

BIGLOBE AS2518を まるごと仮想環境へ“コピー”してみた

萩原 学

TIS株式会社

田島 照久

NTTコミュニケーションズ
株式会社

滝口 敏行

ビッグロブ株式会社

武藤 匠汰

伊藤忠テクノソリューションズ
株式会社

沖縄オープンラボトリ

2024/01/19 JANOG53

登壇者紹介

- 萩原 学 TIS株式会社
- 田島 照久 NTTコミュニケーションズ株式会社
- 滝口 敏行 ビッグロース株式会社
- 武藤 匠汰 伊藤忠テクノソリューションズ株式会社



@沖縄オープンラボラトリ

Model Driven Network DevOps プロジェクト

<https://www.okinawaopenlabs.org/mdnd>

このセッションの流れ

- この取り組みのモチベーション 萩原 (5分)
- AS2518での"コピー"とオペレーション評価 滝口 (20分)
- 過程で発生した技術的な課題と対策 武藤 (10分)
- "コピー"の一步先へ進めるには 田島 (10分)
- まとめ・議論したいこと

- QA/議論 (15分)

このセッションでやりたいこと

- ネットワーク「全体がどう動くのか」を検証したい…が、「全体の構造」を再現するだけのリソースはまずない。
- 昨年度(JANOG51)、NW機器コンフィグを起点に、本番環境NWを"コピー"して「本番同等の規模・動作」を再現させるしくみとその運用プロセスについて検証してみた。
 - 「やってみないとわからない」を実際にやってみる



- 実際に商用NWを "コピー" できるか? … BIGLOBE/AS2518での挑戦
- なにをしたのか・どんなことがわかったのか・今後の運用プロセスをどう考えるか

この取り組みの モチベーション

この取り組みのモチベーション

ねらうもの

- 既存のネットワーク(Brown Field)
- 個別のルール・運用・例外などが入る、自動化の難しいシステム



ねらわないもの

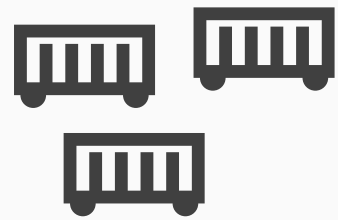
- 新規基盤で新しい技術を導入する
- 一様な構成ルールでフラットに展開される大規模システムとその自動化

運用者の手で「エンジニアリング」する

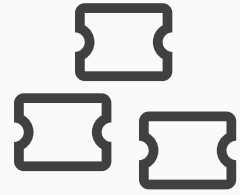
ネットワークのモデル(構成情報)中心なオペレーション

- モデルを介することで人の理解や判断を助けるシステム
- 計算パワー・アルゴリズム・ベストプラクティスの導入と蓄積
- 自分たちで自分たちのやり方を継続的に開発・改善する

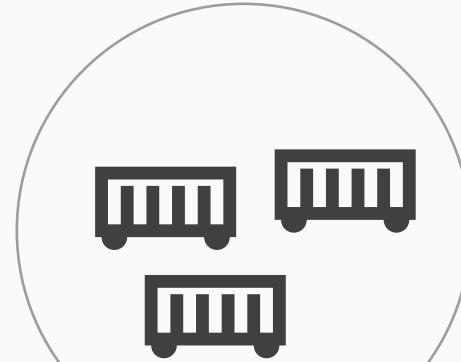
エンジニアリング(例)



製品在庫



注文伝票(紙)



在庫情報I/O
(実世界との
インターフェース)

業務の流れを再現できる
情報全体

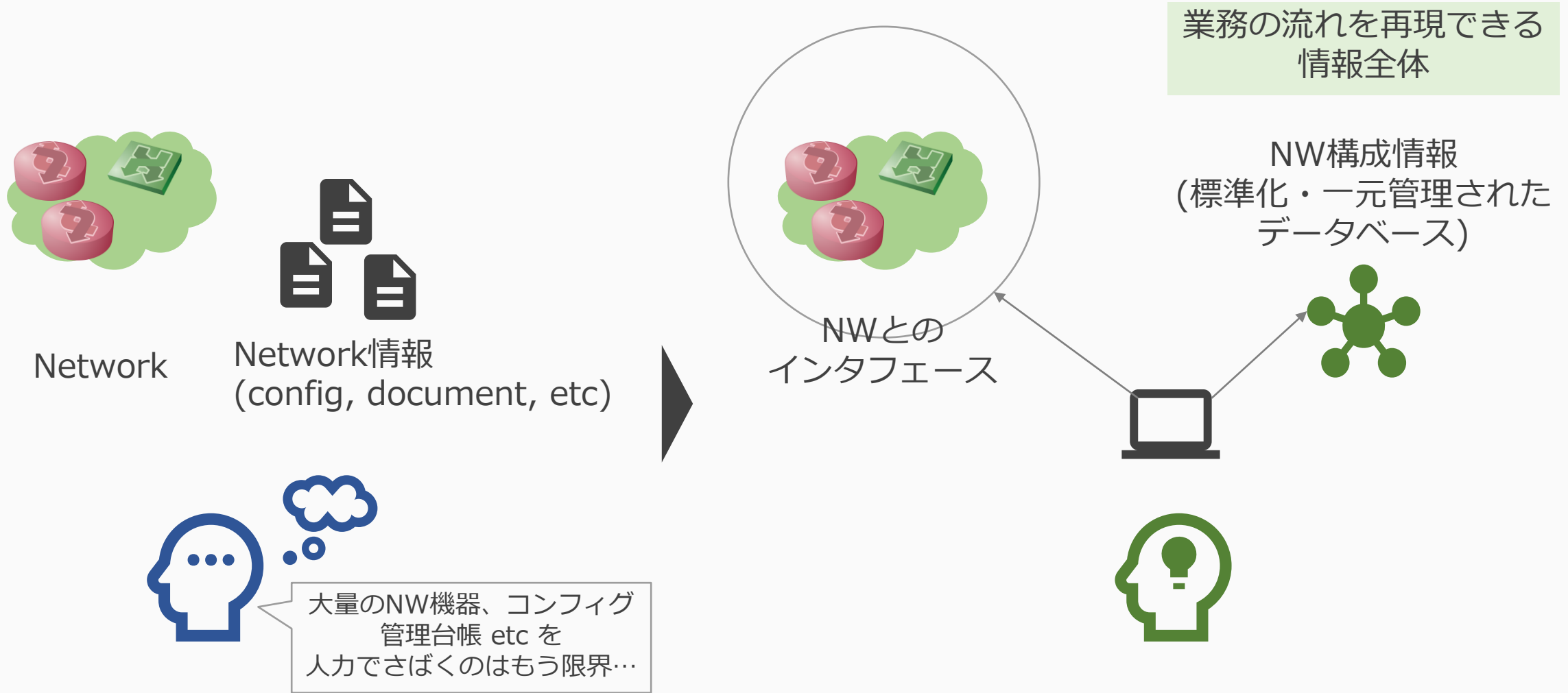
注文・在庫データ
(標準化・一元管理された
データベース)



大量の伝票と製品在庫を
人力でさばくのはもう限界…



エンジニアリング(NW運用)



モデル中心アプローチのねらい

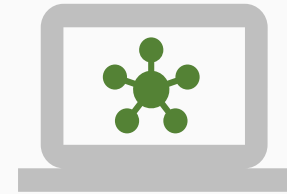


あらゆる情報処理、判断を人が行う
人がボトルネックになる

人が扱える情報量には限界がある
人はスケールしない
人には揺らぎがある

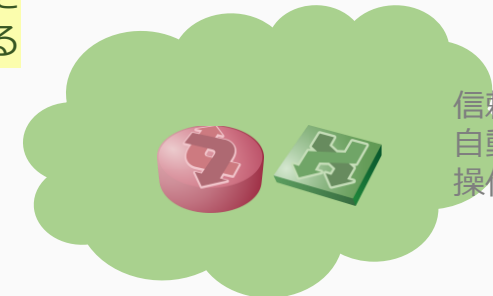
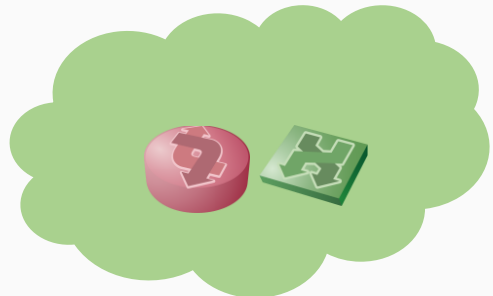


情報処理をシステムで支援する
人は重要な判断や操作にシフトする



モデルと
ロジック

情報量や複雑さに
マシンパワーとロジック(アルゴリズム)で対抗する



信頼度が上がれば
自動化レベルを上げて
操作も機械でできるかも

フェーズ

現行運用

- 手書き(人用)構成図・ドキュメント・台帳等を基にした人力運用
- 構成変更は実機(実環境)直接操作

モデル「育成」期

- 現行運用に対する判断材料の提供・支援等、情報参照での利用
- 構成変更は従来運用を維持
- ➡ モデルの精度・モデルを使ったオペレーションの信頼性向上
 - モデルを信頼できるようにする
 - モデルを「育てる」



ターゲット

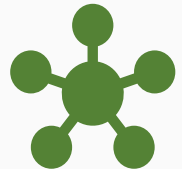
ユースケースのひとつ
…NW動作検証

モデル中心運用

- 構成変更の自動化・効率化へのモデル利用
 - 操作対象をモデルデータに切り替える

育成期ユースケース: NW動作検証

- もっと手軽に「NW全体の動作を」「実際にやって確認」したい!
- そのために
 - 実環境の構成情報をモデル化 → マルチレイヤのトポロジデータ
 - トポロジデータを基に、異なるアーキテクチャのNWとして再構築(再現)
- **従来は検証が難しかったサイズのNWを手軽に再現して、不確実さを低減させたい**
 - 従来は準備が難しいサイズ・時間(短時間)での「砂場」の実現
 - 設定変更時レビューの精度向上・効率化…「やってみる」ことで理解・判断する
 - etc



モデルの作りかた
モデルの活用方法
モデルに対する信頼の確立

議論ポイント

- 実際の商用NW/そこでの運用に適用してみても
 - やれたこと・やれなかったこと(その理由)
 - 苦労したこと・問題点・対応
-
- モデルベースにNWを取り扱うことで、今後の運用プロセスをどう変えられるか?

BIGLOBE AS2518での“コピー” とオペレーション評価

BIGLOBE での評価

別紙参照

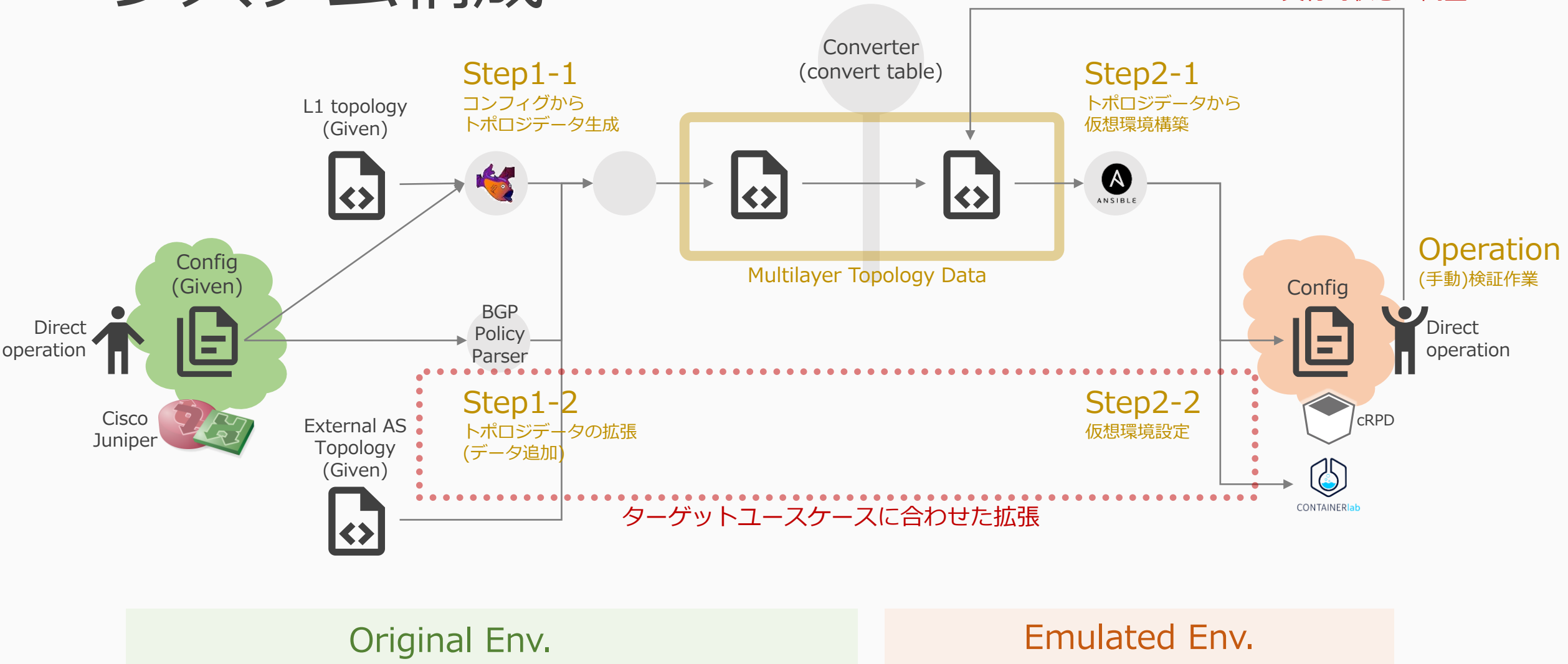
過程で発生した 技術的な課題と対策

このパートの概要

- 技術的なトピックの紹介
 - システム構成
 - 仮想環境を構築するまでの具体的な処理
- 苦労したポイント
 - 技術的に苦労したポイント
 - 開発プロセスの中で苦労したポイント

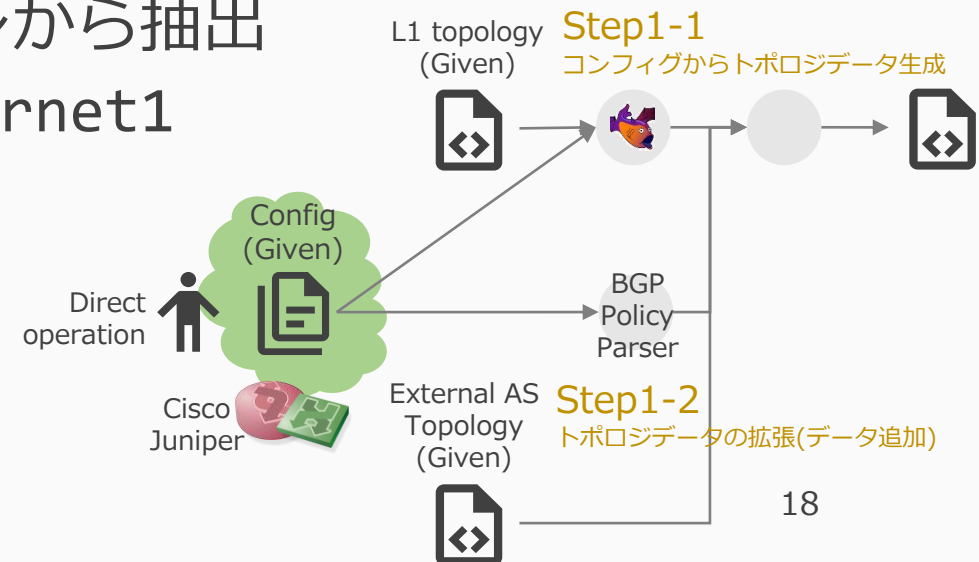
システム構成

実行時状態の調整



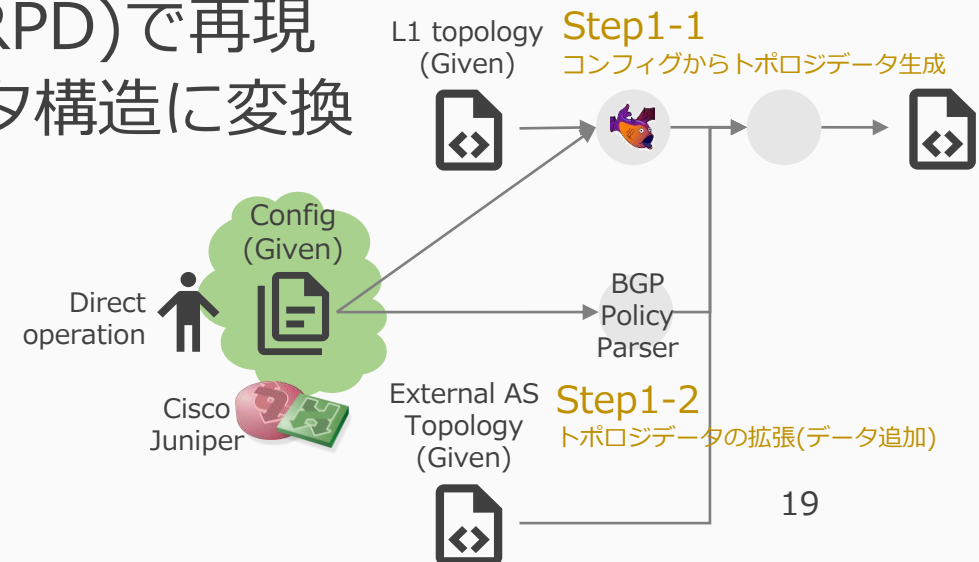
Step1-1 トポロジデータ生成

- 与えられたコンフィグから仮想環境の構築に使用するトポロジデータを生成
- BGPとOSPFの情報はBatfishを通して抽出
 - Batfishが対応していない箇所は独自パッチで対応
- 各機材間のL1接続情報はBatfish+NetBox+スクリプトで抽出
 - インターフェース情報のデスクリプションから抽出
 - デスクリプションの例: Switch-01 Ethernet1



Step1-2 トポロジデータの拡張

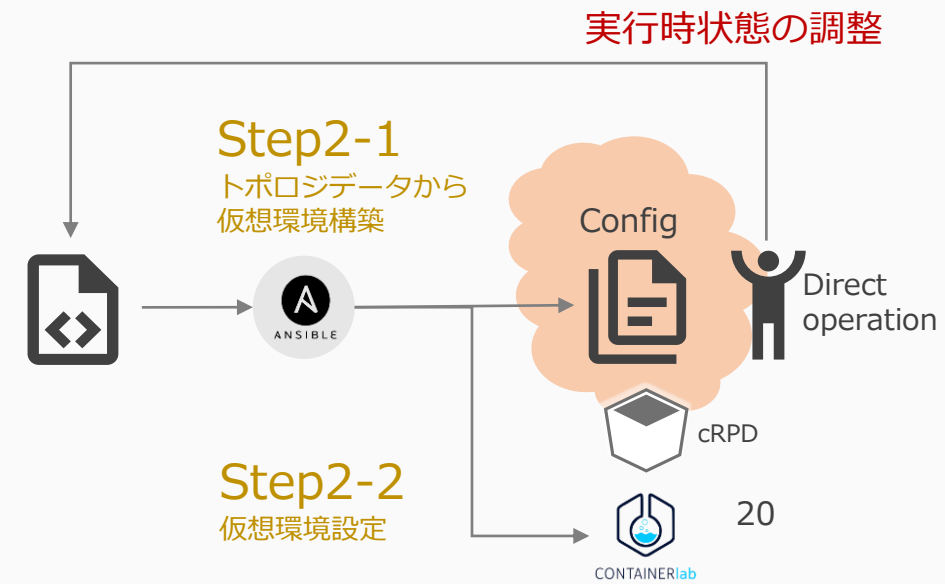
- ユースケースに合わせて必要となるデータを拡張
- その1. 外部ASノードの補完
 - 外部ASのノードも再現したいがコンフィグには情報がない
 - トポロジ拡張用ファイルをコンフィグとは別に与える
- その2. BGPポリシーの共通モデルへの変換
 - 仮想環境ではIOS-XRのノードをJunos(cRPD)で再現
 - JunosとIOS-XRのポリシーを共通のデータ構造に変換



Step2-1

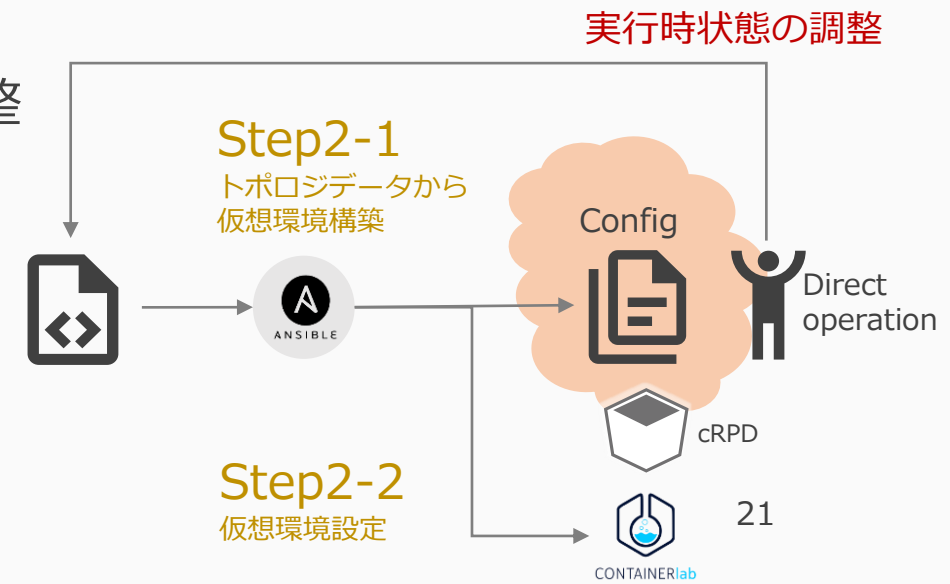
トポロジデータからの仮想環境構築

- ここまで生成したトポロジデータを使用して仮想環境を構築
- 仮想環境の構築にはconatinerlabを使用
 - ネットワーク機器はcRPDで再現
 - L2ドメインはOVSで再現
- トポロジデータからコンフィグを生成して各ノードに投入
 - jinja2を使用してモデルデータから生成



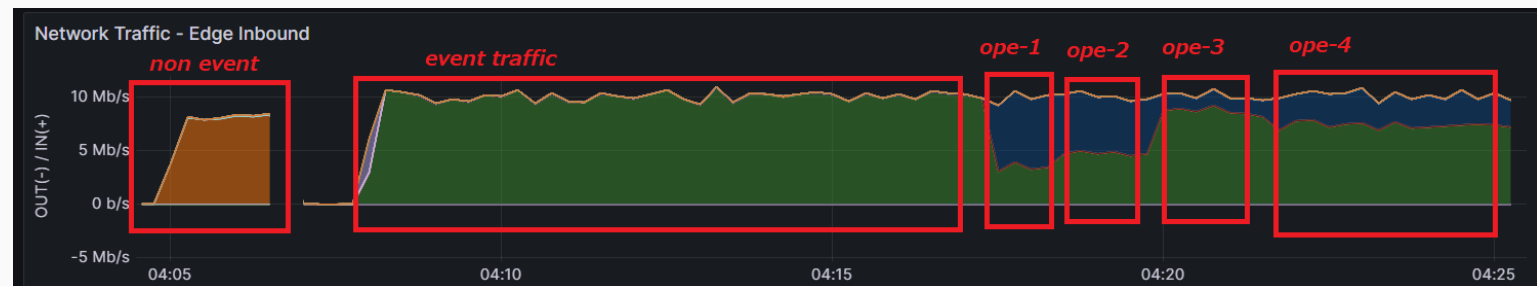
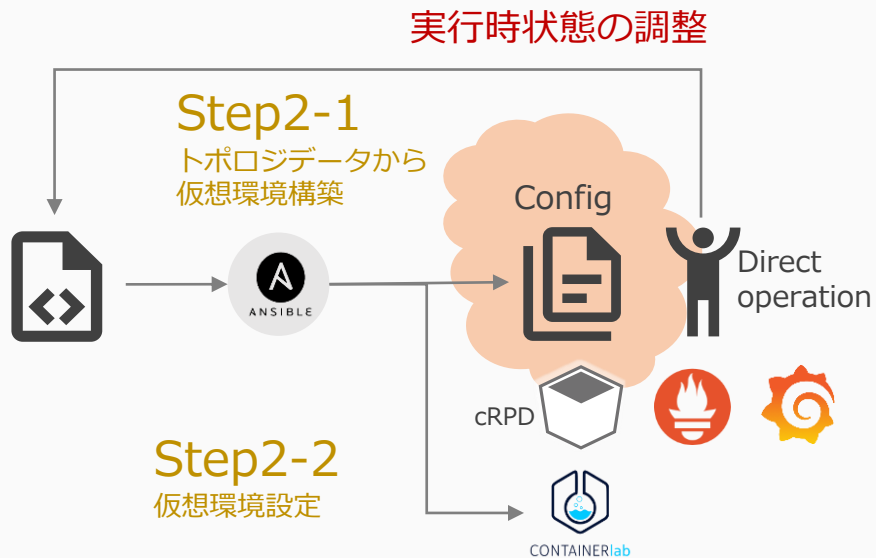
Step2-2 仮想環境設定

- 実施するオペレーションに合わせて仮想環境の設定を変更
- 外部ASから見た優先ピアの変更
 - 自ASのトラフィック受信インターフェースを本番環境と合わせる
 - 内部的には外部ASノードに対してLPを変えたコンフィグを投入
- トラフィックの模擬
 - 外部ASのノードの奥にiperfを実行するノードを用意
 - 流す際にはプリフィックスごとの帯域が実環境(実測したフロー情報)と同等の比率になるように調整



仮想環境での検証作業

- オペレーションがトラフィックに与える影響を一目で判断できるように可視化の仕組みを用意
- 仮想環境のノードは全てコンテナなのでコンテナのメトリクスを取得するcAdvisorとGrafana+Prometheusで可視化



苦勞ポイント 表記ゆれ

- L1トポロジデータはインターフェースのデスクリプションから抽出
 - Switch-01のEthernet1に接続したインターフェースは Switch-01 Ethernet1という様に書かれている (はず)
- 実環境では表記揺れが存在
 - 今回確認できたものだけで8パターン
 - スペース区切り : Switch-01 ge-0/0/1
 - アンダースコア区切り : Switch-01_ge-0/0/1
 - 先頭にtoが付く : to Switch-01 ge-0/0/1 など...
 - インターフェース名の短縮形も都度対応
 - Eth~, Et~ : Ethernet~
 - Ten~ : TenGigE~
 - Te~ : TenGigaEthernet~
- 本番環境コンフィグで確認して修正の繰り返し

苦勞ポイント

BGPポリシーパーサーの自作

- IOS-XRからJunos(cRPD)へのBGPポリシー変換が必要となり、そのためのパーサを探したが満足にパースできるツールが無かった
 - 特定のOSのみ対応、nestされたifは非対応、など
 - Batfishはパースできていそうだがデータ仕様の情報が無く扱いづらい
 - データ仕様が不明 → パターン列挙してデータ確認: 時間がかかる・見落としリスク
 - この調査にも本番コンフィグでの確認で時間を要す
- 既存ツールを拡張するか? 自作するか?
 - 既存を拡張させるのは時間がかかりそうなのでTTPで自作
- TTPも上手くコントロールするにも時間がかかった
 - 正規表現を書かなくてもパースできるので楽に感じるが、デフォルトの挙動を把握していないと予期しない動きをする

苦勞ポイント ポリシー変換の実装

- IOS-XRとJunos間におけるBGPポリシーコンフィグの差分吸収
 - if-else-elseif、nested-if
 - Junosのポリシーはif-endifを連ねた構造になっている
 - ➡ if文ごとにsubroutineを作ってそこで評価する形で対応
 - next-hop-self
 - XRではneighborブロックに設定されるがJunosではthen句で設定
 - ➡ neighborブロックのパーズ結果からthen句に反映する形で対応
- 全てのポリシーを等価に変換するのは難しい
 - 変換しきれないポリシーは今回のユースケースでの要否を考慮して読み飛ばすなどした
 - OSごとに効率的な書き方が存在するが、ポリシー変換を考えるとそれぞれの最大公約数的な書き方がされていないと厳しい

苦勞ポイント 外部ASの再現

- 外部ASも仮想環境で再現したいが、
コンフィグには外部ASの構造は含まれない
➡ 再現に必要な外部ASの情報を別途与える
- 仮想環境で外部ASを再現する際に必要となる構成情報を定義
 - 外部ASに期待する挙動はフラグを与えてコンフィグ生成時に対応
- ユースケースに応じてどこまで再現するか検討が必要
 - 外部AS間とのリンク数、P2MPな環境でのピア(IX)、 etc.
 - 今回のユースケースではPNIとのP2Pリンクを前提とした構成で再現

苦勞ポイント

本番環境コンフィグでのトライ&エラー

- 実コンフィグでの動作確認はBIGLOBEメンバで実施
 - 本番環境のコンフィグは機微情報を含むため外に出せない
- 開発用にデフォルメしたコンフィグを用意
 - 実コンフィグは巨大なためテストがしにくいという理由もあり
- BIGLOBE本番環境のコンフィグを入れてみるとデフォルメでは発生しなかったシステムのバグに遭遇
 - Batfishに保存されるデータが大きすぎて出力が適切に扱えない
 - 描画ツールでトポロジが正しく描画されない、など
- 結局は実コンフィグでのテストは避けられないため、これを使用したトライ&エラーのオーバーヘッドが大きい
 - テストデータ(デフォルメ): テストのしやすさ + 網羅性の両立が難しい

技術面のふりかえり

- ネットワークの知識もソフトウェアエンジニアリングの知識も必要
- 環境固有の知識も必要
 - 今回の場合はBIGLOBEバックボーンNWとその運用
 - 表記揺れへの対応、実環境との差分比較、運用上やりたいこととその判断ポイント・必要な情報…など
- 今回のシステムをall-in-oneなパッケージとして実現するのは難しい
 - どうしても環境固有の対応が求められる
 - 「これを入れればOK」というものではない、というかできない

このセッションの流れ（再掲）

- この取り組みのモチベーション 萩原（5分）
- AS2518での"コピー"とオペレーション評価 滝口（20分）
- 過程で発生した技術的な課題と対策 武藤（10分）
- "コピー"の一步先へ進めるには 田島（10分）
 - モデルにすると規模・時間軸それぞれを拡大できる
 - 実施の壁は各ステップごとにある
 - チームを作って試行錯誤を回しモデルの精度を上げるのが重要
- QA/議論 (15分)

フェーズ（再掲）

現行運用

- 手書き(人用)構成図・ドキュメント・台帳等を基にした人力運用
- 構成変更は実機(実環境)直接操作

モデル「育成」期

- 現行運用に対する判断材料の提供・支援等、情報参照での利用
- 構成変更は従来運用を維持
- ➡ モデルの精度・モデルを使ったオペレーションの信頼性向上
 - モデルを信頼できるようにする
 - モデルを「育てる」



ターゲット

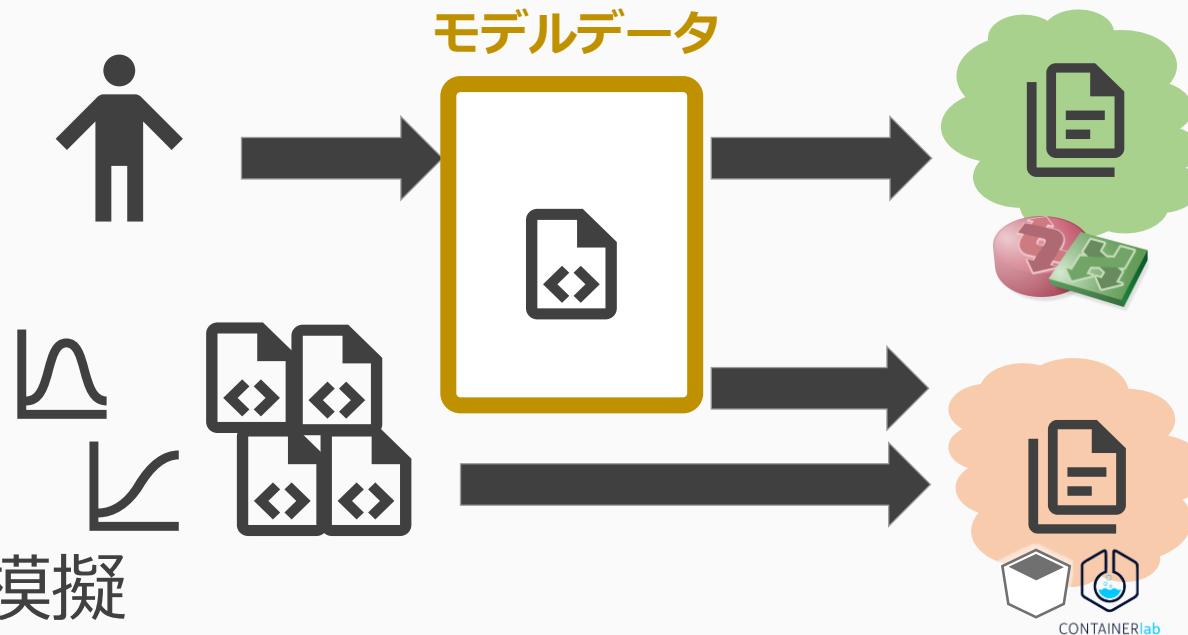
ユースケースのひとつ
…NW動作検証

モデル中心運用

- 構成変更の自動化・効率化へのモデル利用
 - 操作対象をモデルデータに切り替える

モデル中心になるとできること

- モデルを通じた実機の変更作業 w/ 静的検査（設計支援）



- 広い範囲の模擬
 - ピア先の模倣（サイズ方面への拡張）
 - 未来シミュレーション（時間軸方面への拡張）

障壁 (1) : そもそも要るの？

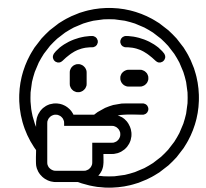
現行運用

- 手書き(人用)構成図・ドキュメント・台帳等を基にした人力運用
- 構成変更は実機(実環境)直接操作

モデル「育成」期

- 現行運用に対する判断材料の蓄積
- 構成変更は従来運用を維持
- ➡ モデルの精度・モデルを使う
 - モデルを信頼できるようにする
 - モデルを「育てる」

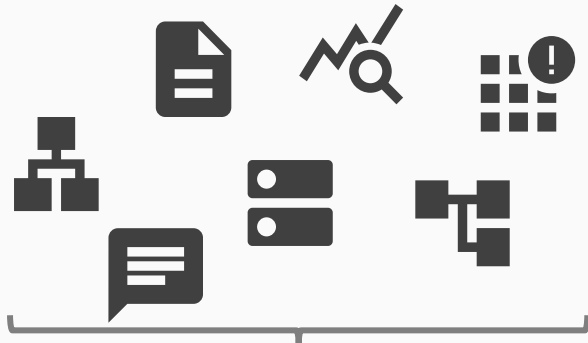
- いまの運用は支障なくできている
- 運用を変えると蓄積されたノウハウが失われてしまう



モデル中心運用

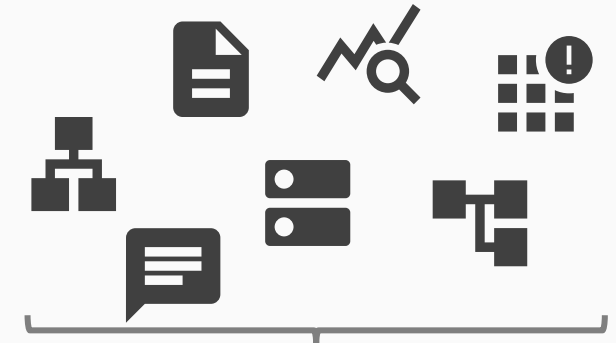
- 構成変更の自動化・効率化へのモデル利用
 - 操作対象をモデルデータに切り替える

モデル中心アプローチのねらい（再掲）

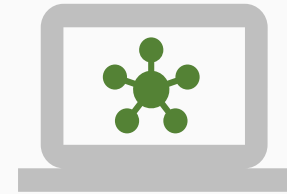


あらゆる情報処理、判断を人が行う
人がボトルネックになる

人が扱える情報量には限界がある
人はスケールしない
人には揺らぎがある

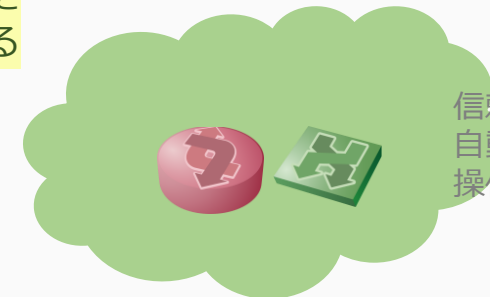
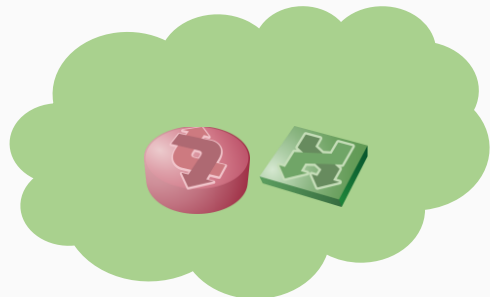


情報処理をシステムで支援する
人は重要な判断や操作にシフトする



モデルと
ロジック

情報量や複雑さに
マシンパワーとロジック(アルゴリズム)で対抗する



信頼度が上がれば
自動化レベルを上げて
操作も機械でできるかも

ネットワークの技術的負債の返済

- 特定の人に依存した運用は「リスク」であり「負債」
 - 中長期的な技術投資は「負債」の返済も視野に入れるべき
 - 既製の市販品導入では解決できない部分もある
 - サービスとして最終的なOK判断を出す稼働にシフトしたい
 - 直近の作業工数では測りきれない
 - このシステムすら、塩漬けすると負債になる
- 継続的にモデルの更新を続けられる体制にする必要がある

障壁 (2) : 併走し始められない？

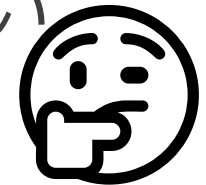
現行運用

- 手書き(人用)構成図・ドキュメント・台帳等を基にした人力運用
- 構成変更は実機(実環境)直接操作

モデル「育成」期

- 現行運用に対する判断材料
- 構成変更は従来運用を維持
- ➡ モデルの精度・モデルを使う
 - モデルを信頼できるように
 - モデルを「育てる」

- 見通しが立たない
- スーパーエンジニアがいない



モデル中心運用

- 構成変更の自動化・効率化へのモデル利用
 - 操作対象をモデルデータに切り替える

チームで地道に着実に進める

- scrumのように細かく回せる体制にする
 - 「育てる」ステップは問題に次ぐ問題で事前に要件を洗い出しきれない
 - 各問題では、原因分析・プログラムの対応・設計ポリシーの変更検討が必要
- 全員が同時にスキルを持つ必要は無く、チームを組む
 - 主にNW担当：configを読み解く＋ワークフローを変更して良いか考える
 - 主にコード担当：configがちゃんとモデルに反映されるようにする
- 達成度が測れる
 - モデル中心＝最後までモデル化するとゴール設定するとルータ数やピア数など「説明に使う数字」を算出可能

例) 数ある工事のうち、PNI工事（20%程度）をカバーできるようになりました

障壁 (3) : 例外をどこまで認める？

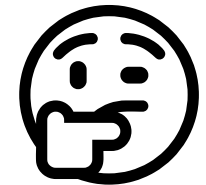
現行運用

- 手書き(人用)構成図・ドキュメント・台帳等を基にした人力運用
- 構成変更は実機(実環境)直接操作

モデル「育成」期

- 現行運用に対する判断材料の提供
- 構成変更は従来運用を維持
- ➡ モデルの精度・モデルを使用
 - モデルを信頼できるように
 - モデルを「育てる」

• 正規化つらい



モデル中心運用

- 構成変更の自動化・効率化へのモデル利用
 - 操作対象をモデルデータに切り替える

設計の正規化

- モデルができてても曖昧なルールのままではconfigが作れない
- ただし、特定のプロトコル仕様の100%対応や現行運用を100%移行した正規化は無理なので、塩梅を考える
 - モデル化へ移行する各ステップで、どんな業務で何ができることを狙うかを検討
 - 小さく始めて効果(実績)とモデルへの信頼を作る例) PNIユースケースだけをまずやる
- BCP上マルチベンダで組むのが必須なら、機能のANDで構成
 - 例) インターフェースdescriptionに使える文字 (文字数)
 - 例) policyに使える条件式や制御文

障壁（4）：モデルを信用できない？

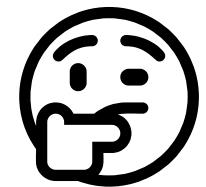
現行運用

- 手書き(人用)構成図・ドキュメント・台帳等を基にした人力運用
- 構成変更は実機(実環境)直接操作

モデル「育成」期

- 現行運用に対する判断材料の提供
- 構成変更は従来運用を維持
- ➡ モデルの精度・モデルを使用
 - モデルを信頼できるようになる
 - モデルを「育てる」

• どうやって「これなら大丈夫」と主張するのか



モデル中心運用

- 構成変更の自動化・効率化へのモデル利用
 - 操作対象をモデルデータに切り替える

モデルと検証環境の妥当性をチェック

- 100%の定量評価はできないので決めの問題になる
 - 目標完成度はシステムを「砂場」にするか「CI」にするかにも依存
 - どこに重点を置くかは各ネットワークによる
- モデルの改修 → 動作確認 → システムの改修 → モデルの改修
 - これをどのくらい早く回すか = ソフトウェア開発そのもの

例) ピアが確立してshow route + iperfでトラフィックが流れた

まとめ と 議論したいこと

- モデル(構成情報)中心の運用へ変化させようとしています
- BIGLOBE AS2518でのユースケースの検証ができました
- 運用を変える際の対処ノウハウを紹介しました

- 個別ルールなど例外がある既存環境の扱いについて
 - 「ウチはこうしている」 etc.
- 解決すべき障壁について
 - 「ウチはこんなところも苦労した」 etc.