



プライベートクラウドを"継承"する Cycloudのネットワークを全面刷新した話

JANOG58 Day1

株式会社サイバーエージェント グループIT推進本部 CIU

小障子 尚太郎(Shotaro Koshoji)

自己紹介

小障子 尚太郎(Koshoji Shotaro)

- **2021/12 株式会社サイバーエージェント 中途入社**
 - 所属: グループIT推進本部 CIU Platform Div NWチーム
 - 担当: プライベートクラウドのネットワーク業務全般
 - AS24284運用(Peering・拠点間接続)
 - IaaS基盤・GPU基盤のNW設計・構築・運用 など
- **JANOG44から参加**



Agenda

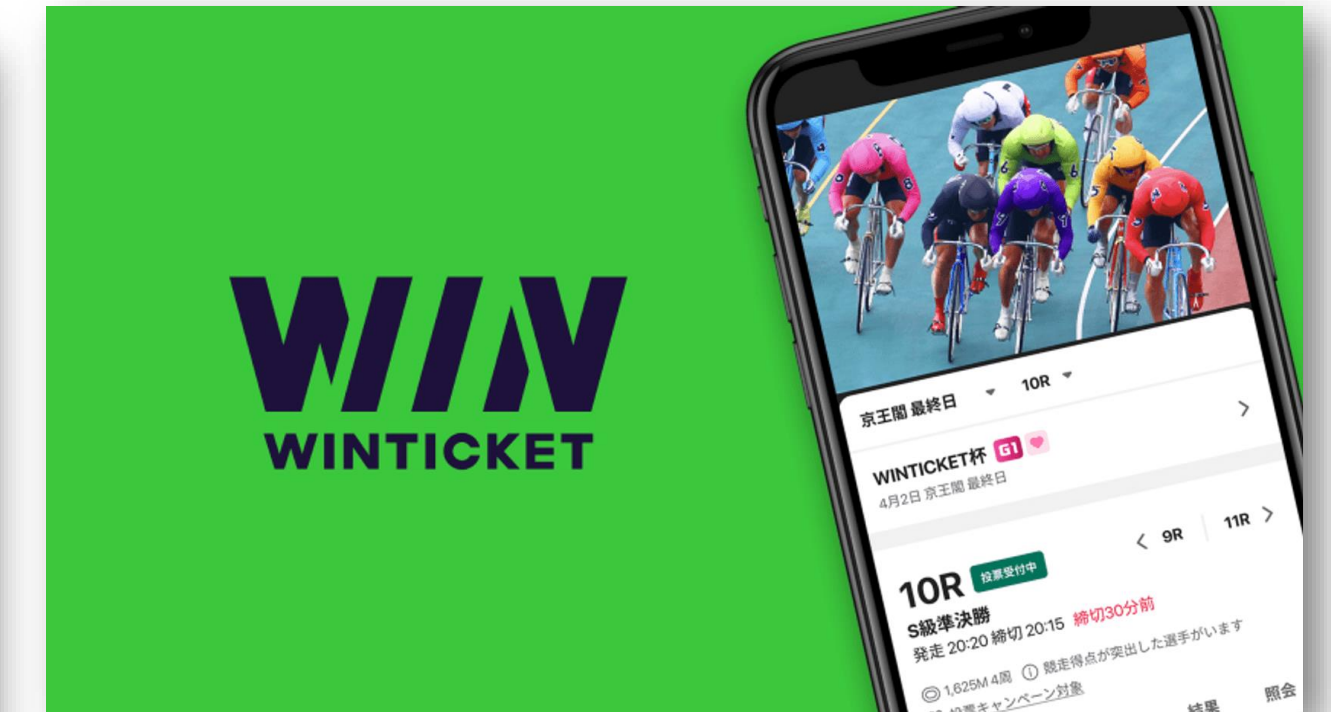
1. Cyccloud 現行リージョンの課題
2. Cyccloud 新リージョンの構成
3. 苦劳したこと・まとめ
4. 議論・会場に聞きたいこと

1. Cycloud 現行リージョンの課題

サイバーエージェントの主要事業と関連会社

主な事業

- メディア&IP事業(e.g. ABEMA)
- 広告事業
- ゲーム事業(e.g. Cygames)
- その他



tapple

 Cygames



CIU (CyberAgent group Infrastructure Unit) は、CyberAgentグループ全体のインフラを支える組織です

Cycloud というブランドでプライベートクラウドを展開しており、IaaSとしてのOpenStackやKaaSであるAKEなど様々なサービスを提供しています

プライベートクラウド - Cycloud



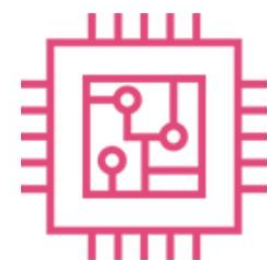
Cycloud

Compute



AKE

GKE相当のKubernetes as a Service



Cycloud Compute

VM、ブロックストレージ、ロードバランサー、ファイアウォールなどのIaaS機能を提供



ML Platform

GPUが利用可能なコンテナプラットフォーム、および機械学習基盤



Cycloud Run

プライベートクラウド Cycloud 向けに提供するサーバーレスサービスです。



CDB

マネージド MySQL サービス



S4

AWS S3に互換性をもったオブジェクトストレージ

CI / CD



hosted runner

安価かつ強力なスペックを持つGitHub Actions Runner



Container Registry

コンテナレジストリ

Platform Management



IAM

Cycloud Platformに統合されたアクセス制御を提供するサービス



Cycloud CLI

Cycloud Platform のためのコマンドラインインターフェイス



Terraform Provider

Cycloud Platform のための Terraform Provider

Tools



action-configure-credentials

クレデンシャルレスにCycloud Platformを操作するための GitHub Action

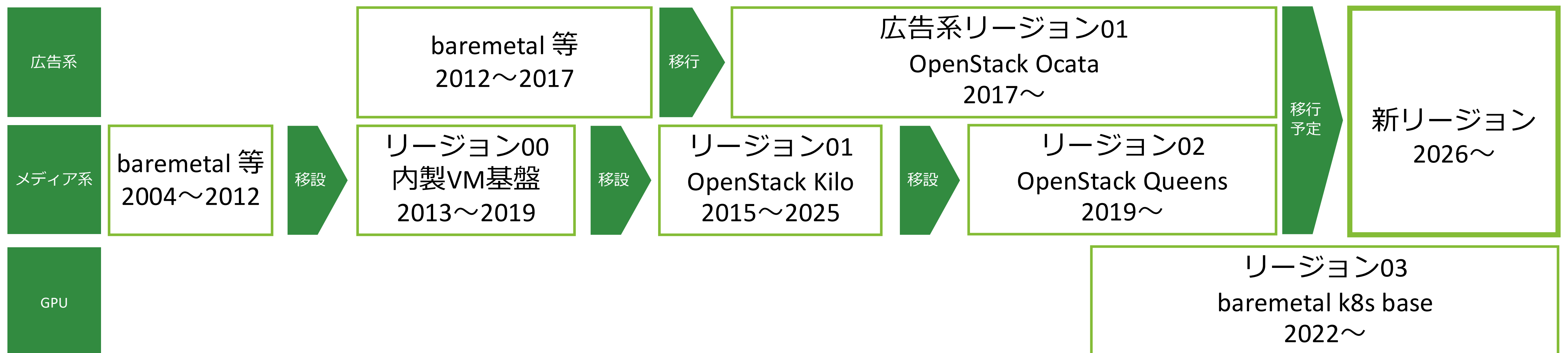


action-setup-cycloud

Cycloud CLI をインストールする GitHub Action

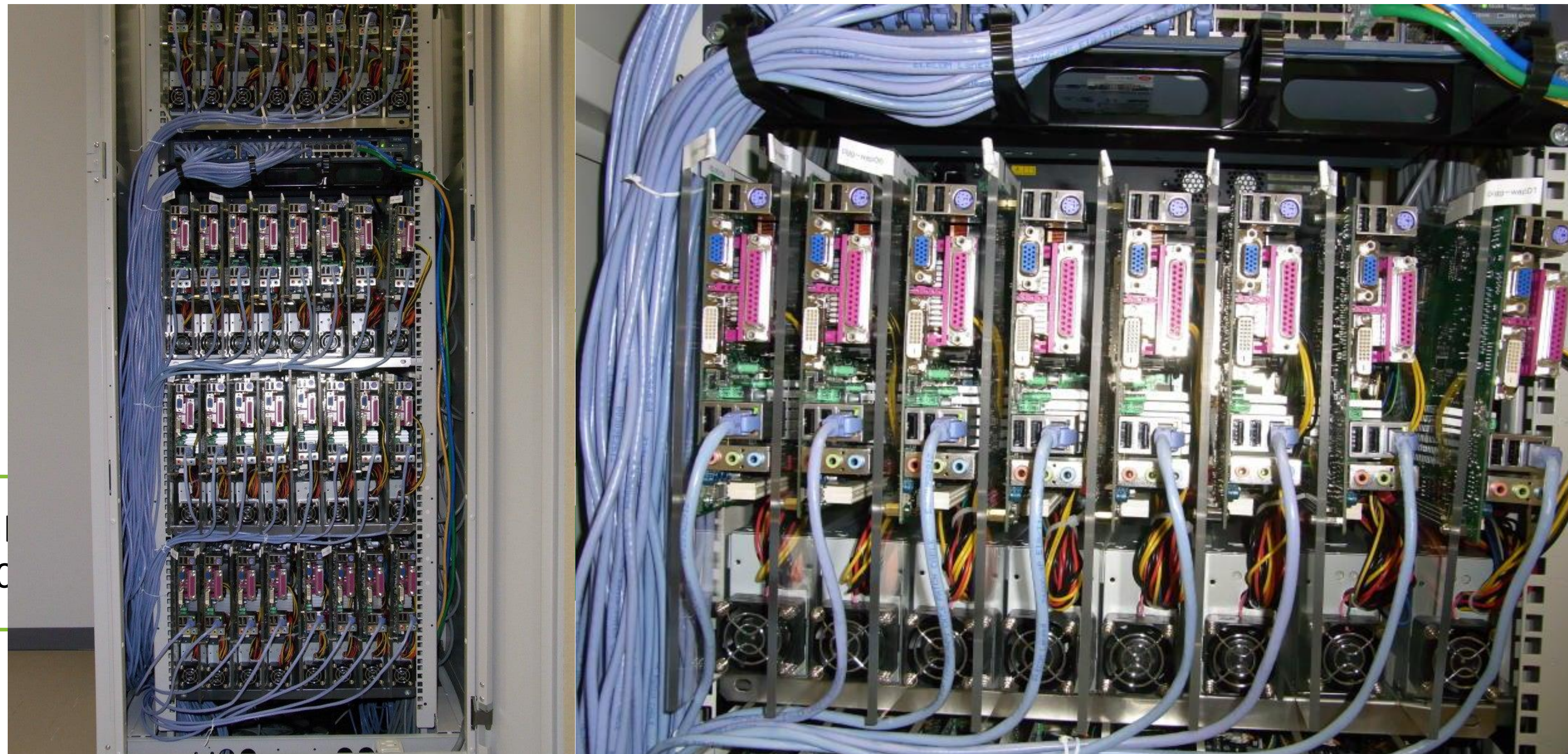
Cycloudのこれまで

- サイバーエージェントでは2004年頃から内製でプライベートクラウドを運用



Cycloudのこれまで

- サイバーエージェントでは2004年頃から内製でプライベートクラウドを運用
自作サーバーでサービスしている時代もありました



広告系

メディア系

GPU

baremetal
2004～20

新リージョン
2026～

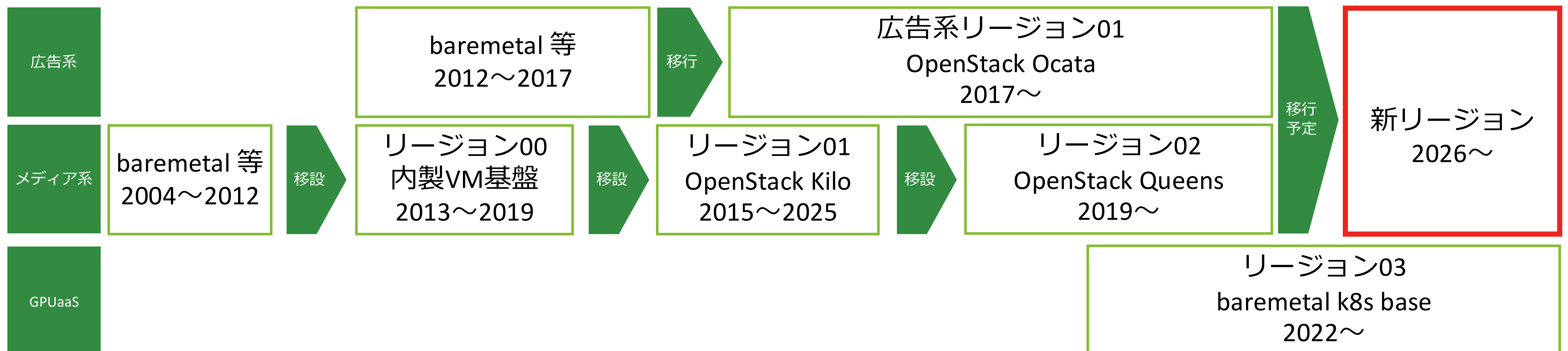
ン03
8s base

<https://ameblo.jp/principia-ca/image-10741032111-10940358224.html>

Cycloudのこれまで

- ・サイバーエージェントでは2004年頃から内製でプライベートクラウドを運用
- ・3つのリージョン(広告系・メディア系・GPUaaS)を運用中
 - ・CIUは広告事業部・メディア事業部のインフラ運用組織が統合された組織
- ・数年毎にデータセンターを含めたリージョン全体の式年遷宮を実施

最新のCPU基盤は2019年より運用しているもので課題も多く、2024年頃から次期基盤の検討開始



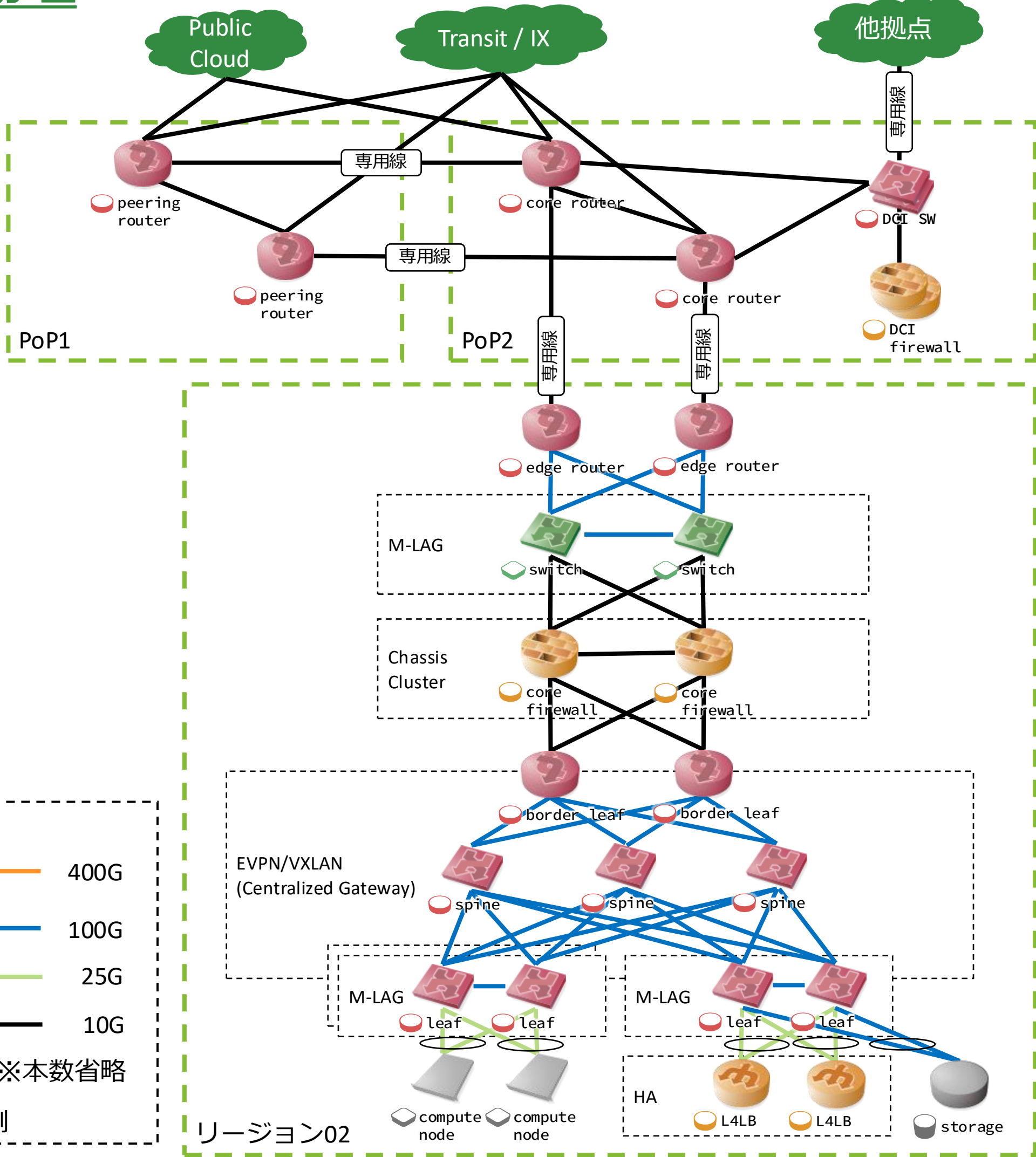
Cycloud 現行リージョン全体の課題

- **式年遷宮によるユーザーの運用負荷**
 - 数年に1度のサービス基盤の移設作業・仕様変更による移行作業が重い
- **式年遷宮前提の構成のため更新が難しく、基盤全体のスペックが年々陳腐化**
 - アップデート・リプレースしない前提の設計が多い
 - 最新技術が導入しにくく技術的負債になっている
- **パブリッククラウドに慣れたユーザーにとって使いにくい仕様**
 - VPC機能がないなどパブリッククラウドとの差異が大きい
 - Public Cloudに慣れたエンジニアが多く、素朴なインフラ仕様では導入・運用しにくい

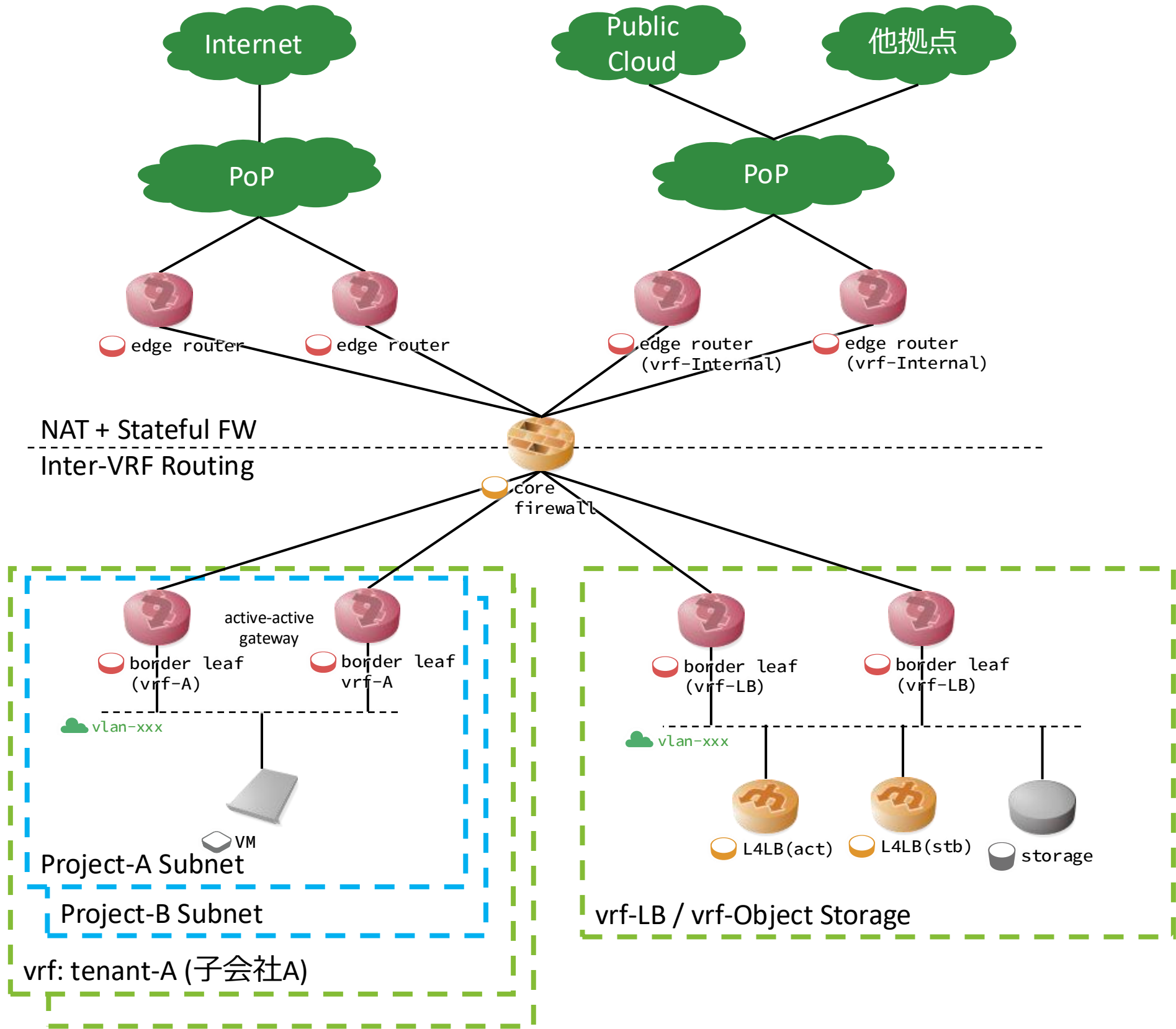
→ 新リージョンの設計に向けてネットワーク視点の課題を分析

現行リージョン02 ネットワーク構成

物理

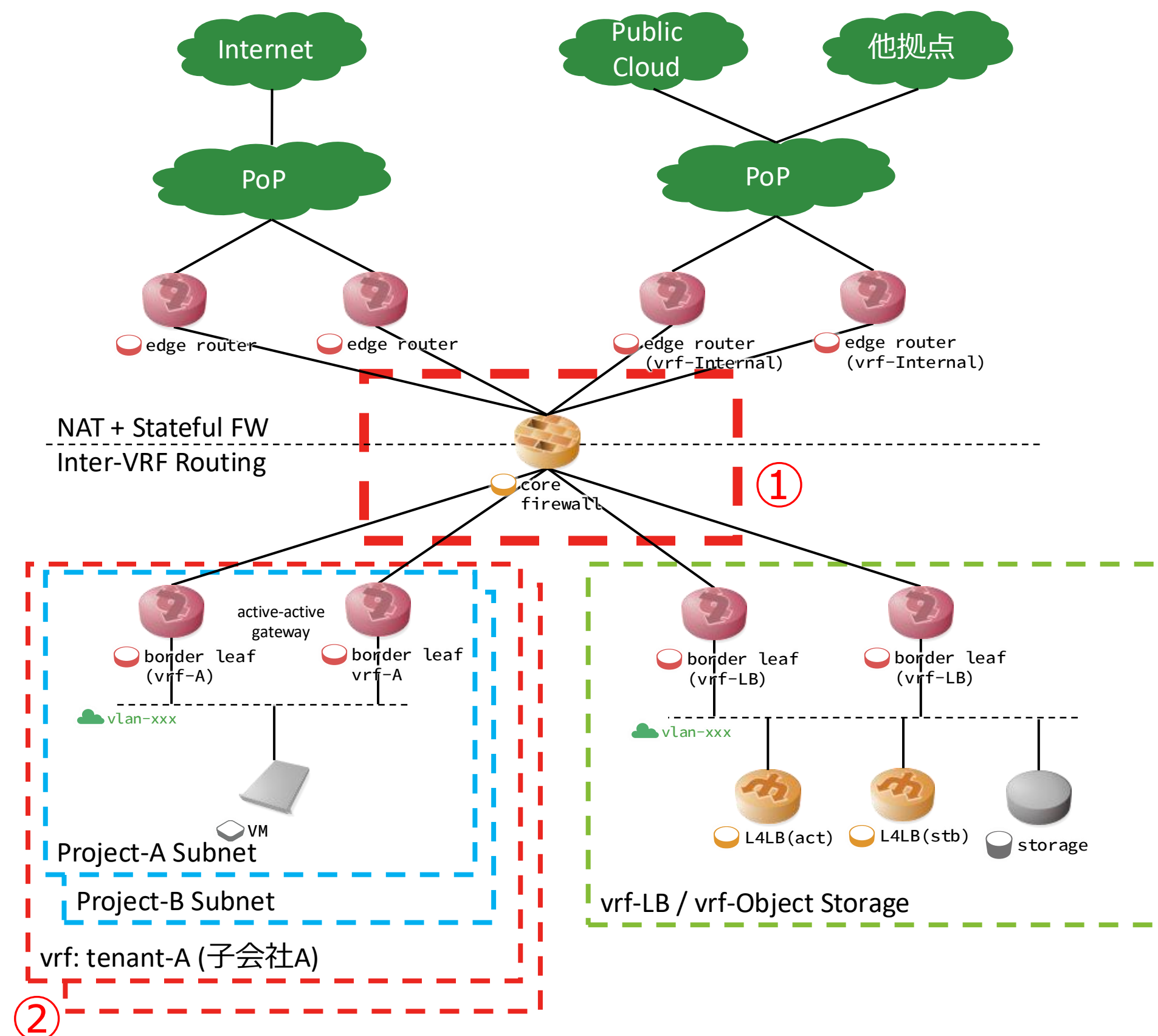


論理



ネットワーク構成の課題 (1)

論理



① テナント分離が子会社単位

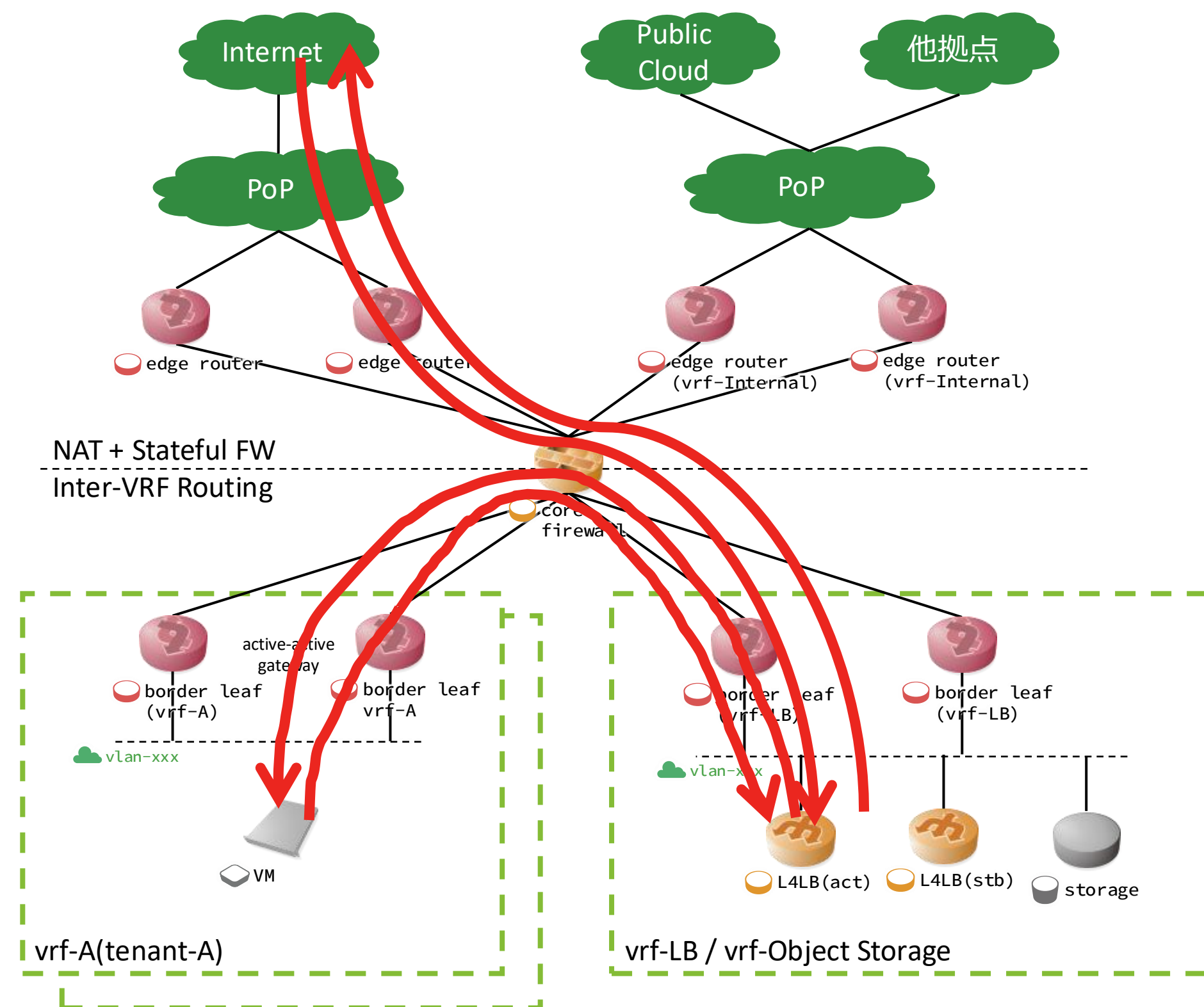
- VRFを子会社単位で作成、サービス間の通信制御はSecurityGroupベース
- セキュリティ的にプロジェクト単位でネットワークを分離したい

② FirewallがCentral Pointとなっている

- 全トラフィックが経由しているため、障害・メンテナンス影響が大きい
- リプレース・OSバージョンアップがやりにくい要因
- 折り返しトラフィックが増幅し帯域が埋まる

Firewall折り返しトラフィック問題

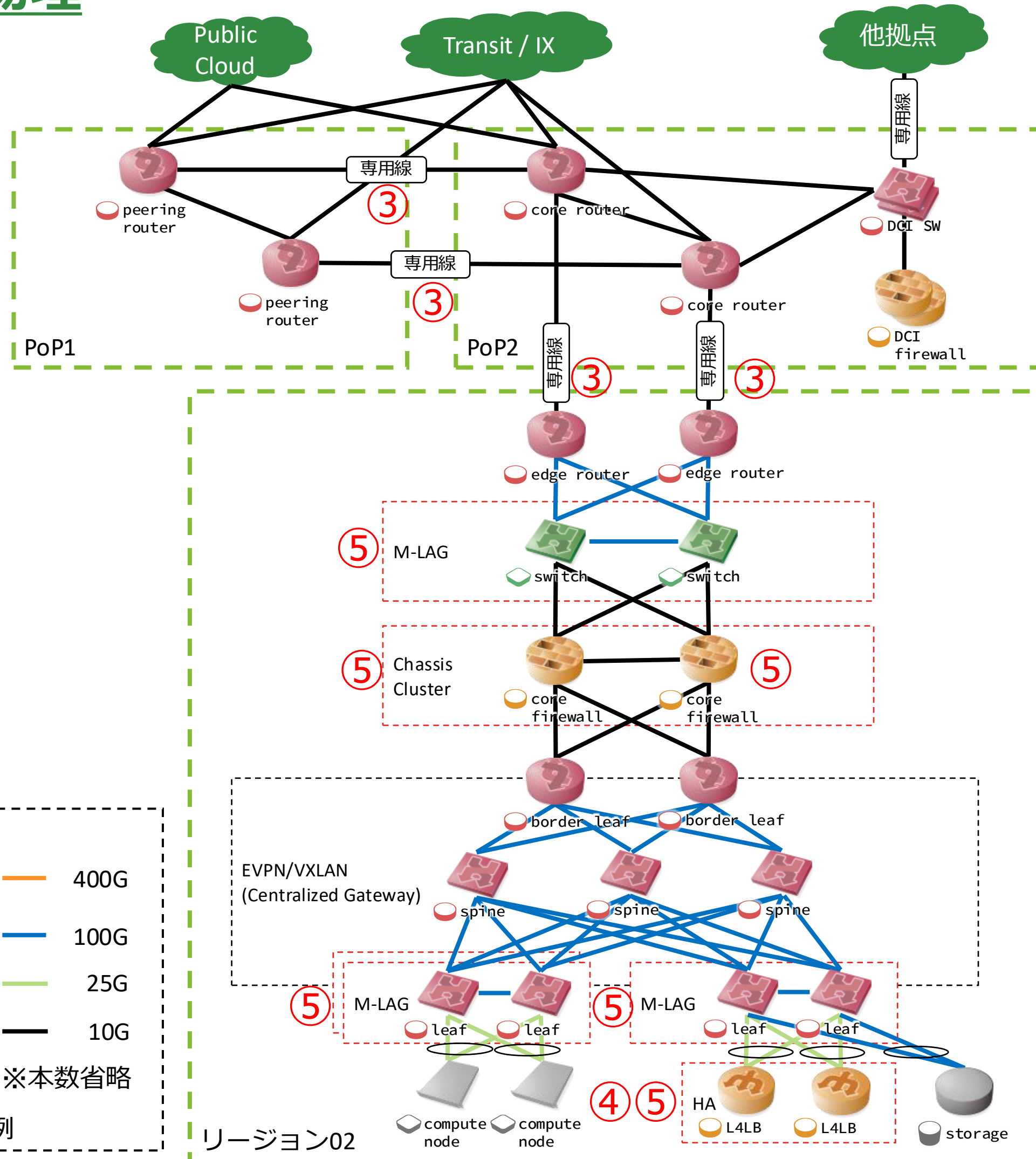
論理



LB <-> VMの折り返し通信で倍の帯域を消費
トラフィック単価の高いFirewallの実利用効率が悪く、増設してもすぐ埋まる

ネットワーク構成の課題(2)

物理



- ③ **10GベースのDCIで突発需要に対応できない**
 - 増設時のリードタイムが課題
 - トラフィックの増えるイベント情報を聞いてから増設では間に合わない
- ④ **L4LBの増設コスパが悪い**
 - 2台セットでの拡張コストが高い
- ⑤ **ベンダー独自の冗長化技術によりリプレース困難**
 - M-LAG / Chassis Cluster / HAは異機種で接続不可

ユーザー利便性の課題

① 手動対応によるスピード感の低下

- VRF / Vlan / IPアドレス / Firewall ポリシーなどの追加・削除が手動設定で対応に時間がかかる
- チケットシステムでの依頼が必要

② Cyclocloud 特有の仕様によるユーザーへの運用・認知負荷

- VMのL3DSRの設定(IPIP など)が複雑である
- ライブマイグレーション不可でユーザー負荷が高い
 - ホスト故障・メンテナンス時にユーザー側でVM待避作業が必要
 - leafスイッチのリプレースが難しい一因

ネットワーク構成の課題まとめ

- **ネットワーク構成の課題**

- ① テナント分離が子会社単位
- ② FirewallがCentral Point
- ③ 10Gベースの専用線により需要への対応が間に合わない
- ④ L4LBの増設のコスパが悪い
- ⑤ ベンダー独自の冗長化技術によりリプレース不可

- **ユーザー利便性**

- ① ネットワーク設定が手動対応であり対応が遅い
- ② Cyclocloud特有のユーザー仕様(パブリッククラウドとの仕様の差異)によるユーザーの運用・認知負荷

これらの課題を解決し長期間運用できるプライベートクラウドを目指す

新リージョンのネットワーク設計の方針

1. ネットワーク分離のためにVPC機能を提供
2. 標準化技術を採用し全機器メンテナンス・拡張可能な構成
 - 現行リージョン以上のコストパフォーマンスも実現する
3. パブリッククラウドに慣れたユーザーでも利用しやすくする
 - ネットワークコンフィグ自動化によるスピード感
 - 余裕のあるネットワークリソース・ハイパーバイザーの性能

**独自のAPIと標準化されたネットワーク仮想化技術により、
ユーザーが意識することなく永続可能なネットワーク構成を検討**

VPC機能の要件

- **テナント間のネットワークアクセスをVPC単位で分離**
 - VPC間・既存リージョンを相互接続(VPC Peering)
 - 増設・移設を見越してデータセンターを跨いだVPC接続にも対応
- **OpenStackのライフサイクルに囚われない運用・機能追加**
- **やらないこと**
 - データセンターを跨いだL2延伸
 - L2をデータセンター内に留める(Availability Zone的概念)
 - IPアドレス重複は許容しない
 - 社内のプライベートIPv4アドレス在庫にまだ余裕がある
 - これらも後から対応可能な構成にする

VPC機能の実現方法検討

案1. ハイパーバイザーで実装

Pros

- ・ ソフトウェアの力でスケールしやすい・柔軟性

Cons

- ・ 自前で開発・運用する人的リソースが必要
- ・ ハイパーバイザーの責務が大きくなる
 - ・ OSバージョンアップなどやりにくくなりそう

案2. ネットワーク機器(EVPN/VXLAN)で実装

Pros

- ・ これまでのEVPN/VXLAN運用知見の活用
- ・ (乱暴だが)買えば動く

Cons

- ・ ASICの制約でリソース(VRF/Vlan/収容ホスト(mac-ip))のスケールが決まる
- ・ マルチベンダー対応が難しい

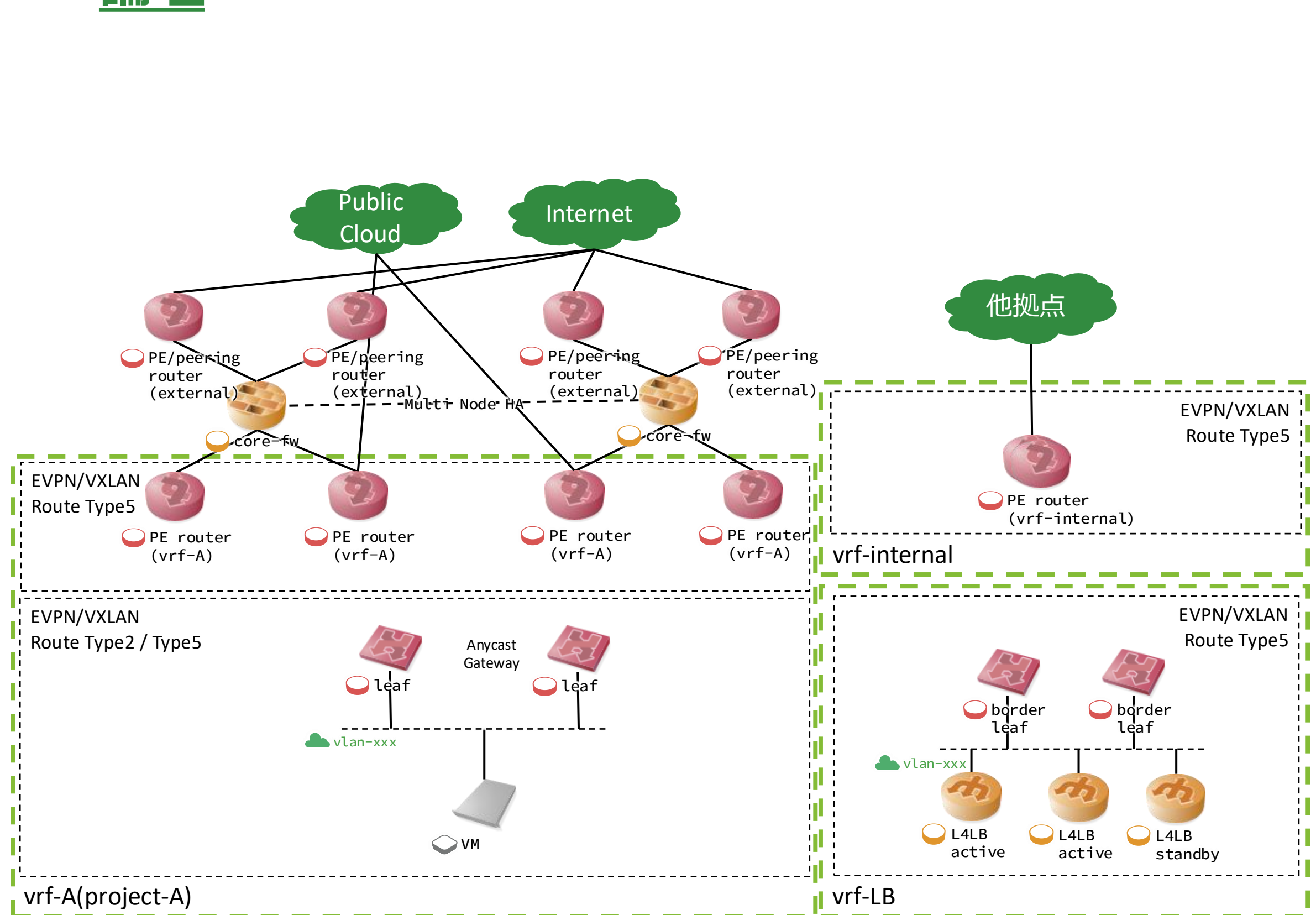
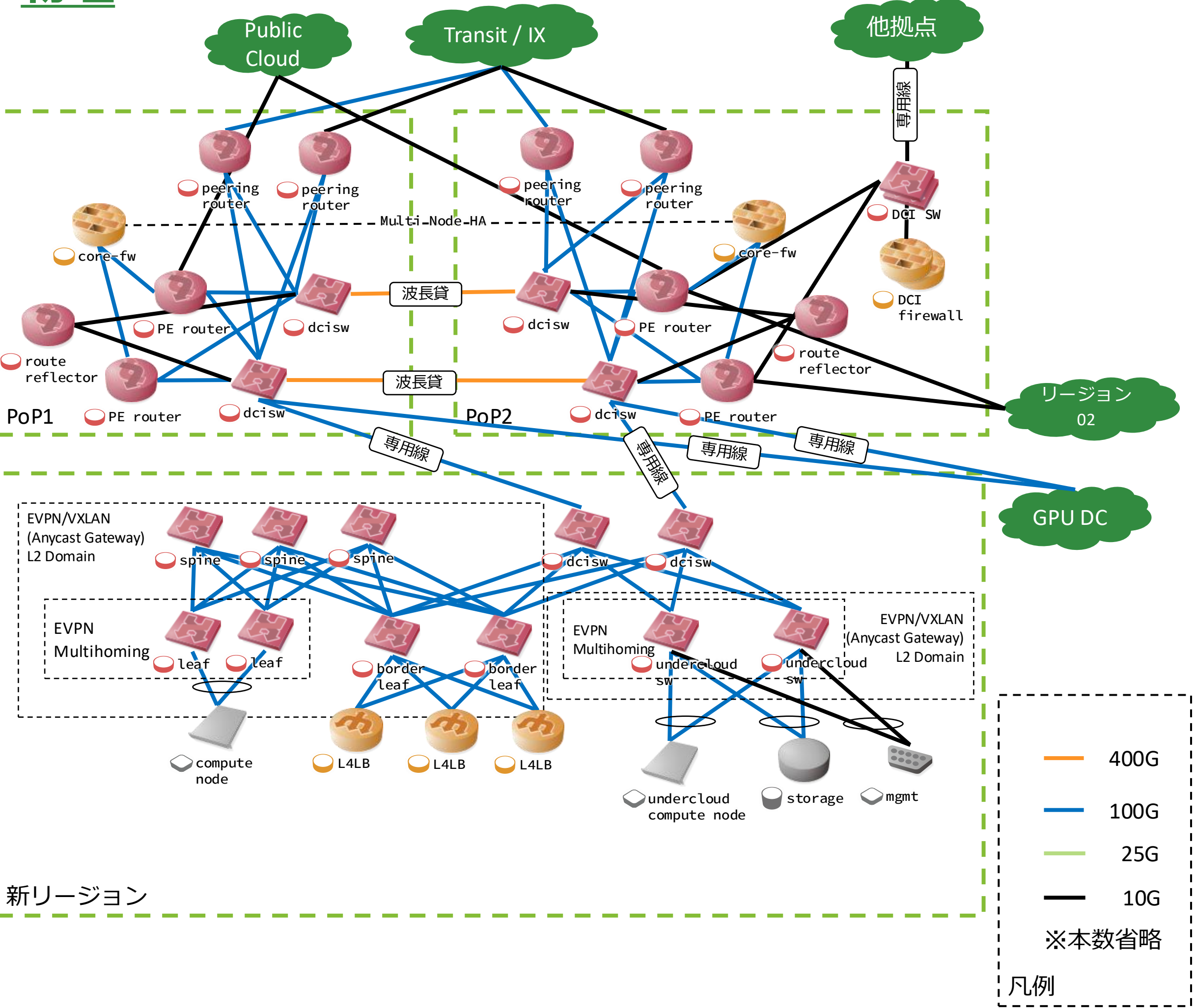
規模的には問題ないので導入・運用しやすいネットワーク機器(EVPN/VXLAN)でVPC機能を実装する方針で設計

2. 新リージョンのネットワーク構成

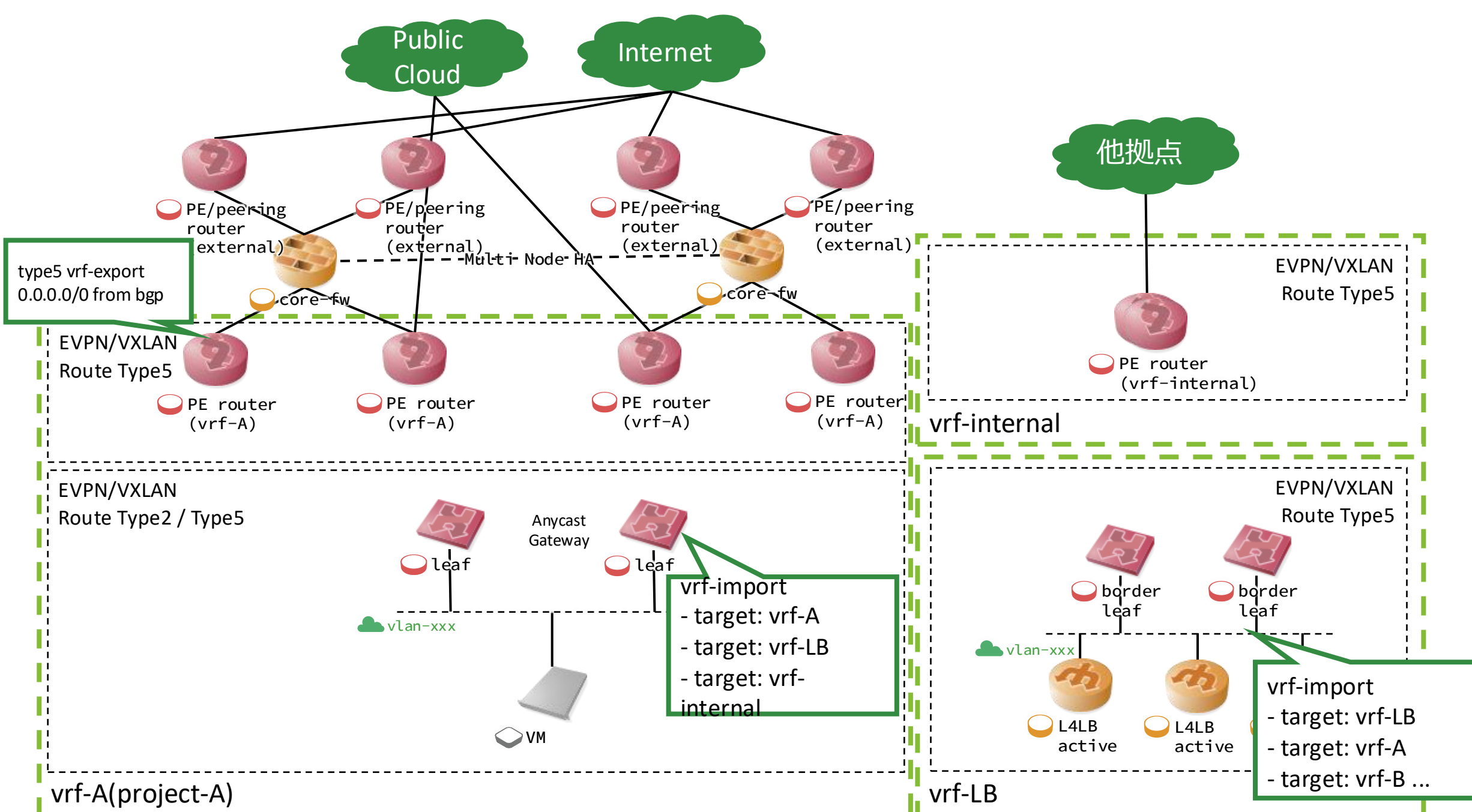
新リージョンのネットワーク構成

物理

論理



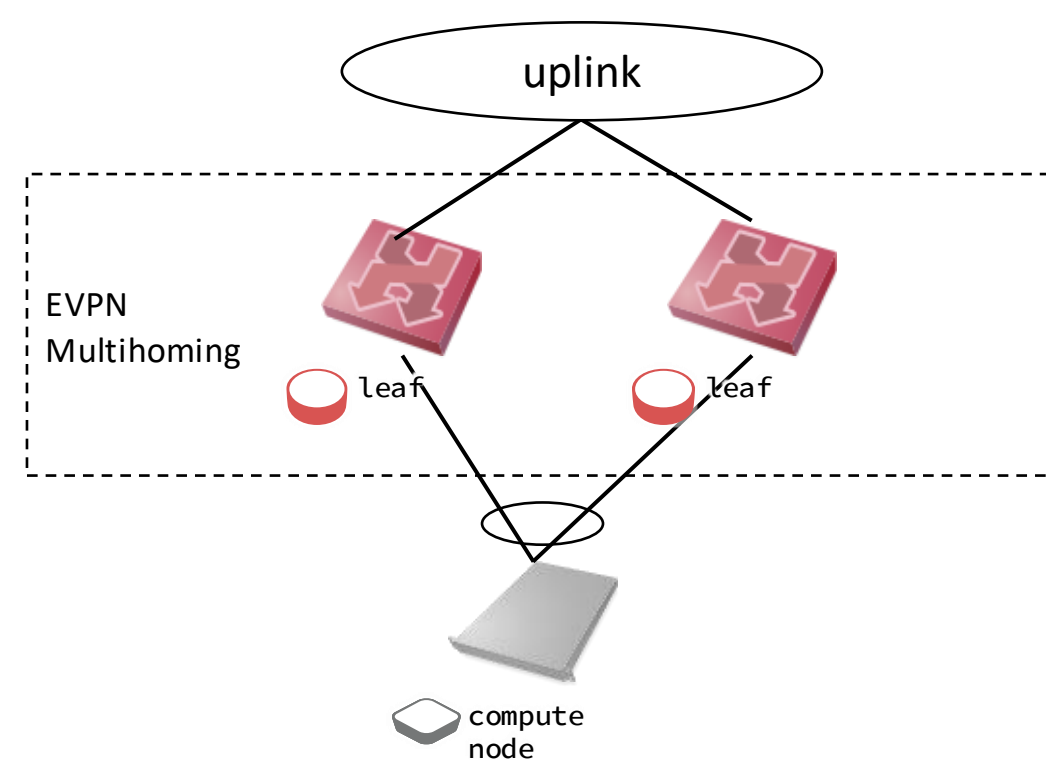
EVPN Route Type-5によるVPC



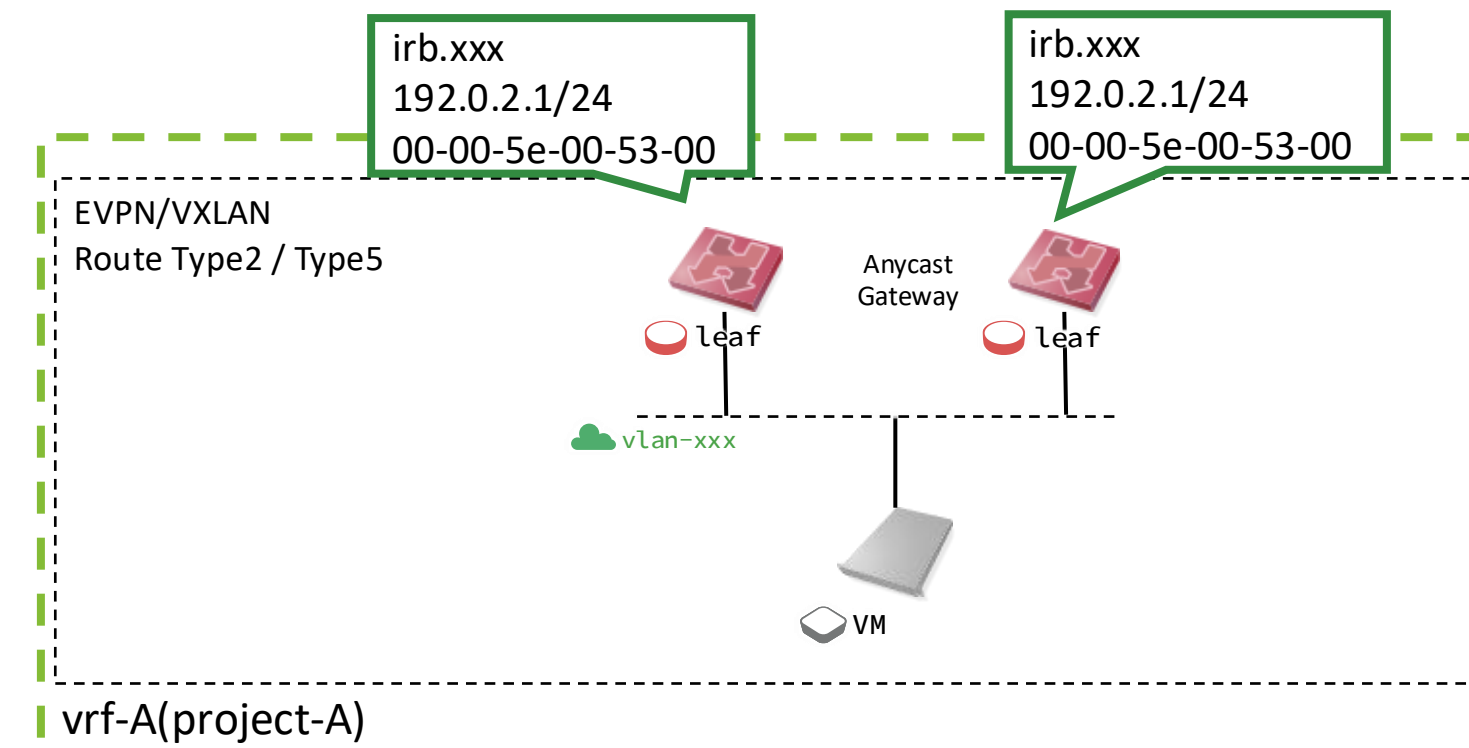
- **EVPN Route Type-5(IP Prefix Route)**
 - IP Prefixの情報をEVPNで広告するRoute Type
 - VRF/L3VNIと紐づけてVTEP間をルーティングできる
- **VRF相互のvrf-importでVPC Peering**
 - Firewallを通さずVPCを相互接続
 - VM間の通信制御はSecurity Groupで担保
- **データセンターを跨いだVPC接続(DCI)**
 - NAT機能をネットワーク拠点(PoP)に集約
 - 将来的なサーバーファームの増強・移設に対応
- **標準化技術により長期間運用しやすい構成**

EVPN Multihoming & Anycast Gateway

- EVPN MultihomingでLeafスイッチのL2冗長を実現
- Anycast GatewayによりVMのDefault GatewayをAct-Actで冗長化
- 標準化技術によりLeafスイッチがリプレース可能に

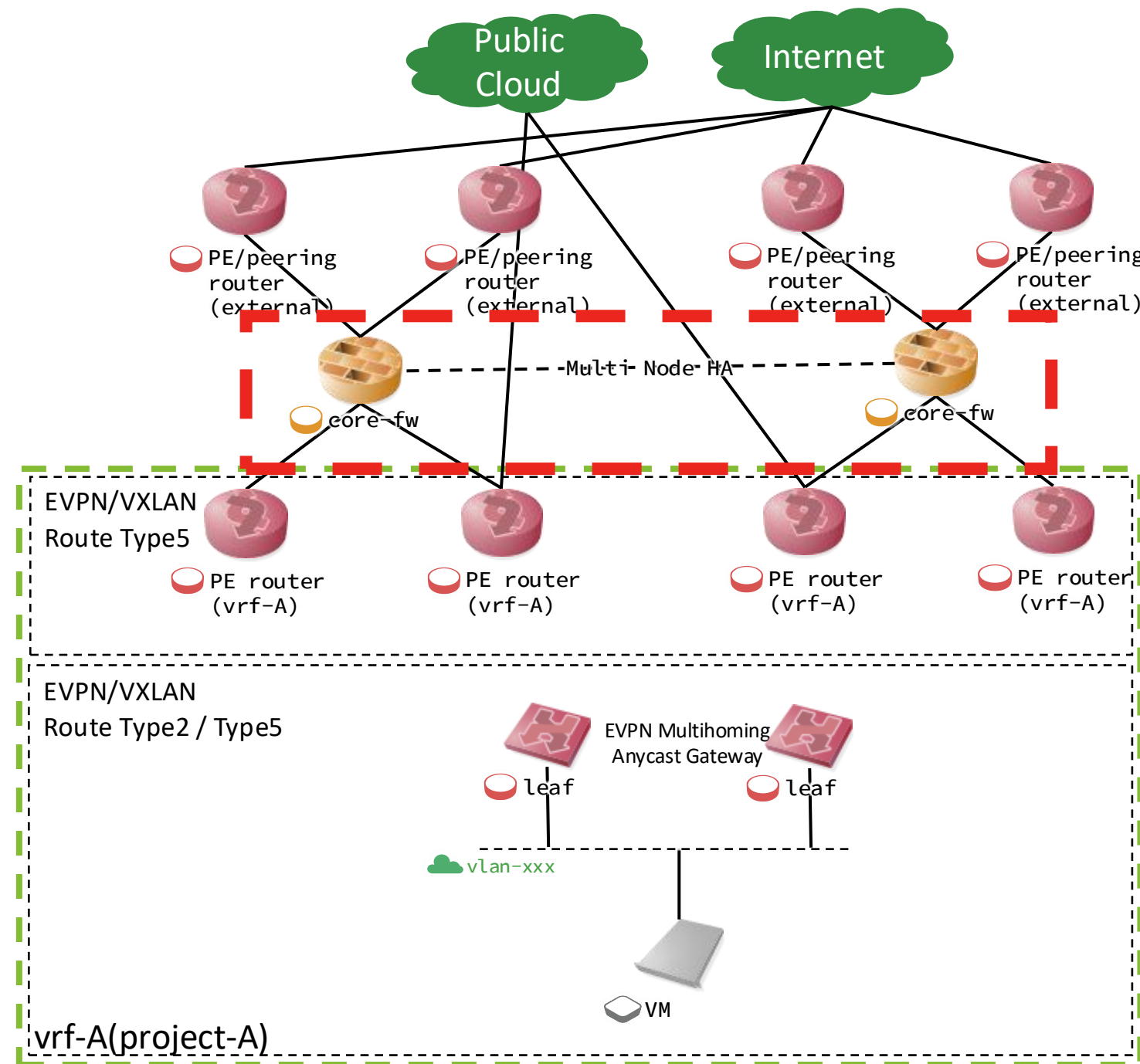


物理構成



論理構成

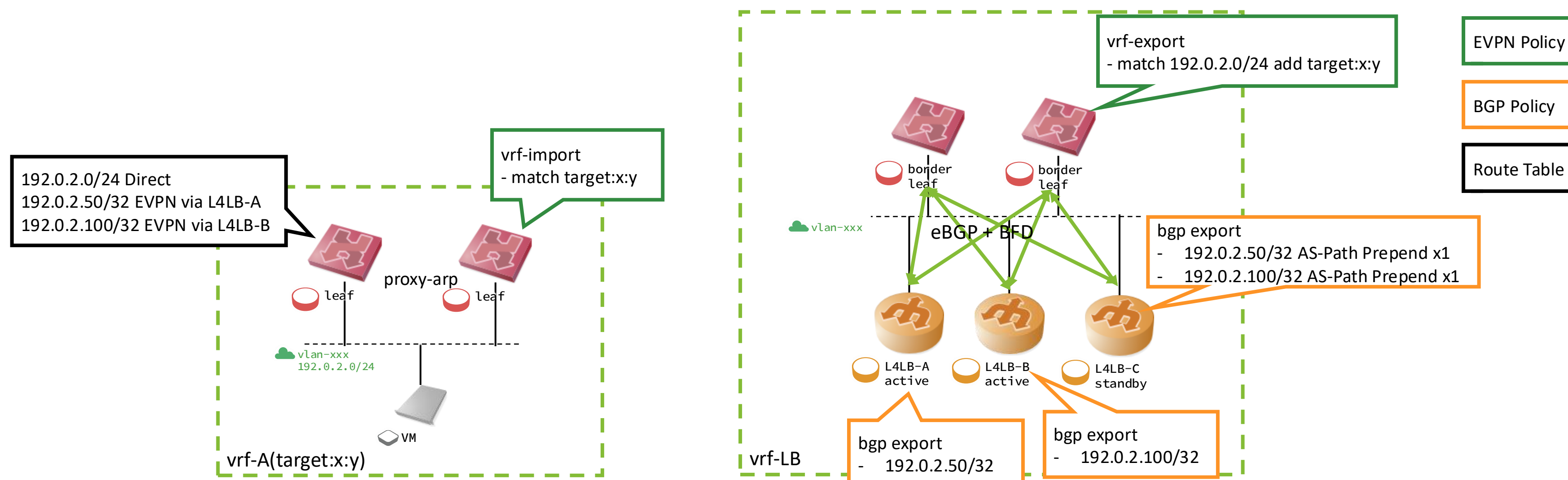
NAT Gatewayの分離・集約



- NAT Gatewayをネットワーク拠点に分離・集約
- 共通リソースとすることでDC追加時の初期コストを削減
- データセンターを跨いだHA構成(Multi Node HA)
 - メンテ影響低減のためのコネクションミラー
 - ベンダー独自技術だが、メンテナンス性・拡張性も問題無し
 - NAT Pool を拠点ごとにアサインすることでリプレイス可能
 - Active – Active 構成、4台までのスケールアウト対応
 - 異バージョンでのコネクションミラー対応
- NAT Gatewayをコスパ良くリプレイス可能な構成を実現

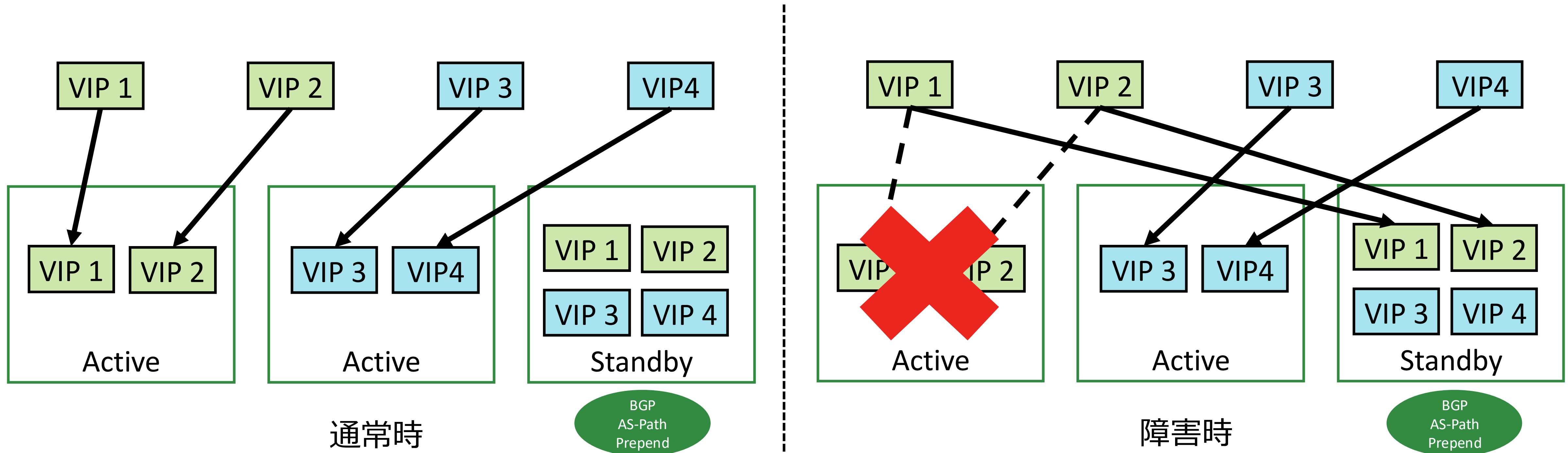
L4 Load BalancerのN+1構成

- BGP + 自作コントローラーでN+1構成のL4LBを開発
 - EVPN Route Type-5 による VPC / Subnetとの接続
- BGPを喋ればベンダー問わずクラスターへの組み込みも可能
 - 現状は物理アプライアンスを採用したが、今後Linuxベースの自作L4LBも検討
- L4LBを1台ずつ増設可能・リプレイス可能な構成を実現



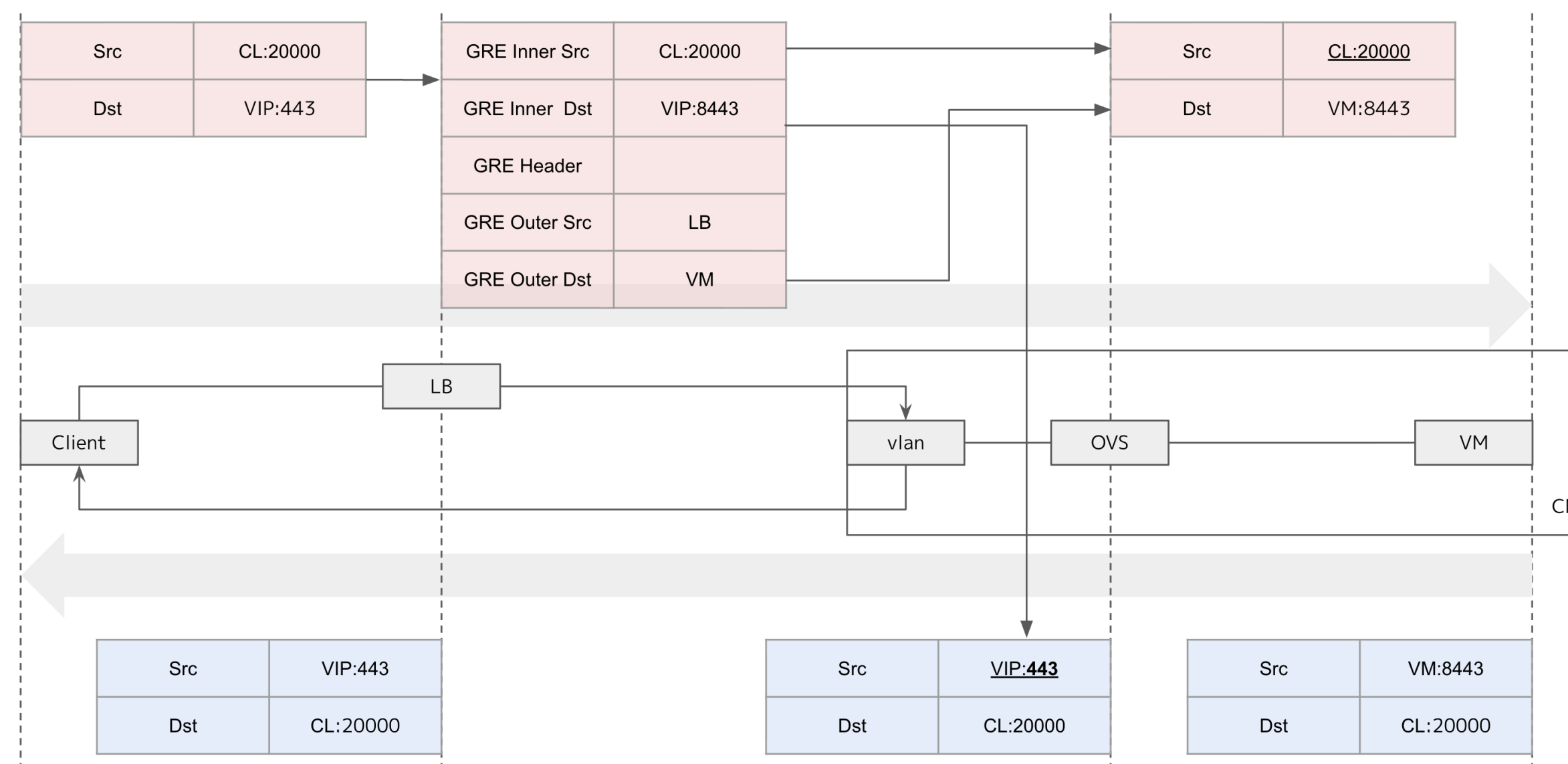
L4 Load Balancer N+1構成の動作イメージ

- BGP+BFDによるVIPの経路切替
- Standby機器に全てのVIPを作成しBGP AS-Path Prependを入れておく



L4 Load Balancer - 透過L3DSR

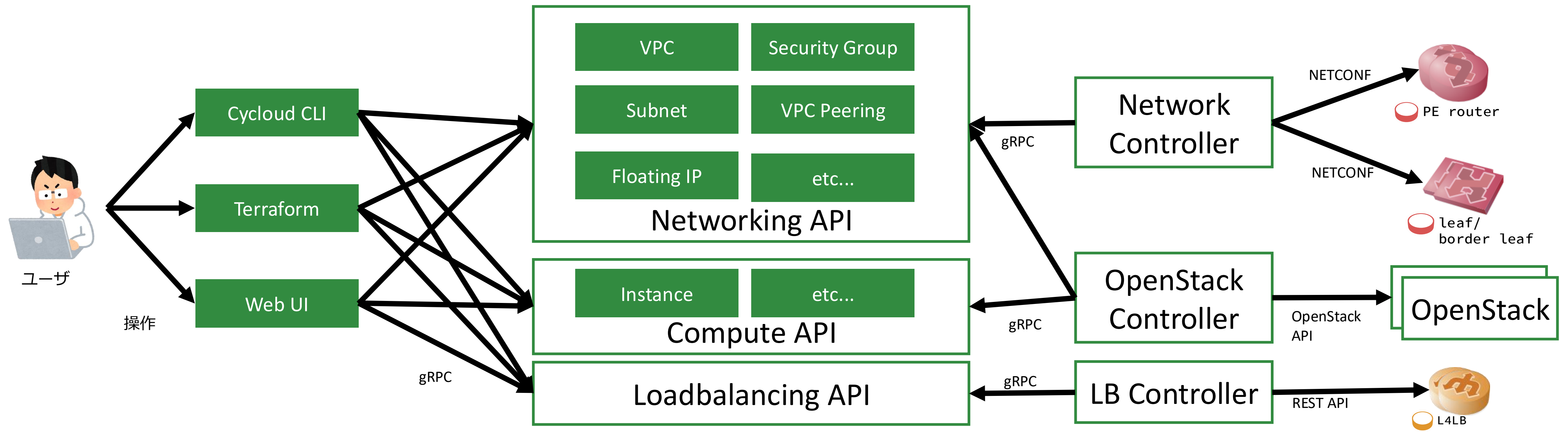
- L3DSRを利用するにはユーザーVMに特別な設定(DSCP, IPIP, GRE 等)が必要
- ハイパーバイザー上でDSCP / GRE / IPIPをいい感じに処理すれば、VM側に特別な設定は要らないのではというアイデア



今回はスケジュール都合と実装上の課題があり断念 😞
現在はユーザーVMでProxy Protocol v2 or L3DSR の設定が必要

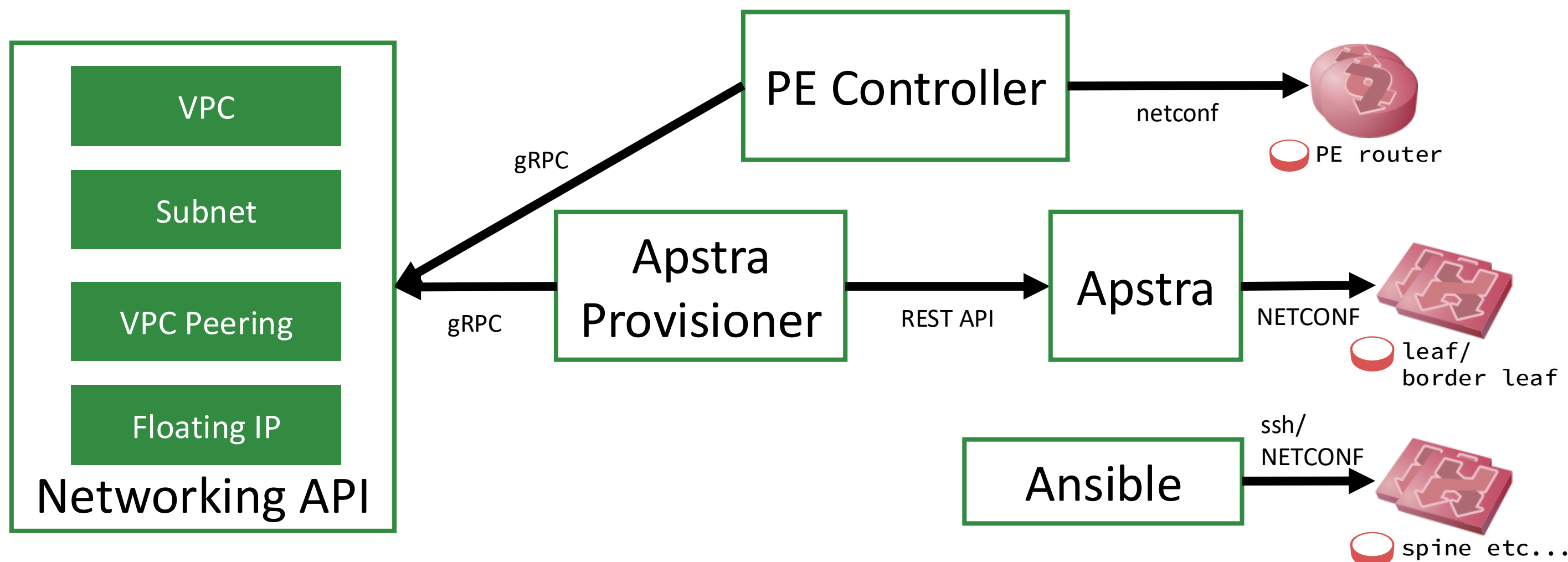
Networking / Compute API

- Network / Compute 機能をgRPC APIとして抽象化
- 独自のAPIにより長期運用に耐える柔軟性・独自機能を実装する余地を作成
- OpenStackのバージョン / ライフサイクルに縛られずに運用・機能追加が可能



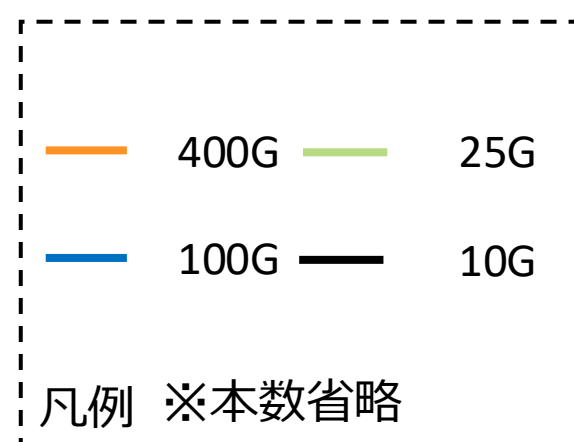
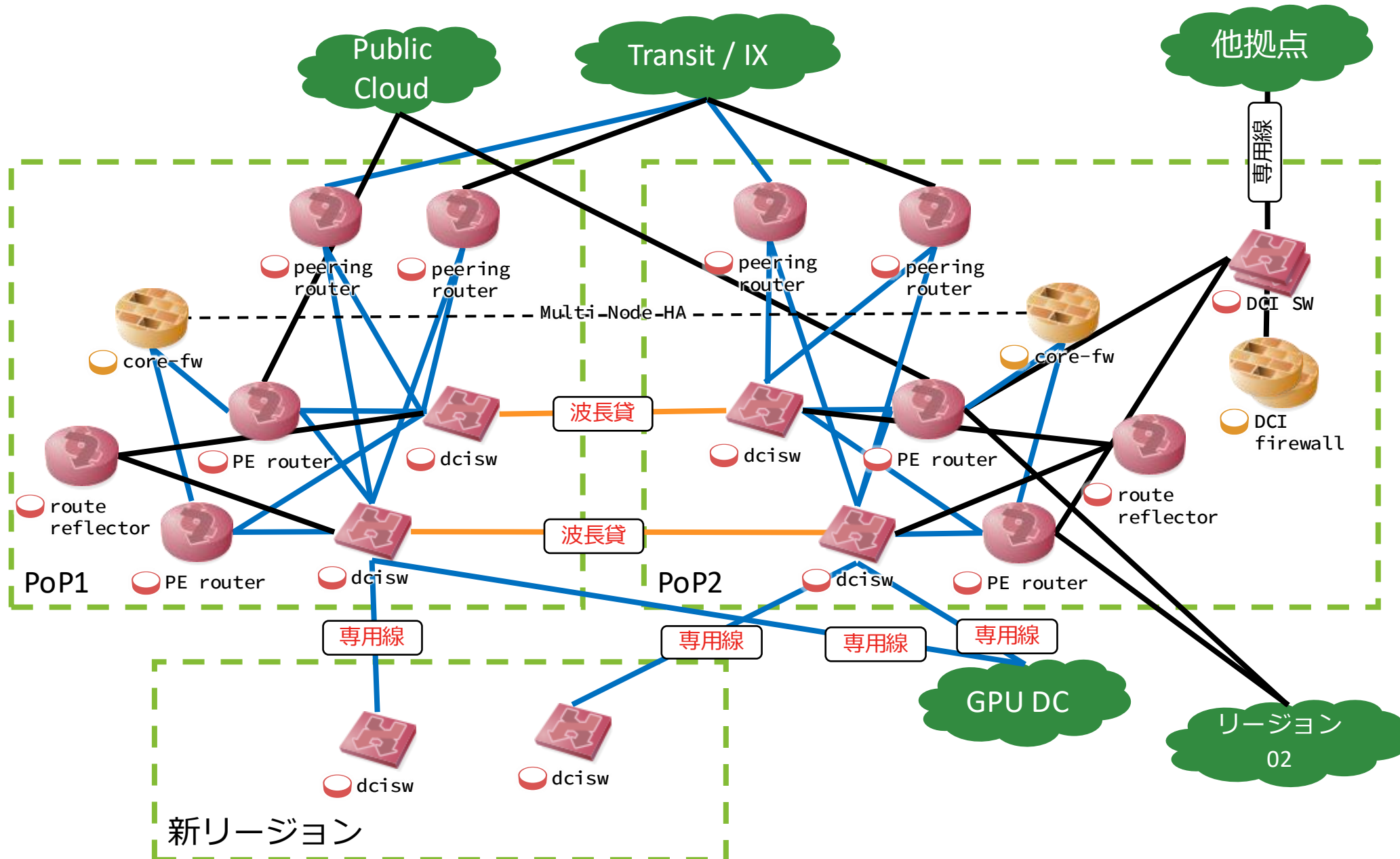
ネットワークコントローラー

- ネットワーク自動化のためコントローラーを機器のロール毎に開発
 - NW機器の実装コストの削減のため商用コントローラー・Ansibleを活用
- VPCやSubnetなどのネットワークリソースの操作を自動化



データセンター間接続の高速化

- 主要なデータセンター間接続を100/400GbE化
- NW拠点間の接続に波長貸しサービスと400G-ZR+0dBmを採用
 - アンプ等も不要で専用線同等の使い勝手
 - 安価だが運用者に伝送の知識も必要
 - 事前に実機での検証を実施
- 突発的な需要にも耐えられる帯域を確保



ハイパーバイザーの高速化

- **約200Gbpsのスループットを実現**
 - ライブマイグレーションにも耐える余裕の帯域を確保
 - 100GbEx2 NIC / OVS-DPDK 構成
- **詳しくはCloud Operators Days Tokyo 2026で発表予定**



200GbE 時代に再考する
ハイパーバイザネットワーク設計

Akira KAMIO

ネットワークコストの最適化

- **スイッチ構成の統一・保守レベルの最適化**

- スイッチ構成を台数の多いLeafの構成に統一
- ネットワークラックも合わせて背面マウントに変更(L4LB以外)
- 検証用機器の削減・在庫融通がしやすい

- **検証環境の仮想化**

- 以前は本番と同じ構成・台数で検証環境を作っていたが仮想化を実施
- 機器選定の要件として仮想版OSが必須に

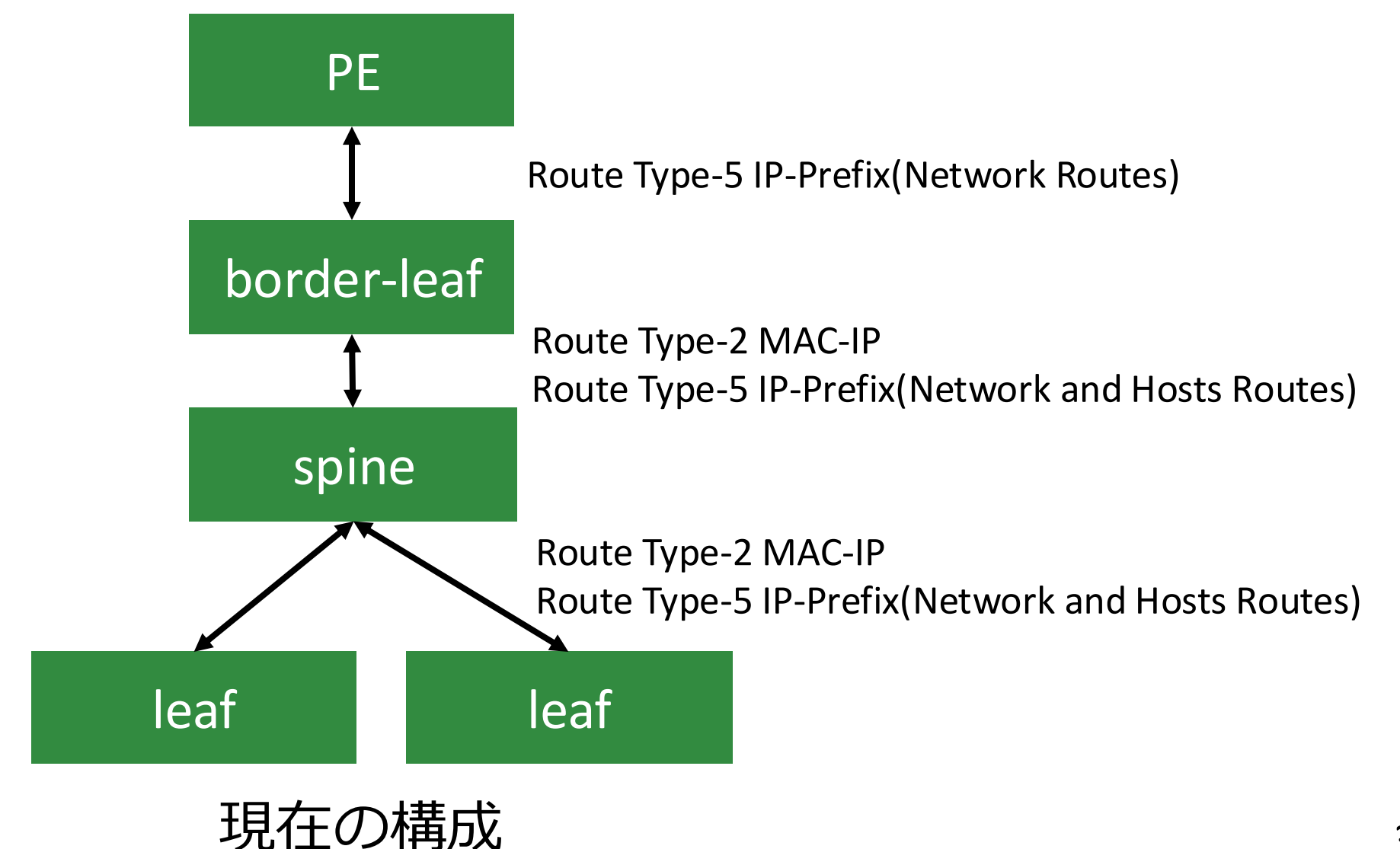
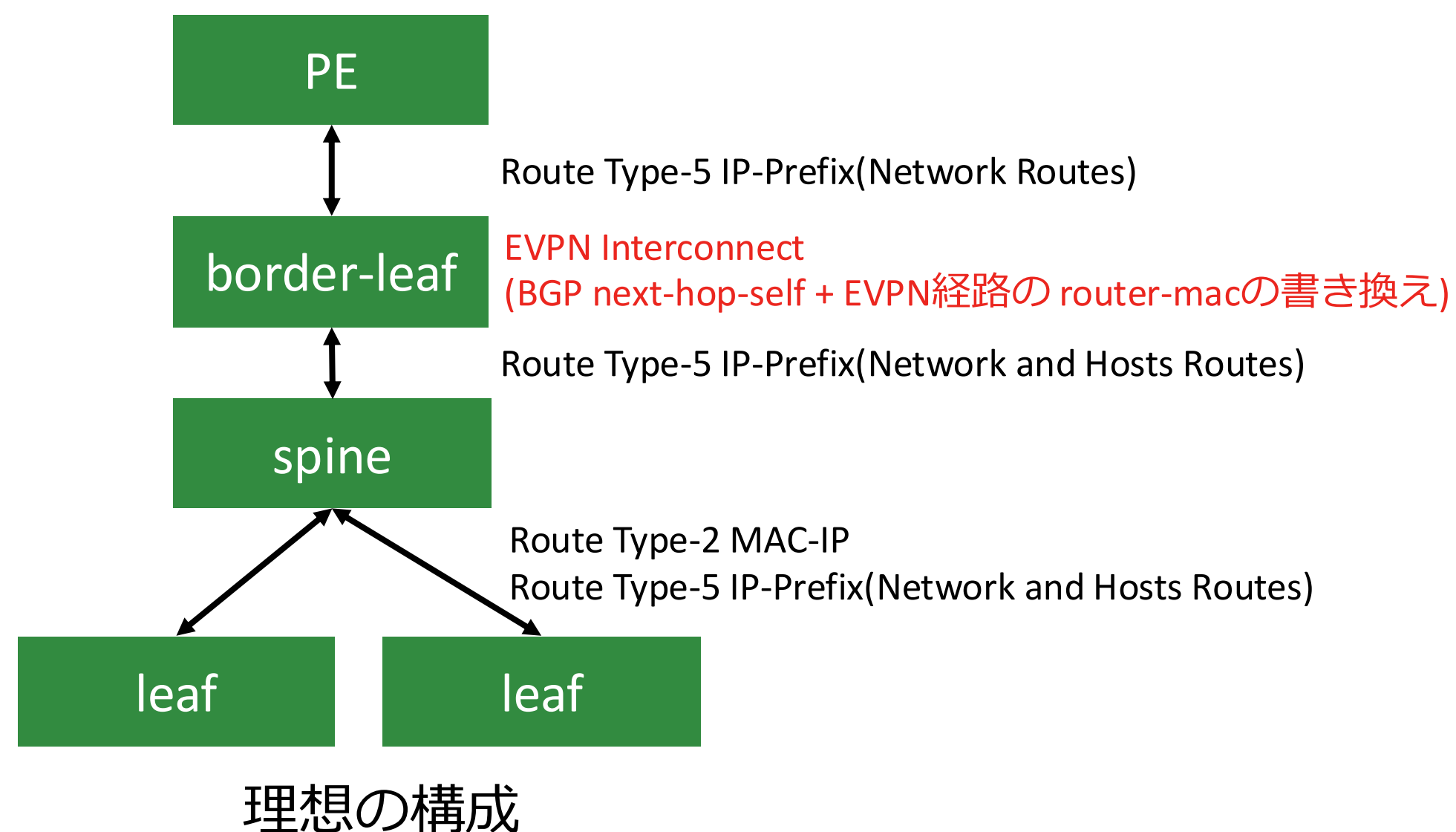
- **3rd Party Optics の採用**

- 純正比で圧倒的コスパを実現
 - 400G-ZR+(後述)は予備パーツを加味すると高かったなので純正を選択
- 保守の切り分け用として純正トランシーバーも数本購入

3. 苦勞したこと・まとめ

苦労したこと: EVPN Interconnect

- EVPN Interconnect機能を使いborder-leafにleafのType5経路を集約する構成
 - border-leaf配下に複数のL2 Domainを作成しやすい構成
- 経路切替が失敗する事象が発生し、リリースまでに原因追求が間に合わせず諦め
 - ASICへの書き込みがうまくいってなさそうな挙動でFIB/RIBレベルでは正しそう
- EVPN/VXLAN 環境ではどこでパケットが消失しているのか切り分けが難しい



その他の苦労したこと

- **L4 Load BalancerのL3DSR構成での負荷試験が難しい**
 - 負荷をかけるマシンでIPIP/GREなどを解く必要がある
 - 今回はdperf <https://dperf.org/> を使用して負荷試験を実施
 - L3DSR構成の厳密なパフォーマンス測定は未実施
- **EVPN/VXLAN環境の負荷試験**
 - 仮想環境上にVM/VPC/Subnetを大量に作成して負荷試験を実施
 - 大量のコンピュータリソースが必要なのでいい方法があったら教えていただきたい
- **N+1 構成の運用が大変**
 - 標準機能ではないのでWebGUI/CLIを見ただけではどれがActiveかわからない
 - コントローラーとBGPの経路状態を見る必要がある
 - 再起動時に勝手に組み込まれないようにAS-PATH Prependするスクリプトなどを作成
- **移行作業・構築作業が大変**
 - バックボーンネットワークの移行作業と新リージョンの構築を約3人で同時実施🔥

まとめ

Cycloud新リージョンでは、「数年ごとに作り直す基盤」から「継続的に更新できる基盤」へ転換

1. ネットワーク構成を刷新
2. ネットワーク機能をAPI / コントローラー化
3. 性能とユーザー体験を改善

プライベートクラウドを“作り直す”のではなく、“継承し続けられる”形へ

所感

- **EVPN Route Type-5が便利**
 - 別拠点も同じような構成に変え始めている
 - マルチベンダーでもちゃんと運用できるのか気になる
- **一般的な100G 32ポートスイッチで動くような100G-ZR 製品が欲しい**
 - 小規模拠点に400Gは不要である
 - 100Gくらいの帯域で使えれば波長貸し・ダークの活用も進みそう
- **生成AIによりネットワークエンジニアの活躍の幅が広がっていることを実感**
 - ネットワークコントローラーなど必要なものを自分たちで開発出来る

今後の展望

- EVPN/VXLANのマルチベンダー検討
- NAT GatewayのVPC・コントローラー対応
- ハイパーバイザー側での仮想ネットワーク対応
- 透過型DSRの実現
- 自作L4LB
- 自作CDN
- GPU基盤のマルチテナント対応

議論ポイント・会場に聞きたいこと

■ テクニカル

- EVPN/VXLANファブリックについて
 - EVPN Route Type 5 についての知見・苦労話
 - マルチベンダーの事例
 - 負荷試験はどうしていますか(VRF/Vlan/MAC-IP等)
 - ファブリックの監視はどうしていますか
- 拠点間接続での波長貸サービス / 400G-ZR 導入についての知見
- LBアーキテクチャについて
 - N+1構成についてどう思いますか
 - 透過DSRを実例があれば知見の共有
- マルチテナント環境における Firewall の在り方をどう考えるか
- NW刷新に伴う検証環境はどのように準備していますか

■ 組織・運用

- ネットワーク以外のチーム（サーバー基盤・アプリケーション等）との連携について
 - 責任分界点の決め方（例: MetalLB等サーバーサイドでBGPを扱う場合の棲み分け）
- プライベートクラウドとパブリッククラウドの棲み分けについてどう考えますか
- プライベートクラウドやネットワーク刷新を経験した方の苦労話・知見の共有
- NWチームにおけるCoding Agent / AIの活用について

FIN

(参考)OpenStackのライフサイクル

OpenStackの更新は難しい

⇒ マシンタイプのライフサイクルに合わせてOpenStackを乗り換える

⇒ ユーザーにはマシンタイプを変更して再起動しているだけに見せる

