# ACI Deep Dive

## Contract and policy enforcement

10 Dec 2019

Roland Ducomble – rducombl@cisco.com

Cisco TS Technical Leader – ACI Solution Support Team

CCIE 3745

V1.1

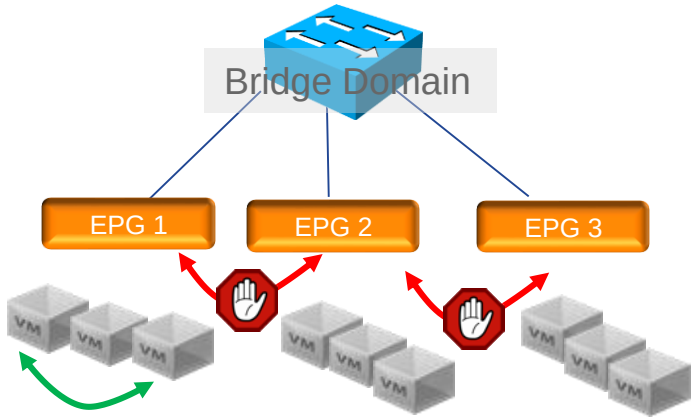
# Agenda

- Contract introduction
- Contract configuration
- Contract operational simplification
- Checking Contract and reading zoning-rule
- Contract Priority rules
- Example of TCAM usage
- L4 operator
- Policy compression

# Contract and policy enforcement introduction

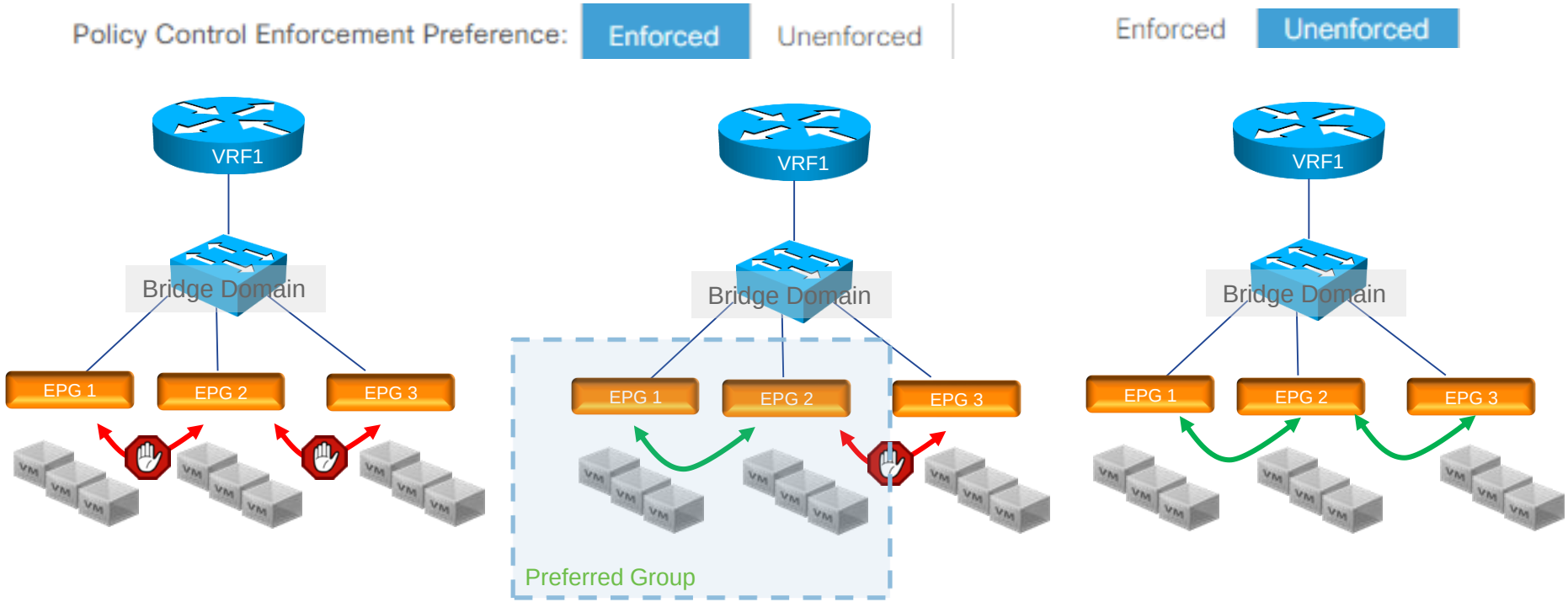
# Traffic between EPGs is not allowed by Default

- Every Bridge Domain can be segmented in Security Zones called EPGs
- Traffic within the EPG is allowed
- Traffic between EPGs by default is not allowed



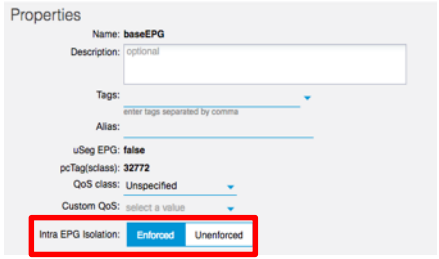

- EPGs can be further segmented into
- Micro EPGs based on: MAC, IP, VM tags etc...
- or Isolated EPGs

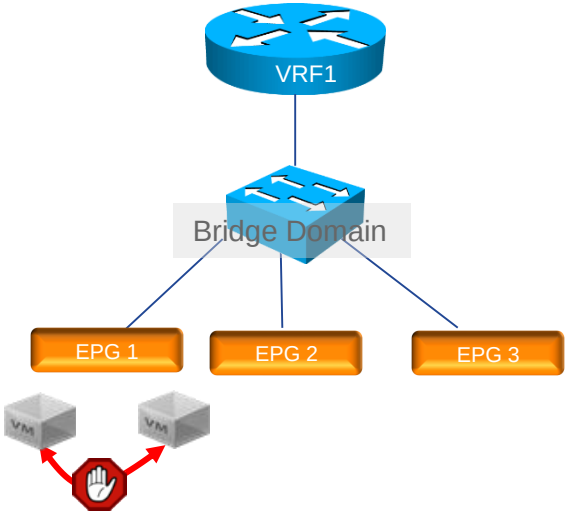
# VRFs can be configured not to allow EPG-to-EPG communication, or to allow it, or to partially allow it



# You can also configure EPGs for intra EPG Isolation

- Intra EPG Isolation **blocks communication between all endpoints** inside the group
- In the same EPG you can mix Physical and Virtual endpoints
- This is similar to the concept of Private VLANs (Isolated VLANs)



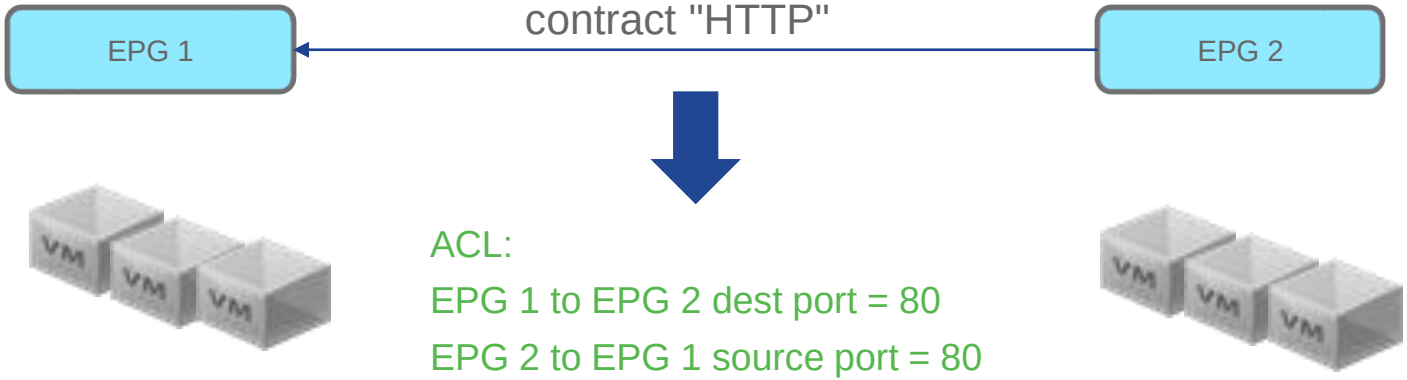




```
<fvTenant name="Tenant1">
  <fvAp name="ap1">
    <fvAEPg isAttrBasedEPg="no" matchT="AtleastOne"
name="baseEPG" pcEnfPref="enforced" prefGrMemb="exclude"
prio="unspecified">
      <fvRsBd tnFvBDName="bd"/>
    </fvAEPg>
  </fvAp>
</fvTenant>
```

# Communication between EPGs is configured via "contracts"

EPG 1 consumes "HTTP"

EPG 2 provides "HTTP„




ACL:

EPG 1 to EPG 2 dest port = 80

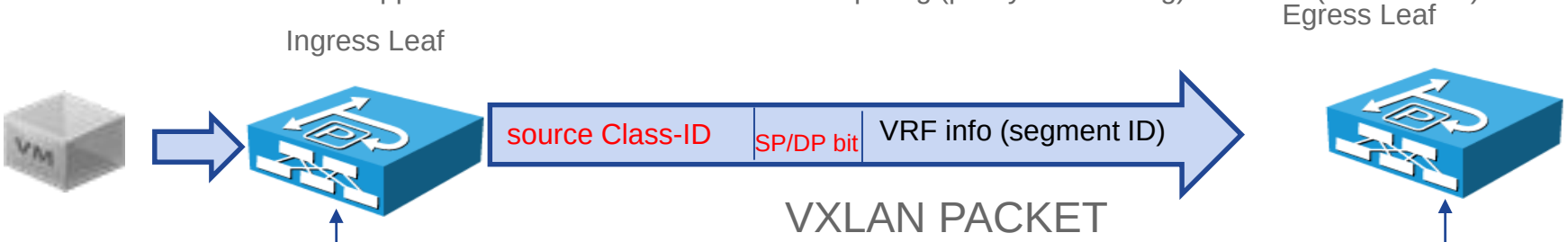
EPG 2 to EPG 1 source port = 80

# What are Contracts in ACI

- They don’t steer traffic (except for service graph)
- They are ACLs
- They control redistribution of routes between VRFs
- Contracts are semantics to specify End Point Group (EPG) to EPG communication in ACI Fabric
- Contracts can be between EPGs or between L3out and EPGs
- Filters take space in the Policy CAM

# What happens on the wire

In hardware each EPG is mapped to a number called "class-id“ or pcTag (policy Control Tag) or sclass (in switch cli)

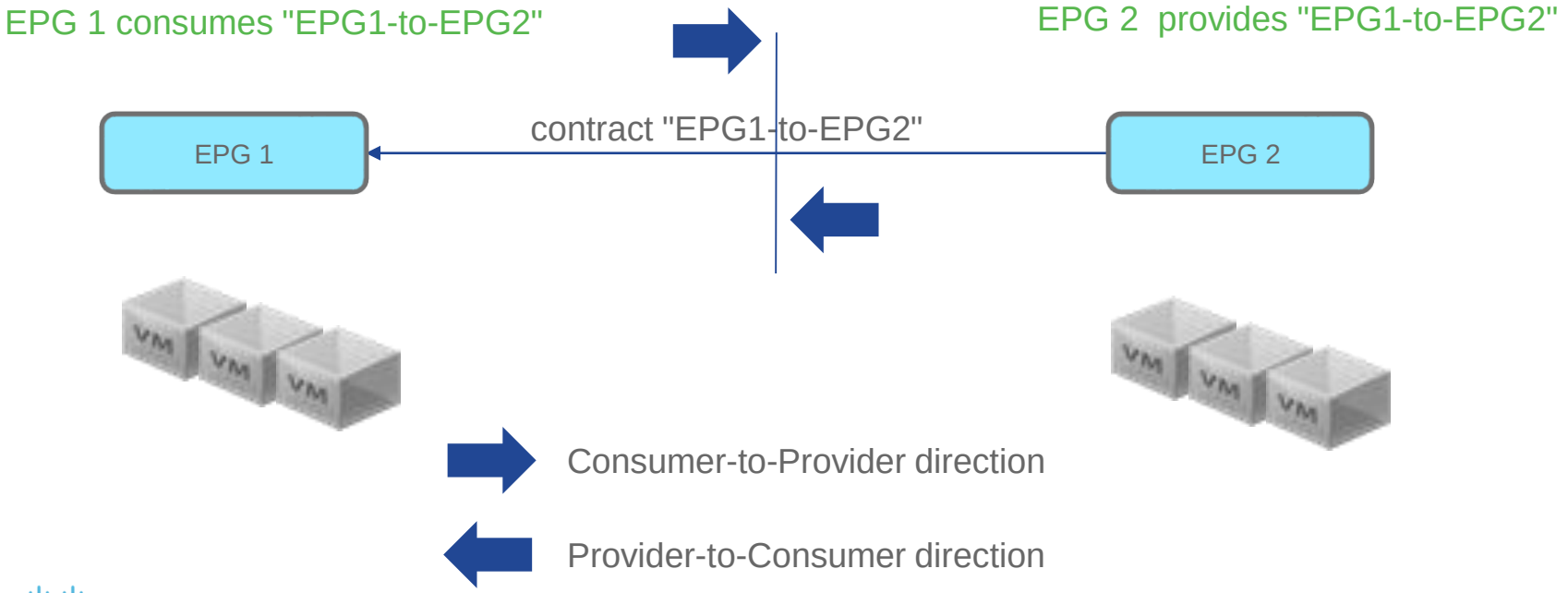



Ingress Leaf assignes the source Class-ID to the packet based on ingress vlan (epg) or src IP (l3 out)
And if the /32 of the destination is known it can also derive the Destination Class—id
And apply the policy (ACL) → sets Policy Applied bit in VXLAN header if done (SP/DP bit)

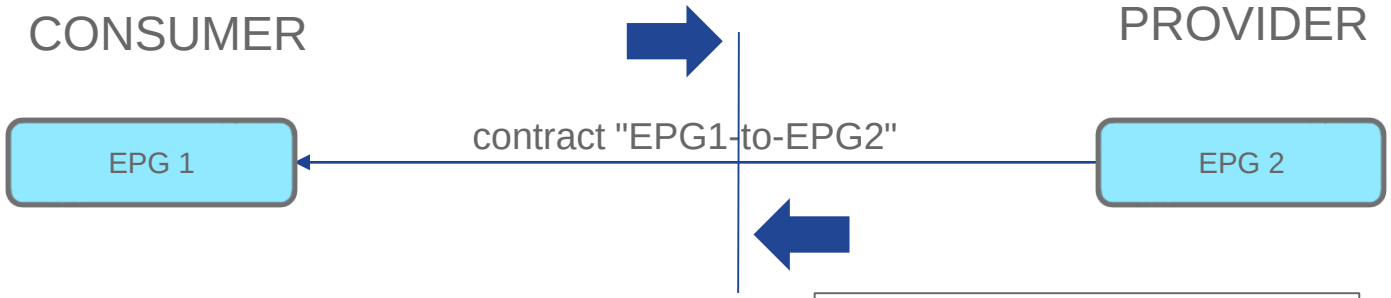
If policy has not been applied, egress leaf checks the EPG that The destination belongs to and applies the policy filtering rule


**SP: Indicates Source Policy (ACL) has been applied**
**DP: Indicates Destination Policy (ACL) has been applied**

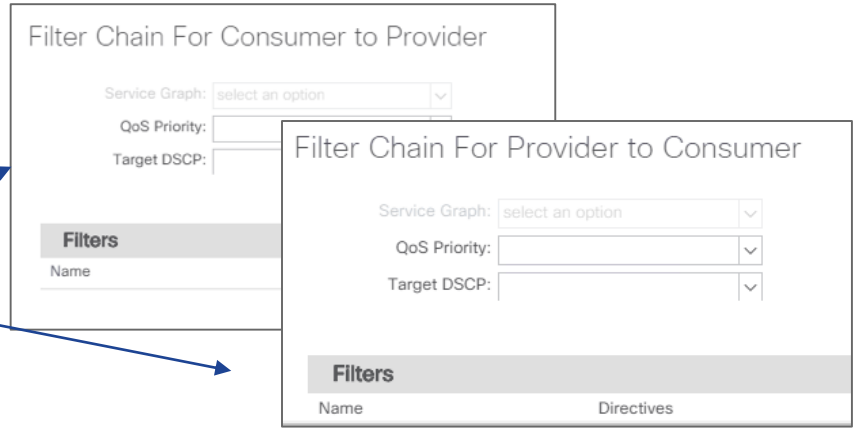
# Provider and Consumer is just to create a direction between EPGs



# You can then add subjects to the contract and define the filters for each direction

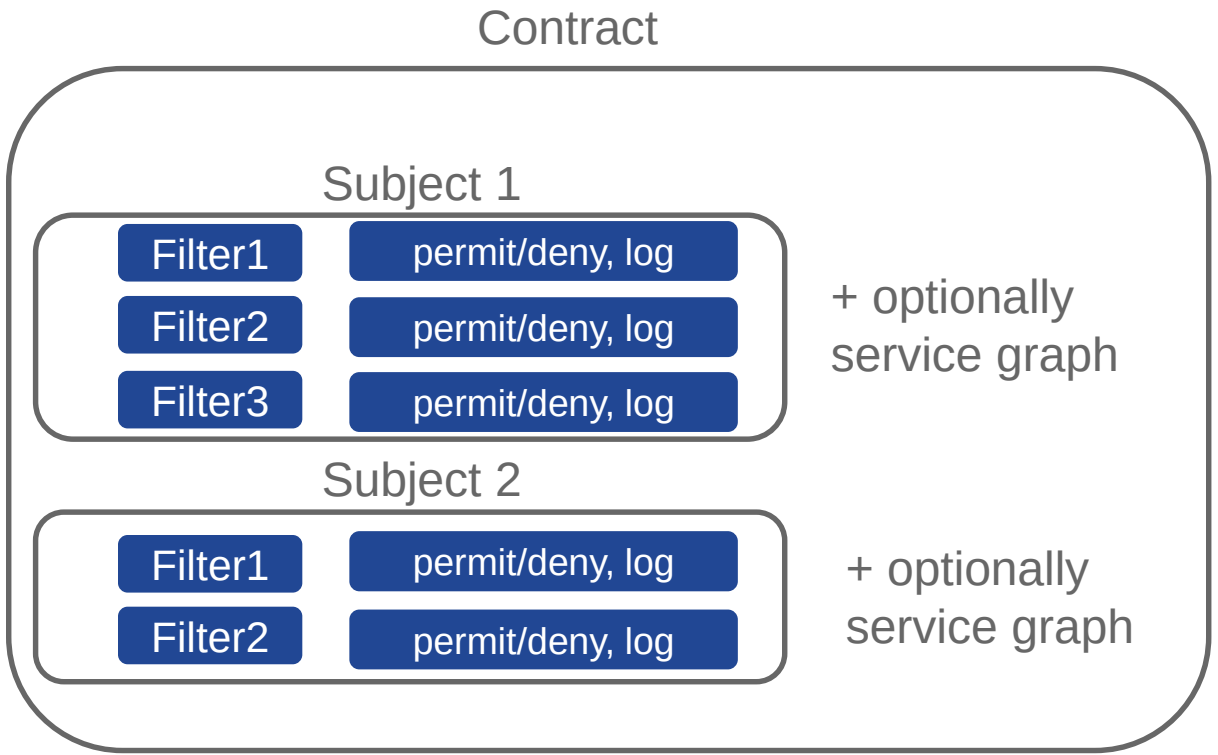



subject1: HTTP permit
subject2: Various other filters
subject3: etc..



# A Contract defines a set of filters in each "direction"

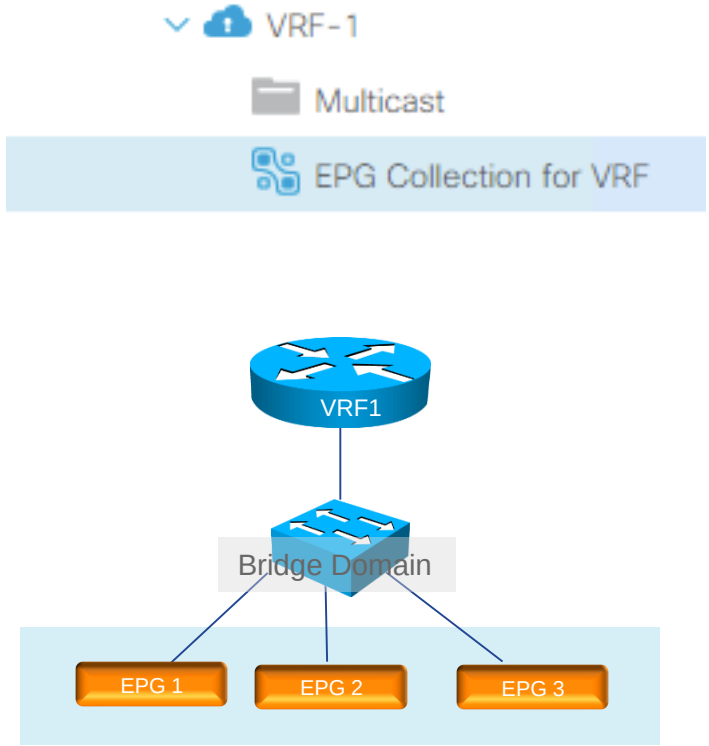
- Contract subjects contain list of filters
- Filters are rules matching protocols, src and dst ports
- Filters are attached to subjects as permit rules
- Filters are attached to subjects with a directive (log, none)



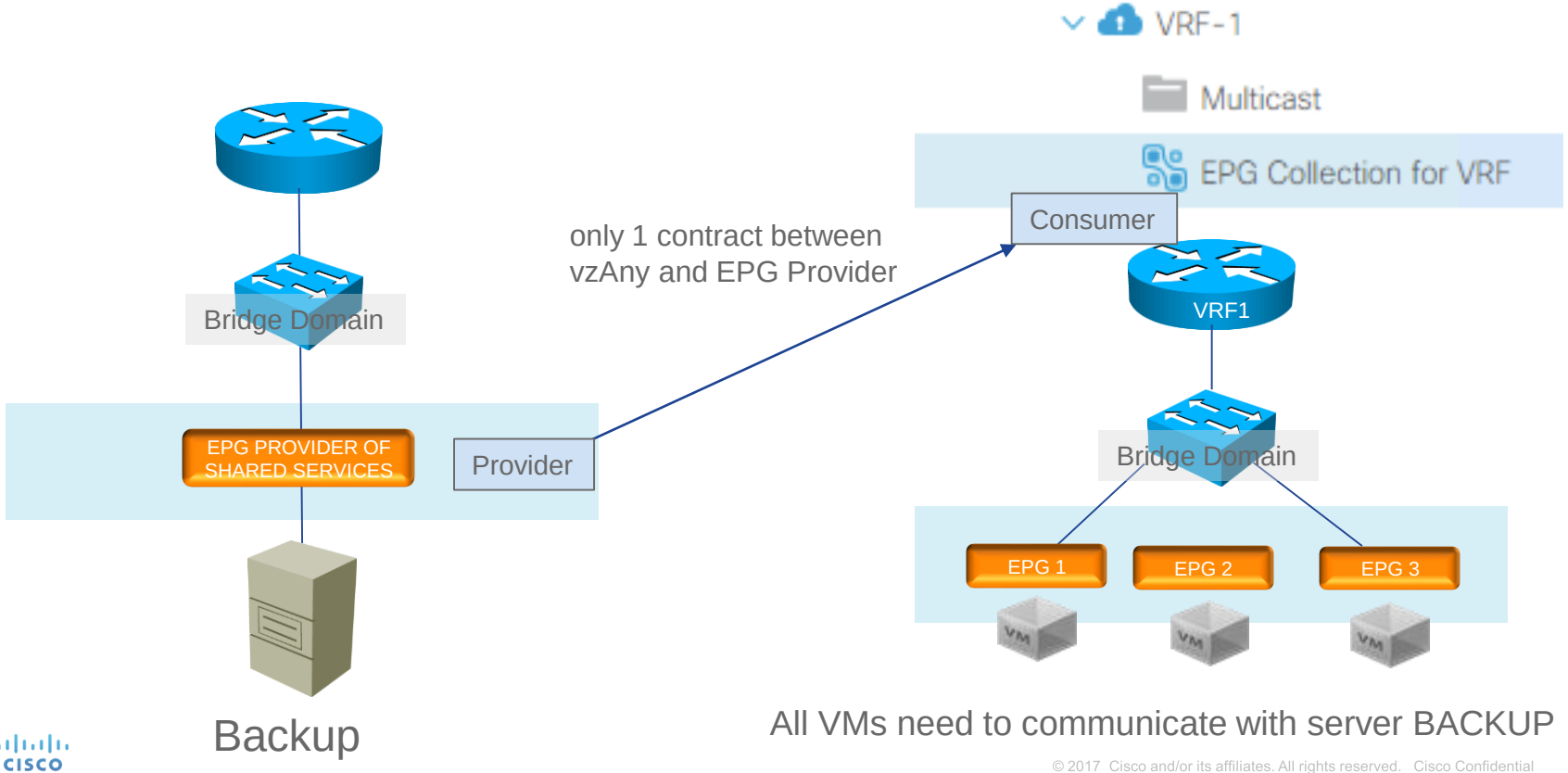
# vzAny

- One special EPG is called:
- vzAny or
- EPG collection for VRF or
- AnyEPG

- A contract defined for vzAny includes all the EPGs under the VRF and the L3Out also

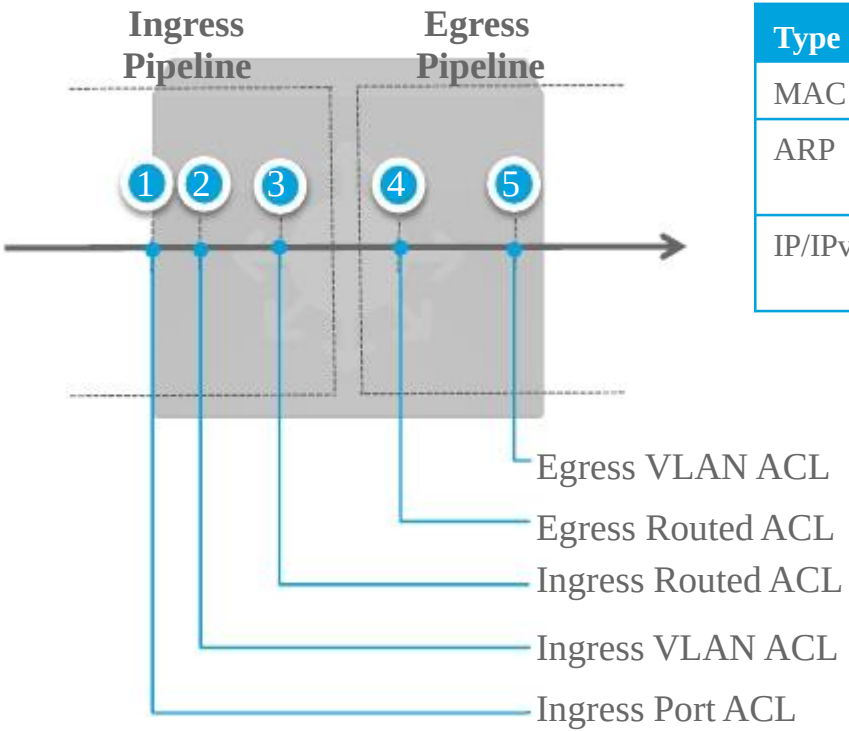


# A Common Use case for vzAny is for Shared Services across VRFs



# Contract configuration

# Classical Policy Enforcement

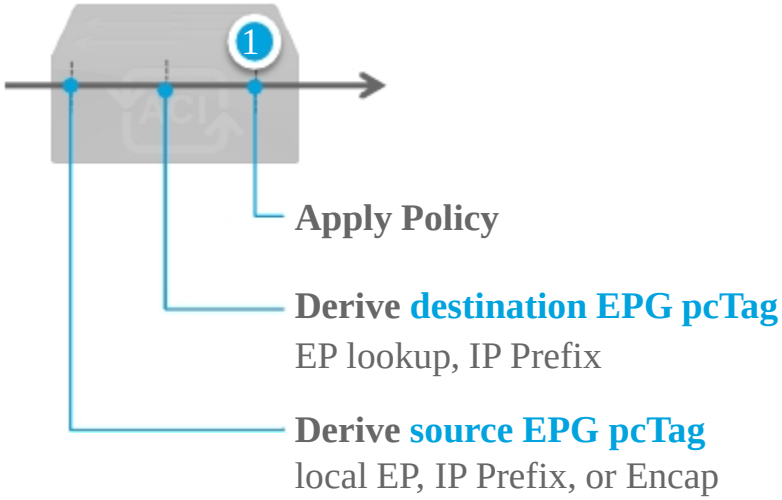



| Type | Access Control Entry (ACE) Format |
|---|---|
| MAC | action src/mask dst/mask ethertype [PD filters] |
| ARP | action opcode srcIp/mask dstIp/mask srcMac/mask dstMac/mask [PD filters] |
| IP/IPv6 | action protocol srcIp/mask srcPort/mask dstIp/mask dstPort/mask [PD filters] |

- Multiple logical locations where ACLs can be applied depending on what type of traffic and what type of filters are needed ( **very flexible** )
- ACE primarily based on src and dst values within frame (may be hard to maintain)
- ACLs often need to be configured and maintained on multiple devices in the network

# ACI Policy Enforcement

| Scope | Access Control Entry (ACE) Format |
|---|---|
| VRF | action src-EPG dst-EPG [filters] |
| VRF | permit any any (unenforced mode) |




- Policy is created based on contract between EPGs with support for L2/L3/L4 filters similar to traditional ACLs.
- Leaf derives **source EPG pcTag** based on:
  - match in **EP database**
    src MAC for L2 traffic or src IP for L3 traffic
  - **longest-prefix match** against src IP
    (IP-based EPG or L3Out external EPG)
  - ingress **port + encap**
- Leaf derives **destination EPG pcTag** based on:
  - match in **EP database**
    dst MAC for L2 traffic or dst IP for L3 traffic
  - **longest-prefix match** against dst IP
    (L3Out external EPG or shared-services)
- **Rules are programmed with scope of VRF. Policy lookup is always** ( **VRF** , **src-EPG** , **dst-EPG** , filter).
- Allow traffic between all EPGs without a contract by setting the **VRF** to **unenforced** mode

# Security in classical network

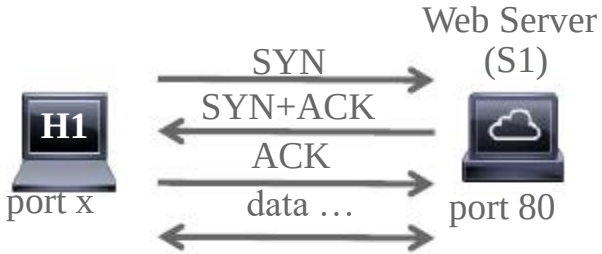
- Each ACL line (ACE – access-list entry) is a tuple containing IP info (Src & dst) and Protocol/port info (src & dst)
- ACL are applied on interface with a direction (in or ou)

# Contract in ACI

- Contract line only contains protocol. IP info are not needed as grouping of “ip/server” come from the EPG concept
- Contract are applied on EPG. Direction is “replaced” by provider/consumer relation

# ACI Policy Enforcement

**Reference** TCP Packet

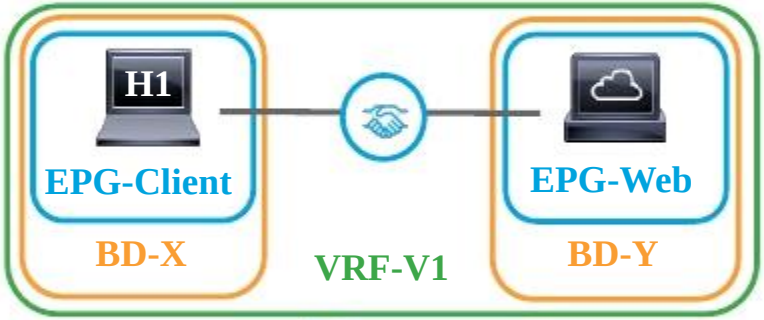




**Classical** Switch ACL

*Generally applied at one or more L3 boundaries*
*assuming H1 and S1 are in different subnets*

```
ip access-list web
  permit tcp host H1 host S1 eq 80
  permit tcp host S1 eq 80 host H1
```

**ACI** Contract




**EPG-Web** is **Providing** a service on port 80

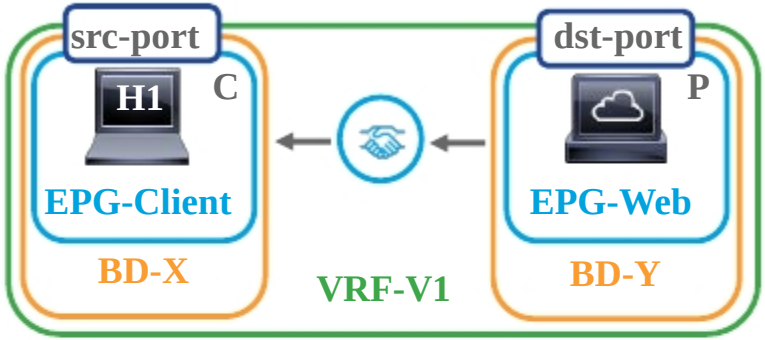
**ACI** Desired Behavior

| Scope | Access Control Entry |
|---|---|
| VRF-V1 | permit **tcp** EPG-Client EPG-Web **eq 80** |
| VRF-V1 | permit **tcp** EPG-Web **eq 80** EPG-Client |

How do we get here?

# ACI Policy Enforcement

Identify Provider ( **P** ) EPG and Consumer ( **C** ) EPG




- With a bidirectional contract, the ' **provider** ' will be the **dst-port** filters and the ' **consumer** ' will be the **src-port** filters (opposite of contract arrows)

❑ Create Filters

| Name | EthType | Proto | Src Port | Dst Port |
|---|---|---|---|---|
| flt-1 | IP | TCP | Any | 80 |
| flt-2 | IP | TCP | 80 | Any |

❑ Create a contract, subject, and filter(s). Apply to EPGs EGP-Web as provider and EPG-Client as consumer

**Option 1** – Unidirectional filters
Apply both flt-1 and flt-2 to subject

**flt-1 (C to P) and flt-2 (P to C)**

permit **tcp** Consumer Provider **eq 80**

permit **tcp** Provider **eq 80** Consumer

**Option 2** – Bidirectional filters with reverse ports

**flt-1 (C to P implied)**

permit **tcp** Consumer Provider **eq 80**

**flt-1 + apply both directions**

permit **tcp** Consumer Provider **eq 80**

permit **tcp** Provider Consumer **eq 80**

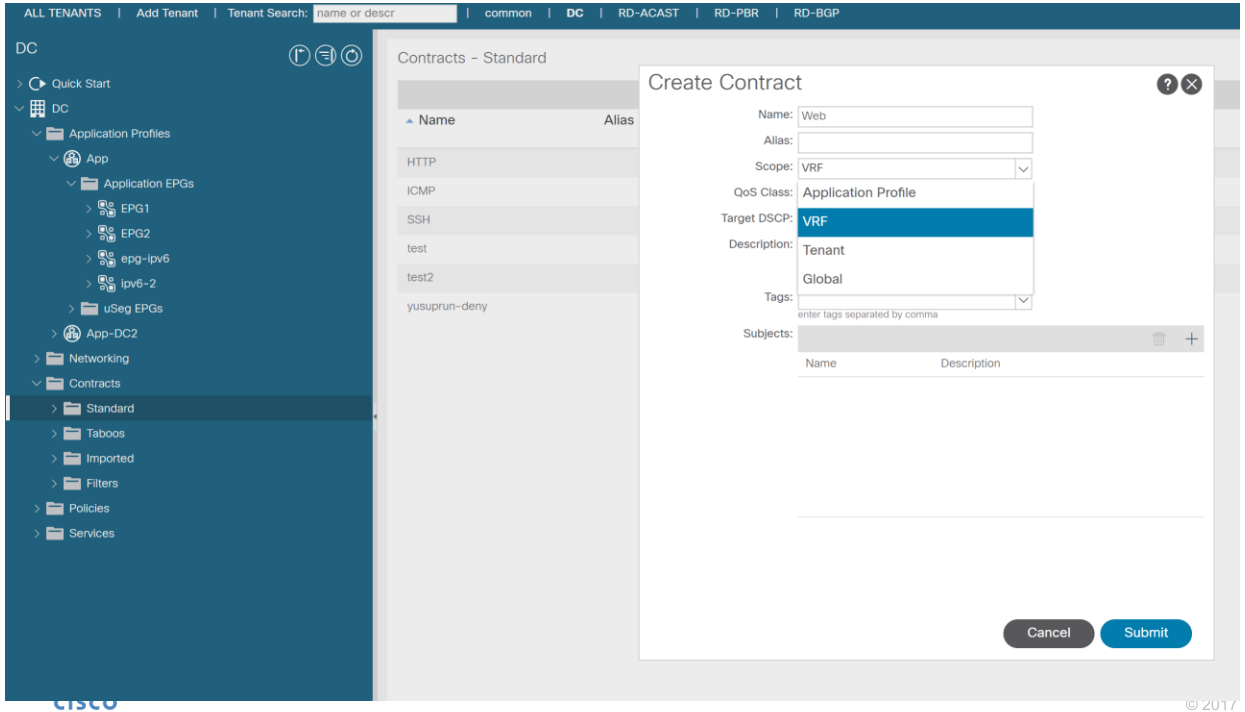
Only flt-1 needed!

**flt-1 + apply both directions + reverse ports**

permit **tcp** Consumer Provider **eq 80**

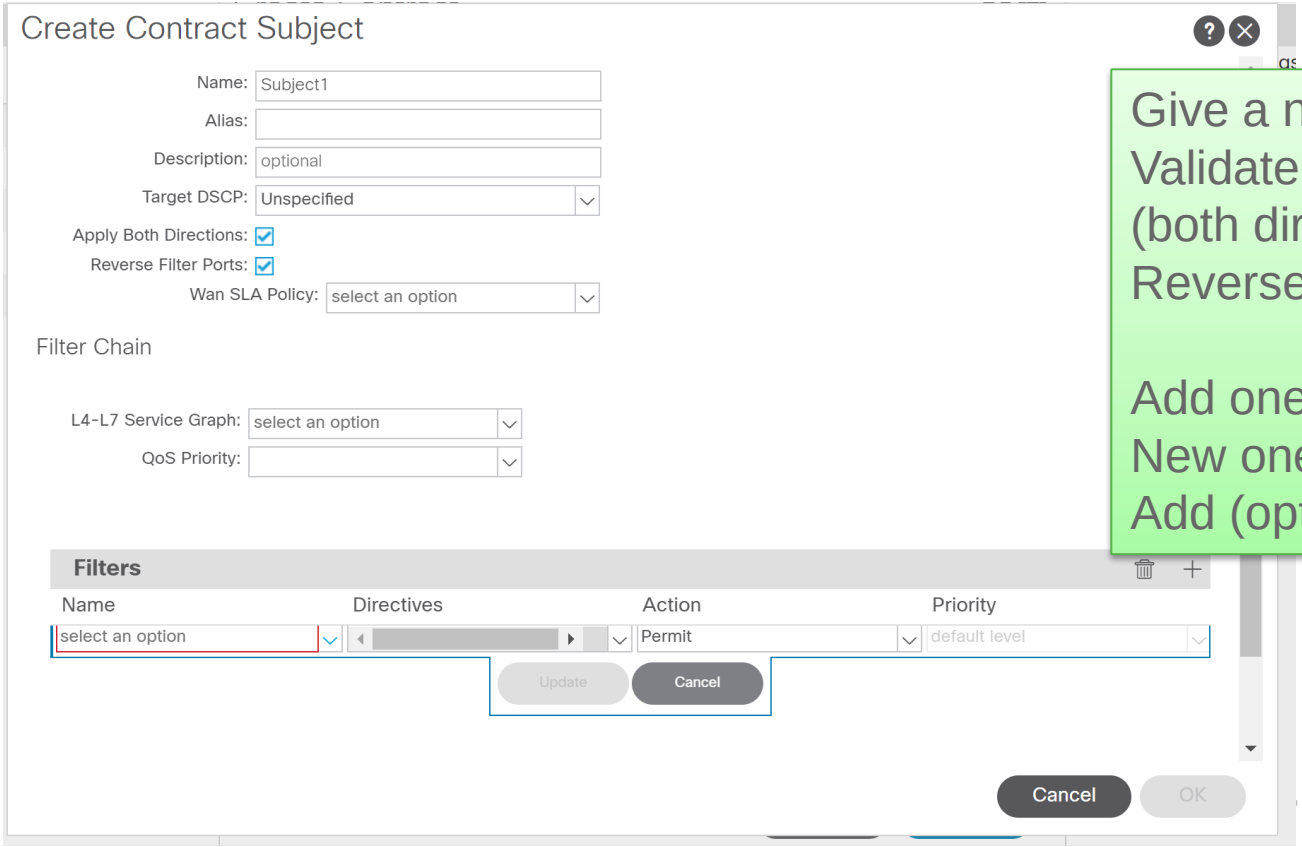
permit **tcp** Provider **eq 80** Consumer

# Config – 1/ Create Contract




Give a name
Choose the scope
Click + to add a subject

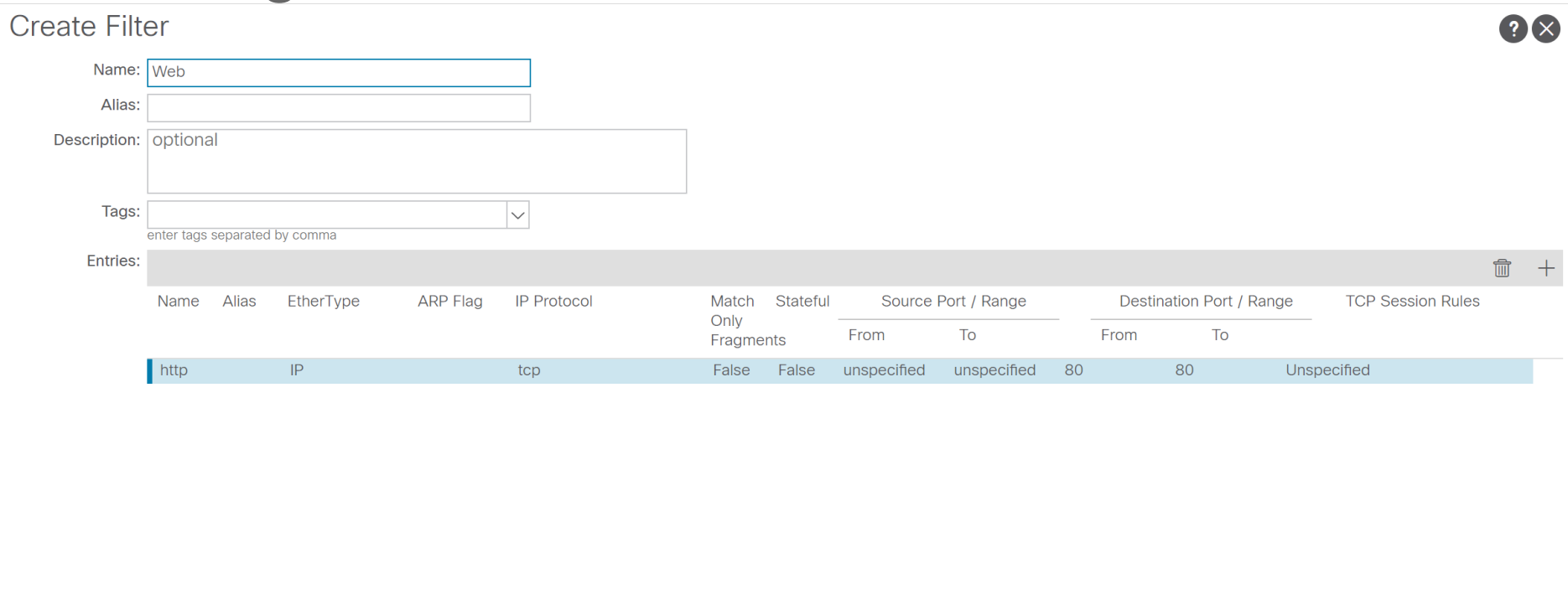
# Config 2 - Contract Subject creation




Give a name
Validate the flags (both direction and Reverse filter)

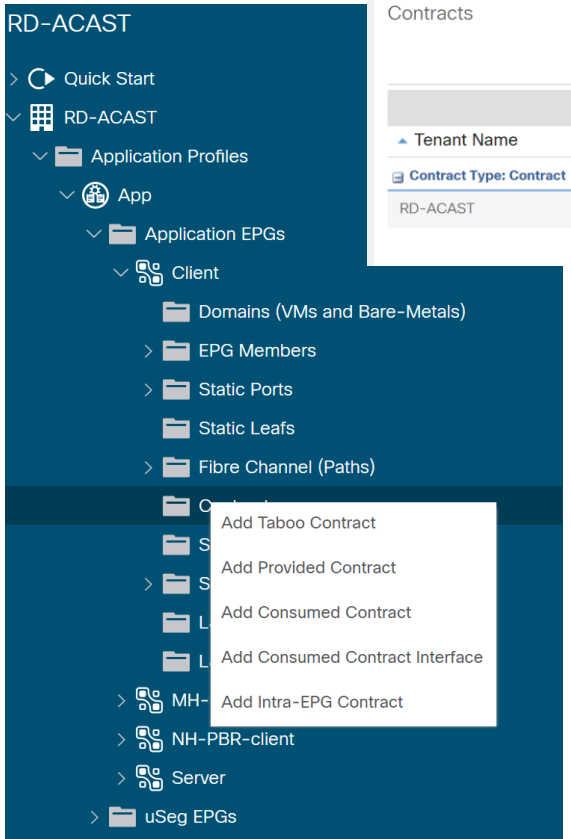
Add one or more filter (existing or New one)
Add (optional) a service graph

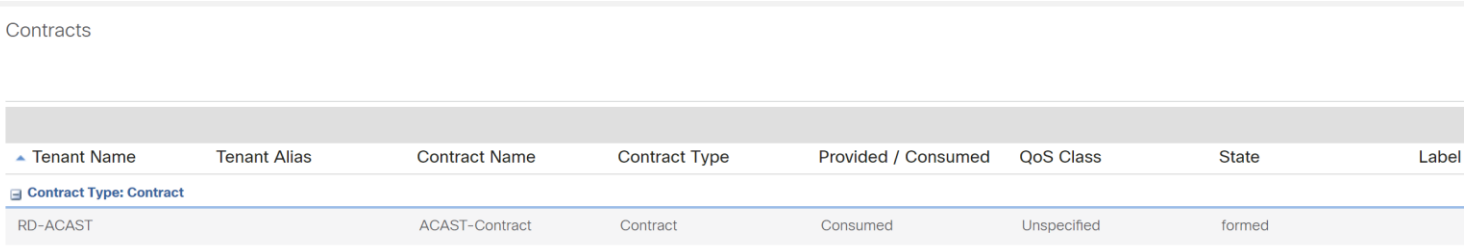
# Config 3 - Contract Filter creation




Each filter may have one or more entries
In typical use case each filter entry only specify destination ports

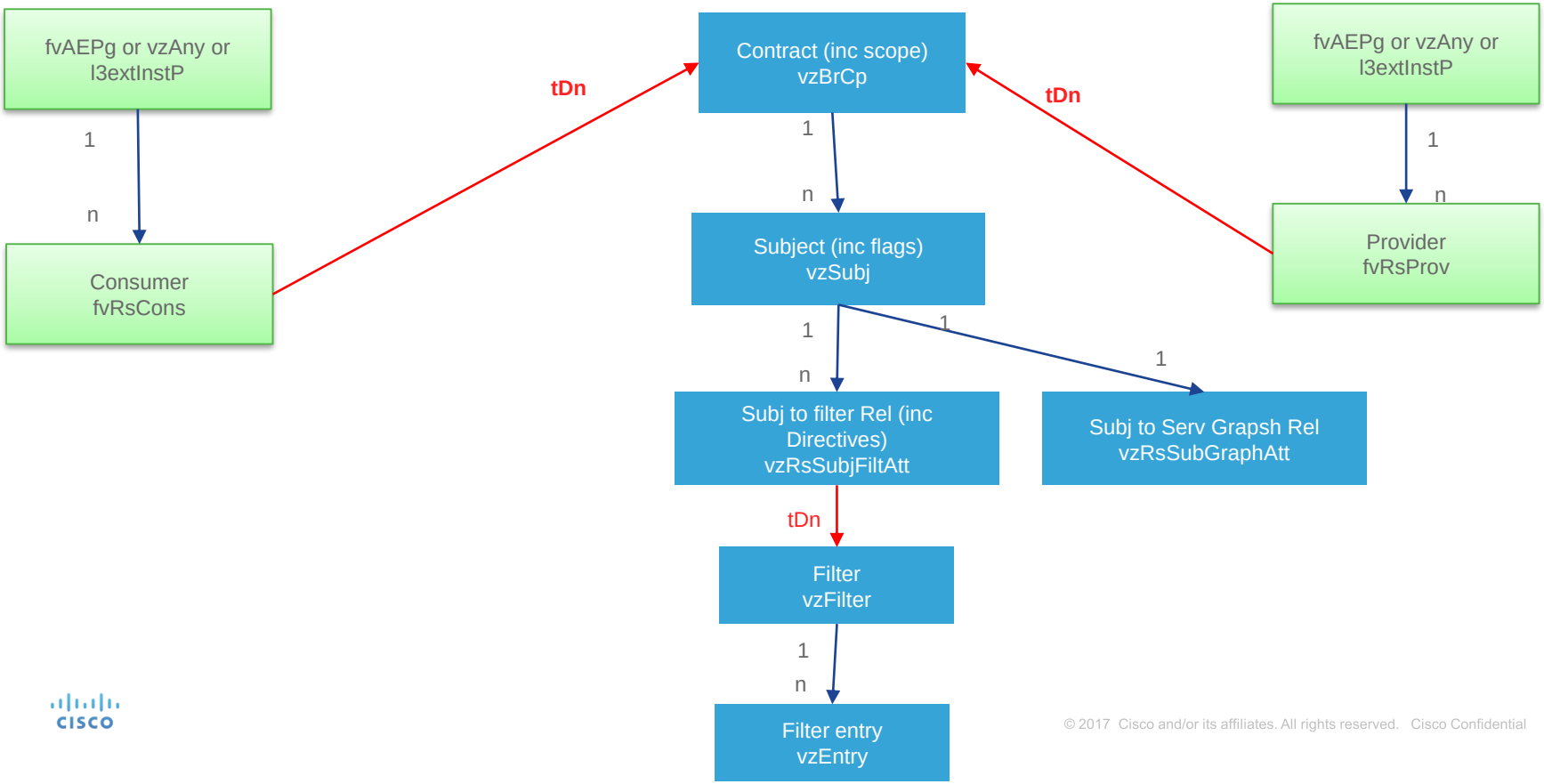
# Config 4 – Apply contract to EPG Consumer and provider



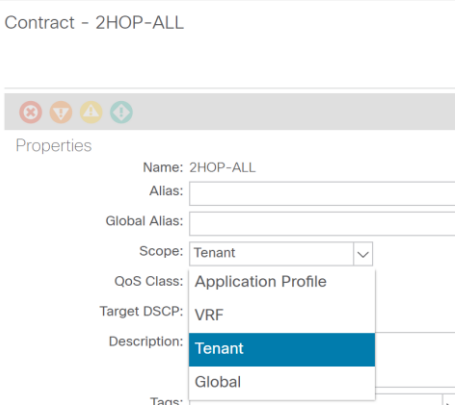




Contract can be applied on
fvAEPg – standard EPG
L3extInstP – L3 out EPG
vzAny – all EPG in VRF

# Contract Object



# More on contract scope




The contract scope will limit which providers and consumers can participate within the same contract.

- **VRF :**The contract can be applied between EPGs within the same VRF.

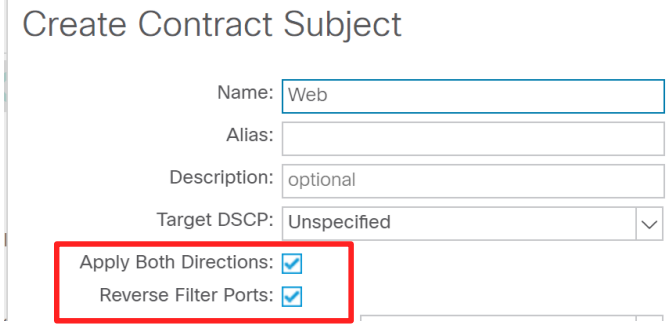
- **Application Profile :** The contract can be applied between EPGs within the same application profile

- **Tenant :** The contract can be applied between EPGs within the same tenant.

- **Global :** The contract can be applied between any EPGs within the fabric.

Note, global contracts not in common tenant need to be exported in order to be consumed by EPG in a different tenant. Consumers of global contracts will use the ‘ **Consumer Contract Interface** ’ Option

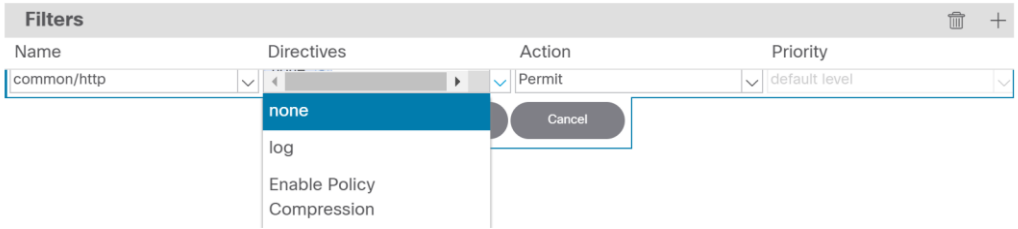
# More on subject flags

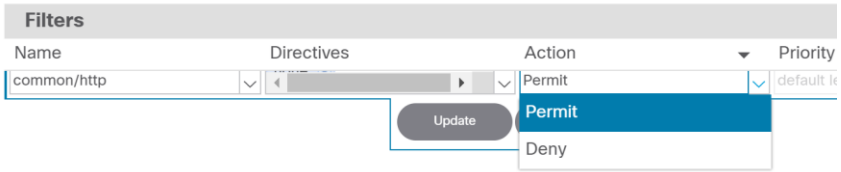



- Apply both direction : creates a duplicate rule for each TCP/UDP filter entry in the subject. That duplicated rule swap the src EPG and the Dest EPG
- Reverse Filter ports : The duplicated rule has Src and Dst port swapped
- Example:
  - Filter rule allowing Dst TCP port 80 from EPG Client to EPG Web
    - Permit Tcp Client any Server 80
  - Apply both direction : we add a rule with Dst TCP port 80 from EPG Web to EPG Client
    - Permit Tcp Client any Server 80
    - Permit tcp server any client 80
  - Reverse filter port : that dup rule is modified to be Src TCP 80
    - Permit Tcp Client any Server 80
    - Permit tcp server 80 client any

Flags should normally
Always be used together
Allow return traffic

# More on subj to filter directive and Action

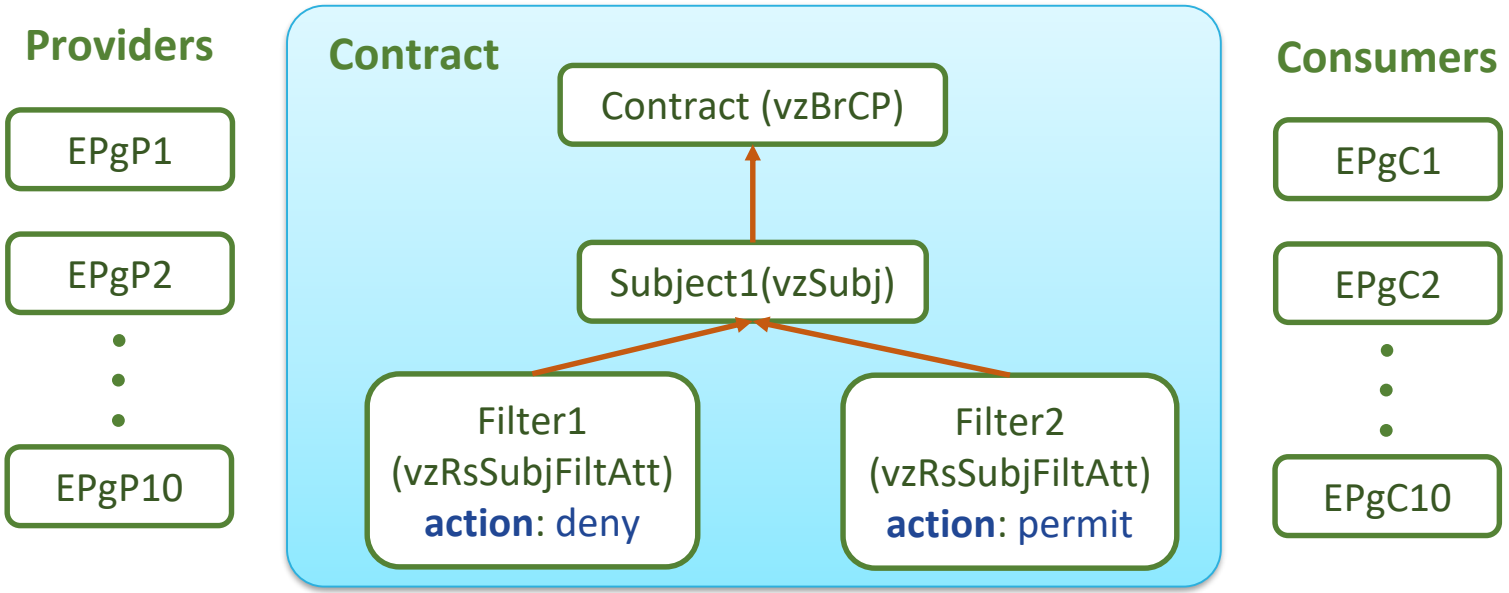



- Log allow to implement permit logging (as of 3.2)
- Enable policy compression spare TCAM resource (more on that later)

- Action is typically permit as we have default deny behavior
- However with deny action (as of 3.2) you can have in same subject interleaved deny vsRSSubjFiltAtt with permit

# Deny Filter detail



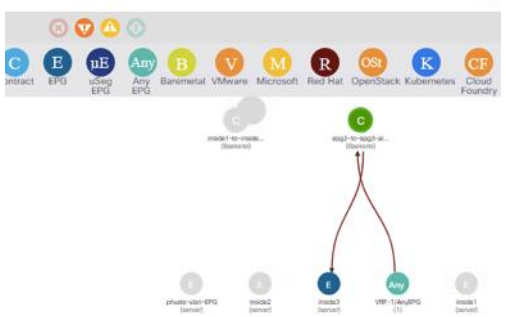

- Traffic between all the provides and the consumers for Filter1 will be dropped
  - Filter 1 deny tcp port 80
  - Filter 2 allow all Tcp traffic
  - → net result we allow all tcp traffic except port 80

# Example – Contract Deny

```
bdsol-aci32-leaf1# show zoning-rule scope 3014656 src-epg 16398
+---------+--------+--------+----------+---------+---------+---------+---------------+--------+---------------+
| Rule ID | SrcEPG | DstEPG | FilterID |   Dir   |  operSt |  Scope  |      Name     | Action |    Priority   |
+---------+--------+--------+----------+---------+---------+---------+---------------+--------+---------------+
|   4140  | 16398  | 49157  |    39    | uni-dir | enabled | 3014656 | CT1-deny-test |  deny  | fully_qual(7) |
|   4193  | 16398  | 49157  |    10    |  bi-dir | enabled | 3014656 | CT1-deny-test | permit | fully_qual(7) |
+---------+--------+--------+----------+---------+---------+---------+---------------+--------+---------------+
bdsol-aci32-leaf1# show zoning-rule scope 3014656 src-epg 49157
+---------+--------+--------+----------+----------------+---------+---------+---------------+--------+---------------+
| Rule ID | SrcEPG | DstEPG | FilterID |      Dir       |  operSt |  Scope  |      Name     | Action |    Priority   |
+---------+--------+--------+----------+----------------+---------+---------+---------------+--------+---------------+
|   4205  | 49157  | 16398  |    10    | uni-dir-ignore | enabled | 3014656 | CT1-deny-test | permit | fully_qual(7) |
|   4191  | 49157  | 16398  |    40    |    uni-dir     | enabled | 3014656 | CT1-deny-test |  deny  | fully_qual(7) |
+---------+--------+--------+----------+----------------+---------+---------+---------------+--------+---------------+
bdsol-aci32-leaf1# show zoning-filter
+----------+------+--------+-------------+------+-------------+--------+-----------+-------------+-------------+-------------+-------+----------
| FilterId | Name | EtherT |    ArpOpc   | Prot |ApplyToFra |Stateful|  SFromPort  |   SToPort   |  DFromPort  |   DToPort   |  Prio |   Icmpv4T
+----------+------+--------+-------------+------+-------------+--------+-----------+-------------+-------------+-------------+-------+----------
|    10    | 10_0 |   ip   | unspecified | tcp  |      no     |   no   | unspecified | unspecified | unspecified | unspecified | proto |
|    39    | 39_1 |   ip   | unspecified | tcp  |      no     |   no   | unspecified | unspecified |    https    |    https    | dport | unspecified |
|    39    | 39_0 |   ip   | unspecified | tcp  |      no     |   no   | unspecified | unspecified |     http    |     http    | dport | unspecified |
|    40    | 40_1 |   ip   | unspecified | tcp  |      no     |   no   |     http    |     http    | unspecified | unspecified | sport | unspc |
|    40    | 40_0 |   ip   | unspecified | tcp  |      no     |   no   |    https    |    https    | unspecified | unspecified | sport | unspe |
+----------+------+--------+-------------+------+-------------+--------+-----------+-------------+-------------+-------------+-------+-----------
```

# Verifying Contract

# The contract filters are programmed in the Policy Cam on the Leaf

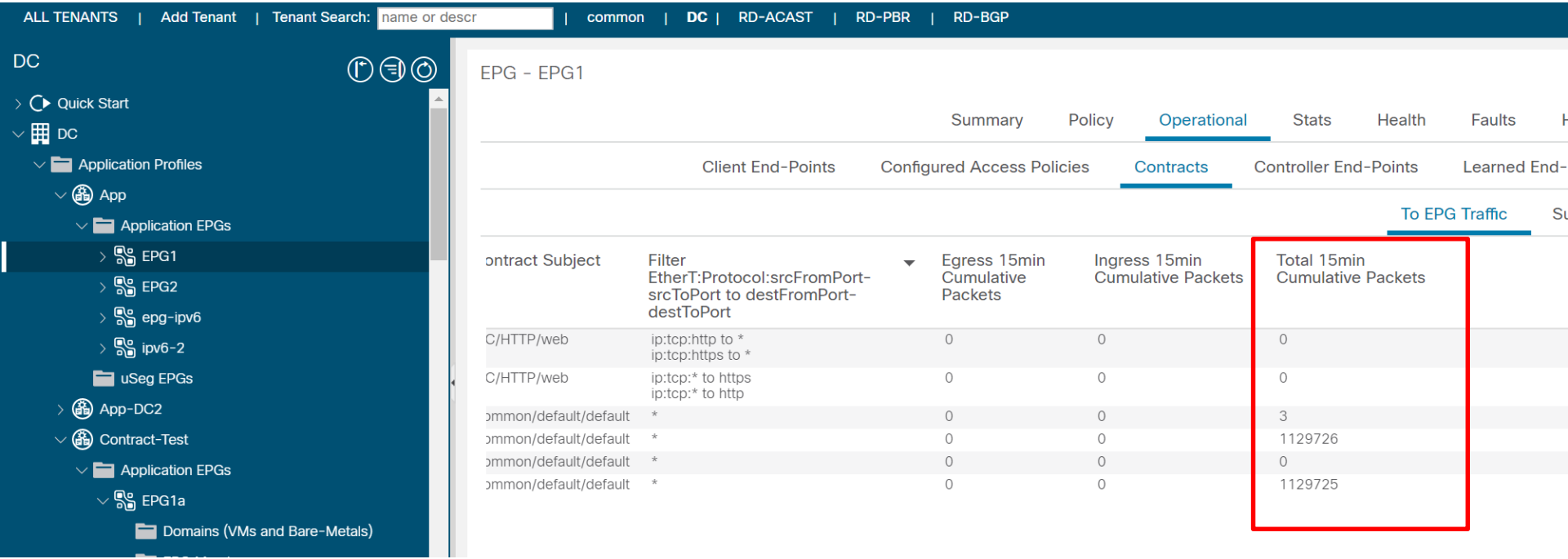


Name: VRF-1
Alias:
Description: optional
Tags:
enter tags separated by comma
Global Alias:
Segment: 2555904

| Provider EPG | Consumer EPG | Contract Information |
|---|---|---|
| Baekerei/client-to-server/inside2<br>Provider Labels: None<br>49166 | Baekerei/client-to-server/inside1<br>Consumer Labels: None<br>32787 | Baekerei/inside1-to-inside2 Scope: context<br>Provider Labels<br>Subject: abc<br>Labels: None<br>Consumer Labels<br>Subject: abc<br>Labels: None |

```
leaf1# show zoning-rule scope VNID-OF-THE-VRF
```

# GUI – Counters contract EPG view



# How to Check in the TCAM

- log into the leaf of interest
- “show zoning-rule”
- You find “rule ID, and the SrcEPG and DestEPG
- The srcEPG and DestEPG are numbers called class-id (pcTag)
- You can find the class-id from the GUI by highlighting the EPG or all class id in GUI – tenant – Operational – Ressource ID

# pcTag and VRF VNID in GUI

Tenant - DC

Summary | Dashboard | Policy | **Operational** | Stats | Health | Faults

Flow

Bridge Domains | VRFs | **EPGs** | External Networks (Routed)

100

| Application Profile Name | AP Alias | EPG Name | Class ID | Scope |
|---|---|---|---|---|
| App-DC2 | | EPG1-DC2 | 32770 | 2162691 |
| App | | EPG1 | 32775 | 3014656 |
| App | | EPG2 | 16387 | 3014656 |
| App | | epg-ipv6 | 49153 | 3014656 |
| App | | ipv6-2 | 49155 | 3014656 |
| Contract-Test | | EPG1a | 16398 | 3014656 |
| Contract-Test | | EPG1b | 49157 | 3014656 |

# Finding pcTag and scope in Object model

- fvAEPg is the class in logical model representing regular epg
- l3extInstP is the class in logical model representing L3 out EPG

```
admin@pod2-apic1:~> moquery -c fvAEPg | egrep -9 "^dn.*DC.*EPG" | egrep "dn|fv.AEP|scope|pcTag"
# fv.AEPg
dn           : uni/tn-DC/ap-App/epg-EPG1
pcTag        : 32770
scope        : 2097153
# fv.AEPg
dn           : uni/tn-DC/ap-App/epg-EPG2
pcTag        : 49153
scope        : 2097153

admin@pod2-apic1:~> moquery -c l3extInstP | egrep -10 "^dn.*DC" | egrep "dn|l3ext|scope|pcTag"
# l3ext.InstP
dn           : uni/tn-DC/out-L3out/instP-L3outEPG
pcTag        : 32772
scope        : 2097153
```

For traffic between VM A and VM B we are interested
In rules between sclass 49153 and 32770 in scope 2097153

# ACL Key in TCAM

Contract are implemented in what we call zoning-rule in TCAM (SGTCAM – sec group tcam)

They key in TCAM is the following :

**Scope – Src sclass – Dst sclass – filter entry**

(vrf vnid – src pcTag – Dst pcTac – Filter entry (proto))

Other field in zoning-rule

Rule priority

Action (permit, deny, log, redirect,..)

# Zoning rule and rules statistics – leaf 4

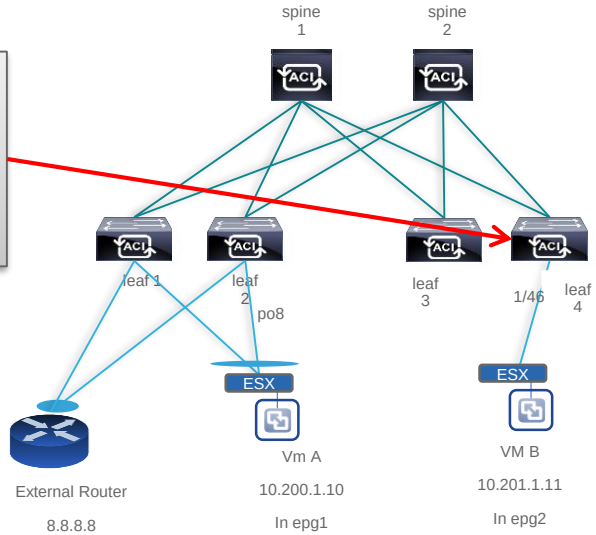
1st on leaf 4 we check any rule between our two epgs

```
pod2-leaf4# show zoning-rule | egrep "32770.*49153.*2097153|Rule|49153.*32770.*2097153"
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action          Priority
4150            32770           49153           5               enabled         2097153         permit          fully_qual(7)
4151            49153           32770           5               enabled         2097153         permit          fully_qual(7)
```

Now we check the statistics for those rules. We see we receive packet In epg 2 (49153) towards epg 1 (32770)

```
pod2-leaf4# show system internal policy-mgr stats  | egrep "4150|4151"
Rule (4150) DN (sys/actrl/scope-2097153/rule-2097153-s-32770-d-49153-f-5) Ingress: 0, Egress: 0
Rule (4151) DN (sys/actrl/scope-2097153/rule-2097153-s-49153-d-32770-f-5) Ingress: 62, Egress: 0
pod2-leaf4# show system internal policy-mgr stats  | egrep "4150|4151"
Rule (4150) DN (sys/actrl/scope-2097153/rule-2097153-s-32770-d-49153-f-5) Ingress: 0, Egress: 0
Rule (4151) DN (sys/actrl/scope-2097153/rule-2097153-s-49153-d-32770-f-5) Ingress: 72, Egress: 0
```

Where is the return traffic ?



# Zoning rule and rules statistics – leaf 1

On leaf 1 we check any rule between our two epgs

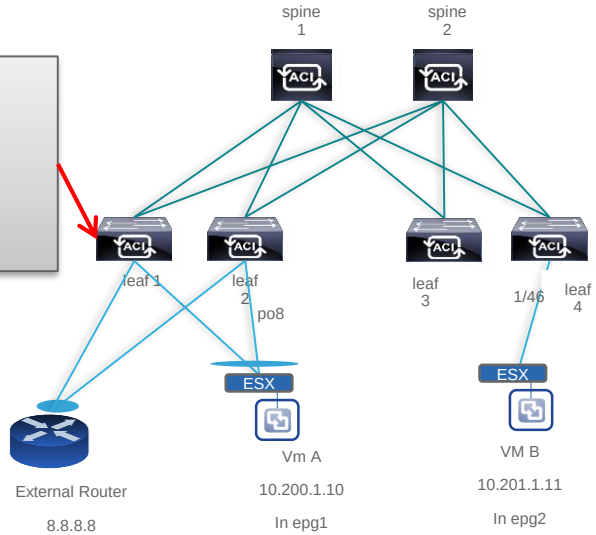
```
pod2-leaf1# show zoning-rule | egrep "32770.*49153.*2097153|Rule|49153.*32770.*2097153"
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action          Priority
4142            32770           49153           5               enabled         2097153         permit          fully_qual(7)
4145            49153           32770           5               enabled         2097153         permit          fully_qual(7)
```

Now we check the statistics for those rules. We see we receive packet In epg 1 (32770) towards epg 2 (49153)

```
pod2-leaf1# show system internal policy-mgr stats | egrep "4142|4145"
Rule (4142) DN (sys/actrl/scope-2097153/rule-2097153-s-32770-d-49153-f-5) Ingress: 533, Egress: 0
Rule (4145) DN (sys/actrl/scope-2097153/rule-2097153-s-49153-d-32770-f-5) Ingress: 0, Egress: 0
pod2-leaf1# show system internal policy-mgr stats | egrep "4142|4145"
Rule (4142) DN (sys/actrl/scope-2097153/rule-2097153-s-32770-d-49153-f-5) Ingress: 543, Egress: 0
Rule (4145) DN (sys/actrl/scope-2097153/rule-2097153-s-49153-d-32770-f-5) Ingress: 0, Egress: 0
```

In this scenario we now see the policy is enforced on ingress leaf in both direction

How can we be sure this is our ICMP filter ?



# Checking Filter on the switch

Also we can check «show zoning-filter »

```
pod2-leaf1# show zoning-filter

FilterId  Name       EtherT      ArpOpc      Prot     MatchOnlyFrag Stateful SFromPort   SToPort     DFromPort   DToPort     Prio      Icmpv4T    Icmpv6T  TcpRules

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

implicit  implicit   unspecified unspecified unspecified no         no       unspecified unspecified unspecified unspecified implicit   unspecified unspecified

implarp   implarp    arp         unspecified unspecified no         no       unspecified unspecified unspecified unspecified dport      unspecified unspecified

default   any        unspecified unspecified unspecified no         no       unspecified unspecified unspecified unspecified def        unspecified unspecified

5         5_0        ip          unspecified icmp        no         no       unspecified unspecified unspecified unspecified sport      unspecified unspecified
```

# Zoning in 4.x software

- Output of zoning rule was revamped in 4.0
- Policy_mgr stats still similar

```
bdsol-aci32-leaf1# show zoning-rule scope 3014656 src-epg 16398
+---------+--------+--------+----------+---------+---------+---------+---------------+--------+---------------+
| Rule ID | SrcEPG | DstEPG | FilterID |   Dir   |  operSt |  Scope  |      Name     | Action |    Priority   |
+---------+--------+--------+----------+---------+---------+---------+---------------+--------+---------------+
|   4140  | 16398  | 49157  |    39    | uni-dir | enabled | 3014656 | CT1-deny-test |  deny  | fully_qual(7) |
|   4193  | 16398  | 49157  |    10    |  bi-dir | enabled | 3014656 | CT1-deny-test | permit | fully_qual(7) |
+---------+--------+--------+----------+---------+---------+---------+---------------+--------+---------------+
bdsol-aci32-leaf1# show system internal policy-mgr stats 4140
Requested Rule Statistics
Rule (4140) DN (sys/actrl/scope-3014656/rule-3014656-s-16398-d-49157-f-39) Ingress: 0, Egress: 0, Pkts: 0
RevPkts: 0
```

# Python script to ease your life

```
bdsol-aci32-leaf1# contract_parser.py --help

usage: contract_parser.py [-h] [--offline OFFLINE] [--offlineHelp] [--noNames]
                          [--noContract] [--noGraph] [--cache CACHE]
                          [--debug {debug,info,warning,error,critical}] [--nz]
                          [--incremented] [--node NODES [NODES ...]]
                          [--contract CONTRACT [CONTRACT ...]]
                          [--vrf VRF [VRF ...]] [--epg EPG [EPG ...]]
                          [--sepg SEPG [SEPG ...]] [--depg DEPG [DEPG ...]]
                          [--protocol PROT [PROT ...]]
                          [--port PORT [PORT ...]] [--sport SPORT [SPORT ...]]
                          [--dport DPORT [DPORT ...]] [--checkMask]

  This script checks zoning rules, filters, and statistics and correlates with
  EPG names. The results are printed in NXOS/IOS-like ACL syntax.
```

# Python script to ease your life

```
bdsol-aci32-leaf1# contract_parser.py --vrf DC:DC | egrep EPG1a

[7:4193] [vrf:DC:DC] permit ip tcp tn-DC/ap-Contract-Test/epg-EPG1a(16398) tn-DC/ap-Contract-Test/epg-EPG1b(49157)  [contract:uni/tn-
DC/brc-CT1-deny-test] [hit=4332]

[7:4140] [vrf:DC:DC] deny ip tcp tn-DC/ap-Contract-Test/epg-EPG1a(16398) tn-DC/ap-Contract-Test/epg-EPG1b(49157) eq 80
[contract:uni/tn-DC/brc-CT1-deny-test] [hit=460]

                          ip tcp tn-DC/ap-Contract-Test/epg-EPG1a(16398) tn-DC/ap-Contract-Test/epg-EPG1b(49157) eq 443

[7:4205] [vrf:DC:DC] permit ip tcp tn-DC/ap-Contract-Test/epg-EPG1b(49157) tn-DC/ap-Contract-Test/epg-EPG1a(16398)  [contract:uni/tn-
DC/brc-CT1-deny-test] [hit=0]

[7:4191] [vrf:DC:DC] deny ip tcp tn-DC/ap-Contract-Test/epg-EPG1b(49157) eq 443 tn-DC/ap-Contract-Test/epg-EPG1a(16398)
[contract:uni/tn-DC/brc-CT1-deny-test] [hit=0]

                          ip tcp tn-DC/ap-Contract-Test/epg-EPG1b(49157) eq 80 tn-DC/ap-Contract-Test/epg-EPG1a(16398)
```

# Deny / Permit log

## Usual Cli logging denied leaked to cpu (small buffer)

```
bdsol-aci32-leaf4# show logging ip access-list internal packet-log deny
[ Wed Apr 12 09:16:26 2017 959265 usecs]: CName: RD:SB(VXLAN: 2097153), VlanType: FD_VLAN, Vlan-Id: 11, SMac: 0x005056a41279,
DMac:0x0022bdf819f1, SIP: 10.20.1.3, DIP: 10.20.21.1, SPort: 0, DPort: 0, Src Intf: Ethernet1/9, Proto: 1, PktLen: 98

[ Wed Apr 12 09:16:25 2017 959208 usecs]: CName: RD:SB(VXLAN: 2097153), VlanType: FD_VLAN, Vlan-Id: 11, SMac: 0x005056a41279,
DMac:0x0022bdf819f1, SIP: 10.20.1.3, DIP: 10.20.21.1, SPort: 0, DPort: 0, Src Intf: Ethernet1/9, Proto: 1, PktLen: 98

[ Wed Apr 12 09:16:24 2017 959875 usecs]: CName: RD:SB(VXLAN: 2097153), VlanType: FD_VLAN, Vlan-Id: 11, SMac: 0x005056a41279,
DMac:0x0022bdf819f1, SIP: 10.20.1.3, DIP: 10.20.21.1, SPort: 0, DPort: 0, Src Intf: Ethernet1/9, Proto: 1, PktLen: 98
```

## New Cli . Keeping track of logged flows

```
bdsol-aci32-leaf4# show logging ip access-list cache deny
Source MAC     Destination MAC    Source IP      Destination IP    S-Port  D-Port   Interface      Protocol     VRF    VRF-Encap StartTimeStamp          EndTimeStamp
PktLen  Hits
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
005056a41279   0022bdf819f1       10.20.1.3      10.20.21.1        0       0        Ethernet1/9    (001)ICMP    RD:SB    2097153   Apr 12 09:13:55 2017    Apr 12 09:16:26
2017    98      112
..
```

# How to read zoning-rule

# Reading Zoning-Rule

Rule ID : no meaning, just an index
SrcEPG : sclass/pcTag that will match for the src of the traffic
DstEPG : sclass/pcTag that will match for the dst of the traffic
FilterId : pointer to Filter table (aka protocol to check)
- implicit – not result of config, this is a default rule
- implicate arp – arp is always permitted by default in the vrf
- default – means the filter applied is allow all (or commond/default)

Scope – vrf vnid (or seg)
Action – Permit, deny, log or a combination
Priority – the number is the priority, the lower the better

```
bdsol-aci32-leaf3#  show zoning-rule | egrep "Rule|==|2228225"
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action                  Priority
=======         ======          ======          ========        ======          =====           ======                  ========
4159            0               0               implicit        enabled         2228225         deny,log                any_any_any(21)
4197            0               0               implarp         enabled         2228225         permit                  any_any_filter(17)
4202            0               15              implicit        enabled         2228225         deny,log                any_vrf_any_deny(22)
4203            0               49153           implicit        enabled         2228225         permit                  any_dest_any(16)
4206            0               15              default         enabled         2228225         permit                  any_dest_any(16)
4207            32770           0               default         enabled         2228225         permit                  src_any_any(15)
4212            0               16387           implicit        enabled         2228225         permit                  any_dest_any(16)
4217            32772           16386           43              enabled         2228225         permit                  fully_qual(7)
4218            32772           16386           37              enabled         2228225         permit                  fully_qual(7)
4219            16386           32772           38              enabled         2228225         permit                  fully_qual(7)
4216            16386           32772           44              enabled         2228225         permit                  fully_qual(7)
```

# Policy rule lookup algorithm

1. Packet enters a leaf , we need to derive pcTAg
    - If ingress on regular EPG it will always be the epg pctag
    - If ingress on l3 out, it will be
        - vrf pcTag if hitting 0.0.0.0/0 in aclqos prefix
        - L3 out epg pcTag if hitting a more specific subnet in aclqos prefix table (policy-mgr prefix in vsh as of 3.2)
2. We make a forwarding decision (route lookup or epm destination)
3. We try to derive dest pcTag
    - If dest is l3 out we will again use aclqos prefix to derive dst pcTag
        - If hitting 0.0.0.0/0 pctag will be 15
        - If hitting more specific subnet it will be the dest l3 out epg pcTag
    - If dest is regular EPG, we will know the pctag from EPM lookup
        - If EPM is populated we derive directly dst pctag of dest EPG
        - If EPM is not populated, we send to fabric (spine or flood) and we insert src pctag in vxlan header to let Dest Leaf to enforce zoning-rule
4. At this stage we should have both src and dst pcTag and we will make zoning-rule lookup by Rule priority either in src leaf or on destination leaf:
    - For every line we look in zoning-rule, we consider it to be a hit if either :
        - src and dst pcTag resulting from packet lookup do match src and dst epg of the line AND the protocol of the frame is matching the filter of that line. Note src and dst epg in zoning-rule with value 0 will be consider as potential Hit always
    - We checked line by priority and scope:
        - We first check line in the scope with lowest priority
        - if no match in lowest priority line , we check every line with the next lowest prioriy (just higher)

Note : Within rule with same Rule priority, we have extra rules (see next section)

- Deny wins over permit/redirect
- L4 rule for entry priority are tiebreaker if needed

# Finding pcTag for a packet (src and dst)

- From Data packet : use ELAM and check sclass and dclass
- Below packet is from pcTag 49155 to pcTag 15
- Next steps :
  - Is it expected pcTag ?
  - What line do we hit in zoning-rule ?

```
module-1(DBG-elam-insel6)# report | egrep class
      sug_lurw_vec.info.ifabric_leaf.dclass: 0xF      - here 15
      sug_lurw_vec.info.ifabric_leaf.sclass: 0xC003   - here 49155
```

# Verifying expected pcTag from cli – L3 out

- For l3 out you need to use aclqos prefix
  - Below src of packet in previous elam was 10.20.0.1/32 on a l3 out hence using 49155 as src sclass
  - Dst ip was 10.30.0.1 on another l3 out hence using 15
  - CLI is replaced after 3.2 by : vsh –c ‘show system internal policy-mgr prefix’

```
module-1(DBG-elam-insel6)# show system internal  aclqos prefix  | egrep "Vrf|==|2228225"
Vrf-Vni VRF-Id Table-Id          Addr                        Class  Shared Remote Complete
======= ====== ============ ============================== ====== ====== ====== ========
2228225 15     0x8000000f     0::/0                         15     0      0      No
2228225 15     0xf            0.0.0.0/0                     15     0      0      No
2228225 15     0xf            10.20.0.1/24                  49155  0      0      No
```

# Verifying expected pcTag from cli – regular EPG

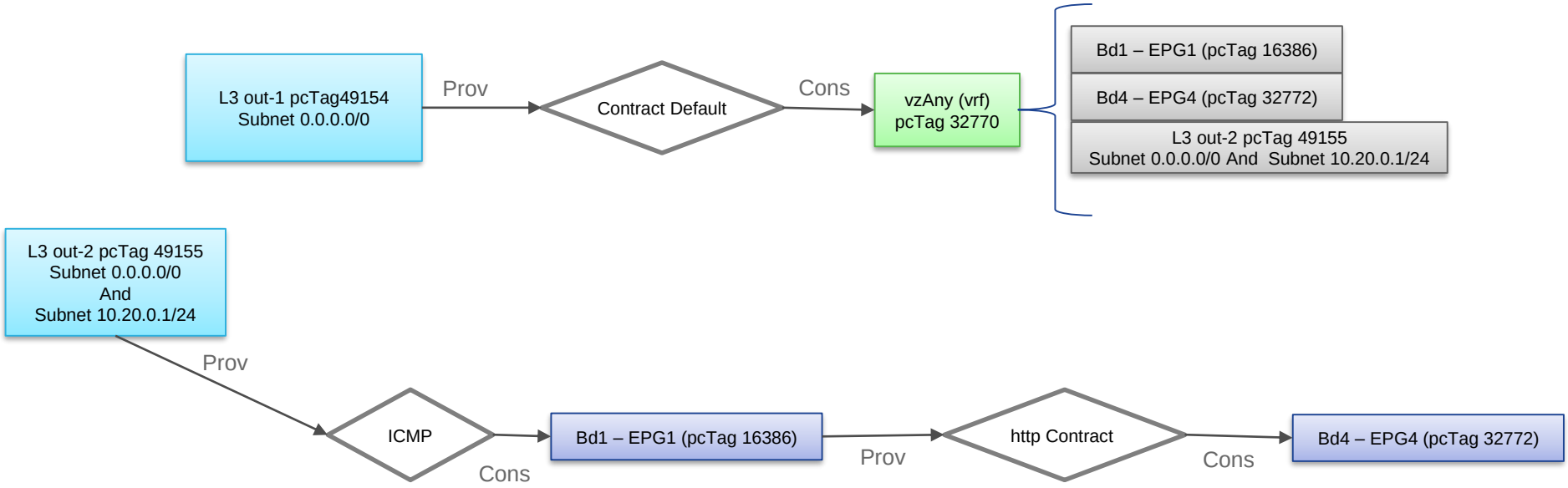
```
bdsol-aci32-leaf3# show system internal epm  endpoint  ip
10.143.1.1

MAC : 0050.56a4.0017 ::: Num IPs : 1
IP# 0 : 10.143.1.1 ::: IP# 0 flags :
Vlan id : 55 ::: Vlan vnid : 8412 ::: VRF name : RD-TR:RD
BD vnid : 15826915 ::: VRF vnid : 2228225
Phy If : 0x1a008000 ::: Tunnel If : 0
Interface : Ethernet1/9
Flags : 0x80004c04 ::: sclass : 16386 ::: Ref count : 5
EP Create Timestamp : 06/06/2018 05:41:06.782885
EP Update Timestamp : 06/06/2018 05:48:49.495349
EP Flags : local|IP|MAC|sclass|timer|
::::


bdsol-aci32-leaf3# show system internal epm  endpoint  ip
10.143.2.1

MAC : 0050.56a4.5e29 ::: Num IPs : 1
IP# 0 : 10.143.2.1 ::: IP# 0 flags :
Vlan id : 61 ::: Vlan vnid : 8413 ::: VRF name : RD-TR:RD
BD vnid : 16580494 ::: VRF vnid : 2228225
Phy If : 0x1a008000 ::: Tunnel If : 0x1801002c
Interface : Tunnel44
Flags : 0x80000c80 ::: sclass : 32772 ::: Ref count : 5
EP Create Timestamp : 06/06/2018 05:41:25.617174
EP Update Timestamp : 06/06/2018 05:45:25.516971
EP Flags : on-peer|IP|MAC|sclass|
::::
```

- For regular EPG
- you can always derive src pcTag (src sclass) using epm local learning
- Dest pcTag (dclass in elam) it will be in EPM if EP is learned, if not we will use a default pcTag (1 I believe) and enforcement will be done on egress leaf)

# Finding which contract line we hit from ELAM ?

```
module-1(DBG-elam-insel6)#  report det | egrep -B 1
          sug_fpc_lookup_vec.fplu_vec.rslt.dciptrslt.pt.hit: 0x1
          sug_fpc_lookup_vec.fplu_vec.rslt.dciptrslt.pt.index: 0x13F64

module-1(DBG-elam-insel6)# dec 0x13f64
81764

module-1(DBG-elam-insel6)# show system internal  aclqos zoning-rules | egrep -A 5  -B 9 81764
============================================
Rule ID: 4196 Scope 11 Src EPG: 49154 Dst EPG: 16387 Filter 65535

  Curr TCAM resource:
  ============================
    unit_id: 0
    === Region priority: 2439 (rule prio: 9 entry: 135)===
        sw_index = 134 | hw_index = 156
    === SDK Info ===
        Result/Stats Idx: 81764
        88
        Tcam Total Entries: 1
        HW Stats: 1107546

============================================
bdsol-aci32-leaf4# show zoning-rule  | egrep 4196
4196        49154         16387           default       enabled    2654211         permit        src_dst_any(9)
bdsol-aci32-leaf4#
```

# Lab Contract layout



# Walking down Zoning-rule example

Order of line check :
4212, 4218,4219,4216 (prio7) - epg1 to epg4
4220, 4221,4222,4223 (prio 7 same level) – l3out-2 to EPG1
4207 (prio 15) - ingress 0.0.0.0/0 L3 out to ANY
4203, 4206 and 4212 (prio 16) - any to egress 0.0.0.0/0 L3 out
4197 (prio 17)
4159 (prio 21)
4202 (prio 22)

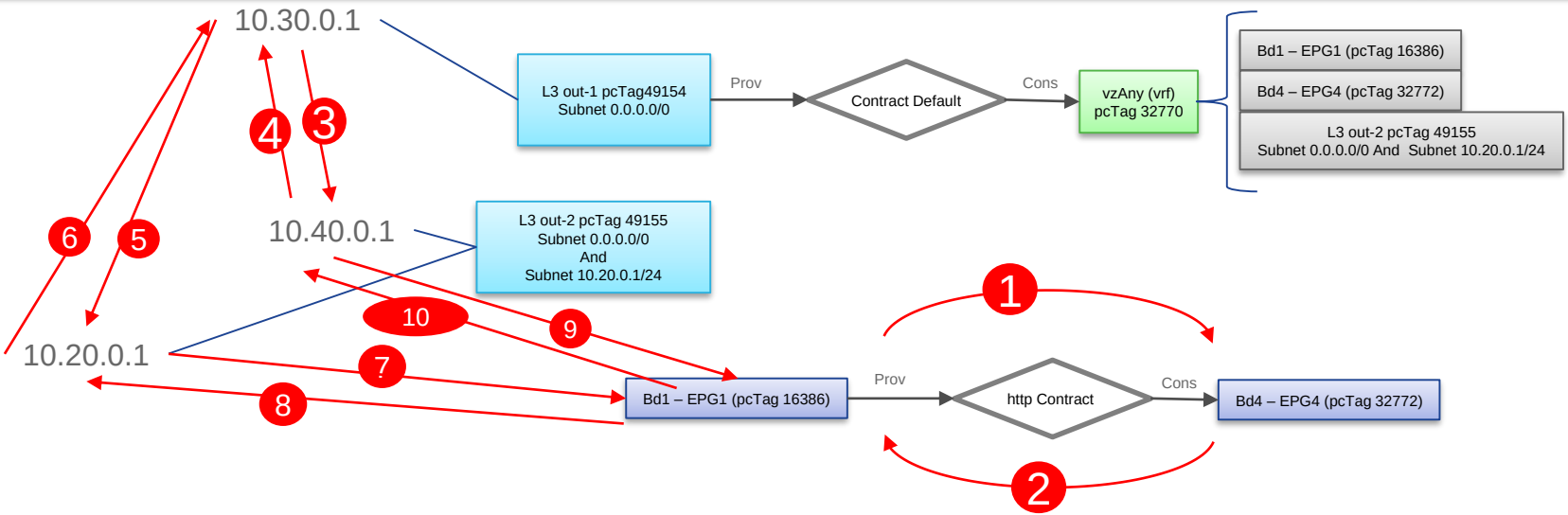
32770 is vrf pcTag
16386 is regular EPG1
32772 is regular EPG4
49154 is L3out-1 EPG
49155 is L3out-2 EPG
49153 is BD1 pcTag (not used)
16387 is BD4 pcTag (not used)

```
bdsol-aci32-leaf3#  show zoning-rule | egrep "Rule|==|2228225"
```

| Rule ID | SrcEPG | DstEPG | FilterID | operSt | Scope | Action | Priority |
|---|---|---|---|---|---|---|---|
| 4159 | 0 | 0 | implicit | enabled | 2228225 | deny,log | any_any_any(21) |
| 4197 | 0 | 0 | implarp | enabled | 2228225 | permit | any_any_filter(17) |
| 4202 | 0 | 15 | implicit | enabled | 2228225 | deny,log | any_vrf_any_deny(22) |
| 4203 | 0 | 49153 | implicit | enabled | 2228225 | permit | any_dest_any(16) |
| 4206 | 0 | 15 | default | enabled | 2228225 | permit | any_dest_any(16) |
| 4207 | 32770 | 0 | default | enabled | 2228225 | permit | src_any_any(15) |
| 4212 | 0 | 16387 | implicit | enabled | 2228225 | permit | any_dest_any(16) |
| 4217 | 32772 | 16386 | 43 | enabled | 2228225 | permit | fully_qual(7) |
| 4218 | 32772 | 16386 | 37 | enabled | 2228225 | permit | fully_qual(7) |
| 4219 | 16386 | 32772 | 38 | enabled | 2228225 | permit | fully_qual(7) |
| 4216 | 16386 | 32772 | 44 | enabled | 2228225 | permit | fully_qual(7) |
| 4220 | 16386 | 15 | 5 | enabled | 2228225 | permit | fully_qual(7) |
| 4221 | 32770 | 16386 | 5 | enabled | 2228225 | permit | fully_qual(7) |
| 4222 | 16386 | 49155 | 5 | enabled | 2228225 | permit | fully_qual(7) |
| 4223 | 49155 | 16386 | 5 | enabled | 2228225 | permit | fully_qual(7) |

| Rule ID | SrcEPG | DstEPG | FilterID | operSt | Scope | Action | Priority |
|---|---|---|---|---|---|---|---|
| 4159 | 0 | 0 | implicit | enabled | 2228225 | deny,log | any_any_any(21) |
| 4197 | 0 | 0 | implarp | enabled | 2228225 | permit | any_any_filter(17) |
| 4202 | 0 | 15 | implicit | enabled | 2228225 | deny,log | any_vrf_any_deny(22) |
| 4203 | 0 | 49153 | implicit | enabled | 2228225 | permit | any_dest_any(16) |
| 4206 | 0 | 15 | default | enabled | 2228225 | permit | any_dest_any(16) |
| 4 | 32770 | 0 | default | enabled | 2228225 | permit | src_any_any(15) |
| 4212 | 0 | 16387 | implicit | enabled | 2228225 | permit | any_dest_any(16) |
| 4217 | 32772 | 16386 | 43 | enabled | 2228225 | permit | fully_qual(7) |
| 4218 | 32772 | 16386 | 37 | enabled | 2228225 | permit | fully_qual(7) |
| 4219 | 16386 | 32772 | 38 | enabled | 2228225 | permit | fully_qual(7) |
| 4216 | 16386 | 32772 | 44 | enabled | 2228225 | permit | fully_qual(7) |
| 4220 | 16386 | 15 | 5 | enabled | 2228225 | permit | fully_qual(7) |
| 4221 | 32770 | 16386 | 5 | enabled | 2228225 | permit | fully_qual(7) |
| 4222 | 16386 | 49155 | 5 | enabled | 2228225 | permit | fully_qual(7) |
| 4223 | 49155 | 16386 | 5 | enabled | 2228225 | permit | fully_qual(7) |

6
3-4-5
2
1
10
9
8
7



# Verify your suspicion with policy-mgr stats

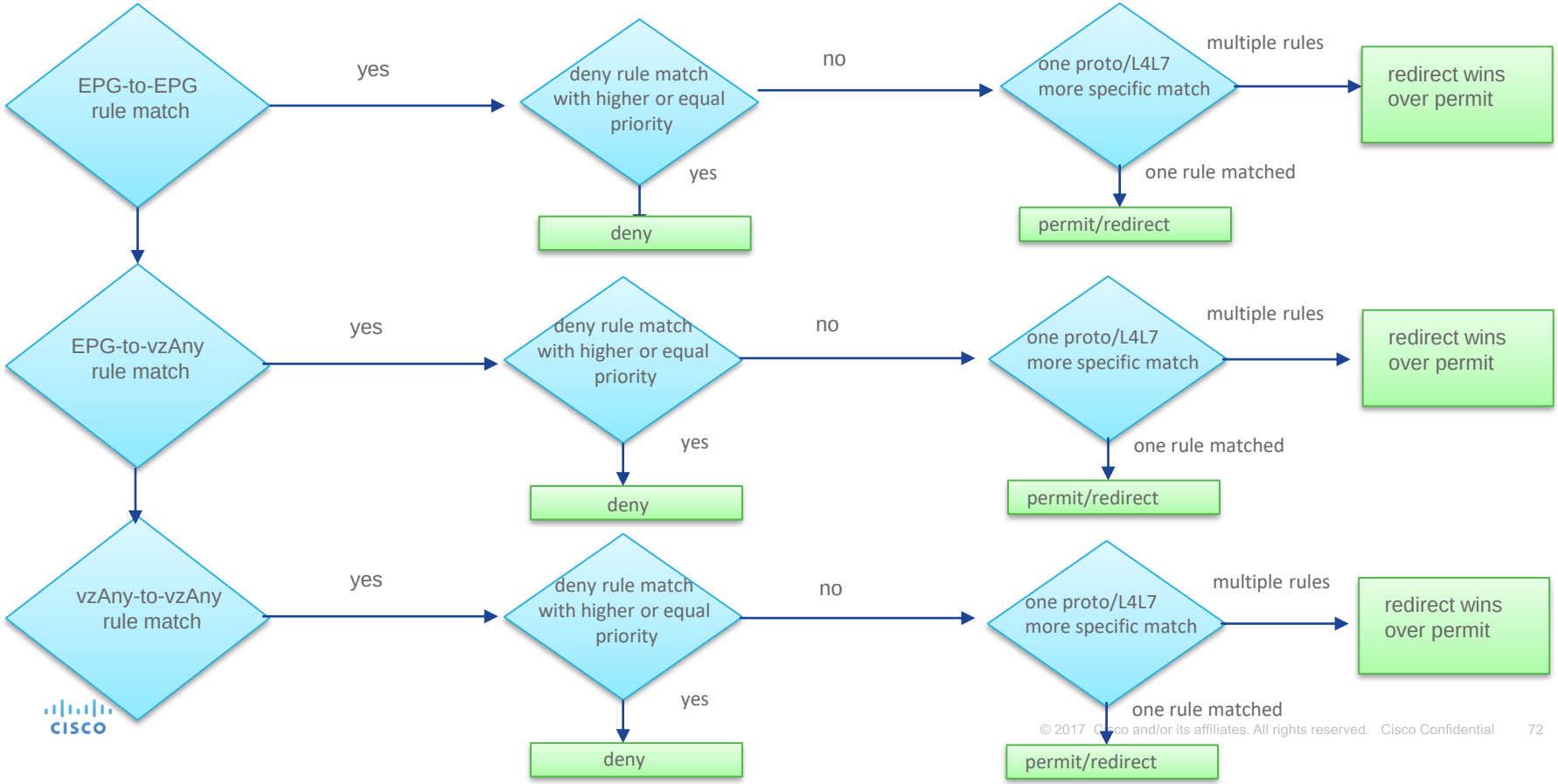
```
bdsol-aci32-leaf3# show system internal  policy-mgr stats | egrep "2228225"
Rule (4159) DN (sys/actrl/scope-2228225/rule-2228225-s-any-d-any-f-implicit) Ingress: 0, Egress: 0, Pkts: 22  RevPkts: 0
Rule (4197) DN (sys/actrl/scope-2228225/rule-2228225-s-any-d-any-f-implarp) Ingress: 0, Egress: 0, Pkts: 9  RevPkts: 0
Rule (4202) DN (sys/actrl/scope-2228225/rule-2228225-s-any-d-15-f-implicit) Ingress: 0, Egress: 0, Pkts: 0  RevPkts: 0
Rule (4203) DN (sys/actrl/scope-2228225/rule-2228225-s-any-d-49153-f-implicit) Ingress: 0, Egress: 0, Pkts: 17  RevPkts: 0
Rule (4206) DN (sys/actrl/scope-2228225/rule-2228225-s-any-d-15-f-default) Ingress: 0, Egress: 0, Pkts: 1771210  RevPkts: 0
Rule (4207) DN (sys/actrl/scope-2228225/rule-2228225-s-32770-d-any-f-default) Ingress: 0, Egress: 0, Pkts: 8309  RevPkts: 0
Rule (4212) DN (sys/actrl/scope-2228225/rule-2228225-s-any-d-16387-f-implicit) Ingress: 0, Egress: 0, Pkts: 17  RevPkts: 0
Rule (4216) DN (sys/actrl/scope-2228225/rule-2228225-s-16386-d-32772-f-44) Ingress: 0, Egress: 0, Pkts: 20  RevPkts: 0
Rule (4217) DN (sys/actrl/scope-2228225/rule-2228225-s-32772-d-16386-f-43) Ingress: 0, Egress: 0, Pkts: 0  RevPkts: 0
Rule (4218) DN (sys/actrl/scope-2228225/rule-2228225-s-32772-d-16386-f-37) Ingress: 0, Egress: 0, Pkts: 0  RevPkts: 0
Rule (4219) DN (sys/actrl/scope-2228225/rule-2228225-s-16386-d-32772-f-38) Ingress: 0, Egress: 0, Pkts: 1  RevPkts: 0
Rule (4220) DN (sys/actrl/scope-2228225/rule-2228225-s-16386-d-15-f-5) Ingress: 0, Egress: 0, Pkts: 2  RevPkts: 0
Rule (4221) DN (sys/actrl/scope-2228225/rule-2228225-s-32770-d-16386-f-5) Ingress: 0, Egress: 0, Pkts: 0  RevPkts: 0
Rule (4222) DN (sys/actrl/scope-2228225/rule-2228225-s-16386-d-49155-f-5) Ingress: 0, Egress: 0, Pkts: 17925  RevPkts: 0
Rule (4223) DN (sys/actrl/scope-2228225/rule-2228225-s-49155-d-16386-f-5) Ingress: 0, Egress: 0, Pkts: 17925  RevPkts: 0
```

# Understanding Rule Priority and Entry Priority

# High Level Filtering Rules

- More specific EPGs win over vzAny
- More specific L4 rules win (it applies between permit and redirect)
- Deny wins over Redirect which wins over Permit

# How to figure out which rule wins



# High Level Filtering Rules

- Higher Rule Priority wins over Lower Priority:
  - More specific EPG (e.g. EPG-to-EPG) wins over vzAny-to-EPG or EPG-to-vzAny
  - vzAny-to-EPG wins over vzAny-to-vzAny

- Within the same Priority
- deny wins over all other actions

- between redirect and permit: more specific protocol and L4 port wins (pls refer to the table of which rule is considered more specific)

# Rule Priority

```
<type name="RulePrio"
      base="scalar:UByte"
      >
    <range min="1" max="23"/>
    <const name="class-eq-filter" value="1"/>
    <const name="class-eq-deny" value="2"/>
    <const name="class-eq-allow" value="3"/>
    <const name="prov-nonshared-to-cons" value="4"/>
    <const name="black_list" value="5"/>
    <const name="fabric_infra" value="6"/>
    <const name="fully_qual" value="7"/>
    <const name="system_incomplete" value="8"/>
    <const name="src_dst_any" value="9"/>
    <const name="shsrc_any_filt_perm" value="10"/>
    <const name="shsrc_any_any_perm" value="11"/>
    <const name="shsrc_any_any_deny" value="12"/>
    <const name="src_any_filter" value="13"/>
    <const name="any_dest_filter" value="14"/>
    <const name="src_any_any" value="15"/>
    <const name="any_dest_any" value="16"/>
    <const name="any_any_filter" value="17"/>
    <const name="grp_src_any_any_deny" value="18"/>
    <const name="grp_any_dest_any_deny" value="19"/>
    <const name="grp_any_any_any_permit" value="20"/>
    <const name="any_any_any" value="21"/>
    <const name="any_vrf_any_deny" value="22"/>
```

# Entry Priority

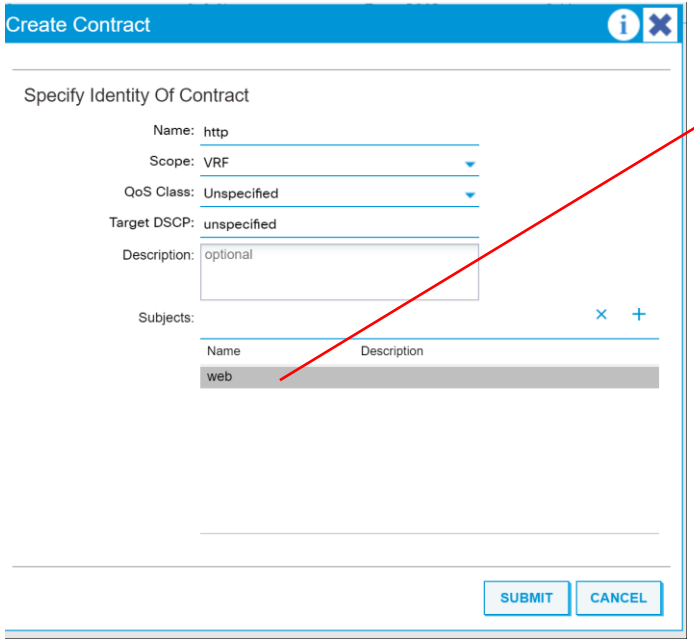
```
<type name="EntryPrio"
      base="scalar:UByte"
      >
    <range min="1" max="8"/>
    <const name="flags" value="1"/>
    <const name="sport_dport" value="2"/>
    <const name="dport" value="3"/>
    <const name="sport" value="4"/>
    <const name="proto" value="5"/>
    <const name="frag" value="6"/>
    <const name="def" value="7"/>
    <const name="implicit" value="8"/>
    <default value="def"/>
</type>
```

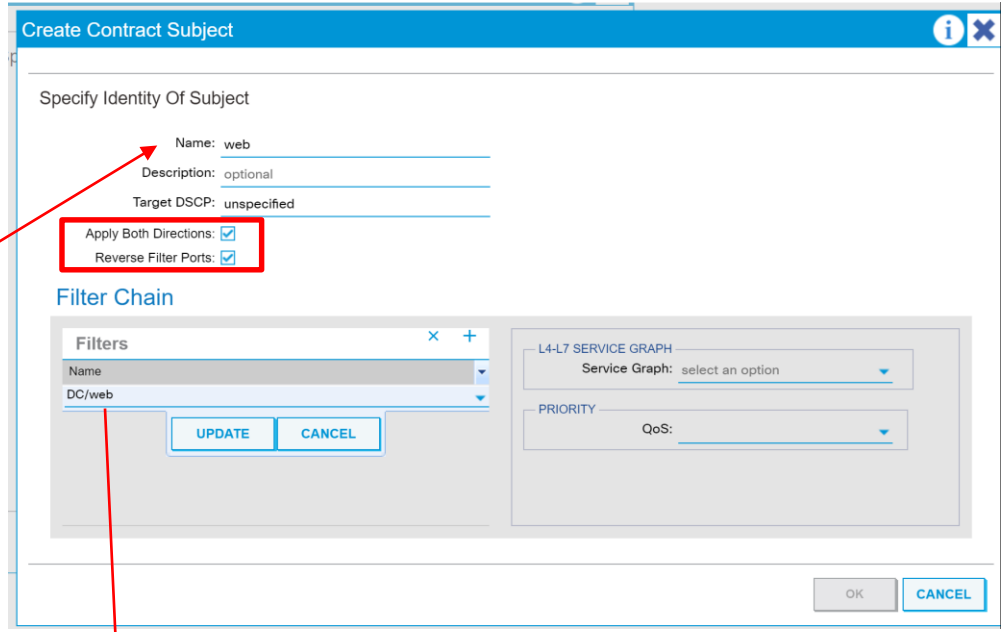
# Example of TCAM usage

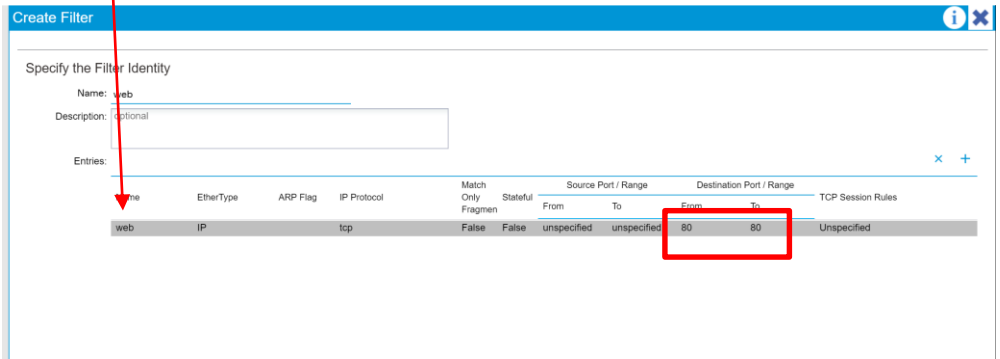
# Example 1- What about contract Direction ?

- Normally never need to apply contract Bidirectionally
- See Provider as the EPG that will allow traffic to Dest Port specified in the contract (we provide Web services in EPG Web by allowing traffic with Dest Port 80 to enter EPG Web where Web server sits)
- For UDP or TCP direction is important !!!
- (not relevant for ICMP ,…)

# Contract Config



# We Provide HTTP Contract on EPG1 consumed on EPG2

```
pod2-leaf3#  show zoning-rule | egrep "32770.*2097153|Rule"
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action          Priority
4172            49153           32770           6               enabled         2097153         permit          fully_qual(6)
4173            32770           49153           7               enabled         2097153         permit          fully_qual(6)



pod2-leaf3# show zoning-filter | egrep "^6|^7|Filter"
FilterId  Name          EtherT      ArpOpc      Prot        MatchOnlyFrag Stateful SFromPort   SToPort     DFromPort   DToPort     Prio
Icmpv4T     Icmpv6T     TcpRules
7         7_0           ip          unspecified tcp         no            no       http        http        unspecified unspecified sport
6         6_0           ip          unspecified tcp         no            no       unspecified unspecified http        http        dport
pod2-leaf3#
```

We allow Filter 6 from EPG2 to EPG1 – Filter 6 allow Traffic From Sport http
We allow Filter 7 from EPG1 to EPG2 – Filter 7 allow traffic To Dort http

# Example 2 - What if I provide and consume Web contract on both EPG ?

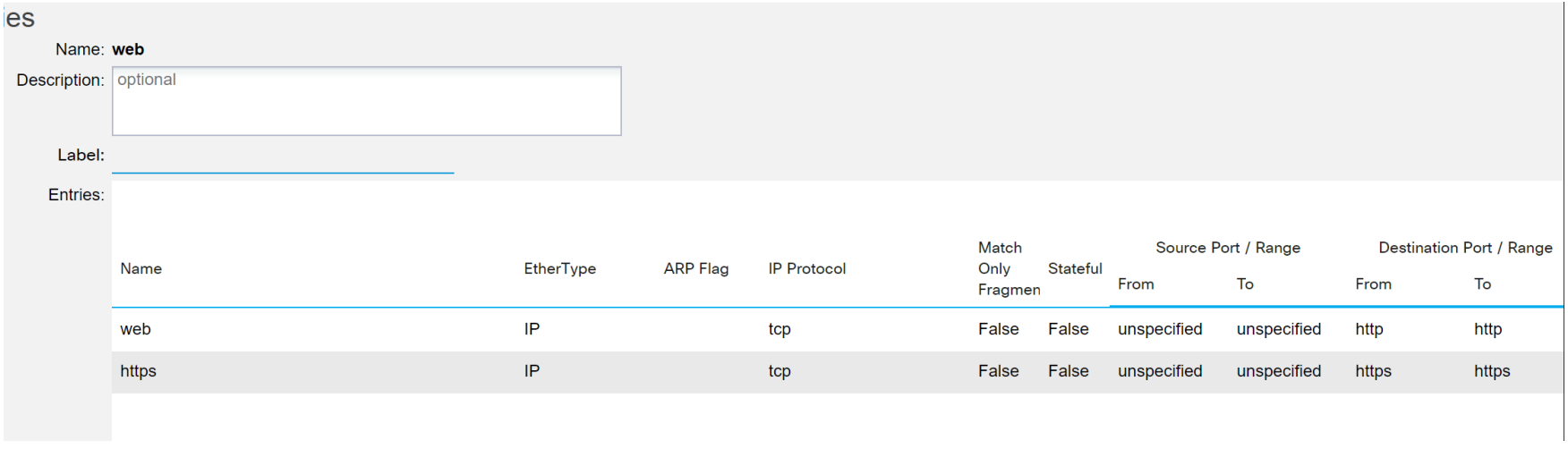
```
pod2-leaf3# show zoning-rule | egrep "32770.*2097153|Rule"
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action          Priority
4172            49153           32770           6               enabled         2097153         permit          fully_qual(6)
4173            32770           49153           7               enabled         2097153         permit          fully_qual(6)
4175            49153           32770           7               enabled         2097153         permit          fully_qual(6)
4174            32770           49153           6               enabled         2097153         permit          fully_qual(6)
pod2-leaf3#



pod2-leaf3# show zoning-filter | egrep "^6|^7|Filter"
FilterId  Name          EtherT      ArpOpc      Prot        MatchOnlyFrag Stateful SFromPort   SToPort     DFromPort   DToPort     Prio
Icmpv4T     Icmpv6T     TcpRules
7         7_0           ip          unspecified tcp         no            no       http        http        unspecified unspecified sport
6         6_0           ip          unspecified tcp         no            no       unspecified unspecified http        http        dport
pod2-leaf3#
```

Now we have following behavior (drawback):

- 4 lines in zoning-rule (TCAM) – double TCAM space
- We allow both Web traffic to enter EPG1 (Web server) and Web traffic to leave EPG1 (Web server) to go to EPG2 (client)

# Example 3 - What is we have multiple line in the filter ?

- Each line in filter will consume an extra line in TCAM as each filter will contain two entries
- Now My filter have http an https filter



es

Name: **web**

Description: optional

Label:

Entries:

| Name | EtherType | ARP Flag | IP Protocol | Match Only Fragmen | Stateful | Source Port / Range | | Destination Port / Range | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | From | To | From | To |
| web | IP | | tcp | False | False | unspecified | unspecified | http | http |
| https | IP | | tcp | False | False | unspecified | unspecified | https | https |

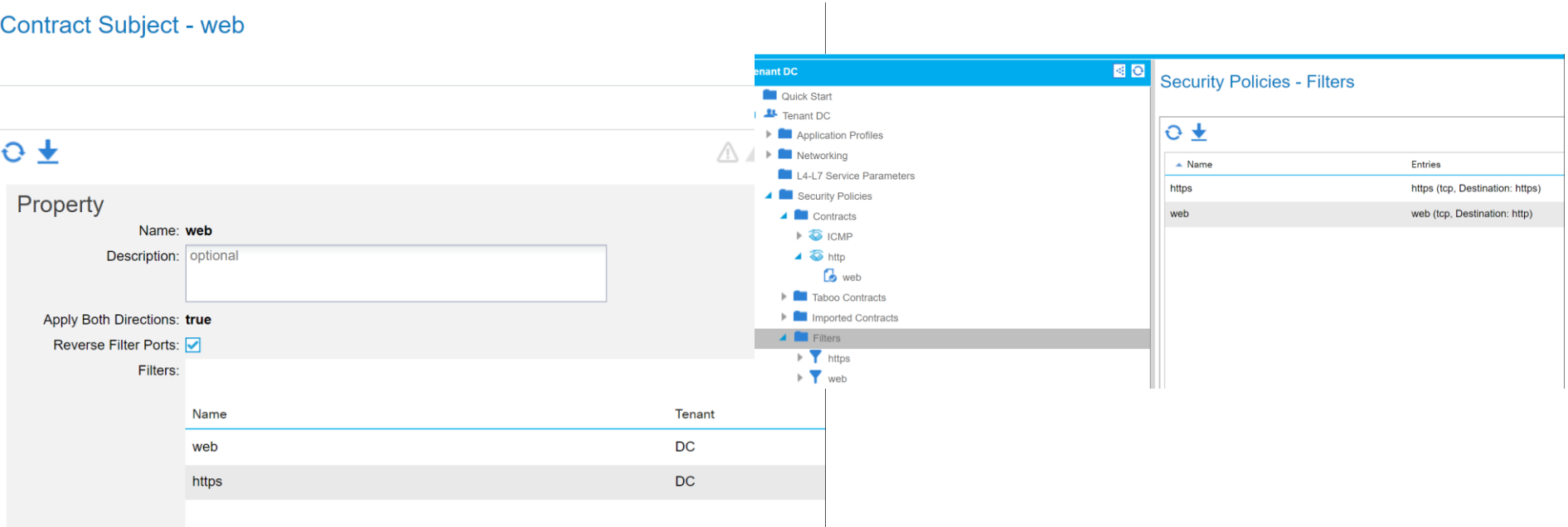
# Single Direction contract But two line per Filter

```
pod2-leaf3# show zoning-rule | egrep "32770.*2097153|Rule"
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action          Priority
4176            32770           49153           50              enabled         2097153         permit          fully_qual(6)
4177            49153           32770           49              enabled         2097153         permit          fully_qual(6)



pod2-leaf3# show zoning-filter | egrep "^50|^49|Filter"
FilterId  Name          EtherT      ArpOpc      Prot        MatchOnlyFrag Stateful SFromPort   SToPort     DFromPort   DToPort     Prio
Icmpv4T     Icmpv6T     TcpRules
50        50_1          ip          unspecified tcp         no            no       http        http        unspecified unspecified sport
50        50_0          ip          unspecified tcp         no            no       https       https       unspecified unspecified sport
49        49_1          ip          unspecified tcp         no            no       unspecified unspecified http        http        dport
49        49_0          ip          unspecified tcp         no            no       unspecified unspecified https       https       dport
pod2-leaf3#
```

Now out filter (50  and 49) each have two lines
50_1 providing http and 50_0 providing https
49_1 consuming http and 49_0 consuming https

# Example 4 - What is  we use two one line filter instead ?

- Now I am not using anymore a filter with 2 protocol, but I am rather using two different filters , one for http, one for https.
- Then the subject contains both filters (functionally the same as previous



# Single direction contract – two filter in the subject one line per filter

```
pod2-leaf3# show zoning-rule | egrep "32770.*2097153|Rule"
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action          Priority
4178            32770           49153           7               enabled         2097153         permit          fully_qual(6)
4179            49153           32770           6               enabled         2097153         permit          fully_qual(6)
4176            32770           49153           50              enabled         2097153         permit          fully_qual(6)
4177            49153           32770           49              enabled         2097153         permit          fully_qual(6)


pod2-leaf3# show zoning-filter | egrep "^7|^6|^50|^49|Filter"
FilterId  Name          EtherT      ArpOpc      Prot        MatchOnlyFrag Stateful SFromPort   SToPort     DFromPort   DToPort     Prio
Icmpv4T     Icmpv6T     TcpRules
7         7_0           ip          unspecified tcp         no            no       http        http        unspecified unspecified sport
6         6_0           ip          unspecified tcp         no            no       unspecified unspecified http        http        dport
50        50_0          ip          unspecified tcp         no            no       https       https       unspecified unspecified sport
49        49_0          ip          unspecified tcp         no            no       unspecified unspecified https       https       dport
```

Compare with previous example :
Programing wise we now have 4 zoning-rule each with one line filter with 4 different filters.
From Resource consumption perspective : No changes , 4 lines consumed in each case
Fonctionality wise nothing changes

# Example 5 – Multiple EPG Web server

- What if  we want to enable Web port 80 from EPG1 to EPG2 AND Web port 80 from EPG3 to EPG4
- Approach 1 – we provide Web contract (single filter port 80) on epg1 and epg3 and consume it in epg2 and epg4

```
pod2-apic1# moquery -c fvAEPg | egrep -7 "dn.*tn-DC.*EPG" | egrep "#|dn|pcTag"
# fv.AEPg
dn              : uni/tn-DC/ap-App/epg-EPG1
pcTag           : 32770
# fv.AEPg
dn              : uni/tn-DC/ap-App/epg-EPG2
pcTag           : 49153
# fv.AEPg
dn              : uni/tn-DC/ap-App/epg-EPG3
pcTag           : 49156
# fv.AEPg
dn              : uni/tn-DC/ap-App/epg-EPG4
pcTag           : 49159
```

# Resulting programmation

```
pod2-leaf3# show zoning-rule | egrep "32770.*2097153|49156.*2097153|Rule"
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action          Priority
4290            32770           49153           7               enabled         2097153         permit          fully_qual(6)
4291            49153           32770           6               enabled         2097153         permit          fully_qual(6)
4224            49153           49156           6               enabled         2097153         permit          fully_qual(6)
4225            49156           49153           7               enabled         2097153         permit          fully_qual(6)
4226            49159           32770           6               enabled         2097153         permit          fully_qual(6)
4227            32770           49159           7               enabled         2097153         permit          fully_qual(6)
4228            49159           49156           6               enabled         2097153         permit          fully_qual(6)
4229            49156           49159           7               enabled         2097153         permit          fully_qual(6)
FilterId  Name          EtherT      ArpOpc      Prot        MatchOnlyFrag Stateful SFromPort   SToPort     DFromPort   DToPort     Prio
7         7_0           ip          unspecified tcp         no            no       http        http        unspecified unspecified sport
6         6_0           ip          unspecified tcp         no            no       unspecified unspecified http        http        dport
pod2-leaf3#
```

Now we have 8 lines for only http :
For filter 6 we have both EPG2 (49153) talking to EPG1 (32770) and to EPG3 (49156) for port 80
And also EPG4 (49159) taking to talking to EPG1 (32770) and to EPG3 (49156) for port 80

The 4 remaining rules provides return traffic (filter 7)

This is not what we wanted as EPG1 should not provide web to EPG4
and EPG3 should nor provide Web to EPG2

# Example 6 – Two different contract

- we still want to enable Web port 80 from EPG1 to EPG2 AND Web port 80 from EPG3 to EPG4
- Approach 2 – we create a second contract http2 using same filter (web port 80) :
  - We apply contract http between epg1 to epg2
  - We apply contract http2 between epg3 and epg4

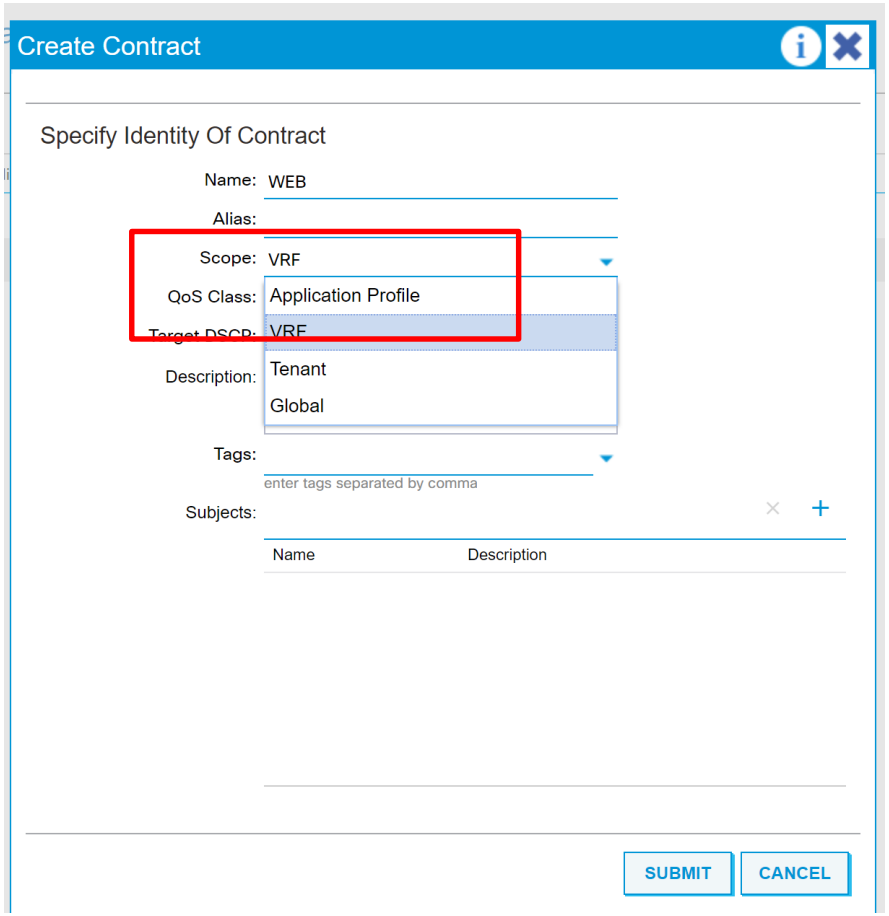
# Resulting programmation

```
pod2-leaf1# show zoning-rule  | egrep 2097153
4290            32770           49153           7               enabled         2097153         permit          fully_qual(6)
4291            49153           32770           6               enabled         2097153         permit          fully_qual(6)
4228            49159           49156           6               enabled         2097153         permit          fully_qual(6)
4229            49156           49159           7               enabled         2097153         permit          fully_qual(6)
pod2-leaf1#
```

Now have what we want epg2 (49153) can only reach epg1 (32770)
And epg4 (49159) can only reach epg3 (49156).

No need to create different filter, filter can be reused (note if we create different Filter you would achieve exact same result and programmation

# Example 6 – Contact scope alternative




## Contract Scope

The contract scope will limit which providers and consumers can participate within the same contract.

**VRF**
The contract can be applied between EPGs within the same VRF.

**Application Profile**
The contract can be applied between EPGs within the same application profile

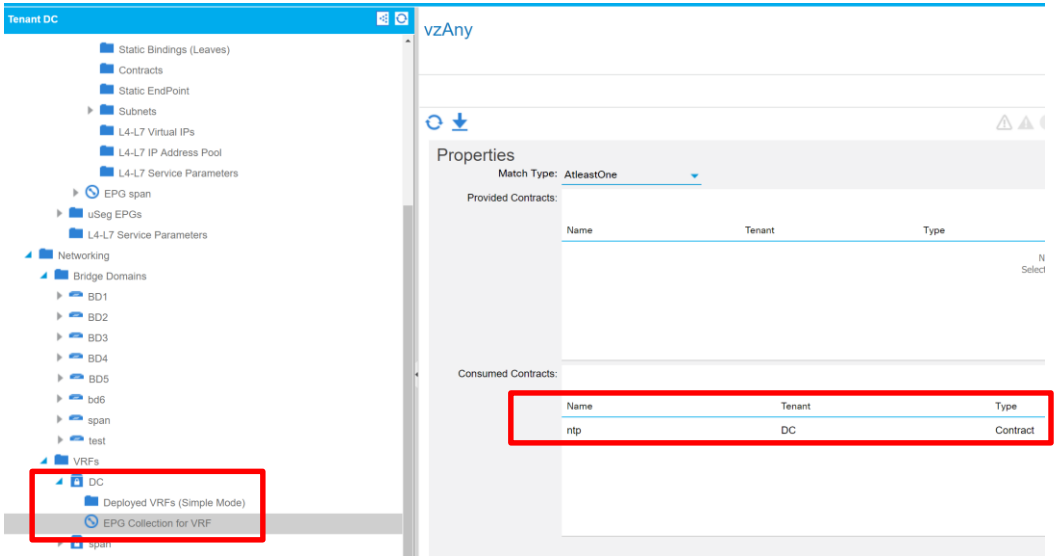
**Tenant**
The contract can be applied between EPGs within the same tenant.

**Global**
The contract can be applied between any EPGs within the fabric. Note, global contracts not in common tenant need to be exported in order to be consumed by EPG in a different tenant. Consumers of global contracts will use the ' **Consumer Contract Interface** ' Option

By using the web contract scope as Application Profile. Then we can reuse the same contract Between EPG1 to EPG2 place in App1
And EPG3 to EPG4 placed in App2
Functionnally and resource wise this is equivalent

# Example 7 – Usage of vzAny

- Here we have in EPG5, NTP servers and we want all EPG in the VRF to be able to reach those NTP servers.
- EPG5 will provide NTP contract and we will consume NTP contract in the vrf (vzAny consume)



# Result

```
pod2-apic1# moquery -c fvAEPg  | egrep -8 "dn.*tn-DC.*EPG"  | egrep "#|dn|pcTag"
# fv.AEPg
dn                  : uni/tn-DC/ap-App/epg-EPG5
pcTag               : 32779


pod2-leaf3# show zoning-rule | egrep "32779.*2097153"
4202            32779           0               34              enabled         2097153         permit          src_any_filter(12)
4203            0               32779           33              enabled         2097153         permit          any_dest_filter(13)
pod2-leaf3#
pod2-leaf3# show zoning-filter  | egrep "^34|^33"
34        34_0          ip          unspecified tcp         no            yes      123         123         unspecified unspecified flags
33        33_0          ip          unspecified tcp         no            yes      unspecified unspecified 123         123         dport
pod2-leaf3#
```

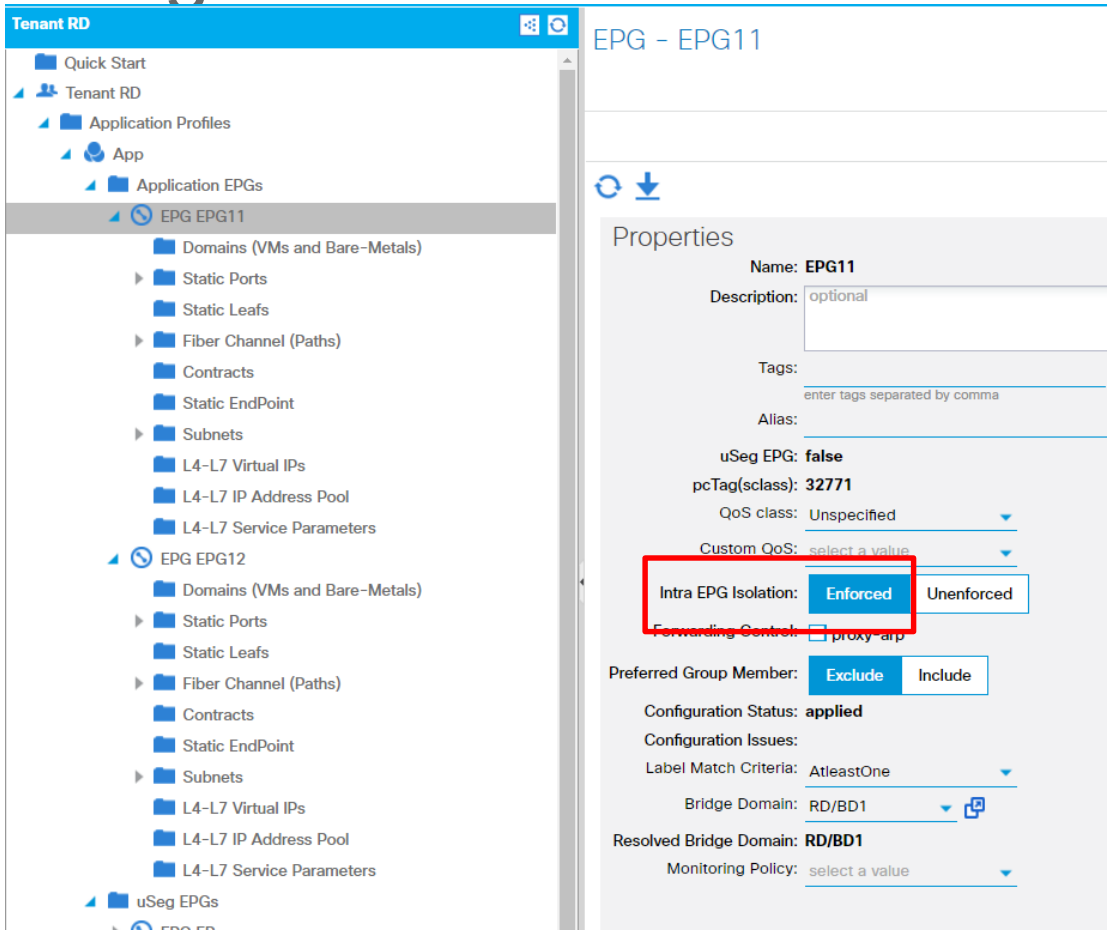
Any EPG in the scope (vrf 2097153) can go to EPG 32779 with filter 33
That allows dest port 123 (NTP)
And vice versa

# Example 8 – Intra EPG isolation

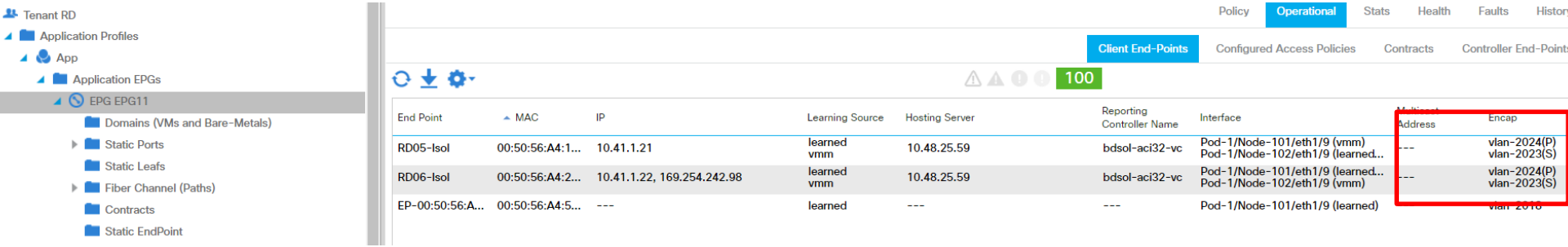
- Now we do not want any communication between VM in same EPG

- Solution – Intra EPG isolation flags
- This turns out the VLAN to a Isolated Private VLAN (both in Vcenter and in ACI switch)

# Intra-EPG isolation config



# ACI Gui effect



# Vlan setting in ACI

```
bdsol-aci32-leaf1# show vlan extended  | egrep "RD.*:EPG11|vlan-2023"

 35   RD:App:EPG11                     active    Eth1/9

 35   enet  CE         vlan-2023
```

```
module-1# show system internal eltmc info vlan 35  | egrep "vlan_|access|isol"
            vlan_id:              35   :::      hw_vlan_id:              28
          vlan_type:         FD_VLAN   :::         bd_vlan:              33
   access_encap_type:          802.1q   :::    access_encap:            2023
     pd_vlan_ft_mask:             0x8
        vlan_ft_mask:          0x7830
            vlan_id:              35   :::           isEpg:               1
          bd_vlan_id:              33   :::         hwEpgId:           11308
            isolated:               1   :::   primary_encap:            2024
          vlan_type:              13
        access encap:            2023
```

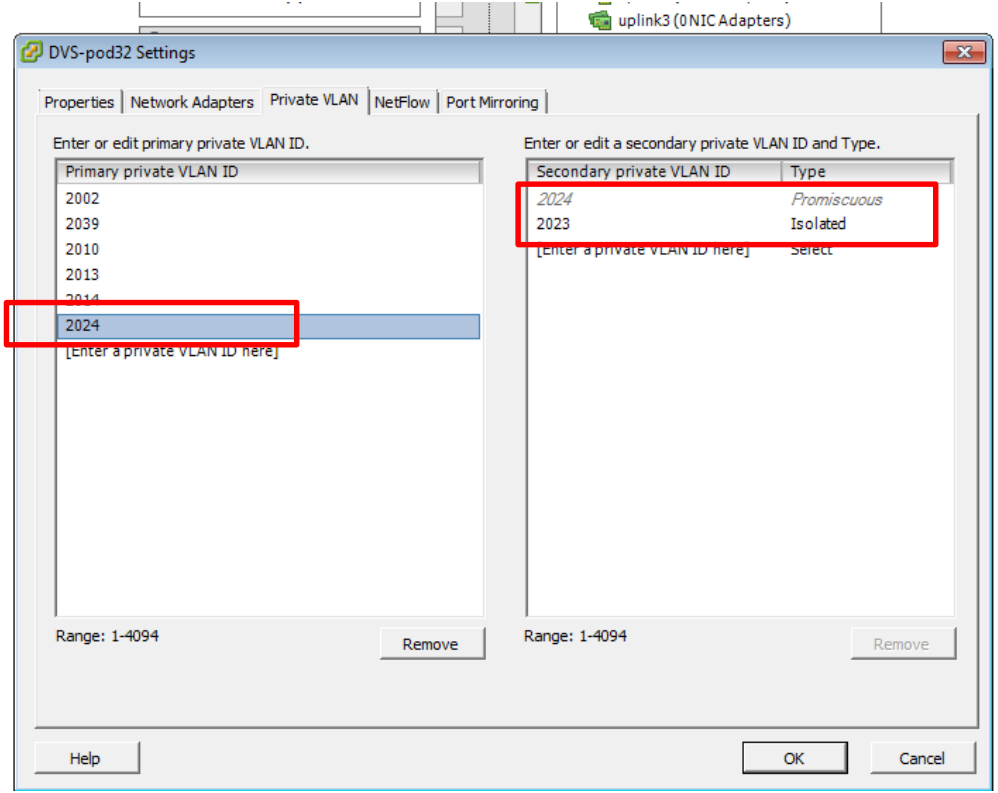
# Policy enforcement at leaf level

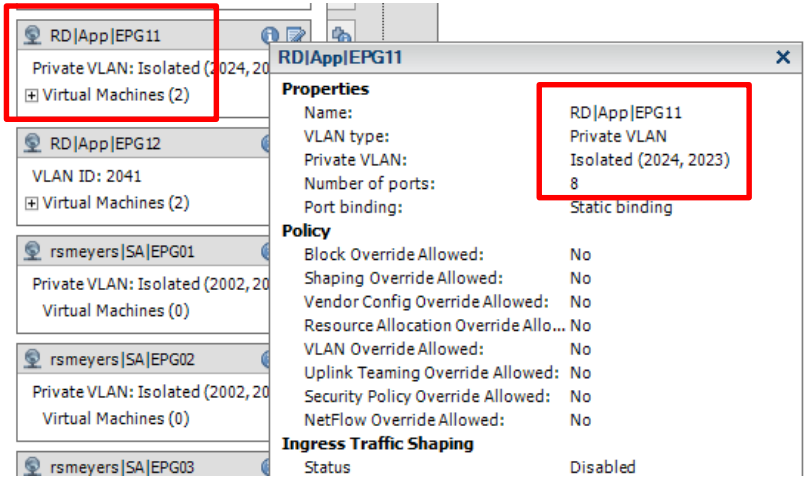
```
apic1# moquery -d uni/tn-RD/ctx-RD | egrep scope
scope            : 2686976
apic1#
apic1# moquery -d  uni/tn-RD/ap-App/epg-EPG11 | egrep pcTag
pcTag          : 32771
apic1#
```

```
bdsol-aci32-leaf1# show zoning-rule | egrep " 32771.*2686976 "
4170       32771      32771      implicit    enabled   2686976    deny,log                  class-eq-deny(1)
bdsol-aci32-leaf1# show system internal  policy-mgr stats | egrep 4170
Rule (4170) DN (sys/actrl/scope-2686976/rule-2686976-s-32771-d-32771-f-implicit) Ingress: 0, Egress: 0, Pkts: 42
RevPkts: 0
bdsol-aci32-leaf1#
```

A rule is added to drop Intre EPG traffic (src epg = Dst epg)
See priority is 1 (highest)

# Policy enforcement in Vcenter



# Example 9 – Preferred group

- Intermediate between enforce mode (default deny rule) and unenforced (default permit rule – no other rule)
- With preferred group , some epg in the vrf are in the preferred group where communication is all open (unenforced behavior) and the remaining EPG are following enforced rule

# Example 9 – Preferred Group – vrf config

- Preferred group must be enable at vrf level first
  - vzAnyGroupDef and fvCtx
  - `<fvCtx name="lab"> <vzAny prefGrMemb="enabled"/> </fvCtx>`

```
admin@apic1:~> moquery -d  anydefcont/anygroupdef-[uni/tn-
RD/ctx-SB/any]
Total Objects shown: 1

# vz.AnyGroupDef
anyDn         : uni/tn-RD/ctx-SB/any
childAction   :
descr         :
dn            : anydefcont/anygroupdef-[uni/tn-RD/ctx-SB/any]
lcOwn         : local
modTs         : 2017-10-23T06:40:20.385+00:00
monPolDn      :
name          :
nameAlias     :
prefGrMemb    : enabled
rn            : anygroupdef-[uni/tn-RD/ctx-SB/any]
status        :
```

```
# vz.Any
childAction   :
configSt      : not-applied
descr         :
dn            : uni/tn-RD/ctx-SB/any
lcOwn         : local
matchT        : AtleastOne
modTs         : 2017-10-23T06:40:20.385+00:00
monPolDn      : uni/tn-common/monepg-default
name          :
nameAlias     :
pcTag      nn    : any
prefGrMemb    : enabled
rn            : any
status        :
uid           : 0
useAnyDef     : yes
```

# Example 9 – Preferred Group – EPG config

- Preferred group can then be enable at the epg level (or l3extInsP)
  - `<fvAEPg name="mail" prefGrMemb="include">`

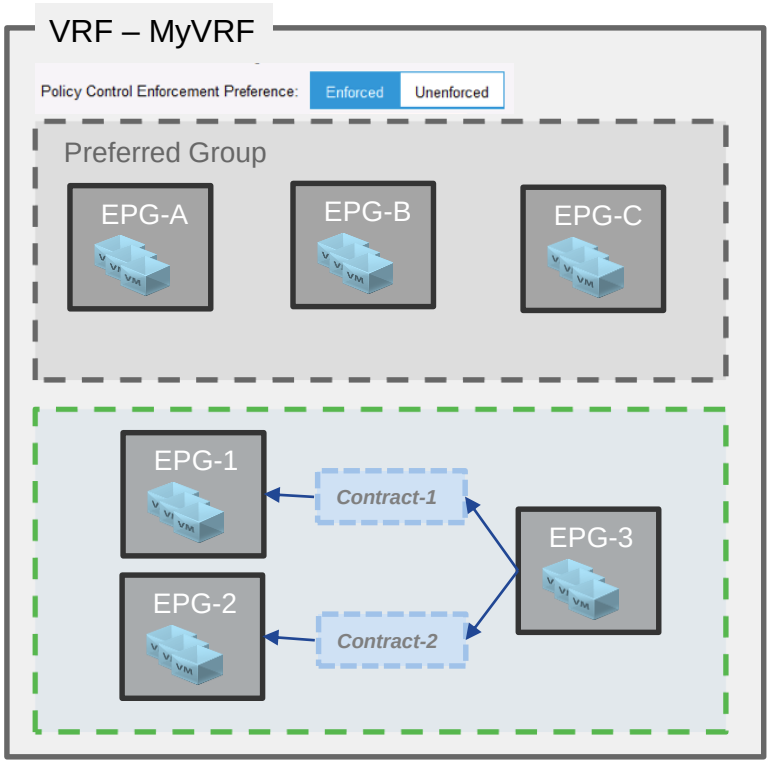
```
admin@apic1:~> moquery -c fvAEPg -f 'fv.AEPg.prefGrMemb
=="include"'
Total Objects shown: 2

# fv.AEPg
name                  : SB-epg1
childAction           :
configIssues          :
configSt              : applied
descr                 :
dn                    : uni/tn-RD/ap-SB/epg-SB-epg1
extMngdBy             :
fwdCtrl               :
isAttrBasedEPg        : no
isSharedSrvMsiteEPg   : no
lcOwn                 : local
matchT                : AtleastOne
```

```
modTs                 : 2017-10-
23T05:44:32.918+00:00
monPolDn              : uni/tn-common/monepg-
default
nameAlias             :
pcEnfPref             : unenforced
pcTag                 : 16388
prefGrMemb            : include
prio                  : unspecified
rn                    : epg-SB-epg1
scope                 : 2097153
status                :
triggerSt             : triggerable
txId                  : 5764607523041268653
uid                   : 15374
```

# Example 9 – preferred group –zoning rule

- We install 3 type of zoning-rule
  1. EPG to EPG zoning-rule for contract between EPG out of the preferred group with lowest priority (typically 6) – Usual enforce behavior
  2. Any to/from EPG for EPG out of preferred group to deny all other traffic to/from those epg with med prio (17 or 18 I believe)
  3. Any to Any with permit all to allow traffic inside the preferred group

# Contract Preferred Group In Practice




EPGs with contracts (priority 7)

| Source | Destination | Filter | Action |
|---|---|---|---|
| EPG-3 | EPG-1 | contract-1 | permit |
| EPG-1 | EPG-2 | contract-1-r | permit |
| EPG-3 | EPG-2 | contract-2 | permit |
| EPG-2 | EPG-3 | Contract-2-r | permit |

Rules for the Excluded EPGs added by Preferred Group (priority 18)

| Source | Destination | Filter | Action |
|---|---|---|---|
| EPG-1 | any | implicit | deny |
| any | EPG-1 | implicit | deny |
| EPG-2 | any | implicit | deny |
| any | EPG-2 | implicit | deny |
| EPG-3 | any | implicit | deny |
| any | EPG-3 | implicit | deny |

Rule for the Preferred Group EPGs (priority 20)

| Source | Destination | Filter | Action |
|---|---|---|---|
| any | any | implicit | permit |

# Preferred Group rules priority

- The **deny rules** for EPGs that are not part of the Preferred Group have a priority of 18 and 19
- 0 15 implicit enabled 2260992 deny,log grp_any_dest_any_deny(19) (*)
- 0 49161 implicit enabled 2260992 deny,log grp_src_any_any_deny(18)
- 0 49161 implicit enabled 2260992 deny,log grp_any_dest_any_deny(19)
- The implicity permit from the Preferred Group has priority of 20
- 0 0 implicit enabled 2260992 permit **grp_any_any_any_permit(20)**

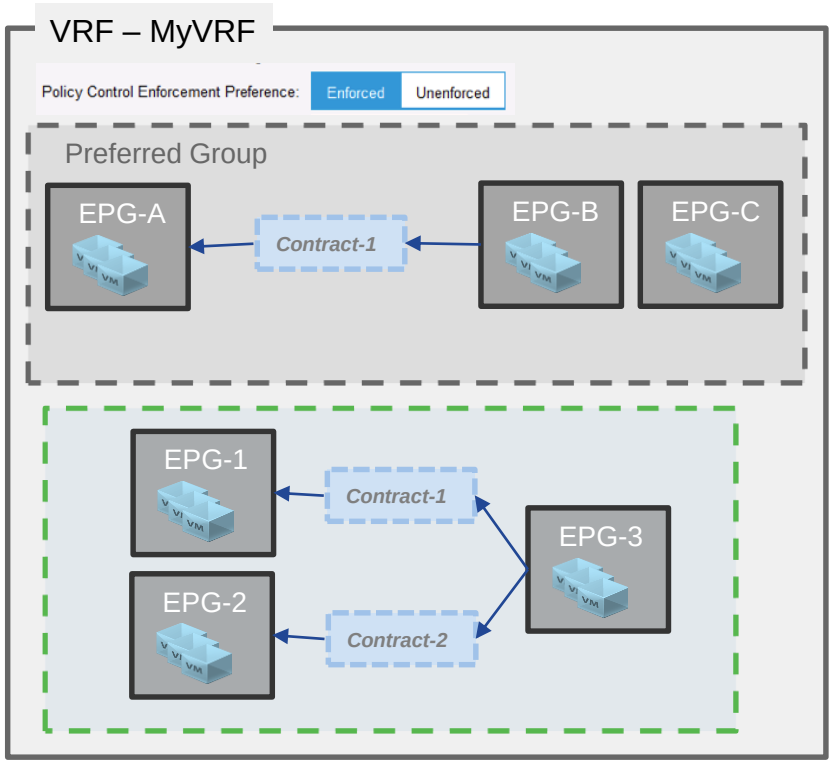
# Example 9

Epg 32771 and 49155 are out of preferred group and allow filter 5 between them
Epg 32773, 49155, 32771 and 32772 are outside of preferred group and consume
Two lines per EPG to exclude what is not permitted by contract
For EPG in preferred group, we have nothing but a permit any any with lower prio.

```
bdsol-aci32-leaf1#  show zoning-rule  | egrep 2097153
4113            0               0               implicit        enabled         2097153         permit          grp_any_any_any_permit(20)
4114            0               0               implarp         enabled         2097153         permit          any_any_filter(17)
4115            0               15              implicit        enabled         2097153         deny,log        grp_any_dest_any_deny(19)
4133            0               49154           implicit        enabled         2097153         permit          any_dest_any(16)
4130            0               16386           implicit        enabled         2097153         permit          any_dest_any(16)
4216            0               16387           implicit        enabled         2097153         permit          any_dest_any(16)
4217            32771           49155           5               enabled         2097153         permit          fully_qual(7)
4218            49155           32771           5               enabled         2097153         permit          fully_qual(7)
4220            32773           0               implicit        enabled         2097153         deny,log        grp_src_any_any_deny(18)
4221            0               32773           implicit        enabled         2097153         deny,log        grp_any_dest_any_deny(19)
4222            49155           0               implicit        enabled         2097153         deny,log        grp_src_any_any_deny(18)
4223            0               49155           implicit        enabled         2097153         deny,log        grp_any_dest_any_deny(19)
4224            32771           0               implicit        enabled         2097153         deny,log        grp_src_any_any_deny(18)
4225            0               32771           implicit        enabled         2097153         deny,log        grp_any_dest_any_deny(19)
4164            32772           0               implicit        enabled         2097153         deny,log        grp_src_any_any_deny(18)
4165            0               32772           implicit        enabled         2097153         deny,log        grp_any_dest_any_deny(19)
bdsol-aci32-leaf1#
```

Be aware of scalability :
Each EPG non part of preferred group will take 2 extra entry (+ their configured contract )

# What if you configure a contract between EPGs that are part of a preferred Group




EPGs with contracts (priority 7)

| Source | Destination | Filter | Action |
|---|---|---|---|
| EPG-A | EPG-B | contract-1 | permit |
| EPG-B | EPG-A | contract-1-r | permit |
| EPG-3 | EPG-1 | contract-1 | permit |
| EPG-1 | EPG-2 | contract-1-r | permit |
| EPG-3 | EPG-2 | contract-2 | permit |
| EPG-2 | EPG-3 | Contract-2-r | permit |

Rules for the Excluded EPGs added by Preferred Group (priority 18

| Source | Destination | Filter | Action |
|---|---|---|---|
| EPG-1 | any | implicit | deny |
| any | EPG-1 | implicit | deny |
| Etc… | | | |

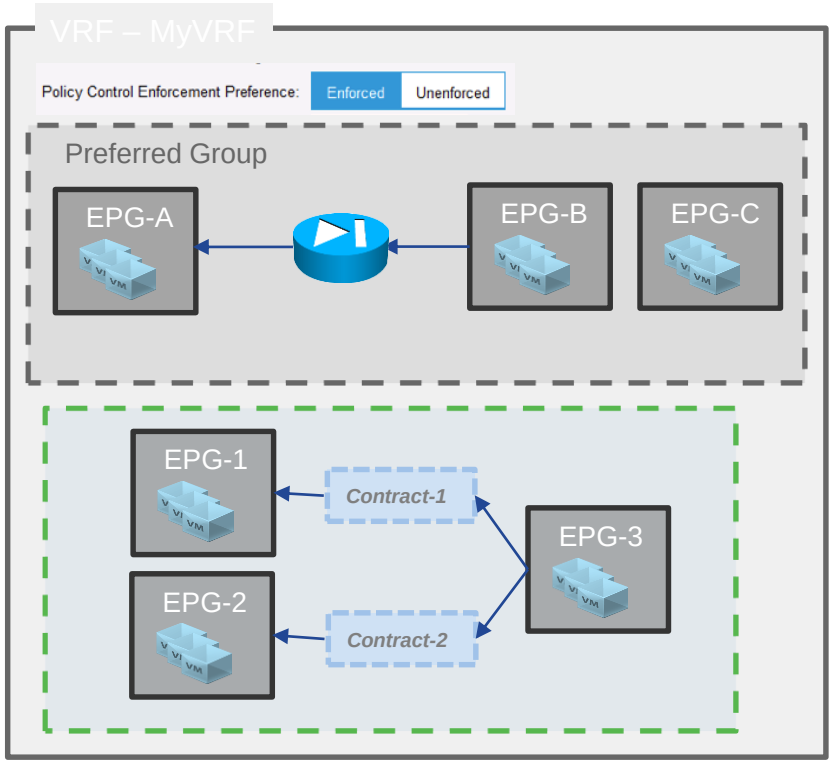
Rule for the Preferred Group EPGs (priority 20)

| Source | Destination | Filter | Action |
|---|---|---|---|
| any | any | implicit | permit |

# What if you configure a contract between EPGs that are part of a preferred Group

- IF you configure contracts between EPGs that are also preferred group EPGs, the contract entries are programmed with priority 7
- If the contract has a permit, these rules serve no purposes
- If the contract is associated with a service graph redirect it allows the service graph to work between EPGs that are part of the preferred group

# Service Graph redirect between Preferred Groups EPGs




EPGs with contracts (priority 7)

| Source | Destination | Filter | Action |
|---|---|---|---|
| EPG-A | EPG-B | contract-1 | redirect |
| EPG-B | EPG-A | contract-1-r | redirect |
| EPG-3 | EPG-1 | contract-1 | permit |
| EPG-1 | EPG-2 | contract-1-r | permit |
| EPG-3 | EPG-2 | contract-2 | permit |
| EPG-2 | EPG-3 | Contract-2-r | permit |

Rules for the Excluded EPGs added by Preferred Group (priority 18)

| Source | Destination | Filter | Action |
|---|---|---|---|
| EPG-1 | any | implicit | deny |
| any | EPG-1 | implicit | deny |
| Etc… | | | |

Rule for the Preferred Group EPGs (priority 20)

| Source | Destination | Filter | Action |
|---|---|---|---|
| any | any | implicit | permit |

# Example 10 : alternative to BiDir Contract

- We saw in earlier example. Assuming we want to allow one port dest (aka port 80 to EGP Web)
- Option 1 : Bidir  subject contract provided by Web consumed by client → 2 entry
- Option 2 : Unidir subject, one with src port, one with dst port and two contract → 2 entry as well but more complex

What is we have 100 EPG client that needs to talk port 80 to the same Web EPG !!! Both options leads to 200 lines

# Bidirectional Contract/Filters

- When a policy is configured for bidirectional filters two entries are created for each specific filter one for each direction
- When the bidirectional attribute is not used a single entry should be created for the entire context allowing ‘established’ connections (check for ACK bit)

## Policy TCAM

| Scope | SRC_EPG | DST_EPG | L2/L3 Protocol | SRC_L4_Port | Dst_L4_Port or APP_CAM Index |
|---|---|---|---|---|---|
| Tenant_1 | WEB | APP | IP | * (Any) | 80 |
| Tenant_1 | APP | WEB | IP | 80 | * (Any) |
| Tenant_1 | WEB | APP | IP | * (Any) | 443 |
| Tenant_1 | APP | WEB | IP | 443 | * (Any) |
| Tenant_2 | Any | Any | IP | * (Any) - Established | * (Any) |
| Tenant_2 | APP | WEB | IP | * (Any) | 80 |
| Tenant_2 | APP | WEB | IP | * (Any) | 443 |

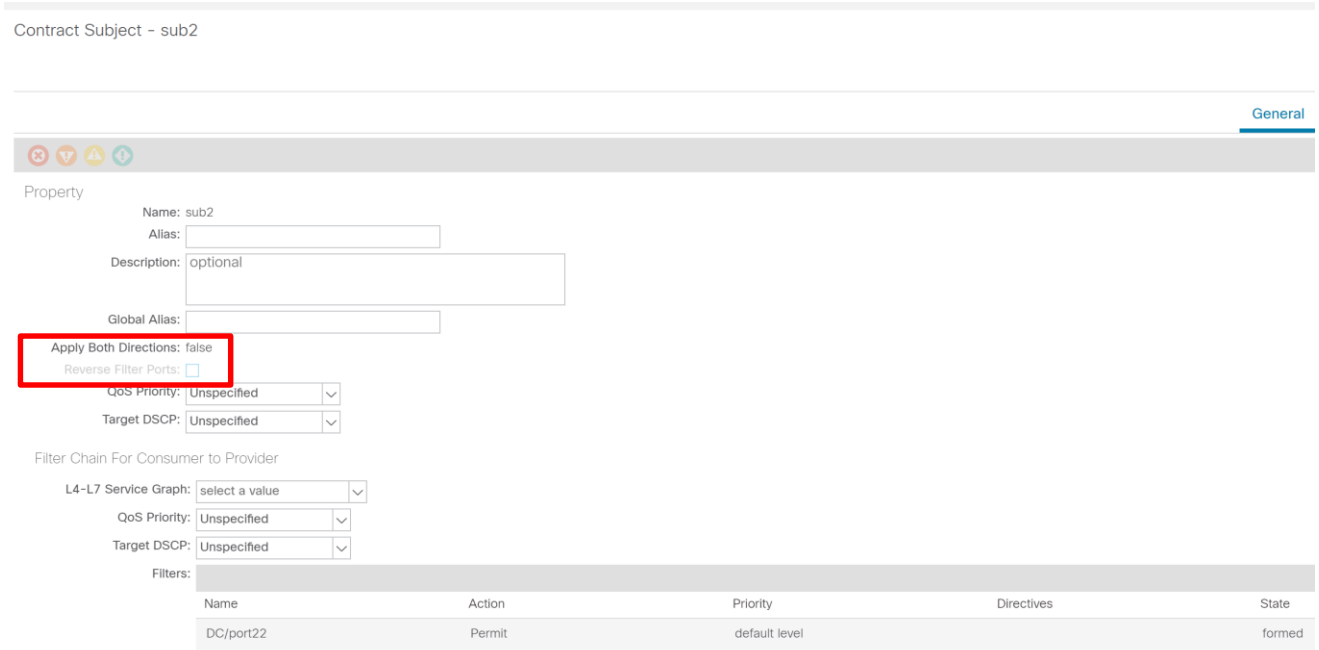
**Tenant 1 has configured Bidirectional Filter**

**Symmetric TCAM Entries are Installed**

**Tenant 2 has ‘not’ configured Bidirectional Filter**

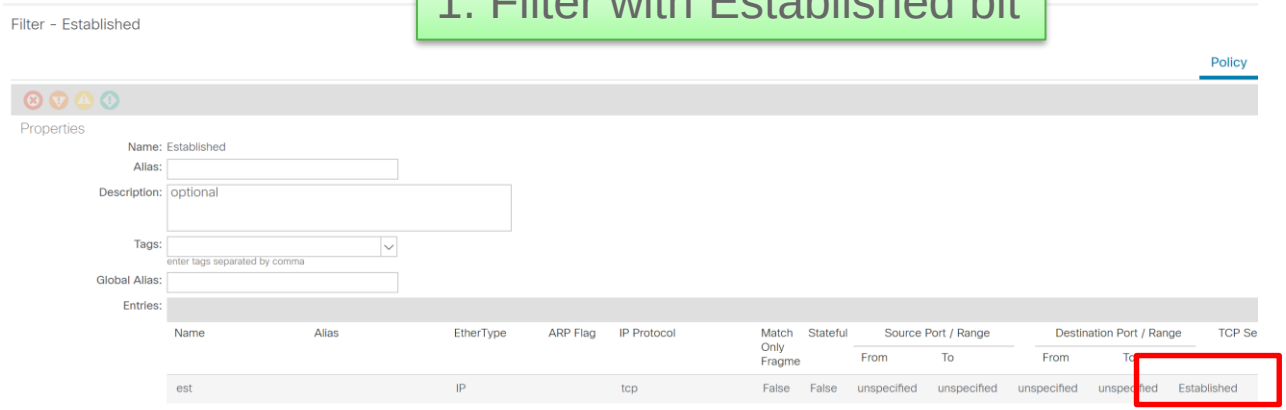
**A single VRF scoped allow any-any established entry is recommended This allow the return traffic**

# Example : EPG to EPG contract unidir to Dest port 22

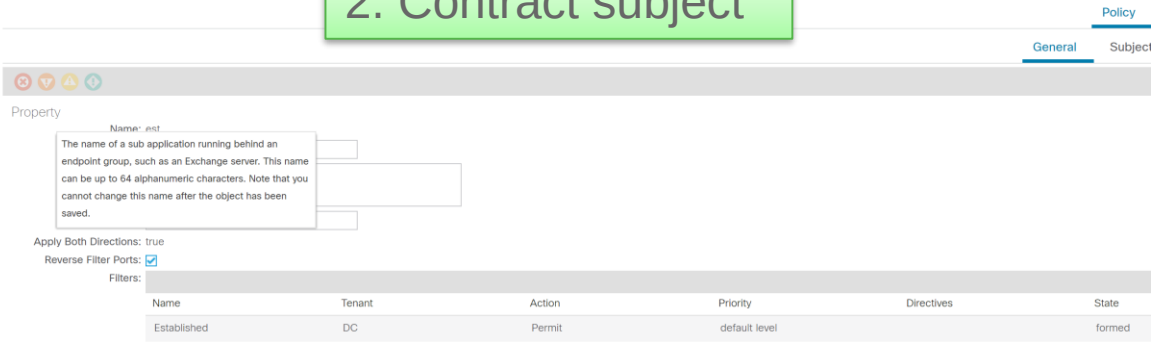


# Any to Any established config

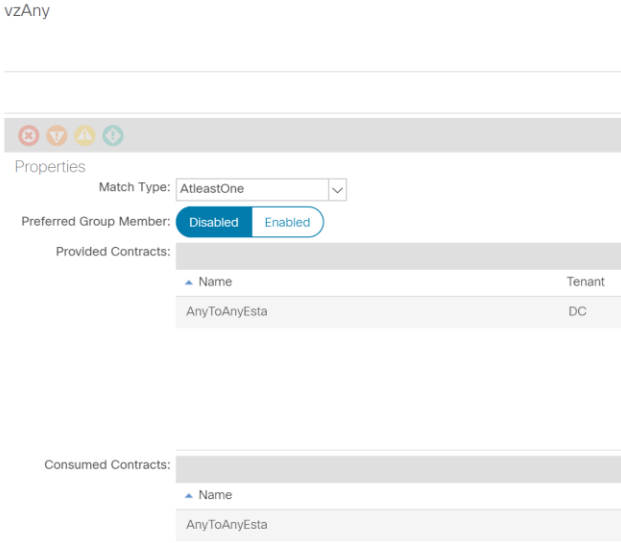
1. Filter with Established bit




2. Contract subject




3. vzAny consume /provide



# Zoning-rule with the established trick

- Epg 32775 can talk to epg 16387 on port 22 but there is no return rule from 16387 to 32775.
- Communication is allowed per rule 4435 any to any with filter allowing Established TCP flows

```
bdsol-aci32-leaf3#  show zoning-rule scope 3014656

+---------+--------+--------+----------+-----------------+---------+---------+---------------+------------+----------------------+
| Rule ID | SrcEPG | DstEPG | FilterID |      Dir        |  operSt |  Scope  |      Name     |   Action   |       Priority       |
+---------+--------+--------+----------+-----------------+---------+---------+---------------+------------+----------------------+
|   4437  | 32775  | 16387  |    14    |    uni-dir      | enabled | 3014656 | CT1-deny-test |   permit   |    fully_qual(7)     |
|   4435  |   0    |   0    |    1     |    uni-dir      | enabled | 3014656 |  AnyToAnyEsta |   permit   |  any_any_filter(17)  |
bdsol-aci32-leaf3# show zoning-filter  filter 1
+----------+------+--------+------------+------+-------------+----------+------------+------------+------------+------------+-------+------------+------------+----------+
| FilterId | Name | EtherT |    ArpOpc  | Prot | ApplyToFrag | Stateful |  SFromPort |   SToPort  | DFromPort  |  DToPort   |  Prio |  Icmpv4T   |  Icmpv6T   | TcpRules |
+----------+------+--------+------------+------+-------------+----------+------------+------------+------------+------------+-------+------------+------------+----------+
|    1     | 1_0  |   ip   | unspecified | tcp |      no     |    no    | unspecified | unspecified | unspecified | unspecified | flags | unspecified | unspecified |   est    |
|    14    | 14_0 |   ip   | unspecified | tcp |      no     |    no    | unspecified | unspecified |     22      |     22      | dport | unspecified | unspecified |          |
+----------+------+--------+------------+------+-------------+----------+------------+------------+------------+------------+-------+------------+------------+----------+
bdsol-aci32-leaf3#
```

# Example 10 (cont).

- With the established approach in vzAny we have
- One line from each Client EPG to Web port 80 (100 lines)
- One line per VRF for vzAny to vzAny established
- → total 101 lines instead of 200 ☺

- The Established flag is designed to allow TCP traffic of existing connections:  ACK or RST

# Example 11 – Taboo Contract

- A taboo contract is applied to a single EPG.
- There are no consumer or provider
- The taboo contract will prevent any traffic matching the filter in Taboo to enter the EPG
- Next slides shows a taboo for port tcp 24 applied to an epg

# Example 11 - Taboo zoning-rule

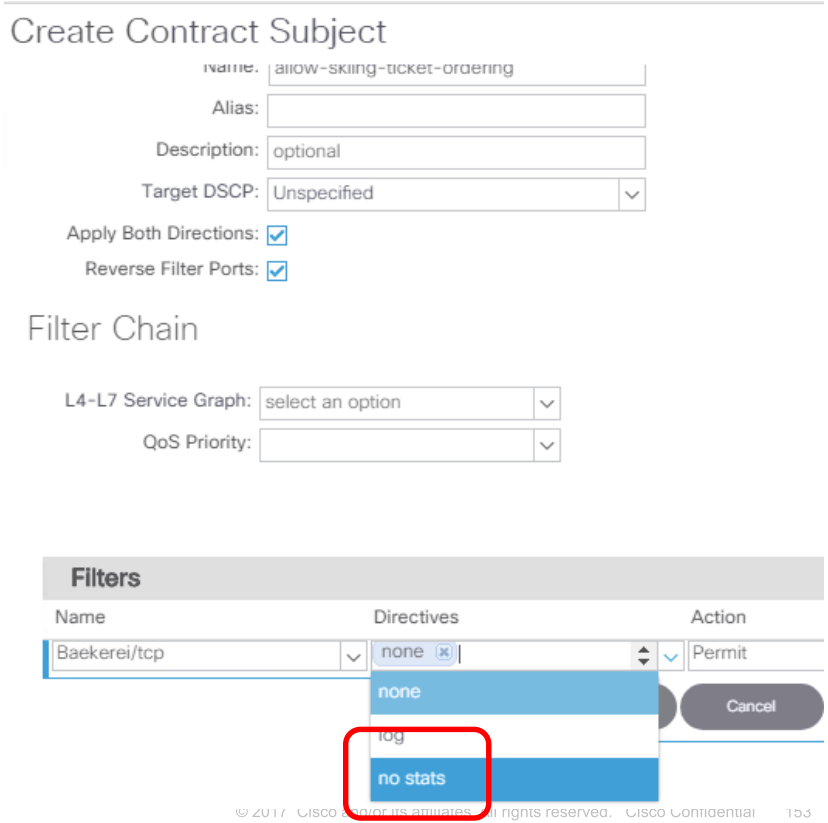
```
bdsol-aci32-leaf1# show zoning-rule scope 3014656 dst-epg 32775
+---------+--------+--------+----------+----------------+---------+---------+---------------+--------+---------------+
| Rule ID | SrcEPG | DstEPG | FilterID |      Dir       |  operSt |  Scope  |      Name     | Action |    Priority   |
+---------+--------+--------+----------+----------------+---------+---------+---------------+--------+---------------+
|   4210  |   0    | 32775  |    93    |    uni-dir     | enabled | 3014656 |               |  deny  | black_list(5) |
+---------+--------+--------+----------+----------------+---------+---------+---------------+--------+---------------+
bdsol-aci32-leaf1# show zoning-filter filter 93
+----------+------+--------+-------------+------+-------------+----------+-------------+-------------+-----------+---------+-------+-------------+-------------+
| FilterId | Name | EtherT |    ArpOpc   | Prot | ApplyToFrag | Stateful |  SFromPort  |   SToPort   | DFromPort | DToPort |  Prio |   Icmpv4T   |   Icmpv6T   |
+----------+------+--------+-------------+------+-------------+----------+-------------+-------------+-----------+---------+-------+-------------+-------------+
|    93    | 93_0 |   ip   | unspecified | tcp  |      no     |    no    | unspecified | unspecified |     24    |    24   | dport | unspecified | unspecified |
+----------+------+--------+-------------+------+-------------+----------+-------------+-------------+-----------+---------+-------+-------------+-------------+
bdsol-aci32-leaf1#
```

The rule is Any to EPG with Priority 5 (black-list)
Higher than any epg to epg fully qualified rule (7)

# Policy Compression

# ACI 3.2 introduced the first optimization for the policy-cam

- With ACI 3.2 it is possible to program -EX leafs and newer in such a way that bidirectional subjects take one entry only in the policy-cam
- This option in ACI 3.2 is called "no stats„
- Option was renamed policy-compression in 4.x software.

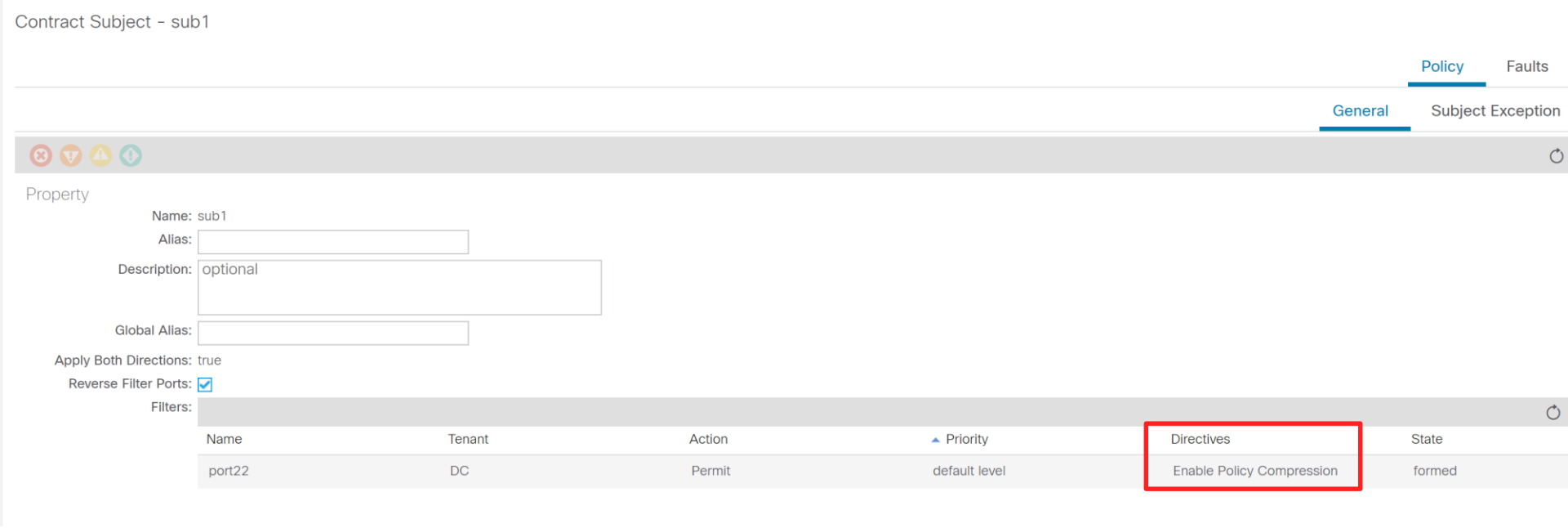


# Criteria for Bi-directional TCAM Compression (Rel 3.2)

- Only the contracts which follow the below guidelines will become candidates of bi-directional compression.
  - Contract->subject->apply-both-direction
  - Contract->subject->reverse-filter-ports
  - Contract->subject->filter-group->no-stats
  - Fully qualified Rules
  - Action: permit or permit+log

- Note: The directive “no-stats” in 3.2 was renamed to “Enable Policy Compression” in Rel 4.0.

# GUI no stats- policy compression config



# 3.2 – no stats - policy compression

```
bdsol-aci32-leaf3# show zoning-rule scope 3014656  | egrep 32775
|   4441  | 32775  | 16387  |    14    |     bi-dir     | enabled | 3014656 | CT1-deny-test | no_stats,permit |    fully_qual(7)     |
|   4354  | 16387  | 32775  |    15    | uni-dir-ignore | enabled | 3014656 | CT1-deny-test | no_stats,permit |    fully_qual(7)     |
bdsol-aci32-leaf3# show system internal policy-mgr stats | egrep "4441|4354"
Rule (4354) DN (sys/actrl/scope-3014656/rule-3014656-s-16387-d-32775-f-15) Ingress: 0, Egress: 0, Pkts: 0  RevPkts: 0
Rule (4441) DN (sys/actrl/scope-3014656/rule-3014656-s-32775-d-16387-f-14) Ingress: 0, Egress: 0, Pkts: 0  RevPkts: 0
bdsol-aci32-leaf3#
bdsol-aci32-leaf3#
bdsol-aci32-leaf3# vsh_lc
module-1# show system internal aclqos zoning-rules 4441

 ASIC type is Sug

========================================
Rule ID: 4441 Scope 11 Src EPG: 32775 Dst EPG: 16387 Filter 14
  Bidir compressed
    unit_id: 0
    === Region priority: 1923 (rule prio: 7 entry: 131)===
        sw_index = 137 | hw_index = 47

  Curr TCAM resource:
  ============================
    === SDK Info ===
        Result/Stats Idx: 81873
        68
        Tcam Total Entries: 1
        HW Stats: 0
module-1# show system internal aclqos zoning-rules 4354

 ASIC type is Sug
Rule ID: 4354 does not exist
module-1#
```

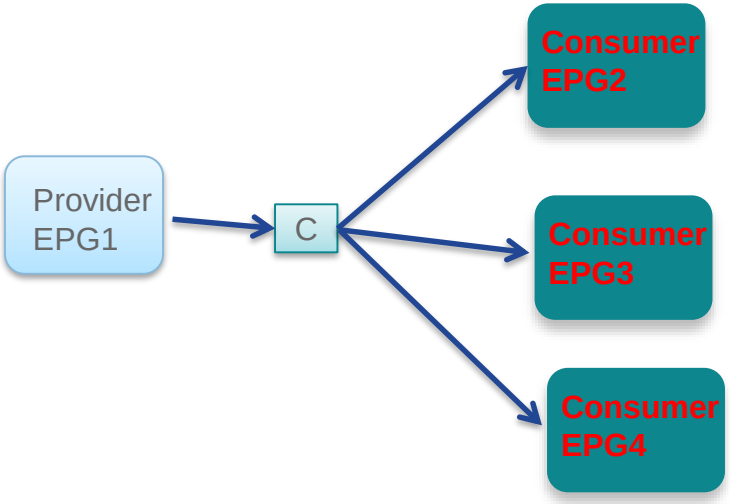
Though there are 2 lines
In zoning-rule they are combined to
One in aclqos in lc at the cost of
Disabling the stats

# Caveat

- CSCvq80820  - Compressed stateful filters drop TCP SYN
  - Only if the compressed rule as the stateful option configured

# 4.0 : Contract reuse

- New level of indirection is introduced in FX based leaf to address contract scale for many-to-one contract relationship(many consumer EPG access provider EPG via same contract)
- Not supported in EX or earlier (EX Leaf support only bi-directional compression)



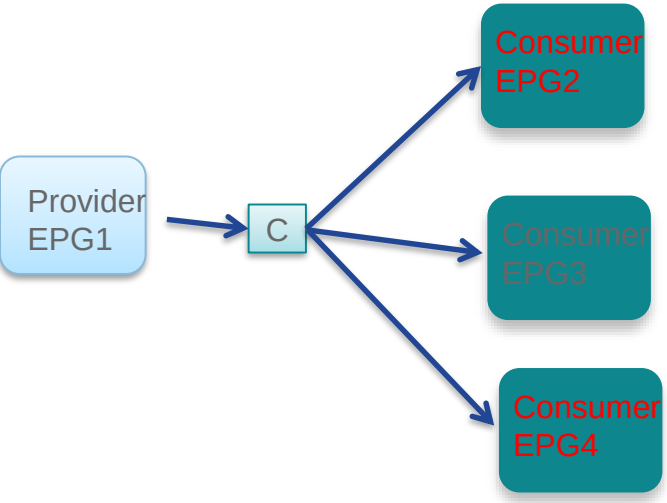

NS/Donner/Sugarbowl Implementation

| sclass | dclass | protocol | sport | dport |
|---|---|---|---|---|
| EPG1 | EPG2 | tcp | * | 443 |
| EPG1 | EPG2 | tcp | * | 80 |
| EPG1 | EPG3 | tcp | * | 443 |
| EPG1 | EPG3 | tcp | * | 80 |
| EPG1 | EPG4 | tcp | * | 443 |
| EPG1 | EPG4 | tcp | * | 80 |

One copy of contract rules per (sclass, dclass) combination

# 4.0 : contract reused – how to compress

- All provider-consumer EPG pair refer to same set of rules in the policy CAM
- **No statistics for these compressed rules**
- This built-in HW support allows improving zonerule scale.




1st Stage: Policy Group Label Lookup

| sclass | dclass | PG-Label |
|---|---|---|
| EPG1 | EPG2 | 10 |
| EPG1 | EPG3 | 10 |
| EPG1 | EPG4 | 10 |

2nd Stage: Policy Hash TCAM Lookup

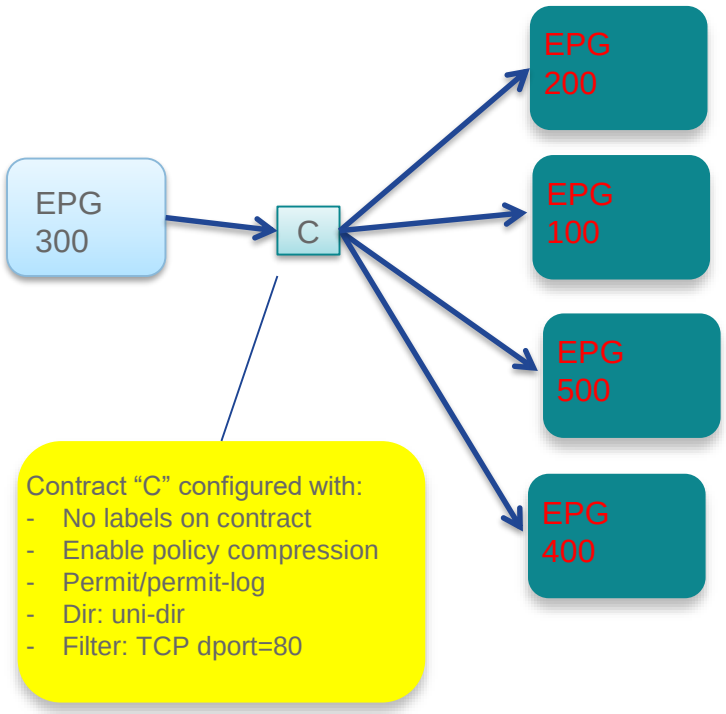
| PG-Label | Protocol | sport | dport |
|---|---|---|---|
| 10 | tcp | * | 443 |
| 10 | tcp | * | 80 |
| | | | |
| | | | |

One copy of contract rules for all consumer EPG

# Criteria for Policy TCAM Compression

- Contract should not have any labels configured on them.
- “Enable Policy Compression” directive
- Fully qualified Rules
- Action: permit or permit+log

# Rule compression illustration (PT-Indirection)




1st Stage: Policy Group Label Lookup

| class0 | class1 | PG-Label |
|---|---|---|
| 100 | 300 | 10 |
| 200 | 300 | 10 |
| 300 | 400 | 20 |
| 300 | 500 | 20 |

2nd Stage: Hash TCAM Lookup

| PG-Label | Protocol | sport | dport | Dir |
|---|---|---|---|---|
| 10 | tcp | * | 80 | 0 |
| 20 | tcp | 80 | * | 1 |

# Policy Group Label Lookup

- PG Label lookup is direction agnostic, i.e
  - sclass=a & dclass=b &
  - sclass=b & dclass=a will derive the same PG label.
- Hence grouping for contract between multiple EPG pairs is possible (first group for forward direction say with label ‘A’ and second group for reverse direction with same label ‘A’)

# Feature Internals

- Space for PG label lookup Table (TCAM stage 1 look up) is repurposed dynamically from existing Policy TCAM space. In default profile there is a cap of max 40k entries for this table.
- If compression eligible rules are present, PG label lookup table is carved out and will be in use till compression rules are not deleted. Once those rules are deleted the space can be reused for regular TCAM entries.
- If PG label lookup Table space runs out and there is space in regular TCAM those rules will be programmed in uncompressed form.
- PG label Table space utilization available in capacity dashboard.
- No need to change leaf profile to enable this feature.
- There are certain scenarios where policy compression won’t save any space. For example: only 1 consumer EPG and 1 Provider EPG per contract.

# Example – policy compression not enable

```
pod11-leaf1# show zoning-rule scope 2654210 | egrep 16386
|   4112  | 16387  | 16386  |    15    |     bi-dir     | enabled | 2654210 | testCompress |  permit  |    fully_qual(7)     |
|   4111  | 16386  | 16387  |    16    | uni-dir-ignore | enabled | 2654210 | testCompress |  permit  |    fully_qual(7)     |
|   4184  | 16386  | 16388  |    16    | uni-dir-ignore | enabled | 2654210 | testCompress |  permit  |    fully_qual(7)     |
|   4177  | 16388  | 16386  |    15    |     bi-dir     | enabled | 2654210 | testCompress |  permit  |    fully_qual(7)     |
|   4183  | 49153  | 16386  |    15    |     bi-dir     | enabled | 2654210 | testCompress |  permit  |    fully_qual(7)     |
|   4190  | 16386  | 49153  |    16    | uni-dir-ignore | enabled
```

Policy CAM

Rules: <1% 118 of 65536

Labels: 0% 0 of 0

- 4 EPG
- 1 Provides a contract
- 3 other consume that contract
- No label consumed all rules in aclqos
- See stats index different for each

```
module-1# show system internal  aclqos zoning-rules 4112

 ASIC type is Hom

========================================
Rule ID: 4112 Scope 12 Src EPG: 16387 Dst EPG: 16386 Filter 15
    unit_id: 0
    === Region priority: 1923 (rule prio: 7 entry: 131)===
        sw_index = 111 | hw_index = 47

  Curr TCAM resource:
  ============================
    === SDK Info ===
        Result/Stats Idx: 81873
        70
        Tcam Total Entries: 1
        HW Stats: 0
module-1# show system internal  aclqos zoning-rules 4111

 ASIC type is Hom

========================================
Rule ID: 4111 Scope 12 Src EPG: 16386 Dst EPG: 16387 Filter 16
    unit_id: 0
    === Region priority: 1924 (rule prio: 7 entry: 132)===
        sw_index = 112 | hw_index = 52

  Curr TCAM resource:
  ============================
    === SDK Info ===
        Result/Stats Idx: 81868
        75
        Tcam Total Entries: 1
        HW Stats: 0
```

# Policy compression enabled

```
pod11-leaf1# show zoning-rule scope 2654210 | egrep 16386
|   4177  | 16388  | 16386  |    15    |     bi-dir      | enabled | 2654210 | testCompress | no_stats,permit |     full
|   4183  | 16387  | 16386  |    15    |     bi-dir      | enabled | 2654210 | testCompress | no_stats,permit |     full
|   4184  | 49153  | 16386  |    15    |     bi-dir      | enabled | 2654210 | testCompress | no_stats,permit |     full
|   4111  | 16386  | 16387  |    16    | uni-dir-ignore | enabled | 2654210 | testCompress | no_stats,permit |     full
|   4112  | 16386  | 16388  |    16    | uni-dir-ignore | enabled | 2654210 | testCompress | no_stats,permit |     full
|   4190  | 16386  | 49153  |    16    | uni-dir-ignore | enabled | 2654210 | testCompress | no_stats,permit |     full
```

Policy CAM

Rules: <1% 113 of 55296

Labels: <1% 3 of 40960

```
module-1# show system internal  aclqos zoning-rules 4177
 ASIC type is Hom

============================================
Rule ID: 4177 Scope 12 Src EPG: 16388 Dst EPG: 16386 Filter 15
  Bidir compressed
Contract ID: 2
Sub Rule Id: 262146, Region Id: 1987
Rule compressed status: True     PG Label : 131074

  Curr TCAM resource:
  ============================
    === SDK Info ===
        Result/Stats Idx: 81856
        83
        Tcam Total Entries: 1
        HW Stats: 0
module-1# show system internal  aclqos zoning-rules 4183
 ASIC type is Hom

============================================
Rule ID: 4183 Scope 12 Src EPG: 16387 Dst EPG: 16386 Filter 15
  Bidir compressed
Contract ID: 2
Sub Rule Id: 262146, Region Id: 1987
Rule compressed status: True     PG Label : 131074

  Curr TCAM resource:
  ============================
    === SDK Info ===
        Result/Stats Idx: 81856
        83
        Tcam Total Entries: 1
        HW Stats: 0
```

```
Rule ID: 4184 Scope 12 Src EPG: 49153 Dst EPG: 16386 Filter 15
  Bidir compressed
Contract ID: 2
Sub Rule Id: 262146, Region Id: 1987
Rule compressed status: True     PG Label : 131074

  Curr TCAM resource:
  ============================
    === SDK Info ===
        Result/Stats Idx: 81856
        83
        Tcam Total Entries: 1
        HW Stats: 0

module-1# show system internal  aclqos zoning-rules 4111
 ASIC type is Hom
Rule ID: 4111 does not exist

module-1# show system internal  aclqos zoning-rules 4112
 ASIC type is Hom
Rule ID: 4112 does not exist

module-1# show system internal  aclqos zoning-rules 4190
 ASIC type is Hom
Rule ID: 4190 does not exist
```

**Uni Dir rule are not created (per no stats Feature of 3.2)**
**However PG label and Rule compressed Are new in 4.x**

# Tcam table pointer based on Rule ID, subRule ID and Region Id

```
module-1# show system internal aclqos zoning-rules tcam-tbl-name rule-id 4177 subrule-id 262146 regionid 1987

 ===== Rule Id 4177 Sub Rule Id 262146 Region Id 1987 =====

    show plat int hom table roc_hom_fpc_pttcam 6164 1 sl 0
    show plat int hom table roc_hom_fpc_pttcamdata 83 1 sl 0

        Tcam Total Entries: 1
==== Policy Group Table ====
         Entry : 0
         show platform internal hom table roc_hom_fpx_fptile 244 fp 11 tile 0 | grep policy_grp
module-1# show system internal aclqos zoning-rules tcam-tbl-name rule-id 4183 subrule-id 262146 regionid 1987

 ===== Rule Id 4183 Sub Rule Id 262146 Region Id 1987 =====

    show plat int hom table roc_hom_fpc_pttcam 6164 1 sl 0
    show plat int hom table roc_hom_fpc_pttcamdata 83 1 sl 0

        Tcam Total Entries: 1
==== Policy Group Table ====
         Entry : 0
         show platform internal hom table roc_hom_fpx_fptile 1010 fp 11 tile 0 | grep policy_grp
module-1# show system internal aclqos zoning-rules tcam-tbl-name rule-id 4184 subrule-id 262146 regionid 1987

 ===== Rule Id 4184 Sub Rule Id 262146 Region Id 1987 =====

    show plat int hom table roc_hom_fpc_pttcam 6164 1 sl 0
    show plat int hom table roc_hom_fpc_pttcamdata 83 1 sl 0

        Tcam Total Entries: 1
==== Policy Group Table ====
         Entry : 0
         show platform internal hom table roc_hom_fpx_fptile 487 fp 11 tile 0 | grep policy_grp
```

All 3 points to the same fpc table

# PG table lookup (stage 1)

```
module-1#          show platform internal hom table roc_hom_fpx_fptile 244 fp 11 tile 0 | grep policy_grp
                tile_entry_policy_grp_entry_0_vld=0x1
                tile_entry_policy_grp_entry_0_sg_label=0xc
                tile_entry_policy_grp_entry_0_class0=0x4002
                tile_entry_policy_grp_entry_0_hash_rslt_msb3=0x2
                tile_entry_policy_grp_entry_0_label=0x20002
module-1#          show platform internal hom table roc_hom_fpx_fptile 1010 fp 11 tile 0 | grep policy_grp
                tile_entry_policy_grp_entry_0_vld=0x1
                tile_entry_policy_grp_entry_0_sg_label=0xc
                tile_entry_policy_grp_entry_0_class0=0x4002
                tile_entry_policy_grp_entry_0_class1=0x3
                tile_entry_policy_grp_entry_0_hash_rslt_msb3=0x3
                tile_entry_policy_grp_entry_0_label=0x20002
module-1#          show platform internal hom table roc_hom_fpx_fptile 487 fp 11 tile 0 | grep policy_grp
                tile_entry_policy_grp_entry_0_vld=0x1
                tile_entry_policy_grp_entry_0_sg_label=0xc
                tile_entry_policy_grp_entry_0_class0=0x4002
                tile_entry_policy_grp_entry_0_class1=0x1
                tile_entry_policy_grp_entry_0_hash_rslt_msb3=0x5
                tile_entry_policy_grp_entry_0_label=0x20002
```

**Sg_label is vrf id in Hw (here 12  = 0xc)**
**Label1 store de pg label : 0x20002 = 131074**
**Class0 0x4002 = 16386 is the provider EPG pcTAg**

# TCAM table lookup (stage 2)

```
module-1#     show plat int hom table roc_hom_fpc_pttcam 6164 1 sl 0 | egrep "pt_tcam_key_i|class_dir"
ENTRY[006164] = pt_tcam_key_ifabric_misc0_sg_label=0xc
                pt_tcam_key_ifabric_port0=0x50
                pt_tcam_key_ifabric_protocol=0x6
                pt_tcam_key_ifabric_pg_label=0x20002
                pt_tcam_key_ifabric_pg_label_vld=0x1
                pt_tcam_key_mask_ifabric_class_dir=0x1
                pt_tcam_key_mask_dci_sclass_dir=0x1
```

Class_dir = 1 means wildcard (do not care direction)
Proto 0x6 for TCP
Port0 = 0x50 → 80 for http
…

```
module-1# show plat int hom table roc_hom_fpc_pttcamdata 83 1 sl 0 all | egrep deny
                bndl_pt_all_pt_data_service_info_copy_info_set_idx=0x0     bndl_pt_all_pt_data_deny=0x0
module-1#
```

Denys is not set to 0x1 so it is a permit

# Policy compression enabled

```
module-1# show platform internal hal  policy indirection  vrf all
==========================================================================================================================
                                                                                Seg Label : 12
==========================================================================================================================

--------------------------------------------------------------------------------------------------------------------------
                                                                          EPG Map - Size:  : 3
--------------------------------------------------------------------------------------------------------------------------
| Key          | Hash | FP   | Tile | Entry | MSB3 | PGID   | VLD | OCID     | NCID     | CIDs
--------------------------------------------------------------------------------------------------------------------------
| 16388-16386 | 244  | 11   | 0    | 0     | 2    | 131074 | 1   | -1       | 2        | 2
| 16387-16386 | 1010 | 11   | 0    | 0     | 3    | 131074 | 1   | -1       | 2        | 2
| 49153-16386 | 487  | 11   | 0    | 0     | 5    | 131074 | 1   | -1       | 2        | 2
--------------------------------------------------------------------------------------------------------------------------
                                                    Contract Map - Size:  1
--------------------------------------------------------------------------------------------------------------------------
Key         FCount       RCount
2           1             3
--------------------------------------------------------------------------------------------------------------------------
                                                  Contract Filter Map - Size:  1
--------------------------------------------------------------------------------------------------------------------------
Key             RSet
    15-1       [1987,4183,262146] [1987,4177,262146] [1987,4184,262146]
--------------------------------------------------------------------------------------------------------------------------
                                                  Contract EPG Map - Size:  3
--------------------------------------------------------------------------------------------------------------------------
Key            RSet
 16388-16386  [1987,4177,262146]
 16387-16386  [1987,4183,262146]
 49153-16386  [1987,4184,262146]
--------------------------------------------------------------------------------------------------------------------------
Uncompressed rule set size - 0
Uncompressed rule set - []
--------------------------------------------------------------------------------------------------------------------------
```

Key 16388-16388 is epg pair
Each pair use same PGID
Rset Contains the BiDir Rule ID

Last part shows mapping of epg pair
With Rset
1987 and 262146 are region id and subrule id

# Policy-Group Table and Policy-Cam Tables *How many entries?*

- If a contract is re-used,
- the entries that don't have "compression" configured are programmed in the policy-cam as usual: source class-id, destination class-id, filters etc...
- The filters with the "compression" option configured are programmed in the policy-cam table with a label

- If there is at least one EPG pair with compression
- The policy-group table is carved from the main policy-cam table, whose size is less than the initial one.
- But the aggregated capacity is much bigger due to the fact that this policy-cam may be used just for filters which can be re-used multiple times.
- As an example for 40k Policy-Group EPG pairs, this requires 10k entries from the existing policy-cam table, which will change size from 64k to 64k-10k = 54k.
- For 80k Policy-Group EPG pairs, this requires 20k entries from the policy-cam table. The size of the policy-cam table changes from 128k to 128k-20k = 108k

# Policy-cam scale with Filter Re-use

- The Aggregate Capacity depends on the amount of filter re-use and on the use of bidirectional subjects.
- In the worst case each EPG pair points to a different label which may consist of one or more filters.
- For a policy-cam with the default profile that means that a 64k policy-cam can provide the equivalent of:
  - 40k (EPG pairs with label indirection with 1 filter) + 14k (EPG pairs without label indirection) = 54k x 2 (bidirectional)
  - But if there's no reuse no policy-group table is allocated, hence
  - 64k x 2 = 128k
  - Up to:
  - 40k (EPG pairs with label indirection to 54k filters) = 2.160 x 10^9 x 2 entries

| FX ToR | | | |
|---|---|---|---|
| Profile | PG Table (EPG pairs only) | Policy Cam (for filters and/or EPG pairs + filters) | Aggregated Capacity (it depends on filter re-use) |
| Default | 40k | 64k-10k = 54k | 128k - 40k x 54k x 2 |
| IPv4 Scale | 40k | 64k-10k = 54k | 128k - 40k x 54k x 2 |
| HDS | 80k | 128k-20k = 108k | 160k – 80k x 108k x 2 |
| High LPM | Feature not available | 8k | 8k |

# Each EPG pair can only have one contract optimized

- The Policy-Group table contains EPG pairs and the label that points to the policy-cam table.
- The EPG pair can be associated with only one label
- Hence only one contract can be compressed per EPG pair
- Example:
- Contracts: WEB, MGMT, SSH
- EPGmgmt---MGMT---EPG2, EPG3, EPG4, EPG5
- EPGclient---WEB---EPG2, EPG3, EPG4, EPG5

- With Compression:
  - EPGmgmt to EPG2
  - EPGmgmt to EPG3
  - Etc…
  - All share the same filters from contract MGMT
  - EPGclient to EPG2
  - EPGclient to EPG3
  - Etc…
  - All share the same filters from contract WEB
- Now if I add a contract
  - EPGclient---SSH---EPG3, EPG5
- Can it be also compressed?
- No because there is already compression for theEPGclient EPG3 and for EPGclient EPG5 with contract WEB

# Policy Compression with –FX leafs
# Policy Group label lookup and Policy CAM

- On FX leafs it is possible to program the EPG pairs with a pointer to a set of filter entries.
- The Poligy Group label table is where the EPG pairs are programmed
- This spares policy-cam space because the same filter entries can be re-used.
- At the light of this pls keep into account that the "compression" feature doesn't help with the number of EPG pairs but it helps in case the configuration consists of complex filters.
- The space for PG label lookup Table is repurposed dynamically from existing Policy TCAM space.
- If compression eligible rules are present, the PG label lookup table is carved out and will be in use till compression rules are not deleted. Once those rules are deleted the space can be reused for regular TCAM entries.
- If PG label lookup Table space runs out and there is space in regular TCAM those rules will be programmed in uncompressed form.
- The PG label Table space utilization is available in capacity dashboard.
- There is no need to change leaf profile to enable this feature

# Limitations

- It is not possible to convert a pre-existent Subject: you must delete the subject and reconfigure it with the compression option.
- In an EPG pair only one contract can be compressed. The feature analyses all the contracts and select the one that gives the best saving.
- Enabling Policy Compression disables individual filter rule statistics.
- Policy Compression can be enabled for permit and permit-log rules only.
  - No Compression for Rules with the actions Deny , Redir , Copy or Deny-log
- Policy Compression can be enabled for user-defined rules only
- Policy Compression cannot be enabled on Contracts that have labels and subject exceptions associated with them.
- Policy Compression is not enabled for vzAny contracts