

# Model – supporting sketches
























Nigel Davis (Ciena)

20150330

# Additions and changes in this version

- Updated to align with change from SN to FD and SNC to FC

# Key

Alternatives Symbols	Primary Symbol	
		FC [TMF SNC, FDFr, MFDFr, XC (and now FRE/FC)]
		FC (emphasising EndPoints  which supports the pointer to the LTP) [EndPoint is equivalent to aList and zList] of TMF SNC/FDFr (and now EndPoint of FRE/FC)]
		LTP bound to physical port (TMF PTP (and now TPE with physical port))
		LTP without direct physical port that is dependent on another LTP (TMF CTP and now dependent TPE)
		LTP without direct physical port that is not dependent on another LTP (TMF FTP and now TPE that is floating)
		Adapter and or LTP Pool - absorbed into LTP [Is G.805 adaptation function and is Absorbed into TMF PTP/CTP/FTP/TPE]
		Termination function - absorbed into LTP [Is G.805 Trail Termination and is Absorbed into TMF PTP/CTP/FTP/TPE]
		Termination Connection Point (TCP) - absorbed into LTP [Is G.805 TCP and is Absorbed into TMF PTP/CTP/FTP/TPE]
		Connection Point (CP) - absorbed into LTP [Is G.805 CP and is Absorbed into TMF PTP/CTP/FTP/TPE]
		Inverse Multiplex Point (IMP) - absorbed into LTP [Absorbed into TMF PTP/CTP/FTP/TPE]
		ForwardingDomain [TMF MLSN, FlowDomain (and now ForwardingDomain)]
		NE [roughly TMF ME]
		Protection switch in an FC
		FC decomposition (half switch) showing common point in grey
		An association
		Route structure
		Link (emphasising Link Ends and conceptual relationship to FC)
		Link End



Two protection switches  
that are inverse ganged

CP = Connection Point  
 AP = Access Point  
 TCP = Termination Connection Point  
 TTP = Trail Termination Point  
 CTP = Connection Termination Point  
 PTP = Physical Termination Point  
 LT = Layer Termination  
 TPE = Termination Point Encapsulation

# Derivation of LTP and LP

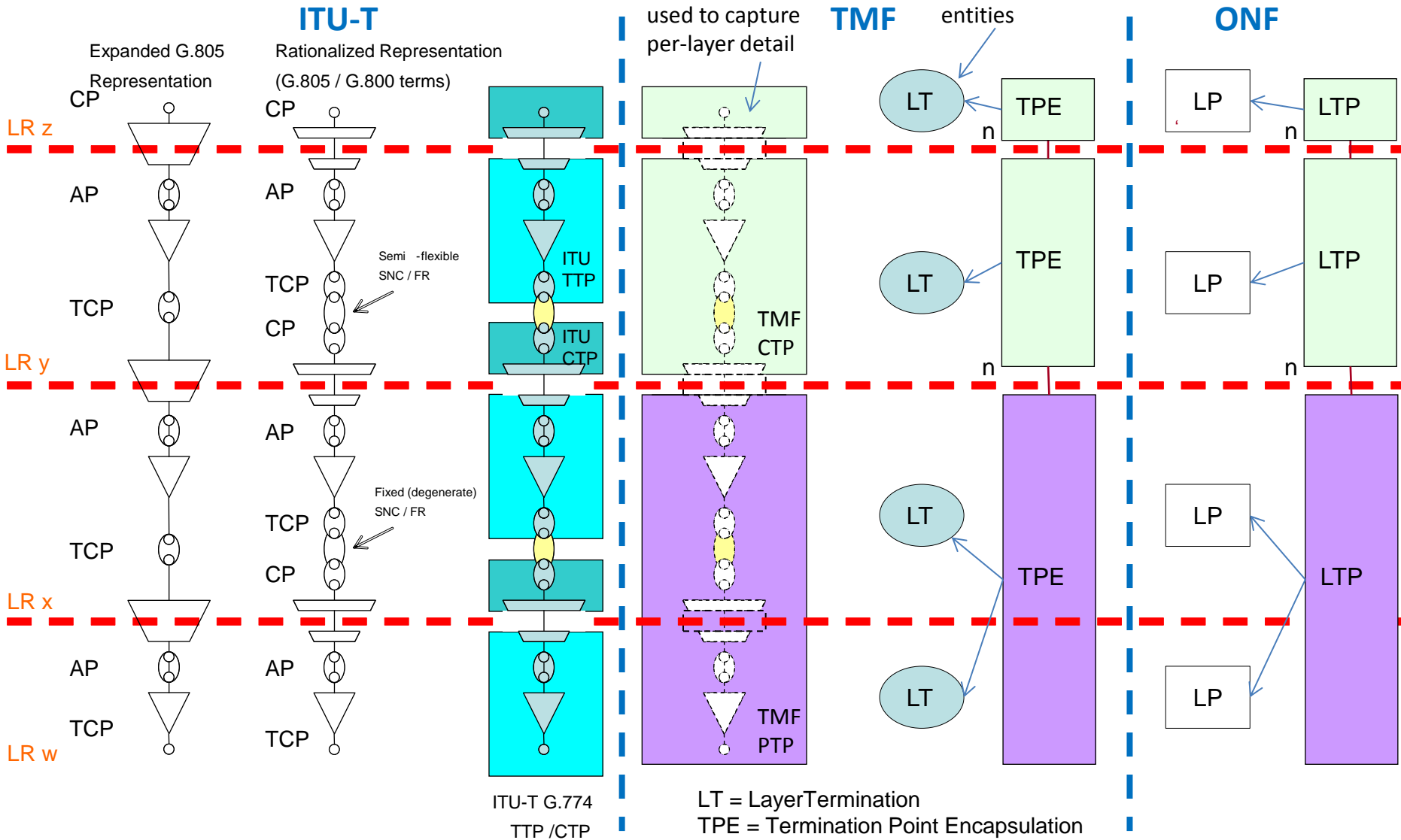
Layer examples  
 LR x = MS  
 LR y = VC4 (flexible)  
 LR z = 140 (flexible)

Layered parameter list used to capture per-layer detail

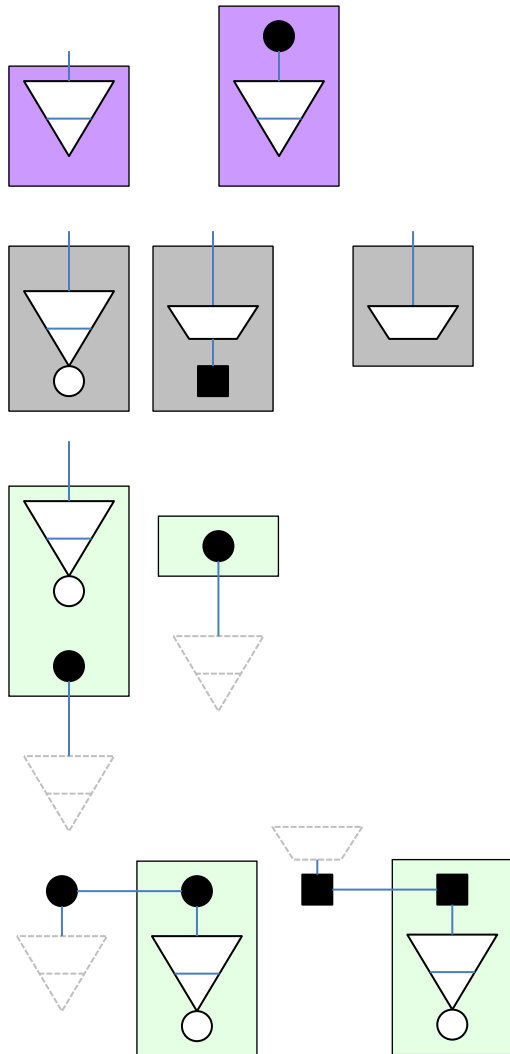
Per-layer detail captured in LT entities

TMF

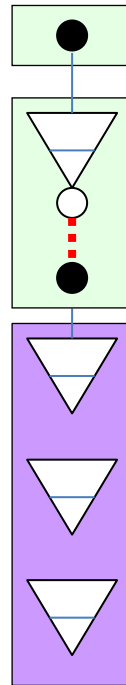
ONF



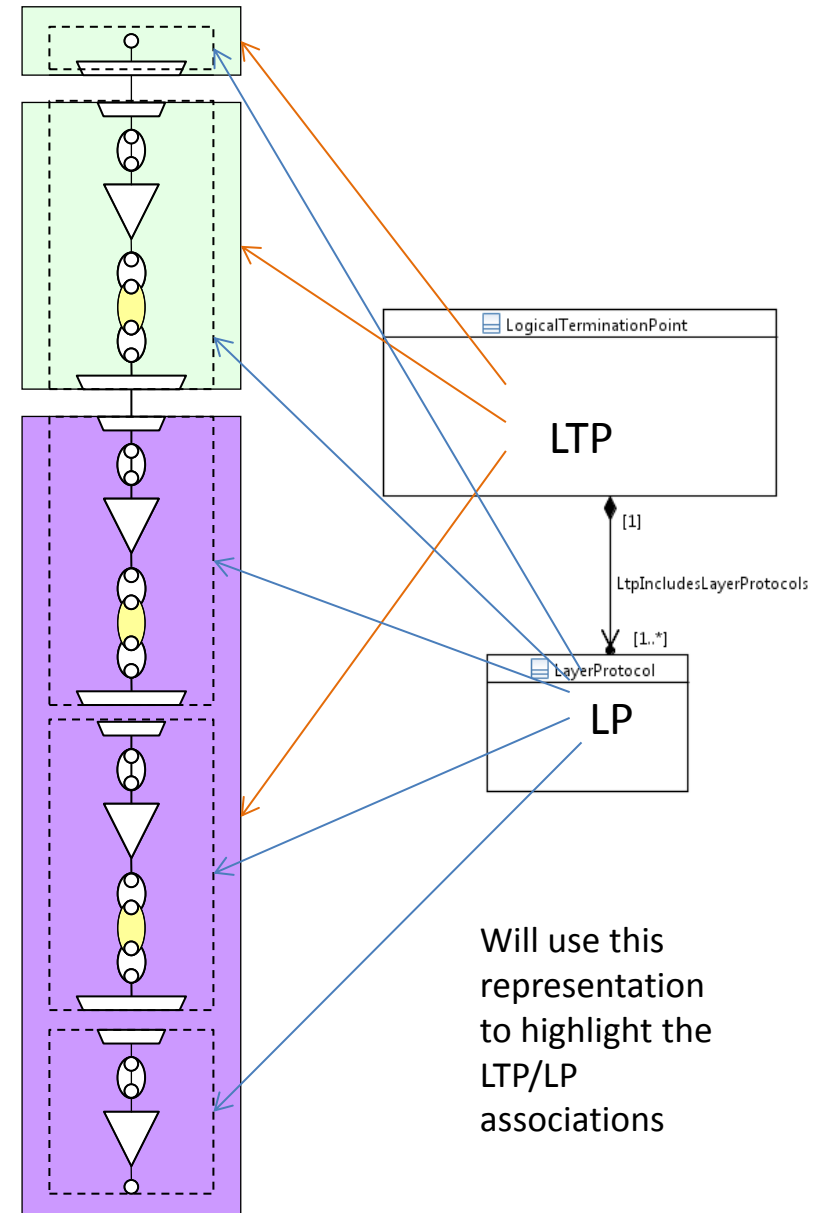
Examples of LTPs (using figures consistent with those used in TM Forum for PTP, CTP and FTP)

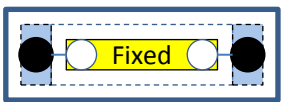


Port with various layers and flexibilities modelled as LTPs (PTPs and CTPs)

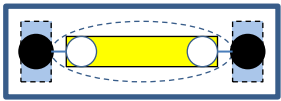


More precise view of port





Dual Port (encapsulating zero length link)



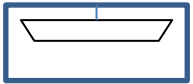
Dual Port Client Mapping  
(note this has n encapsulated CPs)



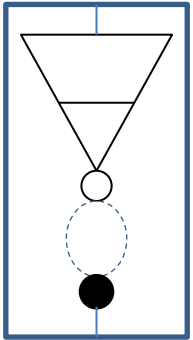
Exposed CP



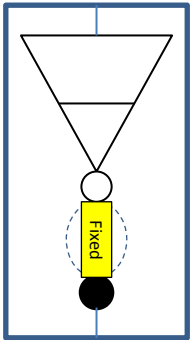
Adapter (with pool)



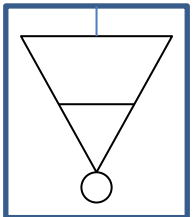
Pool



Full Layer, Flexible with  
Optional Exposed CP



Full Layer, Fixed



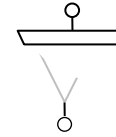
Exposed TCP

# LP cases

Encapsulated  
Structure in LP



FC



CP



TCP



IMP



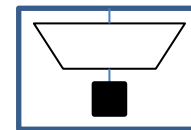
ForwardingDomain



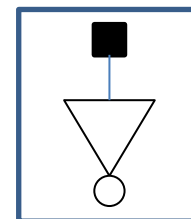
Link



Link End



Concatenating Adapter

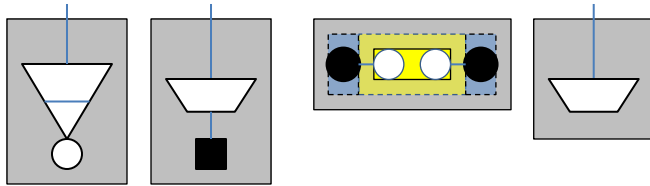
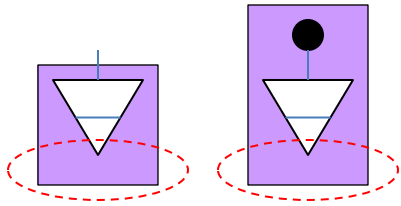


Concatenatable Exposed TCP

Distinguishing characteristic i.e. naming source

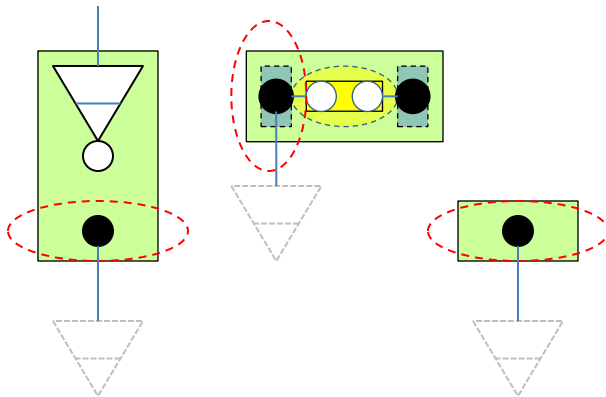
# LTP cases

LTP server access via physical port  
(all PTP cases)

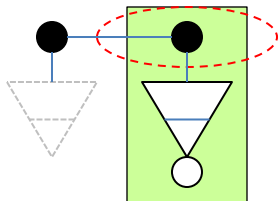


Floating LTP

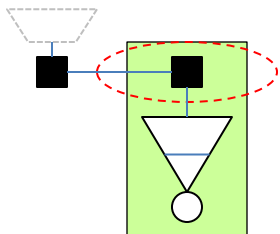
Diagrams show single  
LT versions only  
clearly the LTP can be  
multi-layered



LTP server bound to another LTP client

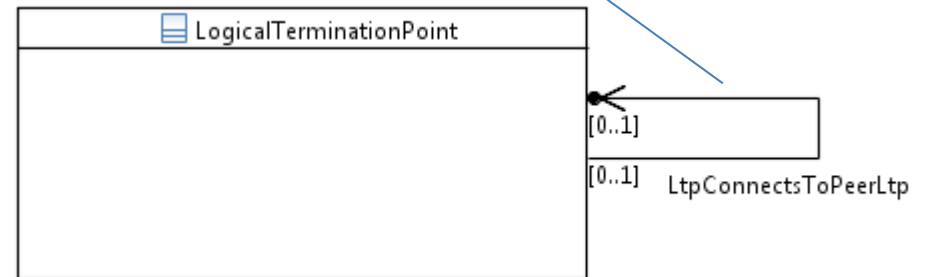
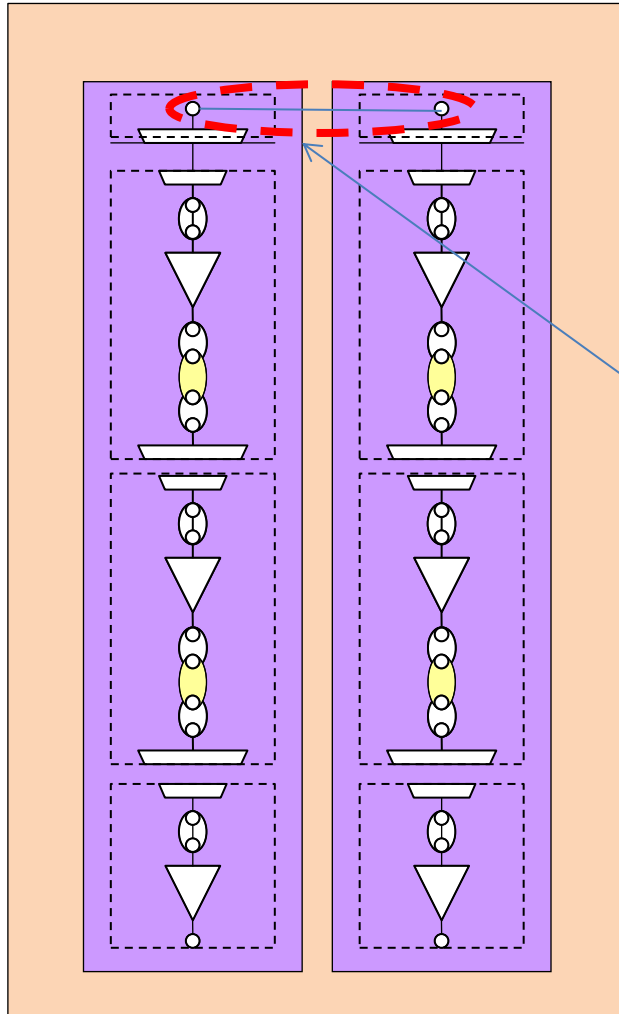


LTP client bound to another LTP client



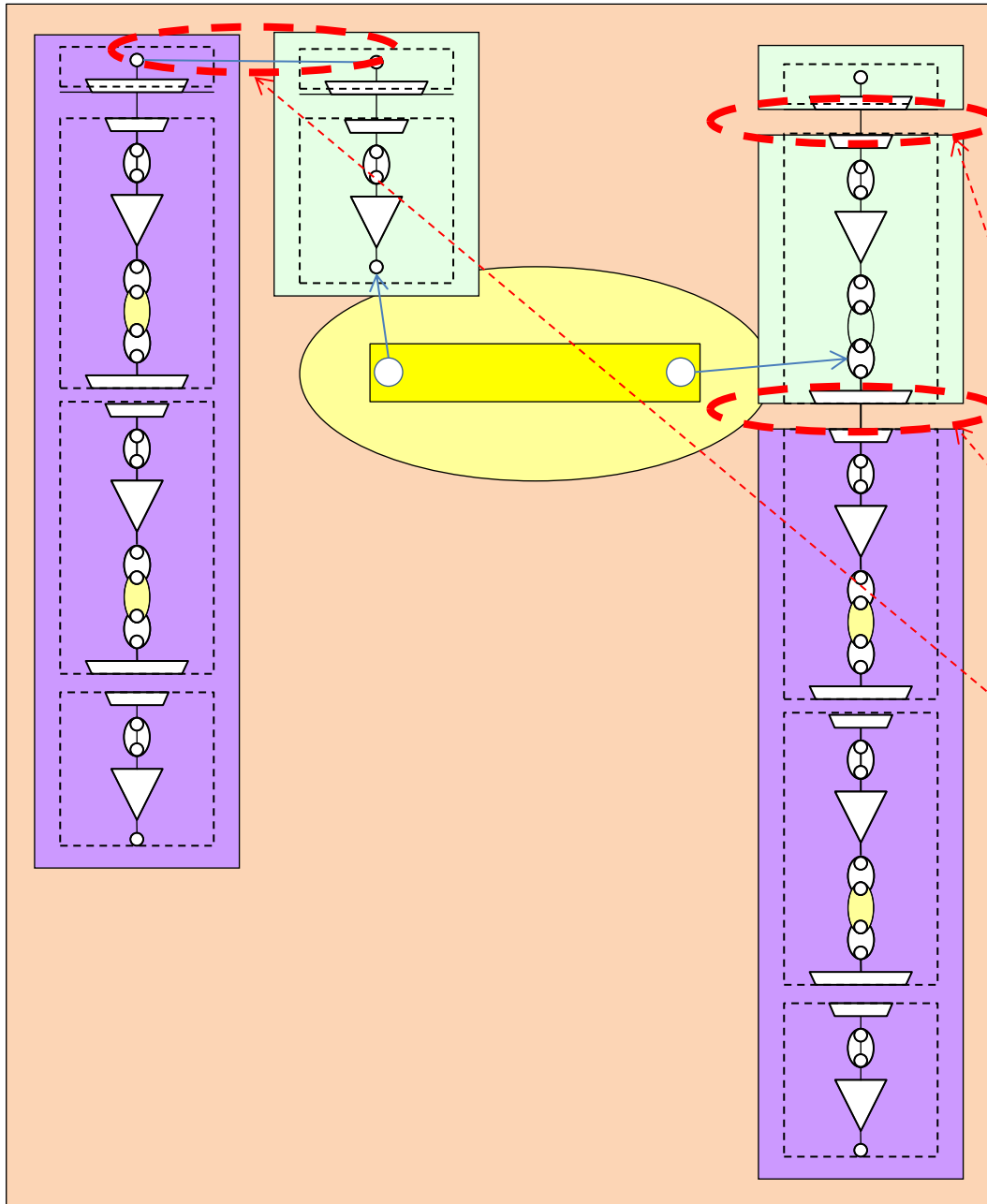
LTP client bound to another LTP server

# Amplifier/Regen case

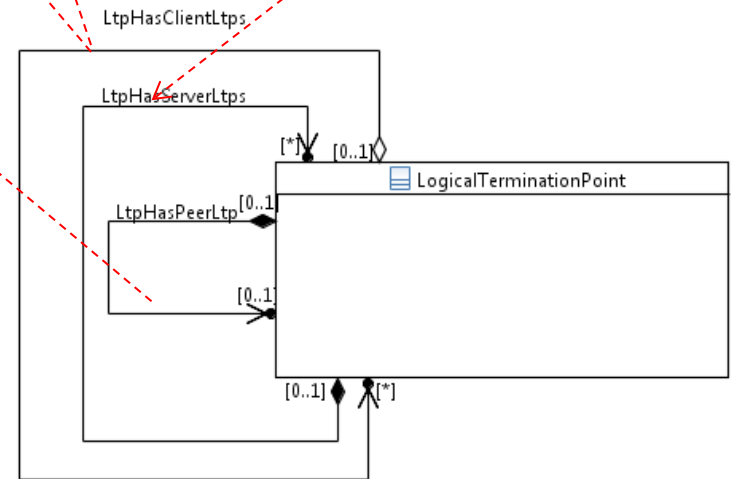


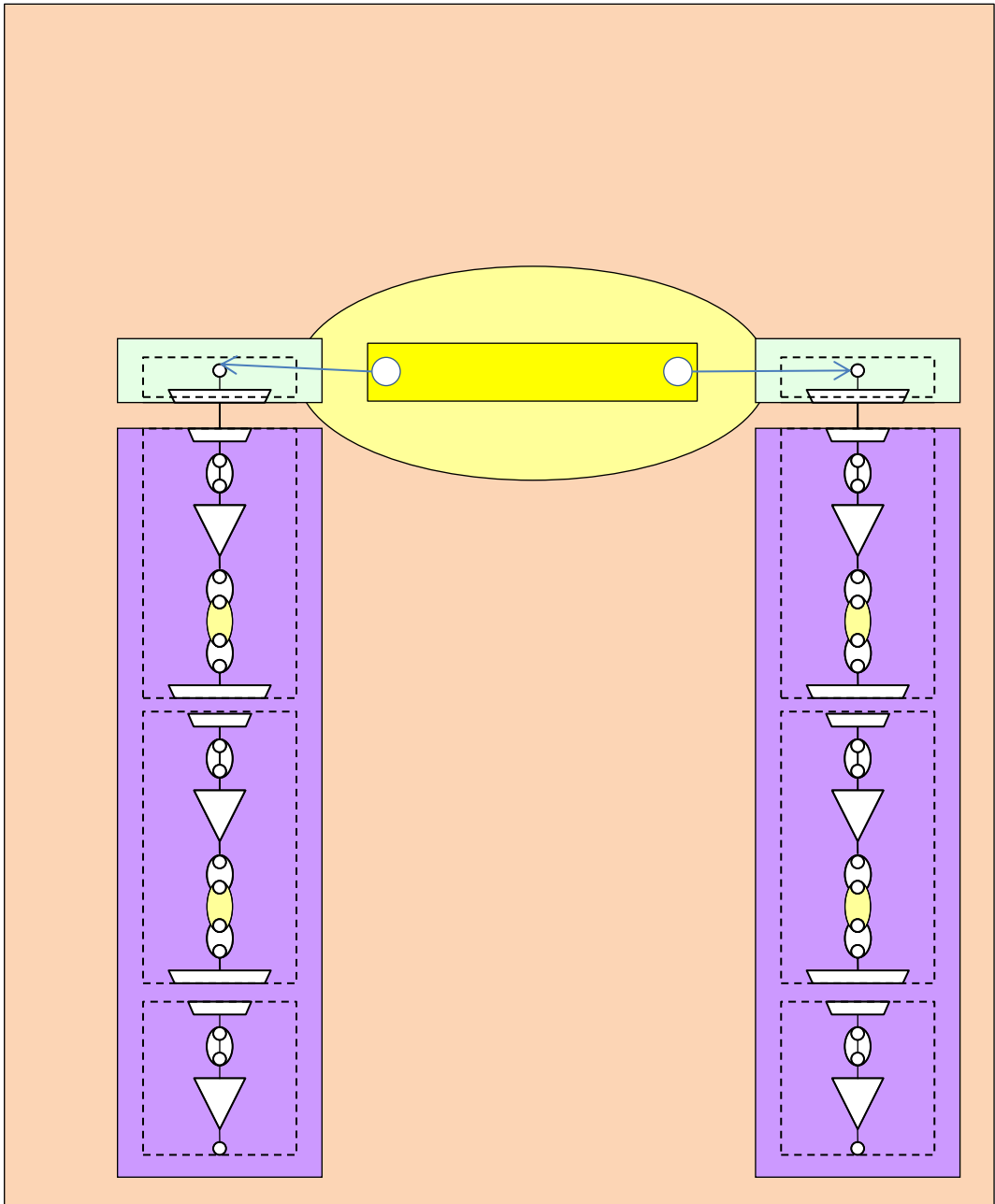


# Trib on multi-layer XC case

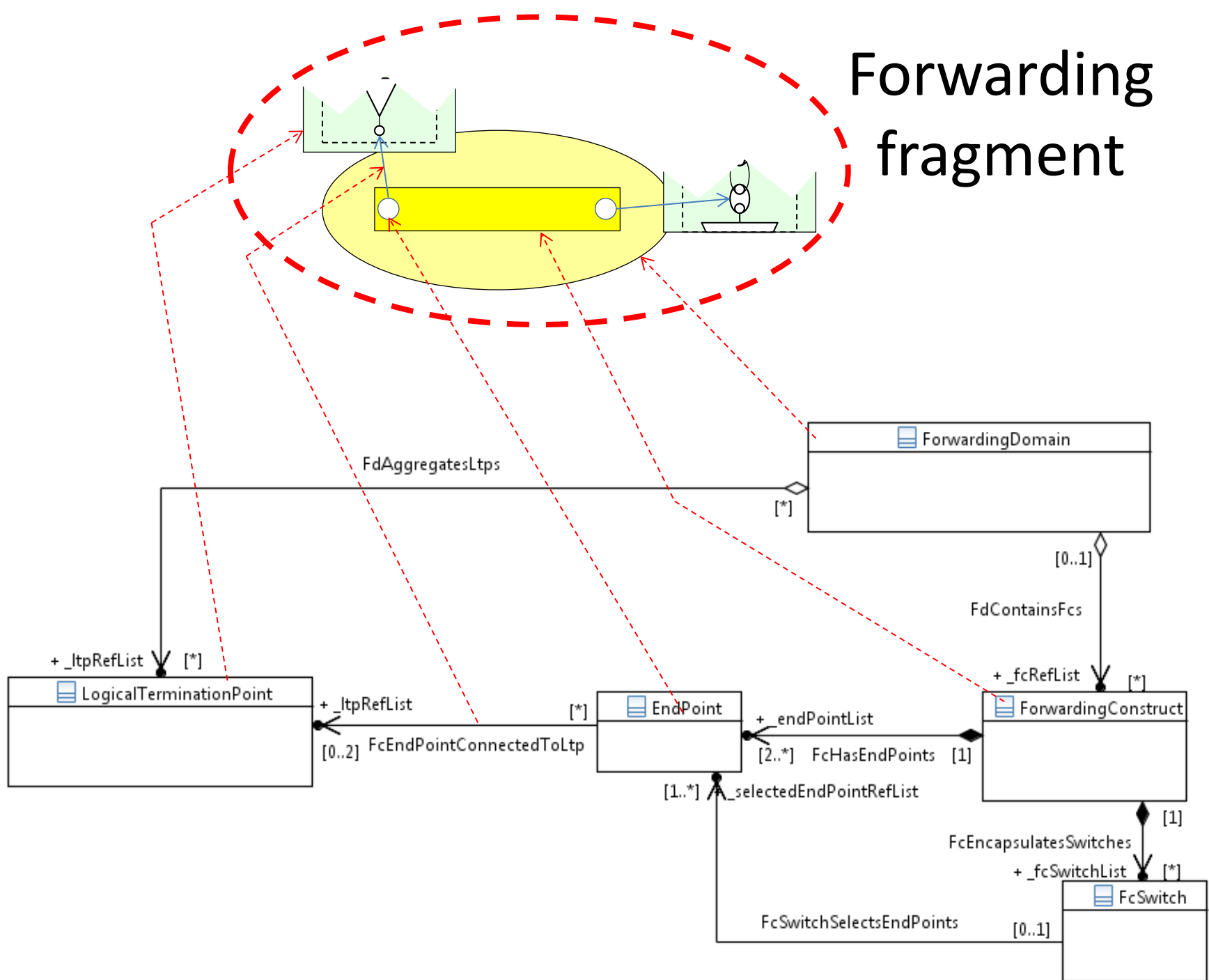


This association is used for inverse multiplexing (inc VCAT)

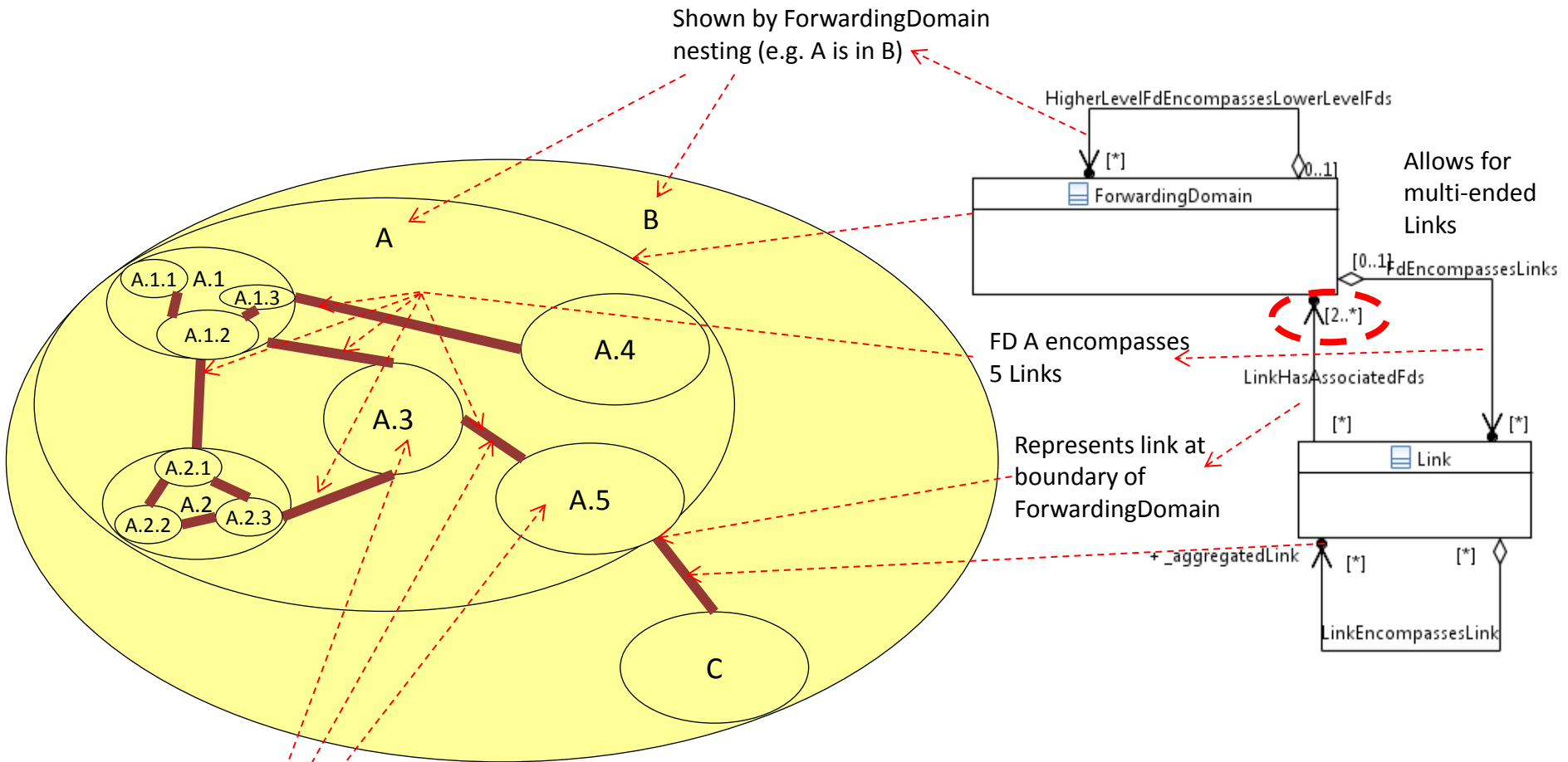




# Forwarding fragment

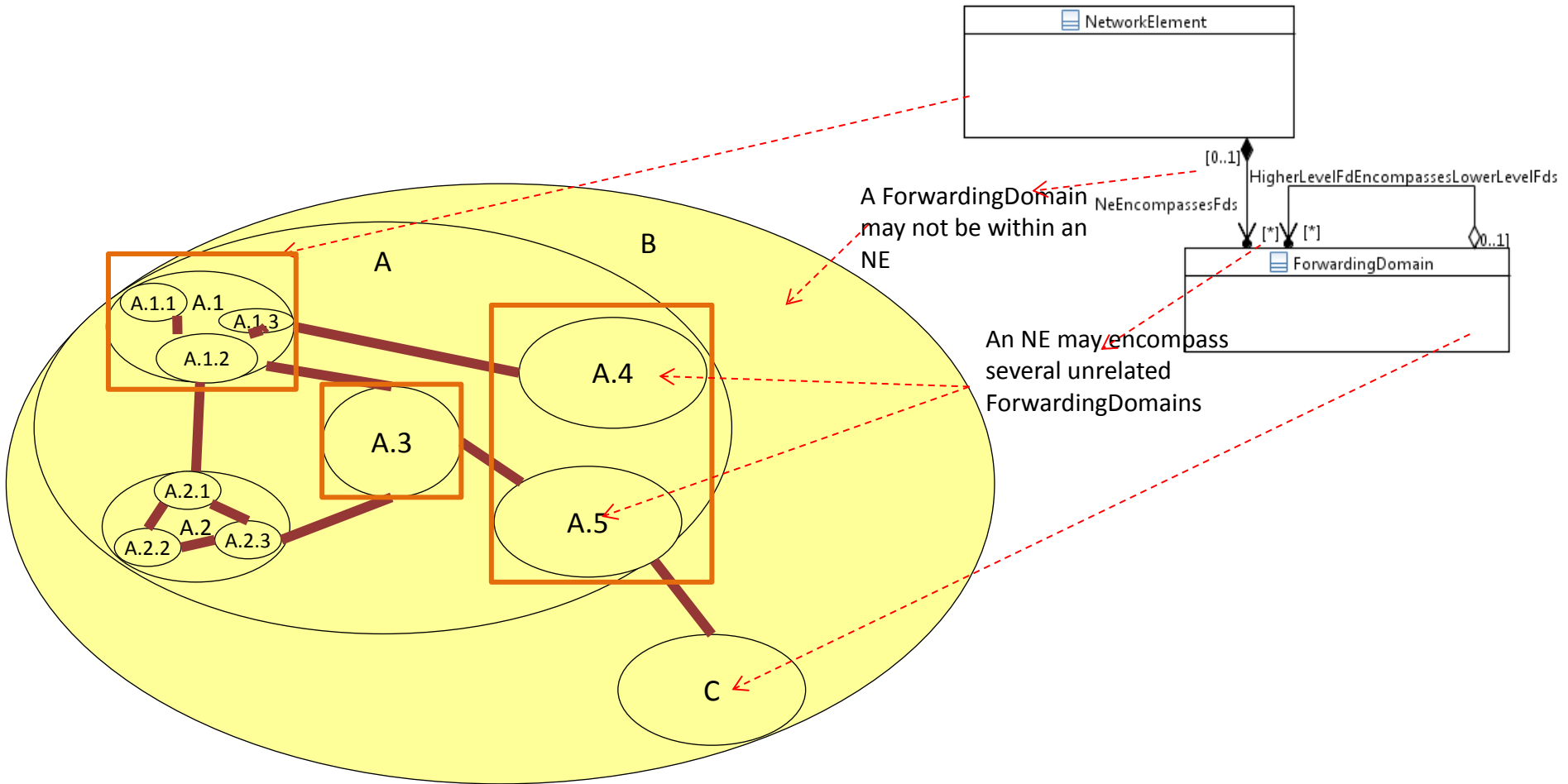


# ForwardingDomain recursion with link



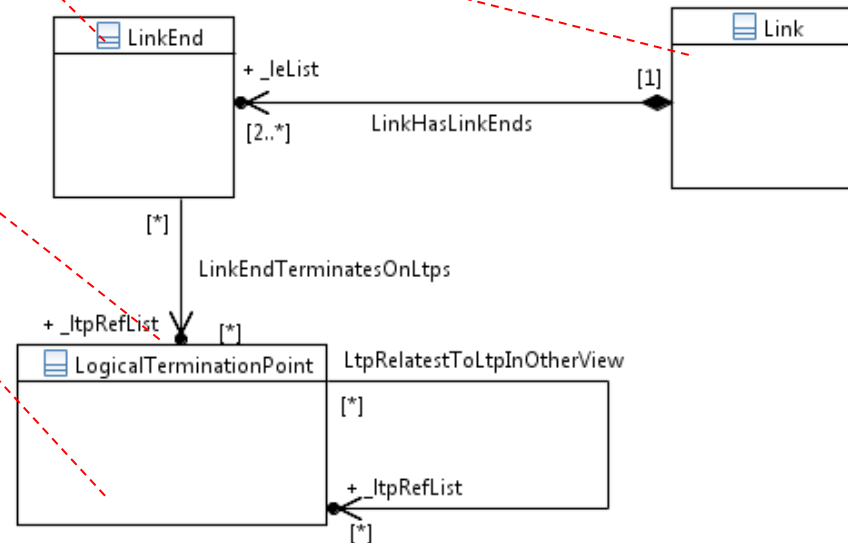
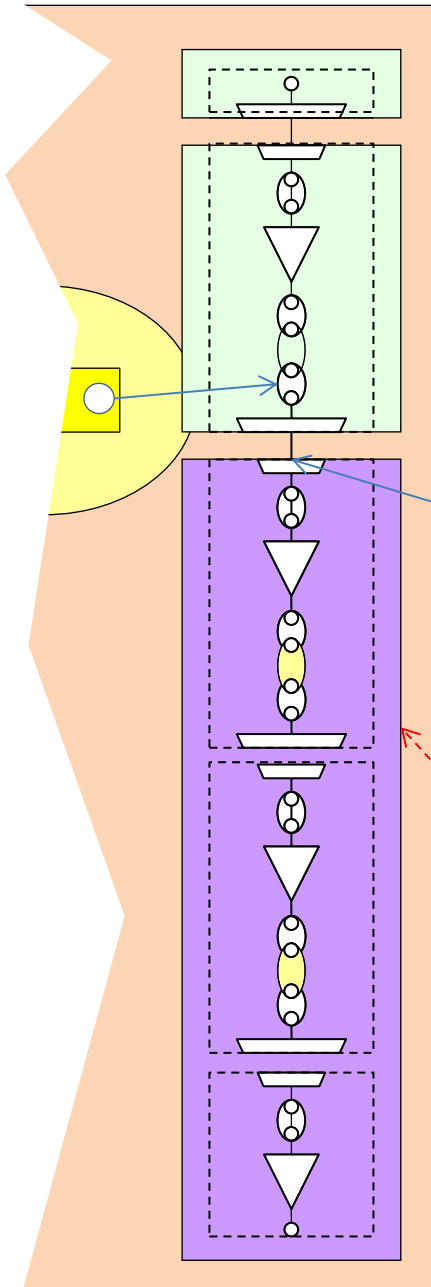
A Link is wholly in a specific ForwardingDomain if all ForwardingDomains that it is associated to are in that ForwardingDomain. Hence no specific association is necessary in the model.

# ForwardingDomain and NE



# LTP Pool considerations

- For cases where there is no physical LTP a “floating” LTP is used.
- Where the situation is fully virtualized a “floating” LTP with only the pooling function is used.
- An inter-view relationship to link contents of a “floating” LTP with the contents of a physically bound LTP is shown (preliminary). This is essentially internally to the controller



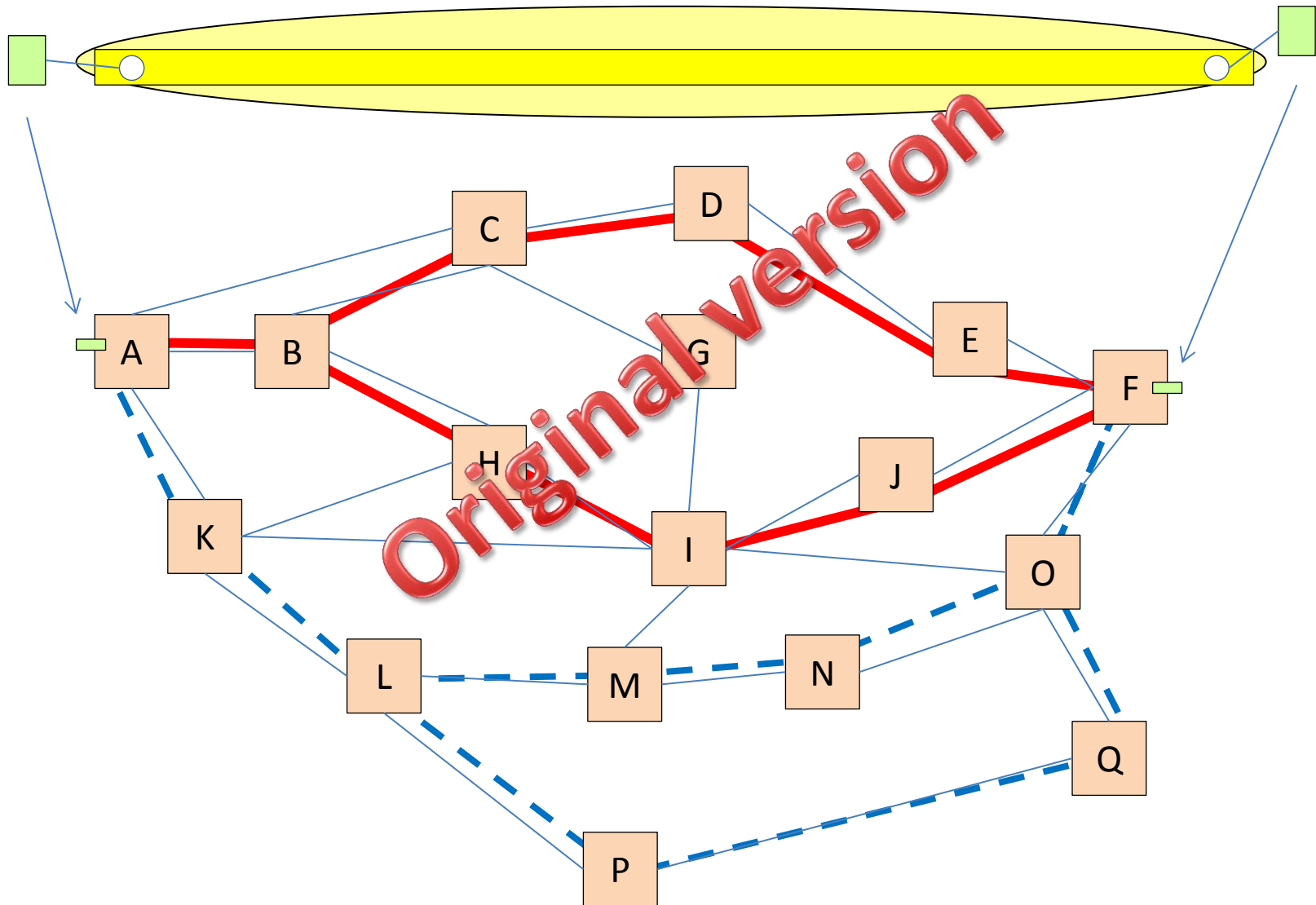
# What is a route?

Two perspectives

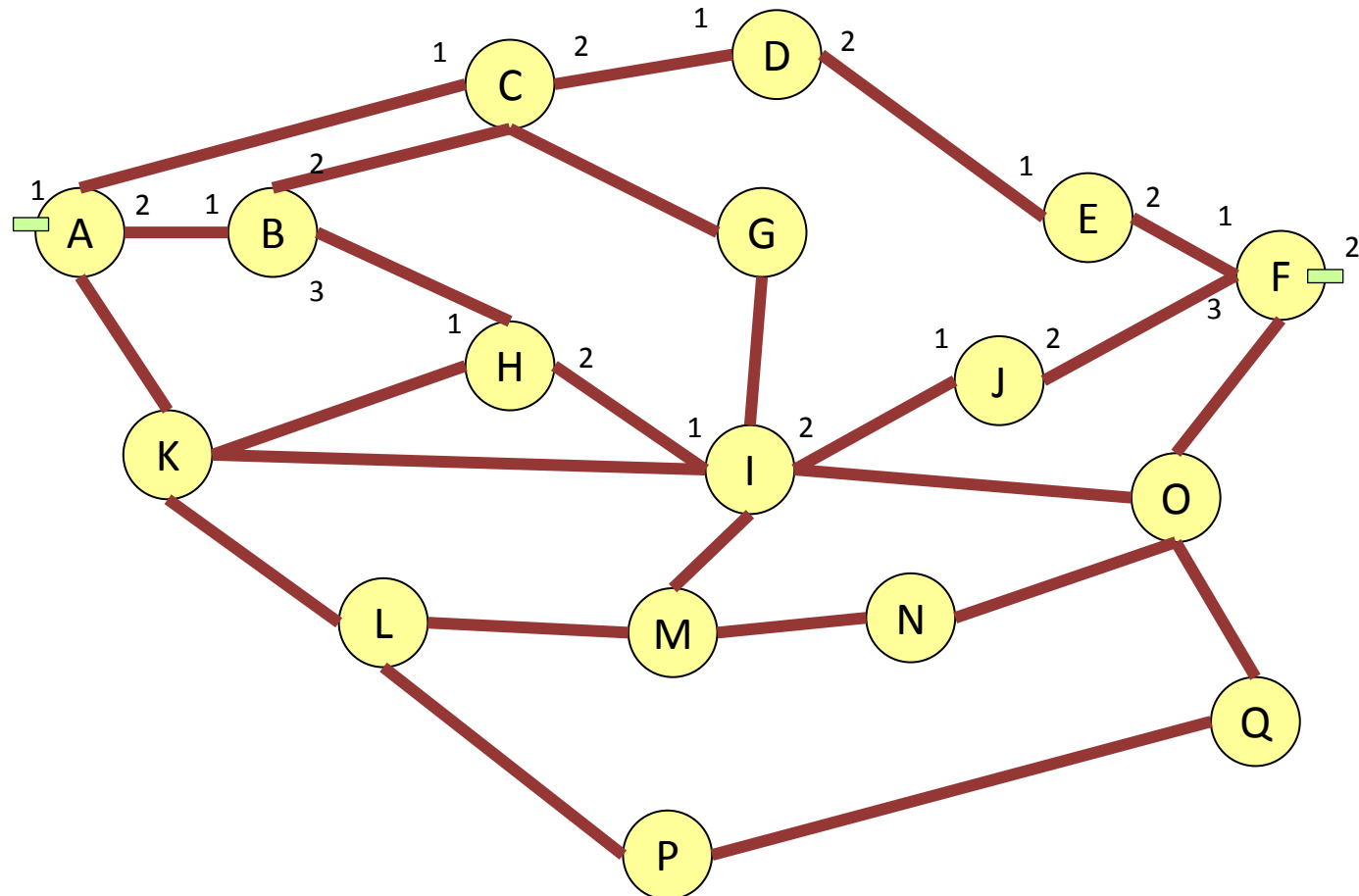
- Two concepts that could be called route
  - Optional Alternative layouts
    - Consider ASON control plane with retain home route where the home route has failed and an alternative is being used
  - Description of a layout
    - Consider a case of protection where there are “two distinct ways” through the network
- Two ways of describing a layout
  - Partition into smaller FCs to encapsulate the alternative
    - For this to be useful (as the FCs are essentially unordered) the Link topology is required. It is expected that any application interested will have the link topology
  - Specific alternative paths of enabled flow



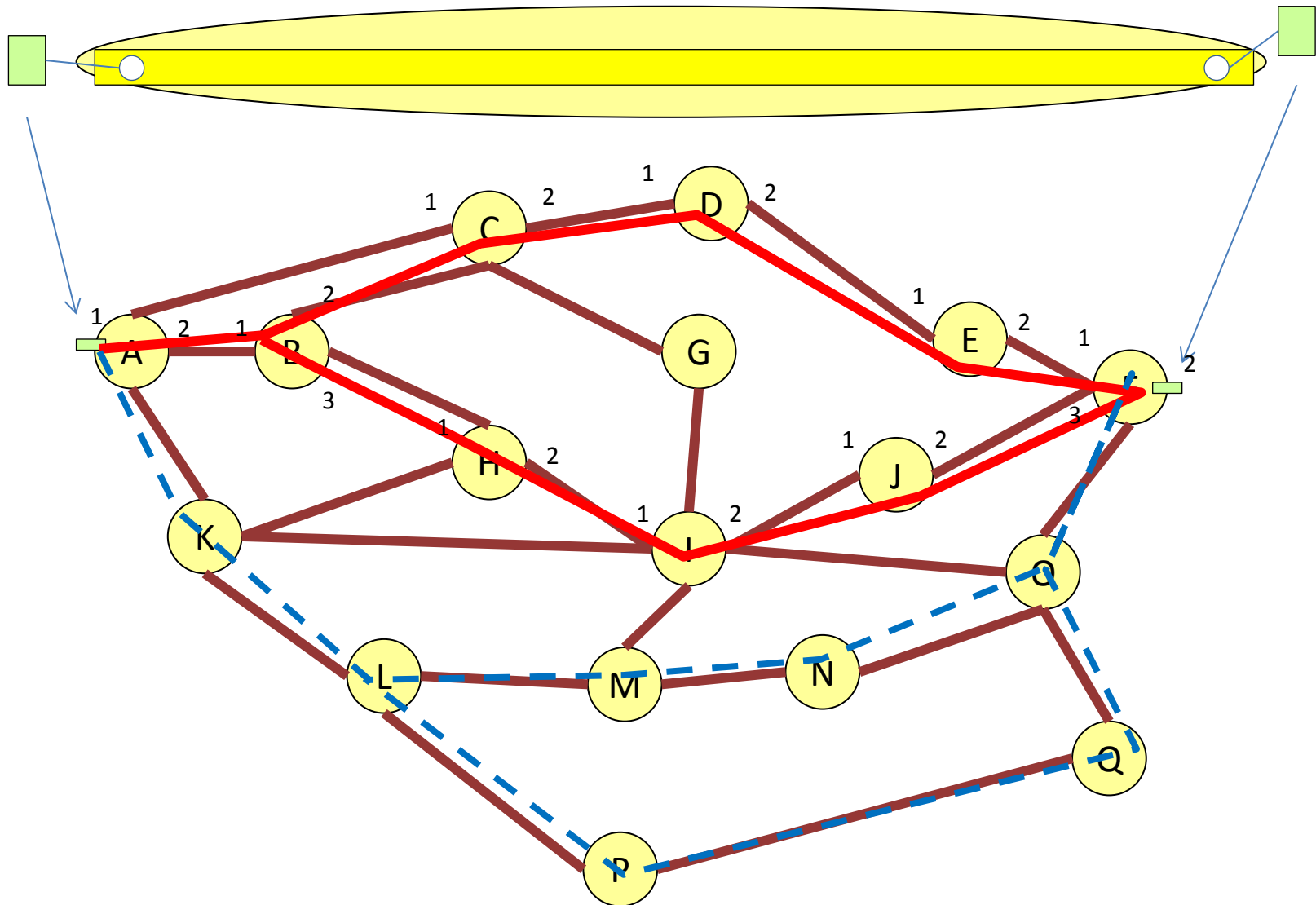
# FC with Optional alternative layouts



# Arbitrary network



# FC with Optional alternative layouts



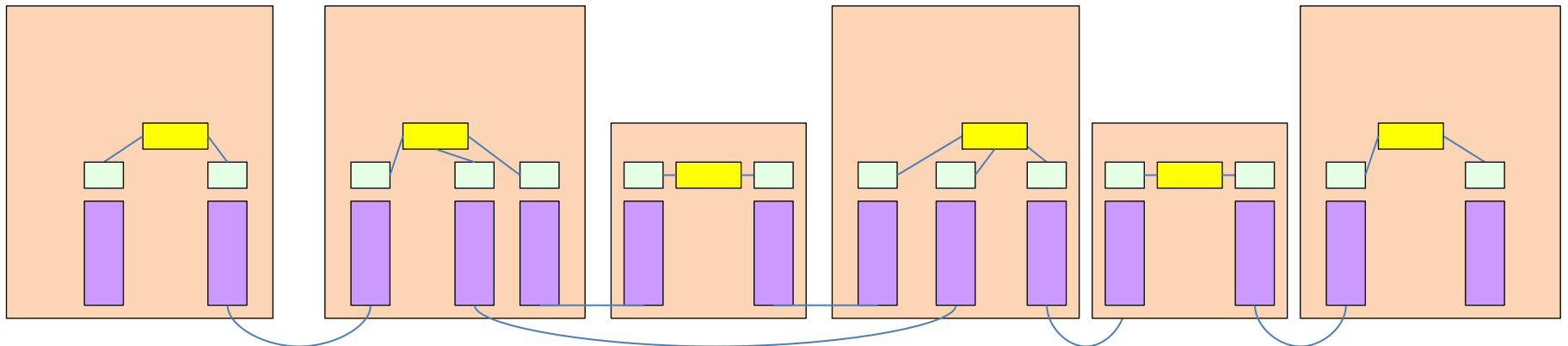
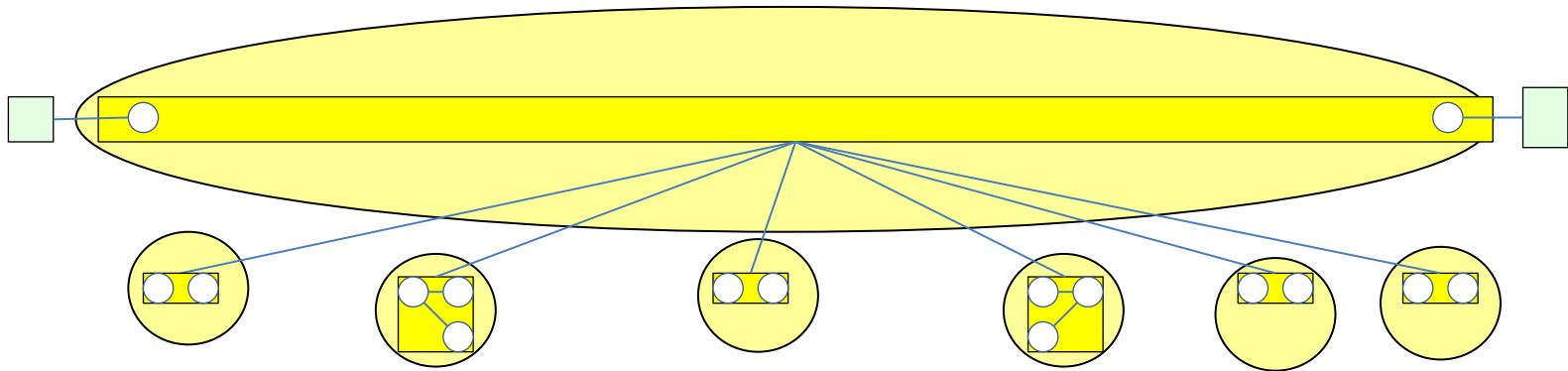
# Interpretation of Route and partition

- Route decomposition
  - FC A1-F2 has two routes, red and blue where each route has an FC
    - A1-F2
  - FC A1-F2 representing the red route has a route that has FCs
    - A1-A2, B1-F2
  - FC B1-F2 has two routes where each has FC
    - B1-F2
  - FC B1-F2 of route1 has FCs
    - B1-B2, C1-C2, D1-D2, E1-E2, F1-F2
  - FC B1-F2 route2 has FCs
    - B1-B3, H1-H2, I1-I2, J1-J2, F3-F2
- Partition
  - There are two FCs A1-F2, one in place (red) and one “pending” (blue)
  - The red FC partitions into FCs
    - A1-A2, B1-B2-B3, C1-C2, D1-D2, E1-E2, F1-F2-F3, H1-H2, I1-I2, I1-J2
- Note that
  - Some of the FCs in the route are exactly the same as the FCs in the decomposition where as some are specifically only relevant in the route view
  - The route is an ordered list whereas the partition is an unordered list

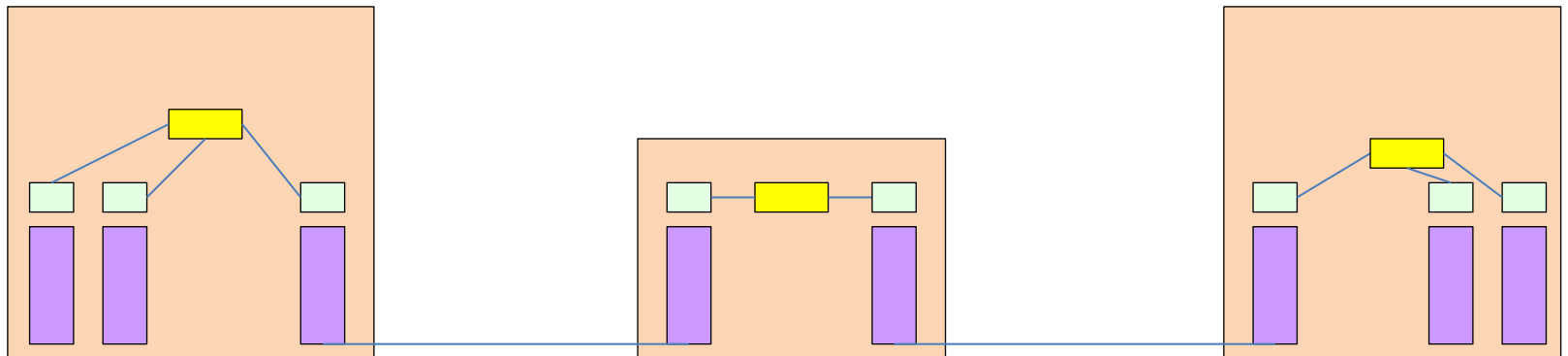
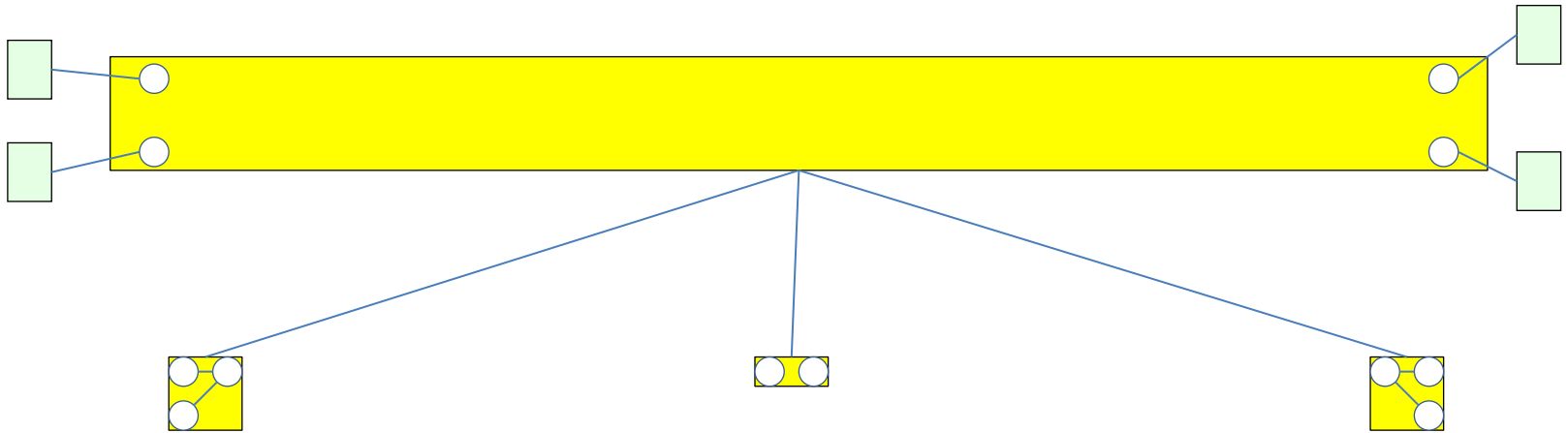
# Further points

- The route is represented by a route object
  - I have made a final adjustment to arrive at a coherent model that avoids an earlier complexity where, on the previous slide, the partition led to two FCs whereas the route form led to two routes
    - If an FC is used to represent the route this issue is removed completely
- Route
  - The route is described by a sequence of ports describe in terms of port pairs (FCs) highlighting switches and other relevant properties
- Comments and response
  - Clearly B1-B2 and B1-B3 are a part of the same FC <n> Exactly... well yes and no... From an NE perspective we can either consider them as two separate FCs both with a unidirectional switch connection towards the common B1 point where the switch goes “high impedance” when open and where the switches in the two FCs are inverse-ganged hence avoiding contention (this is the route form) or we can consider it as one FC with a switch that selects B2 or B3 (this is the partition form) </n>
  - Protection switching commands (automatic and manual) should be be directed to the FC (e.g. B1-2-3): The FC would update the route status. <n> Commands can be sent to the FC B1-F2 to select FC 1 or FC 2 of its route or can be sent to the FCs in B and F in the partition </n>

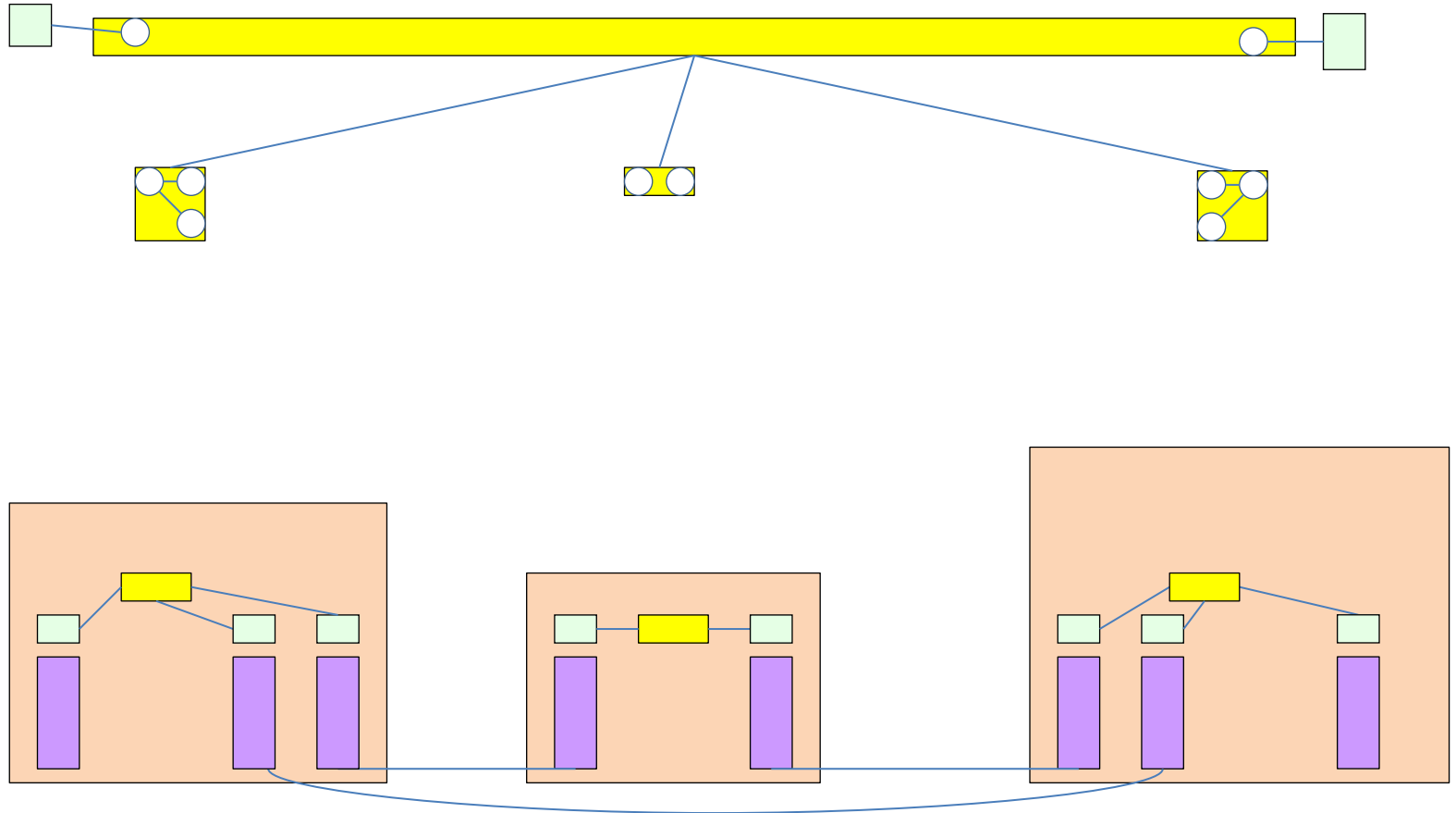
# FC partition as a description of a layout



# FC Partition (note the exposure of four ends) (ignoring ForwardingDomains)

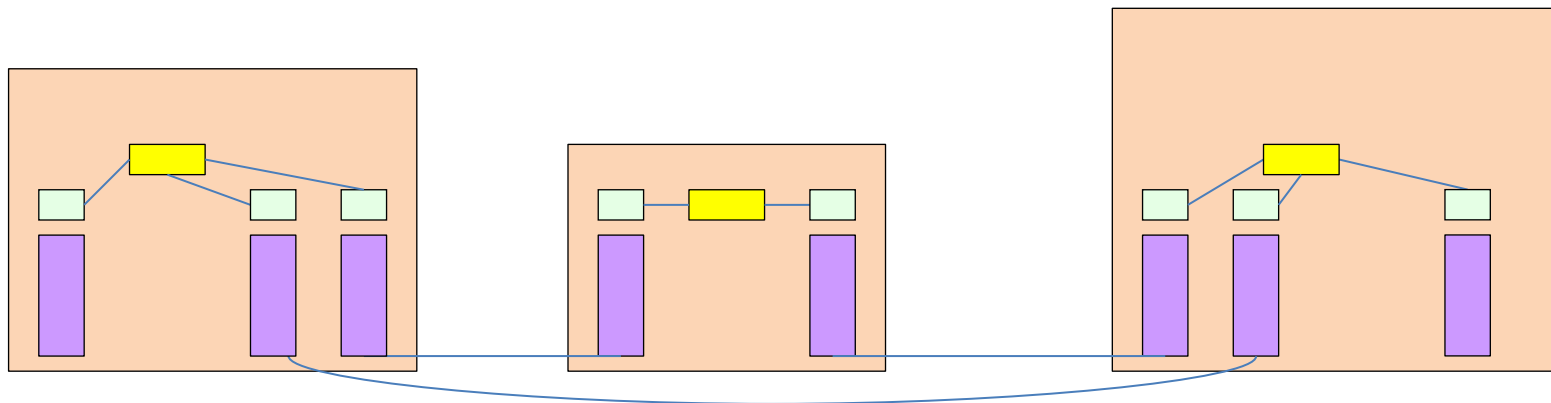
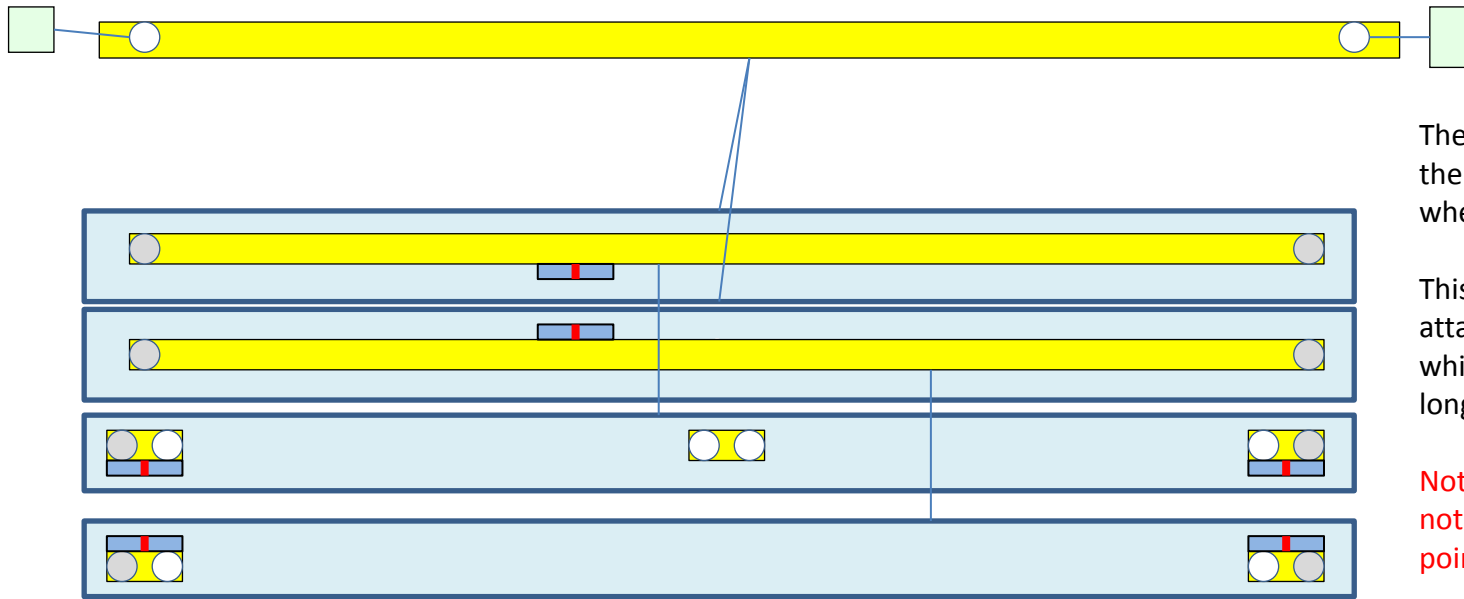


# FC showing partition





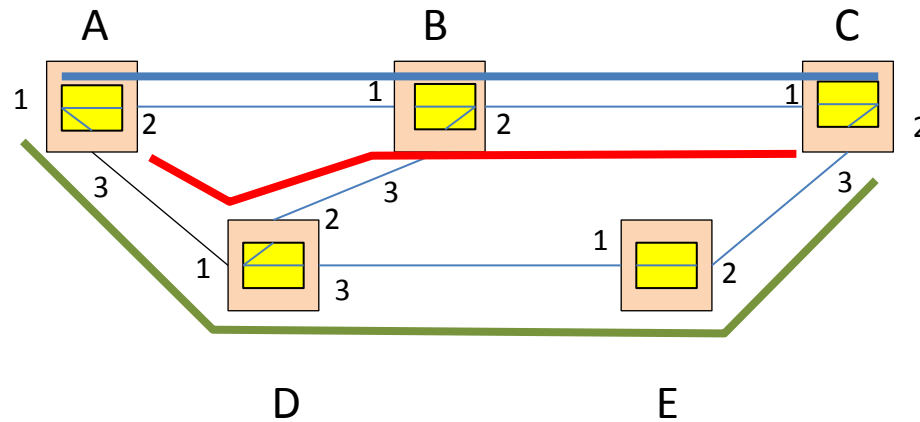
# Showing two alternative ways through



But what about directional asymmetry?

# Route v decomposition

What if there are many ways end-end

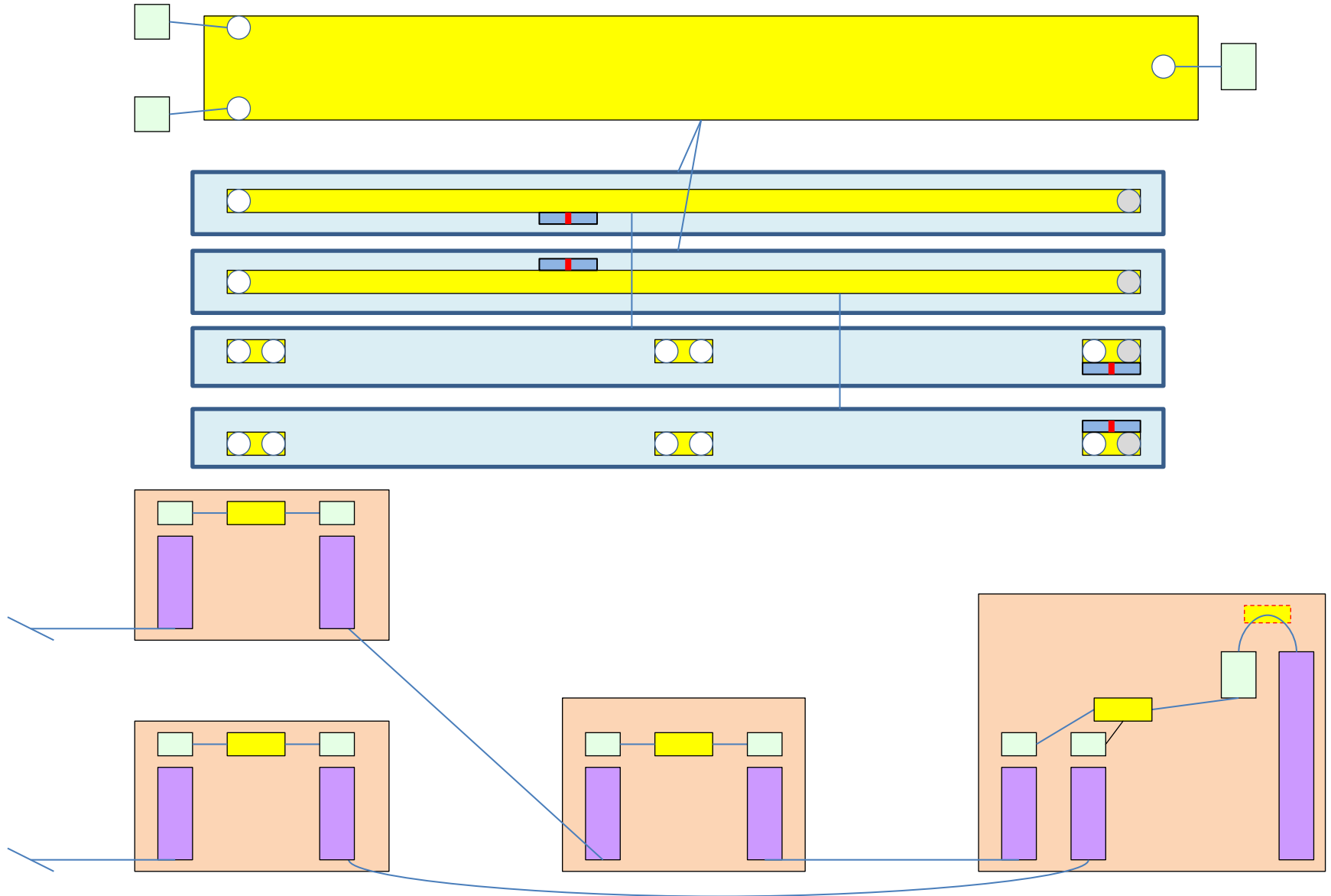


For complex layouts the route approach is not an efficient way of expressing the layout

# Potential route solution

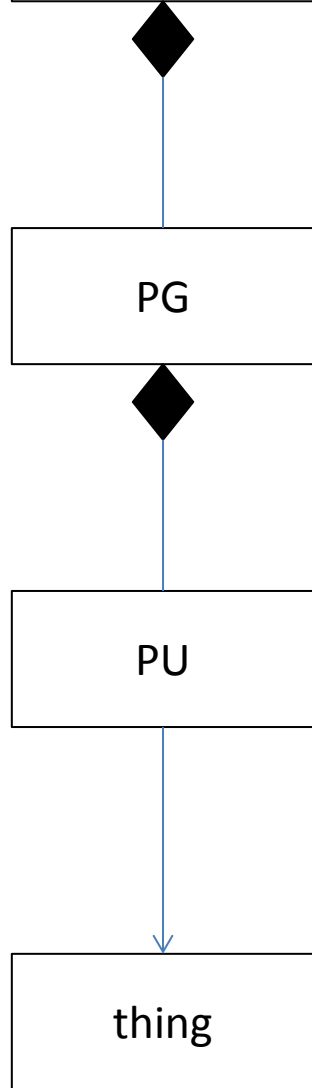
- A1-C2 has three routes each of which has one FC
  - A1-C2
- FC A1-C2 (1) has route which has FCs
  - A1-A2, B1-B2, C1-C2
- FC A1-C2 (2) has route which has FCs
  - A1-A3, D1-D2, B3-B2, C1-C2
- FC A1-C2 (3) has route which has FCs
  - A1-A3, D1- D3, E1-E2, C3-C2

# Showing two alternative ways through

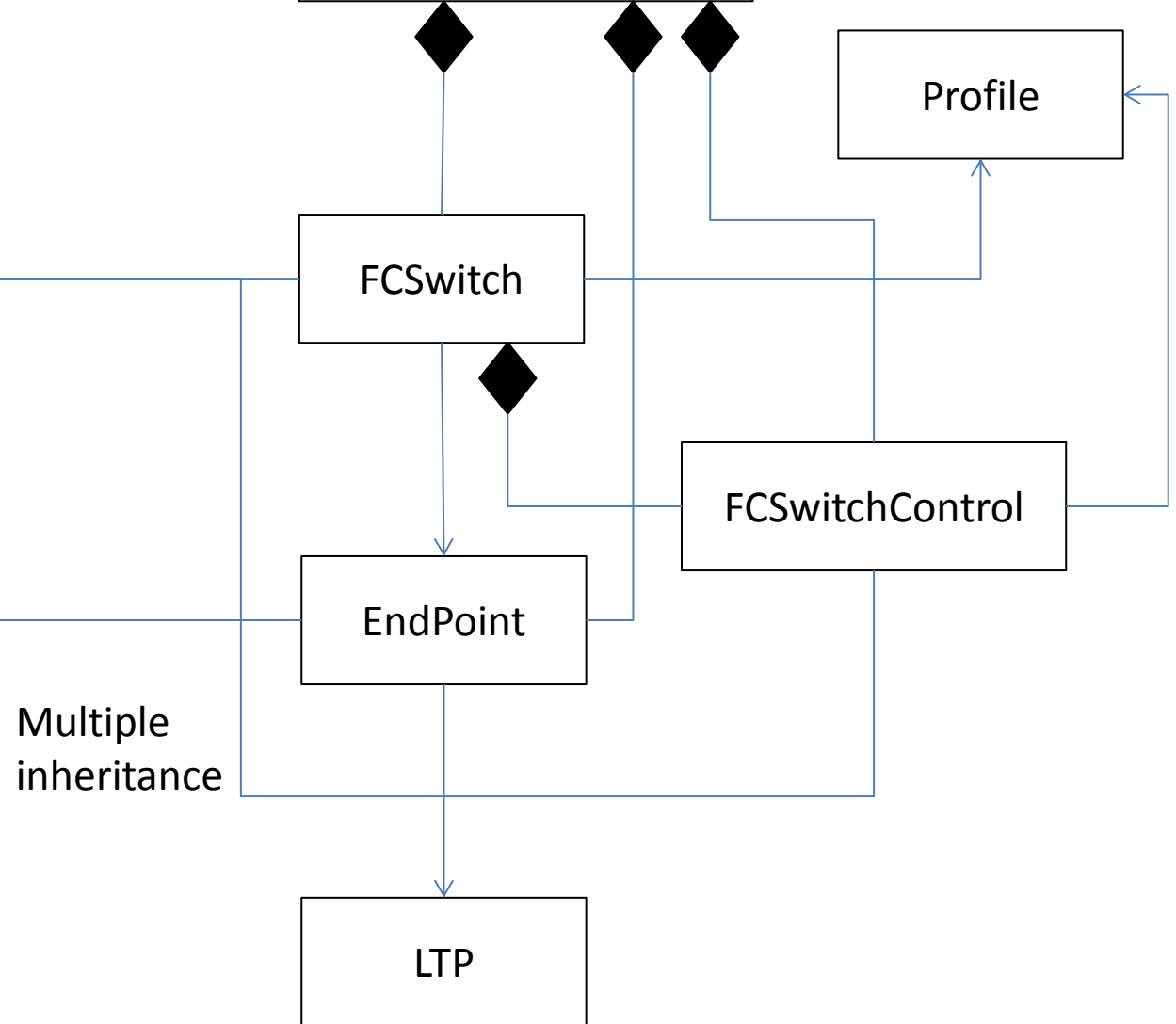


# Protection

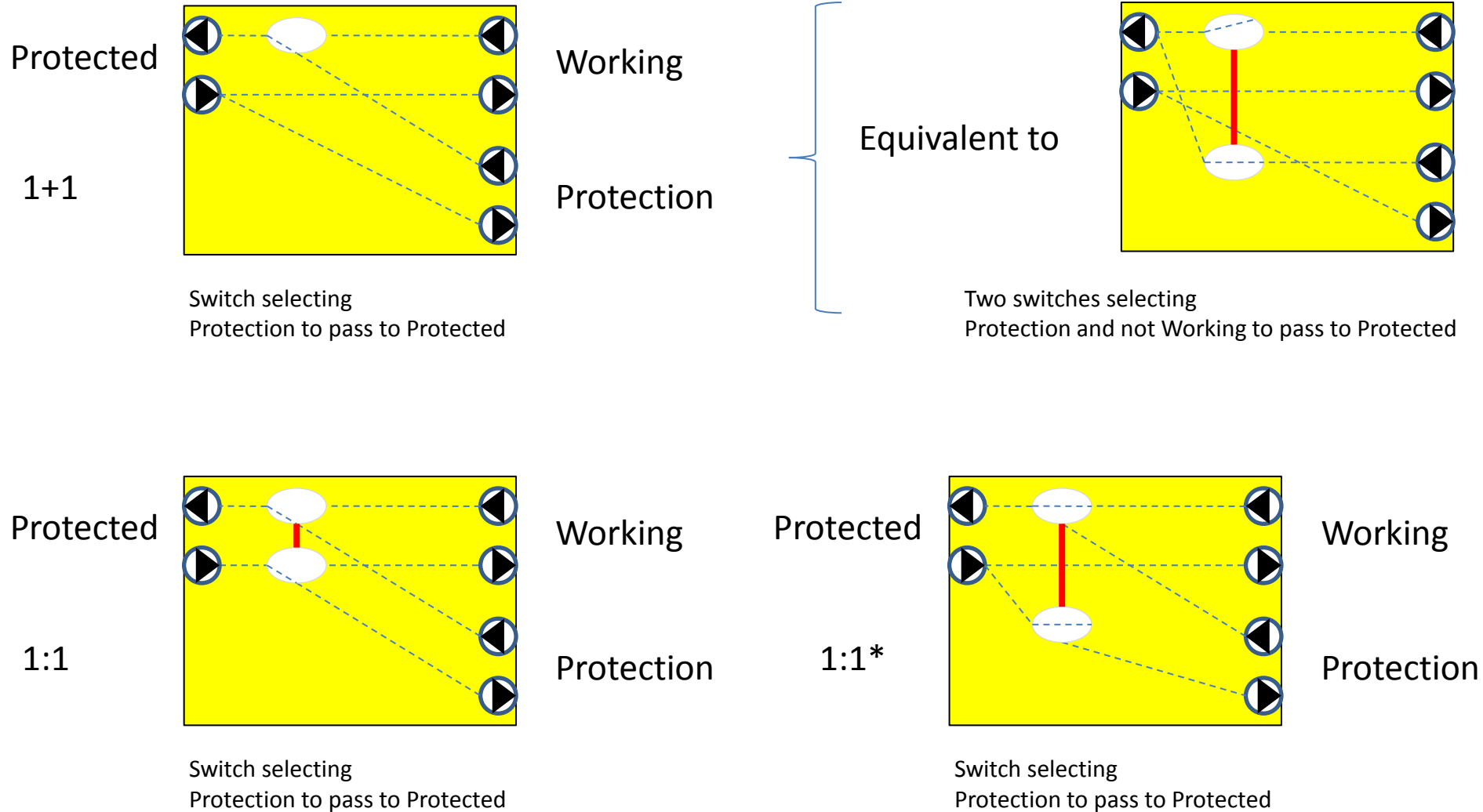
FD



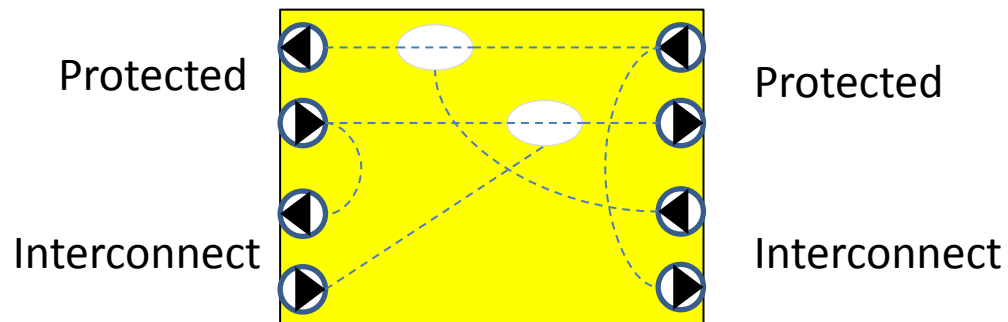
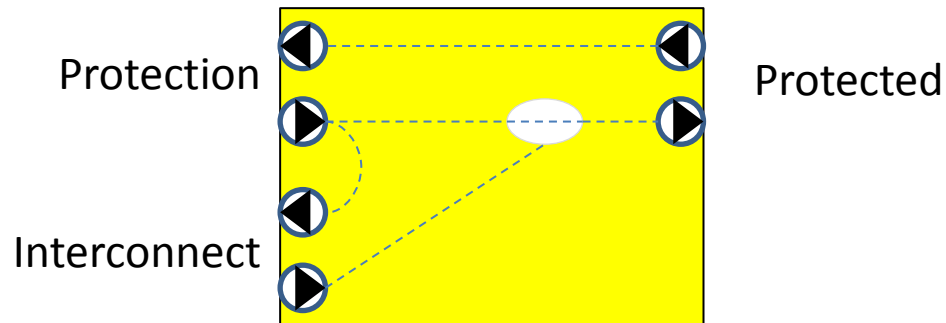
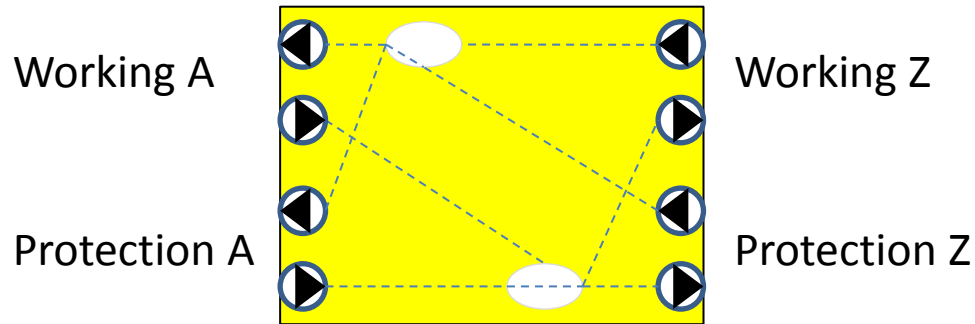
FC
----



# Various forms of protection in an FC



# Various forms of protection in an FC





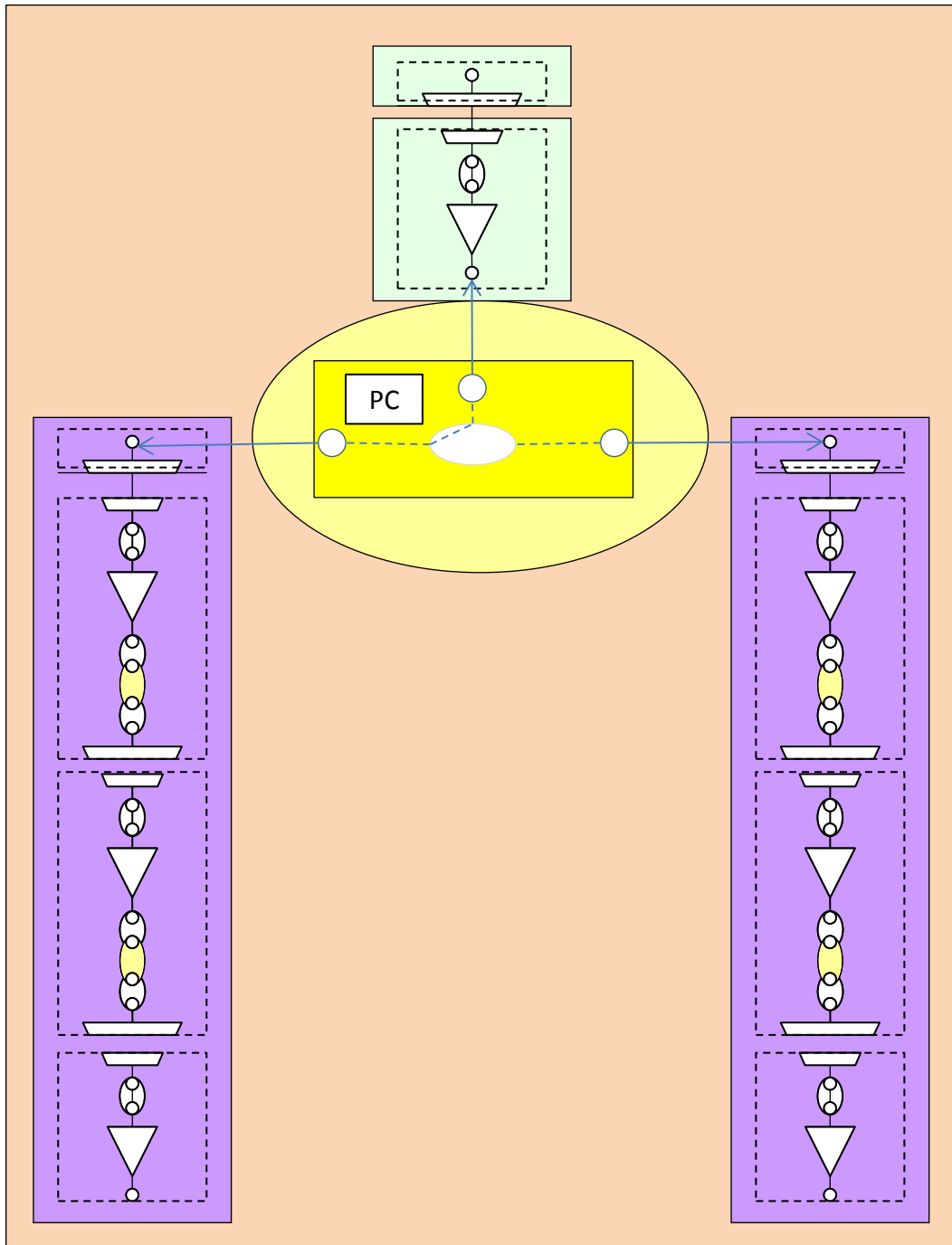
NE - ForwardingDomain

# NE – ForwardingDomain relationships

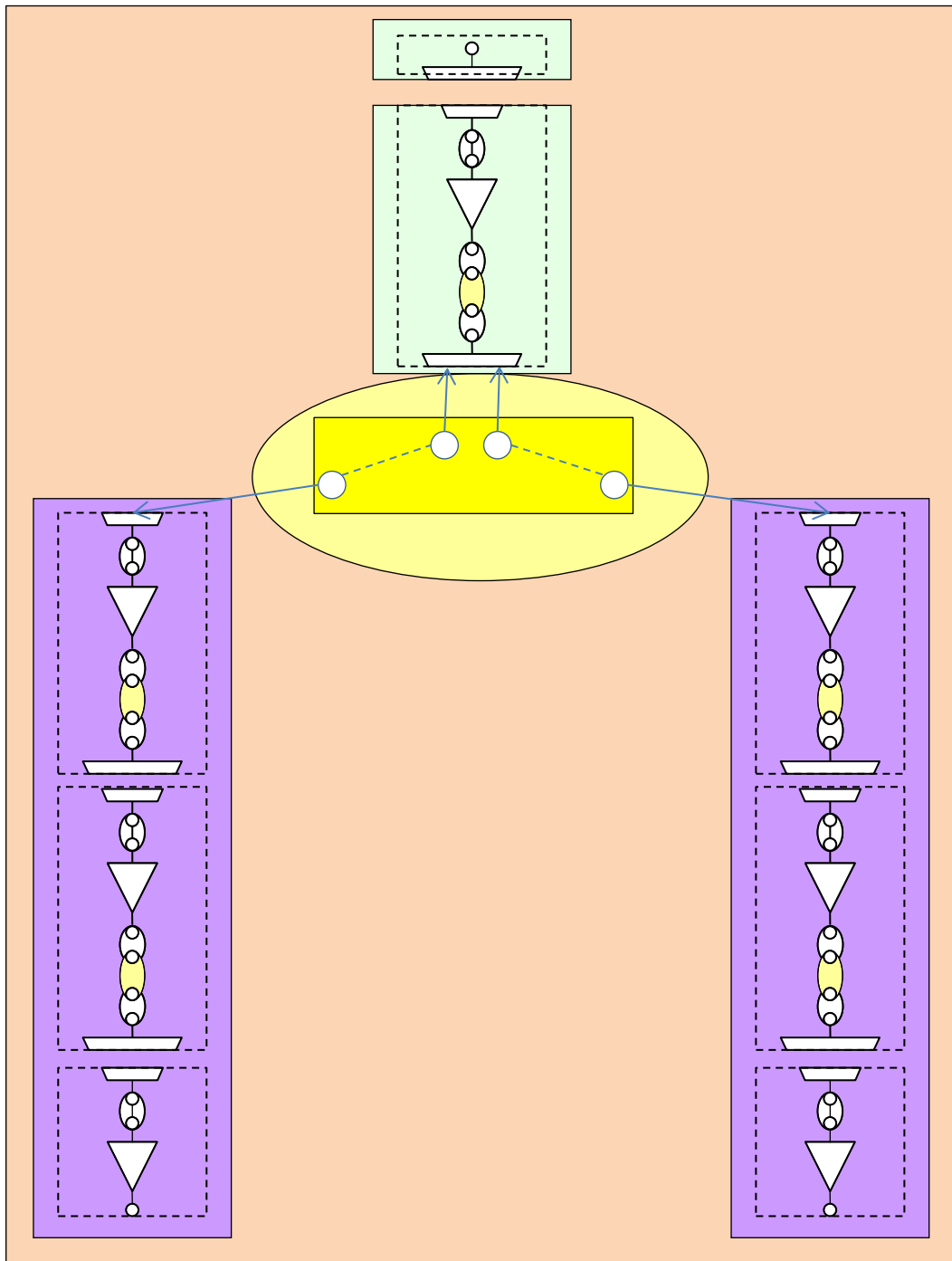
- It is not unreasonable to have a ForwardingDomain that is bounded within the scope of an NE
- BUT considering the NE as bounded within the scope of a ForwardingDomain does not seem reasonable
  - The NE is one of
    - A hybrid of physical and logical
    - The management scope
  - Where as the ForwardingDomain is purely logical
- Discuss...
  - The physical model and what we need to model

Protection

# Strict Protection via Lag



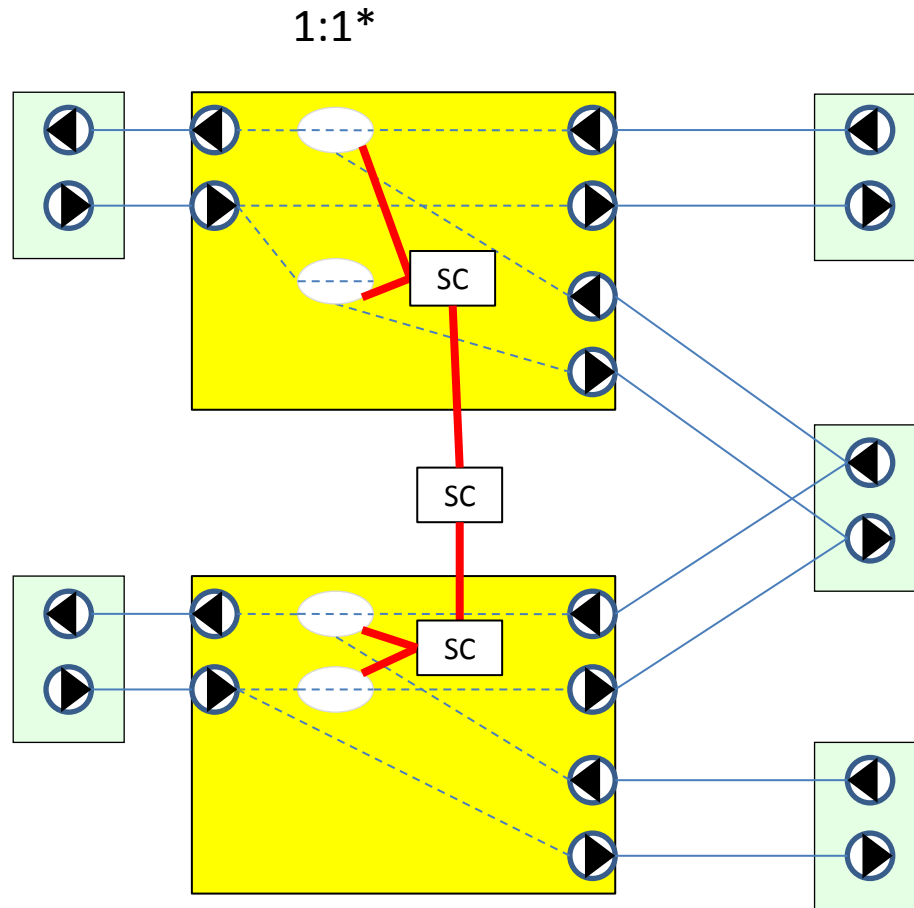
- Diagram of FC shows only one direction of traffic
- PC is a protection coordinator
  - This will be renamed switch coordinator as it may coordinate Loopback switches



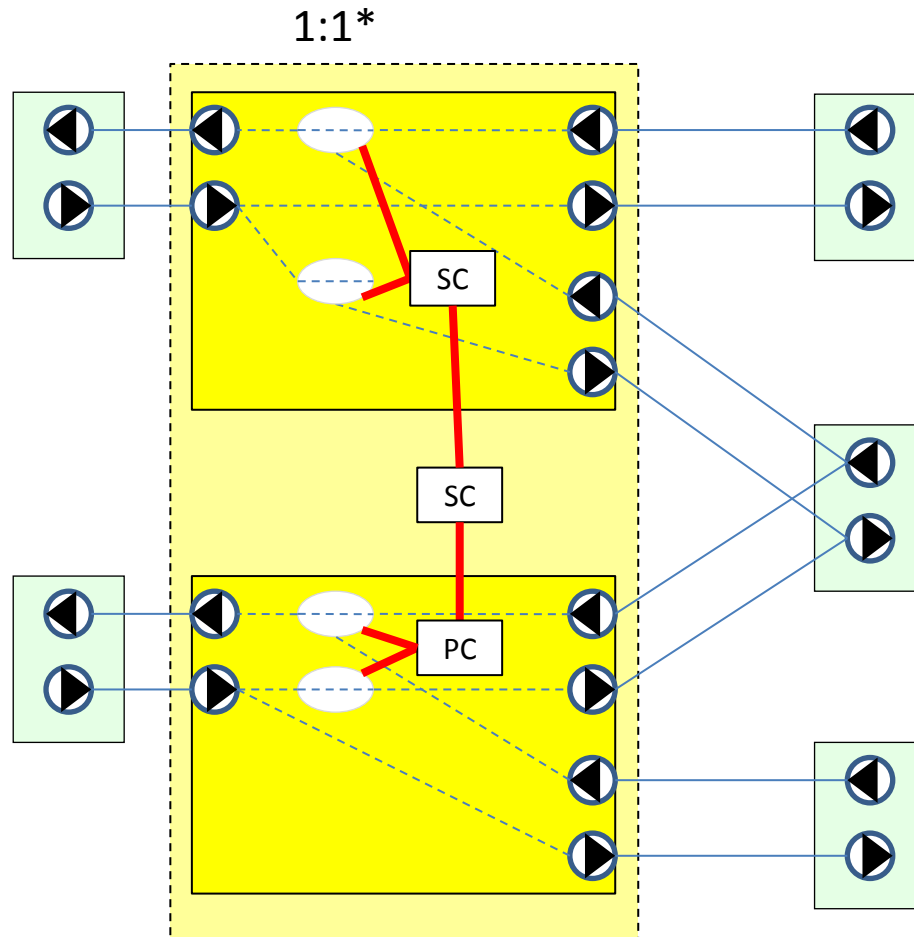
# Aggregation via LAG

- Could add multiple server side CTPs with separate FCs (to be consistent with TMF)

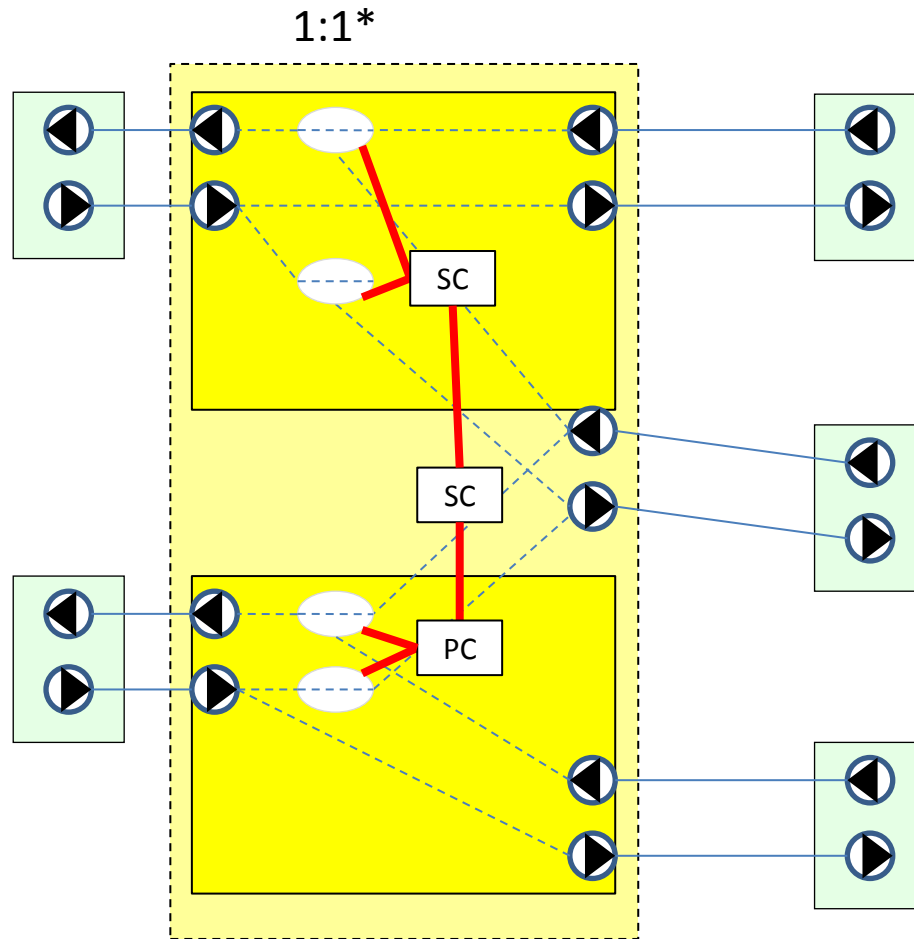
# Various forms of protection in an FC



# Various forms of protection in an FC

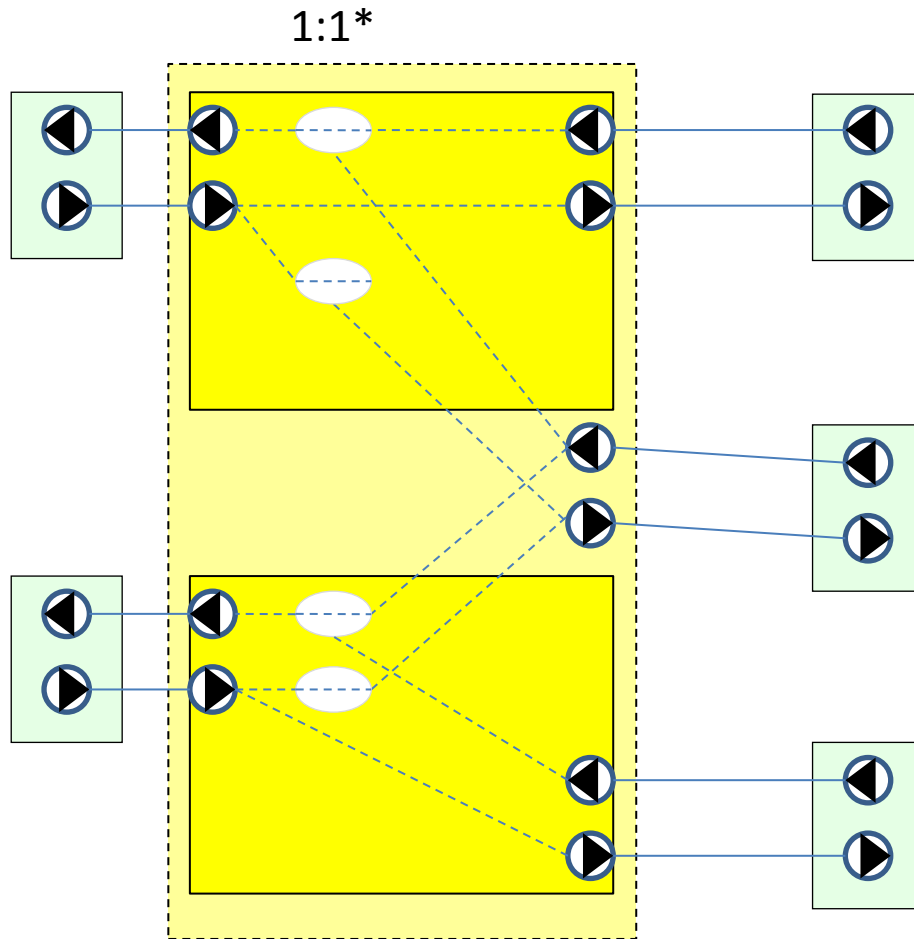


# Various forms of protection in an FC

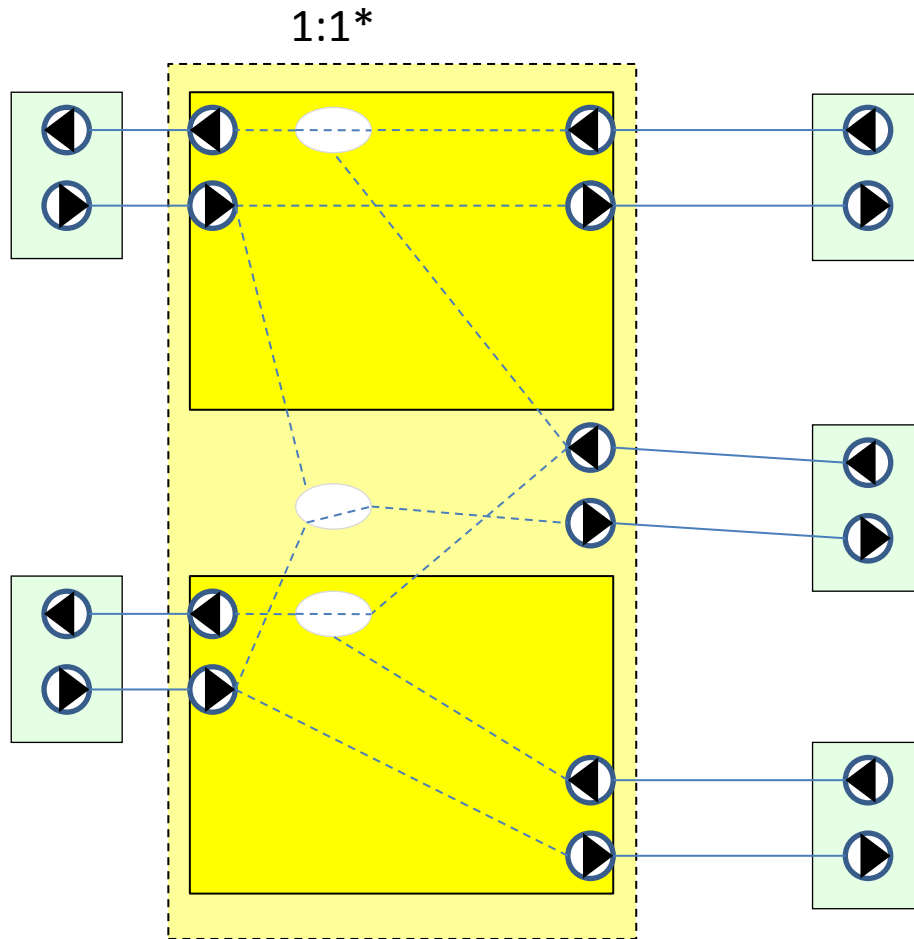




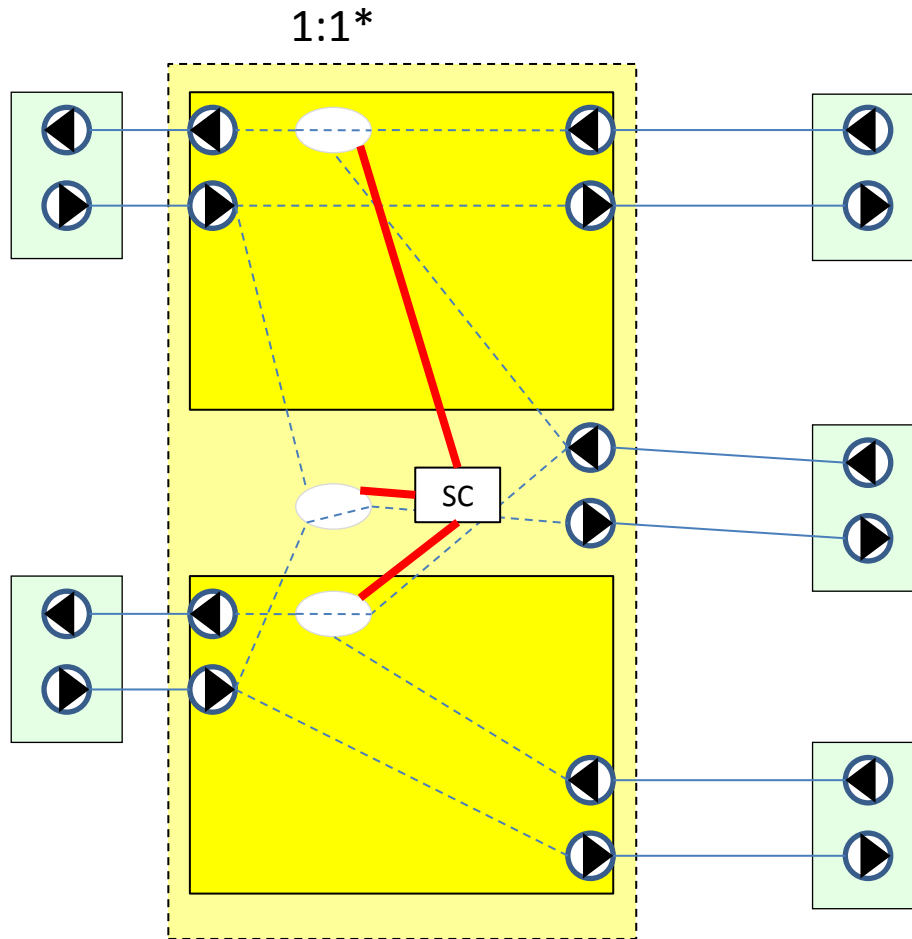
# Various forms of protection in an FC



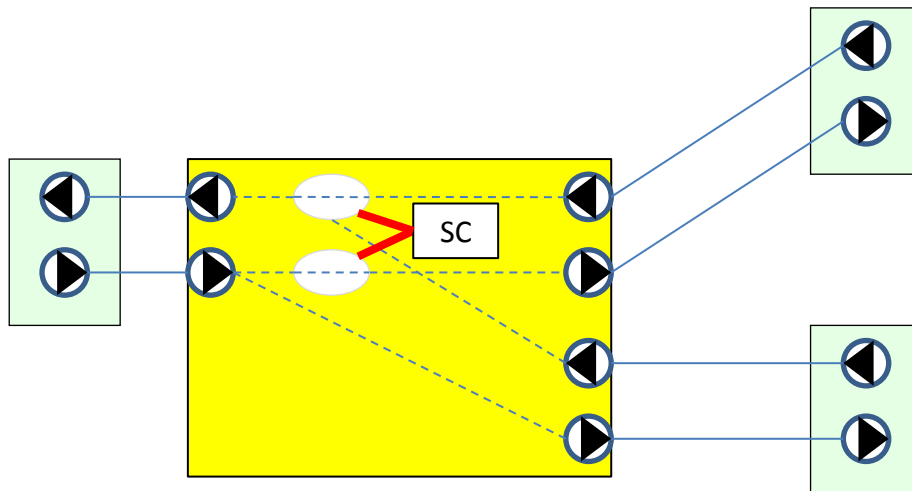
# Various forms of protection in an FC



# Various forms of protection in an FC



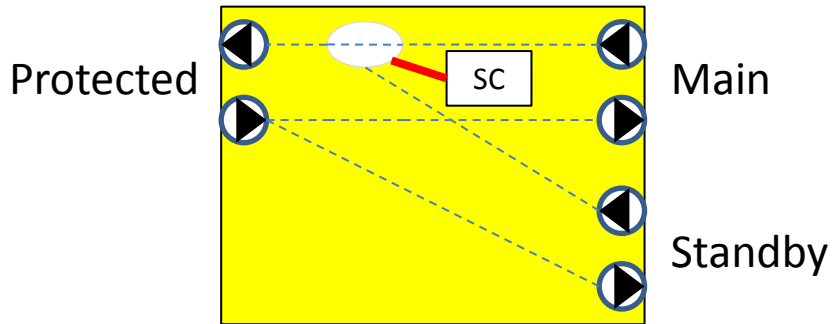
# Various forms of protection in an FC



# Protection controls

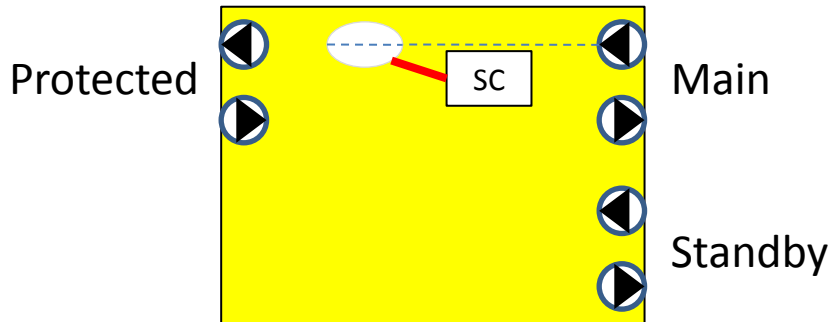
- Forced (state of switch assembly)
  - A particular selection is made and cannot be overridden by any automatic or other manual action
  - A force switch request cannot be ignored and will be followed regardless of whether traffic will be preserved or not
  - May apply to several switches simultaneously due to coordination of switching
  - That a switch assembly is forced is a state of the switch assembly reflected via the coordinator
  - When a force is removed the switch will freerun on automatic abiding by lock-out restrictions and will apply a wait to restore timer or a hold off timer if a switch is to be made
- Manual (applied to switch assembly – on shot)
  - A particular selection is made that can be overridden by automatic action
  - That a switch assembly position was manually set is not remembered??
  - A manual switch request will be ignored if any automatic process deems that it has a better choice (and hence manual can only apply where the selection is between points of equal priority)
- Locked Out (state of Endpoint of FC or even LTP??)
  - A locked out point cannot be selected by manual or by forced or by any automatic actions
  - If a port is selected and it is then locked out the automatic switch scheme will choose the next preferred port that has not failed, if all have failed it will choose the most preferred port (though failed)
  - If all ports are locked out the protection scheme will select the preferred port but the switch will not convey any traffic
  - When lock-out is removed and if the port no longer locked out is preferred over the current selection switching will occur immediately
- Hold-off time (all switches caused by a “signal” condition)
  - The duration for which a signal condition indicating need for a switch has to persist before the switch away action is taken
  - Note that if the alternative is also failed no switch will be made until the alternative recovers for the hold-off time
- Wait-to-restore time (revertive only)
  - The duration for which a condition on a preferred path has to persist before a switch to preferred is made
  - If the switch is selecting non-preferred and the preferred recovers the switch back will be immediate

# FC spec v instance



Specification explains:

- The endpoints and their roles
- All flows along
- All switches
- All switch parameters options
- All available controls

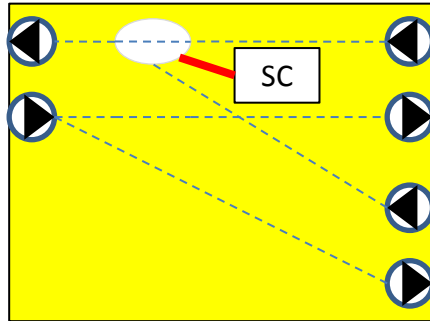


Instance simply provides

- The endpoints and their roles
- All switches each with their current settings
- The current values of each of the parameters of the switch coordinators

# FC spec v instance

Normal



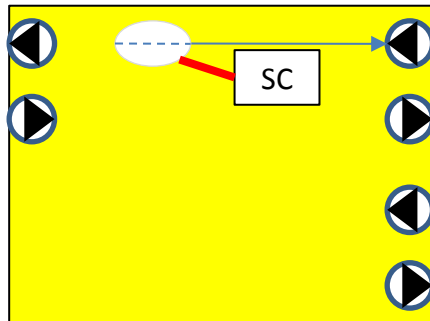
Working

Protection

Specification explains:

- The endpoints and their roles
- All flows
- All switches
- All switch parameters options
- All available controls

Normal



Working

Protection

Instance simply provides

- The endpoints and their roles
- All switches each with their current settings
- The current values of each of the parameters of the switch coordinators

# Capability model

- LTP stacks with LTP capacity interaction rules
- ForwardingDomains with FC type support rules and with orientation restrictions



# Encapsulation principles

- If the positional bounds of two related concept instances are coincident for their entire lifecycle then they may be merged into a single entity instance representing the composite concept and hence share an identifier etc
- If the positional bound of one concept instance is a subset of the positional bound of another concept to which it is related for its entire lifecycle and where that larger concept can be considered as a dominate definition then it may be subsumed into the entity representing the larger concept and hence be identified as part of the entity for that larger concept in terms of attributes of that larger concept
- If the positional bounds of several instances of a concept are all subsets of the positional bound of a another concept to which they are related for their entire lifecycle and where that larger concept can be considered as a dominate definition then they may be subsumed into the entity via a composition relationship
- If a concept instance that bridges two other concept instances (of the same or different types) is, in the particular case, devoid of anything but identity then it may be represented simply by associations between the entities representing the two other concept instances
  - The associations may be two way navigable or one way navigable depending upon the original associations
- If a concept instance that is a leaf is devoid of anything but identity then it may be omitted

# Splitting

- To be added.

Mapping FCs to OF tables

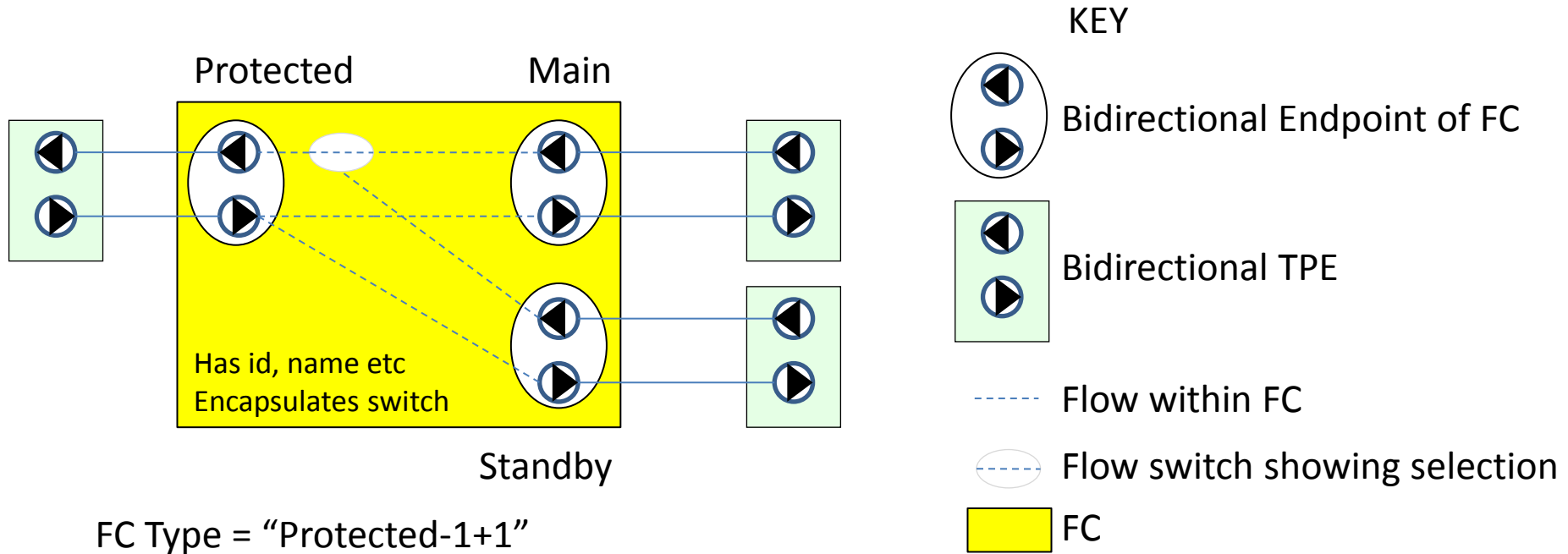
# FC to OF mapping

- OF tables hold the per flow or flow fragment rules
  - If data with values x arrives output on port y (unidirectional)
  - The rules are the machine code
- The FC is a representation of the effect of many rules as perceived by (and as desired by) an external overseeing “user”
  - Bidirectional Hub-spoke configuration with hubs at a and b and spokes at c, d, e etc
  - The FC represents the perception of the thing/resource providing service and hence an aspect of the perception of service

# Consider mapping FC to OF

- For simplicity assume
  - A simple network device
  - A simple case where there is only one flow per logical port and each logical port has a single dedicated physical port (to avoid the channelization etc challenge)
  - The traffic fabric only has three logical ports accessible
  - Forwarding is implemented in a single ASIC
  - The ASIC has an OF table internal form and register access to program that form
  - A Java API is provided by the device driver that allows coordinated access to the ASIC
  - As the ASIC has an internal table form the ASIC registers can be mapped directly to an OF externalized interface
  - The network has only one NE
  - The customer desires a protected service that is dual homed on one side and single homed on the other in a 1+1 configuration
  - The essential implementation in the NE is hence an FC with 1+1 protection
  - <50ms protection is required

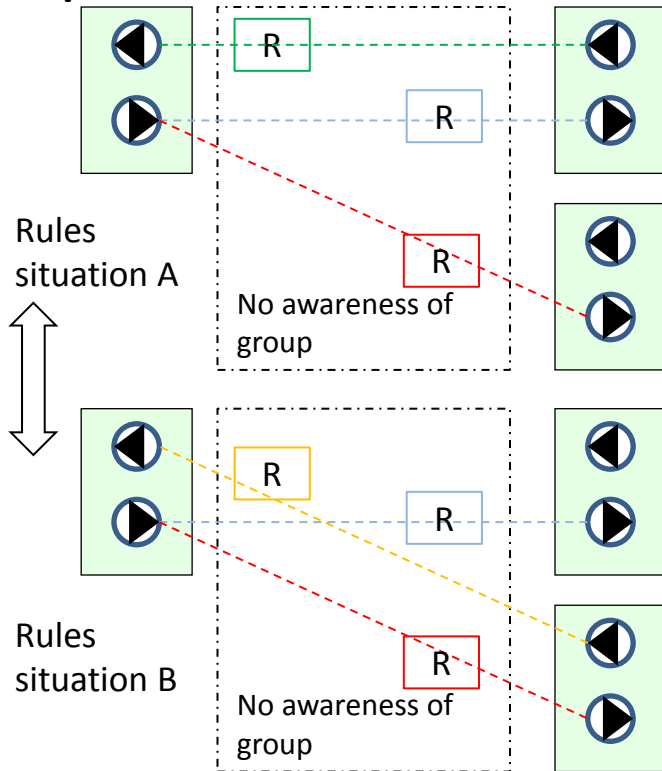
# Mapping protection – a simple case



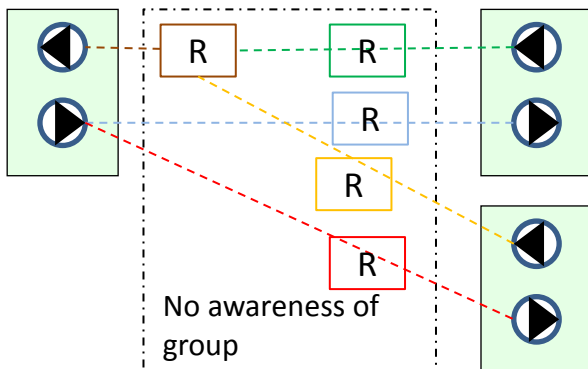
- 1+1 protected form has two permanent flows and two potential flows where one of the two must be an actual flow
- The switch indicates which of the two potential flows is actual

# Table content options

## Option 1







## Option 2

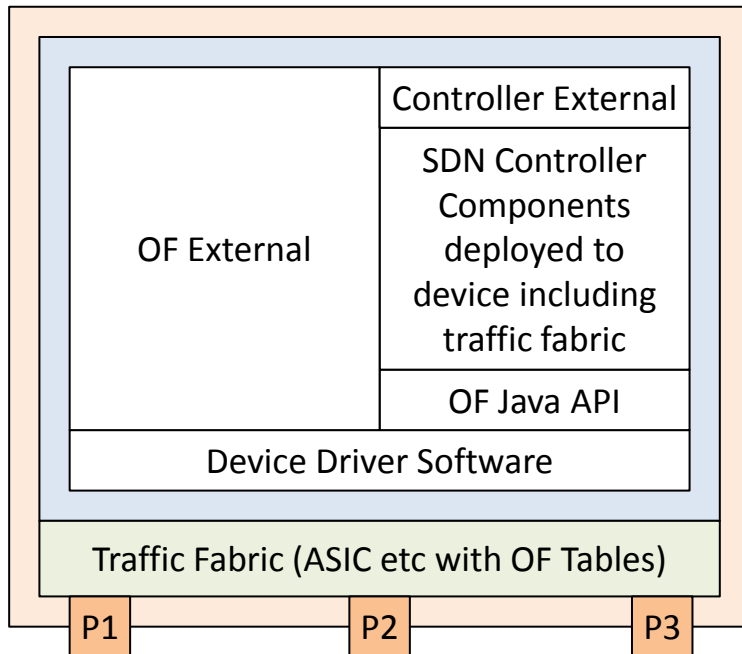


- In both options
  - Simple forwarding rules are held on table to enable traffic flow from left to both right ports (blue and red flows)
  - There is no notion of an FC (as there does not need to be)
  - All flow rules are independent except where highlighted by flow serialization
- In option 1
  - Simple forwarding rules are held in the table in situation A to allow traffic from top right to left (green flow)
  - Controller updates forwarding rule for situation B to allow traffic from lower right to left (orange flow)
- In option 2
  - Tables include switching rule with conditions etc to select feed to flow to left port (brown) from green and blue flows from right port
  - The green, blue and brown rules are related by flow serialization but this is not “known” in the tables

# Mapping considerations

## KEY

-  Single physical device containing generalized compute and traffic hardware
-  Generalized compute hardware with infrastructure to allow controller component deployment
-  Traffic Hardware
-  Physical Port



- Two table options
  - Tables have rules for current flows only
    - Switching is achieved by locally resident controller software reconfiguring the tables
  - Tables have rules for current flows and potential for flow
    - Switching is achieved by ASIC interpreting OAM conditions etc and changing current flow
- Controller External can be
  - FC form
  - OF table form
- OF interface is either
  - Internal to device compute
  - External to device



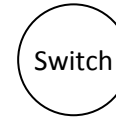
# Generalized Layered implementation showing alternative hardware/software formations

Note device boundary will always necessarily have thin mediation software (not shown to avoid clutter) to map from internal parallel presentation to external serialized presentation

## KEY



Hardware implementation



Software implementation

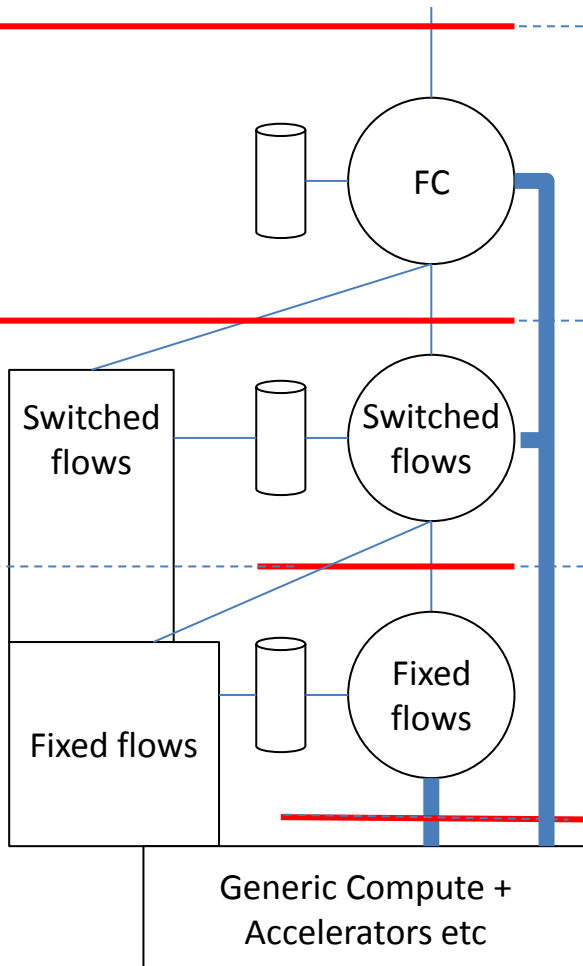


Stored data/code/rules (tables etc)



Software is running on

Alternative Device (NE) Boundaries



### FC view (not OF)

Bidirectional complex treatment. Has control/management switch coordination based on shared information with other control/management components. FCs have ids **and mappings to subordinate flows (in terms of ids and flow rules etc)**. This does NOT need to be on compute next to the genuine data plane (but could be).

### Switched flows-view (OF?)

Fast control/management. Provides switched flows where the flow is changed based on comparison of quality of several incoming streams. Each flow may have an id. Points in the flow have ids. Rules have ids. **No need for FC id.** Considering comms round trip & 50ms switch requirement, if software, this almost certainly needs to be on compute next to the "genuine" data plane. If hardware it is with the genuine data plane.

### Static flow-view (OF)

"Genuine" data plane. Moves (forwards) data "from one place to another" within the device where the move is based solely on fixed rules and properties of the incoming data content where each flow is treated independently. The points bounding the flows have ids. The rules have ids. This is positioned specifically where the data to be moved is positioned, there is no option (other than via overall network engineering)

### Generic compute ++ resource view

Generic compute. Allows loading of software and data and subsequent running of the software to act on the data. The Fixed flows software is potentially running on the generic compute (although in some cases it may be specialist compute). The fixed flows hardware may also take advantage of generic compute)

# Observations wrt IDs

- Each FC is supported by a number of rules
- Each rule supports one or more FCs?
- If rule supports more than one FC it will need to indicate this so that it is not deleted when the FC is deleted
- When an FC is deleted all rules that support it alone need to be deleted
- Each rule that supports an FC should be maintained in the FC repository and should be validated and corrected if changed through some other process
  - If this is the case then the rules in the traffic fabric do NOT need FC identifiers as the image in the repository translation in the controller provides the mapping (and will deal with the case where a rule supports more than one FC)
- If rules are found that do not correspond to any FC they should be deleted?
- If the controller loses all information on the rules and FCs it is NOT possible to reverse engineer the full FCs from the rules although it would be possible to determine the effect of the rules and to arrange these effects into probable FCs
  - An identifier in the traffic fabric that relates otherwise unrelated rules may help here
- Issues reported against rules can use the rule id and correlated this via the repository in the controller to determine the impacted FC
- If alarms are reported in the context of rules (and intermediate points between rules) then the rule id can be mapped to the FC id

- Transactional activities
- Action on several things at once that have a common handle
- Discovery of service from the data plane is not possible as the switch validly has a limited view
- If there are resilient controllers they need to have a shared view of the service and the FC and the mapping to rules

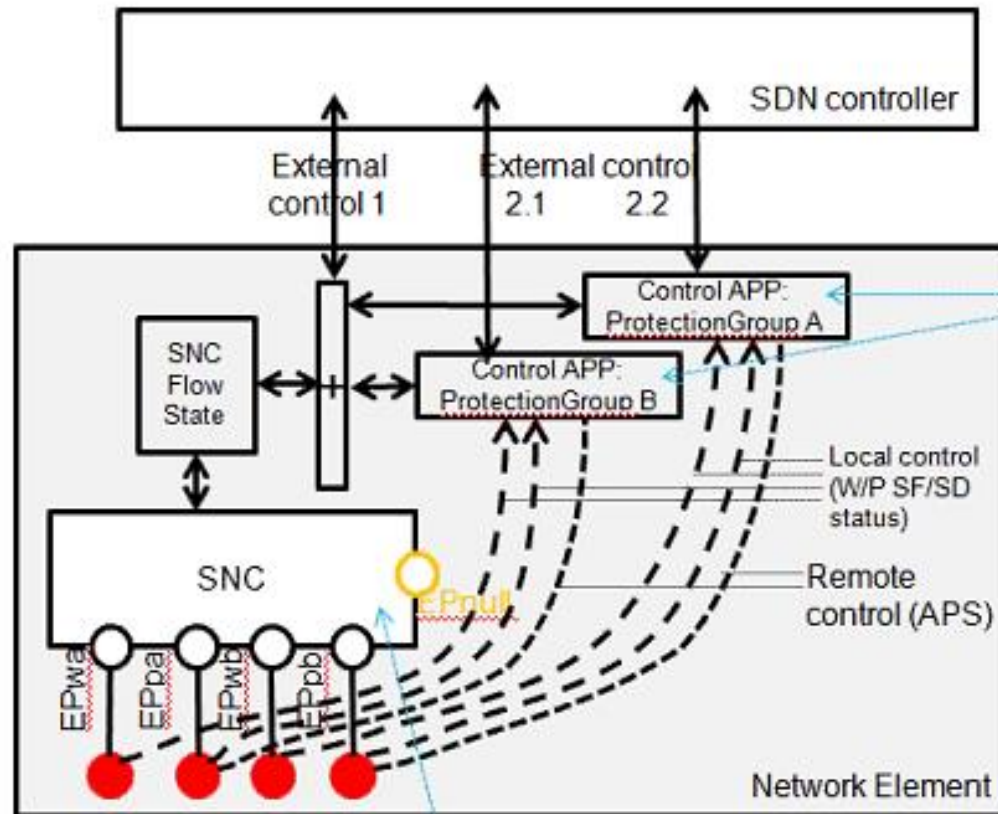
# Summary of observations



- OK 1. LTP(CTP, TTP), SNC, SNC Flow and ProtectionGroup managed objects should be structurally separated.
- No 2. LTP(CTP, TTP) managed objects can be chained (as in a 'Table Type Pattern') within the boundaries of a multi-layer LTP~~X~~ managed object. The Term&Map connectivity between a CTP and a TTP in such LTP~~X~~ is included in the chain. The LTP is multi-layered. The LTPP is questionable in itself
- No 3. LTP(CTP) managed objects should support two modes: inactive and active. Need to consider potential as not a set of instances
- In inactive mode, the object only resides in the control plane and no data plane resources are yet allocated to it.
  - In active mode, the object has been allocated data plane resources. This LTP mode is controlled by the Protection Group managed object.
4. A SNC managed object is connected with a number of EndPoint objects. Some of these EndPoint objects may connect to an inactive LTP(CTP) object. The Protection Group managed object will have to activate the LTP when it decides to set up an SNC Flow to this EndPoint.
5. The protection control APP associated with a ProtectionGroup managed object should be capable to control the SNC Flow state associated with multiple SNC managed objects.
- E.g. in case of 1:N and  $(1:1)^N$  linear protection, linear group protection (e.g. SNCG/I), shared ring protection and shared mesh protection.
6. The protection control APP associated with a ProtectionGroup managed object should be capable to control the LTP mode of LTP instances associated with multiple SNC managed objects.
7. One SNC managed object may be associated with multiple ProtectionGroup managed objects.



# Architectural elements in a back-to-back protected SNC

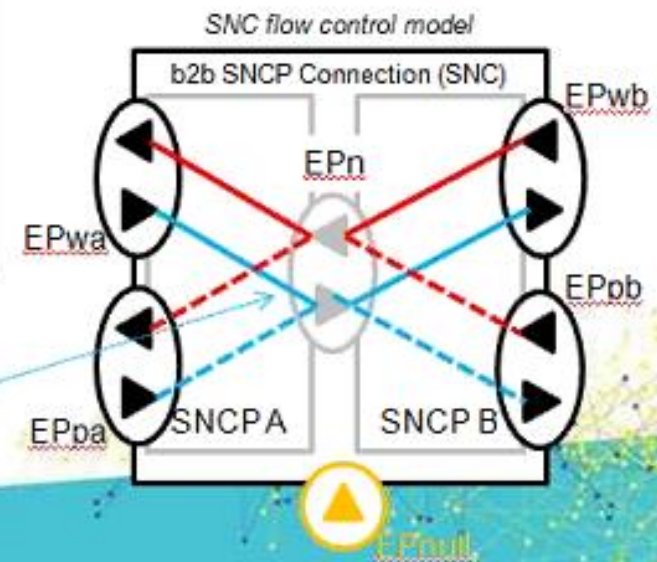


Multiple protection control APPs control Flow State in one SNC.

ProtectionGroup: SNC = M:1

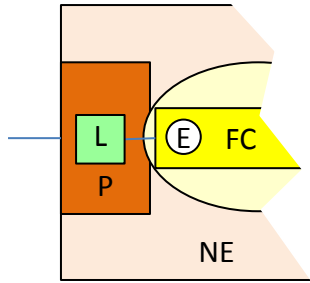
Separate administration of ProtectionGroup and SNC.

Flow State in SNC is controlled by a combination of independent control APPs, which control the flow state between a subset of EPs at the edge of the SNC and an additional logical EP that can be assumed to exist internally in the SNC.

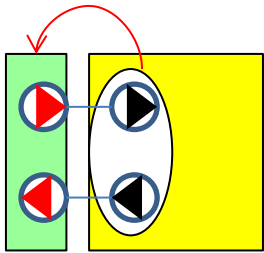


# Case

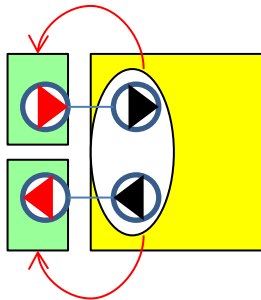
# Simple layout variety



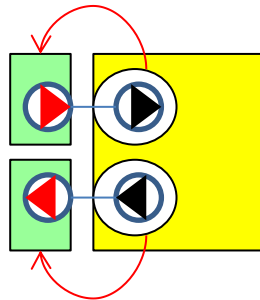
- Each case below has a simple flow legality
- It is not clear whether all cases are required but all are supported by the model.



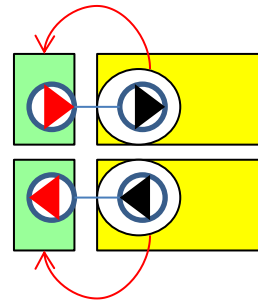
A) This case occurs when there is a strong bidirection treatment of the LTP server layers (inc physical) and the FC (Forwarding) is also essentially bidirectional.



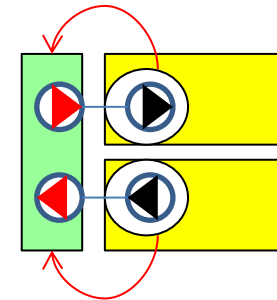
B) This case occurs when the flow to the left of the termination points diverges and hence needs to be treated unidirectionally (and is also treated as unidirectional in the server including physical) but the Forwarding connection) is essentially bidirectional (to the right).



C) This case most likely occurs when the role of the points in the Forwarding Relationship (connection) differ although associated with the same single bidirectional flow from some perspective.



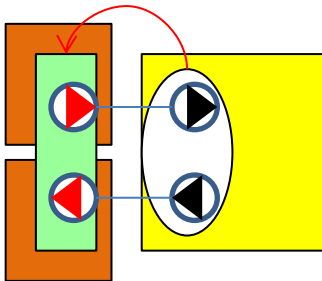
D) Two independent unidirectional flows.



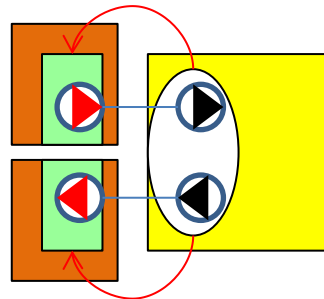
E) This case occurs when there is a strong bidirection treatment of the server layers (inc physical) but the signal diverged via the Forwarding Relationship in such a way that there is no shared fate, transaction etc that needs to be managed. It is clear that case (C) could also be used.

# “Addressing” variety

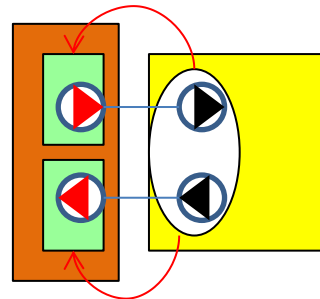
- This discussion assumes an “address” oriented approach to access to the LTP and FC where the address if the LTP includes equipment that they are supported by



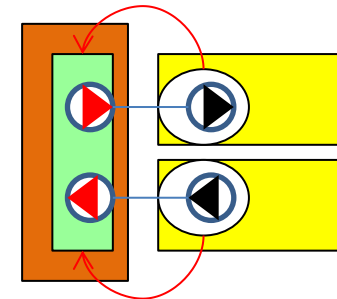
A1) The termination point is named from one of the Equipments (chosen to best match the name provided by the NE). In general the a view of the supporting equipment should be made available independent of the name.



B1) The termination point are each named from the Equipment they are supported by



B2) The termination points are both named from the one Equipment. Some other identifier (port, direction or whatever) differentiates the two.

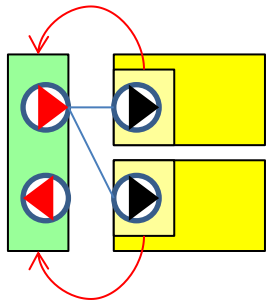


E1) The bidirectional termination point is named from the Equipment it is supported by

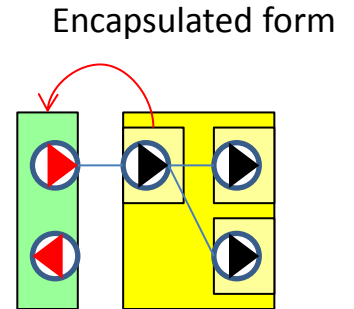
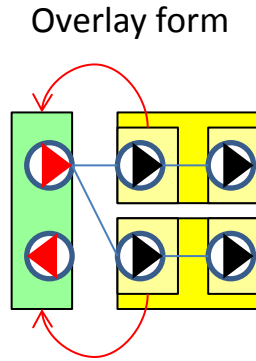


# Flow Legality

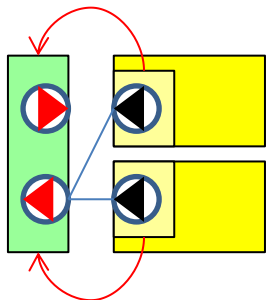
- This discussion considers a number of split flow cases
- The second case shows the protection alternatives



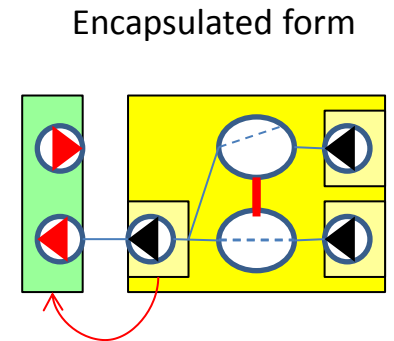
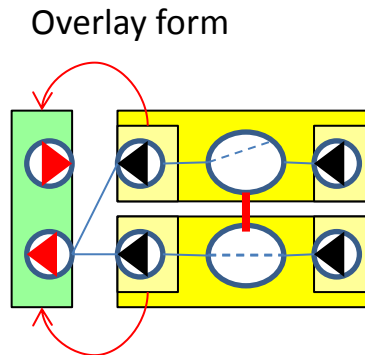
Legal.  
Overlay model for Broadcast.



Equivalent

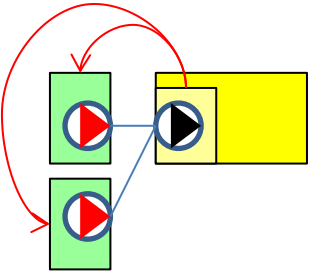


Only legal if flows switched.  
Overlay model for Broadcast.

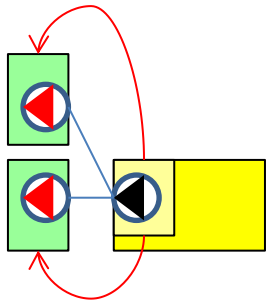
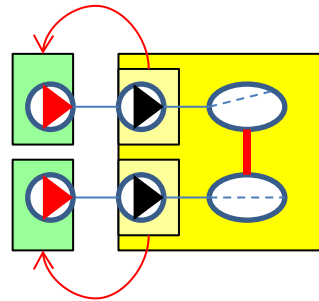


Equivalent

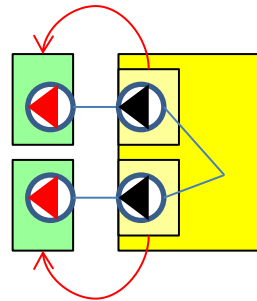
# Flow Legality

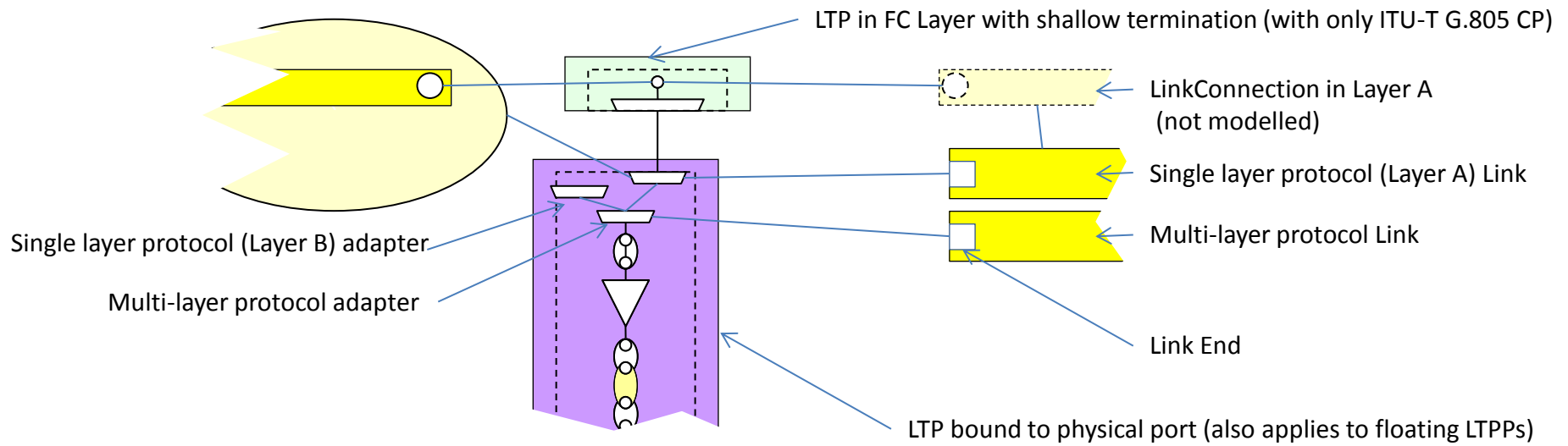


Never Legal.  
Corrected form →

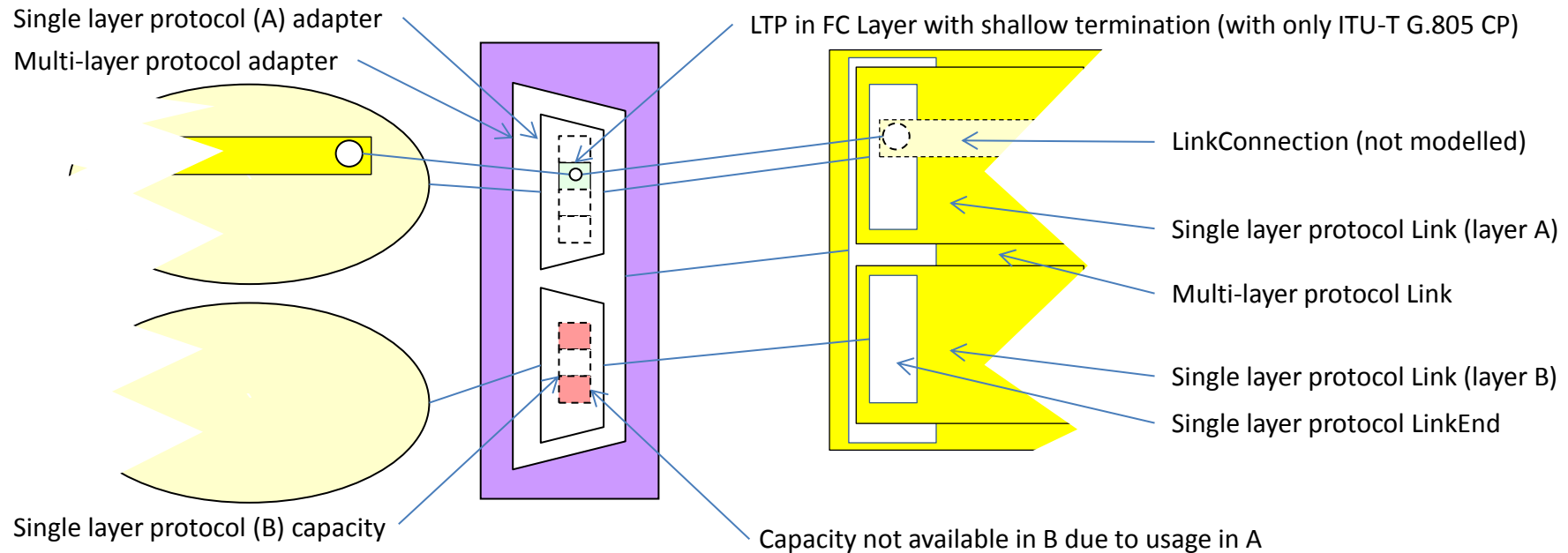


Never Legal.  
Corrected form →

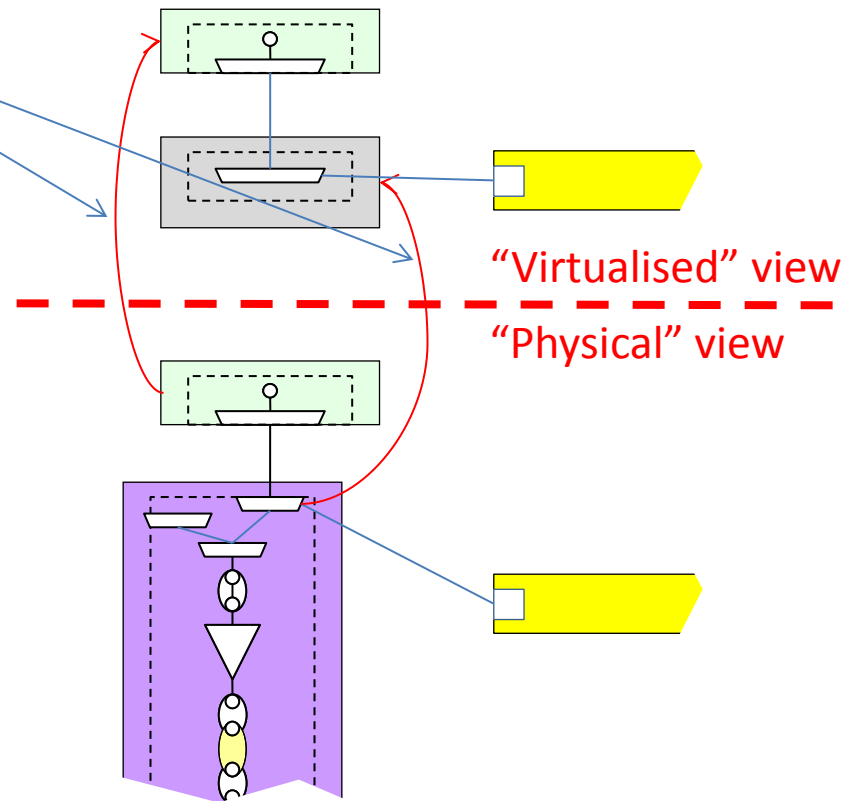
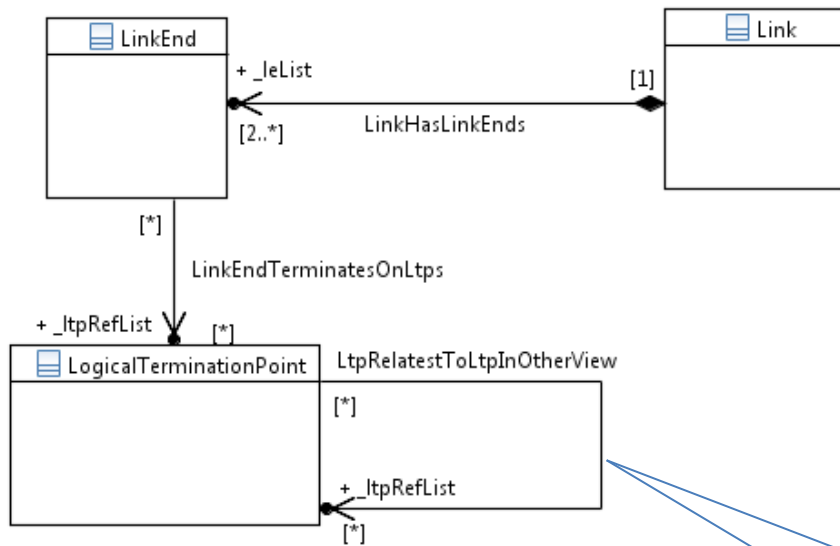




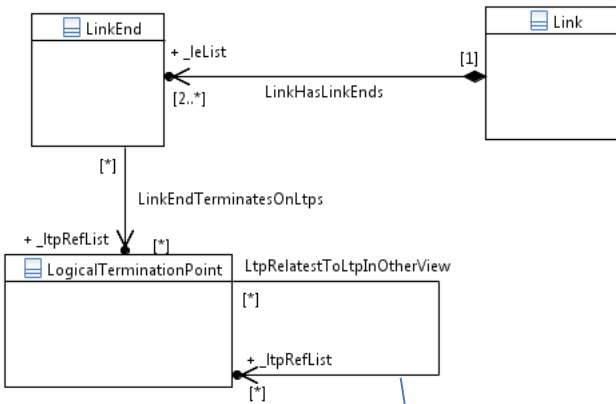
### Showing layering in elevation (above)



### Showing plan view and multiple channels (above)

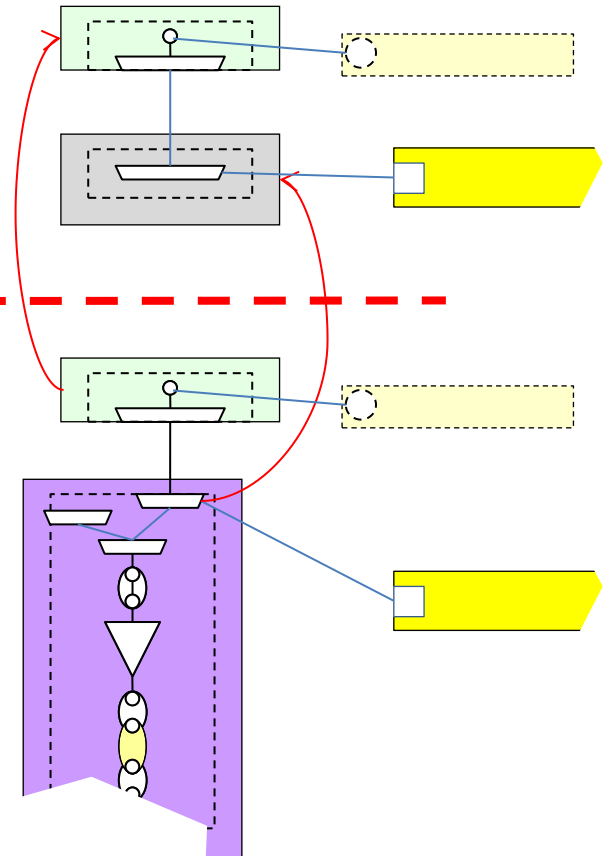
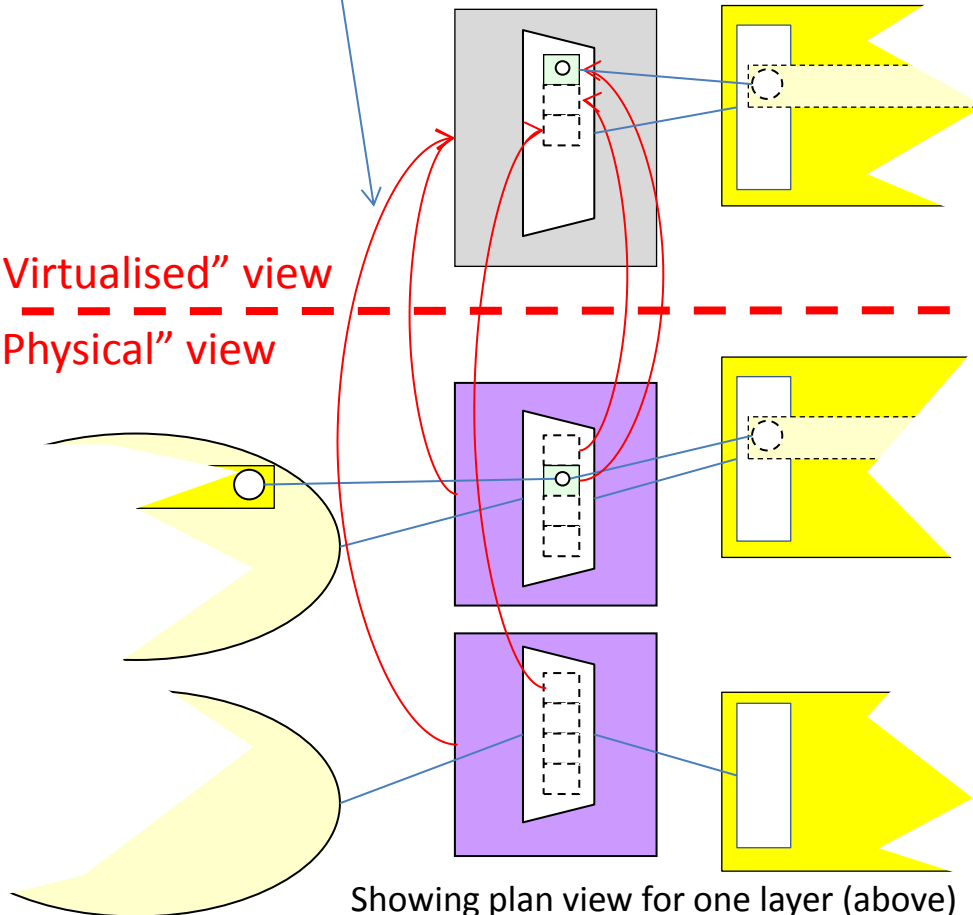


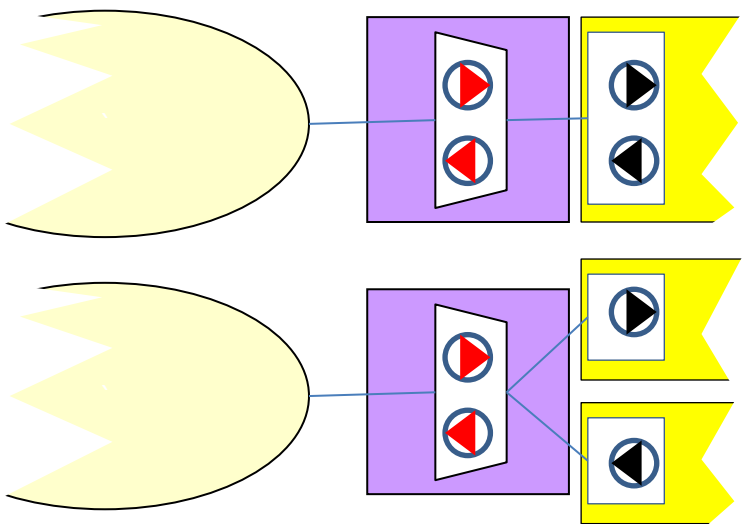
Note that the model view has been updated since the call with the ViewAbstractionRule. Although already in the model this was not shown on the figure presented. It is a preliminary piece of modelling that does require further work.

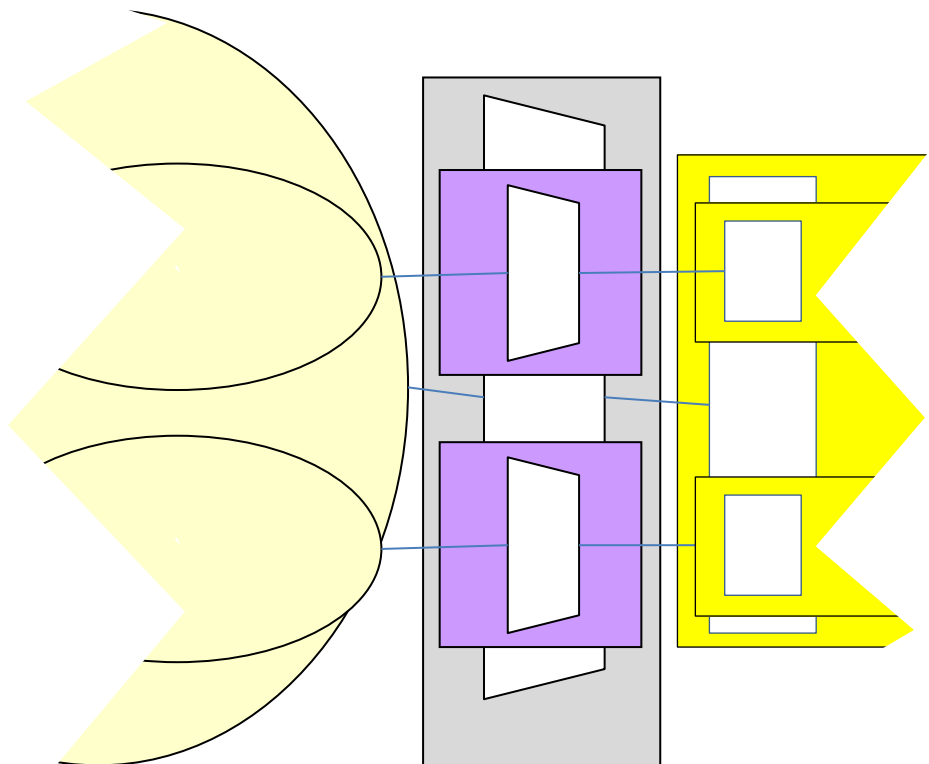


LtpRelatesToLtpInOtherView is used to relate LTPs at one level of virtualization with those at another

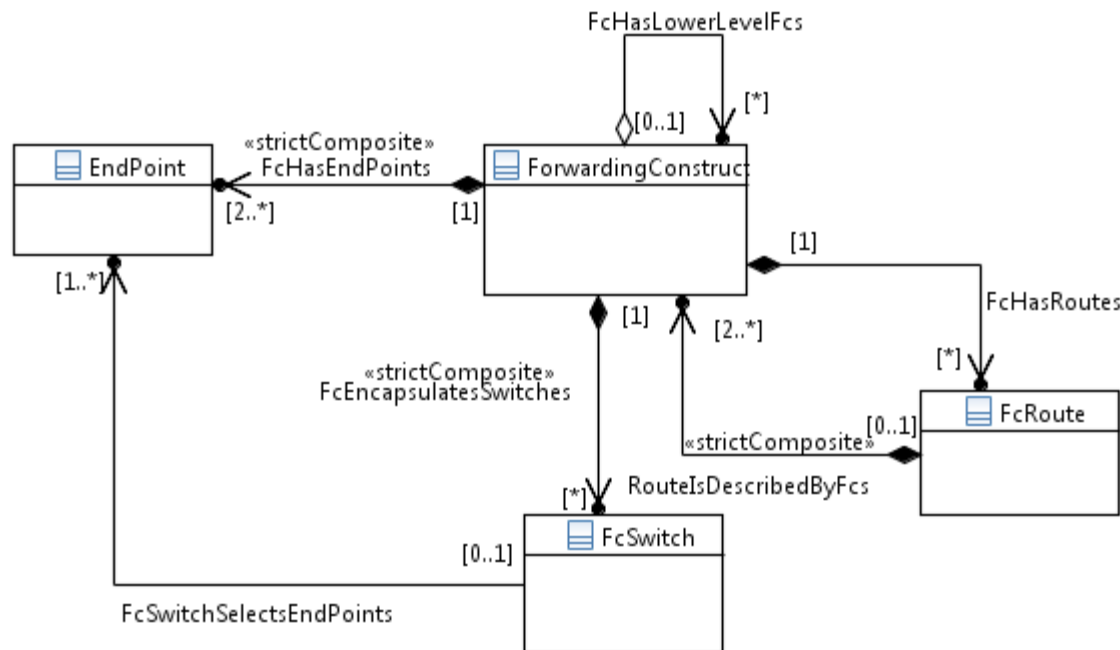
“Virtualised” view  
 “Physical” view







# Strict Composition



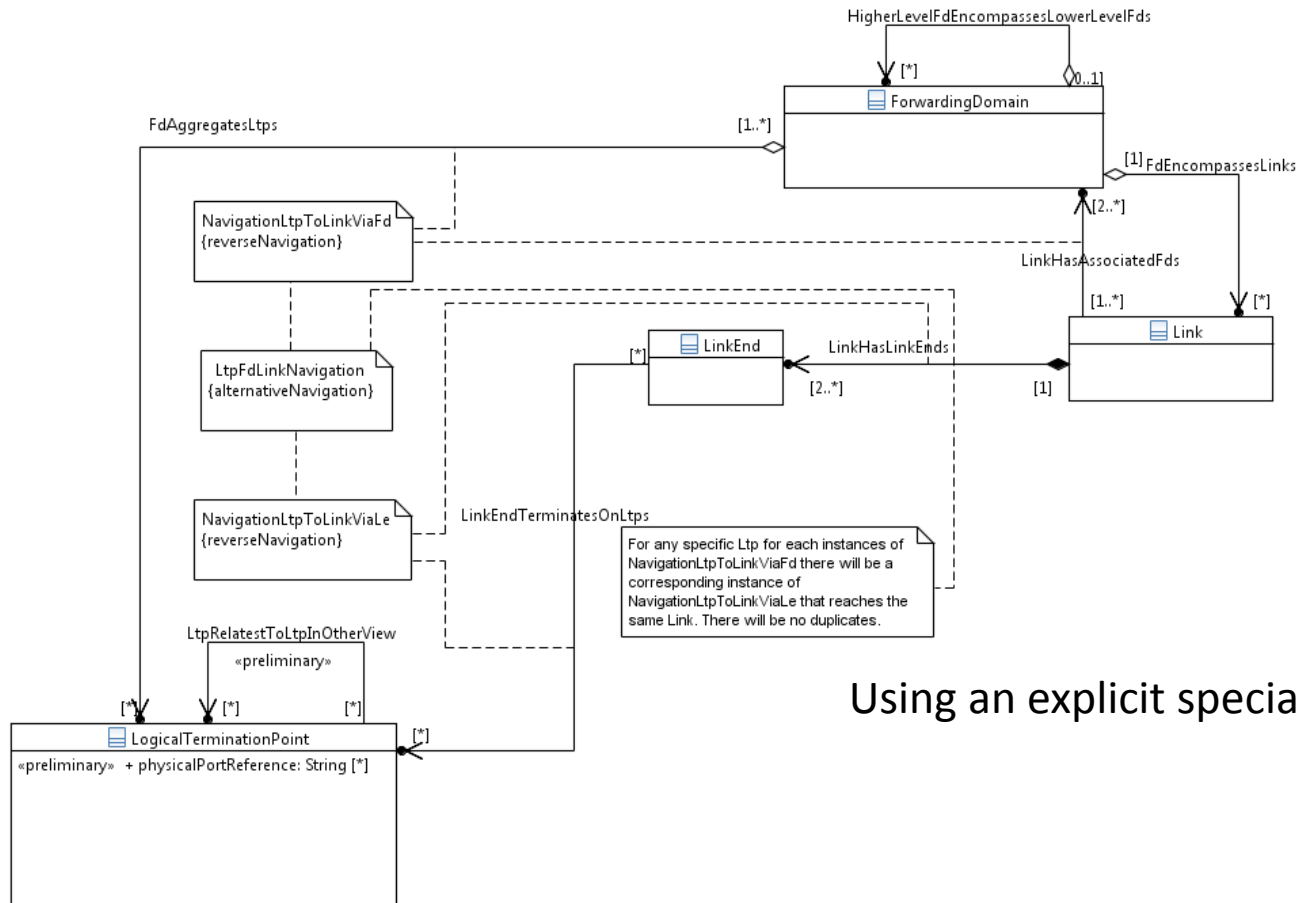
Using a stereo type to identify complete dependency.

So in the model above reporting the FC without its endpoints is not useful  
whereas reporting the FC without its FcRoute is useful

A route has a life bounded by that of the containing FC as does an EndPoint

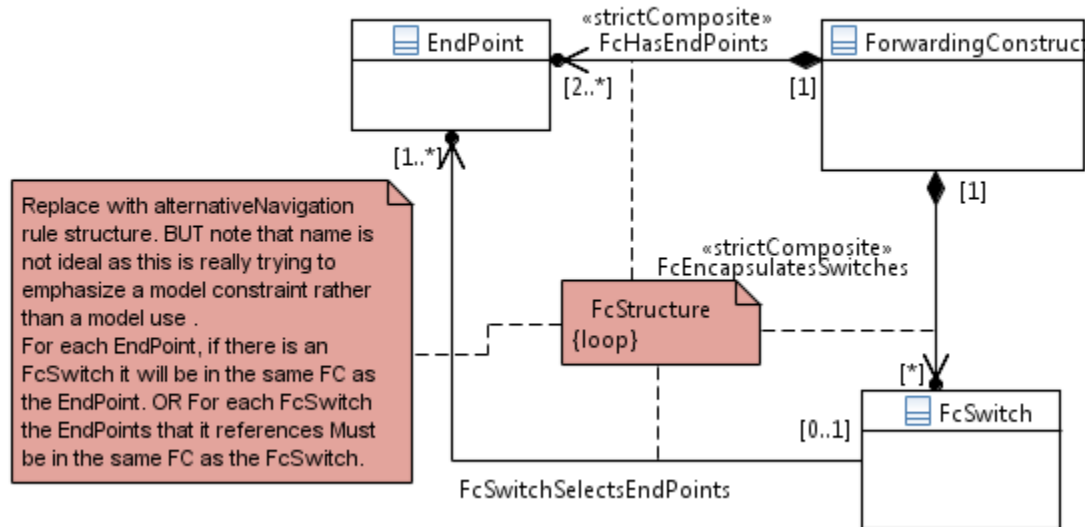


# Alternative navigation



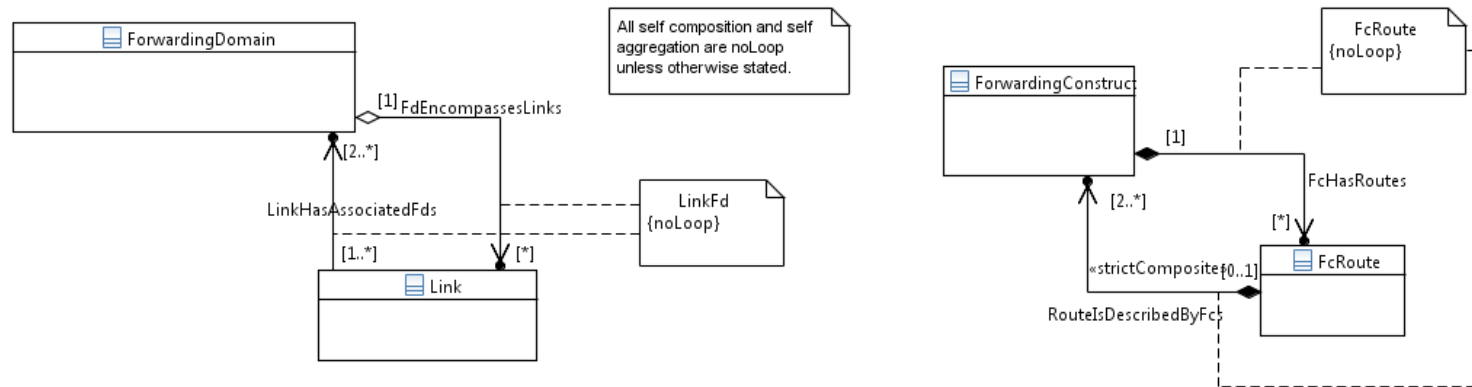
Using an explicit specialized rule

# Alternative navigation



# No Loop rule

Using an explicit specialized rule



# Multiplicity rationale

