DC aware TE topology model

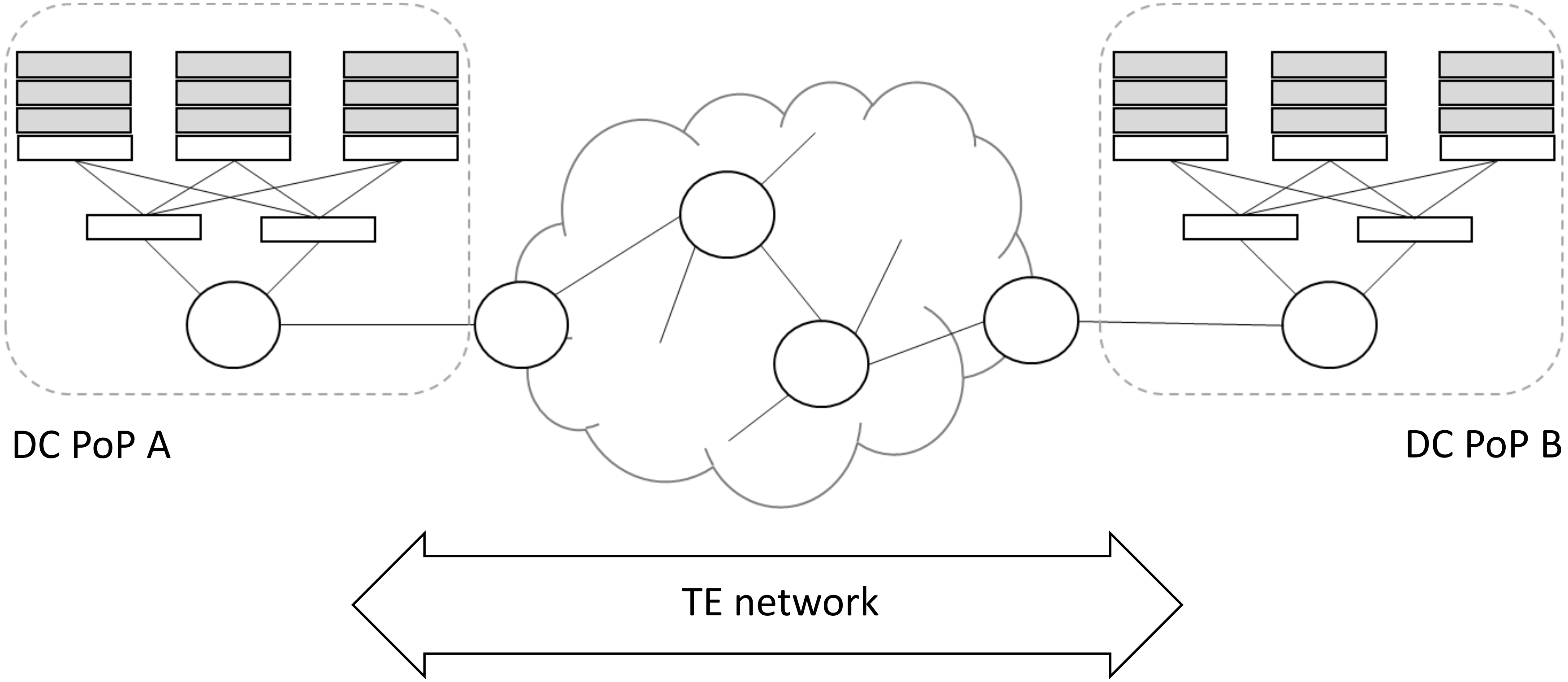
draft-llc-teas-dc-aware-topo-model-04

Luis M. Contreras *(Telefonica)*

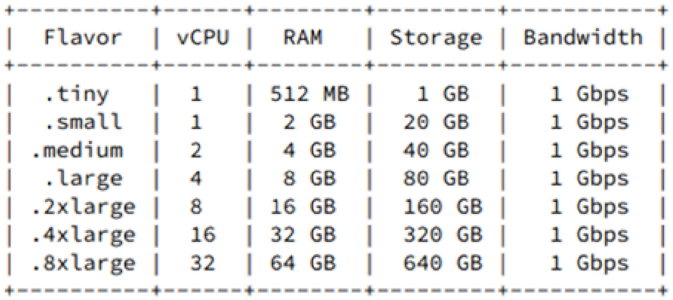
Xufeng Liu *(Alef Edge)*

Problem statement

- Wide deployment of computing facilities across service provider’s Networks, in the form of DC PoPs (as edge and/or central cloud)
- Interesting **to have joint topological view of both networking and computing resources** available to assist on **TE decisions that could require combined awareness of network and compute domains**
- Similar approach as the one followed in *draft-ietf-teas-sf-aware-topo-model* but concentrated on available DC resources instead of functions



- DC PoPs described in terms of resource capabilities such as CPU, memory, storage, etc
- Alternatively, they could be described in terms of resource bundles (quotas, flavors)



Flavor	vCPU	RAM	Storage	Bandwidth
.tiny	1	512 MB	1 GB	1 Gbps
.small	1	2 GB	20 GB	1 Gbps
.medium	2	4 GB	40 GB	1 Gbps
.large	4	8 GB	80 GB	1 Gbps
.2xlarge	8	16 GB	160 GB	1 Gbps
.4xlarge	16	32 GB	320 GB	1 Gbps
.8xlarge	32	64 GB	640 GB	1 Gbps

draft-llc-teas-dc-aware-topo-model

- Attempt to provide a model for characterizing the resource-related information of a compute domain, in a per DC PoP basis
  - The goal is to integrate such compute information, integrated with the topological information of the network
- Cloud managers (Kubernetes, OpenStack) as means for collecting compute node information (via APIs)
- Different cloud solutions impose different ways of modelling the compute resources and assets
  - E.g., OpenStack, Kubernetes

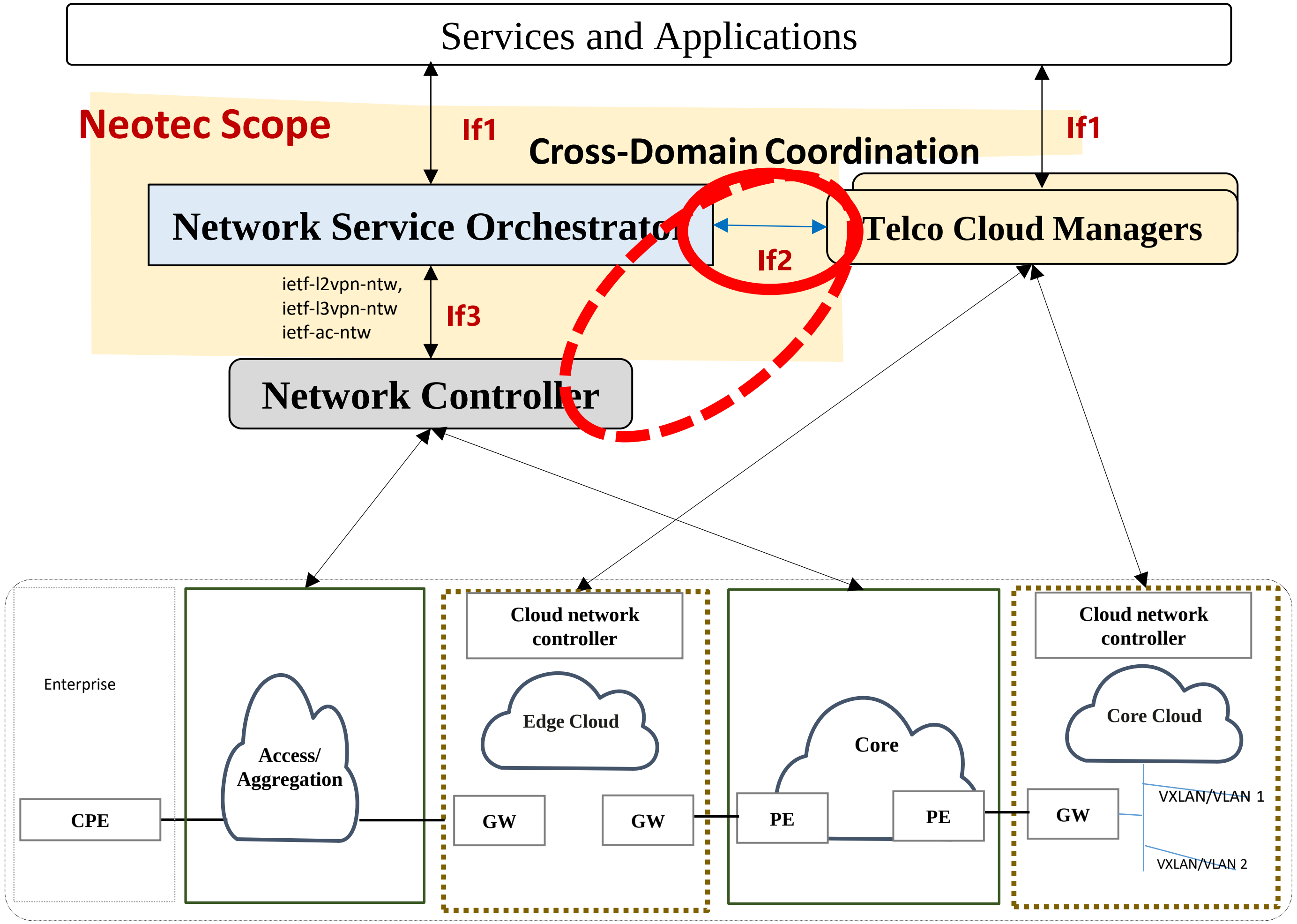
Examples

```
module: ietf-kubernetes-info
       +--rw dcpop
          +--rw dcpop-id?   string
          +--rw dc* [id]
             +--rw kubernetes
                +--rw nodes
                |  +--rw node* [id]
                |     +--rw name                string
                |     +--rw cpu
                |     |  +--rw capacity         uint64
                |     |  +--rw allocatable      uint64
                |     |  +--rw usage            uint64
                |     +--rw memory
                |     |  +--rw capacity         uint64
                |     |  +--rw allocatable      uint64
                |     |  +--rw usage            uint64
                |     +--rw pods
                |        +--rw max-pods         uint32
                |        +--rw running-pods     uint32
                +--rw pods
                   +--rw pod* [id]
                      +--rw namespace          string
                      +--rw name               string
                      +--rw cpu
                      |  +--rw request         uint64
                      |  +--rw limit           uint64
                      |  +--rw usage           uint64
                      +--rw memory
                      |  +--rw request         uint64
                      |  +--rw limit           uint64
                      |  +--rw usage           uint64
                      +--rw status
                         +--rw phase           enumeration
                         +--rw conditions*     string
```

```
module: ietf-openstack-info
    +--rw dcpop
       +--rw dcpop-id?   string
       +--rw dc* [id]
          +--rw openstack
             +--rw compute-nodes
             |  +--rw node* [name]
             |     +--rw name                   string
             |     +--rw vcpus
             |     |  +--rw total               uint64
             |     |  +--rw allocated           uint64
             |     |  +--rw used                uint64
             |     +--rw memory
             |     |  +--rw total               uint64
             |     |  +--rw allocated           uint64
             |     |  +--rw used                uint64
             |     +--rw instances
             |        +--rw max-instances       uint32
             |        +--rw running             uint32
             +--rw instances
                +--rw instance* [id]
                   +--rw id                    string
                   +--rw name                  string
                   +--rw project-id            string
                   +--rw vcpus
                   |  +--rw allocated          uint64
                   |  +--rw limit              uint64
                   |  +--rw used               uint64
                   +--rw memory
                   |  +--rw allocated          uint64
                   |  +--rw limit              uint64
                   |  +--rw used               uint64
                   +--rw status
                      +--rw state              enumeration
                      +--rw conditions*        string
```

Network Telco-cloud Orchestration Interfaces Goal

- **Interface 1(If 1) : 1)Intent-driven service deployment and scaling policy with service and SLO requirements can be directly mapped to cloud-network alliance policies. E.g. low-latency 100ms service, the system automatically selects edge nodes whose latency is less than 100 ms and reserves dedicated network bandwidth for the node. 2) Cloud aware network topology and metrics information**
- **Interface 2(If 2) : Cloud exposes the resource and operation metrics to the orchestrator, for network aware service placement and scaling policies**
- **Interface 3(If 3) : Network exposes the resource and operation metrics to the orchestrator for cloud resource aware network connectivity's and service QoS policy**



Note: Telco Cloud Managers could be NFVO, OpenStack or Kubernetes platform

Neotec Goal:

① Central Cloud: Elastic Scaling on Demand (Millisecond-Level Scaling)

② Edge Cloud: Deterministic Low Latency (<10ms End-to-End)

③ Global Efficiency: Cross-Domain Resource Utilization Improved, and agile Domain Coordination

Need for the model

- Means for an optimal orchestration in the telecom-cloud network operation
- Facilitator of interoperability through proper resource abstraction
- Enabler seamless integration of cloud and network services
- Allows the exposure of combined network-cloud information
- Permits a more efficient dynamic resource allocation cross-domain (i.e., for cloud and network)