# MIXIの
# マルチテナント
# ハイブリッドインフラ

2023/08/30

株式会社MIXI　吉野純平

# 自己紹介

- 吉野純平
- 株式会社MIXI 執行役員 CTO
- 2008年新卒
- ネットワーク、物理インフラ、アプリケーション運用、ミドルウエア運用、ハードウエア開発などなどやってきました。

# 既出のサマリー

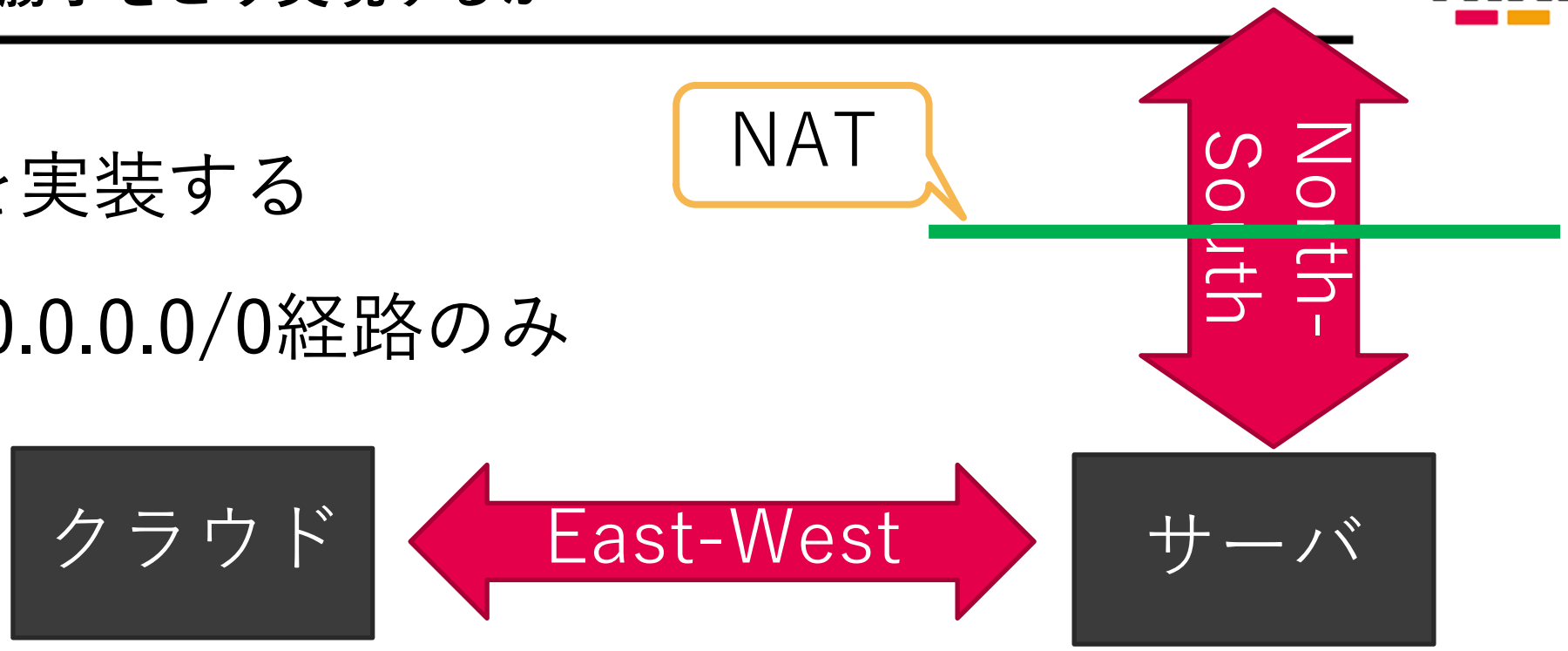
- 2014年からサーバのL3化以外はコンセプト変わらず
- 新規は基本パブリッククラウドで作る
- パブリッククラウドの使い勝手と似せられるところは似せる
- マルチクラウドでの通信を意識（8社と物理接続経験あり、ほぼ商用利用）
- サーバでFRR動かす。bgp unnumbered + PXE等。GSHUT万歳
- MPLSベースのマルチテナントL3ネットワーク(Not SR)
- オンプレでは、VMは使わずベアメタルサーバで利用
- LBは、L4でSNAT + proxy protocol（今日は新しいやつの話をしたい）
- 映像系もあるよ！（大量のマルチキャスト通信、1usec精度の時刻同期）

# 着想のきっかけと勝てると思ったポイント

- 着想のきっかけはTungsten Fabric（旧OpenContrail）
- マルチテナントをMPLSでL3VPNにて実装
- 汎用ASICのスイッチで作ったので安上がり
- キャリアさんが使ってそうな道は枯れているに違いない
- 一旦テナント数を10とした

# クラウドと同じ使い勝手をどう実現するか

- VPCとEIPを実装する
- サーバには0.0.0.0/0経路のみ

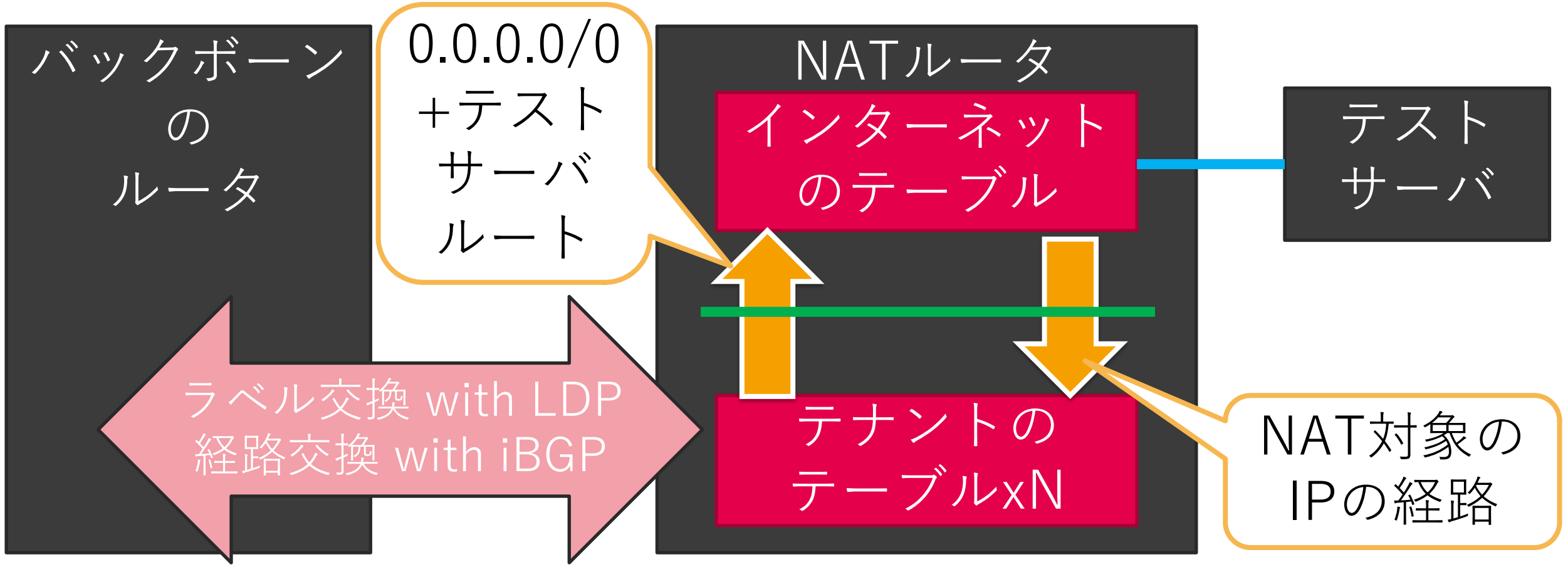



- サーバにはprivateアドレスのみ
- インターネットに行くときはEIPに変換される

# クラウド接続

- VRF単位でのBGP接続が基本
- クラウドごとのお作法をエッジに実装
- Graceful Restartが必須なケースもある
- GRE終端機能を活用したこともある
  - https://speakerdeck.com/isaoshimizu/monster-strike-x-ibm-cloud
- 経路制御コミュニティがある場合もある

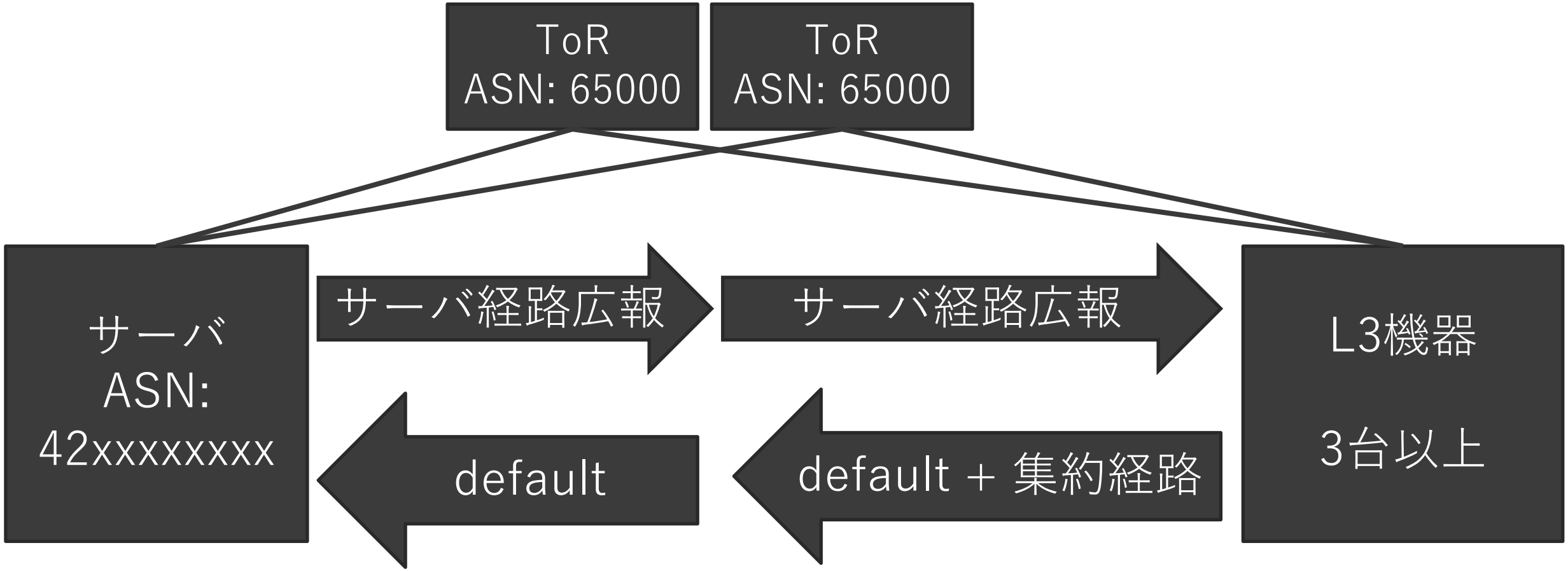
# ANYCASTなNATルータを網に複数設置




. 全ルールが正しく動くかチェックしてから戻せる

. ロスがない＝＞停止含む作業は24/365で実施できる

# ToRの構成（よくあるやつ）

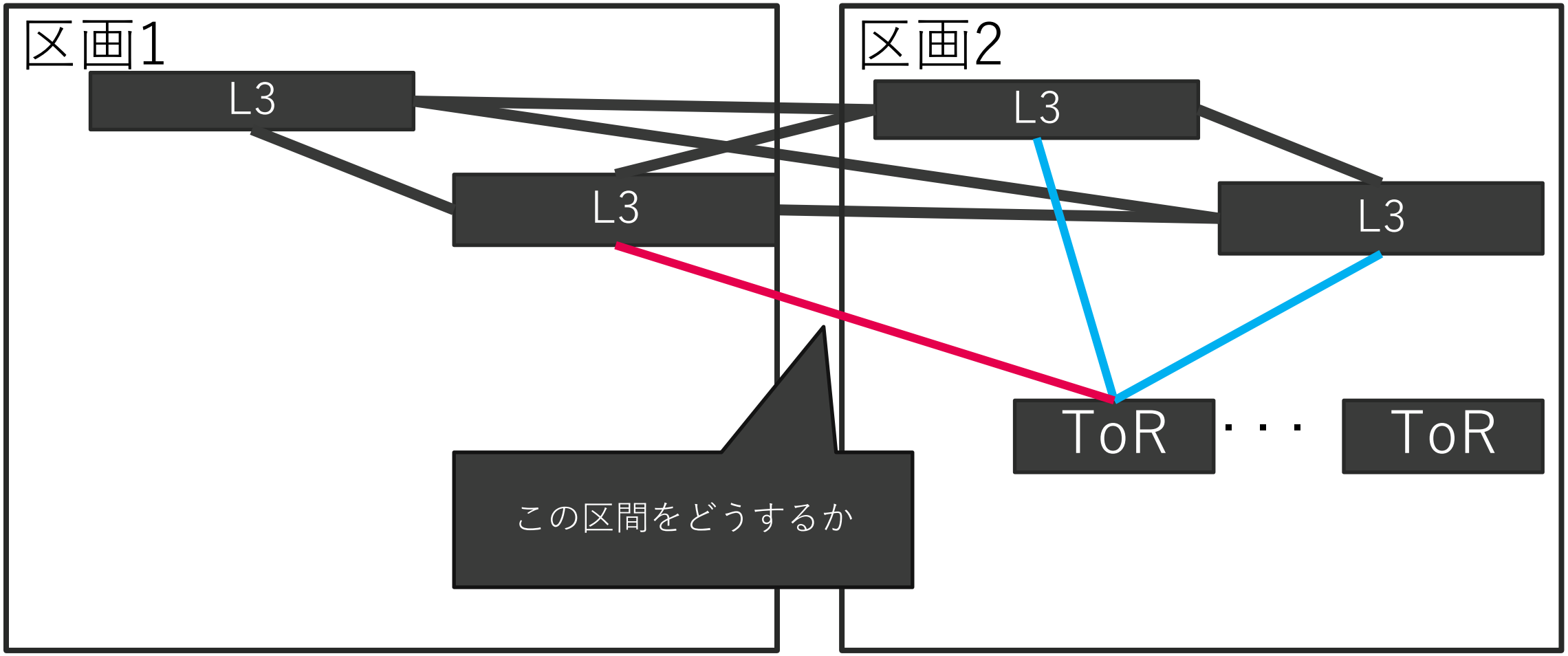



- よくありがちなL3構成
- defaultはAnycastのNATのものを使用

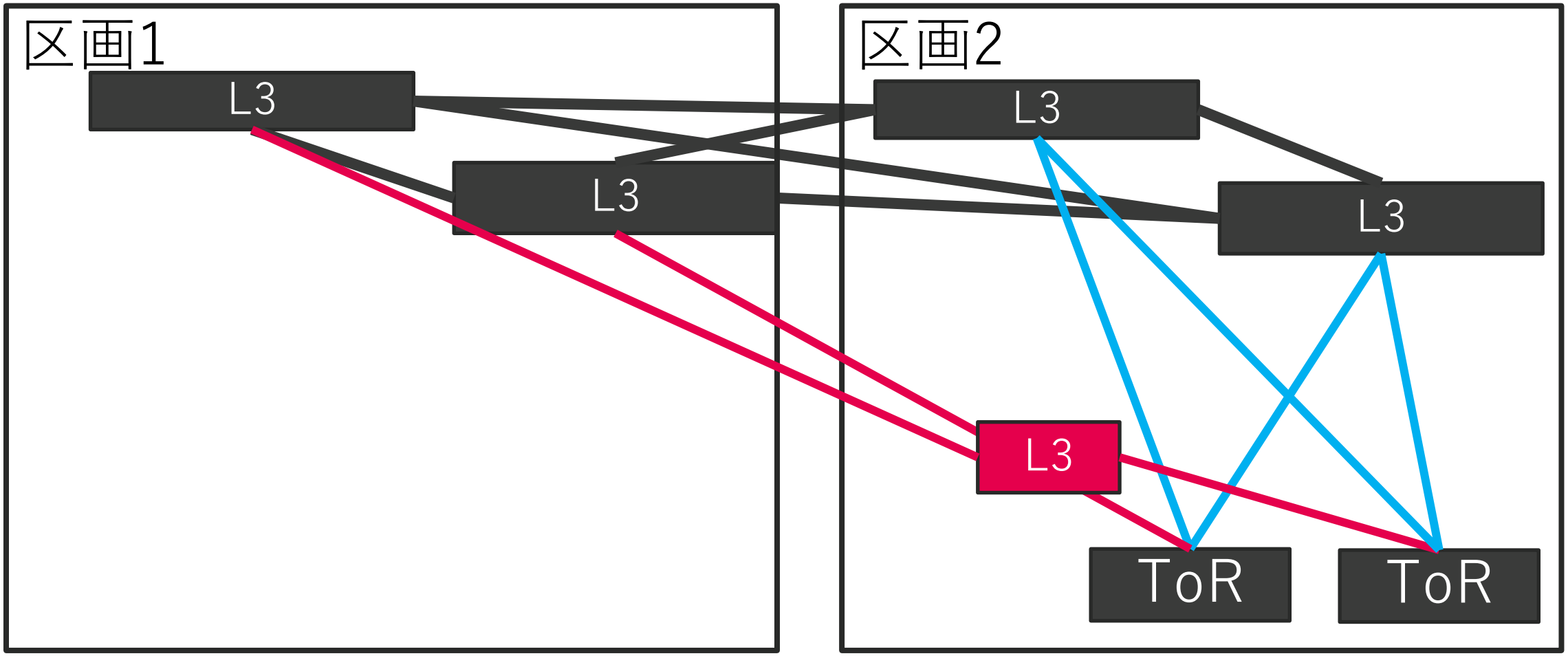
# ダイバーシティと運用負荷軽減

- 全ての部屋のL3は2社以上で構成
- 2社の機材でチップも違う（今は）
- バージョンもずらす（が最近は効果薄いかも）
- １つのメーカーがルーティングのバグで全滅しても半身はちゃんとトポロジ的に疎通維持できるようにする
- ToRはそこまでできてない
- L3機器の数を以下に減らすかが戦い

# だがしかし、N+2をやりたい



# 案1: 要件絞ったL3を仕方なく増やす




結局L3増えてるジャーンっていうのはある。
帯域とか収容の効率は良い。

# 案２：OTN/Muxponderを使う

- Linkdown転送される回線を拠点跨ぎで作る
  - MPLSベースではなくMuxponderでそれを実現
  - OTUC2を100G/40G/10G等に分割して使用
  - OTU4を10Gに分割して使用
- スイッチASICタイプのホワイトボックス伝送では不可
- 時刻同期系プロトコル等もついでに運べる点は嬉しい

# LB:クラウド接続するとDSR構成が使いづらい

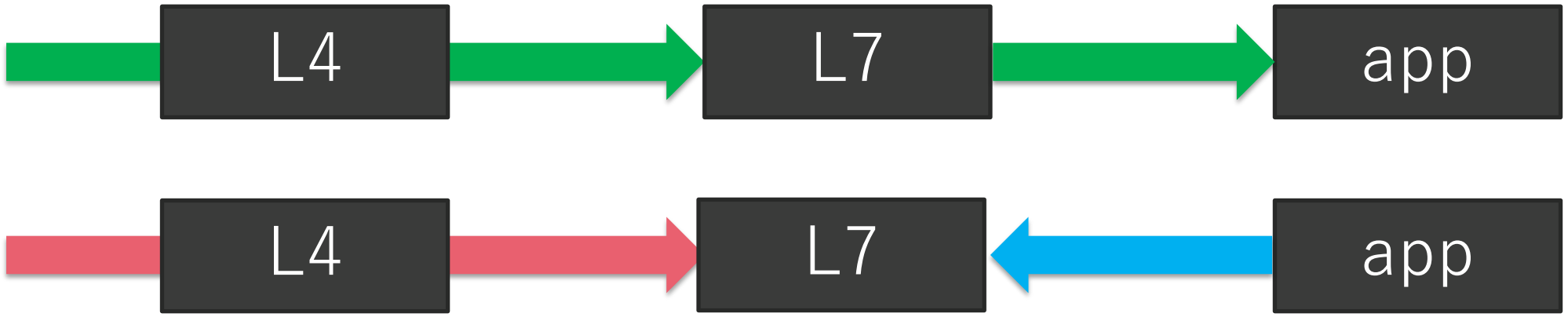
- クラウド跨ぎでDSRはできない
- 他のASのIPをソースとしたトラフィックをインターネットには送れない
- proxy protocolを使ってシンプルにsource nat構成

# L7終端とappの連携の工夫を検討中

- 2019年ごろ検証していた構成
- LBの課題
  - 大量になってくるとLBに追加や抜くのがだるい
    - 増減をツール化や一貫性を維持する仕組みなどなど
  - ウエイト調整も悩ましい

# L7終端とappのプロセスの接続方向性を逆転させる

- Appプロセスから L7終端場所まで接続しにくる
- AppがsocketをconnectしてリクエストをAcceptして待つのではないようにする



# メリット

プロセス数しか負荷がこない。
Backlogが埋まるという概念がない。

受ける一方で、client portの枯渇を踏みづらい


バランシング対象の追加が不要
ヘルスチェックは受け取ったコネクションだけを見れば良い

# デメリットになりそうなこと？

- L7終端点に死んだappのsocketを残したくない
  - 不要なtimeout待ち等をしないようにしたい
  - Appがいなくなっているsocketをうまく閉じる
  - 勝手にいなくなった等を防止するためにtcp keepalive等
- Appを最小限変更でこれにできるか
  - Appの中のサーバ部分を変更したとして、
  - 綺麗にGraceful shutdown等でデプロイできるのか
  - まあやればできそうではある

# まとめ

- MPLSベースのマルチテナントハイブリッドを紹介
- いかにL3機材を減らせるかのチャレンジ
  - OTNも一部利用
- LBも余裕ある時に模索しているのを共有した