JANOG40@福島
SDNのこれまでとこれからと

VMware K.K.
井上一清
2017年7月27日

免責事項

- 本資料に掲載されている内容は技術情報の提供を目的としたものとなります。
  VMware の製品/サービスにこれらの機能を搭載することを約束するものではありません。

- この資料に記載されている技術内容はVMware製品とは関わりのないものもあります。

- 本セッションでの発言はVMware社としてのものではなく、個人の発言となります。

SDNのこれまで

- 2010年頃、数多のSDNベンダーが登場

2007年設立。2012年にVMwareが買収

2010年設立。2014年にオープンソース化

2010年設立

2010年設立。現在は物理Fabricに注力

2011年設立。2016年にVMwareがAssetを買収

2012年設立。2015年にNokiaが買収

2012年頃登場

2013年設立

2013年設立

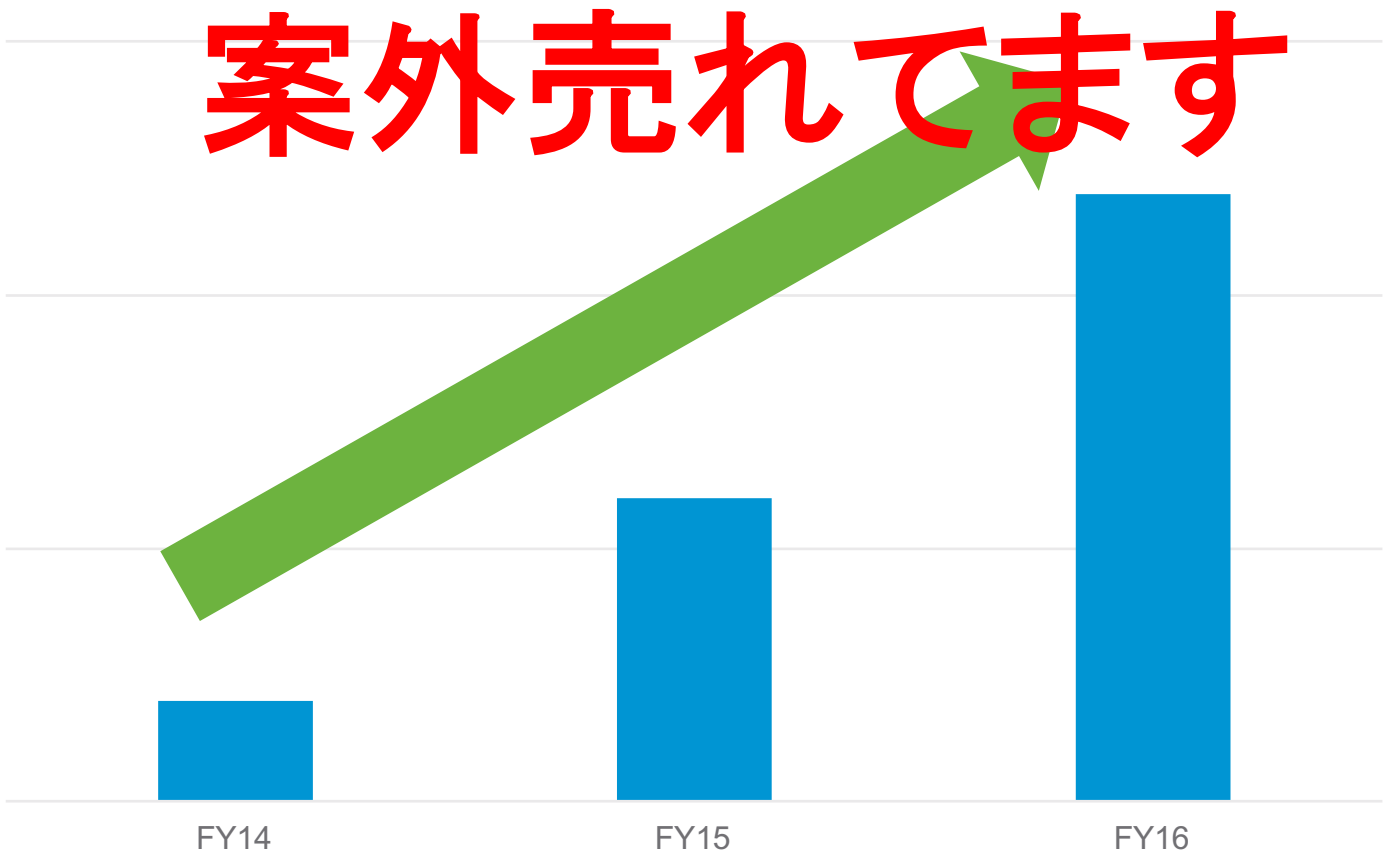
SDNを語っていた日々

- 言いたい放題言っていた201x年
- 色々なものを検証して、結局断った201x年

SDNは単なるバズワードだったのか？
バブルは弾けたのか？

SDNって売れてるの？

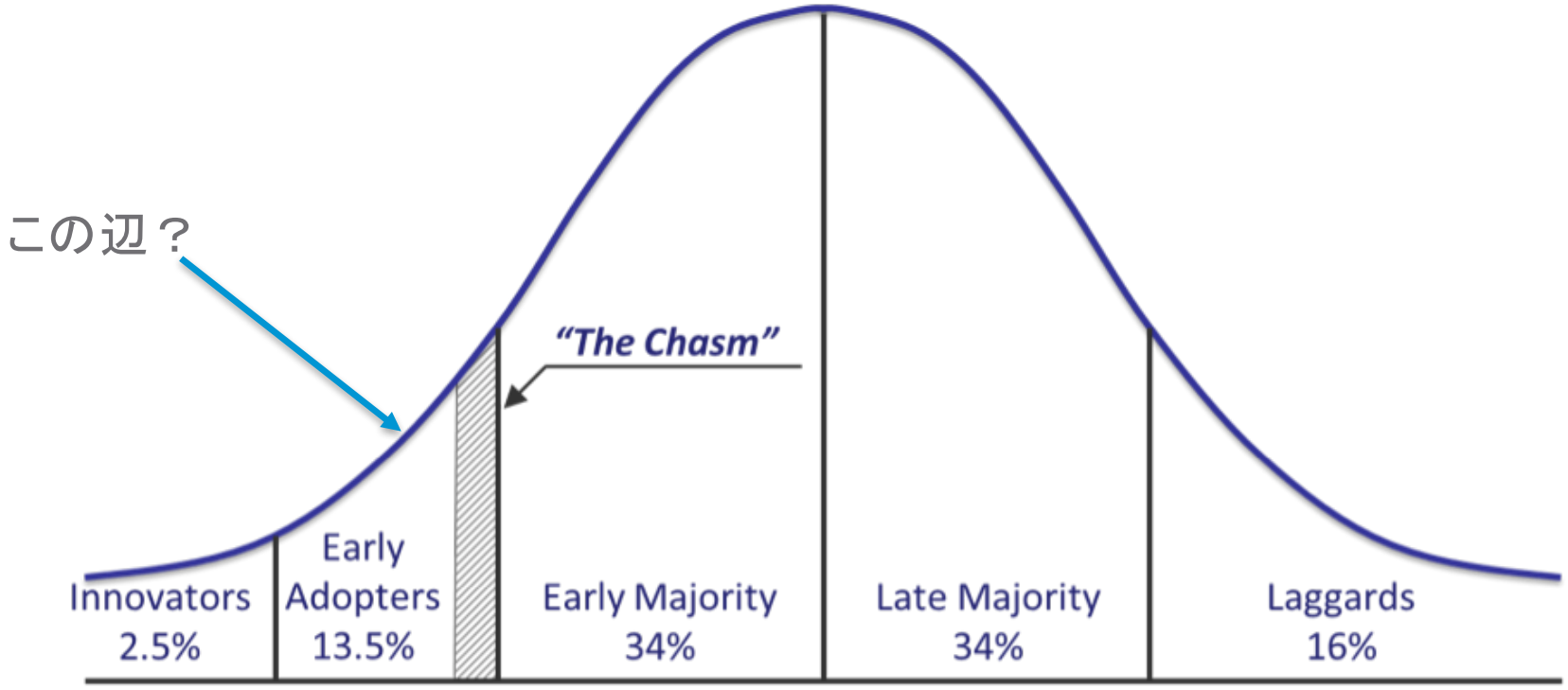
ネットワーク仮想化製品の利用社数



日本の方が伸び率はさらに上

キャズムは越えたの？

- まだまだだと思います。。



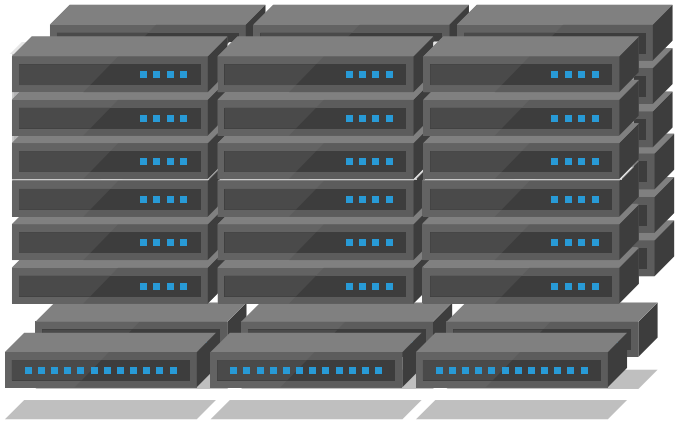
いまひとつな理由

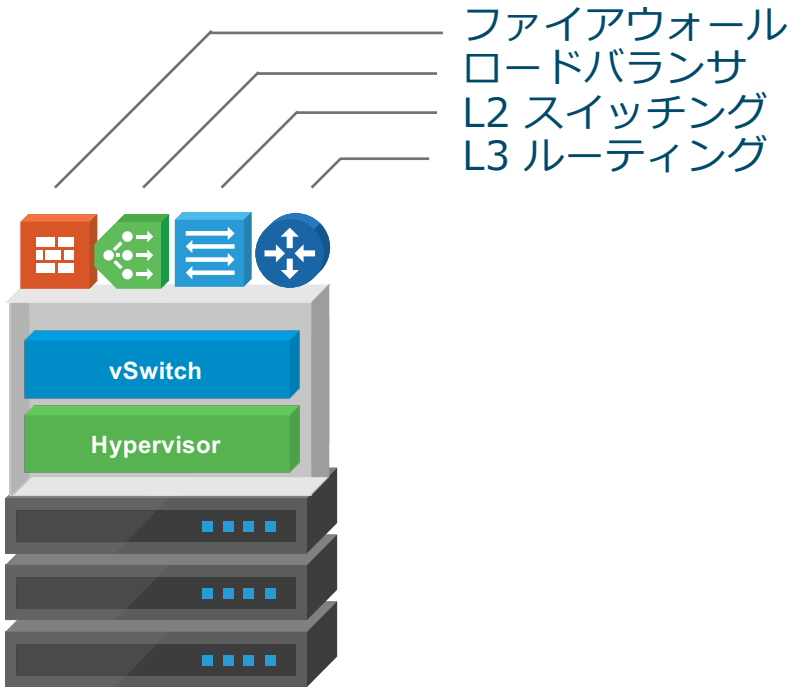
- 訴求が足りないから？
- 必要性があまりないから？
- 不安があるから？
- 良くわからないから？
- 面倒くさいから？
- 今のままでも別に良いから？

改めて、ネットワーク仮想化を見直してみよう

ネットワークの仮想化

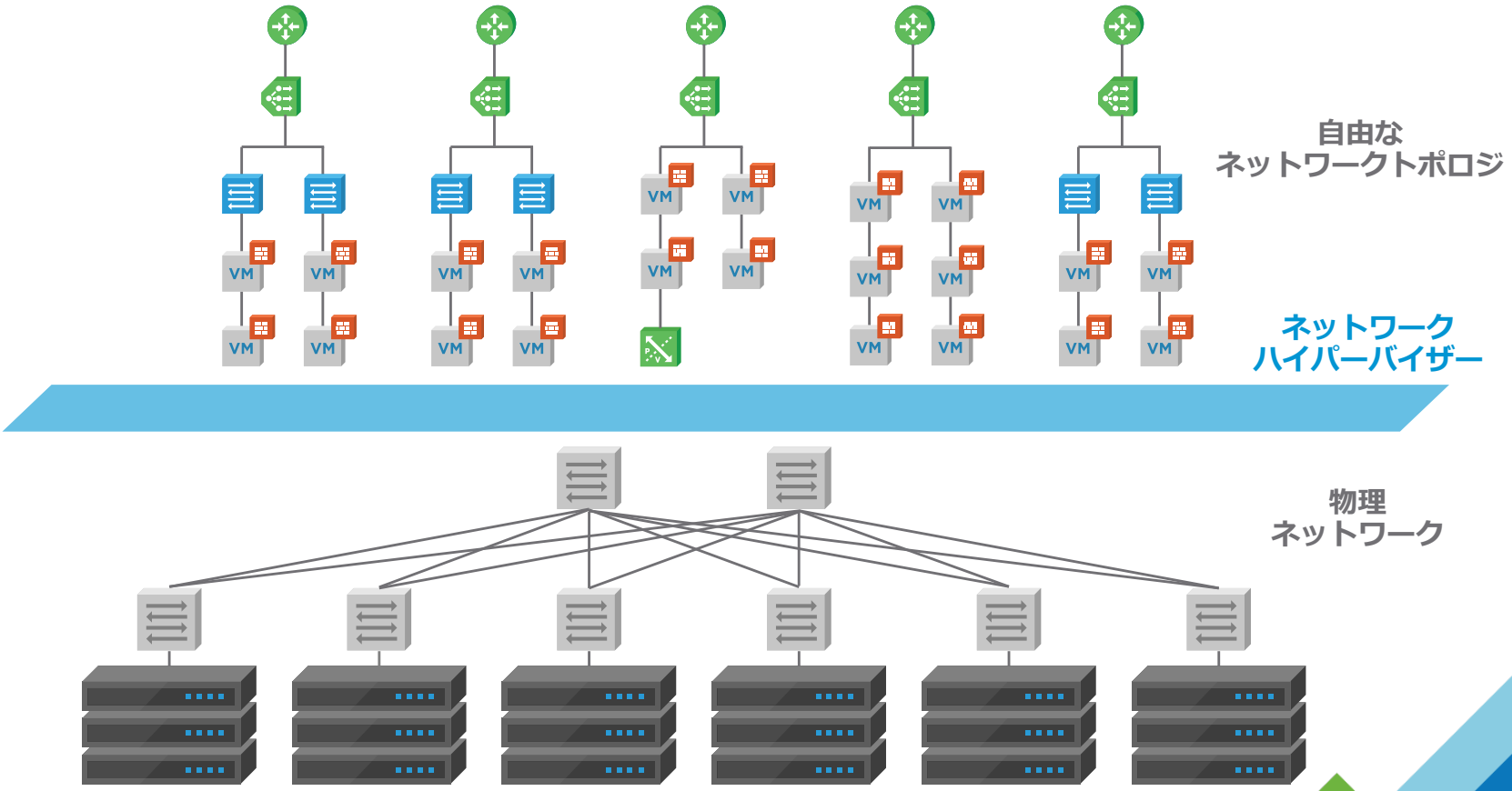
- 仮想化されたx86 サーバー上で様々なネットワーク機能を提供





ネットワークの仮想化

- Logical Switching
- Logical Routing
- Load Balancing
- Physical to Virtual
- Firewalling & Security



ネットワークの仮想化で使われている技術

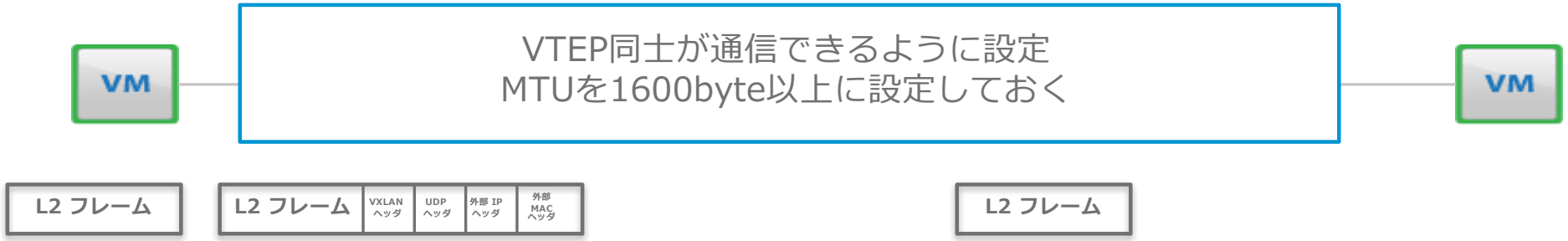
- L2： Overlay：VxLAN, NVGRE, Geneve, STT
- L3： 分散論理ルーティング
- L4： 分散FW, 分散LB
- Other： L7LB, L2VPN, SiteVPN, ClientVPN, IPS, WAF,

場所、環境に依存しないシームレスなネットワーク
East-Westトラフィックの合理的、最適な処理
Cross Cloudな設計

L2

L2 Packet Walk (VxLAN)

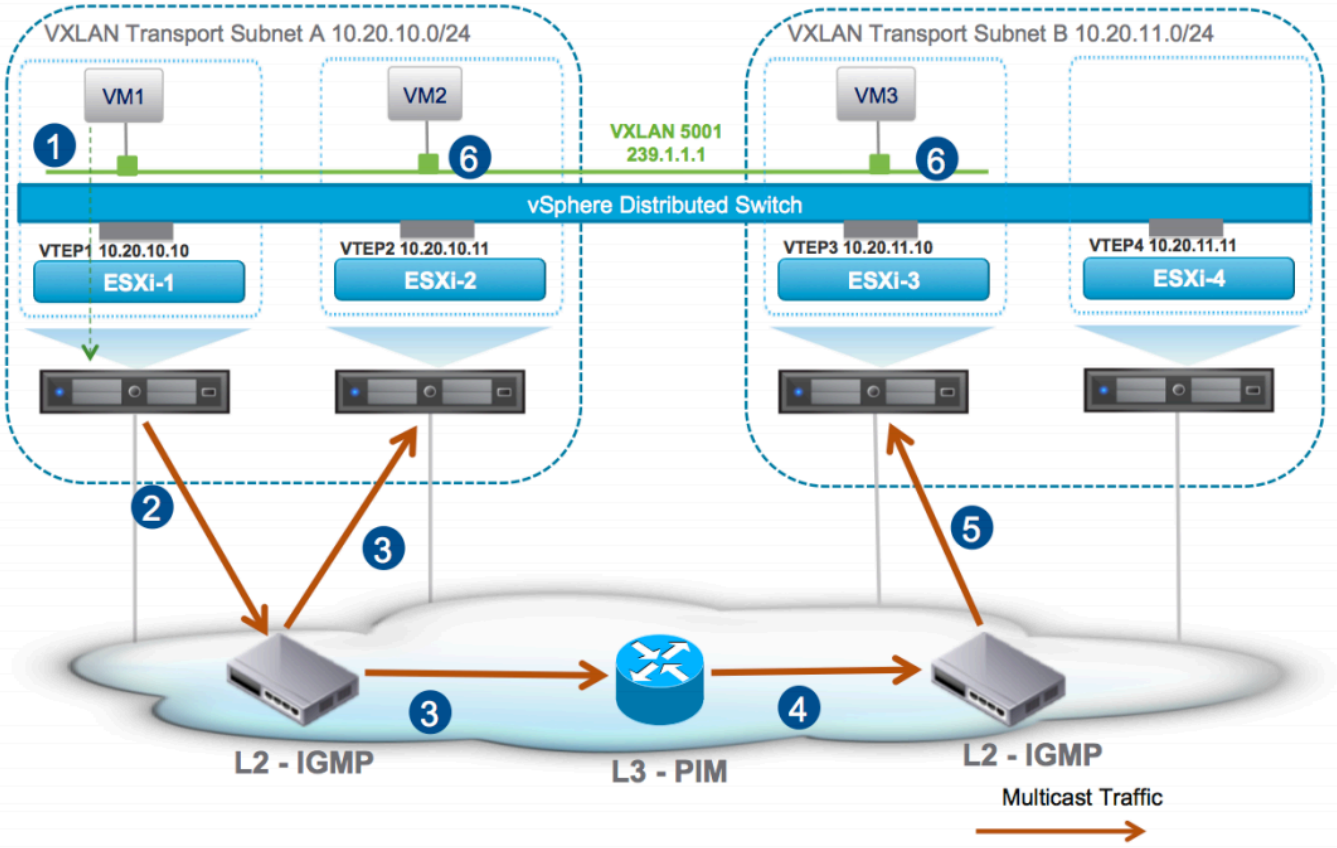
- VxLANは業界標準のIPオーバーレイ技術:
  IPネットワーク上でレイヤ2トラフィックをトンネル



1. 仮想マシンが標準の L2 フレームを送信
2. 送信元のハイパーバイザー（VTEP）が VXLAN、UDP、IP ヘッダを付加
3. 物理ネットワークが標準の IP フレームとしてフレームを転送
4. 送信先のハイパーバイザー（VTEP）がヘッダのカプセル化を解除
5. 元の L2 フレームを仮想マシンに送信

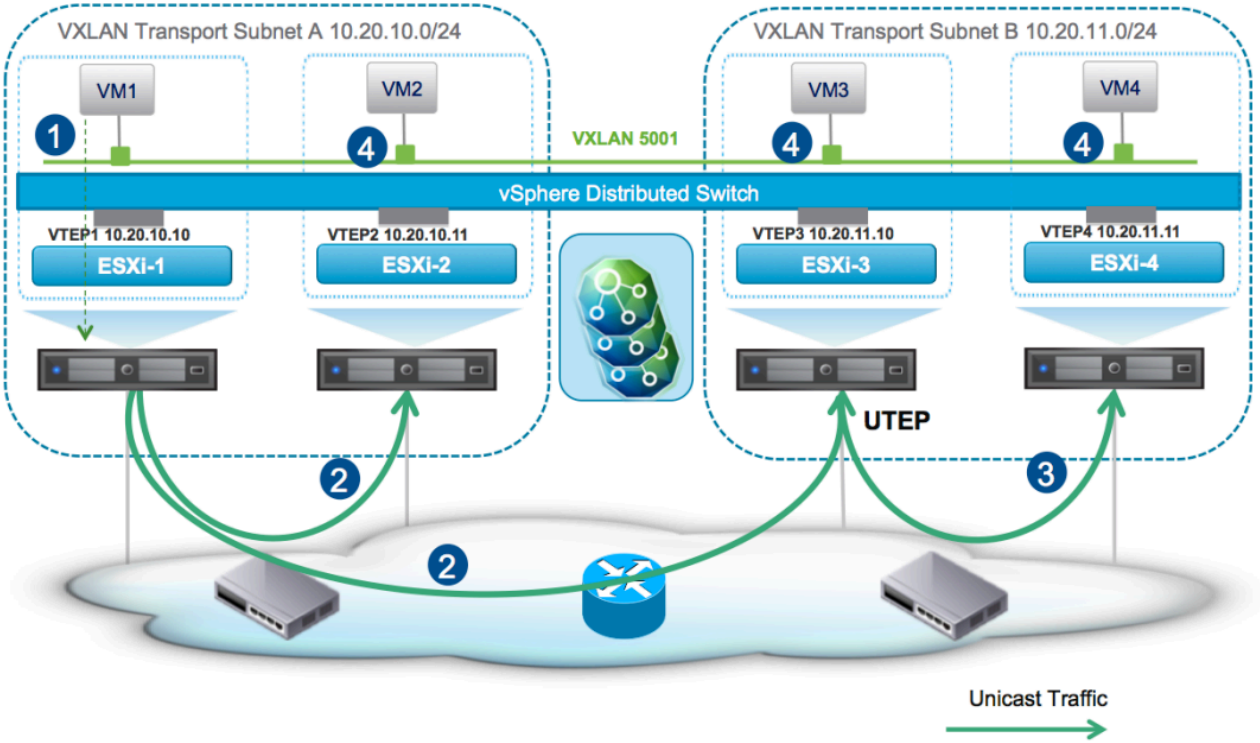
BUM処理（Multicast Mode）

- VTEP側でのReplication処理は必要ないが、物理ネットワーク側でMulticast Routingが必要
  RFC準拠なVxLAN方式だが採用実績ほぼゼロ



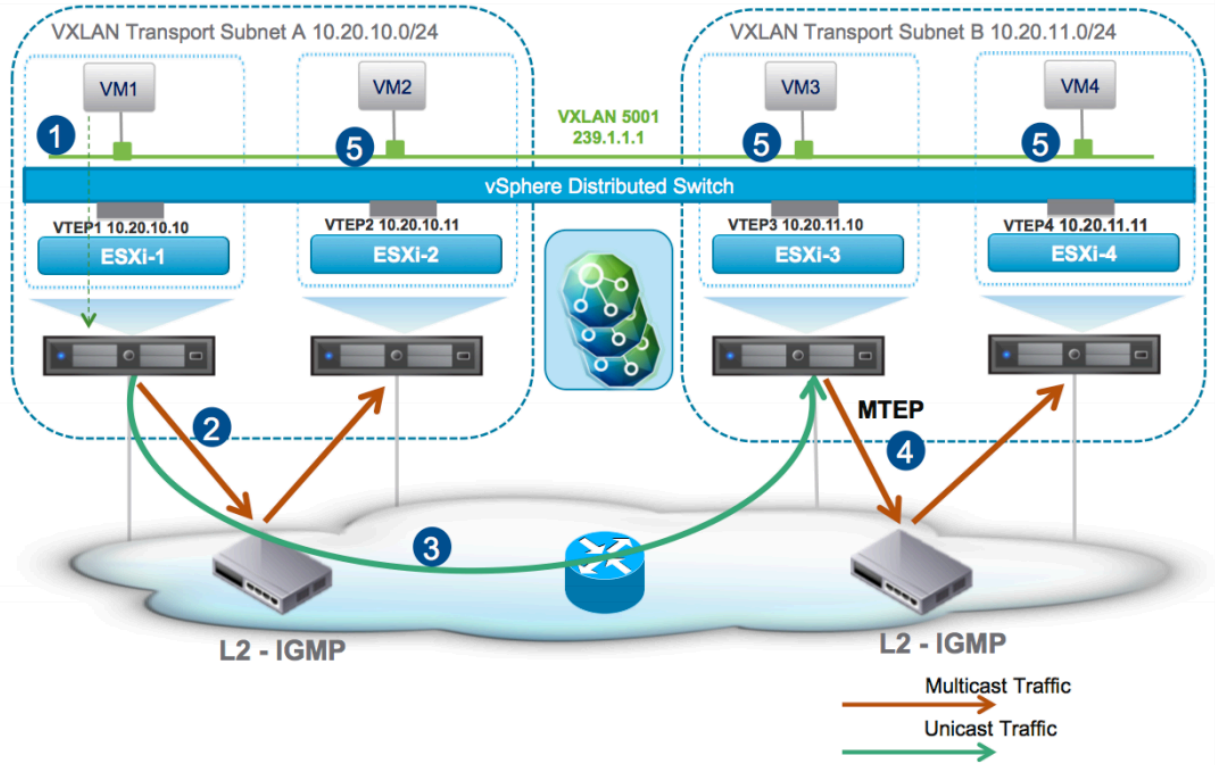
BUM処理（Unicast Mode）

- 物理ネットワーク側は特段何もする必要なし
  送信元VTEPと代表のUTEPでReplication
  最も採用されているパターン

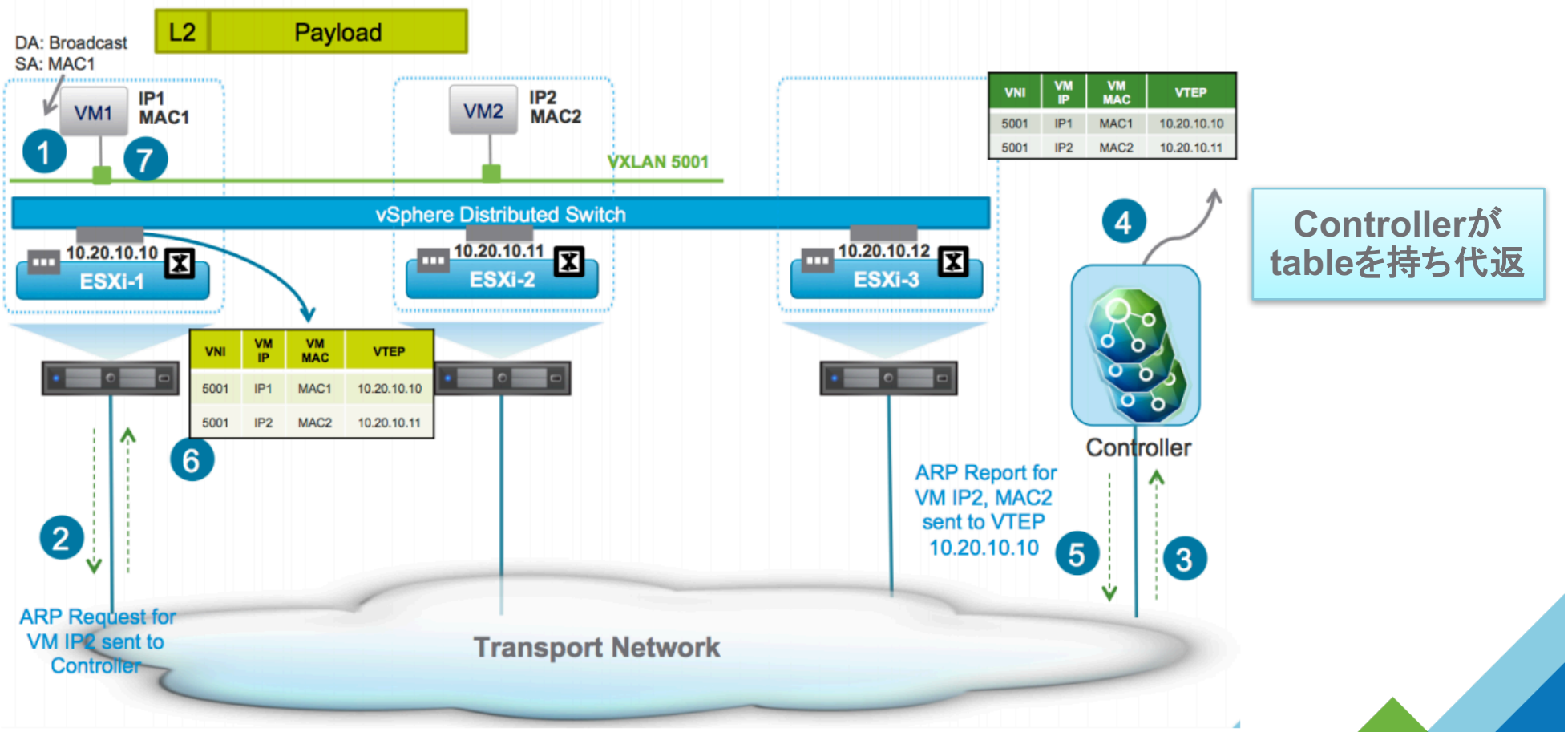


BUM処理（Hybrid Mode）

- 同一セグメントのVTEP間はMulticast、別セグメントのVTEPにはReplication
  大規模ネットワークで使われるパターン

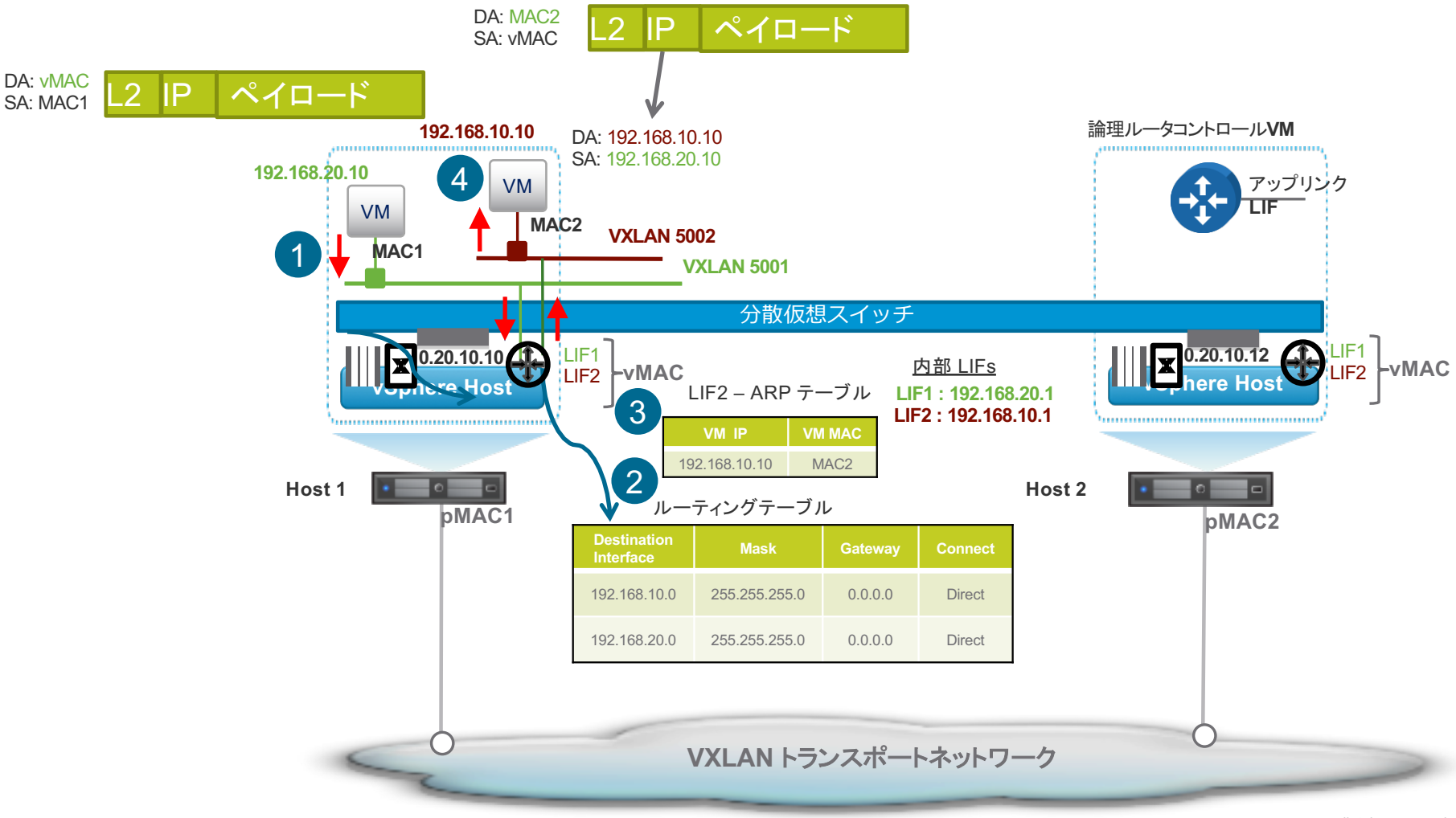


ARP解決 ～ARPを蔑ろにする者はARPに泣く～

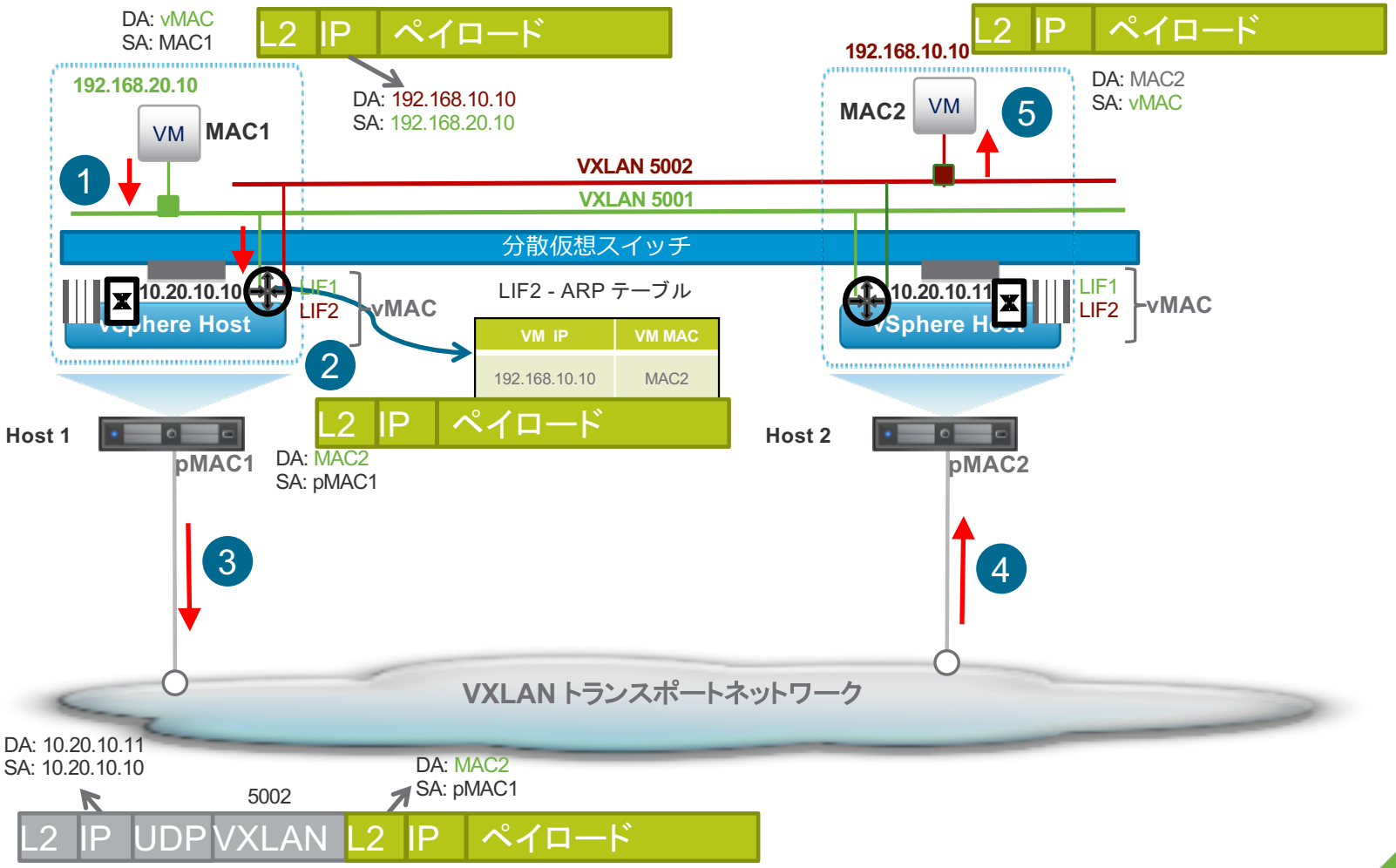


L3

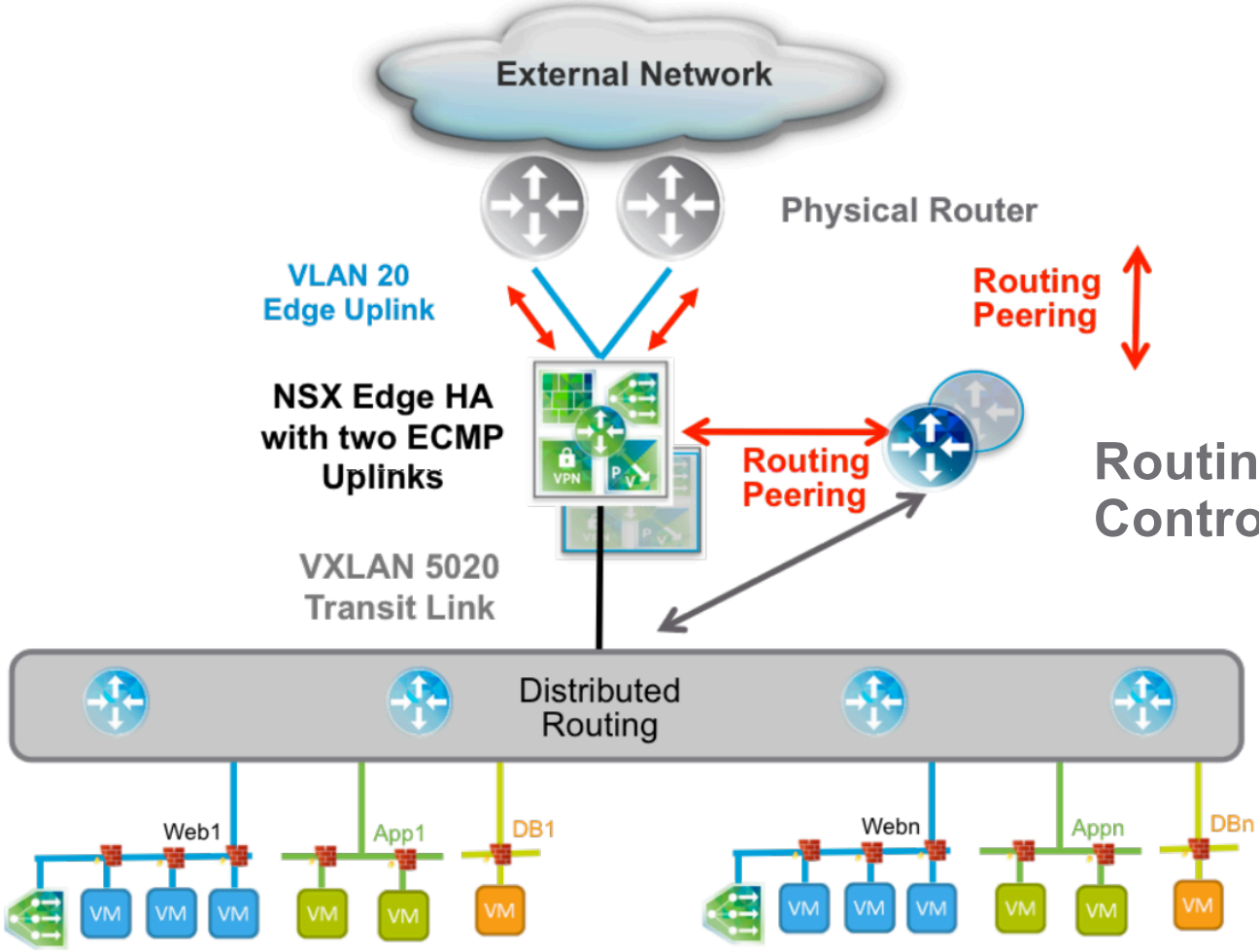
分散ルーティングのトラフィックフロー（同一ホスト）



分散ルーティングのトラフィックフロー（別ホスト）



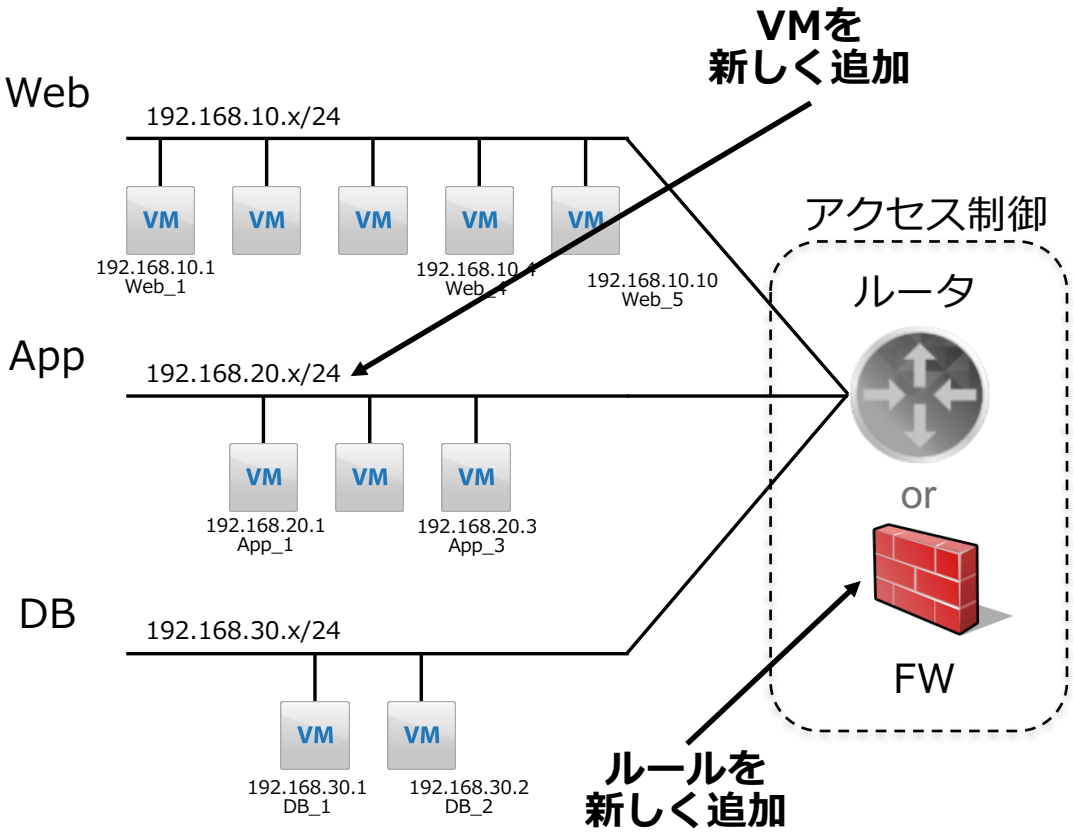
分散論理ルータとの接続



Routing情報のやり取りは
Control用のVMで実施

Firewall

通常のFWの場合



形式的、固定的なFW設定
- VMのIPが決まらないとFWの設定ができない

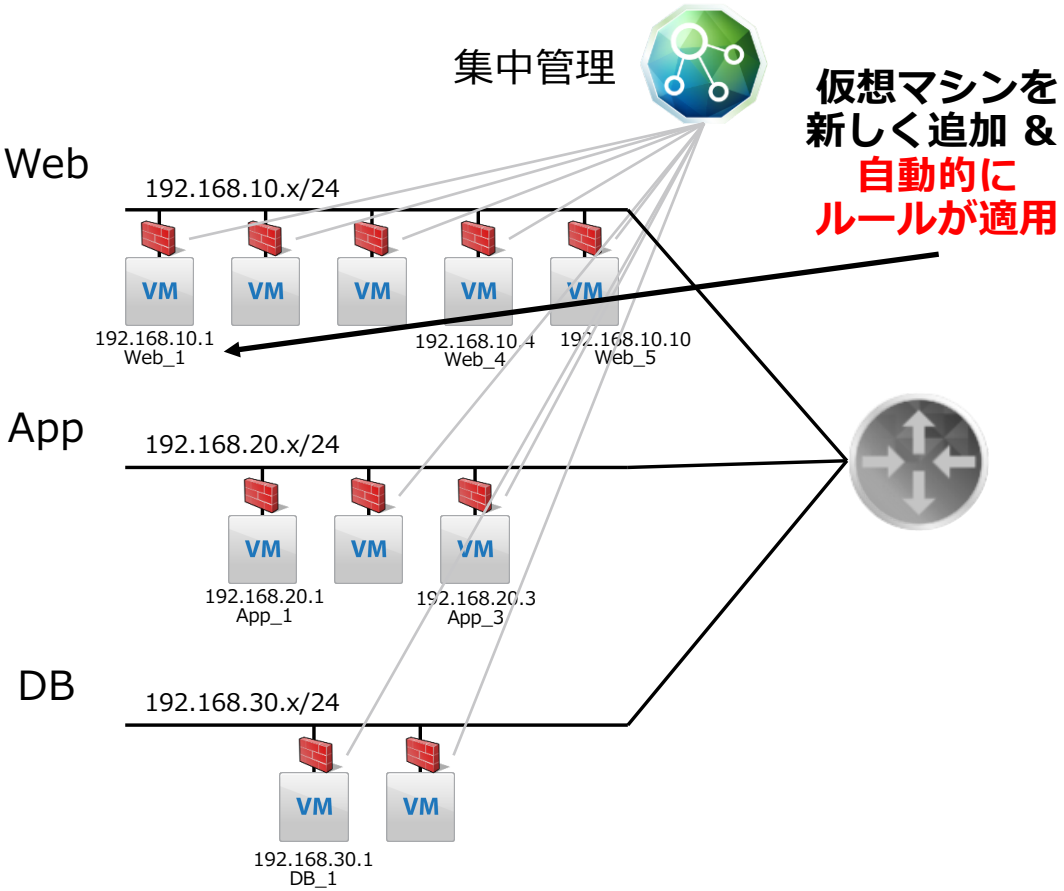
オブジェク的な設定が困難
- Web to App ：許可
- Web to Web ：拒否 ←難しい

ルールは、IPアドレスベース
- 192.168.10.1-4 to 192.168.20.1-3：サービス：許可

FWルールの崩壊
- 膨大なルール数の管理負荷増大
  - ルールは削除できず、増えるばかり

1台でも侵食されたら全滅の可能性も

分散FWの場合



送信元	送信先	サービス	ポリシー
Web＊	App＊	TCP	許可
App＊	DB＊	TCP3306	許可
Any	Any		Block

ルール適用箇所の分散化
- 仮想マシン単位でのルール

ポリシーとルールの相関性の実現
- ポリシーもルールもオブジェクトベース
  - Web層 to App層 ：サービス：許可

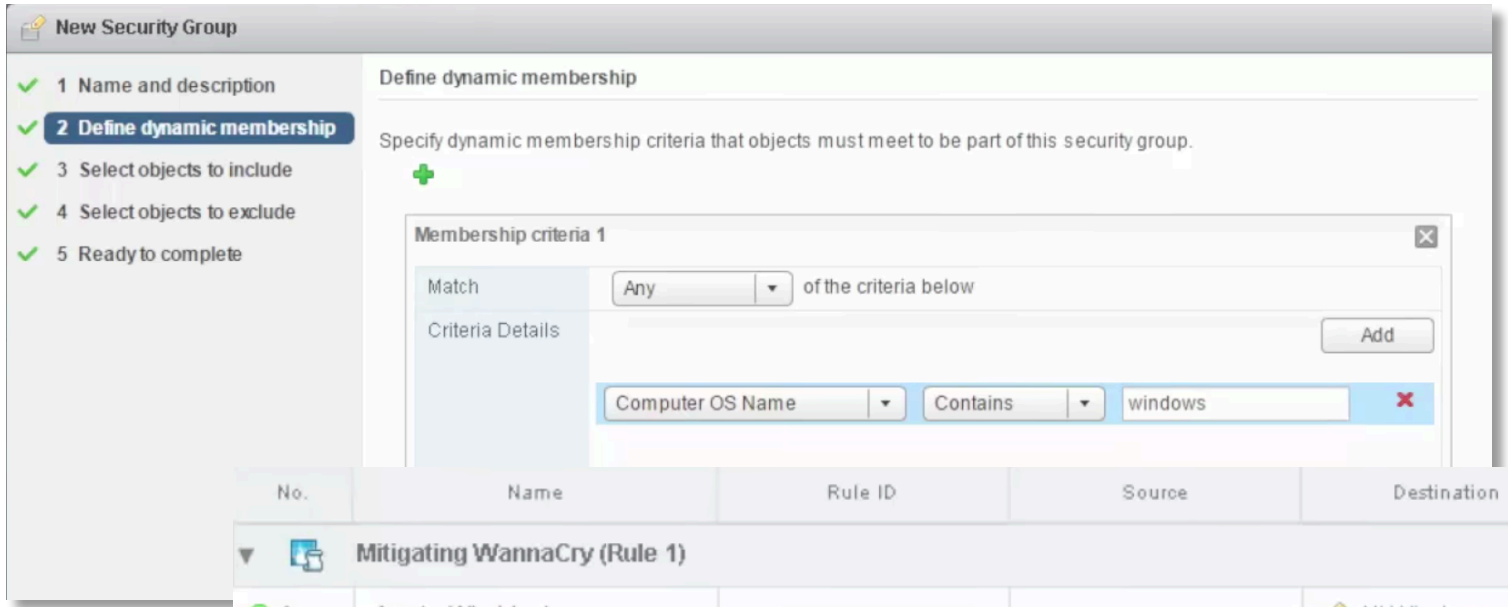
ルール適用の自動化
- 管理するのはポリシー
  - 追加も削除も自動化

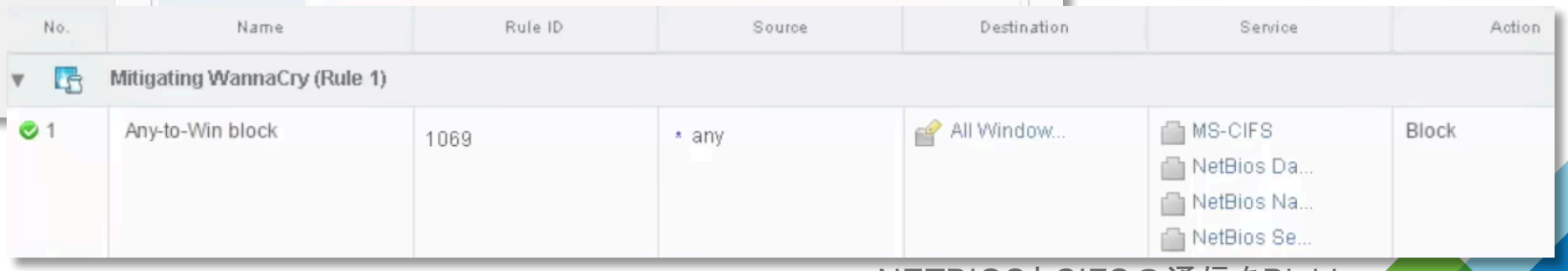
・クラスター
・VM名
・VM タグ
・OS
etc

不必要な通信を容易にblockできる

（例）How to mitigate WannaCry

Security organizations recommend blocking the following ports to the vulnerable systems
- 137 and 138 UDP (NETBIOS Name Service and NETBIOS Datagram Service resp.)
- 139 TCP (NETBIOS Session Service)
- 445 TCP (Microsoft CIFS)





WindowsのGroupを作成し、

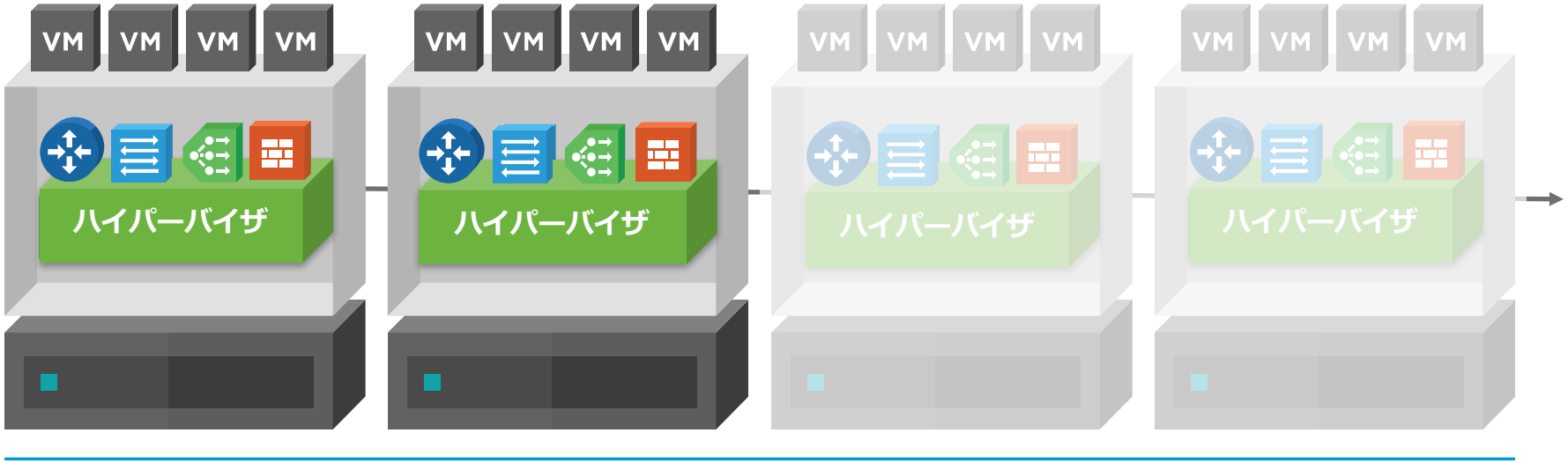
NETBIOSとCIFSの通信をBlobk

（例）ウィルスにやられたVMを自動隔離

Tag = ‘ANTI_VIRUS.VirusFound’
セキュリティ グループ = 隔離ゾーン

Tag = ‘VDI’
セキュリティ グループ = VDI

分散FWによるスケールアウト



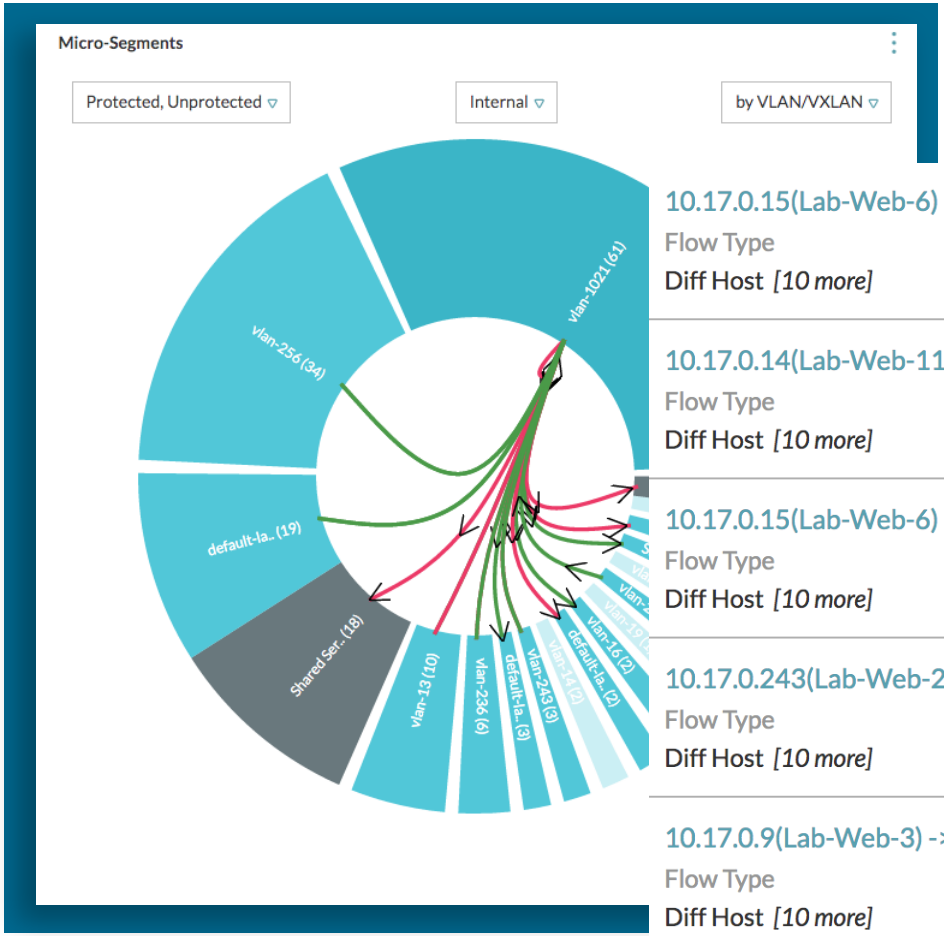
並べるだけです

仮想ネットワークの可視化

- ネットワークを仮想化するとトラフィック処理がどうなっているのかわかりづらくなる印象がある（気がする）
- 仮想スイッチでflowを取ればEast-Westのトラフィックも丸見え
- ネットワーク構成も可視化しやすくなる

ネットワークを仮想化することで、可視化がし易くなる

HyperVisor上の仮想SWからflowを取得



10.17.0.15(Lab-Web-6) -> 10.17.13.17(Lab-Midtier-13) [port:8080]

Flow Type	Src IP	Dst IP	Port	Protocol
Diff Host *[10 more]*	10.17.0.15	10.17.13.17	8080	TCP

10.17.0.14(Lab-Web-11) -> 10.17.13.10(Lab-Midtier-7) [port:8080]

Flow Type	Src IP	Dst IP	Port	Protocol
Diff Host *[10 more]*	10.17.0.14	10.17.13.10	8080	TCP

10.17.0.15(Lab-Web-6) -> 10.17.13.10(Lab-Midtier-7) [port:8080]

Flow Type	Src IP	Dst IP	Port	Protocol
Diff Host *[10 more]*	10.17.0.15	10.17.13.10	8080	TCP

10.17.0.243(Lab-Web-2) -> 10.17.13.22(Lab-Midtier-12) [port:8080]

Flow Type	Src IP	Dst IP	Port	Protocol
Diff Host *[10 more]*	10.17.0.243	10.17.13.22	8080	TCP

10.17.0.9(Lab-Web-3) -> 10.17.13.20(Lab-Midtier-6) [port:8080]

Flow Type	Src IP	Dst IP	Port	Protocol
Diff Host *[10 more]*	10.17.0.9	10.17.13.20	8080	TCP

IPだけではなく、VM名も一発で見える

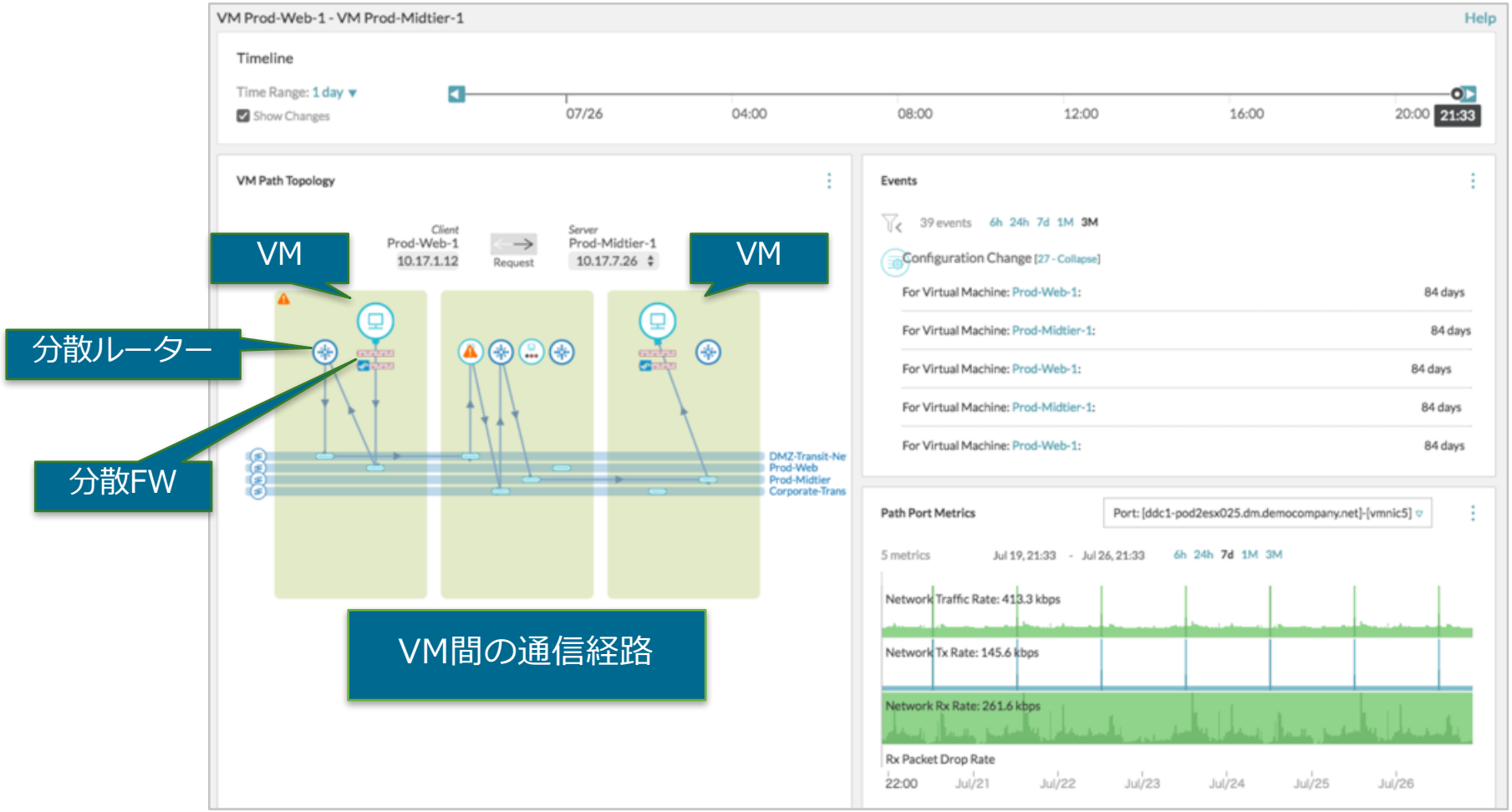
構成情報(Path)の可視化

- 通信は、基本的にはtable情報を追いかけていけば良い
  - mac-table
  - arp-table
  - routing-table
  - firewall

仮想スイッチ、仮想ルータが一元的に用意されて、
統一化されたAPIがあれば構成情報を組み立てやすい

構成情報を時系列に持つことで、過去の構成を再現することも可能

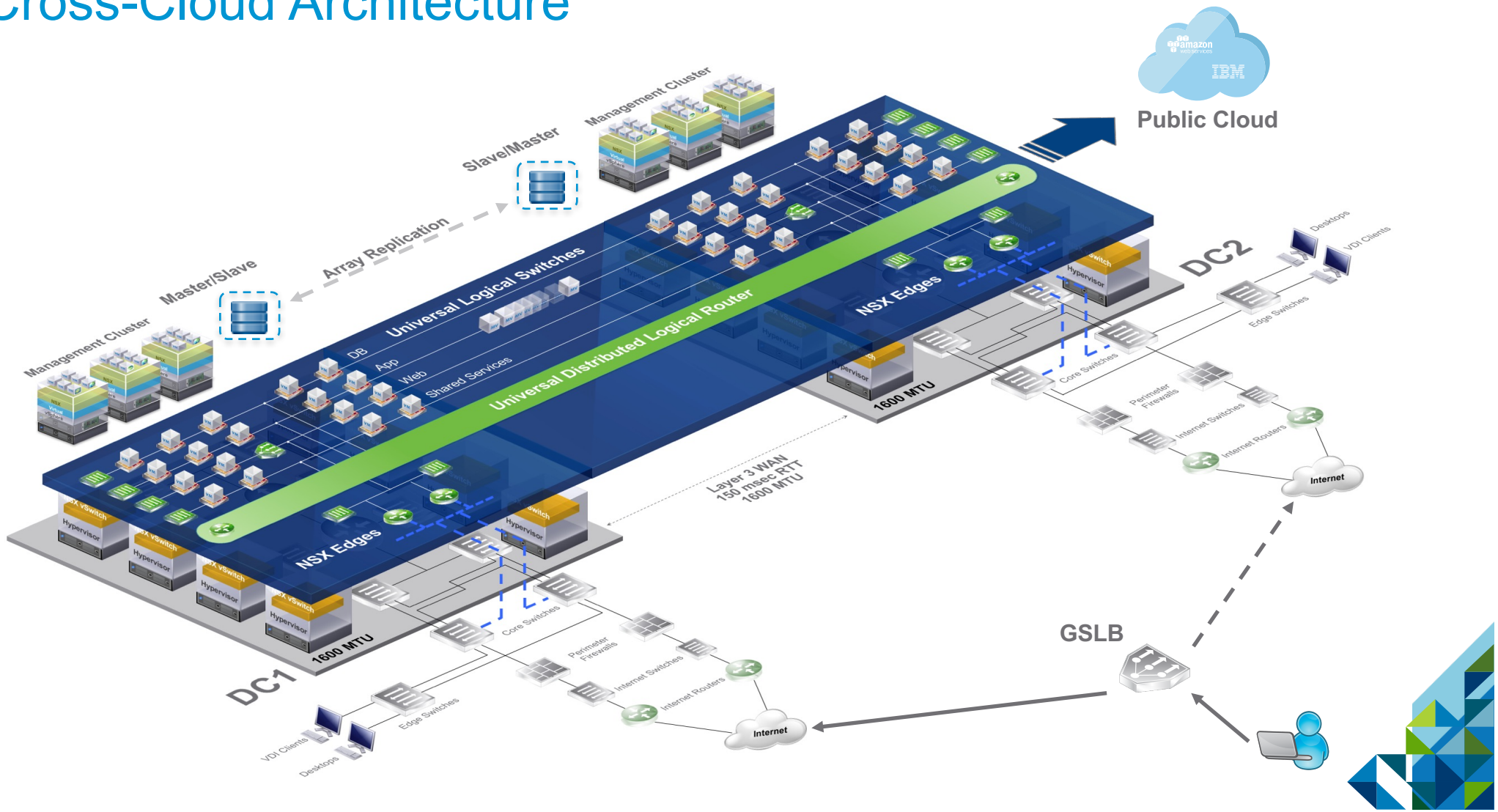
構成情報の可視化



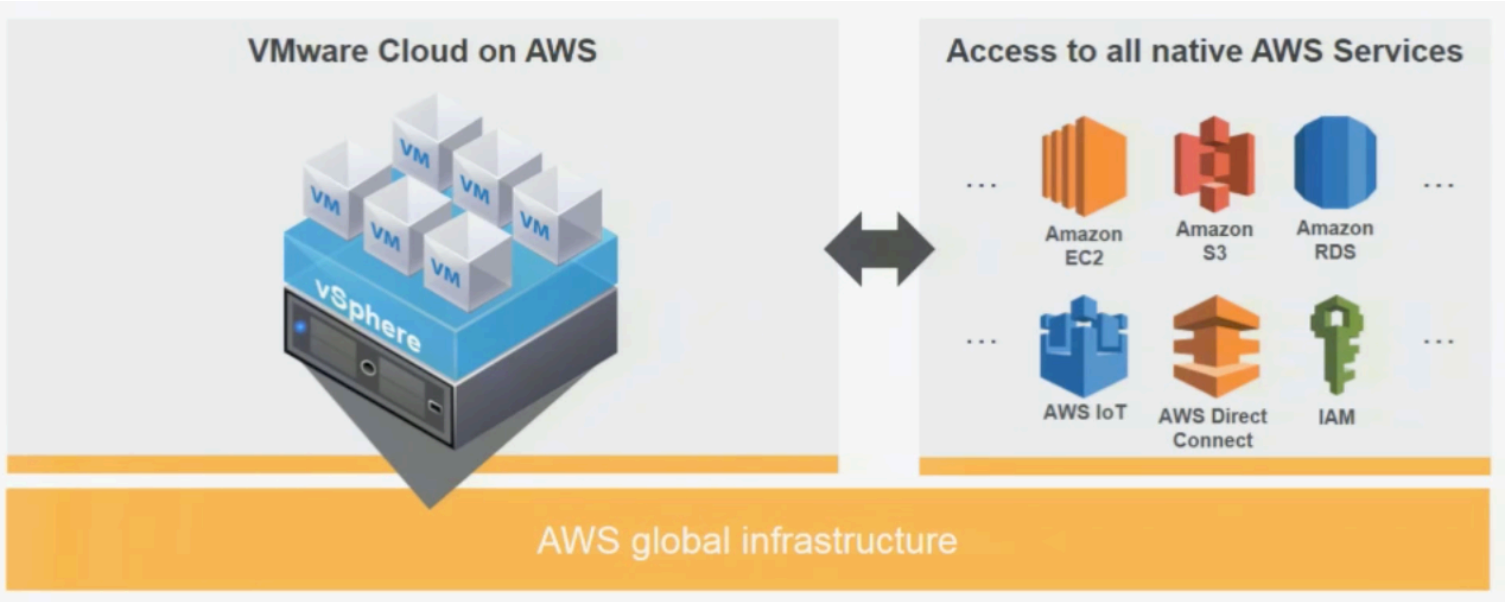
構成情報を時系列に持つことで、過去の構成を再現することも可能

ネットワークの仮想化による今後

Cross-Cloud Architecture

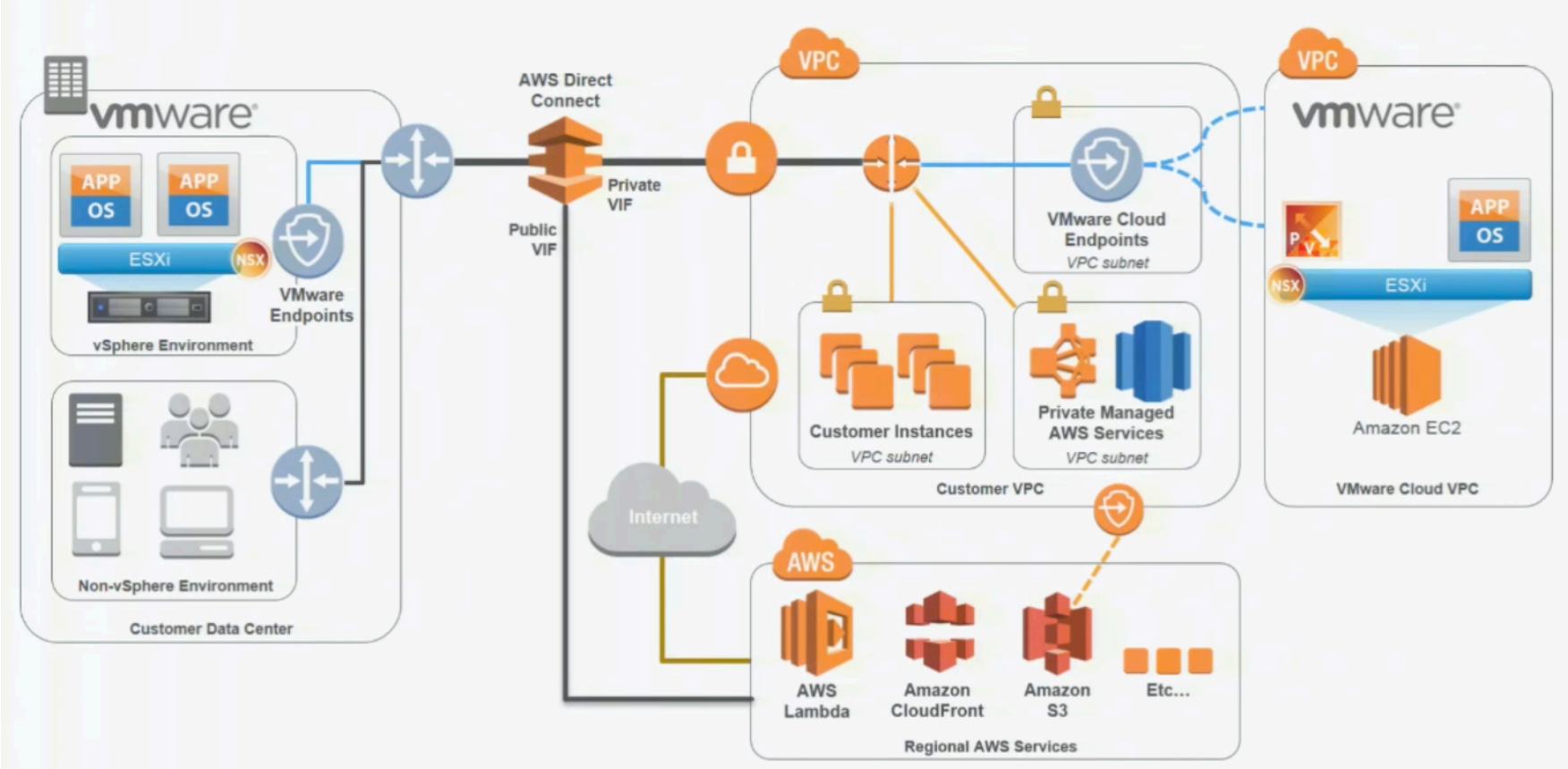


Cross Cloud



AWSのインフラ上にvSphere(サーバー仮想化), NSX(NW仮想化), vSAN(ストレージ仮想化)を実現し、AWSのクラウドネイティブなサービスと接続

Cross Cloud



Cross Cloud

Datacenterはいくつ?

バージニアの詳細を教えて

バージニアDCに5ホスト追加して

アイルランド、バージニア、オレゴンの3DCです。

ホストは8台あり、正常状態です。
キャパシティ使用率は87%です。

バージニアDCに5ホスト追加しました

ご清聴、ありがとうございました！

For more information, please contact me:

vmware
Issei Inoue: {
r: senior systems engineer
g: networking + security
m: +81 1596 2825
e: iinoue@vmware.com
f: https://www.facebook.com/inoue.issei
t: https://twitter.com/inoueissei
}