

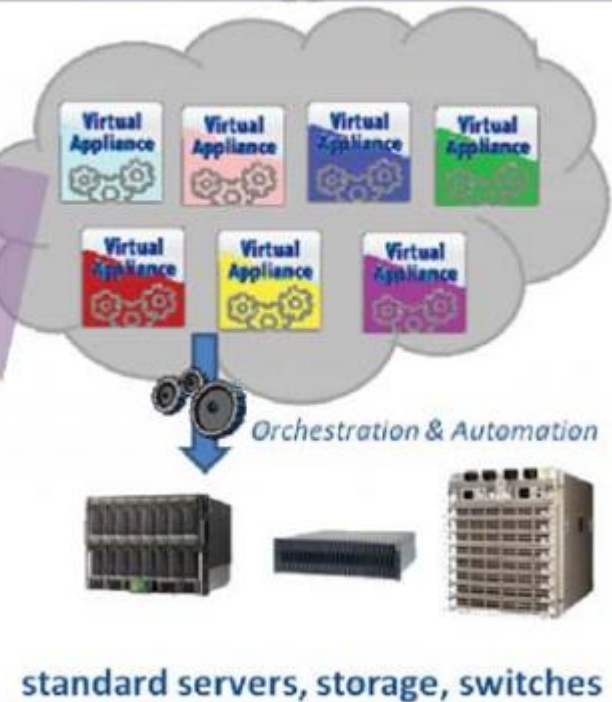


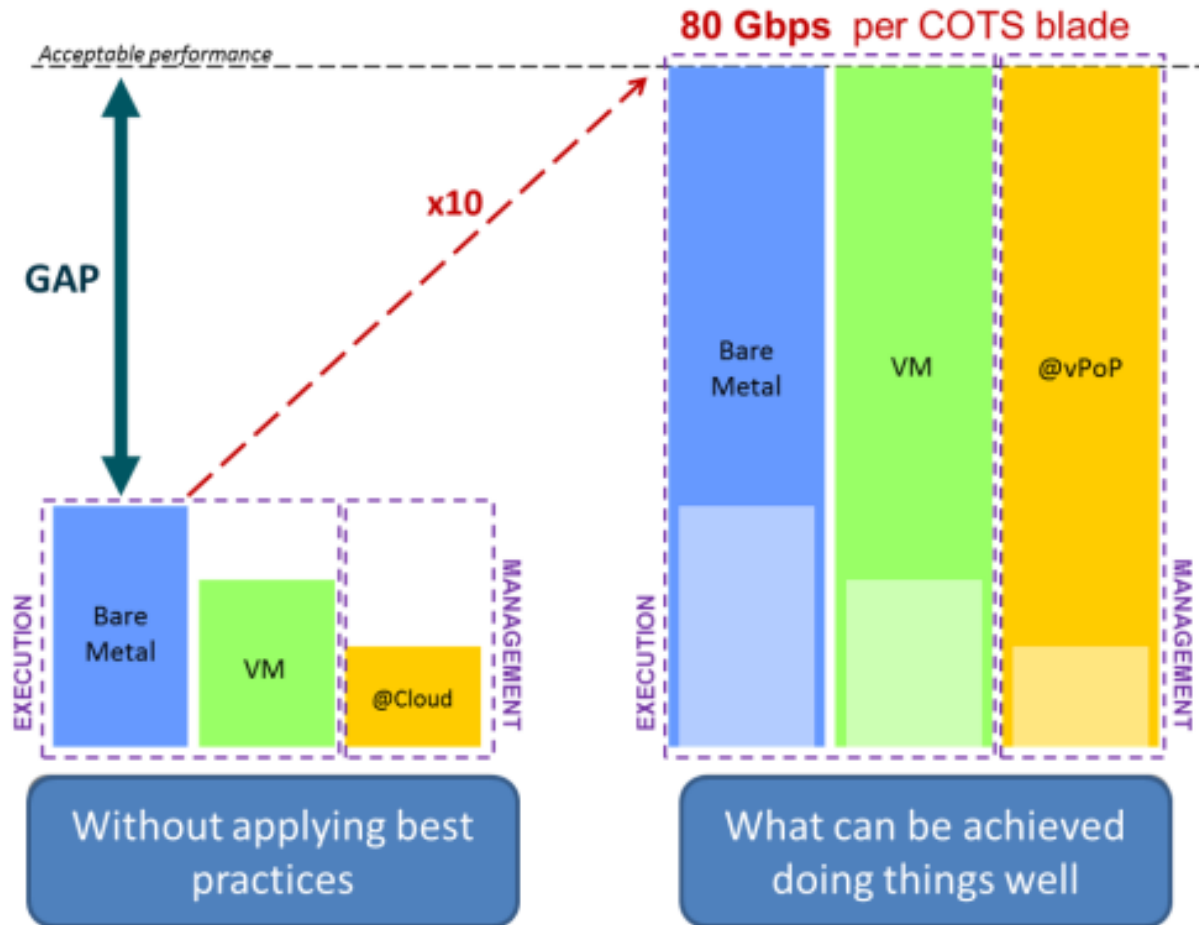
Network Functions Virtualization

Classical Network Model: Hardware Appliances



The New Network Model: Virtual Appliances





The goal has been achieved at laboratory, but now we should go into production!

NFV Potential Use Cases

USE CASE	DESCRIPTION
Virtualization of mobile core network nodes	Virtualization of core network nodes, including IMS. Affected functions could include packet data network gateways, serving gateways, mobility management entities, and mobile home subscriber servers.
Virtualization of mobile base stations	Aims at realizing the base station function (at least specific functional block) with software based on standard IT platform. Mainly focused on LTE LTE-A, but similar concept can be applied to 2G, 3G and WiMax.
Virtualized home environment	Aims to shift functionality away from the home to a network-located environment as a way to solve many installation and lifecycle upgrade problems, consolidating the corresponding workloads into equipment installed in the network operator premises. Virtualization targets include: residential gateway; set-top box; WiFi access points; home eNodeB.
Virtualized network function as a service	Possible virtualization targets: enterprise access router/enterprise CPE, provider edge router, enterprise firewall, enterprise NG-FW, enterprise WAN optimization, deep packet inspection (appliance or a function), IPS – and other security appliances, network performance monitoring.
Service chains with NFV	Virtualizing the appliance functions and putting them into applications on a server in a single location or area – making service analysis more efficient and streamlining the flow of traffic in the network.
Virtualization of CDNs	Virtualization of content delivery networks (CDNs) potentially covers all components of the CDN, though the initial impact would probably be on cache nodes for achieving acceptable performance (e.g., throughput, latency).

Low and Mid-End Router ≠ Aggregation and High-End Router

NFV Use Case: Virtual CPE Router

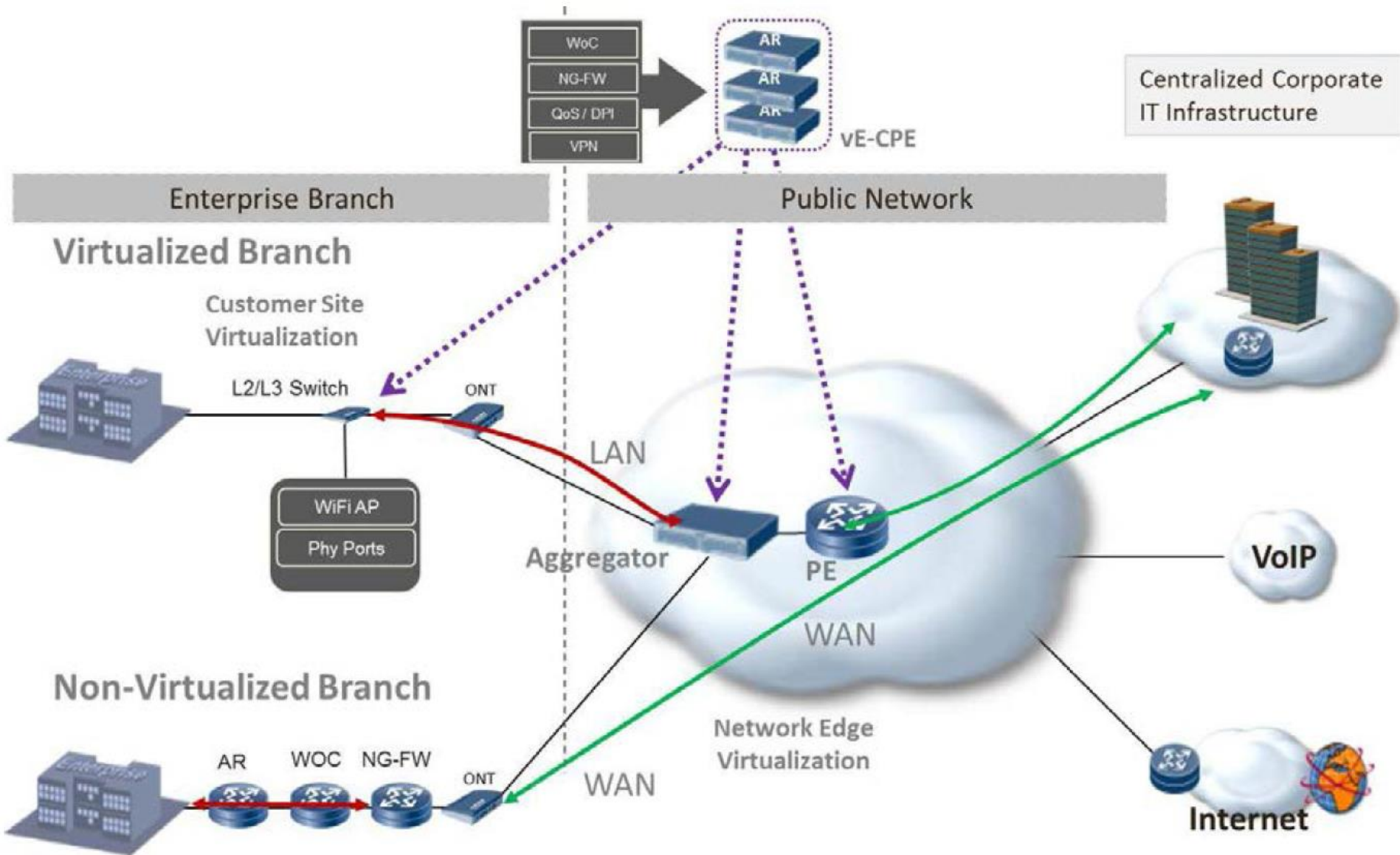
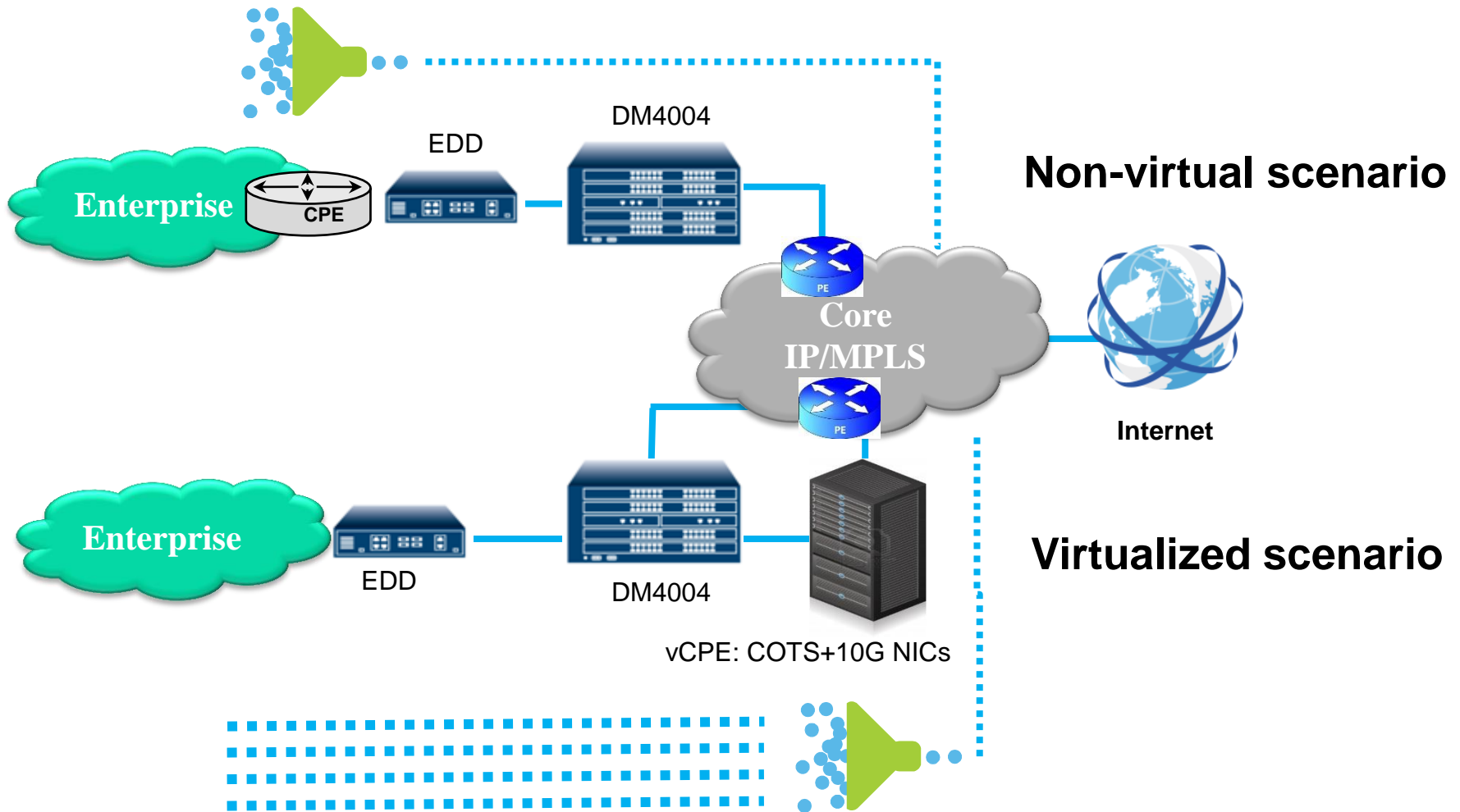


Figure from ETSI® GS NFV 001 V1.1.1 (2013-10)

Market Opportunity: NFV coupled with Ethernet Access



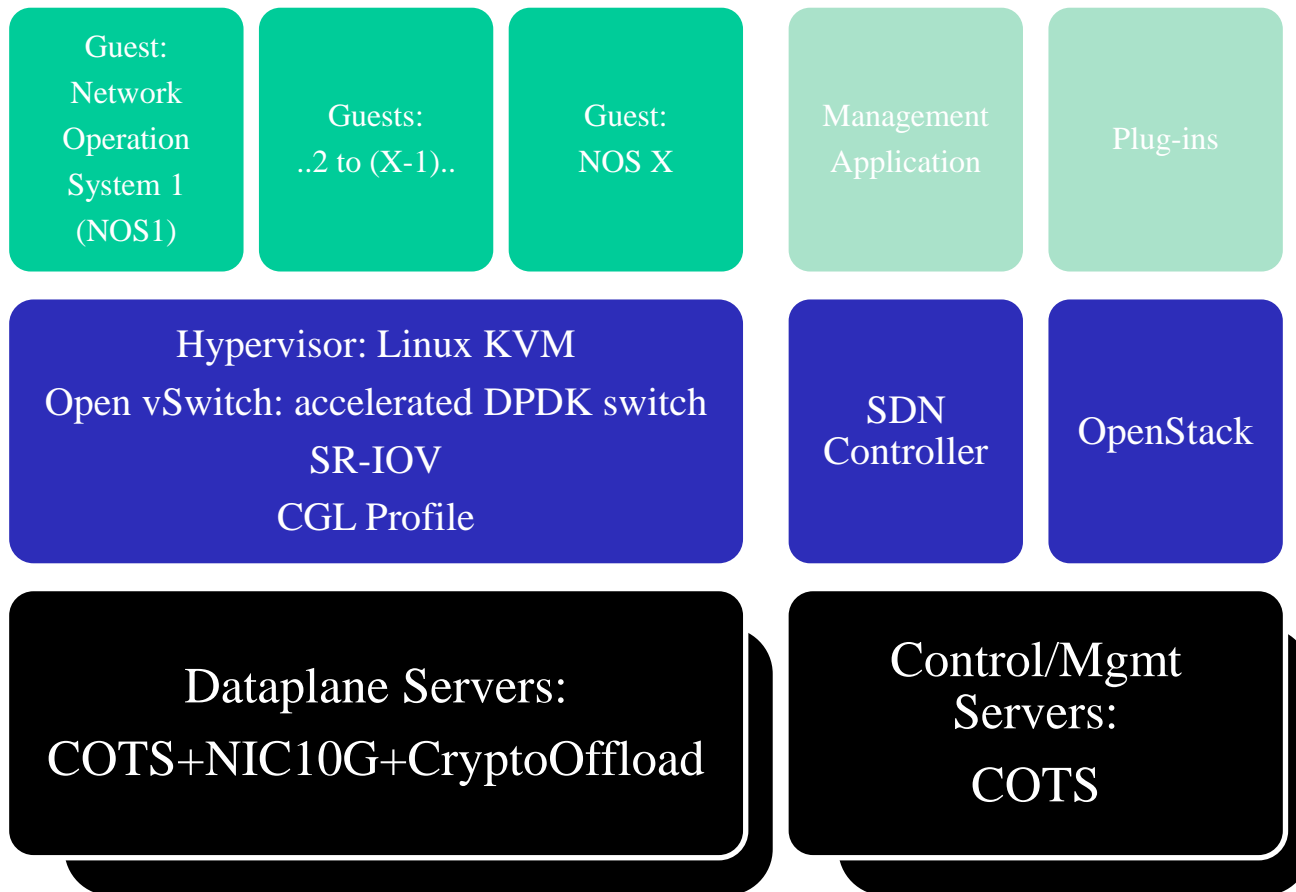
Virtualization Targets:

- AR – Enterprise Access Router => Scale Up and Down
 - [client] Can I upgrade my circuit?
 - [provider] Yes! You have 100M now, and a dual homed 1G access. How much do you want, from 100M up to 1G?
- FW / NG-FW – Enterprise Firewall => Managed Firewall as a Service
 - [client] Can I add a Managed Firewall to my link?
 - [provider] Yes! What are the basic rules that you would like to start with? I can add this to your circuit right now for xxxx/Month.
- Network Performance Monitoring => tight integration with the OSS/BSS

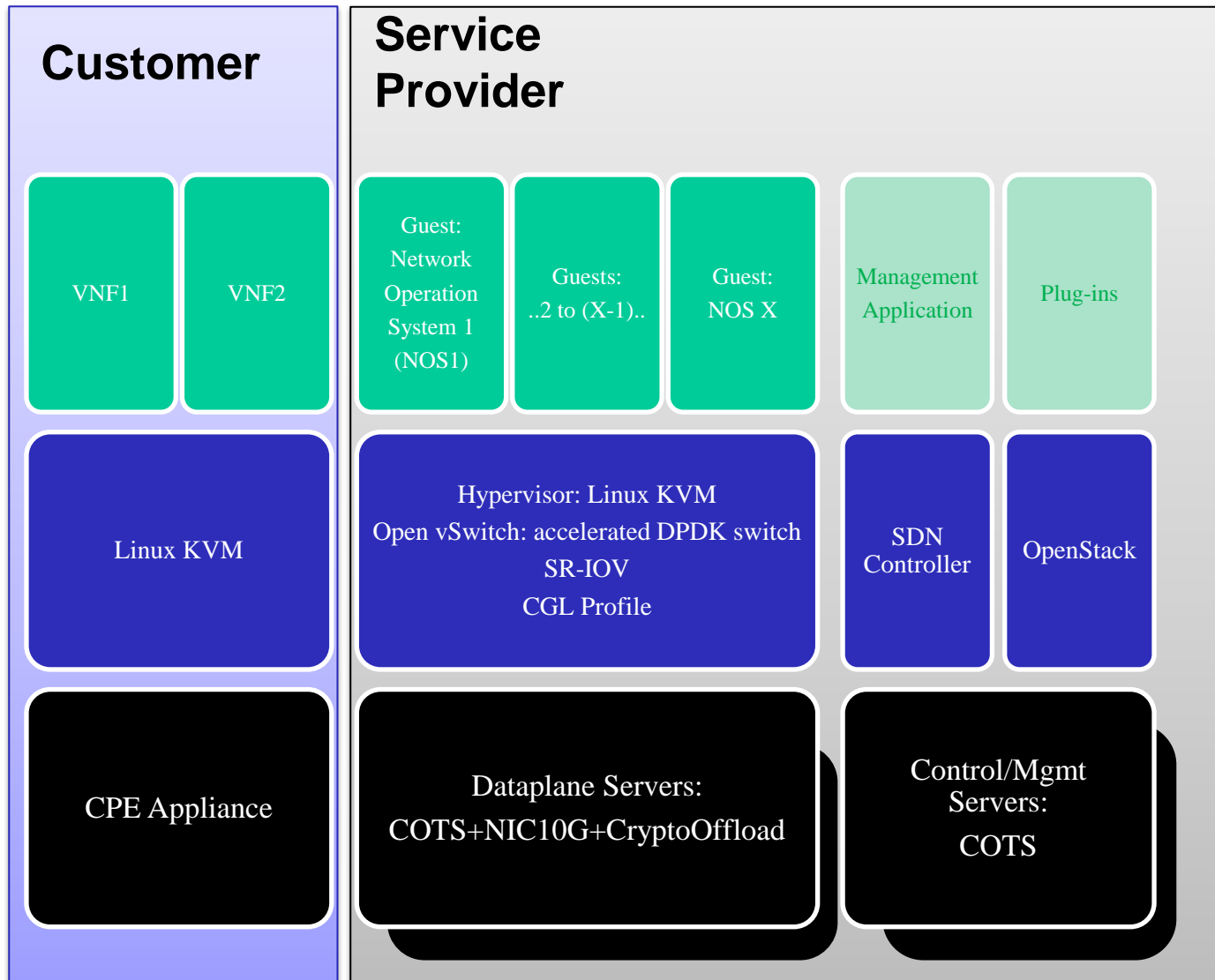
Other possibilities => XaaS (everything as a service!)

- WOC – WAN Optimization Controller
- DPI – Deep Packet Inspection
- IDS / IPS – Intrusion Detection/Protection System

NFV Infrastructure

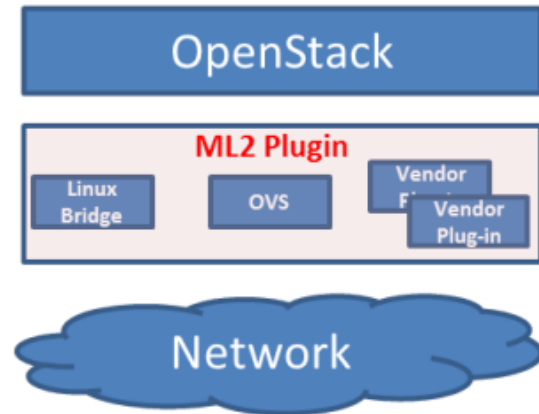
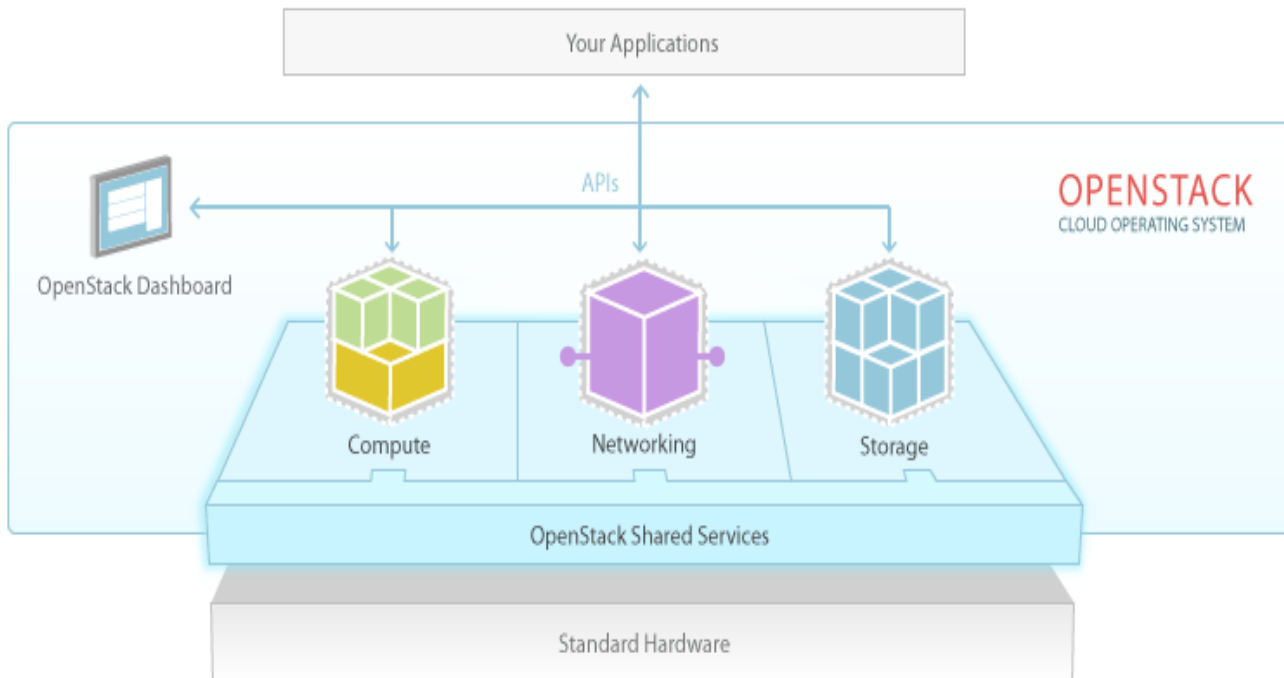


Distributed NFV Infrastructure



OPENSTACK ICEHOUSE:

- Linux KVM
- Neutron ML2 Plugin
 - Linux Bridge
- SR-IOV



OPENSTACK NEUTRON

Use Case: DmView

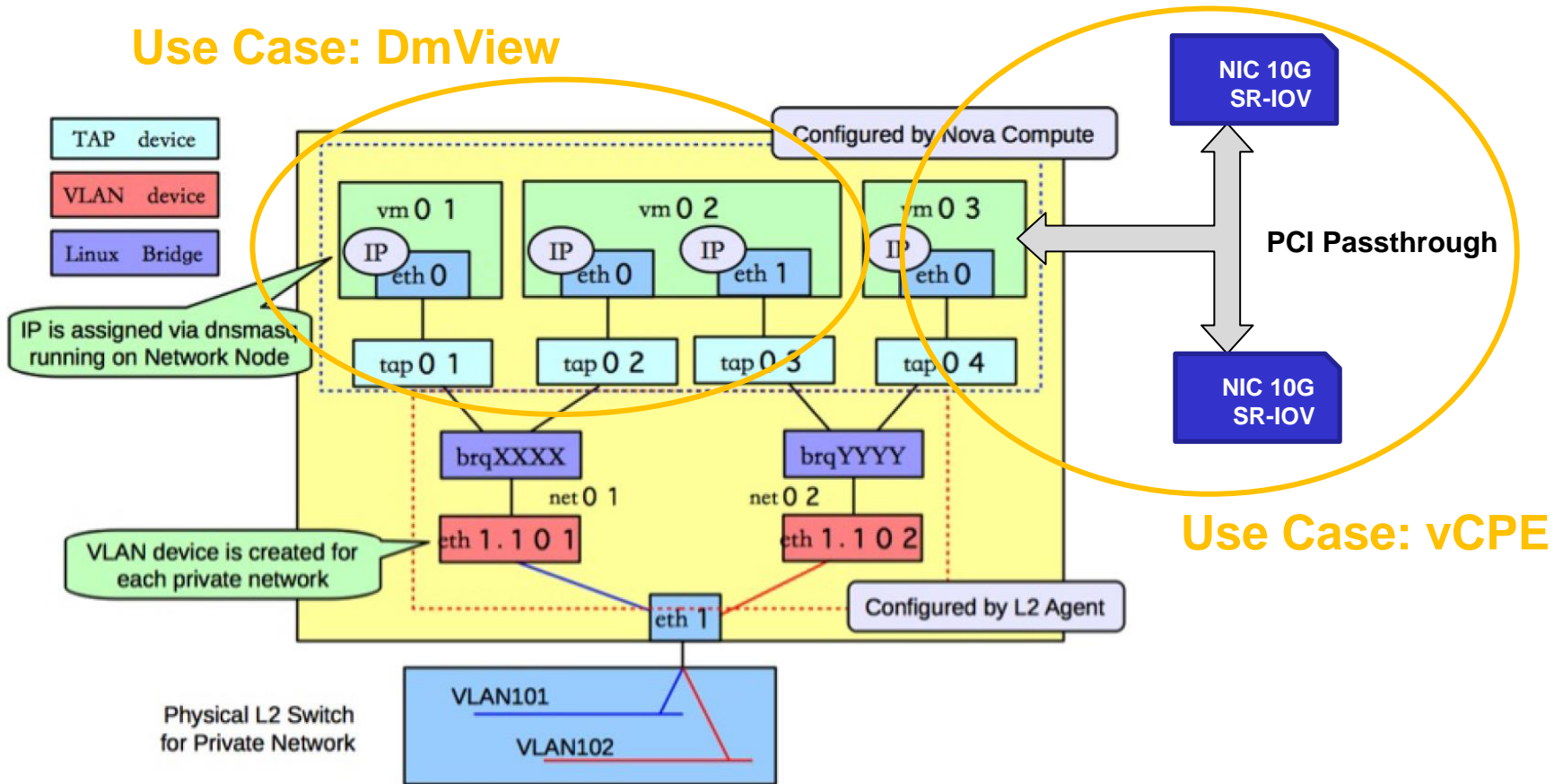
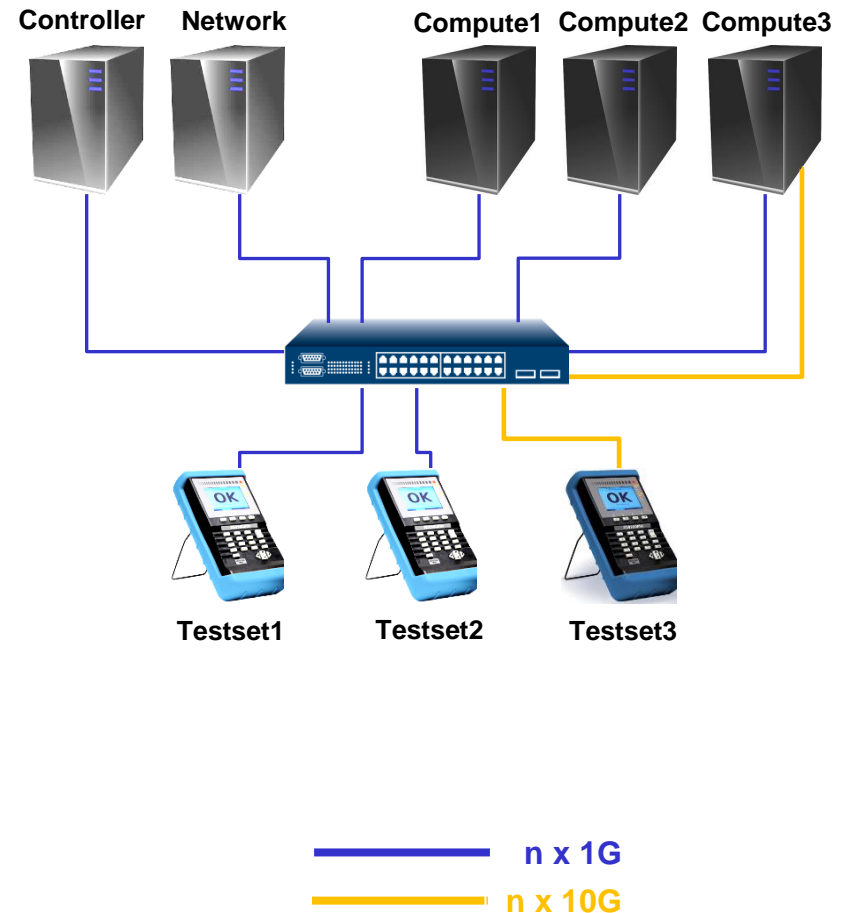


Figure 7.10. Linux Bridge: Scenario 1: Compute host configuration

Physical Infrastructure:

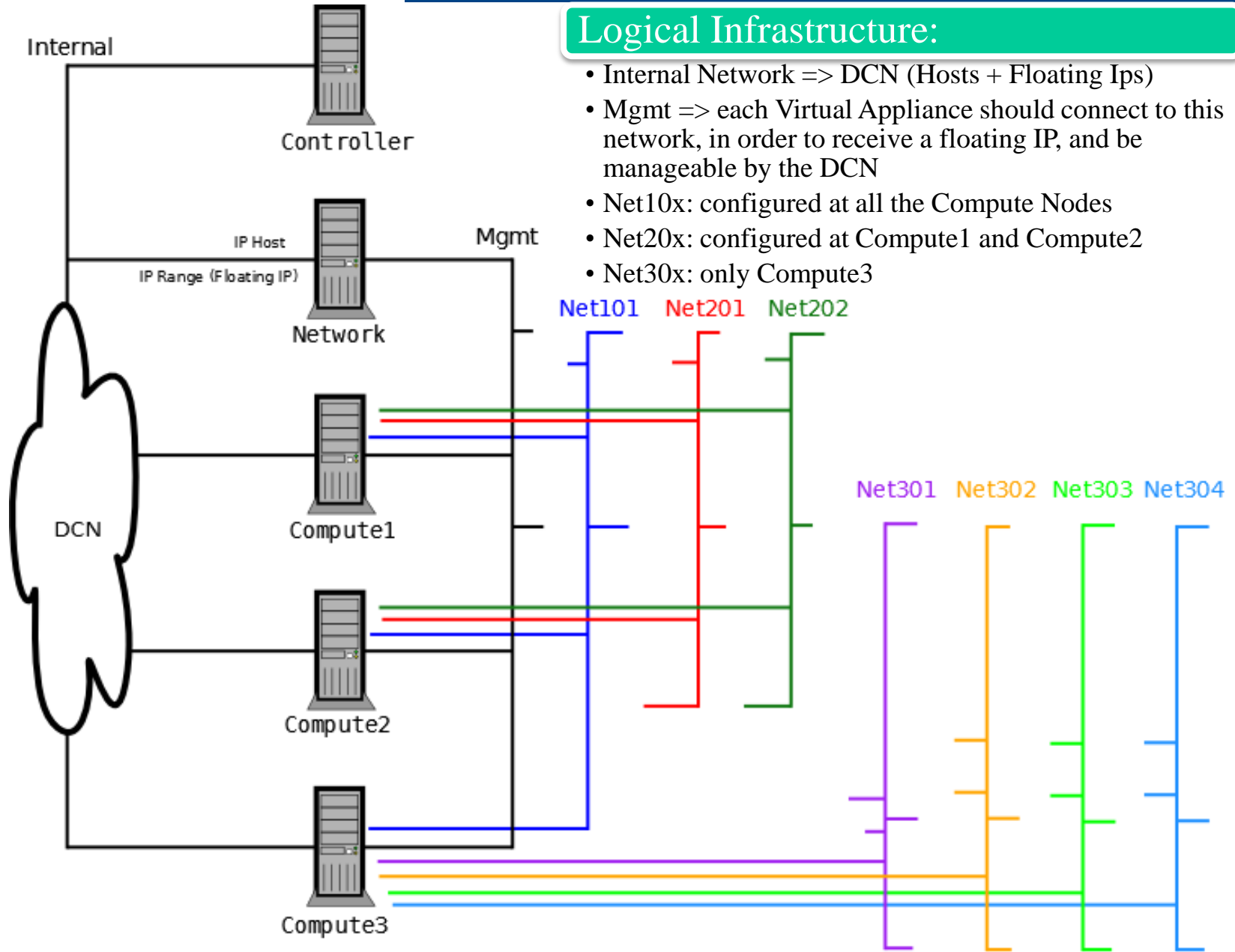
- Virtual Machine: Controller Node
- Virtual Machine: Network Node
- Bare Metal : Compute Node 1
- Bare Metal : Compute Node 2
- Bare Metal : Compute Node 3
- Switch: DM4100 24GT+4XX
- Testset 1: TSW900ETH 2x1G
- Testset 2: TSW900ETH 2x1G
- Testset3: BL2000A 2x10G



Testbed

Logical Infrastructure:

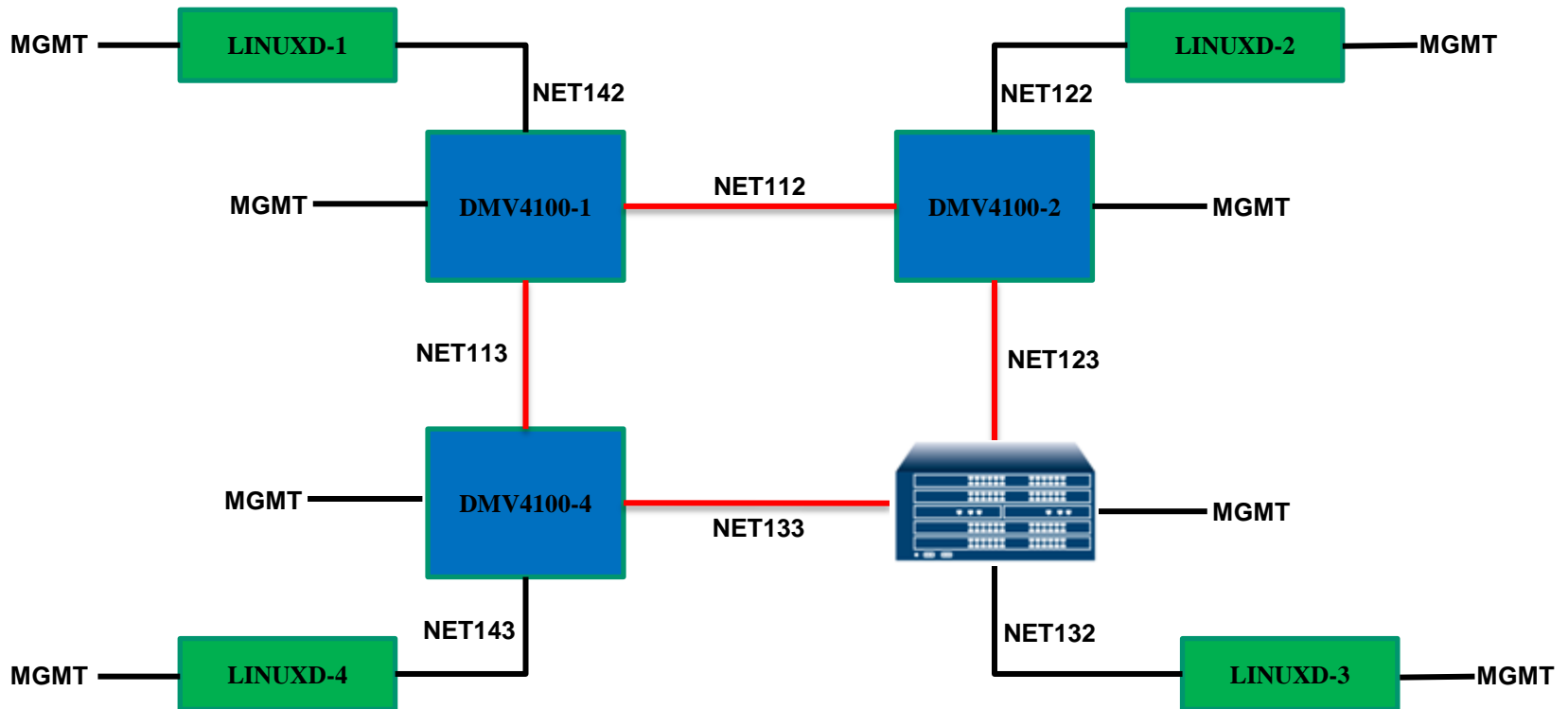
- Internal Network => DCN (Hosts + Floating Ips)
- Mgmt => each Virtual Appliance should connect to this network, in order to receive a floating IP, and be manageable by the DCN
- Net10x: configured at all the Compute Nodes
- Net20x: configured at Compute1 and Compute2
- Net30x: only Compute3



Use Case: DmView

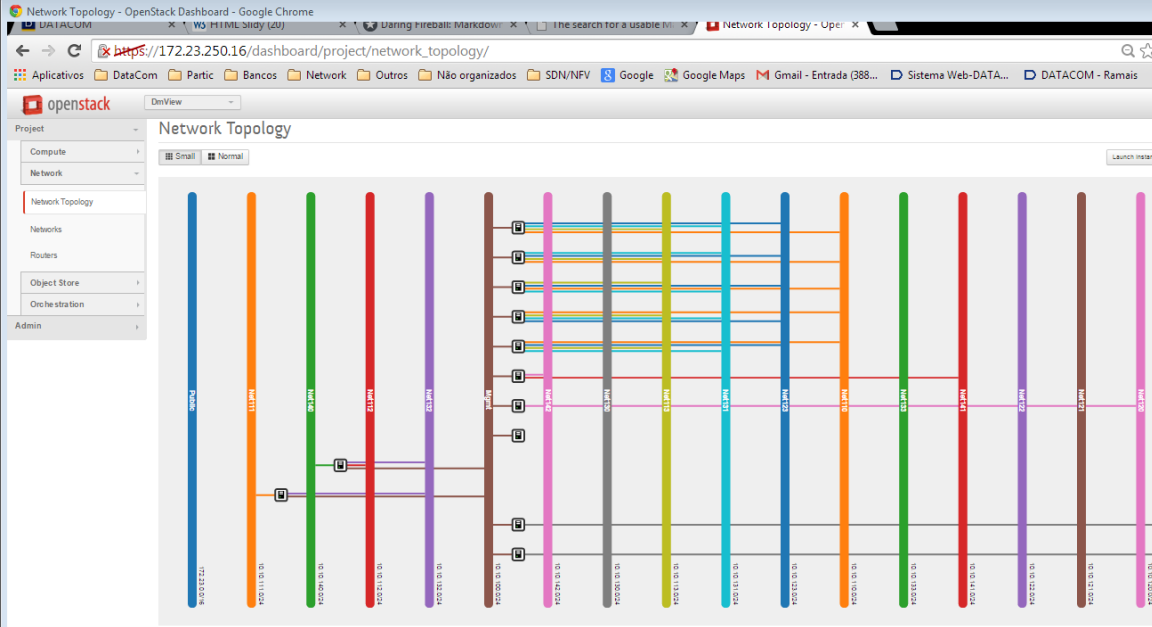
DmView development

- Any choice of topology
- DM4100 VMs
 - 1GB RAM, 1 vCPU, 1GB Disk
 - Up 7 Dataplane Interfaces + 1 Mgmt
- Mix of virtual and bare metal appliances



Use Case: DmView

Project	Host	Name	Image Name	IP Address	Size	Status	Task	Power State	Uptime	Actions
DmView	oscompue1.localdomain	DMA100-G	DMA100-13.6-final	Net131 10.10.131.10 Net123 10.10.123.9 Net110 10.10.110.45 Mgmt 10.10.100.118 172.23.250.118 Net113 10.10.113.11	m3.tiny-4100 (1GB RAM) 1 VCPU 1 0GB Disk	Active	None	Running	2 weeks, 2 days	Get Instance Stop
DmView	oscompue2.localdomain	DMA100-H	DMA100-13.6-final	Net131 10.10.131.11 Net123 10.10.123.10 Net110 10.10.110.46 Mgmt 10.10.100.119 172.23.250.119 Net113 10.10.113.12	m3.tiny-4100 (1GB RAM) 1 VCPU 1 0GB Disk	Active	None	Running	2 weeks, 2 days	Get Instance Stop
DmView	oscompue1.localdomain	DMA100-I	DMA100-13.6-final	Net131 10.10.131.9 Net123 10.10.123.7 Net110 10.10.110.45 Net111 10.10.111.8 Mgmt 10.10.100.118 172.23.250.108	m3.tiny-4100 (1GB RAM) 1 VCPU 1 0GB Disk	Active	None	Running	2 weeks, 2 days	Get Instance Stop
DmView	oscompue2.localdomain	DMA100-J	DMA100-13.6-final	Net113 10.10.113.10 Net131 10.10.131.9 Net123 10.10.123.8 Net110 10.10.110.47 Mgmt 10.10.100.117	m3.tiny-4100 (1GB RAM) 1 VCPU 1 0GB Disk	Active	None	Running	2 weeks, 2 days	Get Instance Stop
DmView	oscompue1.localdomain	DMA100-L	DMA100-13.6-final	Net131 10.10.131.9 Net123 10.10.123.8 Net110 10.10.110.44	m3.tiny-4100 (1GB RAM) 1 VCPU 1 0GB Disk	Active	None	Running	2 weeks, 2 days	Get Instance Stop



The 'Launch Instance' dialog is shown with the 'Networking' tab selected. It displays 'Selected Networks' and 'Available networks'.

Selected Networks:

- nic1: Mgmt (10.10.100.0/24)
- nic2: Net110 (10.10.110.0/24)
- nic3: Net130 (10.10.130.0/24)
- nic4: Net143 (10.10.143.0/24)

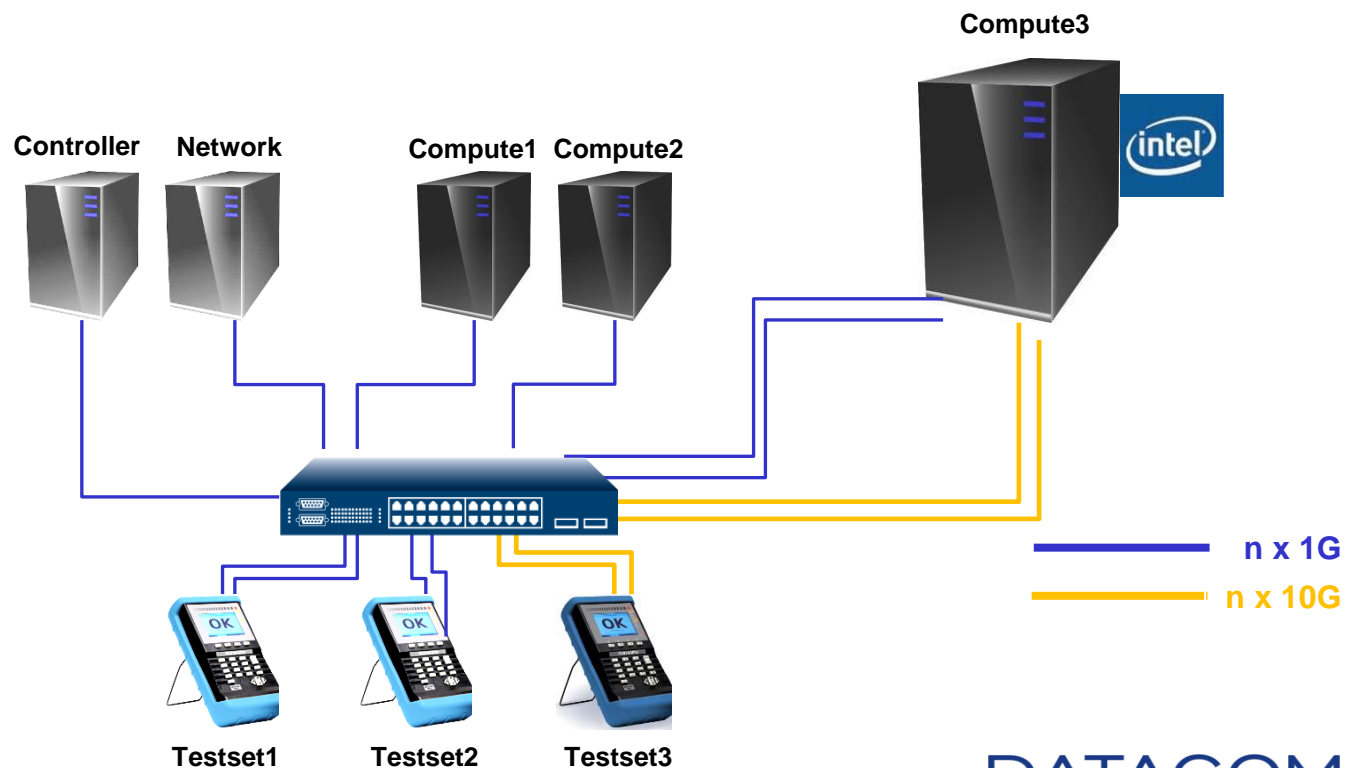
Available networks:

- Net111 (10.10.111.0/24)
- Net140 (10.10.140.0/24)

Instructions: Choose network from Available networks to Selected Networks by push button or drag and drop. you may change nic order by drag and drop as well.

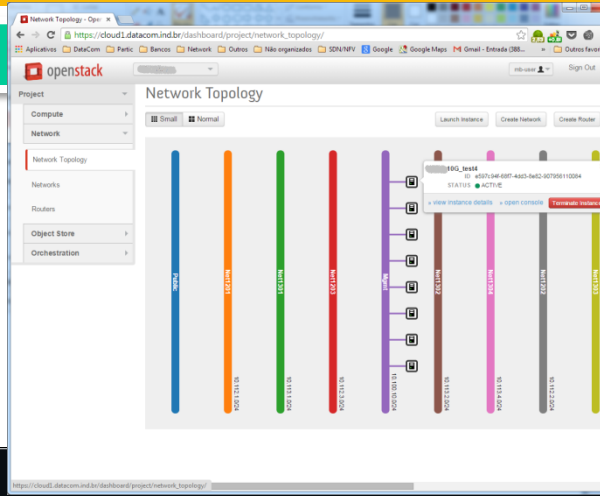
L3 Router

- Minimum vCPE has:
 - 2 dataplane interfaces through SR-IOV
 - 1 mgmt interface through Linux Bridge
 - 8 GB RAM
 - 5 vCPU



Project vCPE

- Single vCPE @ BER 10⁻⁹
 - up to 2,5Gbps FD => 7,44 Mpps
- Each Server @ BER 10⁻⁹
 - 8 x vCPE @ 1GB FD => 8 x 2,45 Mpps
- We can get more performance out of the server!



Launch Instance

Details * Access & Security * **Networking** * Post-Createion Advanced Options

Selected Networks

nic1 [NIC_1] [ACTIVE]

Available networks

- Net1201
- Net1301
- Net1203
- Net1302
- Net1304
- Net1202
- Net1303
- Net1103
- Net1102
- Net1204
- Net1101
- Net1104

Cancel Launch

Instances - OpenStack Dashboard

https://cloud1.datacom.ind.br/dashboard/project/instances/

Project: **openstack**

Instances

Filter [] Filter [] + Launch Instance Soft Reboot Instances Terminate Instances

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Uptime	Actions
10G_test4		10.100.10.58	m2.tiny-vimos-2x10G 8GB RAM 5 VCPU 5.0GB Disk	-	Active	nova	None	Running	1 week, 4 days	Create Snapshot More
10G_test3		10.100.10.57	m2.tiny-vimos-2x10G 8GB RAM 5 VCPU 5.0GB Disk	-	Shutoff	nova	None	Shutdown	1 week, 4 days	Start Instance More
fake		10.100.10.56	m2.tiny-vimos-6x10G 8GB RAM 5 VCPU 5.0GB Disk	-	Shutoff	nova	None	Shutdown	1 week, 4 days	Start Instance More
10G_test2		10.100.10.54	m2.tiny-vimos-10G 8GB RAM 5 VCPU 5.0GB Disk	-	Active	nova	None	Running	1 week, 6 days	Create Snapshot More
10G_test1		10.100.10.53	m2.tiny-vimos-10G 8GB RAM 5 VCPU 5.0GB Disk	-	Active	nova	None	Running	1 week, 6 days	Create Snapshot More
1G_test6		10.100.10.48 172.25.16.131	m2.tiny-vimos-10G 8GB RAM 5 VCPU 5.0GB Disk	-	Active	nova	None	Running	1 week, 6 days	Create Snapshot More
1G_test5		10.100.10.47	m2.tiny-vimos-1G 8GB RAM 5 VCPU 5.0GB Disk	-	Active	nova	None	Running	1 week, 6 days	Create Snapshot More
c1		10.100.10.20 172.25.16.132	m1.tiny 512MB RAM 1 VCPU 1.0GB Disk	-	Active	nova	None	Running	1 month, 2 weeks	Create Snapshot More

Displaying 8 items

Next Steps

- Evolve our testing environment concerning scalability and use cases tests
- OPNFV
- OPENSTACK => Juno Oct/2014; Kilo Apr/2015
- SDN + NFV
- PoCs and ideas are welcomed!

Thank you!

Marcelo Barcelos

R&D

+55 (51) 3933 3000

www.datacom.ind.br

barcelos@datacom.ind.br

DATAKOM

OPNFV

⁴ ETSI NFV ISG Architectural Framework: www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf

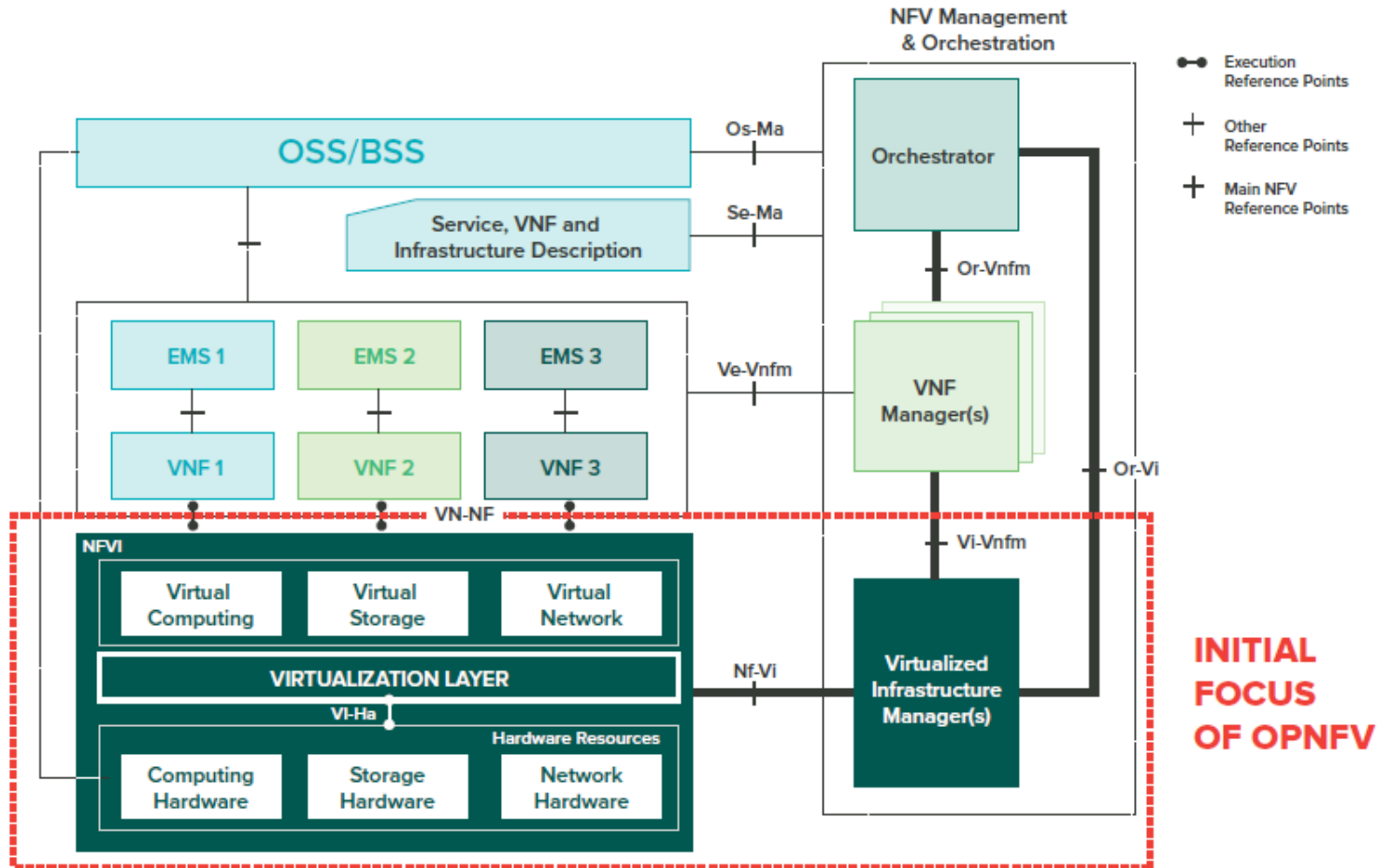


Figure 1: NFV Reference Architectural Framework

Figure from OPNFV White Paper

The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.

The SDN architecture is:

- **Directly programmable**
 - Podemos *programar* o nosso control plane, criando novas regras em L2, L3 e MPLS
- **Agile**
 - Como o control plane agora pode gerenciar diversos planos de encaminhamento, alterando as regras eu posso reconfigurar instantaneamente o comportamento da minha rede
- **Centrally managed**
 - Grande potencial para abstrações
- **Programmatically configured**
 - Podemos *programar* as ações de config, OAM, segurança, otimização
- **Open standards-based and vendor-neutral**
 - Modelo de negócio Open Source

