



Horizontal NFVの実践と課題

Isamu Monji (文字 勇)

Business Solutions Architect, Cisco Systems

October 15, 2020

自己紹介

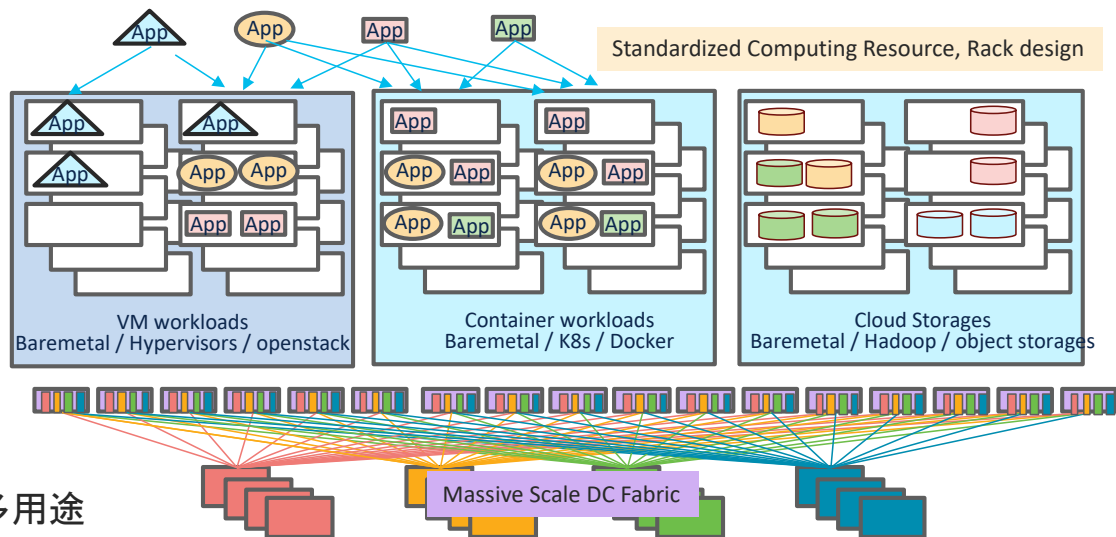
- 文字 勇 (もんじ いさむ)
- 某 通信事業者にて数年 Network Engineer職
 - 流通業界のNetwork、新規事業の立ち上げのようなものも
- 2001年～ Cisco Systems にて Systems Engineer, SE Manager, Solution Architect
 - 主に通信事業者様を担当するSales部門にて
 - Broadband Access、Telecom Infrastructure、Mobile Core、Small Cell、Stadium Entertainment、他
- 現在アーキテクト職として : 通信事業者様の新しいインフラ構築に携わる
 - 4G/5G Mobile Core, (v)RAN, Transport, DC, NFV, Automation,

Agenda

- “Horizontal NFV” と “Vertical NFV”
- Horizontal NFVの事例
- 大規模なNFVでの共通NFV基盤：メリット
- 直面した課題
- まとめ：今後の拡大へ向けて

Public Cloudは広く一般に普及

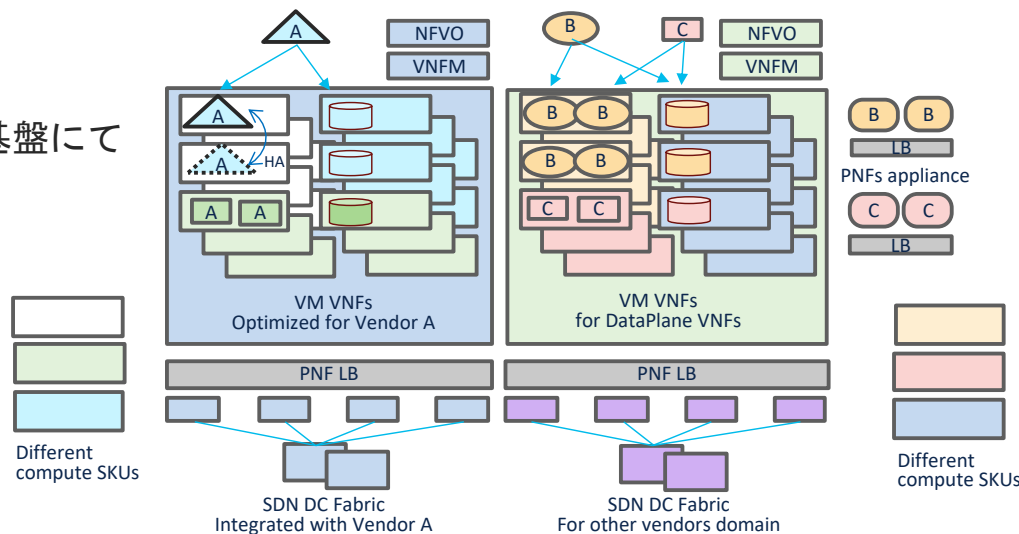
- VM, Containerなど必要な Resourceを自由に確保
- App/Workloadを載せるだけ
- Storageも必要な量だけ、
随時Scale Up
- ユーザからはインフラが
どうなっているか見えない
(見る必要が無い、意識させない)
- 開発環境、商用サービス、Hybrid、多用途



➡ 通信事業者のインフラを支えるNetwork Functionには

通信業界でもNFVは数年来導入が拡大

- “Virtual NFV”が少なくない
- VNF vendorとしての性能保証はvalidate済の基盤にて (HW, HyperVisor, Storage, etc)
- 多数のVMから成るVNF、Networking Design、作り込んだ冗長化・自動化・運用、大規模導入には各社のVNFMにて
- System Integrationとあわせて Compute, Storage, DC Fabricも提供

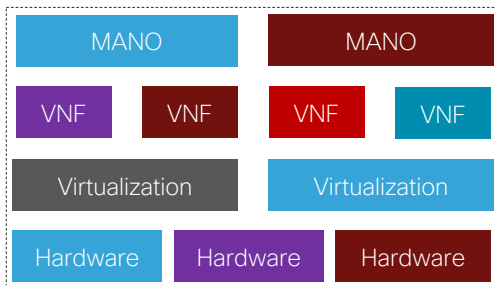


- 極端な例では VNF Vendor毎に個別のVirtual NFV Stackが並ぶ
- Computeの例を見てもSKUが多種に分かれ、調達・構築・Lifecycle管理も個別に行われる
- 部分的にNFVの導入は進むが、完全仮想化には進みにくい(全て仮想化する方針でも無い)

“Horizontal NFV” は実現したい理想のモデル

1

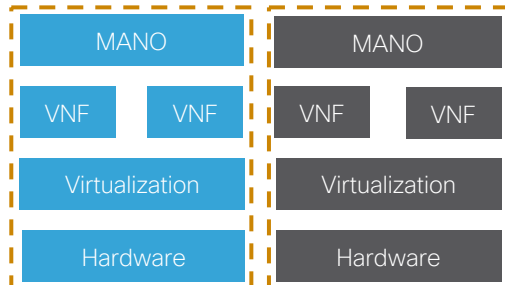
Do it Yourself (DIY)



- 標準化により各種Componentの疎結合は実現できる
- オープンソース、Vendor特性・固有、その他の組み合わせによるフルカスタム・モデル
- 新しい技術、Solutionを柔軟に取り込めるアーキテクチャ
- 複雑化は否めず、大規模展開や運用には課題

2

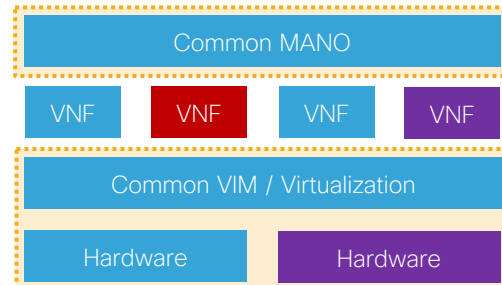
Vertical Solution Stack



- 主要なVNFコンポーネントに関連しHWとSWの性能保証の観点も含む
- 仮想化とCOTS HWなどメリットは享受しつつ、アプライアンスモデルにも近い
- 各VNF 機能、ドメインごとにStackが作られることも少なくない

3

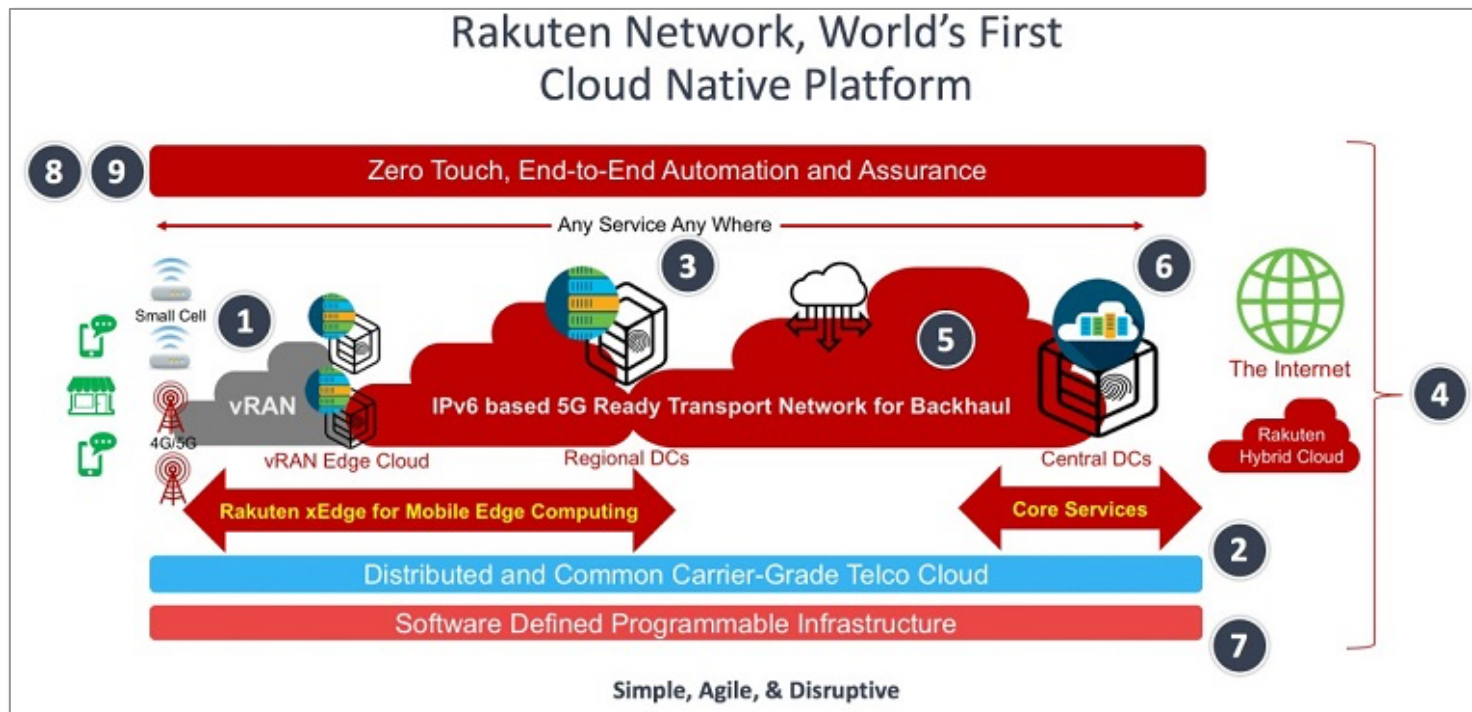
Common Horizontal Platform



- NFV基盤 HW, 仮想化基盤、MANOレイヤを共通化
- VNFは各Vendor, 各機能を組み合わせ、共通基盤上で稼働
- 適用範囲の水平拡大に期待
- VNF Vendorは性能評価や動作保証が難しい

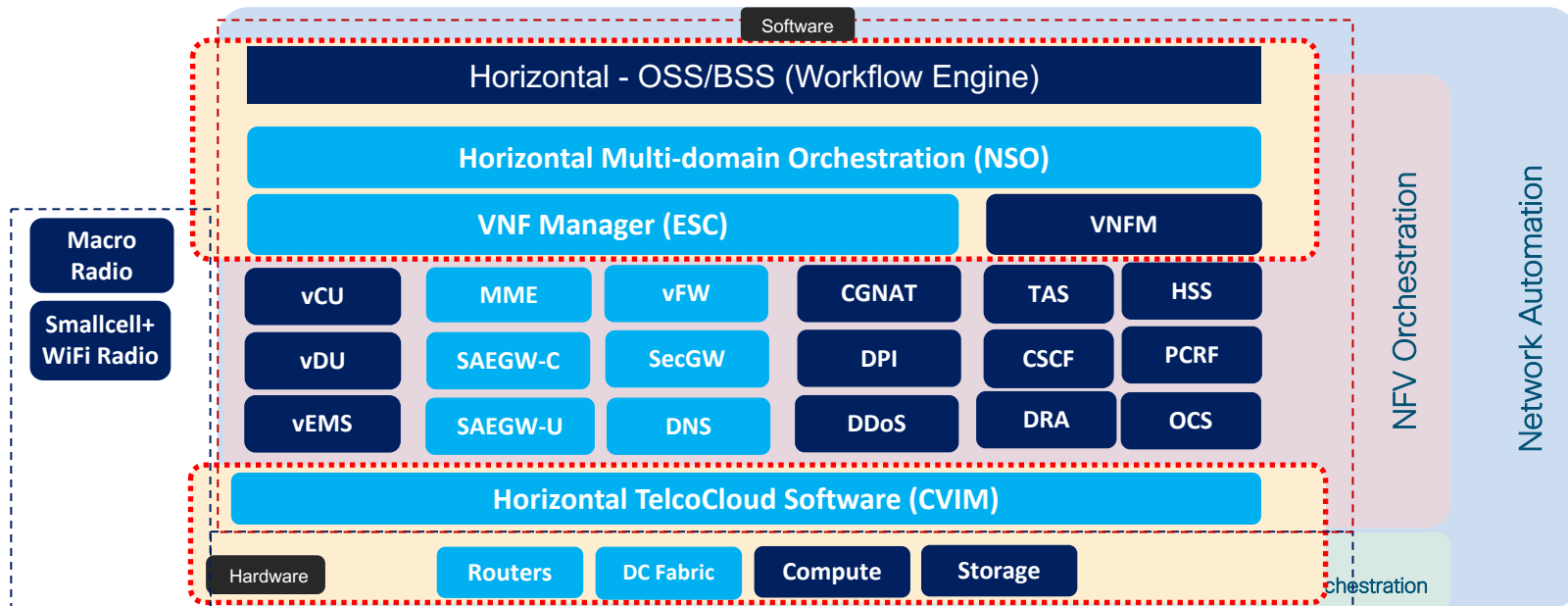
Horizontal NFVの事例：楽天モバイル(株)様

完全仮想化 モバイルインフラの実現は”Horizontal NFV”が非常に重要



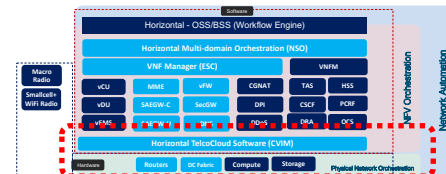
Horizontal NFVの事例 : アーキテクチャと主なVNF

共通 NFVO Automation / Orchestration : Common OSS, Cisco NSO, ESC, other VNFM



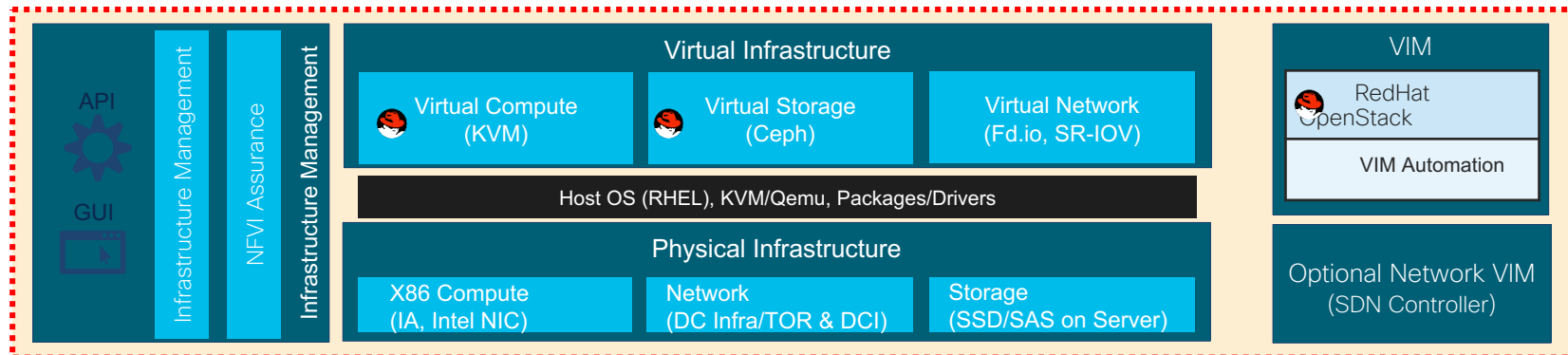
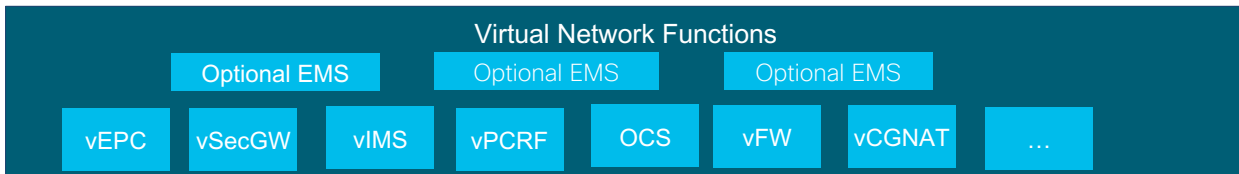
共通 NFVI 基盤 : Cisco VIM + Cisco ACI + x86 Servers

Cisco VIM : NFV向け Openstack環境の自動化

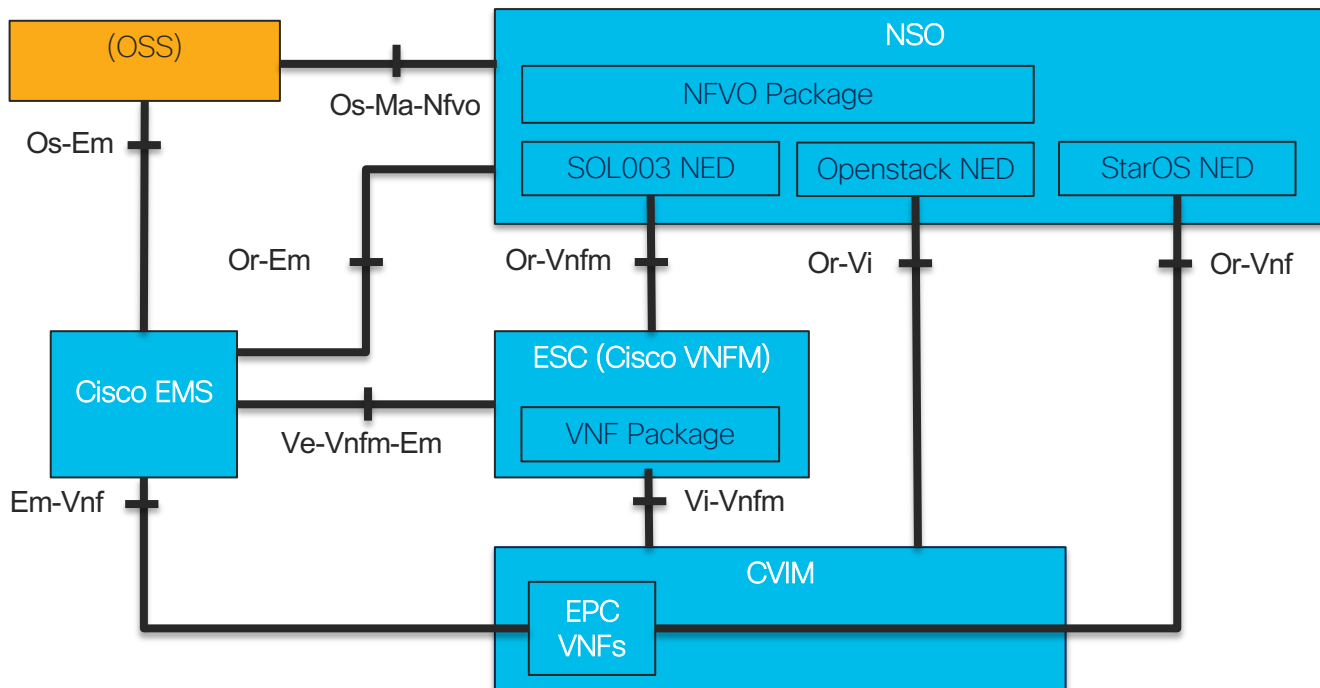
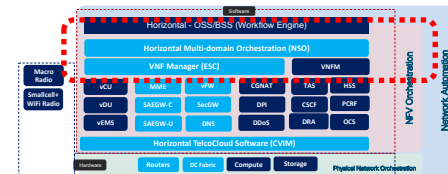


OSS, BSS, Portal, CRM

NFVO, Resource Orchestration & VNF Service Orchestration



VNF Automation Architecture : Cisco vEPCの例



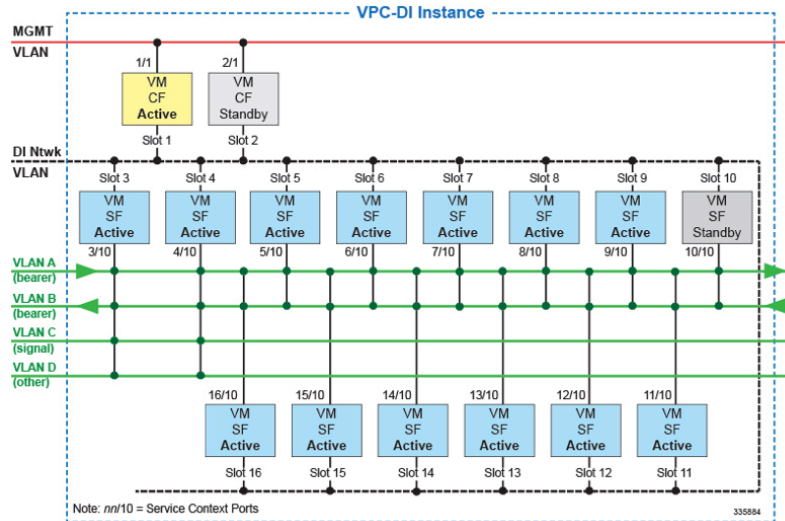
- 常にOSSからVNF立ち上げ
- NFVO (NSO)からCompute Resource Assign、VNFM (ESC)からVM立ち上げ、NSOからDay1 Config投入
- 異なるVNF vendorにもNFVO, VNFMを共通化
- 異なるVendorのVNFMも許容する: 複雑な構成のVNF, 開発期間の観点など

共通NFV 基盤 : 成功のポイント (設計面)

- VNF Workloadによらない Compute Nodeの標準化 : 少ないHW SKU
 - セッション・コントロール系 (CPU intensive), ユーザデータ系 (Data Plane Performance)、など
コアDCのVNFは基本的に共通SKU : SR-IOV NIC, CPU, Memory, Local Storage,
- PODデザイン、Networkingデザインの(できる限り)モデル化、シンプル化
 - Controller, Compute, Storage のラックデザイン標準化、ToR/Spine 物理接続も
 - Logical NW : Data Plane (Internal, External), L2 segment, L3 gateway, VRF Segregation
- デザインの標準化はAutomationを救う

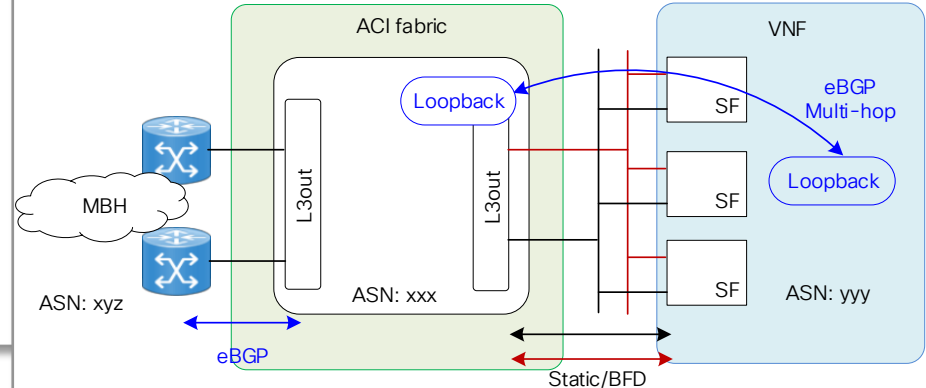
Cisco vEPC の場合 : Control Planeの例

vPC-DI VMs / Network



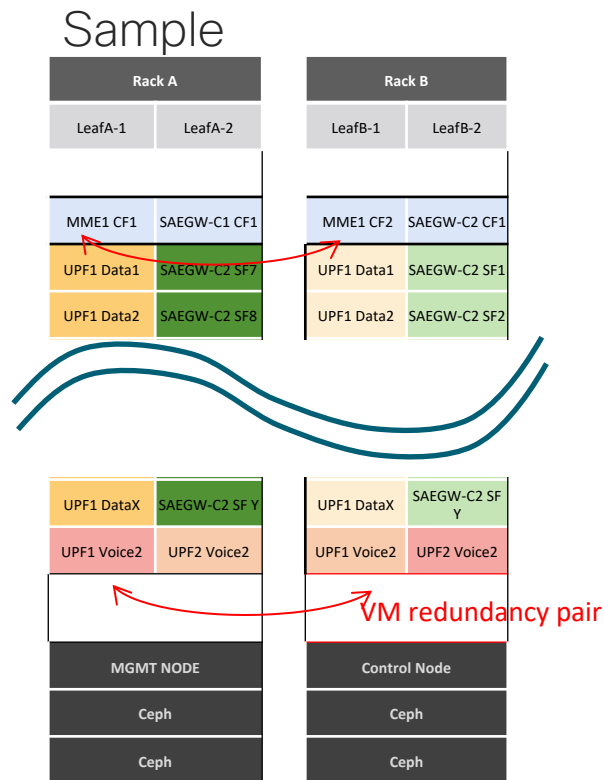
- vMME vPC-DI VNF instance
- 2 x CF, N x SF
- L3 Gateway, BGP advertise

eBGP peering with vEPC VNF



DynamicなCompute ResourceのAllocationは理想、But...

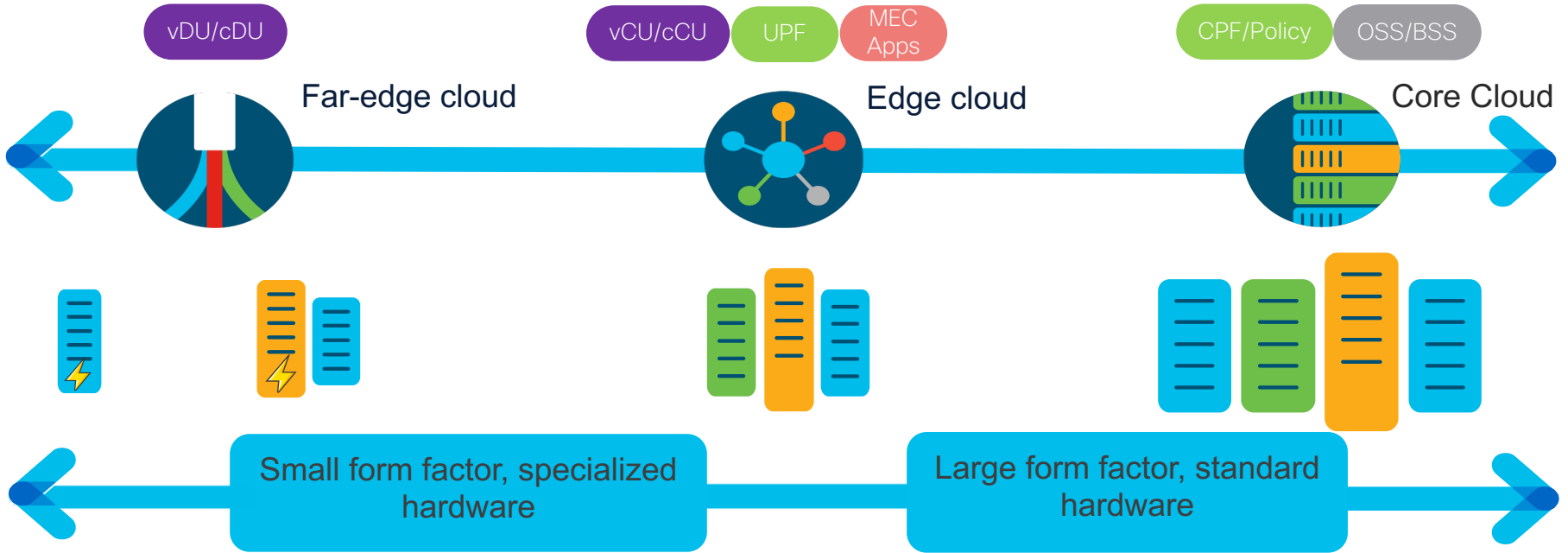
- VNFに要求される性能 : 例: UPF N Gbps (UL + DL)
 - VM placement, CPU pinning
- VNF あるいは VMの単位でのRedundancy
 - Anti-affinity も
 - 物理Rackを跨いだ耐障害性
- 初期構築時のPOD Sizing
 - 必要セッション数、Traffic Capacity
 - DC内・DC間冗長化



Horizontal NFV により得るメリット

- 共通基盤の上に新しいVNFを載せられる柔軟性
 - デザインの標準化、Networkingのモデル化、通信Flow、
 - 新サービスを迅速に提供できる俊敏性
- 機能ドメインをまたがる、共通のリソース管理、HW保守・増強
 - 複数のVertical Stack -> Horizontal Stackの運用性
 - Telco NFVのクラウド運用化に近づく
- 物理的なロケーションの拡大・水平展開
 - Distributed DC、Edge Computing
 - 規模の拡大

Horizontal NFVはDistributed DCを実現する



- Abstracted infrastructure layer
- Automated lifecycle management
- Centralized, end-to-end monitoring

- End-to-end service orchestration
- Network functions deployed anywhere

Horizontal NFV 直面した課題

- 各VNF vendor毎に異なる要件：どのように吸収するか
 - VM Flavor, Dimensioning
 - PerformanceとCPU pinning / NUMA aware、H/T
 - L2/L3 Networking
 - VM level Redundancy, VNF redundancy design
- リソースは有限
 - 理想は Flavorに基づいてVNFをDeployしていくこと
 - Under Dimensioning / Over Dimensioning
- 共通基盤でのQualification, Validation
 - 各VNF vendorのUnit Test onto NFV基盤
 - 総合試験環境：Multivendor Integration

リソースと初期 Dimensioning

- DCサイズ、POD サイズ、物理リソース
- Target Session、Traffic Capacity
- VM Flavor : 明確
 - vCPU, vMEM, IOPS, Disk, H/T
- 必要なトータルリソース : Dimensioning
 - 各VNF vendorごと
 - 共通POD or 分離

POD A

		Session M					
	Node	VNF	Compute	vCPU	vMem	vDisk	IOPS
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							

Sample

	POD A	Compute	vCPU	vMem	vDisk	IOPS
TOTAL :						

まとめ：今後の拡大へ向けて

- Horizontal NFV 共通基盤は、VNFデザインの標準化・シンプル化は大事
 - 来るコンテナ環境にも重要
 - Microservice, Service Mesh と、Data Plane性能 (SR-IOV ?)
 - Networking モデル化
 - SKUの最小化は大きな価値あり
- 通信インフラに求められる伝送性能、スケールと信頼性を実現しつつ、仮想化による柔軟性・俊敏性は両立できる
- アーキテクチャの標準化・共通基盤化により、どんなサービスをどこへでも展開できる、ようになった!?

