Simplified EDE VLAN Management

Don Fedyk (dfedyk@labn.net)

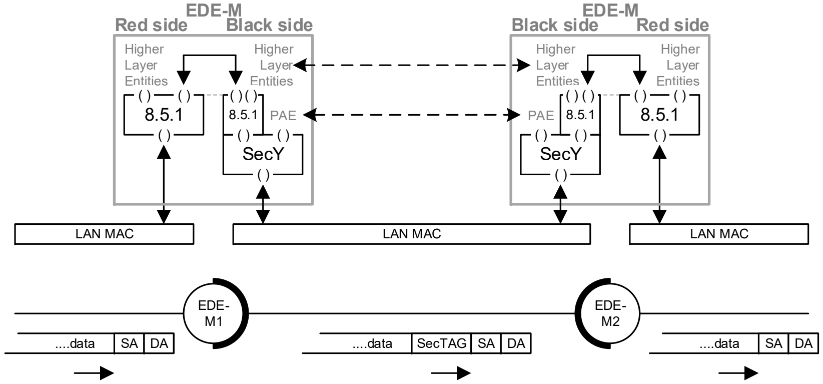
Forward

- This presentation is for a discussion on detailed config.
- It may contain errors/omission and should be consider a work in progress.
- An updated version the presentation will be posted after discussion to correct it but it will remain a work in progress.

Ethernet Data Encryption (EDE) devices

- EDE come in several types
- EDE-M VLAN unaware – handled by existing YANG models
- EDE-CS – Provider Bridge C-VLAN & S-VLAN like Components.
- EDE-CC – Two C-VLAN like components
- EDE-SS – Two S-VLAN like components

Ethernet Data Encryption



802.1AE-2018 Figure 15-2

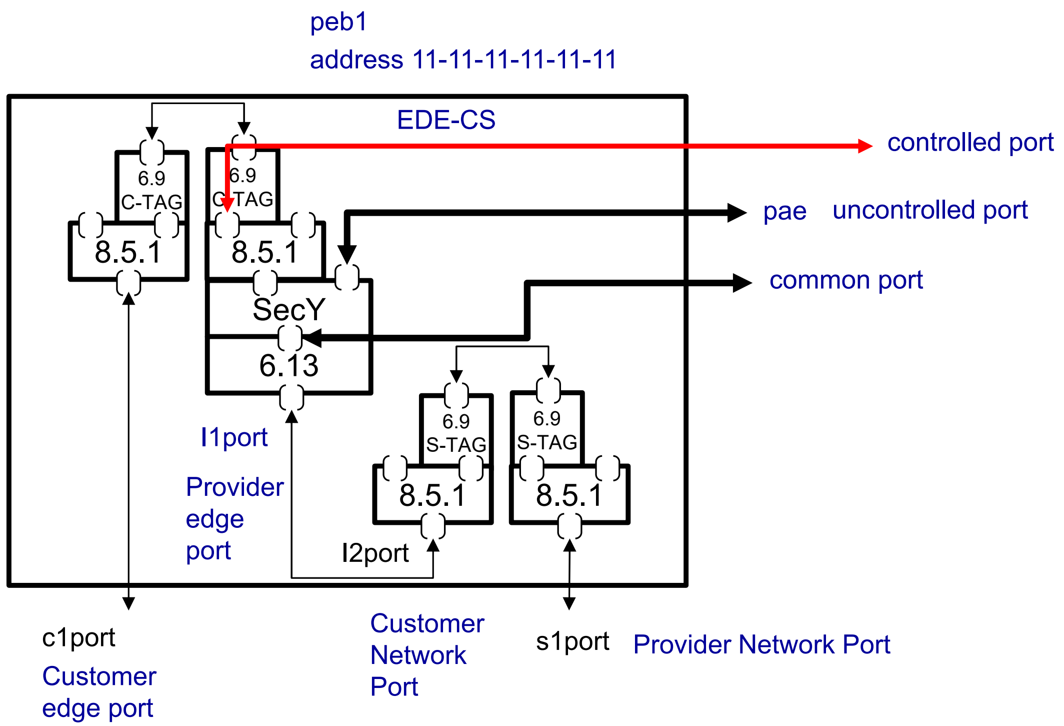
A model of configuring MACsec Shim on a bridge com

EDE-M The VLAN unaware device

Nothing to do here!

Revisiting MACsec Config for EDEs

The EDE is C and
S-VLAN aware
MACsec remains
A shim but the
combinations
Resulting in
Tagging
Of the data on
the wire.



YANG models all these ( ) ports

Last Meeting

- Discussed a prototype provision of what was needed from the Bridge and Provider YANG
- Got hung up on the Mapping of components – things mostly work for an EDE-CS but there gap extending to EDE-CC and EDE-CS
- As coded the C-Components and the S-Components have behavioral characteristics and you can’t just interchange them
- Plus it seemed the mapping of C-VID to S-VID was limited

See dk-fedyk-dot1aedk-privacy-config-0317-v00

The simplest model

Last time Approach was

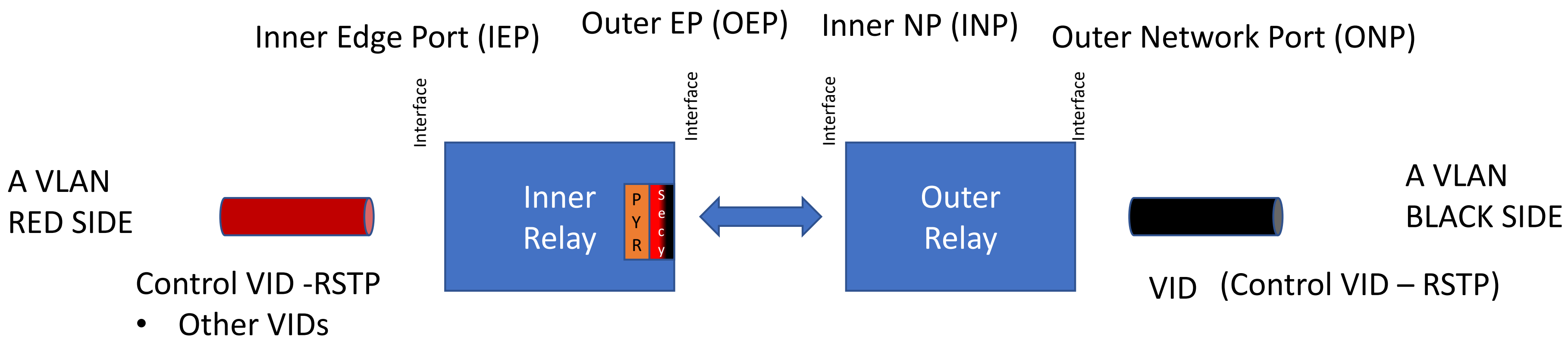
- Provision components that were needed for the Bridge and Provider Bridge model

This time start from basics:

  - Only allow functions that are needed
  - Note this is not a replacement. The goal of this exercise is to reuse existing components and build a structure that satisfies all the EDE models.
  - Then this delta could be added back to the Bridge and Provider bridge models

    Spoiler the delta is small!

EDEs Simple Bridge Relays –
What's Needed for Generic Tagging



Inner Edge Port (IEP)

Outer EP (OEP)

Inner NP (INP)

Outer Network Port (ONP)

Interface

Interface

Interface

Interface

A VLAN
RED SIDE

Inner
Relay

P
Y
R

S
e
c
y

Outer
Relay

A VLAN
BLACK SIDE

Control VID -RSTP

- Other VIDs

VID (Control VID – RSTP)

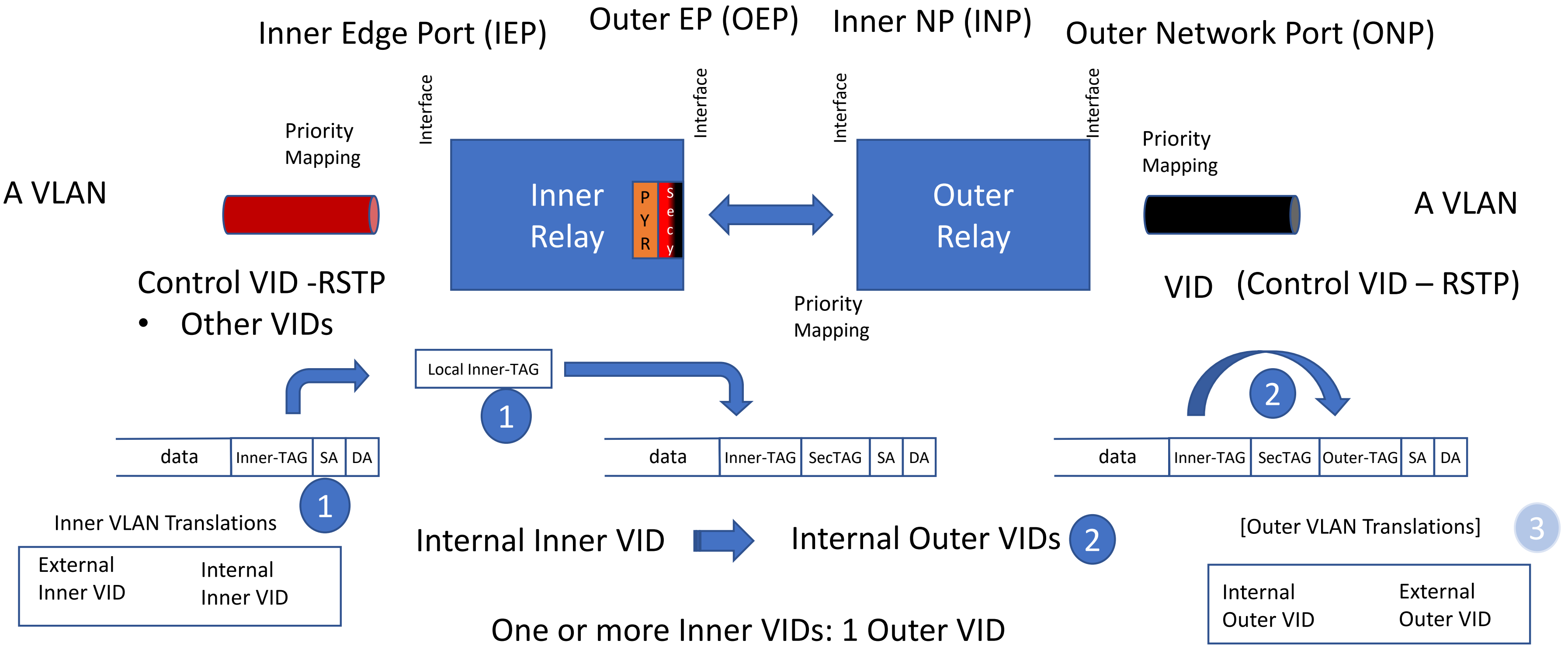
data	Inner-TAG	SA	DA

data	Inner-TAG	SecTAG	SA	DA

data	Inner-TAG	SecTAG	Outer-TAG	SA	DA

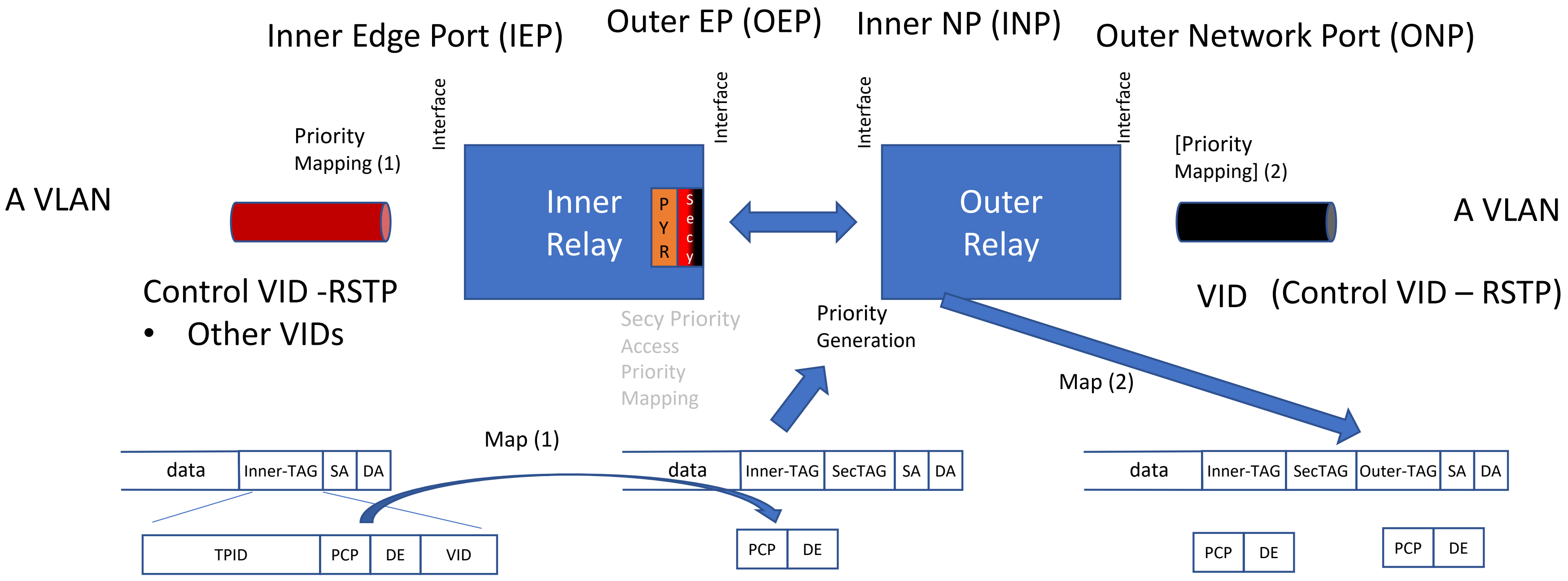
MACsec and MAC Privacy can exist on an interface typically where shown but not limited

EDEs Simple Bridge Relays – What's Needed for Generic Tagging



One or more Inner VIDs: 1 Outer VID

EDEs Simple Bridge Relays – What's Needed for Generic Priority



One or more Inner VIDs: 1 Outer VID

How many VID values in a single Tagged VLAN?

a) No more than 4094?

b) Varies - Statically Configured

c) Varies - Dynamically Control Plan Learned

d) All of the above

If you picked all of the above then you are right!

802.1 Bridges
Organized by
Bridge relays

Bridging uses FDB Filtering

- Forward all but …
- Filter all but …
- VID -> FID
- FID->Control Plane (VLAN)

Other groups models
Often Organized by
Interfaces/Ports

On Interfaces means many times we need mappings:

- All VIDs
- List of VIDs
- Ranges

Not Equivalent!

VID Mappings
VID bundling - grouping

- VID –> FID
- FID – Control plane or Static Config

This is the original VLAN Bridge model
This mapping only requires a single VLAN tag but works for the outer VLAN tag

- Inner VID –> Outer VID
- Outer VID -> Control Plane/Config
- Needs a Shared VLAN map to be equivalent
- And VID Filtering? It can be Implicit

This is very close but only works for
2 tags

IETF and many Vendors use Tagged config model

Breaking it down – What do we need?

Bridge Inner Edge Port

- Tag Type (component-type c-tag s-tag)
  - VID Translator / Interface
    - Untagged -> Primary VID (PVID)
    - Priority Tagged -> PVID
    - VID -> to other VID one-to-one Mapping (Ingress and egress)
      - Ease of input
  - Priority MAP / Interface
    - Input 8 Priority Code Points (PCPs) to 8 PCPs
    - Or Input 16 (8PCPs+2DE) to 16

Inner VID to Outer VID

1 Inner VID to oner or more Outer VID

Assume one Inner Bridge for now
Multiple Bridges ~ Virtual Interfaces

Interface VID Translator - Existing

```
list vid-translations {

       key "local-vid";
       description
         leaf local-vid {
         type dot1qtypes:vlanid;
         description
           "The Local VID after translation received at the ISS or
           EISS.";
         reference
           "12.10.1.8 of IEEE Std 802.1Q-2018
           6.9 of IEEE Std 802.1Q-2018";
       }
       leaf relay-vid {
         type dot1qtypes:vlanid;
         description
           "The Relay VID received before translation received at ISS
           or EISS.";
         reference
           "12.10.1.8 of IEEE Std 802.1Q-2018
           6.9 of IEEE Std 802.1Q-2018";
       }
     }
```

```
+--rw vid-translations* [local-vid]
      |  +--rw local-vid    dot1qtypes:vlanid
      |  +--rw relay-vid?   dot1qtypes:vlanid
      +--rw egress-vid-translations* [relay-vid]
         +--rw relay-vid    dot1qtypes:vlanid
         +--rw local-vid?   dot1qtypes:vlanid
```

Question can we should we use vid-ranges to ease input?

Currently VLANID is value – UP to 4090 individual entries YANG Ensures 1:1
VID ranges is a string – Values and correlation YANG cannot ensure 1:1 but ranges much more compact – Multiple mappings per line. Much better for input.
Currently YANG only ensure 1:1 mapping. It will only allow
X to Y where X is guaranteed to be unique but Y is not checked.
String would mean any mapping.
The point is both need backend handling and YANG checking even on value is minimal

Bridge Priority Map - Existing

```
+--rw port-type?                    identityref
+--rw pvid?                      dot1qtypes:vlan-index-type  +--rw default-priority?                dot1qtypes:priority-type
+--rw priority-regeneration
|  +--rw priority0?   priority-type
|  +--rw priority1?   priority-type
|  +--rw priority2?   priority-type
|  +--rw priority3?   priority-type
|  +--rw priority4?   priority-type
|  +--rw priority5?   priority-type
|  +--rw priority6?   priority-type
|  +--rw priority7?   priority-type
+--rw pcp-selection?                  dot1qtypes:pcp-selection-type
+--rw pcp-decoding-table
|  +--rw pcp-decoding-map* [pcp]
|     +--rw pcp           pcp-selection-type
|     +--rw priority-map* [priority-code-point]
|        +--rw priority-code-point    priority-type
|        +--rw priority?            priority-type
|        +--rw drop-eligible?       boolean
+--rw pcp-encoding-table
|  +--rw pcp-encoding-map* [pcp]
|     +--rw pcp           pcp-selection-type
|     +--rw priority-map* [priority dei]
|        +--rw priority             priority-type
|        +--rw dei                boolean
|        +--rw priority-code-point?   priority-type
+--rw use-dei?                        boolean
+--rw drop-encoding?                     boolean
+--rw service-access-priority-selection?    boolean
+--rw service-access-priority
|  +--rw priority0?   priority-type
|  +--rw priority1?   priority-type
|  +--rw priority2?   priority-type
|  +--rw priority3?   priority-type
|  +--rw priority4?   priority-type
|  +--rw priority5?   priority-type
|  +--rw priority6?   priority-type
|  +--rw priority7?   priority-type
+--rw traffic-class
|  +--rw traffic-class-map* [priority]
|     +--rw priority               priority-type
|     +--rw available-traffic-class* [num-traffic-class]
|        +--rw num-traffic-class    uint8
|        +--rw traffic-class?      traffic-class-type
```

Port-type = Customer Edge port, Provider Port Type, …
PVID – Primary VID
Priority
PCP
DE
Priority ….

Assume that we have a VLAN tag with PCP for the common case

Focus on Marking in and out PCP as primary case (the other modes are still there)

```
+--rw pcp-selection?                dot1qtypes:pcp-selection-type
   +--rw pcp-decoding-table
   |  +--rw pcp-decoding-map* [pcp]
   |     +--rw pcp            pcp-selection-type
   |     +--rw priority-map* [priority-code-point]
   |        +--rw priority-code-point    priority-type
   |        +--rw priority?             priority-type
   |        +--rw drop-eligible?        boolean
   +--rw pcp-encoding-table
   |  +--rw pcp-encoding-map* [pcp]
   |     +--rw pcp            pcp-selection-type
   |     +--rw priority-map* [priority dei]
   |        +--rw priority              priority-type
   |        +--rw dei                   boolean
   |        +--rw priority-code-point?   priority-type
```

PCP mapping In

PCP mappng out

Inner VID to Outer VID

- All to One (ranges)
- Set (ranges, lists)
- Individual
- Possible - Explicitly Block some VIDs

Use Explicit Model only forward specified VIDs

Inner VID to Outer VID

```
list outer-vid-inner-vid{
key “outer-vid";
     description
leaf outer-vid {
       type dot1q-types:vlanid;
       description
         “Outer VLAN identifier.";
       reference
         "12.13.2.1 of IEEE Std 802.1Q-2018";
     }
     leaf inner-vid {
       type dot1q-types:vid-range-type;
       description
         “Inner VLAN identifiers associated with this
bridge port.";
       reference
         "12.13.2.1 of IEEE Std 802.1Q-2018";
     }
```

```
+--rw outer-vid-inner-vid* [outer-vid]
       +--rw outer-vid    dot1qtypes:vlanid
       +--rw inner-vid?   dot1qtypes:vid-range-type
```

This is variation of the CVID registration Table
But it provides 1 Inner TAG to Multiple Outer TAGs
And it is very compact. (Using ranges).

In fact where no VID translation is needed this
becomes the essential VID configuration.

Notes

- There are multiple controls that enable VID forwarding
- Mapping the Inner relay VID to a outer relay VID requires the equivalent of a VID to FID mapping otherwise a bridge is likely to filter the VID.
- The current C-VID registration table allows multiple C-VIDs mapped to an S-VID but does not allow Multiple CVIDs mapped to multiple S-VID

Breaking it down – What do we need?

Bridge Outer Network Port

- Tag Type
  - VID Translator / Interface
    - Untagged -> PVID
    - Priority Tagged → PVID
    - VID tagged -> to other VIDs
  - Priority MAP / Interface
    - Input 8 PCPs to 8 PCPs
    - Or Input 16 (8PCP+2DE) to 16

Already covered

Assume one Bridge for now
Multiple Bridges ~ Virtual Interfaces

Breaking it down – What do we need?

Outer Edge Port or Inner Network Port

- Priority MAP
  - Input 8 PCPs to 8 PCPs
  - Or Input 16 (8PCP+2DE) to 16

Already covered

Outer Edge Port Shims

- MACsec SecY
- Port Access Entity
- MAC Privacy PrY

This falls out if we get the above right

Now what does it look like from a VID Configuration perspective

Inner Relay

- Determine TPID type (allows C-VID or SIV-D)

IF (VID Translation required)

  - Incoming MAP External local VID to relay VID
    - This feature is optional if local VID == Relay VID in the VLAN
    - It is useful when the VLAN used for bridged has a different VLAN ID (typically because the admin authority of the VLAN ID is not the same.

THEN

  - Incoming MAP inner relay VID to one or more outer relay VIDs

Now what does it look like from a VID perspective

Outer Relay

- Determine TPID type

IF (VID Translation required)

  - Outgoing MAP Outer relay VID to local Interface VID
    - This feature is optional if local VID equals Relay VID in the VLAN
    - It is useful when the VLAN used for bridged has a different VLAN ID (typically because the admin authority of the VLAN ID is not the same all along the path.

How to add this back to the Bridge Model

- Existing Bridge model with a few new identity's and the new Inner to Outer VLAN Map
- One option Add new component types
  - New inner-vlan-component with TPID-Config
  - New outer-vlan-component with TPID-Config
- Add a Inner-VID to Outer-VID Map
  - Allows all combinations.
  - Question Does this need an untagged flag?

Questions

- The configuration maps all VLANs in a bridge through the MACsec [MAC Privacy] on the bridge leg.
- For traffic that is not to be MACsec[MAC Privacy] How do we specify controls?
    - Multiple inner bridge relays can filter VIDs – is anything else required?

```
RPC Data Reply 46 for session 2:

rpc-reply {
  data {
    bridges {
      bridge EDE-CC1 {
        name EDE-CC1
        address 11-11-11-11-11-11
        bridge-type dot1q:ede-double-tag-bridge
        component inner-relay1 {
          name inner-relay1
          type dot1q:inner-relay-component
          bridge-vlan {
            tag-type dot1q-types:c-vlan
            vlan 1 {
              vid 1
              name ivid1
            }
          }
        }
        component outer-relay1 {
          name outer-relay1
          type dot1q:outer-relay-component
          bridge-vlan {
            tag-type dot1q-types:c-vlan
            vlan 1 {
              vid 1
              name ovid1
            }
          }
        }
      }
    }
```

```
interfaces {
      interface iep1 {
        name iep1
        type ianaift:bridge
        bridge-port {
          bridge-name EDE-CC1
          component-name inner-relay1
          port-type dot1q:inner-edge-port
        }
      }
      interface inp1 {
        name inp1
        type ianaift:bridge
        bridge-port {
          bridge-name EDE-CC1
          component-name outer-relay1
          port-type dot1q:inner-network-port
        }
      }
      interface inp2 {
        name inp2
        type ianaift:bridge
        bridge-port {
          bridge-name EDE-CC1
          component-name outer-relay2
          port-type dot1q:inner-network-port
        }
      }
      interface inp3 {
        name inp3
        type ianaift:bridge
        bridge-port {
          bridge-name EDE-CC1
          component-name outer-relay3
          port-type dot1q:inner-network-port
        }
      }
```

```
interface oep1 {
        name oep1
        type ianaift:bridge
        secy {
          controlled-port-number 1
          generation {
            max-transmit-channels 4
            max-transmit-keys 4
            protect-frames false
            always-include-sci false
            use-es false
            use-scb false
          }
          controlled-interface {
          }
          uncontrolled-interface {
            admin-point-to-point-mac auto
          }
          common-port {
          }
        }
        pae {
          pae-system 1
          port-type real-port
        }
        bridge-port {
          bridge-name EDE-CC1
          component-name inner-relay1
          port-type dot1q:outer-edge-port
          outer-vid-inner-vid 2 {
            outer-vid 2
            inner-vid 29
          }
        }
      }
```

```
interface oep2 {
        name oep2
        type ianaift:bridge
        secy {

        }
        bridge-port {
          bridge-name EDE-CC1
          component-name inner-relay1
          port-type dot1q:outer-edge-port
          outer-vid-inner-vid 20 {
            outer-vid 20
            inner-vid 7,1
          }
        }
      }
      interface oep3 {
        name oep3
        type ianaift:bridge
        secy {
        }
        bridge-port {
          bridge-name EDE-CC1
          component-name inner-relay1
          port-type dot1q:outer-edge-port
          outer-vid-inner-vid 85 {
            outer-vid 85
            inner-vid 9,12
          }
        }
      }
      interface onp1 {
        name onp1
        type ianaift:bridge
        bridge-port {
          bridge-name EDE-CC1
          component-name outer-relay1
          port-type dot1q:outer-network-port
        }
      }
    }
    nacm {
    }
    system {
      pae-system {
        name pae1
      }
    }
  }
}
```

Comments?