YANG based Config for MAC Privacy 802.1AEdk Third Attempt

Don Fedyk (dfedyk@labn.net)

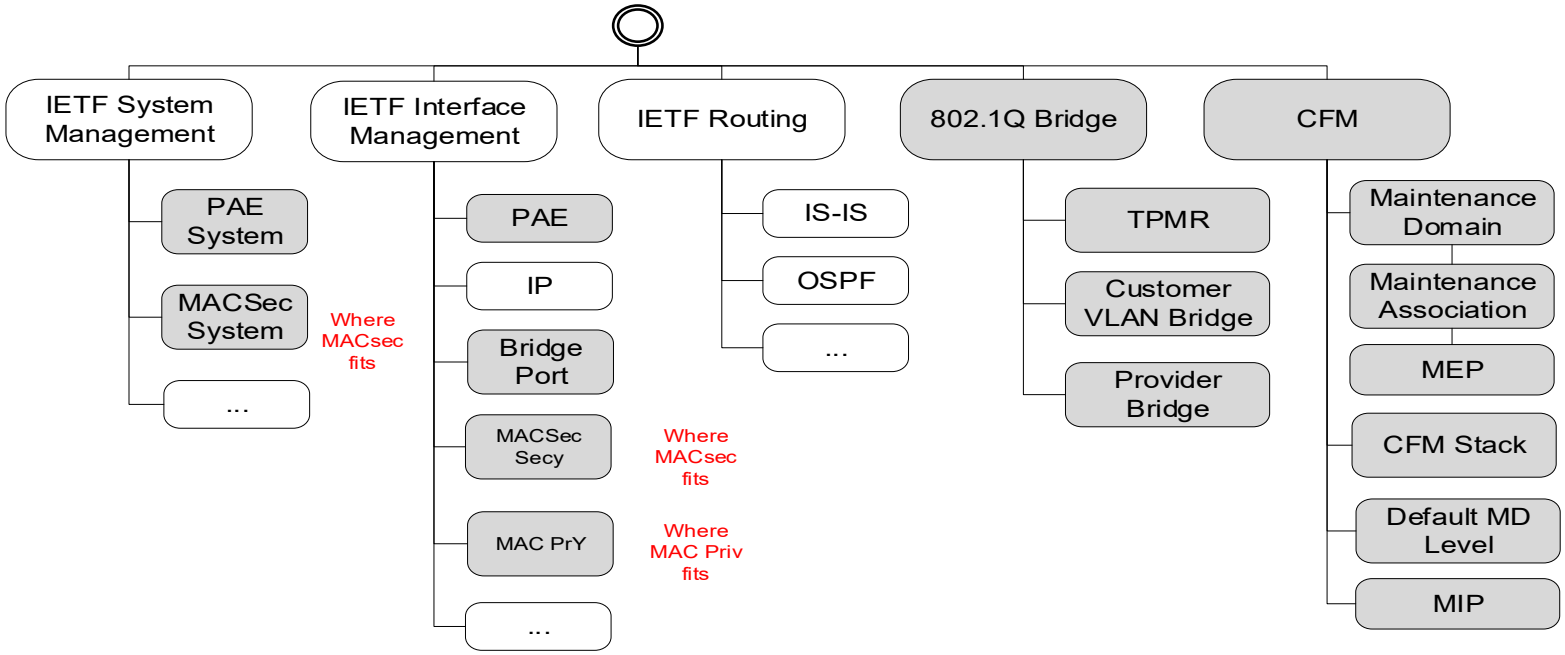
Outline

- Proto Config for MAC Privacy
- Moving configuration to standard language

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.

Instance Diagram for MACSec and MAC Privacy



MACsec and MAC Privacy YANG Some lessons learned

- Instance Model – Where the YANG trees lives
- YANG Models – What to configure and what to display
  - Our bridge Model is a large superset that supports many permutations.
  - The model contains a lot of detail.
  - The tree provides a useful summary (a slice of the instance model)
- Validation
  - Pyang – validates a single model
  - Various other tools
- Instance Configuration – IEEE is in general only beginning to look at this
  - Yuma123
  - Confd (free version)
  - Yanglint (Used by IETF)

Validation versus Instance configuration

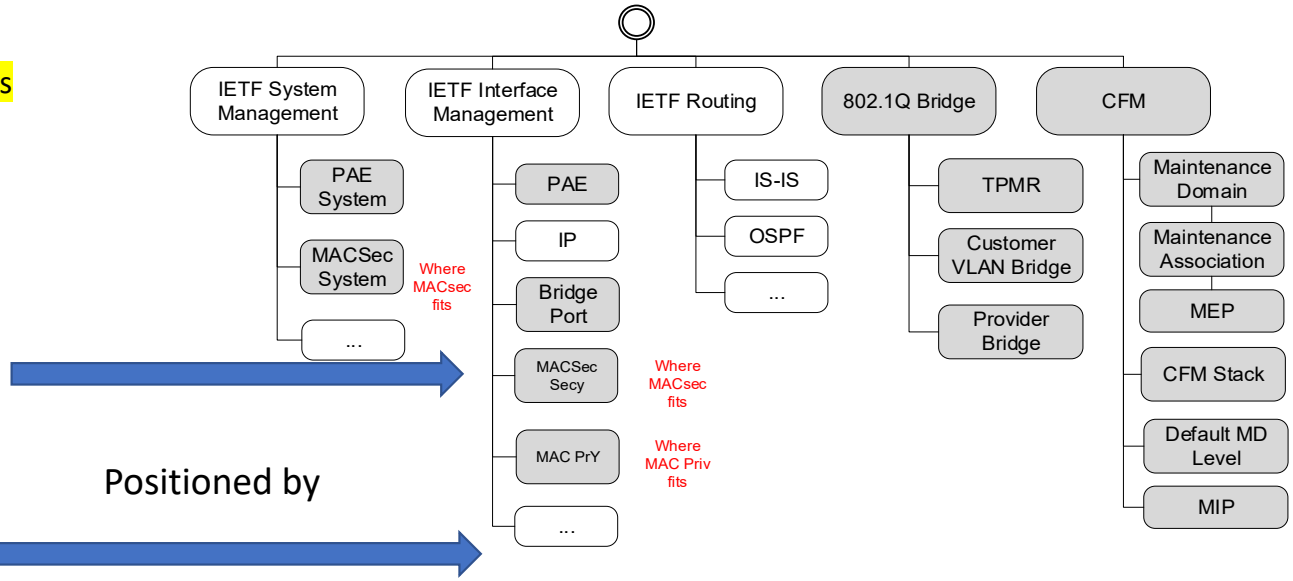
- Validation
  - YANG syntax is correct
  - YANG xpath is syntactically correct x=y (but x may be apples and y may be oranges)
  - The whole set of permutations it the tree file or the xml decription.
- Instance configuration
  - Config values are tested reference pointers are checked
  - YANG syntax is correct and multiple modules that are not related can exist side by side
  - x = y and x is the set of apple types and y is a type of apple (Macintosh but not iphone!).
  - A slice of valid configuration references links are tested

MACsec and MAC Privacy

- Mainly 2 New Modules
- ieee802-dot1ae
  - ieee802-dot1ae-types
- ieee802-dot1ae-pry

Depends on:

  - ietf-yang-types
  - ietf-inet-types
  - iana-if-type
- ieee802-dot1q-bridge
  - ieee802-dot1q-types
- ieee802-dot1x
  - ieee802-dot1x-types
- ietf-interfaces
- ietf-system
- ieee802-types



ieee802-dot1q-bridge.tree
Lots of stuff just by itself

Yanglint

- https://github.com/CESNET/libyang
  - Parsing (and validating) schemas in YANG format.
  - Parsing (and validating) schemas in YIN format.
  - Parsing, validating and printing instance data in XML format.
  - Parsing, validating and printing instance data in JSON format (RFC 7951).
  - Manipulation with the instance data.
  - Support for default values in the instance data (RFC 6243).
  - Support for YANG extensions.
  - Support for YANG Metadata (RFC 7952).
  - yanglint - feature-rich YANG tool.
  - Current implementation covers YANG 1.0 (RFC 6020) as well as YANG 1.1 (RFC 7950).
- Loads multiple modules
- IETF uses for example config
  - Can test operational tree as well
  - Gives a Slice of the larger tree.

Configure a simple VLAN bridge.

```
<qb:bridges
  xmlns:qb="urn:ieee:std:802.1Q:yang:ieee802-dot1q-bridge"
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:sy="urn:ietf:params:xml:ns:yang:ietf-system"
  xmlns:in="urn:ieee:std:802.1Q:ieee802-dot1q-types"
  xmlns:it="urn:ieee:std:802.1Q:ieee802-types"
  xmlns:ae="urn:ieee:std:802.1AE:yang:ieee802-dot1ae"
  xmlns:py="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-pry"
  xmlns:at="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-types"
  xmlns:dx="urn:ieee:std:802.1X:yang:ieee802-dot1x"
  xmlns:xt="urn:ieee:std:802.1X:yang:ieee802-dot1x-types"
  xmlns:yt="urn:ietf:params:xml:ns:yang:ietf-yang-types"
  xmlns:in="urn:ietf:params:xml:ns:yang:ietf-inet-types">
  <qb:bridge>
    <qb:name>bridge1</qb:name>
    <qb:address>10-10-10-10-10-10</qb:address>
    <qb:bridge-type>qb:customer-vlan-bridge</qb:bridge-type>
    <qb:component>
      <qb:name>cv1</qb:name>
      <qb:id>1</qb:id>
      <qb:type>qb:c-vlan-component</qb:type>
    </qb:component>
    <qb:component>
      <qb:name>cv2</qb:name>
      <qb:id>2</qb:id>
      <qb:type>qb:c-vlan-component</qb:type>
    </qb:component>
  </qb:bridge>
</qb:bridges>
```

```
<if:interfaces
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:sy="urn:ietf:params:xml:ns:yang:ietf-system"
  xmlns:qb="urn:ieee:std:802.1Q:yang:ieee802-dot1q-bridge"
  xmlns:in="urn:ieee:std:802.1Q:ieee802-dot1q-types"
  xmlns:it="urn:ieee:std:802.1Q:ieee802-types"
  xmlns:ae="urn:ieee:std:802.1AE:yang:ieee802-dot1ae"
  xmlns:py="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-pry"
  xmlns:at="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-types"
  xmlns:dx="urn:ieee:std:802.1X:yang:ieee802-dot1x"
  xmlns:xt="urn:ieee:std:802.1X:yang:ieee802-dot1x-types"
  xmlns:yt="urn:ietf:params:xml:ns:yang:ietf-yang-types"
  xmlns:in="urn:ietf:params:xml:ns:yang:ietf-inet-types"
  xmlns:ia="urn:ietf:params:xml:ns:yang:iana-if-type">
    <if:interface>
      <if:name>eth0</if:name>
      <if:type>ia:bridge</if:type>
      <qb:bridge-port>
        <qb:bridge-name>bridge1</qb:bridge-name>
        <qb:component-name>cv1</qb:component-name>
        <qb:port-type>qb:c-vlan-bridge-port</qb:port-type>
        <qb:pvid>1</qb:pvid>
    </qb:bridge-port>
    </if:interface>
    <if:interface>
      <if:name>eth1</if:name>
      <if:type>ia:ethernetCsmacd</if:type>
      <qb:bridge-port>
        <qb:bridge-name>bridge1</qb:bridge-name>
        <qb:component-name>cv2</qb:component-name>
        <qb:port-type>qb:c-vlan-bridge-port</qb:port-type>
        <qb:pvid>1</qb:pvid>
      </qb:bridge-port>
    </if:interface>
</if:interfaces>
```

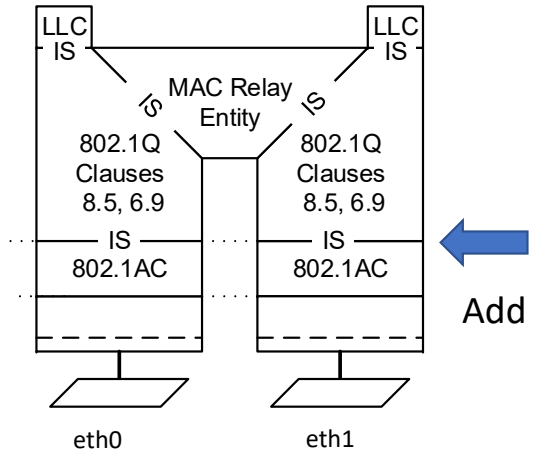
Yanglint JSON output for a VLAN Bridge

```
data  -t config   -f json basic-vlan-bridge.xml
{
  "ieee802-dot1q-bridge:bridges": {
    "bridge": [
      {
        "name": "bridge1",
        "address": "10-10-10-10-10-10",
        "bridge-type": "customer-vlan-bridge",
        "component": [
          {
            "name": "cv1",
            "id": 1,
            "type": "c-vlan-component"
          },
          {
            "name": "cv2",
            "id": 2,
            "type": "c-vlan-component"
          }
        ]
      }
    ]
  },
```

```
 "ietf-interfaces:interfaces": {
   "interface": [
     {
       "name": "eth0",
       "type": "iana-if-type:bridge",
       "ieee802-dot1q-bridge:bridge-port": {
         "bridge-name": "bridge1",
         "component-name": "cv1",
         "port-type": "c-vlan-bridge-port",
         "pvid": 1
       },
       "ieee802-dot1x:pae": {

       }
     },
     {
       "name": "eth1",
       "type": "iana-if-type:ethernetCsmacd",
       "ieee802-dot1q-bridge:bridge-port": {
         "bridge-name": "bridge1",
         "component-name": "cv2",
         "port-type": "c-vlan-bridge-port",
         "pvid": 1
       },
       "ieee802-dot1x:pae": {

       }
     }
   ]
 }
}
```



Add MACsec

Not too complicated
- No explicit VID tagging just PVID

Add MACsec (delta to interfaces)

```
</if:interface>
    <if:interface>
      <if:name>eth1</if:name>
      <if:type>ia:ethernetCsmacd</if:type>
      <qb:bridge-port>
        <qb:bridge-name>bridge1</qb:bridge-name>
        <qb:component-name>cv2</qb:component-name>
        <qb:port-type>qb:c-vlan-bridge-port</qb:port-type>
        <qb:pvid>1</qb:pvid>
      </qb:bridge-port>
    <ae:secy>
      <ae:controlled-port-number>1</ae:controlled-port-number>
      <ae:verification>
       <ae:validate-frames>strict</ae:validate-frames>
       <ae:replay-protect>true</ae:replay-protect>
      </ae:verification>
      <ae:generation>
       <ae:max-transmit-channels>16</ae:max-transmit-channels>
       <ae:max-transmit-keys>16</ae:max-transmit-keys>
       <ae:protect-frames>true</ae:protect-frames>
       <ae:always-include-sci>true</ae:always-include-sci>
       <ae:use-es>true</ae:use-es>
       <ae:use-scb>true</ae:use-scb>
       <ae:user-priority-tc>
         <ae:user-priority>0</ae:user-priority>
         <ae:traffic-class>0</ae:traffic-class>
         <ae:access-class-de0>0</ae:access-class-de0>
         <ae:access-class-de1>0</ae:access-class-de1>
       </ae:user-priority-tc>
       <ae:user-priority-tc>
         <ae:user-priority>1</ae:user-priority>
         <ae:traffic-class>1</ae:traffic-class>
         <ae:access-class-de0>1</ae:access-class-de0>
         <ae:access-class-de1>1</ae:access-class-de1>
       </ae:user-priority-tc>
```

```
  <ae:user-priority-tc>
         <ae:user-priority>2</ae:user-priority>
         <ae:traffic-class>2</ae:traffic-class>
         <ae:access-class-de0>2</ae:access-class-de0>
         <ae:access-class-de1>2</ae:access-class-de1>
       </ae:user-priority-tc>
       <ae:user-priority-tc>
         <ae:user-priority>3</ae:user-priority>
         <ae:traffic-class>3</ae:traffic-class>
         <ae:access-class-de0>3</ae:access-class-de0>
         <ae:access-class-de1>3</ae:access-class-de1>
       </ae:user-priority-tc>
      </ae:generation>
    </ae:secy>
      <dx:pae>
       <dx:pae-system>pae1</dx:pae-system>
      </dx:pae>
    </if:interface>
</if:interfaces>
<sy:system
  xmlns:sy="urn:ietf:params:xml:ns:yang:ietf-system"
  xmlns:yt="urn:ietf:params:xml:ns:yang:ietf-yang-types"
  xmlns:it="urn:ieee:std:802.1Q:ieee802-types"
  xmlns:xt="urn:ieee:std:802.1X:yang:ieee802-dot1x-types"
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:ia="urn:ietf:params:xml:ns:yang:iana-if-type"
  xmlns:dx="urn:ieee:std:802.1X:yang:ieee802-dot1x">
  <sy:contact>test</sy:contact>
  <dx:pae-system>
     <dx:name>pae1</dx:name>
     <dx:system-access-control>enabled</dx:system-access-control>
  </dx:pae-system>
</sy:system>
```

Note partial file

Yanglint JSON output for a VLAN Bridge with MACsec

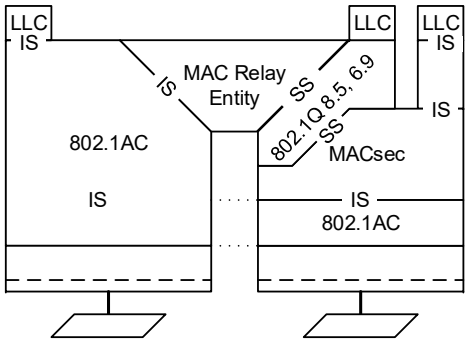
```
> data  -t config   -f json basic-vlan-bridge-with-macsec.xml    {
{
  "ieee802-dot1q-bridge:bridges": {
    "bridge": [
      {
        "name": "bridge1",
        "address": "10-10-10-10-10-10",
        "bridge-type": "customer-vlan-bridge",
        "component": [
          {
            "name": "cv1",
            "id": 1,
            "type": "c-vlan-component"
          },
          {
            "name": "cv2",
            "id": 2,
            "type": "c-vlan-component"
          }
        ]
      }
    ]
  },
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:bridge",
        "ieee802-dot1q-bridge:bridge-port": {
          "bridge-name": "bridge1",
          "component-name": "cv1",
          "port-type": "c-vlan-bridge-port",
          "pvid": 1
        },
        "ieee802-dot1x:pae": {

        }
      },
```

```
"name": "eth1",
"type": "iana-if-type:ethernetCsmacd",
"ieee802-dot1q-bridge:bridge-port": {
  "bridge-name": "bridge1",
  "component-name": "cv2",
  "port-type": "c-vlan-bridge-port",
  "pvid": 1
},
"ieee802-dot1ae:secy": {
  "controlled-port-number": 1,
  "verification": {
    "validate-frames": "strict",
    "replay-protect": true
  },
  "generation": {
    "max-transmit-channels": 16,
    "max-transmit-keys": 16,
    "protect-frames": true,
    "always-include-sci": true,
    "use-es": true,
    "use-scb": true,
    "user-priority-tc": [
      {
        "user-priority": 0,
        "traffic-class": 0,
        "access-class-de0": 0,
        "access-class-de1": 0
      },
      {
        "user-priority": 1,
        "traffic-class": 1,
        "access-class-de0": 1,
        "access-class-de1": 1
      },
      {
        "user-priority": 2,
        "traffic-class": 2,
        "access-class-de0": 2,
        "access-class-de1": 2
      },
```

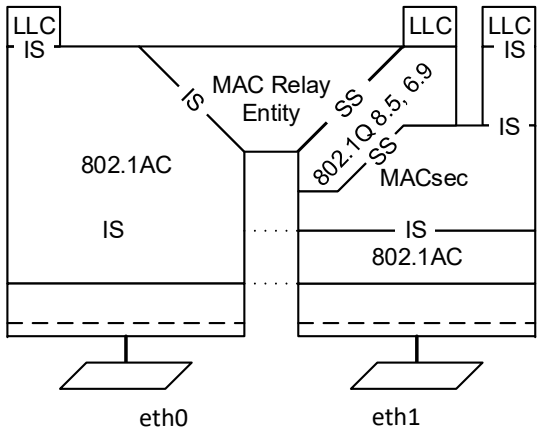
```
            {
              "user-priority": 3,
              "traffic-class": 3,
              "access-class-de0": 3,
              "access-class-de1": 3
            }
          ]
        }
      },
      "ieee802-dot1x:pae": {
        "pae-system": "pae1"
      }
    }
  ]
},
"ietf-system:system": {
  "contact": "test",
  "ieee802-dot1x:pae-system": {
    "name": "pae1",
    "system-access-control": "enabled"
  }
}
}
v
```

Note Abbreviated
TC table is 8 priorities 4 are shown etc



Yanglint JSON output for a VLAN Bridge with MACsec

- MACsec Position is controlled by the interface
- Interface position is controlled by the link to the component
- VLAN Behavior is controlled by the Bridge Port



Adding MAC Privacy is similar

```
<qb:bridges
  xmlns:qb="urn:ieee:std:802.1Q:yang:ieee802-dot1q-bridge"
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:sy="urn:ietf:params:xml:ns:yang:ietf-system"
  xmlns:in="urn:ieee:std:802.1Q:ieee802-dot1q-types"
  xmlns:it="urn:ieee:std:802.1Q:ieee802-types"
  xmlns:ae="urn:ieee:std:802.1AE:yang:ieee802-dot1ae"
  xmlns:py="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-pry"
  xmlns:at="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-types"
  xmlns:dx="urn:ieee:std:802.1X:yang:ieee802-dot1x"
  xmlns:xt="urn:ieee:std:802.1X:yang:ieee802-dot1x-types"
  xmlns:yt="urn:ietf:params:xml:ns:yang:ietf-yang-types"
  xmlns:in="urn:ietf:params:xml:ns:yang:ietf-inet-types">
  <qb:bridge>
    <qb:name>bridge1</qb:name>
    <qb:address>10-10-10-10-10-10</qb:address>
    <qb:bridge-type>qb:customer-vlan-bridge</qb:bridge-type>
    <qb:component>
      <qb:name>cv1</qb:name>
      <qb:id>1</qb:id>
      <qb:type>qb:c-vlan-component</qb:type>
      <qb:bridge-vlan>
       <qb:vlan>
          <qb:vid>2</qb:vid>
          <qb:name>vlan2</qb:name>
       </qb:vlan>
      </qb:bridge-vlan>
    </qb:component>
    <qb:component>
      <qb:name>cv2</qb:name>
      <qb:id>2</qb:id>
      <qb:type>qb:c-vlan-component</qb:type>
      <qb:bridge-vlan>
       <qb:vlan>
          <qb:vid>2</qb:vid>
          <qb:name>vlan2</qb:name>
       </qb:vlan>
      </qb:bridge-vlan>
    </qb:component>
  </qb:bridge>
</qb:bridges>
```

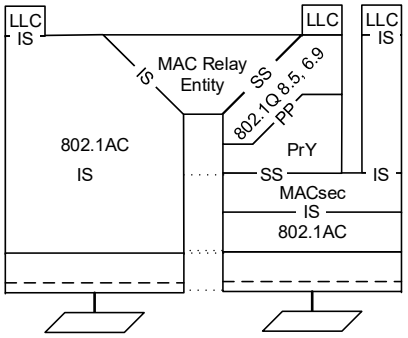
```
<if:interfaces
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:sy="urn:ietf:params:xml:ns:yang:ietf-system"
  xmlns:qb="urn:ieee:std:802.1Q:yang:ieee802-dot1q-bridge"
  xmlns:in="urn:ieee:std:802.1Q:ieee802-dot1q-types"
  xmlns:it="urn:ieee:std:802.1Q:ieee802-types"
  xmlns:ae="urn:ieee:std:802.1AE:yang:ieee802-dot1ae"
  xmlns:py="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-pry"
  xmlns:at="urn:ieee:std:802.1AE:yang:ieee802-dot1ae-types"
  xmlns:dx="urn:ieee:std:802.1X:yang:ieee802-dot1x"
  xmlns:xt="urn:ieee:std:802.1X:yang:ieee802-dot1x-types"
  xmlns:yt="urn:ietf:params:xml:ns:yang:ietf-yang-types"
  xmlns:in="urn:ietf:params:xml:ns:yang:ietf-inet-types"
  xmlns:ia="urn:ietf:params:xml:ns:yang:iana-if-type">
    <if:interface>
      <if:name>eth0</if:name>
      <if:type>ia:bridge</if:type>
      <qb:bridge-port>
        <qb:bridge-name>bridge1</qb:bridge-name>
        <qb:component-name>cv1</qb:component-name>
        <qb:port-type>qb:c-vlan-bridge-port</qb:port-type>
    </qb:bridge-port>
    </if:interface>
    <if:interface>
      <if:name>eth1</if:name>
      <if:type>ia:ethernetCsmacd</if:type>
      <qb:bridge-port>
        <qb:bridge-name>bridge1</qb:bridge-name>
        <qb:component-name>cv2</qb:component-name>
        <qb:port-type>qb:c-vlan-bridge-port</qb:port-type>
      </qb:bridge-port>
    <ae:secy>
      <ae:controlled-port-number>1</ae:controlled-port-number>
      <ae:verification>
       <ae:validate-frames>strict</ae:validate-frames>
       <ae:replay-protect>true</ae:replay-protect>
      </ae:verification>
      <ae:generation>
       <ae:max-transmit-channels>16</ae:max-transmit-channels>
       <ae:max-transmit-keys>16</ae:max-transmit-keys>
       <ae:protect-frames>true</ae:protect-frames>
       <ae:always-include-sci>true</ae:always-include-sci>
       <ae:use-es>true</ae:use-es>
       <ae:use-scb>true</ae:use-scb>
```

```
<ae:user-priority-tc>
        <ae:user-priority>0</ae:user-priority>
        <ae:traffic-class>0</ae:traffic-class>
        <ae:access-class-de0>0</ae:access-class-de0>
        <ae:access-class-de1>0</ae:access-class-de1>
      </ae:user-priority-tc>
      <ae:user-priority-tc>
        <ae:user-priority>1</ae:user-priority>
        <ae:traffic-class>1</ae:traffic-class>
        <ae:access-class-de0>1</ae:access-class-de0>
        <ae:access-class-de1>1</ae:access-class-de1>
      </ae:user-priority-tc>
      <ae:user-priority-tc>
        <ae:user-priority>2</ae:user-priority>
        <ae:traffic-class>2</ae:traffic-class>
        <ae:access-class-de0>2</ae:access-class-de0>
        <ae:access-class-de1>2</ae:access-class-de1>
      </ae:user-priority-tc>
      <ae:user-priority-tc>
        <ae:user-priority>3</ae:user-priority>
        <ae:traffic-class>3</ae:traffic-class>
        <ae:access-class-de0>3</ae:access-class-de0>
        <ae:access-class-de1>3</ae:access-class-de1>
      </ae:user-priority-tc>
     </ae:generation>
   </ae:secy>
```

Adding MAC Privacy is similar

```
<py:pry>
        <py:mac-privacy>enabled</py:mac-privacy>
        <py:user-priority-to-pry>
         <py:user-priority>0</py:user-priority>
         <py:privacy-type>py:frame-a</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>1</py:user-priority>
         <py:privacy-type>py:express-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>2</py:user-priority>
         <py:privacy-type>py:express-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>3</py:user-priority>
         <py:privacy-type>py:express-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>4</py:user-priority>
         <py:privacy-type>py:standard-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>5</py:user-priority>
         <py:privacy-type>py:standard-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>6</py:user-priority>
         <py:privacy-type>py:standard-channel</py:privacy-type>
       </py:user-priority-to-pry>
        <py:user-priority-to-pry>
         <py:user-priority>7</py:user-priority>
         <py:privacy-type>py:standard-channel</py:privacy-type>
       </py:user-priority-to-pry>
       <py:privacy-channel>
         <py:pc>py:standard-channel</py:pc>
         <py:max-per-second-bitrate>10000000000</py:max-per-second-bitrate>
         <py:max-mppdu-size>1500</py:max-mppdu-size>
         <py:mppdu-priority>3</py:mppdu-priority>
       </py:privacy-channel>
      </py:pry>
      <dx:pae>
        <dx:pae-system>pae1</dx:pae-system>
      </pae>
    </if:interface>
</if:interfaces>
```

```
<sy:system
  xmlns:sy="urn:ietf:params:xml:ns:yang:ietf-system"
  xmlns:yt="urn:ietf:params:xml:ns:yang:ietf-yang-types"
  xmlns:it="urn:ieee:std:802.1Q:ieee802-types"
  xmlns:xt="urn:ieee:std:802.1X:yang:ieee802-dot1x-types"
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:ia="urn:ietf:params:xml:ns:yang:iana-if-type"
  xmlns:dx="urn:ieee:std:802.1X:yang:ieee802-dot1x">
  <sy:contact>test</sy:contact>
  <dx:pae-system>
     <dx:name>pae1</dx:name>
     <dx:system-access-control>enabled</dx:system-access-control>
  </dx:pae-system>
</sy:system>
```



Yanglint JSON output for a VLAN Bridge with MACsec

```
> data  -t config   -f json basic-vlan-bridge-with-pry.xml     {
{
  "ieee802-dot1q-bridge:bridges": {
    "bridge": [
      {
        "name": "bridge1",
        "address": "10-10-10-10-10-10",
        "bridge-type": "customer-vlan-bridge",
        "component": [
          {
            "name": "cv1",
            "id": 1,
            "type": "c-vlan-component"
          },
          {
            "name": "cv2",
            "id": 2,
            "type": "c-vlan-component"
          }
        ]
      }
    ]
  },
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:bridge",
        "ieee802-dot1q-bridge:bridge-port": {
          "bridge-name": "bridge1",
          "component-name": "cv1",
          "port-type": "c-vlan-bridge-port",
          "pvid": 1
        },
        "ieee802-dot1x:pae": {

        }
      },
```

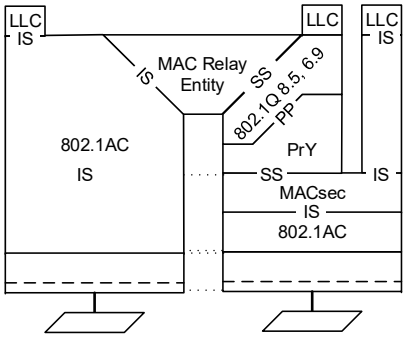
Note Abbreviated
TC table is 8 priorities 4 are shown etc

```
"name": "eth1",
"type": "iana-if-type:ethernetCsmacd",
"ieee802-dot1ae-pry:pry": {
  "mac-privacy": "enabled",
  "user-priority-to-pry": [
    {
      "user-priority": 0,
      "privacy-type": "frame-a"
    },
    {
      "user-priority": 1,
      "privacy-type": "express-channel"
    },
    {
      "user-priority": 2,
      "privacy-type": "express-channel"
    },
    {
      "user-priority": 3,
      "privacy-type": "express-channel"
    },
    {
      "user-priority": 4,
      "privacy-type": "standard-channel"
    },
    {
      "user-priority": 5,
      "privacy-type": "standard-channel"
    },
    {
      "user-priority": 6,
      "privacy-type": "standard-channel"
    },
    {
      "user-priority": 7,
      "privacy-type": "standard-channel"
    }
  ],
  "privacy-channel": [
    {
      "pc": "standard-channel",
      "max-per-second-bitrate": "10000000000",
      "max-mppdu-size": 1500,
      "mppdu-priority": 3
    }
  ]
},
```

```
"ieee802-dot1q-bridge:bridge-port": {
          "bridge-name": "bridge1",
          "component-name": "cv2",
          "port-type": "c-vlan-bridge-port",
          "pvid": 1
        },
        "ieee802-dot1ae:secy": {
          "controlled-port-number": 1,
          "verification": {
            "validate-frames": "strict",
            "replay-protect": true
          },
          "generation": {
            "max-transmit-channels": 16,
            "max-transmit-keys": 16,
            "protect-frames": true,
            "always-include-sci": true,
            "use-es": true,
            "use-scb": true,
            "user-priority-tc": [
              {
                "user-priority": 0,
                "traffic-class": 0,
                "access-class-de0": 0,
                "access-class-de1": 0
              },
              {
                "user-priority": 1,
                "traffic-class": 1,
                "access-class-de0": 1,
                "access-class-de1": 1
              },
              {
                "user-priority": 2,
                "traffic-class": 2,
                "access-class-de0": 2,
                "access-class-de1": 2
              },
```

Yanglint JSON output for a VLAN Bridge with MACsec


```
{
                "user-priority": 3,
                "traffic-class": 3,
                "access-class-de0": 3,
                "access-class-de1": 3
              }
            ]
          }
        },
        "ieee802-dot1x:pae": {
          "pae-system": "pae1"
        }
      }
    ]
  },
  "ietf-system:system": {
    "contact": "test",
    "ieee802-dot1x:pae-system": {
      "name": "pae1",
      "system-access-control": "enabled"
    }
  }
}
v
```

What about VLAN Control in single bridge

- Untagged ports can be tagged with PVID
- Tagged port can “flow through” if the TPID is of the same type.
- No Explicit VLAN tag control.
- Plenty of control plane control of filtering MACs and VIDs

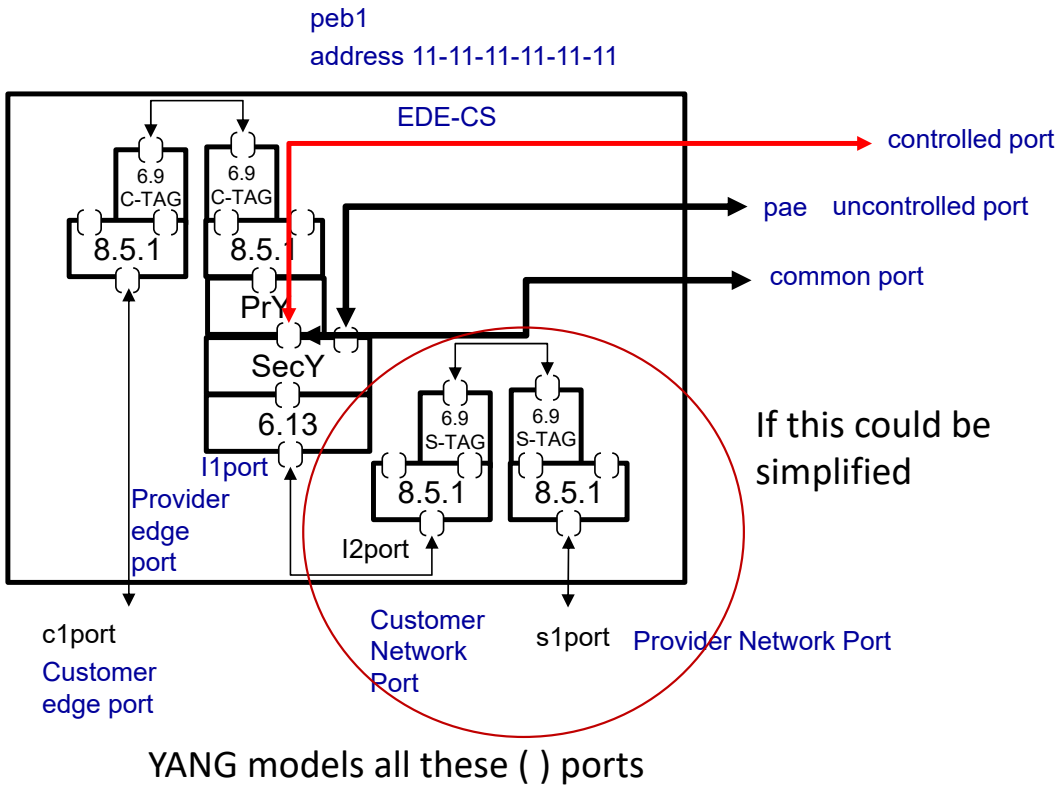
What about EDEs?

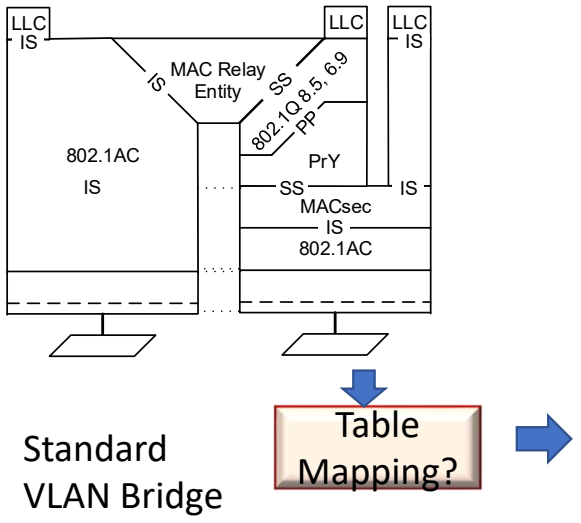
- This is simple Bridge with a control plane controlling the VLANs
- The simple bridge does not cover the details of the EDE components
- Prior Presentations looked at Provider Bridged components
  - http://www.ieee802.org/1/files/public/docs2020/dk-fedyk-dot1aedk-simple-management-0520-v00.pdf
  - Conclusion was Map ports or handles to inner relay (PEP)
  - Then Incoming MAP inner relay to the Edge Component.
  - Edge compose applies outer VID
- Are there simple tagging configuration where a table could handle the bulk of EDE cases in the simple model?
  - Less flexibility than the complete Provider Bridge general model but good enough?

The rest of this presentation is for Discussion

- Is there a table format that can summarize the outer component for the simple EDE cases?

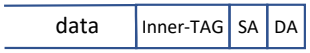
MACsec Config for EDEs

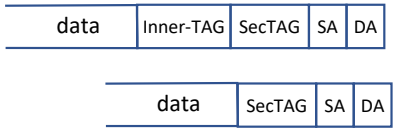


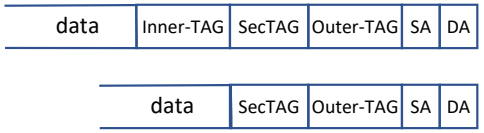


Simple Cases we need

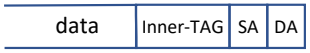
Detailed Model handled today







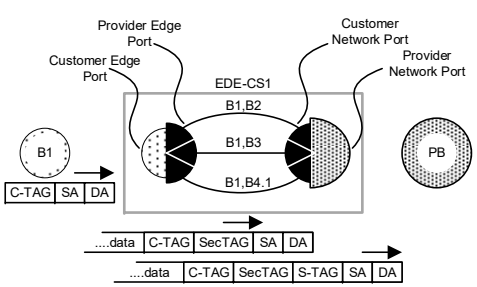
Incoming TAG

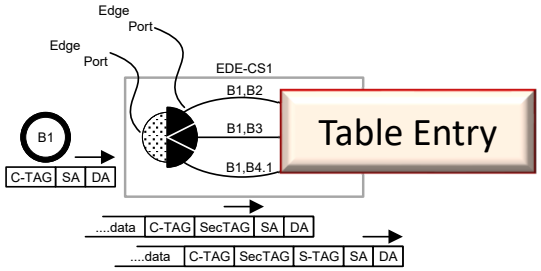


Possible Simple Model ?

Table Mapping?

Outgoing TAGs


Provider Edge Port
Customer Edge Port
EDE-CS1
Customer Network Port
Provider Network Port
B1,B2
B1,B3
B1,B4.1
B1
PB
C-TAG SA DA
....data C-TAG SecTAG SA DA
....data C-TAG SecTAG S-TAG SA DA


Edge Port
Edge Port
EDE-CS1
B1,B2
B1,B3
B1,B4.1
Table Entry
B1
C-TAG SA DA
....data C-TAG SecTAG SA DA
....data C-TAG SecTAG S-TAG SA DA

Cases we need: To be completed

Incoming TAG	Inner TAG	Outer TAG	Needed ?	Notes
None	C-TAG	S-TAG		
None	C-TAG	C-TAG		
None	S-TAG	S-TAG		
None	C-TAG	None		
None	S-TAG	None		
C-TAG	Original C-TAG	S-TAG		
C-TAG	Original C-TAG	C-TAG		
C-TAG	Original C-TAG	None		
S-TAG	Original S-TAG	S-TAG		
S-TAG	Original S-TAG	C-TAG		
S-TAG	Original S-TAG	None		

Conclusions

- To be filled by meeting notes.