



NXTWORK2019

ジュニパーが考える
Intent-Driven Networkへの取り組み
～ クラウド、ネットワーク、セキュリティにもDevOpsを ～

2019年10月9日

ジュニパーネットワークス株式会社

上田 昌広



AGENDA

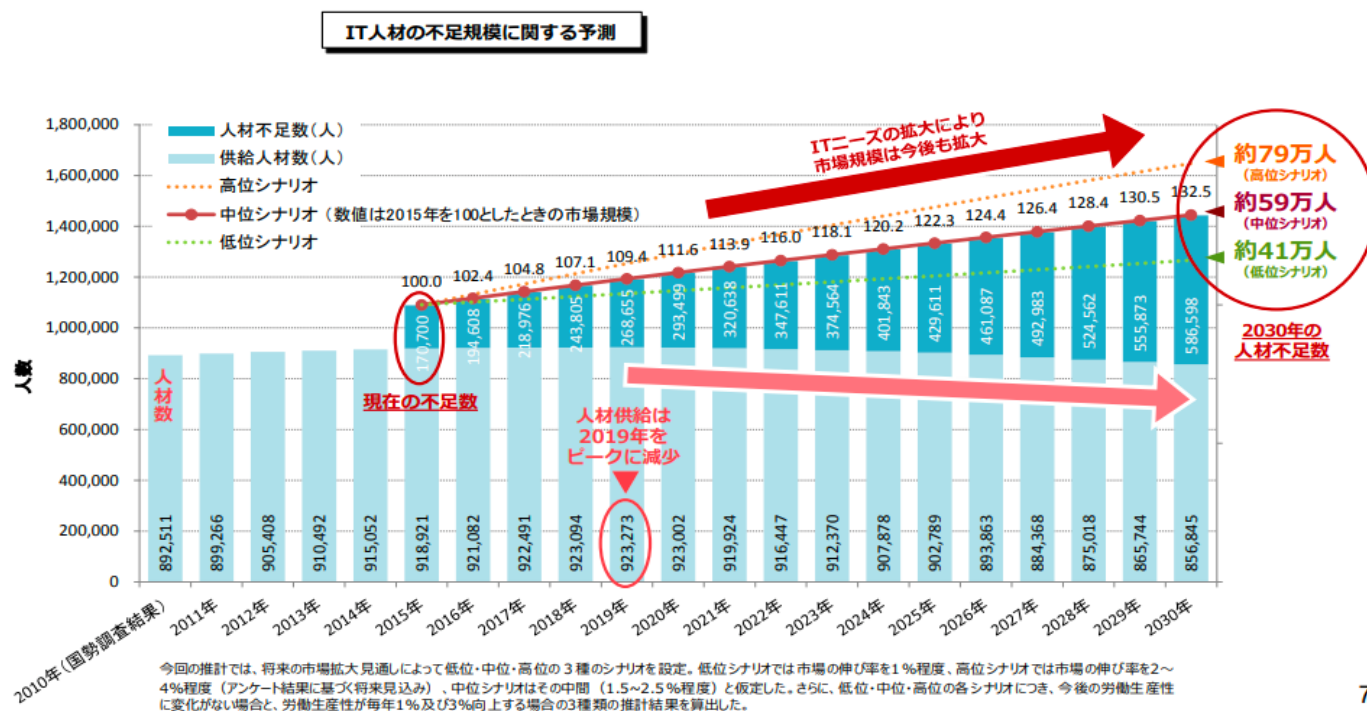
- マーケット・トレンド
- Intent-Drivenとは
- Network Reliability Engineer (NRE)とは
- TelcoCloud
- その他
- 最後に



AGENDA

- マーケット・トレンド
- Intent-Drivenとは
- Network Reliability Engineer (NRE)とは
- TelcoCloud
- その他
- 最後に

IT人材は2019年がピーク



2030年には

約59万人

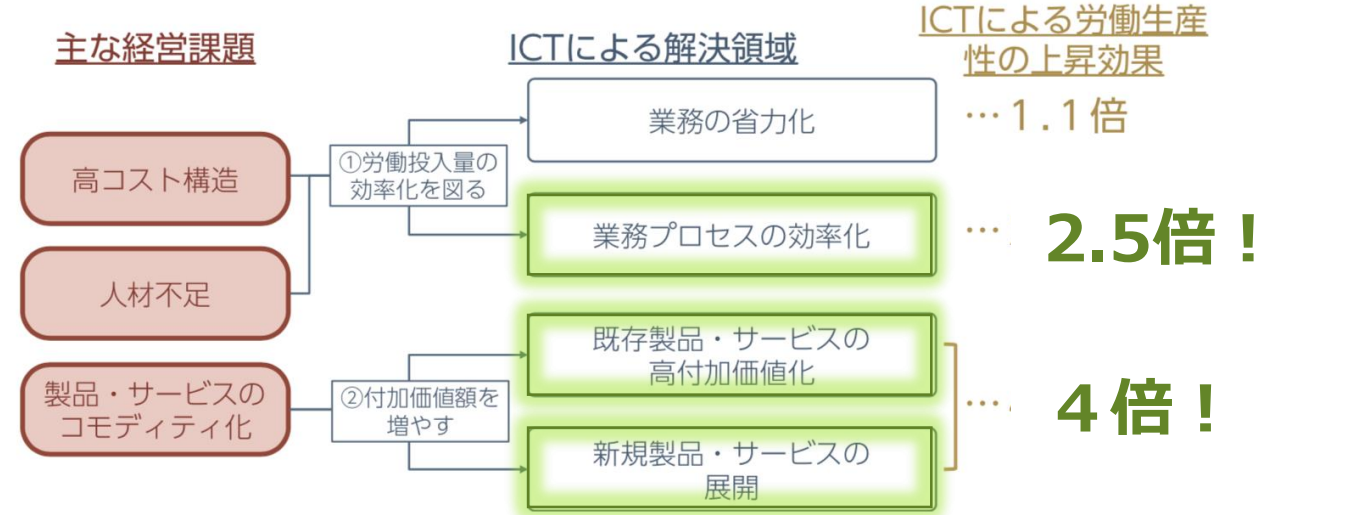
のIT人材が不足

※必要な人材の3割以上が不足

出典：IT人材の最新動向と将来推計に関する調査結果（経済産業省）

経営課題に対するICT解決ソリューションの効果

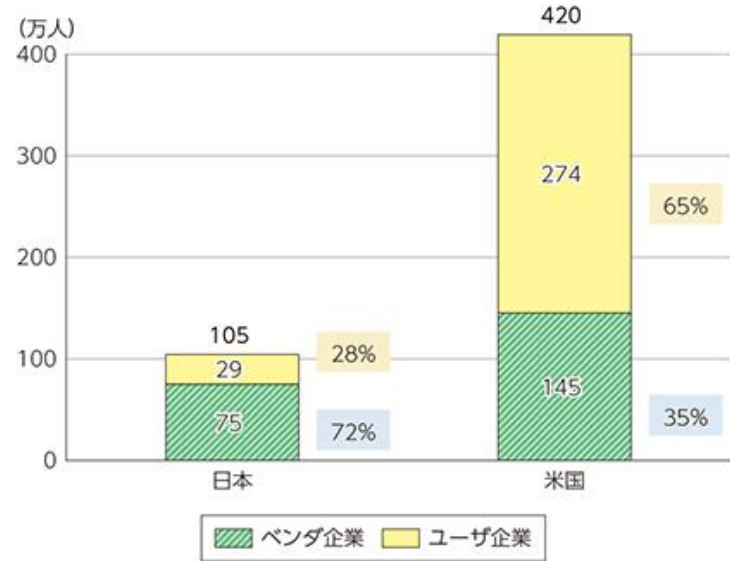
図表 3-2-4-1 ICTによる生産性向上の効果



(出典) 総務省「ICTによるイノベーションと新たなエコノミー形成に関する調査研究」(平成30年)

既存プロセスの効率化 < 付加価値・新規サービス

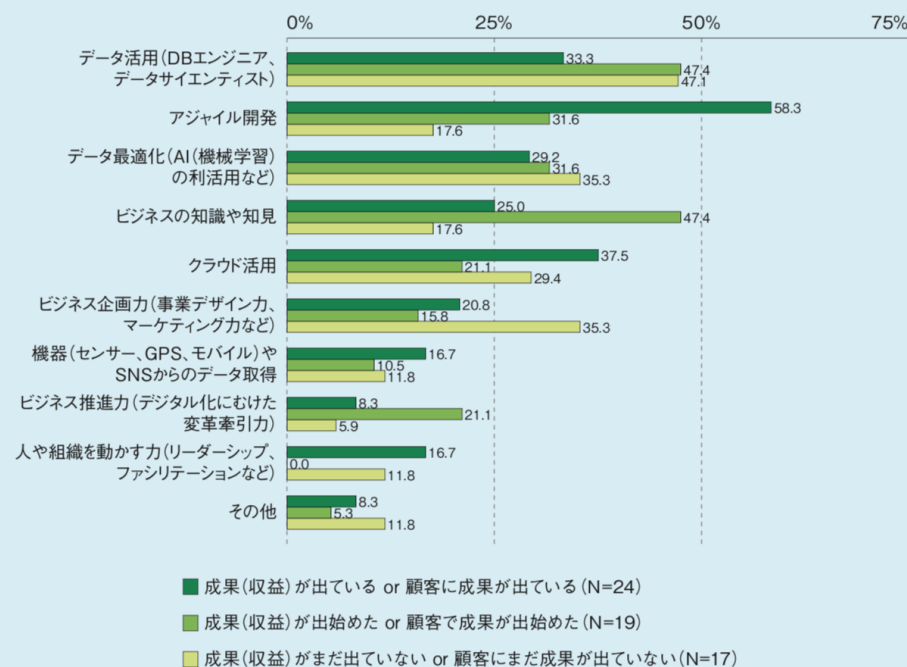
業務プロセス&サービスがわかる
ユーザ企業ITエンジニアが少ない



(出典) 情報処理推進機構「IT人材白書2017」より

人材スキルの傾向

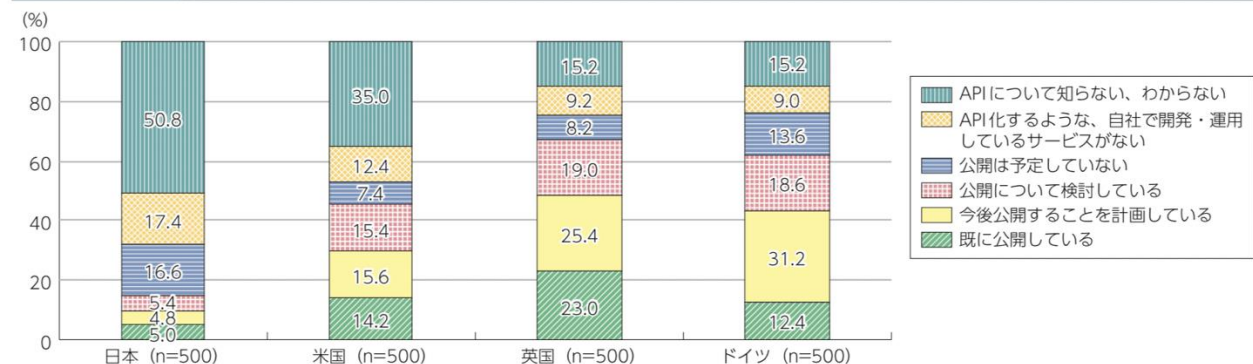
図表C-6 デジタル系企業がデジタル化に携わる人材を採用する上でどのようなスキルを持つ人材を重点的に採用したか【デジタル化の進み具合】



IPA IT人材白書2019より

- データサイエンスはこれから
- ソフトウェアのアジャイル開発は成果が出てきている

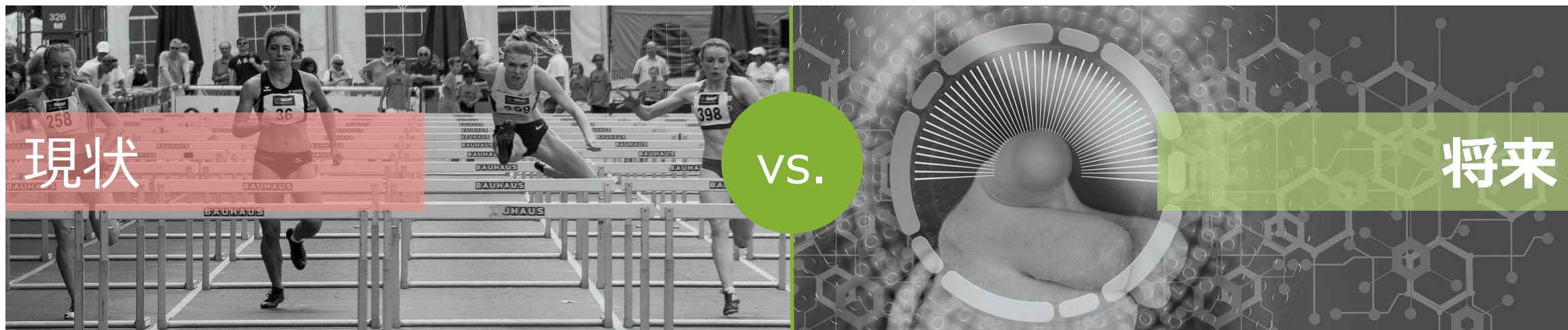
図表3-3-1-1 APIの認知・公開状況



(出典) 総務省「ICTによるイノベーションと新たなエコノミー形成に関する調査研究」(平成30年)

- 日本におけるAPIの知見、経験はこれから
- システム連携、アジャイル開発、DevOpsには必須

ビジネス成長力を止めないIT基盤にするために



IT活用の実態：

- 事後対応
- ITシステムの不良債権化
- 終わりのないセキュリティ対策
- 自動化、AIへの対応の複雑さ

競争力の強化：

- Pro-Activeな対応（可視化、予兆検知）
- アジャイルサービス基盤
- 迅速かつ柔軟なセキュリティ課題への対策
- 自動化による生産性の向上

信頼性

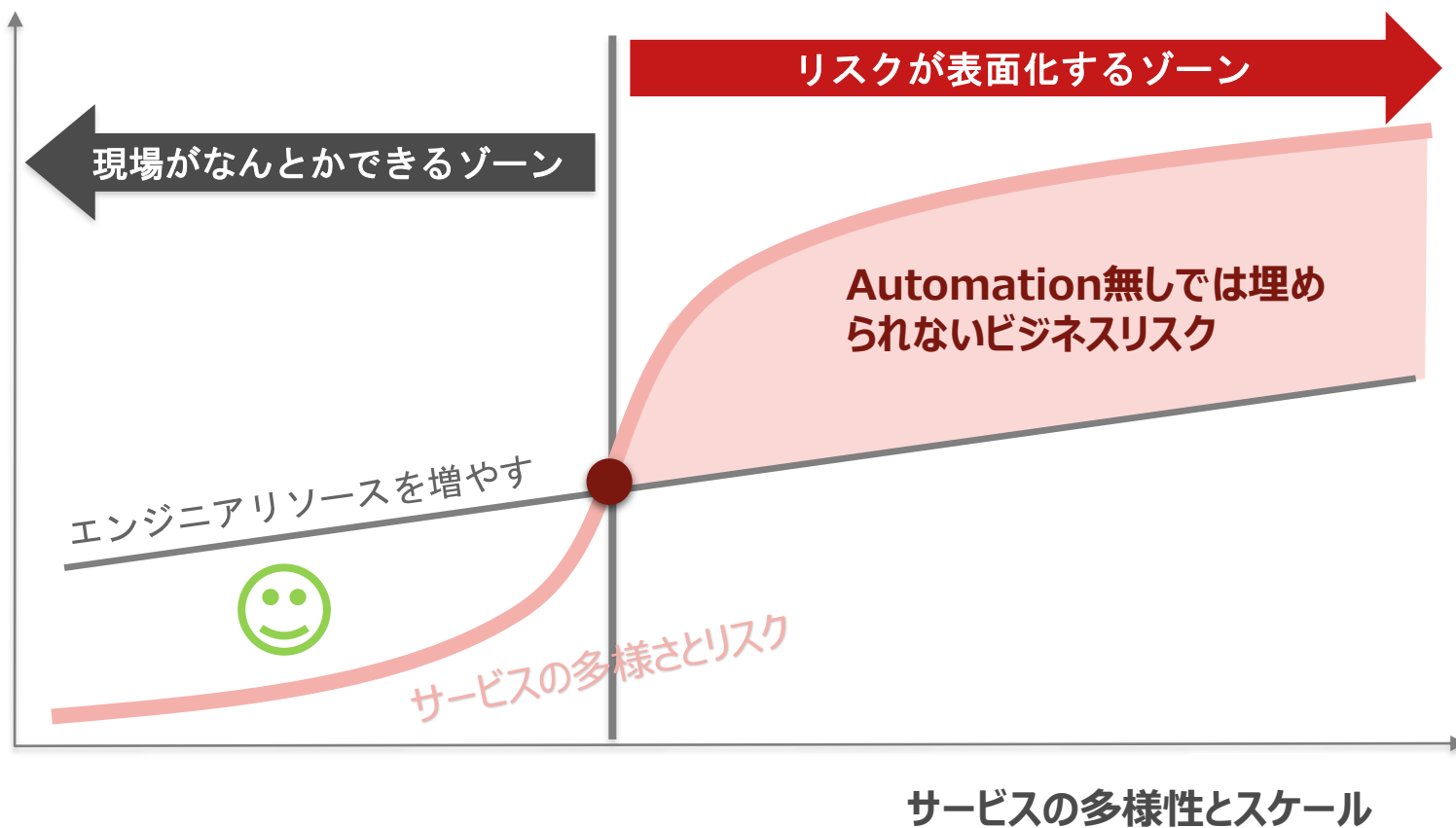
スピード

柔軟性

業務プロセスが変わるので根本的な見直しが必要 → 経営層の判断が必須

サービスの多様性 & スケールとリスクの関係

複雑さとリスク



人的リソースの割当増では限界が…

ネットワークも
DevOpsを活用した
プロジェクト展開が求
められてきている



NRE

(Network Reliability Engineer)

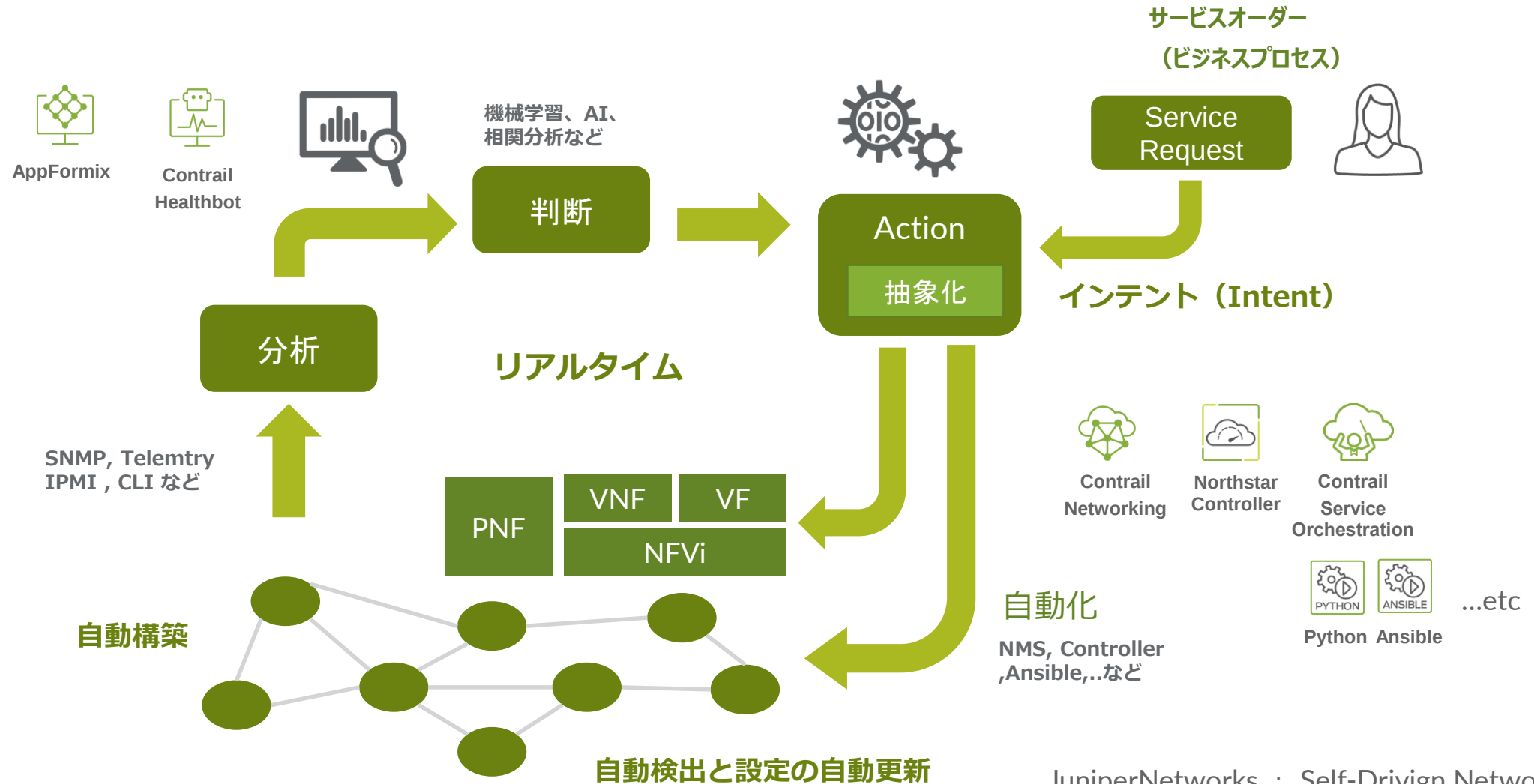


AGENDA

- マーケット・トレンド
- Intent-Drivenとは
- Network Reliability Engineer (NRE)とは
- TelcoCloud
- その他
- 最後に

INTENT-DRIVEN と CLOSED LOOP

Intent-Driven Network は Intentオブジェクトで定義された状態を自動で保つ、Closed-Loopを使ったシステム

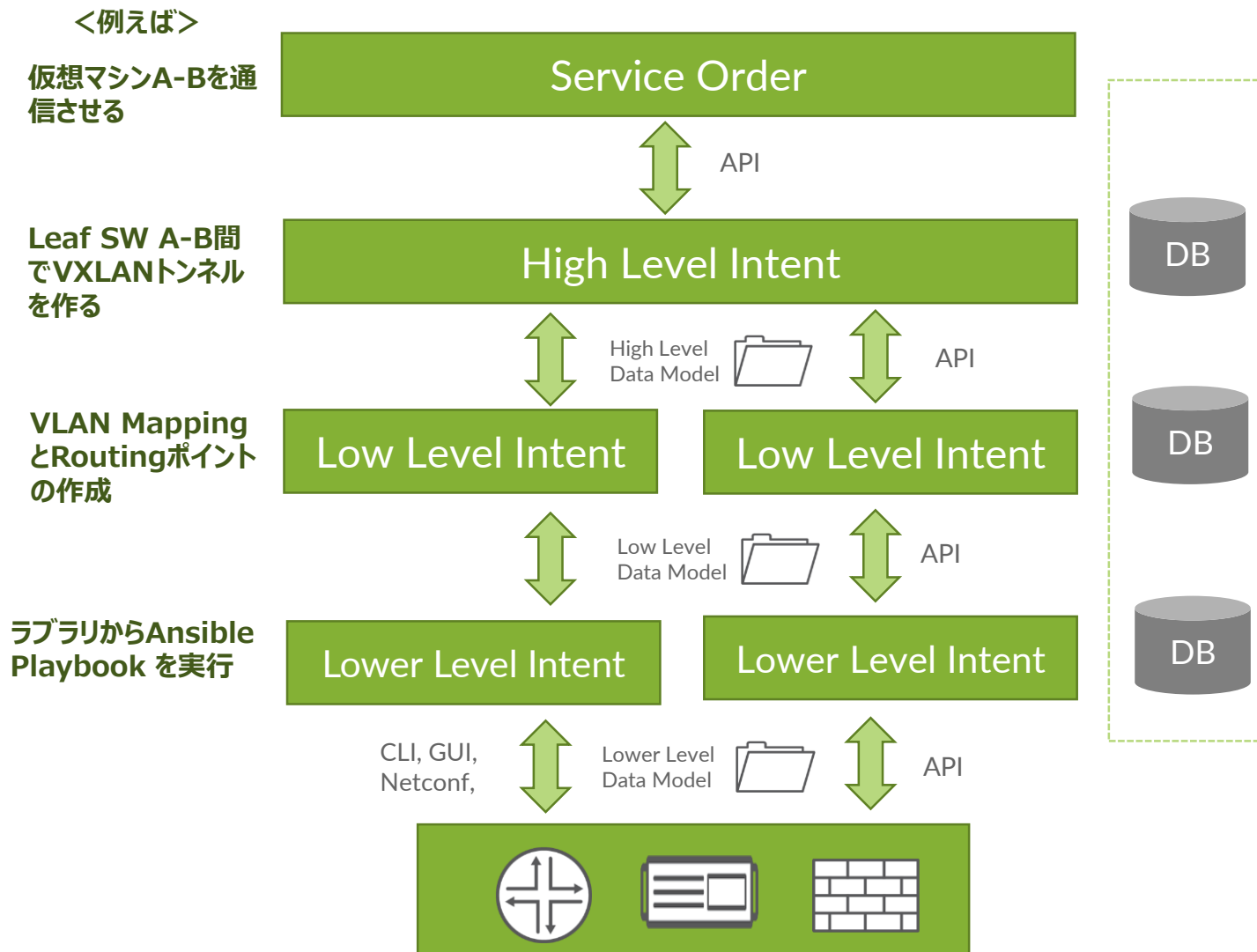


JuniperNetworks : Self-Drivgn Networking™

INTENT-DRIVEN NETWORKのコンセプト

- Intent-Driven Network 自体は Kubernetesなど活用されている
- Intent-Driven NetworkはマイクロサービスAPIやサービスデリバリーを意図したData Modelを トップ・ツー・ボトム のプロセスでインプリされる
- **Intent** は“意志” や“意図” などとされますが、提供するサービスをデリバリーするための一貫したオブジェクト

Data Model イメージコンセプト

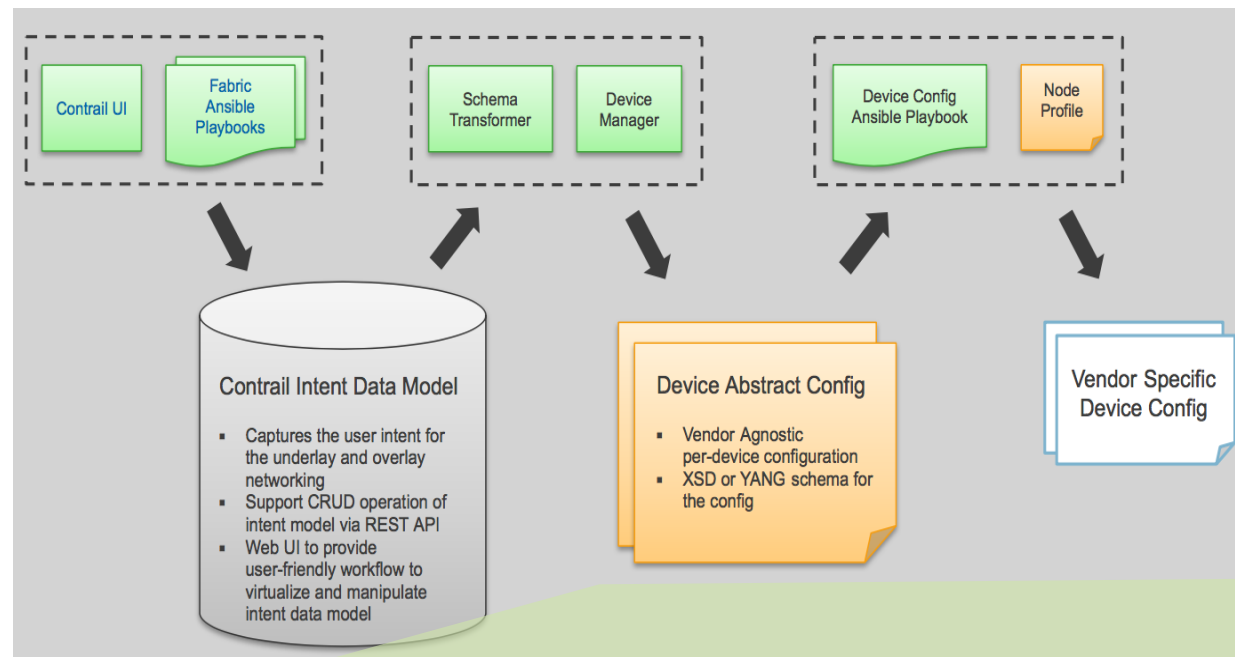


INTENT-DRIVEN NETWORKのコンセプト

- オペレーションの効率化を図ることで運用リスクを大きく改善
 - **Validation** : ネットワークワイドでの妥当性の確認
 - **Instantaneous Roll back/ Forward** : バージョニング管理されたロールバック、フォワード
 - **Blast-radius containment** : 新しいポリシー影響範囲の明確化
 - **Fallback** : 特定のコンフィグが成功／エラーを引き起こしていないかの確認及びリトライや代替え作などのフォールバック機能
- APIを経由したCRUD(Create, Reuse, Update, Delete)オペレーションを実現し、そのIntentオブジェクトを変更することでのみネットワークを期待するステートにする（他の作業は許可されません = 一貫性の確保）

例えば FABRIC SWITCH CONTROLLERなら

- Device On-Boarding
- Device Role指定
- GUIによるIntent指示
- 抽象レイヤを介したCRUDオペレーション
- スイッチへのコンフィグ実施





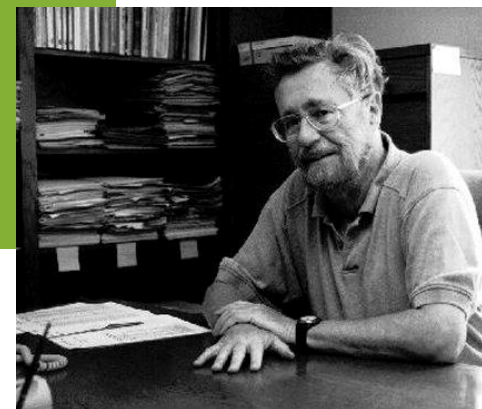
AGENDA

- マーケット・トレンド
- Intent-Drivenとは
- Network Reliability Engineer (NRE)とは
- TelcoCloud
- その他
- 最後に

シンプル化は信頼性の前提条件である

SIMPLICITY
IS PREREQUISITE TO
RELIABILITY

- Edsger W. Dijkstra
(Computer Scientist and father of the SPF algorithm)



ネットワークアーキテクチャーの進化とNETOPSの現実



82%

データセンターの82%のNetOpsがまだマニュアル運用に頼っている現実

… まだ進化できていない部分

ネットワーク自動化実現のための取り組み中…

- プログラマビリティとAPI、APIプロトコル
- SDN, NFV, 仮想NW
- SDKとツールキット
- Ansible, Puppet, Chef, Saltなどのツール連携

ただ、今日のNetOpsの現実…

- マニュアル運用が大多数
- CLIからGUIへのシフトで進化は見られるが、APIへはこれから
- 深夜作業は今まで通り
- 変更はサービス障害の主な原因であることは変わらない
- エンジニアはCLI/UI使いが多く、運用プロセスへ統合する“エンジニアリング”はこれから必要になる

NETWORKRELIABILITY ENGINEER (NRE)

(NRE = NETOPS + SRE)

従来のNetOps

- サービス実現にあたっての定型オペレーション業務
- ITに関するトラブルに対応する障害対応業務
- インフラに関する管理業務（構成管理やキャパシティ管理など）
- ワークフローオペレーションの効率化（自動化）



Site Reliability Engineer (SRE)

- ソフトウェア開発アプローチの採用
- 変更への即応性や信頼性の高いシステム基盤を設計
- 運用管理の自動化/サービス展開へのパイプラインの仕組みを設計 & 構築
- 開発者が利用しやすい標準化されたポリシーやルールの整備 など

Network Reliability Engineer (NRE)

DEVOPSによる手法（例）OPENSHIFT

Code

開発ツールを使ってWed、CLI, IDAインターフェースを使用して新しいアプリケーションを作成しコードをGitへ送る。

OpenShiftはバイナリコード展開を受け付けるか、既存のCI/CD環境へのインテグレーションを行う。

Build

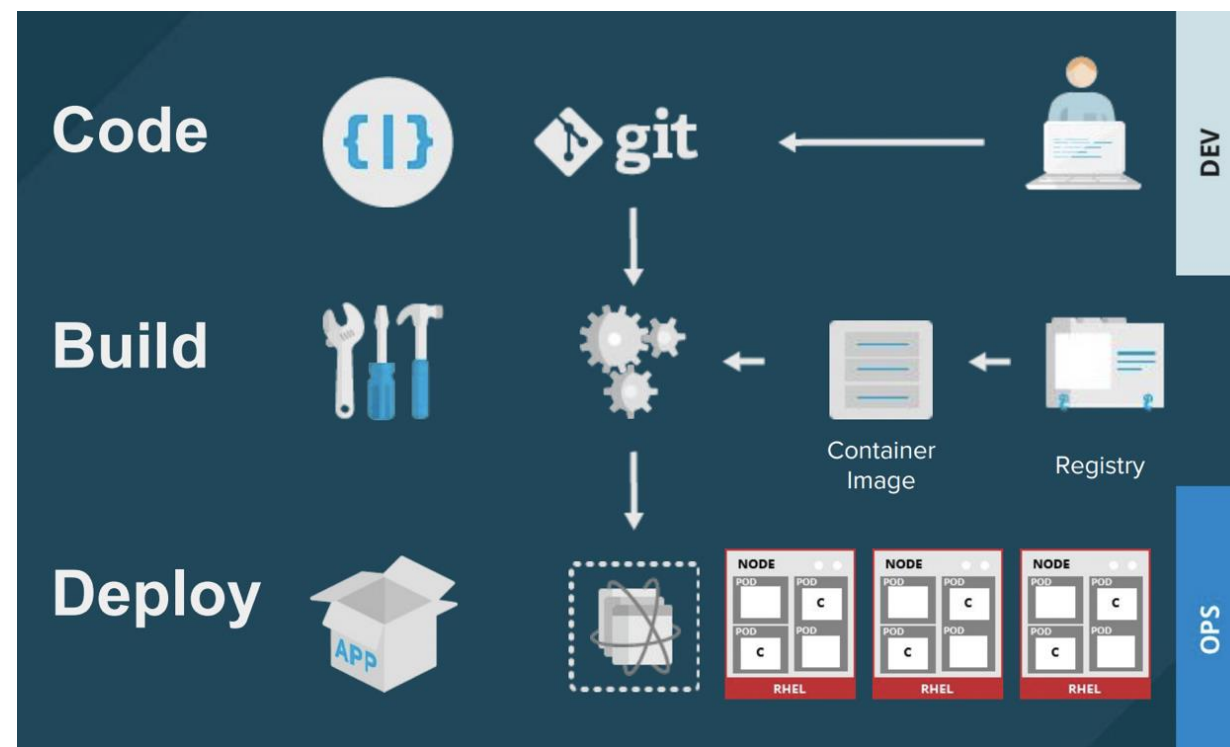
OpenShiftはSource-to-Image (S2I)を使い自動でDockerイメージ構築プロセスを開始。S2IはDockerレジストリを基にContainerイメージを作成。ビルドイメージは手動でも自動でもGitWebhookを使って選択可能。その後ビルドパイプラインへ追加。

Deploy

OpenShift はKubernetesスケジューラを使用して複数のノードにアプリケーションコンテナを自動で展開。ユーザはアプリケーションをの設定、展開、ロールバックなど必要なアクションを自動で実現可能に。

アプリケーションにフォーカス

- デプロイ/スケール/アップデート/リソース管理
- アプリのコンテナ化、マイクロサービスの利点を活用
- CI/CD環境
- インフラとアプリをつなぎ合わせる



RedHat OpenShift Overviewより

NETWORKRELIABILITY ENGINEER (NRE) (NRE = NETOPS + SRE)



NREとは

- ソフトウェア開発アプローチをネットワークオペレーションへ適用
- 商用展開 & 運用における自動化に直結するパイプライン上で作業する
- 従来のエンジニアのタスク（構築、テスト、導入、運用）とSREのタスク（[DevOps](#)の実装）を組み合わせ

「どれだけ速く運転できるのか？」



「どうやって十分な速さで運転するか？」

NREの主な作業例

- ネットワークを [DevOps](#)パイプライン上に構築して導入
- ネットワークのダイナミックなステートに対して自動的に対処
- ワークフローの自動化、定型的な繰り返し作業を削減
- プロアクティブなテストでトラブルシューティングを自動化
- 自動化された対処で信頼性を設計
- エラーの見込み(許容)とサービスレベル目標をトラックし監視する *)

* Service Level Objectives/Agreements/Indicators (SLO/SLA/SLI), MTTR, MTBFを継続的にモニタ&トラッキングし改善を行います。ネットワークの一部の信頼性だけではなく全体システムの信頼性向上を透過的かつ俯瞰的に管理する

NREのメリット

信頼性

- NRE の最大の価値は信頼性です。技術が進歩するスピードとビジネスのスピードは経済性の点で重要ですが、信頼性というベースがなければ役に立ちません。DevNetOps の原則では、段階的な小さな変更による進化とスピードを重視しているので多くの製品には進化型アーキテクチャの俊敏性が進んで取り入れられます。

障害回避

- NRE では、負荷がかかった場合にネットワークのパフォーマンスがどのように低下して切断につながるかを徹底的に解析 & 理解することで、インシデント対応を自動化して定型化することができます。これによりサービス障害を防止するためのプロアクティブなアプローチを確立できます。

高いモチベーション

- NRE により、各リソースレベルで導入の懸念が減少し、仕事に対する満足度が向上することが報告されています。

運用管理のシンプル化

- 全体的には、NRE により、シンプルな運用と管理が実現します。ネットワーク運用は、制御、保護、監査に対する多くのダイナミックな要素が存在するため、非常に複雑になります。NRE では、信頼できる情報のコード化と自動対応することで複雑さを解決し [Self-Driving Network](#) を目指しています。

SDN, NFVとの関係性

• NRE、SDN、NFV の関係

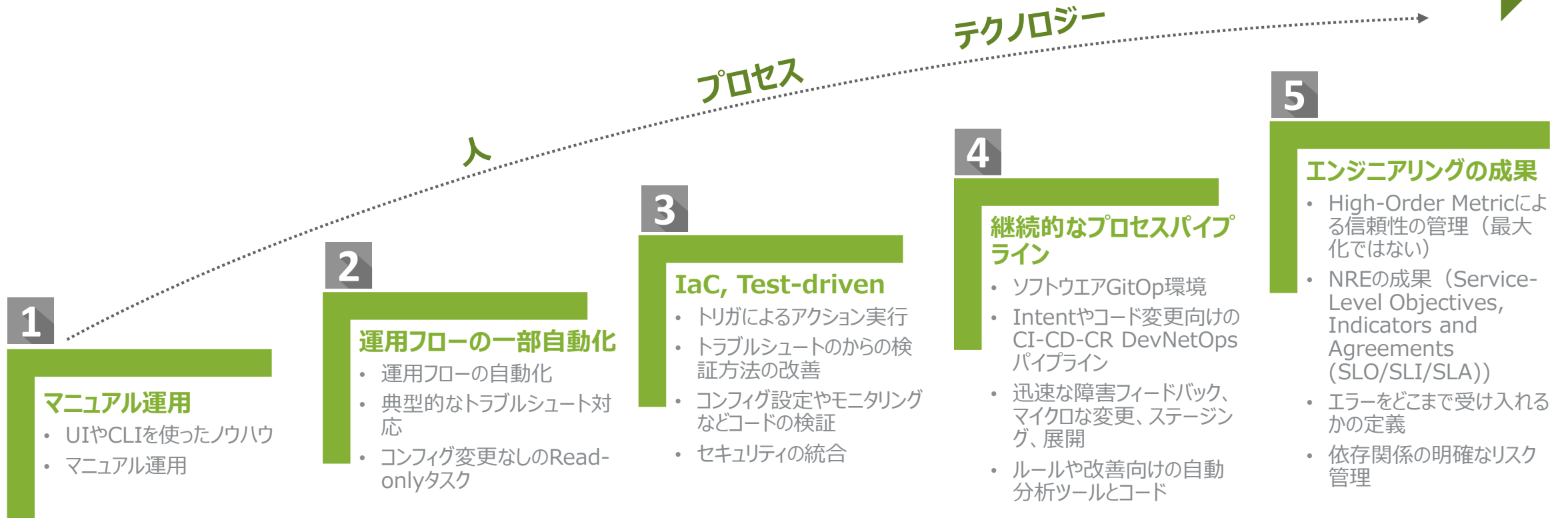
- SD, NFV環境へNREを適用することは非常に親和系が高い。SDNシステムでは特定のドメイン内のネットワーク アーキテクチャ全体の制御と設定を自動化して抽象化するためNRE を取り入れることは簡単です。もちろん、NRE を取り入れるために再度システムを作り直す必要はありません。

• エラーへの許容

- SRE や DevOps と同様、NRE と DevNetOps でも、失敗の許容を重視する傾向があり、その結果、修正が迅速に行われ、教訓が得られる
- Kaizen（改善）はバランスが取れていることではなく、バランスを回復することです。最初に失敗が許容されなければ回復することはできない（多くの現地調査からこのアプローチがより良い結果につながることが判明している）
- NRE が目指しているのは完璧さではなく進化。NRE はエンジニアリング期間と運用期間を分けて、失敗の把握、教訓の記録、継続的な改善を、ツールと自動化プロセスに継続的に取り込みます。
 - 属人的なノウハウ蓄積をシステム化する

自動化されたNETOPS の5ステップ

Network Reliability Engineer/ing (NRE)



目指すゴール



エンジニア:
Network Reliability
Engineers (NRE)

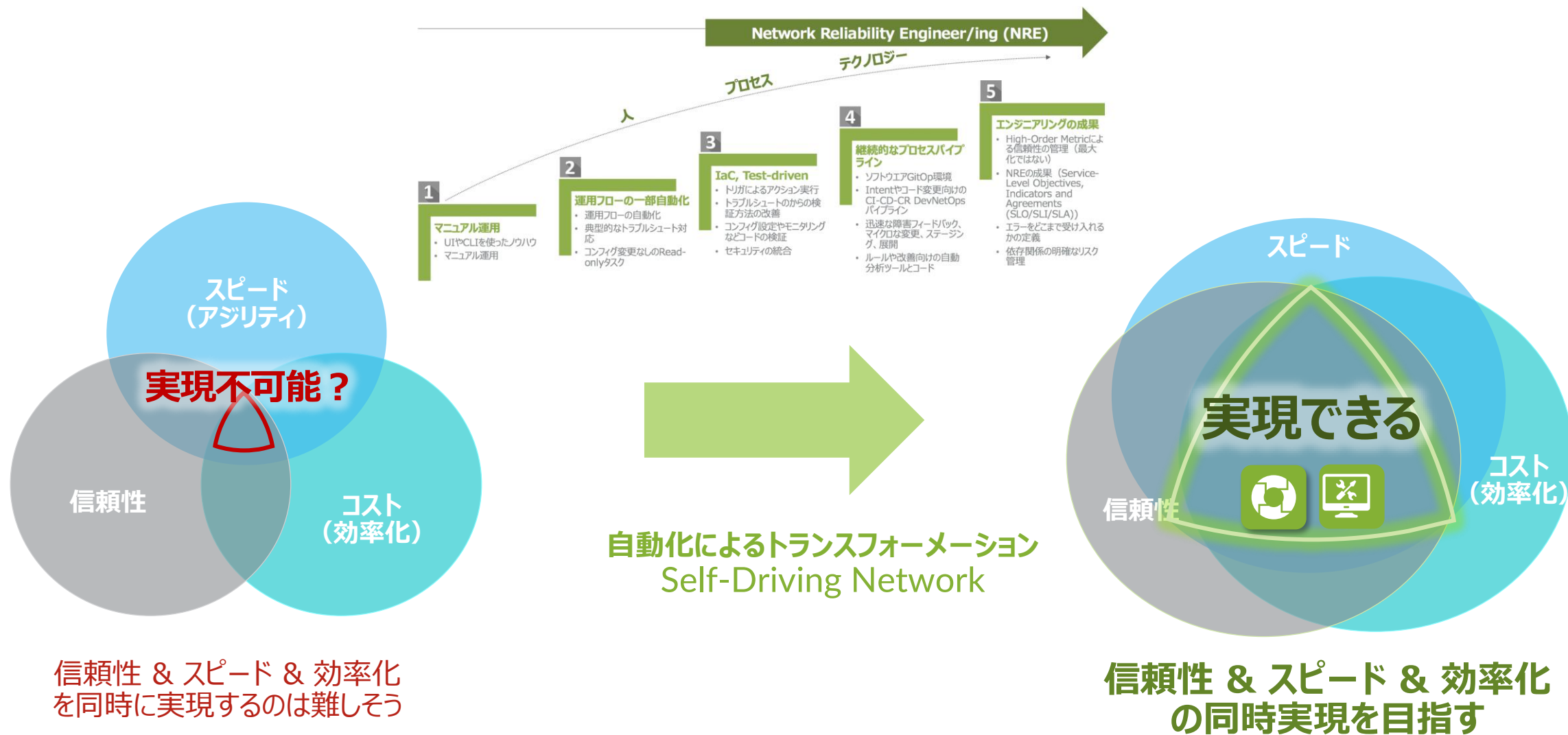


プロセス:
“DevNetOps”
と NRE'ing



テクノロジー:
抽象化, 自動化
自律化した NetOps

信頼性 & スピード & コスト削減を実現するために



A low-angle, black and white photograph of a modern glass skyscraper, showing the intricate grid of windows and structural elements reaching towards the sky.

AGENDA

- マーケット・トレンド
- Intent-Drivenとは
- Network Reliability Engineer (NRE)とは
- TelcoCloud
- その他
- 最後に

TELCO CLOUD がめざすもの

プロダクト先行型 → サービス・ドリブン型

	現在のオペレーション	将来のオペレーション
新機能開発期間	6-24 か月 サプライヤー	2 か月 サプライヤー、SP
ソフトウェア品質(品質安定に要する期間)	最大6度目のリリース	1度目のリリース
リリース導入検討からデプロイまでの期間	3-5年	3-6か月
検証自動化(準備、分析、実行、文書化)	40%	100%
ソフトウェアの有効性確認	数日 平日日勤帯	数秒 7 x 24
ソフトウェアアップグレード実施	夜間のメンテナンスウィンドウ	日中・サービス提供中
変更管理の自動化	10%	100%
CPU,メモリの最高稼働率	30%	80%
リリースを導入するのに要するコスト	5 年, 10人	6か月, 5人

サービスドリブン : IaC と DevOps

1

INFRASTRUCTURE AS CODE



Git Repository

Inventory

Ansible playbooks and roles

Test suites

Pipelines

Software Ansible Artifacts



2

継続的な自動化デプロイとライフサイクルマネジメントを実現するためのDevops

Code

Build

Test

Deploy

Monitor

CI

Continuous Integration



CD

Continuous Delivery

Build

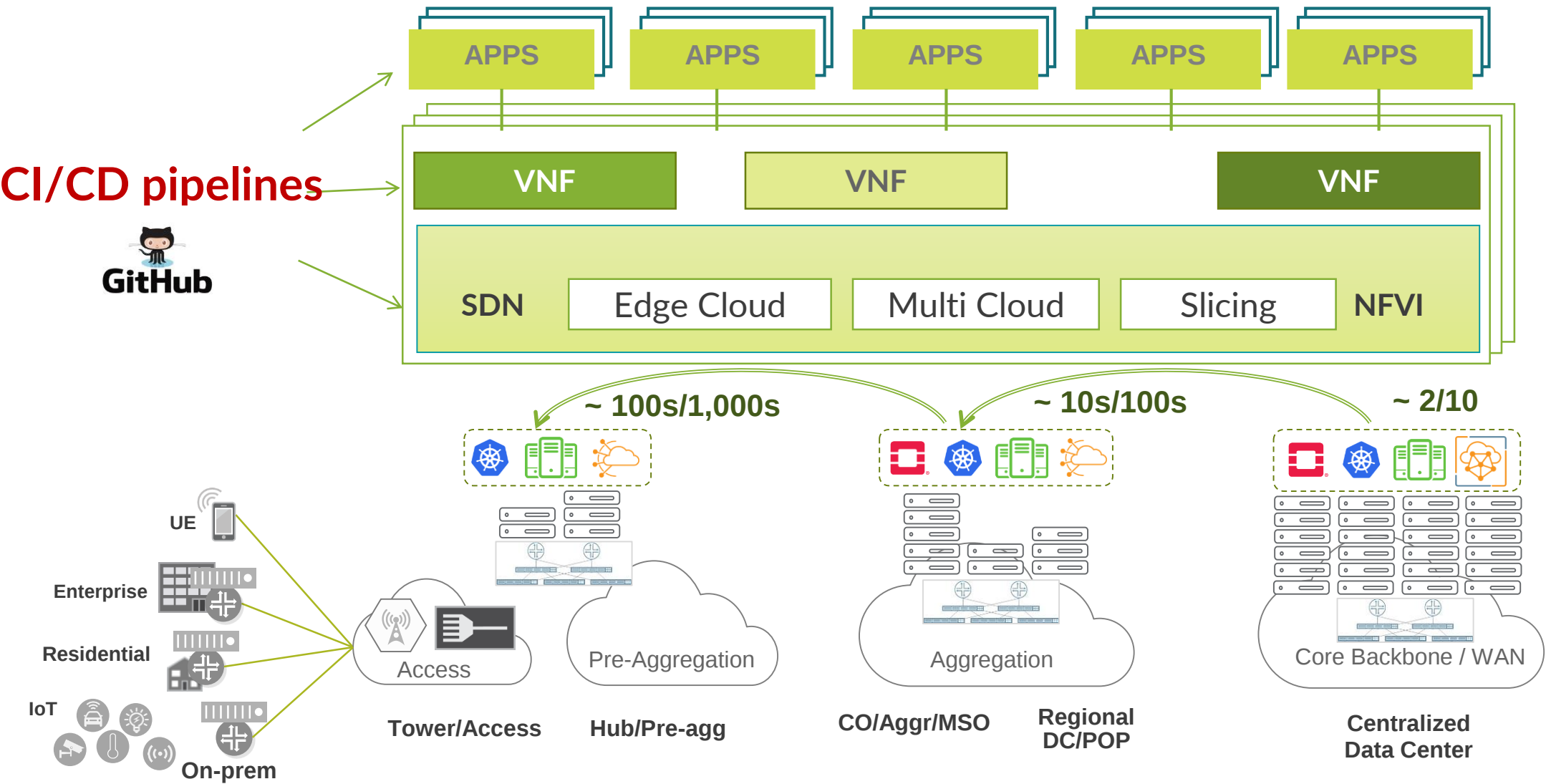
Integration Test

Code Review

Staging

Production

エッジクラウド上のアプリケーションのライフサイクルを自動化するDEVOPS

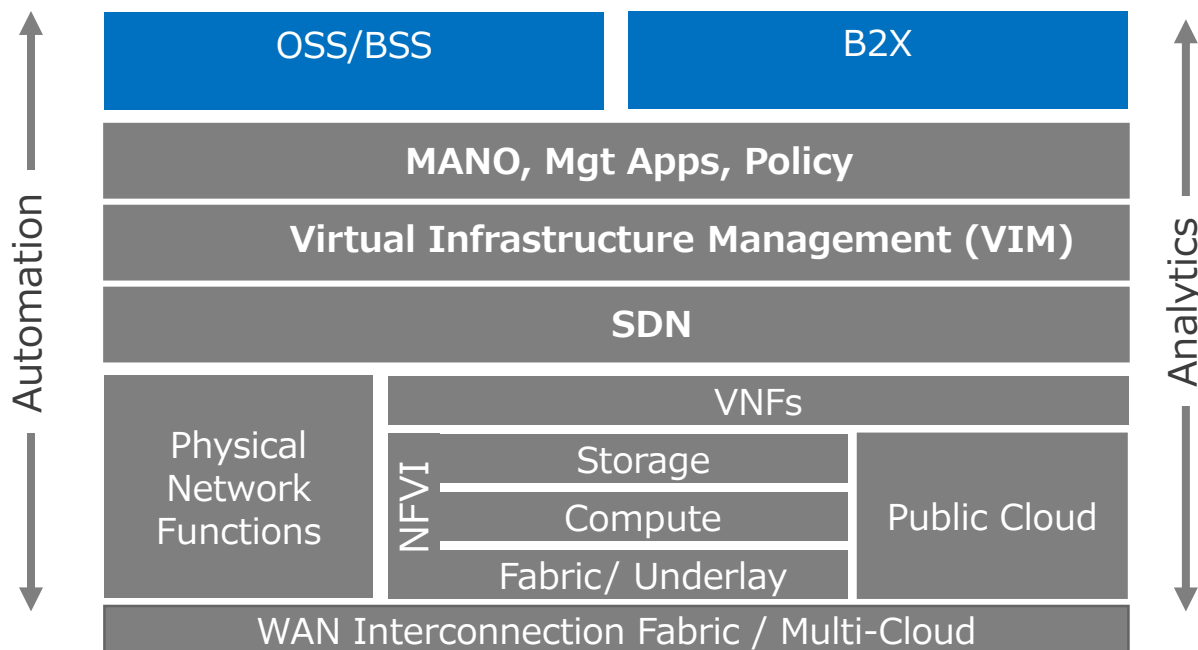


“TELCO CLOUD”とは？

TELCOネットワークとオペレーションにクラウドのDEVOPS特性を適用

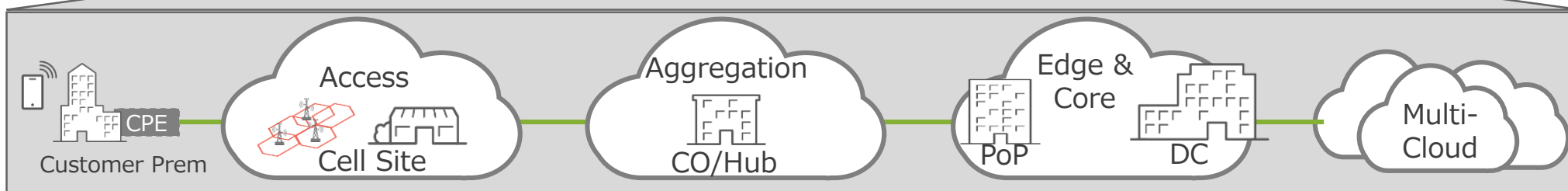
Telco Cloudの利点

- ネットワーク代替性
- インフラの最適化
- 提供サービスの**アジリティ**・イノベーション
- オペレーション**効率化**
- 特有の要件への対応
- **柔軟な**ビジネスモデルへの対応

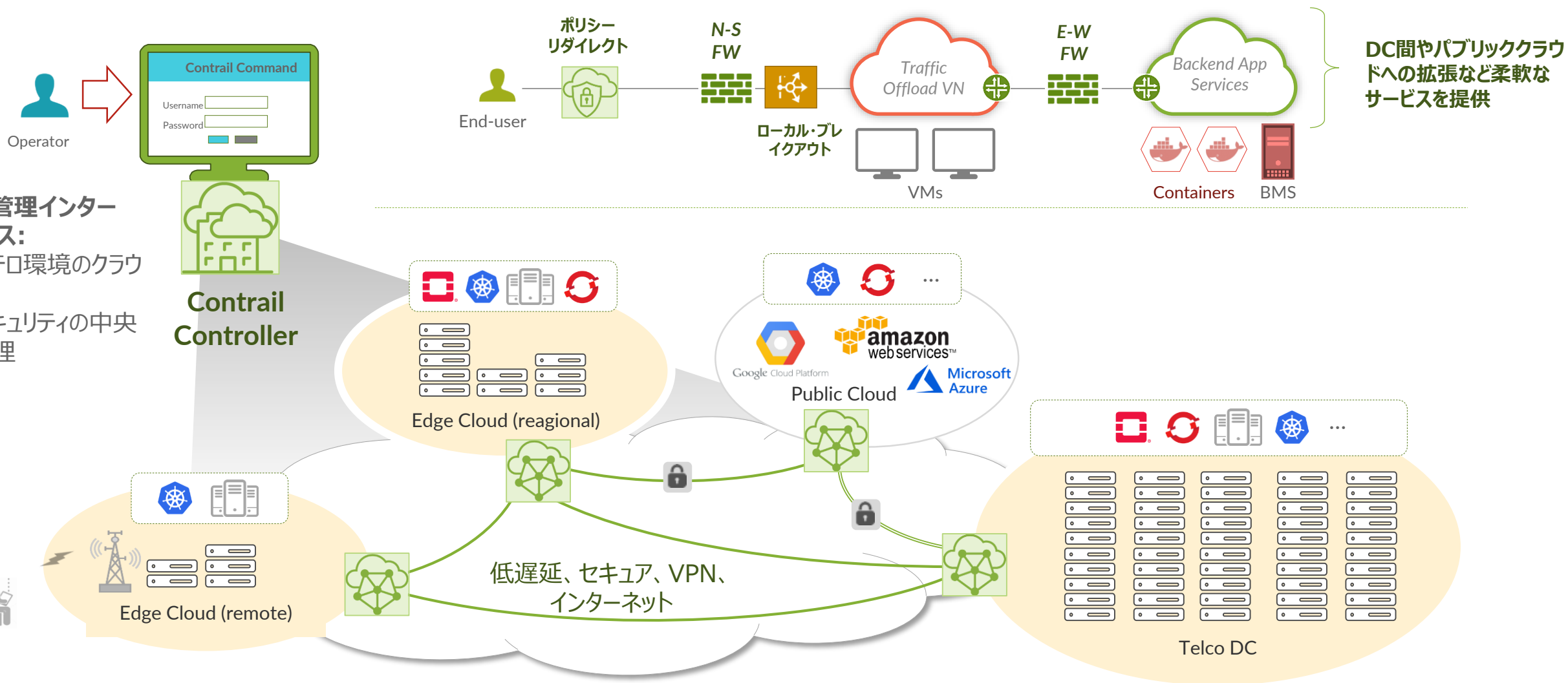


Telco Cloudがもたらすもの

- 仮想化
- SDN化され拡張に優れたインフラ
- **インテント**ベースの管理・**自動化**
- **分析**結果を利用したインテリジェンス
- ネットワークとサービスの**抽象化** (**APIs**)



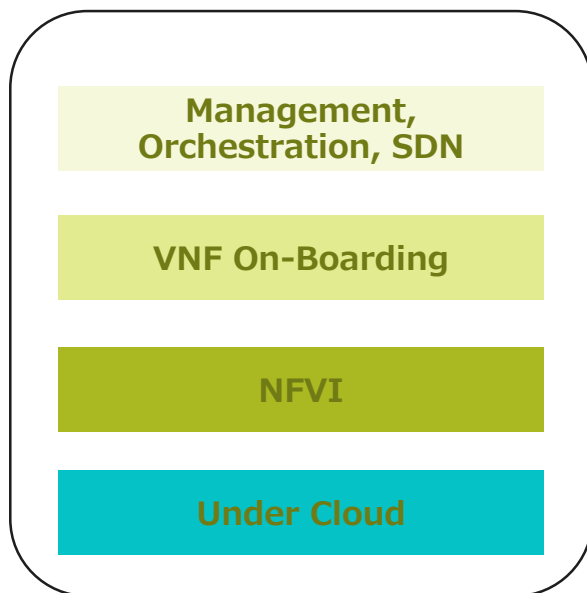
MULTI-CLOUD インフラ – アプリケーション・ドリブン・インフラ



JUNIPER TELCO CLOUD ソリューション

水平統合するNFV基盤ソリューション オープン / 自動化 / スケール / アジャイル

ソリューションスタック



Secure. Automated. Cloud.

VIM

VNF

NFVI



サービスオーケストレーター、SDNコントローラー、

Contrail Networking/Security/Edge、AppFormix



VNF On-Boardingの自動化

Juniper VNFs

3rd Party VNFs

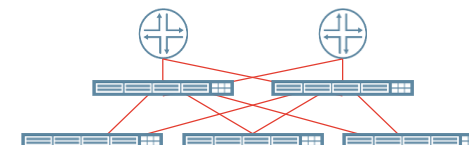
Use cases



VNF エコシステム
(vEPC, 5GCN, vIMS, vRAN, etc)



スケール&ステイブルな アンダーレイ・オートメーション



MX



QFX





AGENDA

- マーケット・トレンド
- Intent-Drivenとは
- Network Reliability Engineer (NRE)とは
- TelcoCloud
- その他
- 最後に

AUTOMATIONにおける確固たる自信 — JUNOS

8 features you need for automation

“I would characterize the transformation that we are seeing in this industry with one word, and that is **AUTOMATION**”


Rami Rahim, CEO at Juniper Networks

The independent and respected industry expert Ivan Pepelnjak identified 8 features for your automation RFP. **How did he rate Juniper?**

1 On-device APIs to read / write configuration and operational data

✓ Juniper Networks: Pass

2 Structured operational data for easy programmatic analysis

✓ Juniper Networks: Pass

3 Structured device configuration data for easy programmatic analysis

✓ Juniper Networks: Pass

4 Atomic configuration changes to avoid partial updates

✓ Juniper Networks: Pass

5 Configuration rollback to minimise risk

✓ Juniper Networks: Pass

6 Full configuration replace that makes templates easy to use

✓ Juniper Networks: Pass

7 Configuration difference analysis to simplify manual approvals

✓ Juniper Networks: Pass

32
8 Industry standard data models for configurations

✓ Juniper Networks: Pass

Juniper:
100% pass rate

Check the original blog to see how other vendors compared

[Read the blog](#)

JUNOS EVO – さらなる革新へ

Openness

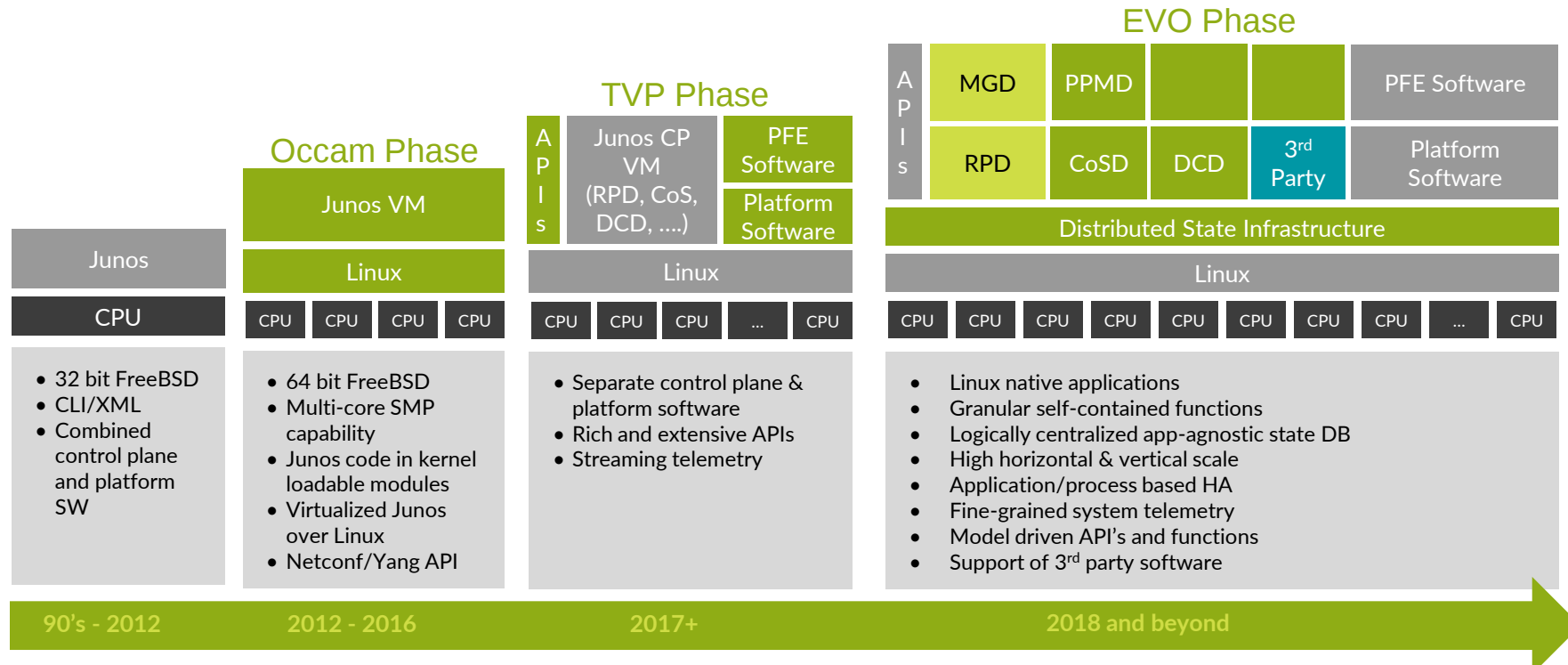
- Linux ネイティブプラットフォームとアプリ
- 3rd party アプリとツールのサポート

モジュラリティ

- コンポーネントレベルの設計と冗長性
- コンポーネントレベルのHitless Upgrade

Logically Centralized Database

- すべてのステートはAPIで管理
- Pub-sub communication between components
- コンポーネント毎の障害分離
- 大幅に改善された診断機能による迅速なデバックを実現



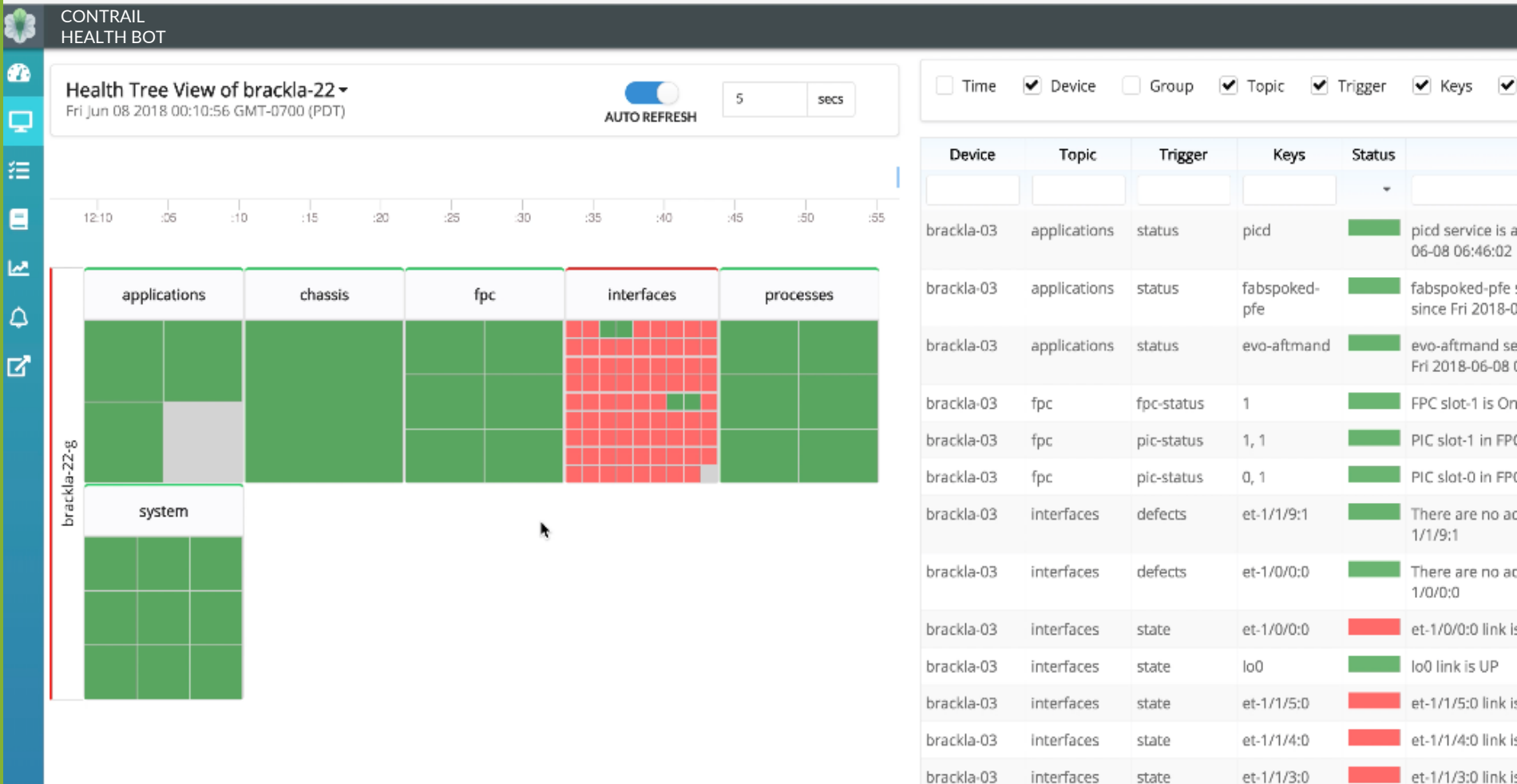


CONTRAIL HEALTHBOT

ネットワークのヘルスチェック と自動化連携ツール

- 問題の本質究明
- SLI/SLO ダッシュボード
- 詳細なデータ分析による
UXの見える化
- トレンドの検知

無償テスト環境：
vlabs.juniper.net

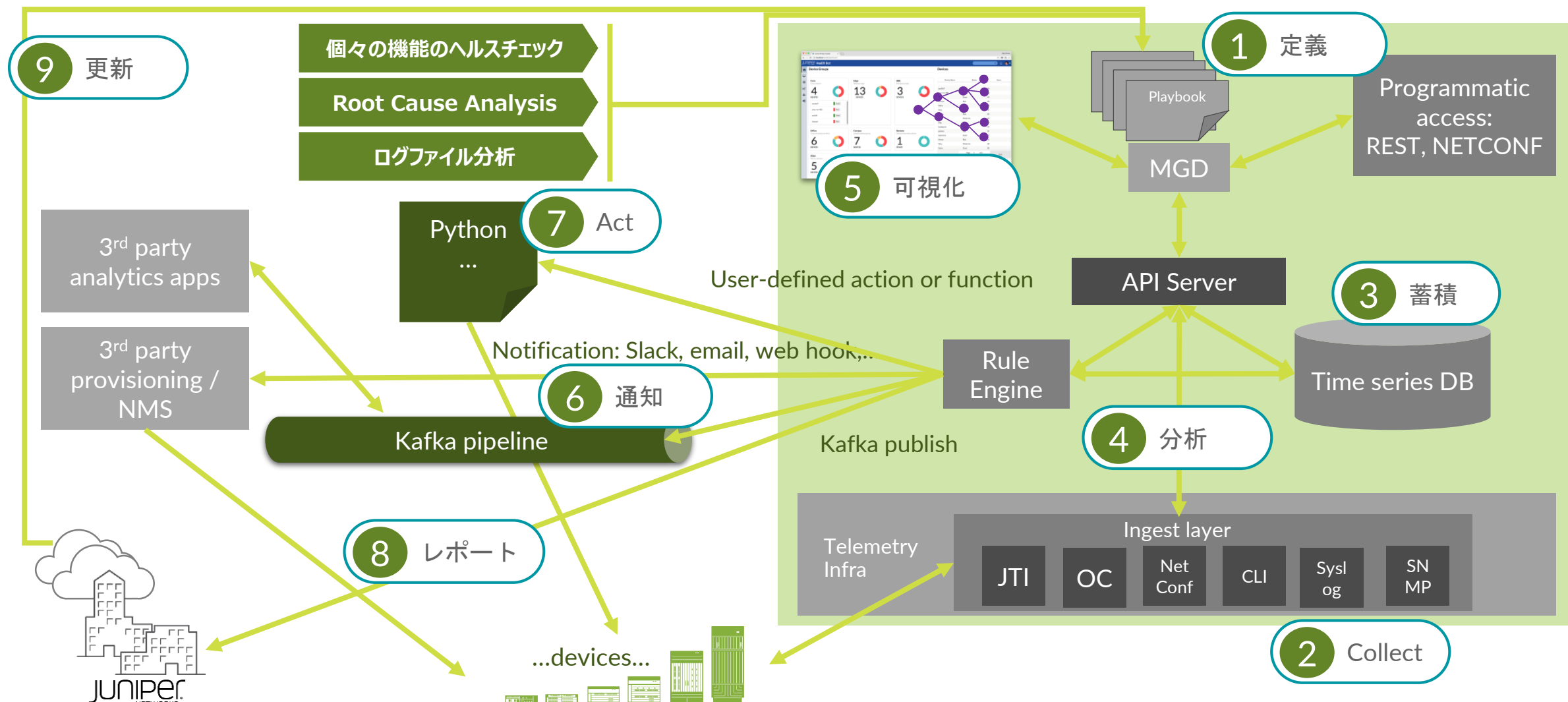


- テレメトリなど詳細な可視化
- アラームとリアクション
- トラブルシュートの定型化とノウハウの蓄積
- RCA処理プロセスのルール
- GitHub上でのPlaybook、RCA公開と蓄積（コミュニティ）



Telemetry, Syslog, SNMP, OpenConfig, NETCONF, JTI, gRPC, gNMI

CONTRAIL HEALTHBOT — CLOSED-LOOP 自動化アーキテクチャー



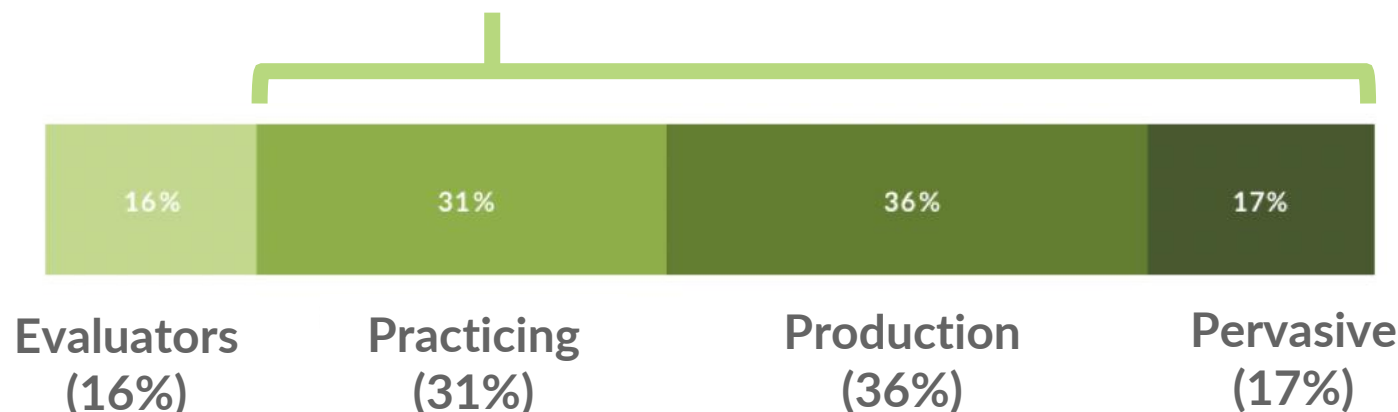


AGENDA

- マーケット・トレンド
- Intent-Drivenとは
- Network Reliability Engineer (NRE)とは
- TelcoCloud
- その他
- 最後に

2019 STATE OF NETWORK AUTOMATION REPORT (SONAR)

Enterpriseユーザの **84%** がNetworkAutomationへ本格的にシフト



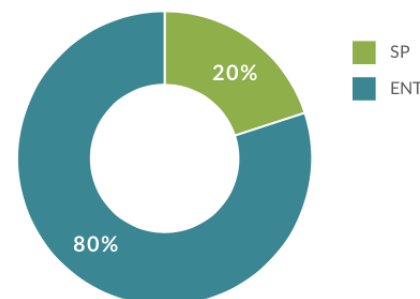
Evaluators : CLIかGUIのみのオペレーション (4%)
or 自動化に着手し始めたばかり(12%)

Practicing : 自動化は進めているがまだテスト環境のみ

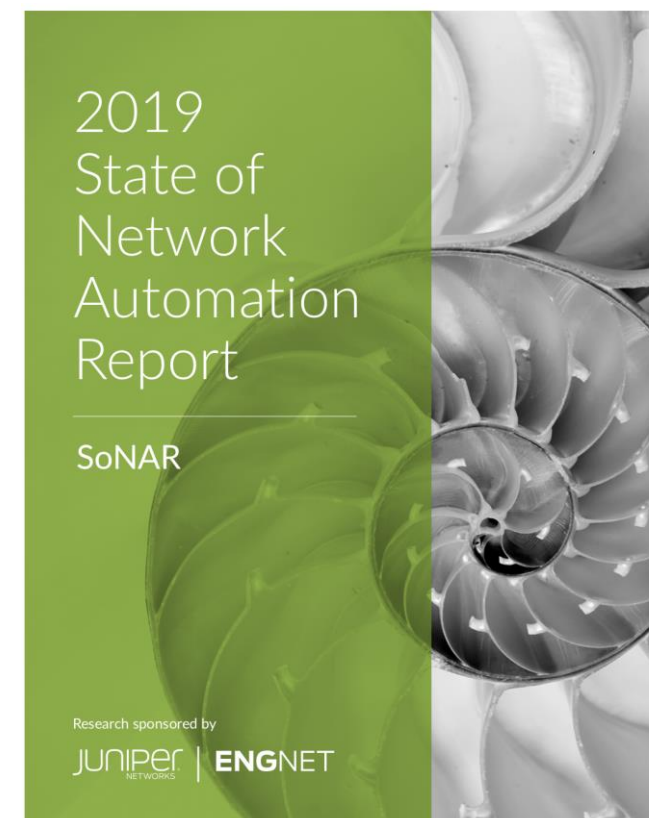
Production : 一部のNWで自動化導入済み

Pervasive : NW全体で自動化導入済み

Service Provider vs. Enterprise



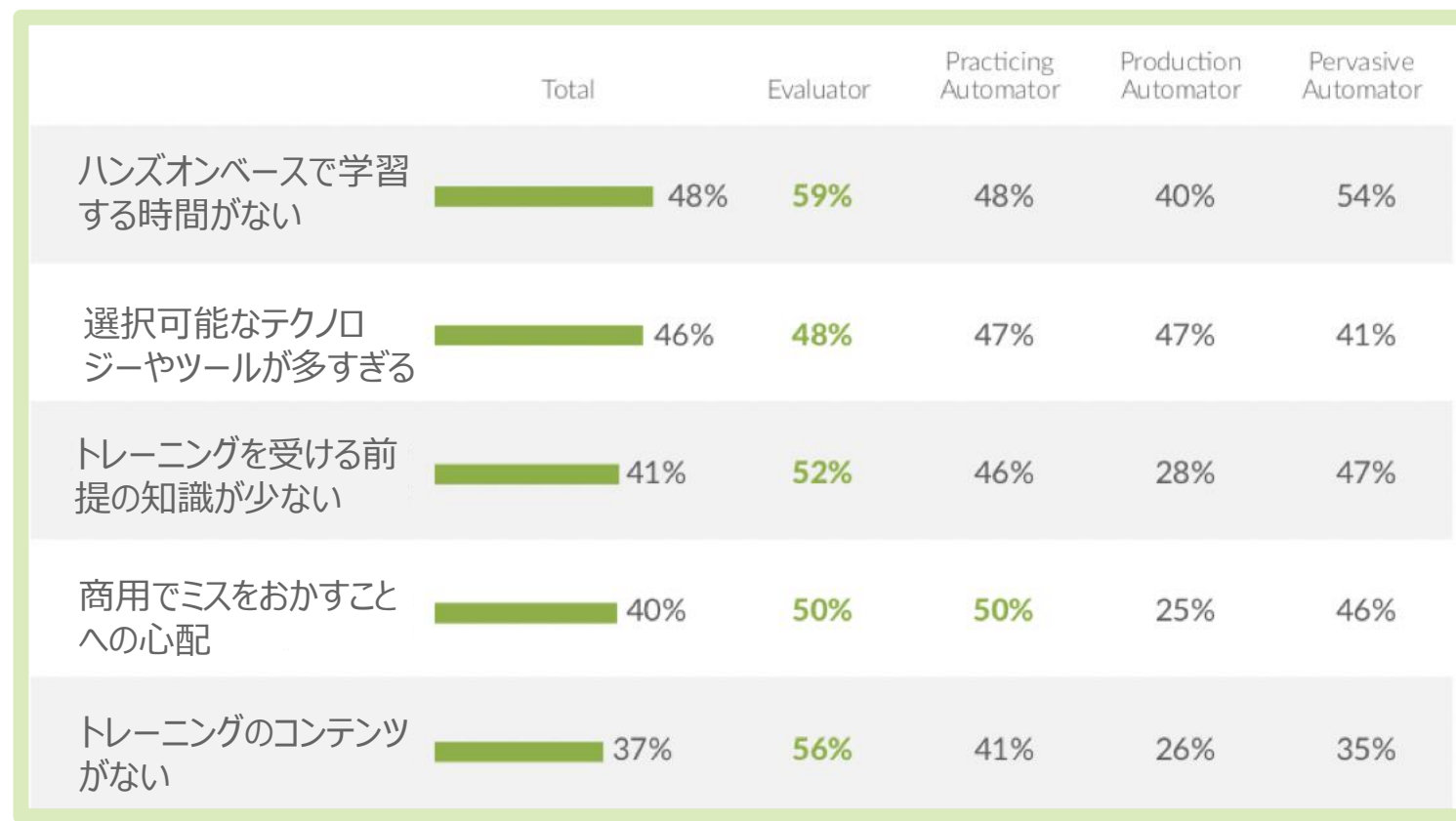
SONaRレポート



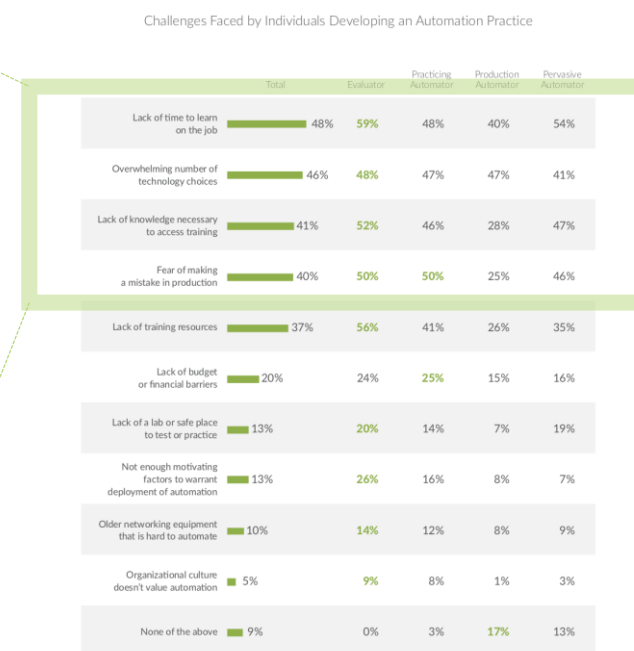
<https://www.juniper.net/us/en/forms/2019-sonar/>

2019 STATE OF NETWORK AUTOMATION REPORT (SONAR)

担当エンジニアが自動化実現に直面する際の難題（トップ 5）



担当エンジニアが自動化実現にあたっての難題



NRE LAB



(Juniperスポンサー)

<https://labs.networkreliability.engineering/>

- 無償
- アカウント登録不要
- ブラウザでハンズオンもトレーニングも完結
- NRE/DevOpsが初めてのネットワークエンジニア向けの学習コース提供

The screenshot displays the NRE LABS website. The header features the 'NRE LABS' logo and a navigation menu. The main heading is 'Automation. Unshackled.' with a subtext 'Learn automation by doing it. Right now, right here, in your browser.' Below this, a paragraph describes NRE Labs as a no-strings-attached, community-centered initiative. Two buttons, 'Watch Intro Video' and 'Find Lesson Content', are visible. The page is divided into three columns highlighting key features: 'Not just for experts.', 'No paywalls here.', and 'Open. Like, for real.' The 'Lesson Catalog' section is organized into three categories: 'Fundamentals', 'Tools', and 'Workflows', each with a list of specific lessons.

NRE LABS

Automation. Unshackled.

Learn automation by doing it. Right now, right here, in your browser.

NRE Labs is a no-strings-attached, community-centered initiative to bring the skills of automation within reach for everyone. Through short, simple exercises, all right here in the browser, you can learn the tools, skills, and processes that will put you on the path to becoming a Network Reliability Engineer.

Watch Intro Video Find Lesson Content

Not just for experts.
NRE Labs assumes nothing about your background or training. Everything is contained in the browser, and there's something here for everyone.

No paywalls here.
This is first and foremost an educational tool aimed at helping the industry consume automation more readily. So you won't find any paywalls or creepy web trackers here.

Open. Like, for real.
NRE Labs is built on Antidote, an open source project published on Github under an Apache v2 license to make industry-wide participation a first-class citizen.

Lesson Catalog

Already know what lesson you're looking for? That's awesome - all lessons are linked directly below.

Fundamentals	Tools	Workflows
These lessons are aimed at teaching the fundamental technologies used needed by automation tools and NRE processes	These lessons will introduce you to the specific tools you're likely to use in your NRE journey.	These lessons will bring it all together and apply fundamental skills and tools experience to accomplish a specific outcome.
Introduction to YAML	Network Unit Testing with JSNAPY	Automating the Troubleshooting Chain
Using Jinja for Configuration Templates	Multi-Vendor Network Automation with NAPALM	Automated STIG Compliance Validation
Version Control with Git	Event-Driven Network Automation with StackStorm	Quick and Easy Device Inventory
Working with REST APIs	Junos Automation with PyEZ	Automated Device Configuration Backup
Linux Basics	Using Robot Framework for Automated Testing	Device Specific Template Generation
Introduction to BASH	Network Automation with Salt	
	Terraform & Junos	

ENGINEERING SIMPLICITY : シンプル化をエンジニアリングする

JUNIPER
NETWORKS

Engineering
Simplicity

シンプル化



信頼性



アジャイル



サービス



効率化



コスト削減

自動化は手段であり
目的ではない

インテントはビジネスの目的プロセスを抽象化して
Actionに橋渡しをする

経営層の決断が必須
(プロセス統合判断の境界を超える)

一朝一夕では実現できない
まずは1プロジェクトから
複数フェーズアプローチ

ご清聴ありがとうございました

JUNIPER[®]
NETWORKS

Engineering
Simplicity