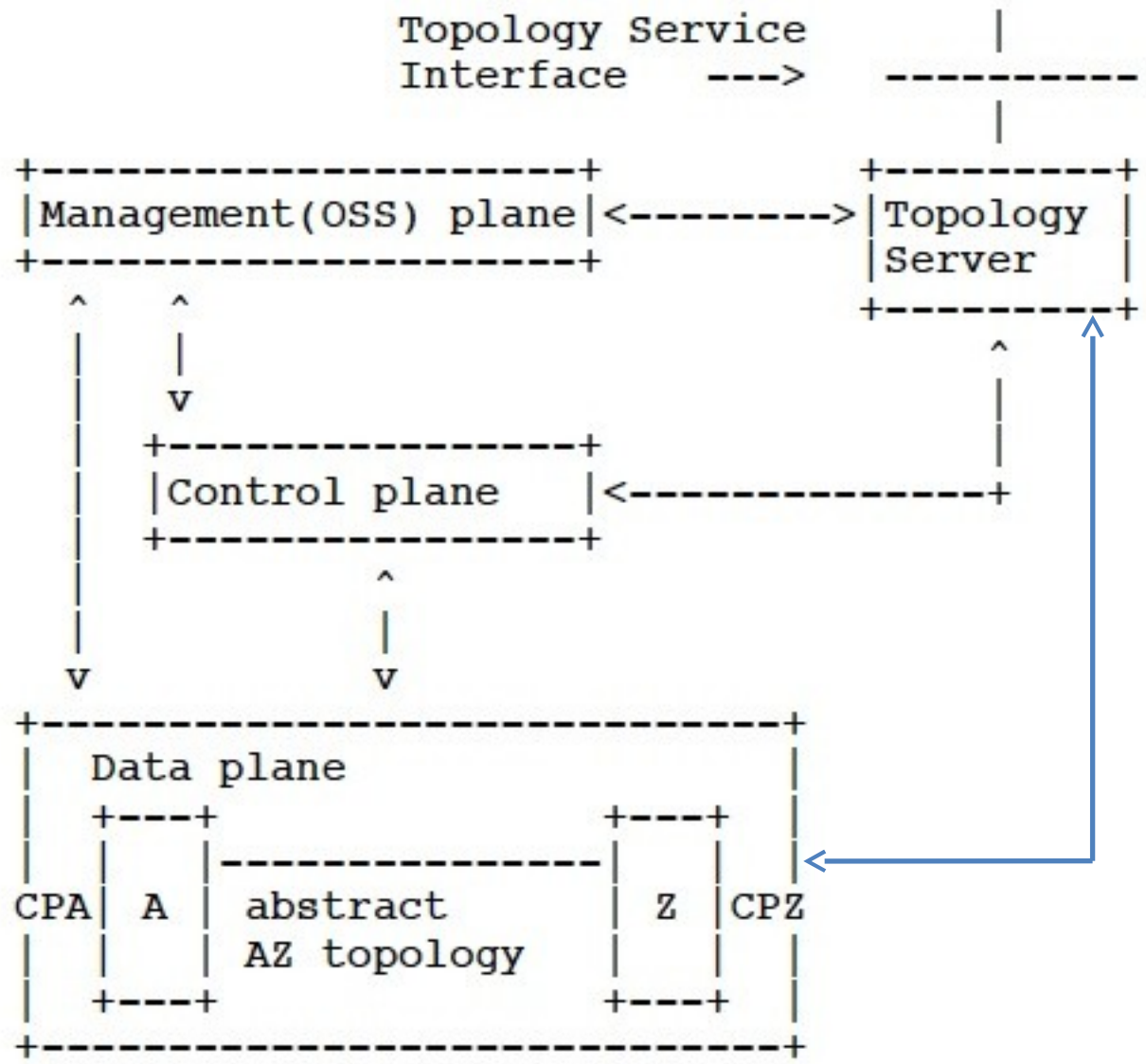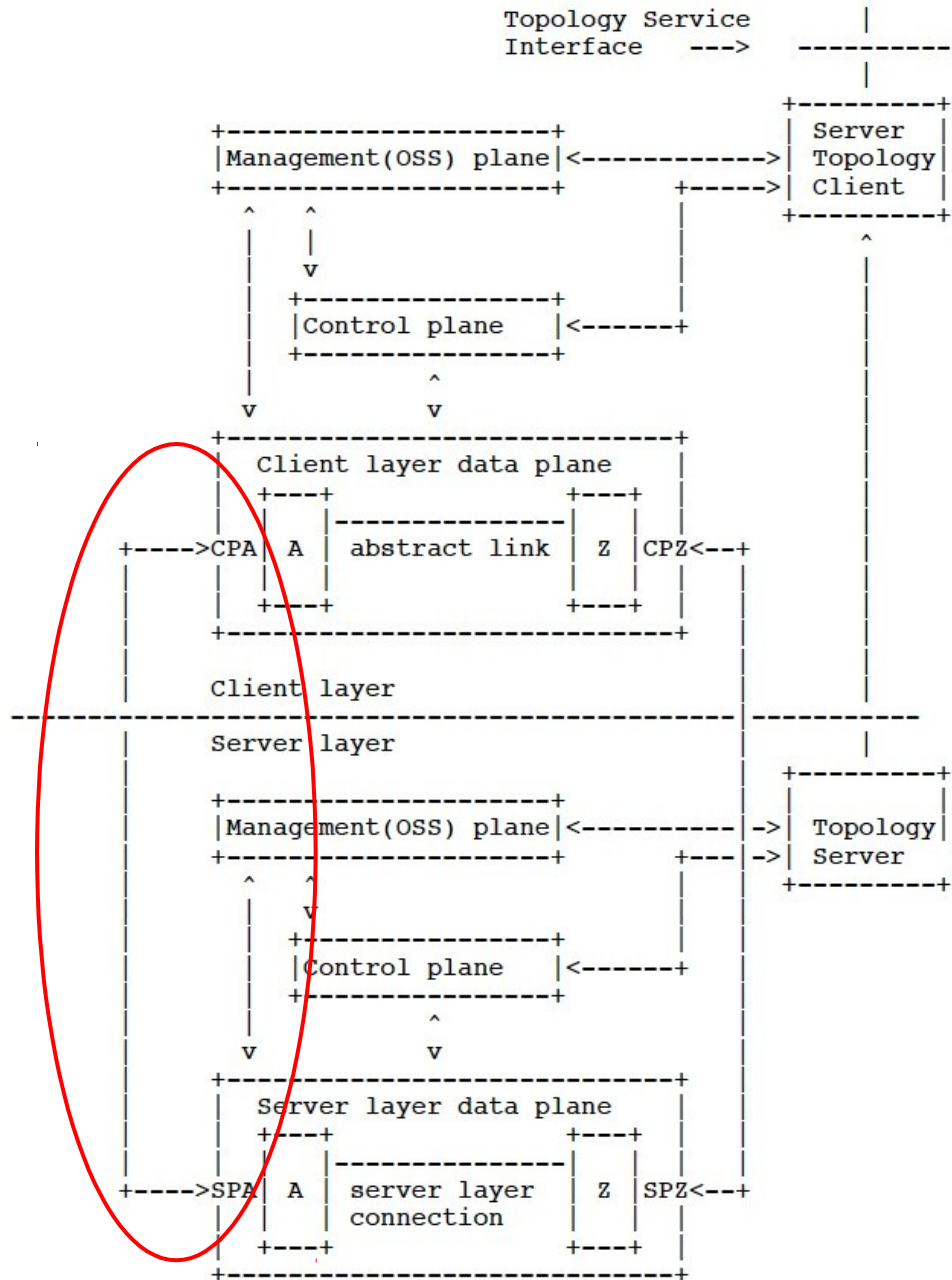draft-doolan-teas-te-topo-mlconsiderations-00

```
                   Topology Service        |
                   Interface       --->    -----------
                                                |
      +----------------------+            +----------+
      |Management(OSS) plane |<---------->|Topology  |
      +----------------------+            |Server    |
           ^        ^                     +----------+
           |        |                           ^
           |        v                           |
           |     +-----------------+            |
           |     |Control plane    |<-----------+
           |     +-----------------+
           |            ^
           |            |
           |            v
           v            v
      +--------------------------------------+
      |   Data plane                         |
      |    +---+                   +---+      |
      |    |   | ---------------   |   |      |
      |CPA | A |   abstract     |  | Z |CPZ   |<--------------+
      |    |   |   AZ topology  |  |   |      |               |
      |    +---+                   +---+      |               |
      +--------------------------------------+               |
```

```
                                Topology Service         |
                                Interface      --->     ----------
                                                          |
                                                        +----------+
          +---------------------+                       | Server   |
          |Management(OSS) plane|<-------------->|       | Topology |
          +---------------------+           +----->| Client   |
             ^      ^                        |       +----------+
             |      |                        |            ^
             |      v                        |            |
             |   +----------------+          |            |
             |   |Control plane   |<------+  |            |
             |   +----------------+       |  |            |
             |         ^                  |  |            |
             v         v                  |  |            |
          +----------------------------+  |  |            |
          |    Client layer data plane |  |  |            |
          |   +---+             +---+   |  |  |            |
          |   |   | ------------    |   |   | |            |
   +----->CPA| A |   abstract link | Z |CPZ<--+            |
          |   |   | |             | |   |   | |            |
          |   +---+               +---+   |  |            |
          +----------------------------+  |            |
             |                            |            |
          Client layer                    |            |
-------------------------------------------|------------
          Server layer                     |            |
             |                             |  +----------+
          +---------------------+          |  |          |
          |Management(OSS) plane|<----------|->| Topology |
          +---------------------+      +---|->| Server   |
             ^   ^                     |   |  +----------+
             |   |  v                  |   |
             |   | +----------------+  |   |
             |   | |Control plane   |<------+   |
             |   | +----------------+  |   |
             |        ^                |   |
             v        v                |   |
          +----------------------------+   |
          |    Server layer data plane |   |
          |   +---+             +---+   |   |
          |   |   | ------------    |   |   | |
   +----->SPA| A |  server layer   | Z |SPZ<--+
          |   |   | connection     |   |   | |
          |   +---+               +---+   |
          +----------------------------+
```
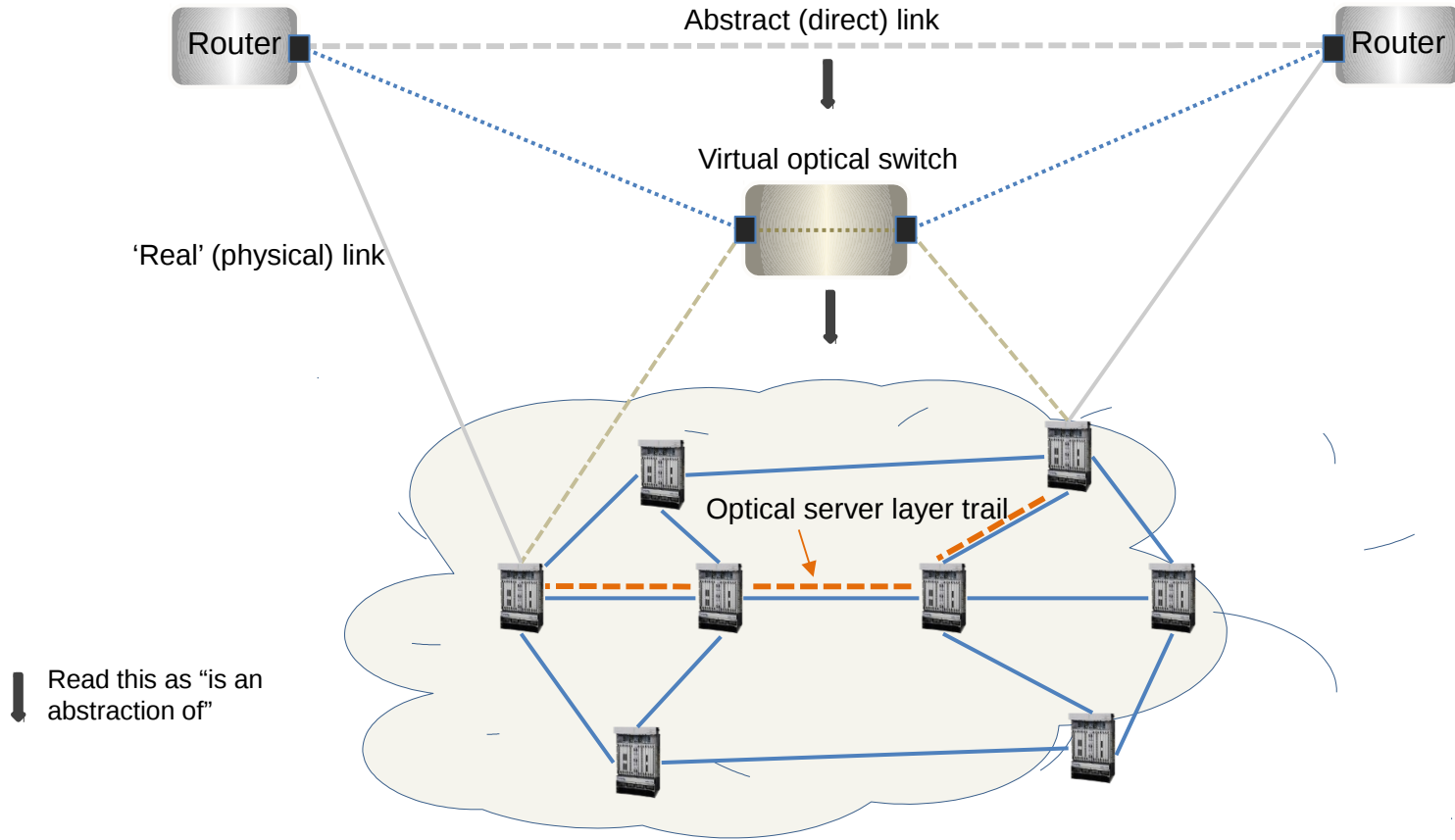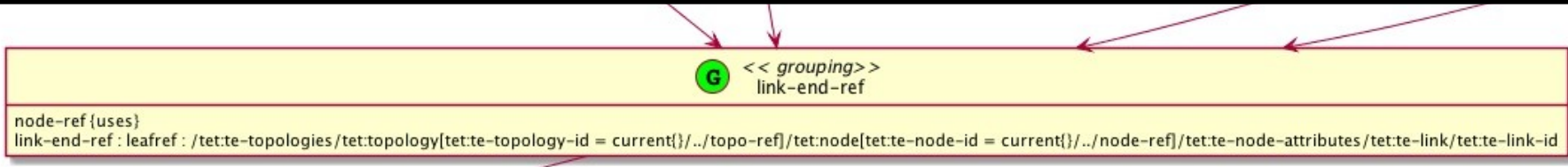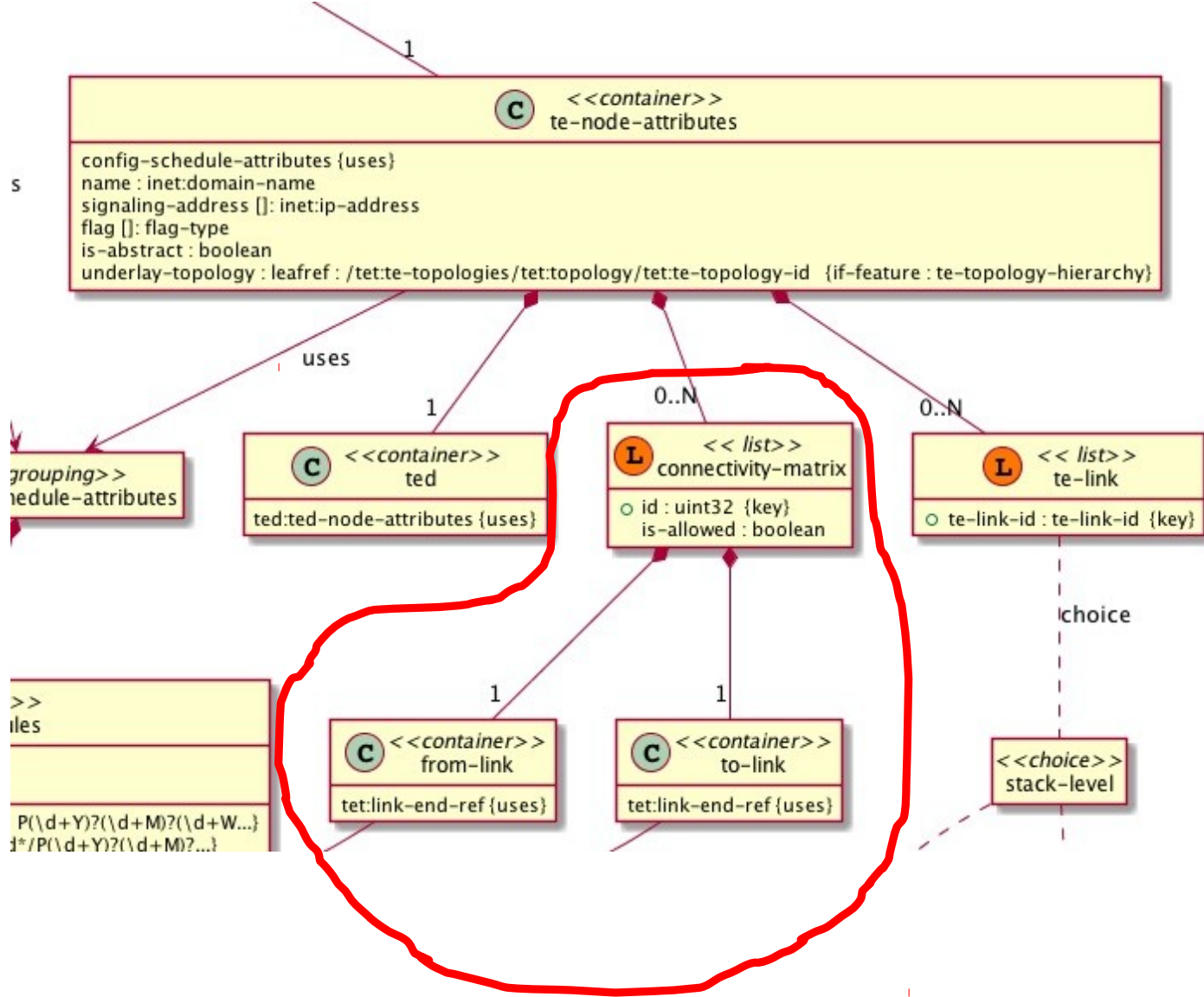
The stuff in 〇 is
here important:

- Policy boundary
- Transitional link
- Naming
  - Namespaces/formats
  - Names vs addresses (ouch)

- These problems are not new:
  - See work in OIF, ITU-T SG15 Q12, ONF
  - An interface that accommodates lessons from work elsewhere may have wider appeal
  - Some of the work elsewhere is based on (or generated) work here e.g. GMPLs extensions
- Other problems (opportunities ☺) remain
  - Some services with wide applicability can be supported by a *very* abstract model.
    - One example: router interconnect over an optical transport network

Virtual switch as an abstraction in a multilayer application

Abstract (direct) link

Router

Router

Virtual optical switch

'Real' (physical) link

Optical server layer trail

Read this as "is an abstraction of"

1

C <<*container*>>
te-node-attributes

config-schedule-attributes {uses}
name : inet:domain-name
signaling-address []: inet:ip-address
flag []: flag-type
is-abstract : boolean
underlay-topology : leafref : /tet:te-topologies/tet:topology/tet:te-topology-id   {if-feature : te-topology-hierarchy}

s

uses

grouping>>
hedule-attributes

1

C <<*container*>>
ted

ted:ted-node-attributes {uses}

0..N

L << *list*>>
connectivity-matrix

○ id : uint32 {key}
   is-allowed : boolean

0..N

L << *list*>>
te-link

○ te-link-id : te-link-id {key}

choice

>>
les

P(\d+Y)?(\d+M)?(\d+W...}
d*/P(\d+Y)?(\d+M)?...}

1

C <<*container*>>
from-link

tet:link-end-ref {uses}

1

C <<*container*>>
to-link

tet:link-end-ref {uses}

<<*choice*>>
stack-level

G << *grouping*>>
link-end-ref

node-ref {uses}
link-end-ref : leafref : /tet:te-topologies/tet:topology[tet:te-topology-id = current{}/../topo-ref]/tet:node[tet:te-node-id = current{}/../node-ref]/tet:te-node-attributes/tet:te-link/tet:te-link-id

Next steps

- WG agreement that this is a model/application that deserves our attention
- Discussion as to how best to implement requirements in the model
- Identification of and reconciliation with other relevant industry models
- Expansion of this draft to address BCP1 around these applications
  - E.g control vs configuration
  - OAM ?
  - Filtering ?
- Intention is to produce an updated draft with YANG specifications for next meeting.

1. KK – "do not call it Framework" ☺