



ネットワーク機器の試験自動化BoF

NW自動化

このBoFの目的

ネットワーク機器の試験自動化に関する知見および課題を共有し、各組織における試験自動化の取り組みの継続的な改善・高度化に繋げていく

時間	内容	発表者
14:20-14:35	試験自動化における一般的な課題と解決方法	伊藤忠テクノソリューションズ株式会社 山口 大樹
14:35-14:50	ネットワーク自動化スクリプティングの昔と今	キーサイトテクノロジー株式会社 松田 健 様
14:50-15:05	試験自動化の歴史と落とし穴	スパイレントコミュニケーションズ ジャパン株式会社 中村 彰宏 様
14:05-15:30	質問・議論の時間	



試験自動化における 一般的な課題と解決方法

山口 大樹

NW 自動化

• NW/インフラに付随する自動化関連開発チーム

- 自動化オーケストレータWeb(OpS)の設計・開発・維持管理
- 自動化エンジン(NSO, Terraform, Ansible等)のライセンス販売＋操作用モジュールの設計・開発
- NWの試験自動化(Robot Framework)、出荷検査自動化開発および関連OSSの開発

Mission

Config EngineのAPIを統括する
Webオーケストレータ等の開発

NSOやAnsible/Robot Framework等の
モデル/PlaybookやAPI開発

Config Engineとの実機結合開発
NW試験の自動化(Robot Framework等)

Product



Value

要件整理支援および業務改善支援
OpS開発および工事業務のDX支援

コンフィグエンジンの技術サポート提供
自動化関連開発手法やナレッジの提供

マルチベンダーでの自動化
リグレーション試験の高度化

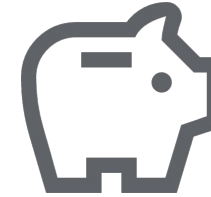
- 山口 大樹(やまぐち だいき)
- 経歴
 - 2019年CTC入社
 - ~2022 通信キャリア向けのNWエンジニア
 - 2023~現在 NW・インフラの試験自動化や商用作業の自動化周りの開発エンジニア
- JANOG暦
 - JANOG51 富士吉田が初参加
- 趣味: サウナ,ライブ鑑賞
 - サウナ・スパ健康アドバイザーという資格を持っています

- NW機器 試験自動化のモチベーション
- NW機器 試験自動化の実装
- 試験自動化事例
- 試験自動化を行う上での課題/議論したいこと
 - 自動化を進める体制
 - NW機器とのインターフェース
 - CLIパーサー
 - 自動化の開発環境

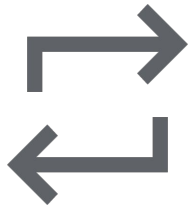
- 開発スケジュール短縮



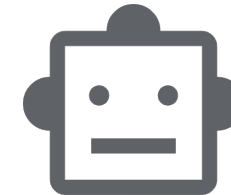
- コスト削減



- 繰り返し実施する工程の効率化
 - OSverUPに伴うリグレッションテスト
 - 障害試験/性能試験

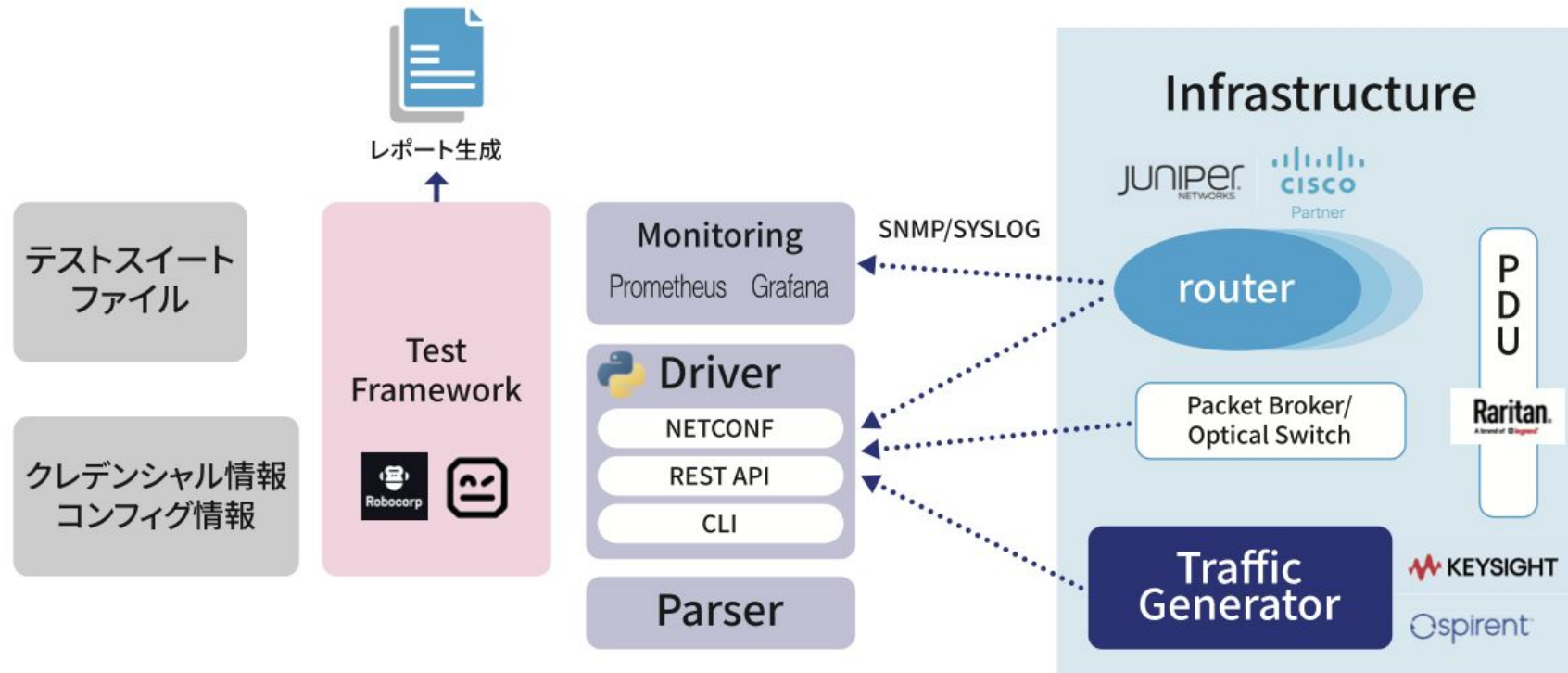


- テスト品質の均一化
 - 作業者のスキルに依存しない
 - ヒューマンエラー防止



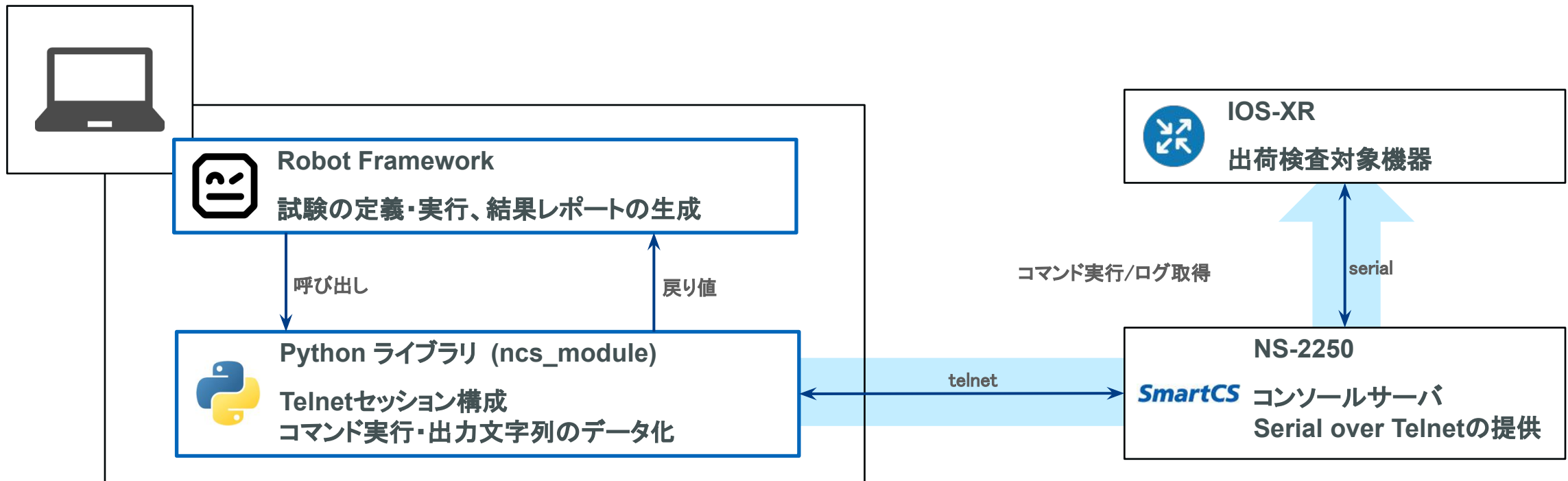
特に単純な作業を繰り返し実施する試験に対して自動化が効果的

- テストフレームワーク: RobotFramework/robocorp-tasks
- Driver: Python(netmiko/ncclient/requests)
- 周辺機器: Tester, Packet Broker, Intelligent PDU, Optical Switch



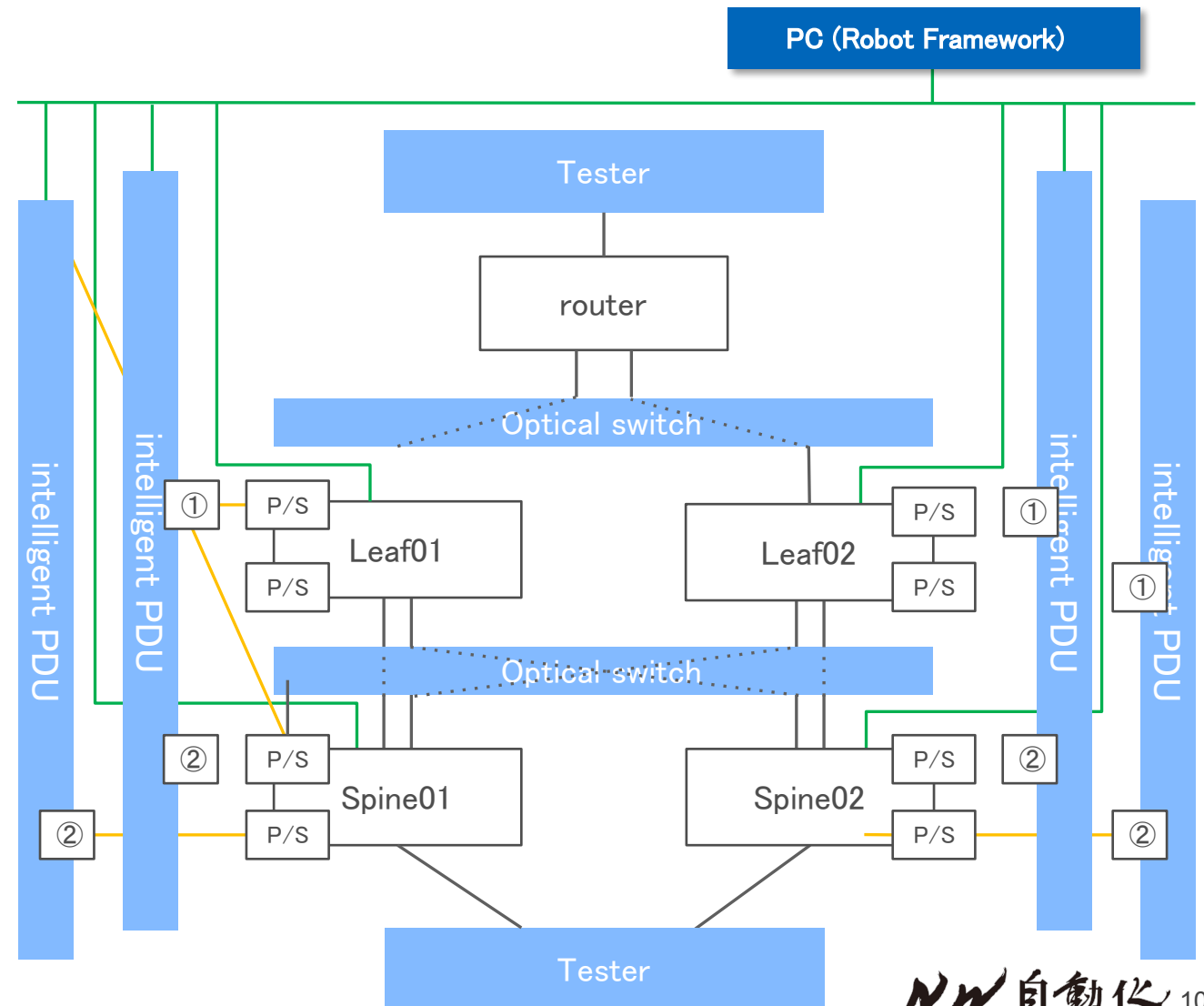
事例. 出荷検査自動化

- ルーターの出荷検査をRobot Frameworkを用いて自動化
- 効果
 - OS upgrade/パッチの適用等が自動化されることによる効率化
 - 台数が多い作業でも、Robot Frameworkであれば検査ミスは最小化され、品質が向上する



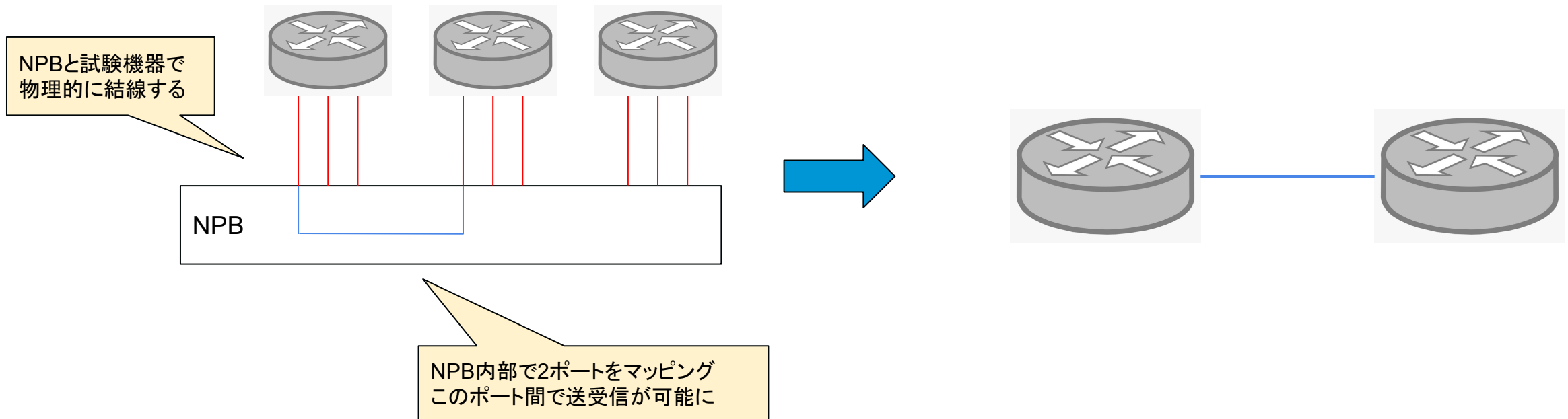
事例. 障害試験自動化

- NWの障害試験および長安試験を自動で実施。
リンク障害や電源障害をOptical switch/intelligent PDUで自動化。
- 効果
 - OS versionUP等に伴うリグレッション試験が自動で実施可能。
 - 手動で実施していた光/電源ケーブル抜線を遠隔でも実施可能に。
長安も自動で実施可能。
 - 平日の夜や土日にも試験時間として利用できることになり、スケジュールが柔軟に。



事例. 構成変更の自動化

- 試験を実施するにあたって、物理構成の変更が課題になることが多い
- ネットワークパケットブローカー(NPB)のポートマッピング機能により、試験機器同士の接続を柔軟に変更可能
 - NPBにはREST API等のインターフェースが実装されており、Robot FramworkよりNPBへの操作を自動化



試験自動化を進める上での課題/CTCの取り組み

- 自動化チームとNWチームに分かれている
- 理想はNWも自動化もわかる人が一気通貫でやれると良いが両方できる人は少ない



自動化チーム

- ソフトウェア開発スキル
- 試験の自動化を実装する



NWチーム

- 試験対象のNW設計知識
- 試験の内容を考える

- メリット
 - 特化すべきドメインを分割できる
 - 自動化ナレッジの集約
- デメリット
 - 試験に対する理解度に差があり、試験項目の曖昧さを解いていく必要がある

- NW自動化を進めていくにあたり、インフラ・ソフトウェア開発スキルの両方が必要だが、両方わかる人材は少ない。
- 我々の場合、1.5ヶ月ほどオンボーディングを実施し基礎的なインフラ知識とソフトウェア開発スキルをインプットしている。
 - インフラ
 - Linux
 - ネットワーク
 - ソフトウェア開発
 - Python
 - Git
 - RESTAPI
 - netmiko/RobotFramework
- その後は実践で経験を積む。
スキルレベルが課題になる場面ではそれぞれ得意なメンバーがフォローする

- CLIで試験実施する場合、出力データの構造化にパーサーが必要。OSSのパーサーを活用する機会が多い。
- 主なOSS CLI parser
 - [textfsm+ntc-template](#)
 - [genie](#)
 - [TTP](#)
- 上記はnetmikoの[send_command](#)メソッドから呼び出しも可能

- 辛いところ
 - 各テンプレートに対応していないコマンド・プラットフォームは都度自作する必要がある
 - OSSのCLIparserでは欲しい情報が取れない場合もある
 - OS側の出力が変わると追従が必要

- 自動化開発側としてはCLIよりもNETCONFやRESTAPIなど使用できる方が開発コストが低い。しかし実際の運用だとCLIが多く、どうしても試験観点としてCLIが残る。

	自動化開発者		NW技術者		運用者	
	CLI	API	CLI	API	CLI	API
心理的障壁	小	小	小	小	小	大
技術的障壁	大	小	小	大	小	大

- CLIを捨ててAPIで試験できるか？
 - CLIベースの運用プロセスを変えていく検討が必要
 - 自動化開発者/NW技術者/運用者など立場によって意見が変わるので、NETCONF等の普及には運用プロセス含めて会話が必要？

- NW機器実機を含めた自動化の開発環境を用意することは期間的にもコスト的にも難しい
- 課題感
 - VMベースのNW機器で実装できるところを進めるがHWが関わる部分は実機での確認が必須。
 - とはいえ実機を何台も用意するのは難しい
- 最近だと開発用NW環境の構築にcontainerlab,コンテナルータの活用が期待できる。ただし、コンテナルータも実機OSと乖離してる部分があるので辛いポイントはある
 - configの互換性
 - 管理IFの扱い
 - c-planeのみ実装など

参考: [JANOG52 コンテナルータをルータ単体として使う野良BoF](#)

全般

- 試験自動化を取り組んでいますか。また、どういう体制で取り組んでいる・取り組もうとしていますか
- 試験自動化を進める上でボトルネックになる部分はどこですか
- 人材の育成・確保ってどうしていますか。
- 自動化の効果測定はしていますか
 - コスト？時間？
- 試験自動化に特化した勉強会とかあれば嬉しいですか？

技術的なこと

- 開発側としてはCLIよりもNETCONFやAPIで試験自動化した方が楽だが、どのようにしていますか。
- 自動化の開発環境はどのように確保していますか
- 開発・実装で何か工夫していることはありますか
- 最近気になっているフレームワークやツールはありますか

CTC

▼ *Challenging Tomorrow's Changes*