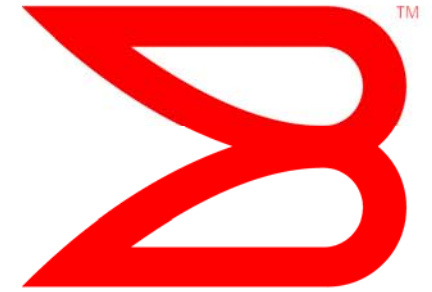


**BROCADE**



# Ethernet Switch ACLs for FCoE

A comparison of ACLs required for :  
Server Provided MAC Address Environments

(aka "Burned In" MAC Address)

and

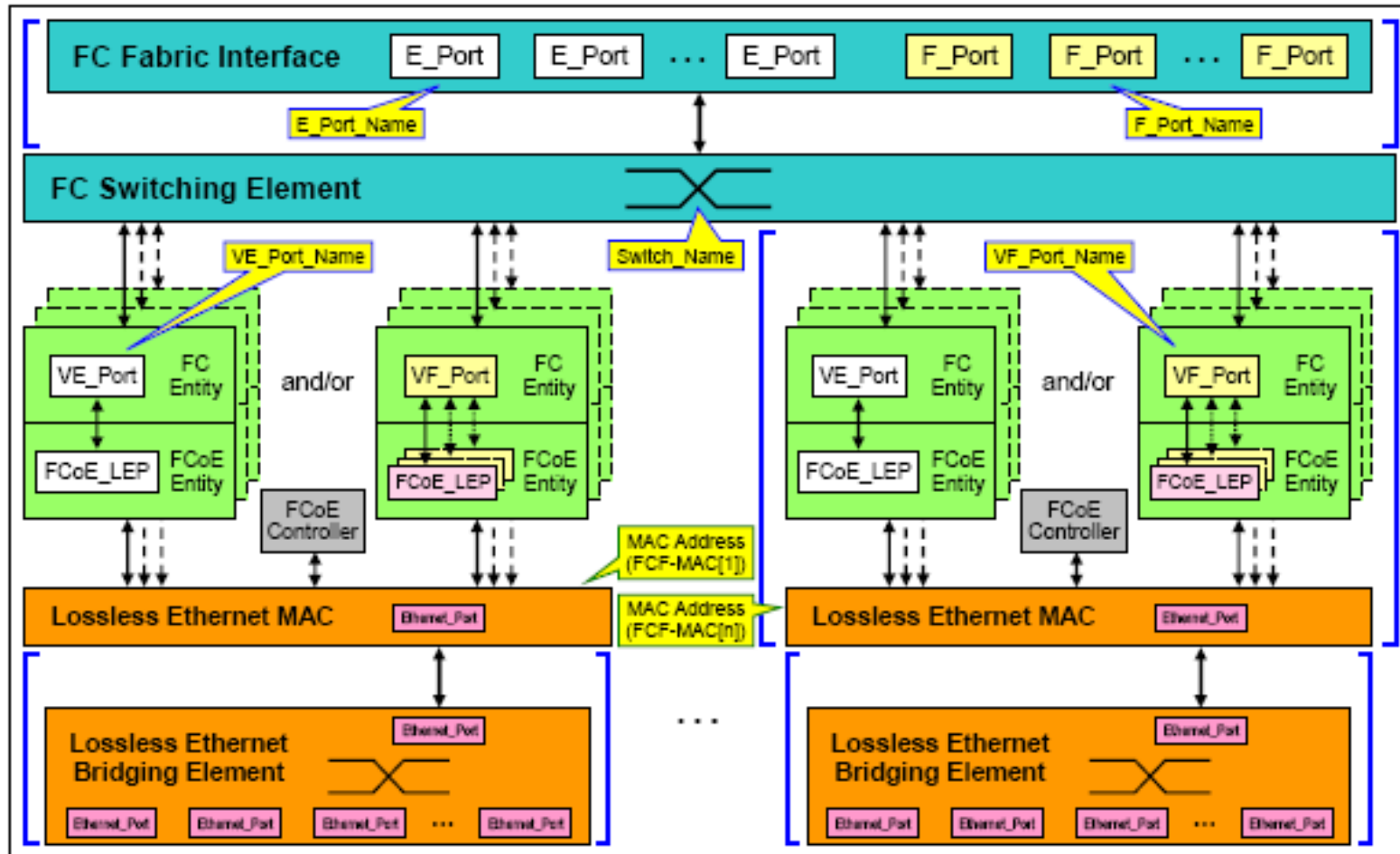
Network Provided MAC Address Environments

(aka Mapped MAC Address)

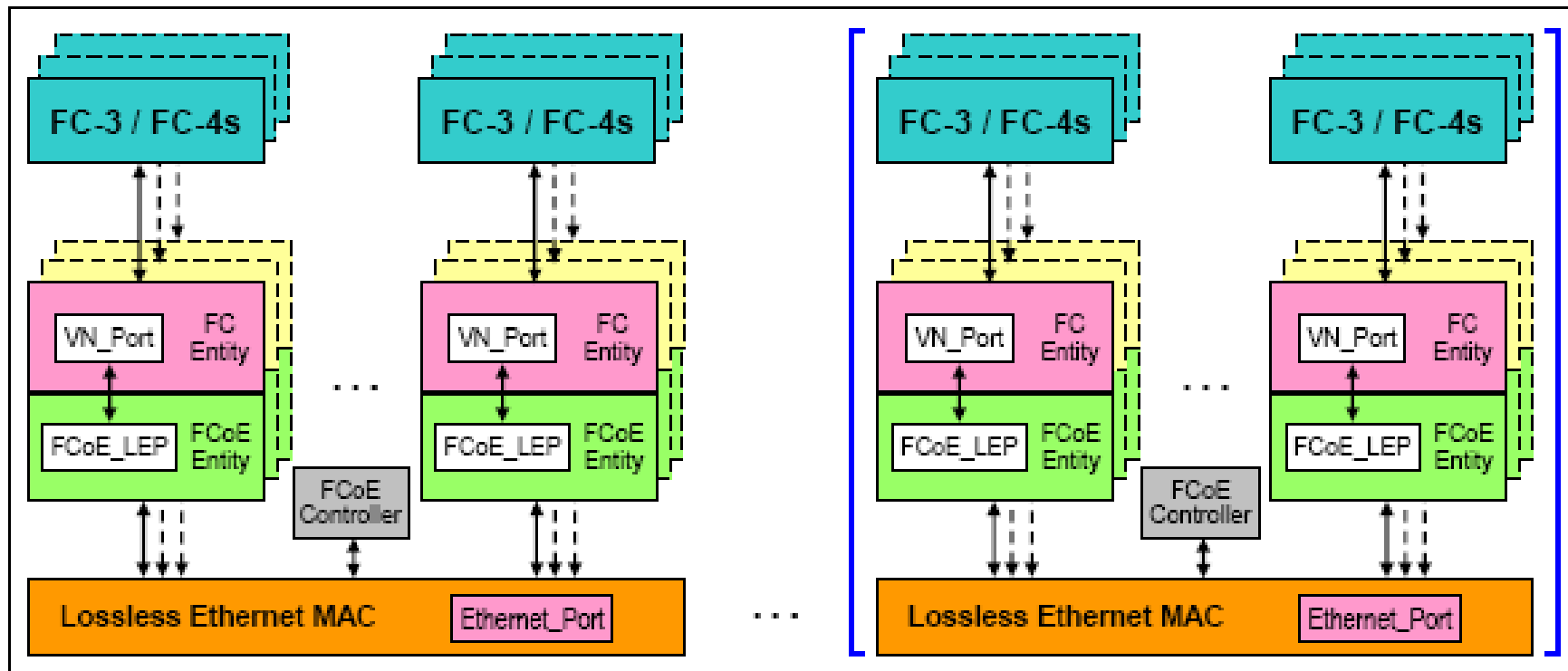
T11/07-656v0

John L. Hufferd  
Robert Snively

# FCoE VE\_Port/VF\_Port Functional Model



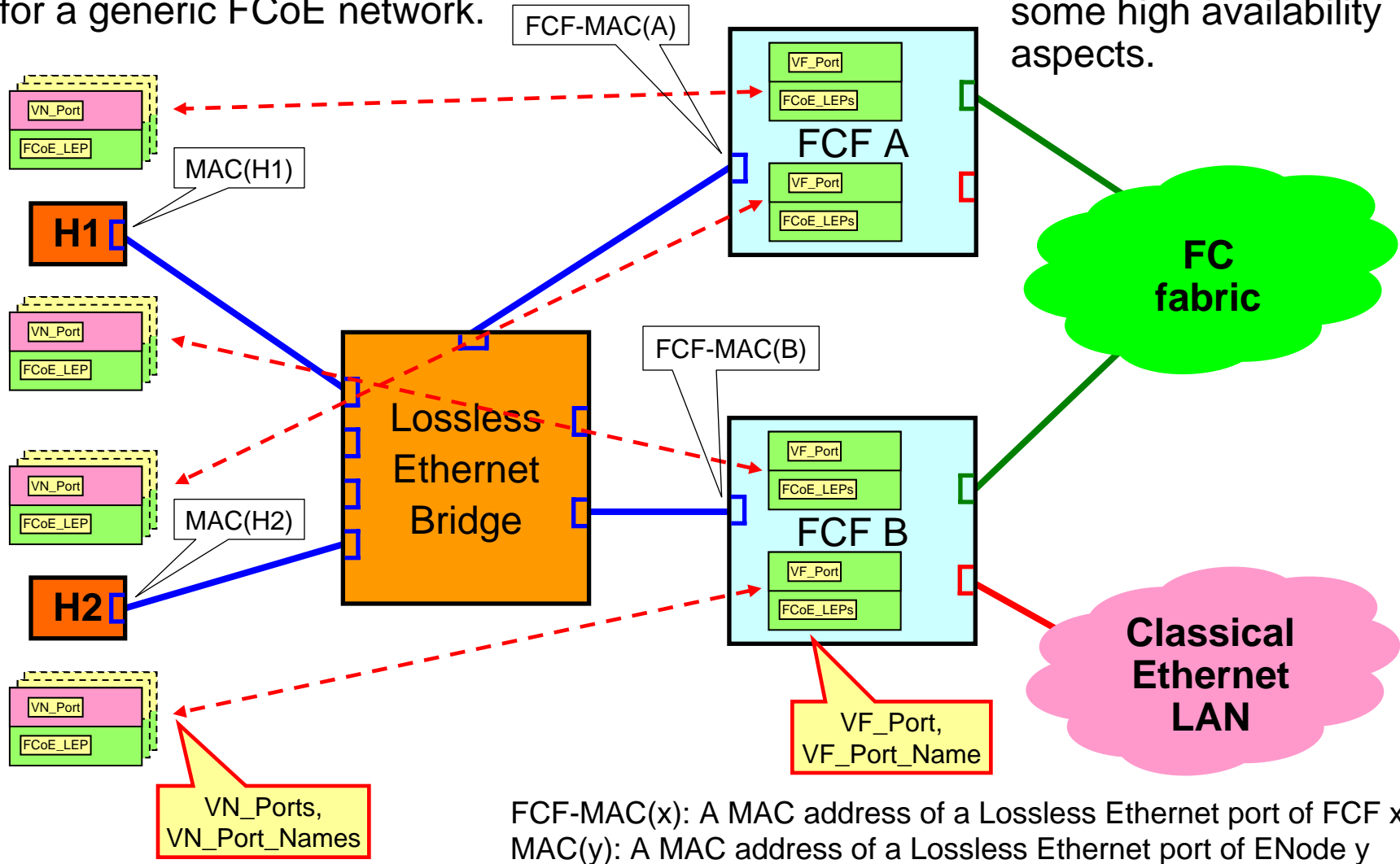
# FCoE VN\_Port/ENode Functional Model



# Generic Topology with NPIV stacks

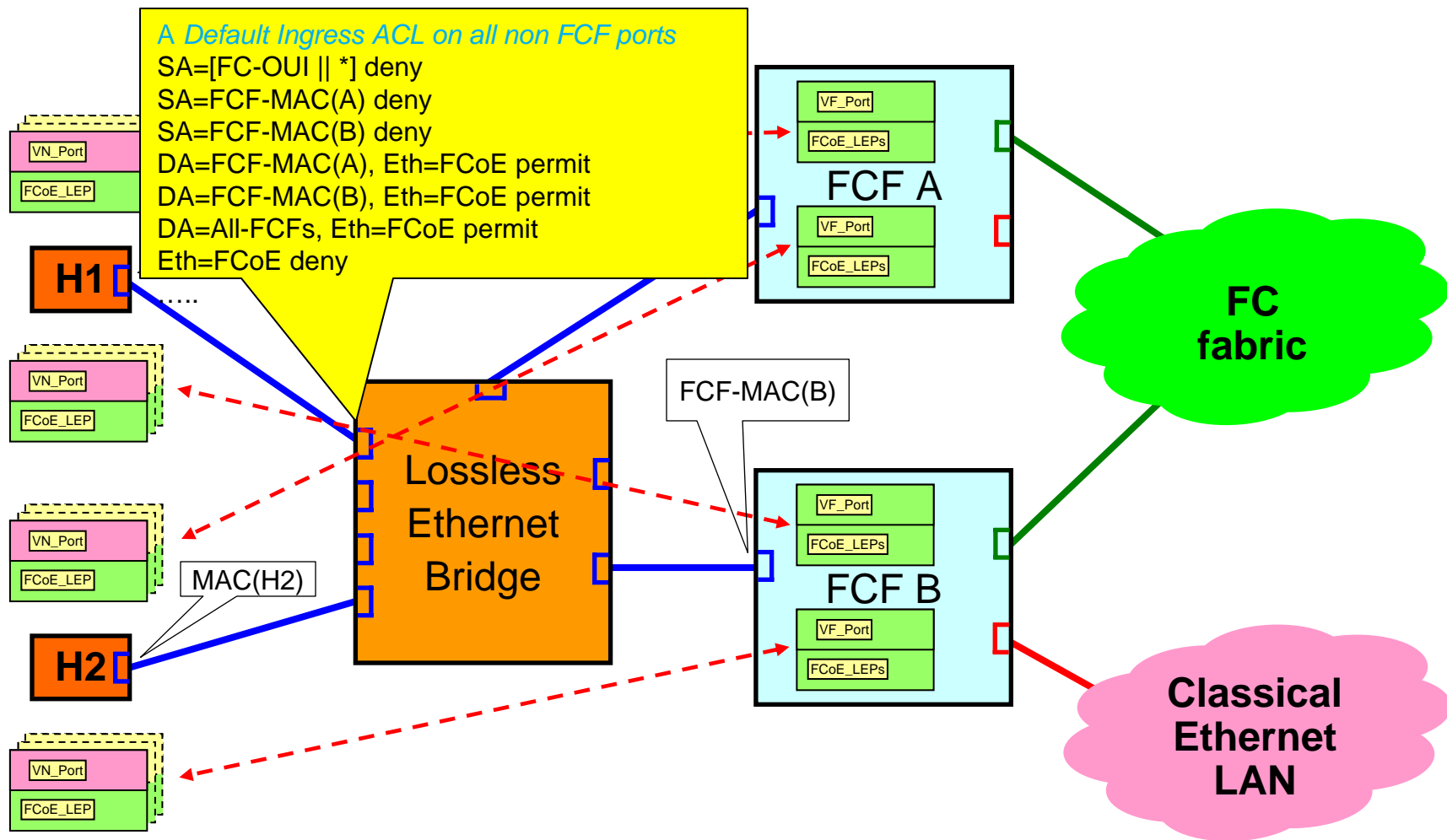
A Discovery protocol is needed for a generic FCoE network.

The configuration has some high availability aspects.



FCF-MAC(x): A MAC address of a Lossless Ethernet port of FCF x  
 MAC(y): A MAC address of a Lossless Ethernet port of ENode y

# Generic Topology ACLs (with Network Provided MAC Addresses)



# Default MAC layer ACL (with Network Provided MAC Addresses)

*Default Ingress ACL on all non FCF ports! (as shown in T11/07-546v0)*

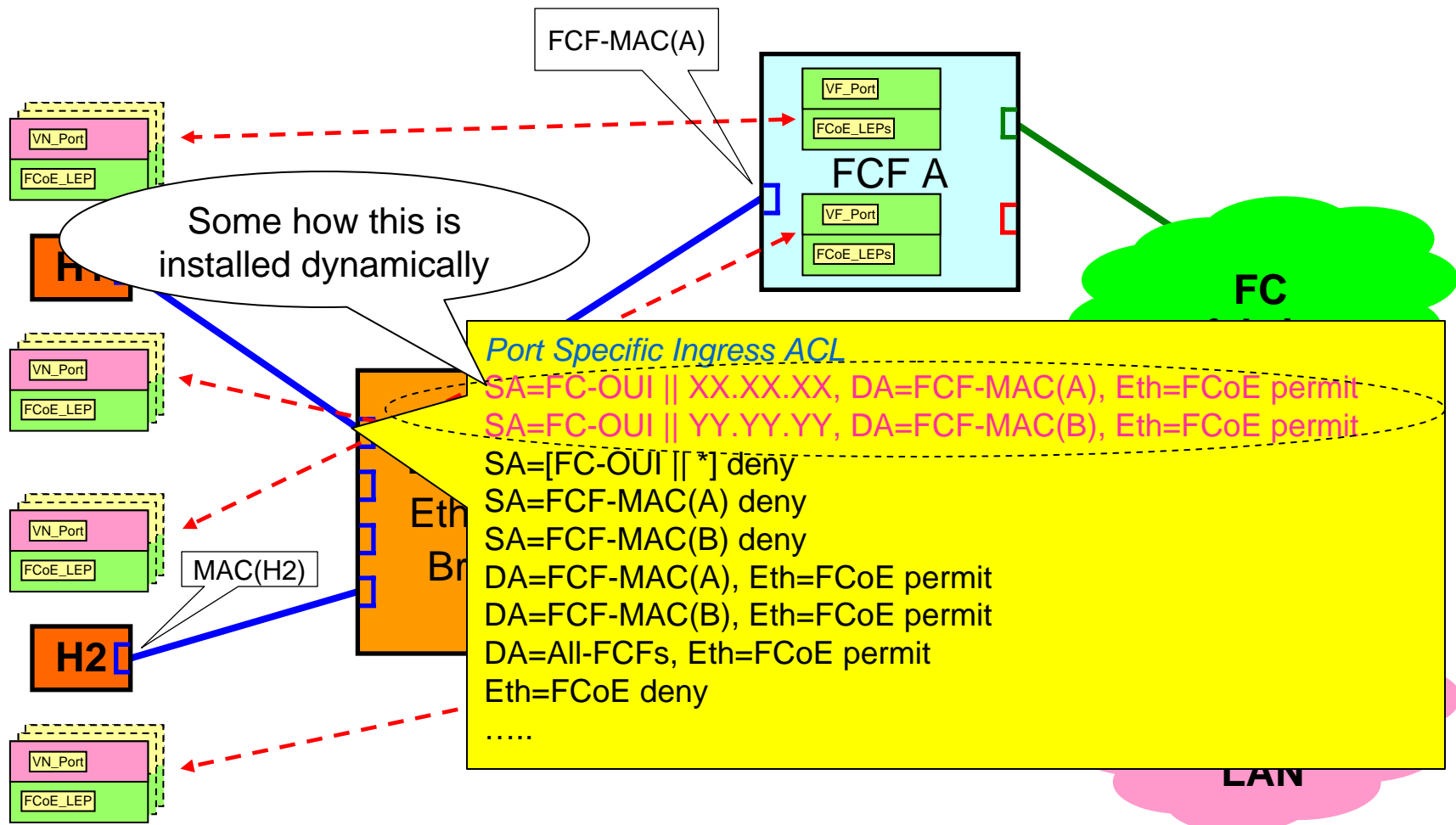
- SA=[FC-OUI || \*] deny {Protect Mapped MAC addresses }
- SA=FCF-MAC(A) deny {Protect one of the FCF burned-in MAC addresses}
- SA=FCF-MAC(B) deny {Protect one of the FCF burned-in MAC addresses}

**... Note: A Rogue could spoof the Switch Learning on SA & impact Discovery with a normal Ping**

- DA=FCF-MAC(A), Eth=FCoE permit {Enables FLOGI }
- DA=FCF-MAC(B), Eth=FCoE permit {Enables FLOGI }
- DA=All-FCFs, Eth=FCoE permit {Enables Discovery Solicitation}
- Eth=FCoE deny {Disable all other FCoE traffic }
- ..... {Non FCoE related entries (e.g. IPv4) }



# Generic Topology Dynamic ACLs (with Network Provided MAC Addresses)



# Dynamically updated MAC layer ACL (with Network Provided MAC Addresses)

Port specific Ingress ACLs (somehow created dynamically by the FCFs) (as shown in T11/07-546v0)

SA=FC-OUI || XX.XX.XX, DA=FCF-MAC(A), Eth=FCoE permit {Dynamically added to top of ACLs}

SA=FC-OUI || YY.YY.YY, DA=FCF-MAC(B), Eth=FCoE permit {Dynamically added to top of ACLs}

... **Note: there could be 100s of those!!! One for each FLOGI & FDISC**

SA=[FC-OUI || \*] deny {Protect Mapped MAC addresses}

SA=FCF-MAC(A) deny {Protect “n” FCF burned-in MAC addresses}

...

SA=FCF-MAC(B) deny {Protect “n” FCF burned-in MAC addresses}

... **Note: A Rogue could spoof the Switch Learning on SA & impact Discovery**

DA=FCF-MAC(A), Eth=FCoE permit {Enables “n” FLOGIs / FDISCs}

...

DA=FCF-MAC(B), Eth=FCoE permit {Enables “n” FLOGIs/ FDISCs}

...

DA=All-FCFs, Eth=FCoE permit {Enables Discovery Solicitation}

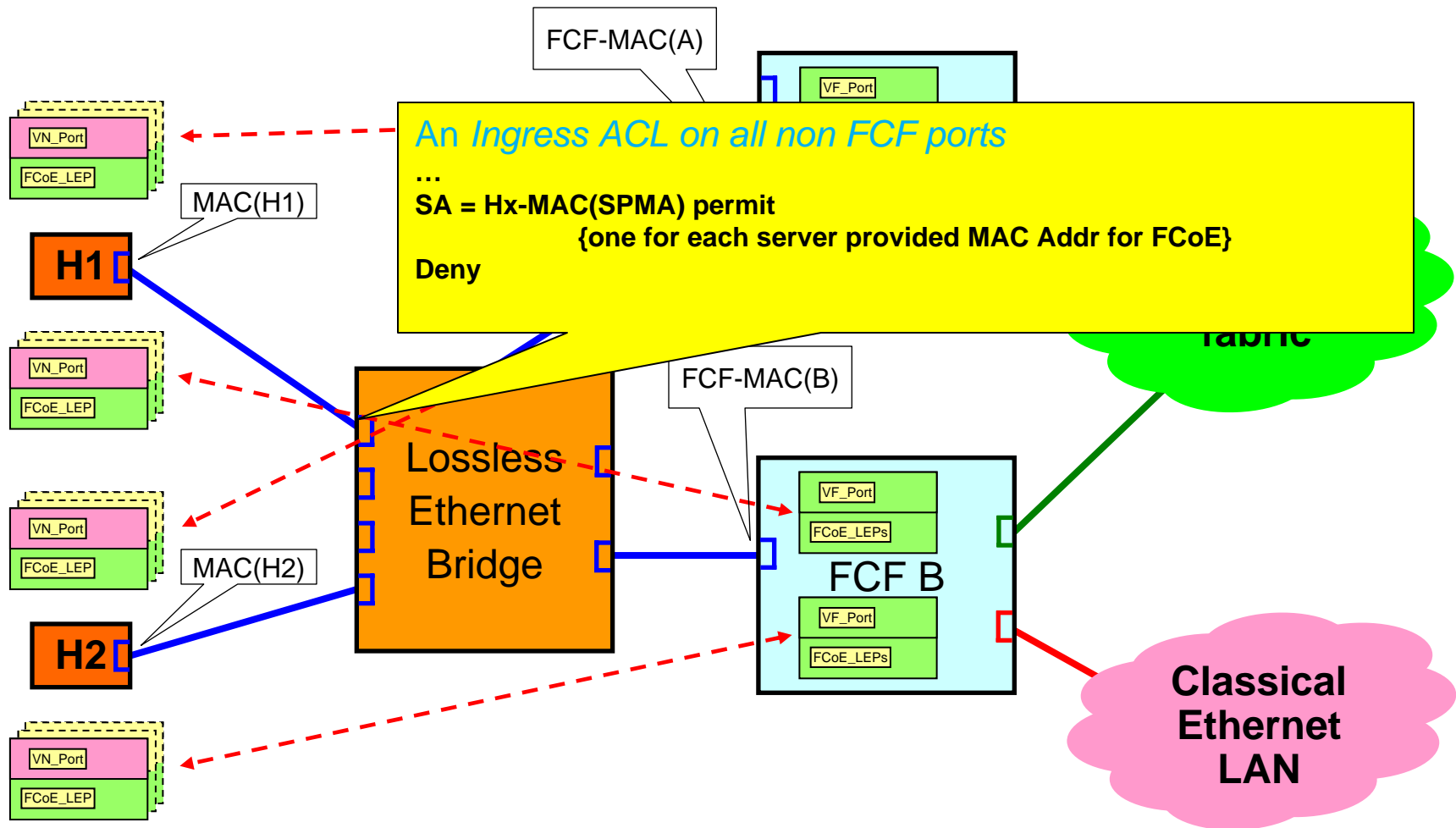
Eth=FCoE deny {Disable all other FCoE traffic}

.....

{Non FCoE related entries (e.g. IPv4)}



# Generic Topology (simple) ACLs (with Server Provided MAC Addresses [SPMA] )



# The really simple Ethernet Switch ACL (with Server Provided MAC Addresses [SPMA] )

*Ingress ACL on all non FCF ports!*

... {Other normal ACLs }

SA = Hx-MAC(SPMA\*) permit

{one for each Server Provided MAC Addr (SPMA), for FCoE, assigned to that switch port}

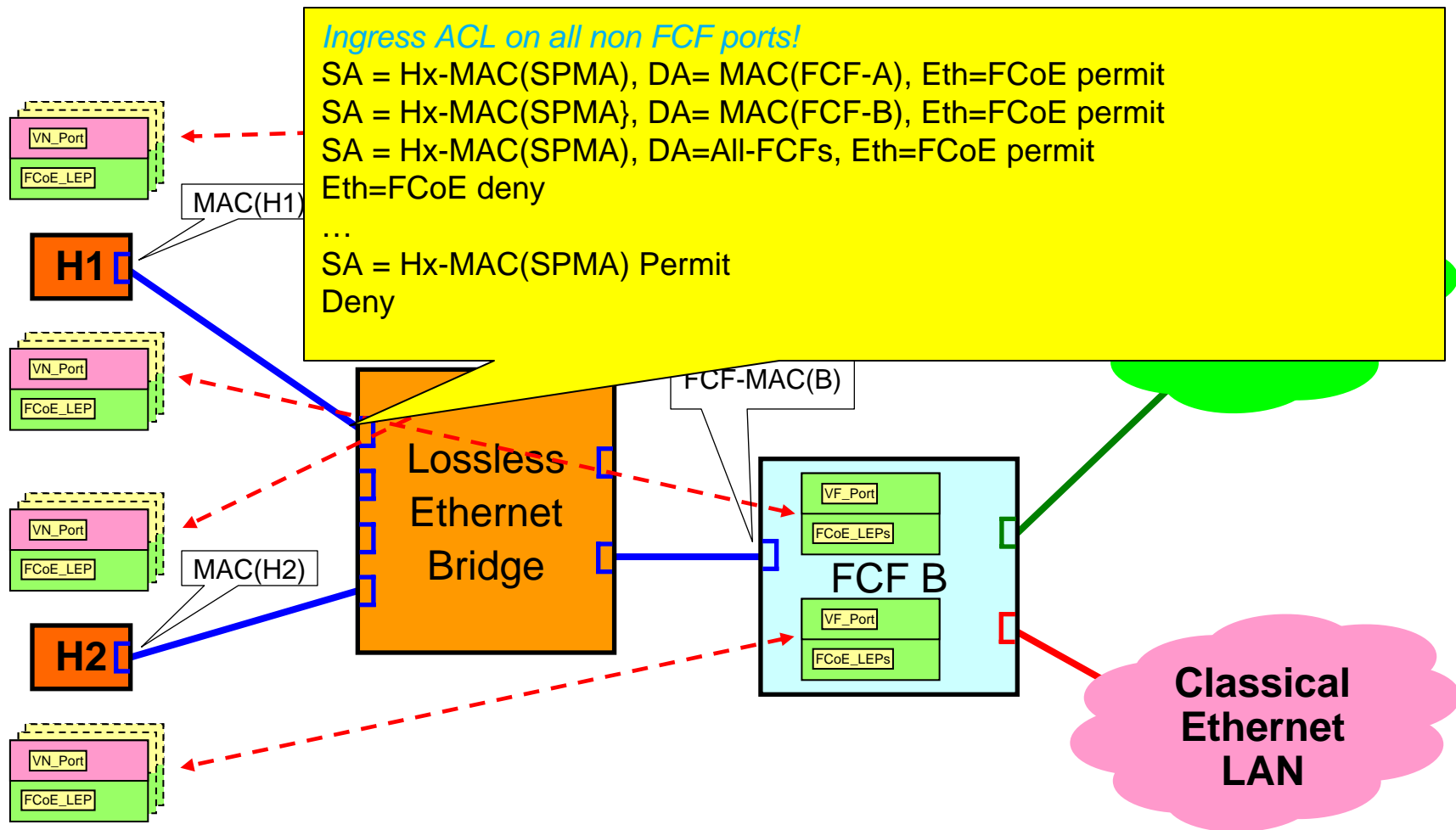
Deny

**Note: The above assumes that if the SA is valid, other stuff is OK**

**\*SPMA = Server Provided MAC Addr**



# Generic Topology & more sophisticated ACLs (with Server Provided MAC Addresses [SPMA] )



# A More sophisticated Ethernet Switch ACL (with Server Provided MAC Addresses [SPMA] )

*Ingress ACL on all non FCF ports!*

SA = Hx-MAC(SPMA), DA= MAC(FCF-A), Eth=FCoE permit {Permits access to FCF-A}

SA = Hx-MAC(SPMA), DA= MAC(FCF-B), Eth=FCoE permit {Permits access to FCF-B}

SA = Hx-MAC(SPMA), DA=All-FCFs, Eth=FCoE permit {Permits discovery of FCFs}

Eth=FCoE deny {Disable other FCoE access}

... {Other normal ACLs}

SA = Hx-MAC(SPMA) Permit {Permits any xmits from SA}

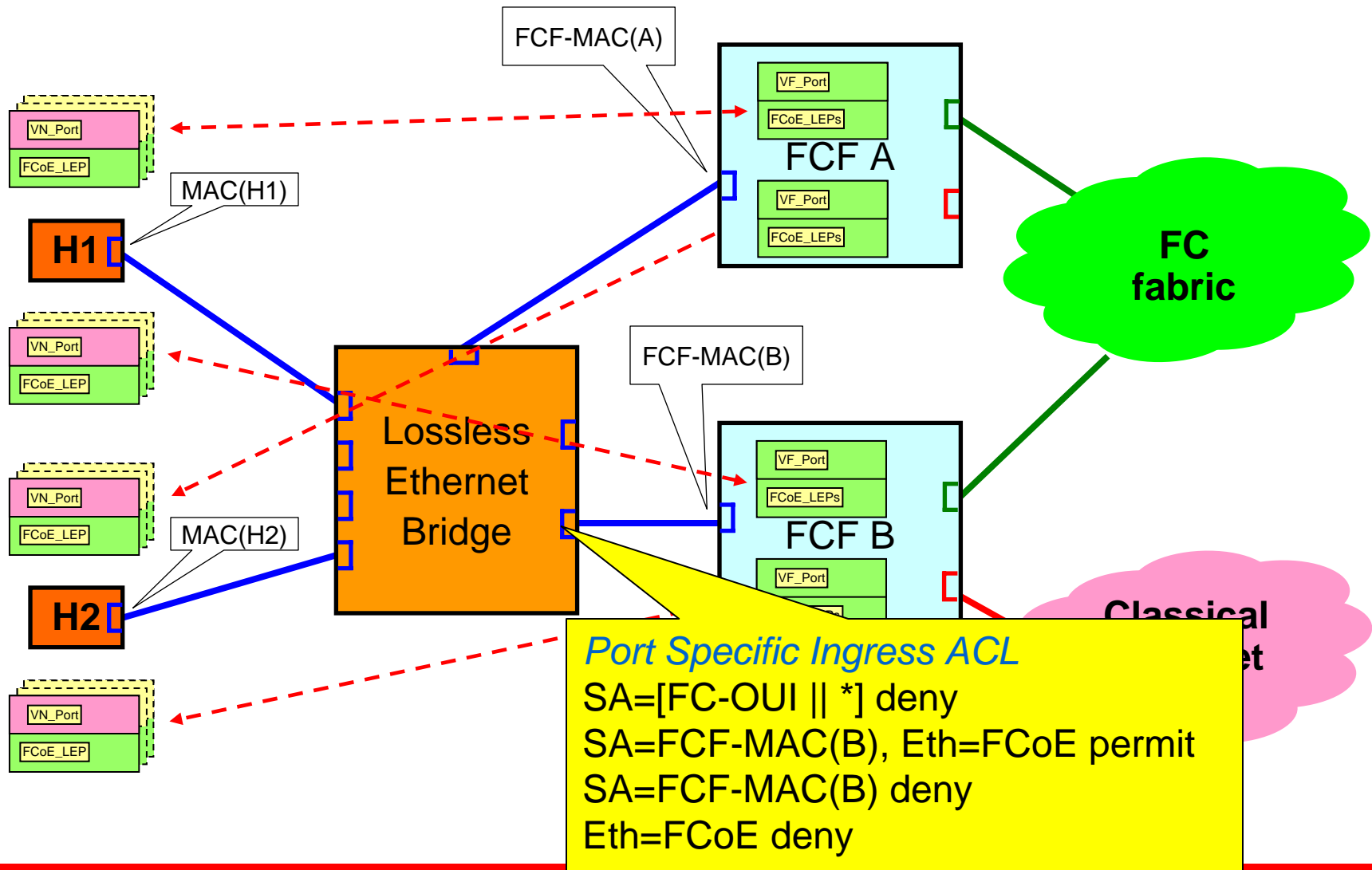
Deny {Prevents use of other SA}

**Note: The above makes sure that Ethernet Rogues do not get access to the “Data Center” Ethernet switches and confuse the learning in the switches**

**Fibre Channel ACLs and Zoning is part of the FC layer in FCoE not Ethernet**



# Ingress ACLs for FCF Ports (with Network Provided MAC addresses)



# Default Ingress ACLs for FCF Ports (with Network Provided MAC addresses)

*Port Specific Ingress ACL (as shown in T11/07-546v0)*

SA=[FC-OUI || \*] deny {Denys all incoming Post FLOGI operations}

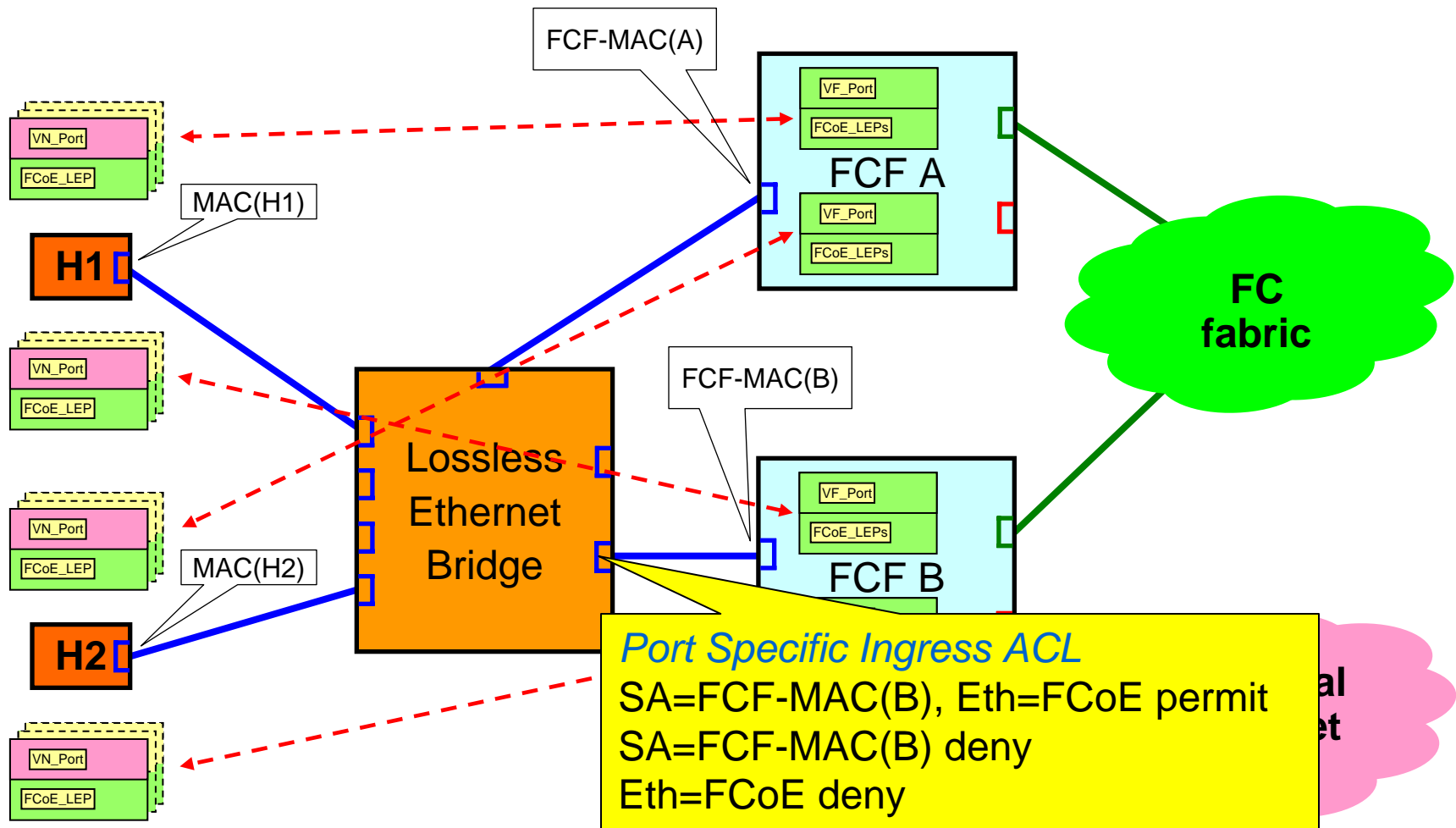
SA=FCF-MAC(B), Eth=FCoE permit {Permits FCoE operation from that SA}

SA=FCF-MAC(B) deny {Prevents any non FCoE with that SA}

Eth=FCoE deny {Prevents any other FCoE operations}



# Ingress ACLs for FCF Ports (with Server Provided MAC addresses)



# Default Ingress ACLs for FCF Ports (with Server Provided MAC addresses)

## *Port Specific Ingress ACL*

SA=FCF-MAC(B), Eth=FCoE permit

{Permits FCoE operations from that SA}

SA=FCF-MAC(B) deny

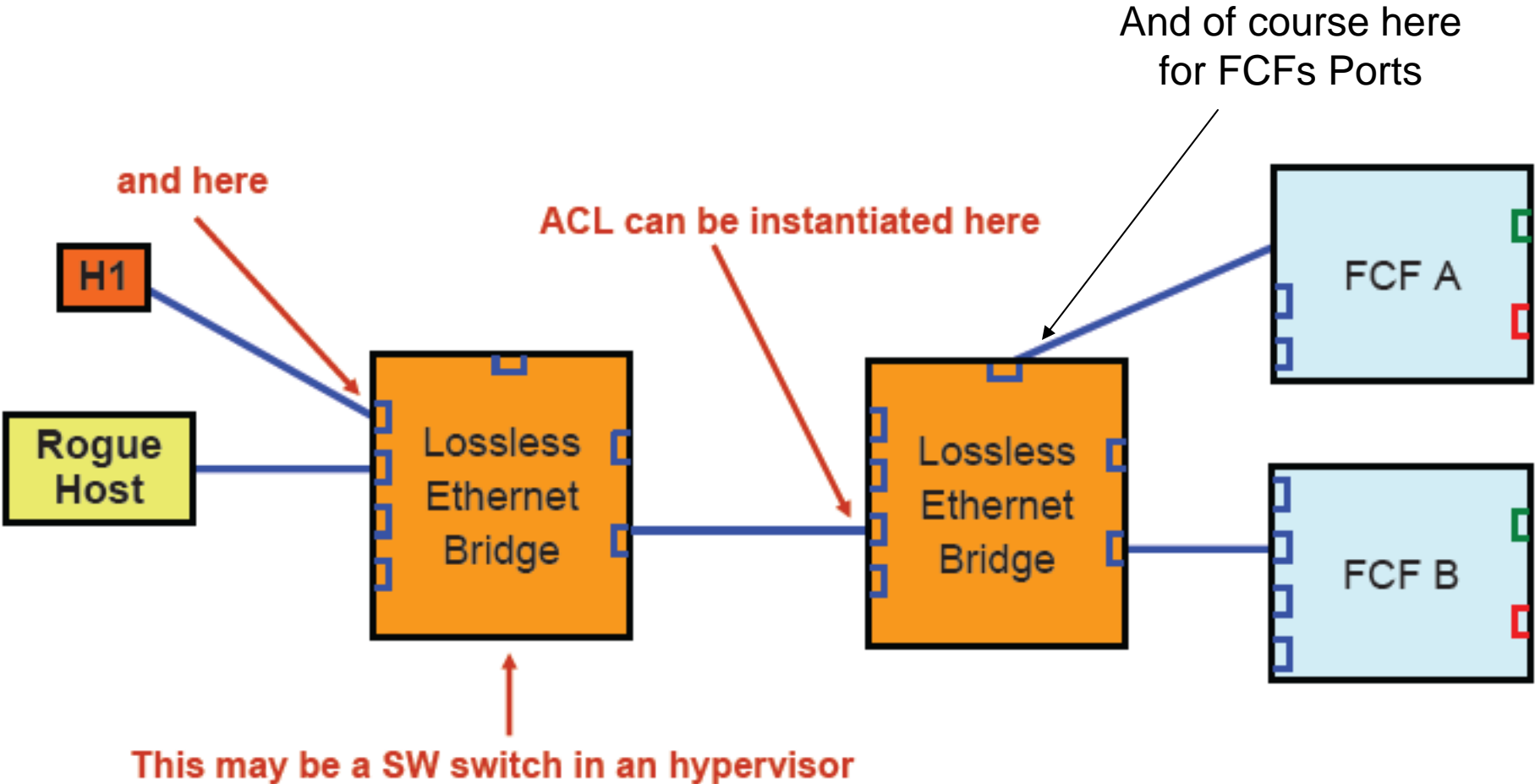
{Prevents any non FCoE with that SA}

Eth=FCoE deny

{Prevents any other FCoE operations}



# Double Ethernet Bridges



# Analysis

<b>Characteristic</b>	<b>Server-provisioned MAC (“Burnt-in”)</b>	<b>Fabric-provisioned MAC (“Mapped”)</b>
Rogue spoofing of switch learning	Denied	Possible
Dynamic ACL modification	Not required	Required, but no standard mechanism
Number of lines of ACL per ingress	Small number	May be large number with Dynamic updates
Static administration	Allowed	Requires subsequent dynamic modification
Able to use current Ethernet Switches	Yes	No; needs new ACL protocol, or Snooping for ACL et.al.
Compatible with 802.1x	Yes	No; requires dynamic update to Radius Server



# Summary

Comparing the Rogue preventing ACLs needed with Server Provided MAC Addresses to the Rogue preventing ACLs needed with Network Provided MAC Addresses, one finds::

The ACLs used with Server Provided MAC Address are:

- Simpler
- Statically defined
  - Can be applied by normal Administration processes
  - Can have Management utilities build and apply the ACL

There is no need to invent a new ACL establishment protocol

There is no need to create a new snooping capability, etc.

All the Ethernet Switch Processes are the same as today

