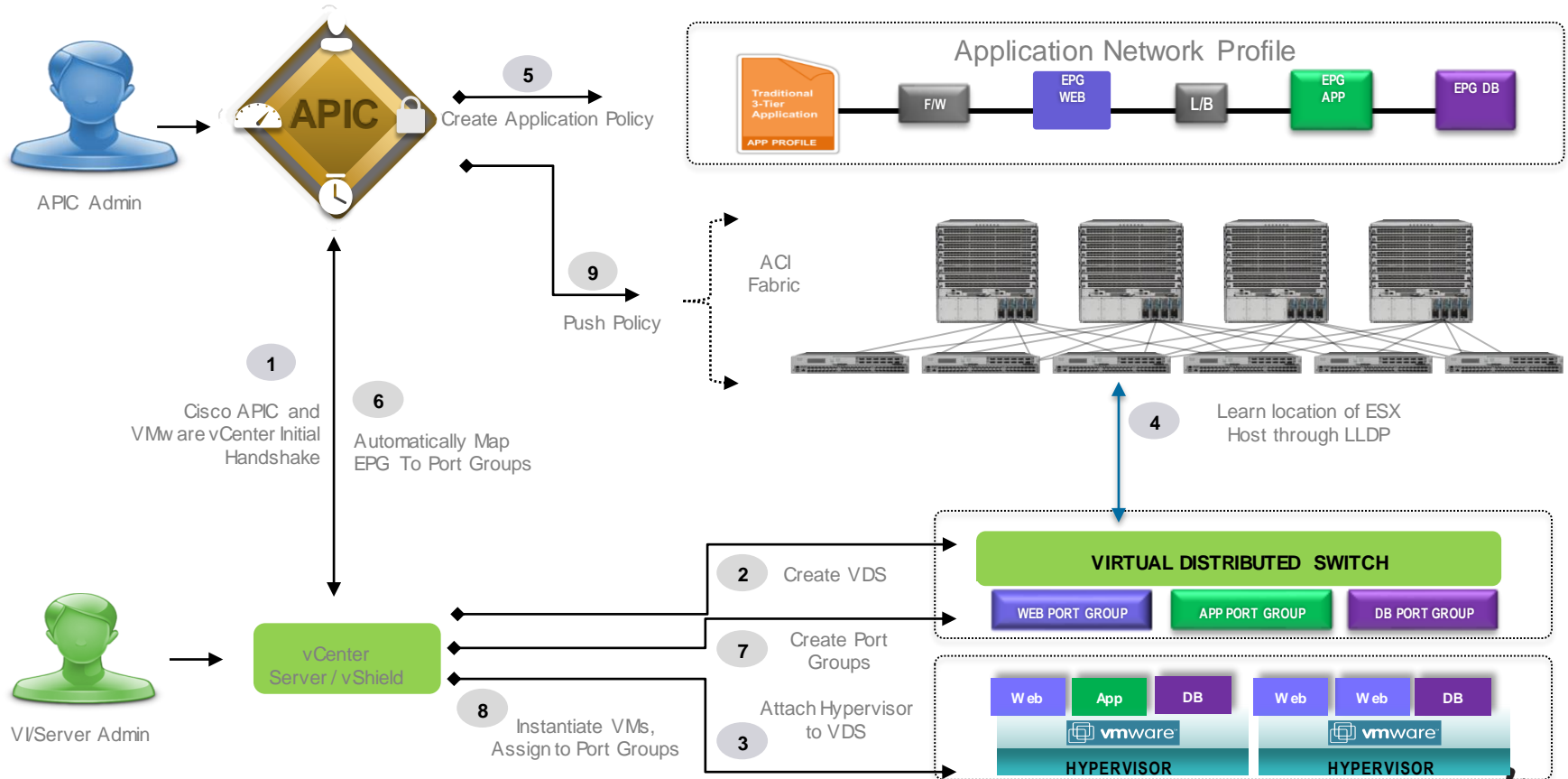
### Application Virtual Switch (AVS)



- Encapsulations: VLAN, VXLAN
- Installation: VIB through VUM or Console
- VM discovery: OpFlex
- Software/Licenses: vCenter with Enterprise+ License



# ACI Hypervisor Integration – VMware DVS/vShield



# ACI Hypervisor Integration – VMware DVS

The screenshot displays the Cisco ACI GUI with the 'VM NETWORKING' tab selected. The left sidebar shows the 'Inventory' tree with 'vmware-dvs' highlighted. The main panel shows the 'Domain - vmware-dvs' configuration page. The 'PROPERTIES' section displays the 'Name: vmware-dvs' and a table of 'Controllers'.

**Domain - vmware-dvs**

**PROPERTIES**


Name: **vmware-dvs**

Controllers:

NAME	STATE	MODEL	SERIAL	REVISION	HYPERVISORS	VIRTUAL MACHINES
vCenter	Online	VMware vCenter Server 5.1.0 bu...	C974564D-08...	5.1.0	3	6

PAGE 1 OF 1 | ITEMS PER PAGE: 15 | DISPLAYING OBJECTS 1 - 1 OF 1

# ACI Hypervisor Integration – VMware



The screenshot displays the vSphere Client interface. The left sidebar shows the inventory tree with the path: localhost > DC3 > vmware-avs > DC3|vmware-avs > uplink > DC3|vmware-dvs > DC3|vmware-dvs-DVUplinks-684 > DC3|vmware-vshield > DC3|vmware-vshield-DVUplinks-687. The main pane is titled "DC3|vmware-vshield" and shows the "Configuration" tab. The "What is a vSphere Distributed Switch?" section explains that a vSphere Distributed Switch acts as a single virtual switch across all associated hosts, allowing virtual machines to maintain consistent network configuration as they migrate across hosts. It also describes the three parts of distributed virtual networking configuration: datacenter level, host level, and virtual machine level. A diagram illustrates the vSphere Distributed Switch connecting to three hosts. The "Recent Tasks" table at the bottom is empty.

localhost - vSphere Client

File Edit View Inventory Administration Plug-ins Help

Home Inventory Networking Search Inventory

localhost

- DC1
- DC2
- DC3
  - DC3|vmware-avs
    - DC3|vmware-avs
      - uplink
        - Coke|www.Coke.com|DB
        - Coke|www.Coke.com|WEB
        - vtep
    - DC3|vmware-dvs
      - DC3|vmware-dvs-DVUplinks-684
        - Coke|www.Coke.com|DB
        - Coke|www.Coke.com|WEB
    - DC3|vmware-vshield
      - DC3|vmware-vshield-DVUplinks-687
        - Coke|www.Coke.com|APP
        - vxx-dvs-687-virtualwire-6-sid-8394309-
        - vxx-vmknipg-dvs-687-4094-57c29b34-
    - VM Network

DC3|vmware-vshield

Getting Started Summary Networks Ports Resource Allocation Configuration Virtual Machines Hosts Tasks & Events Alarms Permissions

close tab

### What is a vSphere Distributed Switch?

A vSphere Distributed Switch acts as a single virtual switch across all associated hosts. This allows virtual machines to maintain consistent network configuration as they migrate across hosts.

Distributed virtual networking configuration consists of three parts. The first part takes place at the datacenter level, where vSphere Distributed Switches are created, and hosts and distributed port groups are added to vSphere Distributed Switches. The second part takes place at the host level, where host ports and networking services are associated with vSphere Distributed Switches either through individual host networking configuration or using host profiles. The third part takes place at the virtual machine level, where virtual machine NICs are connected to distributed port groups either through individual virtual machine NIC configuration or by migrating virtual machine networking from the vSphere Distributed Switch itself.

Explore Further

Recent Tasks

Name, Target or Status contains: Clear

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Time	Start Time	Completed Time
------	--------	--------	---------	--------------	----------------	----------------------	------------	----------------

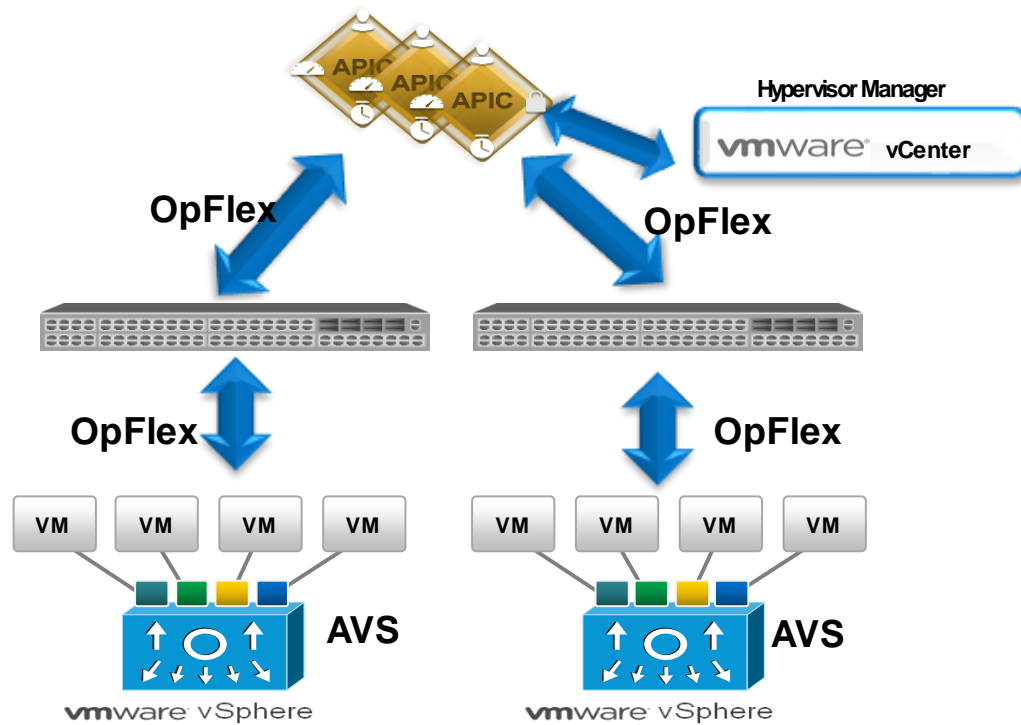
Tasks Alarms

root

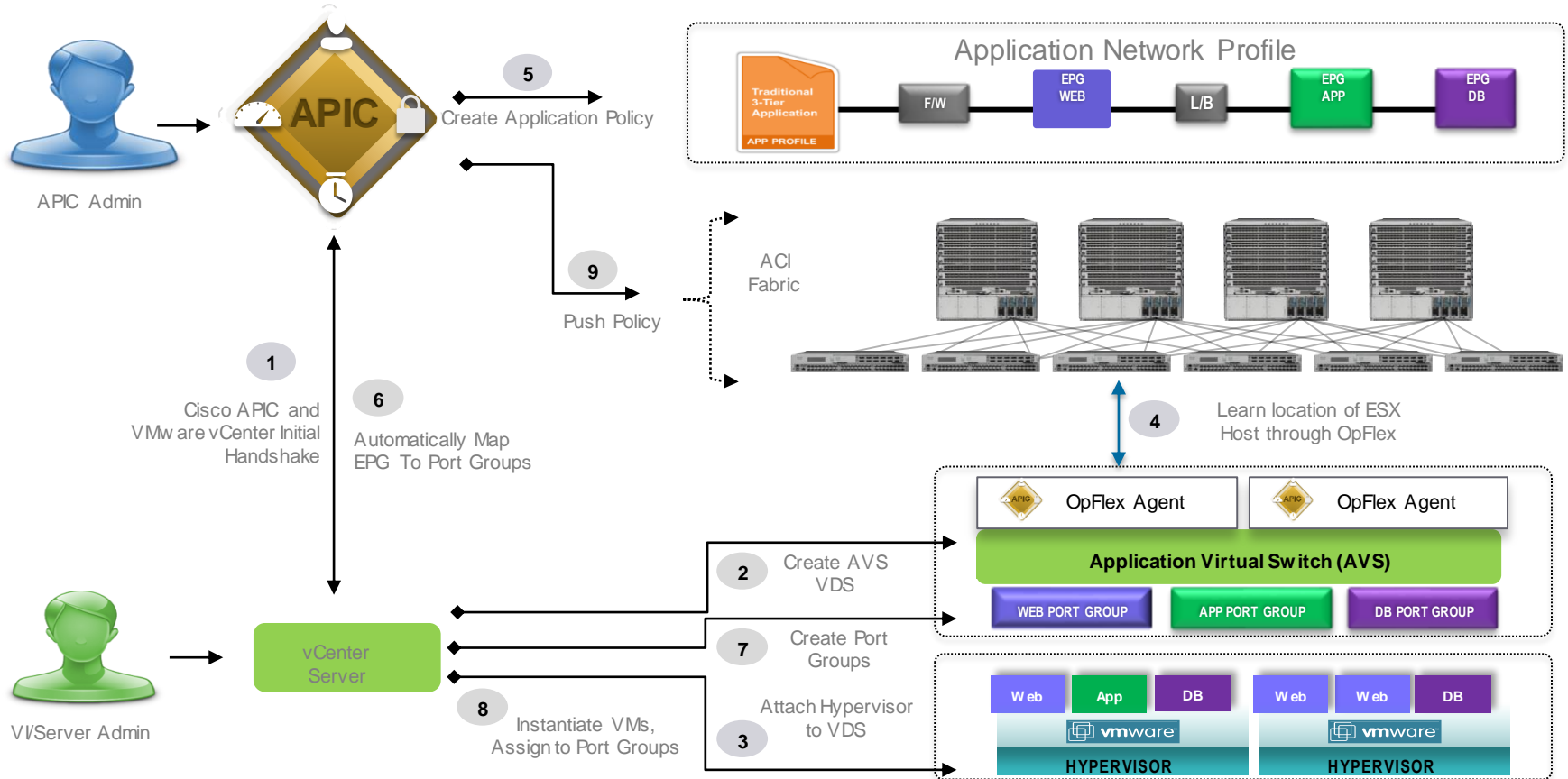
2:22 PM 5/5/2014

# Application Virtual Switch with OpFlex in ACI Fabric

- AVS: First Virtual Leaf to implement OpFlex
- Network policy communicated from APIC to AVS through N9k using OpFlex
- Increased control plane scale through APIC Cluster and Leaf Node
- APIC communicates with vCenter Server for Port Group creation



# ACI Hypervisor Integration – AVS

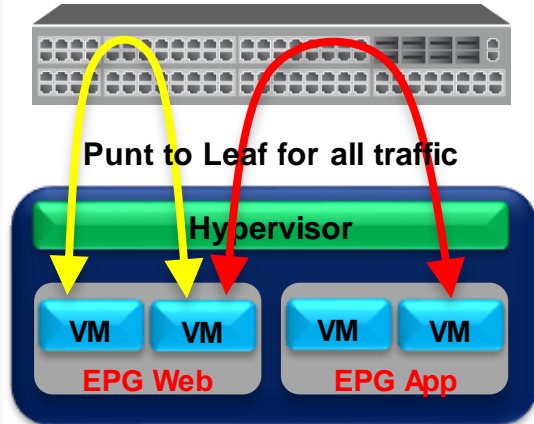


# Virtual Leaf Switching Modes

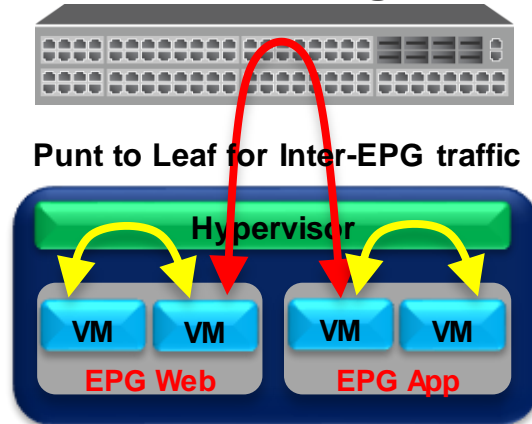
- FEX Mode: All traffic sent to Leaf for switching
- Local Switching (LS) Mode: Intra-EPGs traffic switched on the same host
- Full Switching (FS) Mode: Full APIC policy enforcement on server

FCS

## FEX Mode

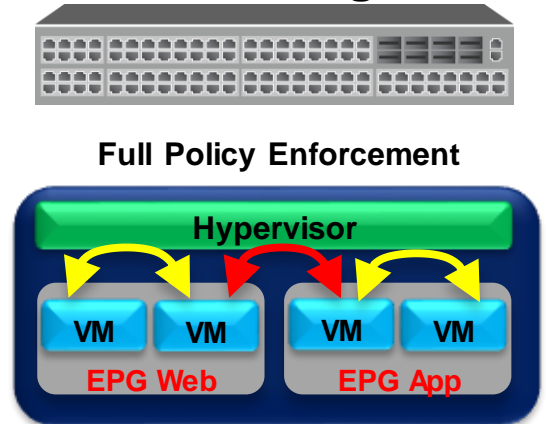


## Local Switching Mode



CY15

## Full Switching Mode



# ACI Hypervisor Integration – Cisco AVS

## CREATE VCENTER DOMAIN



Specify vCenter domain users and controllers

Name:

Virtual Switch: ☐ VMWare vSphere Distributed Switch ☒ Cisco AVS

Switching Preference: ☒ Fex Enable ☐ Fex Disable

Associated Attachable Entity Profile:

VXLAN Pool:

Multicast Address:

vCenter Credentials:

Profile Name	Username	Description
vmwareAdmin	root	

vCenter:

Name	IP	Type	Stats Collection
vCenter	192.168.30.3	vCenter	Disabled

Name of VMM Domain

Type of vSwitch (DVS or AVS)

Switching mode (FEX or Normal)

Associated Attachable Entity Profile (AEP)

VXLAN Pool

Multicast Pool

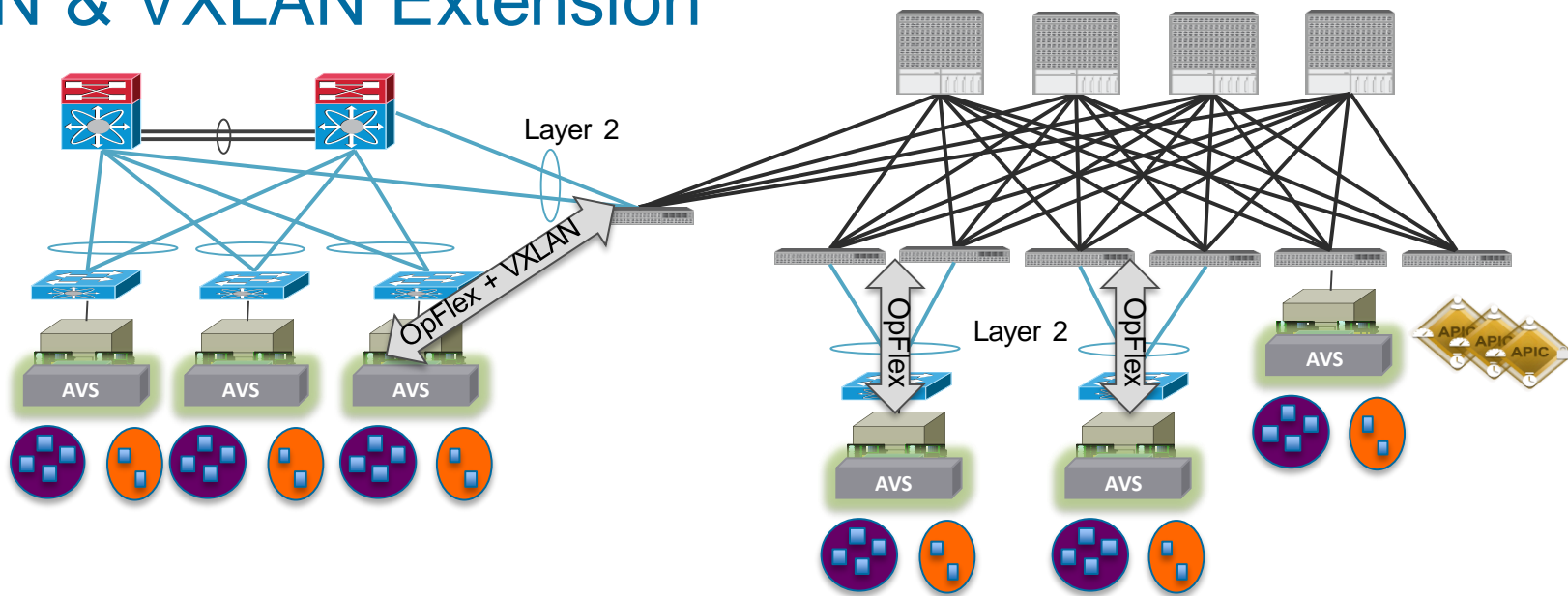
vCenter Administrator Credentials

vCenter server information



# Extending ACI to Existing Virtual & Physical Networks

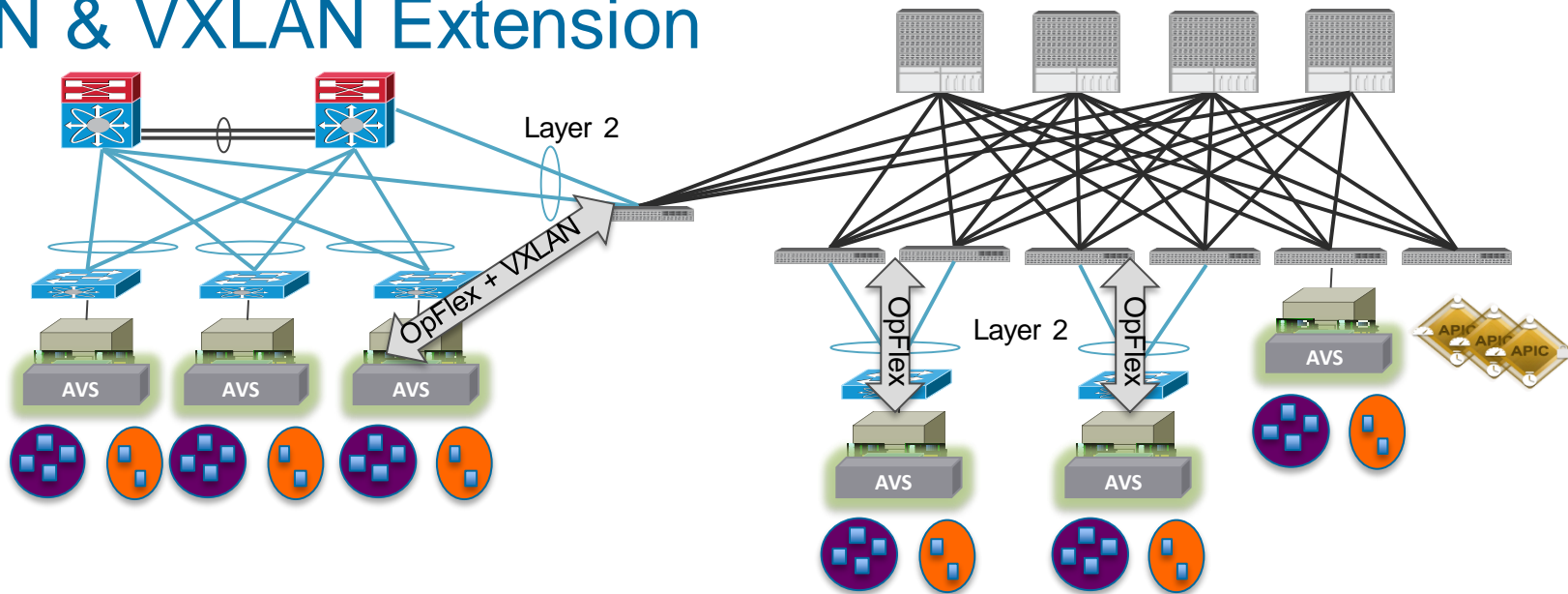
## VLAN & VXLAN Extension



- AVS supports OpFlex to integrate with APIC
- Supports a **Full multi-hop Layer 2 Network** between Nexus 9k and AVS: Investment Protection
- Layer 2 network is required to support OpFlex bootstrapping in this phase

# Extending ACI to Existing Virtual & Physical Networks

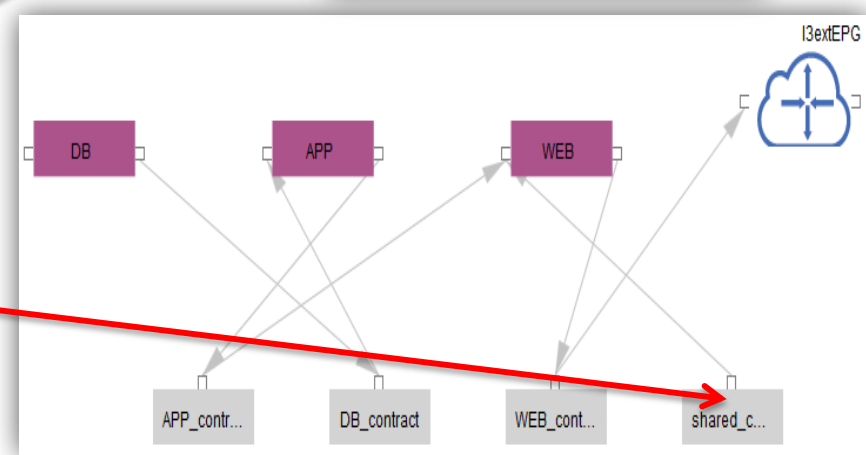
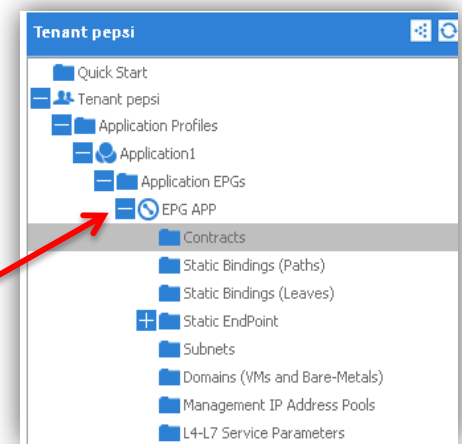
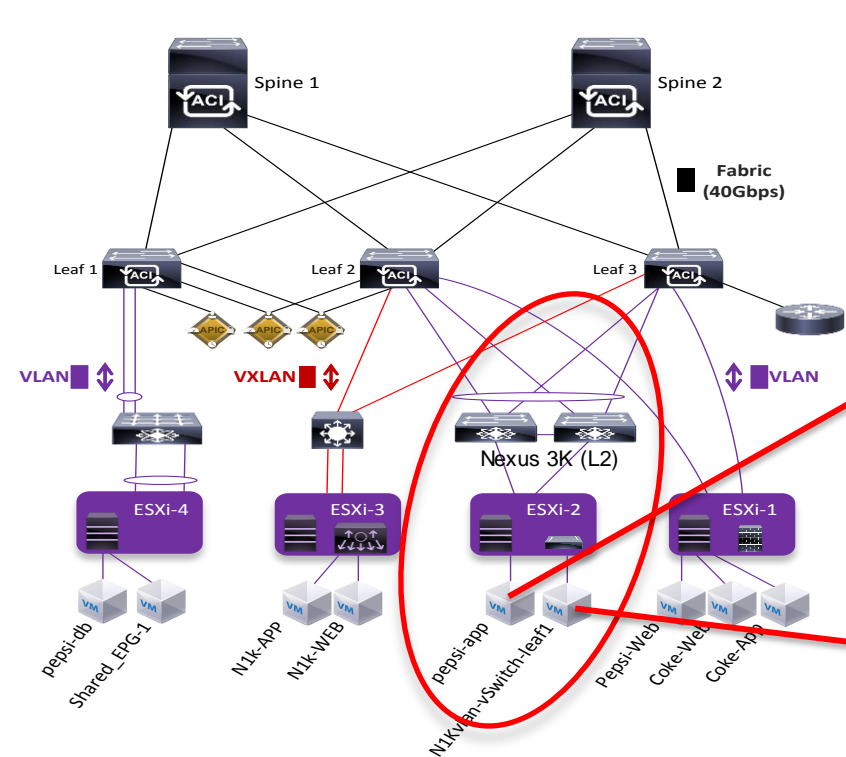
## VLAN & VXLAN Extension



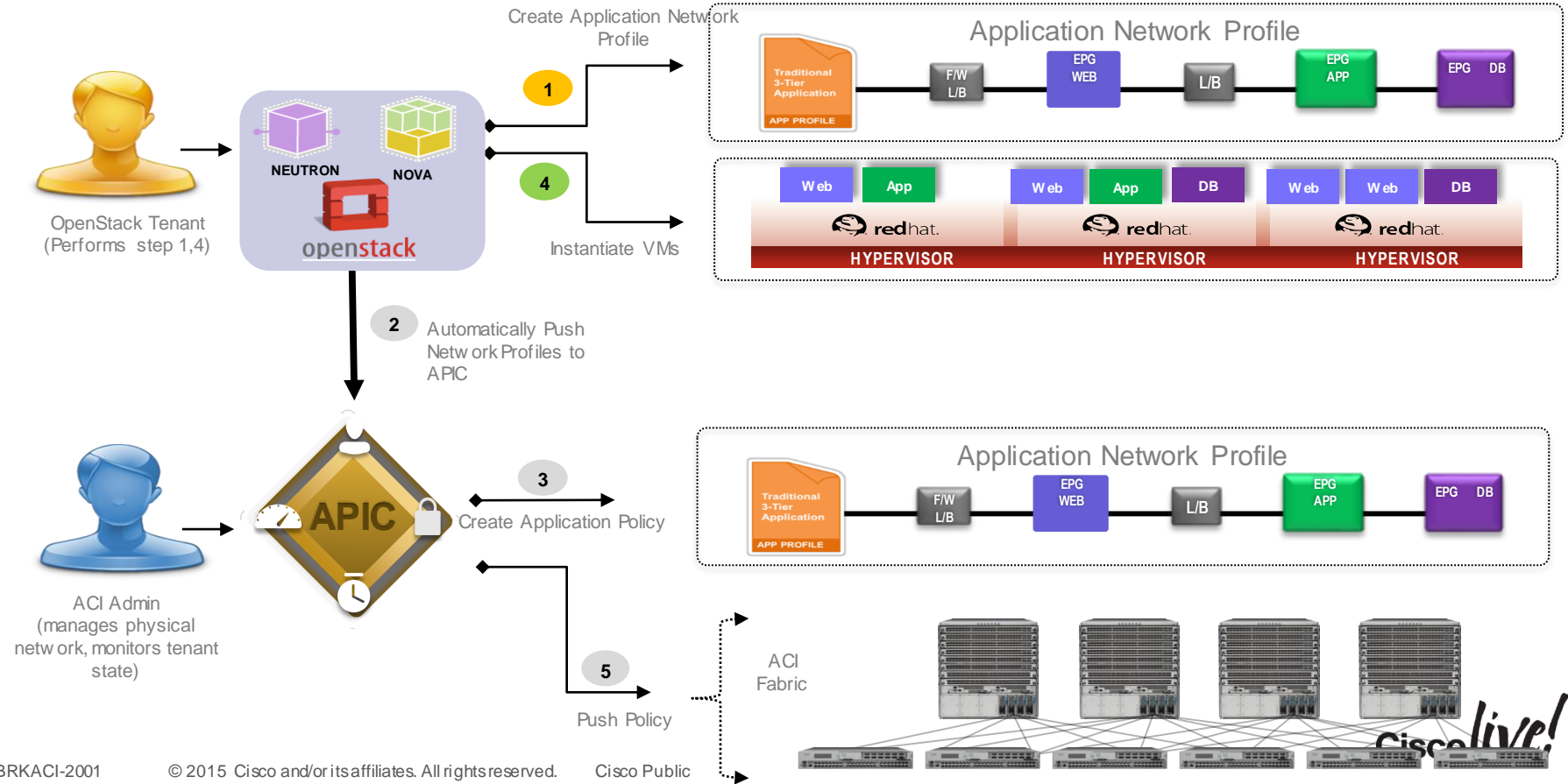
- Supports VLAN and VXLAN for transport (Recommend VXLAN to automate new workload)
- Existing Network need to have 1 Infrastructure VLAN for VXLAN transport
- Multicast: Turn on IGMP Snooping
- Recommend 1 L2 Multicast group per EPG

# ACI - INVESTMENT PROTECTION FOR INTEGRATION / MIGRATION


## Remote VTEP (Virtual) via AVS



# ACI OpenStack Integration



# ACI OpenStack & ODL Integration



[Main page](#)  
[Recent changes](#)  
[Random page](#)  
[Help](#)  
▼ [Tools](#)  
[What links here](#)  
[Related changes](#)  
[Special pages](#)  
[Printable version](#)  
[Permanent link](#)  
[Page information](#)

Page **Discussion**

## Group Policy:Main

**Contents** [\[hide\]](#)

- 1 Introduction
- 2 User Guide
- 3 Developer Guide
  - 3.1 Documentation
  - 3.2 Project Tracking
  - 3.3 Project Subgroups
- 4 Other Documents
- 5 Google+/YouTube information
- 6 All subpages

### Introduction

The group-based policy project defines an application-centric policy model for OpenDaylight that separates information about application connectivity requirements from information about the underlying details of the network infrastructure.



[https://wiki.opendaylight.org/view/Group\\_Policy:Main](https://wiki.opendaylight.org/view/Group_Policy:Main)

<https://github.com/noironetworks/vxlan-gbp>




## VXLAN Group Policy Extension

[3 commits](#) [1 branch](#) [0 releases](#) [1 contributor](#)

[branch: master](#) [vxlan-gbp](#)

[Fix markdown](#)

 **tgraf** authored 19 days ago

latest commit [9087c55bd2](#)

<a href="#">README.md</a>	Fix markdown	19 days ago
<a href="#">Vagrantfile</a>	Vagrantfile and basic README	19 days ago
<a href="#">bootstrap.sh</a>	Vagrantfile and basic README	19 days ago

### README.md

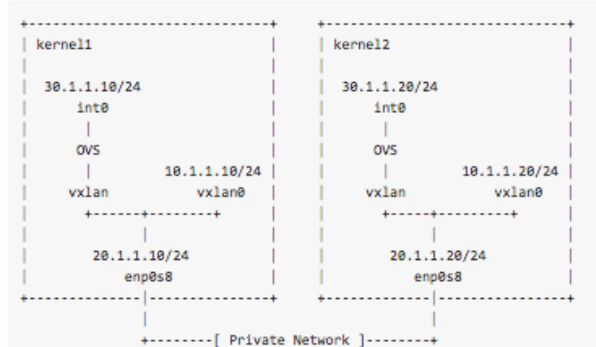
## VXLAN Group Policy Extension

This tutorial walks through the steps required to get a VXLAN-GBP testbed up and running to play with the technology.

### FastTrack: Vagrant

Running `vagrant up` will provision two Fedora 20 based VMs named `kernel1` and `kernel12` with a VXLAN-GBP enabled kernel, `iproute2`, and Open vSwitch. Use `vagrant ssh` to log into the VMs.

This will give you the following topology:

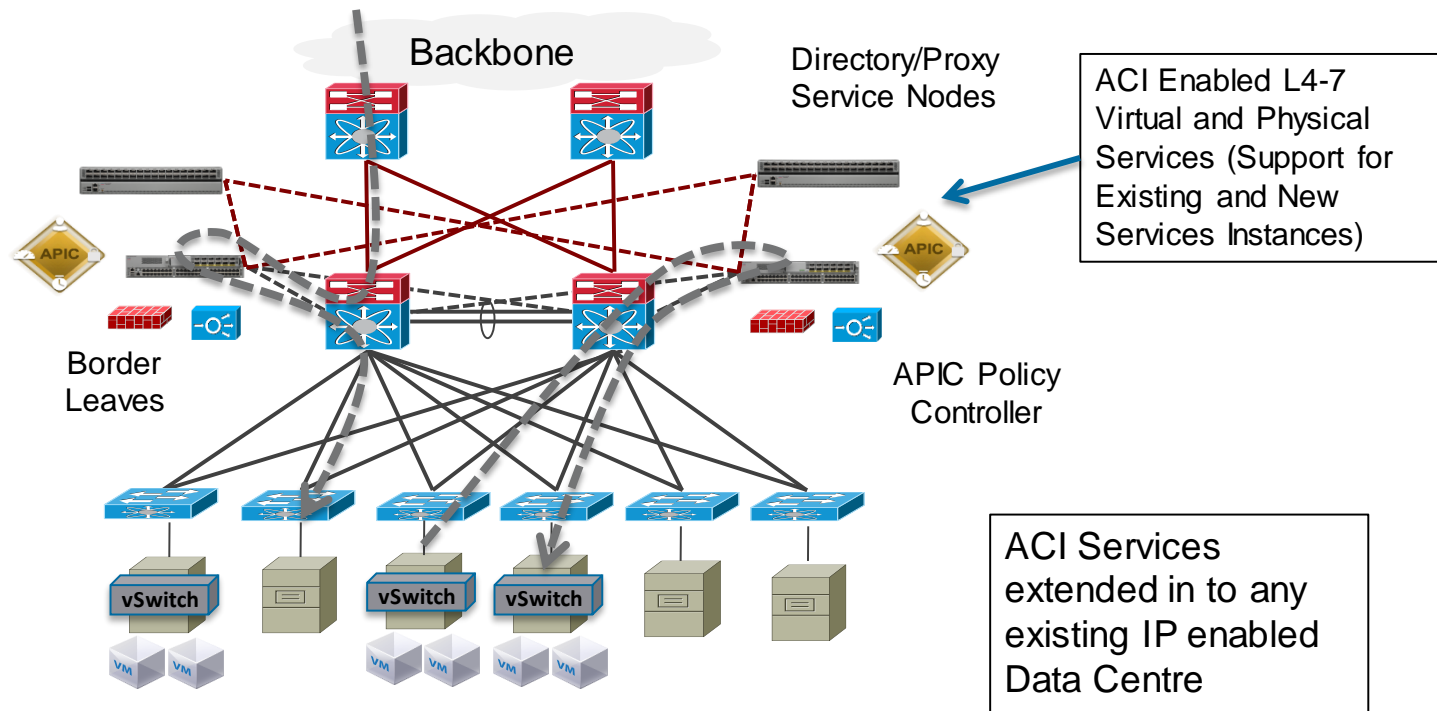


```
graph TD
    subgraph kernel1 [kernel1]
        int0[30.1.1.10/24 int0] --> OVS1[OVS]
        OVS1 --> vxlan0[10.1.1.10/24 vxlan0]
    end
    subgraph kernel12 [kernel12]
        int0[30.1.1.20/24 int0] --> OVS2[OVS]
        OVS2 --> vxlan0[10.1.1.20/24 vxlan0]
    end
    vxlan0 --> PrivateNetwork[Private Network]
    PrivateNetwork --> enp0s8_1[20.1.1.10/24 enp0s8]
    PrivateNetwork --> enp0s8_2[20.1.1.20/24 enp0s8]
```

# Extending ACI in to Current Data Centre's

## Upgrade to ACI Based Services Block

1. Install an ACI Services Block
2. Leverage Existing L4-7 services nodes 'or' Leverage new services that become fully automated via APIC device package interface
3. Extend VLAN == EPG from existing network into ACI Services Pod
4. Migrate default GW to ACI services nodes
5. Manage all access control and services through APIC while maintaining existing switching





A nighttime photograph of a city street with a pedestrian bridge and tall buildings in the background. The foreground is dominated by long-exposure light trails from vehicles, creating a sense of motion and connectivity.

# Extending ACI into Existing Data Centres Adding Remote Switch Nodes







# ACI Spine Proxy == LISP Proxy Tunnel Router + Map-Server

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: August 18, 2014

V. Moreno  
F. Maino  
D. Lewis  
M. Smith  
S. Sinha  
Cisco Systems  
February 14, 2014

## LISP Deployment Considerations in Data Center Networks draft-moreno-lisp-datacenter-deployment-00

### Abstract

This document discusses scenarios and implications of LISP based overlay deployment in the Data Center. The alternatives for topological location of the different LISP functions are analyzed in the context of the most prevalent Data Center topologies. The role and deployment of LISP in the Wide Area Network and Data Center Interconnection are also discussed.

### Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

### 4.2.2. Map-Server/Resolver in-line

The Map-Server/Resolver function can be deployed in-line on one or more network nodes that participate in the clos topology either as Spine nodes or Leaf nodes. In-line Map-Server/Resolvers will receive both Control Plane and Data Plane traffic.

#### 4.2.2.1. Map-Server/Resolver at the Spine

The Map-Server/Resolver function can be deployed on Spine nodes. In order to guarantee resiliency, Map-Server/Resolvers are deployed on at least two Spine nodes, but preferably on all Spine nodes.

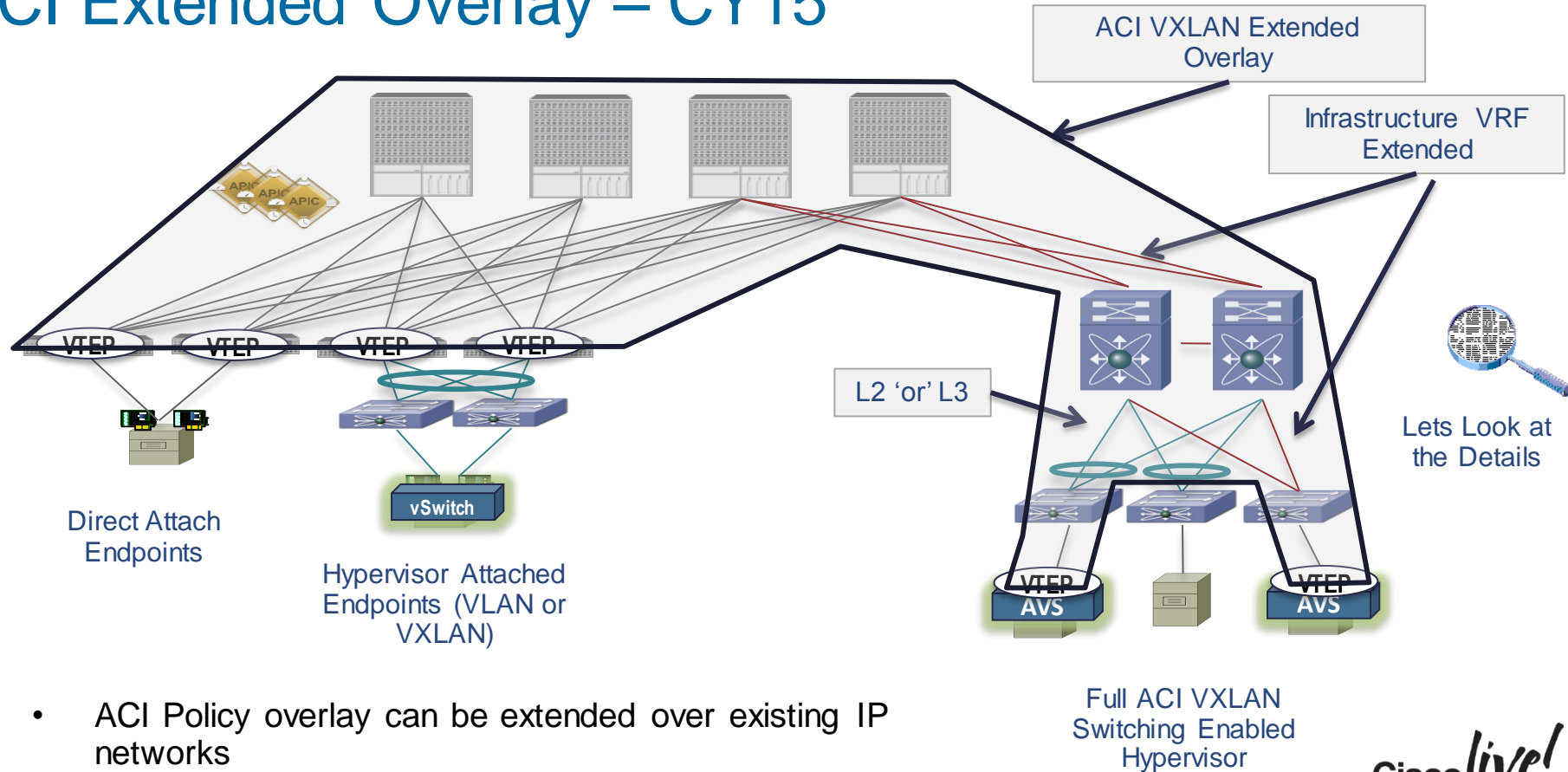
Any map-register messages must be sent to all Map-Server/Resolver enabled Spine nodes in order to ensure synchronization of the mapping system. Therefore in a system where all Spine nodes host the Map-Server/Resolver functionality, all Spine nodes will have complete mapping state for all EIDs in the fabric. Alternatively, map-register messages can be sent to a single Map-Server and the state may be relayed by other methods to other Map-Servers in the Spine.

When all Spine nodes host the Map-Server/Resolver functionality, all data plane traffic that cuts across different leaf nodes will traverse a Spine node that has the full LISP mapping state for the fabric. In this deterministic topology it is possible to implement avoid transient drops that may occur when looking up destinations that have not been previously cached at the ITRs.

<http://tools.ietf.org/html/draft-moreno-lisp-datacenter-deployment-00>

# Extension of the ACI Overlay to Remote AVS

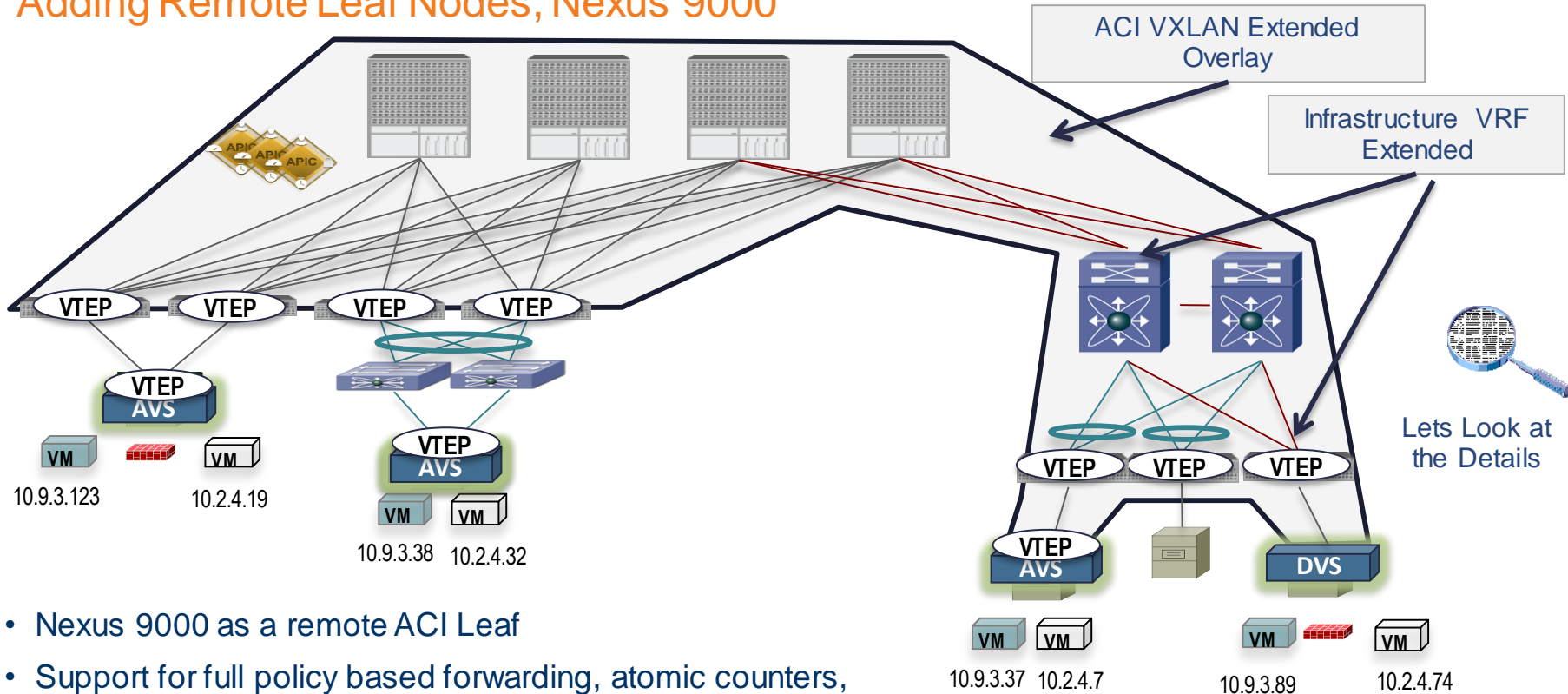
## ACI Extended Overlay – CY15



- ACI Policy overlay can be extended over existing IP networks

# Forwarding within the Extended Overlay

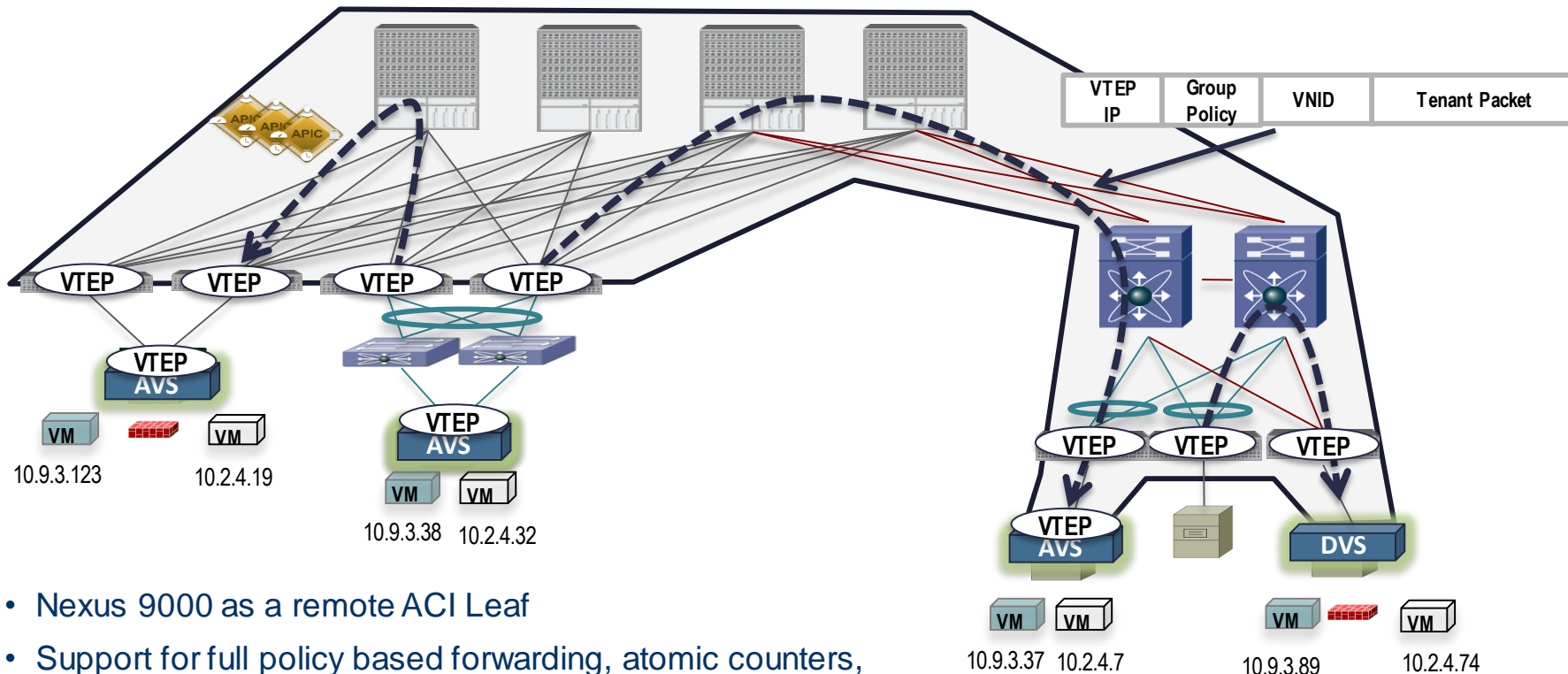
## Adding Remote Leaf Nodes, Nexus 9000



- Nexus 9000 as a remote ACI Leaf
- Support for full policy based forwarding, atomic counters, zero touch install, health scores

# Forwarding within the Extended Overlay

## Adding Remote Leaf Nodes, Nexus 9000

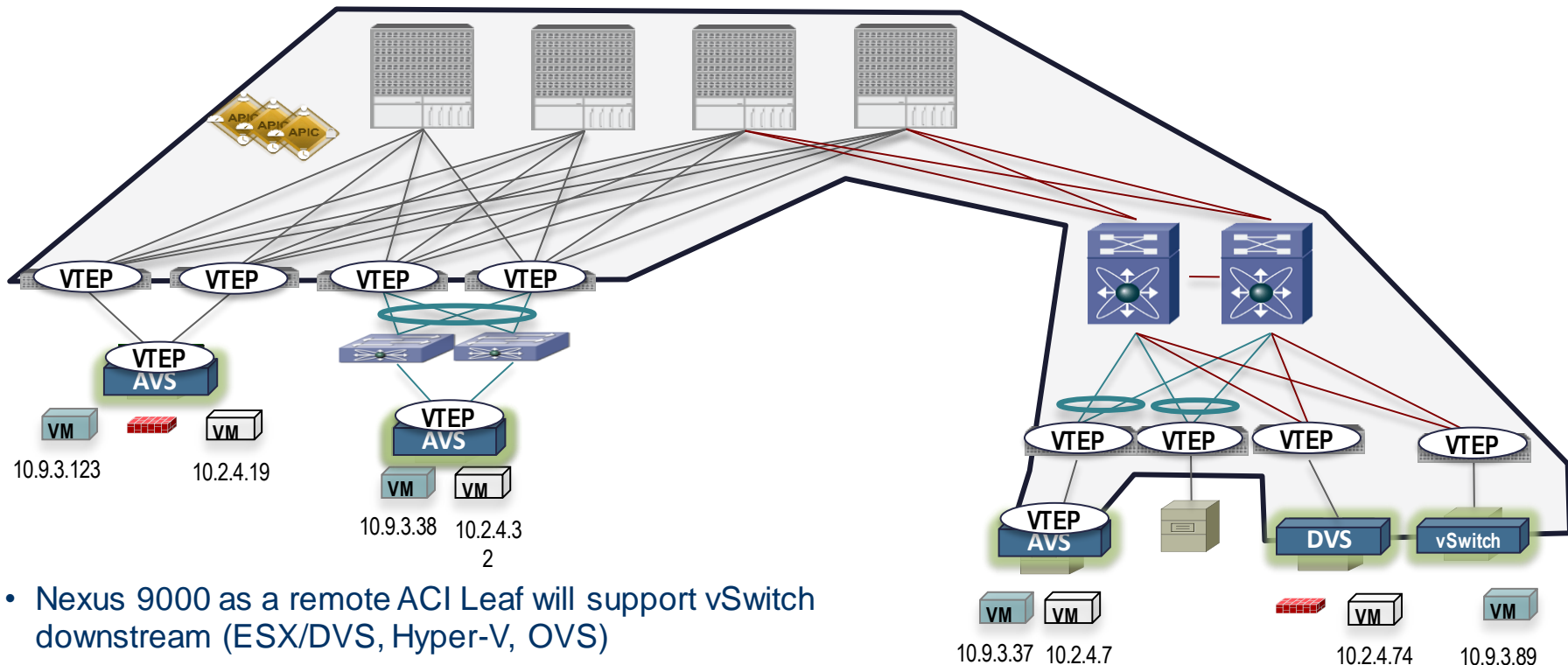


- Nexus 9000 as a remote ACI Leaf
- Support for full policy based forwarding, atomic counters, zero touch install, health scores



# Forwarding within the Extended Overlay

## Adding Remote Leaf Nodes, Nexus 9000

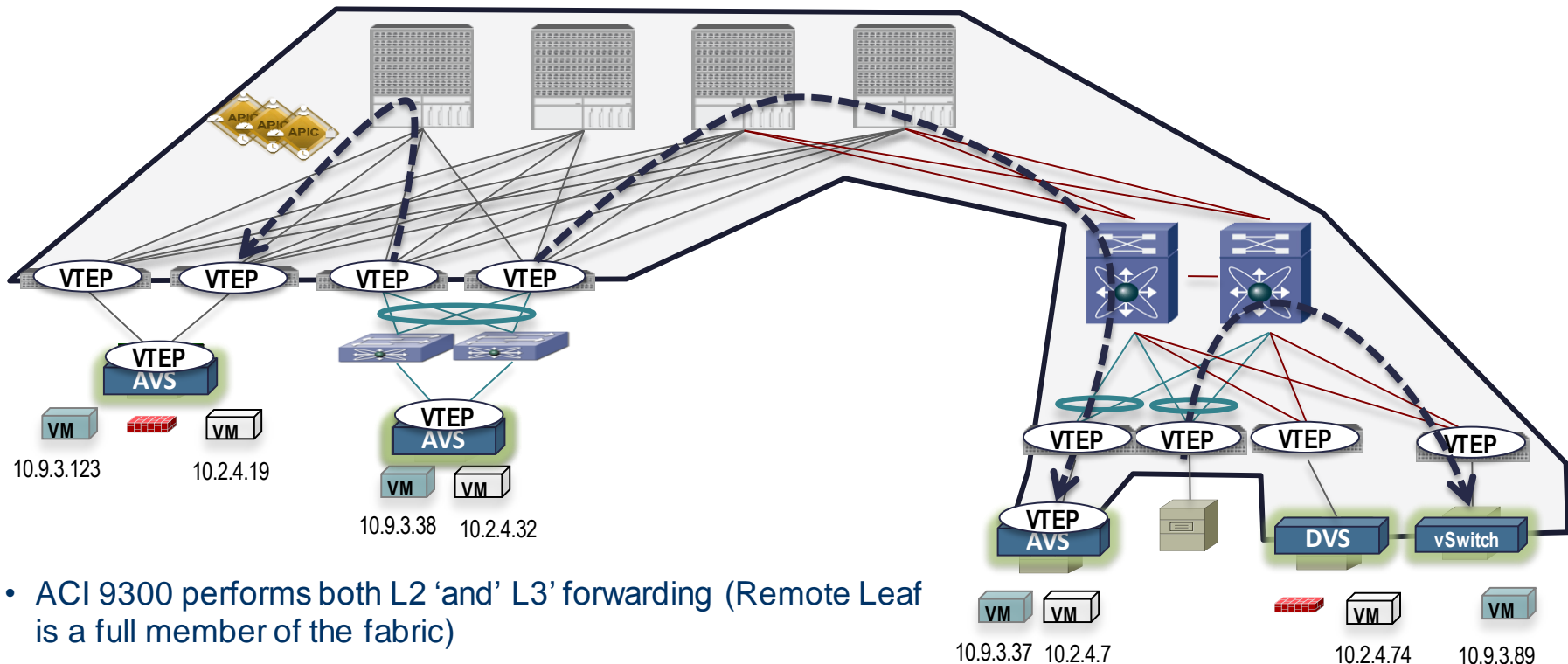


- Nexus 9000 as a remote ACI Leaf will support vSwitch downstream (ESX/DVS, Hyper-V, OVS)
- Leverage Existing Hypervisor implementations



# Forwarding within the Extended Overlay

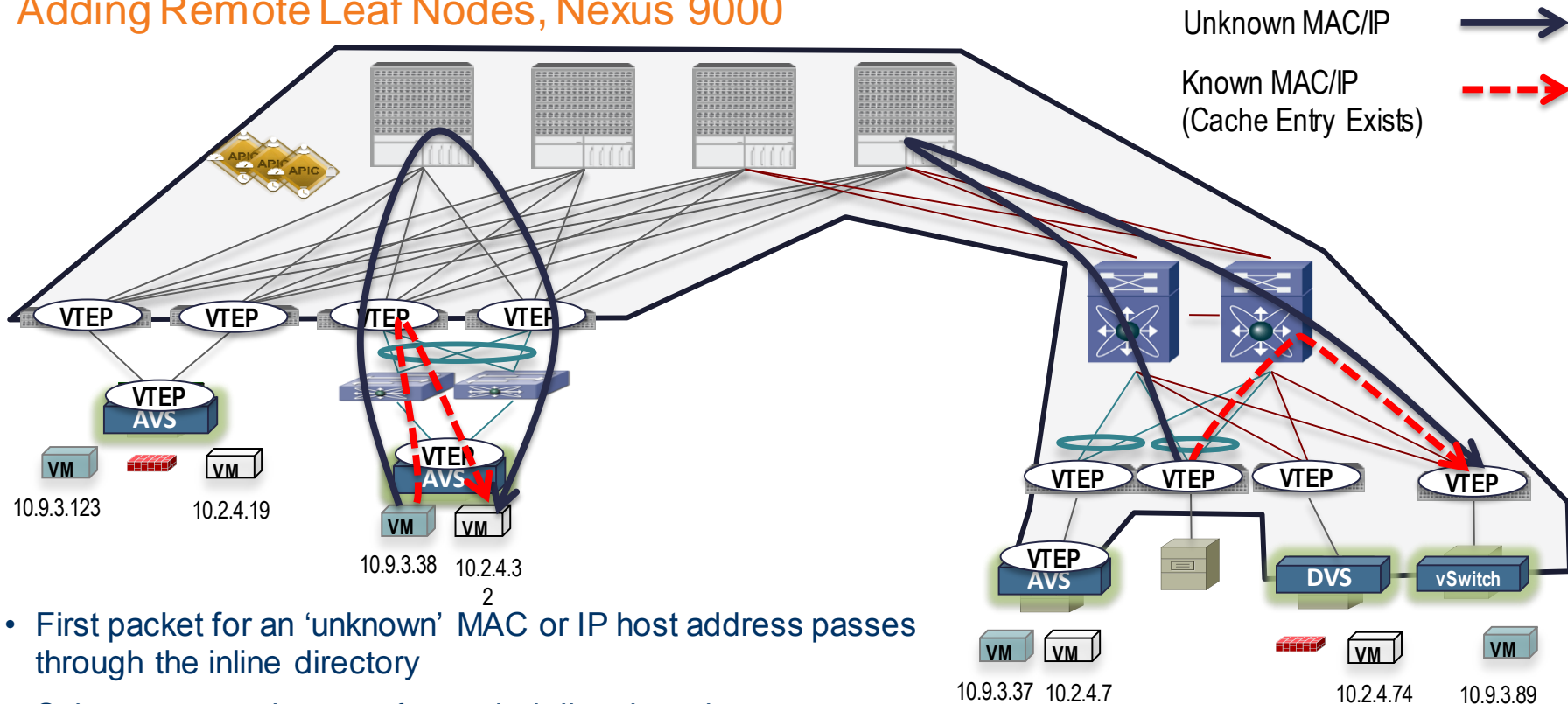
## Adding Remote Leaf Nodes, Nexus 9000



- ACI 9300 performs both L2 'and' L3' forwarding (Remote Leaf is a full member of the fabric)
- Remote 9300 Leaf performs full local inter EPG policy forwarding

# Forwarding within the Extended Overlay

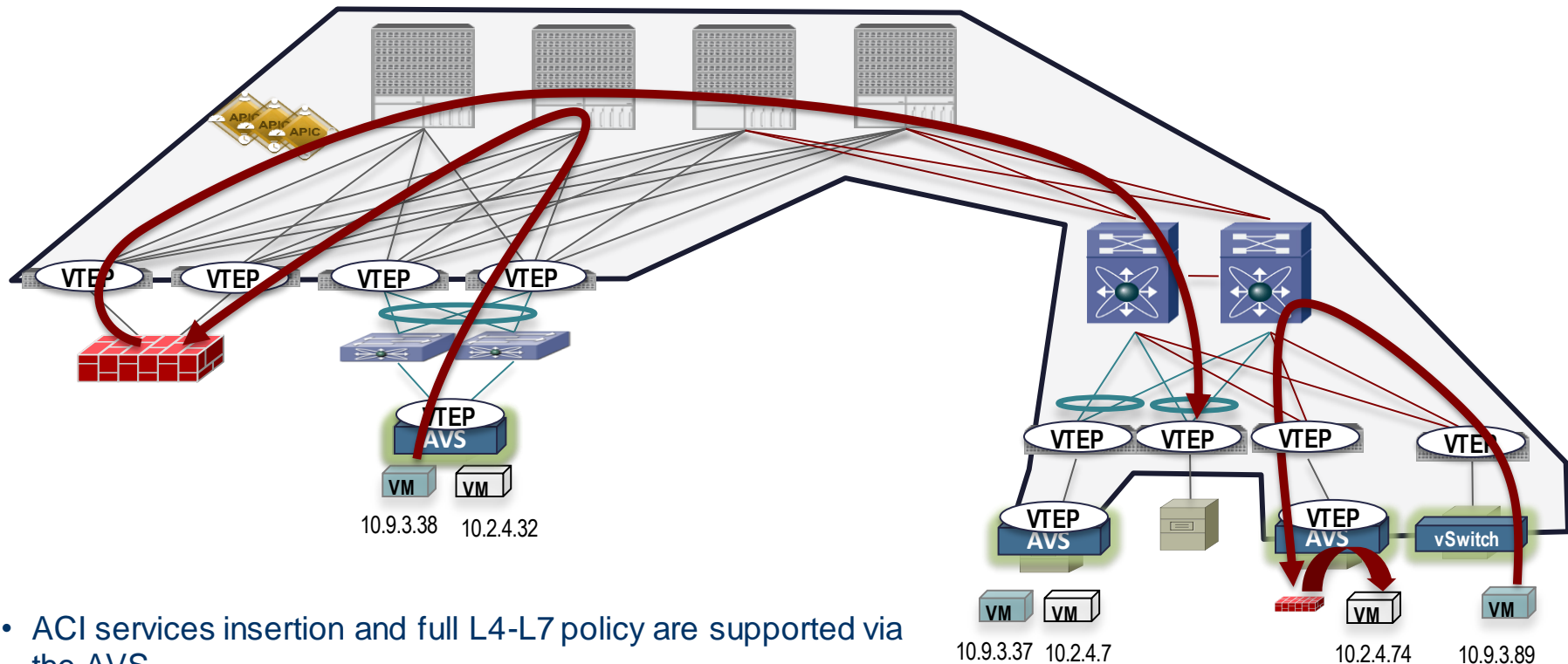
## Adding Remote Leaf Nodes, Nexus 9000



- First packet for an 'unknown' MAC or IP host address passes through the inline directory
- Subsequent packets are forwarded directly to the target VTEP (local switching)

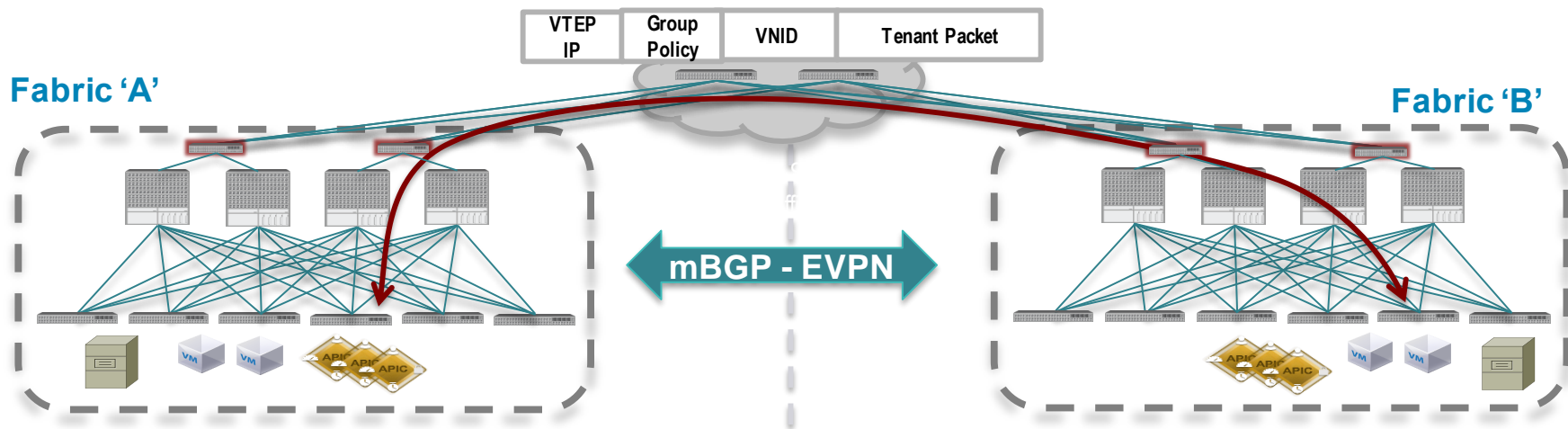
# Forwarding within the Extended Overlay

## Adding Remote Leaf Nodes, Nexus 9000



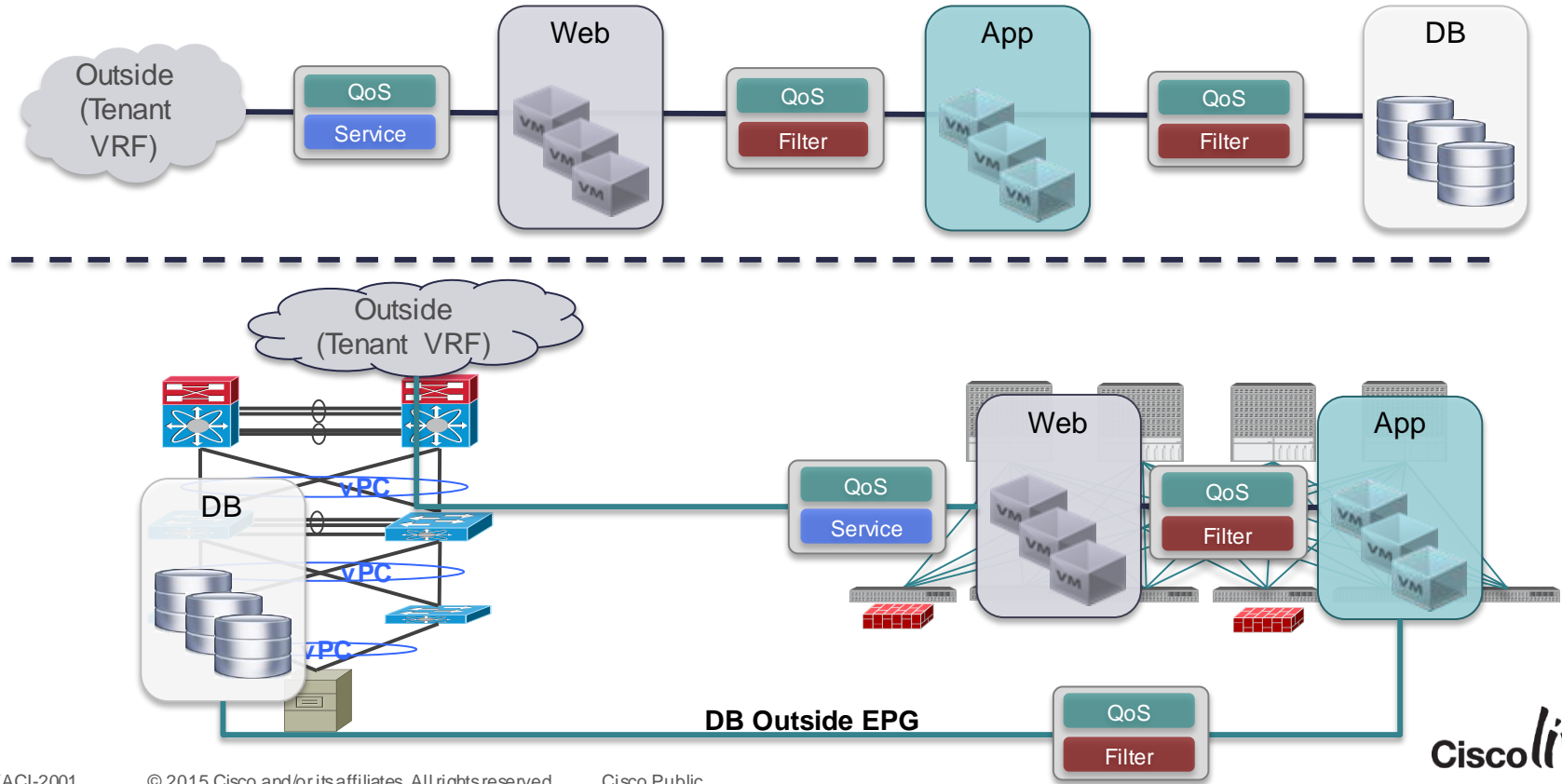
- ACI services insertion and full L4-L7 policy are supported via the AVS

# Multi-Site Fabrics



- Host Level Reachability Advertised between Fabrics via BGP
- Transit Network is IP Based
- Host Routes do not need to be advertised into transit network
- Policy Context is carried with packets as they traverse the transit IP Network
- Forwarding between multiple Fabrics is allowed (not limited to two sites)

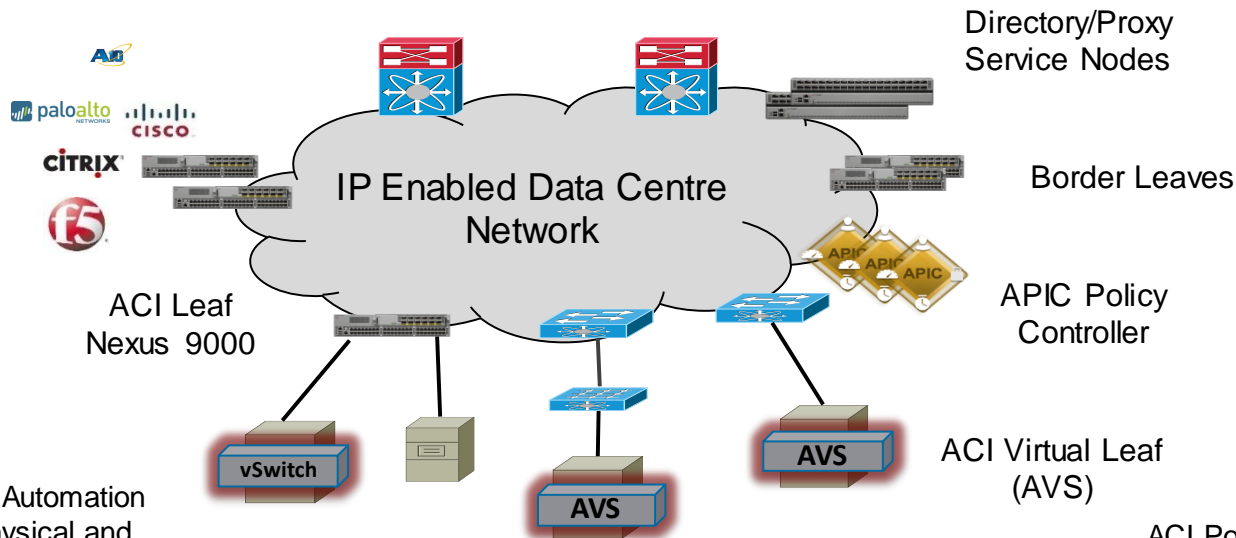
# Applications will Spread Across Existing and New Infrastructure



# ACI

It's an 'IP' Network

ACI Enabled L4-7  
Virtual and Physical  
Services (Support for  
Existing and New  
Services Instances)



ACI Services  
extended in to any  
existing IP  
enabled Data  
Centre

ACI Policy and Automation  
Extended to Physical and  
Existing Virtual Servers via  
Cisco Nexus 9000

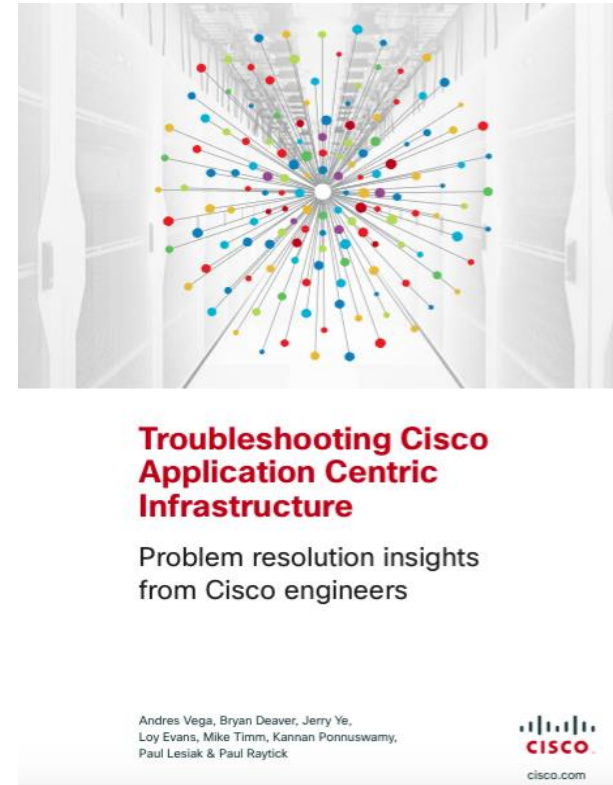
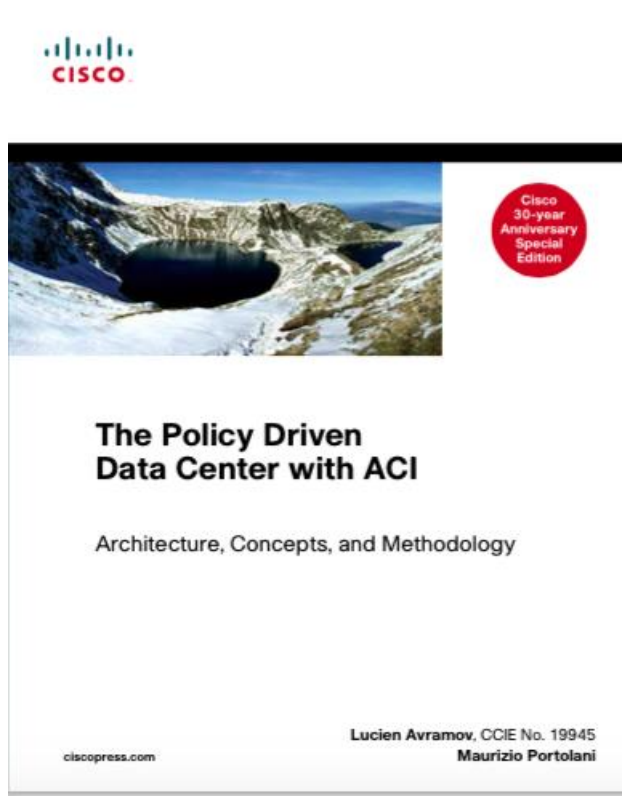
ACI Virtual Leaf  
(AVS)

ACI Policy and Automation  
Extended to Virtual Servers  
via Cisco AVS

Extending ACI Policy and Automation into the Existing Data Centre



# Recommended Readings





# Call to Action

- Visit the World of Solutions for
  - Cisco Campus – (speaker to add relevant demos/areas to visit)
  - Walk in Labs – (speaker to add relevant walk in labs)
  - Technical Solution Clinics
- Meet the Engineer (Speaker to specify when they will be available for meetings)
- Lunch time Table Topics
- DevNet zone related labs and sessions
- Recommended Reading: for reading material and further resources for this session, please visit [www.pearson-books.com/CLMilan2015](http://www.pearson-books.com/CLMilan2015)

A long-exposure photograph of a city street at night. The foreground is filled with vibrant, multi-colored light trails from moving vehicles, creating a sense of motion. In the background, a modern pedestrian bridge with blue lighting spans the street. Tall buildings with illuminated windows and storefronts line the street, and several flags are visible on the left. The overall scene is a dynamic urban nightscape.

Q & A

# Complete Your Online Session Evaluation

## Give us your feedback and receive a Cisco Live 2015 T-Shirt!

Complete your Overall Event Survey and 5 Session Evaluations.

- Directly from your mobile device on the Cisco Live Mobile App
- By visiting the Cisco Live Mobile Site  
<http://showcase.genie-connect.com/clmelbourne2015>
- Visit any Cisco Live Internet Station located throughout the venue

T-Shirts can be collected in the World of Solutions on Friday 20 March 12:00pm - 2:00pm



### Learn online with Cisco Live!

Visit us online after the conference for full access to session videos and presentations. [www.CiscoLiveAPAC.com](http://www.CiscoLiveAPAC.com)

**Cisco** *live!*



Thank you.



**CISCO**