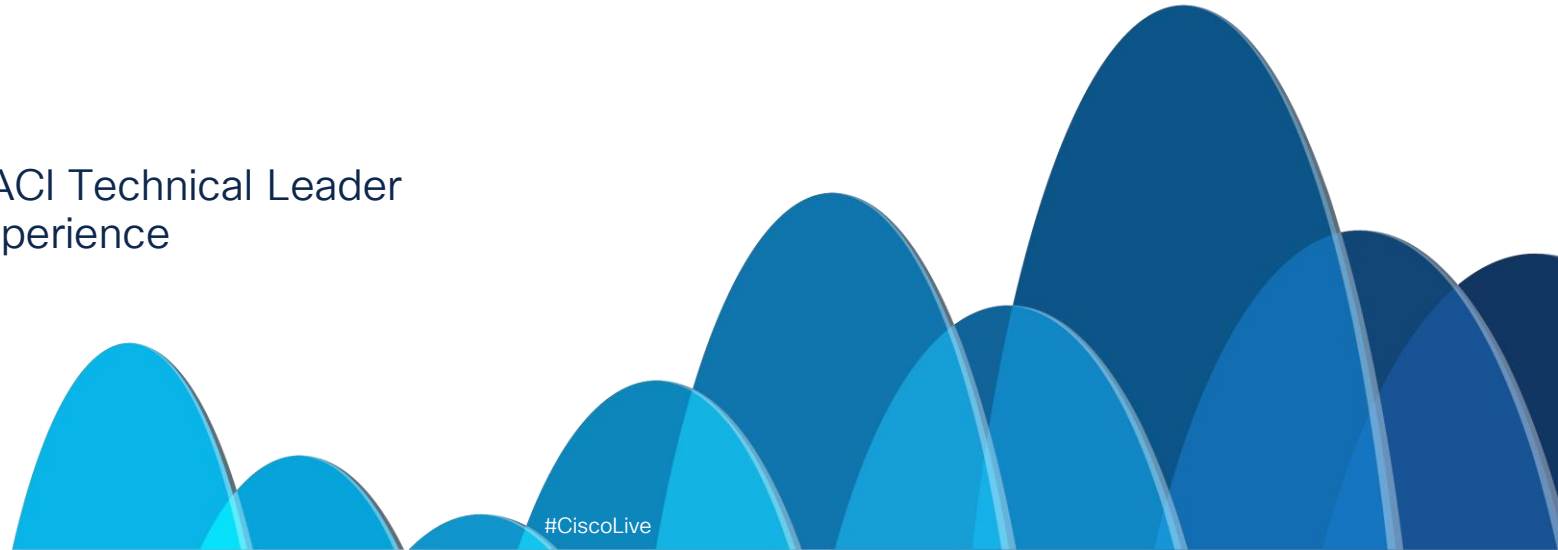
The bridge to possible

# A Network Engineer's Blueprint for ACI Forwarding

Joe Young, ACI Technical Leader
Customer Experience

BRKDCN-3900



cisco Live!

#CiscoLive

# Cisco Webex App

https://ciscolive.ciscoevents.com/ciscolivebot/#BRKDCN-3900

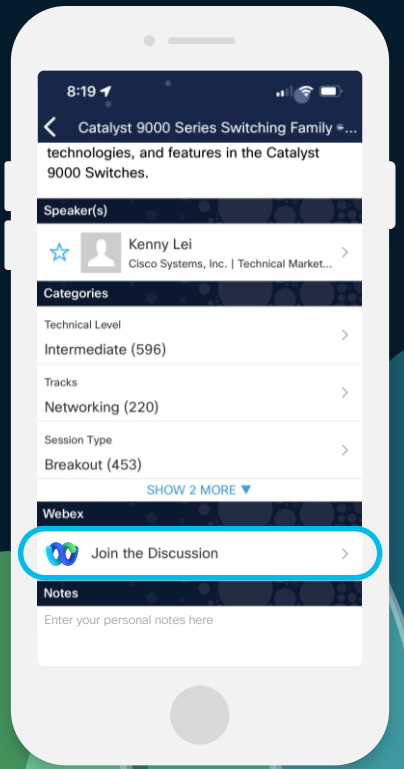
## Questions?

Use Cisco Webex App to chat with the speaker after the session

## How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

## Webex spaces will be moderated by the speaker until June 7, 2024.



# Agenda

- What's Different About ACI Forwarding?
  - (iVXLAN, contracts, endpoint learning)
- Proxy Forwarding
- ACI Forwarding Tables
  - Endpoint tables, routing tables, hardware lookups
- Understanding the Configuration Options
- The Anatomy of an ACI Switch

# Agenda

- Understanding the Tools
  - UI Tools
  - Elam
  - Ftriage
  - Span / ERSPAN
  - Flow Telemetry / netflow
- Debugging and Walking Through ACI Flows
  - (Routed, Bridged, BUM, Proxied)

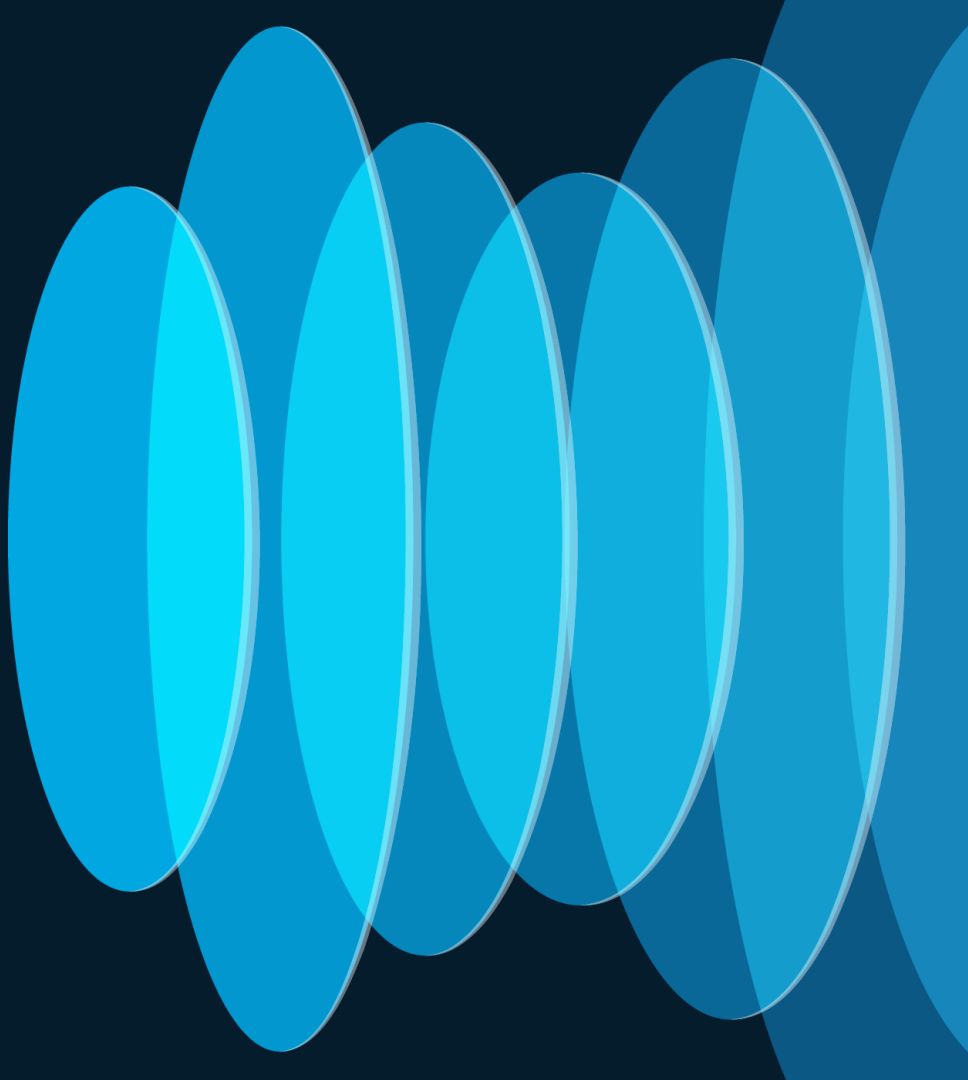
# Glossary of Acronymns

| Acronyms | Definitions |
|---|---|
| ACI | Application Centric Infrastructure |
| APIC | Application Policy Infrastructure Controller |
| EP | Endpoint |
| EPG | Endpoint Group |
| BD | Bridge Domain |
| VRF | Virtual Routing and Forwarding |
| COOP | Council of Oracle Protocol |
| VxLAN | Virtual eXtensible LAN |

VxLAN packet acronyms

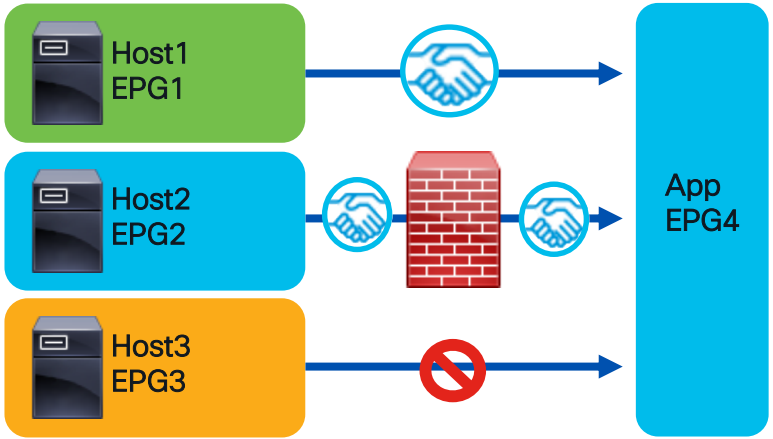
| Acronyms | Definitions |
|---|---|
| d*XXX*o | Outer Destination XXX<br>(dIPo = Outer Destination IP) |
| s*XXX*o | Outer Source XXX<br>(sIPo = Outer Source IP) |
| d*XXX*i | Inner Destination XXX<br>(dIPi = Inner Destination IP) |
| s*XXX*i | Inner Source XXX<br>(sIPi = Inner Source IP) |
| GIPo | Outer Multicast Group IP |
| VNID | Virtual Network Identifier |

# What's Different About ACI Forwarding?



# What is “Application Centric”?

- Traditional networks use ACL’s to classify traffic
  - Usually based on L3 or L2 addresses
  - Makes security decisions (permit, deny, log, etc)
  - Makes forwarding decisions (policy based routing)
- ACI can classify traffic based on its EPG or ESG
- Traffic inherits the forwarding and security policy of the EPG or ESG



# How is “Application Centric” Achieved?

## Sources and Destinations Must be Classified into EPG’s

### Endpoints

- Used by App EPG’s and ESG’s
- Represents the network identity of an end device
- Learned dynamically or configured statically

### Policy-Prefixes

- Used by External EPG’s
- Classifies destination by longest prefix match
- Also used for shared-services
- Configured

### PcTags

- The security ID of an EPG / ESG
- Used in contracts. Ex: Permit PcTag 1000 to PcTag 2000
- Sclass/dclass imply PcTag direction

### Contracts

- Defines security and sometimes forwarding (pbr) policy between epgs
- Essentially an ACL between PcTags
- Consumer/Provider rather than src/dest

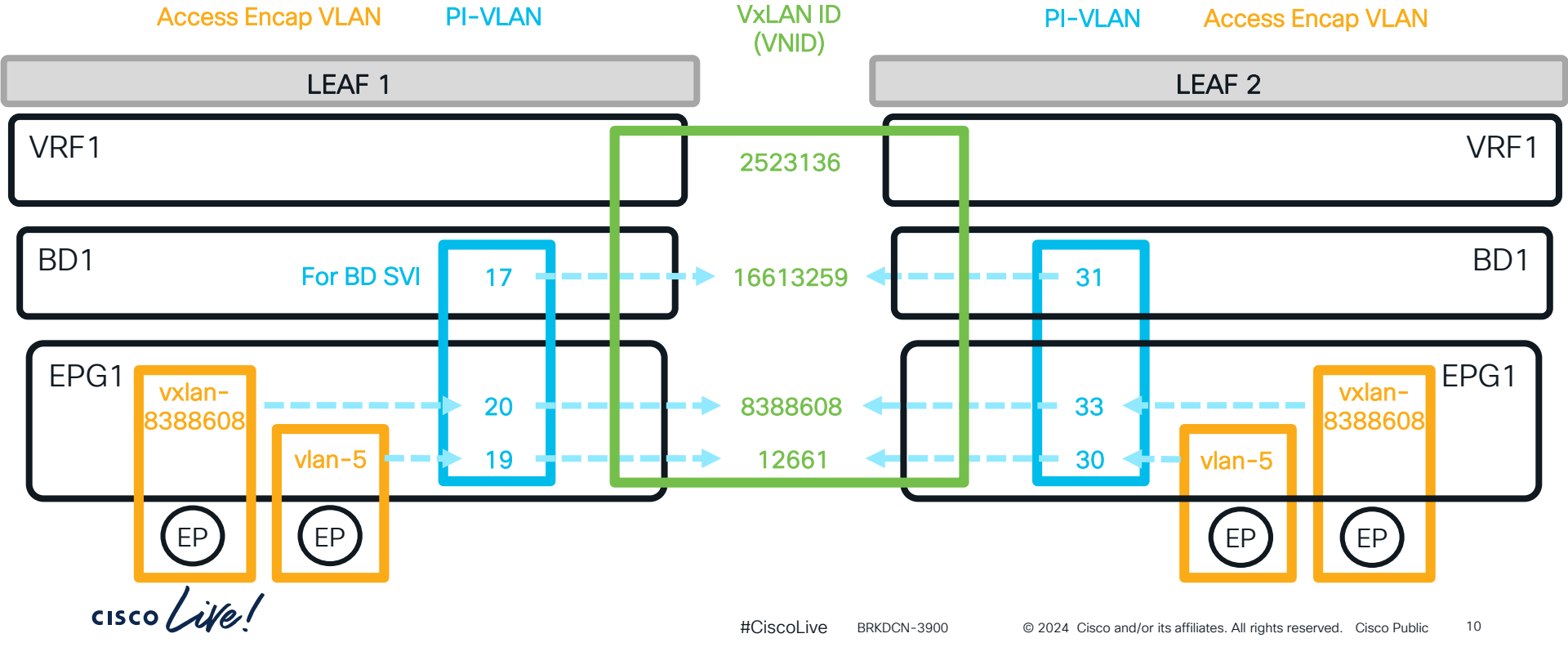
# Vlan Types

※ PI-VLAN : Platform Independent VLAN

VLAN ID for external devices (user configured value)

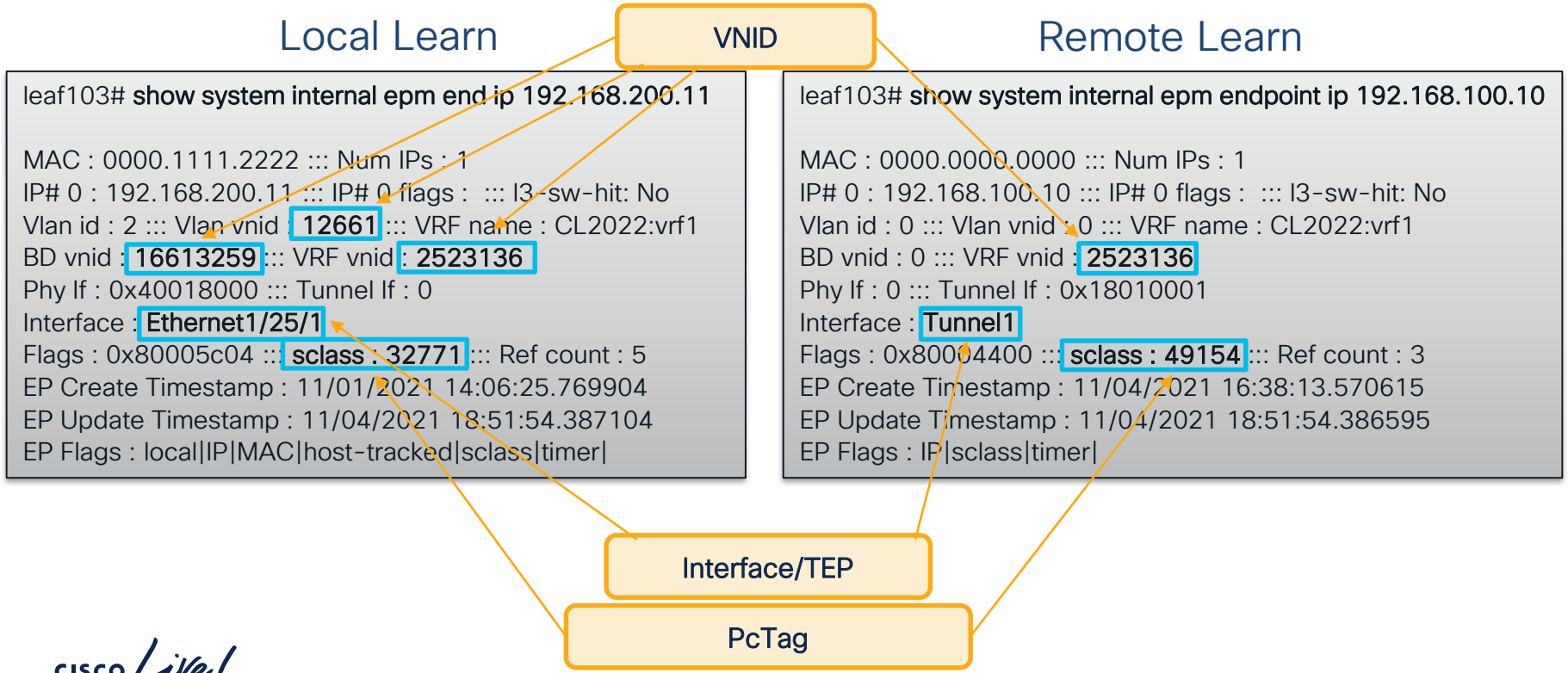
Internal ID on LEAF (not shared across LEAFs)

For forwarding (global value for entire fabric)



# What is an Endpoint?

## An Endpoint joins both forwarding and security policy



# What is a TEP? (Tunnel Endpoint)

- IP addresses allocated for overlay communication
- VXLAN Traffic is sent to the TEP + VNID of destination

**Most Common TEP Types**

| TEP Type | What is it? | What is it for? |
|---|---|---|
| Physical TEP (PTEP) | Unique Overlay IP Address for each individual Leaf/Spine | Non-vpc dataplane, l3out communication, apic-leaf comm, etc |
| VPC TEP (VTEP) | Unique Overlay IP Address for each VPC Pair | Traffic destined to endpoints that are connected behind VPC |
| Proxy TEP | Spine Anycast IP's used for proxy traffic | Leafs send to these TEPs when doing proxy forwarding |

```
a-leaf101# show ip interface loopback0
IP Interface Status for VRF "overlay-1"
lo0, Interface status: protocol-up/link-up/admin-up, iod: 4, mode: ptep
```

# What are Tunnels?

- Leafs/Spines Install Tunnel Interface to each known TEP.
- Used for VXLAN Dataplane

How are Tunnels Learned?

Dataplane Learns

```
leaf# moquery -c tunnelIf -f 'tunnel.If.id=="tunnel1"'

id               : tunnel1
dest             : 10.0.72.67
idRequestorDn    : sys/*/db-dtep/dtep-[10.0.72.67]
```

Through BGP (l3out routes)

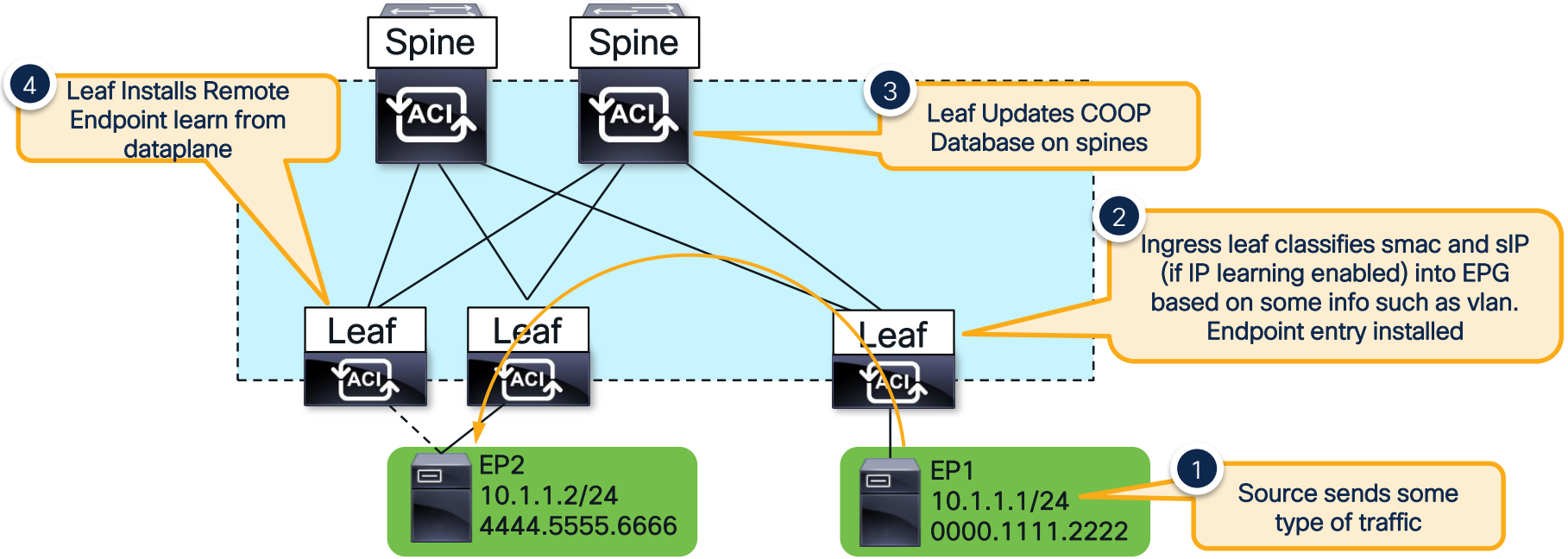
```
leaf# moquery -c tunnelIf -f 'tunnel.If.id=="tunnel1"'

id               : tunnel1
dest             : 10.0.72.64
idRequestorDn    : sys/bgp/*/db-dtep/dtep-[10.0.72.64]
```

Local POD ISIS Database

```
leaf# moquery -c tunnelIf -f 'tunnel.If.id=="tunnel1"'

# tunnel.If
id               : tunnel1
dest             : 10.0.152.64
idRequestorDn    : sys/isis/*/lvl-l1/db-dtep/dtep-[10.0.152.64]
```

# How is an Endpoint Learned?

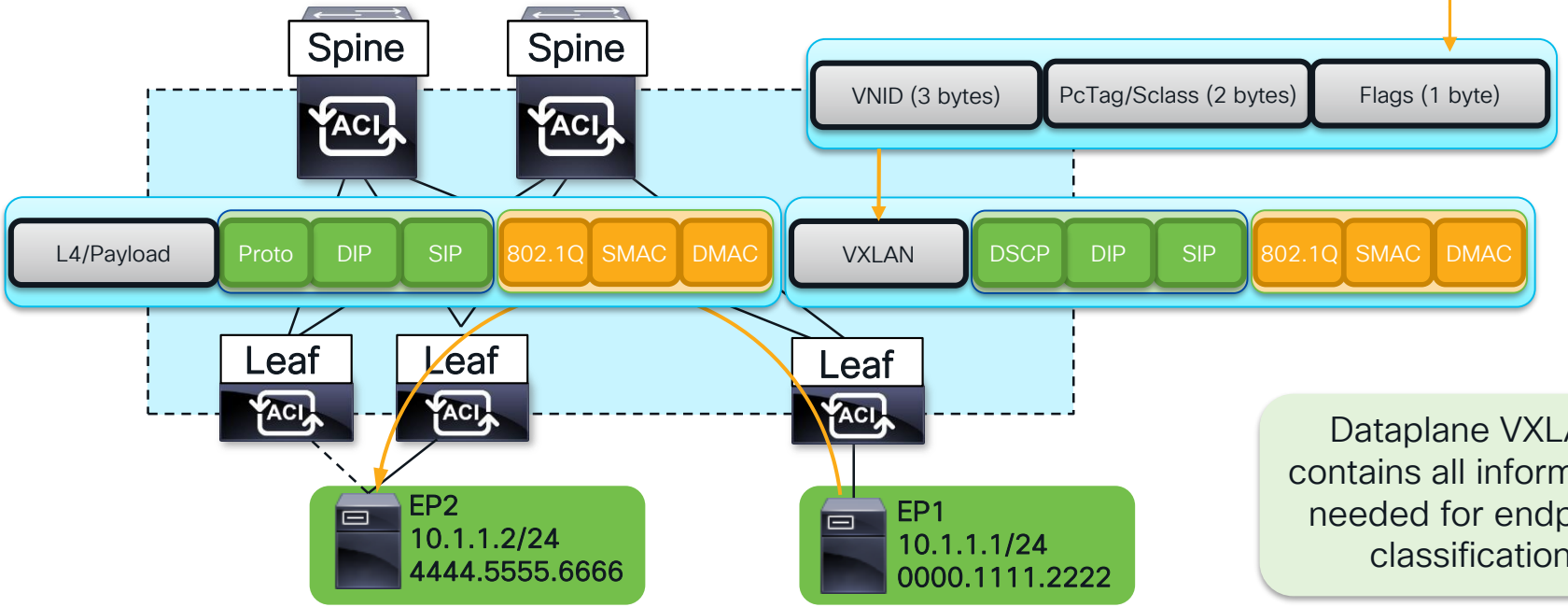
How does the Egress leaf classify traffic into the correct EPG?



# Overlay iVXLAN

## ACI uses VXLAN with some additional bits

Bit pos 4 – Source Policy Applied
Bit pos 5 – Destination Policy Applied
Bit pos 7 – Don't learn




Dataplane VXLAN contains all information needed for endpoint classification

# How is Traffic Classified with no EP Learn?

## In most of these cases, the pcTag is based on a policy-prefix lookup

There will be no endpoint learn in several cases

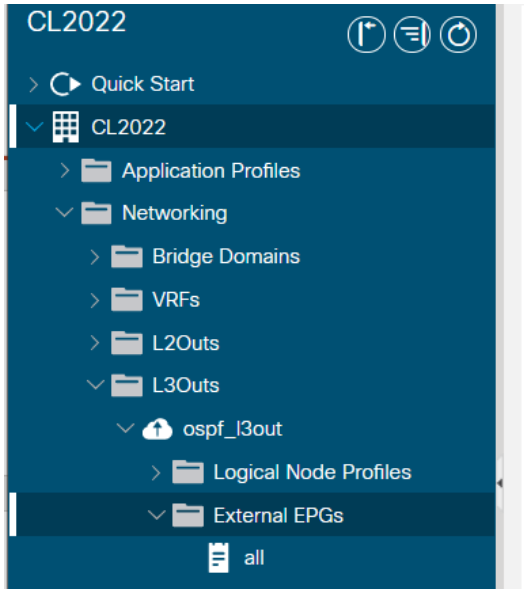
- Source/dest is behind an l3out
- Source/dest is in another vrf
- Endpoint learning is disabled by some option

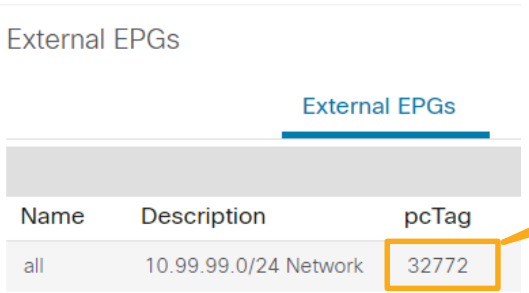
If ingress leaf doesn't apply policy, egress leaf should (indicated via policy-applied bits in ivxlan header)

# How is Traffic Classified with no EP Learn?

## Destination Behind L3out

```
leaf101# vsh_lc -c "show forwarding route 10.99.99.100 platform vrf CL2022:vrf1"
!
Policy Prefix 10.99.99.0/24
!
vrf: 16(0x10), routed_if: 0x0 epc_class: 32772(0x8004)
```




Classification based on longest l3out policy prefix

# How is Traffic Classified with no EP Learn?

## Destination is unknown and is proxied

```
leaf101# show ip route 192.168.200.20 vrf CL2022:vrf1

192.168.200.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 10.0.176.66%overlay-1, [1/0], 4d05h, static, tag 4294967294
        recursive next hop: 10.0.176.66/32%overlay-1

leaf101# vsh_lc -c "show forwarding route 192.168.200.20 platform vrf CL2022:vrf1"
!
Policy Prefix 0.0.0.0/0
!
Vrf: 16(0x10), routed_if: 0x0 epc_class: 1(0x1)
```

"Pervasive" indicates this is a BD or EPG subnet (fvSubnet). Send to spine proxy-addr

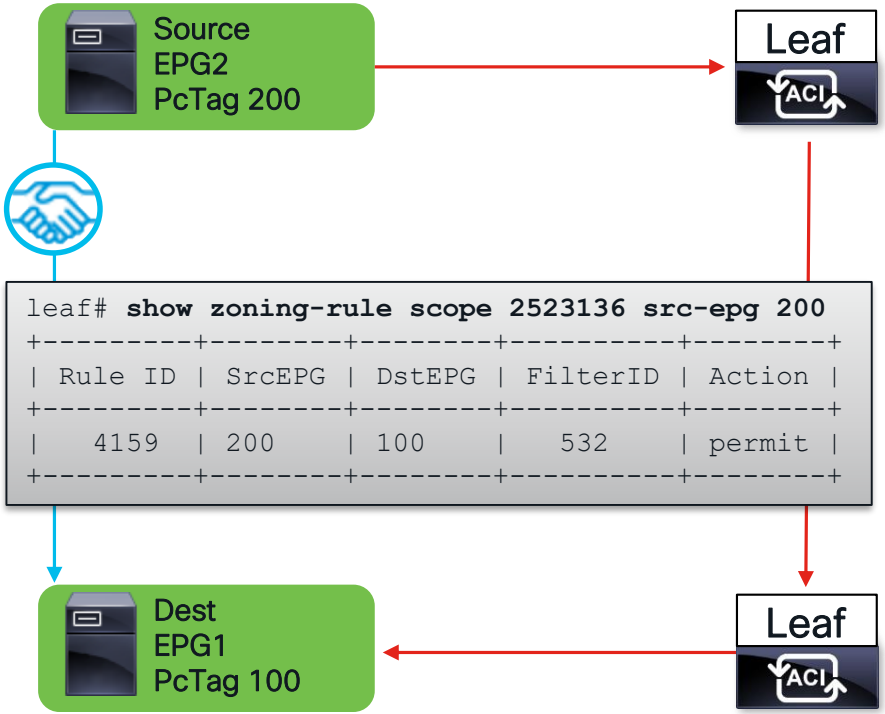
-pcTag of 1 indicates the fabric owns the subnet, don't apply policy
-policy applied flags not set in ivxlan header

## Don't apply policy, Forward to proxy Anycast!

```
leaf101# show isis dtep vrf overlay-1 | egrep "Type|PROXY"
DTEP-Address      Role    Encapsulation   Type
10.0.176.66       SPINE   N/A             PHYSICAL,PROXY-ACAST-V4
10.0.176.65       SPINE   N/A             PHYSICAL,PROXY-ACAST-MAC
10.0.176.64       SPINE   N/A             PHYSICAL,PROXY-ACAST-V6
```

Check hidden slide for impact of “Policy Control Enforcement Direction” setting

# Contracts and Forwarding




Ingress

Contract Found?

Yes → Set policy-applied bits in ivxlan. Permit, deny, redir, log

No → If LPM is BD/EPG subnet, forward and don’t set policy-applied bits in ivxlan. Otherwise, drop!

Egress

Policy-Applied Bits set?

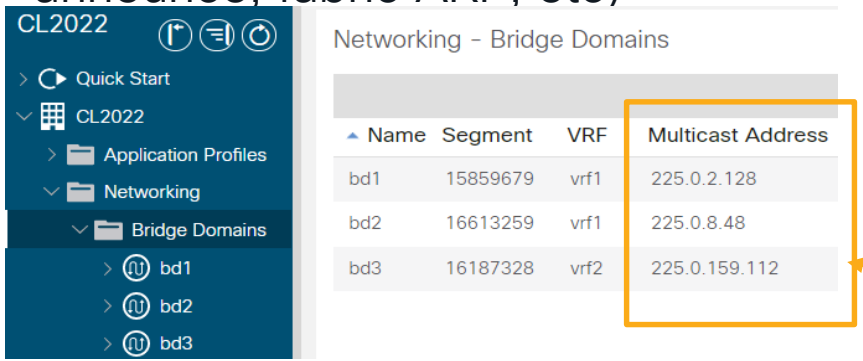
Yes → Don’t do contract lookup. Forward.

No → Do contract lookup. Permit, deny, redir, log

# What About Flooded Traffic?

The following traffic may be flooded:

- Broadcast
- Multicast
- Unknown Unicast
- Control Plane maintenance (EP announce, fabric ARP, etc)




How does ACI flood?

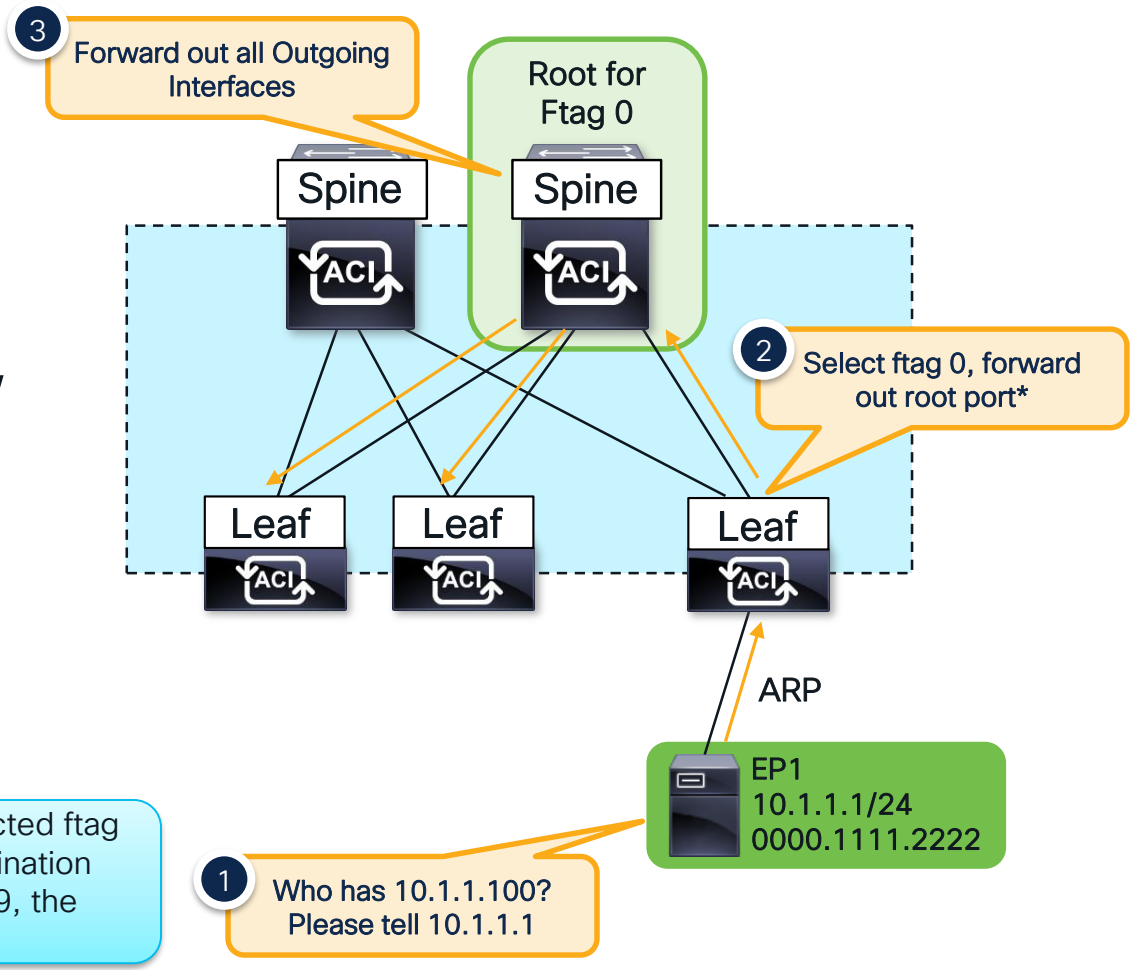
- Flooded traffic is sent to the BD GiPo (I2 flood) or VRF GiPo (I3 flood)
- The GiPo is an overlay multicast address allocated to a BD or VRF
- Flooding is done on a loop-free tree called an FTAG

Security policy NOT applied

GiPo

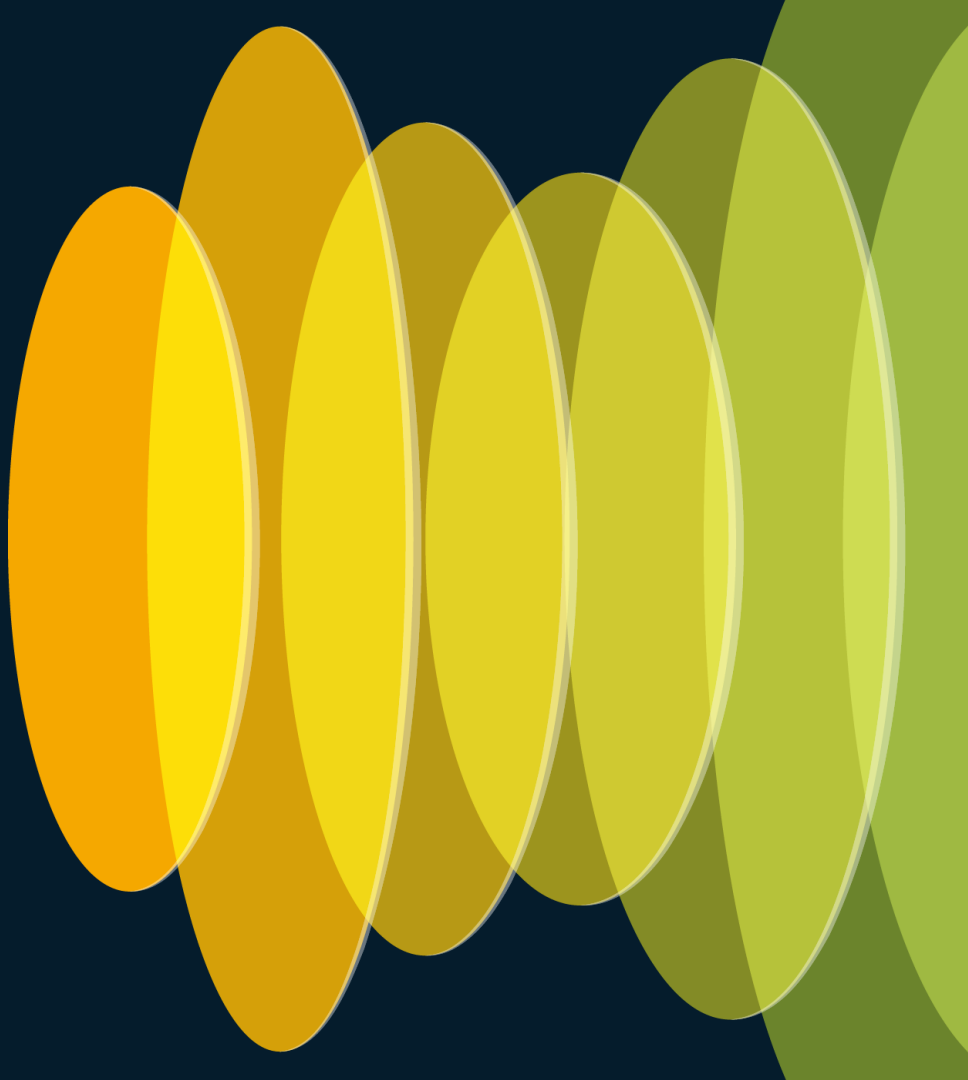
# What are FTAGs?

- FTAGs are loop-free trees within the overlay used by flooded traffic
- FTAGs are picked per flow from values 0 – 0xc
- One spine is root for each tree
- Outgoing interfaces calculated by ISIS



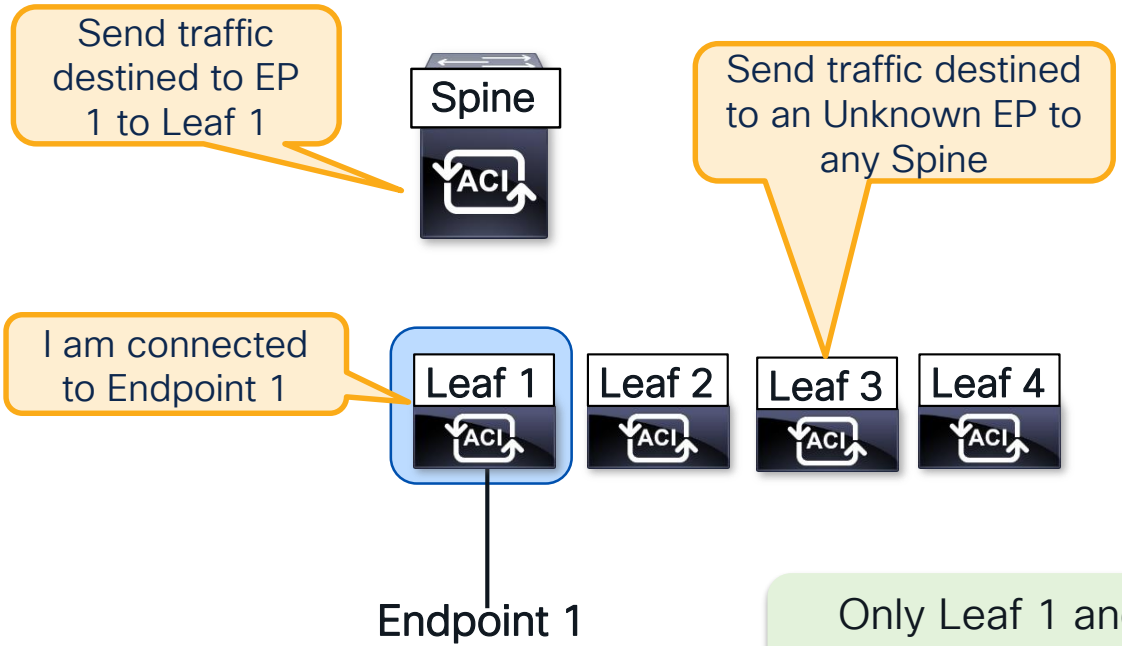

*Note, the ingress leaf communicates the selected ftag to the rest of the fabric by adding it to the destination gipo. If the gipo is 225.0.0.0 and the ftag is 0x9, the destination address would be 225.0.0.9

# Proxy Forwarding



# What is Proxy Forwarding?

## Why? Scaling out Endpoint Learning




Spines own separate anycast TEP's for mac, ipv4, and ipv6 proxy lookups

Only Leaf 1 and Spines have to program Endpoint 1 in hardware

# How to check the Spine-Proxy TEP

```
leaf1# show ip route vrf CL2022:vrf1

192.168.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 10.0.16.64%overlay-1, [1/0], 00:21:39, static
```

BD Subnet (Pervasive Route)

next-hop should be SPINE-PROXY

```
leaf1# show isis dteps vrf overlay-1 | grep PROXY
10.0.16.65          SPINE    N/A                PHYSICAL,PROXY-ACAST-MAC
10.0.16.64          SPINE    N/A                PHYSICAL,PROXY-ACAST-V4
10.0.16.67          SPINE    N/A                PHYSICAL,PROXY-ACAST-V6
```

next-hop of Pervasive Route is IPv4 Spine Proxy TEP

## Three types of Spine Proxy TEP

- Proxy-Acast-MAC
  - ✓ Spine-Proxy for L2 traffic (L2 Unknown Unicast mode “Hardware Proxy”)
- Proxy-Acast-V4
  - ✓ Spine-Proxy for IPv4 traffic (includes ARP Request with ARP Flooding mode “OFF”)
- Proxy-Acast-V6
  - ✓ Spine-Proxy for IPv6 traffic

# What is COOP?

## COOP is the proxy-database of ACI

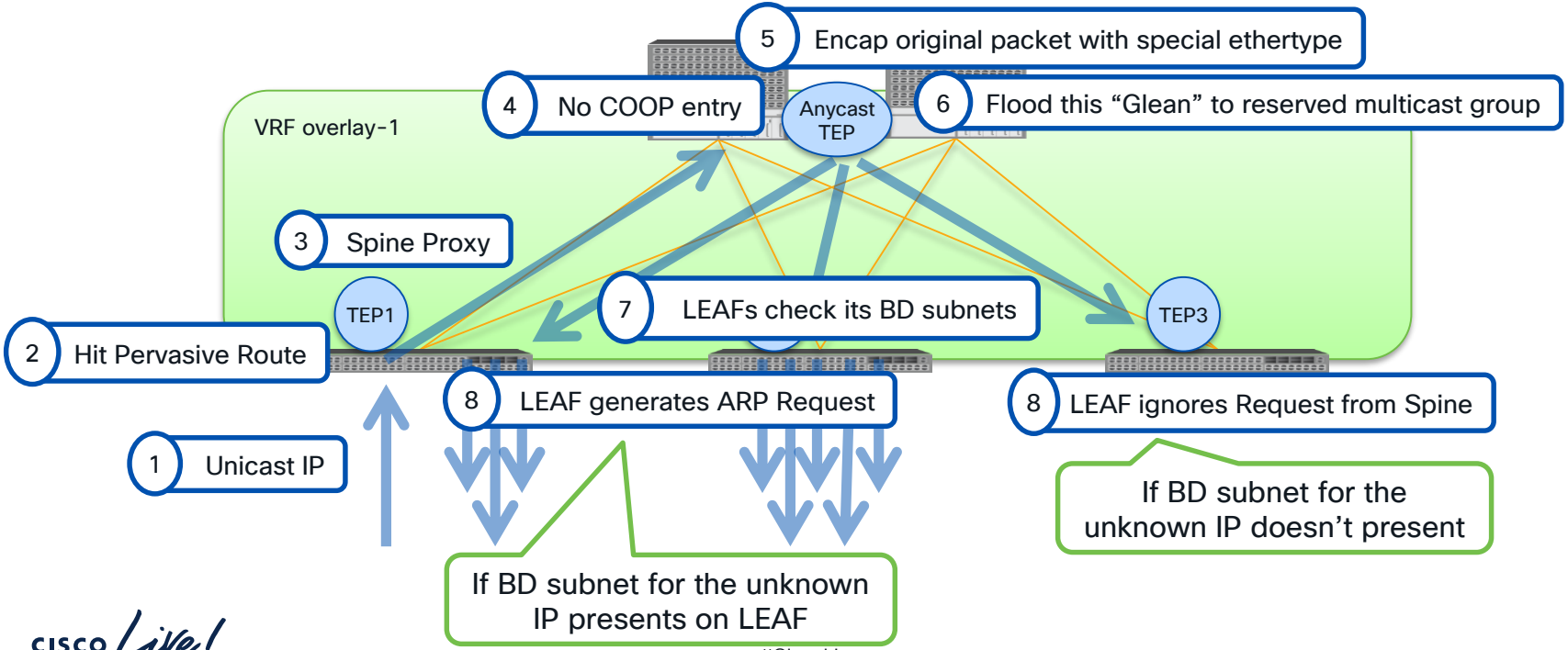
- Council of Oracles Protocol – A TCP protocol for citizens (Leafs) to publish records to oracles (Spines).
- Used for announcing endpoints, fabric owned IP's, multicast information, and more
- Synced across Pods/Sites with BGP EVPN
- Each Endpoint Record contains all information to forward (VNID, leaf TEP, mac, etc)
- COOP records pushed into hardware on spines
- For modular spines, scale is achieved by pushing each EP onto only two Fabric Modules

# What if the Endpoint isn't in COOP? (ARP Glean)

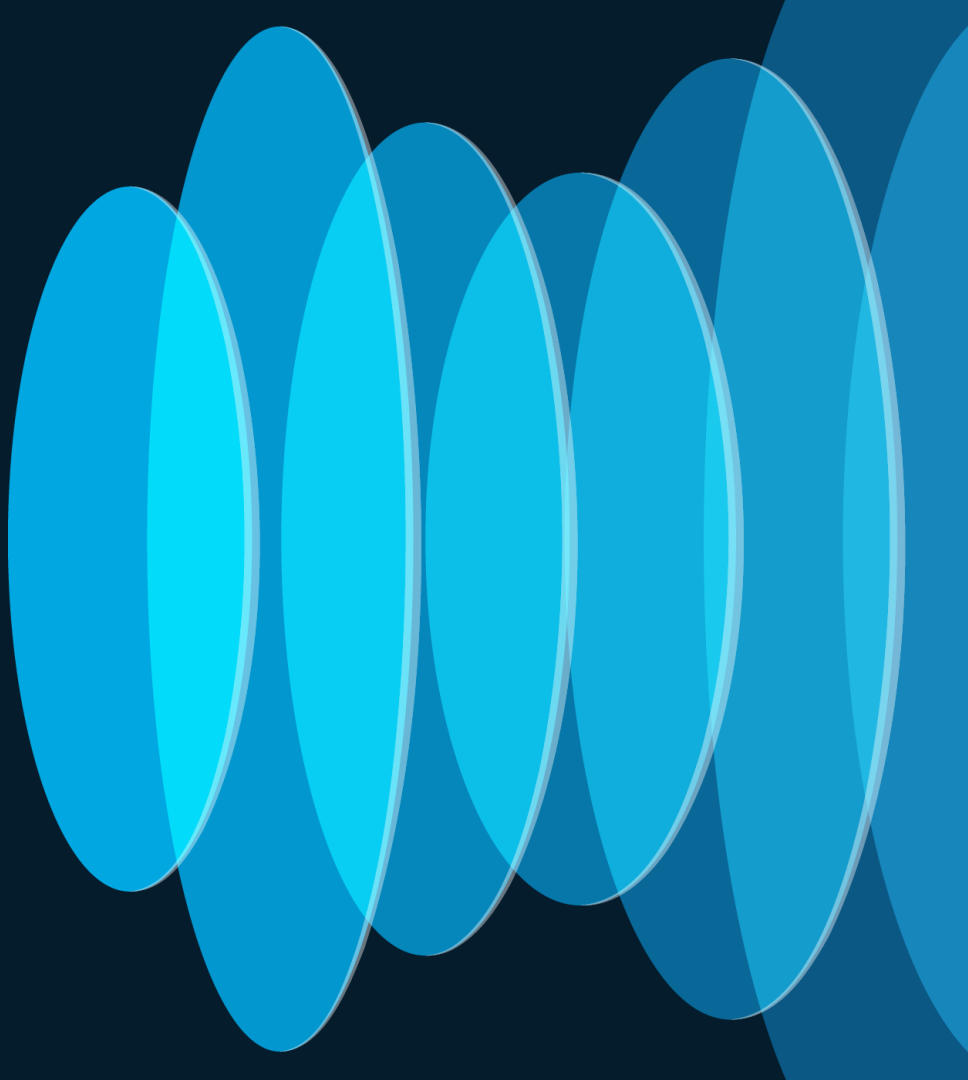
What if Spine's COOP DB doesn't know the destination when proxy'ed?

X L2 Traffic : Drop

✓ L3 Traffic : ARP Glean



# How ACI Builds Forwarding Tables



# Building Adjacency Tables

## ACI combines ARP and MAC Tables into the Endpoint Table

### Legacy Behavior

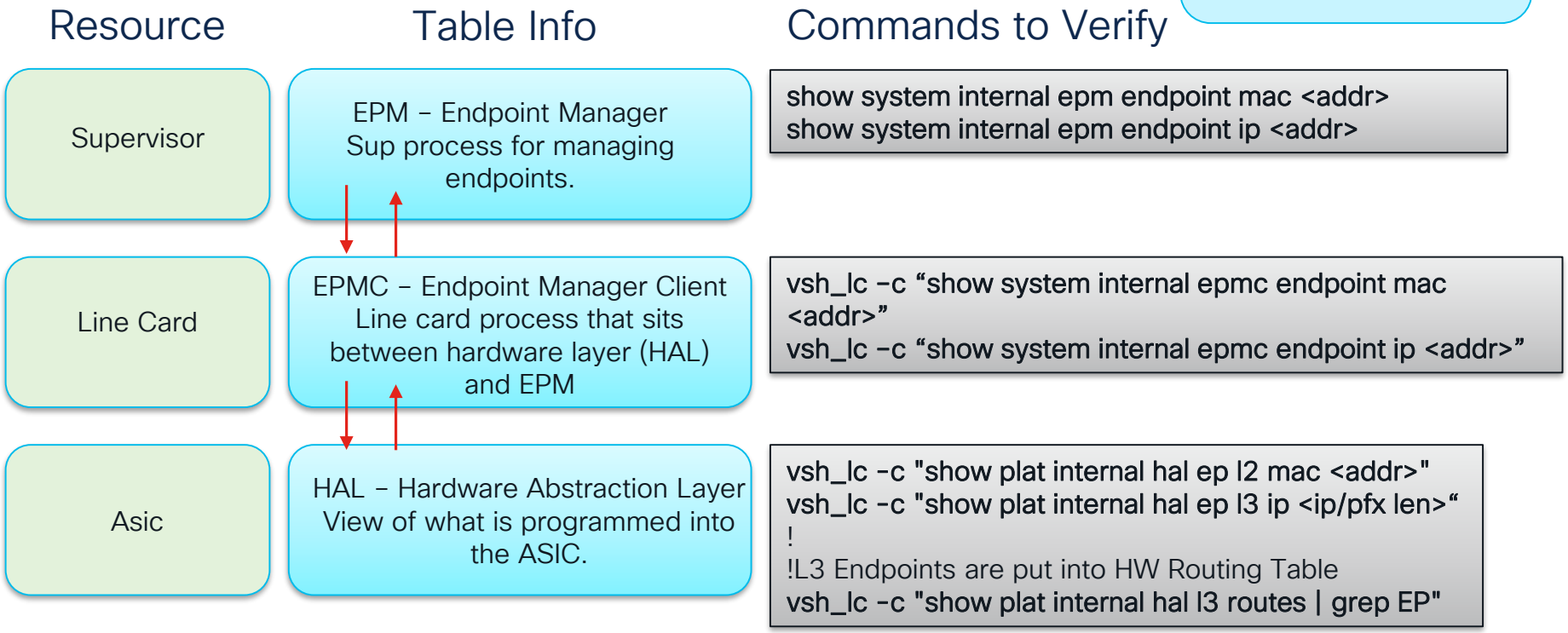
- ARP/ND tables map Layer 3 to Layer 2
- ARP/ND tables are updated by control-plane messages
- MAC Address Table used for switching decisions
- Mac Address Table updated by dataplane

### ACI Behavior

- Endpoint table contains endpoints, which are Layer 2 addresses OR Layer 3 addresses OR a combination of Layer 2 and Layer 3 addresses
- By default, both Layer 2 and Layer 3 information is updated by dataplane
- Used for security and forwarding policy

# Building Endpoint Tables

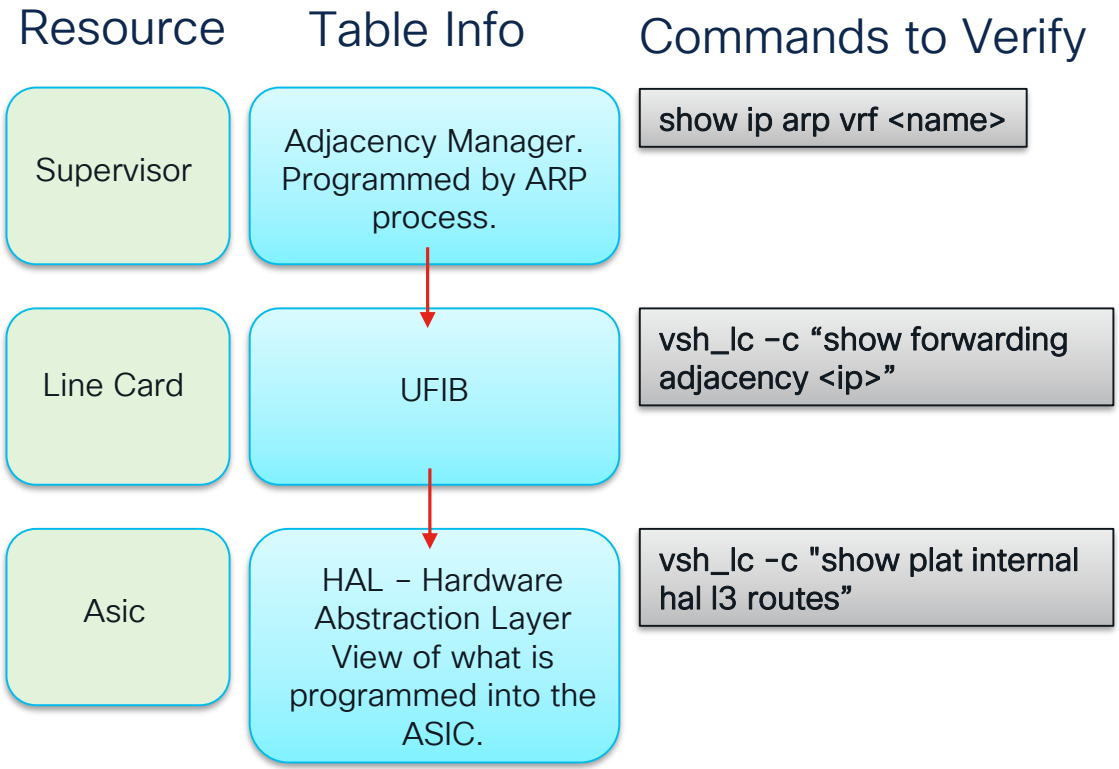
Endpoints can be programmed via software process or by hardware dataplane learns (HAL)



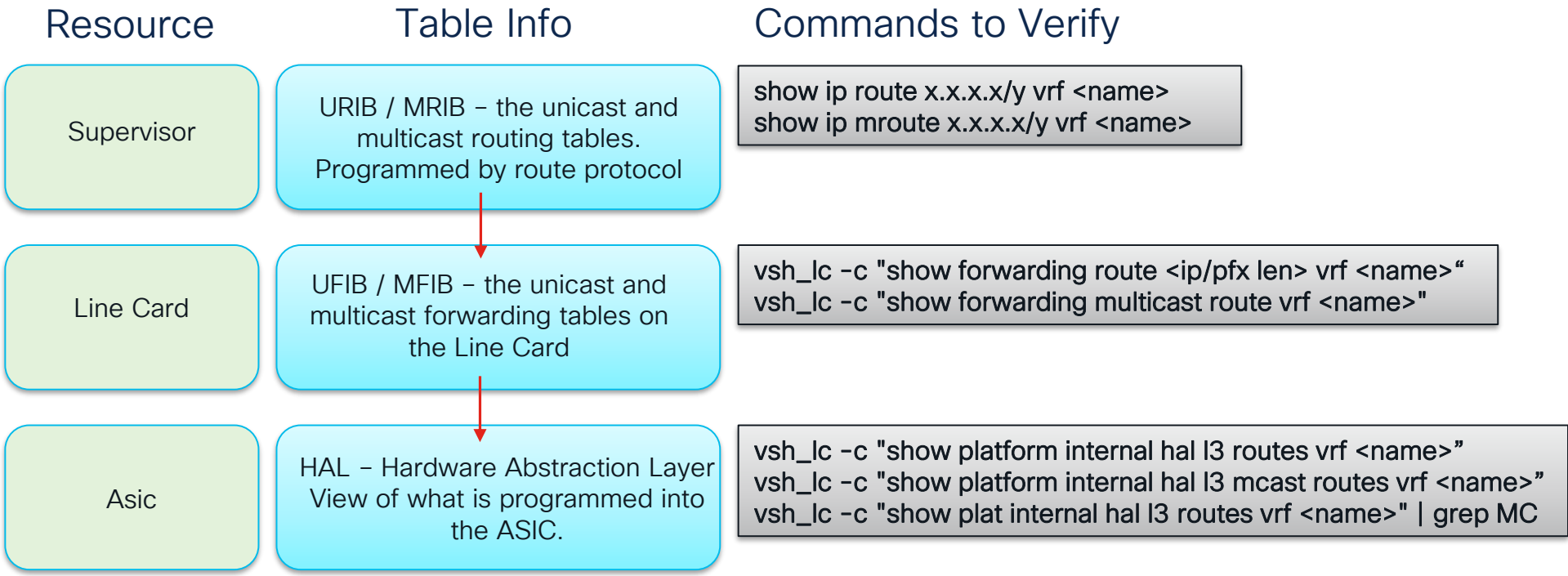
# What about ARP?

## ARP Tables are still used in ACI for...

- L3outs
- Overlay adjacencies
  - VXLAN Endpoints (AVE, K8s, Openstack, etc)
  - APIC / Fabric node adjacencies



# Building Routing Tables



Check Endpoint Table before Routing Table

# Troubleshooting TIP

## When Troubleshooting Layer 3 Flows Always...

1) Check if there is an Endpoint Learn

show endpoint ip <addr>
show system internal epm endpoint ip <addr>

If not then...

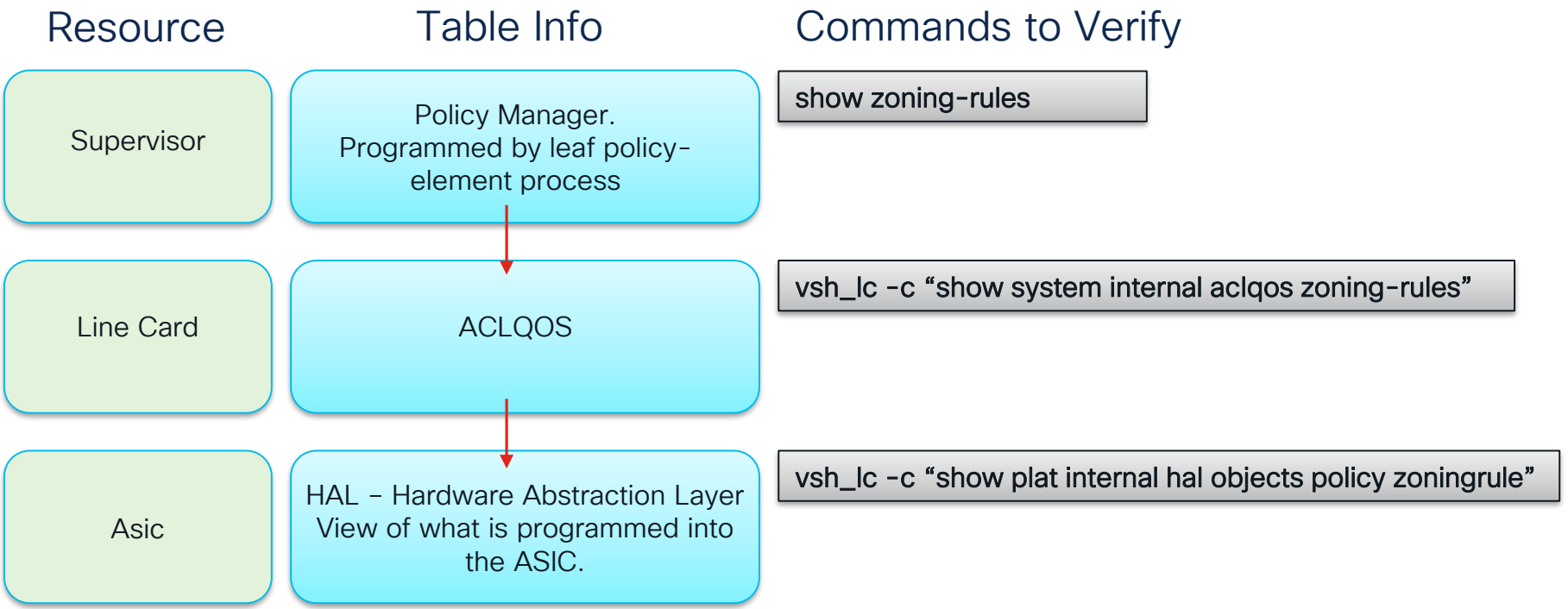
2) Check if there is a BD (pervasive) static route

If not then...

3) Check if there is an External Route

show ip route x.x.x.x/y vrf <name>

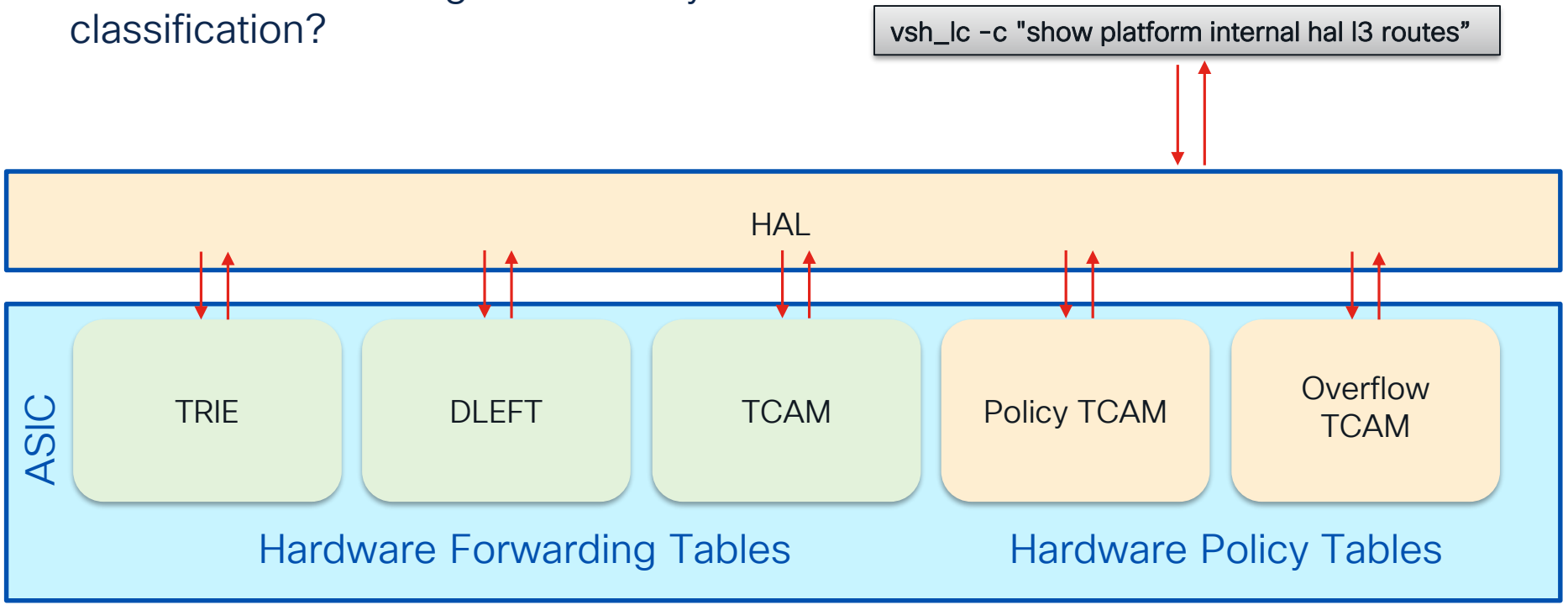
# Programming Contracts



# HAL – Hardware Abstraction Layer

Applicable to EX and Later Hardware

Wouldn't it be great if there was a single point to validate forwarding and security classification?



# HAL – Hardware Abstraction Layer

Applicable to EX and Later Hardware

## L3 Lookup of Hardware Tables

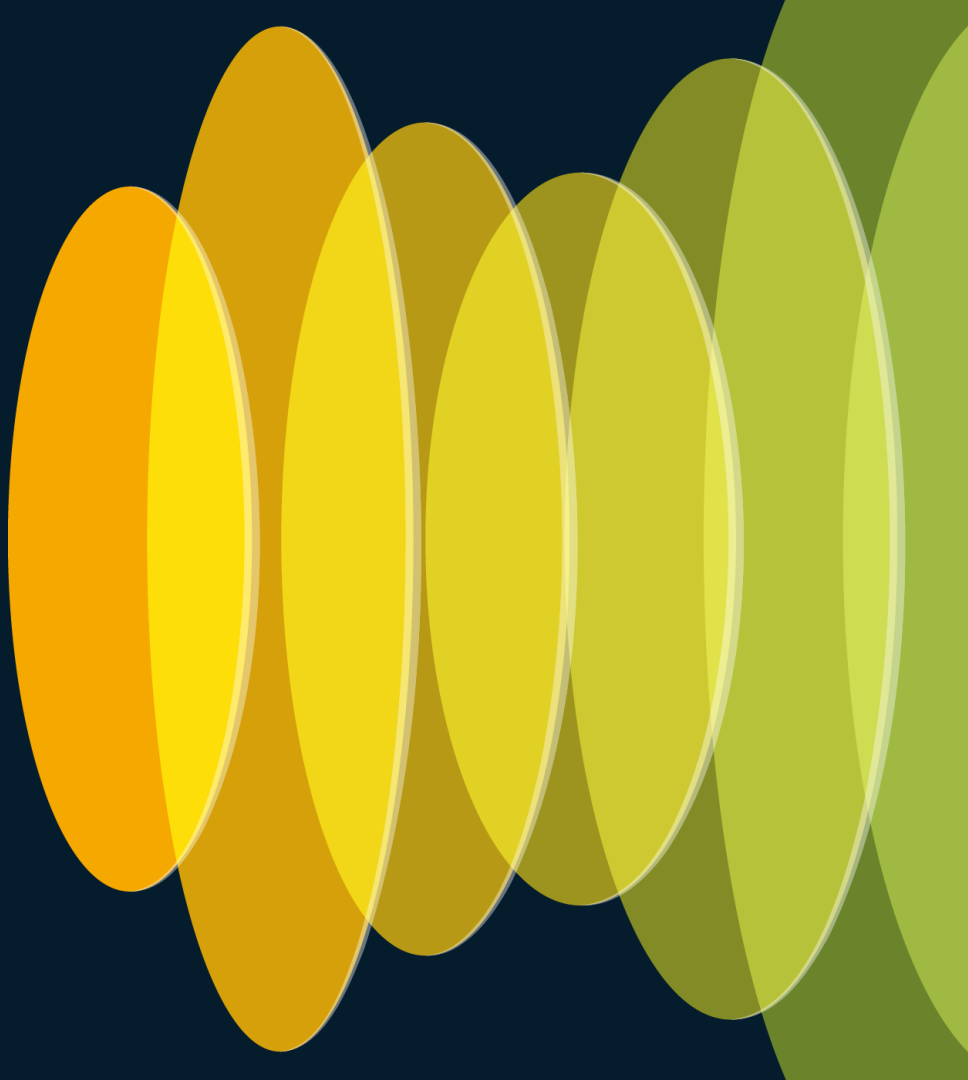
```
module-1# show plat internal hal l3 routes vrf CL2022:vrf1
-----------------------------------!!-------------------------
|     |                       |    | LID |!!                    |
| VRF | Prefix/Len            | RT | Type|!!|CLSS| Flags          |
|-----|-----------------------|----|-----|!!|----|----------------|
|-----|-----------------------|----|-----|!!|----|----------------|
| 4626|     192.168.100.10/ 32| EP | TRIE|!! c002 | le,bne,sne, dl |
| 4626|        10.99.99.0/ 24| UC | TCAM|!! 8004 | sc,spi,dpi     |
| 4626|    192.168.255.0/ 24 | UC | TCAM|!!   24 | sc,spi,dpi, dr |
| 4626|     192.168.200.11/ 32| EP | TRIE|!! 8003 | sc, le,sne     |
 -----------------------------------!!-------------------------
```

Consolidated view of routes for Endpoints, Shared Services, and External routes

PcTag from destination EPG…used for contract lookup

Much more info available in full output!

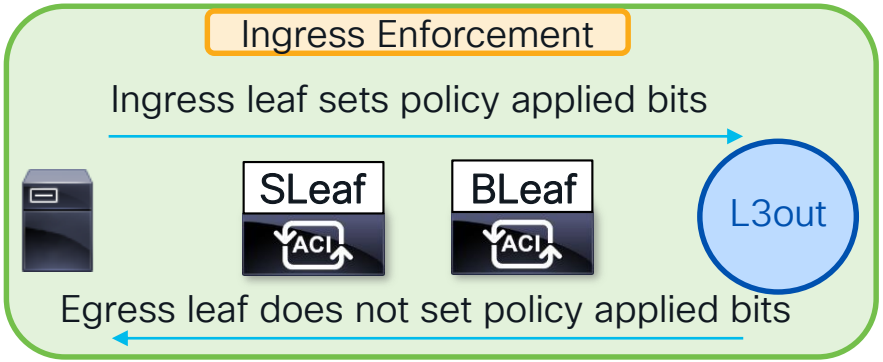
# Understanding the Configuration Options

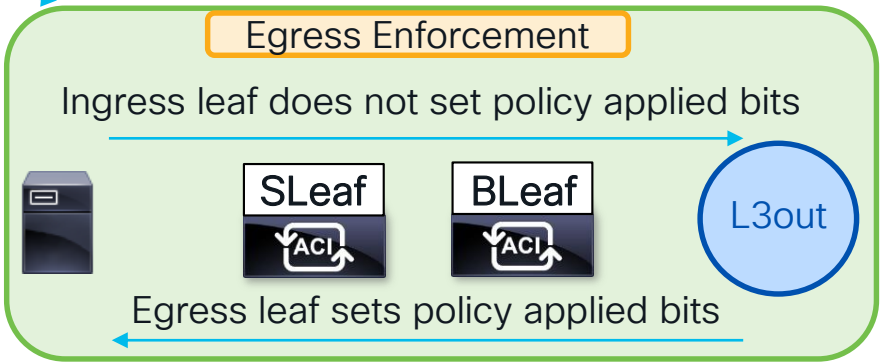


cisco Live!

# VRF Level Forwarding Options

| Feature | What Does it Do? |
|---|---|
| Policy Control Enforcement Preference | If disabled, policy is never applied between EPGs. If enabled, contracts are enforced. |
| IP Dataplane Learning | If Disabled, ACI uses legacy behavior for learning endpoints. Layer 3 endpoints are learned by ARP/GARP/ND and Layer 2 endpoints are learned by dataplane. |
| Policy Control Enforcement Direction | If set to Ingress, contract enforcement for l3out flows is done on service leaf. Egress enables enforcement on Border Leaf (requires remote learning to be enabled) |



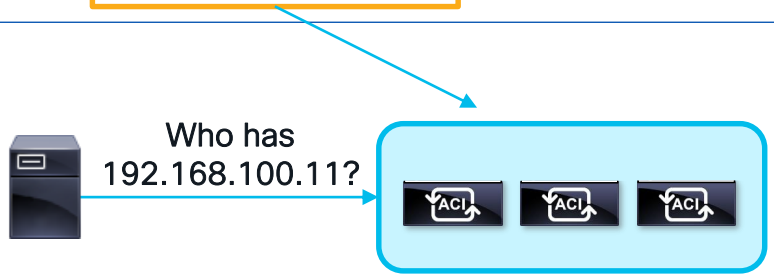
# Bridge-Domain Level Forwarding Options

| Feature | What Does it Do? |
|---|---|
| L3 Unknown Multicast Flooding | For non-link-local L3 multicast traffic in a PIM-disabled BD, should a leaf with no snooping entries flood in BD (flood) or wait for joins (OMF)? |
| Multidestination Flooding | For L2 mcast and broadcast, flood, drop, or flood within epg encap? If flooding with EPG encap, proxy-arp is required for cross-epg L2 communication |
| L2 Unknown Unicast | If destination mac is unicast and unknown, flood or proxy to spines? |

Proxied, L2 Unknown Unicast is dropped if the Destination MAC isn't known in COOP

# Bridge-Domain Level Forwarding Options

| Feature | What Does it Do? |
|---|---|
| Limit IP Learning to Subnet | Only learn IP's if they are within the configured BD subnet for local learns. |
| Unicast Routing | Enable IP learning as well as unicast routing (if a BD subnet is configured) |
| IP Data-plane Learning | Configured underneath the BD subnet. When disabled, IP/IPv6 learning is done via ARP/ND |
| ARP Flooding | When disabled, ARP is unicast routed based on the Target IP (if known) |




```
leaf# show endpoint ip 192.168.100.11
leaf# show ip route 192.168.100.11 vrf CL2022:vrf1

192.168.100.0/24, ubest/mbest: 1/0, direct, pervasive
    *via 10.0.176.66%overlay-1, [1/0], 01w00d, static
        recursive next hop: 10.0.176.66/32%overlay-1
```

Proxy!

# EPG Level Forwarding Options

| Feature | What Does it Do? |
|---|---|
| Flood in Encapsulation | Feature is enabled for just the EPG (rather than all epg's in the BD). Requires proxy arp for L2 traffic between encaps. |
| L4-L7 Virtual IP's | Designed for Direct Server Return flows. This disables dataplane learning per IP. IP is learned by ARP/ND. |
| Disable DP Learning Per-IP/Prefix | Disables dataplane learning. More specific than VRF-level option. In most cases should be used for DSR too. |

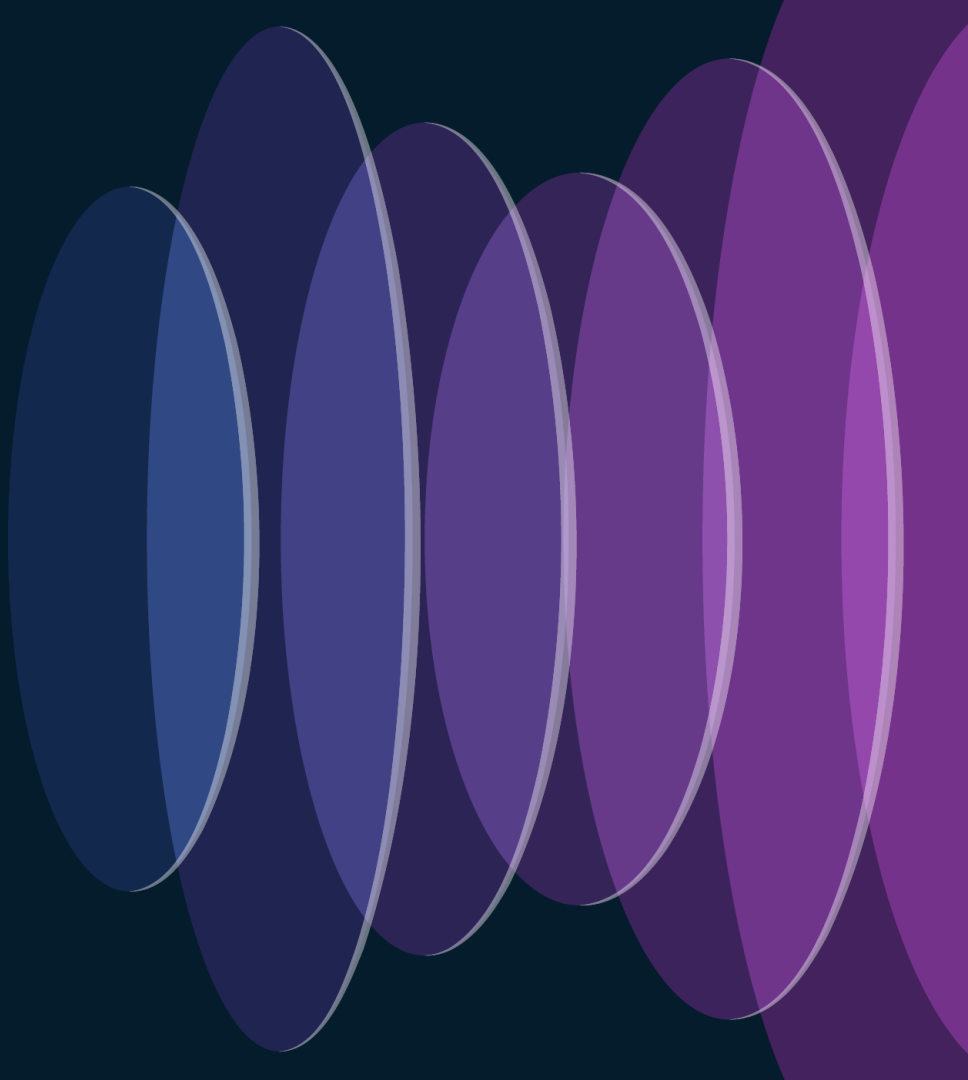
New in 5.2, can also be configured on BD

# Global Forwarding Options

| Feature | What Does it Do? |
| --- | --- |
| Enforce Subnet Check | Don't learn an IP (both local and remote) if it is not within a configured BD subnet in the VRF. |
| Disable Remote EP Learning on BL's | Remote IP learning is disabled for Unicast flows on a leaf in a specific VRF if an l3out exists in the same VRF |

- Multicast sources are still learned
- Also implicitly disabled when intersite l3out is configured

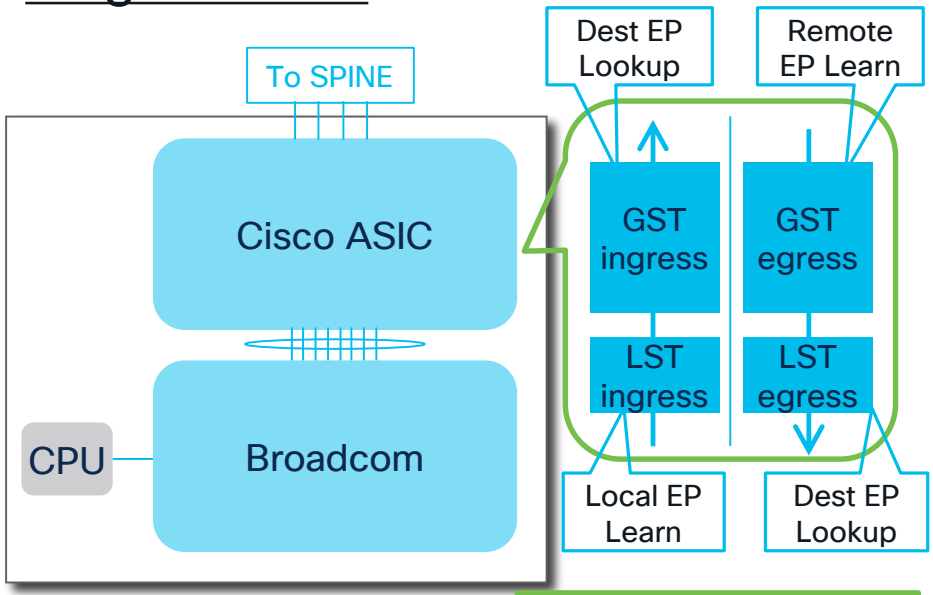
# The Anatomy of an ACI Switch



# LEAF ASIC Generations

※ LST: Local Station Table, GST: Global Station Table
※ FP Tile: Forwarding and Policy Tile

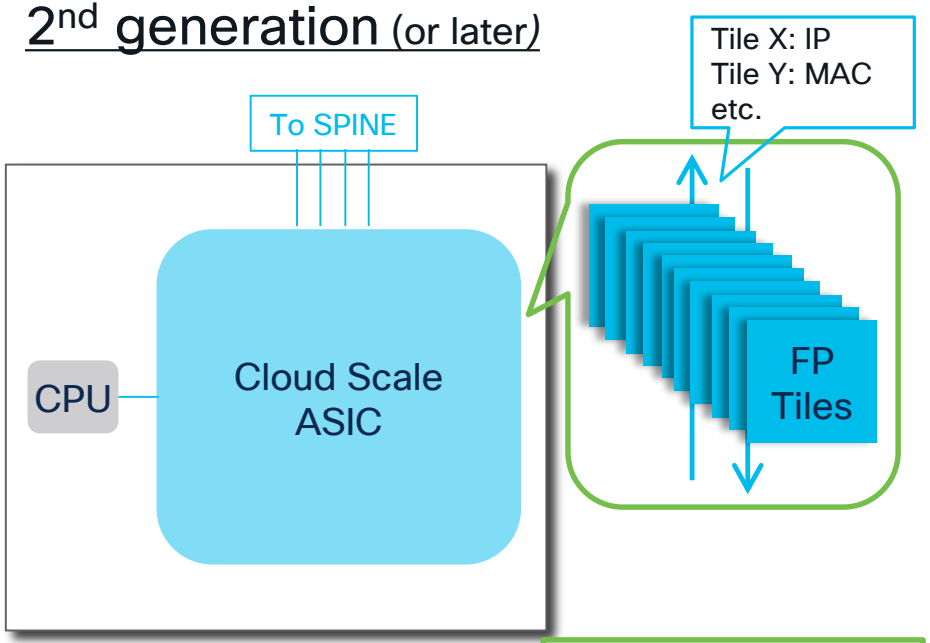
## 1st generation




N9K-C9332PQ
N9K-C9372PX
N9K-C9372PX-E
N9K-C9372TX
N9K-C9372TX-E
N9K-C9396PX
N9K-C9396TX
N9K-C93120TX
N9K-C93128TX

- Complete separation of
  + Ingress and Egress
  + Source Learn and Destination Lookup
- Separate GST/LST for IP and MAC

## 2nd generation (or later)

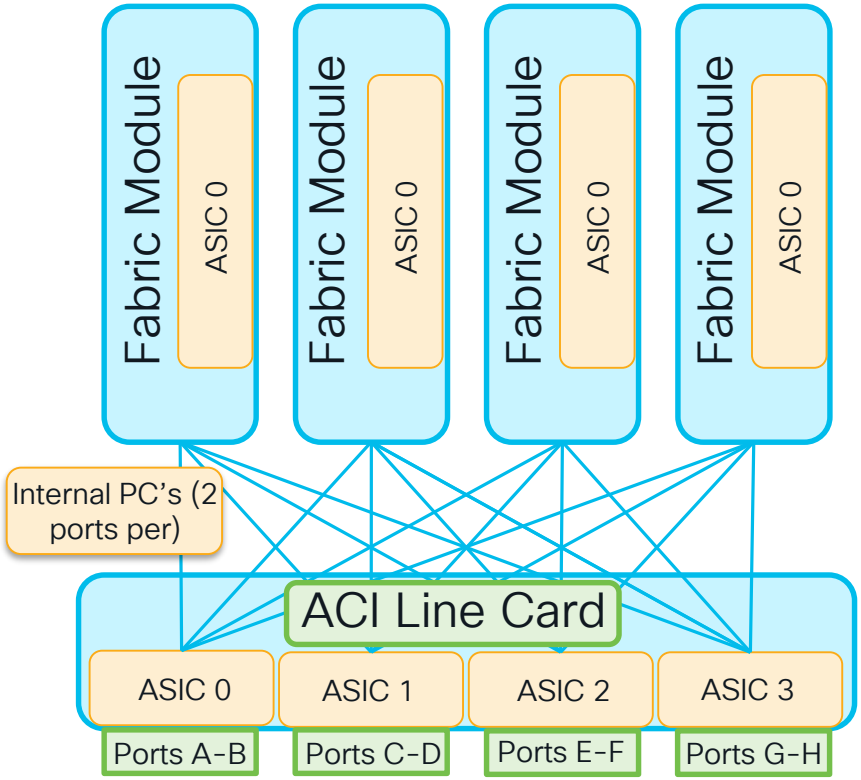



N9K-C*-EX
N9K-C*-FX
N9K-C*-FX2
N9K-C*-FX3
N9K-C*-FXP
N9K-C*-GX
N9K-C*-GX2

- More flexible/scalable with configurable tiles
- Abstracted with HAL
- Tile X for both source learn and destination lookup

※ number of ASIC per card depends on model

# Inside an ACI Modular Spine




What are the strange IP's on the Fabric Modules?

```
sp# vsh -c "slot 26 show plat internal hal l3 routes"
40.0.99.139/ 32
3.124.199.13/ 32
0.156.151.177/ 32
```

Where are the linecard forwarding tables?

```
sp# vsh -c "slot 2 show plat internal hal l3 routes"
<no output>
```

# Inside an ACI Modular Spine

## How is traffic forwarded?

### For Proxied Traffic

- Depending on if the dest IP is the L2 or L3 Proxy TEP the VRF VNID + Dest IP OR BD VNID + Dest MAC is used to hash a synthetic Dest IP and VRF ID
- Synthetic information is used on LC to hash the uplink port to FM
- FM routing lookup is based on Synthetic IP
- Each Synthetic IP is owned by two FM's
- FM uses vnTag to tell egress LC which front panel port to use

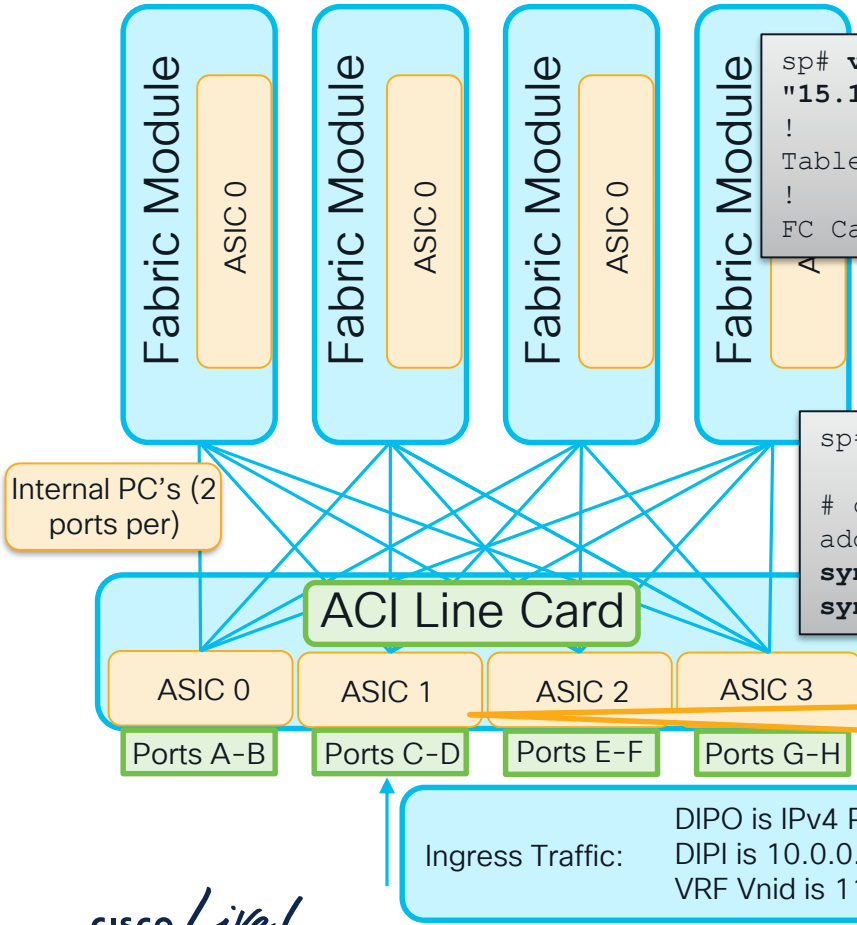
# Inside an ACI Modular Spine

## How is traffic forwarded?

### For Transit Traffic

- Line card hashes across ALL FM uplinks
- ALL FM's have overlay TEP routes
- FM uses vnTag to tell egress LC which front panel port to use

# Inside an ACI Modular Spine

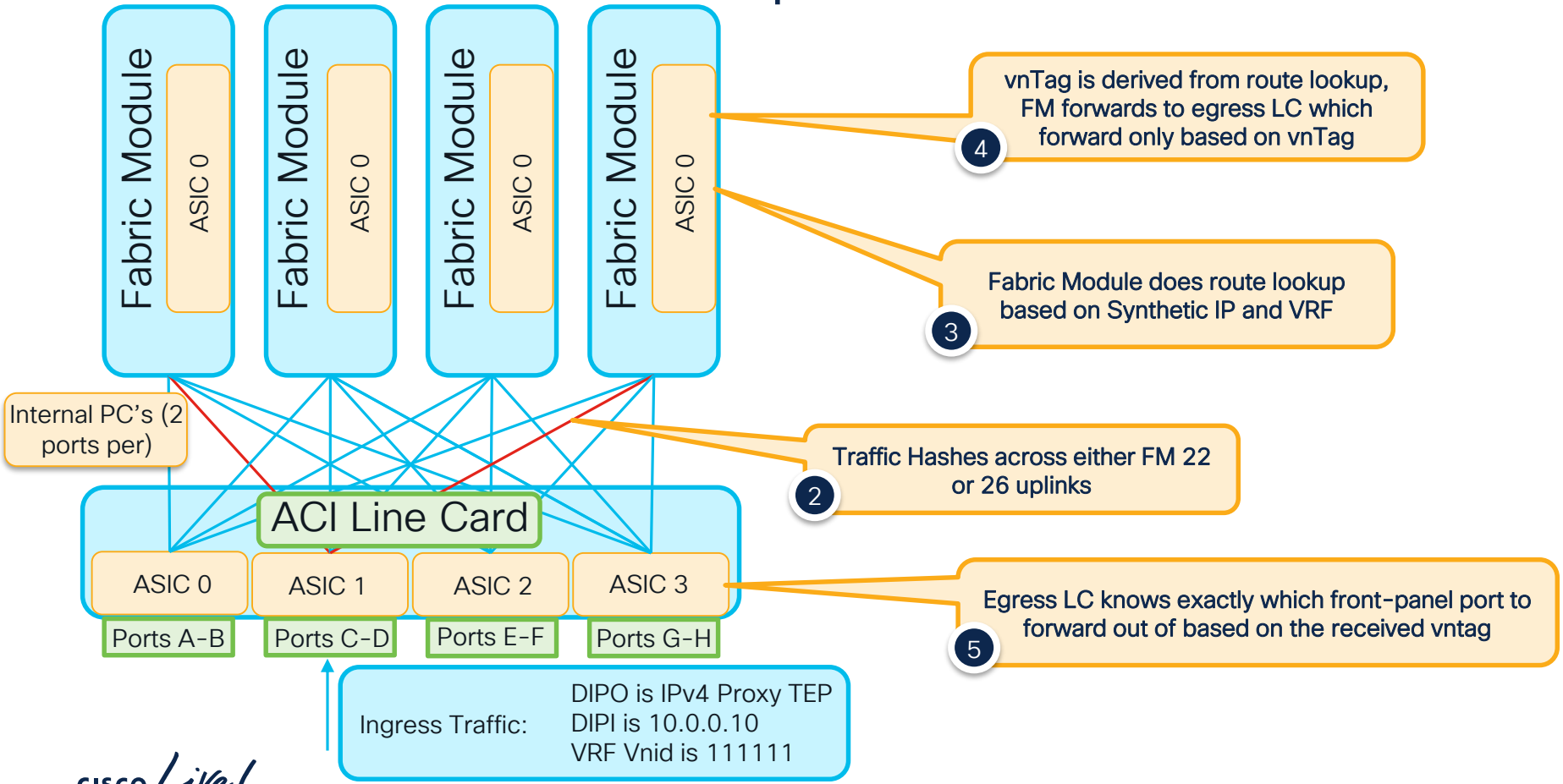



```
sp# vsh -c "slot 26 show forwarding route platform" | grep -A 10
"15.180.164.253"
!
Table: (IN-HW) Type=100 Vrf=750 Synth=0
!
FC Cards/ASICs : FC22/ASIC-0     FC26/ASIC-0
```

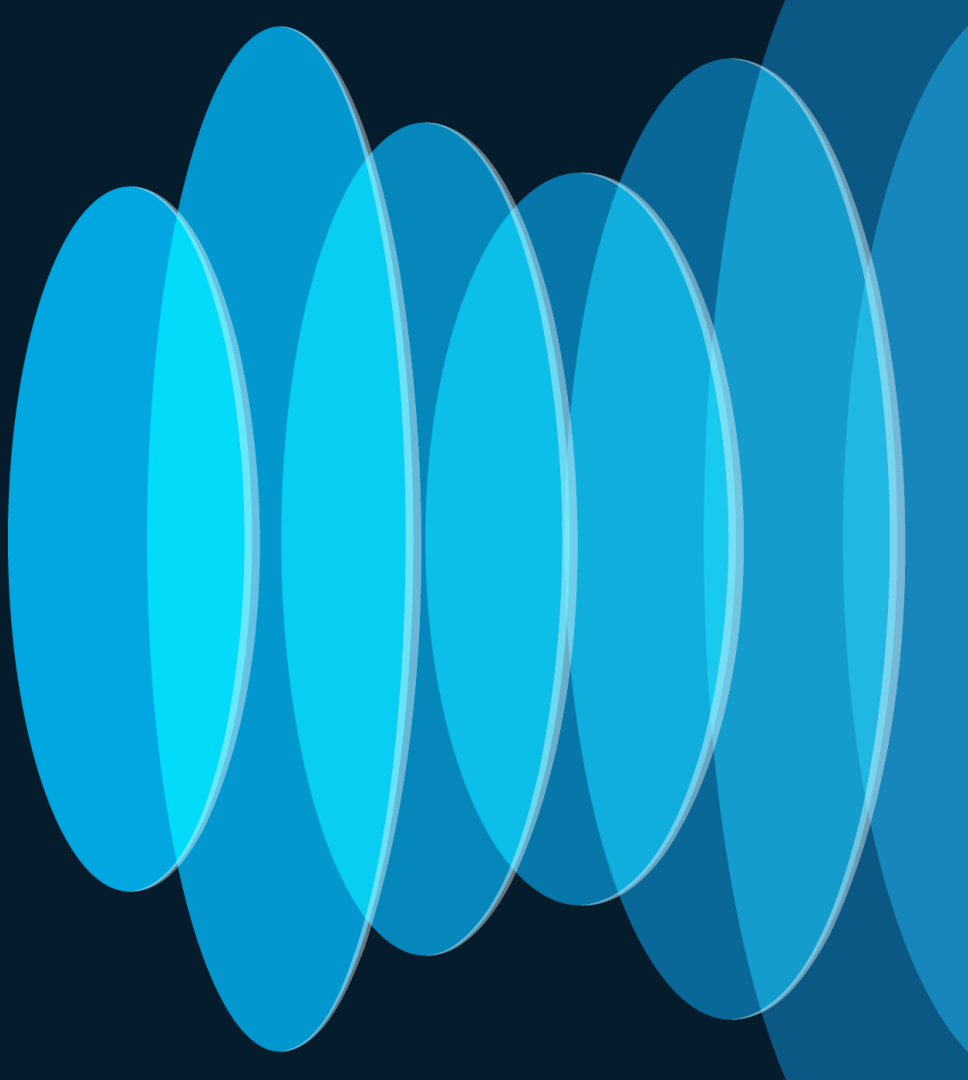
Synthetic info programmed on FM's 22 and 26

```
sp# moquery -c coopIpv4Rec -f 'coop.Ipv4Rec.addr=="10.0.0.10"'

# coop.Ipv4Rec
addr          : 10.0.0.10
synthIp       : 15.180.164.253
synthVrf      : 250
```

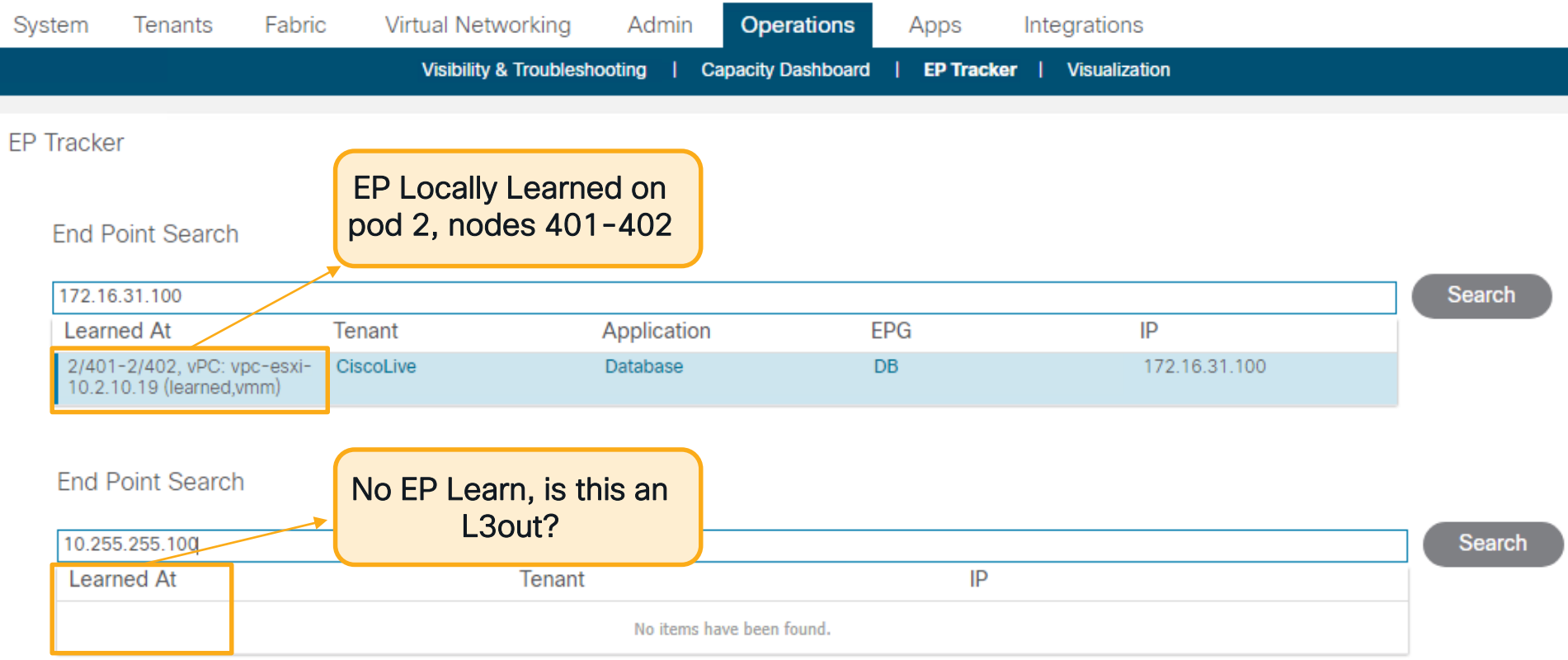
# Inside an ACI Modular Spine



# Understanding the Tools

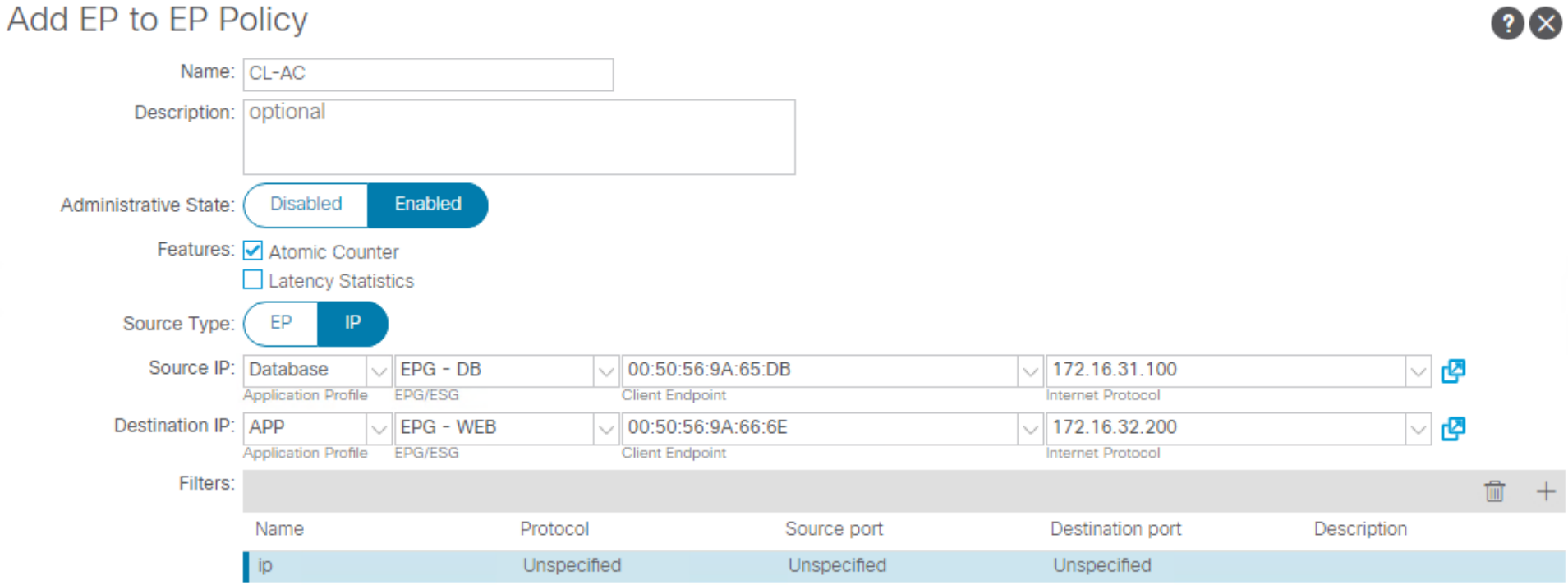


# Start with High-level Tools
## Use Endpoint Tracker for Building a Topology



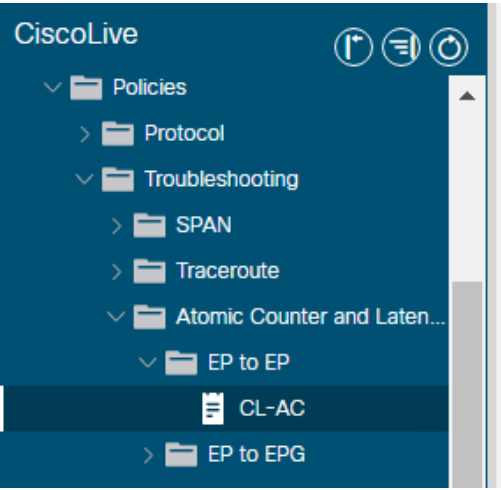
# Start with High-level Tools

## Use Atomic Counters to Check for Overlay Drops and Latency (PTP)



# Start with High-level Tools

## Use Atomic Counters to Check for Overlay Drops and Latency (PTP)



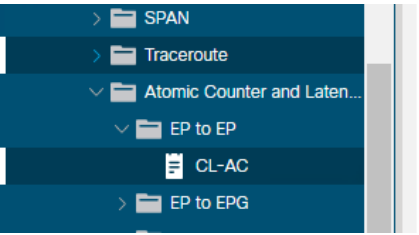

EP to EP CL-AC

EP-to-EP Atomic Counter - CL-AC

| Source | Destination | Last Collection (30 seconds) Pkt | | | |
|---|---|---|---|---|---|
| | | Transmit | Admitted | Dropped | Excess |
| uni/tn-CiscoLive/ap-Databas... | uni/tn-CiscoLive/ap-APP/epg... | 29 | 29 | 0 | 0 |

104 Microseconds of delay in overlay

No overlay drops!

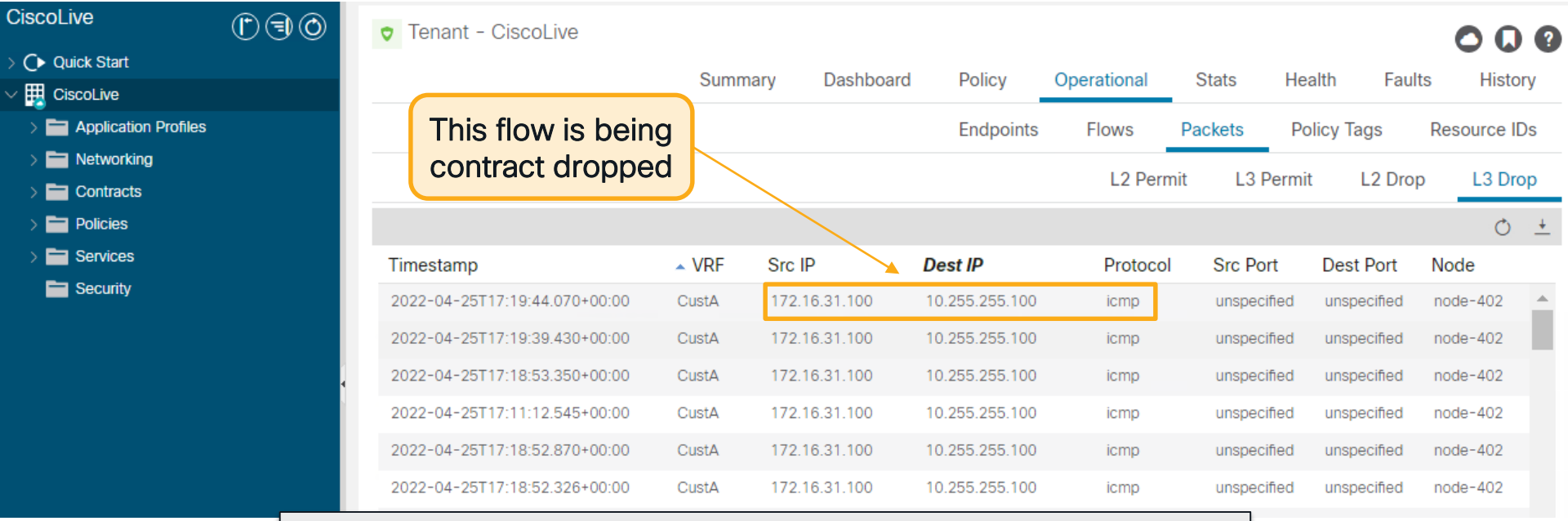



EP-to-EP Latency Average - CL-AC

| Last 30 Seconds Collection 04/25/2022 16:06:05 | | | Cumulative (04/25/2022 15:04:45 - 04/25/2022 16:06:05) | | |
|---|---|---|---|---|---|
| Average(µs) | Standard Deviation(µs) | Packet Count | Average(µs) | Max(µs) | Packet Count |
| 104.8575 | 0.0000 | 29 | 104.8575 | 104.8575 | 3768 |

# Start with High-level Tools

## Use Tenant Visibility tools to check for Contract Drops




```
apic4# show acllog deny l3 pkt tenant common  vrf  CORE
srcIp dstIp protocol srcPort dstPort node srcIntf vrfEncap
----- ----- -------- ------- ------- ---- ------- --------
<EMPTY>
```

# Contract Parser

## The script checks zoning rules, filters, statistics against EPG names

```
Leaf# contract_parser.py --help
 --nz, --nonzero       display only entries with non-zero hits
 --incremented         display only entries that have incremented since last  checked
 --node NODES [NODES ...]
                       display entries specific to one or more leaf nodes
 --contract CONTRACT [CONTRACT ...]
                       display only rules that match a specific contract. The
                       name of the contract is in the form
                       uni/tn-<tenant>/brc-<contract>
 --vrf VRF [VRF ...]   display entries for a specific vrf. The integer vnid
                       of the vrf can be provided or the vrf name in the form
                       <tenant>:<vrf>
 --epg EPG [EPG ...]   display entires for specific EPG. The integer pcTag or
                       DN name can be provided. Note the dn is a partial dn
                       in the form
                       tn-<tenant>/ap-<applicationProfile>/epg-<epg>
```

# Start with High-level Tools

## Port Counters are as Useful as Ever

```
leaf1# show interface eth1/8
Ethernet1/8 is up
admin state is up, Dedicated Interface
  Last link flapped 03:07:41
  RX
    3527922 unicast packets  !ommitted
    4041582 input packets  609518993 bytes
    12 jumbo packets  0 storm suppression bytes
    0 runts  0 giants  0 CRC  0 Stomped CRC  0 no buffer
    0 input error  0 short frame  0 overrun !ommitted
    0 watchdog  0 bad etype drop  0 bad proto drop !ommitted
    0 input with dribble  0 input discard
    0 input buffer drop  0 input total drop
  TX
    32262479565 unicast packets  !ommitted
    32395063346 output packets  49034781261
    32249687943 jumbo packets
    0 output error  0 collision  0 deferred
    0 lost carrier  0 no carrier  0 babble  0 output discard
    0 output buffer drops  0 output total drops
```

Frames received with bad FCS

Frame transmitted with stomped CRC

Buffer drops, sign of congestion

Indicates a previously stomped frame was received

What is a Stomp?

- When a frame is received with a bad FCS and/or is malformed

AND

- The frame is cut-through switched

The switch will invert the new CRC to tell the first store-and-forward device to drop it

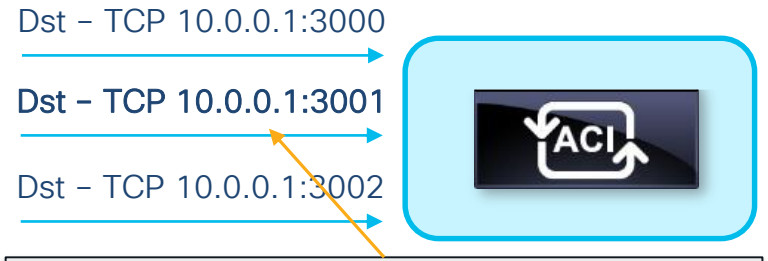
# Start with High-level Tools

## Using moquery to check port counters fabric-wide

```
#Check Fabric-wide for FCS Errors
moquery -c rmonDot3Stats -f 'rmon.Dot3Stats.fCSErrors>="1"' | egrep "dn|fCSErrors”

#Check Fabric-wide for total CRC Stomp + FCS Errors
moquery -c rmonEtherStats -f 'rmon.EtherStats.cRCAlignErrors>="1"' | egrep "dn|cRCAlignErrors”

#Check Fabric-wide for Output Buffer Drops
moquery -c rmonEgrCounters -f 'rmon.EgrCounters.bufferdroppkts>="1"' | egrep "dn|bufferdroppkts”

#Check Fabric-wide Output Errors
moquery -c rmonIfOut -f 'rmon.IfOut.errors>="1"' | egrep "dn|errors”
```

# ELAM – Embedded Logic Analyzer Module

- It is a tripwire in hardware
- The first frame to match a specified condition ‘trips’ it
- Report is created with vast amount of data regarding asic decisions




```
vsh_lc
debug platform internal tah elam asic 0
 trigger reset
 trigger init in-select 6 out-select 1
  set outer ipv4 dst_ip 10.0.0.1
  set outer l4 dst-port 3001
  start
```

```
module-1(DBG-elam-insel6)# stat
 ELAM STATUS
==========
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Triggered

module-1(DBG-elam-insel6)# ereport | grep "drop reason"
RW drop reason                          : no drop
LU drop reason                          : no drop
```

Matching frame was caught!

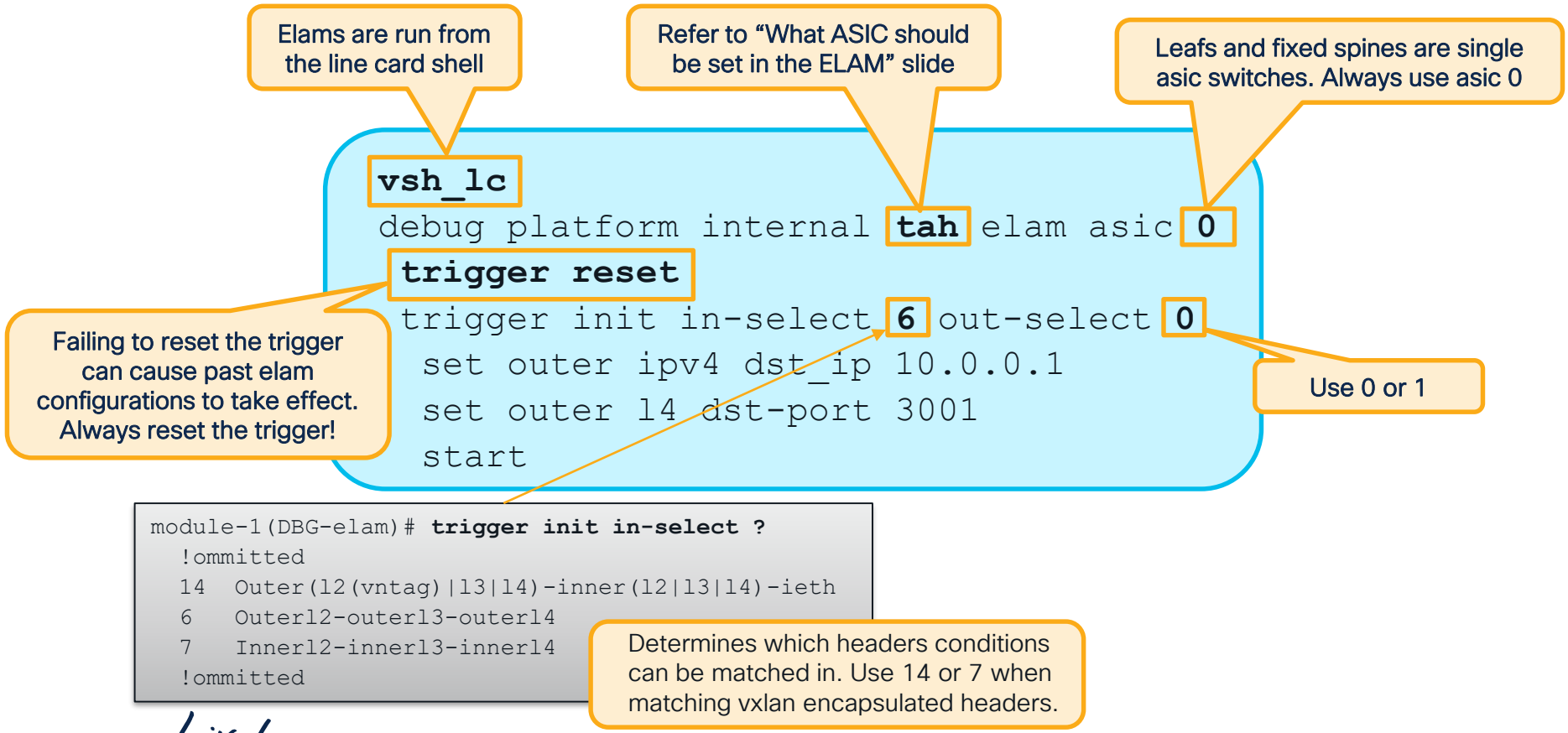
Frame was not dropped in lookups!

# What ASIC should be set in the ELAM?

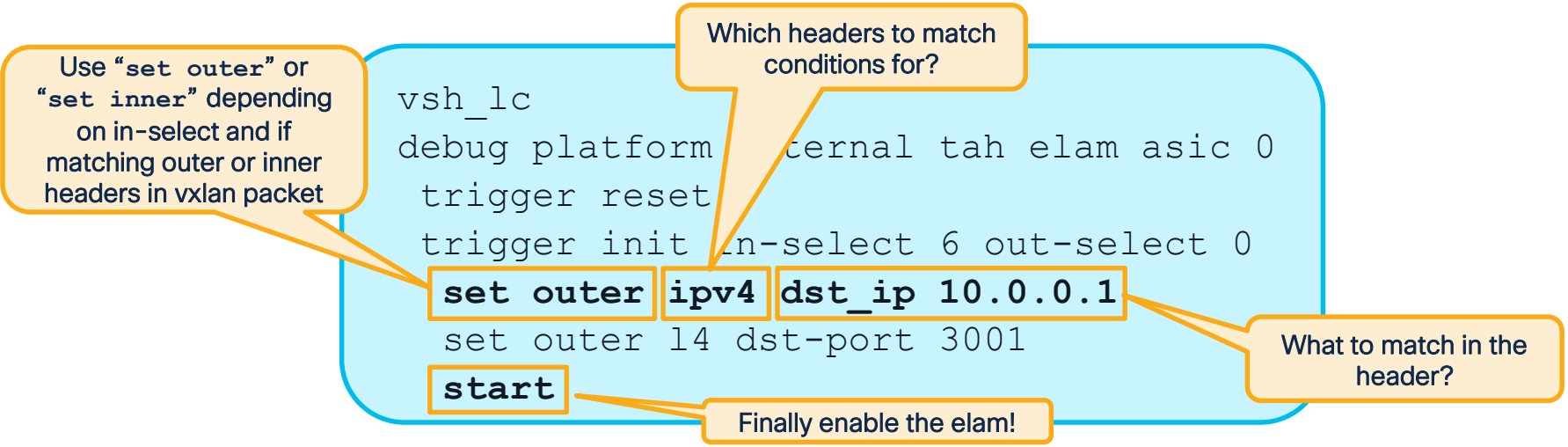
```
vsh_lc
debug platform internal <asic> elam asic 0
```

| Model | Role | Asic for Elam |
|---|---|---|
| N9K-C*C | Fixed Spine | roc |
| N9K-C*GX | Fixed Spine / Leaf | app |
| N9K-C*-GX2 | Fixed Spine / Leaf | cho |
| N9K-C*-EX | Leaf | tah |
| N9K-C*-FX/FXP/FX2/FX3 | Leaf | roc |
| N9K-X97*-EX | Spine LC | tah |
| N9K-X97*-FX | Spine LC | roc |
| N9K-X97*-GX | Spine LC | app |
| N9K-C95*-FM-E | Spine FM | tah |
| N9K-C950*-FM-E2 | Spine FM | roc |
| N9K-C95*-FM-G | Spine FM | app |

# Steps to Using Elam on Gen2+ Leaf or Fixed Spine



# Steps to Using Elam on Gen2+ Leaf or Fixed Spine




When running `stat` if `Triggered` is seen, this means a matching packet was received

# Reading an Elam

**ereport** available since 4.2

## At a high-level…

```
module-1(DBG-elam-insel6)# ereport
!ommitted
------------------------------------------------------------------
Outer L3 Header
------------------------------------------------------------------
L3 Type                : IPv4
IP Version             : 4
DSCP                   : 0
IP Packet Length       : 84 ( = IP header(28 bytes) + IP payload )
Don't Fragment Bit     : set
TTL                    : 64
IP Protocol Number     : ICMP
Destination IP         : 192.168.200.11
Source IP              : 192.168.100.10
!omitted
Contract Result
Contract Drop          : no
Contract Logging       : no
Contract Applied       : yes
Contract Hit           : yes
```

- ereport provides a simple, human-readable report output
- ereport requires >= 5.2 code for modular spines
- Groups data into outer/inner, headers, and lookup results

# What if Elam Shows a Drop?

**ereport** available since 4.2

```
ereport
Lookup Drop
------------------------------------
LU drop reason    : SECURITY_GROUP_DENY
```

## Common Drop Reasons

| Drop Code | What Does it Mean? | What to Do? |
|---|---|---|
| ACL_DROP | For traffic destined to the CPU on an FX switch it is expected and cosmetic. Also seen when traffic was received from a fabric port and the leaf has a remote EP learn with no bounce flag. | Ignore if its an FX switch and destined to local switch IP/process. Otherwise, check for incorrect EP learn. |
| DCI_*_XLATE_MISS | For multisite / remote-leaf, there was no matching vnid or pctag translation found. | Check contracts between local and remote resources. |
| INFRA_ENCAP_SRC_TEP_MISS | No route and/or tunnel found back to the outer source IP | Check for a tunnel pointing back to the outer source IP. Also, check for a route in overlay. |
| **SECURITY_GROUP_DENY** | **Frame was contract dropped** | **Make sure a contract is configured to allow the flow.** |
| SRC_VLAN_MBR | Received vlan not programmed on ingress port. | Check if the frame was correct tagged/untagged. Make sure no invalid-path faults exist for the epg. |
| UC_PC_CFG_TABLE_DROP | No route was found for the destination. | Check the routing table for the destination. |
| VLAN_XLATE_MISS | Received vlan doesn't exist on the switch. | Check if the frame is tagged with correct vlan. Check for invalid-path faults on the epg. |

# Steps to Using Elam on Gen2+ Modular Spine

## Challenges of Modular Spines

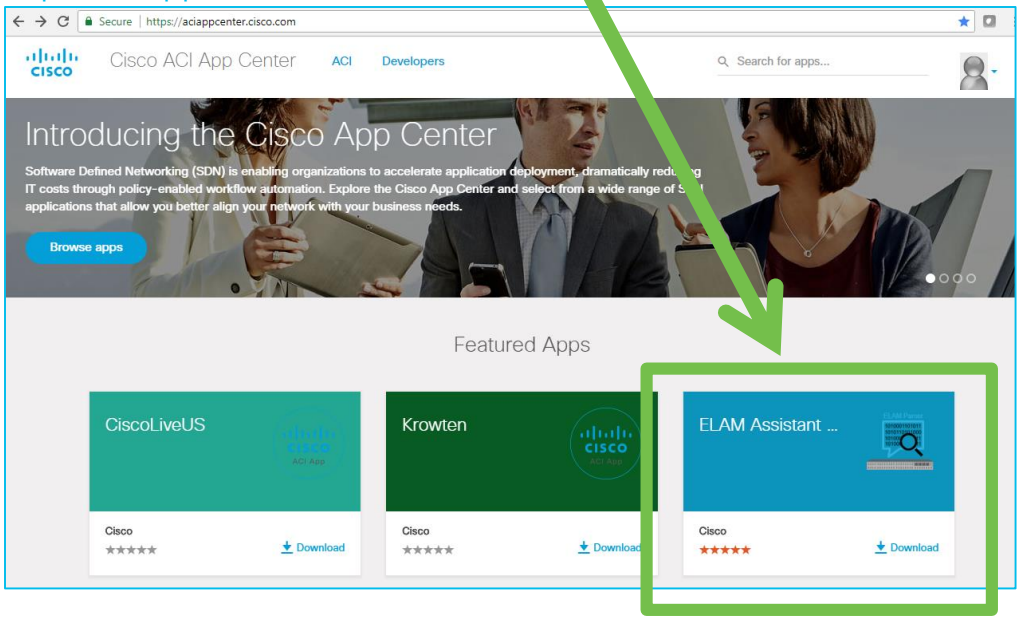
- Line cards (and potentially FM's) have multiple asics
- Elam must specify asic number
- Ingress/Egress ports may be internal LC – FM connections
- ereport only available in 5.2 and later

Fortunately, spine elams aren't needed as commonly as leaf elams!

# Shouldn’t ELAM be More Simple?

Elam Assistant in DCAppCenter

https://dcappcenter.cisco.com




ELAM (Embedded Logic Analyzer Module)

- Perform an ASIC level packet capture

ELAM Assistant

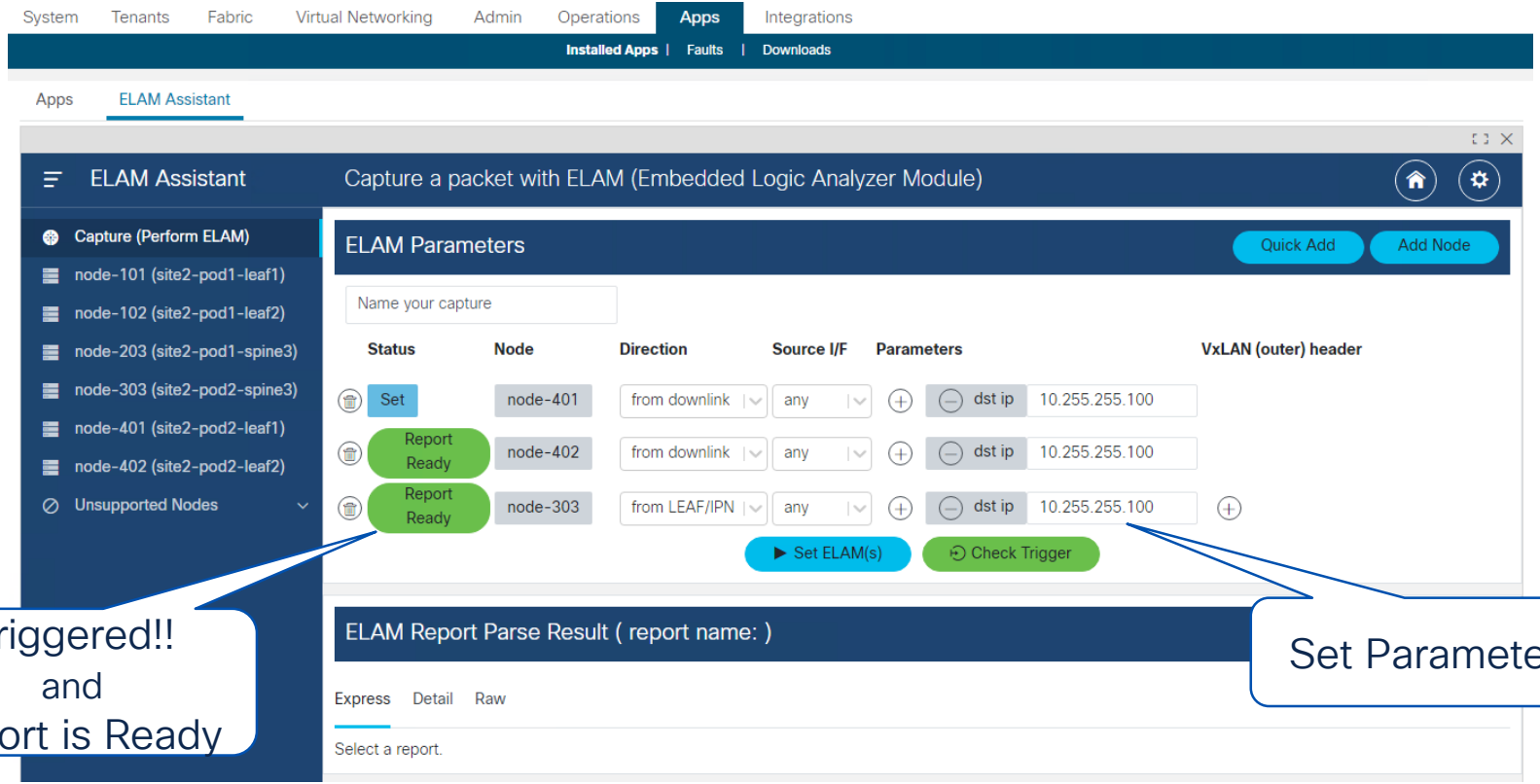
- You can perform ELAM like a TAC engineer!
- With a nicely formatted result report

Detail Explanations:

- https://dcappcenter.cisco.com/elam-assistant.html
- How to use video, pictures
  - ➢ A download link for ELAM Assistant

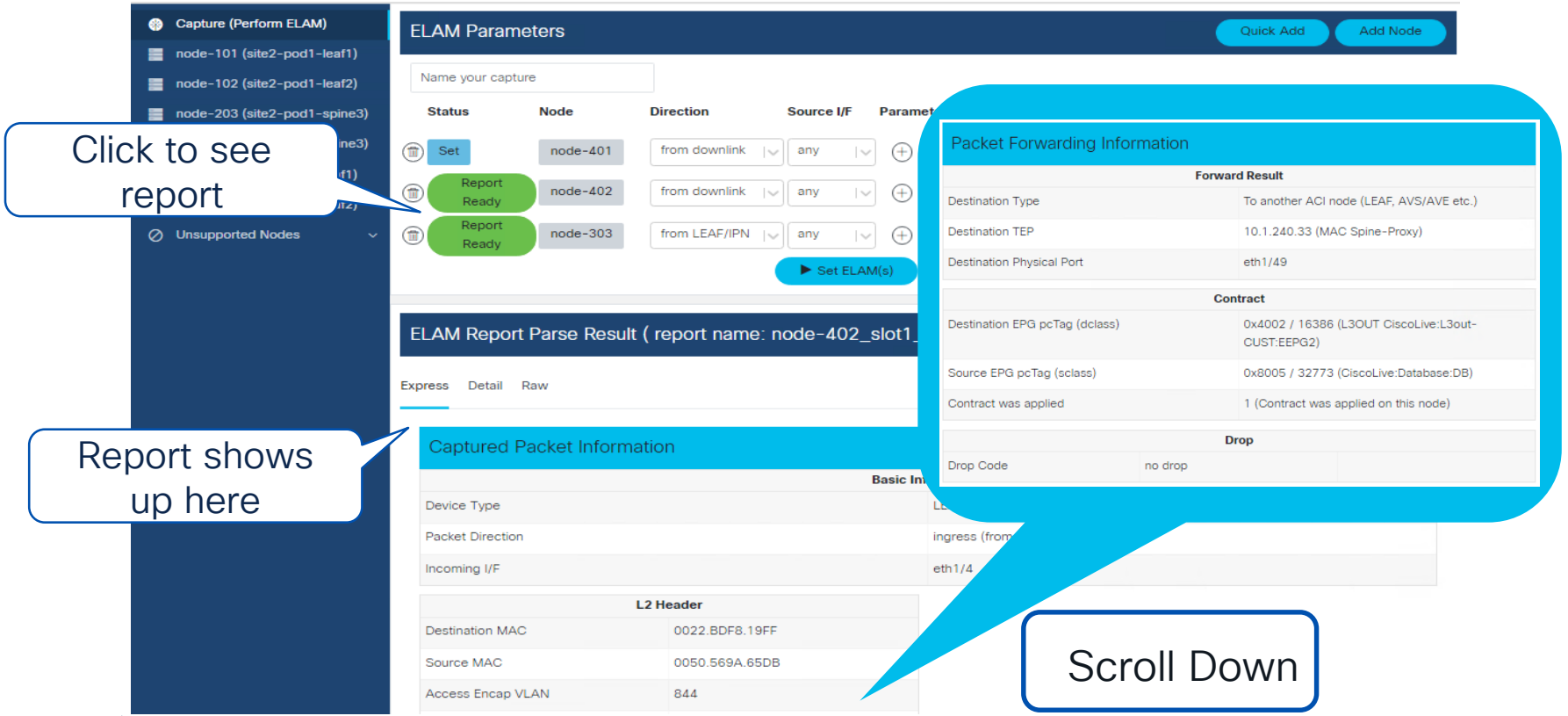
# ELAM Assistant in ACI AppCenter (example)

## 1. Perform an Elam



# ELAM Assistant in ACI AppCenter (example)

## 2. Read a Report



# FTRIAGE – Automating Elams

Orchestrate End-to-End ELAMs from the APIC!

```
apic1# ftriage route -ii LEAF:101,102 -dip 10.99.99.100 -sip 192.168.100.10
20:19:54 INFO main:1295 L3 packet Seen on  leaf102 Ingress: Eth1/34 (Po5) Egress: Eth1/54  Vnid: 2523136
20:19:55 INFO main:1364 leaf102: Packet's egress outer [SIP:10.0.176.67, DIP:10.0.64.70]
20:19:55 INFO main:1371 leaf102: Outgoing packet's Vnid: 2523136
20:19:56 INFO main:353  Computed ingress encap string vlan-3501
20:20:03 INFO main:464  Ingress BD(s) CL2022:bd1
20:20:03 INFO main:476  Ingress Ctx: CL2022:vrf1 Vnid: 2523136
!
20:21:46 INFO main:1295 L3 packet Seen on  spine1005 Ingress: Eth1/1 Egress: Eth1/3  Vnid: 2523136
20:22:38 INFO  fib:737  spine1005: Transit in spine
20:23:32 INFO main:1295 L3 packet Seen on  leaf103 Ingress: Eth1/29 Egress: Eth1/27/4  Vnid: NULL
!
20:24:02 INFO  fib:219  leaf103:  L3 out interface Ethernet1/27/4
20:24:10 INFO main:781  Computed egress encap string vlan-1055
20:24:17 INFO main:1796 Packet is Exiting fabric with peer-device: N3K-1 and peer-port: Ethernet1/31
```

# SPAN / ERSPAN

## Don’t neglect old friends!

- Both local span and erspan supported
- ERSPAN requires an l3 endpoint learned anywhere in the fabric
- Still the best tool for checking –
  - Packet contents
  - Frame format
  - Retransmissions
  - …and anything else that can be seen in a pcap

# Other Tools Requiring External Resources

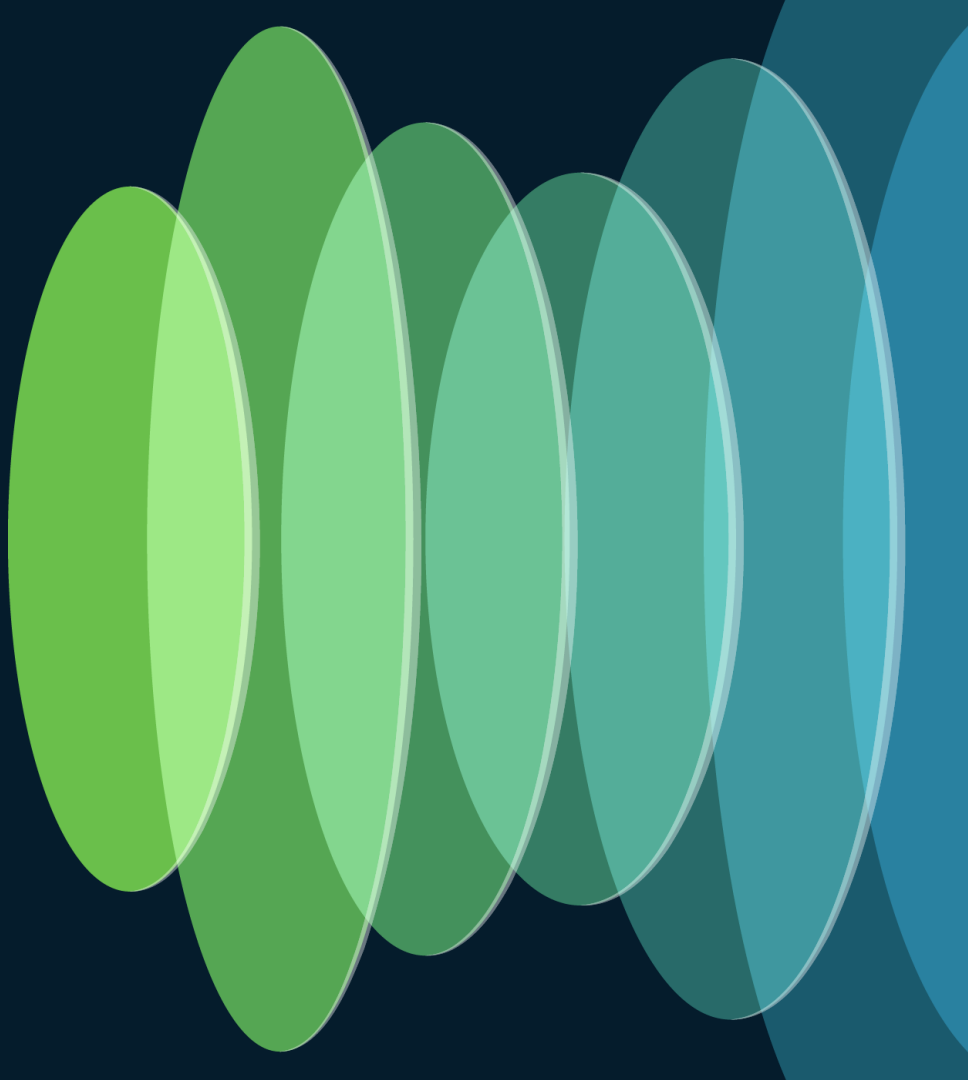
## Netflow

- Captures flow information based on specified criteria
- Useful for troubleshooting packet loss and latency

## Flow Telemetry

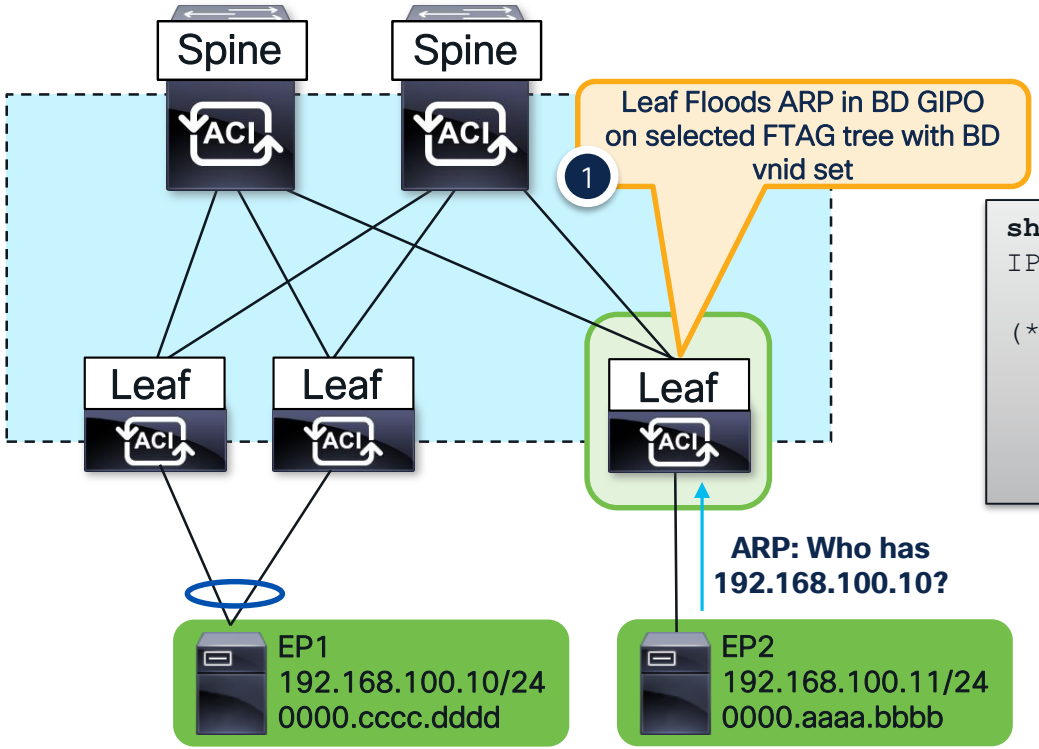
- Hardware directly streams flow data to Nexus Dashboard Insights
- Useful for troubleshooting packet loss and latency
- Latency measurements leverage PTP for additional accuracy
- NDI can perform additional flow analytics

# Debugging ACI BUM Flows



# ARP – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

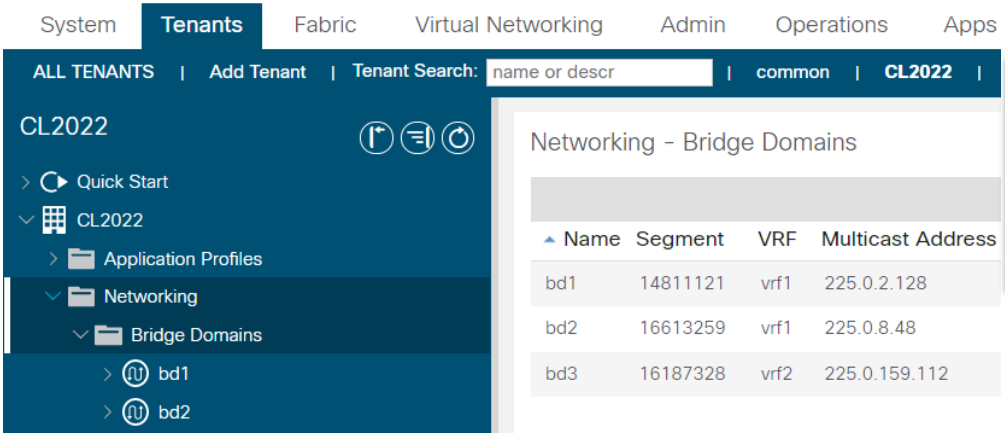



Check GIPO Route

```
show ip mroute 225.0.2.128 vrf overlay-1
IP Multicast Routing Table for VRF "overlay-1"

(*, 225.0.2.128/32), uptime: 22w2d, isis
  Incoming interface: Null, RPF nbr: 0.0.0.0
  Outgoing interface list: (count: 2)
    Ethernet1/29.9, uptime: 8w2d
    Ethernet1/30.10, uptime: 22w2d
```

# ARP – How to Find the GiPo

## From the GUI...




## From the APIC CLI...

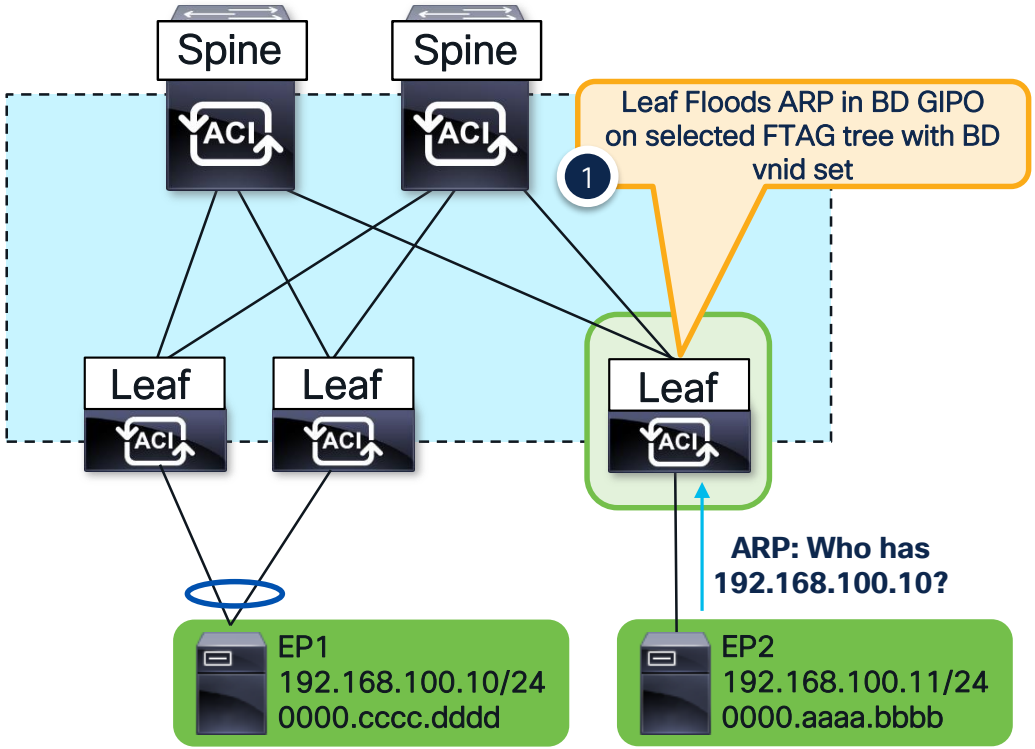
```
moquery -c fvBD -f 'fv.BD.dn*"tn-CL2022/BD-bd1"'

# fv.BD
arpFlood                    : yes
bcastP                      : 225.0.2.128
dn                          : uni/tn-CL2022/BD-bd1
```

## From the Switch CLI...

```
moquery -c l2BD -f 'l2.BD.name=="CL2022:bd1"' -x rsp-subtree=full rsp-subtree-class=fmcastGrp
# fmcast.Grp
  addr          : 225.0.2.128
  dn            : sys/ctx-[vxlan-2523136]/bd-[vxlan-14811121]/fmgrp-[225.0.2.128]
  rn            : fmgrp-[225.0.2.128]
```

# ARP – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled




ELAM the ARP request!

```
vsh_lc
debug plat internal app elam asic 0
 trigger reset
  trigger init in-select 6 out-select 0
  set outer arp source-ip 192.168.100.11
  set outer arp target-ip 192.168.100.10
  start
!
stat
 ELAM STATUS
==========
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Armed
Asic 0 Slice 2 Status Triggered
Asic 0 Slice 3 Status Armed
```

# ARP – Ingress Leaf Elam Results (ereport)

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

```
Outer L2 Header
-----------------------------------
Access Encap VLAN    : 3502( 0xDAE )

Outer L3 Header
-----------------------------------
ARP Opcode           : Request( 0x1 )
ARP Sender IP        : 192.168.100.11
ARP Target IP        : 192.168.100.10

Contract Result
------------------------
Contract Drop        : no
Contract Applied     : no

FINAL FORWARDING LOOKUP
-------------------------------------------------------------------------------
Bits set in Final Forwarding Block:    : IFABRIC_IG MC TENANT MYTEP BRIDGE MISS FLOOD

Lookup Drop
----------------------------
LU drop reason       : no drop
```

Make sure this matches what is expected

Frame is flooded in the Bridge Domain!

Not Dropped in lookups!

# ARP – How to Find the FTAG

## No other way than Elam…

```
module-1(DBG-elam-insel6)# ereport | grep "nopad.ftag"
      wol_lu2ba_sb_info.mc_info.mc_info_nopad.ftag: 0x8
```

Selected ftag is 0x8

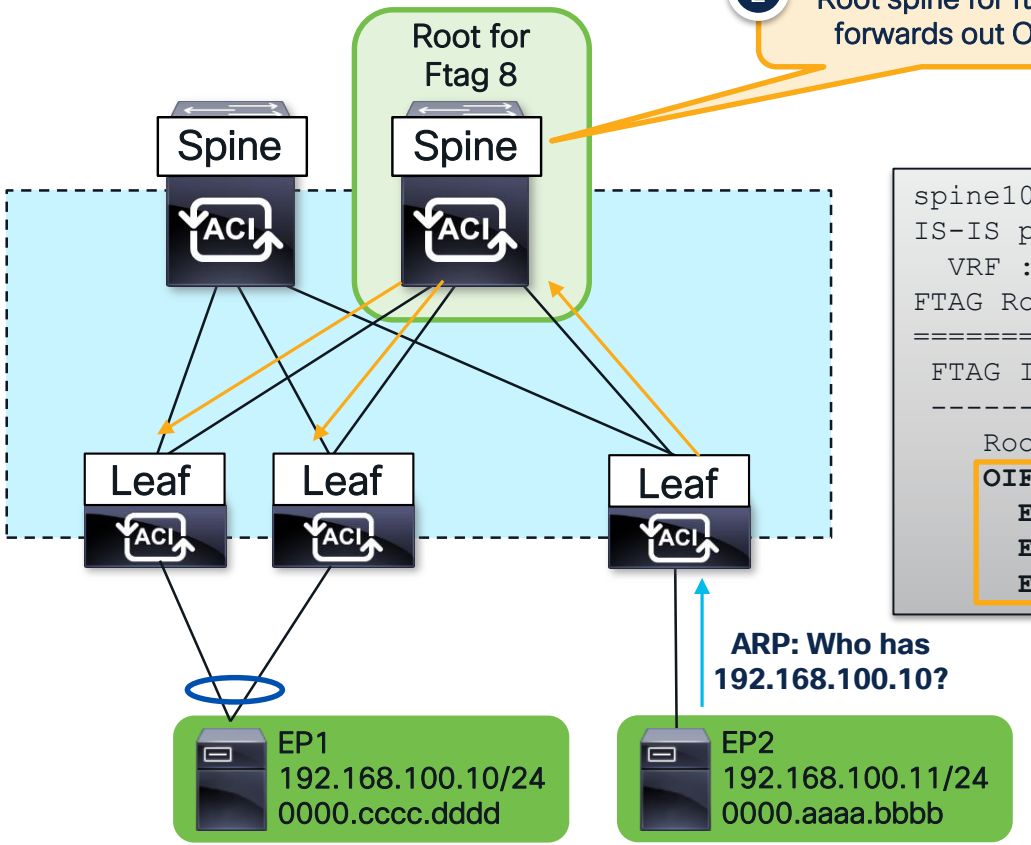
- Leaf forwards to root port and OIF's for ftag 8
- Since GIPO is 225.0.2.128, Dest multicast address is 225.0.2.136 (gipo + ftag)
- Check ftag topology with **`show isis internal mcast routes ftag`**

```
leaf103# show isis internal mcast routes ftag
IS-IS process: isis_infra
  VRF : default
FTAG Routes
====================================
 FTAG ID:   8  [Enabled] Cost:(   1/   6/   0)
 ----------------------------------
    Root port: Ethernet1/29.9
    OIF List:
```

Leaf appends ftag to gipo and forwards out Eth1/29 to spine

# ARP - Spine

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled



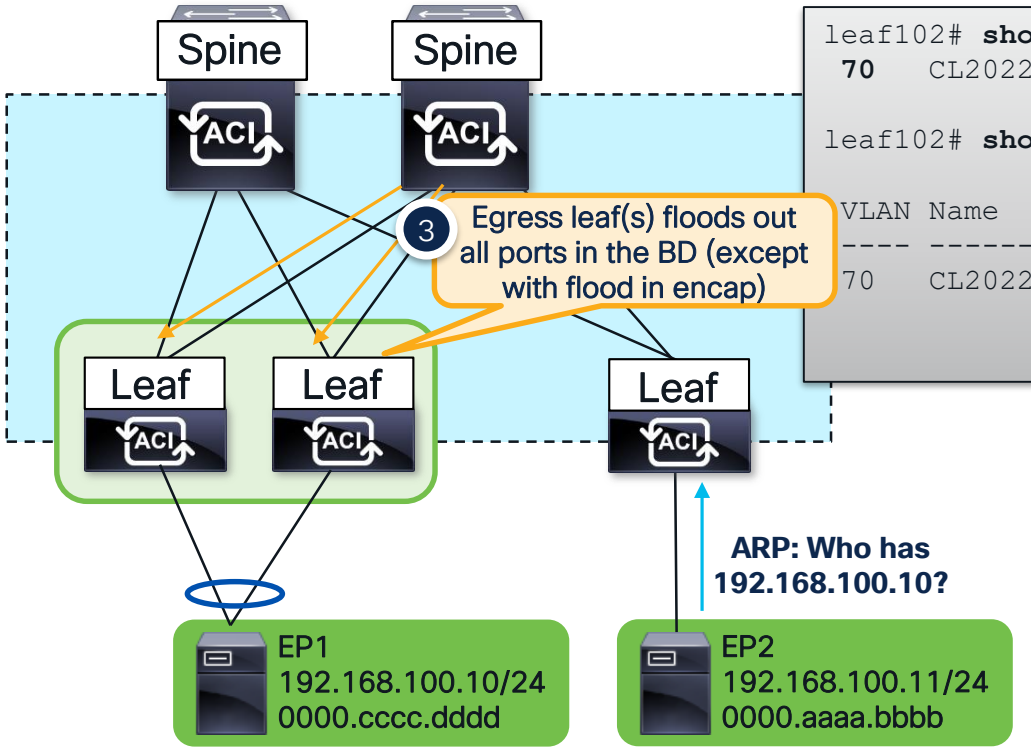

This spine is the root!

```
spine1005# show isis internal mcast routes ftag
IS-IS process: isis_infra
  VRF : default
FTAG Routes
====================================
 FTAG ID:    8 [Root] [Enabled] Cost:(   0/   0/   0)
 ----------------------------------
    Root port: -
    OIF List:
      Ethernet1/1.20
      Ethernet1/2.21
      Ethernet1/3.19
```

Spine forwards out OIFs

# ARP – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

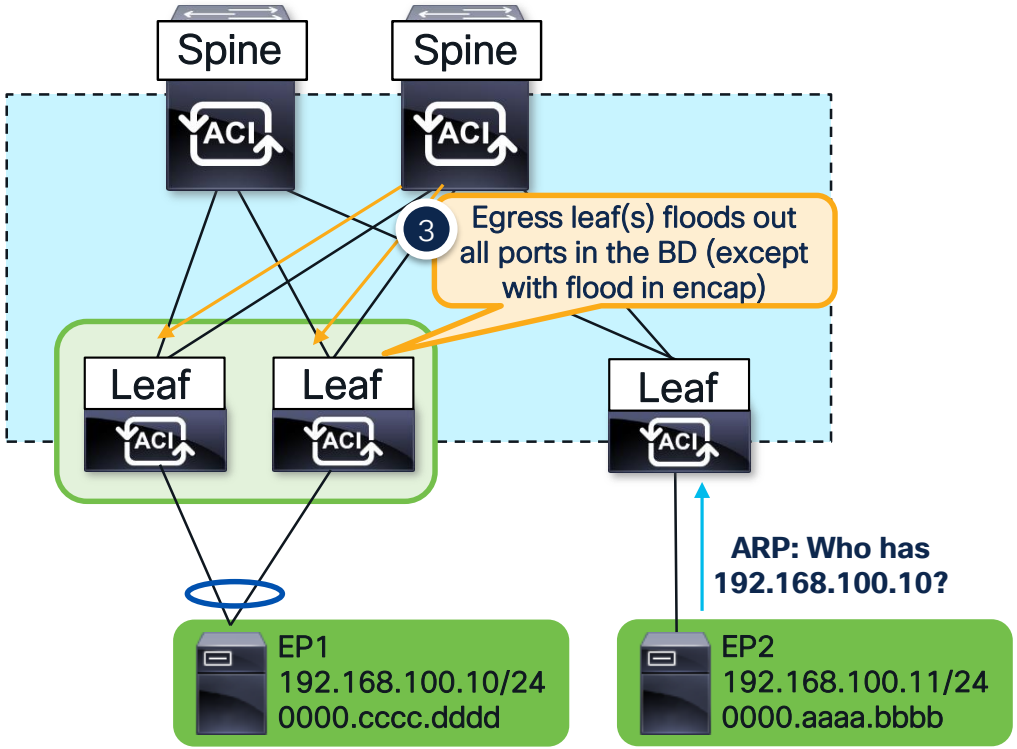



```
leaf102# show vlan | grep CL2022:bd1
 70   CL2022:bd1                            active

leaf102# show vlan id 70 extended

VLAN Name            Encap                Ports
---- --------------- -------------------- ---------------------
70   CL2022:bd1      vxlan-14811121       Eth1/33, Eth1/34,
                                          Eth1/43, Eth1/48, Po5,
                                          Po6, Po8, Po9
```

Flood out these ports and encaps in this BD

# ARP – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled




ELAM the ARP request!

```
vsh_lc
debug plat internal tah elam asic 0
 trigger reset
 trigger init in-select 14 out-select 1
  set inner arp source-ip 192.168.100.11
  set inner arp target-ip 192.168.100.10
  set inner l2 dst_mac ffff.ffff.ffff
  start

stat
 ELAM STATUS
==========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

# ARP – Egress Leaf Elam Results (ereport)

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

```
Outer L3 Header
------------------------------
Destination IP    : 225.0.2.136

Inner L3 Header
---------------------------------
ARP Sender IP     : 192.168.100.11
ARP Target IP     : 192.168.100.10

Outer L4 Header
--------------------------------------
VRF or BD VNID    : 14811121( 0xE1FFF1 )

Contract Result
----------------------
Contract Drop     : no

FINAL FORWARDING LOOKUP
------------------------------------------------------------------------------------
Bits set in Final Forwarding Block:  : IFABRIC_EG MC INFRA ENCAP MYTEP BRIDGE MISS FLOOD

Lookup Drop
---------------------------
LU drop reason    : no drop
```

Destination is GIPO (225.0.2.128) + FTAG (0x8)

Frame is flooded in the Bridge Domain!

Not Dropped in lookups!

# ARP – Egress Leaf Port is VPC

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

- Both VPC members receive a flooded copy
- One VPC member is the Designated Forwarder (DF) for the flow
- DF is hashed per flow
- Only DF floods out VPC interfaces

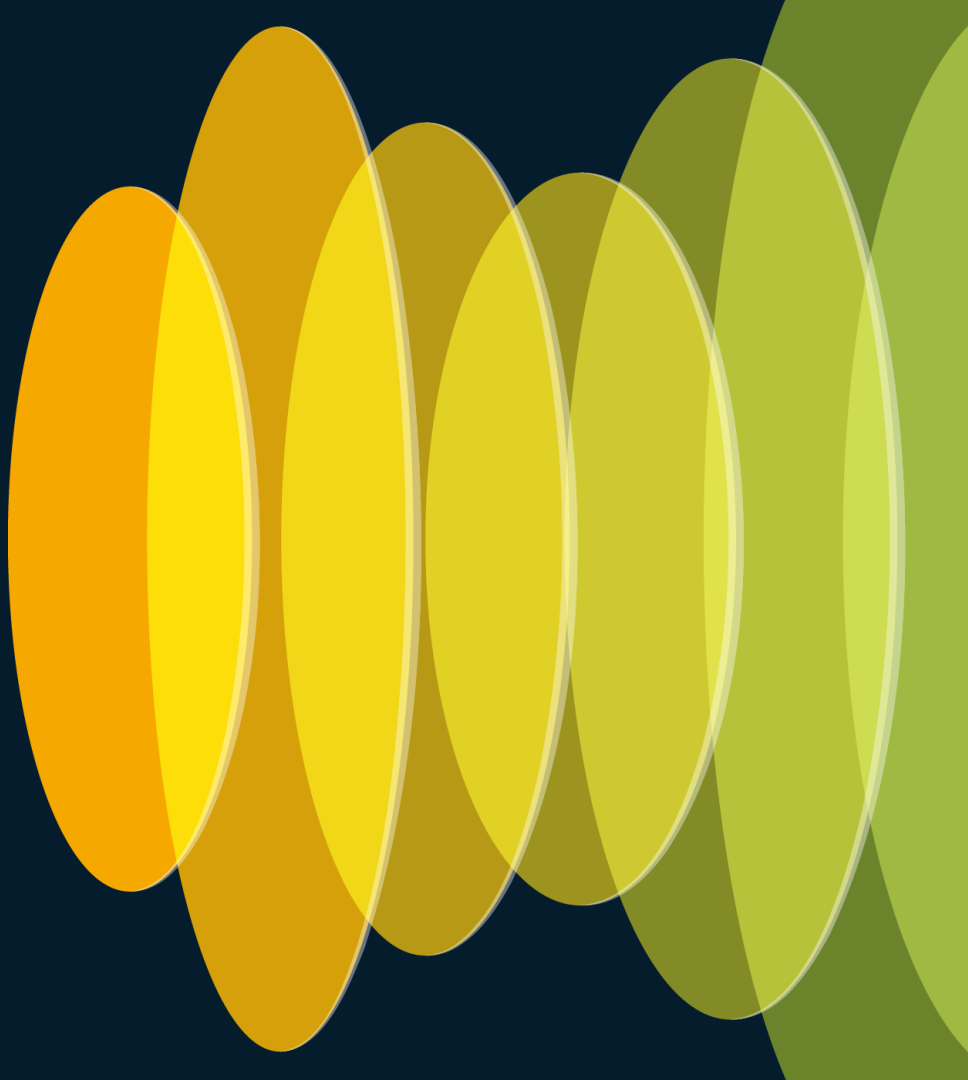
Non-DF Leaf

```
module-1(DBG-elam-insel14)# ereport | grep df | grep vpc
  sug_lub_latch_results_vec.lub4_1.vpc_df: 0x0
        sug_fpx_lookup_vec.lkup.dciptvec.pt.vpc_df: 0x0
          sug_fpc_lookup_vec.fplu_vec.lkup.dciptvec.pt.vpc_df: 0x0
          sug_fpc_lookup_vec.fplu_vec.lkup.dciptvec.pt.vpc_df: 0x0
```

DF Leaf

```
module-1(DBG-elam-insel14)# ereport | grep df | grep vpc
  sug_lub_latch_results_vec.lub4_1.vpc_df: 0x1
        sug_fpx_lookup_vec.lkup.dciptvec.pt.vpc_df: 0x1
          sug_fpc_lookup_vec.fplu_vec.lkup.dciptvec.pt.vpc_df: 0x1
          sug_fpc_lookup_vec.fplu_vec.lkup.dciptvec.pt.vpc_df: 0x1
```

# Debugging ACI Bridged Flows



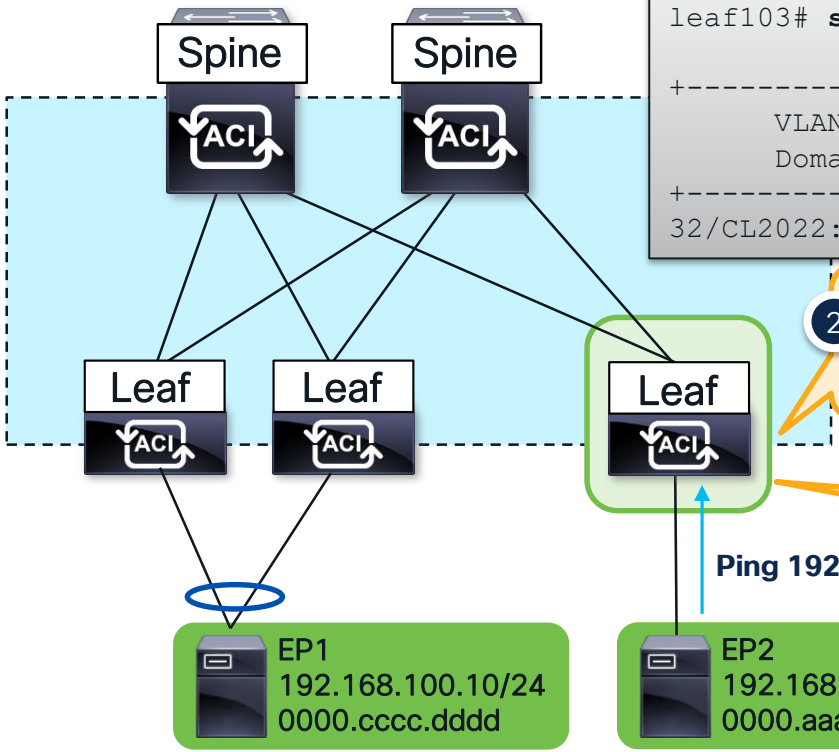
cisco Live!

# Known Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

Lookup dst mac in ingress BD

```
leaf103# show endpoint mac 0000.cccc.dddd

+--------------+--------------+-----------------+-------------+
      VLAN/         Encap           MAC Address        Interface
      Domain        VLAN            IP Address
+--------------+--------------+-----------------+-------------+
32/CL2022:vrf1    vxlan-14811121     0000.cccc.dddd          tunnel1
```




2 Since dst mac is not the router (GW) mac, leaf does mac lookup in BD (usually determined by ingress vlan)

```
leaf103# show int tun1
Tunnel destination 10.0.176.67
```

1 Leaf looks at the dst mac to determine if it should route or switch

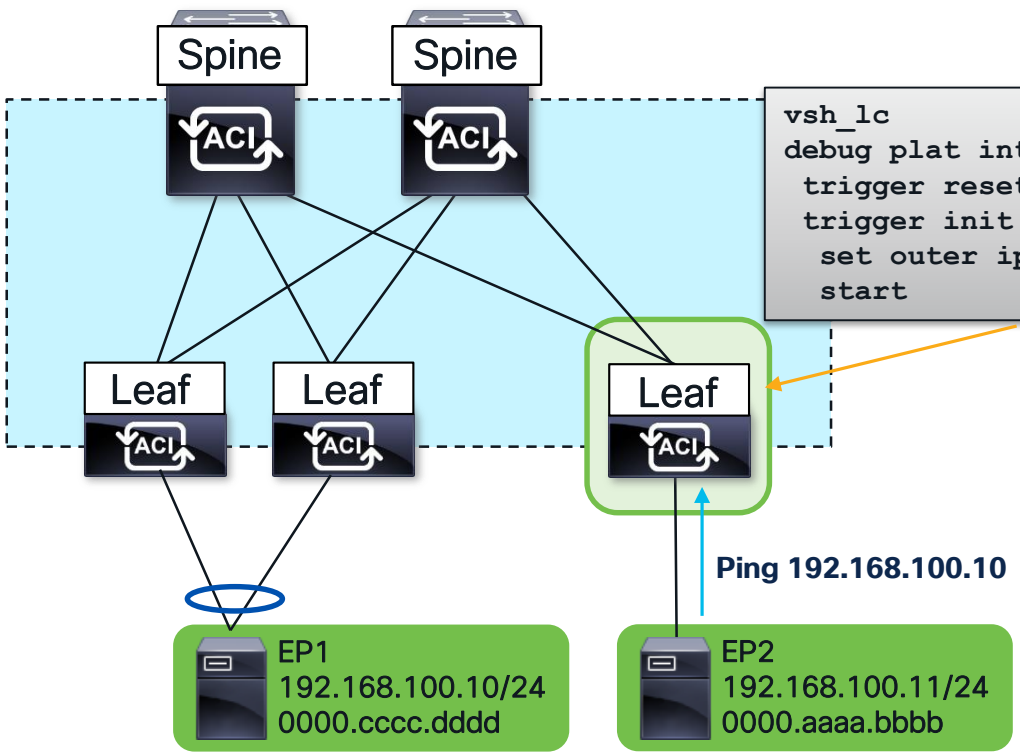
Ping 192.168.100.10

EP1
192.168.100.10/24
0000.cccc.dddd

EP2
192.168.100.11/24
0000.aaaa.bbbb

# Known Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood




ELAM

```
vsh_lc
debug plat internal app elam asic 0
 trigger reset
 trigger init in-select 6 out-select 0
  set outer ipv4 src_ip 192.168.100.11 dst_ip 192.168.100.10
  start
```

# Known Unicast – Ingress Leaf

## Forwarding Verifications

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

```
Outer L2 Header
------------------------------------
Destination MAC     : 0000.cccc.dddd
Source MAC          : 0000.aaaa.bbbb
Access Encap VLAN   : 3502( 0xDAE )

Outer L3 Header
------------------------------------
IP Protocol Number : ICMP
Destination IP     : 192.168.100.10
Source IP          : 192.168.100.11

Other Forwarding Information
------------------------------------
Encap Index is valid       : yes
Encap Index                : 34( 0x22 )

FINAL FORWARDING LOOKUP
------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_IG UC TENANT MYTEP BRIDGE HIT

Lookup Drop
-------------------------
LU drop reason    : no drop
```

Dest mac that is looked up within BD

Make sure this is the expected vlan

Dest is tunnel

```
show plat internal hal tunnel rtep apd
=====================================
ifId     IP             RwEncapIdx
=====================================
18010001 10.0.176.67    22
```

Forward to this overlay TEP

Not Dropped in lookups!

Unicast + Bridge (L2 lookup) + Destination Known

# Known Unicast – Ingress Leaf

## Forwarding Verifications

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

```
ereport | grep "ovector "
ovector    : 152( 0x98 )
```

```
show platform internal hal l2 port gpd

==========================================

IfId     Ifname      As AP   Sl Sp Ss Ovec
==========================================
1a01c000 Eth1/29     0  59   2  18 18 98
```

Traffic is forwarded out Eth1/29!

# Known Unicast – Ingress Leaf

## Contract Verification

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

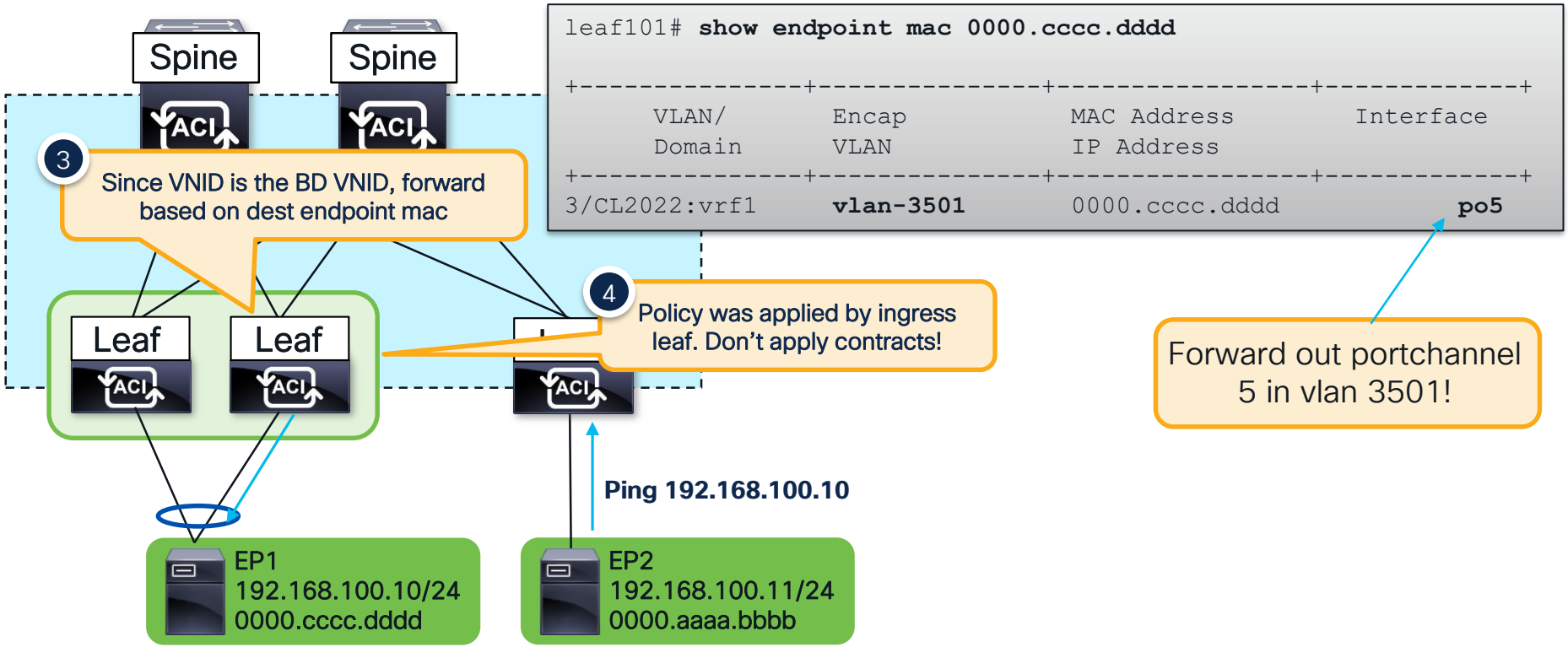
```
Contract Lookup Key
------------------------------------------------
IP Protocol                     : ICMP( 0x1 )
L4 Src Port                     : 2048( 0x800 )
L4 Dst Port                     : 35914( 0x8C4A )
sclass (src pcTag)              : 49154( 0xC002 )
dclass (dst pcTag)              : 49154( 0xC002 )
src pcTag is from local table   : yes
Unknown Unicast / Flood Packet  : no

Contract Result
------------------------------------------------
Contract Drop                                : no
Contract Applied                             : yes
Contract Hit                                 : yes
Contract Aclqos Stats Index                  : 131025
( show sys int aclqos zoning-rules | grep -B 9 "Idx: 131025" )
```

Source and Dest EPG is the same. Implicitly permit! (unless isolation enabled)

Contract Applied and no Drop!

# Known Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

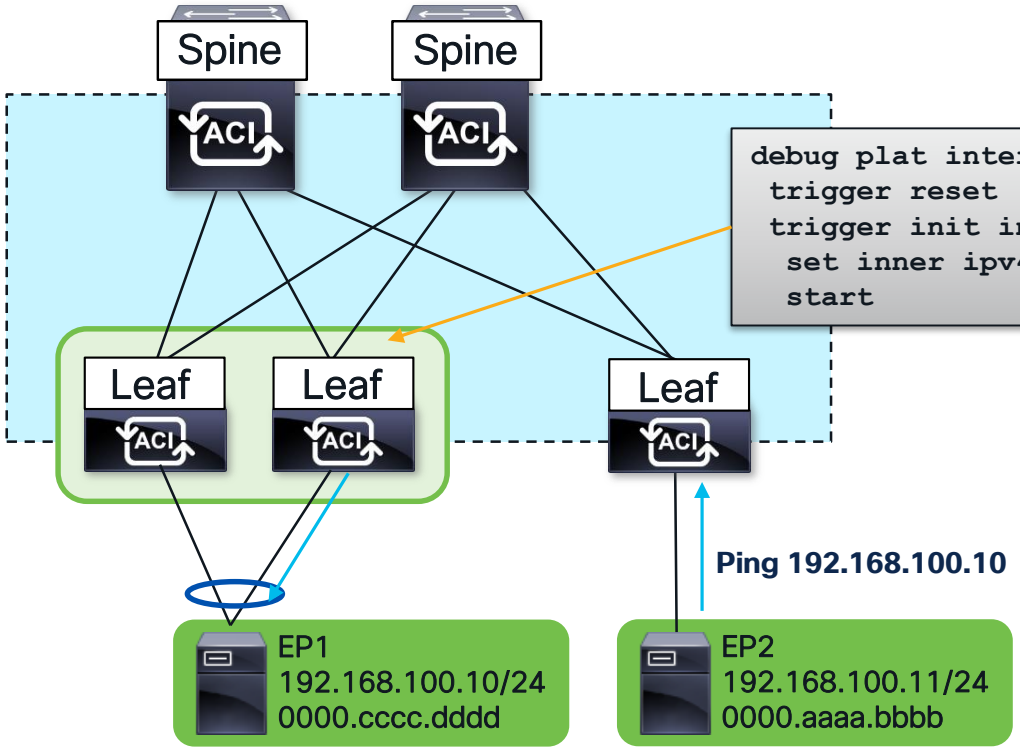



```
leaf101# show endpoint mac 0000.cccc.dddd

+---------------+---------------+-----------------+-------------+
      VLAN/          Encap            MAC Address        Interface
      Domain         VLAN             IP Address
+---------------+---------------+-----------------+-------------+
3/CL2022:vrf1      vlan-3501        0000.cccc.dddd             po5
```

Forward out portchannel 5 in vlan 3501!

# Known Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood




Elam

```
debug plat internal tah elam asic 0
 trigger reset
 trigger init in-select 14 out-select 0
  set inner ipv4 src_ip 192.168.100.11 dst_ip 192.168.100.10
  start
```

# Known Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

```
Inner L2 Header
-------------------------------
Inner Destination MAC  : 0000.cccc.

Inner L3 Header
-------------------------------
Destination IP         : 192.168.100

Outer L4 Header
-------------------------------
L4 Type                : iVxLAN
Src Policy Applied Bit : 1
Dst Policy Applied Bit : 1
VRF or BD VNID         : 14811121( 0xE1FFF1 )

Sideband Information
-------------------------------
ovector                : 146( 0x92 )

FINAL FORWARDING LOOKUP
-------------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_EG UC INFRA ENCAP MYTEP BRIDGE HIT

Lookup Drop
-----------------------
LU drop reason    : no drop
```

Contracts have already been applied. No need to check.

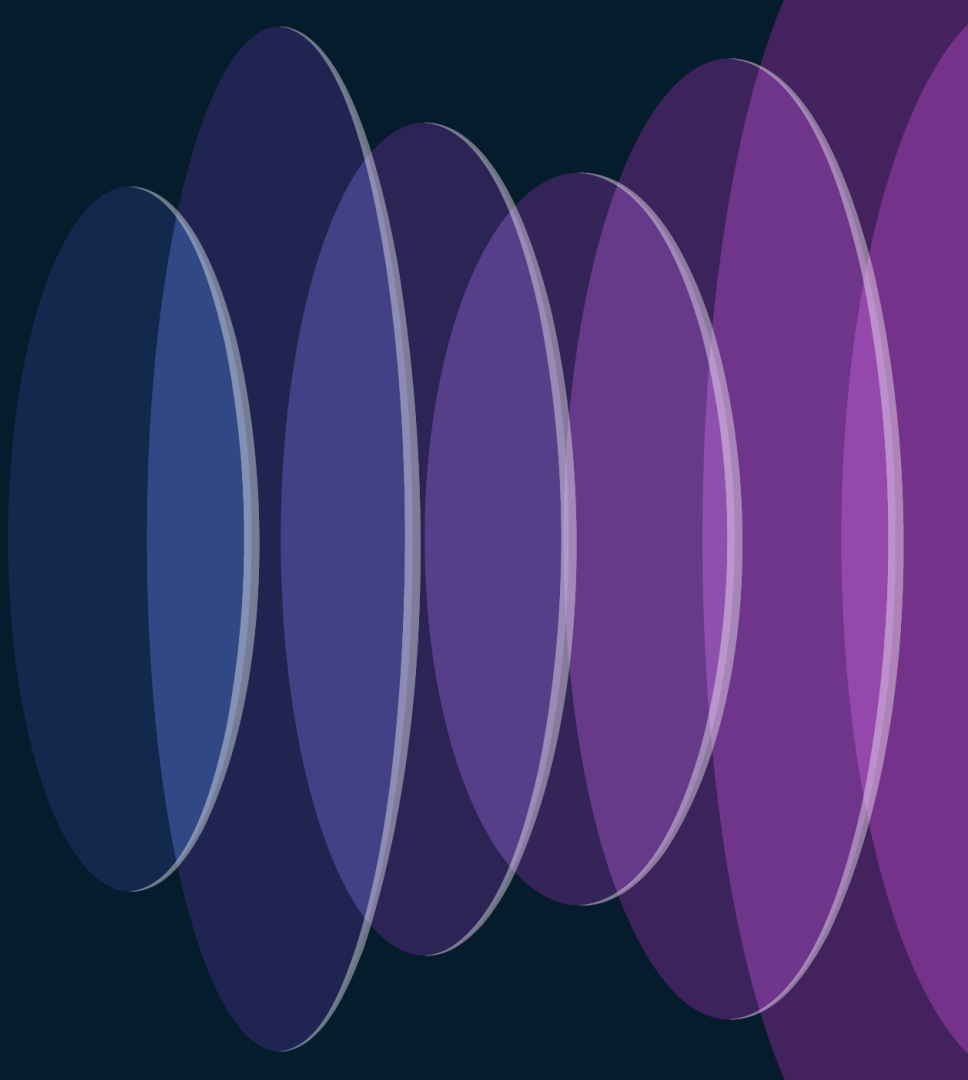
Mac lookup done in bridge domain with this VNID

```
show platform internal hal l2 port gpd
=======================================
IfId     Ifname     As AP  Sl Sp Ss Ovec
=======================================
1a021000 Eth1/34    0  32  1  9  12 92
```

Forward out Eth1/34!

Unicast + Bridge (L2 lookup) + Destination Known

# Debugging ACI Routed Flows



cisco Live!

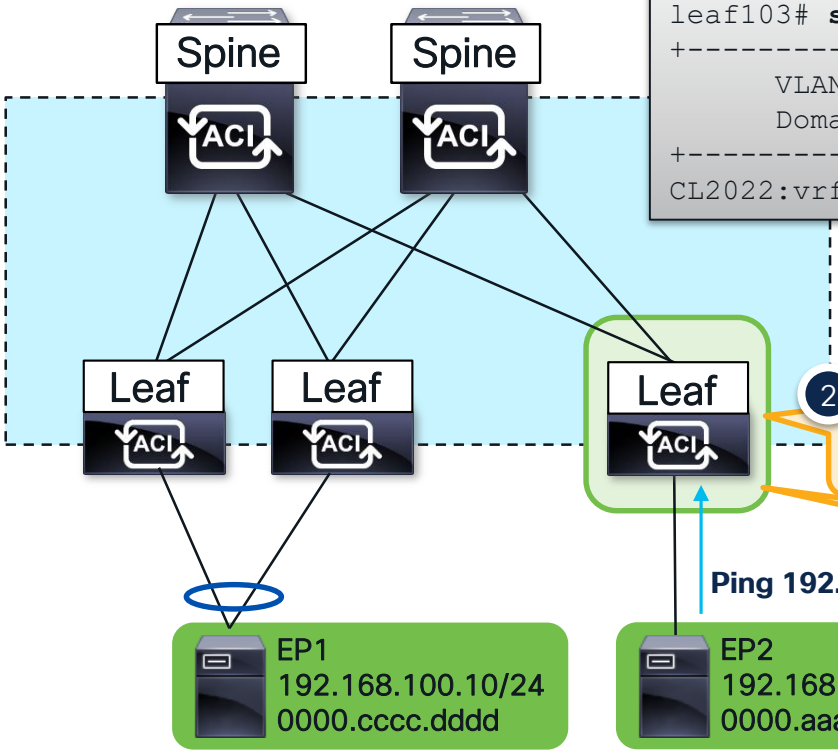
# Known Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled

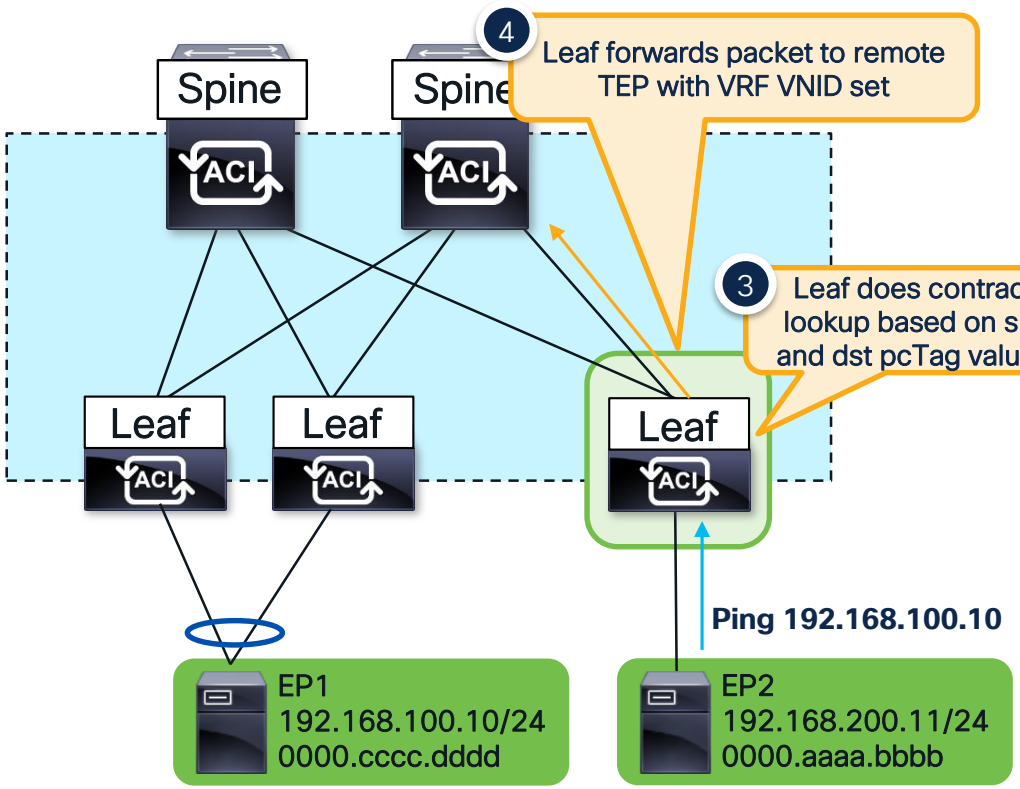
Lookup dst IP in ingress VRF

```
leaf103# show endpoint ip 192.168.100.10
+---------------+-----------------+-------------+
      VLAN/          MAC Address         Interface
      Domain         IP Address
+---------------+-----------------+-------------+
CL2022:vrf1          192.168.100.10       tunnel1
```

```
leaf103# show int tun1
Tunnel destination 10.0.176.67
```



# Known Unicast – Ingress Leaf




Bridge Domain Settings: Unicast Routing Enabled

Get Sclass

```
103# show sys internal epm endpoint ip
192.168.200.11
!omitted
BD vnid : 16613259 ::: VRF vnid : 2523136
sclass : 32771
```

Get Dclass

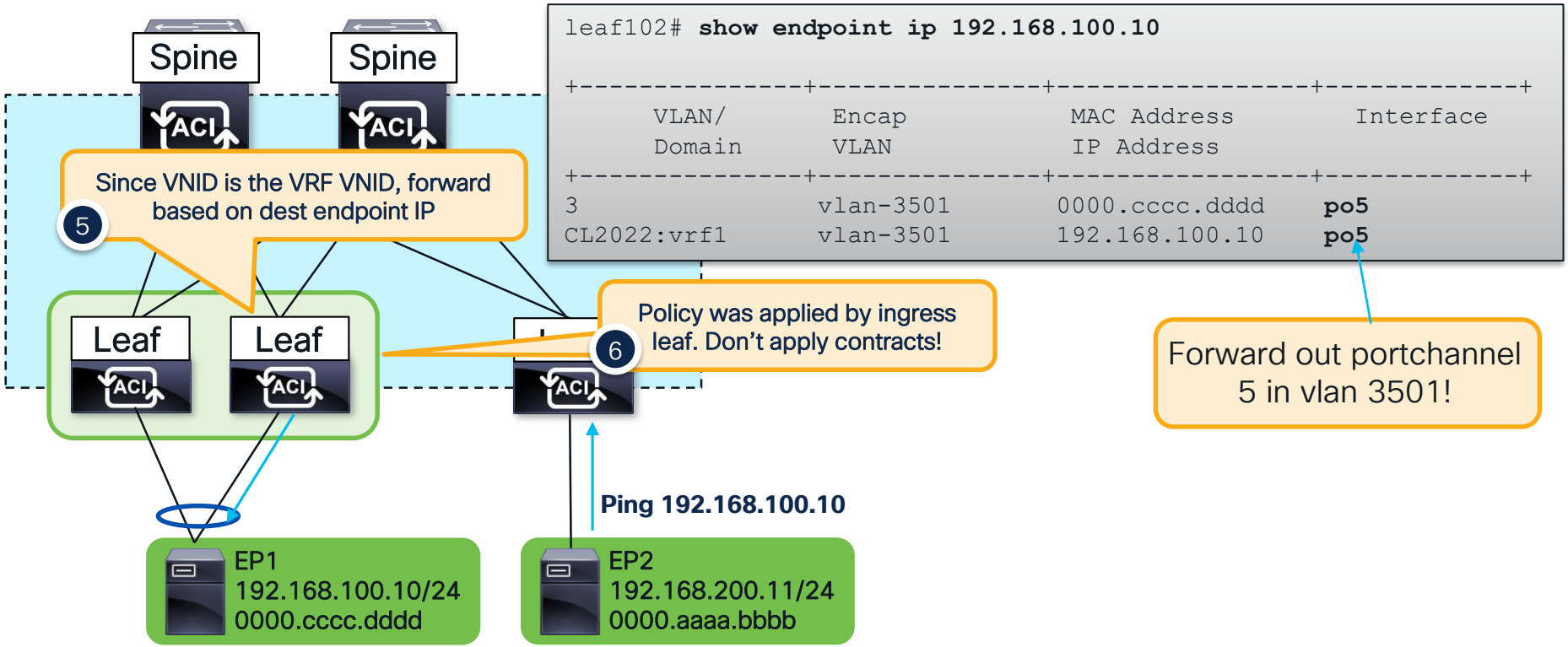
```
103# show sys internal epm endpoint ip
192.168.100.10
!omitted
BD vnid : 0 ::: VRF vnid : 2523136
sclass : 49154
```

Check Contract

```
103# show zoning-rule src-epg 32771
dst-epg 49154 scope 2523136
+---------+------------------+--------+
| RuleID  |       Name       | Action |
+---------+------------------+--------+
|   4209  | CL2022:allow-all | permit |
+---------+------------------+--------+
```

# Known Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



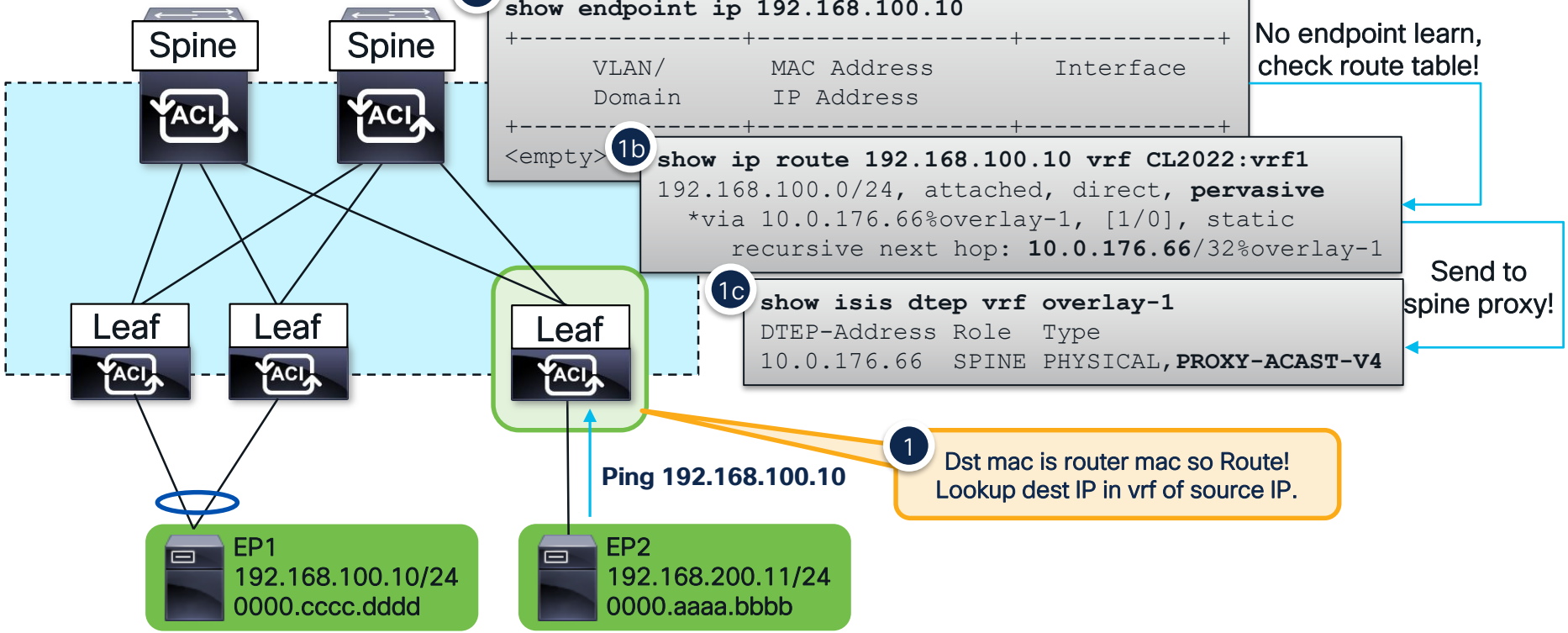

```
leaf102# show endpoint ip 192.168.100.10

+---------------+---------------+-----------------+-------------+
      VLAN/          Encap            MAC Address        Interface
      Domain         VLAN             IP Address
+---------------+---------------+-----------------+-------------+
3                   vlan-3501        0000.cccc.dddd     po5
CL2022:vrf1         vlan-3501        192.168.100.10     po5
```

Forward out portchannel 5 in vlan 3501!

# Proxied Unicast – Ingress Leaf

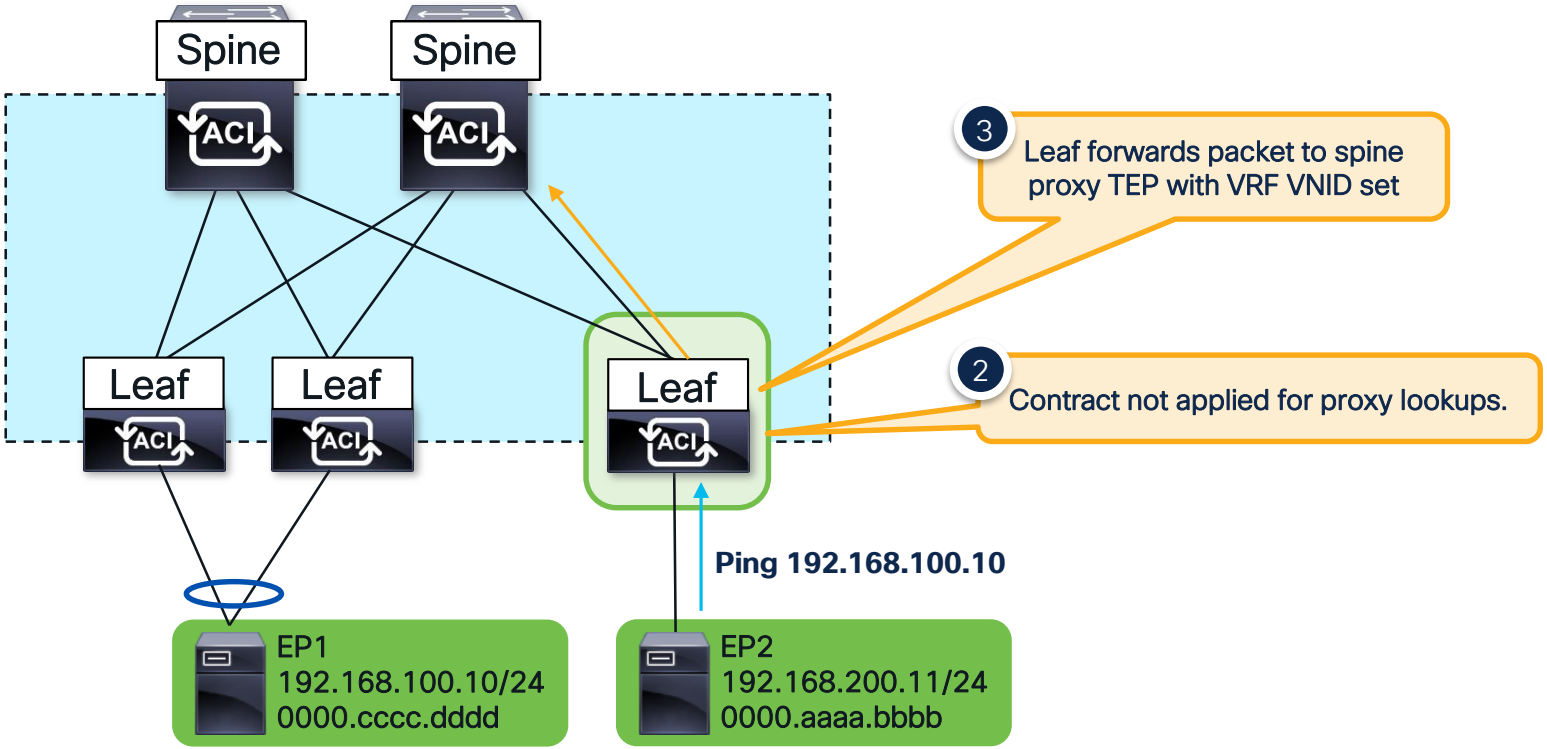
Bridge Domain Settings:
Unicast Routing Enabled

Lookup dst IP in ingress VRF



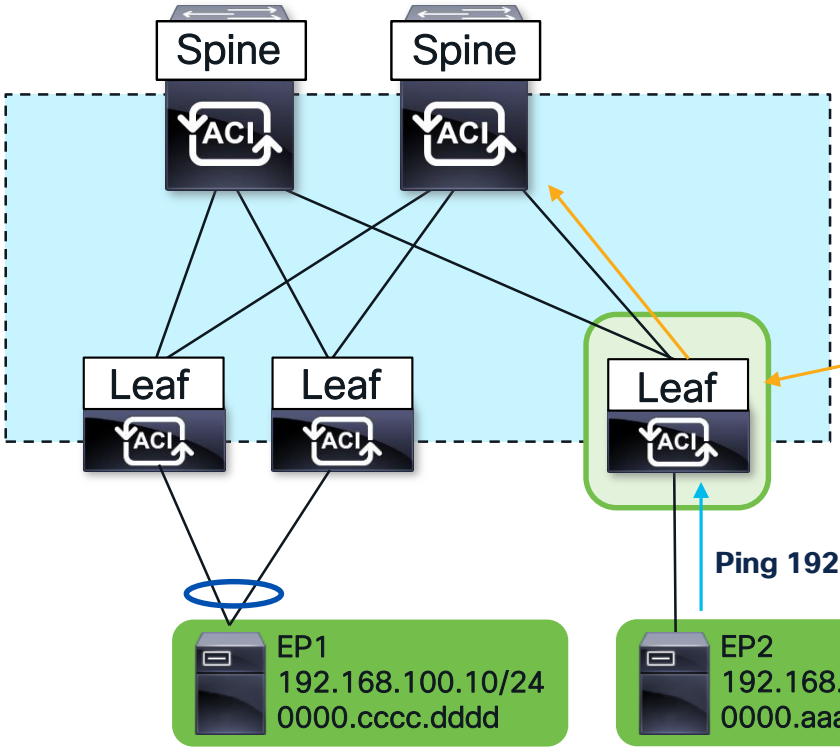
# Proxied Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



# Proxied Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled




ELAM

```
vsh_lc
debug plat internal app elam asic 0
 trigger reset
 trigger init in-select 6 out-select 0
  set outer ipv4 src_ip 192.168.200.11
  set outer ipv4 dst_ip 192.168.100.10
  start
stat
 ELAM STATUS
==========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

# Proxied Unicast – Ingress Leaf

## Forwarding Verifications

Bridge Domain Settings:
Unicast Routing Enabled

```
Outer L2 Header
------------------------------------
Destination MAC     : 0022.BDF8.19FF
Access Encap VLAN   : 3769( 0xEB9 )

Outer L3 Header
------------------------------------
Destination IP                  : 192.168.100.10
Source IP                       : 192.168.200.11

Other Forwarding Information
------------------------------------
Encap Index is valid       : yes
Encap Index                : 1( 0x1 )

FINAL FORWARDING LOOKUP
------------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_IG UC_TENANT MYTEP ROUTE HIT

Lookup Drop
-------------------------
LU drop reason    : no drop
```

ACI Router Mac. Route this packet!

Make sure this is the expected vlan

Dest is tunnel

```
show plat internal hal tunnel rtep apd
=======================================
ifId      IP             RwEncapIdx
=======================================
18010007  10.0.176.66    1
```

Forward to this overlay TEP

Unicast + Route (L3 lookup) + L3 Route Found

Not Dropped in lookups!

# Proxied Unicast – Ingress Leaf

## Forwarding Verifications

Bridge Domain Settings:
Unicast Routing Enabled

```
ereport | grep "ovector "
ovector    : 152( 0x98 )
```

```
show platform internal hal l2 port gpd

=========================================

IfId     Ifname      As AP   Sl Sp Ss Ovec
=========================================
1a01c000 Eth1/29     0  59   2  18 18 98
```

Traffic is forwarded out Eth1/29!

# Proxied Unicast – Ingress Leaf

## Contract Verification

Bridge Domain Settings:
Unicast Routing Enabled

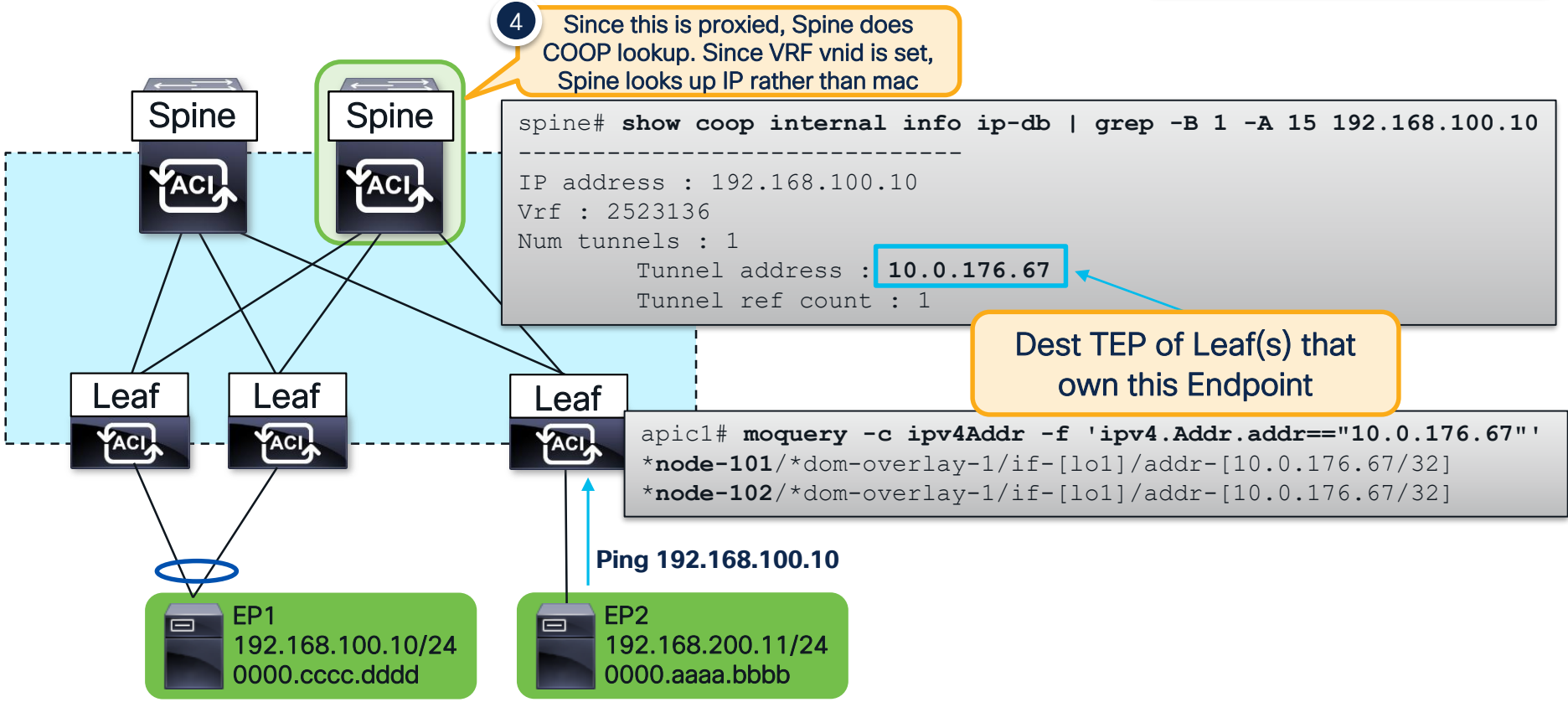
```
Contract Lookup Key
------------------------------------------------
IP Protocol                     : ICMP( 0x1 )
L4 Src Port                     : 2048( 0x800 )
L4 Dst Port                     : 31219( 0x79F3 )
sclass (src pcTag)              : 32771( 0x8003 )
dclass (dst pcTag)              : 1( 0x1 )
src pcTag is from local table   : yes
Unknown Unicast / Flood Packet  : no

Contract Result
------------------------------------------------
Contract Drop                                : no
Contract Applied                             : no
Contract Hit                                 : yes
Contract Aclqos Stats Index                  : 131025
```

Dest EPG is 1 for fabric owned subnets

Contract not applied since this is proxied!

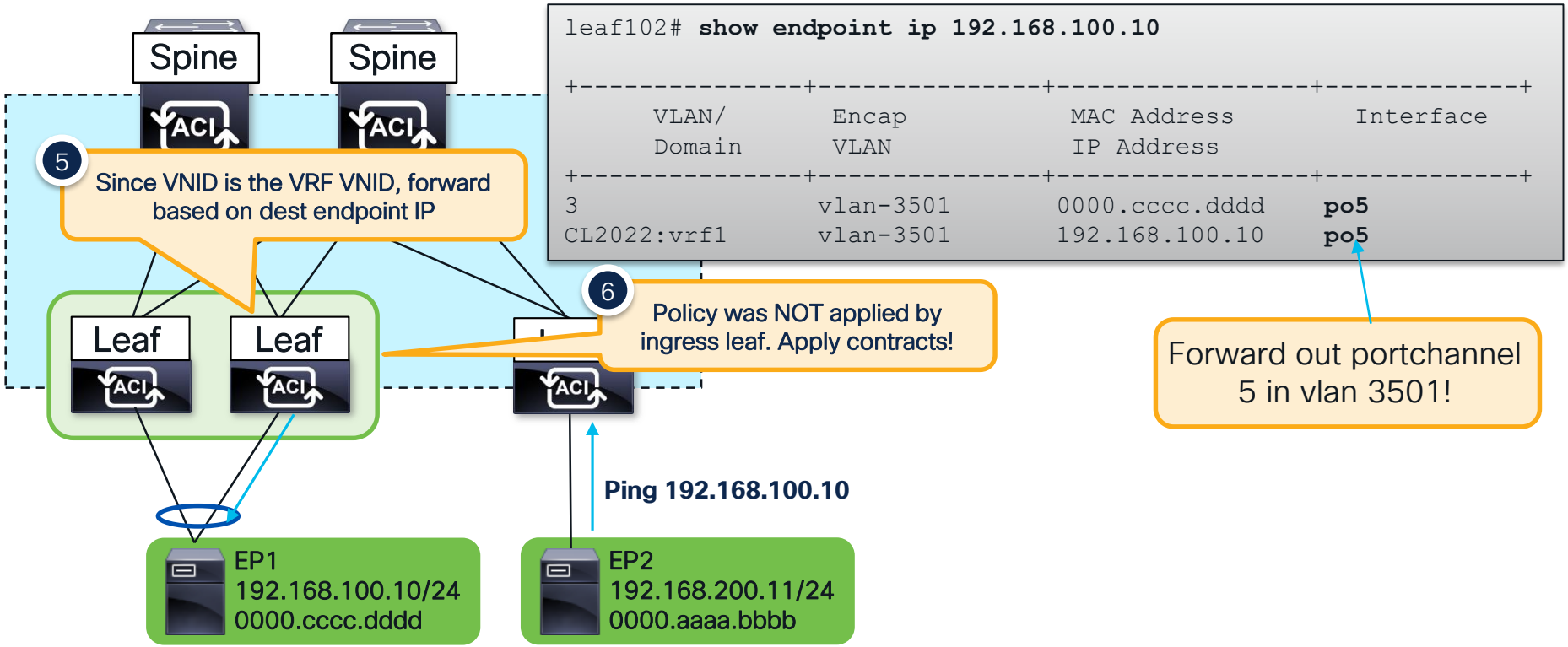
# Proxied Unicast – Spine

Bridge Domain Settings:
Unicast Routing Enabled



# Proxied Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled

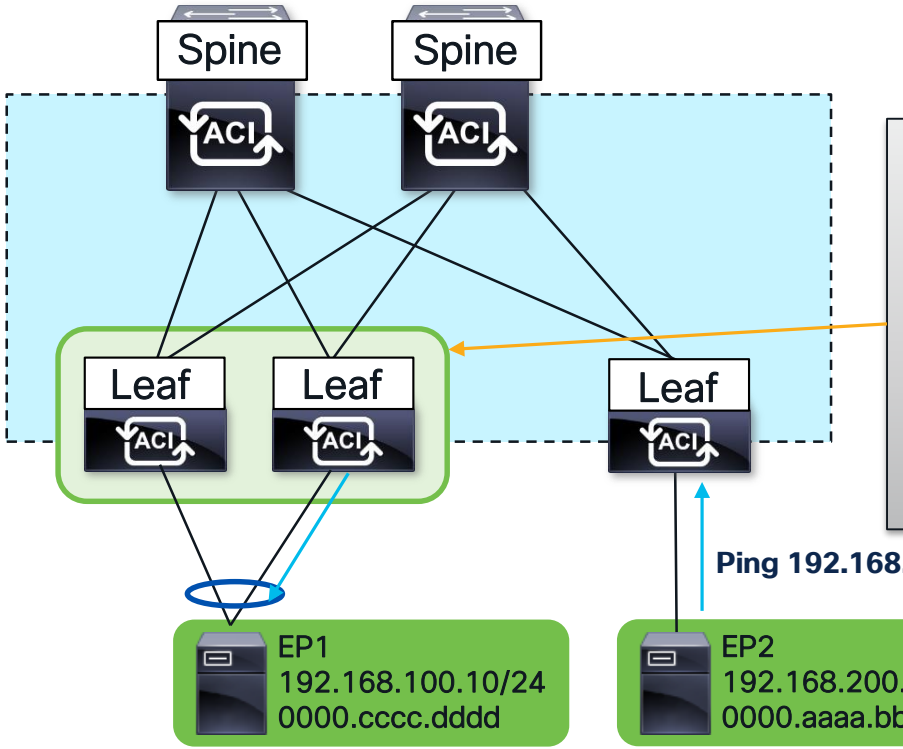



```
leaf102# show endpoint ip 192.168.100.10

+---------------+---------------+-----------------+-------------+
      VLAN/          Encap            MAC Address        Interface
      Domain         VLAN             IP Address
+---------------+---------------+-----------------+-------------+
3                   vlan-3501        0000.cccc.dddd     po5
CL2022:vrf1         vlan-3501        192.168.100.10     po5
```

Forward out portchannel 5 in vlan 3501!

# Proxied Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled




Elam

```
debug plat internal tah elam asic 0
 trigger reset
 trigger init in-select 14 out-select 0
  set inner ipv4 src_ip 192.168.200.11
  set inner ipv4 dst_ip 192.168.100.10
  start
stat
 ELAM STATUS
==========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

**Ping 192.168.100.10**

# Proxied Unicast – Egress Leaf

## Contract Verification

Bridge Domain Settings:
Unicast Routing Enabled

```
Contract Lookup Key
-------------------------------------------------
IP Protocol                     : ICMP( 0x1 )
L4 Src Port                     : 2048( 0x800 )
L4 Dst Port                     : 33226( 0x81CA )
sclass (src pcTag)              : 32771( 0x8003 )
dclass (dst pcTag)              : 49154( 0xC002 )
src pcTag is from local table   : no
Unknown Unicast / Flood Packet  : no

Contract Result
-------------------------------------------------
Contract Drop                                : no
Contract Applied                             : yes
Contract Hit                                 : yes
Contract Aclqos Stats Index                  : 131025
```

Source and Dest EPG used for contract lookup.

Contract Applied and no Drop!

But how do I know which contract this is actually hitting?

# Proxied Unicast – Egress Leaf

## Contract Verification

Bridge Domain Settings:
Unicast Routing Enabled

```
Contract Result
--------------------------------
Contract Drop                  : no
Contract Applied               : yes
Contract Hit                   : yes
Contract Aclqos Stats Index    : 81836
```

Hardware Index of matching contract

Run this from vsh_lc

```
show sys int aclqos zoning-rules | grep -B 9 "Idx: 81836"
=========================================
Rule ID: 4234 Scope 16 Src EPG: 32771 Dst EPG: 49154 Filter
532
============================
    === SDK Info ===
        Result/Stats Idx: 81836
```

Zoning-rule ID

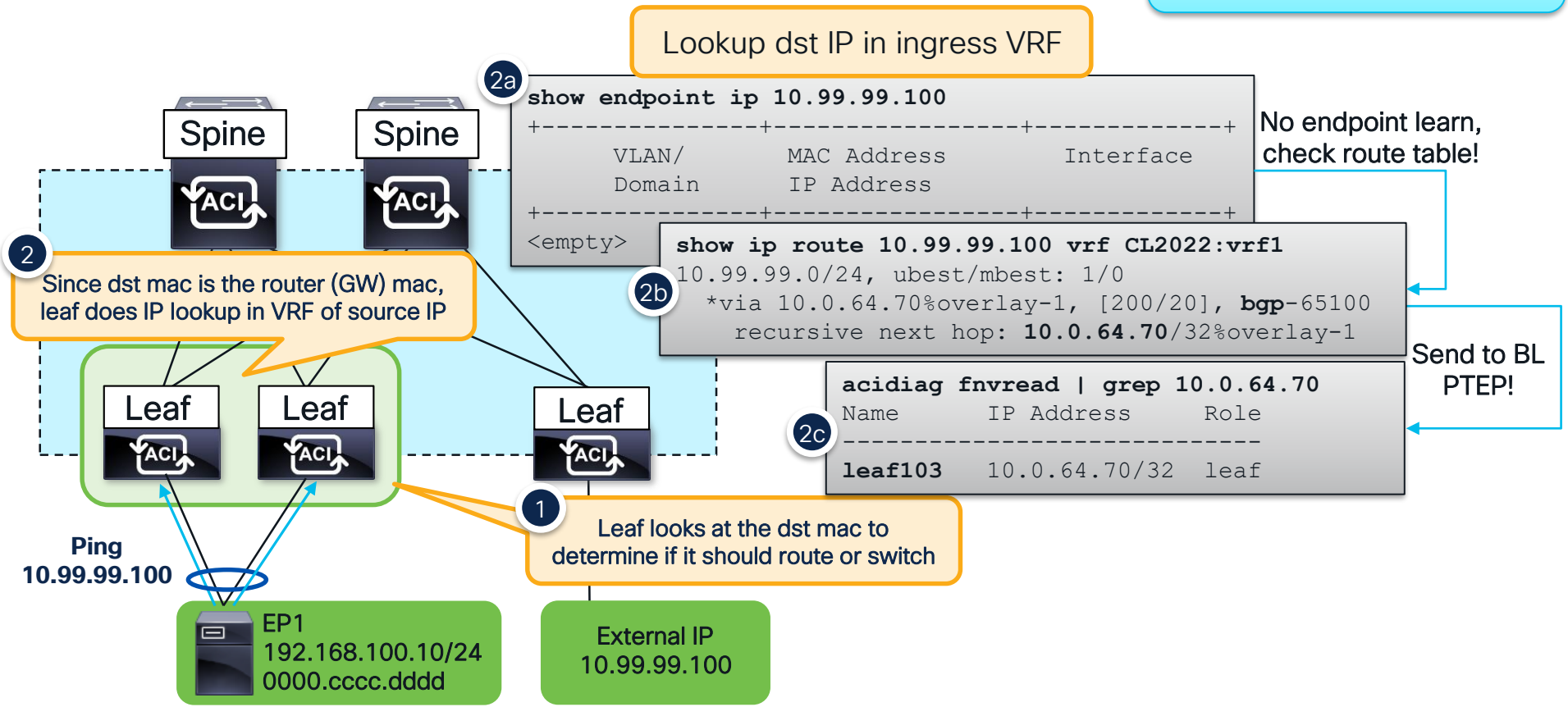
Run this from normal shell

```
show zoning-rule rule-id 4234
+---------+--------+--------+----------+---------+------------------+---
| Rule ID | SrcEPG | DstEPG | FilterID |  Scope  |       Name       | Action |
+---------+--------+--------+----------+---------+------------------+--------+
|   4163  | 32771  | 49154  |   532    | 2523136 | CL2022:allow-all | permit |
+---------+--------+--------+----------+---------+------------------+--------+
```

Traffic hit this contract!

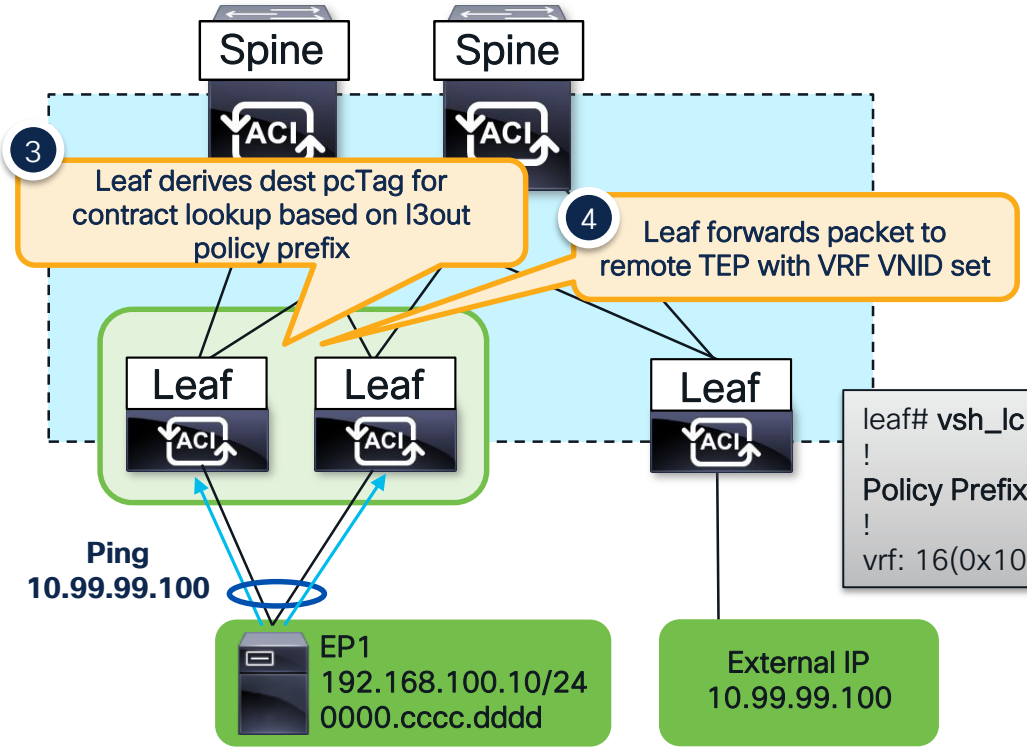
# L3Out Destination – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



# L3Out Destination – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled




External EPGs

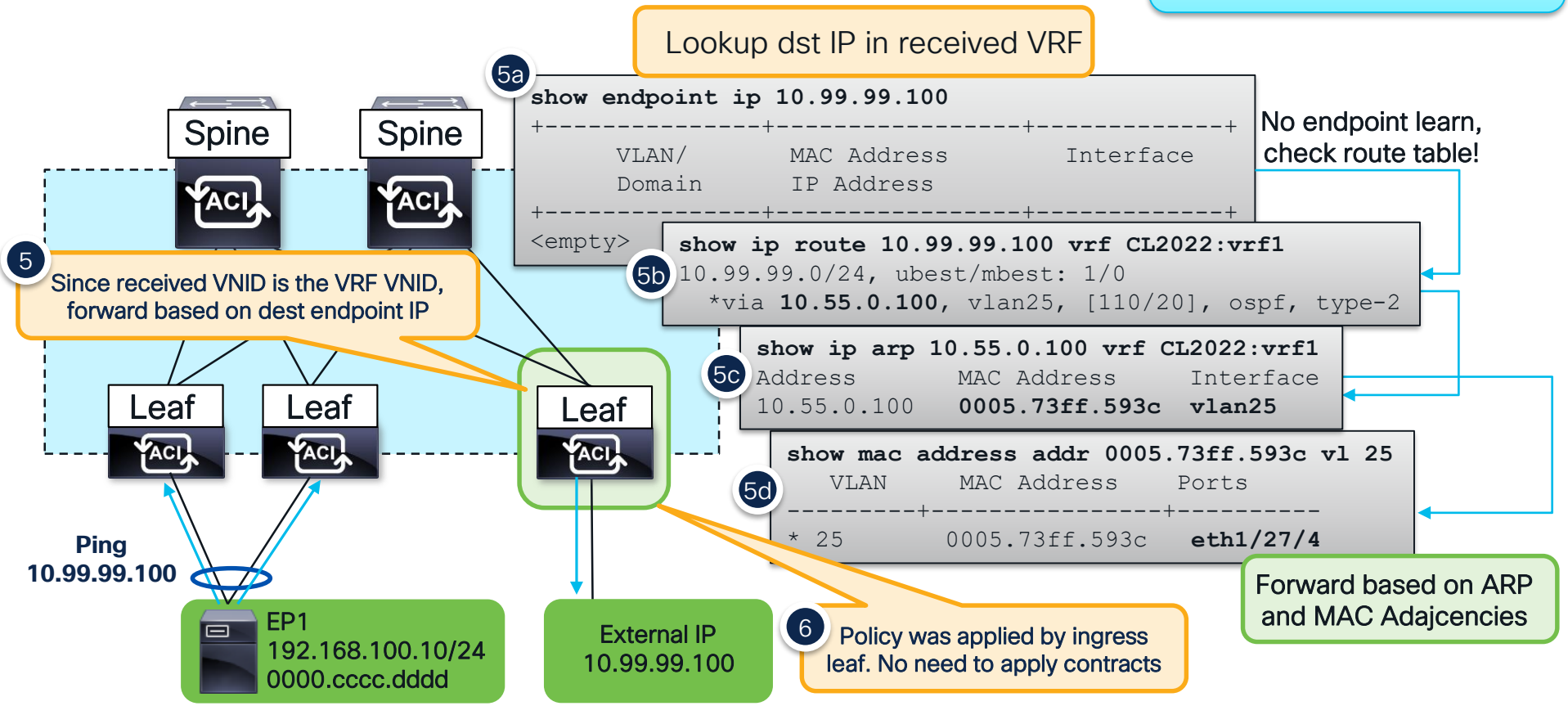
External EPGs

| Name | Description | pcTag |
|---|---|---|
| all | 10.99.99.0/24 Network | 32772 |

```
leaf# vsh_lc -c "show forwarding route 10.99.99.100 platf vrf CL2022:vrf1"
!
Policy Prefix 10.99.99.0/24
!
vrf: 16(0x10), routed_if: 0x0 epc_class: 32772(0x8004)
```

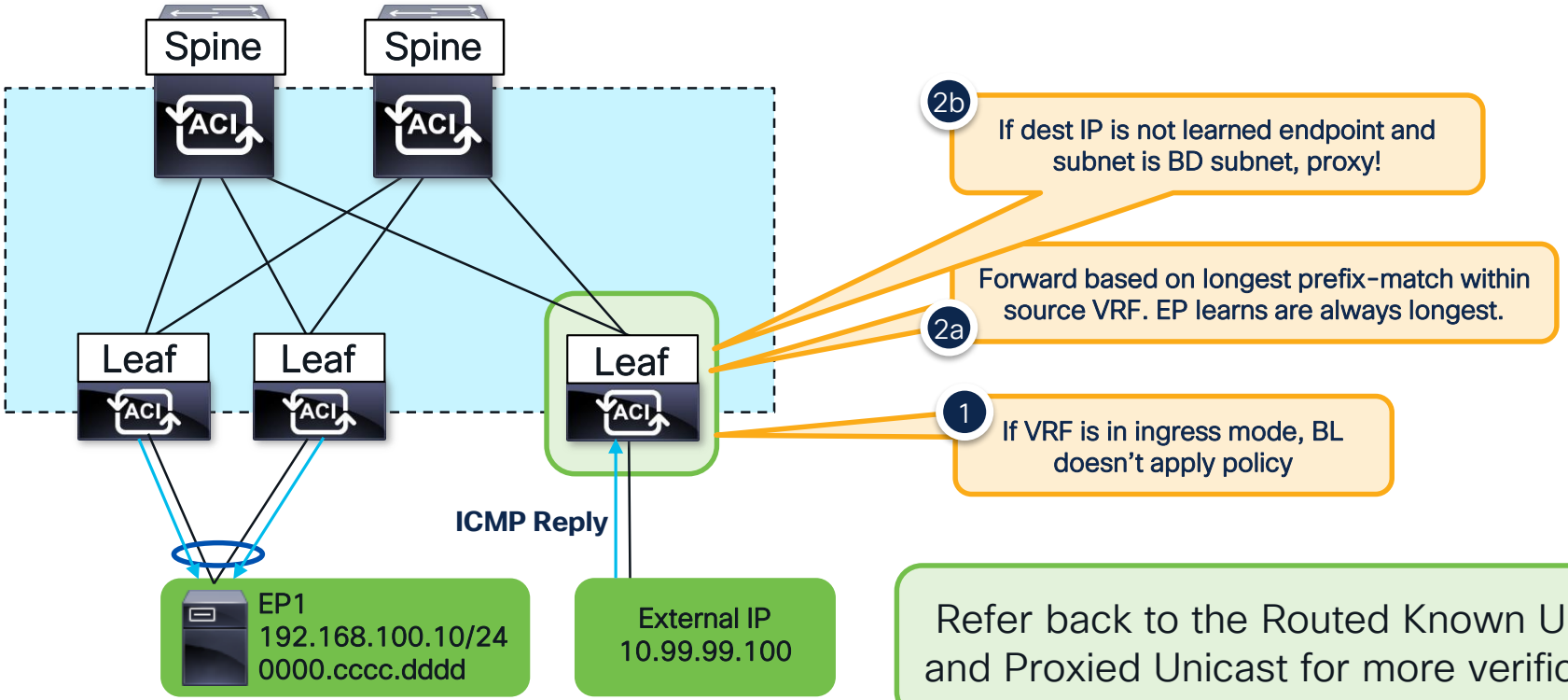
# L3Out Destination – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



# L3Out Source – Ingress Border Leaf

Bridge Domain Settings:
Unicast Routing Enabled




Refer back to the Routed Known Unicast and Proxied Unicast for more verifications

# Complete Your Session Evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to **win 1 of 5 full conference passes** to Cisco Live 2025.


**Earn 100 points** per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn **exclusive prizes!**

Complete your surveys in the **Cisco Live mobile app.**

# Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

# Thank you