

ネットワークのデジタルツイン に求める要件は何ですか？ ～理想的な仮想環境への期待と現状～

事前公開資料

TIS株式会社

NTTドコモビジネス株式会社

伊藤忠テクノソリューションズ株式会社

萩原 学

田島 照久

山口 大樹

事前公開資料の目的

- ネットワーク・デジタルツイン (NDT)
 - そもそもどんなモノで、どこに課題があるのかという共通認識がない
 - 例) シミュレータとデジタルツインは何が違う？
 - 例) 従来の「ラボ環境」とは何が違う？
 - 議論の土台になる共通認識を作る
- 共通認識を作るために
 - NDTについての解像度を上げる
 - 今回ターゲットにするNDTをしぼりこむ
 - NDTを運用に導入しようとしたときの課題点を知ってもらう

議論ポイント

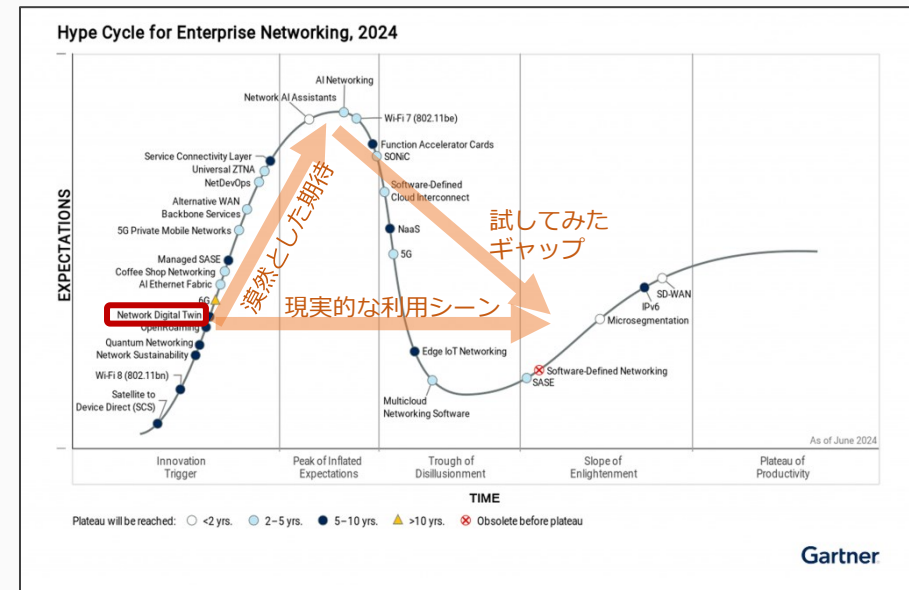
- 従来のラボ環境にせよ、NDTにせよ、100%商用環境を再現可能な技術はない
 - (コストが許す範囲で)既存のラボ環境でやれることはおおむねやってるはず
 - それだけでは足りないこと、NDTに期待できるのはどこまでか？



(従来のラボ環境の基準で)
「〇〇が検証できないならNDTなんて意味ない」
➡ なぜ？ 本当にそう？

- All-or-Nothingで話をしてしまうと今のやり方から抜け出せない
- 何が言えればどんなことが実現できるのかをみんなで考えたい

あと何を・どこまでできれば、今のシンドイやり方を、部分的にでも変えられると思いますか？



Hype Cycle™ for Enterprise Networking, 2024 -
Telefonica Global Solutions
<https://globalsolutions.telefonica.com/en/news/hype-cycle-for-enterprise-networking-2024/>

フォーカスしないこと

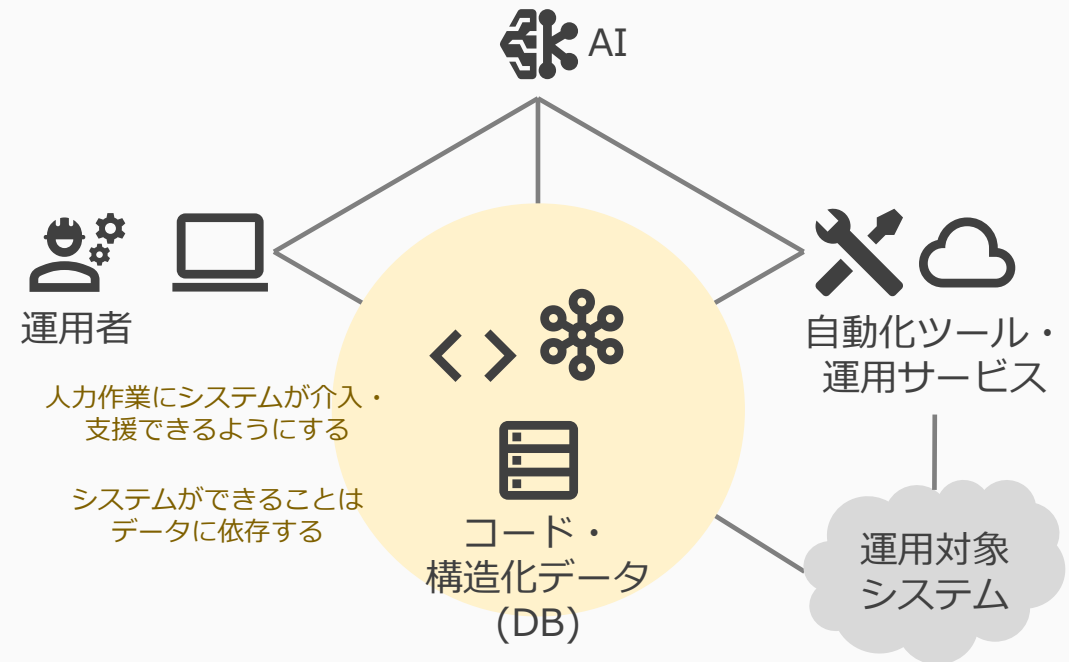
- べき論、理想論、NDTの定義
 - 例) NDTはこうあるべきだ
 - ➡ 具体的な使い方、導入～実用上の課題点、これから実際にやれることについて議論したい
- 個々の機能の差分
 - 例) XXでは〇〇ができるのか、できないのか
 - ➡ なぜ・どんな状況で・何を実現したいから"〇〇が必要だ"という話をしたい

なぜNDTが必要なのか？

モデル中心運用

- Excel台帳から構造化データに基づいた運用システムへ
 - 「高度な運用」を実現するために何が必要か？

人がやっていることを
システム(and/or AI)にオフロードしないと
「高度にする」のは難しい
➡そのための「モデル中心」



モデルによって解決したいこと

- 実際に運用しているネットワーク(商用環境)とその「状況」をデータとして表現する
 - 脳内メモリに納まりきらない「複雑さ」をシステムでフォローする
- 複雑さのフォロー
 - 過去事象の分析
 - 障害原因分析(RCA)
 - 将来事象の予測
 - 構成変更作業、オペレーションの妥当性検証、サービス影響確認
 - 小規模な検証環境では見通すのが難しい動きの検知・発見
 - KKD(勘・経験・度胸)運用の低減
 - 「本番一発勝負」の回避

その手段,
実現方法



ネットワーク
デジタルツイン

参考：過去報告

- これまで…NDTが実際に作れるかどうかを中心
 - [JANOG51: もし商用ネットワークをまるごと仮想環境に“コピー”できたらうれしいですか?](#) (2023年1月)
 - [JANOG53: BIGLOBE AS2518をまるごと仮想環境へ“コピー”してみた](#) (2024年1月)
 - [OOD2024: 運用者の試行錯誤を想定した NWモデル上での並列検証システム](#) (2024年12月)
- 今回は「作れる」から「実際に運用の中で応用できる」に視点を移動

そもそも 「デジタルツイン」って何?

デジタルツインに対する解像度を上げる

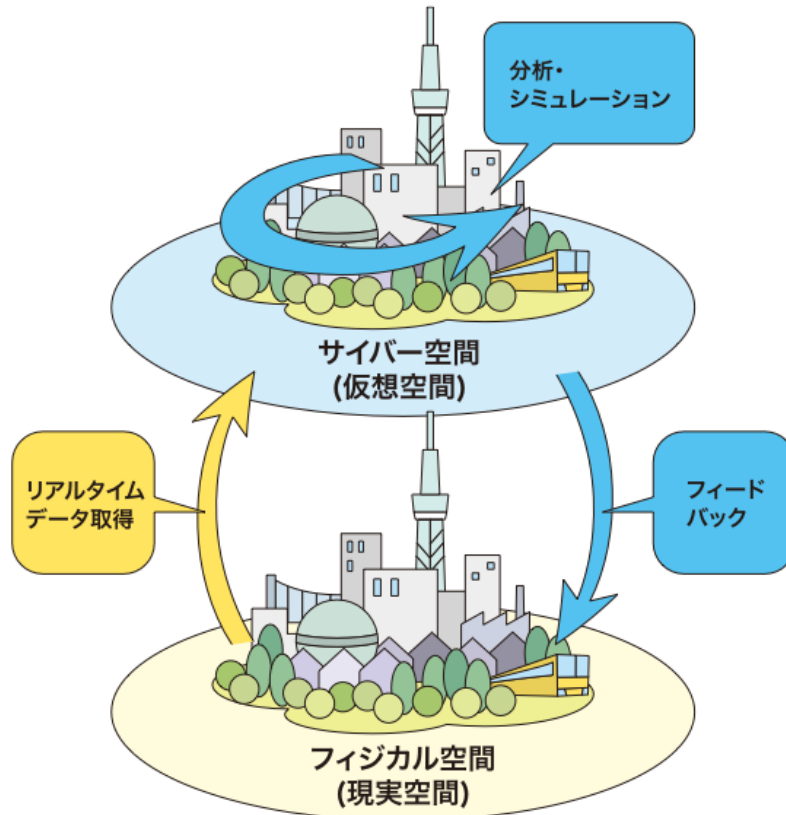
デジタルツイン

ネットワーク・デジタルツイン(NDT)は、これをネットワークシステムに対して応用するもの

IETF(NMRG)やITUのNDT関連資料では現実-仮想「マッピング」みたいな言い方

デジタルツイン (Digital Twin) とは現実世界から集めたデータを基にデジタルな仮想空間上に双子 (ツイン) を構築し、さまざまなシミュレーションを行う技術である。

総務省 | 令和6年版 情報通信白書 | デジタルツイン
<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r06/html/nd217530.html>

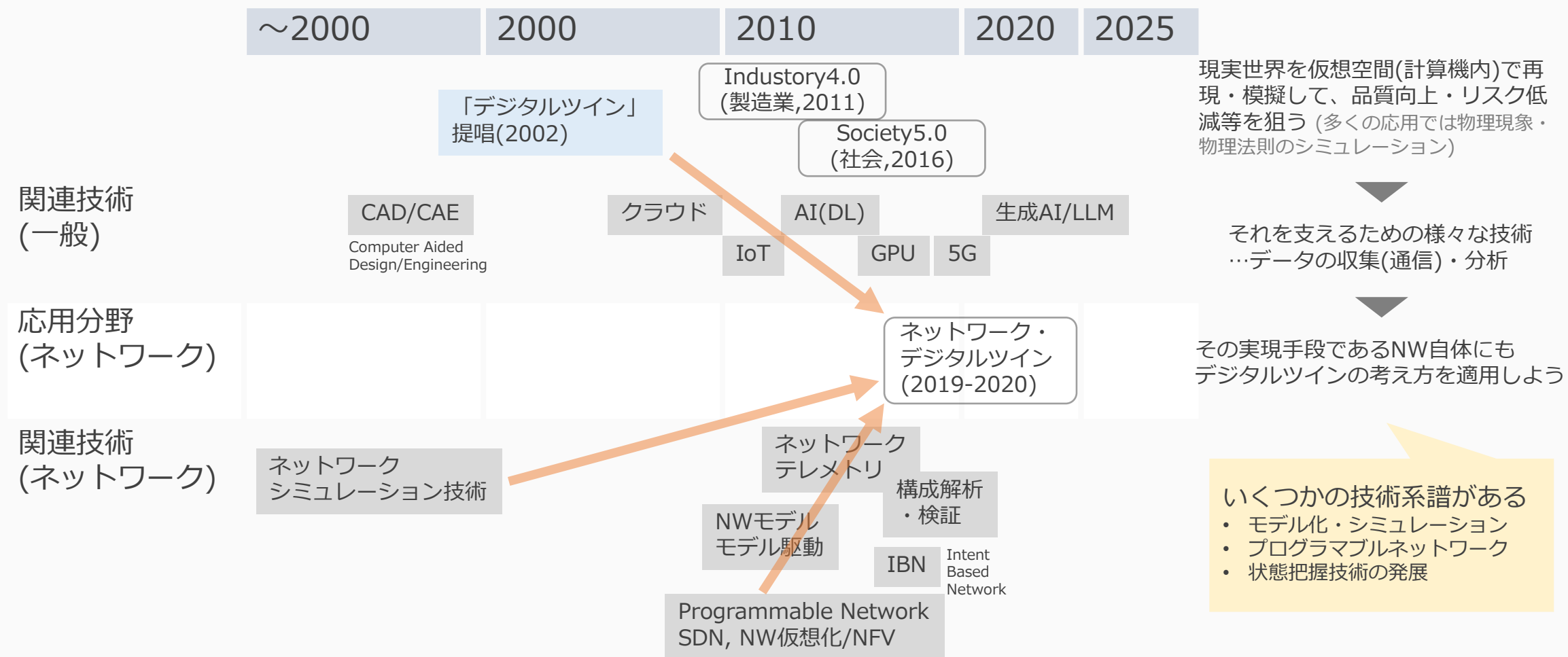


情報通信白書 for Kids : インターネットの活用 : デジタルツインって何？
https://www.soumu.go.jp/hakusho-kids/use/economy/economy_11.html

	デジタルツイン	シミュレーション
再現環境の違い	現実世界をコンピュータ上の仮想空間に再現する	コンピュータ以外(模型など)でも実行されることがある
リアルタイム性の違い	現実世界の状況に合わせて、その都度変化するためリアルタイム性が高い	「ある時点でのモデル」を活用する
精度の違い	実在する現実世界を再現する	いくつかの仮説をベースに実行する

デジタルツインとは？わかりやすく最新技術を徹底解説。メリットや導入事例も！ | SCSK IT Platform Navigator
<https://www.scsk.jp/sp/itpnavi/article/2024/11/digitaltwin.html>

NDT: どこから・何のために？



NDT: 実装の分類

方式		数理モデル型		仮想NW型	
概要		商用NWからある時点のトポロジ・ノード設定・状態を収集(snapshot)し、それらのデータを基に実NWの動作シミュレーション・問題分析をする。 ➡ データモデルとしてのNDT		商用NWのある時点のトポロジ・ノード設定を(snapshot)を基に、仮想NWを構築する。 実NWをエミュレーションする。 ➡ 実働する仮想NWとしてのNDT	
特徴	環境サイズ	◎	扱える環境サイズ(ノード数)が大きい	△	扱える環境サイズに限界がある (物理NWよりは大きくしやすい)
	状態再現	◎	元環境の状態に沿って分析可能	×	元環境の状態を直接再現できない (構成を合わせて状態を近づけることしかできない)
	将来構成	×	NWトポロジ変更など、現在の状態を基にできない将来構成での動作を予測できない	◎	実働するNWノードを立てるので、トポロジ変更等による将来構成での動作を予測できる
プロダクト		Forward Networks, NetworkBrain		Cisco Modeling Labs, OSS(GNS3, Containerlab) etc	

過去状況の再現・問題分析に強い

これからの予測(構成変更)に強い

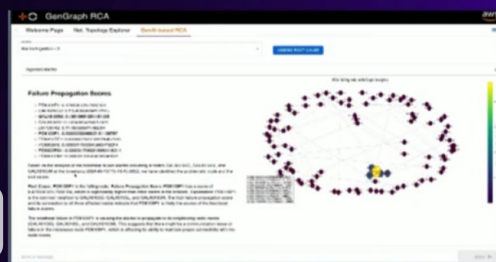
⚠ 従来の「ラボ環境」とのちがいは?

参考: NDT事例

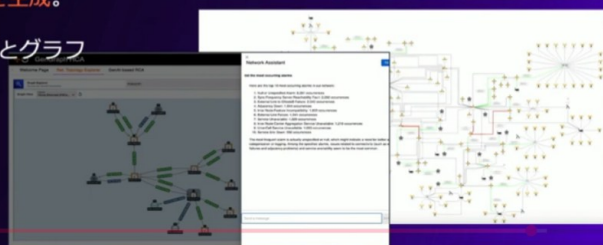
事例 Orange 様の取り組み

- ネットワークポロジをグラフとし管理することで依存関係を捉え、従来のルールベースでは難しかった RCA をグラフ技術で可能に。
- ダイナミックに変化するネットワークポロジを動的に可視化し、デジタルツインを形成した。
- 生成 AI による自然言語による調査とレポートを生成。
- 今回は実験的な取り組みだが他の都市への展開とグラフ分析の改良を予定。

ネットワーク障害 RCA



ネットワークポロジのデジタルツイン



ネットワークポロジの可視化と生成 AI アシスタント

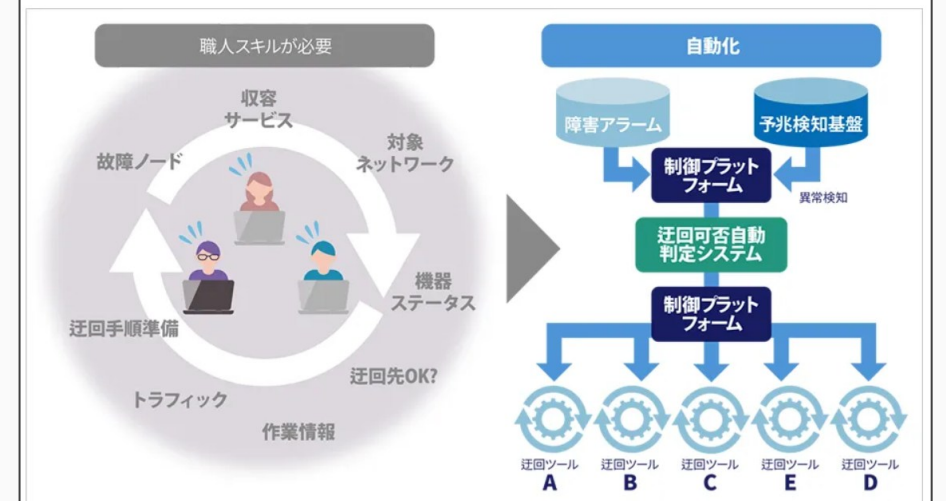
2025/3/11

【開催報告 & 資料公開】テレコム業界向け AWS re:Invent 2024 Recap – インダストリー 編 | Amazon Web Services ブログ

<https://aws.amazon.com/jp/blogs/news/reinvent2024-recap-telecom/>

(1) 「迂回可否自動判定システム」の詳細

異常検知時の迂回判断では、該当機器のトラフィックの流れやネットワークの冗長構成、迂回路路などの設計情報に加えて、作業や故障によって変化するネットワークの状態などを総合的に把握して判断する必要があります。そのため定型的な自動化が難しく、従来は熟練の技術者が都度状況を確認し、判断していました。今回開発した「迂回可否自動判定システム」では、機器や関連システムから取得した設定情報やステータス情報、作業情報、アラーム情報をほぼリアルタイムに収集し、ネットワークの構成変化に特化したデジタルツイン上で状況を分析します。これにより、実際の状態に基づいた自動判定を実現し、サービス復旧までの時間を大幅に短縮しました。また、監視工数の削減や個人のスキルに依存しない安定的な運用体制の構築にも寄与しています。



2025/11/25
デジタルツインを活用したメトロネットワークの運用自動化システムを全国展開～ゼロタッチ運用を実現し、国内で初めてTM Forumの「Autonomous Networks」レベル3認定を取得～ | 企業・IR | ソフトバンク
https://www.softbank.jp/corp/news/press/sbkk/2025/20251125_02/

我々が求めるNDT

実装としての前提と、期待する効果

NDTを用いた検証への期待

従来のラボ環境での検証

仮想検証環境

- サイズは物理より増やしやすい
- 使うモノが同じとはいえない
- 同じく都度設計して環境を作る

物理検証環境

- 必然的にサイズ限定
- (ある程度)同じモノを使える
- 範囲を限定するために都度設計する



商用環境のNDTに対する期待

- 大規模・複雑で、人力では把握しきれないところで深刻なトラブルが起きる
…そもそも「把握しきれない」ことが前提
- 都度設計しない(現状をそのまま再現する)

⚠ 従来の検証環境の「外側」にあることにリーチしようとしている

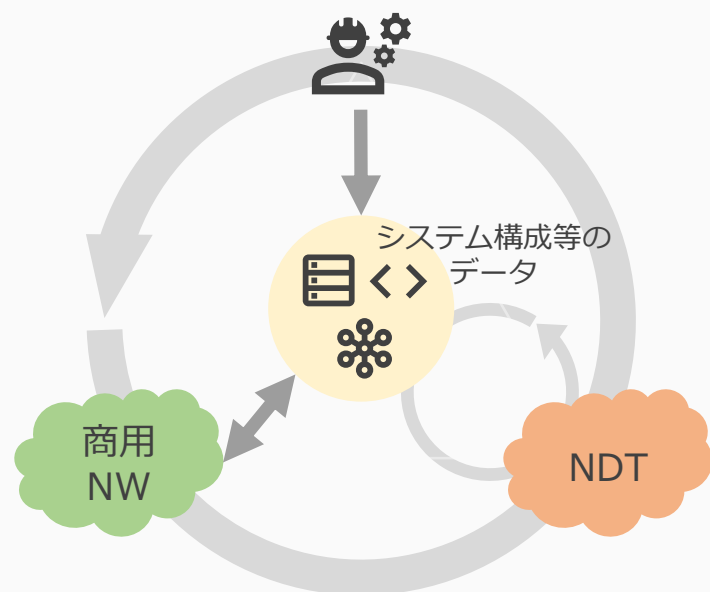
NDT: 従来の"ラボ環境"との違い?

従来: 設計上想定される機能、動作パターンの確認



NDT: 想定しきれない現実世界の構造・状態に基づく

- ・ 「現実世界」の構造や動き(ふるまい)の再現を基準にする
- ・ 現実世界の時間軸(リアルタイム性)を考慮する



データモデリング・データを基にした効率化

- ・ 対象を詳細に表現するデータモデル
- ・ モデル上での検証(verify)・シミュレーション・分析
- ・ 自動化・宣言的アプローチ (仮想NW型)

クラウドネイティブ的な考え方・運用プラクティス

- ・ CI/CDプロセス
- ・ Reconciliation Loop
- ・ 一時的(ephemeral)な検証環境 (仮想NW型)
 - ・ APIベースのリソース操作

NDT: われわれのターゲット

現実世界のモノ	既存の物理ネットワーク
現実世界の人	ネットワークオペレータ
現実世界で困っていること・ NDTに対する期待	<ul style="list-style-type: none">複雑な商用NWの動きまでを含めて確認することで、作業ミスや設定の見落としなどを防止する="本番一発勝負"をなくすNWオペレータが商用環境で行っている作業を再現したい<ul style="list-style-type: none">操作手順その時のネットワークのふるまい
そのために必要なもの	<ul style="list-style-type: none">実際に動かせる・試行錯誤ができるNW商用同等のオペレーションができる仮想NWとしてのNDT

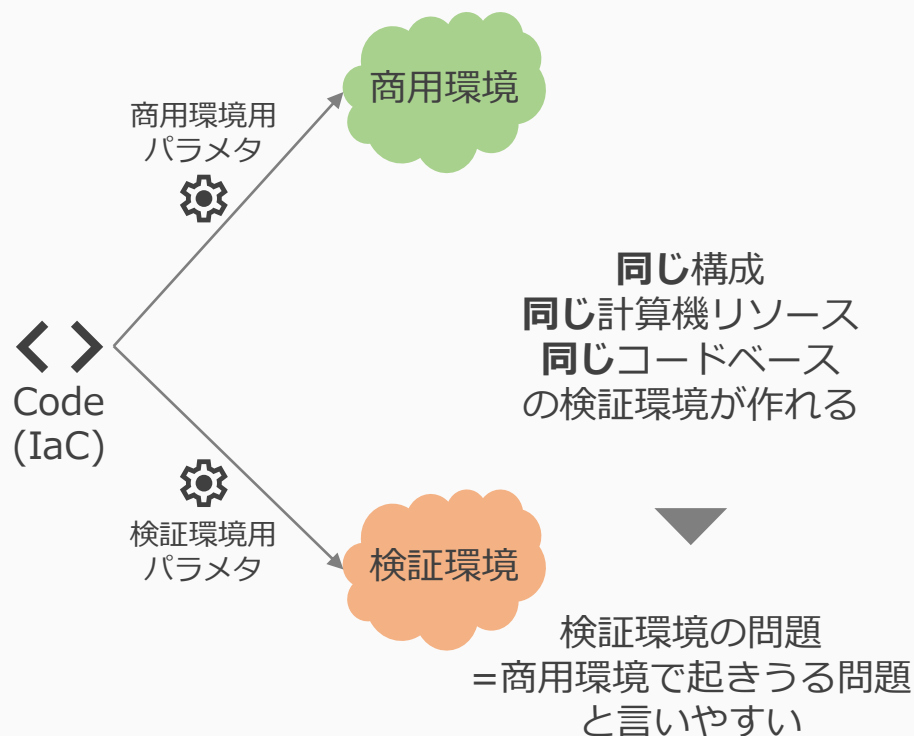
前のセクションで説明

NDTの難しさ

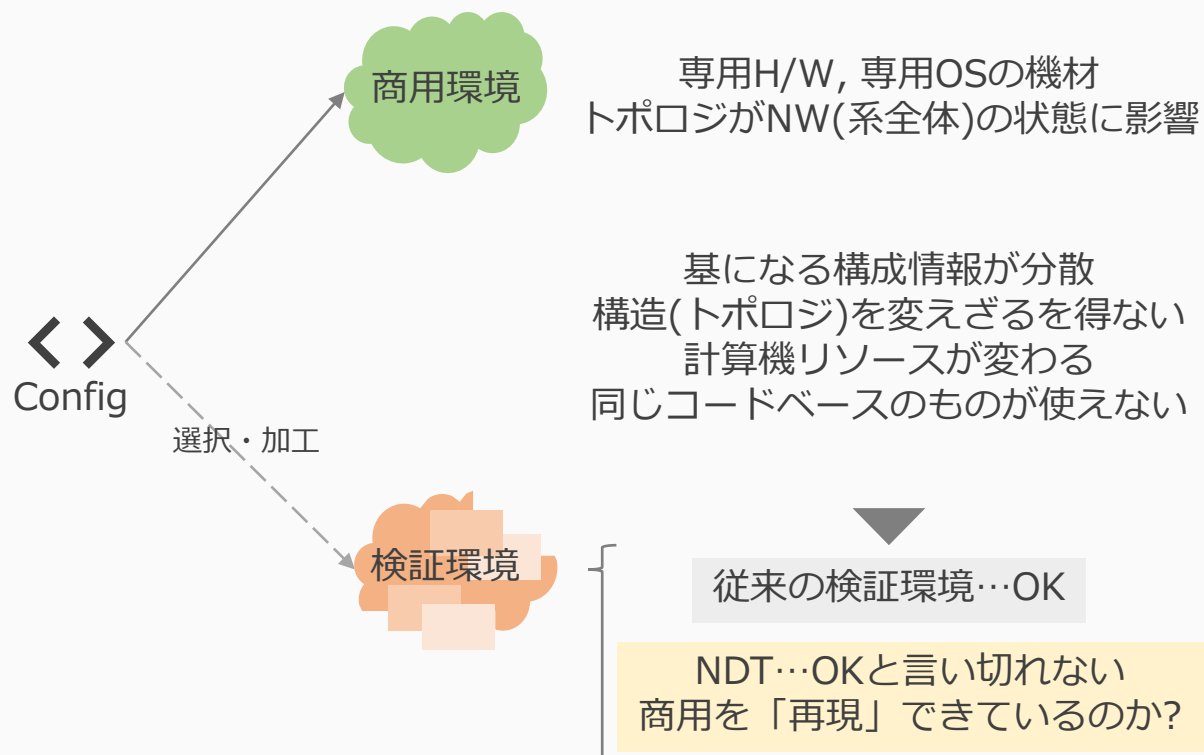
運用導入トライアルでわかったこと

前提: NW検証の難しさ…完全複製は不可能

(例)クラウド上のシステム

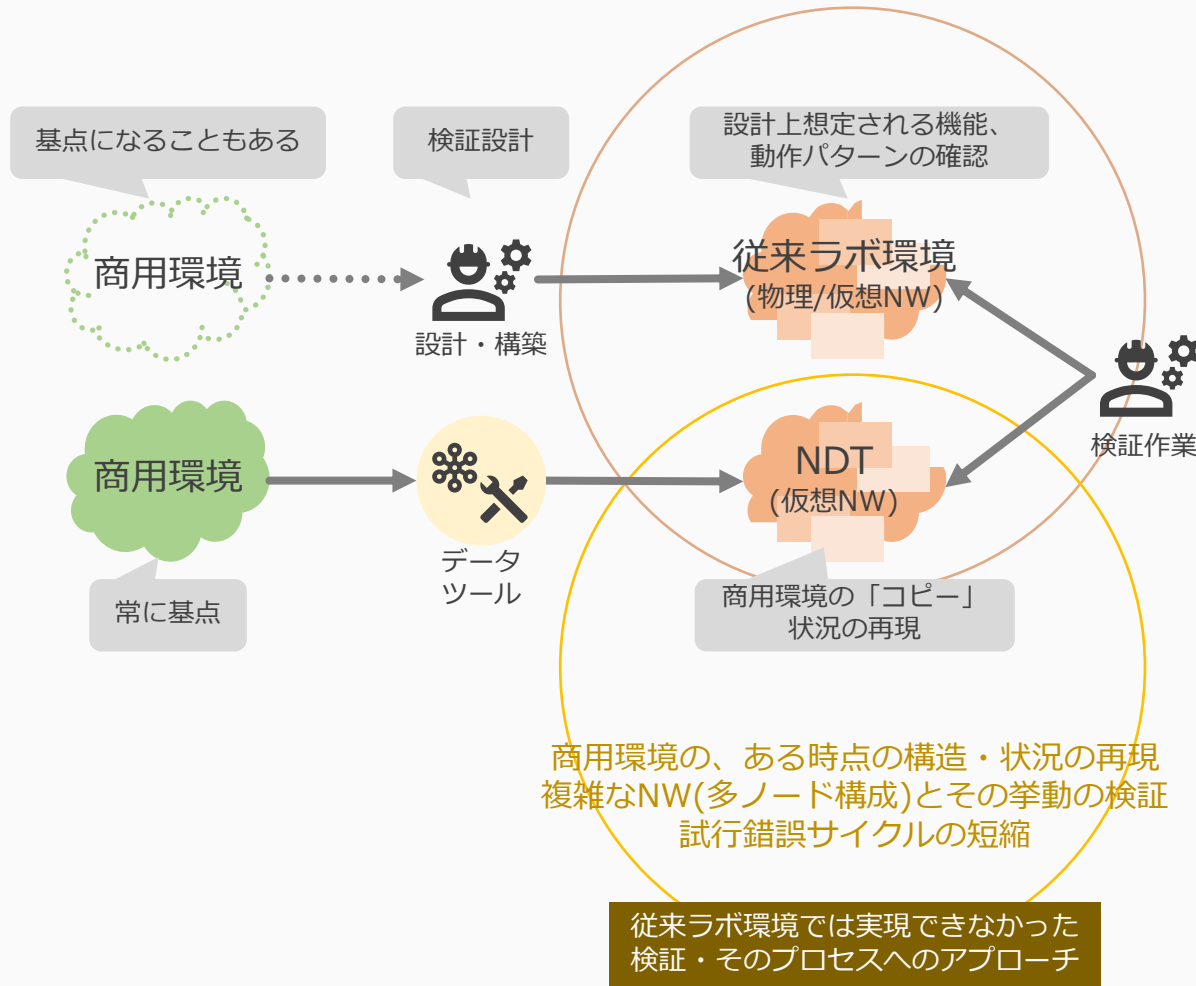


ネットワーク
(既存・物理)



⚠ 100%完全再現出ないのは従来の(物理)検証環境だって同じ NDTでは何に引っ掛かるのかを正しく認識する必要がある

期待とギャップ



検証目的・実現したいことに応じて取捨選択が必要
⚠ NDTで実現したいことに対して何がどこまで必要か？

従来ラボ・NDT問わず常に発生するギャップ

- NWの構造(トポロジ)、再現可能なレイヤ、ノードの設定
- NW全体(ノード)の状態
- 使うモノとそのオペレーション

NDT固有・従来ラボにはない考え方に対するギャップ

- 商用環境の状況に合わせて自動で起動する
 - 複雑な構成・状況をなるべくそのまま再現する
- 一時的(ephemeral)な環境での試行錯誤
 - 蓄積・メンテナンスすべきものは環境ではなくデータ
- 目的…実現したいもの・期待
 - "複雑さ"によって発生する問題へのアプローチ
 - "中身"の完全把握を前提としない
 - …「把握しきれないもの」をどうコントロールするか

NDT利用の壁

従来の検証環境

- シンプル・少数ノード
- 構成や設定を明示的に作る・都度設計する(=中身が分かる)
- (物理環境では)商用環境と同じもの(H/W, OS等)が使える

NDT環境(仮想NW)

- 複雑・多数ノード
- 構成や設定を人が網羅的に見ない・自動構築するので都度設計しない
- (仮想なので)使うものが異なる

NDTに対する期待

- NW環境が大規模化・複雑化
- 従来の検証環境では拾い切れない「複雑さ」を何とかしたい
 - 「やってみないとわからない」
 - 「やってみて初めて分かった」
 - 本番一発勝負・KKD(勘・経験・度胸)運用

NDT利用の壁

- NWの構造・規模(ノード数)を再現する
- 複雑な環境の中身すべてを個別にチェックしきれない⇒ブラックボックス化
- 「このブラックボックスは本当に信じていいのか」…信用問題の発生

課題に対するアプローチ

方針

- ネットワークが「同じ」かどうかには多数の観点が絡むので指標化するのは困難
 - ネットワークの同一性みたいな話を追うアプローチは難易度が高い
 - 仮に指標化ができたとしても、おそらく"テストカバレッジ"ぐらいの参考情報にとどまる
- 結局見たいのは「複雑なNW環境の動き/ふるまい」
 - 中身の詳細を追わなくても「**商用と同じように動く**」と示せば、**実用上問題ないはず**
- 過去(環境の複雑さが起因になって起きた)事象をNDT環境で発生させてみて、実際に同じ動きを再現できることが分ければ、そこを「信用」の基点にできるはず
 - **過去障害の再現**
 - 過去に起きた事象が再現する➡この先やろうとしているオペレーションとそれに伴う事象も予測できるはず

NDTによる再現～切戻し～予測 プロセス

事象の発生/再現

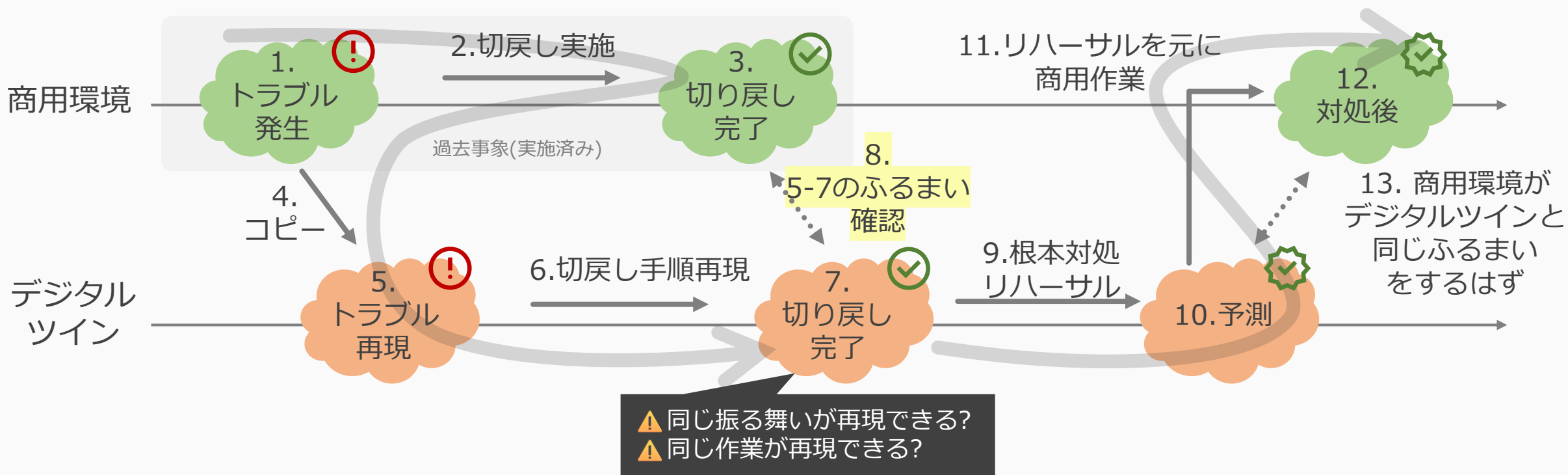
デジタルツイン上で
トラブルを再現

切戻し

商用と同等な手順で切戻し実施
振る舞いが一致するはずだ

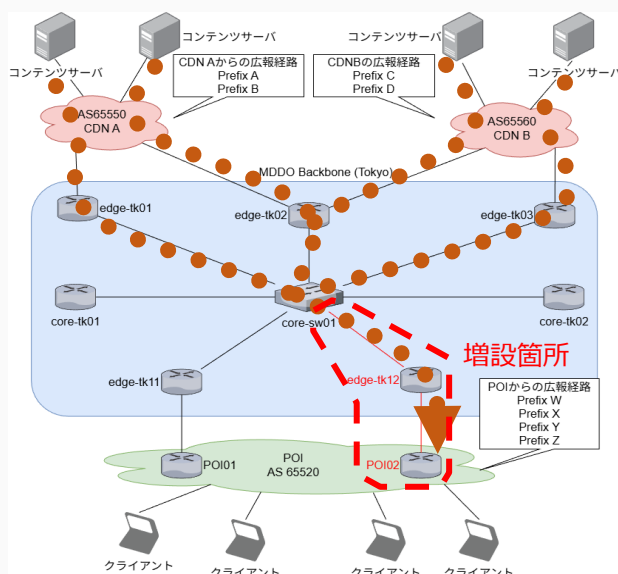
予測

過去事象を同じく再現できるなら
対応策(未来)の検証も信頼できるはず



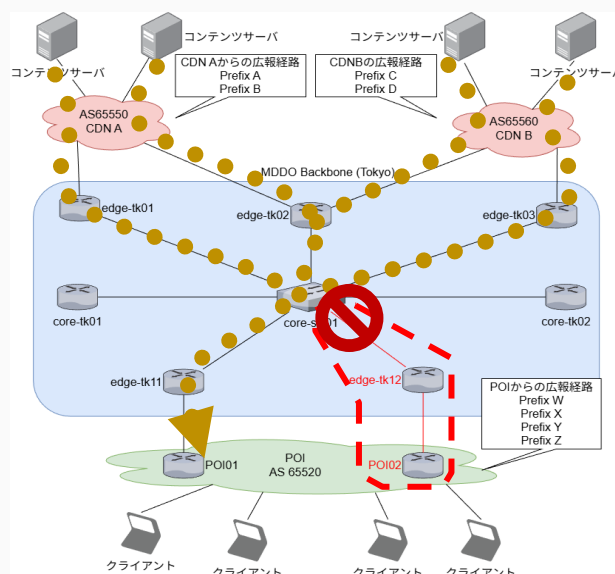
過去障害の再現トライアル

過去(トラブル)の再現



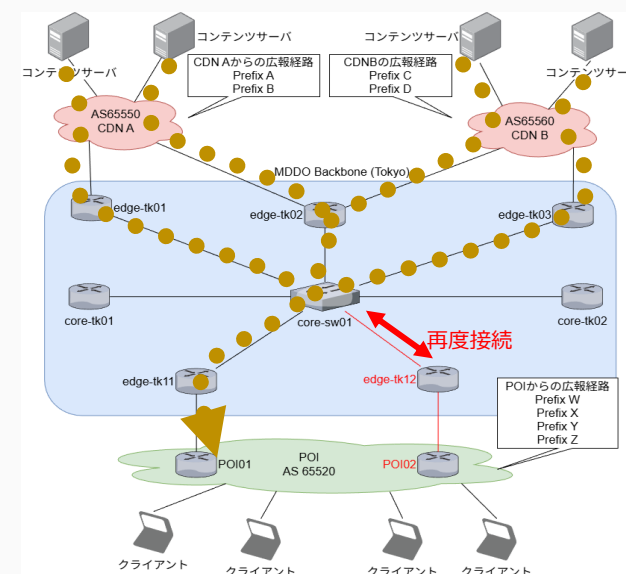
BGP接続箇所の増設時、増設ルータ側に
意図しない経路切り替えが発生

現在(切戻し)の再現



切戻し作業を行い、
元の通信経路へ復旧

根本対処(予測)



NDT上で原因特定
根本対処の手順をNDTで実施し、
効果を判断

4-8: 設定ミス～予想しないトラフィックの発生、環境切戻しまでのNWふるまいを再現

9-13: 原因特定～対策の実施
将来構成でのNWふるまい予測

NW操作(作業)の再現/ギャップ

- ノードのH/W構成依存コマンド実行
- 物理トポロジ操作の読みかえ
 - 操作対象ノードのインタフェース名
 - L3トポロジ操作への読みかえ
- 仮想環境固有のノード状態操作
 - Ad hocなリソース追加ができない
 - Ad hocなトポロジ変更ができない
 - インタフェースshutdownができない

回避不可能

手順スキップ

仮想NWノード
の問題

仮想NWノード実装依存
ツールで名前変換。読替を回避可能な機能を持つ仮想NW
ノードも最近出始めている

NDT実装上の
問題

今回は人力で読み替え
(検証ターゲットはISP NW…L1/L3トポロジのギャップが少
ない環境になっているので要注意)

仮想環境
(仮想化技術)の
問題

- オペレーションで必要なリソースを最初からある程度定義
して仮想環境を起動することで回避
- 直接的なトポロジ変化ではなく、OVSブリッジでのリンク
操作を挟むことでトポロジ変更作業を実現
- NWノード(仮想ノード)からのインタフェースshutではな
く、仮想環境の外(ホストOS)側のコマンド実行でインタ
フェースshut実施で回避

NWのふるまいの再現

- NDTでも**商用環境と同様のふるまいが再現できた**
 - 操作(作業)にギャップはあるものの致命的ではない
 - 狙ったNW全体の動きは確認できた
- **トラブル～切り戻し手順・NW動作の再現**
 - NDT上で根本対処(リハーサル)～動作予測も実施
 - 後は商用作業(NDTで得られた結果のフィードバック)

前頁

まとめ

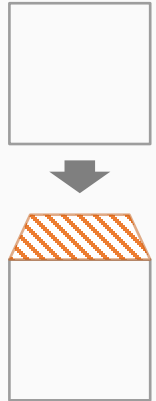
まとめ

- ネットワーク・デジタルツインを実際の運用プロセスに入れようとしたときの壁……「信用」の基点がないこと



- 信用を積み上げるプロセスを作る
 - デジタルツインを少しずつ信用できるようになる
 - 過去事象(障害)の再現 ～ 将来予測 プロセス
 - モデル育成フェーズ
 - トポロジーモデルから生成したデジタルツインの信用を積み上げる

"のりしろ"がないとくっつかない



次頁

フェーズ

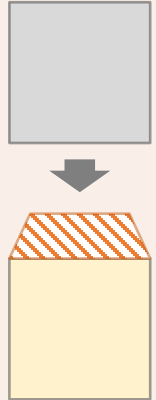
現行運用

- ・ 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
 - ・ モデルを信用できるようにする
 - ・ モデルを「育てる」

従来: 検証環境そのものを「育てる」
➡ 環境を作る元ネタ(モデル)を育てる



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

参考資料

参考資料

- Project紹介ページ: Okinawa Open Laboratory
 - <https://www.okinawaopenlabs.org/mdnd>
- Project Github
 - <https://github.com/ool-mddo/playground>
 - 成果物・デモ素材、公開資料・関連技術情報等の一覧があります