

ISPネットワークのモデルベース再現と BGP運用シミュレーション

ビッグローブ株式会社 前野・滝口・前田

伊藤忠テクノソリューションズ株式会社 山口・武藤

TIS株式会社 萩原

NTTコミュニケーションズ株式会社 田島

課題

- NWの運用で "やってみないとわからない" がなくなる
- なぜ?
 - 本番環境と全く同じ機材・規模で「完全な検証環境」を用意するのは困難
 - 検証環境として再現する範囲が変わった時点で100%の再現にはならない
 - 検証環境には、規模と精度のトレードオフがある
 - 結果として、"システム全体の動作" を事前に検証するのが難しい



"やってみないとわからない" 不確実さを減らすための方法は?

アプローチ

- 不確実さを減らすために?
 - 検証環境の規模と精度のトレードオフ
 - ➡ 制約の異なるコンテナベースのNWをつかう："規模"の再現
 - 既存環境の構成情報をモデル化して、同等のネットワークを再現する
- ターゲット = 既存のネットワーク (Brown field)
 - コンフィグと設定シートで管理されているNW



従来は難しかった規模(ノード数)を再現して
"システム全体の動き" を "実際にやって確認" できるようにする

アプローチ

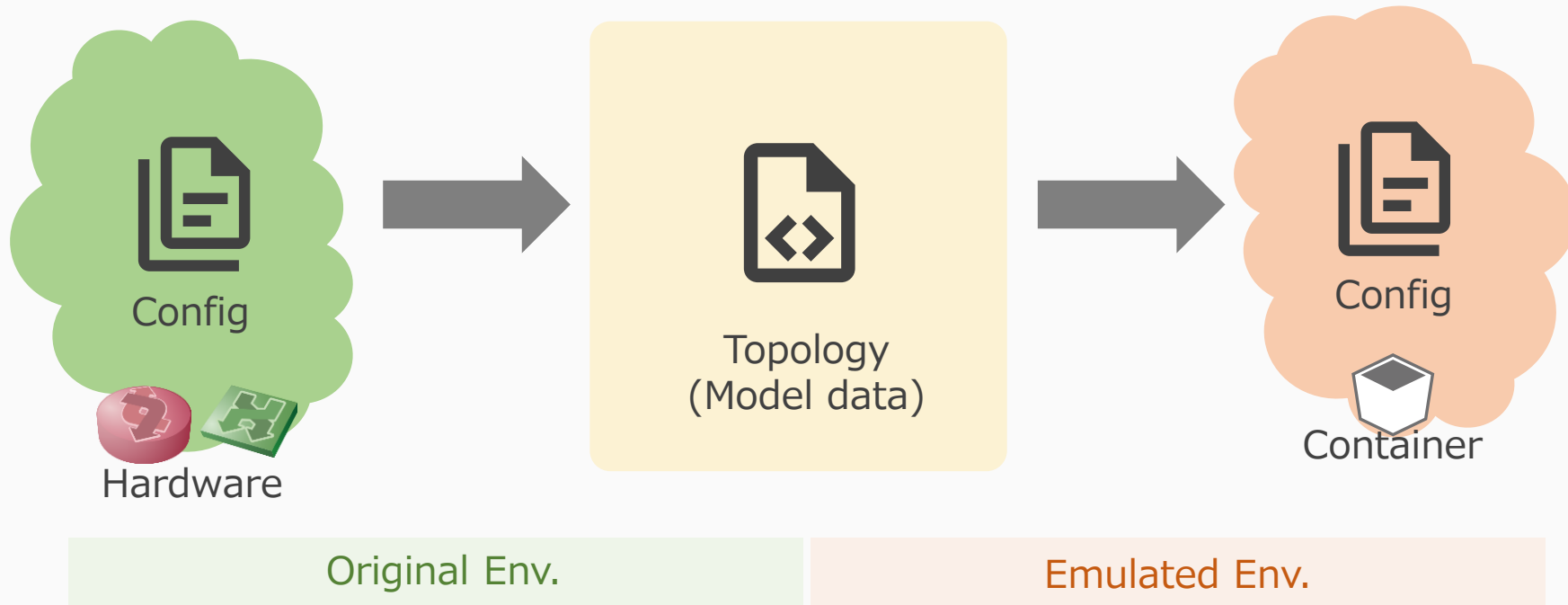
"システム全体の動き"を再現するために、
実環境と同等なNWをコンテナベースで再現する



- 実環境の構成情報を抽出してモデル化する
 - ベンダ非依存
 - マルチレイヤ: 複数レイヤ間の関係性をもつ
 - 各レイヤでのトポロジ + ノード/インタフェースの属性データ
- モデルデータを基にした処理の実装
 - 抽出した構成情報を基にした環境構築の自動化 (検証環境の構築)

同等なNWをコンテナベースで再現

Configから構成情報を抽出・モデル化しコンテナベースの仮想環境で再現



期待される効果・狙い

- 商用同等の規模を素早く仮想環境上に再現できる
 - 運用作業の品質向上
 - システム全体の動きを再現
 - ある時点のNWをスナップショット的に再現してオペレーション可能に
 - 可搬性向上
 - 環境のコピー・クローンが容易
 - 同じ環境を複数台のマシンそれぞれに立てることが可能
- モデルを起点とした運用
 - トポロジー可視化
 - 構成情報をモデル化したことによってできるように
 - モデルによる差分取得
 - ベンダー・装置に非依存で比較

将来像

- 机上レビュー、「やってみないとわからない」 ➡ 実際に「やってみる」運用へ
- Config・台帳ベースの運用 ➡ モデルベースの運用
 - 手書きのNW構成図 ➡ トポロジー可視化による構成管理
- 育成・トレーニングへの活用
 - 商用同等の環境をすぐに構築、壊してもOK



運用業務の品質向上



サービスデプロイまでの
リードタイム短縮



エンジニアリソースの
有効活用

今回の取り組み

- 本システムでラボ環境におけるL1~L3(OSPF)の再現・オペレーションはできた
 - NTTCom検証環境トライアル/セグメント移転シナリオ(FY22)
- 実運用で機能するか未知数 + ラボ検証とのギャップは?
 - 商用環境の規模感・複雑性で動くのか
 - 実運用で何が役に立つのか



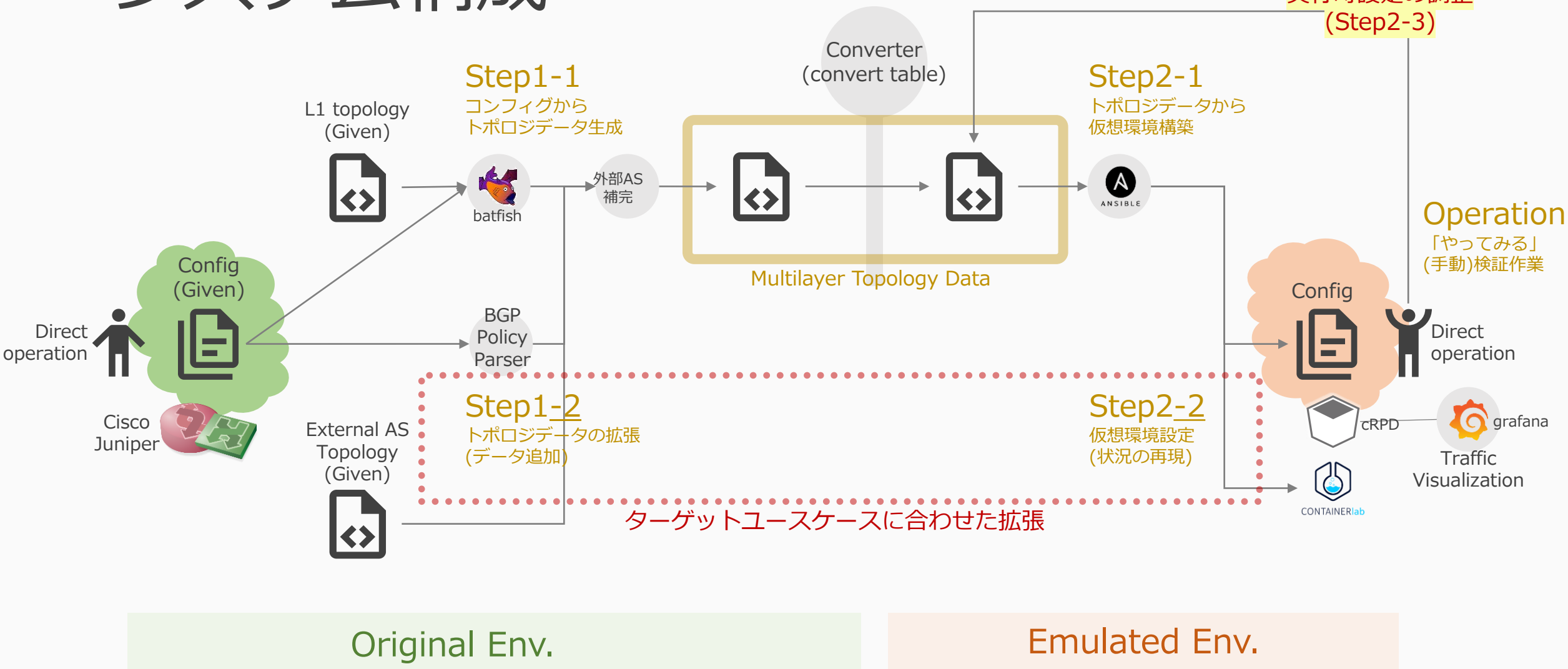
BIGLOBEのバックボーンNWを用いて本システムを評価
実務ベースのユースケースを再現

PJのスタンス

- 自分たちの業務を自分たちで「エンジニアリング」する
 - 各社課題やオペレーションは異なる
 - 自分たちなりにシステム化することで課題を解決していく
- 汎用的なシステムを作ることが目的ではない
(われわれはNW機器メーカーとかじゃないので…)

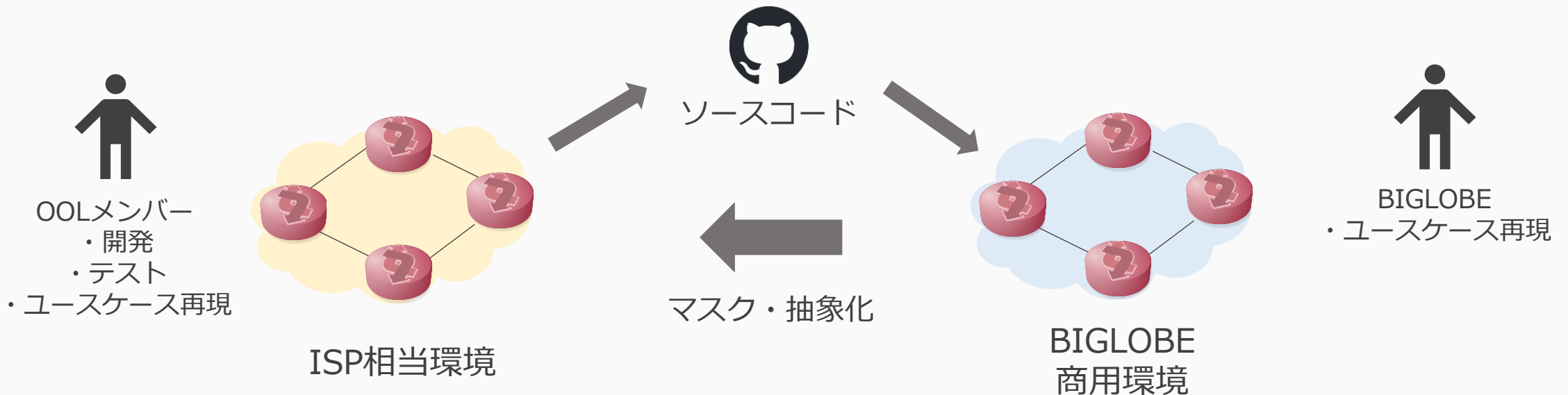
システム構成

実行時設定の調整
(Step2-3)



開発の進め方

- BIGLOBE商用環境でユースケースを再現したいが、商用稼働しているConfigは社外へ公開できない
→ 本番Configを元にISP相当環境を用意して開発を進めた



BIGLOBE トライアル

BIGLOBE ISPバックボーン運用の課題

- 少人数体制(実働メンバ3名)のため環境構築の工数を少しでも削減したい
 - 検証環境を作るのは難しくそれなりの知識が必要
- 全体トポ図が無いためバックボーン全体の動作理解が難しい
 - 構成情報の管理はExcelファイルがメイン
 - 深い理解のためには、Config読み込みや実機での動作確認が必要
- 作業レビューがベテランメンバに集中
 - 設定内容の正当性判断がレビューワーカーの経験に依存
 - 作業レビューワーカーを増やすために、本番に近いシミュレーションができる環境が欲しい



ベテラン以外でも、本番に近い環境を素早く作成して検証できるようにしたい
願わくば本番構成の理解を深める環境としても使いたい

課題解決のためのユースケース設定

- ISP本番環境の再現にあたってBGPの動作は必須
 - これまでのPJ活動でStaticやOSPFの再現はできている
 - BGPの動作に加えポリシー設定の再現までは最低限必要



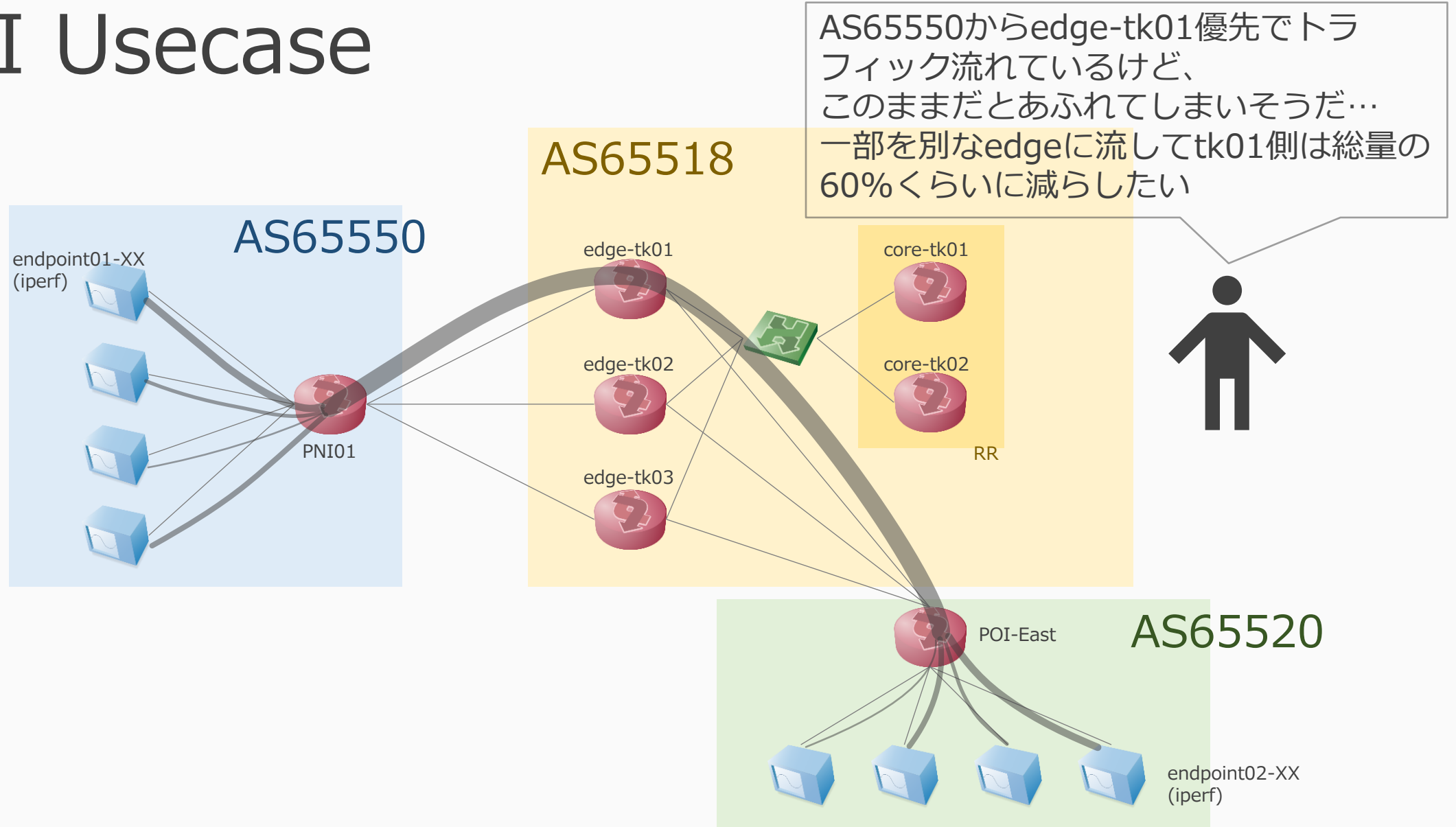
- 多くのISPで発生するBGPを使ったトラフィック制御を再現できれば、あらゆる検証に利用できるのでは？



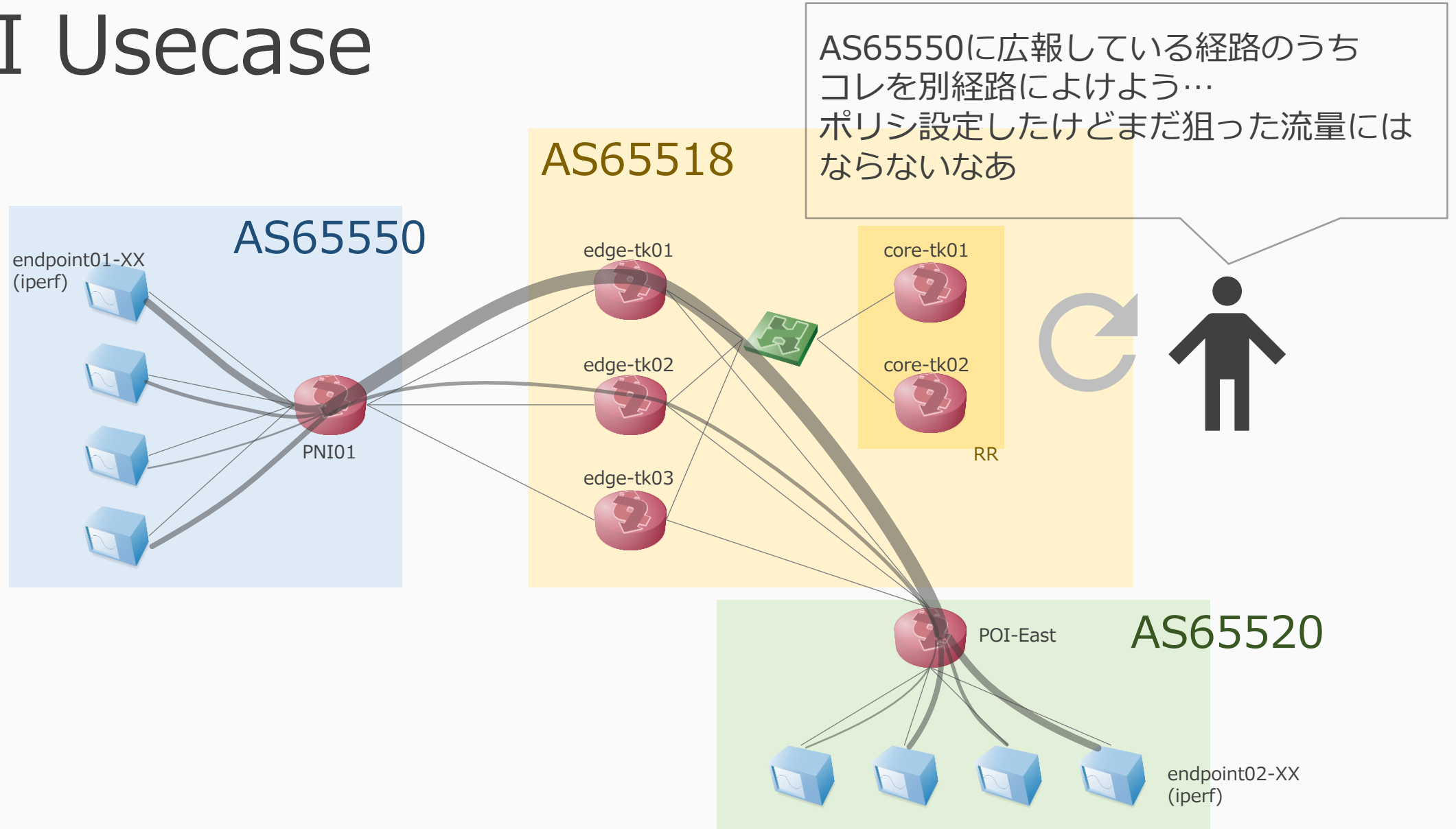
Transit,PNI,IX,CustomerといったBGP接続先によってパラメータ等の違いはあるが、
基本的なポリシーの記法は同じ

シンプルで理解しやすいPNIのトラフィック制御をユースケースとして設定

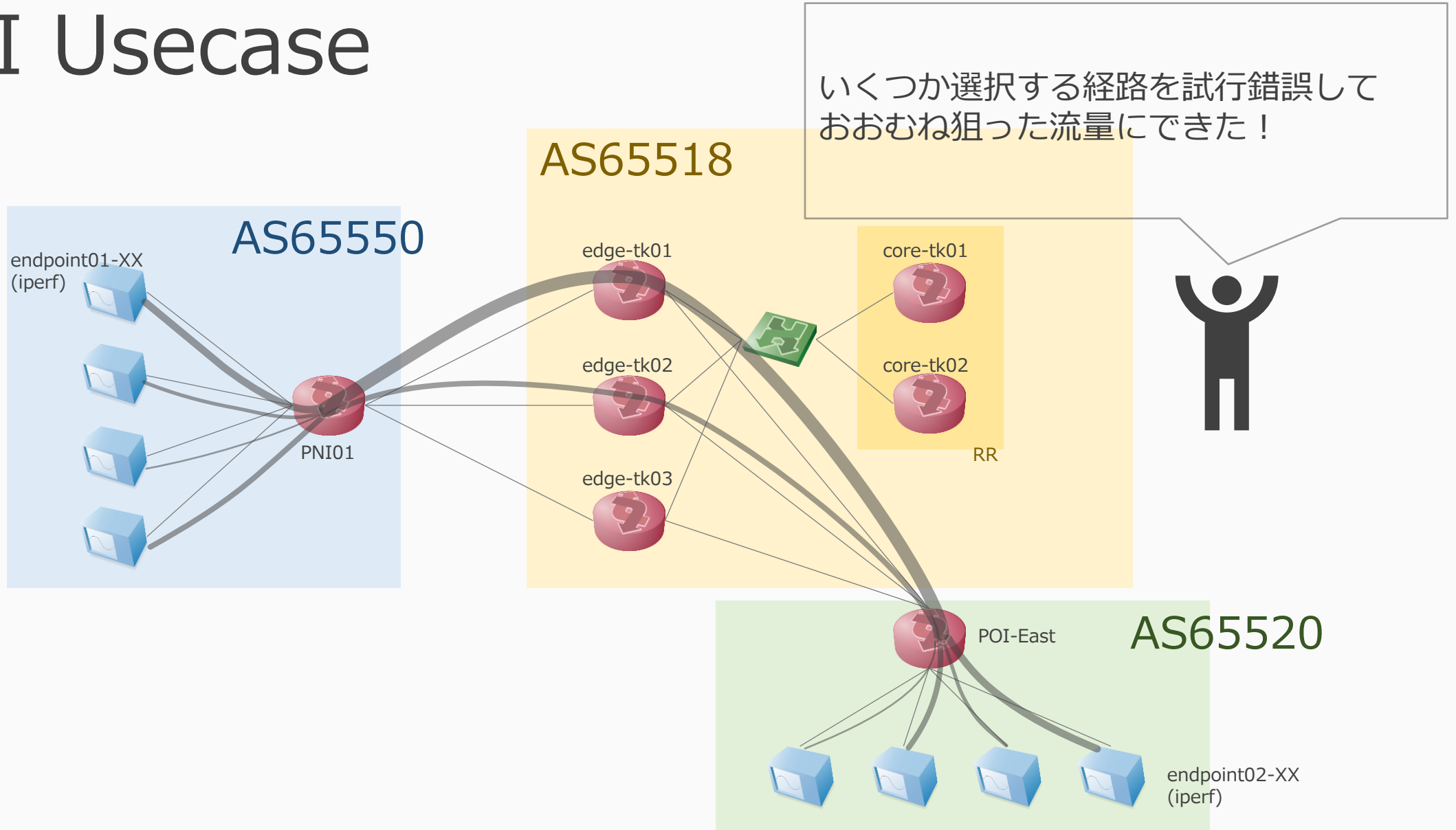
PNI Usecase



PNI Usecase



PNI Usecase

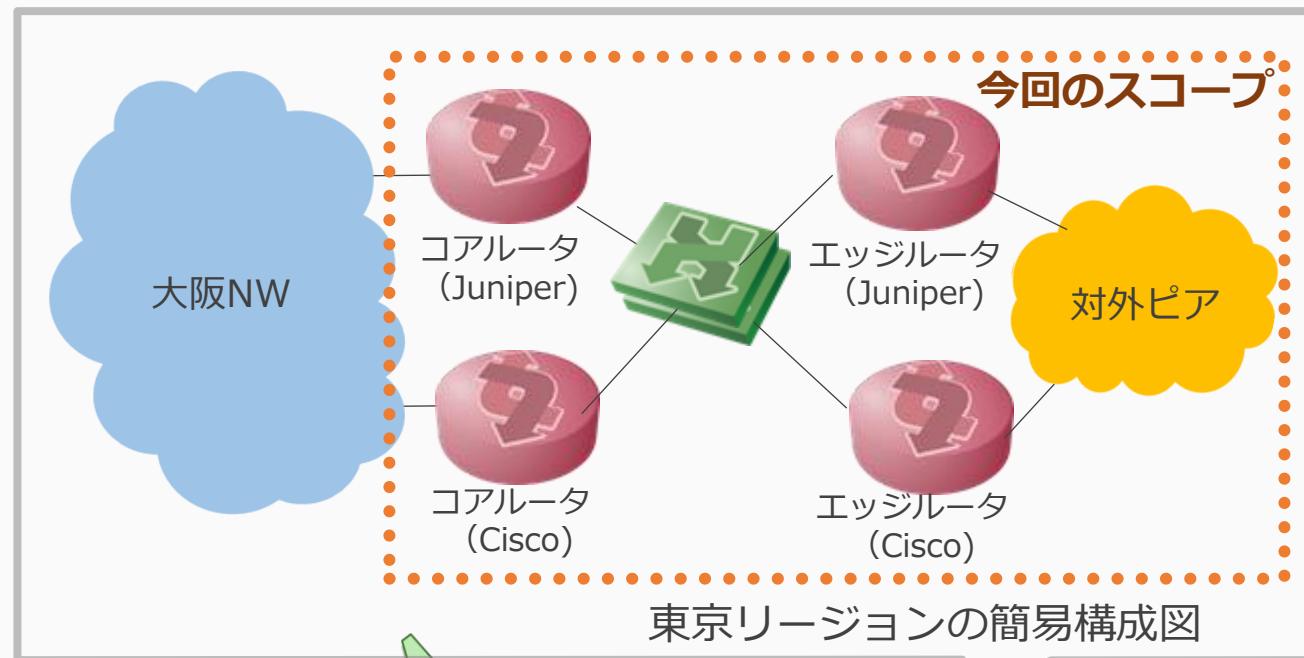


ユースケース再現により期待したこと

- 本番環境と同等の構成で検証できる基盤ができる
 - トラフィックや経路変動のシミュレーション
 - End-to-Endの正常性確認
 - ノード追加/廃止時の動作確認
 - **障害時の影響確認、冗長設計確認**
- 運用改善に繋がる
 - **実Configのモデルデータ化により構成情報の鮮度が保てる**
 - **トポロジ生成により全体構成の把握が容易に**
 - **ベテラン以外でも本番のシミュレーション環境が素早く作成可能**
 - 実Configの逆生成による作業準備とレビュー工数の削減
 - トポロジ生成や評価環境の操作によるチームメンバーの理解度向上

BIGLOBE 本番環境でのトライアル

- BIGLOBE ISPバックボーンの規模感
 - 国内コア拠点：東京、大阪、福岡
 - DCの数：10拠点
 - ネットワーク装置(L2SW/Router)数：50台以上
 - JuniperやCiscoなどのマルチベンダーで構成
 - 総トラフィック：3.2Tbps以上(2023年度6月時点)
 - ピア数：220以上のASと相互接続
- トライアル内容
 - スコープ：
 - 東京リージョンのコアルータ、エッジルータの本番Config
 - 確認内容：
 - 本番Configからモデルデータ/トポロジ生成が可能か
 - モデルデータから構築した検証環境が本番と同等か
 - 実業務/運用にどう活かせるのか



大阪と福岡も概ね同じ構成, Config設計のため、
東京のモデル化ができれば
他の地域にも適用できると判断して検証



トライアルでの期待したことに対する成果

- 本番環境と同等の構成で検証できる基盤ができたか
 - ➡ 追加実装によりPNIのトラフィック制御ユースケースをシミュレーション環境で再現成功(BGPへの対応が去年との差分)
 - ➡ 実際のポリシーとPrefixを利用してトラフィック制御ができることを確認
- 運用改善に繋がったのか
 - ① 実Configのモデルデータ化により構成情報の鮮度が保てる
 - ② トポロジ生成により全体構成の把握が容易に
 - ➡ 最新のconfigからのモデルデータ生成とトポロジ生成は実現
 - ③ ベテラン以外でも本番のシミュレーション環境が素早く作成可能
 - ➡ シミュレーション環境を使って以下を評価中(Janog53で報告予定)
 - ・ ベテランメンバ以外でも過去のPNI移設をシミュレーションできること
 - ・ ベテランメンバ以外でも作業内容の正当性が確認できること

まとめ

狙いに対する成果

- 商用環境の複雑性・規模感に対して動くか
➔ 商用環境同等のNWを仮想環境に再現できた
 - コンテナベースのため分単位で構築可能
 - BGPレイヤーのモデル化・仮想環境への再現
- 何が実運用で役に立ちそうか
 - 再現環境の構築、シミュレーション
 - モデルを元としたトポロジー可視化



「やってみないとわからない」から「やってみてわかった」へ

苦勞したこと・課題

- 協同開発の進め方
 - ISP相当環境で開発 ➡ 本番環境でテストのサイクルを素早く回していく必要がある
 - 実際に使われている機能を全て把握するのは難しい
 - **トライ&エラーをどれだけ素早く回せるか**
- **本番環境の複雑性・分量の多さ・パターンの多さ**
 - OSの違いによるもの
 - 運用者が定義するデータや構成によるもの
- Batfishが対応できない部分 ➡ スクラッチ開発
 - BGP Policy parse
 - BGP confederation
- ベンダ固有の表現方法
 - Policy表現方法の違い(IOS-XR/JUNOS)
 - 対応したいOSが増えるほどコストもかかる

発展

- 実環境と仮想環境のトポロジーモデル差分取得・逆変換
 - 仮想環境のConfigからトポロジーモデルへ変換・実環境との差分取得
 - 差分から実環境へConfig生成・フィードバック
- より複雑なユースケースへの対応
 - IXユースケースへの拡張
 - FW/LB等のコントロール(L4)
 - 他OSへの対応
- 対応プロトコルの拡張
 - IPv6