
cisco
You make possible

ACI Troubleshooting

Layer 3 Out (L3Out)

Takuya Kishida – Technical Marketing, DCBU ACI

BRKACI-2642

cisco Live!
Barcelona | January 27-31, 2020

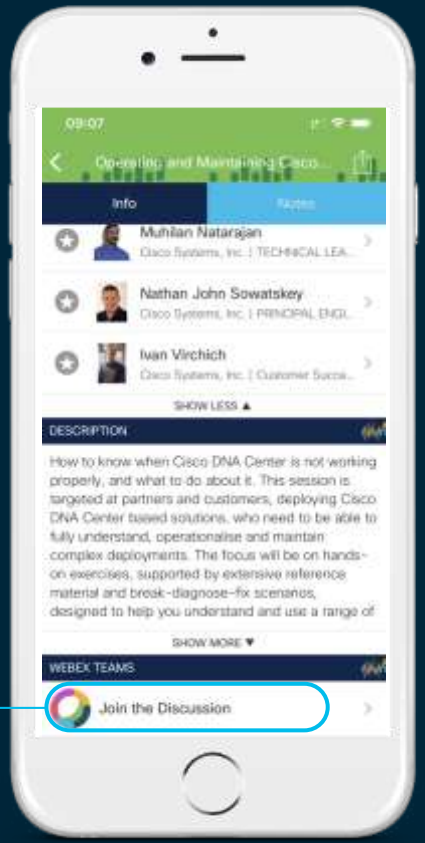
Cisco Webex Teams

Questions?

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

How

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



Agenda

- L3Out Key Components
  - Learning routes (Routing Protocol)
  - Distributing routes within ACI (MP-BGP)
  - Advertising ACI subnet
  - Contract on L3Out (prefix based EPG)
- L3Out Subnet scope options
  - Summary of all options
  - Export Route Control Subnet example

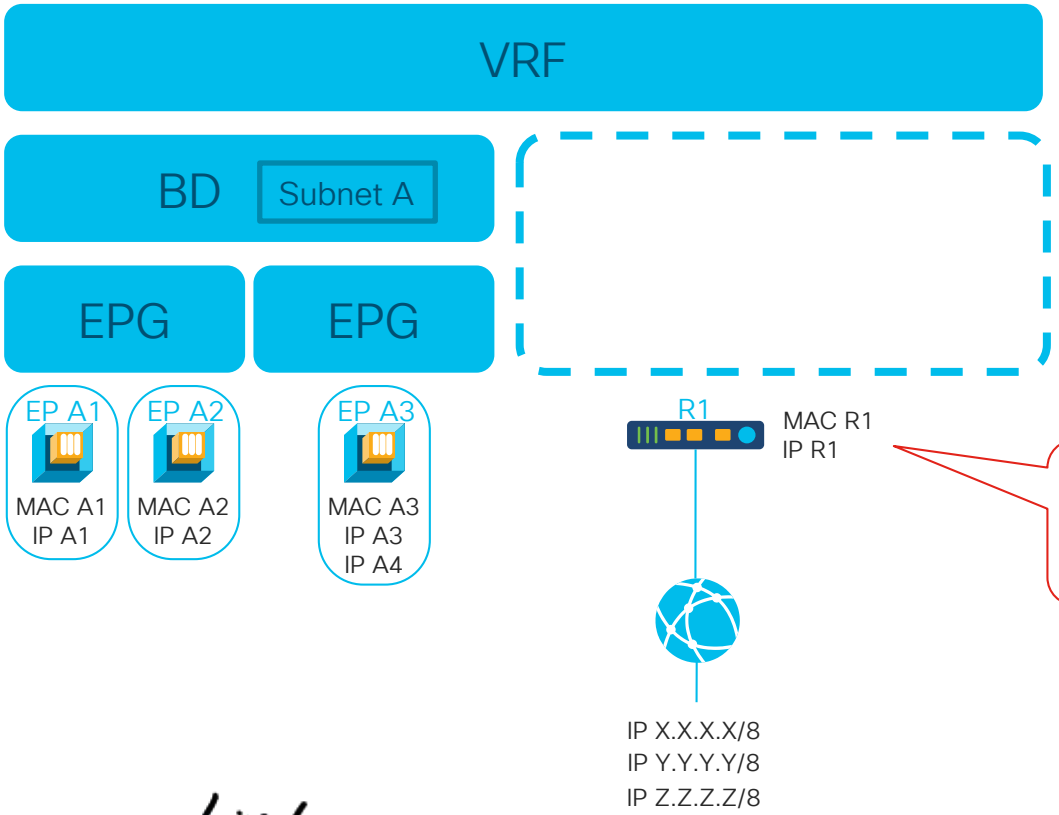
Acronyms/Definitions

Reference Slide 

Acronyms	Definitions
ACI	Application Centric Infrastructure
APIC	Application Policy Infrastructure Controller
EP	Endpoint
EPG	Endpoint Group
BD	Bridge Domain
VRF	Virtual Routing and Forwarding
L3Out	Layer 3 Out (External Routed Network)
L3Out EPG	Layer 3 Out EPG, Prefix Based EPG (External Network Instance)
MP-BGP	Multi Protocol BGP
VPNv4	Virtual Private Network Version 4
RT	Route Target
RD	Route Distinguisher

Why L3Out?

Why L3Out?

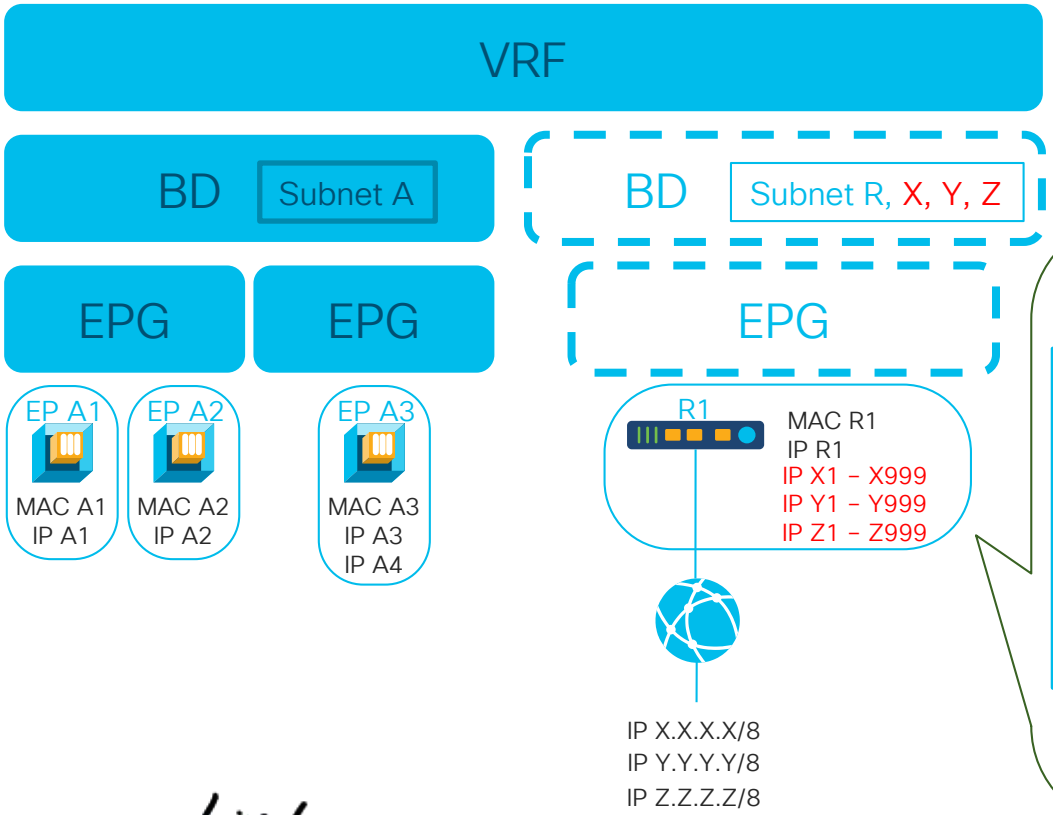


- What is EPG for ?
  - Endpoint (EP) = MAC & /32 IP (or /128)
- What is BD Subnet for ?
  - To be a default gateway
  - For ACI Fabric to know a subnet for EPs in a BD

This is for Spine-Proxy
Please check BRKACI-3545 for details

A network device (ex. router, loadbalancer) as an endpoint?

Why L3Out?



- What is EPG for ?
  - Endpoint (EP) = MAC & /32 IP (or /128)
- What is BD Subnet for ?
  - To be a default gateway
  - For ACI Fabric to know a subnet for EPs in a BD

A network device as an endpoint?

- All IPs as /32 in a single endpoint

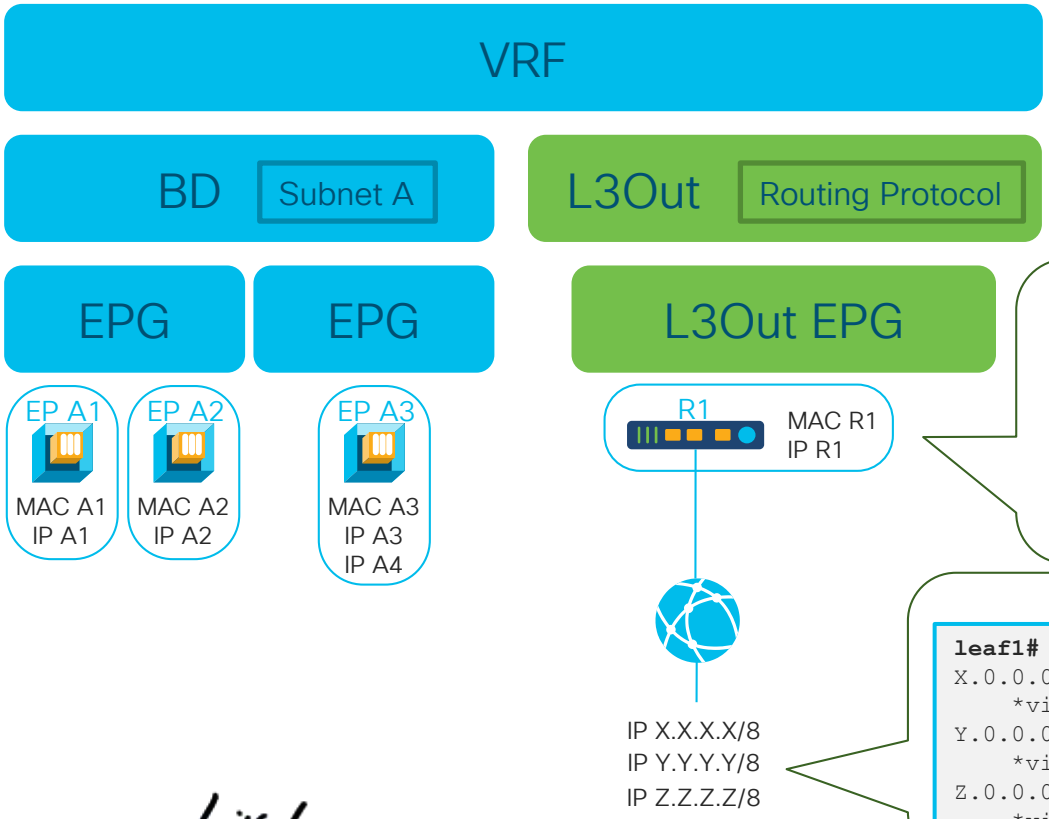
```
leaf1# show endpoint vlan 84
84              vlan-5      0000.0000.R1R1   L        po3
TK:VRF1         vlan-5           R.R.R.1     L        po3
TK:VRF1         vlan-5           X.X.X.1     L        po3
TK:VRF1         vlan-5           X.X.X.2     L        po3
TK:VRF1         vlan-5           X.X.X.3     L        po3
                                     .....
TK:VRF1         vlan-5           Y.Y.Y.1     L        po3
                                     .....
TK:VRF1         vlan-5           Z.Z.Z.1     L        po3
                                     .....
```

※ One endpoint can have up to 1024 IPs in ACI

This does not scale and efficient.
No need to learn each IP as /32.

Why L3Out?

- What is L3Out for ?
  - To connect ACI with other network domain
    = devices with multiple subnet behind it
- How is L3Out different from EPG?
  - Speak Routing Protocol
  - No IP learning as endpoint
  - Next-hop IP is stored in ARP table
    = Same as normal routers



Next-hop MAC in endpoint table

```
leaf1# show endpoint vlan 84
84/TK:VRF1    vxlan-14876665    0000.0000.R1R1    L    po3
```

Next-hop IP in ARP table (only for L3Out)

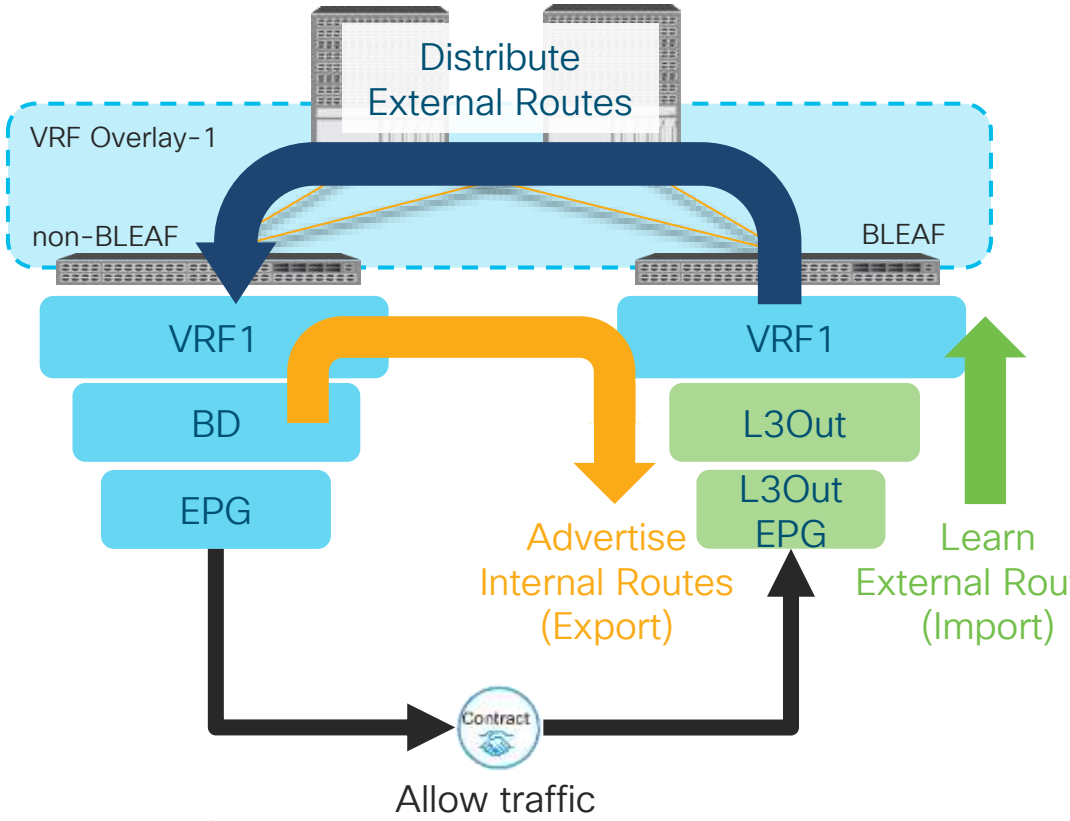
```
leaf1# show ip arp vlan 84
Address          Age        MAC Address     Interface
R.R.R.1          00:07:51   0000.0000.R1R1  vlan84
```

Other routes via Routing Protocol

```
leaf1# show ip route vrf TK:VRF1
X.0.0.0/8, ubest/mbest: 1/0
    *via R.R.R.1, vlan84, [110/5], 2d00h, ospf-default, intra
Y.0.0.0/8, ubest/mbest: 1/0
    *via R.R.R.1, vlan84, [110/5], 2d00h, ospf-default, intra
Z.0.0.0/8, ubest/mbest: 1/0
    *via R.R.R.1, vlan84, [110/5], 2d00h, ospf-default, intra
```

L3Out
Key Components

L3Out Key Components



1. Learn external routes
   - Routing Protocol in L3Out
2. Distribute external routes to other leaves
   - MP-BGP
3. Advertise internal routes (BD subnet) to outside
   - Redistribution
     and
   - Contract
4. Allow traffic with contracts
   - L3Out EPG (Prefix Based EPG)
     and
   - Contract

L3Out Key Components

1. Learn External Routes = Routing Protocol

Configurations

External Routed Networks (L3Out)

- VRF to deploy Routing Protocol
- Routing Protocol parameters
  ex. OSPF area 0.0.0.1 nssa

Node Profile

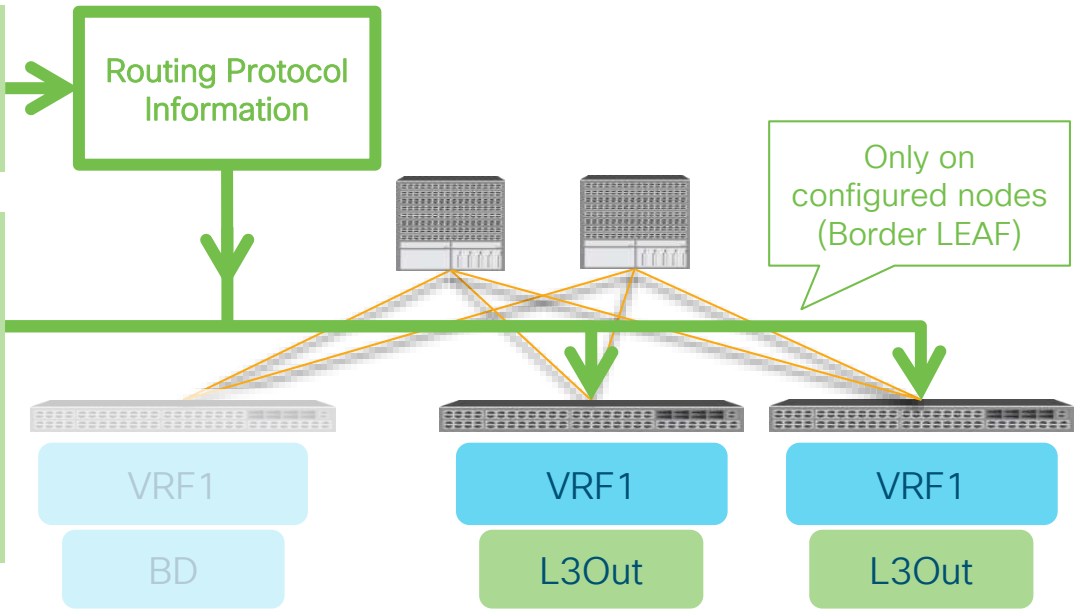
- Node(s) to deploy Routing Protocol
- Static Route (if any)

Interface Profile

- I/F(s) to deploy Routing Protocol
- Routing Protocol I/F parameters
  ex. OSPF hello interval

Networks (L3Out EPG)

- Contract
- Advanced Route Control
  ex. route-map



※ Details for L3Out EPG are in later sections

Verification Examples (OSPF)

1. Is OSPF enabled on a correct I/F?

```
border-leaf# show ip ospf int bri vrf TK:VRF1
 Interface              ID     Area            Cost    State    Neighbors Status
 Vlan58                 134    backbone        4       BDR      2         up

border-leaf# show vlan id 58 extended
 VLAN Name                                Encap            Ports
 ---- ----------------------------------- ---------------- ----------------------
 58   TK:VRF1:l3out-                      vxlan-15695748,  Eth1/3, Po2
      L3OUT_OSPF:vlan-1425                vlan-1425
```

Same CLI verifications are as useful in ACI too

If anything is not as expected, check config or any faults in APIC GUI.

2. Are OSPF parameters matching with neighbors?

```
border-leaf# show int vlan 58 | grep MTU
  MTU 1500 bytes, BW 10000000 Kbit, DLY 1 usec

border-leaf# show ip ospf int vlan 58 | egrep 'IP|State|Timer|auth'
    IP address 15.0.0.3/24, Process ID default VRF TK:VRF1, area backbone
    State BDR, Network type BROADCAST, cost 4
    Timer intervals: Hello 10, Dead 40, Wait 40, Retransmit 5
    No authentication
```

Is MTU matching?
Is Network Mask matching?
Is Area matching?
Is Timer matching?
Is Network Type expected?
etc.

3. Are OSPF neighbors established correctly?

```
border-leaf# show ip ospf neighbors vrf TK:VRF1
Neighbor ID     Pri State            Up Time  Address         Interface
 4.4.4.4          1 FULL/DR          2d06h    15.0.0.4        Vlan58
 9.9.9.9          1 FULL/DROTHER     2d06h    15.0.0.1        Vlan58
```

Can they ping to each other?
leaf# iping –V <VRF> <target IP>
※OSPF DBD requires unicast reachability
etc.

Verification Examples (EIGRP)

1. Is EIGRP enabled on a correct I/F?

```
border-leaf# show ip eigrp int bri vrf TK:VRF1
                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
vlan92             2        0/0        1       0/0          50           0

border-leaf# show vlan id 92 extended
 VLAN Name                             Encap            Ports
 ---- -------------------------------- ---------------- ----------------------
 92   TK:VRF1:l3out-                   vxlan-14712828,  Eth1/3, Po2
      L3OUT_EIGRP:vlan-1426            vlan-1426
```

Same CLI verifications are as useful in ACI too

If anything is not as expected, check config or any faults in APIC GUI.

2. Are EIGRP parameters matching with neighbors?

```
border-leaf# show int vlan 92 | grep MTU
  MTU 1500 bytes, BW 10000000 Kbit, DLY 1 usec

border-leaf# show ip int vlan 92 | grep 'IP addr'
  IP address: 16.0.0.3, IP subnet: 16.0.0.0/24
```

```
border-leaf# show ip eigrp vrf TK:VRF1 | egrep 'AS|K'
IP-EIGRP AS 1 ID 3.3.3.3 VRF TK:VRF1
   Metric weights: K1=1 K2=0 K3=1 K4=0 K5=0
```

Is MTU matching?
Is Network Mask matching?
Is AS matching?
Is K value matching?
etc.

3. Are EIGRP neighbors established correctly?

```
border-leaf# show ip eigrp neighbors vrf TK:VRF1
H   Address                 Interface       Hold  Uptime   SRTT   RTO  Q  Seq
                                            (sec)          (ms)       Cnt Num
0   16.0.0.4                vlan92          12    2d06h     1     50    0   10
1   16.0.0.1                vlan92          13    2d06h     1     50    0   346
```

Verification Examples (BGP)

1. Is BGP neighbor session configured as expected?

```
border-leaf# show ip bgp neighbors vrf TK:VRF1 | egrep 'BGP nei|Using|Opens|hops'
BGP neighbor is 17.0.0.1,  remote AS 65001, ebgp link,  Peer index 1
  Using Loopback6 as update source for this peer
  External BGP peer might be upto to 2 hops away
  Opens:                          1                   1

border-leaf# show ip int lo6 | grep 'IP addr'
  IP address: 3.3.3.3, IP subnet: 3.3.3.3/32
```

Is it correct remote AS?
Is it using correct source I/F with correct IP?
Is enough multi-hop configured for eBGP?
Is Open message exchanged?

2. Is there IP reachability ?

```
border-leaf# iping -V TK:VRF1 17.0.0.1 -S 3.3.3.3
PING 17.0.0.1 (17.0.0.1) from 3.3.3.3: 56 data bytes
64 bytes from 17.0.0.1: icmp_seq=0 ttl=255 time=0.76 ms
64 bytes from 17.0.0.1: icmp_seq=1 ttl=255 time=0.639 ms
   === snip ===

--- 17.0.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
```

Is there an IP reachability to the BGP neighbor from the correct source IP?

3. Are BGP neighbors established correctly?

```
border-leaf# show ip bgp summary vrf TK:VRF1
BGP router identifier 3.3.3.3, local AS number 65003

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
17.0.0.1        4 65001    3300    3302       78    0    0    2d06h 2
```

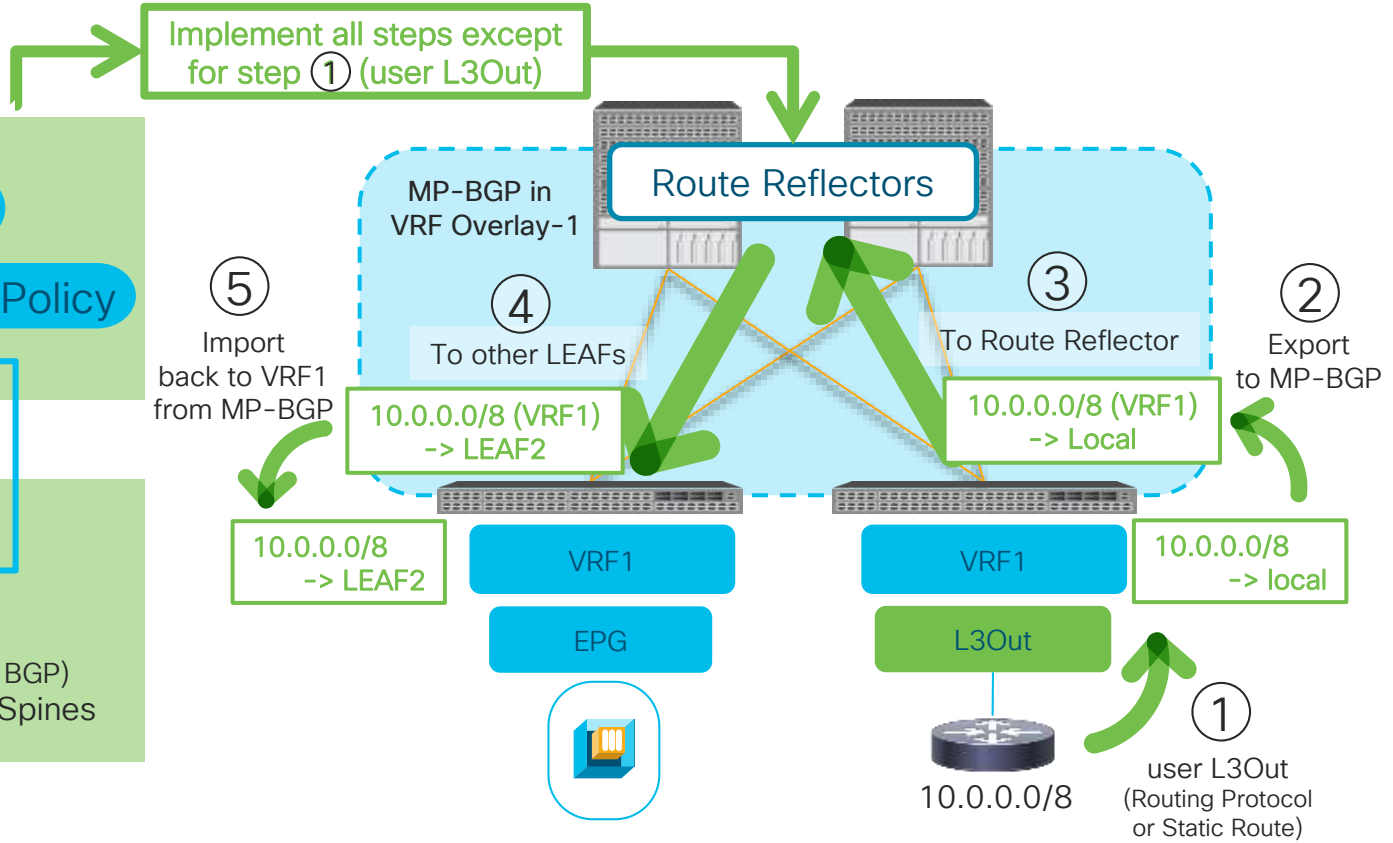
Is it receiving BGP routes?
Is ACI BGP using expected local AS?

L3Out Key Components

2. Distribute External Routes = MP-BGP in infra

Configurations

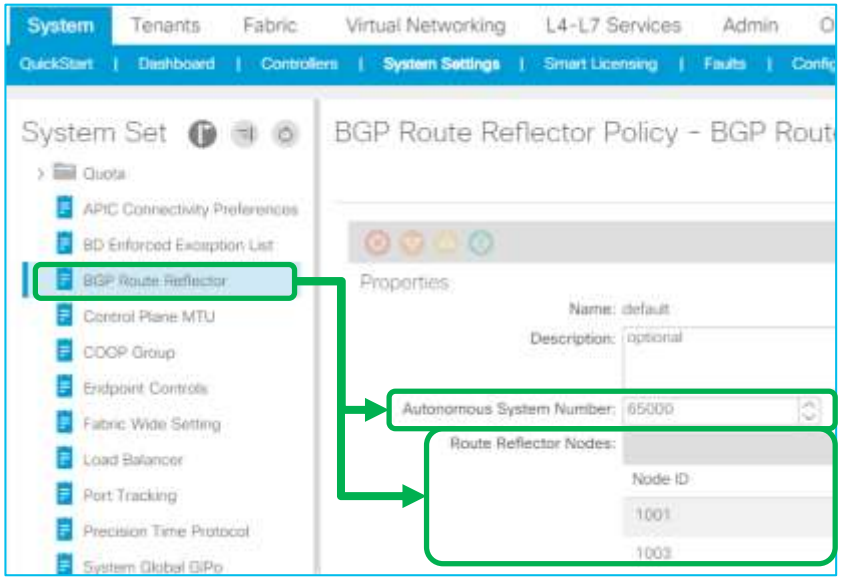
- Pod Profile
- Pod Policy Group
- BGP Route Reflector Policy
  - default
- System Settings
- BGP Route Reflector
  - ACI BGP AS number
    (for both MP-BGP and L3Out BGP)
  - MP-BGP Route Reflector Spines



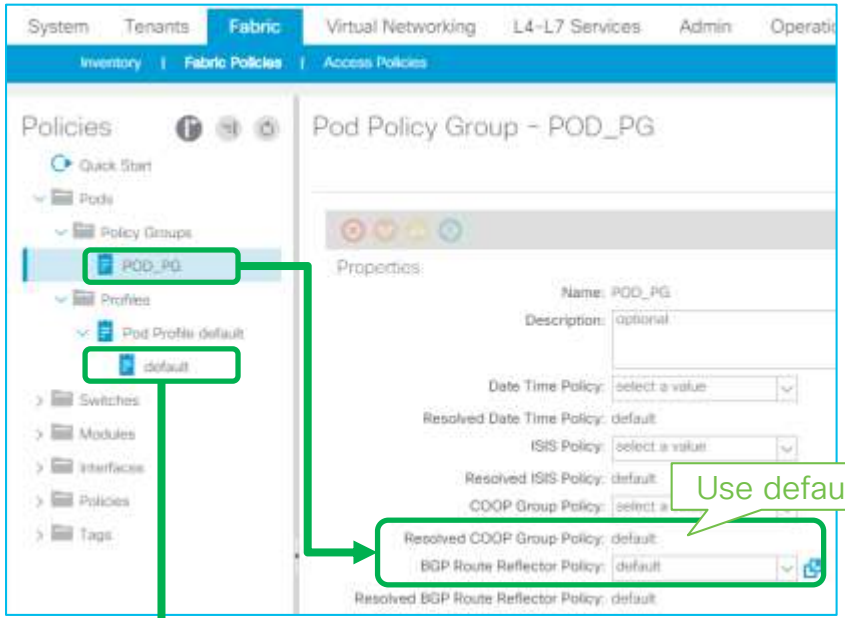
L3Out Key Components

2. Distribute External Routes = MP-BGP in infra

1. Select ACI BGP AS and Route Reflector SPINEs



2. Apply Route Reflector policy to Pod Policy Group



3. Apply Pod Policy Group to Pod Profile

※ L3Out BGP share this same AS with the internal MP-BGP

※ Check appendix for MP-BGP details

CLI Verification

1. Do both border leaf and non-border leaf have BGP sessions with RR spines?

```
leaf# show bgp sessions vrf overlay-1
Neighbor        ASN     Flaps LastUpDn|LastRead|LastWrit St Port(L/R)  Notif(S/R)
10.0.184.65      65003  0     2d07h   |never   |never    E  37850/179  0/0
10.0.184.66      65003  0     2d07h   |never   |never    E  45089/179  0/0

leaf# acidiag fnvread | grep spine
    1001        1       spine1      FGE10000000     10.0.184.65/32   spine         active   0
    1002        1       spine2      SAL10000000     10.0.184.66/32   spine         active   0
```

✓ BGP neighbors are RR spines TEP IPs

2. Is the external route learned on a border leaf?

```
border-leaf# show ip route vrf TK:VRF1
10.0.0.0/8, ubest/mbest: 1/0
    *via 15.0.0.1, Vlan58, [110/5], 2d08h, ospf-default, intra
```

3. Does non-border leaf show the expected border leaf as next-hop?

✓ Next-hops are border Leaf TEP IPs
✓ Learned via iBGP in ACI AS# (65003)

```
non-border-leaf# show ip route vrf TK:VRF1
10.0.0.0/8, ubest/mbest: 2/0
    *via 10.0.184.67%overlay-1, [200/5], 2d08h, bgp-65003, internal, tag 65003
    *via 10.0.184.64%overlay-1, [200/5], 2d08h, bgp-65003, internal, tag 65003

non-border-leaf# acidiag fnvread
      ID    Pod ID                 Name     Serial Number         IP Address     Role        State
LastUpdMsgId
--------------------------------------------------------------------------------------------------
     103        1                 leaf3       SAL10000003      10.0.184.64/32     leaf         active   0
     104        1                 leaf4       SAL10000004      10.0.184.67/32     leaf         active   0
```

L3Out Key Components

3. Advertise BD subnet

Configurations

Bridge Domain (BD)

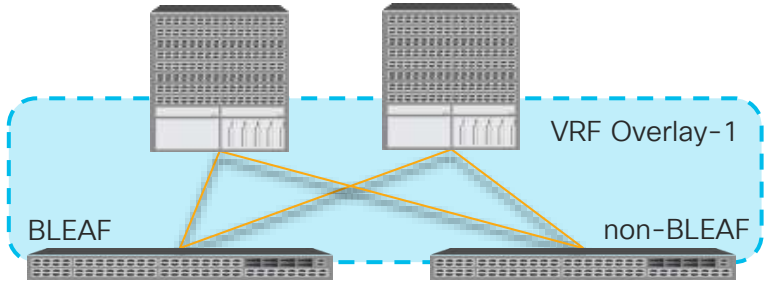
BD Subnet

- Subnet A
  - ✓ "Advertised Externally"

Associated L3Out

- Target L3Out(s) to advertise BD subnets

Redistribution
Direct (Subnet A)
-> L3Out Protocol



VRF1

L3Out

L3Out EPG

VRF1

Subnet A BD

EPG

No BD Subnet A on BLEAF yet
- MP-BGP is only to distribute external routes
- MP-BGP never distributes BD subnets

L3Out Key Components

3. Advertise BD subnet

Configurations

Bridge Domain (BD)

BD Subnet

- Subnet A
  - ✓ “Advertised Externally”

Associated L3Out

- Target L3Out(s) to advertise BD subnets

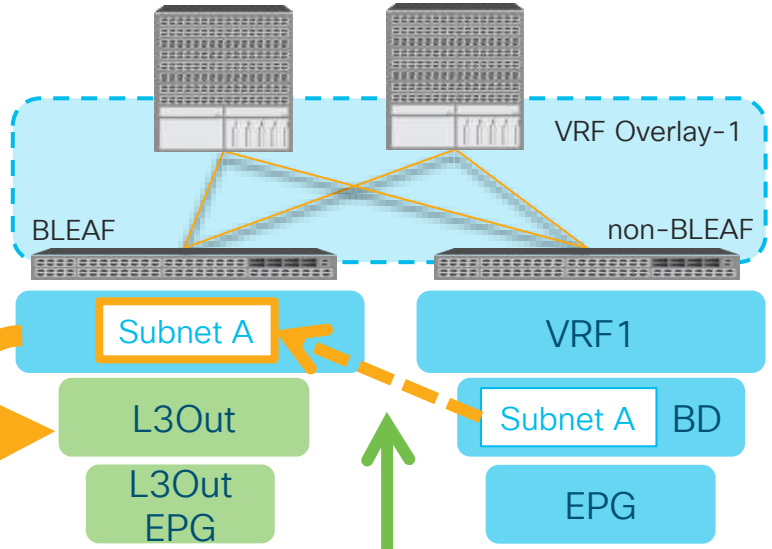
External Routed Networks (L3Out)

Networks (L3Out EPG)

- Contract to EPG

Redistribution
Direct (Subnet A)
-> L3Out Protocol

Static Route (subnet A)
on BLEAF via MO (object)

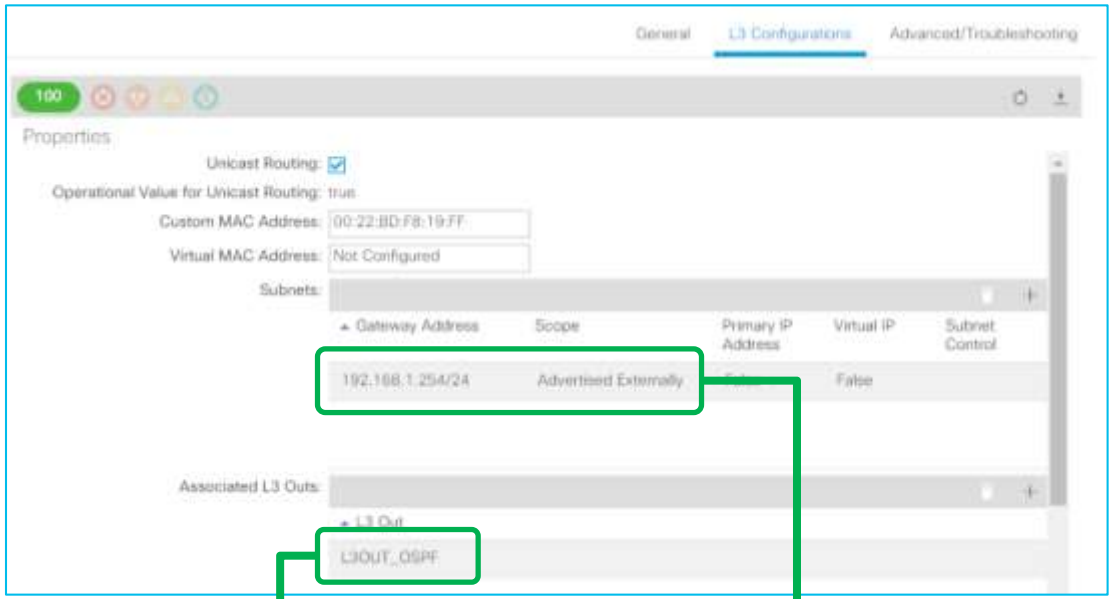


Pushed by APIC. Not MP-BGP.
Please check “pervasive gateway”
in BRKACI-3545

L3Out Key Components

3. Advertise BD subnet

1. L3Out Association from BD (for redistribution)

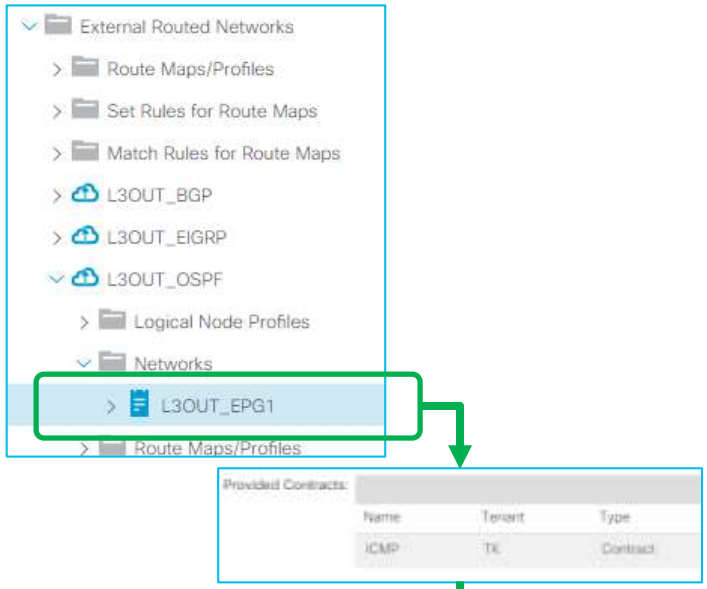



```
border-leaf# show ip ospf vrf TK:VRF1

 Redistributing External Routes from
   direct route-map exp-ctx-st-2326530
```

```
ip prefix-list
    ➢ 192.168.1.0/24
```

2. Contract




```
border-leaf# show ip route vrf TK:VRF1
192.168.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 10.0.184.64%overlay-1, [1/0], 04:32:27, static
```

Make sure the other end of contract is configured correctly as well

CLI Verification (OSPF, EIGRP)

1. Does the border leaf have BD subnet to advertise?

```
border-leaf# show ip route vrf TK:VRF1
192.168.1.0/24, ubest/mbest: 1/0, attached, direct,
pervasive
    *via 10.0.184.64%overlay-1, [1/0], 04:32:27, static
```

If not, check the contract between the L3Out EPG and the EPG for the BD.
This should be pushed by APIC. Not via MP-BGP.

2. Check a route-map name used by the routing protocol on the border leaf for redistribution

```
border-leaf# show ip ospf vrf TK:VRF1

 Redistributing External Routes from
   direct route-map exp-ctx-st-2097152
```

```
border-leaf# show ip eigrp vrf TK:VRF1

  Redistributing:
    direct route-map exp-ctx-st-2097152
```

Check next page for BGP

3. Does the route-map have expected BD subnet?

```
border-leaf# show route-map exp-ctx-st-2097152
route-map exp-ctx-st-2097152, deny, sequence 1
  Match clauses:
    tag: 4294967295
  Set clauses:
route-map exp-ctx-st-2097152, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-st49158-2097152-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
```

IP prefix-list should have the BD subnet.
If not, check APIC config and any faults.

- ✓ Is “Advertise Externally” on the BD subnet checked?
- ✓ Is L3Out associated to the BD?

```
border-leaf# show ip prefix-list IPv4-st49158-2097152-exc-int-inferred-export-dst
ip prefix-list IPv4-st49158-2097152-exc-int-inferred-export-dst: 1 entries
   seq 1 permit 192.168.1.254/24
```

CLI Verification (BGP)

1. Does the border leaf have BD subnet to advertise?

```
--- snip ---
```

2. Check a route-map name used by BGP outbound rule for each neighbor

```
border-leaf# show bgp process vrf TK:VRF1
 Information for address family IPv4 Unicast in VRF TK:VRF1
    Redistribution
        direct, route-map permit-all
```

BGP redistributes all direct routes first,
then limit the routes with an outbound route-map.

```
border-leaf# show ip bgp neighbors vrf TK:VRF1 | egrep '^BGP|Out'
BGP neighbor is 17.0.0.1,  remote AS 65001, ebgp link,  Peer index 1
  Outbound route-map configured is exp-l3out-L3OUT_BGP-peer-2097152, handle obtained
```

3. Does the BGP outbound route-map have the expected BD subnet?

```
border-leaf# show route-map exp-l3out-L3OUT_BGP-peer-2097152
route-map exp-l3out-L3OUT_BGP-peer-2097152, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-peer49157-2097152-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
route-map exp-l3out-L3OUT_BGP-peer-2097152, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
```

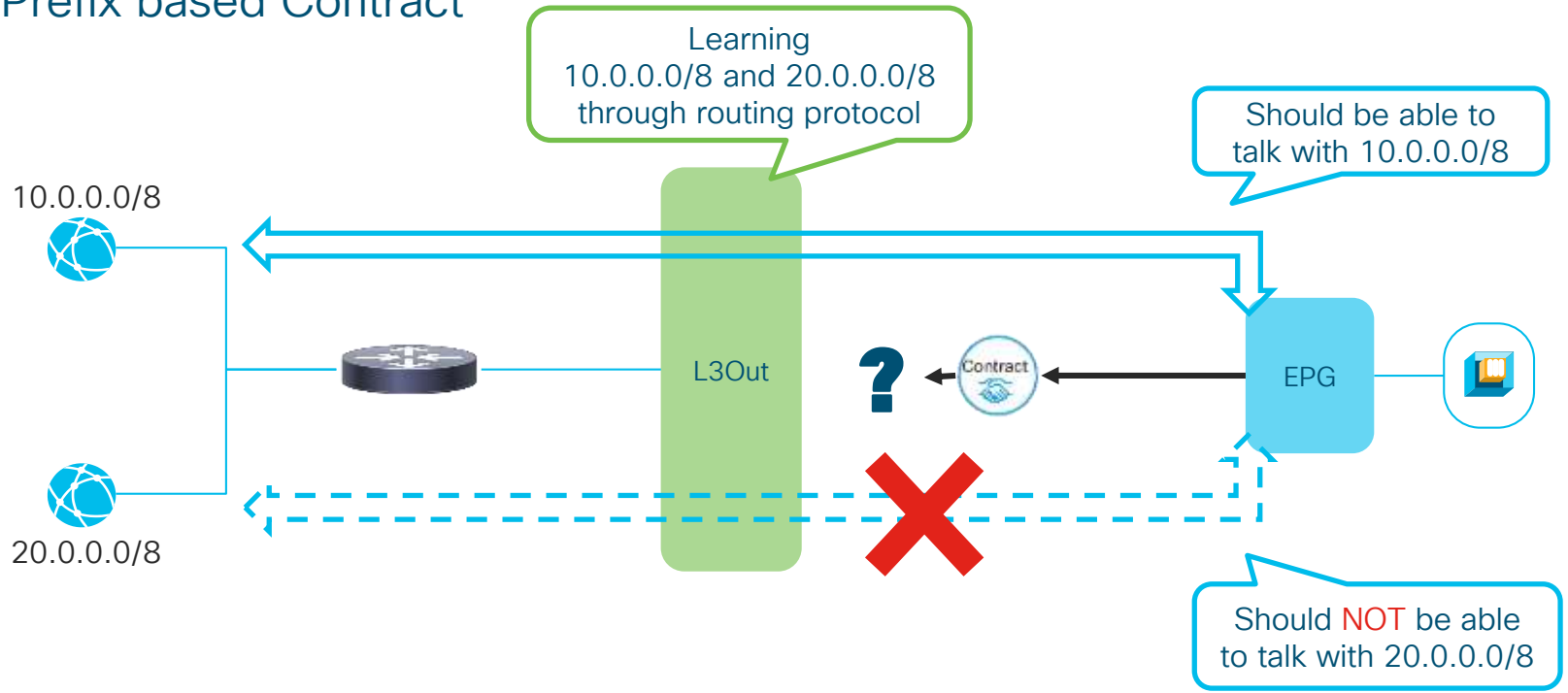
IP prefix-list should have the BD subnet.
If not, check APIC config and any faults.

- ✓ Is “Advertise Externally” on the BD subnet checked?
- ✓ Is L3Out associated to the BD?

```
border-leaf# show ip prefix-list IPv4-peer49157-2097152-exc-int-inferred-export-dst
ip prefix-list IPv4-peer49157-2097152-exc-int-inferred-export-dst: 1 entries
   seq 1 permit 192.168.1.254/24
```

L3Out Key Components

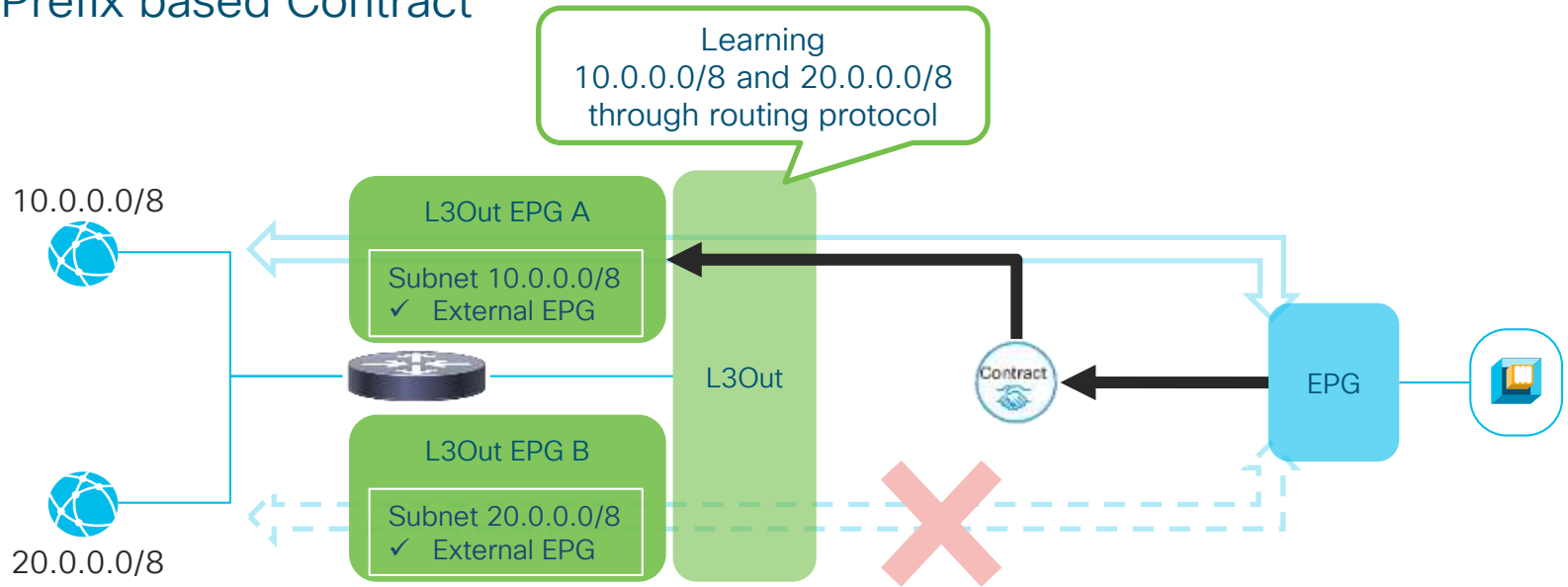
4. Prefix based Contract



How do we accomplish this ??

L3Out Key Components

4. Prefix based Contract



Prefix Based EPG (= L3Out EPG)

L3Out Key Components

4. Prefix based Contract

Configurations

External Routed Networks (L3Out)

Node Profile

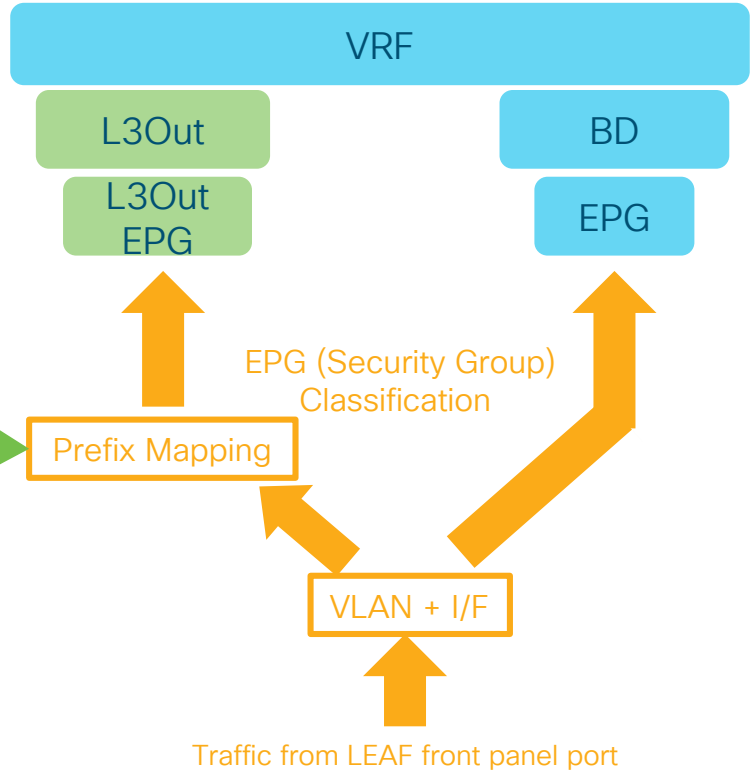
Interface Profile

Networks (L3Out EPG)

- A subnet with scope "External Subnets for the External EPG"

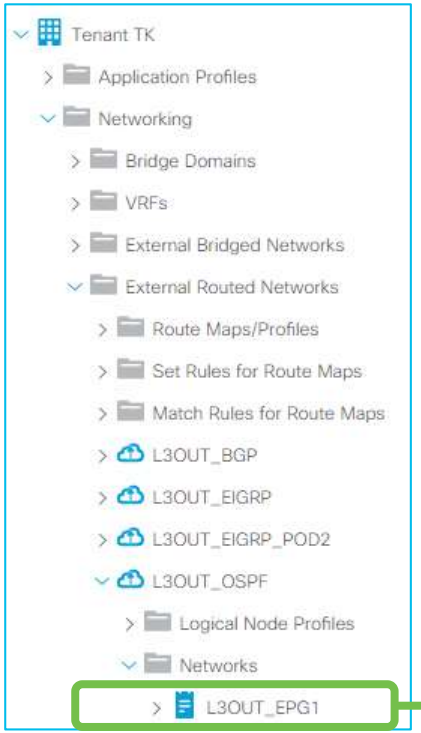
This scope is VRF wide.
No overlapping with other L3Out EPGs in the same VRF

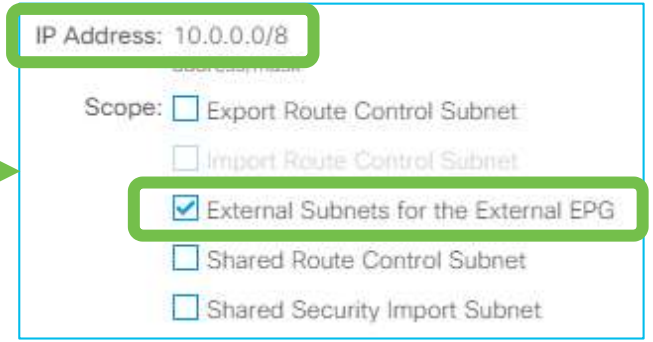
EPG Classification based on prefix



L3Out Key Components

4. Prefix based Contract





VRF1 – 10.0.0.0/8 => pcTag 49158

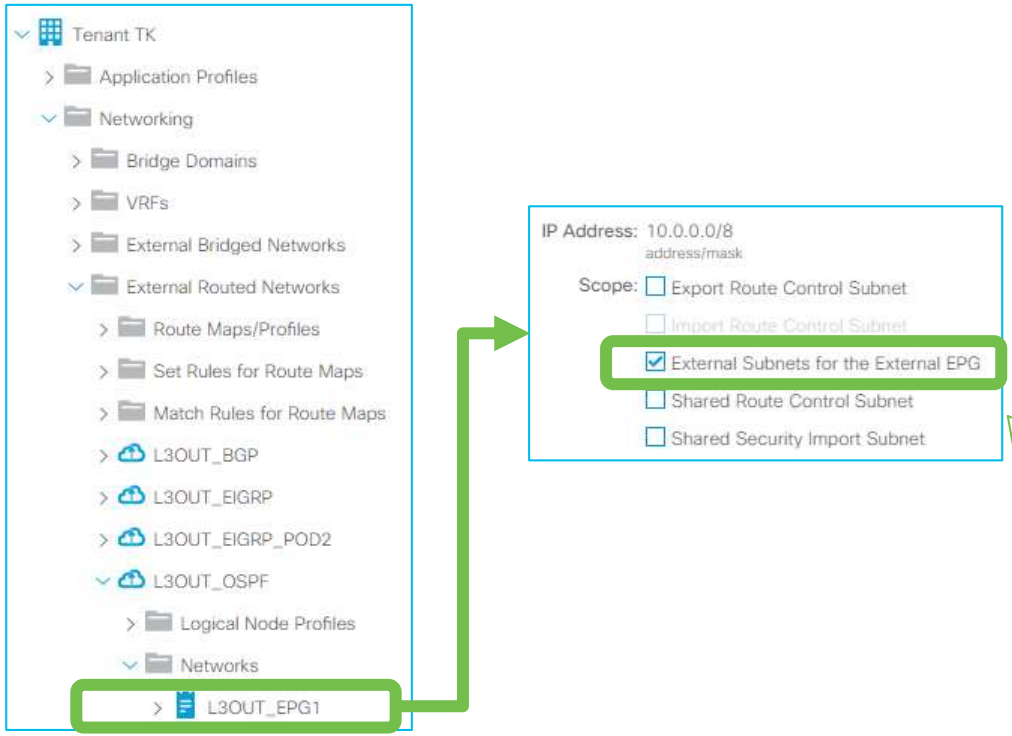
```
leaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152
```

```
leaf# vsh_lc -c 'show system internal aclqos prefix' | egrep 'Vrf|10.0.0.0'
Vrf-Vni VRF-Id Table-Id      Addr         Class  Shared Remote Complete
2097152 8      0x8         10.0.0.0/8     49158  0      1      No

 === use this command from 3.2 ===
leaf# vsh -c 'show sytem internal policy-mgr prefix'
```

L3Out Key Components

4. Prefix based Contract



"External Subnets for the External EPG" is to declare this subnet belongs to this L3Out EPG

- To create prefix to pcTag mapping

NOTE:
It has nothing to do with routing table or routing protocol behavior unlike other Route Control Subnet scopes

A common mistake is selecting both "External Subnets for the External EPG" and "Export Route Control Subnet" for the same subnet, which implies a conflicting situation where the subnet behind the L3Out but the same L3Out is also expected to advertise/redistribute the subnet back to where it came from. It may not cause an immediate issue but unnecessary redistribution should always be avoided.
Check L3Out Subnet scope section for details.

CLI Verifications

1. Check if there is any contract drops

```
leaf# show logging ip access-list internal packet-log deny
[ Wed May  8 18:34:31 2019 155907 usecs]: CName: TK:VRF1(VXLAN: 2719744), VlanType: FD_VLAN, Vlan-Id: 26, SMac: 0x0050569185d1,
DMac:0x0022bdf819ff, SIP: 192.168.1.1, DIP: 10.0.0.1, SPort: 58968, DPort: 80, Src Intf: port-channel1, Proto: 6, PktLen: 74
```

Contract Drop on this leaf shows up in this command.
Check both ingress/egress leaf just in case,
or see appendix for Policy Control Enforcement Direction

2. Check VRF VNID

```
leaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152
```

pcTag/contract is per VRF
except for shared service (VRF route leaking)

3. Check source (or destination) EPG pcTag

```
leaf# show system internal epm endpoint ip 192.168.1.1 | egrep
Vlan id : 30 ::: Vlan vnid : 9025 ::: VRF name : TK:VRF1
BD vnid : 16318374 ::: VRF vnid : 2097152
Flags : 0x80005c04 ::: sclass : 49100 ::: Ref count : 5
EP Flags : local|IP|MAC|host-tracked|sclass|timer|
```

If your source/destination is an endpoint, it should be in here.
sclass = pcTag = EPG ID for contract

Make sure the external IP is not here.
This pcTag takes precedence over “prefix-pcTag mapping table”. If it is, check the traffic path that caused ACI to learn the external IP as an endpoint.

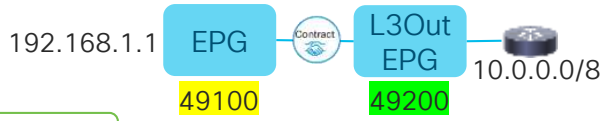
4. Check destination (or source) L3Out prefix based EPG pcTag

```
leaf# vsh_lc -c 'show system internal aclqos prefix' | egrep 'Vrf|10.0.0.0'
Vrf-Vni VRF-Id Table-Id      Addr         Class  Shared Remote Complete
2097152 8      0x8         10.0.0.0/8     49200  0      1      No

 === use this command from 3.2 ===
leaf# vsh -c 'show sytem internal policy-mgr prefix'
```

“External Subnet for the External EPG” config is reflected here.
This is Longest Prefix Match.

CLI Verifications



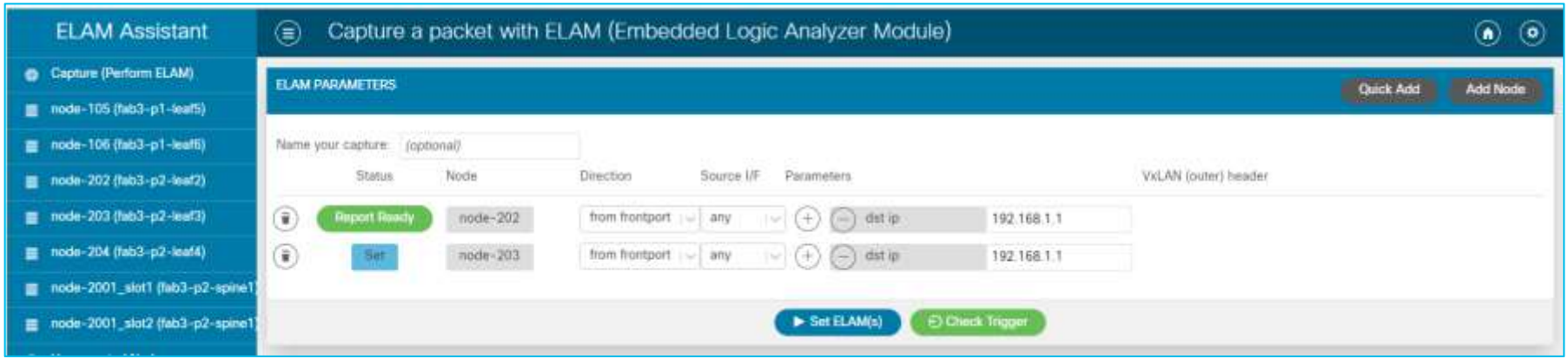
5. Check contracts between two pcTags

scope = VRF VNID

```
leaf# show zoning-rule scope 2097152 | egrep 'Rule|49100|49200'
Rule ID     SrcEPG      DstEPG      FilterID    operSt      Scope       Action      Priority
4165        49100       49200       5           enabled     2097152     permit      fully_qual(7)
4124        49200       49100       5           enabled     2097152     permit      fully_qual(7)
```

```
leaf# show zoning-filter filter 5
FilterId  Name    EtherT  ArpOpc       Prot    ~snip~ SFromPort   SToPort     DFromPort   DToPort     ~snip~
========  ======  ======  ==========   ======= ~snip~ =======     ====        ====        ====        ~snip~
5         5_0     ip      unspecified  icmp    ~snip~ unspecified unspecified unspecified unspecified ~snip~
```

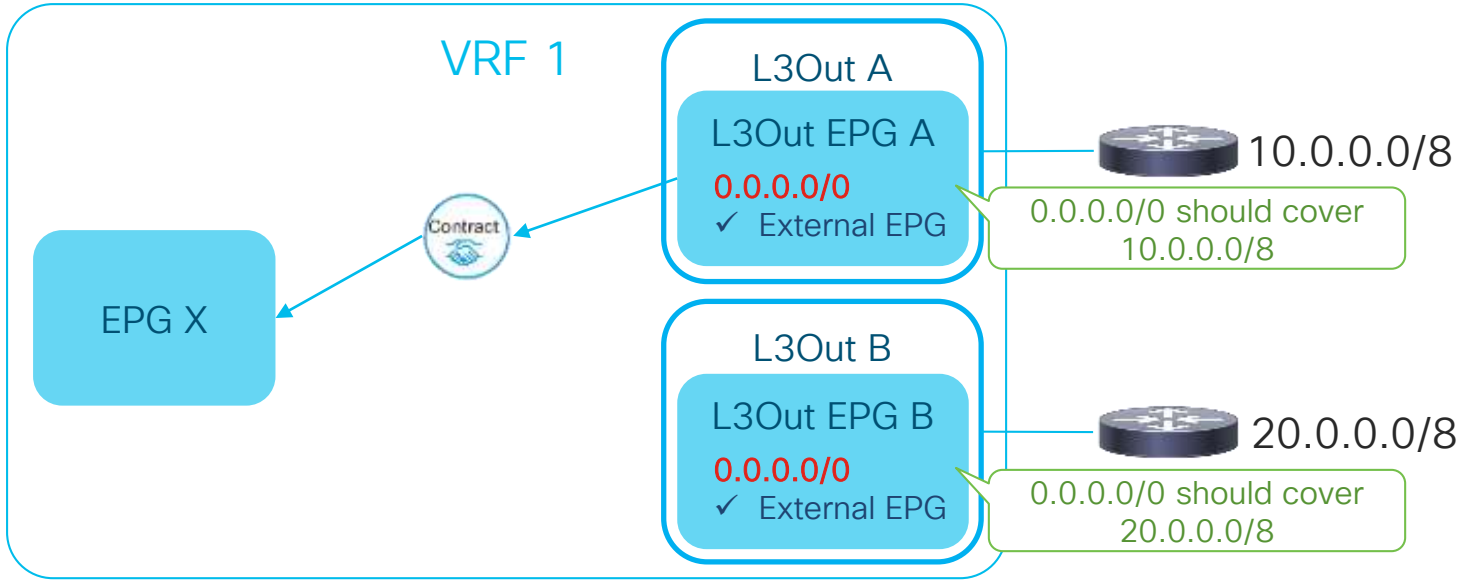
6. Check ELAM to see if the traffic is using correct src pcTag and dst pcTag



https://dcappcenter.cisco.com/elam-assistant.html

L3Out Contract

Common Issue (L3Out EPGs with 0.0.0.0/0)



Both *10.0.0.0/8* and *20.0.0.0/8* can talk to EPG X
even though there is no contract between ***L3Out EPG B*** and ***EPG X***

- Prefix-pcTag mapping is per VRF.
  0.0.0.0/0 for L3Out A and B ends up in the same entry.

Do not overlap External EPG subnets

L3Out Contract

Common Issue (L3Out EPGs with 0.0.0.0/0)

1. Check VRF VNID

```
leaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152
```

2. Check source (or destination) EPG pcTag

```
leaf# show system internal epm endpoint ip 192.168.1.1 | egrep 'VR
Vlan id : 30 ::: Vlan vnid : 9025 ::: VRF name : TK:VRF1
BD vnid : 16318374 ::: VRF vnid : 2097152
Flags : 0x80005c04 ::: sclass : 49100 ::: Ref count : 5
```

"0.0.0.0/0 -> 15" is the only pcTag entry in this VRF.

- Both L3Out A & B will share it since there is no other granular LPM entries

NOTE:
- 0.0.0.0/0 always use pcTag 15
- This is not a routing table. It doesn't matter even if the routing table has more granular routes

3. Check destination L3Out 0.0.0.0/0 EPG pcTag

```
leaf# vsh_lc -c 'show system internal aclqos prefix' | egrep 'Vrf|2097152'
Vrf-Vni VRF-Id Table-Id        Addr        Class  Shared Remote Complete
2097152 8      0x8          0.0.0.0/0     15     0      0      No
```

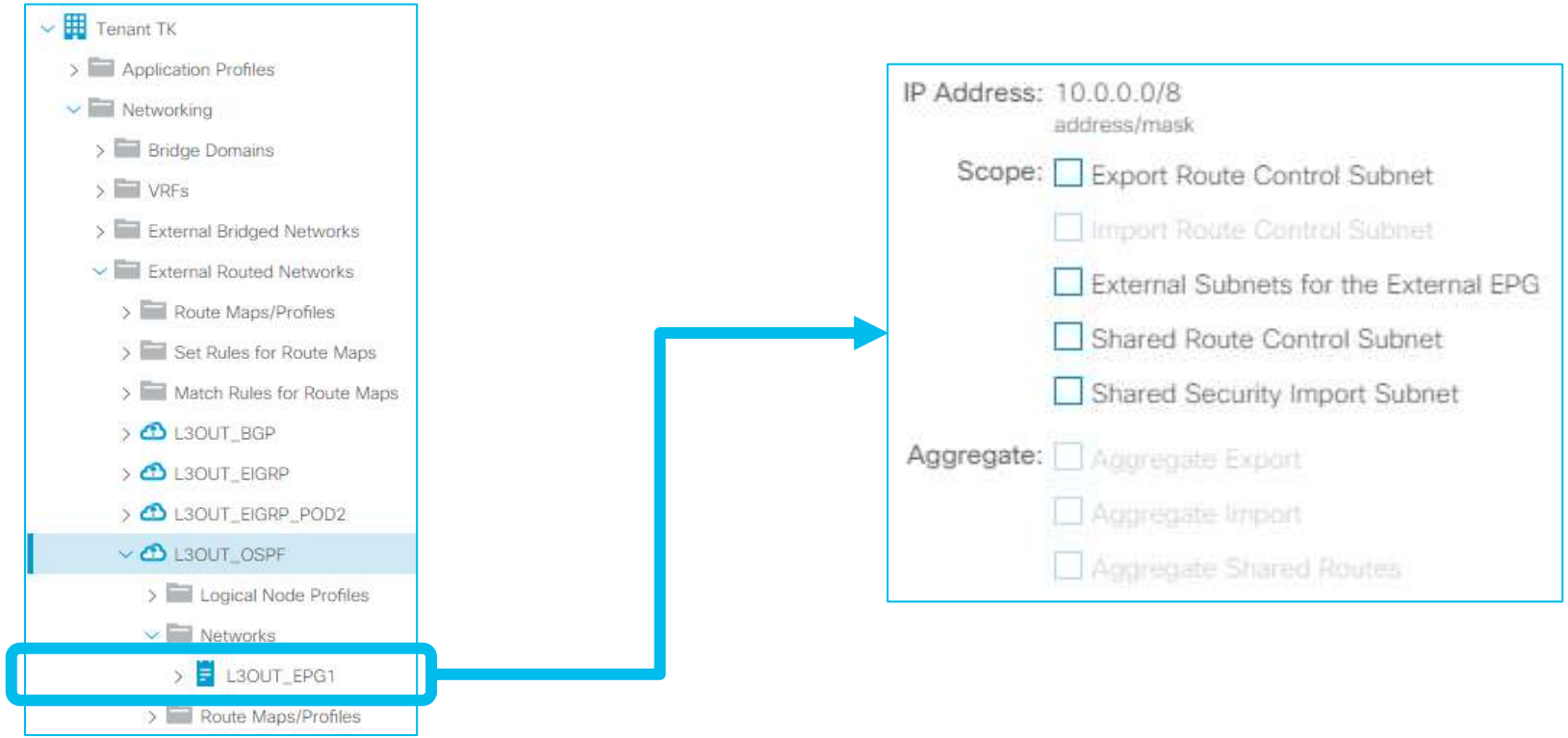
4. Check contracts between pcTags

```
leaf# show zoning-rule scope 2097152 | egrep 'Rule|49162'
Rule ID     SrcEPG      DstEPG      FilterID     operSt          Scope          Action          Priority
4165        49100       15          5            enabled         2097152        permit          fully_qual(7)
```

This contract is due to "EPG X <-> L3Out A"
But any traffic that hits 0.0.0.0/0 in the prefix table can use this rule

L3Out Subnet Scope

L3Out Subnet Scope



L3Out Subnet Scope

Route Control for Routing Protocol

- Export Route Control Subnet
- Import Route Control Subnet
- Shared Route Control Subnet

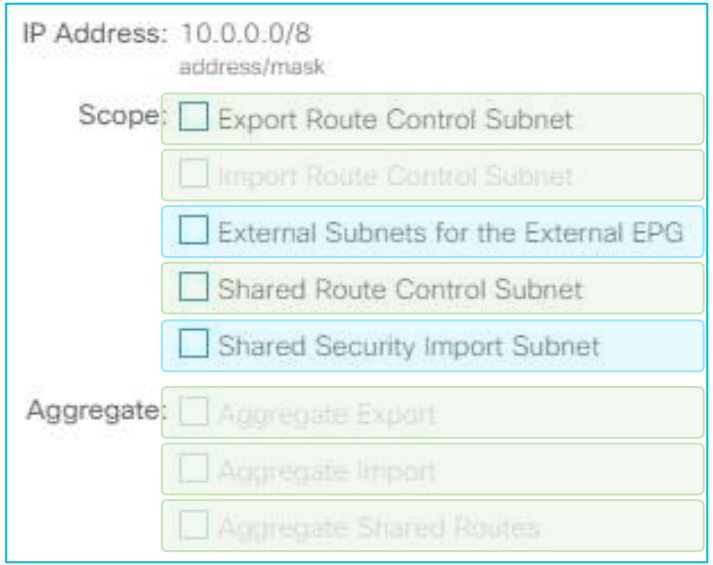
Traffic Classification for Contract

- External Subnets for the External EPG
- Shared Security Import Subnet

Aggregate

- Aggregate Export
- Aggregate Import
- Aggregate Shared Routes

Grouping by functionality

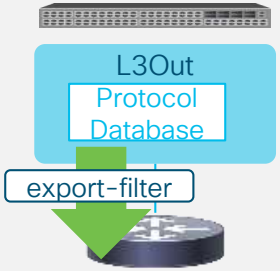


L3Out Subnet Scope Summary

Only for contracts
No impact in routing table

Route Control for Routing Protocol

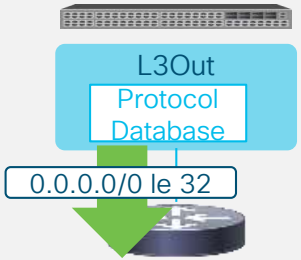
Export Route Control Subnet
(Mainly for Transit Routing)



Advertise the route from ACI to outside

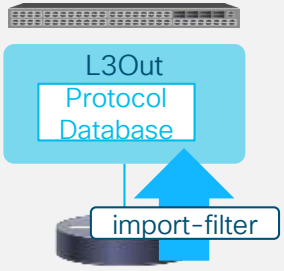
aggregation

Aggregate Export



Advertise all routes from ACI to outside

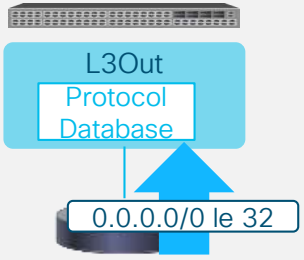
Import Route Control Subnet
(Only for OSPF or BGP)



Receive the route from outside (by default, receive all)

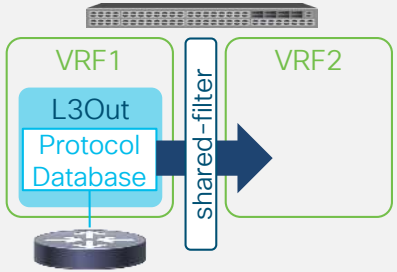
aggregation

Aggregate Import



Receive all routes from outside

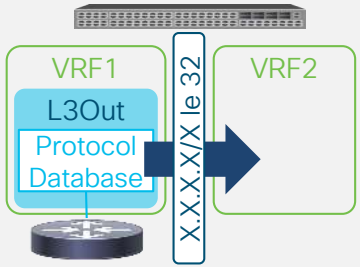
Shared Route Control Subnet



Leak the external route to different VRF

aggregation

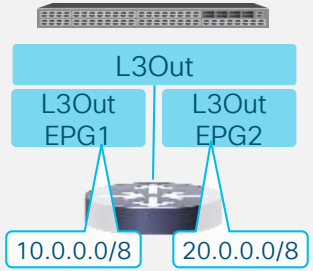
Aggregate Shared Route



Leak multiple external routes to different VRF

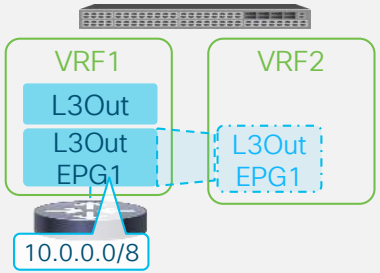
Subnet Classification

External Subnet for the External EPG



Group subnets into each L3Out EPG (pcTag)

Shared Security Import

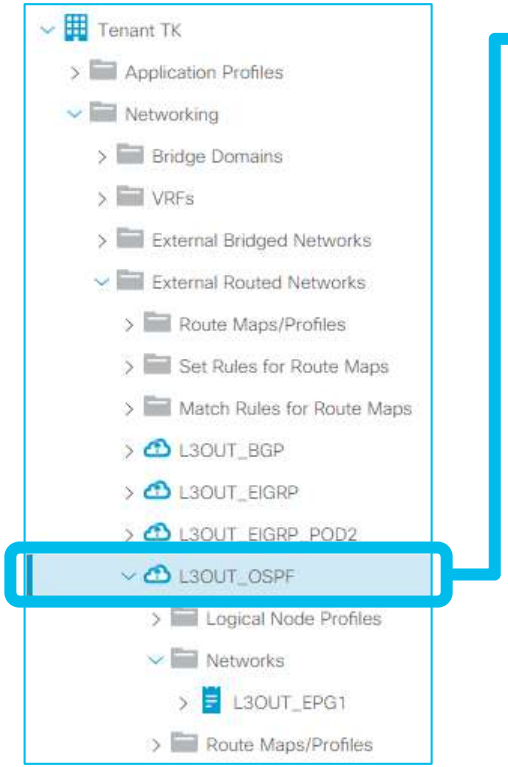


Leak prefix-pcTag mapping to different VRF

Route Control Enforcement

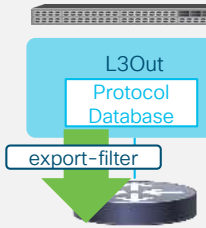
Import is disabled by default.
- Receive all routes by default.
  No import route control.

Export is always enabled.



Available only when enabled

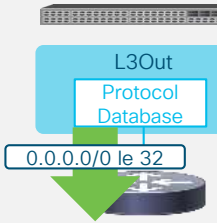
Export Route Control Subn



Advertise the route from A to outside

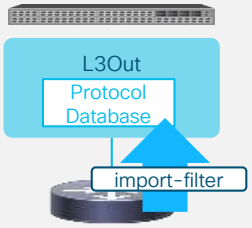
aggregation

Aggregate Export



Advertise all routes from A to outside

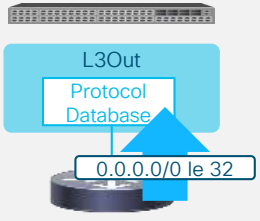
Import Route Control Subnet
(Only for OSPF or BGP)



Receive the route from outside (by default, receive all)

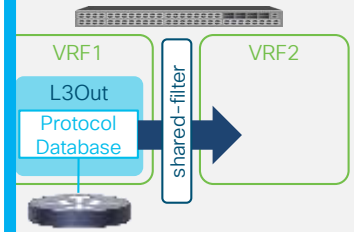
aggregation

Aggregate Import



Receive all routes from outside

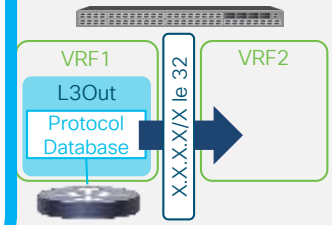
Shared Route Control Subnet



Leak the external route to different VRF

aggregation

Aggregate Shared Route



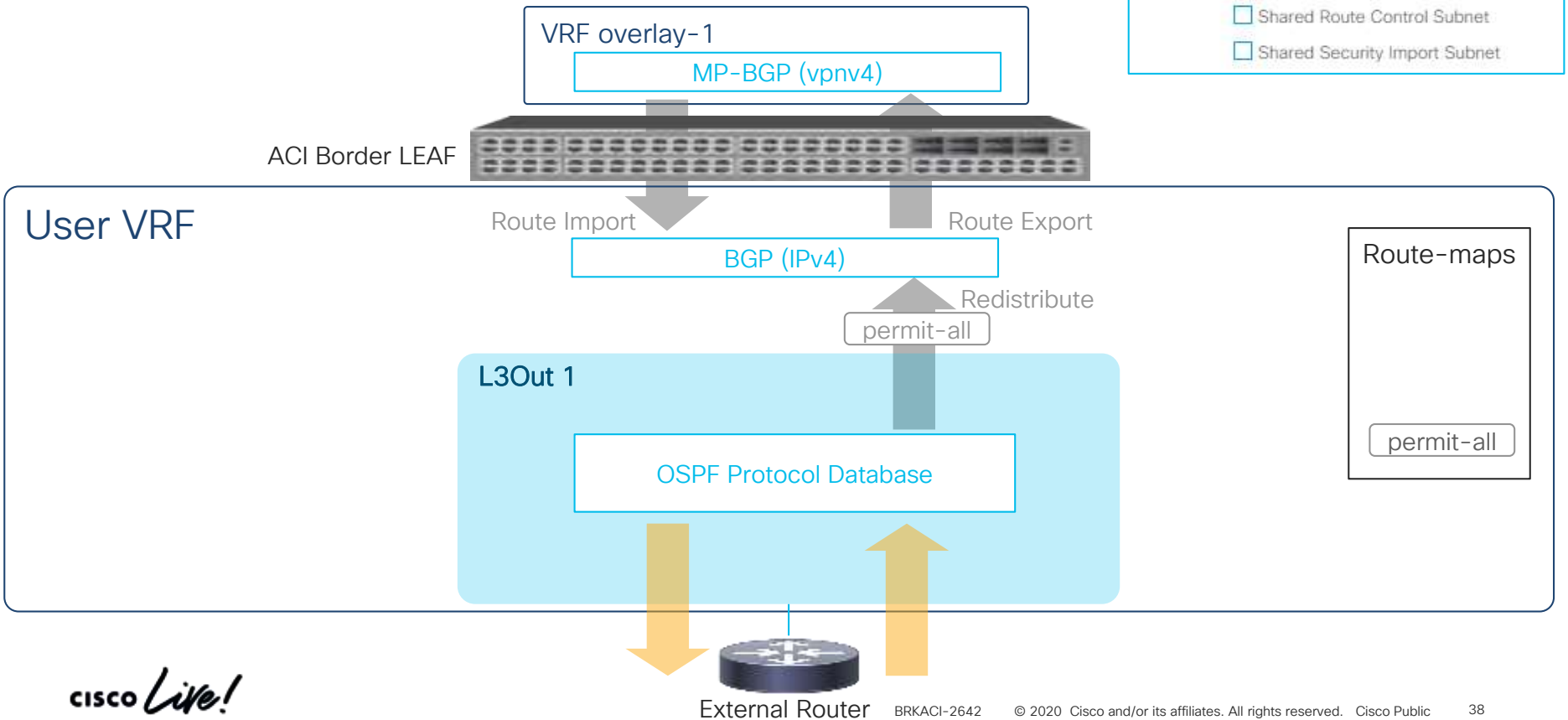
Leak multiple external routes to different VRF

cisco Live!

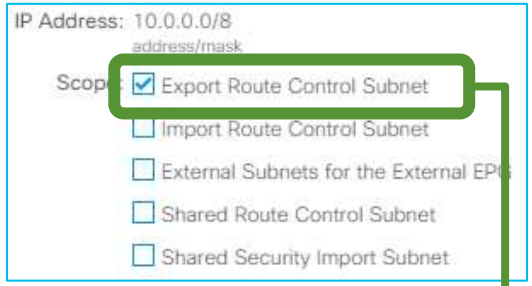
Export Route Control (OSPF)

IP Address: 10.0.0.0/8
address/mask

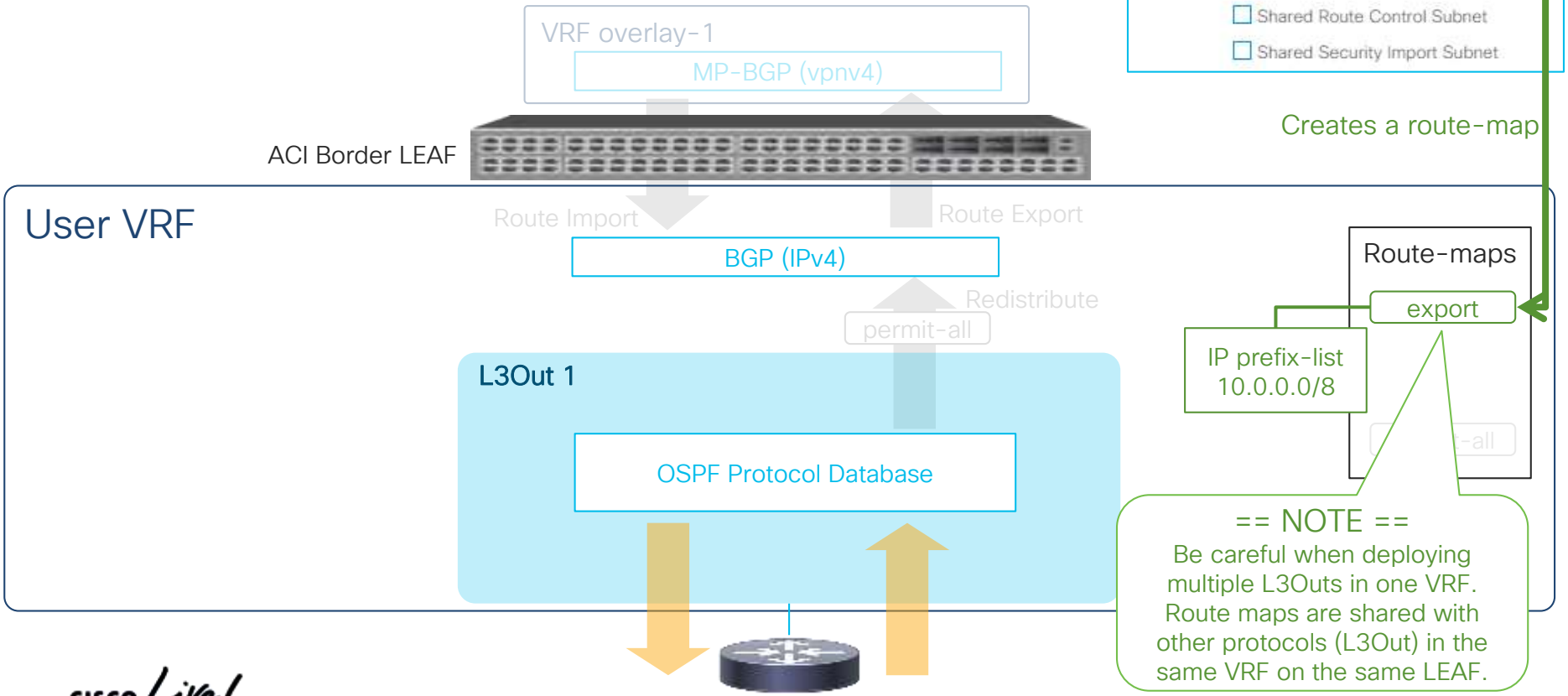
Scope:
- ☐ Export Route Control Subnet
- ☐ Import Route Control Subnet
- ☐ External Subnets for the External EPG
- ☐ Shared Route Control Subnet
- ☐ Shared Security Import Subnet



Export Route Control (OSPF)



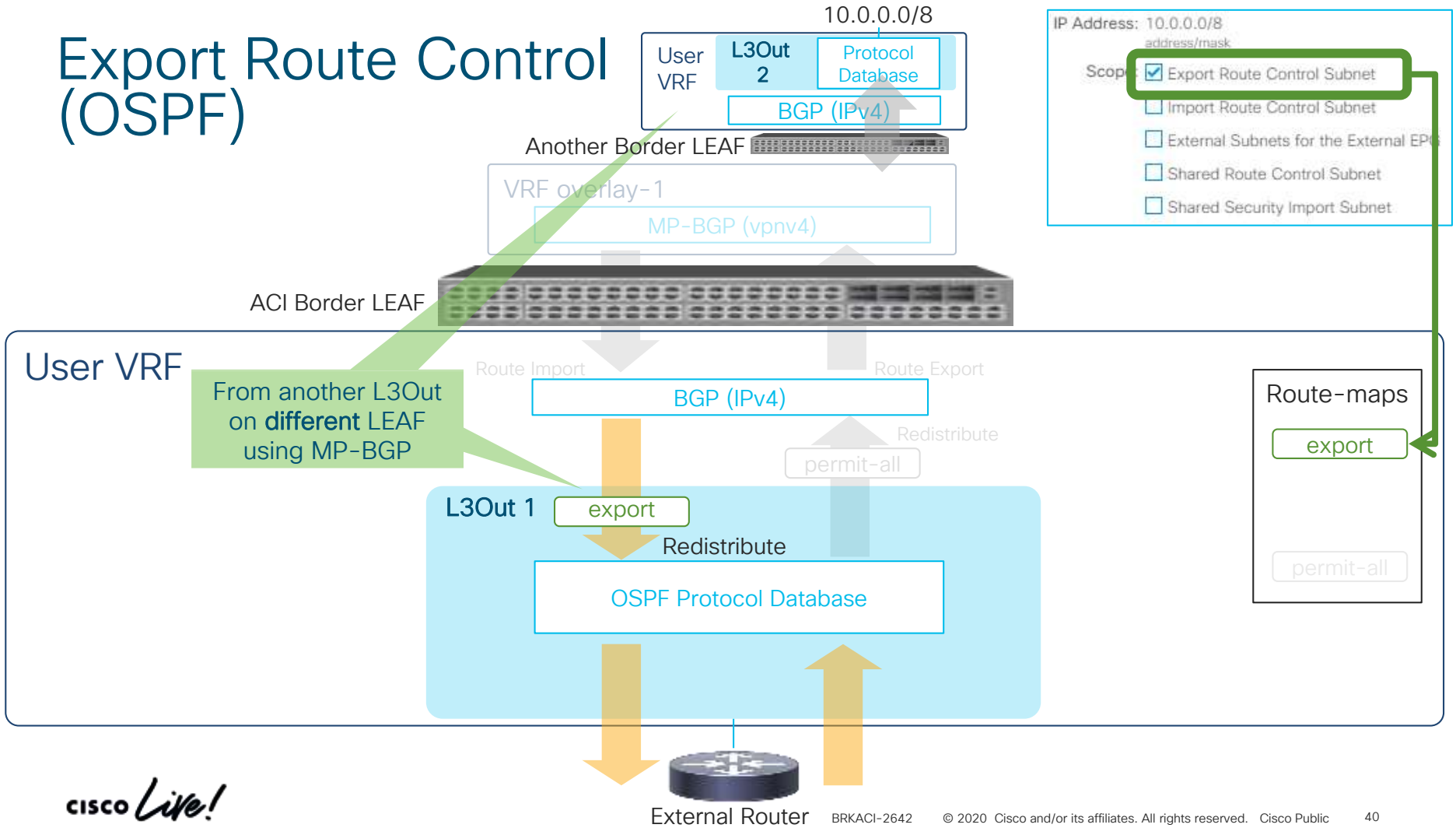
Creates a route-map



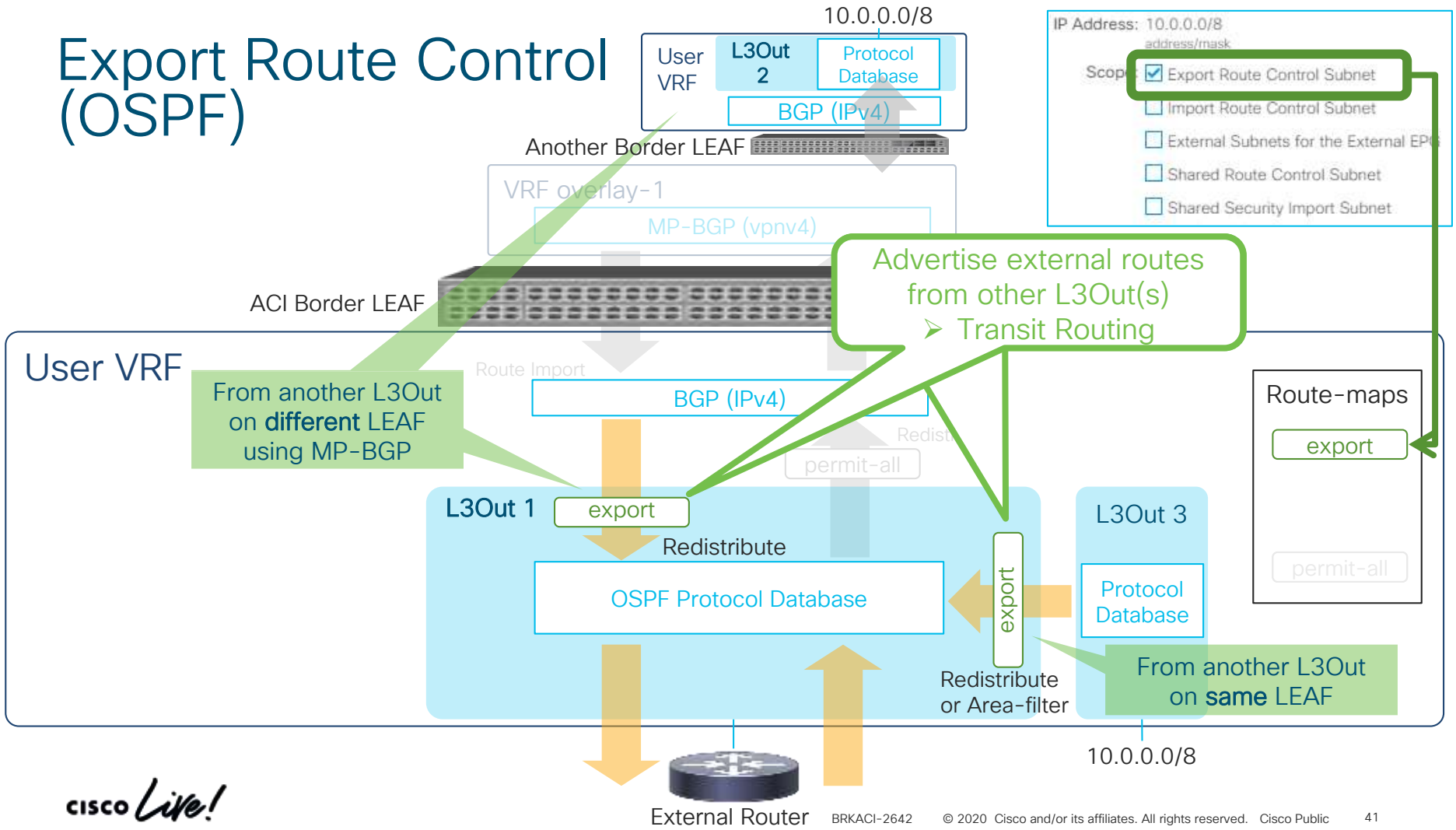
== NOTE ==

Be careful when deploying multiple L3Outs in one VRF. Route maps are shared with other protocols (L3Out) in the same VRF on the same LEAF.

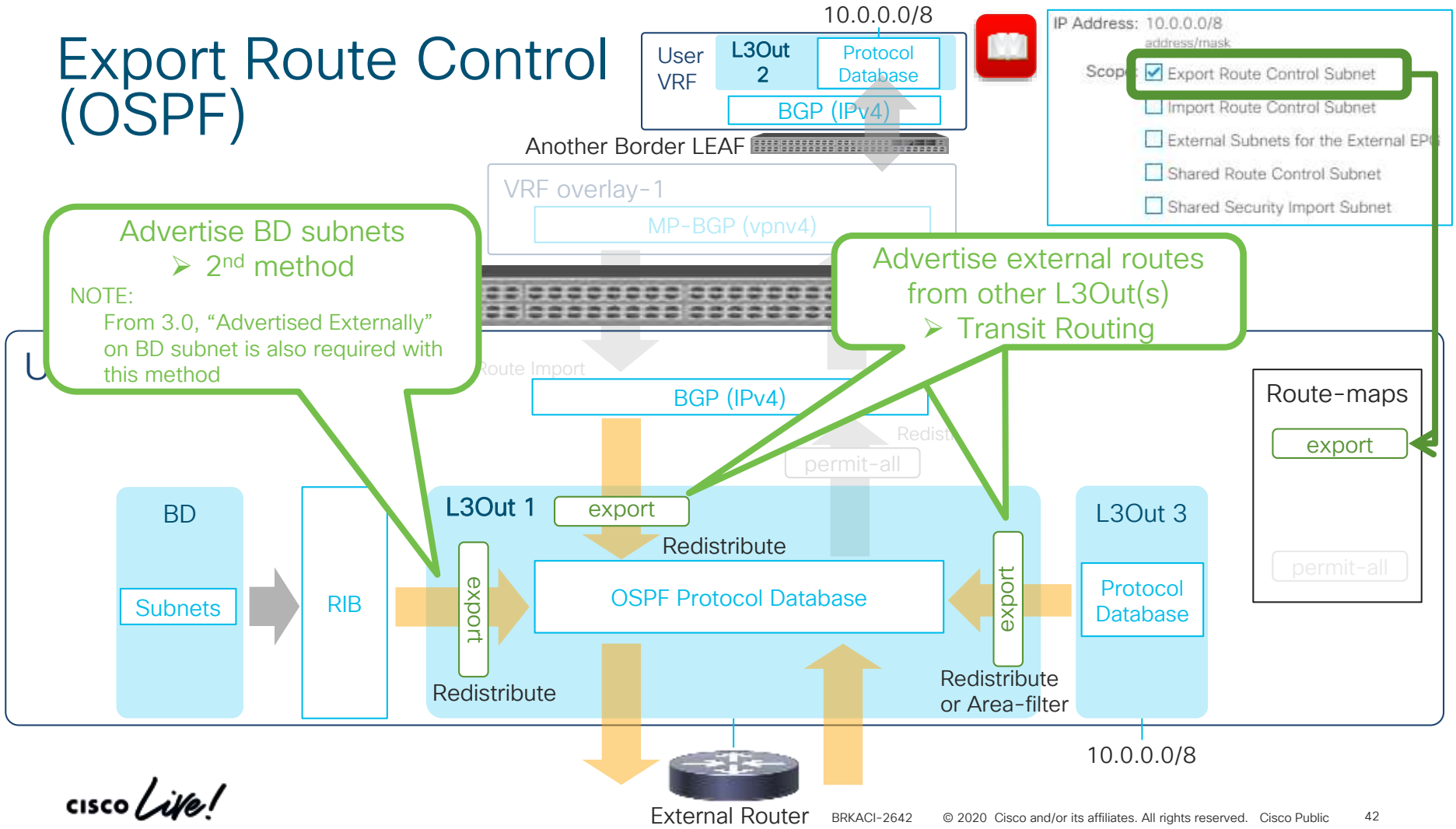
Export Route Control (OSPF)



Export Route Control (OSPF)



Export Route Control (OSPF)



CLI Verification (OSPF/EIGRP)

1. OSPF/EIGRP Redistribution route-map

```
border-leaf# show ip ospf vrf TK:VRF1
 Redistributing External Routes from
   static route-map exp-ctx-st-2097152
   direct route-map exp-ctx-st-2097152
   bgp route-map exp-ctx-proto-2097152
   eigrp route-map exp-ctx-proto-2097152
   Area (backbone)
        Area-filter in 'exp-ctx-proto-2097152'
```

It shares the same route-map with other protocols in the same VRF on the same LEAF

route-map naming:
exp-ctx-st-<vrf vnid> or
exp-ctx-proto-<vrf vnid>

```
border-leaf# show ip eigrp vrf TK:VRF1
  Redistributing:
    static route-map exp-ctx-st-2097152
    ospf-default route-map exp-ctx-proto-2097152
    direct route-map exp-ctx-st-2097152
    bgp-65003 route-map exp-ctx-proto-2097152
```

EIGRP doesn't support Transit Routing on a same LEAF.
- No equivalent filter like OSPF area-filter in EIGRP

2. route-map and ip prefix-list

```
border-leaf# show route-map exp-ctx-proto-2097152
route-map exp-ctx-proto-2097152, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-proto49158-2097152-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
```

All Export Route Control subnet on a same LEAF is added here

Same goes to exp-cxt-st-2097152

```
border-leaf# show ip prefix-list IPv4-proto49158-2097152-exc-ext-inferred-export-dst
ip prefix-list IPv4-proto49158-2097152-exc-ext-inferred-export-dst: 1 entries
   seq 1 permit 10.0.0.0/8
```

CLI Verification (BGP)

BGP has a route-map per L3Out
- A bit more granular control

route-map naming:
exp-l3out-<bgp l3out name>-peer-<vrf vnid>

1. BGP outbound route-map

```
border-leaf# show ip bgp neighbors vrf TK:VRF1
BGP neighbor is 17.0.0.1,  remote AS 65001, ebgp link,  Peer index 1
  Outbound route-map configured is exp-l3out-L3OUT_BGP-peer-2097152, handle obtained
```

2. route-map and ip prefix-list

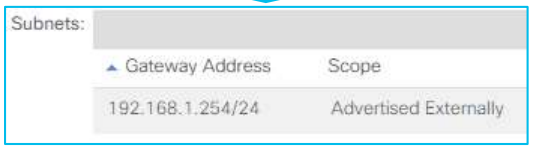
All Export Route Control subnets from the same BGP L3Out is added here

```
border-leaf# show route-map exp-l3out-L3OUT_BGP-peer-2097152
route-map exp-l3out-L3OUT_BGP-peer-2097152, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer49157-2097152-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map exp-l3out-L3OUT_BGP-peer-2097152, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
```

```
border-leaf# show ip prefix-list IPv4-peer49157-2097152-exc-ext-inferred-export-dst
ip prefix-list IPv4-peer49157-2097152-exc-ext-inferred-export-dst: 4 entries
   seq 1 permit 10.0.0.0/8
```

BD Subnet and Export Route Control

```
border-leaf# show ip route vrf TK:VRF1
192.168.1.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 11.0.248.0%overlay-1, [1/0], 00:00:05, static, tag 4294967295
```



"Advertised Externally" removes VRF tag from BD subnet

```
border-leaf# show ip route vrf TK:VRF1
192.168.1.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 11.0.248.0%overlay-1, [1/0], 00:00:05, static
```

Prior to 3.0

IP prefix-list from "Export Route Control" for 192.168.1.0/24

```
border-leaf# show route-map exp-ctx-st-2097152
route-map exp-ctx-st-2097152, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-st49158-2097152-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
```

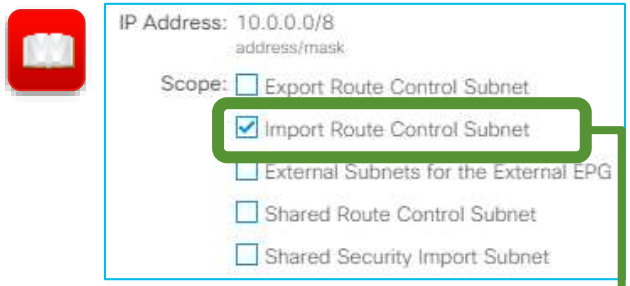
From 3.0

New rule to prevent BD subnets without "Advertised Externally" from being advertised

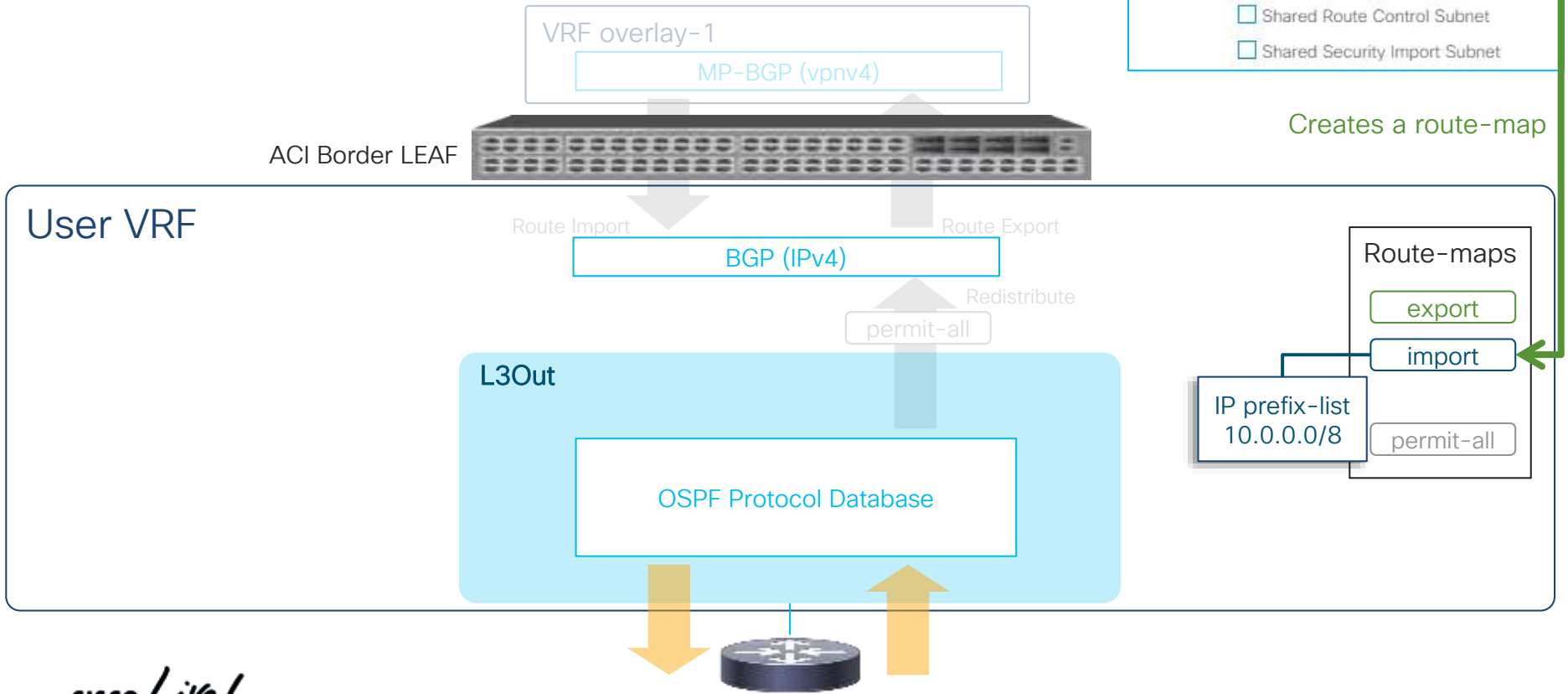
IP prefix-list from "Export Route Control" for 192.168.1.0/24

```
border-leaf# show route-map exp-ctx-st-2097152
route-map exp-ctx-st-2097152, deny, sequence 1
  Match clauses:
    tag: 4294967295
  Set clauses:
route-map exp-ctx-st-2097152, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-st49158-2097152-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
```

Import Route Control (OSPF)



Creates a route-map



Import Route Control (OSPF)

IP Address: 10.0.0.0/8
address/mask

Scope:
- Export Route Control Subnet
- Import Route Control Subnet
- External Subnets for the External EPG
- Shared Route Control Subnet
- Shared Security Import Subnet

If it's not allowed to be imported, it should not be exported to other L3Outs.

When L3Out2 is not OSPF
Redistribution is blocked by table-map (only routes in RIB can be redistributed)
When L3Out2 is OSPF
Block advertisement to another OSPF area via area-filter

ACI Border LEAF

User VRF

Limit external routes to be used in RIB.
It is still in OSPF LSDB.

BD

Subnets

RIB

import

Table-map

OSPF Protocol Database

import

area-filter out

export

Protocol Database

L3Out 2

Route-maps

export

import

permit-all

Still need export in L3Out2 (OSPF) on top of import in L3Out1 (OSPF)

External Router

CLI Verification (OSPF)

```
border-leaf# show ip ospf vrf TK:VRF1
 Table-map using route-map exp-ctx-2097152-deny-external-tag

 Area (backbone)
      Area-filter out 'imp-ctx-ospf-area20971520'
```

- Table-map to prevent the routes from being used in RIB
- “Area-filter out” to prevent the routes from being advertised to another OSPF area on a same LEAF (Transit Routing)

```
border-leaf# show route-map exp-ctx-2097152-deny-external-tag
route-map exp-ctx-2097152-deny-external-tag, deny, sequence 1
  Match clauses:
    tag: 4294967295
  Set clauses:
route-map exp-ctx-2097152-deny-external-tag, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-ospf-49158-2097152-exc-ext-inferred-import-dst-rtpfx
    ipv6 address prefix-lists: IPv6-deny-all
    ospf-area: backbone
  Set clauses:
route-map exp-ctx-2097152-deny-external-tag, deny, sequence 19999
  Match clauses:
    ospf-area: backbone
  Set clauses:
route-map exp-ctx-2097152-deny-external-tag, permit, sequence 20000
  Match clauses:
  Set clauses:
```

route-map for table-map
1. blocks any routes with VRF tag
2. allow routes with Import Route Control subnet in OSPF area X
3. block any routes from OSPF area X

A prefix configured by “Import Route Control Subnet”

```
border-leaf# show ip prefix-list IPv4-ospf-49158-2097152-exc-ext-inferred-import-dst-rtpfx
ip prefix-list IPv4-ospf-49158-2097152-exc-ext-inferred-import-dst-rtpfx: 1 entries
   seq 1 permit 10.0.0.0/8
```

CLI Verification (OSPF) cont.

```
border-leaf# show ip ospf vrf TK:VRF1
 Table-map using route-map exp-ctx-2097152-deny-external-tag

 Area (backbone)
      Area-filter out 'imp-ctx-ospf-area20971520'
```

- Table-map to prevent the routes from being used in RIB
- "Area-filter out" to prevent the routes from being advertised to another OSPF area on a same LEAF (Transit Routing)

```
border-leaf# show route-map imp-ctx-ospf-area20971520
route-map imp-ctx-ospf-area20971520, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-ospf-rt-ospf-import49158-2097152-exc-ext-inferred-import-dst-
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
```

route-map for area-filter

```
border-leaf# show ip prefix-list IPv4-ospf-rt-ospf-import49158-2097152-exc-ext-inferred-import-dst-
ip prefix-list IPv4-ospf-rt-ospf-import49158-2097152-exc-ext-inferred-import-dst-: 1 entries
   seq 1 permit 10.0.0.0/8
```

A prefix configured by "Import Route Control Subnet"

CLI Verification (BGP)

BGP uses an inbound route-map (per L3Out) instead of table-map

```
border-leaf# show ip bgp neighbors vrf TK:VRF1
BGP neighbor is 17.0.0.1,  remote AS 65001, ebgp link,  Peer index 1
  Inbound route-map configured is imp-l3out-L3OUT_BGP-peer-2097152, handle obtained
```

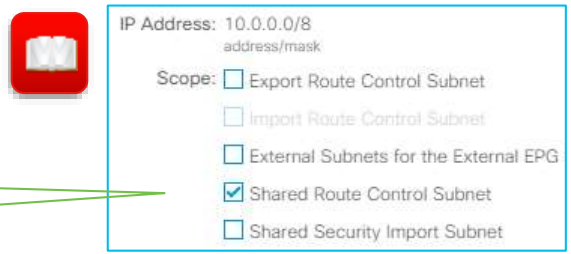
```
border-leaf1# show route-map imp-l3out-L3OUT_BGP-peer-2097152
route-map imp-l3out-L3OUT_BGP-peer-2097152, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-peer49157-2097152-exc-ext-inferred-import-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
```

```
border-leaf# show ip prefix-list IPv4-peer49157-2097152-exc-ext-inferred-import-dst
ip prefix-list IPv4-peer49157-2097152-exc-ext-inferred-import-dst: 1 entries
   seq 1 permit 10.0.0.0/8
```

A prefix configured by “Import Route Control Subnet”

Shared Route Control Subnet (VRF Route Leaking)

Configuration in VRF1 L3Out



VRF overlay-1

MP-BGP (vpnv4)

User VRF1

Route Import

Route Export

BGP (IPv4)

Redistribute

permit-all

L3Out

OSPF Protocol Database

User VRF2

Route Import

Route Export

BGP (IPv4)

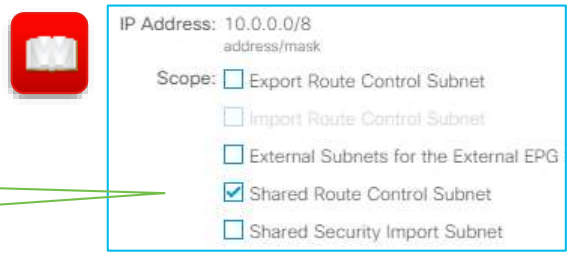
Import RT
<AS>:<VRF2 VNID>

Route-maps

== default ==
Only import L3Out routes for same VRF (VRF2) from other LEAF

Shared Route Control Subnet (VRF Route Leaking)

Configuration in VRF1 L3Out



VRF overlay-1

MP-BGP (vpnv4)

User VRF1

Route Import

Route Export

BGP (IPv4)

Redistribute

permit-all

L3Out

OSPF Protocol Database

User VRF2

Route Import

Route Export

BGP (IPv4)

Import RT

<AS>:<VRF2 VNID>

<AS>:<VRF1 VNID>

Route-maps

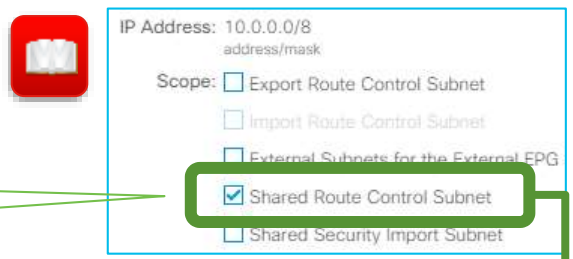
Contract

EPG

Contract across VRFs

- Import VRF1 routes as well

Shared Route Control Subnet (VRF Route Leaking)



Configuration in VRF1 L3Out

VRF overlay-1

MP-BGP (vpnv4)

User VRF1

Route Import

Route Export

BGP (IPv4)

Redistribute

permit-all

L3Out

OSPF Protocol Database

User VRF2

shared

Route Import

Route Export

BGP (IPv4)

Import RT
<AS>:<VRF2 VNID>
<AS>:<VRF1 VNID>

Route-maps

shared

IP prefix-list
10.0.0.0/8

Contract

EPG

Limit routes to be imported (leaked)

CLI Verification

1. MP-BGP Import rule with another VRF VNID route-target and a route-map

```
leaf# show bgp process vrf TK:VRF2

    Information for address family IPv4 Unicast in VRF TK:VRF2

    Import route-map 2588672-shared-svc-leak
    Export RT list:
        65003:2588672
    Import RT list:
        65003:2097152
        65003:2588672
    Label mode: per-prefix
```

- It always has Import and Export RT for its own VRF2 VNID (65003:2588672)
- VRF Route Leaking is handled by Import RT and Import route-map (highlighted ones)

```
leaf# show vrf TK:VRF1 detail extended | egrep 'RD|vxl'
    RD: 10.0.184.64:2
    Encap: vxlan-2097152

leaf# show vrf TK:VRF2 detail extended | egrep 'RD|vxl'
    RD: 10.0.184.64:13
    Encap: vxlan-2588672
```

VRF VNID can be checked with this command to confirm Import RT is correct

CLI Verification

2. A route-map for shared service (VRF Route Leaking)

```
leaf# show route-map 2588672-shared-svc-leak
route-map 2588672-shared-svc-leak, deny, sequence 1
  Match clauses:
    pervasive: 2
  Set clauses:
route-map 2588672-shared-svc-leak, permit, sequence 2
  Match clauses:
    extcommunity  (extcommunity-list filter): 2588672-shared-svc-leak
  Set clauses:
route-map 2588672-shared-svc-leak, permit, sequence 1000
  Match clauses:
    ip address prefix-lists: IPv4-2097152-32771-18-2588672-shared-svc-leak
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
```

1. Prevent BD subnet (pervasive route) from being imported via MP-BGP.
   BD subnet distribution should be done by APIC instead of MP-BGP.
2. Allow importing any routes from the same VRF.
   Extended community list has RT for the same VRF VNID.
3. Allow importing certain routes from another VRF

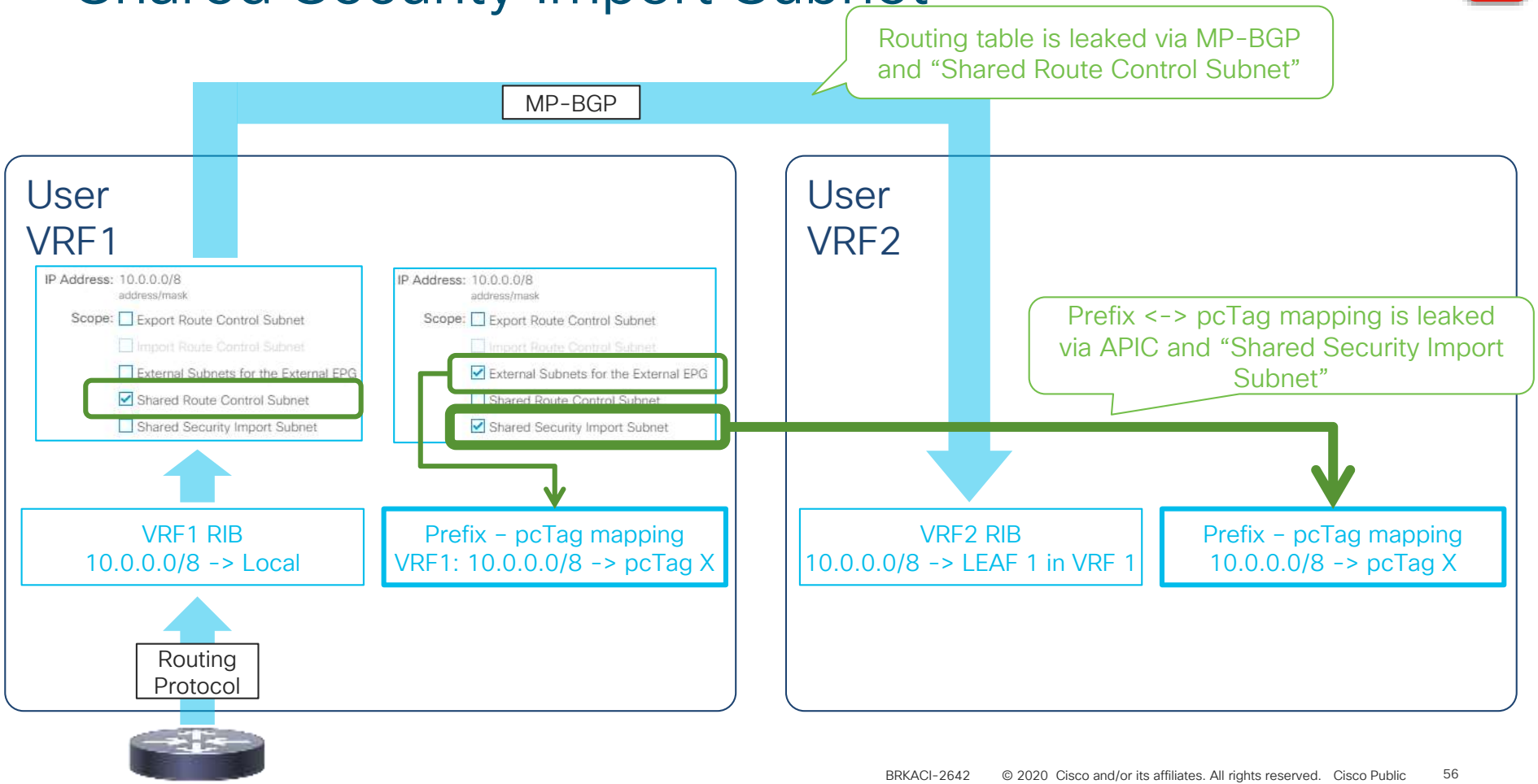
```
leaf# show ip extcommunity-list 2588672-shared-svc-leak
Standard Extended Community List 2588672-shared-svc-leak
    permit RT:65003:2588672
```

RT for the same VRF VNID
Not for VRF Route Leaking

```
leaf# show ip prefix-list IPv4-2097152-32771-18-2588672-shared-svc-leak
ip prefix-list IPv4-2097152-32771-18-2588672-shared-svc-leak: 1 entries
   seq 1 permit 10.10.10.0/8
```

IP Prefix-List from Shared Route Control Subnet

Shared Security Import Subnet



CLI Verification

1. Check VRF VNID

```
leaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152

leaf# show vrf TK:VRF2 detail extended | grep vxlan
    Encap: vxlan-2588672
```

2. Prefix – pcTag mapping table

pcTag (class) for shared route

prefix-pcTag mapping is sahred to VRF2 (VNID 2588672)

```
1st-gen-leaf# vsh_lc -c 'show system internal aclqos prefix' | egrep '^Shared|52.52.52'
Shared Addr     Mask      Scope Class   RefCnt
10.0.0.0        ffffff    0     18      1
```

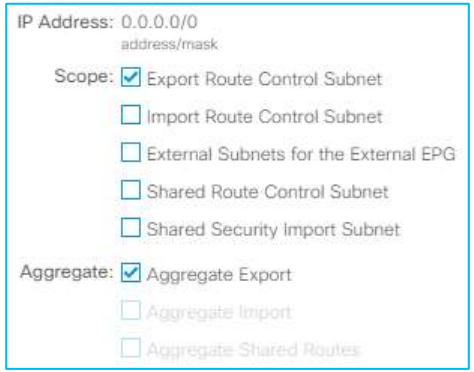
```
2nd-gen-leaf# vsh_lc -c 'show system internal aclqos prefix' | egrep 'Shared|52.52.52'
Vrf-Vni VRF-Id Table-Id          Addr                    Class  Shared Remote Complete
2097152 8      0x8               10.0.0.0/8              18     0      1      No
2588672 11     0xb               10.0.0.0/8              18     1      1      No
```

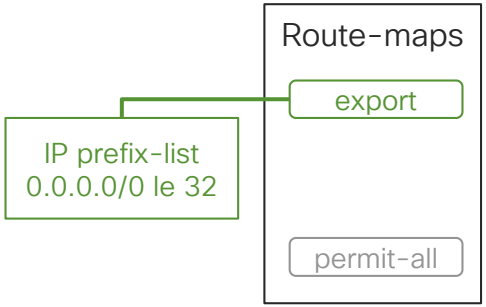
```
leaf# vsh -c 'show system internal policy-mgr prefix'
Vrf-Vni VRF-Id Table-Id Table-State  VRF-Name          Addr      Class Shared Remote Complete
2097152 8      0x8           Up      TK:VRF1       10.0.0.0/8    18   True   True   False
2588672 11     0xb           Up      TK:VRF2       10.0.0.0/8    18   True   True   False
```

From 3.2 release, use this command regardless of leaf generations

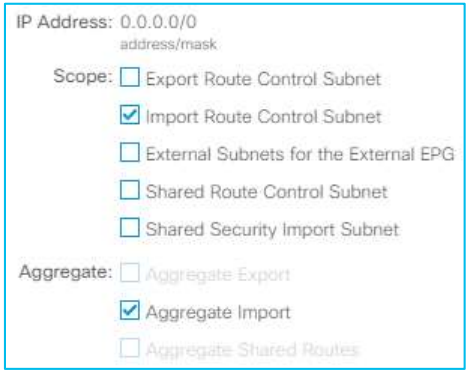
Aggregate Route Control

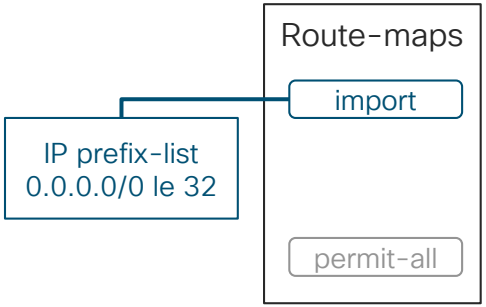
== Export ==



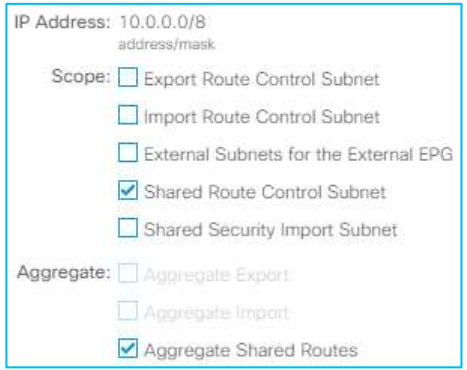


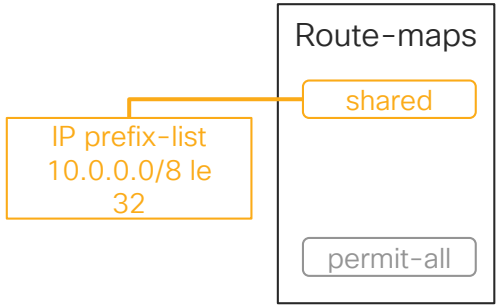
== Import ==





== Shared ==





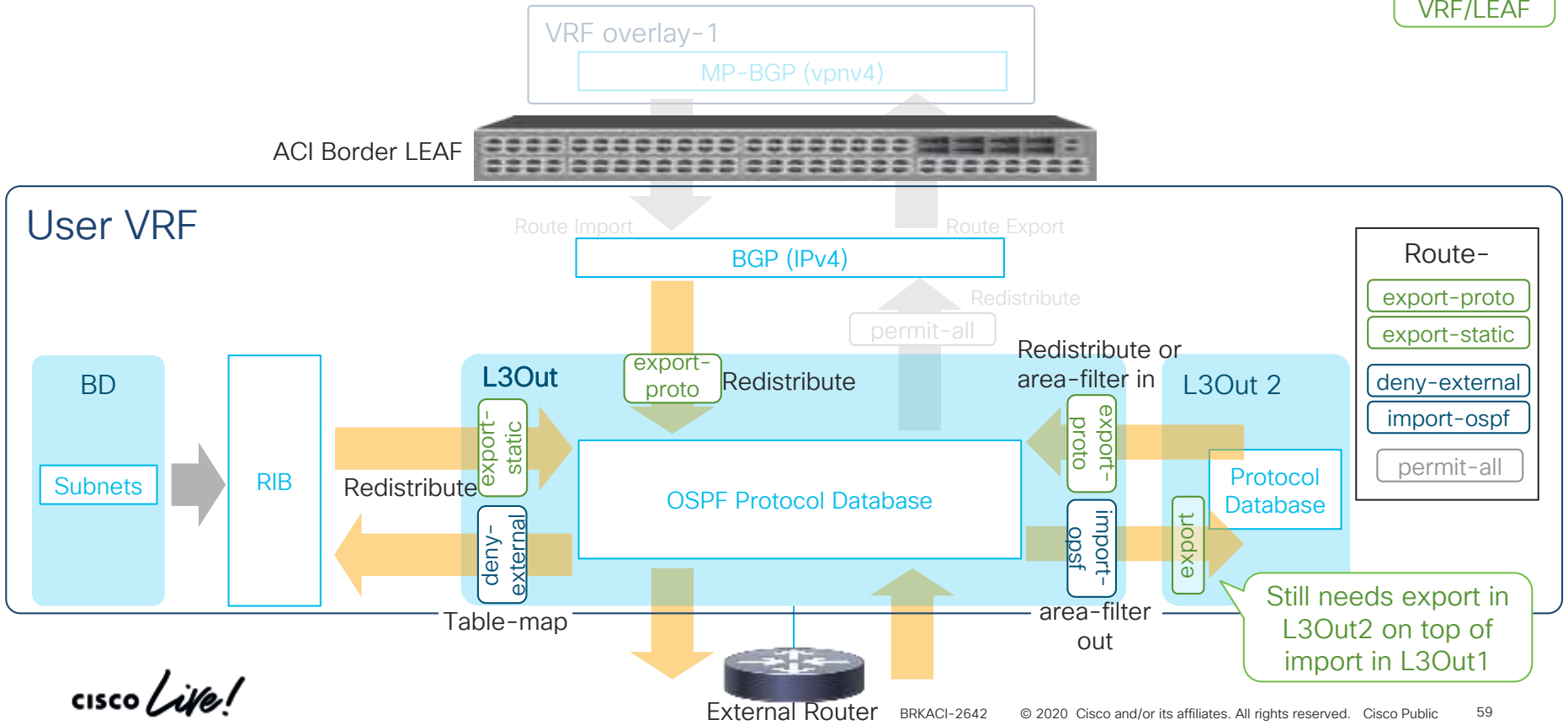
Only “Aggregate Shared Routes” support non-0.0.0.0/0 aggregation

OSPF route-map

Export Route Control Subnet xxxxxxxx

Import Route Control Subnet xxxxxxxxx

Per VRF/LEAF



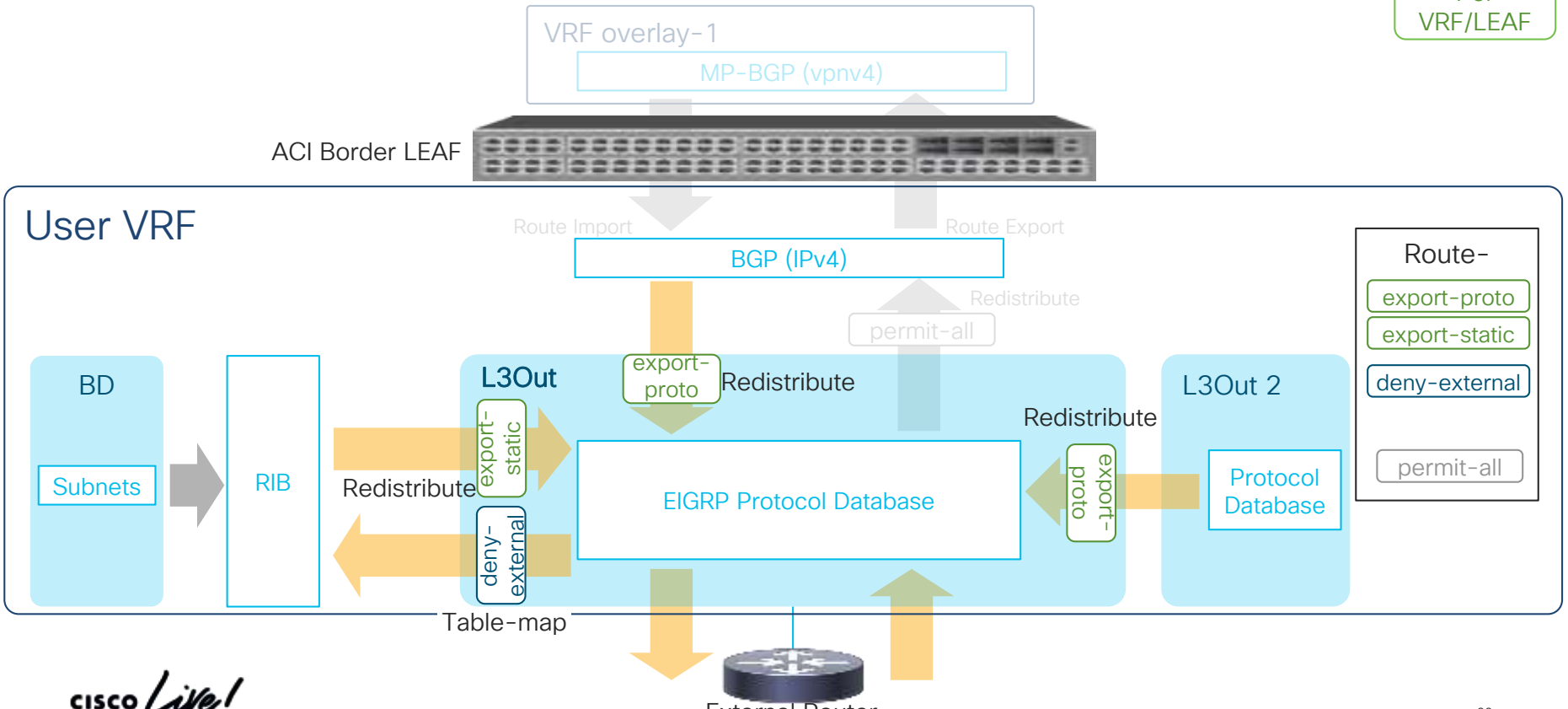
EIGRP route-map

Export Route Control Subnet xxxxxxxx

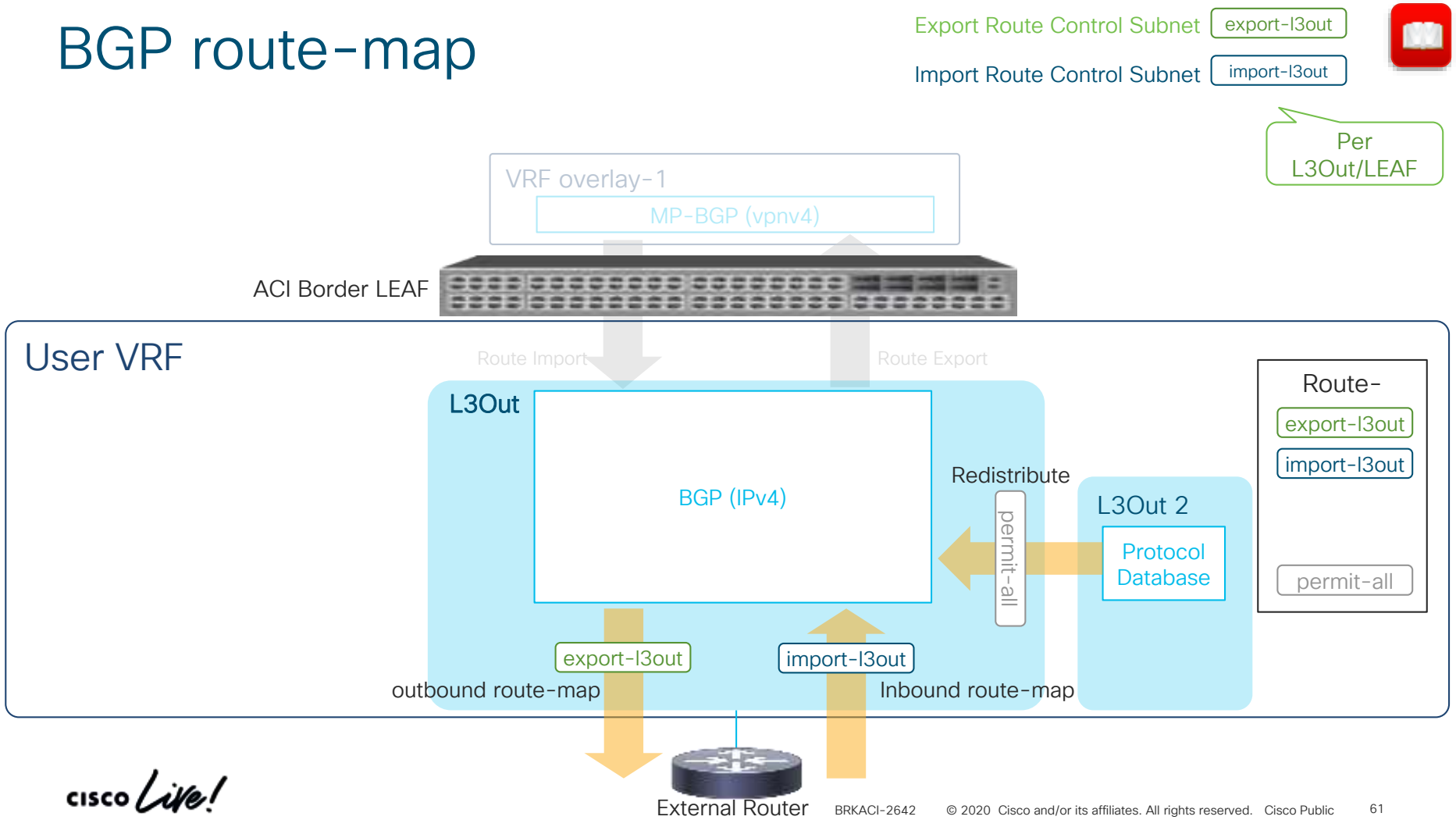
Import is not supported for EIGRP
Used only for VRF Tag

deny-external

Per VRF/LEAF



BGP route-map

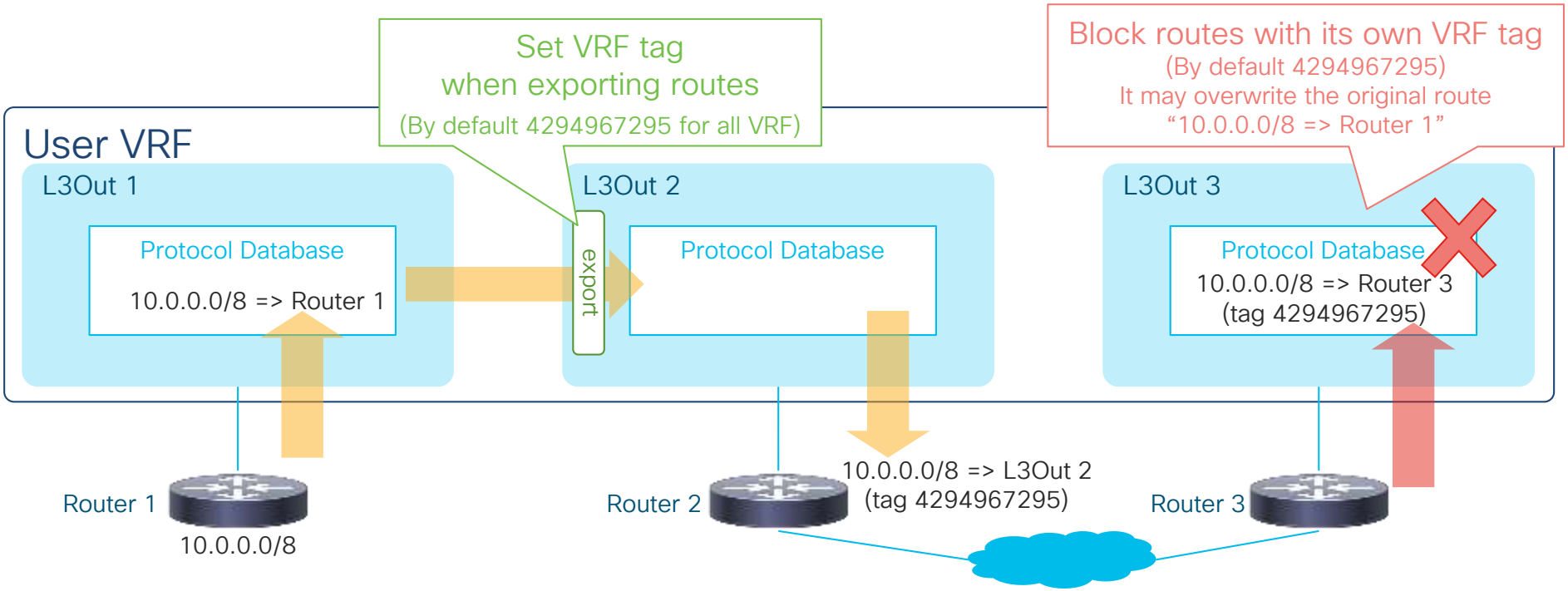


Routing Loop

Loop Avoidance for OSPF/EIGRP – VRF Tag

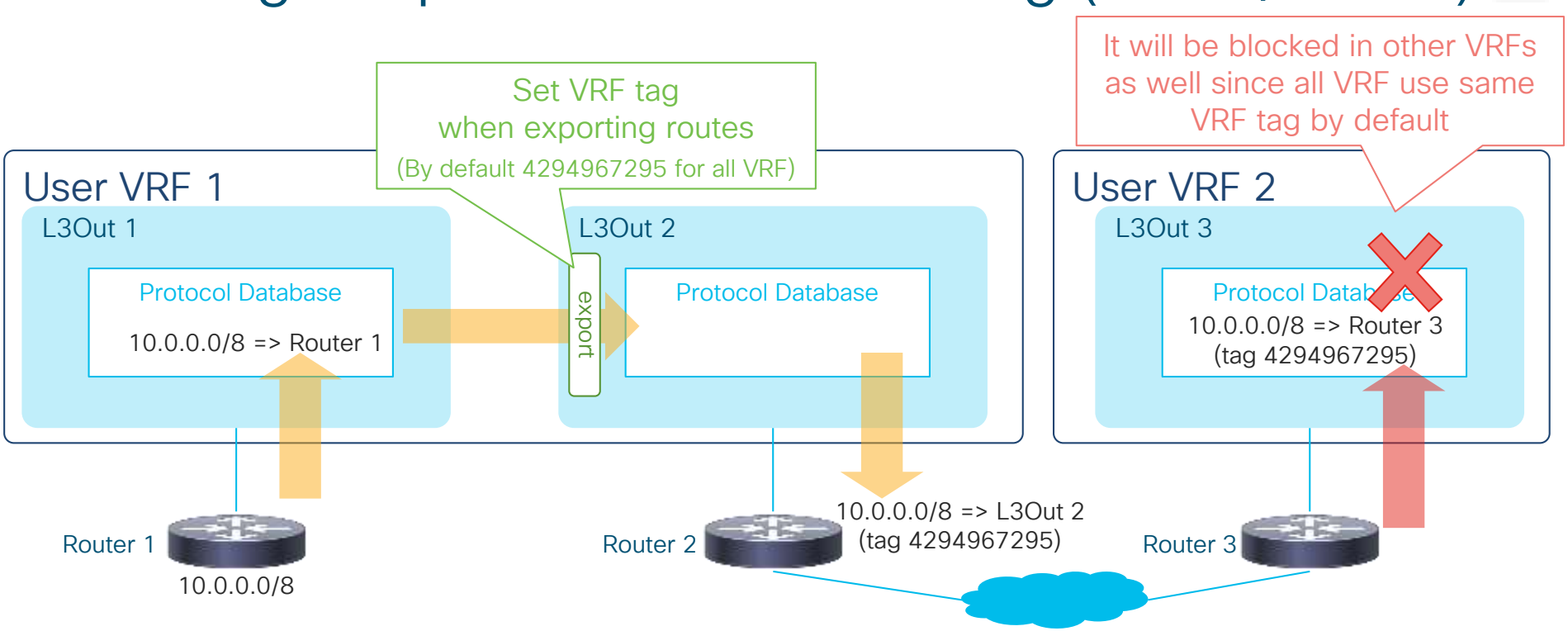
EIGRP and MP-BGP Redistribution Issue

Routing Loop Avoidance – VRF tag (OSPF/EIGRP)



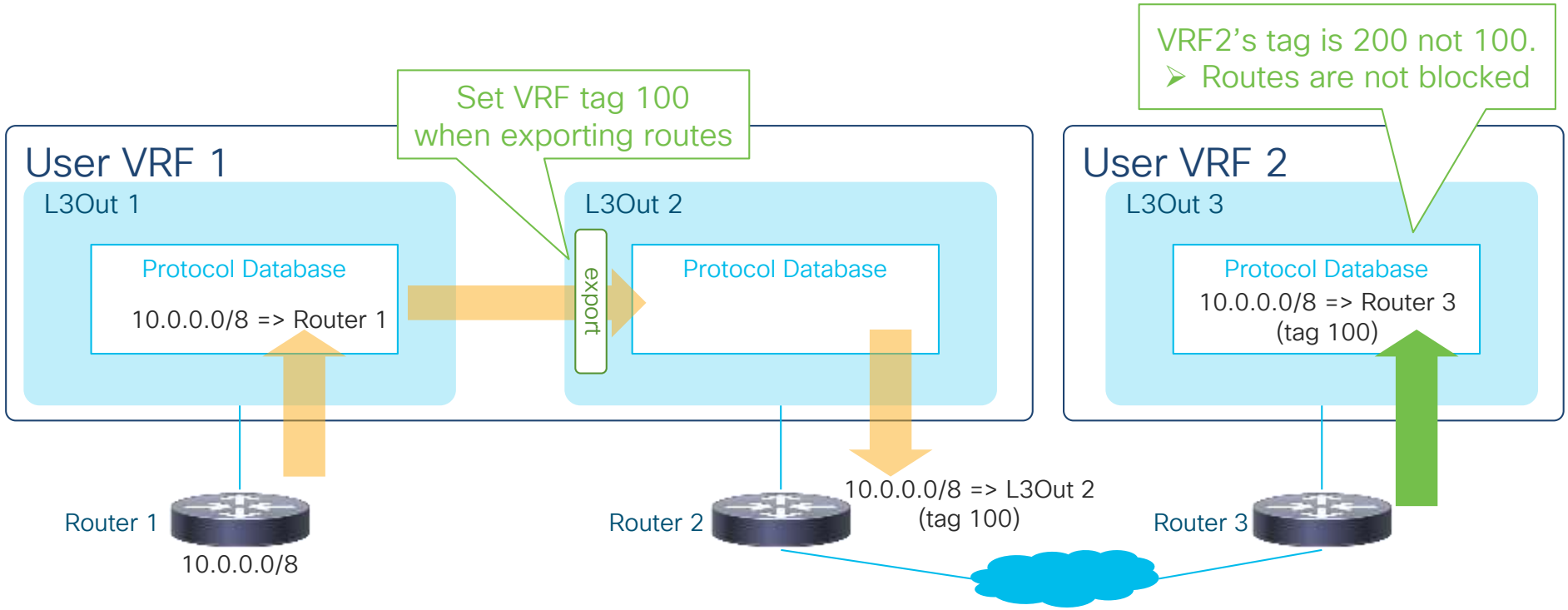
※ VRF tagging for exported routes and blocking routes with VRF tag are always enabled

Routing Loop Avoidance – VRF tag (OSPF/EIGRP)

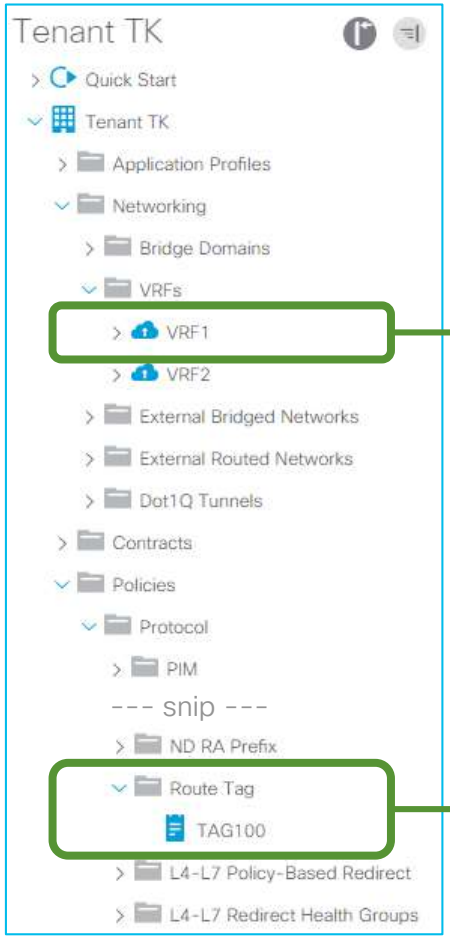


※ VRF tagging for exported routes and blocking routes with VRF tag are always enabled

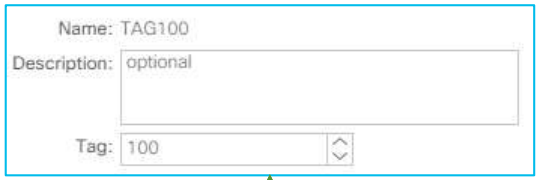
Routing Loop Avoidance – VRF tag (OSPF/EIGRP)



Routing Loop Avoidance – VRF tag (OSPF/EIGRP)

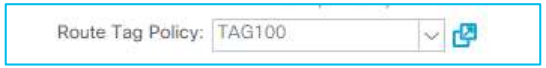


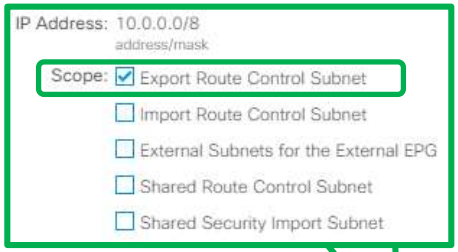
VRF tag can be configured per VRF
In this example VRF1's tag is 100



※ VRF tag is only for OSPF and EIGRP

Routing Loop Avoidance – VRF tag (OSPF/EIGRP)






```
leaf# show ip ospf vrf TK:VRF1 | egrep 'route-map|Redis'
 Table-map using route-map exp-ctx-2097152-deny-external-tag
 Redistributing External Routes from
   static route-map exp-ctx-st-2097152
   direct route-map exp-ctx-st-2097152
   eigrp route-map exp-ctx-proto-2097152
   bgp route-map exp-ctx-proto-2097152
```

Export routes with VRF tag

```
leaf# show route-map exp-ctx-proto-2097152
route-map exp-ctx-proto-2097152, permit, sequence 15802
  Match clauses:
    ip address prefix-lists: IPv4-proto49158-2097152-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 100
```

```
leaf# show ip prefix-list IPv4-proto49158-2097152-exc-ext-inferred-export-dst
ip prefix-list IPv4-proto49158-2097152-exc-ext-inferred-export-dst: 1 entries
   seq 1 permit 10.0.0.0/8
```

Routing Loop Avoidance – VRF tag (OSPF/EIGRP)

Always there with VRF tag

note:

Import Route Control Subnet is added here after VRF tag deny rule when Import Route Control subnet is used.

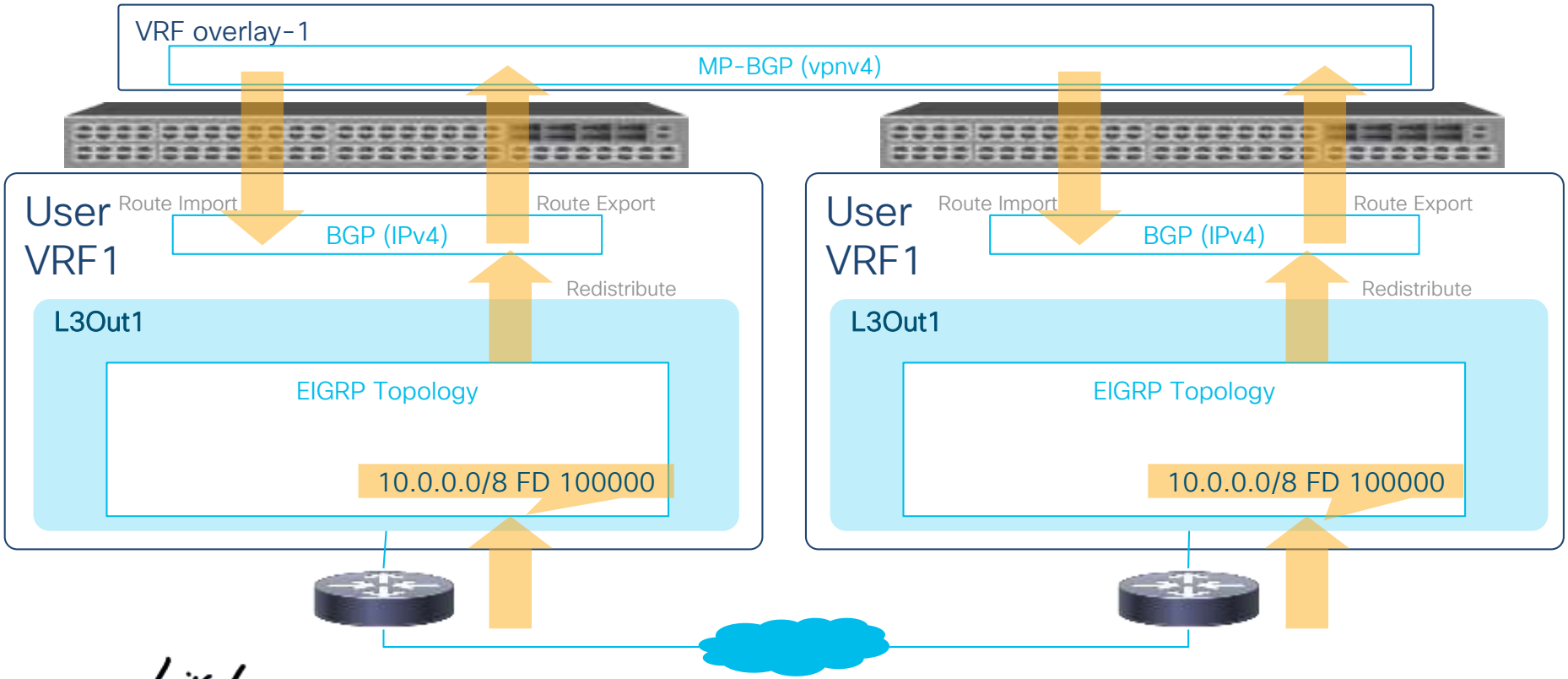
```
leaf# show ip ospf vrf TK:VRF1 | egrep 'route-map|Redis'
 Table-map using route-map exp-ctx-2097152-deny-external-tag
 Redistributing External Routes from
   static route-map exp-ctx-st-2097152
   direct route-map exp-ctx-st-2097152
   eigrp route-map exp-ctx-proto-2097152
   bgp route-map exp-ctx-proto-2097152
```

Block routes with VRF tag

```
leaf# show route-map exp-ctx-2097152-deny-external-tag
route-map exp-ctx-2097152-deny-external-tag, deny, sequence 1
  Match clauses:
    tag: 100
  Set clauses:
route-map exp-ctx-2097152-deny-external-tag, permit, sequence 200
  Match clauses:
  Set clauses:
```

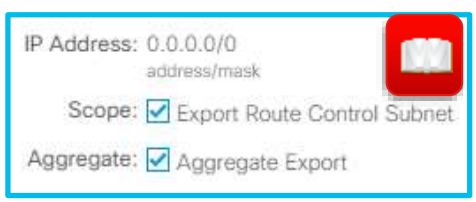
EIGRP & MP-BGP Redistribution Issue

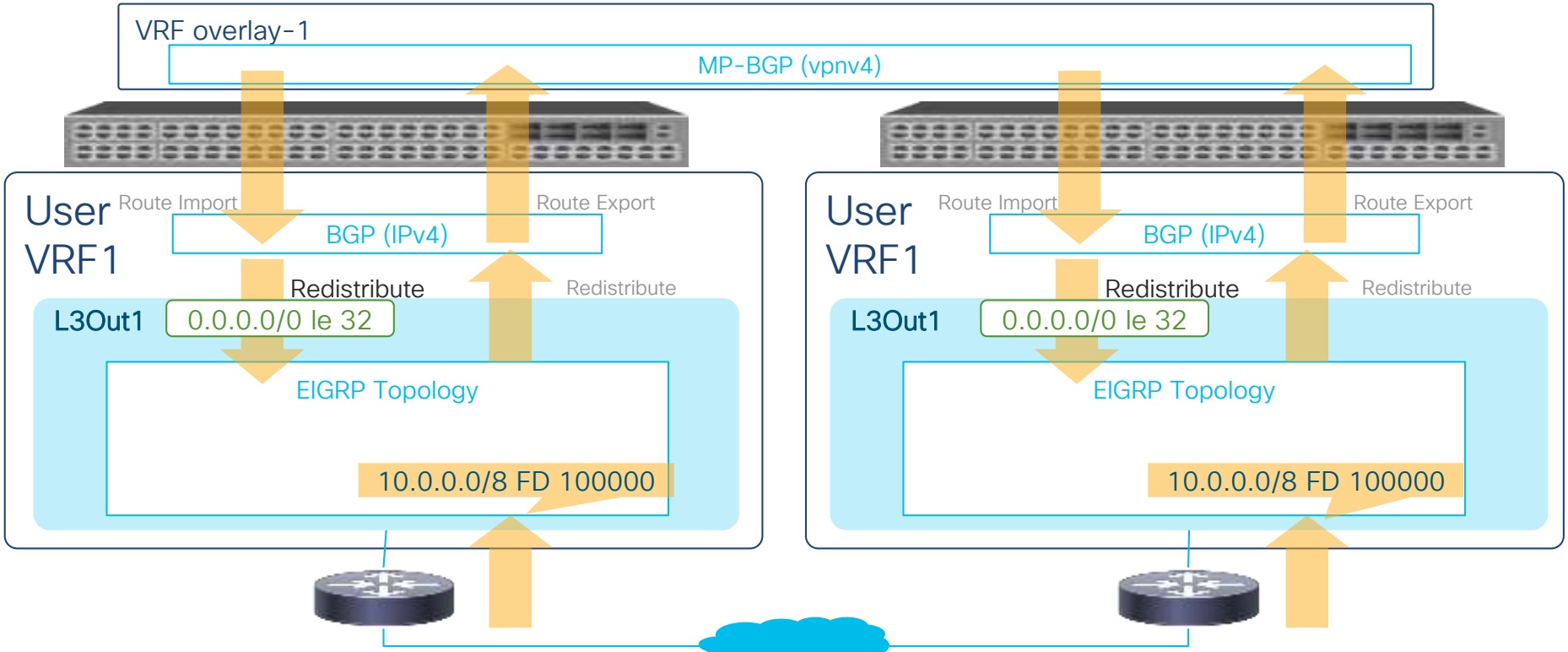
1. 10.0.0.0/8 from L3Out 1 via EIGRP on two Border LEAFs



EIGRP & MP-BGP Redistribution Issue

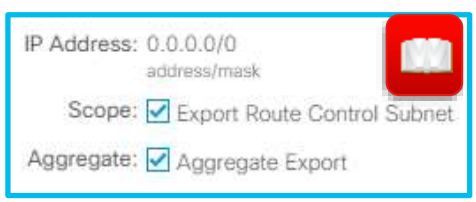
2. L3Out 1 exports all routes (including 10.0.0.0/8)

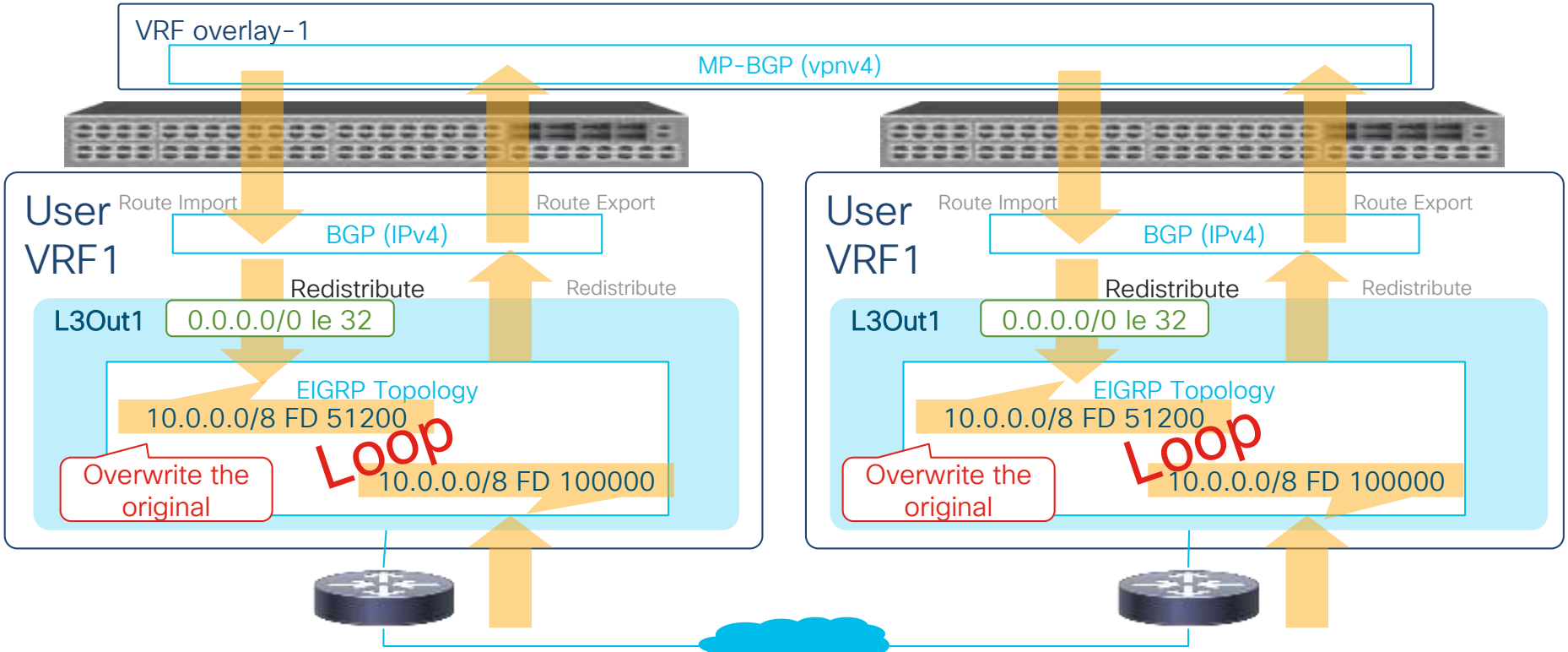




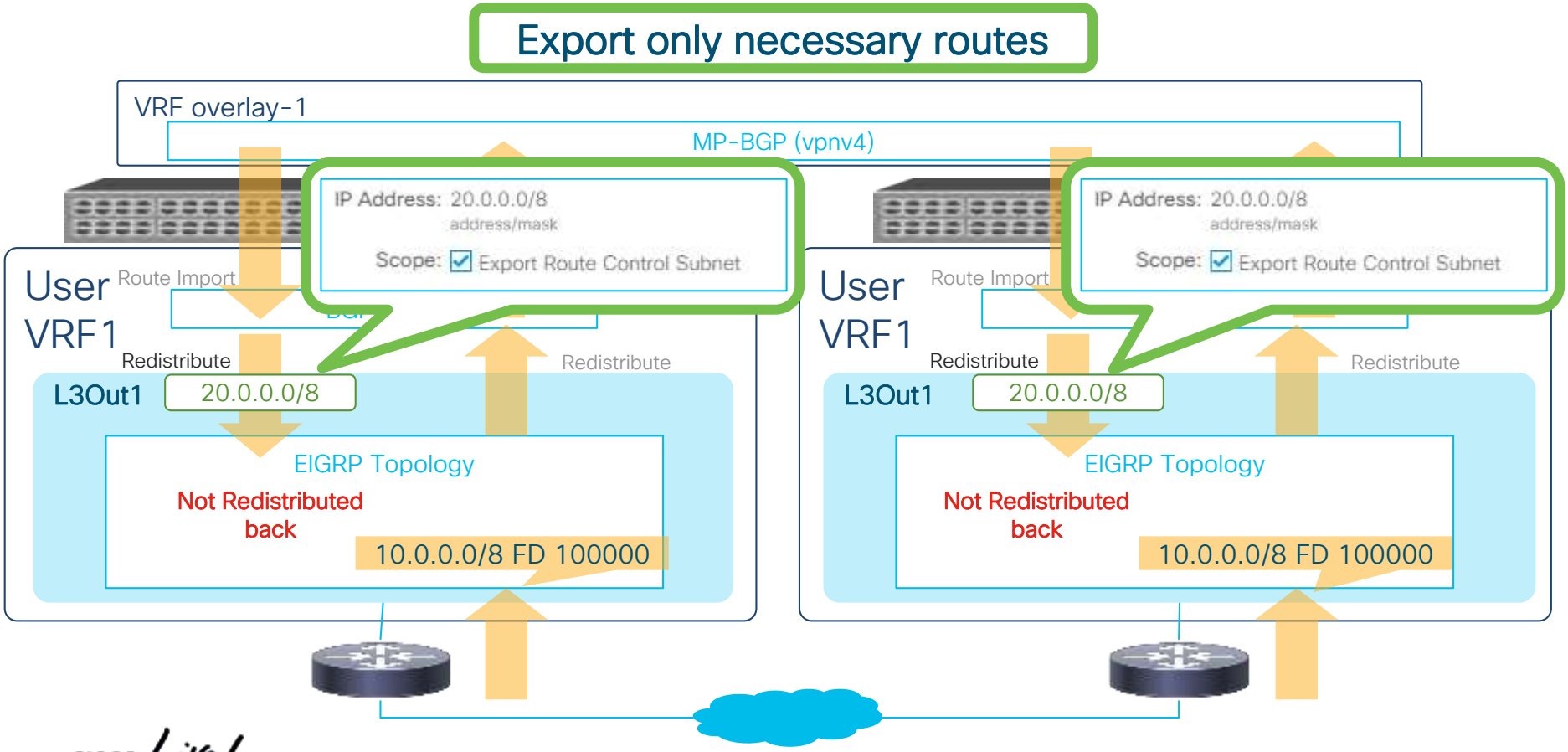
EIGRP & MP-BGP Redistribution Issue

3. Redistributed routes have lower metric than the original

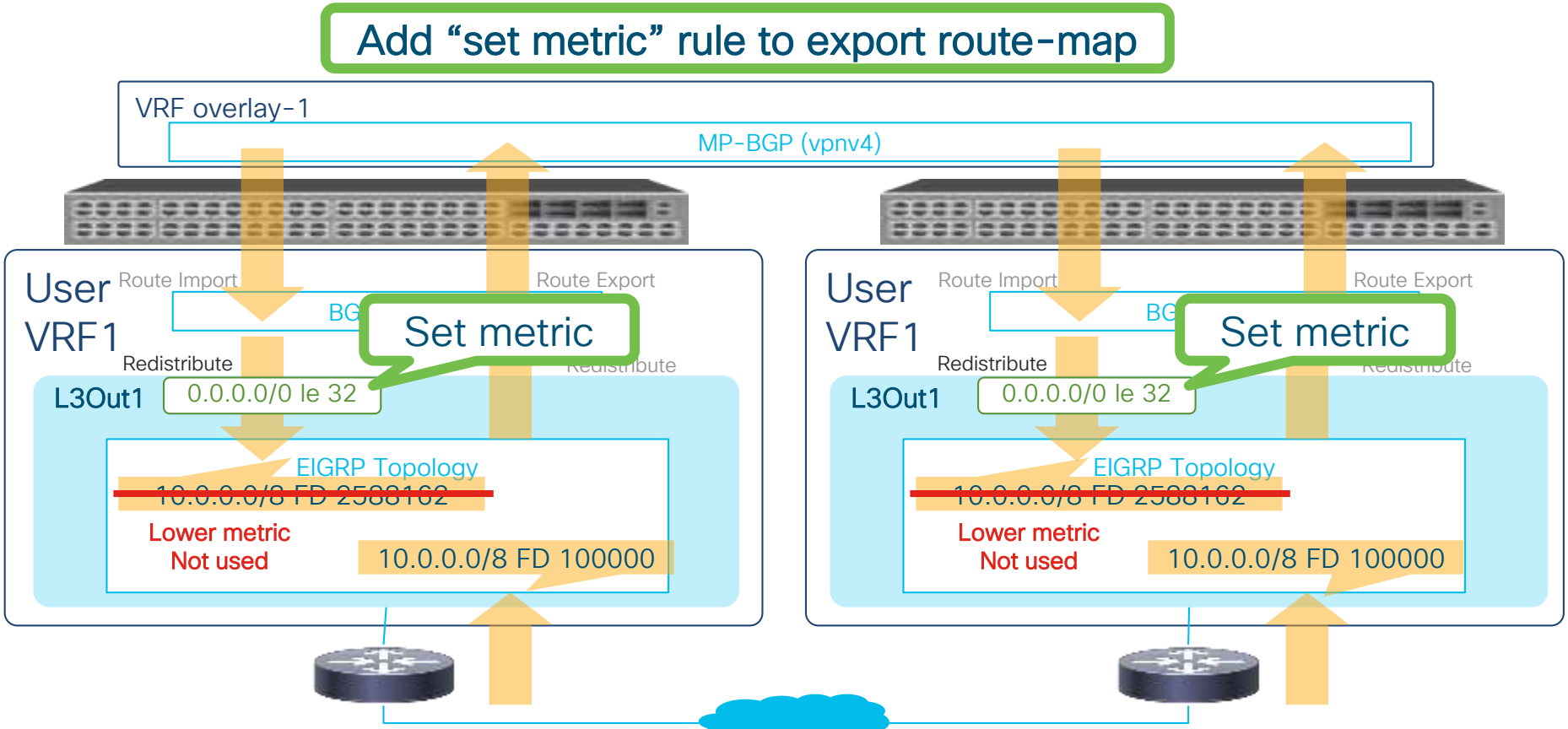




EIGRP & MP-BGP Redistribution Solution1

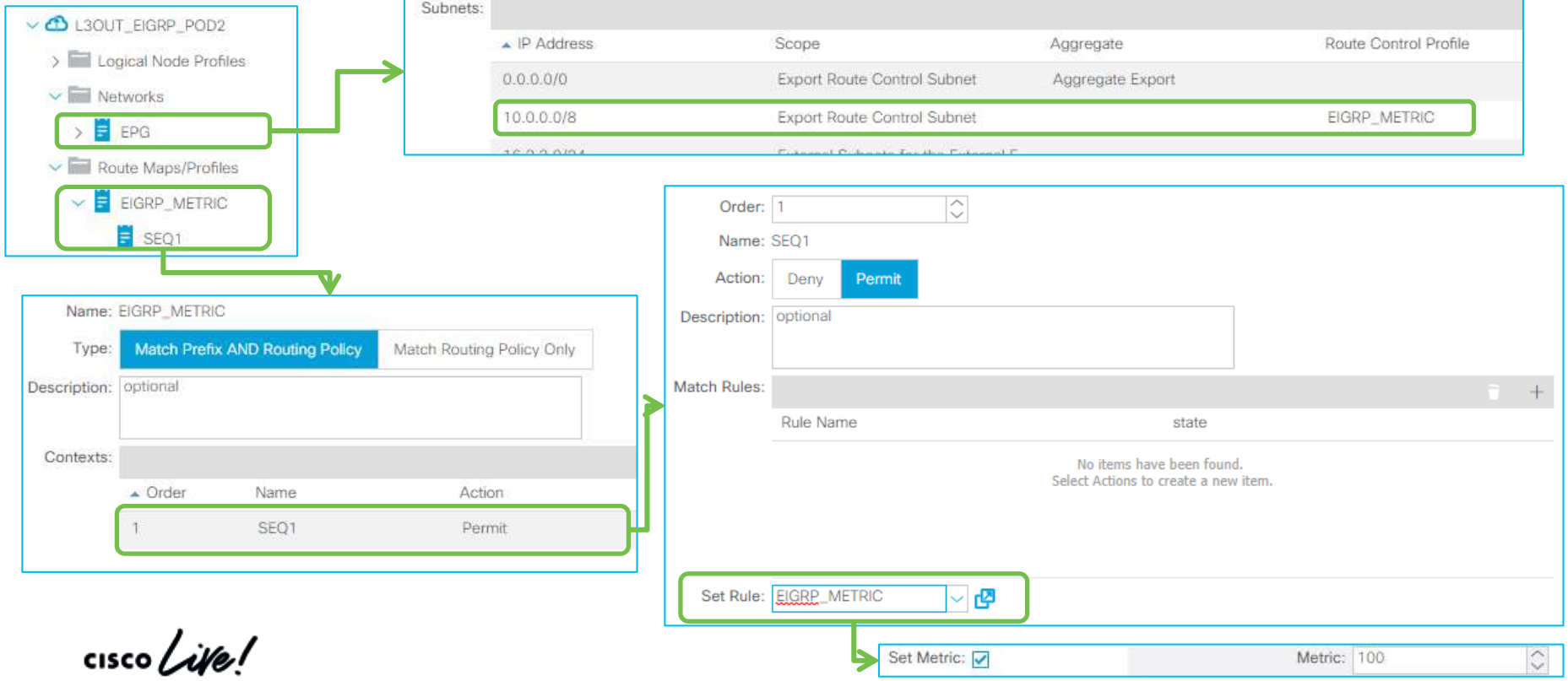


EIGRP & MP-BGP Redistribution Solution2



EIGRP & MP-BGP Redistribution Solution2

Add “set metric” rule to export route-map

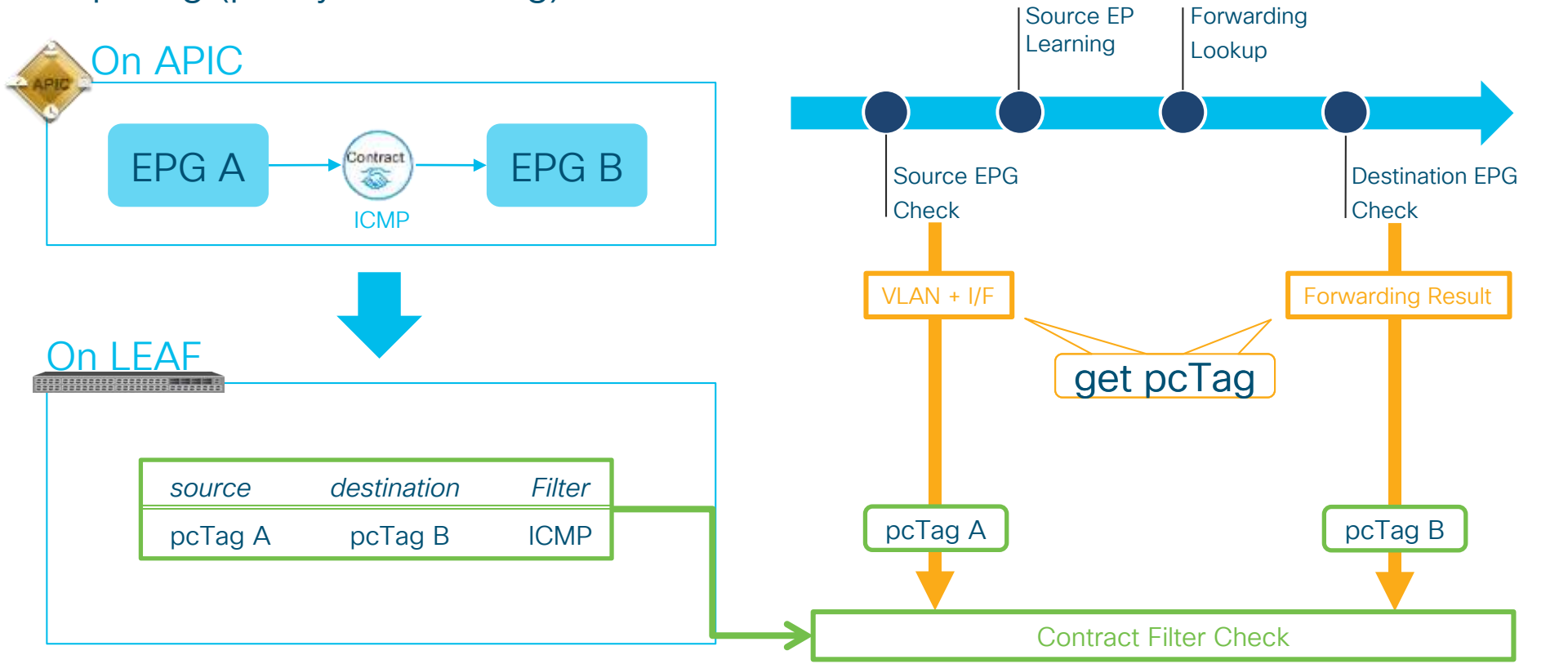


cisco Live!

L3Out Contract deep dive

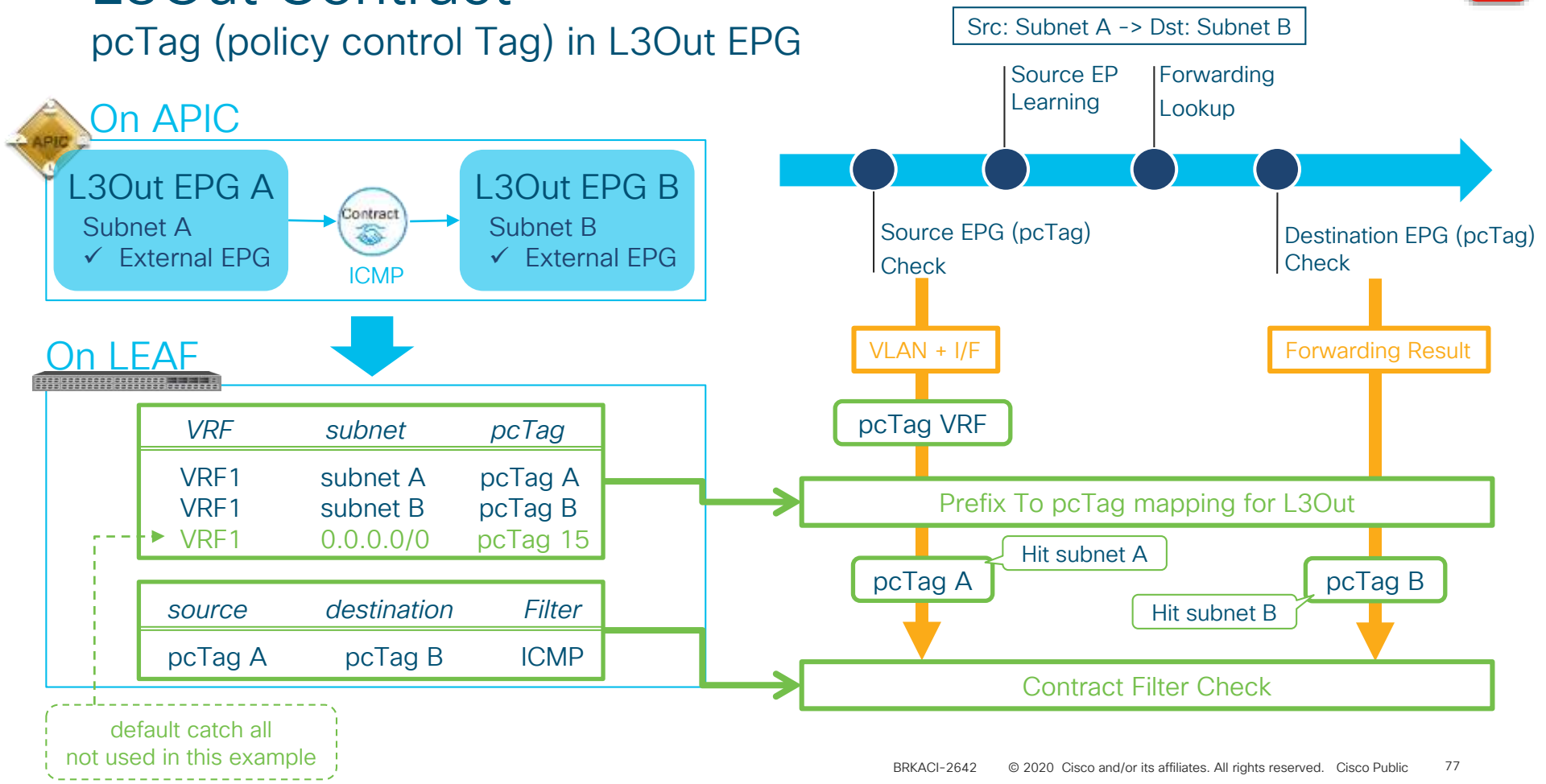
L3Out Contract

pcTag (policy control Tag) in normal EPG



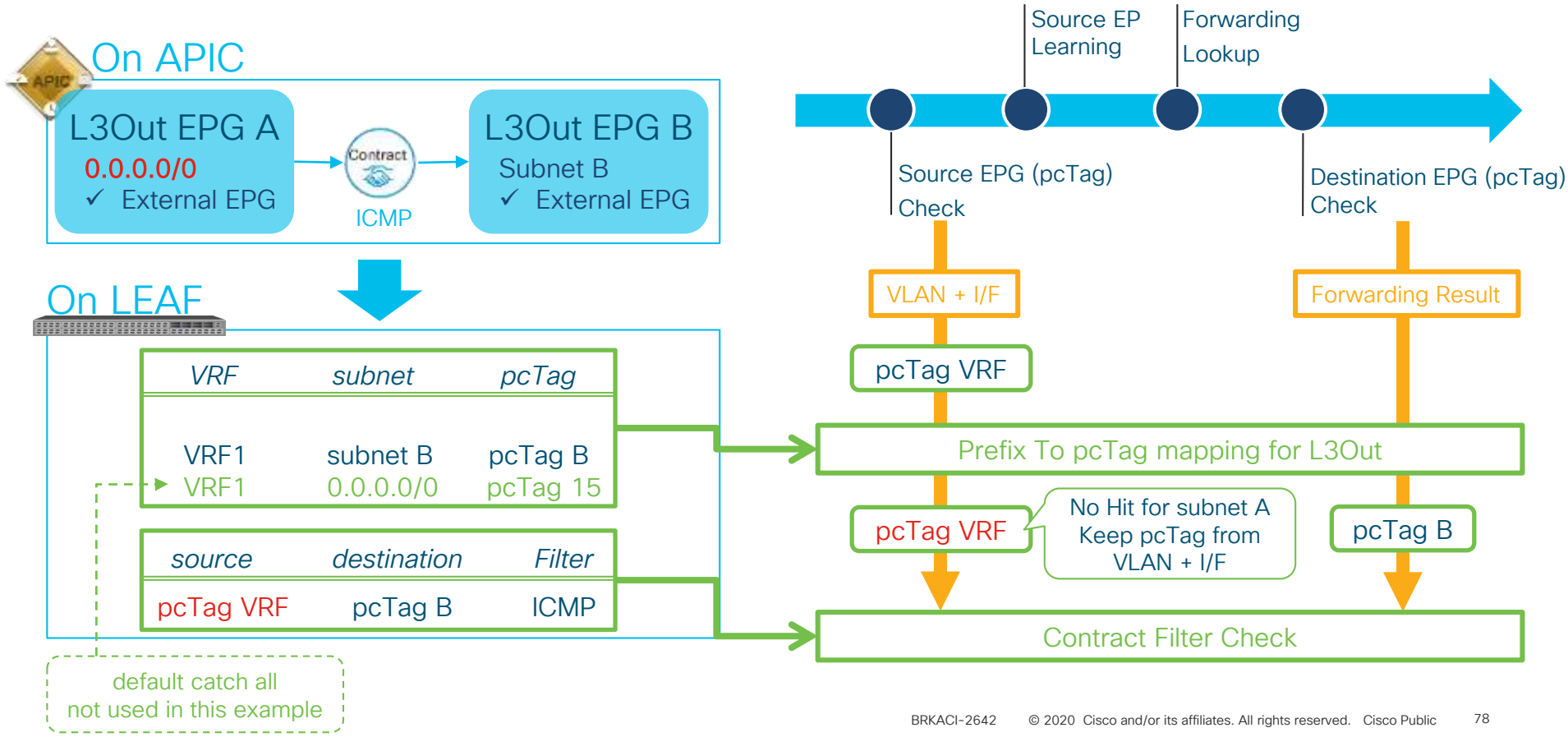
L3Out Contract

pcTag (policy control Tag) in L3Out EPG



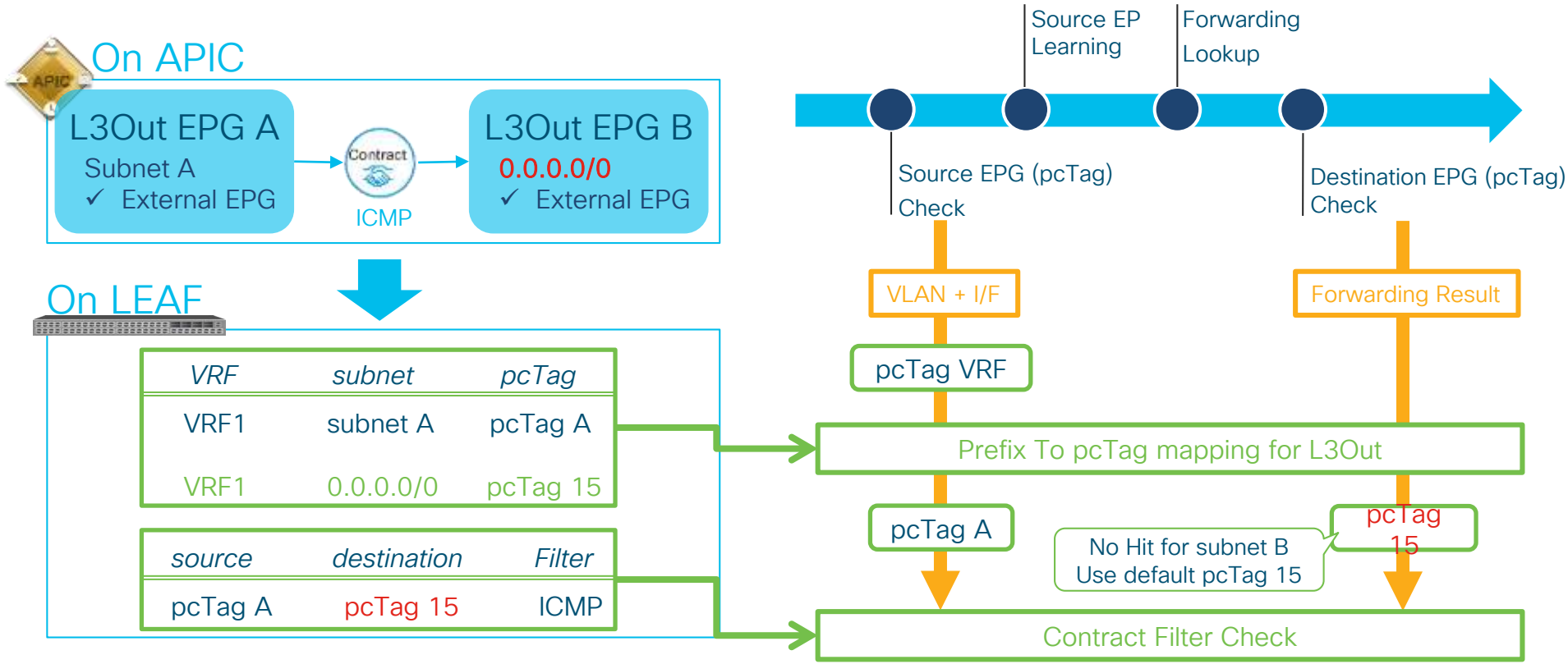
L3Out Contract

pcTag (policy control Tag) in L3Out EPG with 0.0.0.0/0



L3Out Contract

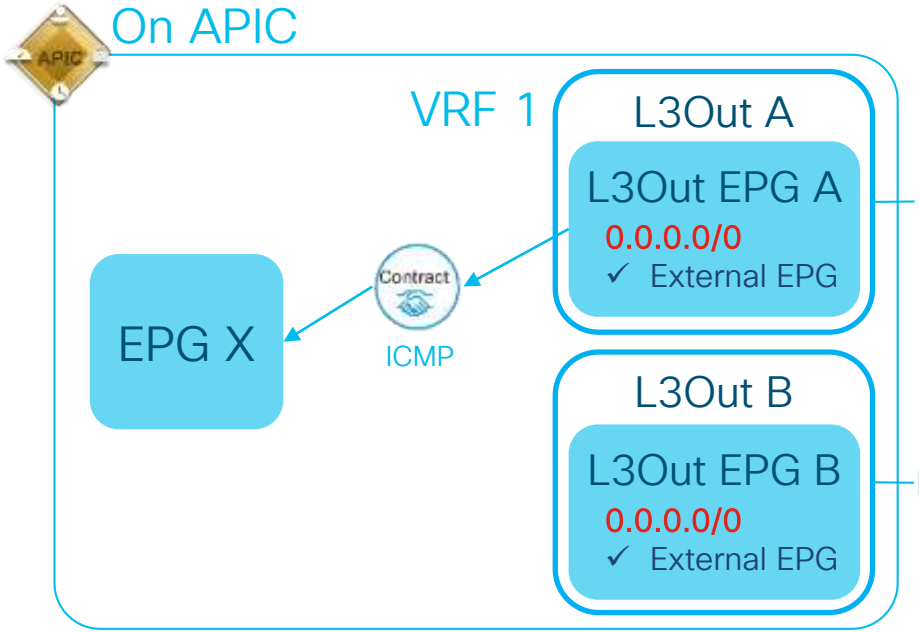
pcTag (policy control Tag) in L3Out EPG with 0.0.0.0/0



L3Out Contract

Common Issue (L3Out EPGs with 0.0.0.0/0)

On APIC



On LEAF

Prefix-pcTag entry is per VRF.
Default catch all (0.0.0.0) is shared with everyone in the VRF.

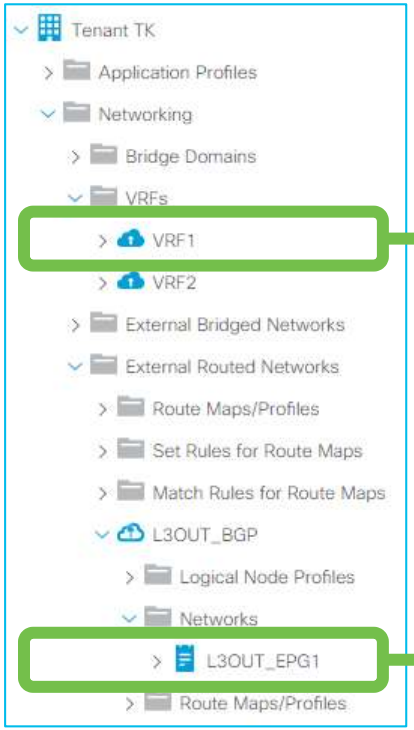
VRF	*subnet*	*pcTag*
VRF1	0.0.0.0/0	pcTag 15

source	*destination*	*Filter*
pcTag X	pcTag 15	ICMP
pcTag VRF	pcTag X	ICMP

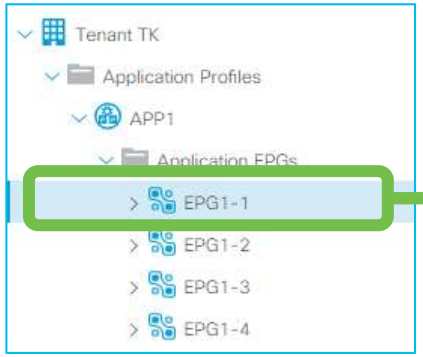
These contracts are from EPG X and L3Out A
However, traffic from/to L3Out B (20.0.0.0/8) will also use default pcTag (VRF or 15) due to 0.0.0.0/0 config.

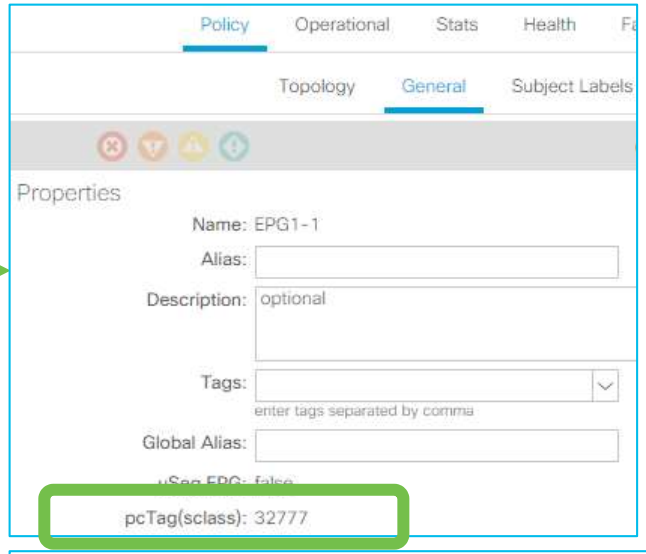
No overlap of External EPG L3Out subnets in same VRF
Use 0.0.0.0/0 (External subnet for the external EPG) only for one L3Out EPG per VRF

How to get pcTag for L3Out



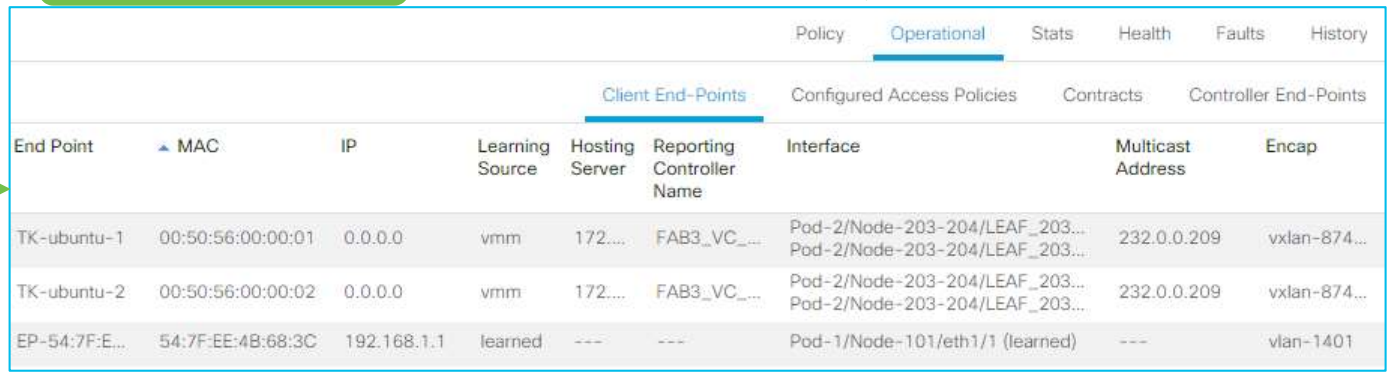
How to get pcTag for normal EPG





== Policy tab ==
➢ Check EPG's pcTag

== Operational tab ==
➢ Check if the endpoint is learned on the expected EPG



End Point	MAC	IP	Learning Source	Hosting Server	Reporting Controller Name	Interface	Multicast Address	Encap
TK-ubuntu-1	00:50:56:00:00:01	0.0.0.0	vmm	172....	FAB3_VC_...	Pod-2/Node-203-204/LEAF_203... Pod-2/Node-203-204/LEAF_203...	232.0.0.209	vxlan-874...
TK-ubuntu-2	00:50:56:00:00:02	0.0.0.0	vmm	172....	FAB3_VC_...	Pod-2/Node-203-204/LEAF_203... Pod-2/Node-203-204/LEAF_203...	232.0.0.209	vxlan-874...
EP-54:7F:E...	54:7F:EE:4B:68:3C	192.168.1.1	learned	---	---	Pod-1/Node-101/eth1/1 (learned)	---	vlan-1401

How to get VRF pcTag

- From APIC

```
admin@apic1:~> moquery -c fvCtx -f 'fv.Ctx.name=="VRF1"' | egrep '#|dn|pcTag'
# fv.Ctx
dn                  : uni/tn-TK/ctx-VRF1
pcTag               : 49153
```

- From LEAF

```
leaf# vsh_lc -c 'show system internal eltmc info vrf TK:VRF1' | grep sclass
                scope:              4   :::          sclass:          49153
```

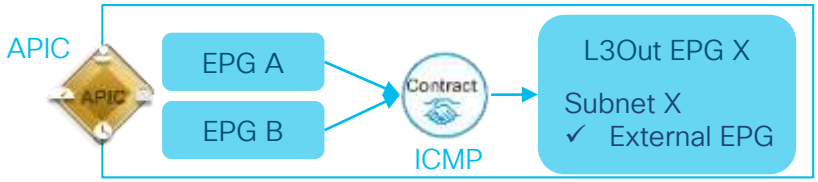
L3Out Contract

Policy Control Enforcement Direction

Under VRF

Policy Control Enforcement Direction: **Egress** Ingress

A feature to save contract TCAM usage on border LEAF



No effects on
EPG <-> EPG traffic

Egress Policy Enforcement

Policy Control Enforcement Direction: **Egress** Ingress

Non-Border LEAF(s)

with EPG A

source	destination	Filter
pcTag A	pcTag X	ICMP

with EPG B

source	destination	Filter
pcTag B	pcTag X	ICMP

Border LEAF(s)

source	destination	Filter
pcTag A	pcTag X	ICMP
pcTag B	pcTag X	ICMP

Ingress Policy Enforcement

default from 1.2

Policy Control Enforcement Direction: Egress **Ingress**

Non-Border LEAF(s)

with EPG A

source	destination	Filter
pcTag A	pcTag X	ICMP

with EPG B

source	destination	Filter
pcTag B	pcTag X	ICMP

Border LEAF(s)

source	destination	Filter
- none -		

L3Out Contract

Policy Control Enforcement Direction

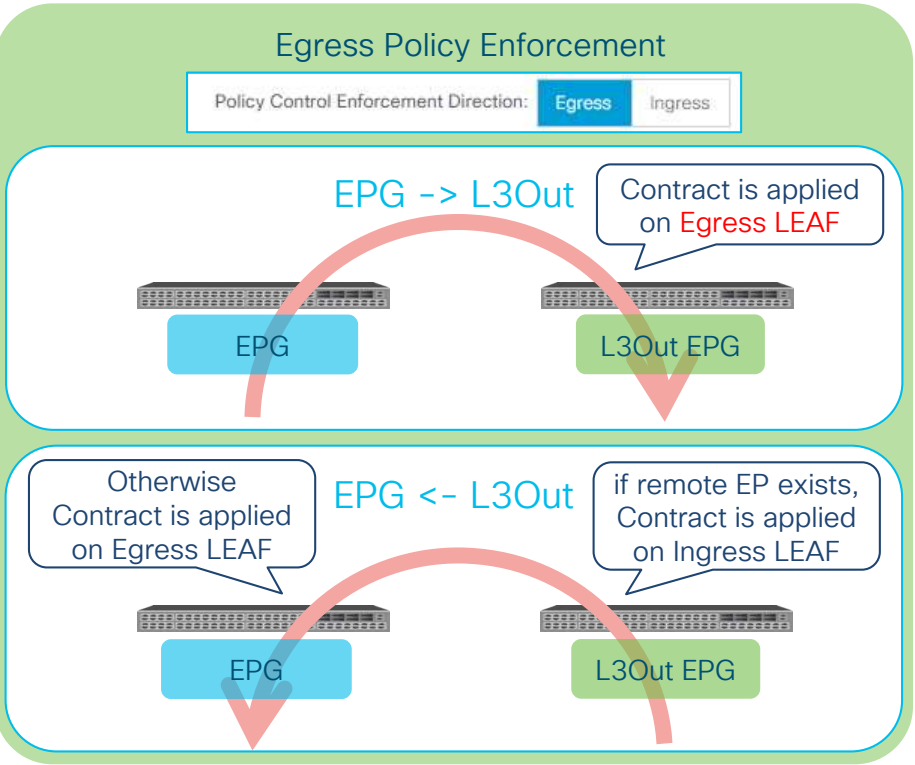
Under VRF

Policy Control Enforcement Direction: Egress | Ingress

How does it affect traffic flow and contract?

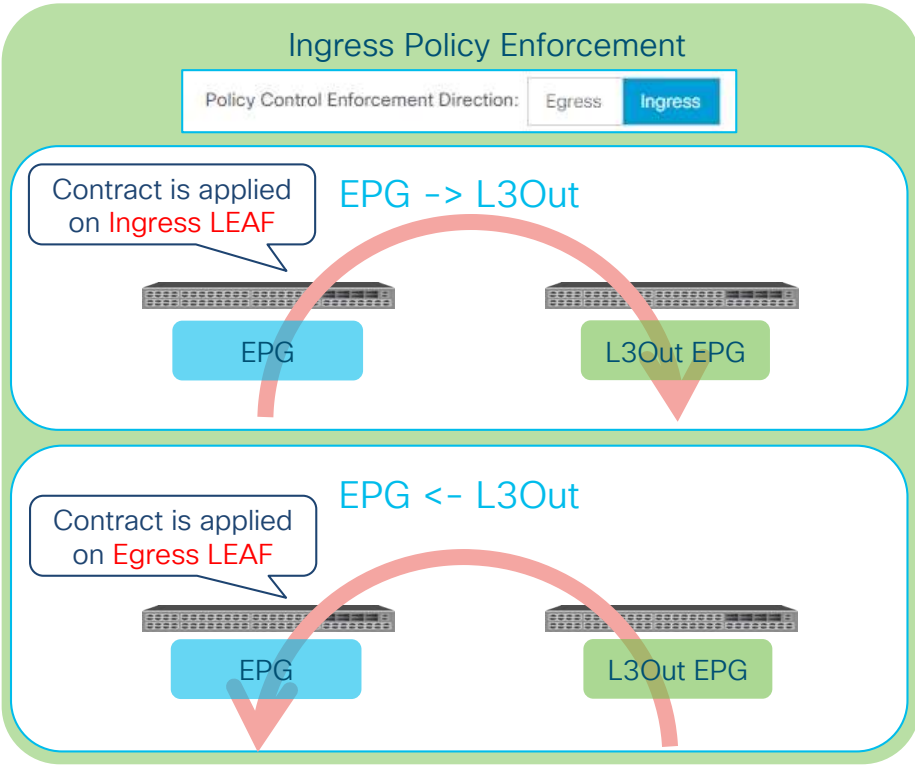
Egress Policy Enforcement

Policy Control Enforcement Direction: Egress | Ingress



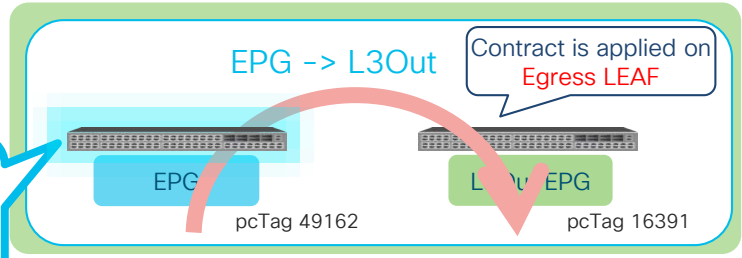
Ingress Policy Enforcement

Policy Control Enforcement Direction: Egress | Ingress



L3Out Contract

CLI Verification



Ingress Compute LEAF Verification for

Policy Control Enforcement Direction: Egress | Ingress

Points to the correct border leaf TEP

Routing Table

```
leaf# show ip route 52.52.52.0/24 vrf TK:VRF1
52.52.52.0/24, ubest/mbest: 1/0
    *via 11.0.64.64%overlay-1, [200/5], 00:00:14, bgp-65003, internal,
tag 65003
         recursive next hop: 11.0.64.64/32%overlay-1
```

Prefix-pcTag mapping table

```
leaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152

leaf# vsh -c 'show system internal policy-mgr prefix' | grep 2097152
leaf#       ---- no output ----
```

No prefix-pcTag for a pure compute LEAF
note: before 3.2 release, use
vsh_lc -c 'show system internal aclqos prefix'

Hardware table
(2nd gen or later LEAF -EX, -FX, -FX2 etc.)

```
leaf# vsh_lc -c 'show system internal eltmc info vrf TK:VRF1' | grep hw_vrf_idx
              vrf_id:              8   :::      hw_vrf_idx:           4614

leaf# vsh_lc -c 'show platform internal hal l3 routes' | egrep 'VRF|52.52.52'
| VRF |      Prefix/Len| RT| RID |    LID    | Type| PID | FPID/| HIT |N|  NB-ID  |  NB Hw | PID | FPID/|   TBI    |TRO|Ifindex|CLSS|CLP| AI |SH|DH| Flags |
| 4614|   52.52.52.0/ 24| UC|  1a2|       117| TCAM| 1823|     0| 1823|E|       1a|      12|   NA|   NA|       NA| NA|      0|   1|  3|   0| 0| 0|
```

check hw_vrf index

pcTag (CLSS) is 1 to bypass contract

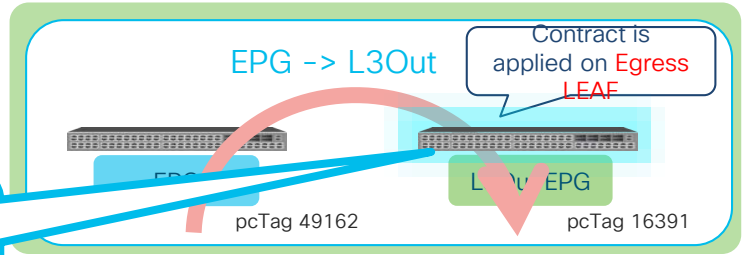
Check contract on LEAF (zoning-rule)

```
leaf# show zoning-rule scope 2097152 | egrep 'Rule|16391'
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action                          Priority
4142            49162           16391           5               enabled         2097152         permit                          fully_qual(7)
```

Zoning-rule on compute LEAF in egress mode (not used in this case)

L3Out Contract

CLI Verification



Egress border LEAF Verification for **Egress** Policy Enforcement

Policy Control Enforcement Direction: Egress | Ingress

Points to the correct external next-hop IP

Routing Table

```
Bleaf# show ip route 52.52.52.0/24 vrf TK:VRF1
52.52.52.0/24, ubest/mbest: 1/0
    *via 15.2.2.1, vlan37, [110/5], 00:25:42, ospf-default, intra
```

Prefix-pcTag mapping table

```
Bleaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152

Bleaf# vsh -c 'show system internal policy-mgr prefix' | grep 52.52.52
Vrf-Vni VRF-Id ~snip~ VRF-Name           Addr     Class Shared Remote
Complete
2097152 8      ~snip~ TK:VRF1    52.52.52.0/24    16391   False  False  False
```

prefix-pcTag mapping for L3Out EPG
note: before 3.2 release, use
vsh_lc -c 'show system internal aclqos prefix'

Hardware table
(2nd gen or later LEAF -EX, -FX, -FX2 etc.)

```
Bleaf# vsh_lc -c 'show system internal eltmc info vrf TK:VRF1' | grep hw_vrf_idx
              vrf_id:              8   :::      hw_vrf_idx:           4614

Bleaf# vsh_lc -c 'show platform internal hal l3 routes' | egrep 'VRF|52.52.52'
| VRF  |     Prefix/Len| RT| RID |     LID    | Type| PID | FPID/| HIT |N|  NB-ID  |  NB Hw | PID | FPID/|    TBI    |TRO|Ifindex|CLSS|CLP| AI |SH|DH| Flags |
| 4614 | 52.52.52.0/ 24| UC|  121|        68| TCAM|  80f|     0|  80f|A|     7afd|    80df|   NA|    NA|        NA| NA|      0|4007|  2|  0| 1| 0|spi,dpi
```

check hw_vrf index

pcTag (CLSS) is 16391 (0x4007)

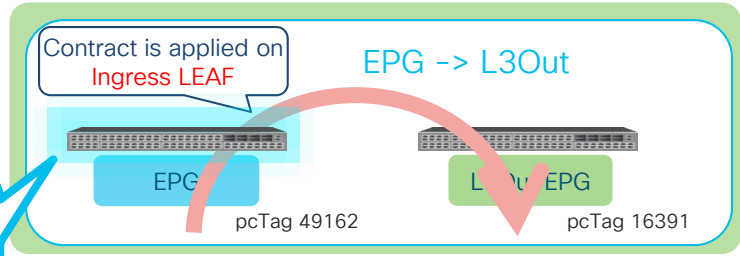
Check contract on LEAF (zoning-rule)

Zoning-rule on border LEAF in egress mode

```
Bleaf# show zoning-rule scope 2097152 | egrep 'Rule|16391'
Rule ID      SrcEPG          DstEPG          FilterID        operSt          Scope           Action                          Priority
4142         49162           16391           5               enabled         2097152         permit                          fully_qual(7)
```

L3Out Contract

CLI Verification



Ingress Compute LEAF Verification for

Policy Control Enforcement Direction: Egress | Ingress

Points to the correct border leaf TEP

Routing Table

```
leaf# show ip route 52.52.52.0/24 vrf TK:VRF1
52.52.52.0/24, ubest/mbest: 1/0
    *via 11.0.64.64%overlay-1, [200/5], 00:00:14, bgp-65003, internal,
tag 65003
         recursive next hop: 11.0.64.64/32%overlay-1
```

Prefix-pcTag mapping table

```
leaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152

leaf# vsh -c 'show system internal policy-mgr prefix' | grep 52.52.52
Vrf-Vni VRF-Id ~snip~ VRF-Name          Addr     Class Shared Remote
Complete
2097152 8      ~snip~ TK:VRF1    52.52.52.0/24    16391   False  True   False
```

prefix-pcTag mapping for L3Out EPG
note: before 3.2 release, use
vsh_lc -c 'show system internal aclqos prefix'

Hardware table
(2nd gen or later LEAF -EX, -FX, -FX2 etc.)

```
leaf# vsh_lc -c 'show system internal eltmc info vrf TK:VRF1' | grep hw_vrf_idx
              vrf_id:              8   :::      hw_vrf_idx:           4614

leaf# vsh_lc -c 'show platform internal hal l3 routes' | egrep 'VRF|52.52.52'
| VRF |     Prefix/Len| RT| RID |    LID    | Type| PID | FPID/| HIT |N|  NB-ID  |  NB Hw | PID | FPID/|   TBI   |TRO|Ifindex|CLSS|CLP| AI |SH|DH| Flags |
| 4614|  52.52.52.0/ 24| UC|  1a2|       117| TCAM| 1823|     0| 1823|A|     7c82|    830a|   NA|   NA|       NA| NA|      0|4007|  2|   0| 0| 0|spi,dpi
```

check hw_vrf index

pcTag (CLSS) is 16391 (0x4007)

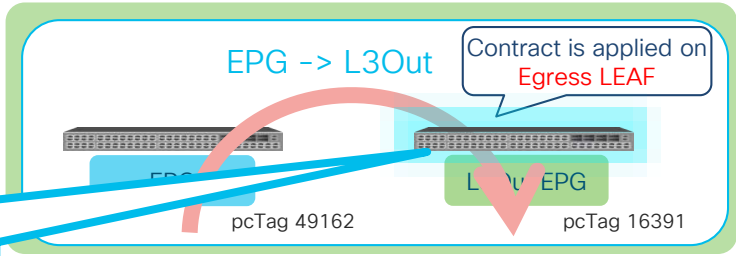
Check contract on LEAF (zoning-rule)

Zoning-rule on compute LEAF in egress mode

```
leaf# show zoning-rule scope 2097152 | egrep 'Rule|16391'
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action                          Priority
4142            49162           16391           5               enabled         2097152         permit                          fully_qual(7)
```

L3Out Contract

CLI Verification



Egress border LEAF Verification for **Ingress** Policy Enforcement

Policy Control Enforcement Direction: Egress | Ingress

Points to the correct external next-hop IP

Routing Table

```
Bleaf# show ip route 52.52.52.0/24 vrf TK:VRF1
52.52.52.0/24, ubest/mbest: 1/0
    *via 15.2.2.1, vlan37, [110/5], 00:25:42, ospf-default, intra
```

Prefix-pcTag mapping table

```
Bleaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152

Bleaf# vsh -c 'show system internal policy-mgr prefix' | grep 52.52.52
Vrf-Vni VRF-Id ~snip~ VRF-Name           Addr     Class Shared Remote
Complete
2097152 8      ~snip~ TK:VRF1    52.52.52.0/24    16391   False  False  False
```

prefix-pcTag mapping for L3Out EPG
note: before 3.2 release, use
vsh_lc -c 'show system internal aclqos prefix'

Hardware table
(2nd gen or later LEAF -EX, -FX, -FX2 etc.)

```
Bleaf# vsh_lc -c 'show system internal eltmc info vrf TK:VRF1' | grep hw_vrf_idx
              vrf_id:              8   :::      hw_vrf_idx:           4614

Bleaf# vsh_lc -c 'show platform internal hal l3 routes' | egrep 'VRF|52.52.52'
| VRF  |     Prefix/Len| RT| RID |     LID    | Type| PID | FPID/| HIT |N|  NB-ID  |  NB Hw | PID | FPID/|    TBI    |TRO|Ifindex|CLSS|CLP| AI |SH|DH| Flags |
| 4614 |  52.52.52.0/ 24| UC|  121|        68| TCAM|  80f|     0|  80f|A|     7afd|    80df|   NA|    NA|        NA| NA|      0|4007|  2|   0| 1| 0|spi,dpi
```

check hw_vrf index

pcTag (CLSS) is 16391 (0x4007)

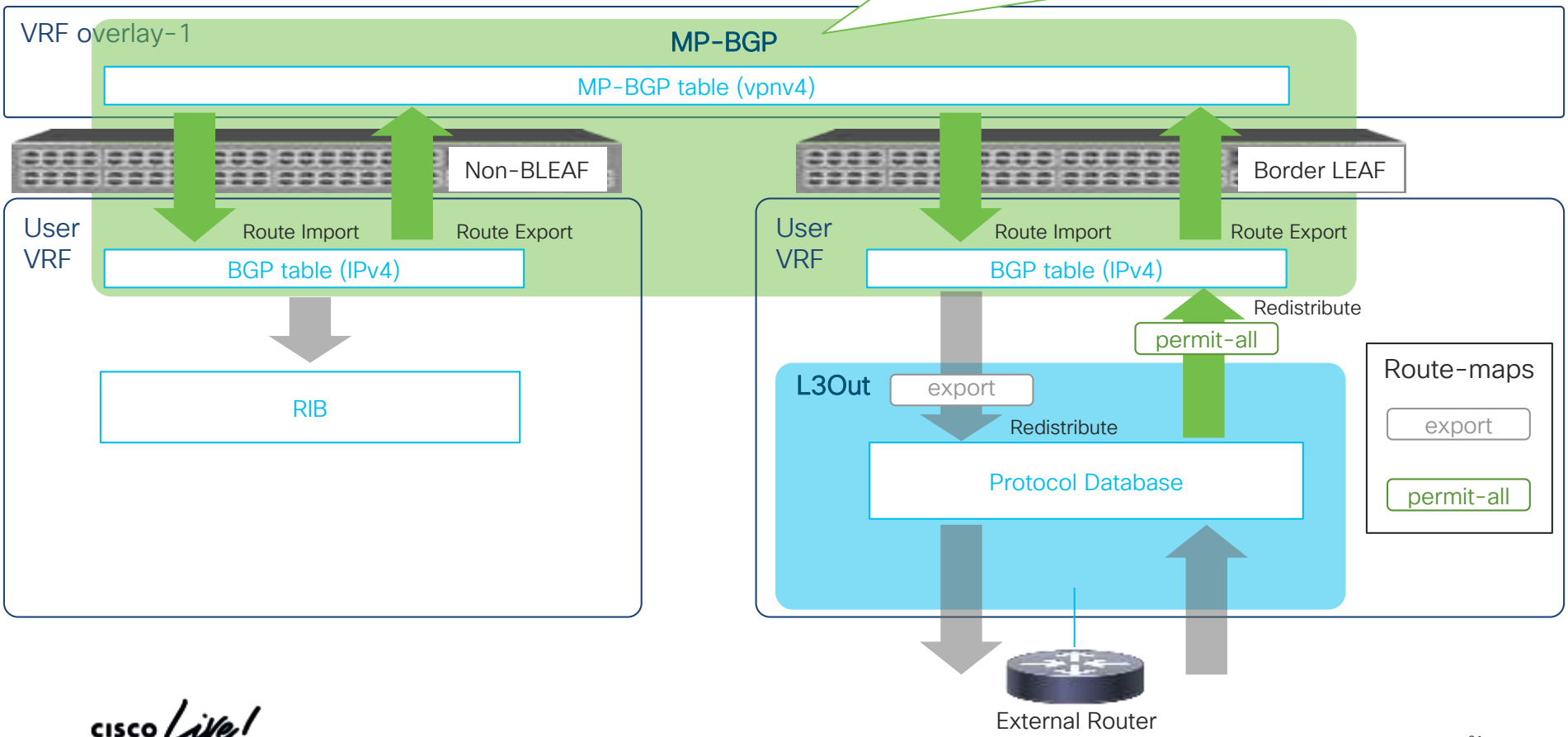
Check contract on LEAF (zoning-rule)

No zoning-rule on border LEAF in ingress mode

```
Bleaf# show zoning-rule scope 2097152 | egrep 'Rule|16391'
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action                          Priority
--- none ---
```

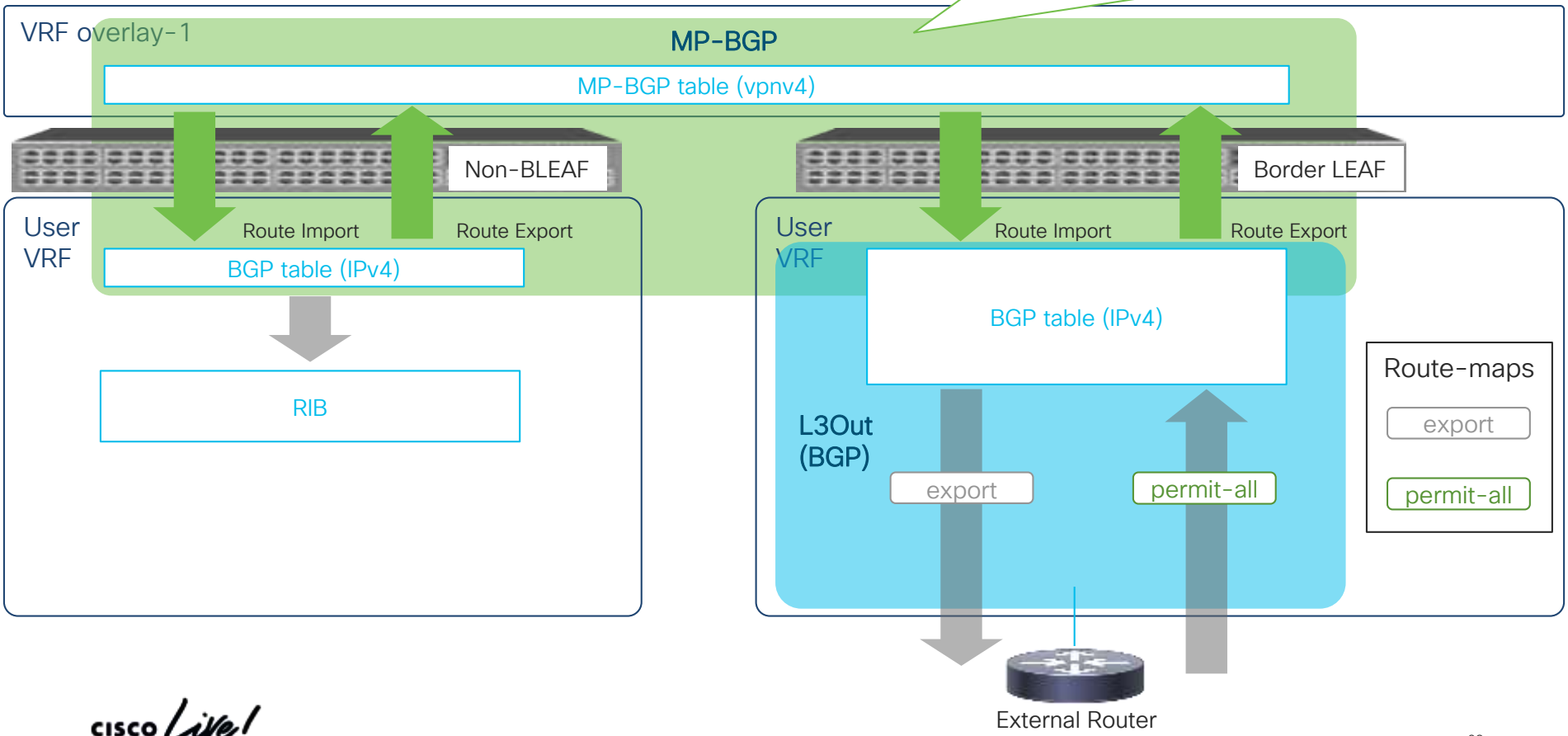
MP-BGP Deep Dive

MP-BGP



MP-BGP with L3Out BGP

MP-BGP is automatically deployed once Route Reflector (and MP-BGP AS) is configured



CLI Verifications

1. BGP process in your VRF with expected Redistribution and Route-Target

```
border-leaf# show bgp process vrf TK:VRF1

VRF RD                              : 10.0.184.64:2

 Information for address family IPv4 Unicast in VRF TK:VRF1
    Redistribution
        direct, route-map permit-all
        static, route-map imp-ctx-bgp-st-interleak-2097152
        eigrp, route-map permit-all
        ospf, route-map permit-all
    Export RT list:
        65003:2097152
    Import RT list:
        65003:2097152

 Information for address family IPv6 Unicast in VRF TK:VRF1
    --- snip ---
```

Automatically created regardless of routing protocol used in L3Out. If not, check Route Reflector policy on APIC

- VRF RD (Route Distinguisher) is based on TEP IP
- BGP redistributes (almost) all external routes to export them into MP-BGP vpnv4 by default.
  Check a later page for the exception on BD subnets (direct routes).
- RT (Route Target) is based on ACI BGP AS and VRF VNID.

2. External routes are redistributed/exported into VPNv4 in VRF overlay-1

```
border-leaf# show bgp vpnv4 unicast vrf overlay-1

 Network            Next Hop            Metric     LocPrf     Weight Path

Route Distinguisher: 10.0.184.64:2    (VRF TK:VRF1)
* i5.5.5.0/24         10.0.184.67              5        100          0 ?
*>r                   0.0.0.0                  5        100      32768 ?
* i15.0.0.0/24        10.0.184.67              0        100          0 ?
*>r                   0.0.0.0                  0        100      32768 ?
```

MP-BGP VPNv4 table can be checked via normal CLI in vrf overlay-1

NOTE:
This example shows two routes are learned locally (r with next-hop 0.0.0.0) and also from another leaf with TEP 10.0.184.67.

CLI Verifications

3. MP-BGP on all leaves should have all external routes in VPNv4 format in VRF overlay-1

```
non-border-leaf# show bgp vpnv4 unicast vrf overlay-1

Route Distinguisher: 10.0.184.64:2
*>i5.5.5.0/24         10.0.184.64              5        100          0 ?
* i                    10.0.184.64              5        100          0 ?

Route Distinguisher: 10.0.184.67:1
*>i5.5.5.0/24         10.0.184.67              5        100          0 ?
* i                    10.0.184.67              5        100          0 ?
```

5.5.5.0/24 is advertised from border-leaf1 (10.0.184.64) and border-leaf2 (10.0.184.67)

Two entries with the same next-hop LEAF TEP means there are two Route Reflectors.

```
non-border-leaf# show bgp vpnv4 unicast 5.5.5.0/24 vrf overlay-1
Route Distinguisher: 10.0.184.67:1
BGP routing table entry for 5.5.5.0/24, version 598 dest ptr 0xaa7e840c

  AS-Path: NONE, path sourced internal to AS
    10.0.184.67 (metric 3) from 10.0.184.65 (1.1.1.101)
      Extcommunity:
          RT:65003:2097152
```

Each VPNv4 route in VRF overlay-1 has RT with its original VRF VNID

4. BGP process is running also in your VRF on non-border-leaf for MP-BGP

```
non-border-leaf# show bgp process vrf TK:VRF1
 Information for address family IPv4 Unicast in VRF TK:VRF1
    Export RT list:
        65003:2097152
    Import RT list:
        65003:2097152
 Information for address family IPv6 Unicast in VRF TK:VRF1
    --- snip ---
```

IPv4 BGP imports all the external routes for its own VRF based on RT from VPNv4 table

If BGP is not running in your VRF on non-border-leaf, check Route Reflector Policy config on APIC

CLI Verifications

5. The external routes are imported in IPv4 BGP table based on RT

```
non-border-leaf# show bgp ipv4 unicast vrf TK:VRF1

   Network            Next Hop            Metric     LocPrf     Weight Path
*>i5.5.5.0/24         10.0.184.64              5        100          0 ?
*|i                   10.0.184.67              5        100          0 ?
```

6. The routing table shows border leaves as next-hop learned from iBGP

```
non-border-leaf# show ip route 5.5.5.0/24 vrf TK:VRF1
5.5.5.0/24, ubest/mbest: 2/0
    *via 10.0.184.67%overlay-1, [200/5], 2d10h, bgp-65003, internal, tag 65003
         recursive next hop: 10.0.184.67/32%overlay-1
    *via 10.0.184.64%overlay-1, [200/5], 2d10h, bgp-65003, internal, tag 65003
         recursive next hop: 10.0.184.64/32%overlay-1
```

CLI Verifications for BD subnet exception

```
border-leaf# show bgp vpnv4 unicast neighbors vrf overlay-1

BGP neighbor is 10.0.184.65,  remote AS 65003, ibgp link,  Peer index 1

  For address family: VPNv4 Unicast
  Outbound route-map configured is deny-pervasive, handle obtained


BGP neighbor is 10.0.184.66,  remote AS 65003, ibgp link,  Peer index 2

  For address family: VPNv4 Unicast
  Outbound route-map configured is deny-pervasive, handle obtained


border-leaf# show route-map deny-pervasive
route-map deny-pervasive, deny, sequence 1
  Match clauses:
    pervasive: 2
  Set clauses:
route-map deny-pervasive, deny, sequence 2
  Match clauses:
    interface: Null0
  Set clauses:
route-map deny-pervasive, permit, sequence 3
  Match clauses:
  Set clauses:
```

BD subnets and Null0 I/F should not be distributed via MP-BGP.

- Outbound route-map to BGP Route Reflector (spines) limits BD subnets (pervasive routes) and Null0.

In this example, 10.0.184.65 and 10.0.184.66 are RR spines.

BD subnets are deployed based on object policies from APIC instead of routing protocol

Reference

ACI Fabric L3Out Guide -
https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/guide-c07-743150.html

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

cisco Live!

Continue your education

Demos in the Cisco campus

Walk-in labs

Meet the engineer 1:1 meetings

Related sessions

Thank you

cisco Live!

You make **possible**