Simplified EDE Management Adjusted for Mapping

Don Fedyk (dfedyk@labn.net)

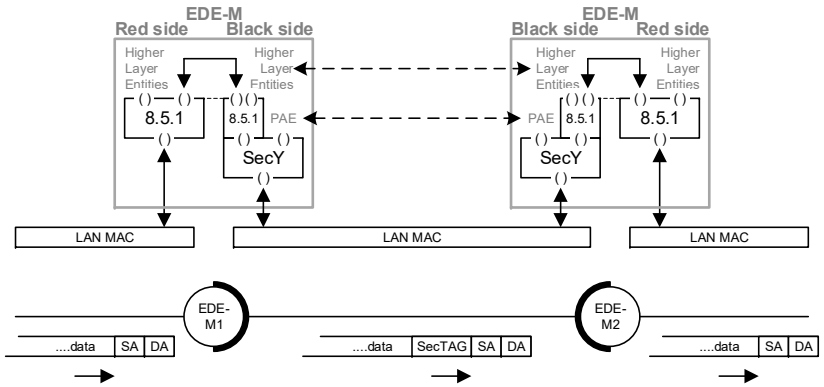
Forward

- 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

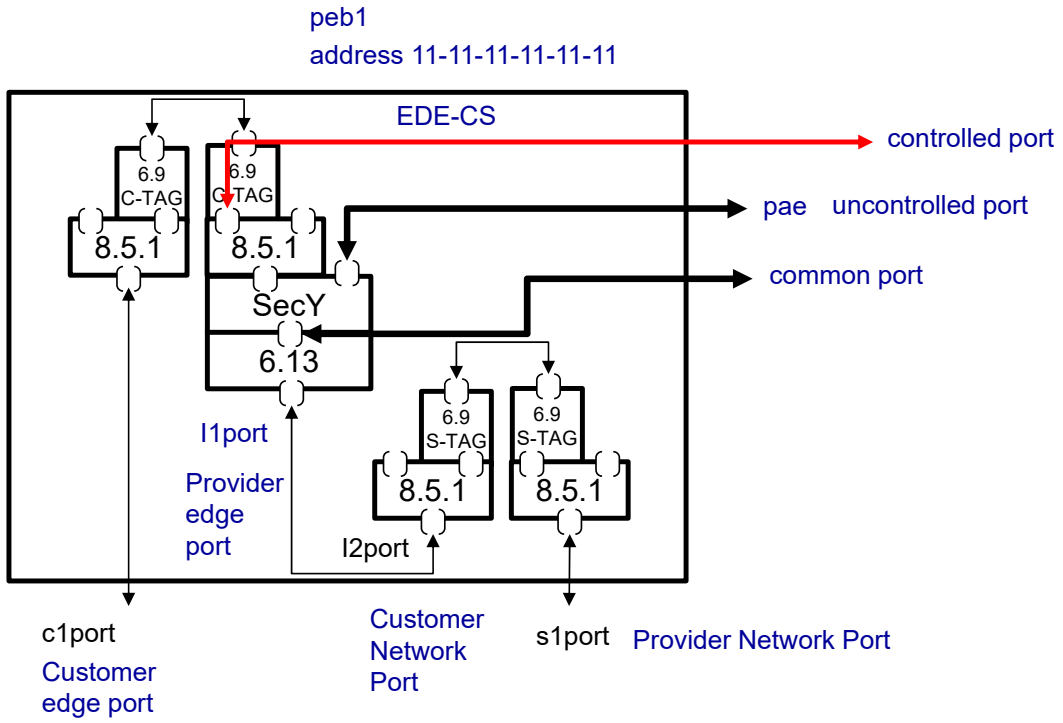
EDE-M The VLAN unaware device

Nothing to do here!

A model of configuring MACsec Shim on a bridge com

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

March Meeting

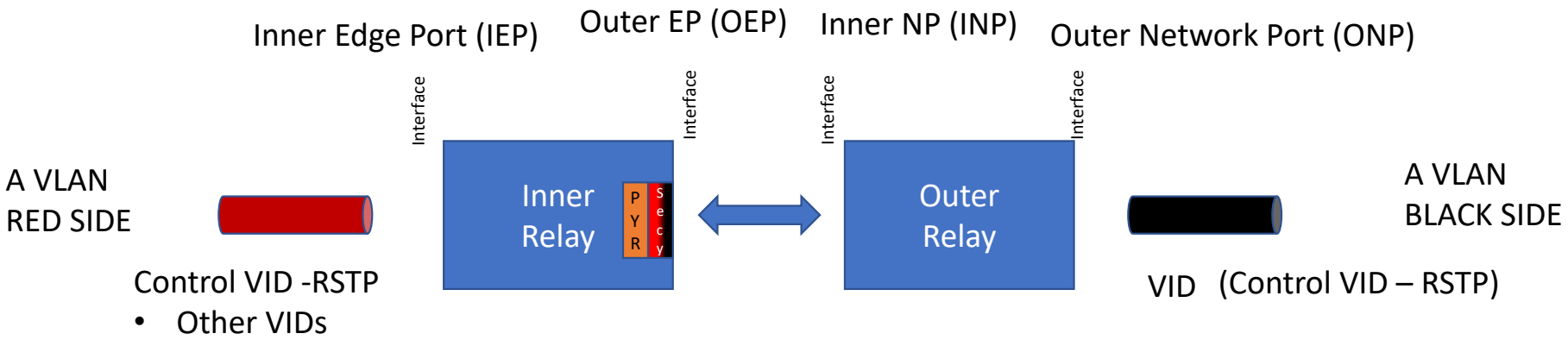
- Discussed a prototype provision of what was needed from the Bridge and Provider YANG
- Got hung up on the Mapping of components – things mostly work for an EDE-CS but there gap extending to EDE-CC and EDE-CS
- As coded the C-Components and the S-Components have behavioral characteristics and you can't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

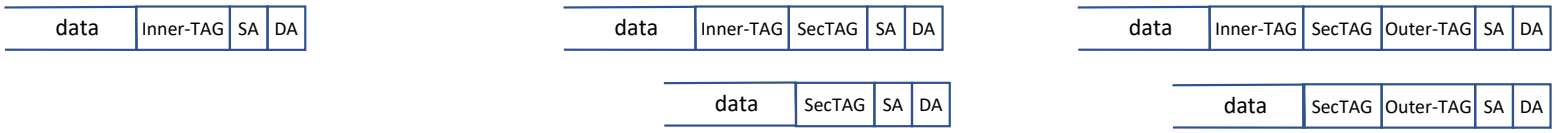
See dk-fedyk-dot1aedk-privacy-config-0317-v00

April Meeting

- Made generic mappings for VID to VID translation
- This does not quite map to what is in Clause 15 for EDE.
- dk-Fedyk-dotaedk-simple-management-0420-v00

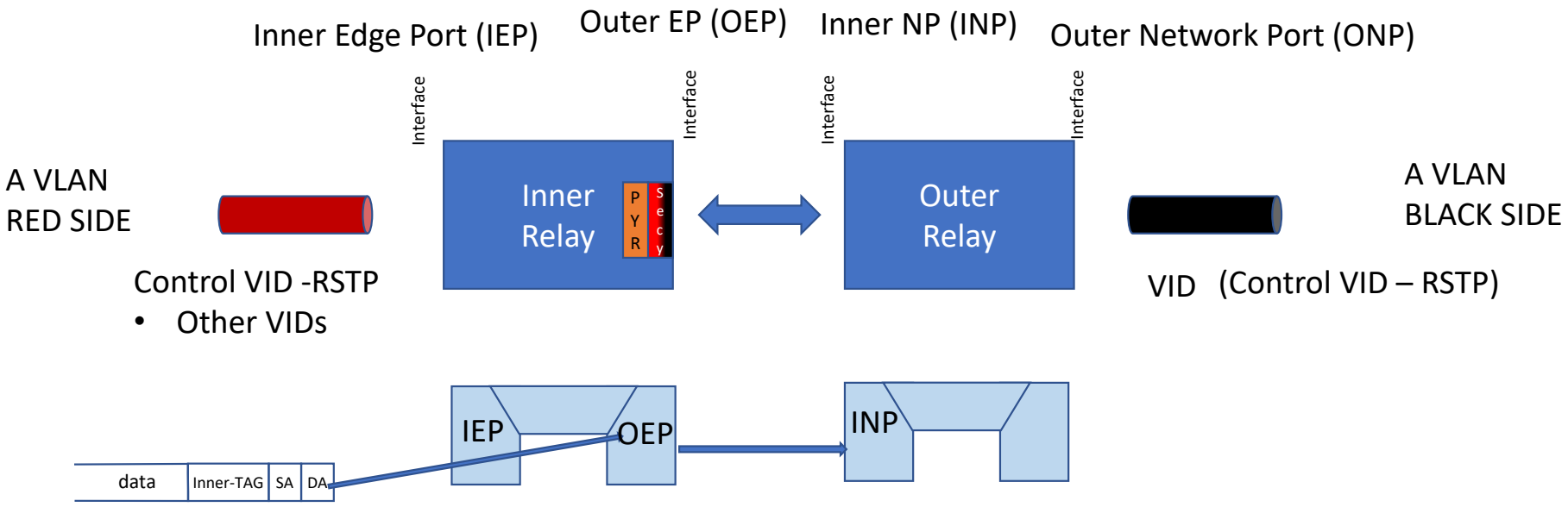
EDEs Simple Bridge Relays – What's Needed for Generic Tagging





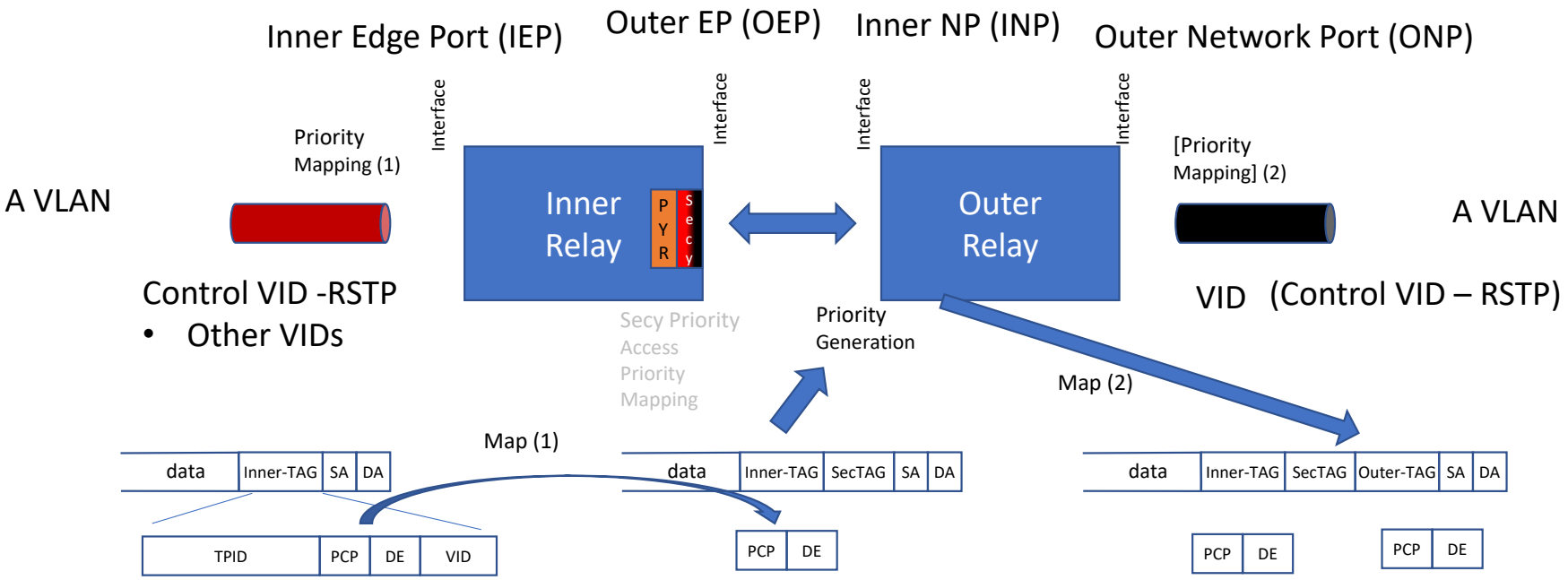
MACsec and MAC Privacy can exist on an interface typically where shown but not limited

EDEs Simple Bridge Relays – But there is More – DA Mapping



Same VID operations but traffic is mapped by DA

EDEs Simple Bridge Relays – Priority Mapping is the same



One or more Inner VIDs: 1 Outer VID

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 Relay OuterEdgePort OEP

    One or more Inner VIDs to OEP or

    One or more MAC Das to OEP

  OEP to InnerNetworkPort

Assume one Inner Bridge for now
Multiple Bridges ~ Virtual Interfaces

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 OEP, MAC DA to OEP

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

Use Explicit Model only forward specified VIDs

Inner VID to OEP, MAC DA to OEP

```
list bridge-edge-port-map{
      key "outer-edge-port";
      description "";
      leaf outer-edge-port {
        type if:interface-ref;
        description
          "Outer Edge Port.";
        reference
          " 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
          " of IEEE Std 802.1Q-2018";
      }
      leaf-list destination-mac {
        type ieee:mac-address;
        description
          "List of matching MAC addresses";
        reference
          " of IEEE Std 802.1Q-2018";
       }
    }
```

Note: The T-PID values for C-VID and S-VID Were assumed by components.

OEP to INP

```
leaf linked-interface {
      type if:interface-ref;
      description
        "Linked component interface";
      reference
        "12.13.2.1 of IEEE Std 802.1Q-2018";
    }

+--rw bridge-edge-port-map* [outer-edge-port]
     |  +--rw outer-edge-port    if:interface-ref
     |  +--rw inner-vid?        dot1q-types:vid-range-type
     |  +--rw destination-mac*   ieee:mac-dash-address
     +--rw linked-interface?                 if:interface-ref
```

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 ?

Inner Relay

- Determine TPID type (allows C-VID or S-VID)

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 or MACs to the Edge Component.

Now what does it look like ?

Map the outer Edge component to the Inner Edge component.

- Advantage here is that all VLAN operation including tagging and untagging are supported.
- (well there is untagging behavior in a provider bridge)
- Should we generalize?

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 edge Component
  - 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?

Some Proto Config

```
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 10 {
              vid 10
              name ivid10
            }
          }
        }
        component outer-relay1 {
          name outer-relay1
          type dot1q:outer-relay-component
          bridge-vlan {
            tag-type dot1q-types:c-vlan
            vlan 10 {
              vid 10
              name ovid10
            }
          }
        }
      }
    }
    interfaces {
      interface iep1 {
        name iep1
        type ianaift:bridge
        bridge-port {
          bridge-name EDE-CC1
          component-name inner-relay1
          port-type dot1q:inner-edge-port
          bridge-edge-port-map oep1 {
            outer-edge-port oep1
            inner-vid 10-20
            destination-mac 10-10-10-10-10-10
            destination-mac 10-10-10-10-10-11
          }
        }
      }
      interface inp1 {
        name inp1
        type ianaift:bridge
        bridge-port {
          bridge-name EDE-CC1
          component-name outer-relay1
          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 pae1
          port-type real-port
        }
        bridge-port {
          bridge-name EDE-CC1
          component-name inner-relay1
          port-type dot1q:outer-edge-port
          linked-interface inp1
        }
      }
      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
      }
    }
  }
}

yangcli don@localhost>
```

Comments?