



APSTRA GENERAL PRESENTATION

Juniper Networks, K.K.

2025/4

JUNIPER
NETWORKS | Driven by
Experience™



Agenda

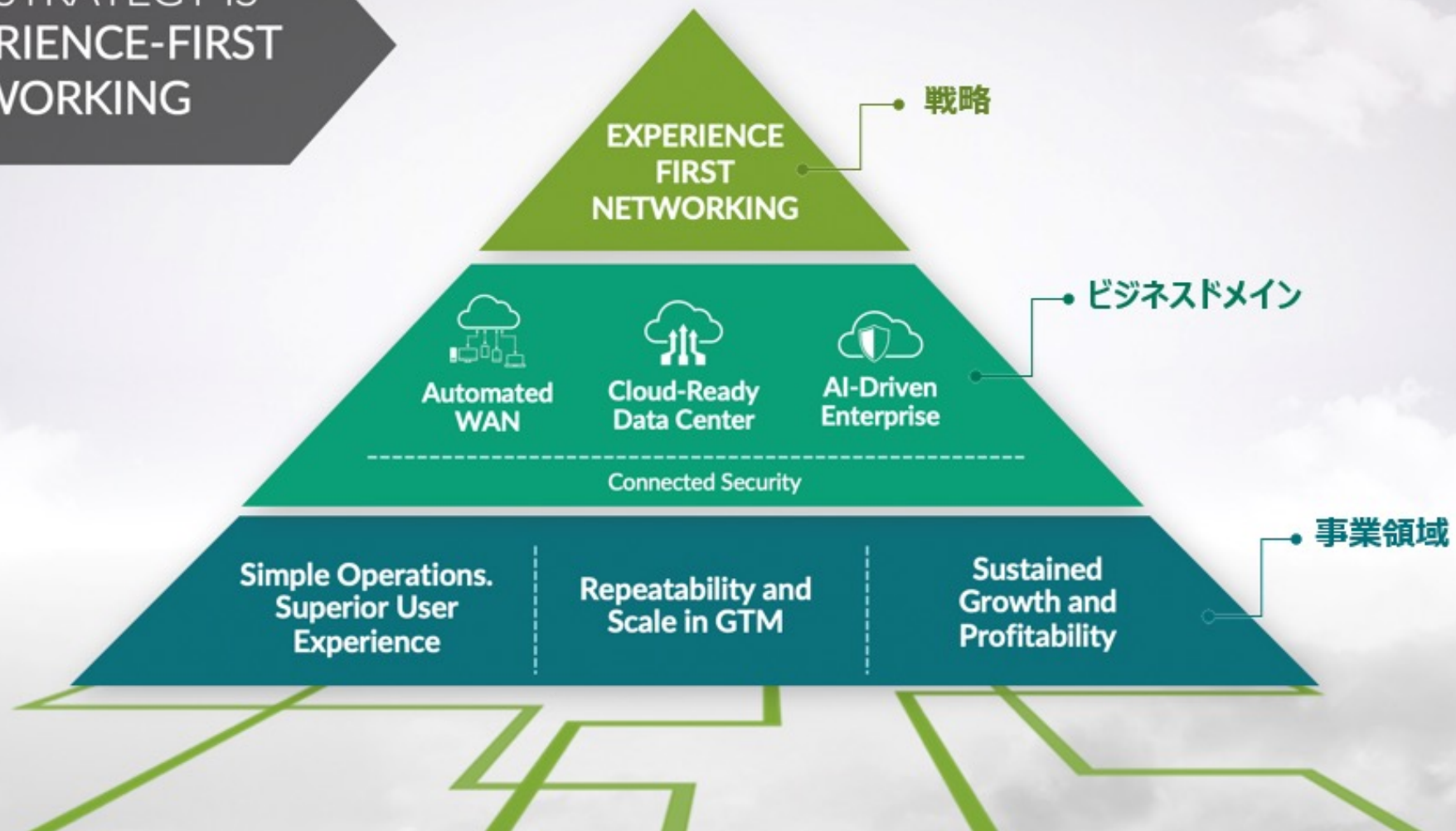
- 製品コンセプト
- JUNIPER APSTRAのご紹介
 - 製品概要
 - デザインパターン(DC Reference, Freeform)
 - 機能紹介(IBN, ACS, AI cluster etc.)
 - コンポーネント
- オペレーションサンプル
- 購買情報
- お役立ち情報

製品コンセプト



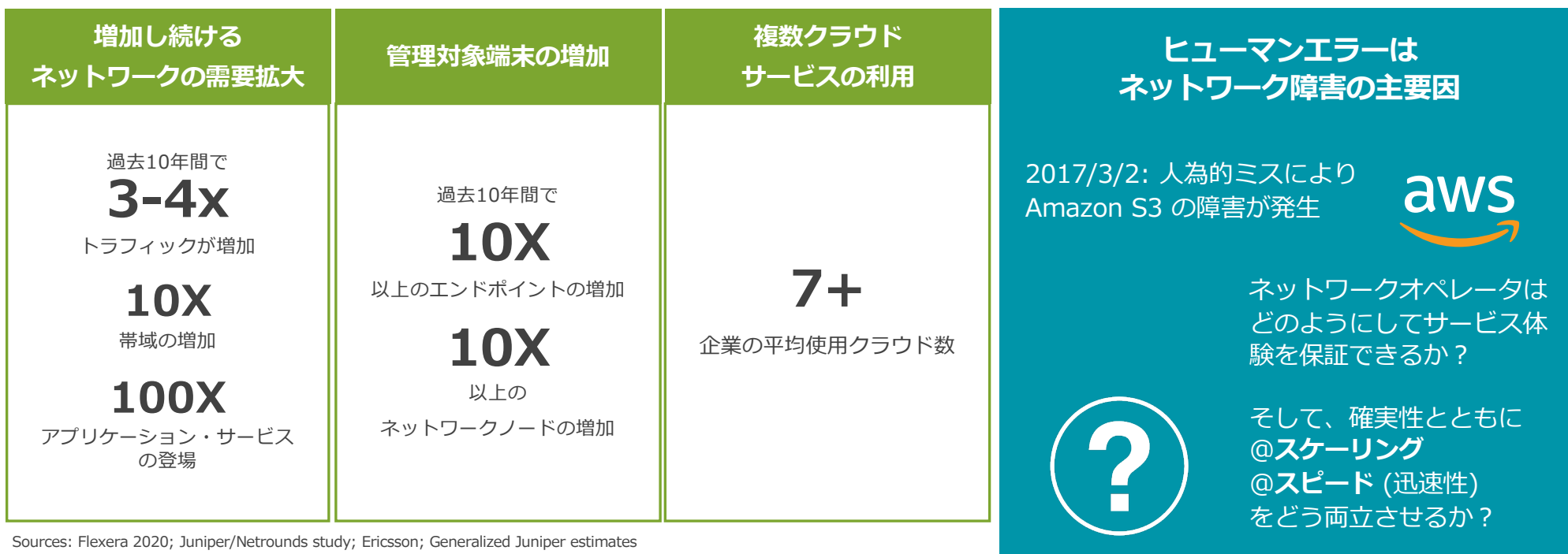
Juniper Apstra™

OUR STRATEGY IS
EXPERIENCE-FIRST
NETWORKING



ヒューマンエラーに起因するネットワーク障害

ネットワークの大規模化とヒューマンエラーの発生率は比例

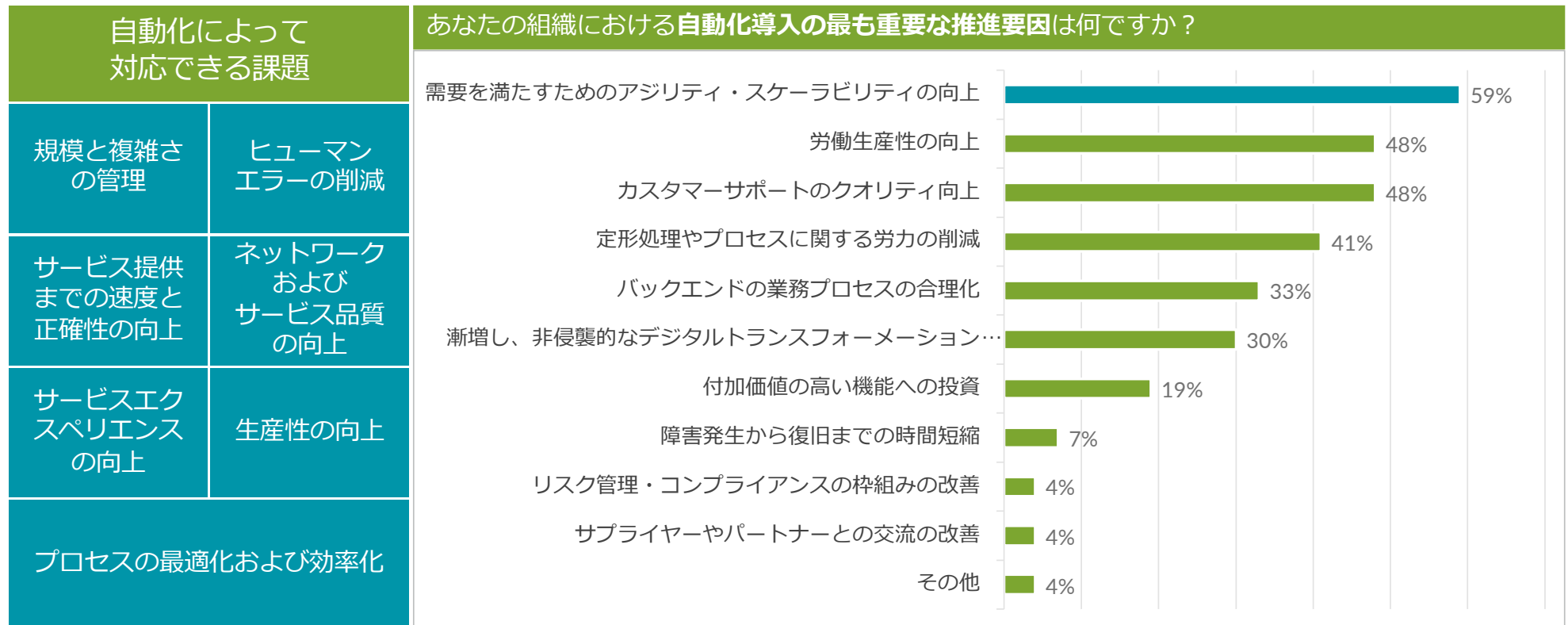


ネットワーク障害の60%はエンドユーザーからの報告か、全く報告されない
OPEXはCAPEXの倍の投資

Sources: : Forrester

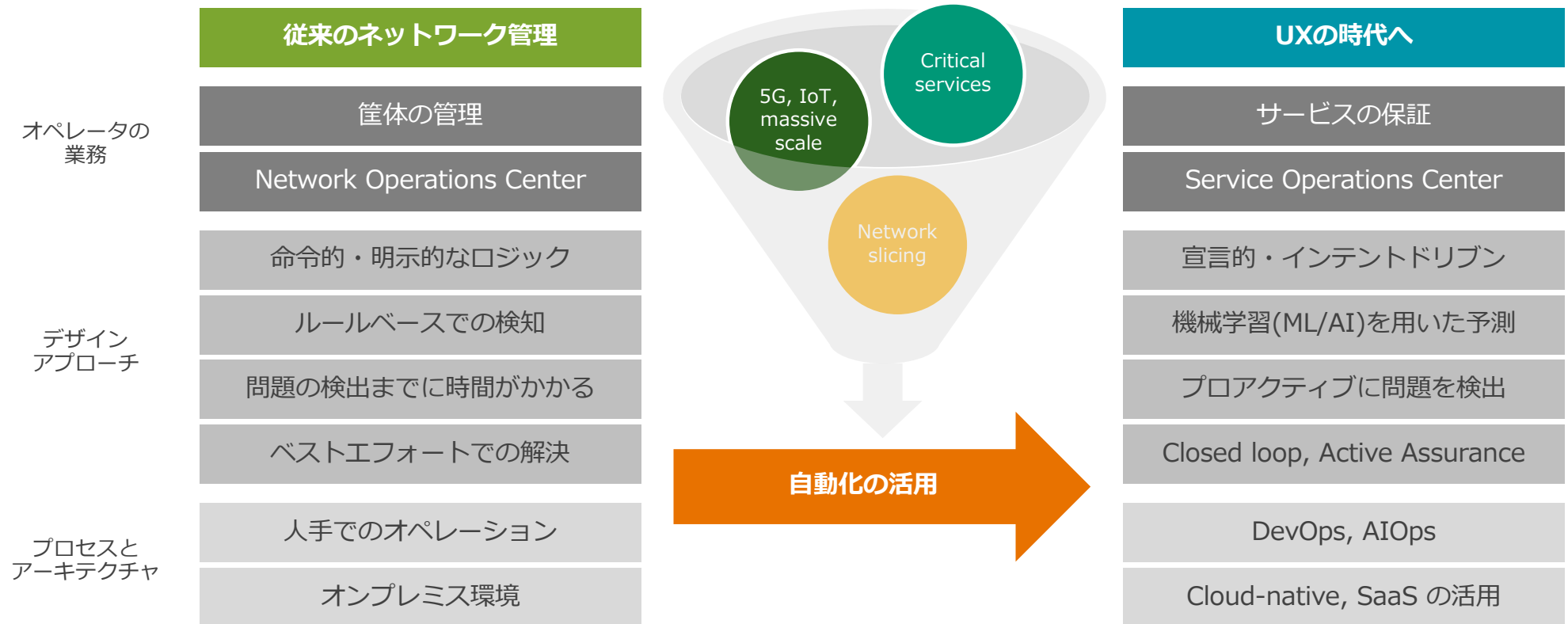
自動化によるヒューマンエラーの排除

自動化による生産性の向上が一層重要化

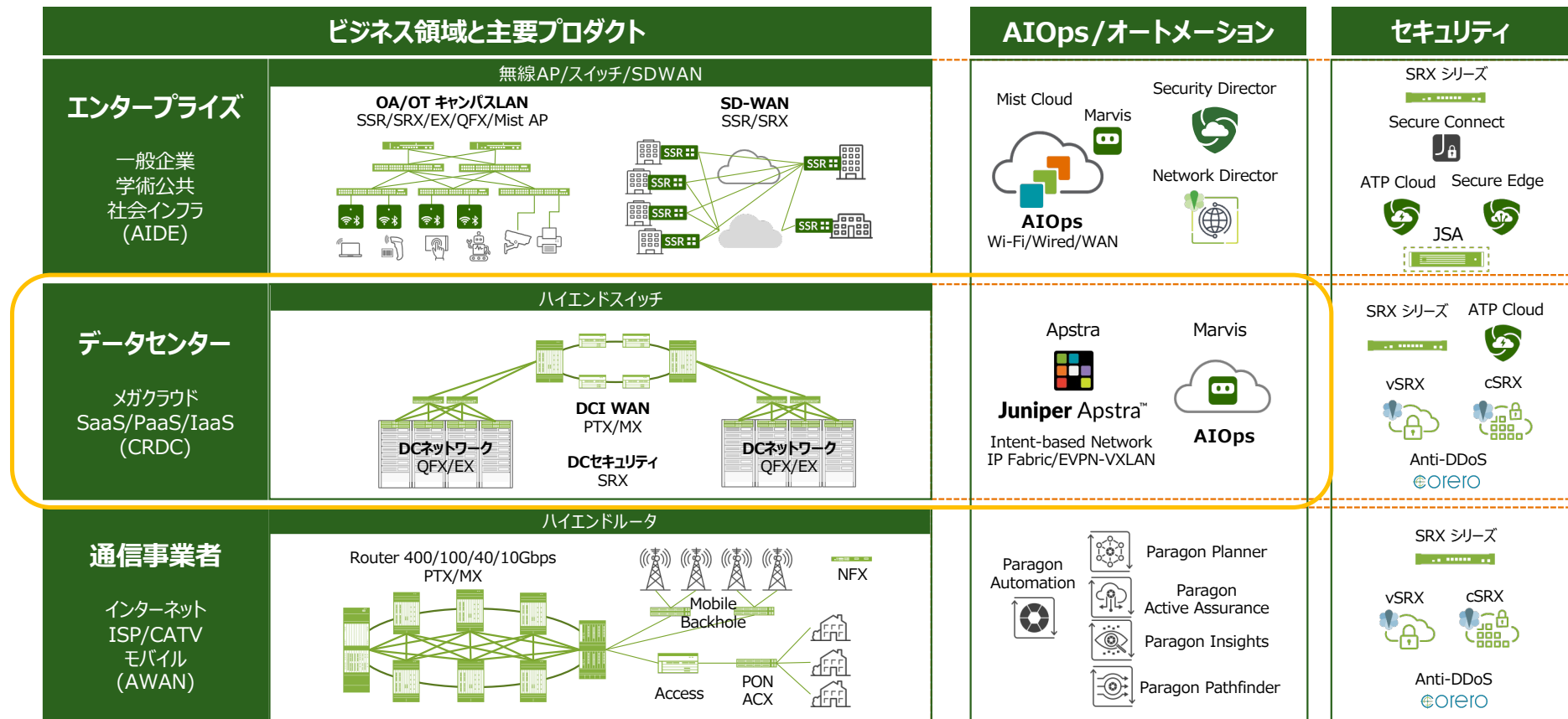


Source: Ernst & Young Global Telecommunication study 2019

Experience First Networking



Juniperのビジネス：ネットワークソリューション



Experience-First Networking データセンタ

オープンテクノロジー

Experience First

Engineering Simplicity



ユーザ体験の向上

Application Experience



運用者体験の向上

Operations Experience

有線系・DC ネットワークの期待



堅牢性と安定性



ヒューマンエラーの回避



即時の状態把握

堅牢性な・正常性確認にはこういったハードルがある？

・設計・運用・障害時それぞれに難しさがある



1. 設計、状態・把握が大変



- ・ 将来にも最適なネットワークの設計は難しく、拡張のたびに追加コストで設計検討
- ・ 設計に携わっていない人への引き継ぎは大変



2. 正常であるかの判断難しい



- ・ 何を持って正常と判断するか。クリティカルなログがない？
- ・ 利用者から連絡がなければ正常？



3. 人的ミスによる障害



- ・ 人が作業した時に毎回人に頼った動作チェックする必要がある
- ・ 設定ミスだけでなく、切り戻しも人的ミスが多い作業になる



4. ドキュメントの最新化が大変



- ・ ドキュメントは常に最新を反映しなければならず確認が大変
- ・ 更新コストも高い

Juniper Apstraは何をしてくれる？

- ・設計・運用・可視化を自動化し、高信頼のネットワークを利用可能に



1.設計・状態の把握が大変



- ・ GUIで設計・確認できる
- ・ Juniper検証済みの設計・コンフィグを自動生成



2.正常であるかの判断難しい



- ・ Apstraが自動的に必要な監視項目を監視
- ・ 異常の時は何が異常か即時に確認



3.人的ミスによる障害



- ・ GUI操作で都度、整合性の自動チェック
- ・ コンフィグの一括ロールバック



4.ドキュメントの最新化が大変



- ・ GUIから最新のトポロジー図、コンフィグ、結線表を入手



5.おまけ



- ・ マルチベンダ対応
- ・ GUIからデバイスOSアップグレードやconfig情報入手が簡単

Juniper Apstraのご紹介

- 製品概要



Juniper Apstra™

Juniper Apstra Intent-based Networking

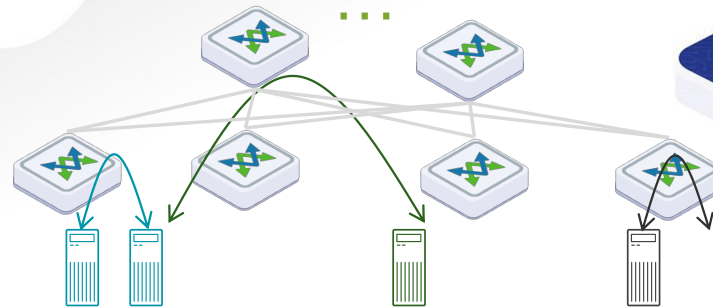
ネットワークを自動構築/監視/診断しエクスペリエンスの向上



Juniper Apstra™

マルチベンダ標準IP ファブリック

JUNIPER
NETWORKS
SONiC ARISTA CISCO



VMware vCenter®/ VMware NSX-T Data Center™ 連携

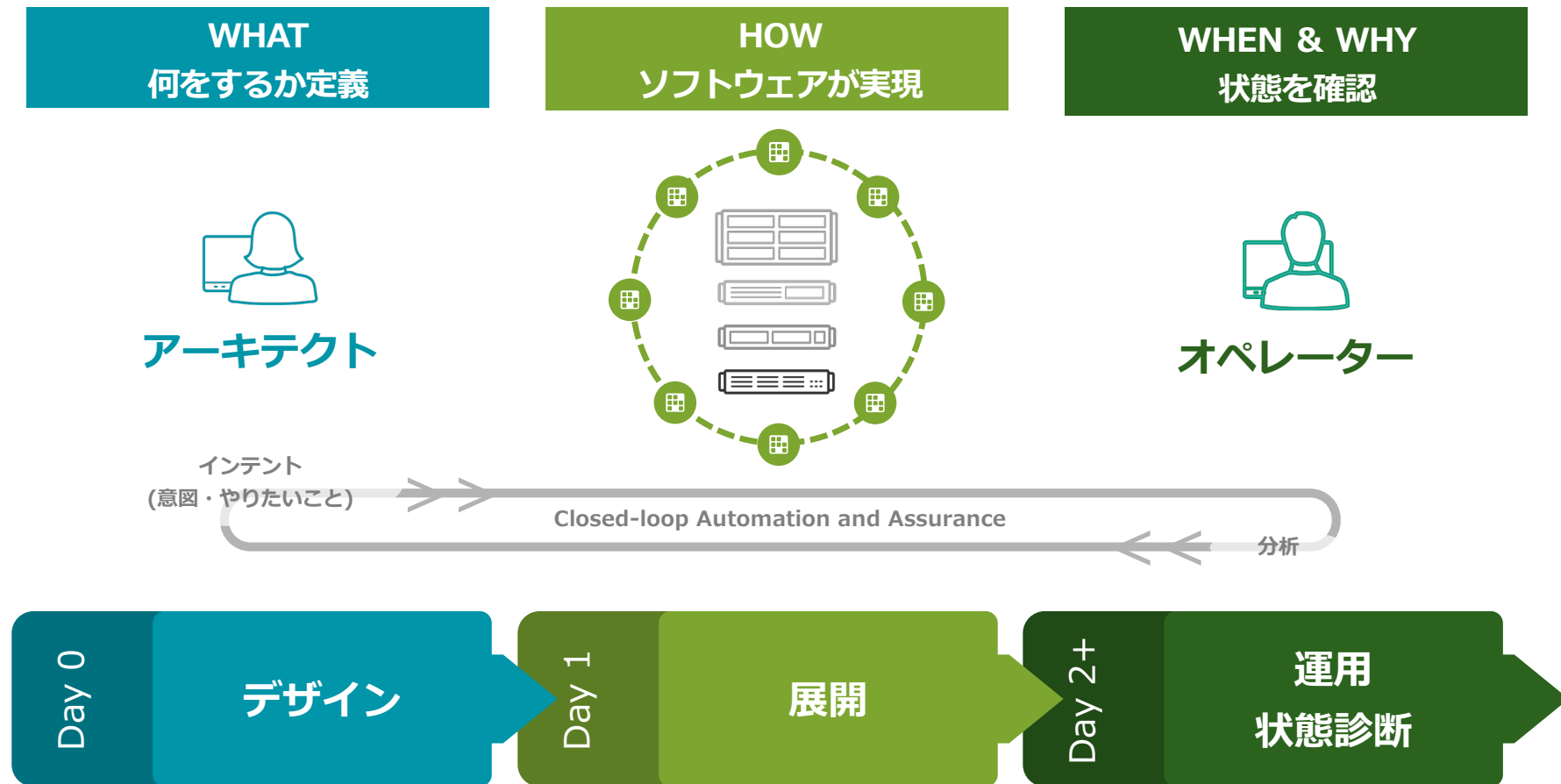


50-90% デリバリー
時間の短縮



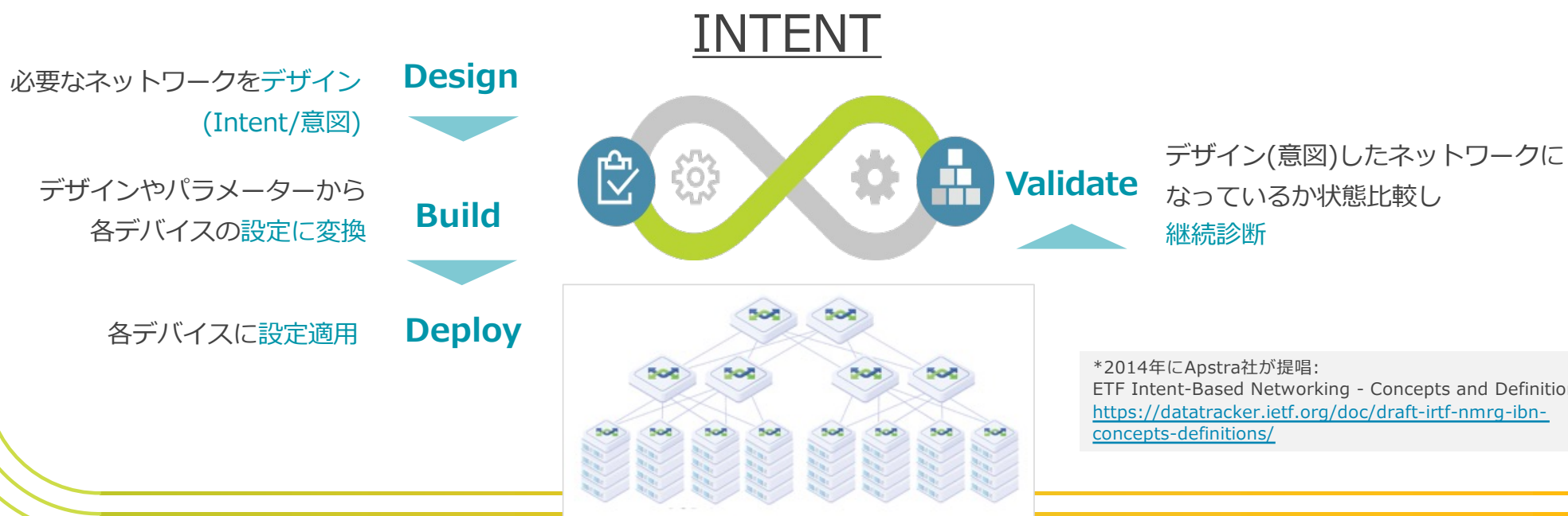
50% 断時間の
短縮

統一化され一貫したエクスペリエンスの提供



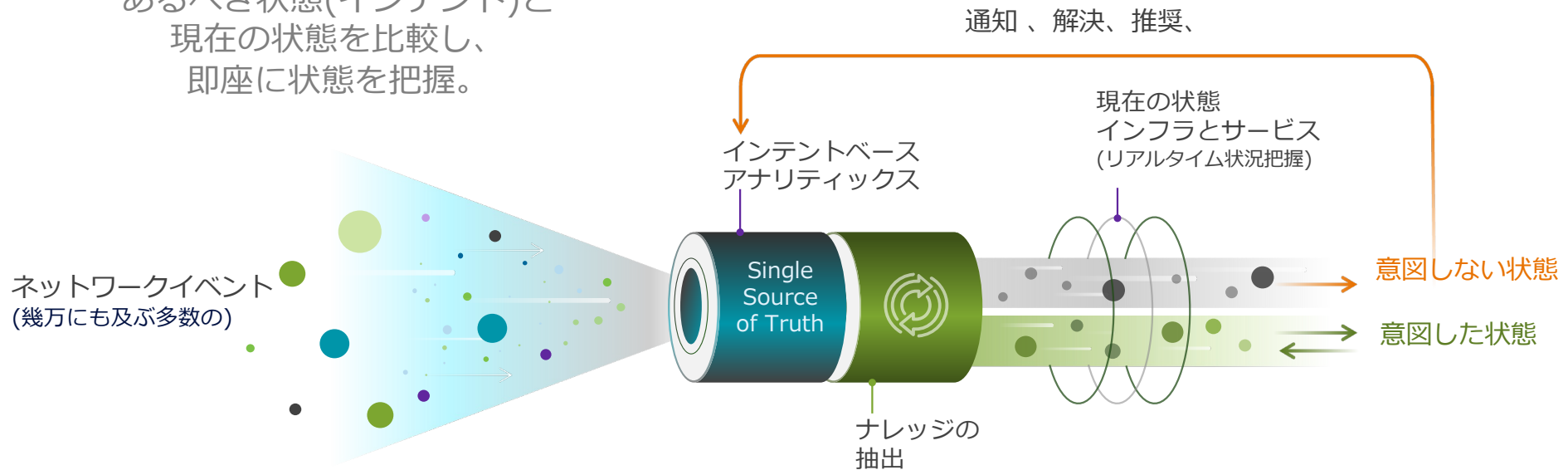
Intent-based Networking(IBN*)とは？

Intent(意図)とデザイン(トポロジー、パラメーター等)から自動化



Day 2+ オペレーション

あるべき状態(インテント)と
現在の状態を比較し、
即座に状態を把握。



ナレッジの
蓄積



クローズドループ
バリデーション



障害回避

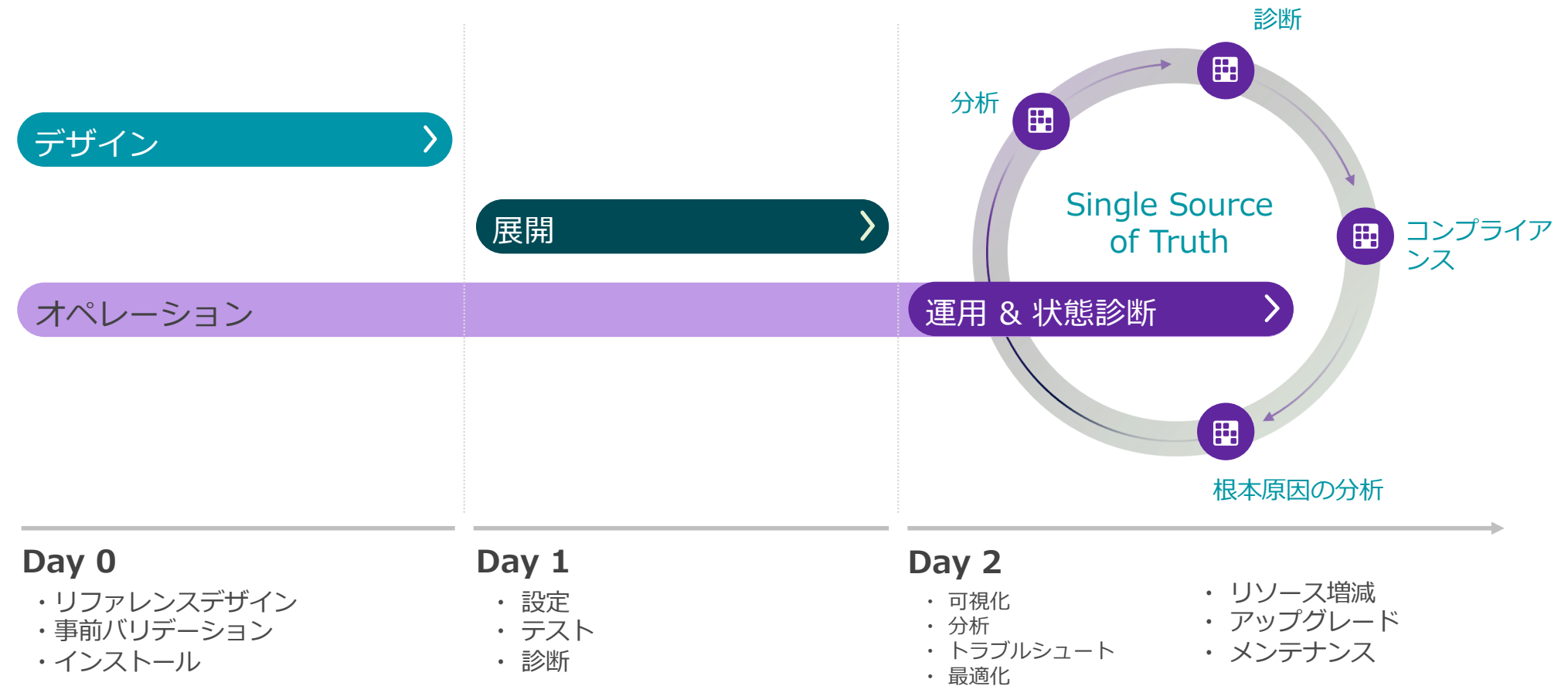


復旧時間の
短縮



管理の改善

運用全体のライフサイクルを自動化



Apstraによる導入効果



容易な運用 効率化

“POCから2週間で
商用導入を可能にした。”

- BeElastic



TCO 削減

“以前は8人で運用していたネットワークが2人で運用することができるようになった。6人は新たなビジネスに取り組むことができるように。”

- Global 500 Manufacturer



セルフドライブ ネットワーク

“私達の成功は如何にはやく新しいサービスを提供し、要求に応じて拡張し、高い品質を提供することが重要です。

JuniperとApstraはそれを実現しました。”

- T-Systems



可観測性 迅速な状態把握

“Apstraによりデバイス、スイッチ、OS、ベンダー、全体像を容易に把握することができます。”

- Bloomberg



Apstraの国内実績と選定理由



データ分析基盤

通信事業者

5Gコア基盤

総合電機メーカー

クラウドサービス

製造系サービス事業者

自社開発基盤

ゲーム会社

開発棟ネットワーク

サービス事業者

アプリサービス基盤

公共、金融、製造など国内外に**採用ユーザーが増加中！**

採用理由

- 簡単な操作、ベンダー依存からの脱却
- 設定ミスの事前検知
- 保証されたコンフィグ
- 豊富な運用機能（切戻し機能など）
- Day2作業後の自動監視

- ブラックボックスのないトラブルシュート
- 大規模ネットワーク対応
- マルチベンダー
- 新規ネットワークOS採用時の学習コスト削減
- API/PythonによるNetOpsオペレーション

Yahoo Japan様によるApstra利用の導入効果

データ分析基盤として数百ラック規模で、1週間以上要していた追加作業が2時間に短縮

Juniper Engineering Simplicity

YAHOO! JAPAN

Clos IP ファブリックの運用基盤に「Juniper Apstra」を採用
効率的なネットワーク設計・構築・運用を実現

サマリー

導入企業
Yahoo! 株式会社

所在地
東京都千代田区豊洲4-1-3
東京のデータセンターに設置されたサーバー

導入
2020年10月

導入台数
100,000台 (2020年10月30日現在)

導入機材
Juniper AOS (2020年10月30日現在)

導入背景
当社は最新のデータセンター「Yahoo! JAPAN」を運用。広域的な集約、集約性を高め、広域集約、イーサネット接続、広域サービスを実現することが必要。

導入前の課題

限られた人員で大規模な Clos IP ファブリックを運用管理したい

異なるベンダーのネットワーク機器を混在させたい

導入後の効果

ネットワーク運用管理作業の大幅なスピードアップを実現

ネットワークの問題点を容易かつ迅速な検出が可能に

ソリューションの価値

- Clos IP ファブリック設計・構築・運用の自動化を実現
- マルチベンダー環境下における Clos IP ファブリックをサポート

Clos IP ファブリックの運用管理に課題が

日本最大級のデータセンター「Yahoo! JAPAN」の運営をはじめ、日本を代表する大手ネットサービス企業として幅広いサービスを展開するYahoo! 株式会社 (以下、Yahoo!)。2020年10月に広域集約を実現する新たなデータセンターの運用を開始。新たなデータセンターの運用を開始するにあたり、当社は最新のデータセンター「Yahoo! JAPAN」を運用。広域的な集約、集約性を高め、広域集約、イーサネット接続、広域サービスを実現することが必要。

このプロジェクトを実現するためには、大規模な Clos IP ファブリックを運用する必要がある。ネットワークインフラを構築する。そのために、当社は最新のデータセンター「Yahoo! JAPAN」を運用。広域的な集約、集約性を高め、広域集約、イーサネット接続、広域サービスを実現することが必要。

「当社は、データセンターとインターネットの間のネットワークを、データセンターのサーバーや機器のトラフィック量の増加が顕著です。こうした場合に海外の大手インターネットサービス企業では、ネットワーク機器を事前にスチームアップできる Clos IP ファブリックで対応しており、弊社もそれに倣ってベストプラクティスを学びました。」

同じく、当社は、当社のデータセンターの一部ネットワークに Clos IP ファブリックを導入。その結果、当社のネットワークも高効率化していることがわかりました。しかし、その結果、当社のネットワークのトラフィックを増加させるためには、ネットワーク機器の迅速な検出、検出後の検出・メンテナンスに迅速に対応する必要があります。今後、ビジネスの成長やデータの増加に伴い、さらにネットワーク

Automation

- Apstra AOS: CLOS Fabricに特化した自動化ツール
 - Intentベース (アドレス、ラック数、アップリンク帯域などを定義するだけ)
 - Config自動生成
 - ケーブルミス、障害解析、Telemetryなど運用向上

自前でできないものをカバー

apstra

150ラック 30ラック 30ラック 30ラック 30ラック

障害停止 85%削減

問題特定時間 70%削減

導入アジリティ 99%向上

変更作業 90%向上

OpEx 83%削減

CapEx 60%削減

<https://www.slideshare.net/techblogyahoo/yjtc18-a1>

Apstra採用による効果

多くの導入効果を実現いただいております



ROI

320%

Payback

<6 months



Day 0

60% savings

Day 1

From 24 hrs →
2 hrs per device

Day 2+

60% savings
ongoing



Apstra Costs

\$275K over 3 years,
includes licenses,
deployment, training

Savings

レガシーツールの
排除

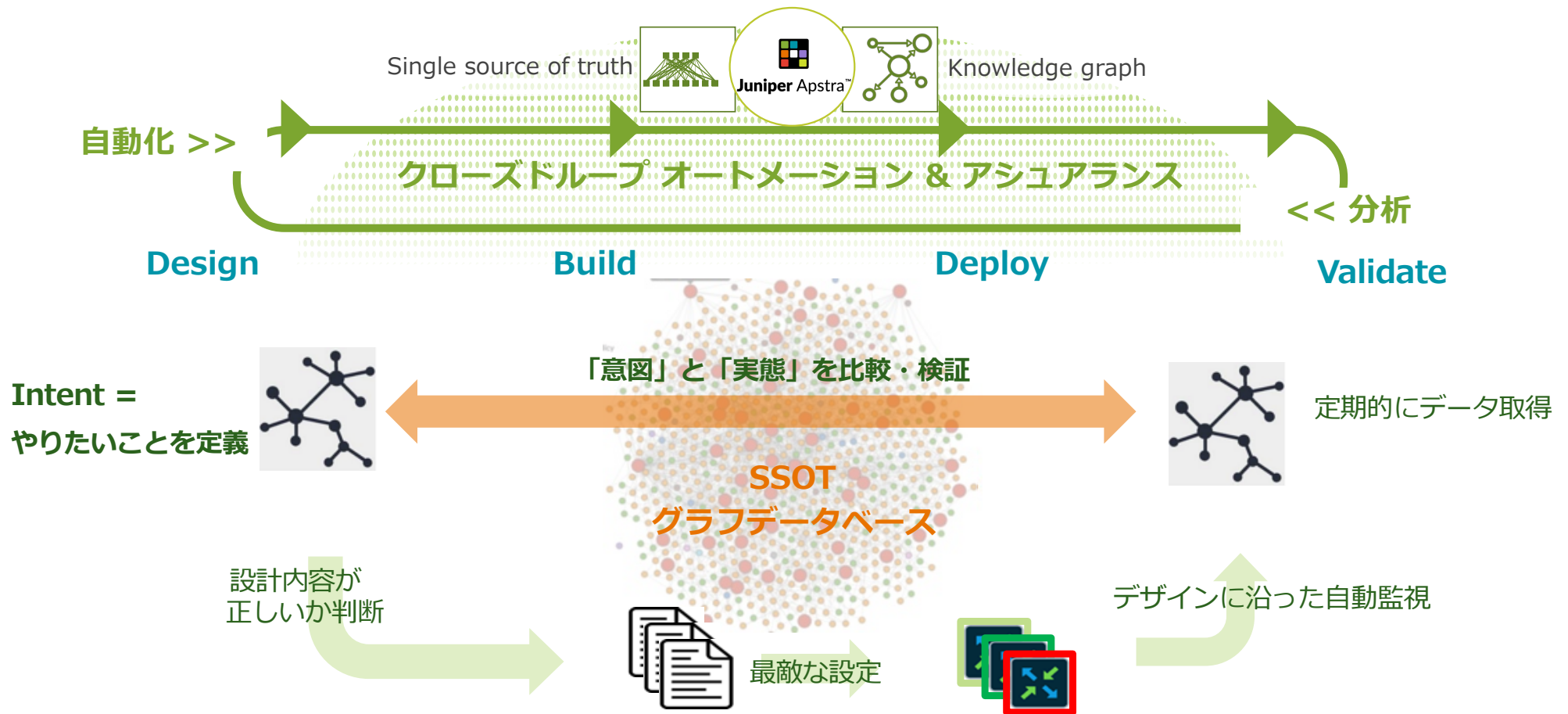
Source: Forrester 2022

The Total Economic Impact™ of Juniper's Apstra Intent-Based Networking Solution

採用効果レポート : Forrester: The Total Economic Impact™ Of Juniper Apstra
<https://www.juniper.net/content/dam/www/assets/white-papers/us/en/2022/forrester-the-total-economic-impact-of-juniper-apstra.pdf>

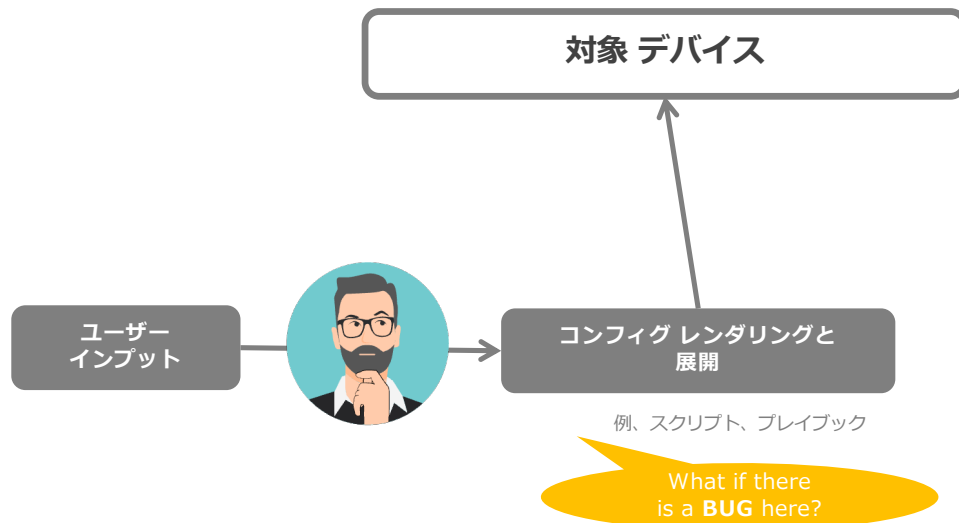


Intent-based Networking with SSOT



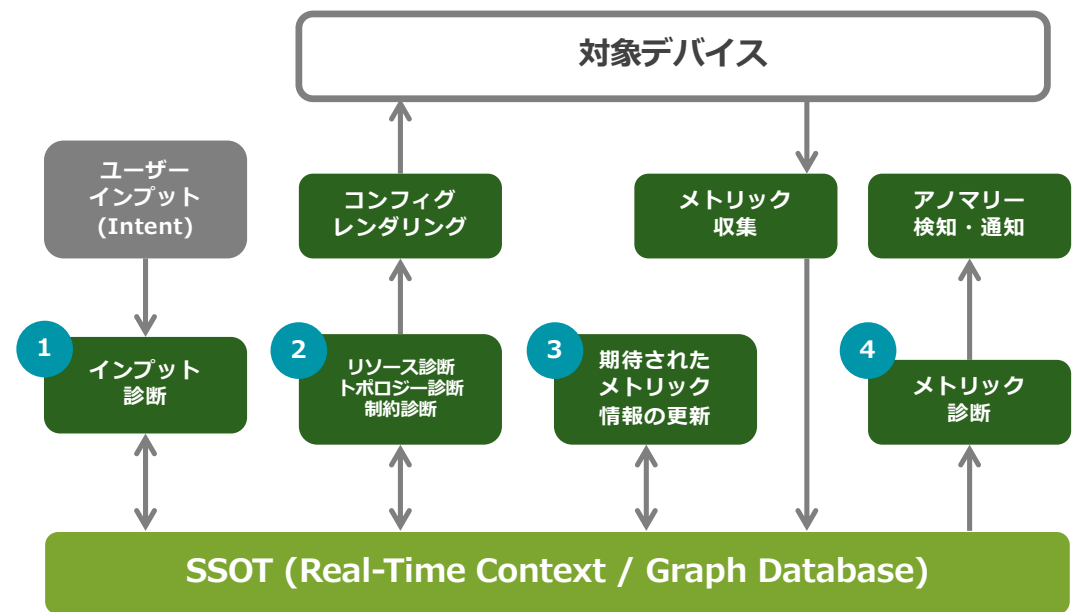
ネットワーク自動化手法の違い

ステートレスな自動化 (スクリプトベース)



自動化に必要な設定や項目を検討しスクリプトを作成
誤った自動化ソフトウェアの設定やBUGにより
大きな障害へとつながるリスク

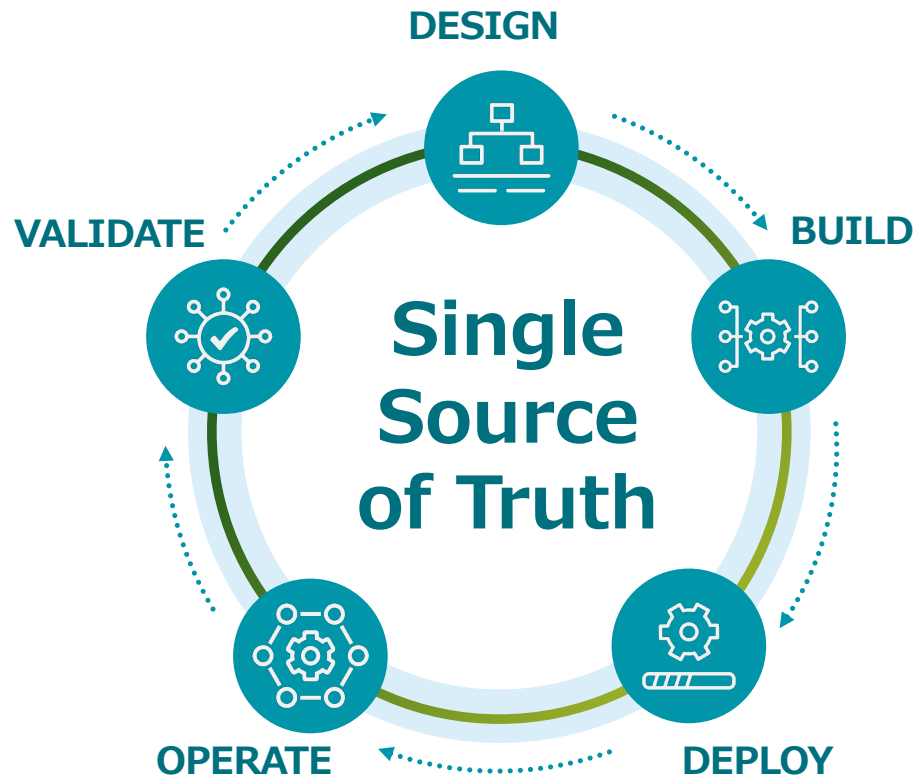
ステートフルな自動化 (IBN with SSOT)



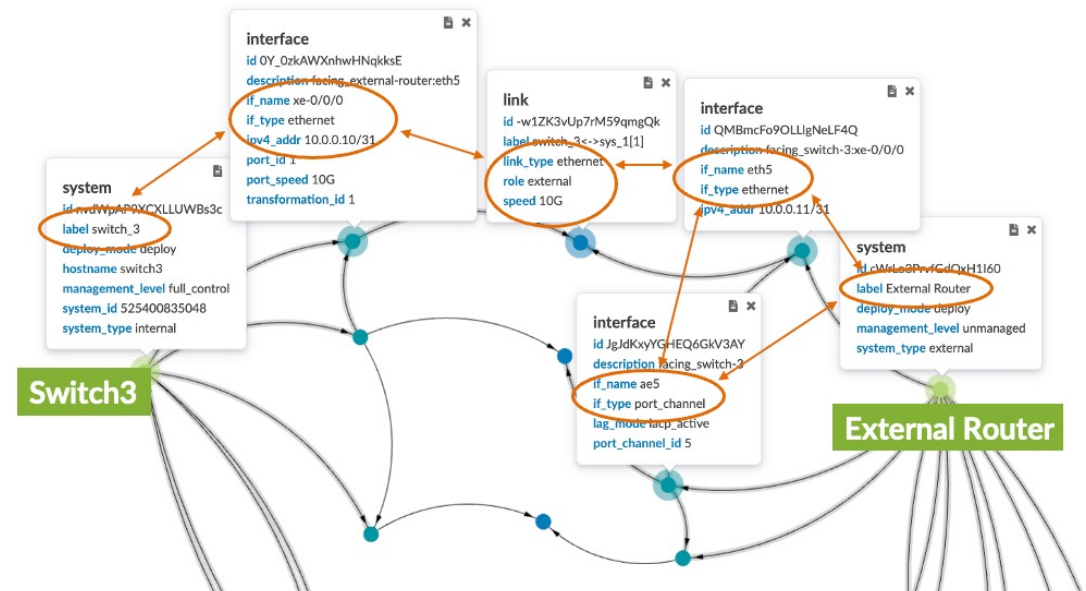
設定の自動化だけでなく、診断の自動化も

IBN with SSOT :多重に診断することでミスのない安定したインフラを実現

IBN: SSoT - Single Source of Truth アーキテクチャ



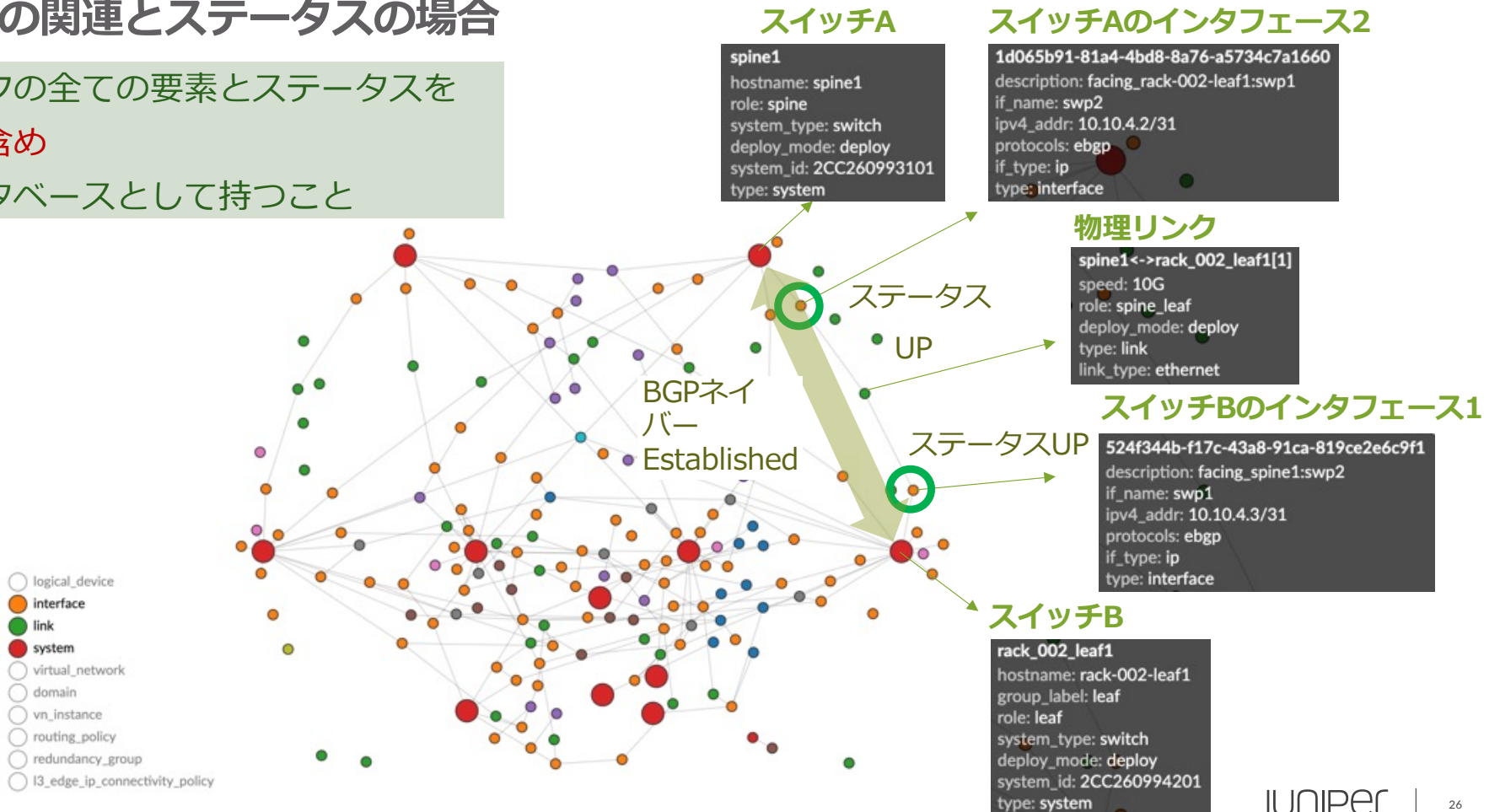
GraphDB内の各ノード
(システム) は、コンテキストデータと他の
ノード (システム) との関係性を保持
整合性確認の実現



What is SSOT Database?

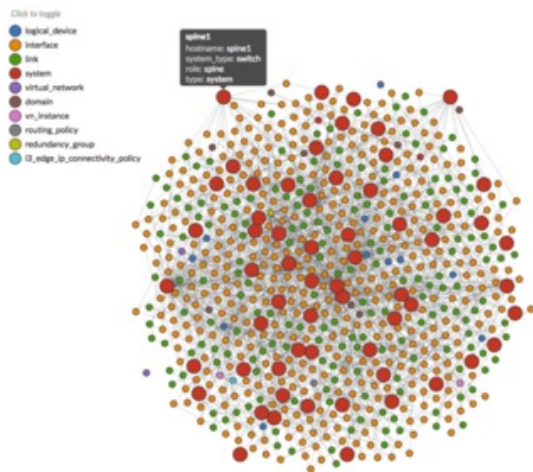
NW機器間の関連とステータスの場合

ネットワークの全ての要素とステータスを
その関係を含め
一つのデータベースとして持つこと



SSOT DBアーキテクチャ

ネットワークの多様な要素とステータスを把握し自動化/可視化



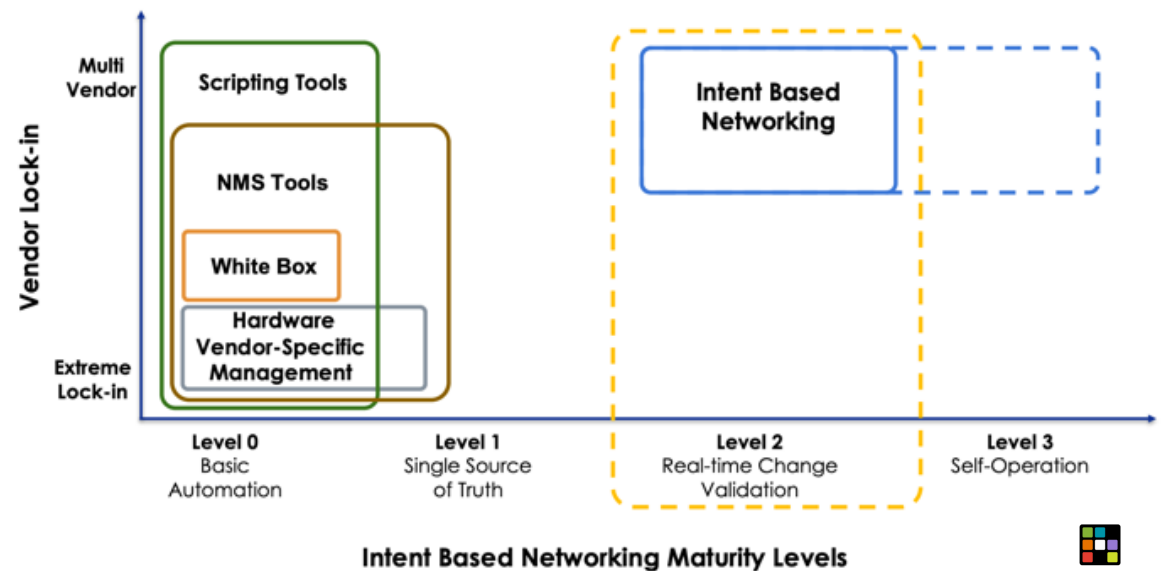
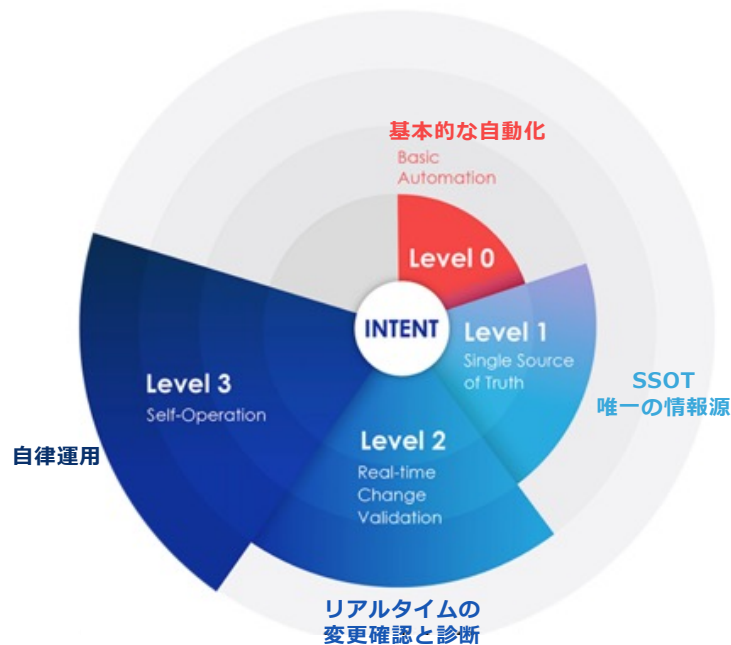
	例	SSOT	Non-SSOT
NWデザインチェック	IPアドレス重複 スピードミスマッチ	トポロジーと設定情報を保持 するためチェック可	アドレス重複は確認可
コンフィグ投入影響	BGPルートマップ設定変更	対向機器のルーティングテー ブルへの影響を確認可	機器間の設定内容やステータ スが紐付いていないため検知 不可
監視項目の自動作成	新規サーバ接続ポートの監 視すべき項目	サーバがチーミングしている ためLAGとポートステータス のモニターを開始	サーバ接続するデザインとブ ロトコルの関連がないため不 可
NWステータスの検証	ルーティングテーブルのル ートチェック	デザインから各機器で学習す べきルートを知っている	デザインとステータスが関連 していないため判断不可

通常それぞれの関係性を把握し自動化や可視化を行なうことは困難
SSOTにより、効果的な自動化を実現し、グラフDB(NoSQL)による高速処理を実現
※大規模OTT事業者も類似のアーキテクチャ各要素を抽象化することで自動化を実現

Intent-based Network Landscape

IBNにより自動化を次のステップへ、そして全自律ネットワークへ

IBNの分類とレベル

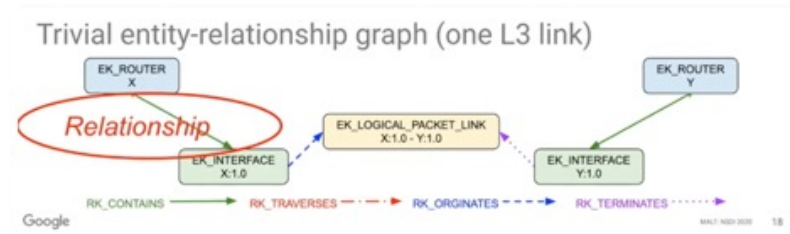
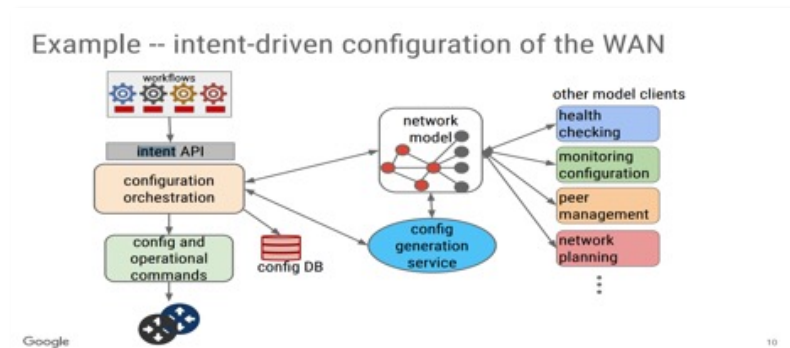


参考: Intent-based Networking with SSOT

- 大規模ネットワークでも進むIBN with SSOTの自動化プラットフォーム
- ネットワークを事前にデザインし展開、デプロイ後は正常性確認

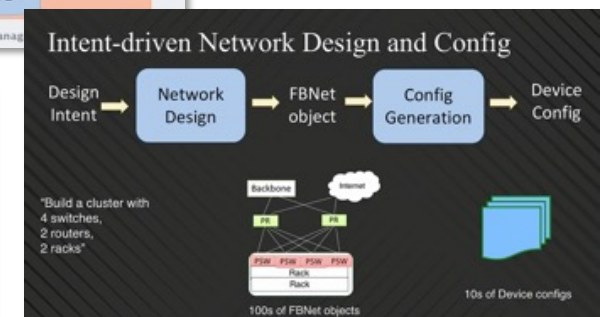
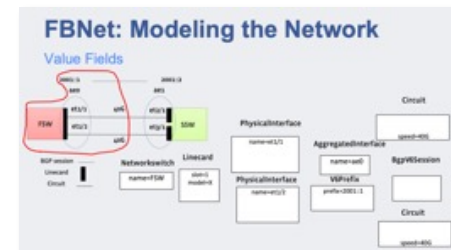
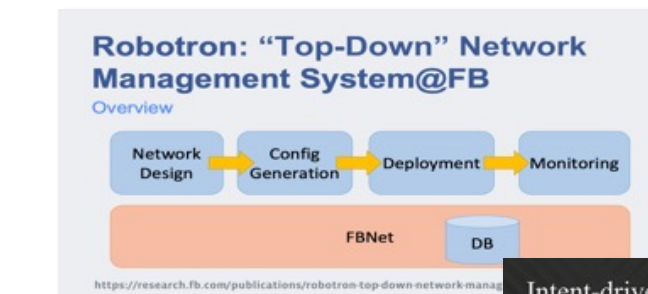
Google MALT

Intent-Driven Configuration Platform



Facebook Robotron and FBNet

Intent-Driven Network Design and Configuration Platform



Juniper Apstraのご紹介

- デザインパターン(DC Reference, Freeform)
- コンポーネント
- Apstra Flow
- Custom Telemetry
- Apstra + Terraform
- Apstra Cloud Services
- AI Cluster Template



Juniper Apstra™

Juniper Apstraの2つのデザインパターン



**DC Reference
Architecture**

Best of Everything

**Freeform
Architecture**

New

容易な自動化を
ベストプラクティスリファレンスと共に

すべてのDC環境の自動化に



Meal kit - DC Reference



DIY - Freeform

Juniper Apstraの2つのデザインパターン



DC Reference Architecture

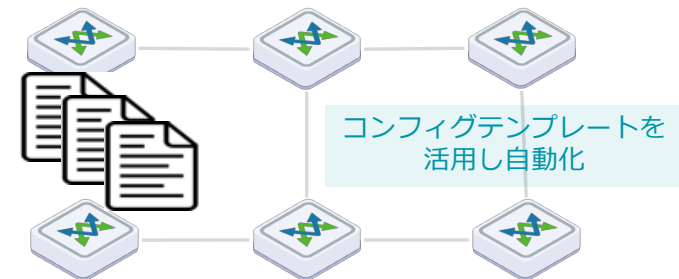
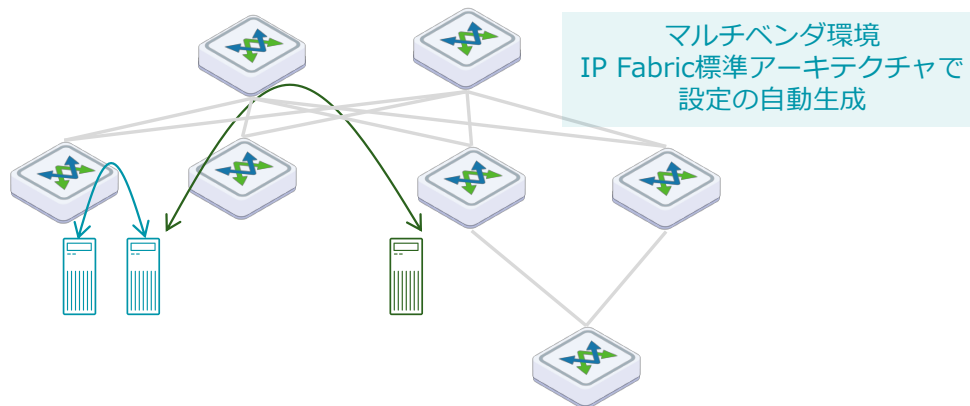
Best of Everything

Freeform Architecture

New

データセンタネットワーク向けの
ベストプラクティスアーキテクチャと
トポロジーを評価済み設定で自動化・可視化

すべてのトポロジーとフィーチャで
どのような設定でも
自動化・可視化基盤として利用可能



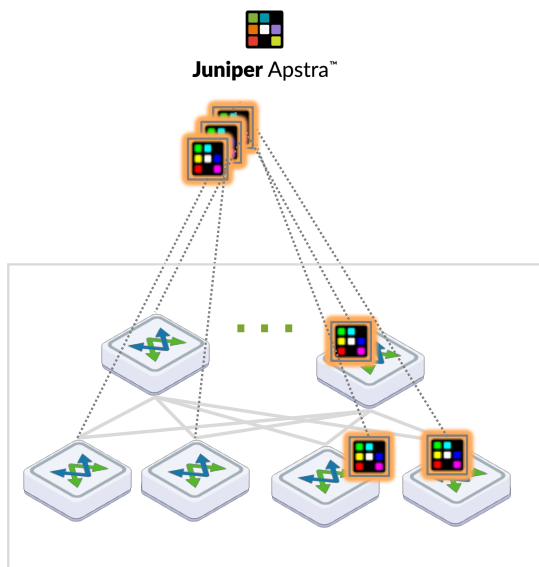
DC Reference と Freeformの違い

	DC Reference Architecture	Freeform Architecture
デザイン	Logical Device/Racks/Templatesから作成（ERB）	トポロジー及びプロトコルの制限なし（デバイス依存）
作成されるコンフィグ	Juniper/Apstraで動作保証済み	ユーザー側にて要検証
監視	フルセット	DC Reference Designの一部のみ
CLI syntaxの知識	基本不要（Configlets機能を利用する場合には必要）	必要
Jinja2の知識	Configletsには必要なことがある	必要(ただ、Jinja2を使わず静的な設定を流し込みも可能)
Time Voyager(ロールバック)	使える	使える
NOS Upgrade	できる	できる
ベンダーサポート	Juniper(JUNOS/EVO)/SONiC/NXOS/EOS	Juniperのサポート済みデバイスのみ(他はroadmap)
Brownfieldサポート	要PS相談	要PS相談
VMware連携	サポート	サポートしない

Juniper Apstra デプロイパターン

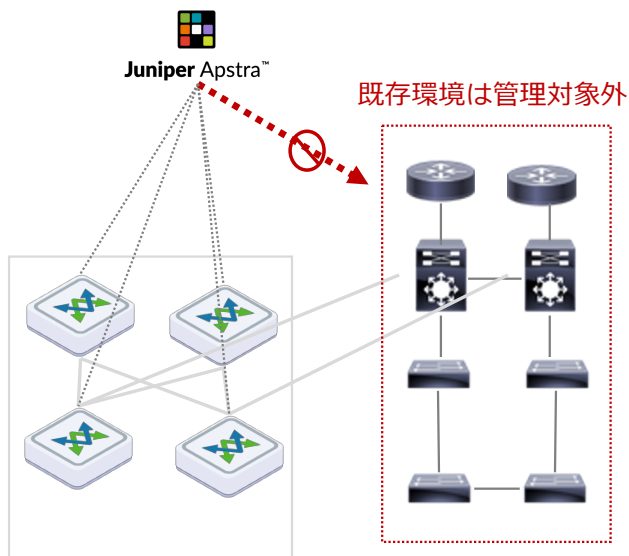
Green Field

新旧データセンター



Green Patches

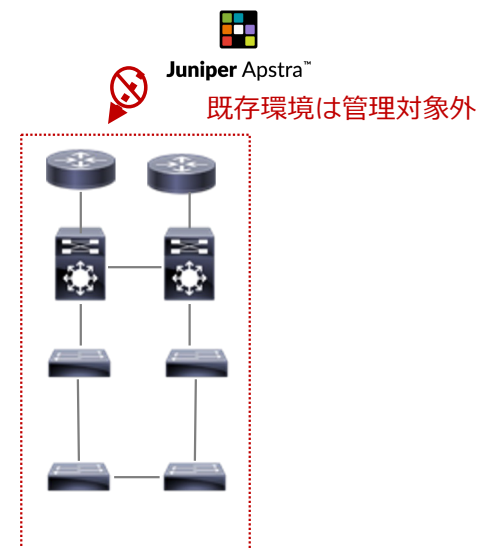
既存ネットワークと接続
既存からのマイグレーション



Apstraのスムーズスタート構成を既存ネットワークに接続し、新規ワークロードはApstra配下とすることや、既存ワークロードを徐々にマイグレーション

Brown Field(要相談)

既存のネットワークを自動化



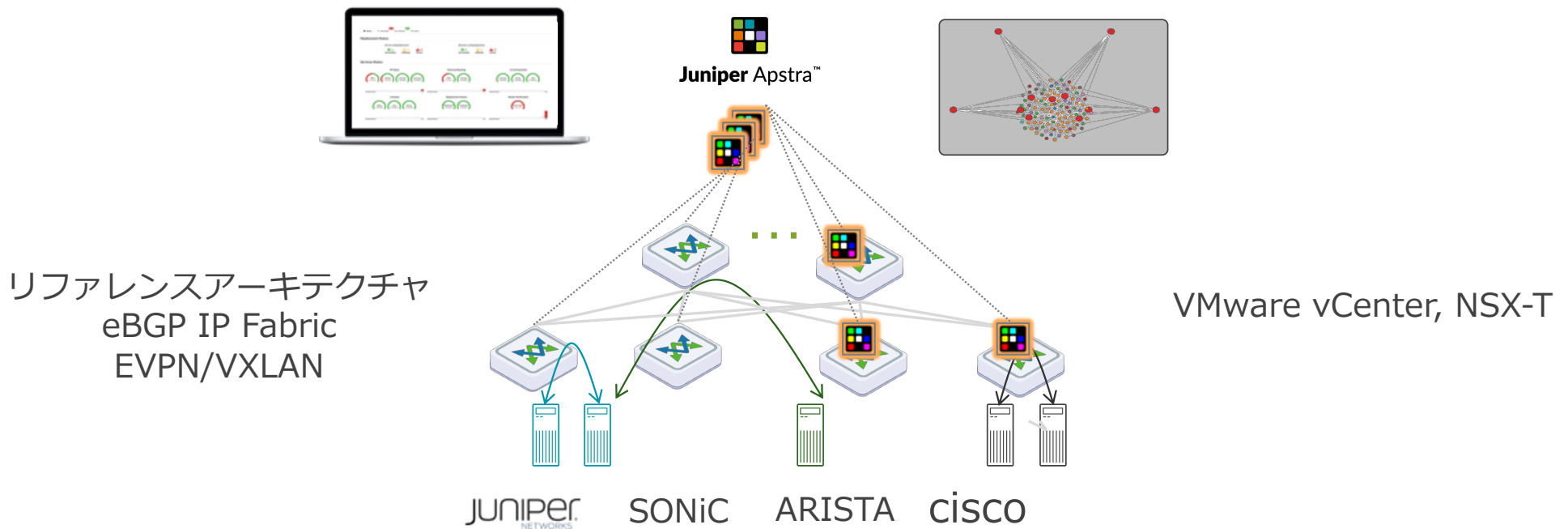
デザインパターン

- DC Reference Architecture



Juniper Apstra™

Juniper Apstra -Intent-based Networking



DCリファレンスデザイン設定の自動化

評価済み設定

リスク軽減

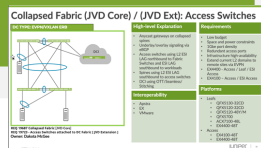
信頼性向上

デバイス単体ではないエンドツーエンド評価テストを複数構成で毎日計1500万以上のテスト項目を実施
充分にテストされたサポート済み(TACサポート有)の設定を自動生成

4 JVD(Juniper Validated Design・評価済みデザインと設定)

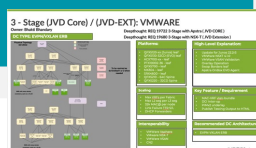
Collapsed Fabric

小規模構成
エッジサイト



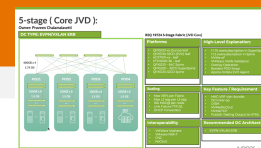
3-Stage EVPN/VXLAN

中規模・大規模スケール



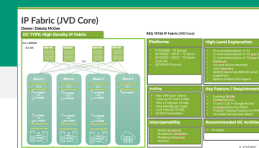
5-Stage EVPN/VXLAN

大規模スケール



IP Fabric

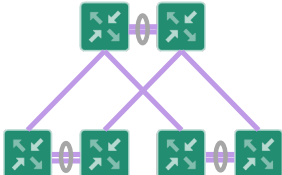
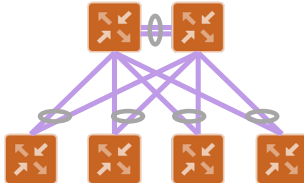
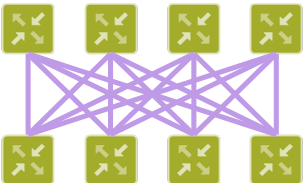
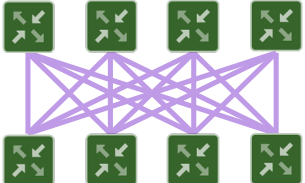
中大規模スケール
L3ファブリック



Apstraにより設計に応じた評価済み設定を自動生成

DCネットワーク技術の進化

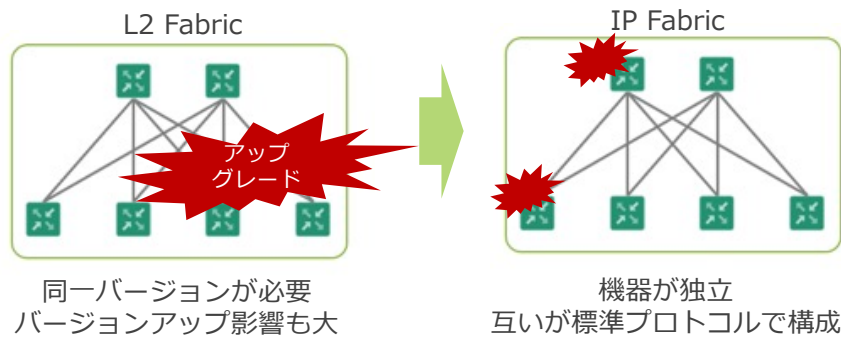
OLD → NEW

		STP + VRRP		マルチシャーシLAG		L2 Fabric		IP Fabric (VXLAN)
								
設計	△	STPの設計が面倒	○	STP排除のため楽	△	設定パラメータが多い	△	設定パラメータが多い
	×	帯域を有効活用できない	△	フレームレベルでLB	△	フレームレベルでLB	○	複数レイヤを使いLB
	△	端末のDGはコア	△	端末のDGはコア	△	端末のDGはコア	○	端末接続スイッチがDG
	○	マルチベンダー接続可	○	マルチベンダー接続可	△	マルチベンダー接続不可	○	マルチベンダー接続可
運用	×	L2ループでNW全体ダウン	×	設定ミスでL2ループ	○	ループ排除で安定	○	ループ排除で安定
	○	トラブルシュート容易 (オープン技術)	○	トラブルシュート容易 (オープン技術)	△	トラブルシュート困難 (ベンダー独自技術)	△	トラブルシュート困難 (BGP運用)
拡張	△	コア拡張は推奨されない	△	コアスイッチは2つまで	○	コアスイッチ3つ以上可	○	コアスイッチ3つ以上可
	○	3階層以上に対応	△	2階層まで	△	2階層まで	○	3階層以上に対応
人材	○	精通した技術者多い	○	技術者多め	△	技術者少ない	△	技術者はまだ少ない

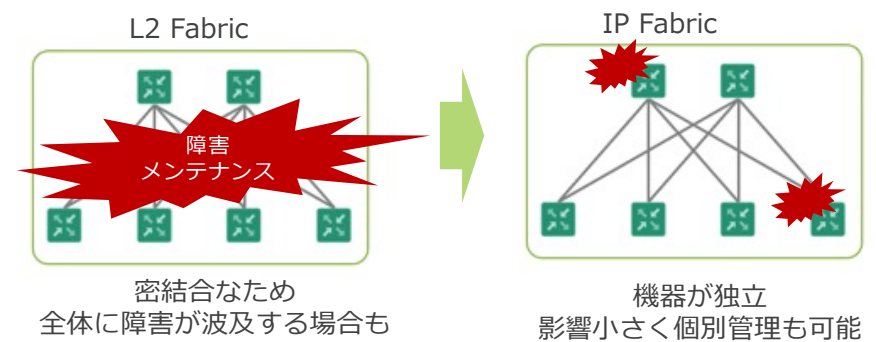
IP Fabric Advantages

疎結合なアーキテクチャは多くの問題を解決する

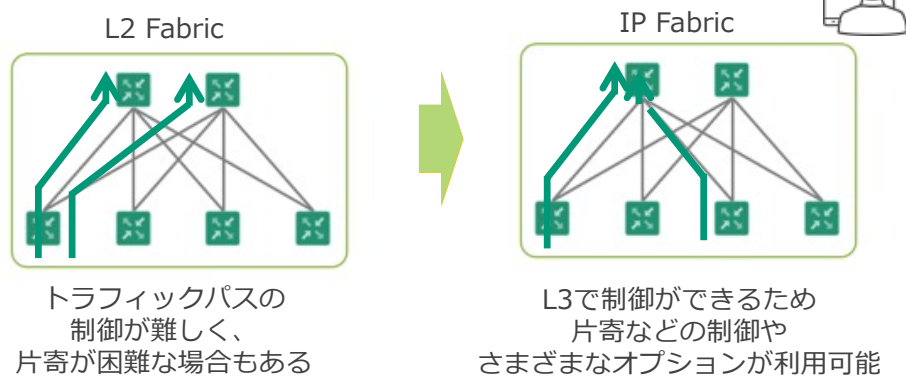
アップグレード



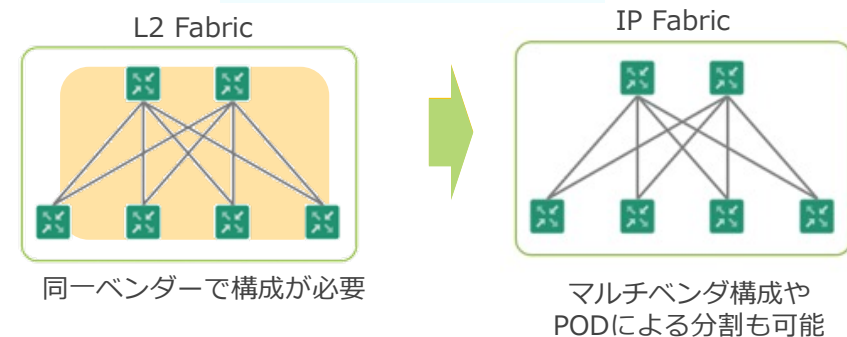
障害影響



トラフィック制御



ベンダーロックイン



EVPN/VXLAN導入に伴う課題

Apstraで全体のライフサイクルを管理・運用・可視化



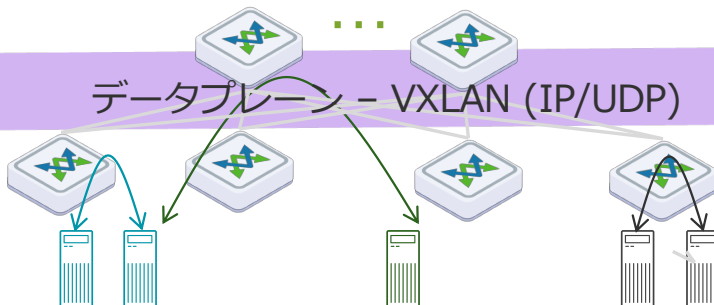
マネージメントプレーン- Apstra

オーバーレイコントロールプレーン- BGP EVPN

アンダーレイコントロールプレーン- BGP unicast

データプレーン - VXLAN (IP/UDP)

VRF

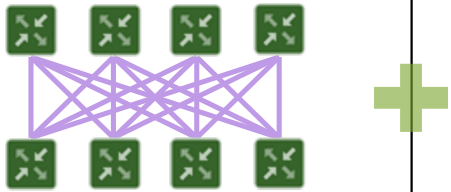


Apstraが複雑さをカバーし
導入を容易に！

- 耐障害性が高く、スケールアウト、L2/L3/マルチテナントが可能で標準化されたEVPN/VXLAN
- 但し、EVPN/VXLANは複数プロトコルを用いるため設定が複雑になる傾向

IP FabricにおけるApstraの優位性

ApstraでEVPN/VXLAN全体のライフサイクルを管理・運用・可視化し導入負担を軽減

	IP Fabric (VXLAN)		Apstra	
			 Juniper Apstra™	
設計	△	設定パラメータが多い	◎	WebUIで自動コンフィグ作成
	○	複数レイヤを使いLB	○	対応
	○	端末接続スイッチがDG	○	対応
	○	マルチベンダー接続可	◎	複数ベンダー間の接続をテスト済み
運用	○	ループ排除で安定	○	スイッチの実装で可
	○	トラブルシュート容易 (オープン技術)	◎	豊富な運用・監視ツールを提供
拡張	○	コアスイッチ3つ以上可	○	対応
	○	3階層以上に対応	○	5-Stage CLOS (3階層)対応
人材	△	技術者はまだ少ない	◎	クラウドラボでトレーニング

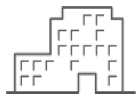
Apstraはさらに様々な自動化を提供



Reference Architectureの採用

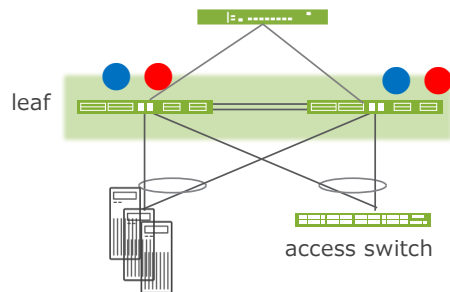
標準技術である **EVPN/VXLAN IP Fabric** のみ使用し、ベンダー**独自技術は除外**

Juniperが培った**ベストプラクティスアーキテクチャ**と
トポロジーを評価済み設定で自動化・可視化



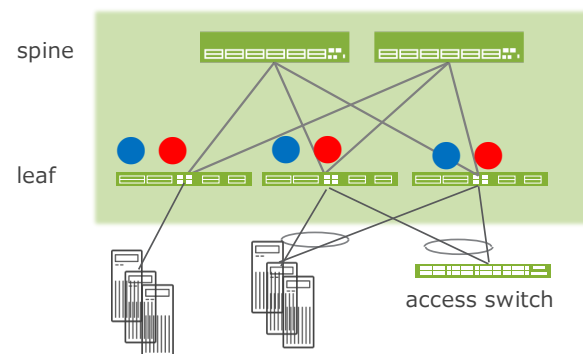
Small/Medium DC

EVPN マルチホーミング
Collapsed Fabric(Juniper製品のみ)



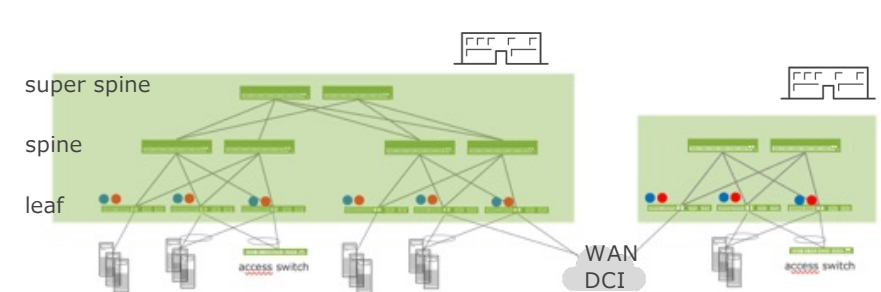
Medium/Large DC

EVPN/VXLAN ERB
(3-stage)



Large DC

EVPN/VXLAN ERB
(5-stage + DCI)



Juniper Apstraが実現すること

ネットワーク(IP Fabric)の自動構築・監視



Juniper Apstraの主要提供機能

構築、運用、監視まで一気通貫で機能を提供



Day 0

Design + Build + Deploy

- Networkデザイン(オフライン)
- Certified & Tested
リファレンスアーキテクチャ
- Config自動生成
- ZTP
- 複数サイト管理 / DCI



Day 1

Operations

- 仮想ネットワークプロビジョニング
- ネットワーク拡張/縮小/変更
- セキュリティポリシー適用
- システムロールバック
- メンテナンスモード/トラフィック迂回
- システム/デバイス アップグレード
- イベントログ/show tech取得



Day 2

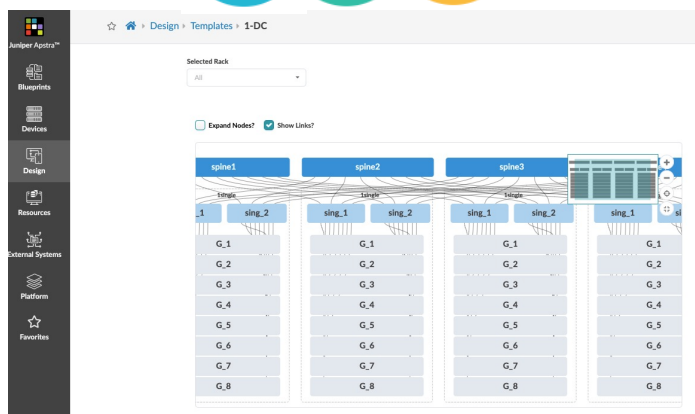
Automation + Assurance

- 自動ネットワーク診断
- ルートコース分析
- リソース枯渇監視
- トラフィック監視

DAY 0/初期セットアップ：ネットワークデザイン

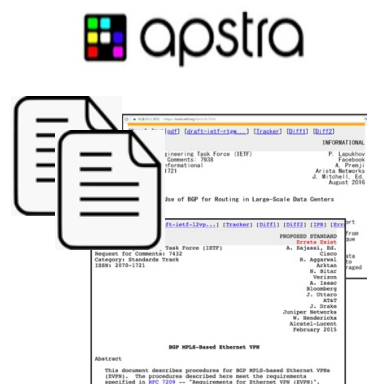
必要なネットワーク構成をGUIでデザインし、設定と可視化項目を自動生成

必要なネットワークを設計



- ・デバイス接続トポロジ
- ・ラック構成
- ・デバイス構成(ポート構成・接続情報)
- ・パラメータ情報(ipアドレス, AS番号)

評価済みの
リファレンス・デザイン



- ・業界標準化済み技術
- ・事前評価済み
- ・設定内容の保証

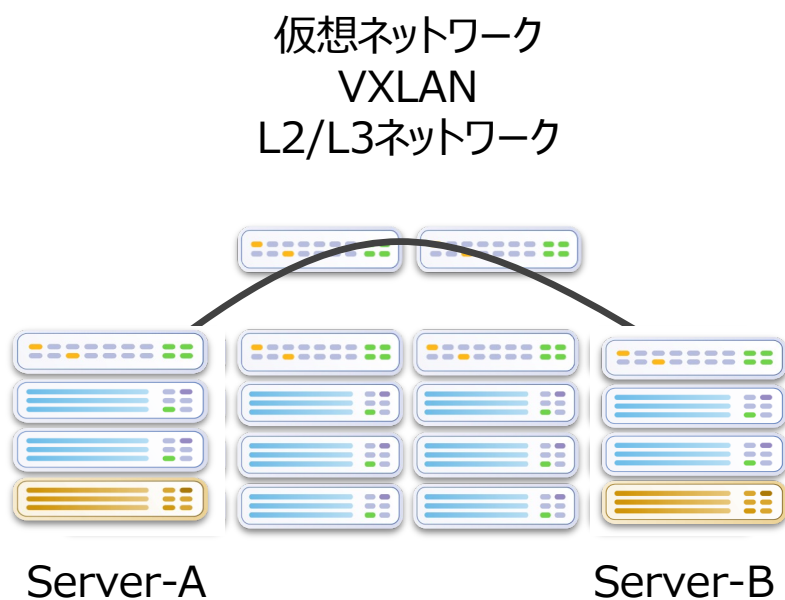
必要なデバイスの設定と
可視化項目の自動生成



Running System

- ・設定自動生成
- ・設定適用後は正常性診断

ネットワーク全体の設定GUI変更・診断

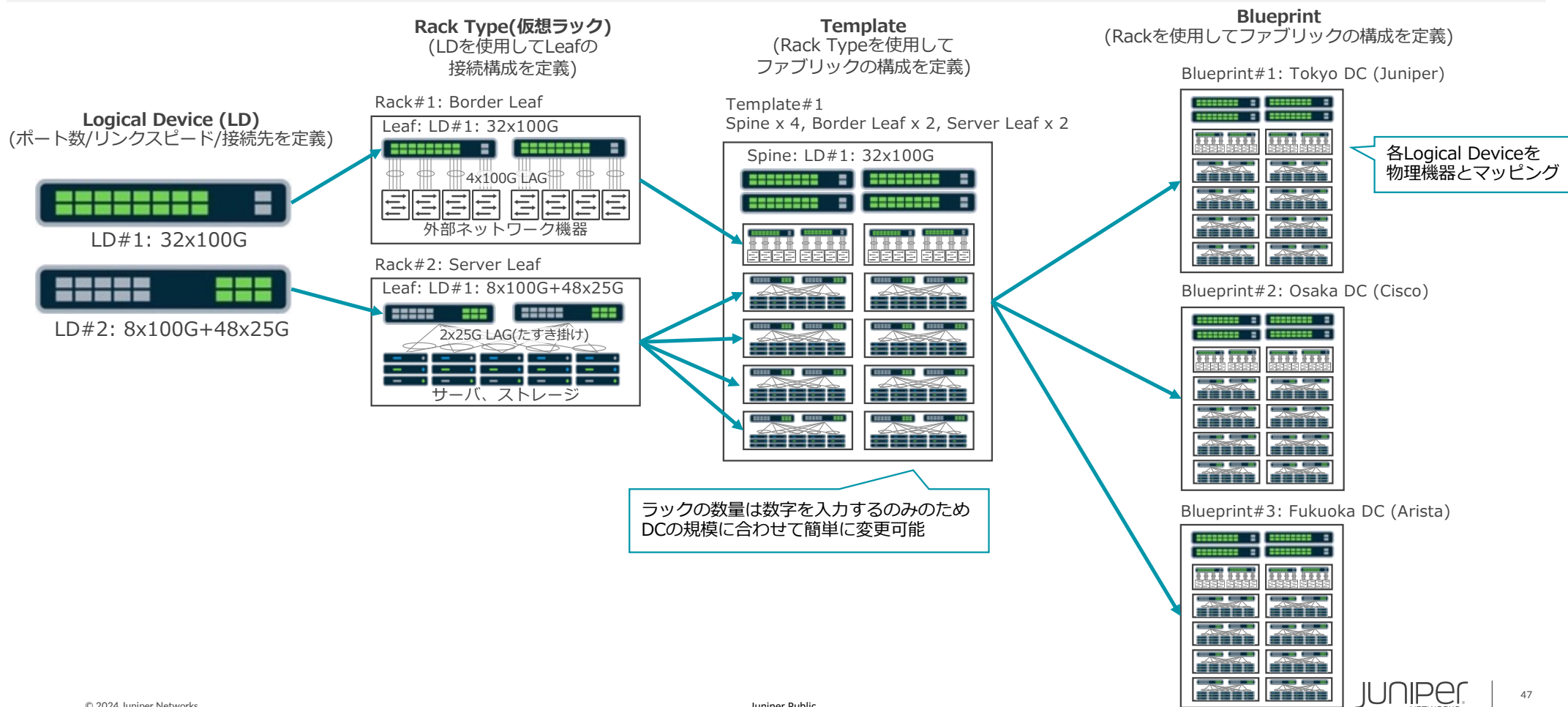


- マルチテナントVRF
- 仮想ネットワークVLAN/VXLAN
- セキュリティフィルターポリシー
- 外部ルーティング
- DCI
- ラック追加
- サーバー追加
-

評価済み設定を自動生成し、適用前の内容確認・自動診断

初期構築：トポロジー定義

必要なトポロジー設計から各デバイス・ラック構成を定義



番外編：Apstraの初期設計方法-その 2

*現在は社内またはパートナーのみに公開。

Web Tool(Clover Designer)でDCネットワークに必要な情報をインプットするだけで、Juniper推奨ハードウェアとApstraの初期構築が可能



要求事項情報をインプット



ファブリックデザインを自動生成



HWデバイス推奨構成の提示



構成 PPT/PDF/XLS/TF展開



Apstraへオンライン展開/TF

- ・サーバー向けポート数・帯域
- ・帯域重複割合
- ・冗長有無



または



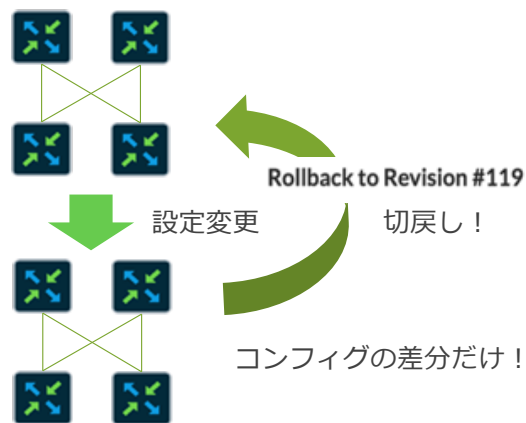
Juniper Apstra™

Operations

ネットワーク運用と設定

ロールバック

WebUIで設定変更した内容を
指定した時点に切戻し



手動切戻し作業は煩雑になりがち、
Apstraだと一発で確実に切戻し

ネットワーク拡張・縮小・変更

ラックの追加・削除



サーバ等エッジ機器の追加・削除
ポートスピード変更



その他

- 個別NOSコマンドをApstraから設定
- AS番号等リソース不足を事前確認
- Show techをApstra経由で取得
- 外部サーバへ監視データを出力可
- Apstraユーザ毎のアクセス制限
- APIによる外部サーバ連携

Configletによる設定の柔軟性

Apstraで設定管理しつつ、各デバイスに個別の独自設定を追加したい場合

- 基本的にコンフィグ設定はApstraから自動的に適用
- 各デバイスに個別の独自設定を追加したい場合、Configletで設定をテンプレート化して適用
- SNMP、NTPなどの管理系や新機能などにも利用可能、RMA時にも設定を保存
- 直接デバイスCLIで設定変更する場合は、アノマリーとしてApstraで検知

コンフィグの
テンプレート化
(Configlets/Jinja)

Config Style

☐ Cumulus ☐ NXOS ☐ EOS ☒ Junos ☐ SONiC

Section

☒ SYSTEM

Template Text

```
system {  
  ntp {  
    server {{ntp_server}} routing-instance {{junos_mgmt_vrf}};  
  }  
}
```

変数化も可能
(property set)

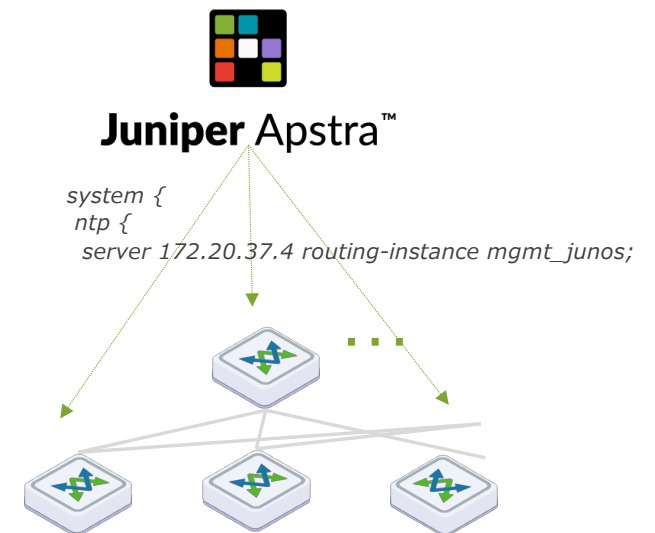
Name

NTP Server

Properties

ntp_server 172.20.37.4

Blueprint/
Deviceに適用

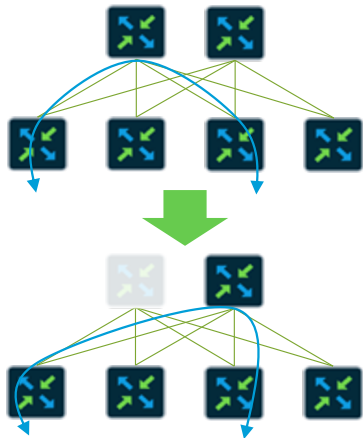


Automation + Assurance

ネットワーク運用とメンテナンス

メンテナンスモード

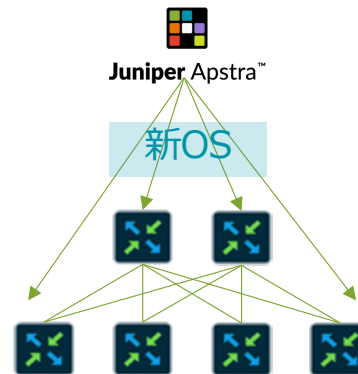
通信経路をスムーズに迂回



NOSバージョンアップや
ハード不良による機器交換時に

NOSバージョンアップ

ネットワークOSを
WebUIからバージョンアップ

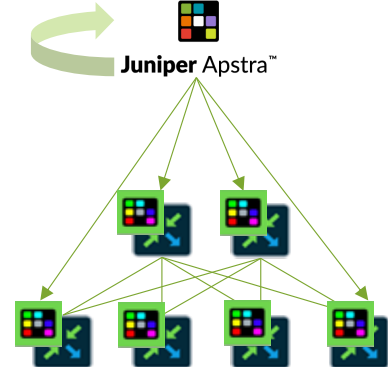


面倒なコマンドラインではなく
ApstraのWebUIから一括で実施

Apstraバージョンアップ

Apstra及びAgentを
バージョンアップ

New!

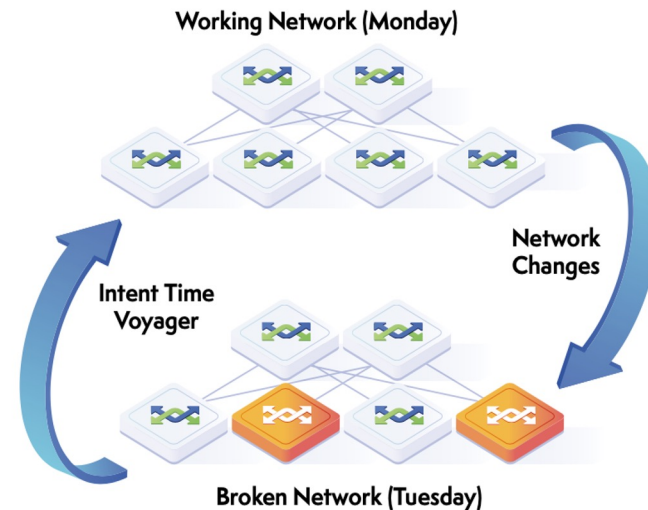
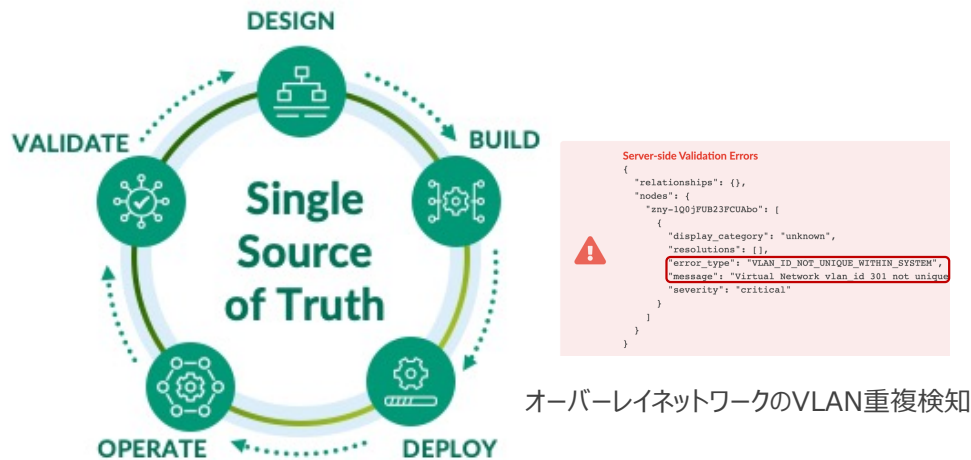


トラフィックへの影響なし
NOSのリロード不要

人的ミスによる障害 – 設定変更整合性チェック–

- ・ミスが発生しがちな、設定変更も、デバイス反映前に整合性のチェック

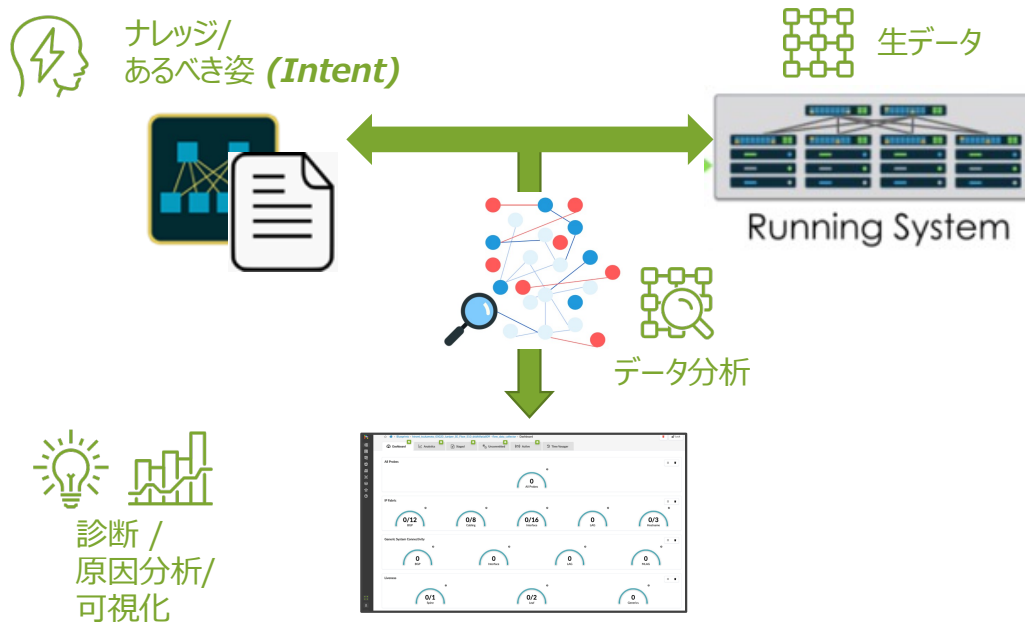
- ・最大100世代まで設定履歴を残し、即時にその時の設定に切り戻し。



設定反映前の自動診断により人的ミスを未然に防止し、切り戻しも容易

Day2: Intent-based Analytics モニタリング

■ デフォルトテレメトリポータル

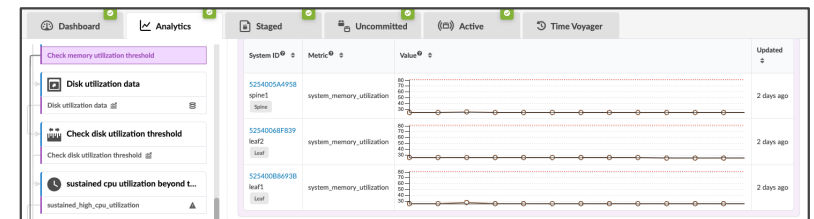


- 物理や論理のあるべき姿と実態を比較し、詳細確認
- 物理ケーブル接続, 経路学習状況, コンフィグ正常投入, プロトコル正常性 BGP, LACP etc.

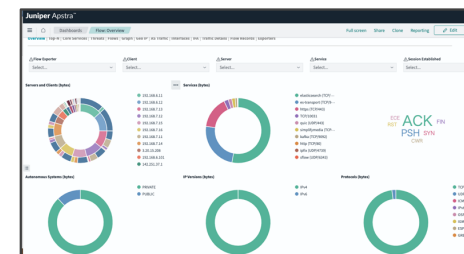
■ Traffic Heat



■ IBA



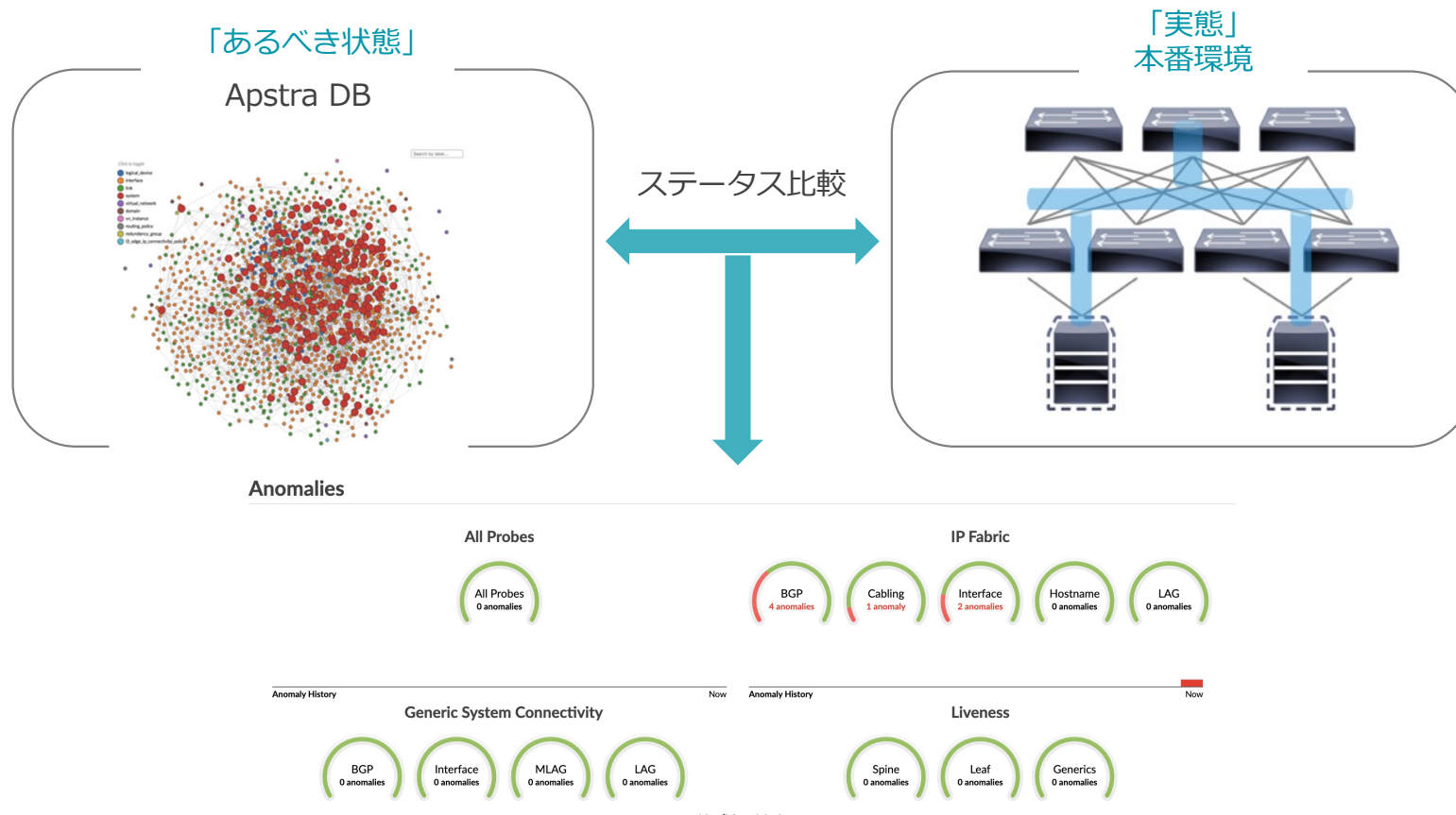
■ Flow Data



*Apstra Flowは
別VMとしてインストール
JUNIPER NETWORKS

Default Telemetry -1

Apstraはグラフデータベースの情報から監視ステータスの**あるべき状態(intent)**を把握し、実際の環境と比較することで正常性を容易に確認



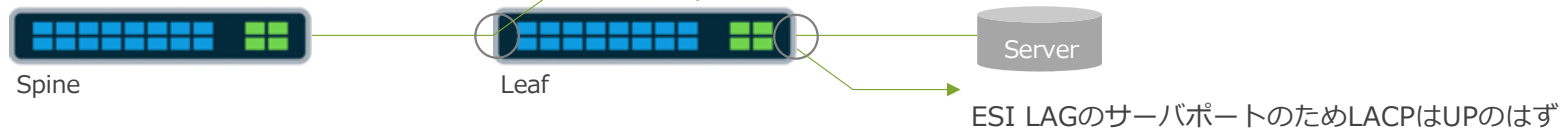
Default Telemetry -2

複数レイヤーの「あるべき姿」に対する「実態」の差分を把握

Layer1 インタフェース



Layer2 LACP



Layer3 BGP

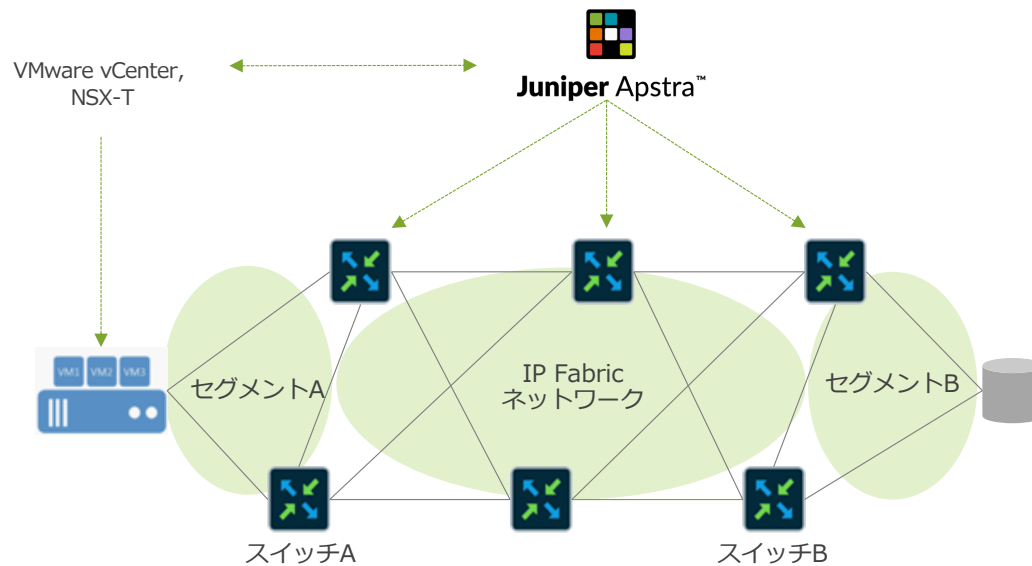


Layer3 ルーティングテーブル



Intent Based Analytics

さまざまな監視項目をIBAとして追加可能



ネットワークレイヤ

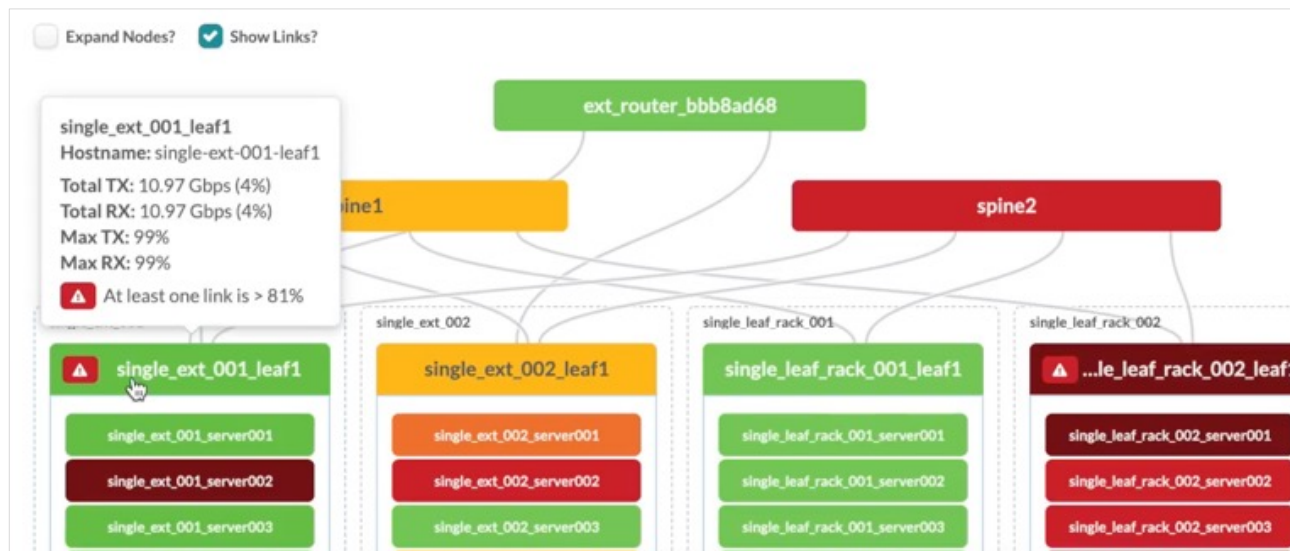
監視例

バーチャルインフラ	VMware vCenter, NSX-Tの 設定の整合性
オーバーレイ	EVPN Type3,5等 特定NWセグメントの学習状況
ネットワークレイヤ3	ルーティングテーブルの整合性 BGPステータス
データプレーン	インタフェースエラーカウンタ インタフェースキュードロップ
ネットワークレイヤ1	トランシーバ光レベル インタフェースフラップ
デバイスヘルス	メモリリーク、CPU使用率 電源・ファン

トラフィック可視化

ネットワーク全体のトポロジーやトラフィック量をヒートマップとして表示

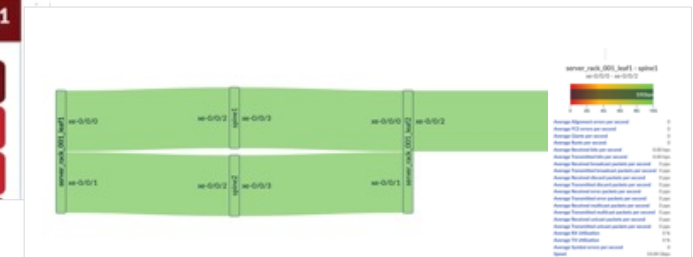
全体のトポロジーとトラフィック量



各デバイスの接続状態とトラフィック量



各エンド端末との接続状態とトラフィック量



Root Cause Identification

複数の事象から真の原因を解析し通知

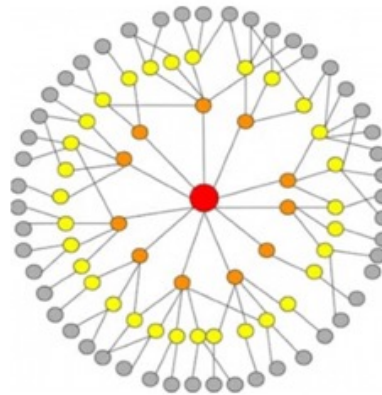
- ネットワーク障害の早期復旧は運用上の最重要課題
- 障害原因の確認、復旧作業にかかる時間の短縮が大事
- 全体のネットワークのデータベースを保持し、ナレッジから問題の根本原因を報告

(例)

インタフェースダウン

LLDPエラー

BGPネイバーダウン



真の原因を解析・報告

GUIとAPIによる運用

GUIによる直感的なオペレーション と APIによる自動化/コード化



視認性の高いGUIで簡単に運用を実現

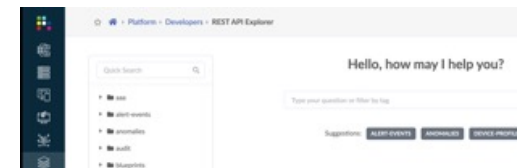


REST API



API利用を容易にするデベロッパーツールをビルトイン

- Rest API Explorer
- Swaggerフレームワーク
- APIStraw
- API Reference



その他にも自動化を支援するための情報を公開

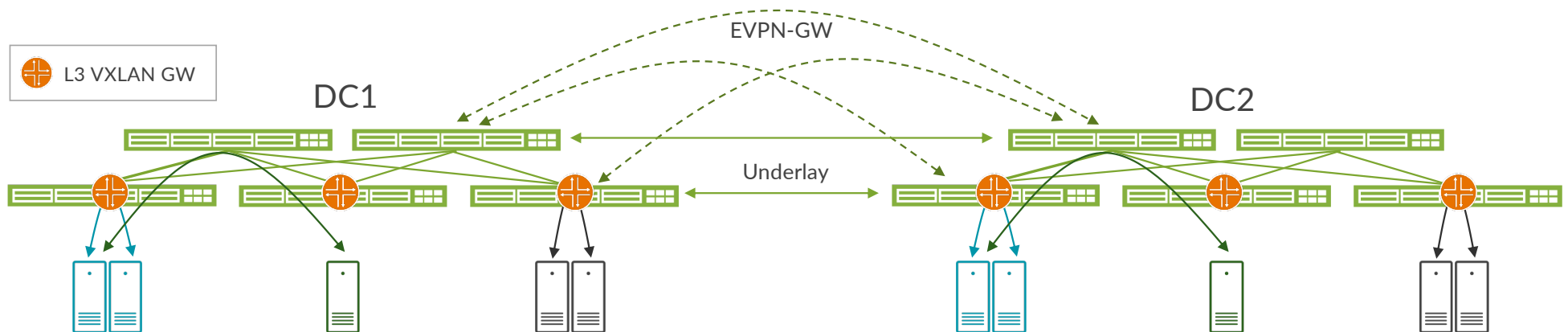
Postman Sample
[Apstra Postman Collections](#)

Python Library
[apstra-api-python Library Reference](#)

サポートするDCI方式

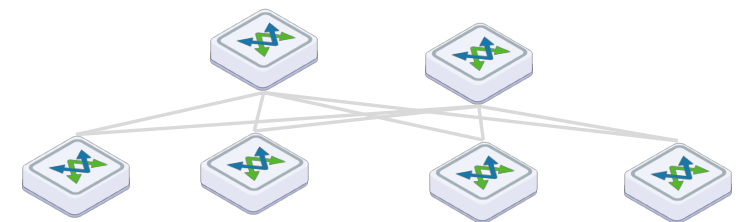
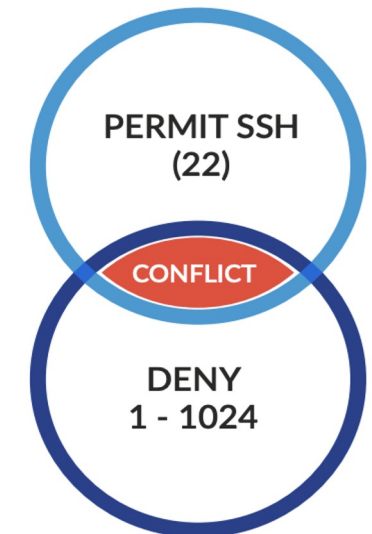
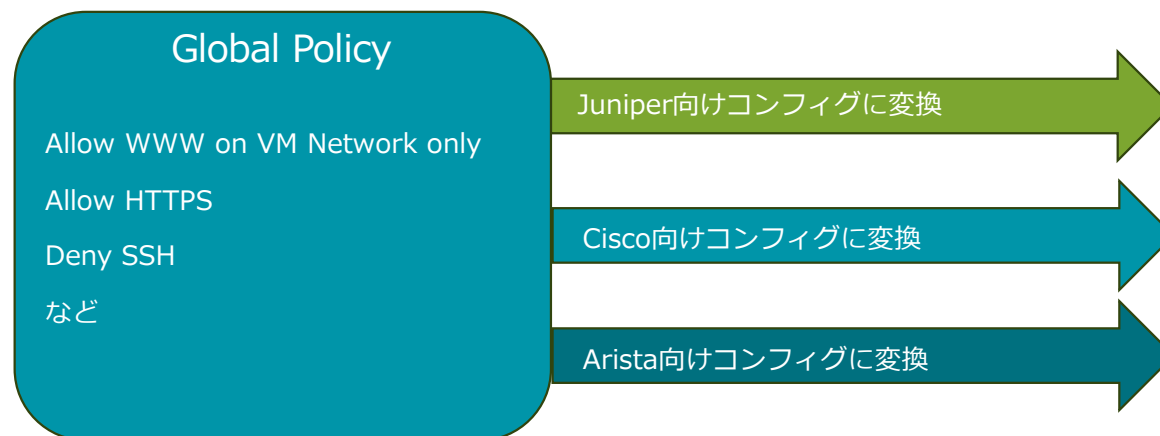
- OTT (Over the Top)
 - インターネットや既存WANを利用してEVPN情報をカプセル化してDC間を接続する方式
 - 分かりやすい設計だがスケールはしづらい
- VXLAN Stitching
 - 異なるネットワークセグメントやVXLANセグメント間を接続する方式
 - Gateway間でEVPVドメインが終端されるためスケールしやすい

DCIで接続する先はApstra管理のDCである必要なし



Security Policy

- ApstraでPolicyでファブリック全体のフィルターポリシーを一元化
- メリット
 - シンプル
 - ファブリック全体でベンダを意識しないACL作成
 - 可視化
 - Policyが適用されるエンドポイントやグループをGUIで表示
 - コンフリクト検知/解決
 - ポリシーにコンフリクトがあると可視化され解決することが可能



Policyの設定方法

1. SourceとDestのエンドポイントを指定する

Application Points

Source Point Type	Destination Point Type
☒ Internal Endpoint	☐ Internal Endpoint
☐ External Endpoint	☒ External Endpoint
☐ External Endpoint Group	☐ External Endpoint Group
☐ Internal Endpoint Group	☐ Internal Endpoint Group
☐ Virtual Network	☐ Virtual Network
☐ Routing Zone	☐ Routing Zone

Source Point	Destination Point
rack-server	external-router

2. RuleにProtocolとActionを指定する

Rules

Name *

deny-icmp

Description

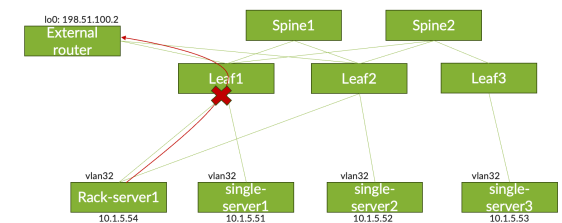
Action *

Deny

Protocol *

ICMP

ポリシーイメージ



3. ACL作成完了

Name	Source Application Point	Destination Application Point	Rule Count	Rule Conflicts	Tags	Enabled	Errors	Actions
inter-external	Internal Endpoint rack-server	External Endpoint external-router	1			ON		

Apstraの機能特徴まとめ

耐障害性の高い標準IPファブリックをintentベースで自動化・可視化

設計移行

- 実現したいネットワーク(Intent)をWebUIからインプット
- SSOTにより設定変更時に誤りのある設定を検知
- 柔軟な構成に対応(3/5 Stage CLOS, 異速度Fabric)
- ハードウェア、OSをマルチベンダで自由に選択
- 1000台以上のスイッチ、複数PODを一つのApstraで管理

運用

- Intentと本番環境の状態をリアルタイムに比較監視(SSOT)
- ロールバック、バージョンアップ、メンテナンスモード
- Apstraの障害、バージョンアップ時に通信影響なし
- 標準機能のみ使用するためブラックボックスなくトラブルシュート可能
- トラフィックの可視化や監視機能が充実



Juniper Apstra™

デザインパターン

- Freeform Architecture



Juniper Apstra™

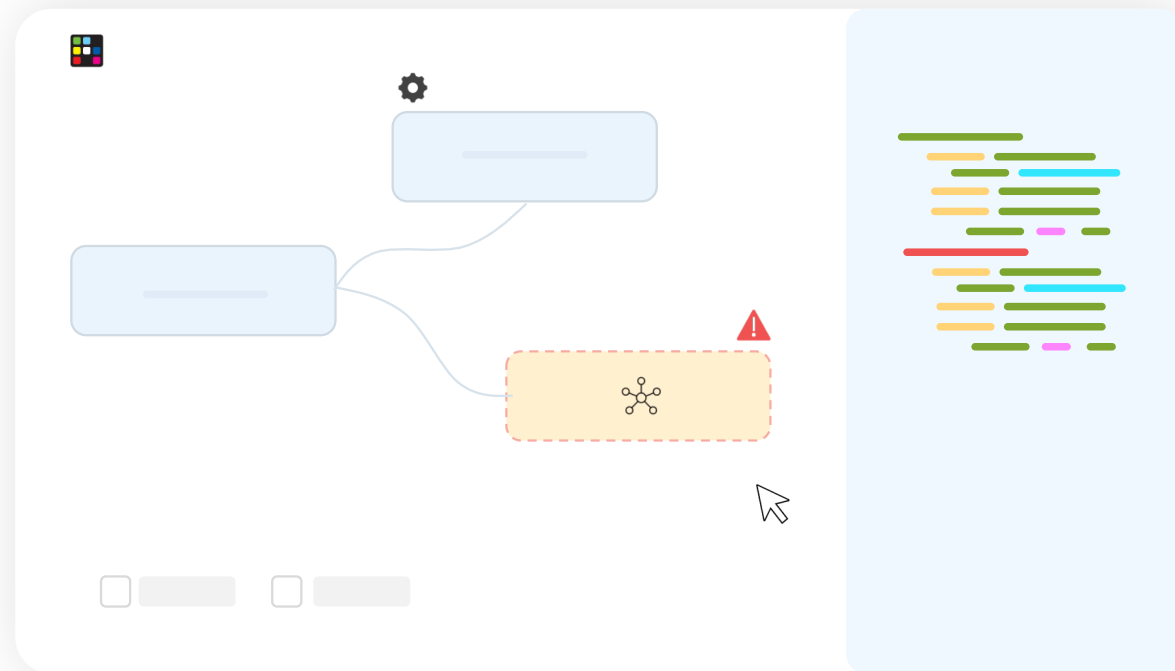
Freeform Overview

どんなデザインでも自動化・可視化基盤として利用可能

- Network Designer & Config Templatesにより柔軟に設定を自動作成
- 可視化、ロールバック、OSアップグレード、など運用に必要な付加価値機能も利用

主要な機能

- Network Designer
- Config Templates & device context
- Property Sets
- Closed-Loop Validation
- Telemetry, IBA
- Time Voyager (Roll Back)
- NOS Upgrades, ZTP,



Config Templates

FreeformではJinja2 Templatesを利用して柔軟な設定対応とデザインに対応します



Jinja2 Confit TemplatesとDevice Contextの
組み合わせで柔軟なConfigを生成

Jinja2 Config Templates

Direct Config

```
1 system {
2   host-name piccadilly-circus;
3 }
4 interfaces {
5   replace: xe-0/0/0 {
6     unit 0 {
7       description "facing_oxford-circus:xe-0/0/0";
8       family inet {
9         address 192.168.0.1/31;
10      }
11    }
12  }
13  replace: xe-0/0/1 {
14    unit 0 {
15      description "facing_green-park:xe-0/0/2";
16      family inet {
17        address 192.168.0.9/31;
18      }
19    }
20  }
21  replace: xe-0/0/2 {
22    unit 0 {
```

Jinja2 Template

```
1 system {
2   host-name {{hostname}};
3 }
```

Nesting - Composable

```
1 {% include "system.jinja" %}
2
3 {% include "interfaces.jinja" %}
4
5 {% include "protocols.jinja" %}
```



Device Context

Topology Editor

Device Profiles

Resource Manager

Property Sets

Tags

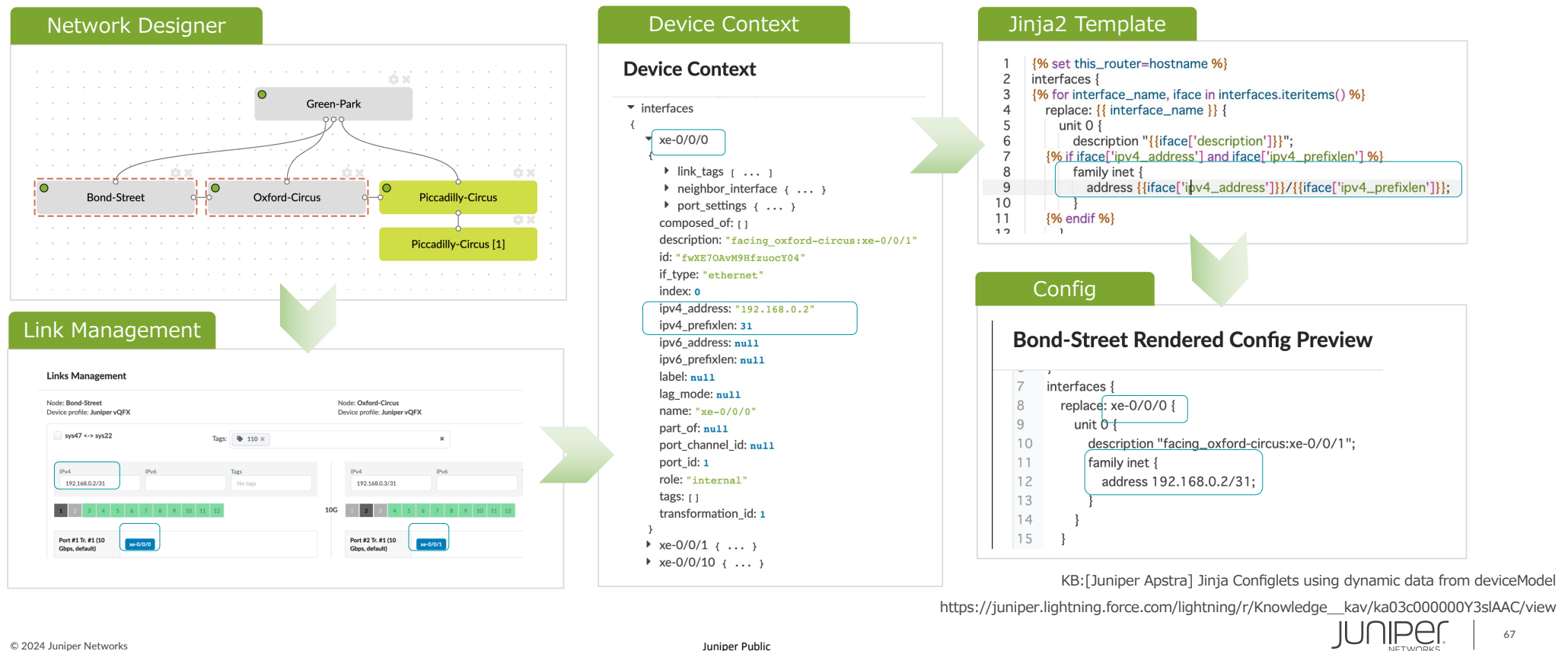
SSOT = GraphDB

※ Jinja2 Templates形式でなくJUNOS CLIをそのまま貼り付けてデバイスに適用することも可能です

Device Context

Network DesignerやDevice Contextの情報を変数値としてコンフィグの自動生成

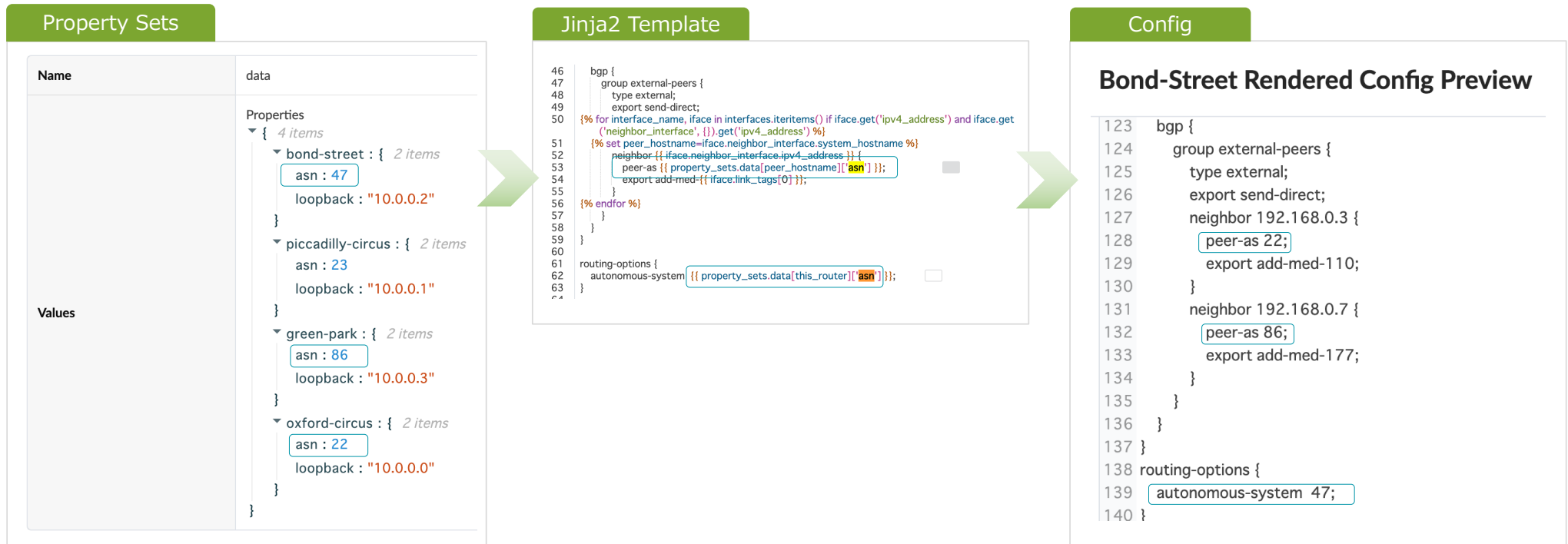
例: Network Designerでアサインしたアドレスから動的にコンフィグ作成



Property Sets

各デバイス固有の静的変数値を設定し、Jinja2で参照してコンフィグを生成できます

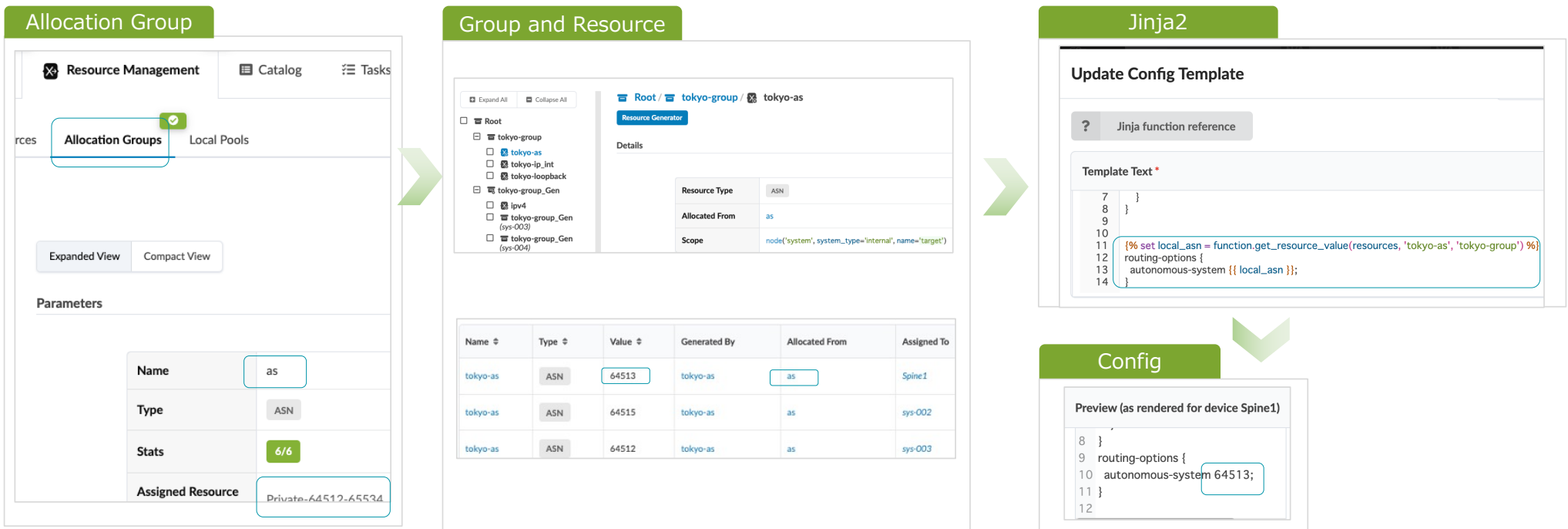
例:各デバイス毎のas番号をproperty setとして定義し、動的にコンフィグの生成



Resource Management

リソースのレンジから動的に各デバイスシステムやリンクにアドレスを自動で割り当て
動変数を用いたConfigを作成することができます

例: リソースレンジから各デバイス毎のas番号を動的にアサインし、動的にコンフィグの生成



Tag Based Dynamic Configuration

タグレベルにより整理、検索、参照などに活用できるほか、Config Templates では
コンフィグ生成時に**動的変数**として設定することもできます

例:リンクにタグ情報を付与し、タグ情報を元にBGPのMEDを設定

Tag

Dashboard Analytics Staged Unco

Topology Systems **Links**

Query: All

Filter selected by ☒ all ☐ selected only ☐ unselected only

	Name	Type	Tags	Speed	Role	Name	Interface Name
☐							Endp
0 selected							
☐	sys47 <-> sys22	Physical	110	10G	internal	Bond-Street	xe-0/0/0
☐	sys47 <-> sys86	Physical	177	10G	internal	Bond-Street	xe-0/0/1

Config Template

```
1 [% set this_router = hostname %]
2 [% set med_tags = namespace(meds=[]) %]
3 policy-options {
4   policy-statement send-direct {
5     term 1 {
6       from protocol direct;
7       then accept;
8     }
9   }
10 }
11
12 [% for interface_name, iface in interfaces.iteritems() %]
13 {#
14   First, building a list of link_meds so we can later deduplicate the
15   add-med-<value> policy statements
16   #}
17   [% do med_tags.meds.extend(iface.link_tags) %]
18   [% endfor %]
19 }
20 [% for link_med in med_tags.meds|sort|unique %]
21 {
22   policy-statement add-med-{{ link_med }} {
23     from {
24       route-filter 0.0.0.0/0 longer;
25     }
26   }
27 }
28
29 bgp {
30   group external-peers {
31     type external;
32     export send-direct;
33   }
34   [% for interface_name, iface in interfaces.iteritems() if iface.get('ipv4_address') and iface.get('neighbor_interface', []).get('ipv4_address') %]
35   {
36     [% set peer_hostname=iface.neighbor_interface.system_hostname %]
37     neighbor {{ iface.neighbor_interface.ipv4_address }} {
38       peer-as {{ property_sets.data[peer_hostname]['asn'] }};
39       export add-med-{{ iface.link_tags[0] }};
40     }
41   }
42 }
```

Config

Bond-Street Rendered Config Preview

```
88 }
89 policy-statement add-med-110 {
90   from {
91     route-filter 0.0.0.0/0 longer;
92   }
93   then {
94     metric add 110;
95   }
96   then {
97     accept
98   }
99 }
100 policy-statement add-med-177 {
101   from {
102     route-filter 0.0.0.0/0 longer;
103   }
104   then {
105     metric add 177;
106   }
107 }
108
109 bgp {
110   group external-peers {
111     type external;
112     export send-direct;
113     neighbor 192.168.0.3 {
114       peer-as 22;
115       export add-med-110;
116     }
117     neighbor 192.168.0.7 {
118       peer-as 86;
119       export add-med-177;
120     }
121   }
122 }
```

Innovative Interactive Editor

Config templateをInteractiveに作成、編集が可能です

Update Config Template

Name *

interfaces.jinja

Config Preview

Device Context

System: Bond-Street

Apply Mode: Complete

Preview

Template Text *

```
1 {% %}
2 interfaces {
3   {% for interface_name, iface in interfaces.iteritems() %}
4     replace: ({{ interface_name }}) {
5       unit 0 {
6         description "{{iface['description']}}";
7         {% if iface['ipv4_address'] and iface['ipv4_prefixlen'] %}
8         family inet {
9           address {{iface['ipv4_address']}}/{{iface['ipv4_prefixlen']}};
10        }
11      {% endif %}
12    }
13  }
14  {% endfor %}
15  replace: lo0 {
16    unit 0 {
17      family inet {
18        address {{ property_sets.data[this_router]['loopback'] }}/32;
19      }
20    }
21  }
```

Preview (as rendered for device Bond-Street)

i

Config is empty.

Save Changes

Update

Jinja function reference Document

Config Templateを作成する際の参考ドキュメント

Apstra Jinja Functions documentation »

Table of Contents

Apstra Jinja helper functions reference

Indices and tables

Next topic

Jinja Common Filters

Quick search

Go

Apstra Jinja helper functions reference

This document serves as a general reference against custom jinja filters, tests, and functions as related to configuration templates and configlets.

Consult <https://jinja.palletsprojects.com/en/2.11.x/templates/> for general Jinja template designer details.

- Jinja Common Filters
 - bool
 - cidr_merge
 - intersect
 - json_query
 - to_cidr
 - to_ip
 - to_mac_format
 - to_netmask
 - to_network
 - to_prefixlen
- Jinja Common Functions
 - gen_acl_port_token
 - max_os_version
 - merge_vlans_to_list
 - min_os_version
 - raise_error
 - re_findall

Example: Extract two resource values from group underlay

Jinja for example *Extract two resource values from group underlay*

```
{% set loopback_ipv4 = function.get_resource_value(resources, 'ipv4_loopback', 'underlay') %}
{% set bgp_asn = function.get_resource_value(resources, 'bgp_asn', 'underlay') %}
Loopback ipv4: {{loopback_ipv4}} BGP ASN: {{bgp_asn}}
```

Device model context for example *Extract two resource values from group underlay*

```
{
  "resources": [
    {
      "tags": [],
      "label": "underlay",
      "groups": [],
      "data": {},
      "id": "test_group_id-underlay",
      "resources": [
        {
          "id": "test_res_id-ipv4-loopback-10.0.0.1/32",
          "assigned_to": [],
          "resource_type": "ip",
          "value": "10.0.0.1/32",
          "label": "ipv4_loopback"
        },
        {
          "id": "test_res_id-bgp_asn-64512",
          "assigned_to": [],
          "resource_type": "asn",
          "value": "64512",
          "label": "bgp_asn"
        }
      ]
    }
  ]
}
```

Expected text output for example *Extract two resource values from group underlay*

Loopback ipv4: 10.0.0.1/32 BGP ASN: 64512

豊富な運用を支援する機能

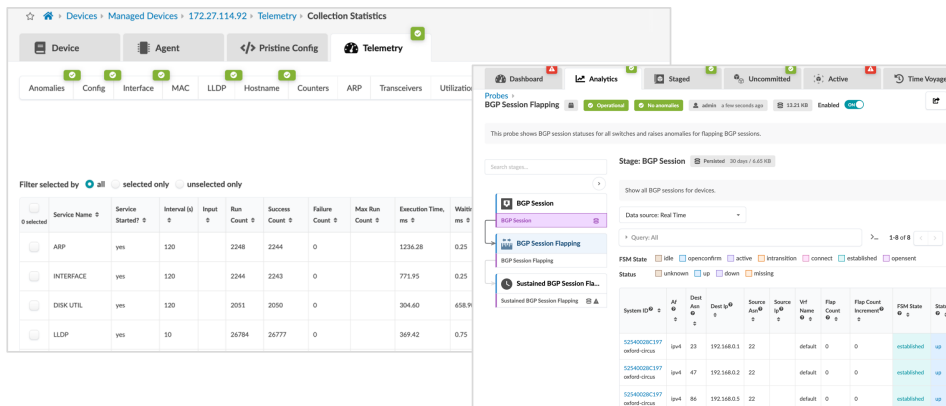
Dashboard

ネットワーク全体の健康状態の把握



Analytics

BGPセッション、Memory、CPUなど各種パラメーターを取得可能



そのほかにも・・・

Time Voyager (Roll Back)

The Time Voyager (Roll Back) section allows users to view and manage historical data. It features a table with columns for Description, Created At, User, and Actions. The table lists several actions, including 'change color', 'Device Remove', and 'change to new-system.json', with their respective creation times and users.

Device OS Upgrade

The Device OS Upgrade section provides a user interface for upgrading the operating system of network devices. It includes a 'Create Onbox Agent(s)' button, a 'Create Offbox Agent(s)' button, and an 'Advanced Settings' button. Below these buttons, there is a 'Query: All' input field and a 'Device' dropdown menu. The 'Filter selected by' section shows 'Upgrade OS Image' and 'unselected only'. The 'Device Information' section includes a 'Collect Show Tech' button and a 'Query: All' input field.

Collect Show Tech

The Collect Show Tech section provides a user interface for collecting technical data from network devices. It includes a 'Choose log source' section with checkboxes for 'AOS Controller', 'Include Backup', and 'Managed Devices'. Below this section, there is a 'Query: All' input field and a 'Page Size' dropdown menu.

ZTP

The ZTP (Zero Touch Provisioning) section provides a user interface for provisioning network devices. It includes a 'Query: All' input field and a 'Page Size' dropdown menu. Below these elements, there is a table with columns for Service Name, IP Address, Service Status, and Last Updated.

Apstra Flow

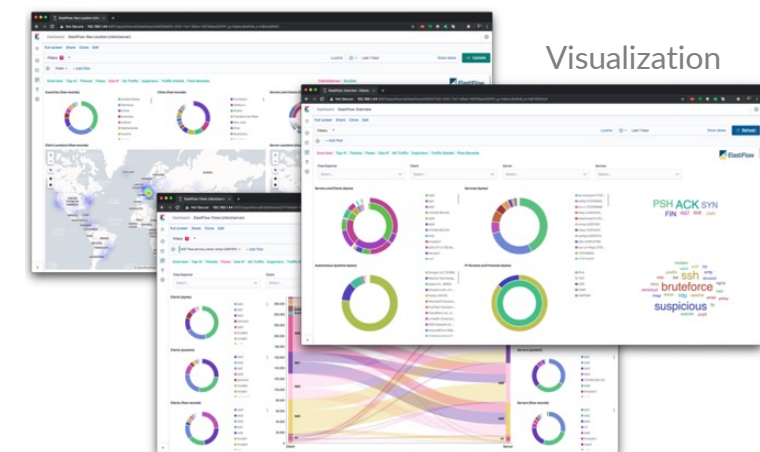


Juniper Apstra™

Day 1/2

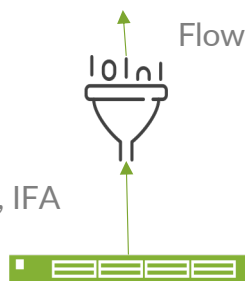
フローモニタリング : Apstra Flow

実トラフィックのフロー情報・可視化し、トラフィックの正常性や傾向分析



Flow collector

sFlow, NetFlow, IPFIX, IFA



MGMT


Juniper Apstra™

*Apstra Flowは別VMとして起動必要

Juniper Public

Data Plane & Flow

Traffic Related

ワークロードが送信しているトラフィックの種類
最大のアプリ/エンドポイント消費者
エンドポイント間のアプリ（フロー）
IPv6のトラフィック量

Apstra Flow サポートハイパーバイザー

ハイパーバイザー	サポートバージョン
Vmware vSphere	7.0, 6.7, 6.5, 6.0
Microsoft Hyper-V	Windows Server 2016 Datacenter Edition
Linux KVM	Ubuntu 22.04, 18.04 LTS

最新のサポート状況は以下をご確認ください。

<https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-flow-data-install-guide/topics/concept/apstra-flow-data-install-supported-hypervisors.html>

Apstra Flow サイジング

サイズ	必要リソース	サポートデバイス数
Small 最大 5,000 FPS	8 vCPU, 32 GB memory, 4 TB disk	最大 : 25 デバイス 1 デバイスあたり平均200 FPS
Medium 最大 10,000 FPS	16 vCPU, 64 GB memory, 4 TB disk	最大 : 50 デバイス 1 デバイスあたり平均200 FPS
Large 最大 15,000 FPS	24 vCPU, 64 GB memory, 4 TB disk	最大 : 75 デバイス 1 デバイスあたり平均200 FPS
X-Large (custom)	N/A	15,000 FPS以上はJuniper社へ お問い合わせください

最新のサポート状況は以下をご確認ください。

<https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-flow-data-install-guide/topics/concept/apstra-flow-data-install-scaling-considerations.html>

Apstra Flow サポート プロトコル/デバイス

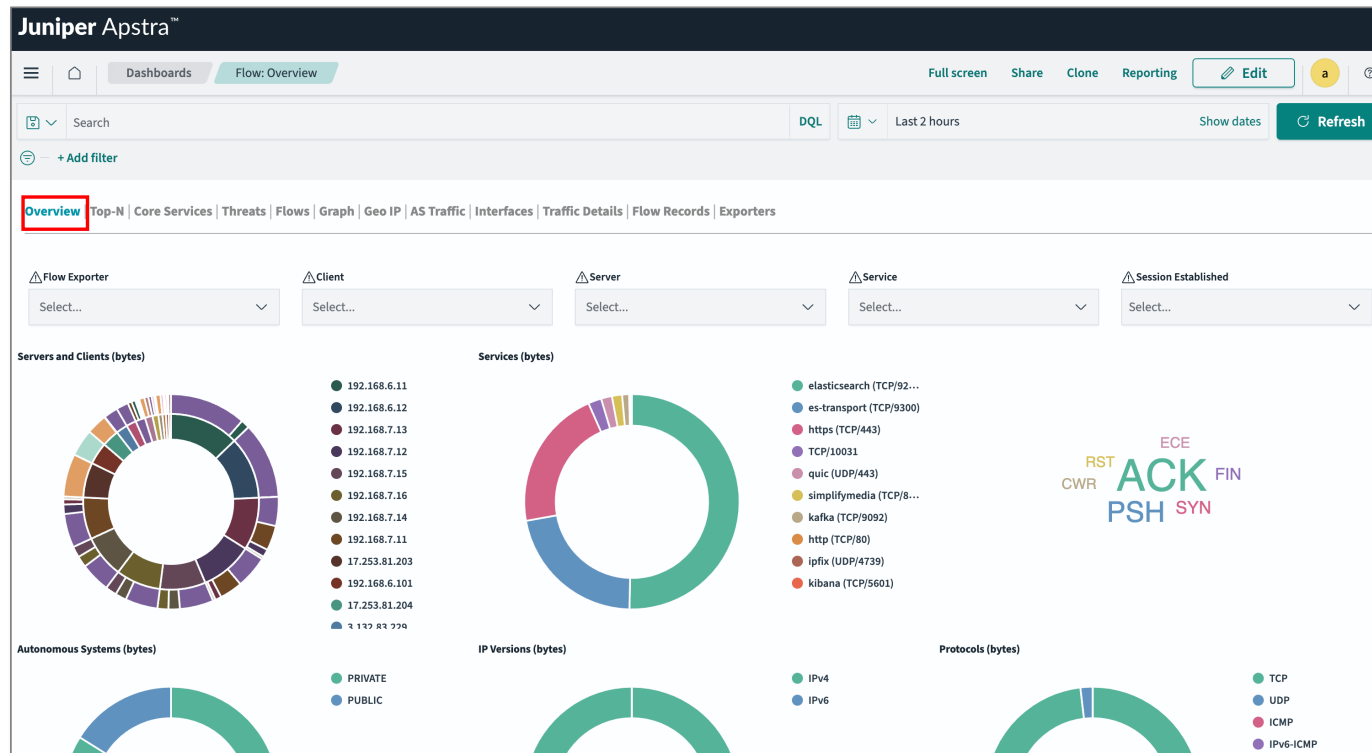
- サポートプロトコル
 - sFlow, NetFlow v1, v5, v6, v7, v9, IPFIX, and IFA protocols.
- サポートデバイス(OS)
 - Juniper Junos OS
 - Juniper Junos Evolved OS (EVO)
 - 注: EVOデバイスはsflowをManagement Interfaceから送信できません。
 - Arista EOS
 - Cisco NXOS
 - Enterprise SONIC devices.

最新のサポート状況は以下をご確認ください。

<https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-flow-data-install-guide/topics/concept/apstra-flow-data-install-device-support.html>

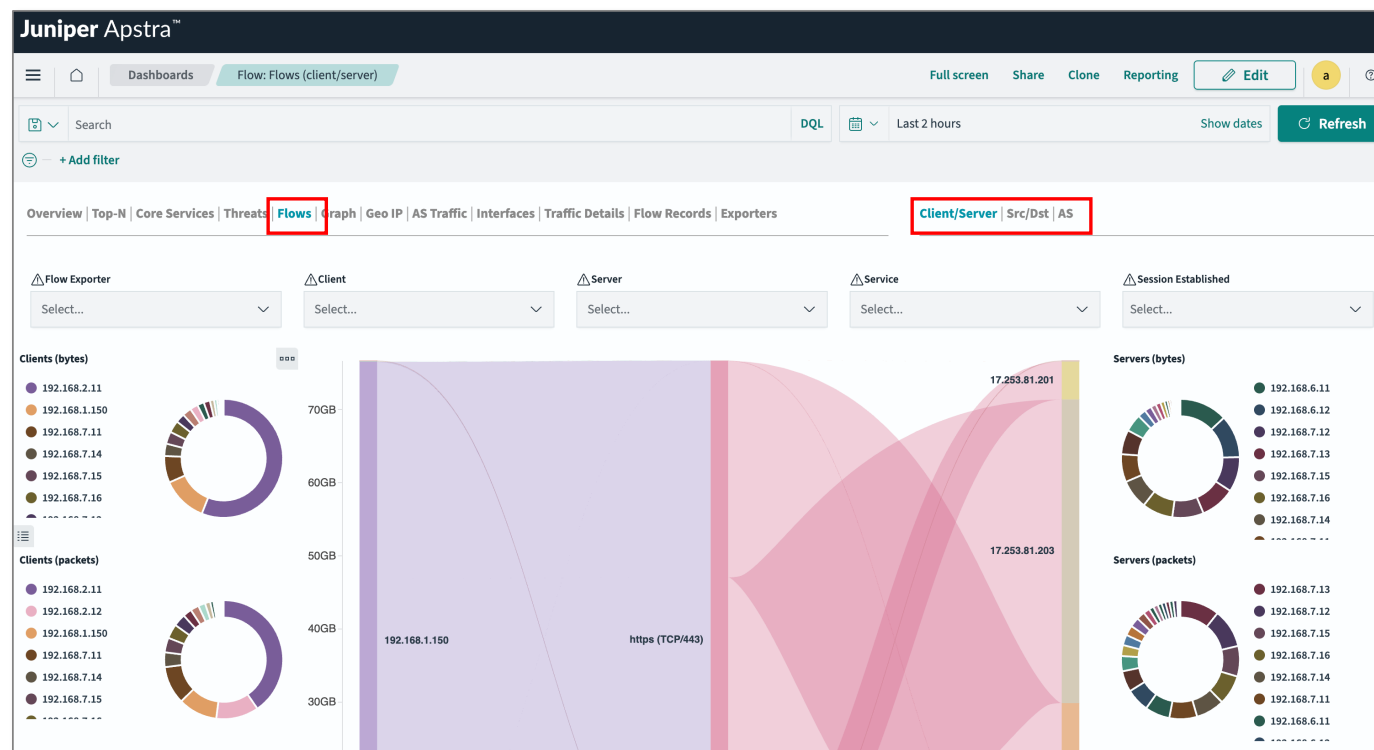
Overview Dashboard

- Overview Dashboard
 - Apstra Flow Collectorで収集している全てのFlow Dataのサマリーダッシュボード
 - Flow Exporter, Client, Server, Service, SessionでFlowの絞り込みが可能



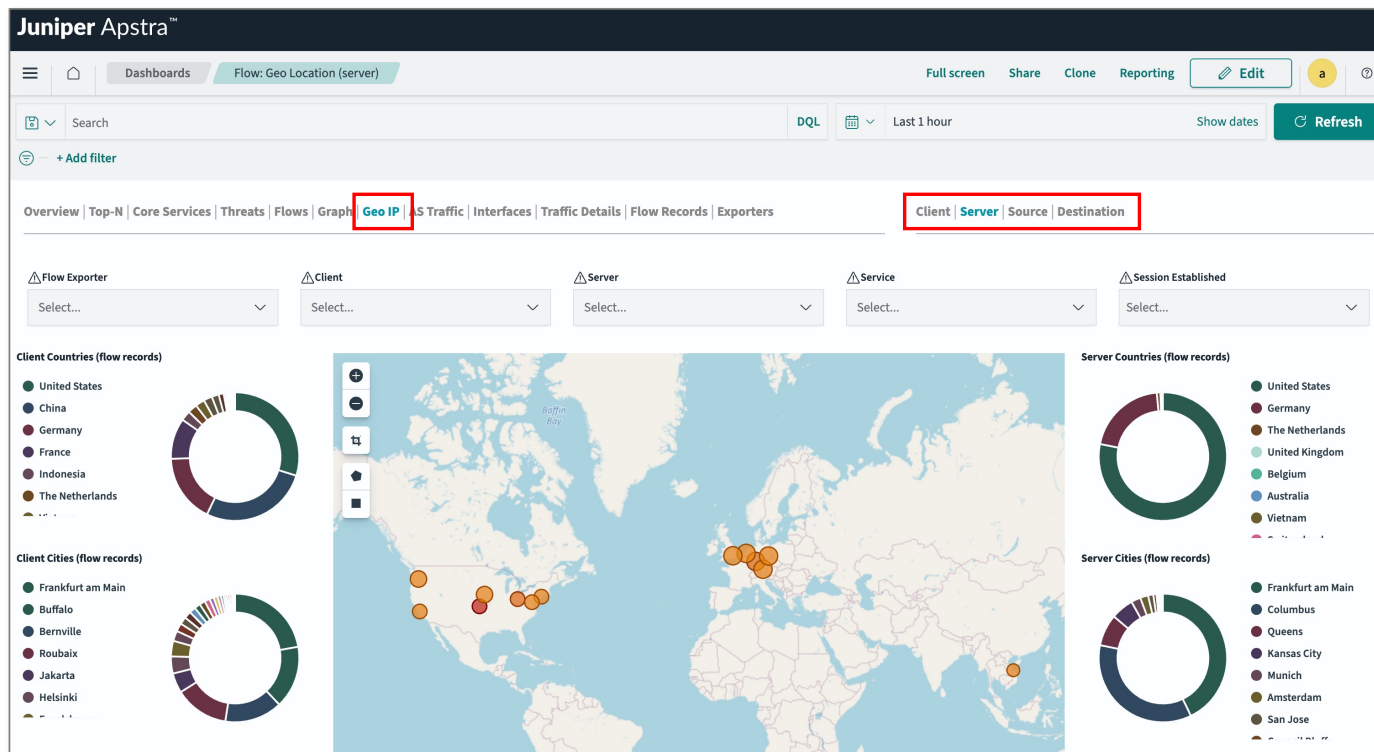
Flow Dashboard

- Flow Dashboard
 - Client, Server, Serviceを指定したFlowの検索



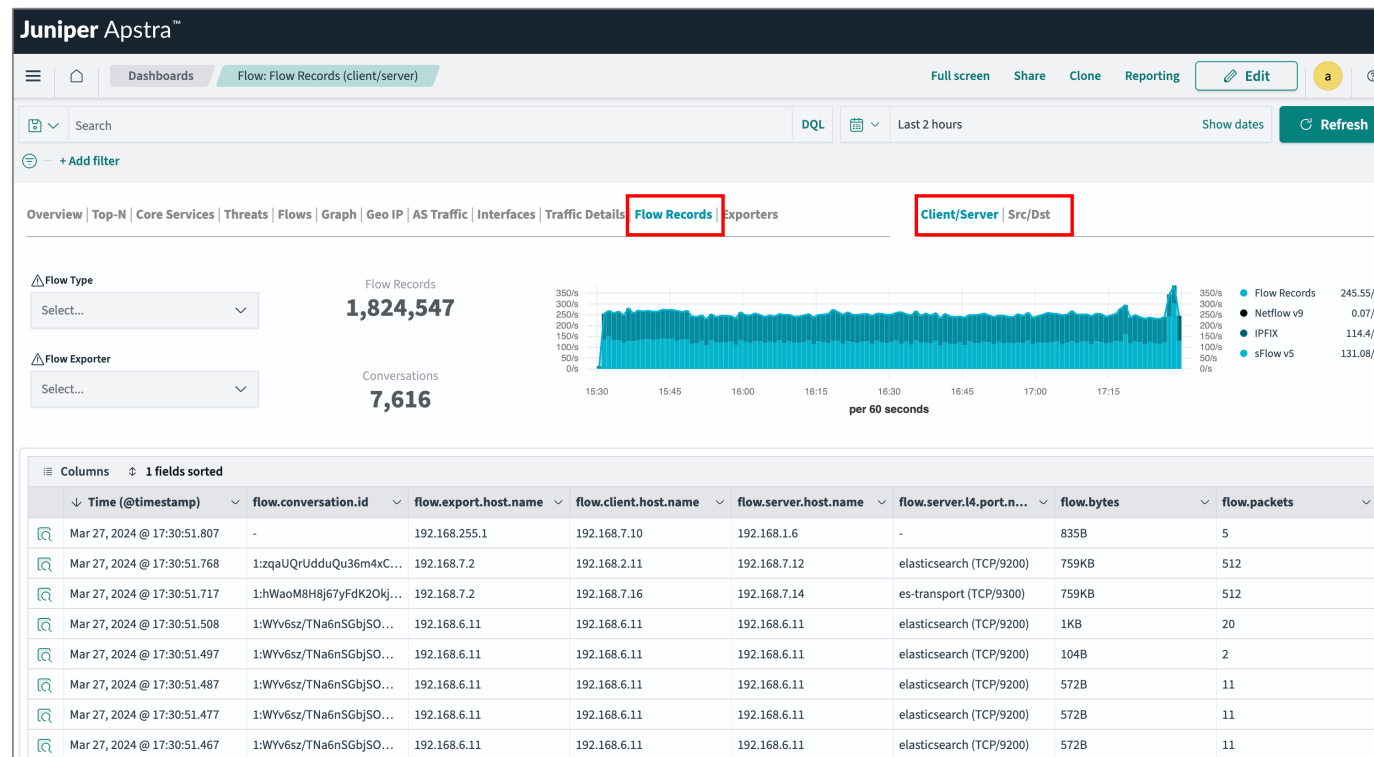
Geo IP Dashboard

- Geo IP Dashboard
 - ASN, IP情報を元にGeo Location表示



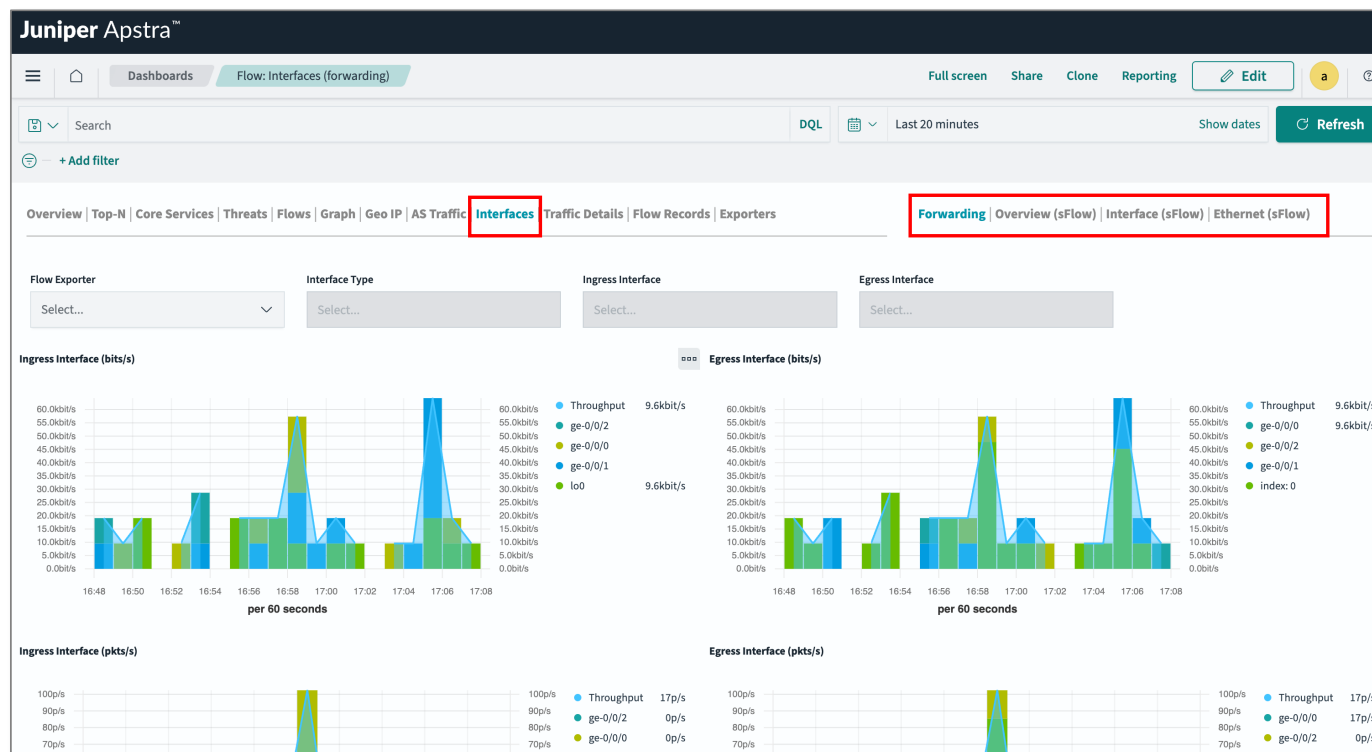
Flow Records Dashboard

- Flow Records Dashboard
 - FlowType(IPFIX, Netflow, sFlow)単位のフローレコード表示



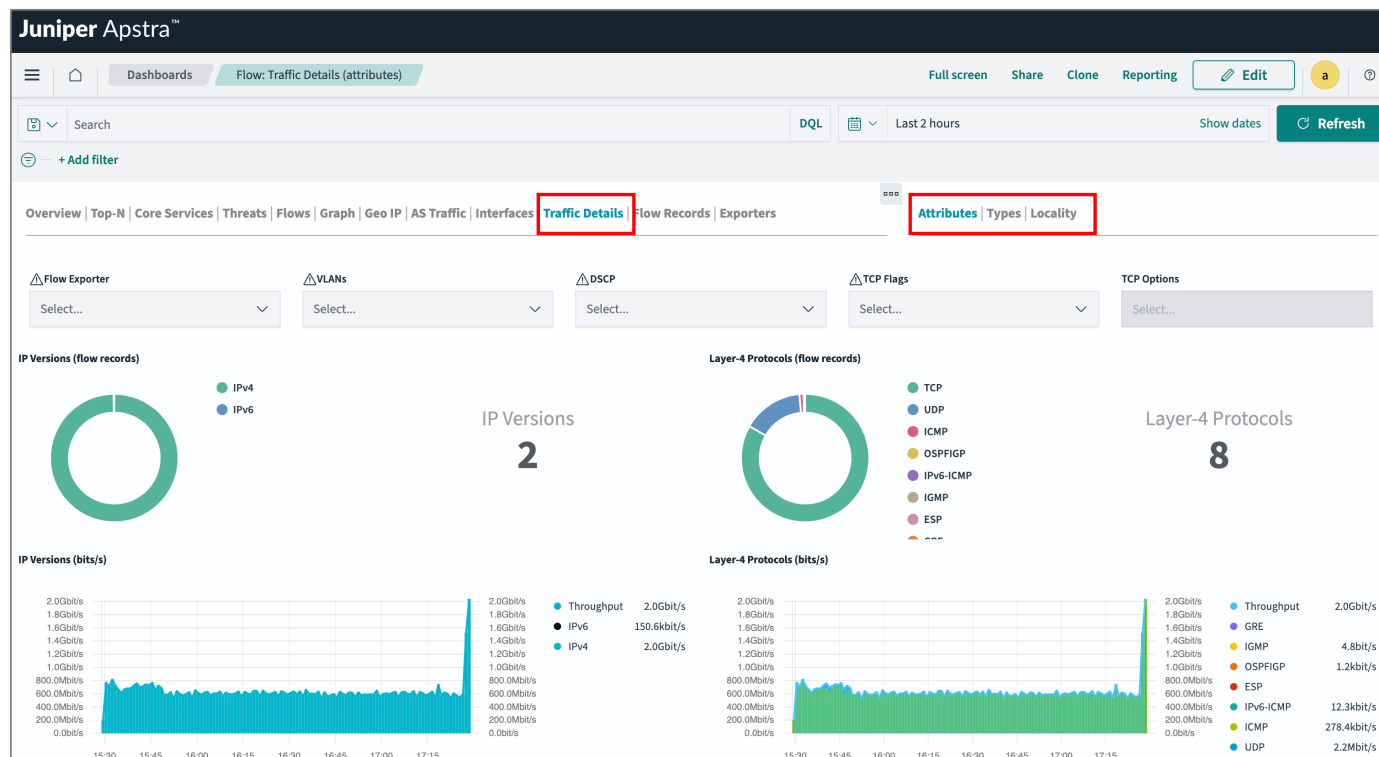
Interface Dashboard

- Interface Dashboard
 - Interface単位のThroughput、Error/Discard Counterの表示



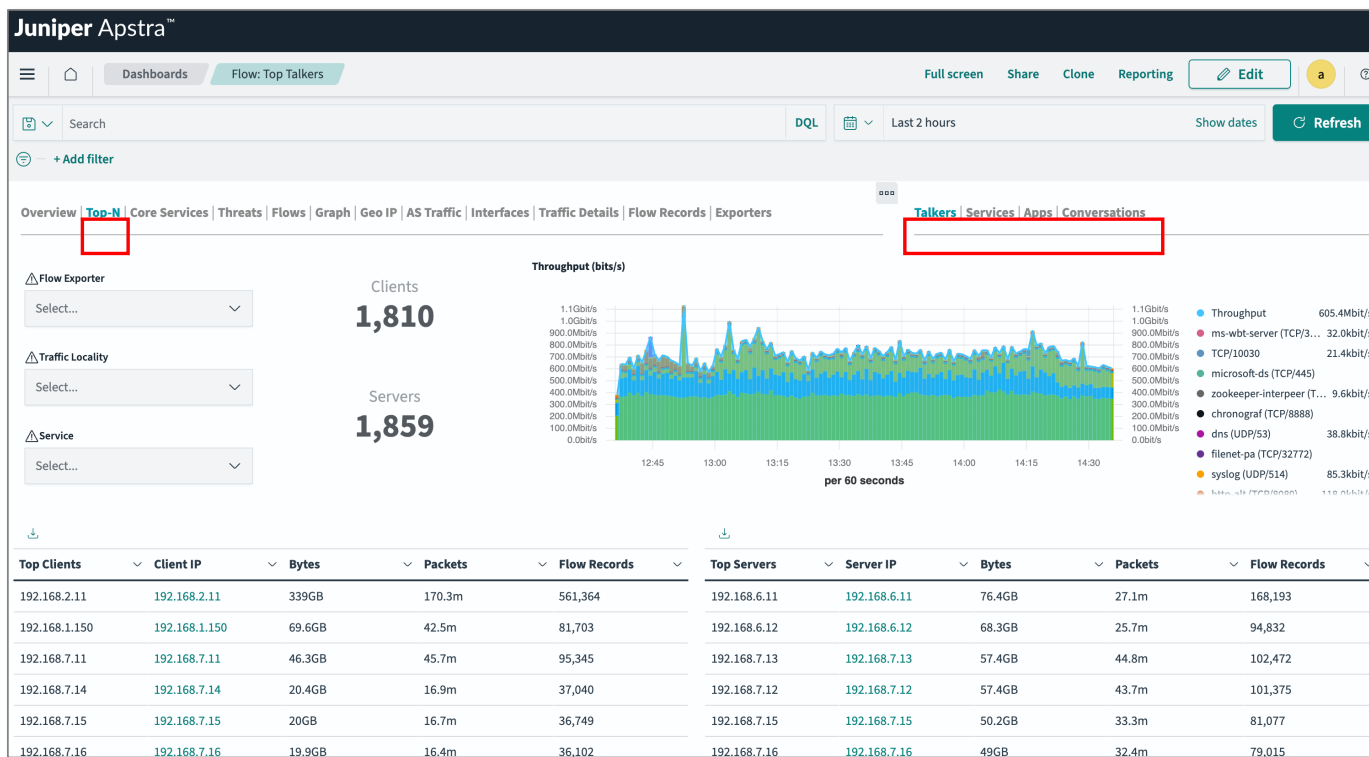
Traffic Details Dashboard

- Traffic Details Dashboard
 - VLAN, DSCP, TCP Flag, L4 Protocol, Service単位のTraffic表示



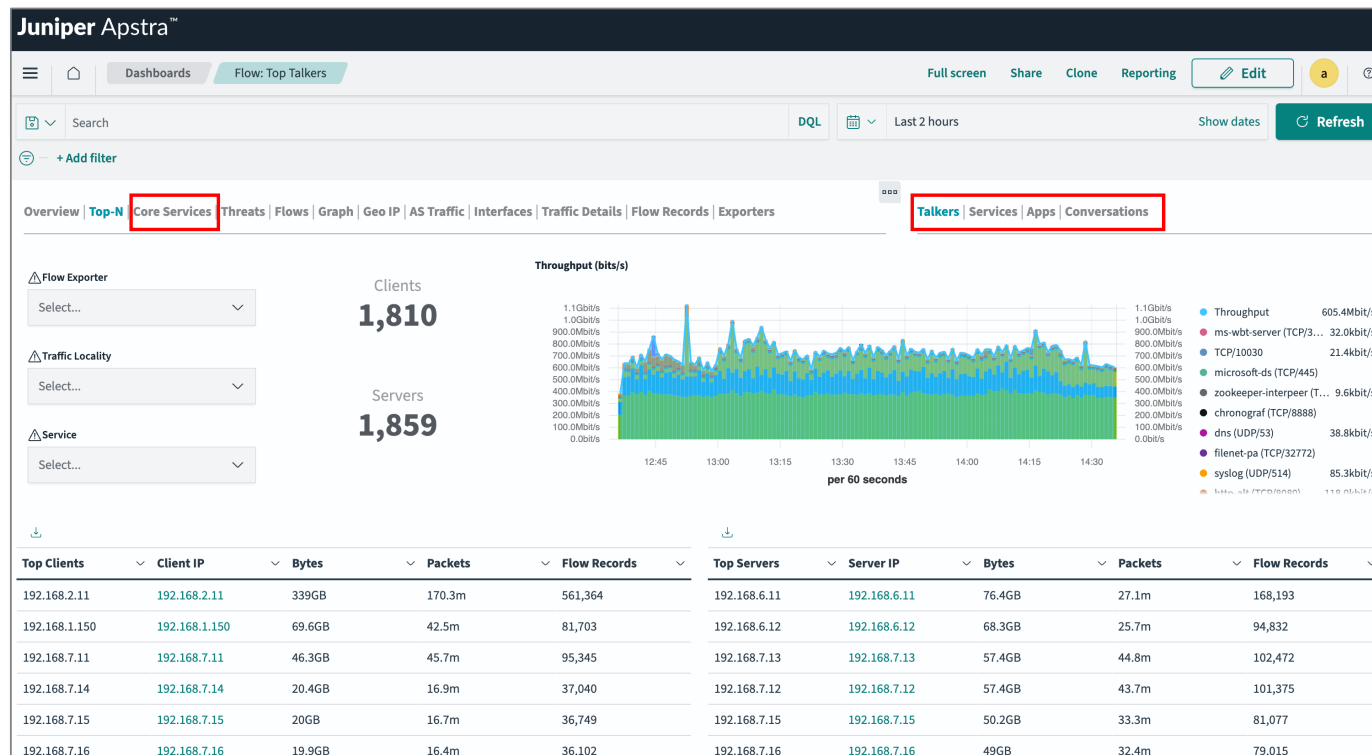
Top-N Dashboard

- Top-N Dashboard
 - 使用率の高いTalkers(Client/Server), Service, Application, Conversationを上位順に表示



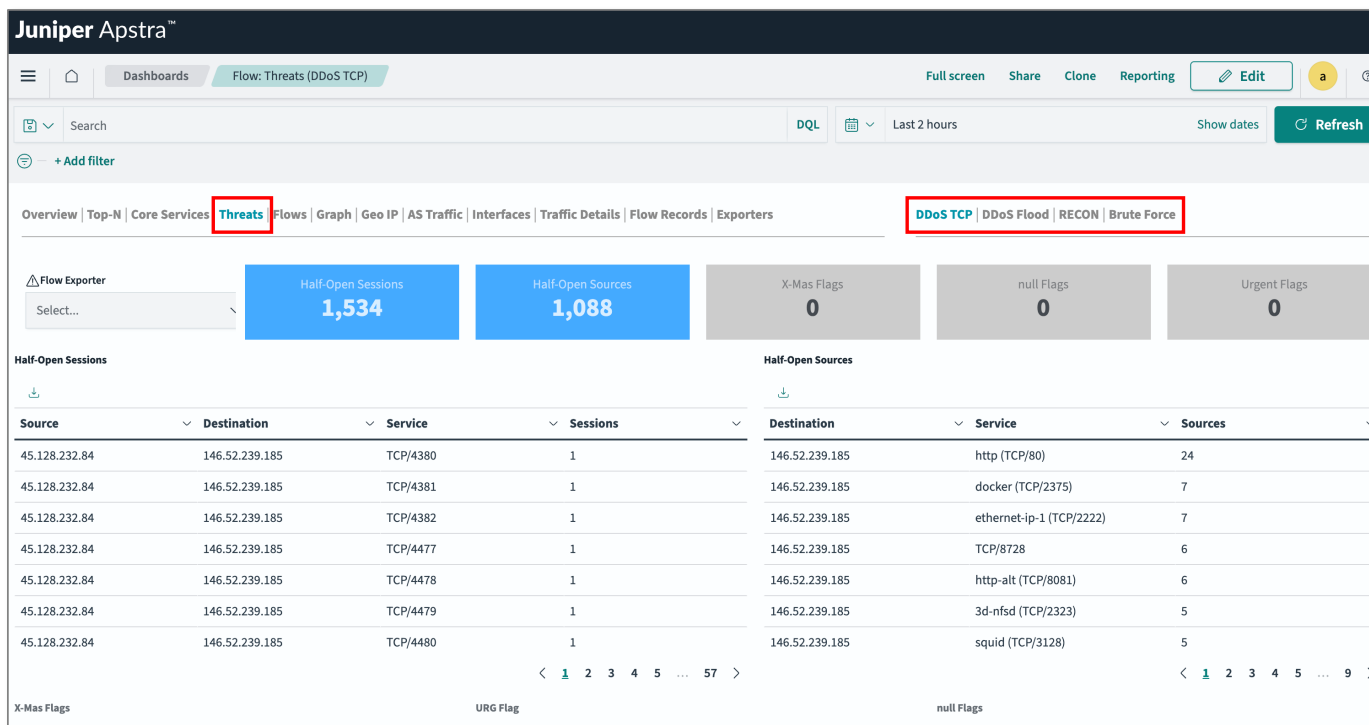
Core Services Dashboard

- Core Services Dashboard
 - DNS, DHCP, Radius, LDAP, NTPのリクエスト、レスポンスのFlow表示



Threats Dashboard

- Threats Dashboard
 - DDoS TCP, DDoS Flood, RECON, Brute ForceのFlow表示

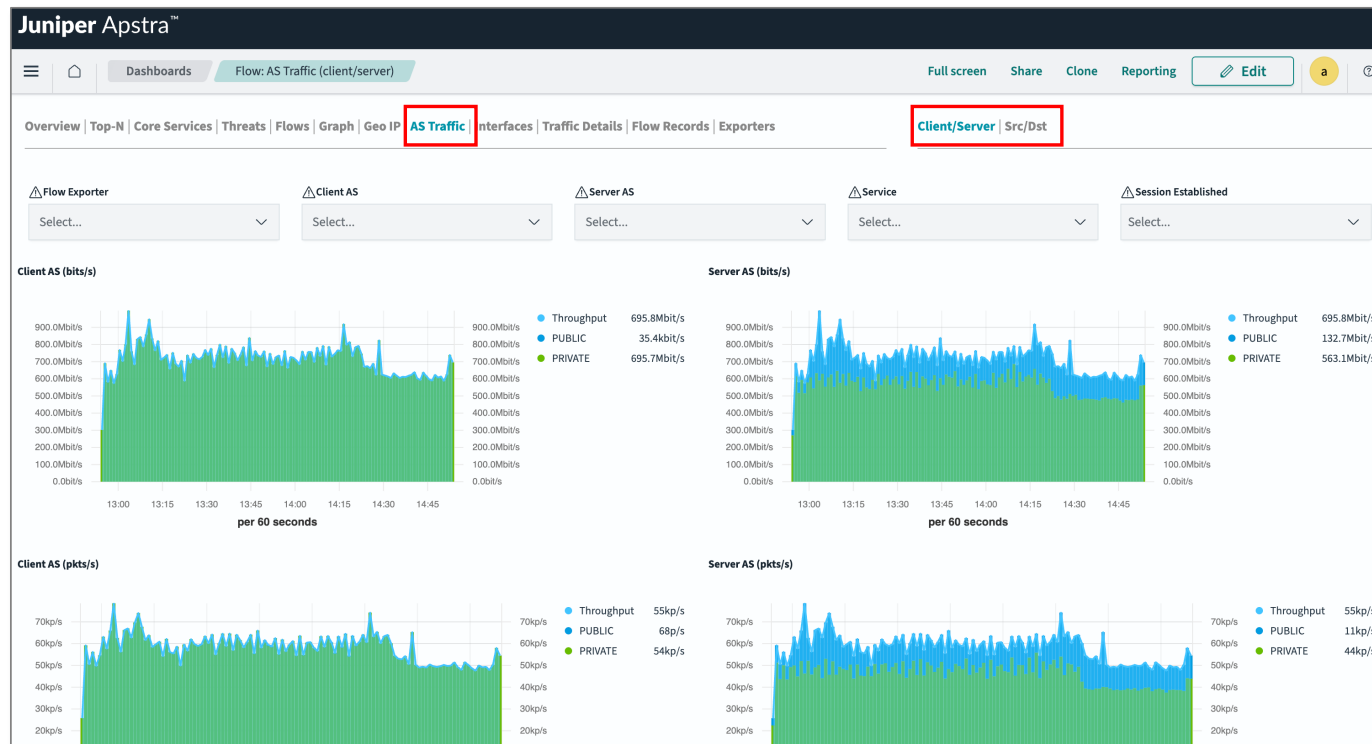


- Graph Dashboard
 - Client, Server, Serviceを指定したFlowをグラフ表示



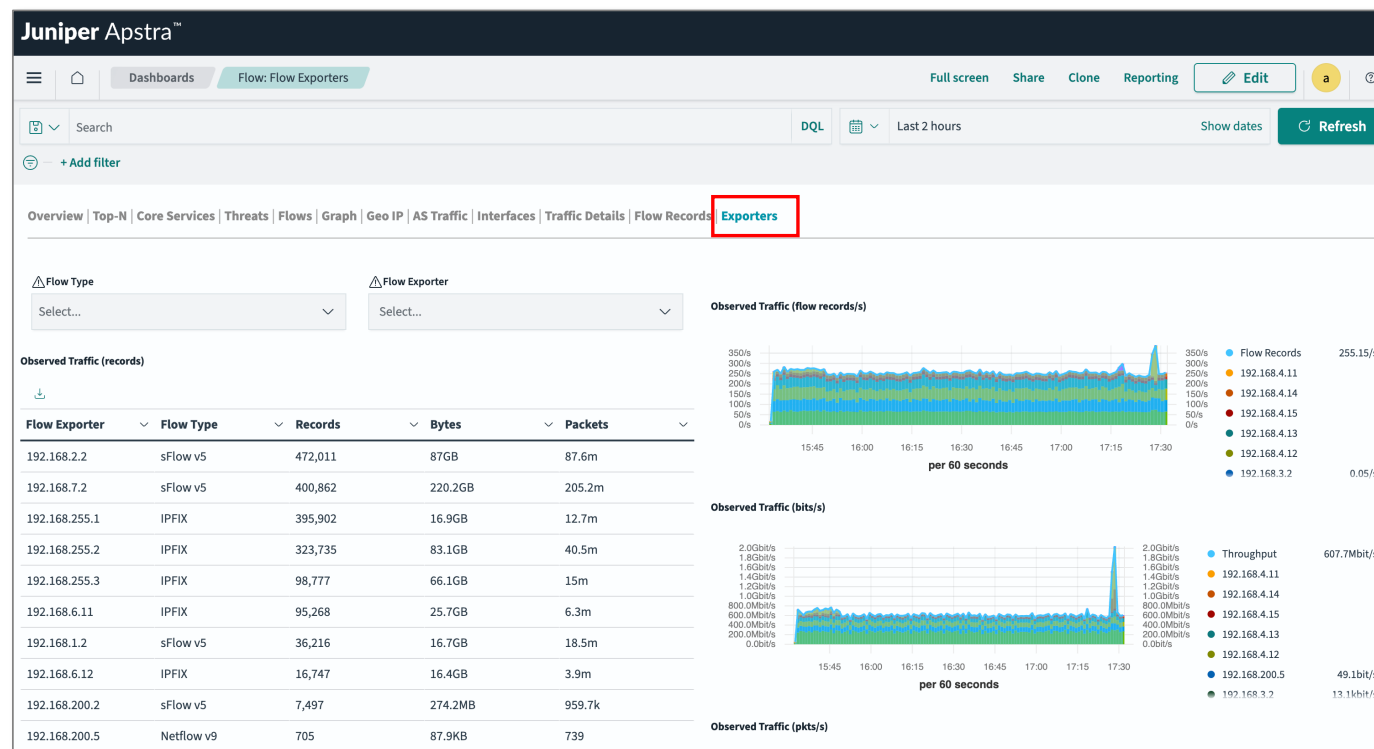
AS Traffic Dashboard

- AS Traffic Dashboard
 - Private AS, Public AS毎のTraffic表示



Exporters Dashboard

- Exporters Dashboard
 - FlowType(IPFIX, Netflow, sFlow)、Flow送信元単位のTraffic Record Traffic表示



Custom telemetry



Juniper Apstra™

IBA – Intent Based Analytics

パラメータモニタリングもデフォルトからカスタマイズまで対応

1. Default probes

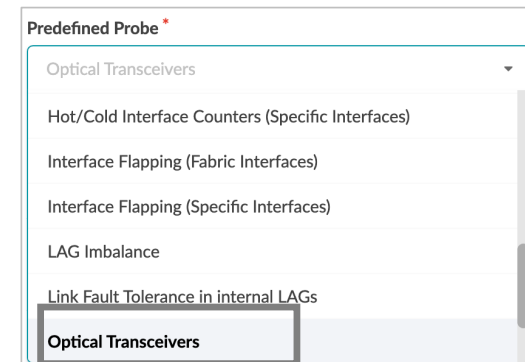
- ブループリントをデプロイするとデフォルトで有効化

2. Pre-defined probes

- カタログからオンデマンドに有効化

3. Custom probes

- ユースケースが1と2でカバーできない場合に新たなProbeを作成
- 初期リリース時はJunos/Junos EVOのみに対応



Pre-defined Probeをは選択



その他のプロトコル: OSPF, IS-IS, RSVP
カーネルまたはプロセスに関する問題
特定のパスやフィールドに基づくセッション統計

JUNIPER
NETWORKS

Default telemetryとIBAの違い

• Default telemetry

- デフォルトで有効化されるtelemetry情報
- 取得する情報やアラート条件はカスタマイズできない
- 可視化もカスタマイズできない
- 履歴も持たない

• Intent Based Analytics (IBA)

- 取得情報の履歴をサポート
- 取得する情報やアラートの閾値などをカスタマイズ可能
- 可視化スタイルをカスタマイズ可能(widgets, dashboards)

• Custom telemetryとは

- IBAで取得する情報をshowコマンドから追加できる機能

事前 to 取得したい情報のshowコマンドを特定

Execute CLI Command

S/N: S2540011DEFF Management IP: 10.28.126.13 Hostname: leaf1

show route summary | match " RIB | FIB | VRF "

Highwater Mark (All time / Time averaged watermark)
RIB unique destination routes: 685 at 2023-05-22 11:12:33 / 571
RIB routes : 1063 at 2023-05-22 11:13:17 / 870
FIB routes : 67 at 2023-05-22 11:09:36 / 36
VRF type routing instances : 4 at 2023-05-22 10:34:13

Text Mode
Text Mode
XML Mode
JSON Mode

Execute

クリップボードにコピー

“Show”コマンドのみサポート
コマンドの自動補完をサポート
パイプでコマンド結果のフィルター可能
XMLやJSON形式もサポート

IBAのTelemetry作成

1. 事前に特定していたshow
コマンドを指定
2. XMLアウトプットをチェック
3. アウトプットの中の取得した
いフィールドを指定

The screenshot shows the 'Create Telemetry Collector' wizard in Juniper IBA. The wizard is at the 'Command' step, with 'Service' and 'Platform' steps completed. It shows a search bar and a 'Browse' button. A message box prompts the user to select a command by typing in the search field or clicking the Browse button. The 'Next' button is visible at the bottom right.

Reports

- ヒストリー分析とトレンド予測

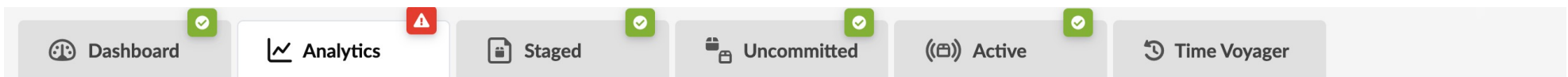


Juniper Apstra™

Reports

・ヒストリー分析とトレンド予測

– レポートできる種類は今後追加される予定

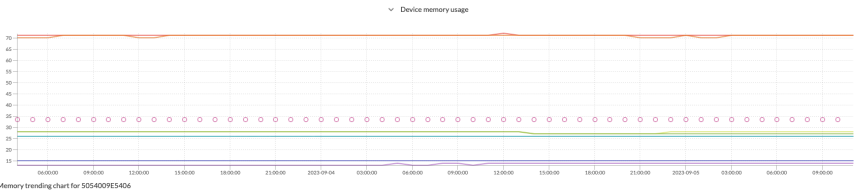


Name	Description	Required Predefined Probes	Actions
Device Health	Analyze device health	▶ Device System Health ✓ ▶ Device Telemetry Health ✓	
Optical XCVR	Analyze optical transceivers telemetry patterns and trends	▶ Optical Transceivers ⊗ ▶ Device Traffic ✓	
Traffic	Analyze device traffic patterns and trends	▶ Device Traffic ✓ ▶ Device System Health ✓	

Device Health Report

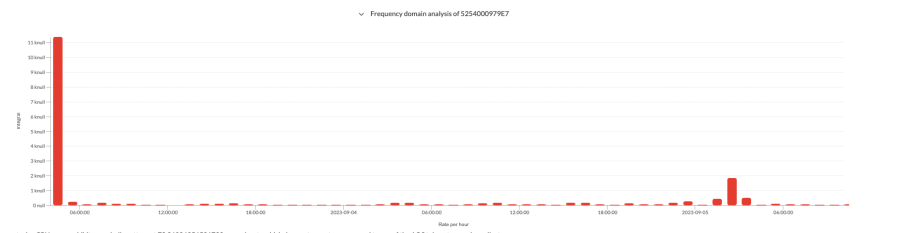
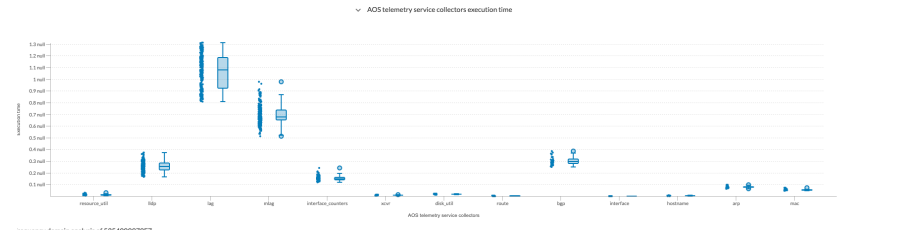
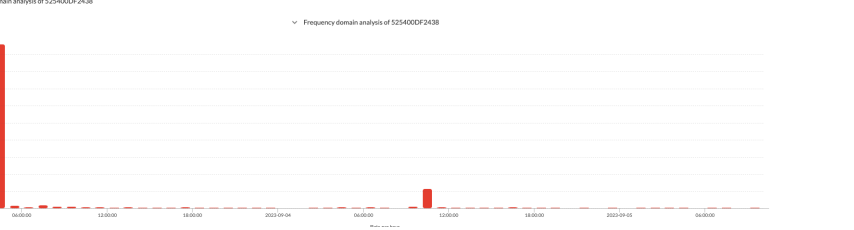
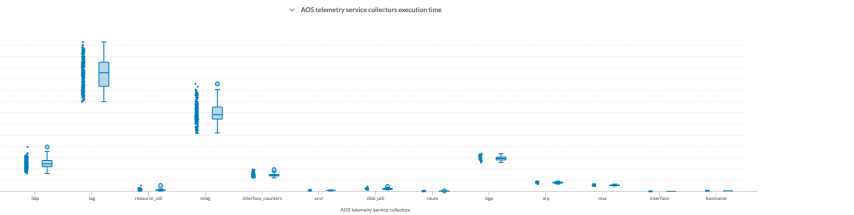
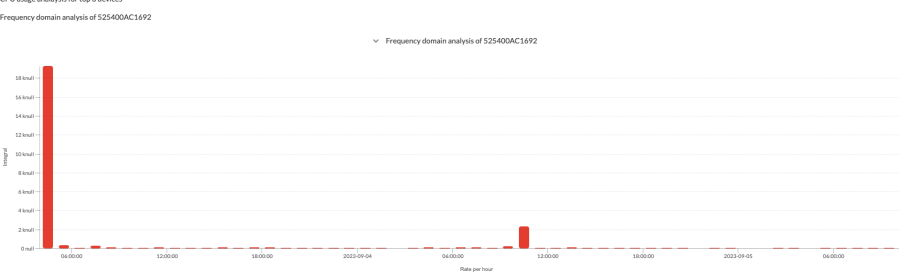
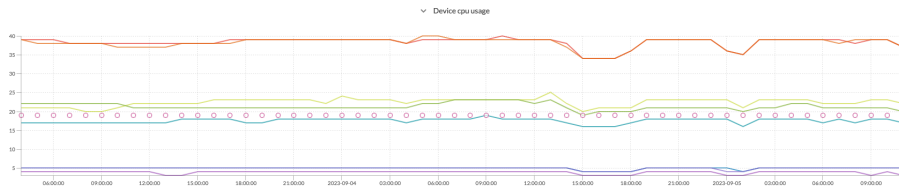
2 Memory usage analysis

All devices memory usage



3 CPU usage analysis

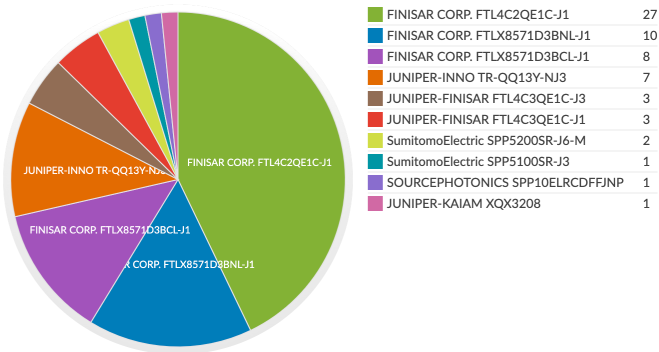
All devices cpu usage



Optical XCVR Report

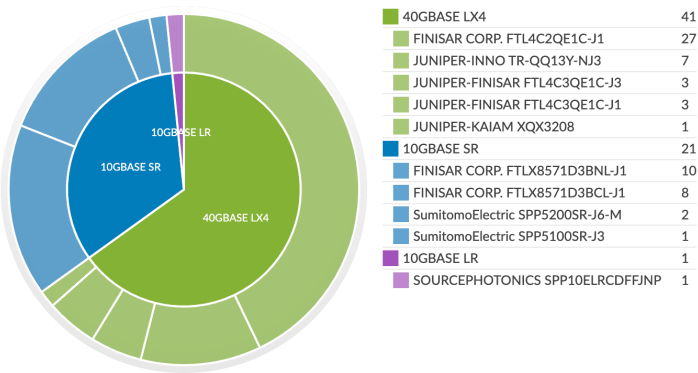
Vendor/Part Inventory Pie Chart

Transceivers by vendor/part



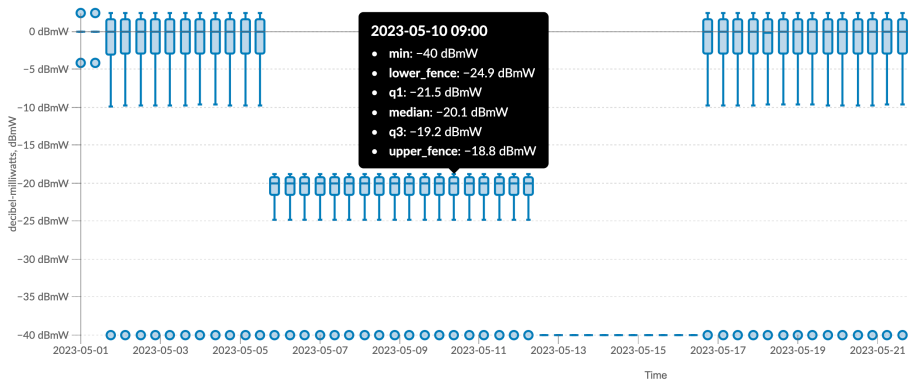
Vendor/Part Inventory by Media-type

Transceivers by media type

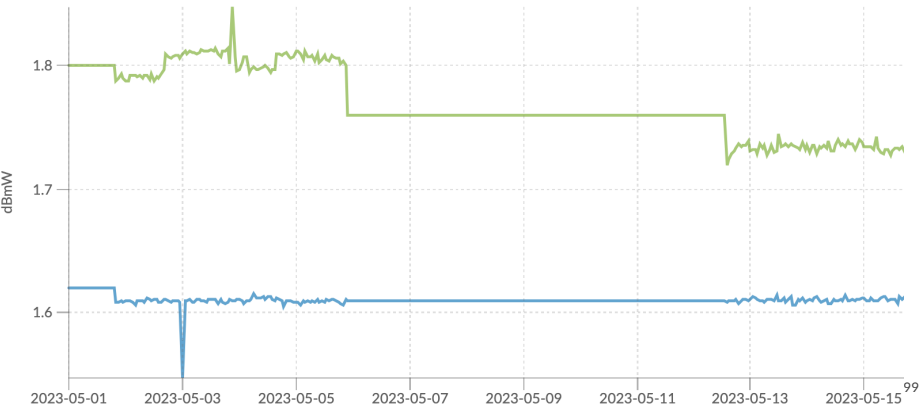


JUNIPER-INNO TR-QQ13Y-NJ3

count = 28, vendor thresholds: warn -12.01 - 5.50dBmW, alarm -16.00 - 6.50dBmW



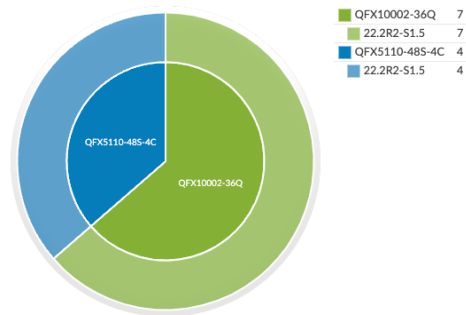
Outliers (2/28)



Traffic Report

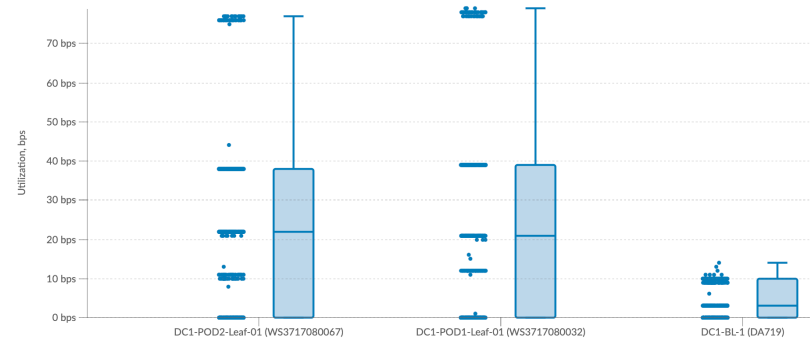
Device Hardware Model and Operation System Inventory Overview

Following chart summarizes the hardware models and corresponding operating systems for all devices deployed in the current data center



Leaf Peak RX Utilization IQR

Leaf Peak RX Utilization IQR IQR



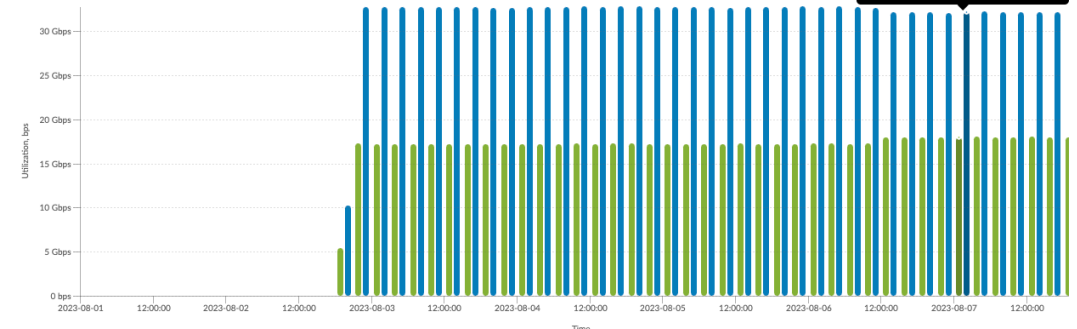
Aggregated Server Facing vs. Fabric Facing Traffic Rate

Interfaces on leaf switches are classified into two groups:

- Fabric facing interfaces which connect to spine switches
- Server facing interfaces which connect to everything else, e.g. hardware servers, generic systems, etc.

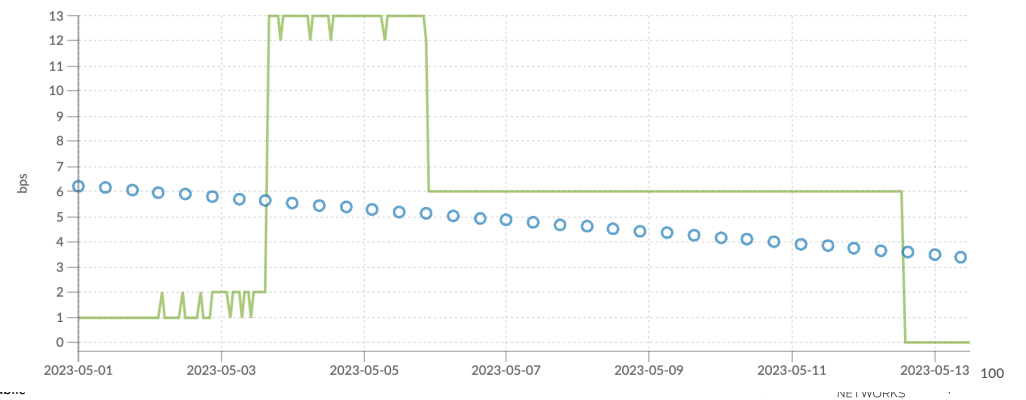
In the following chart,

- Server Facing Traffic** is calculated as the sum of both transmit (TX) and receive (RX) traffic rate from all server facing interfaces from all leaf switches, and
- Fabric Facing Traffic** is the sum of both TX and RX traffic rate of all spine facing interfaces from all leaf switches.



DC1-POD1-Spine-2 (DA738) fabric utilization and trend

Overall fabric traffic rate



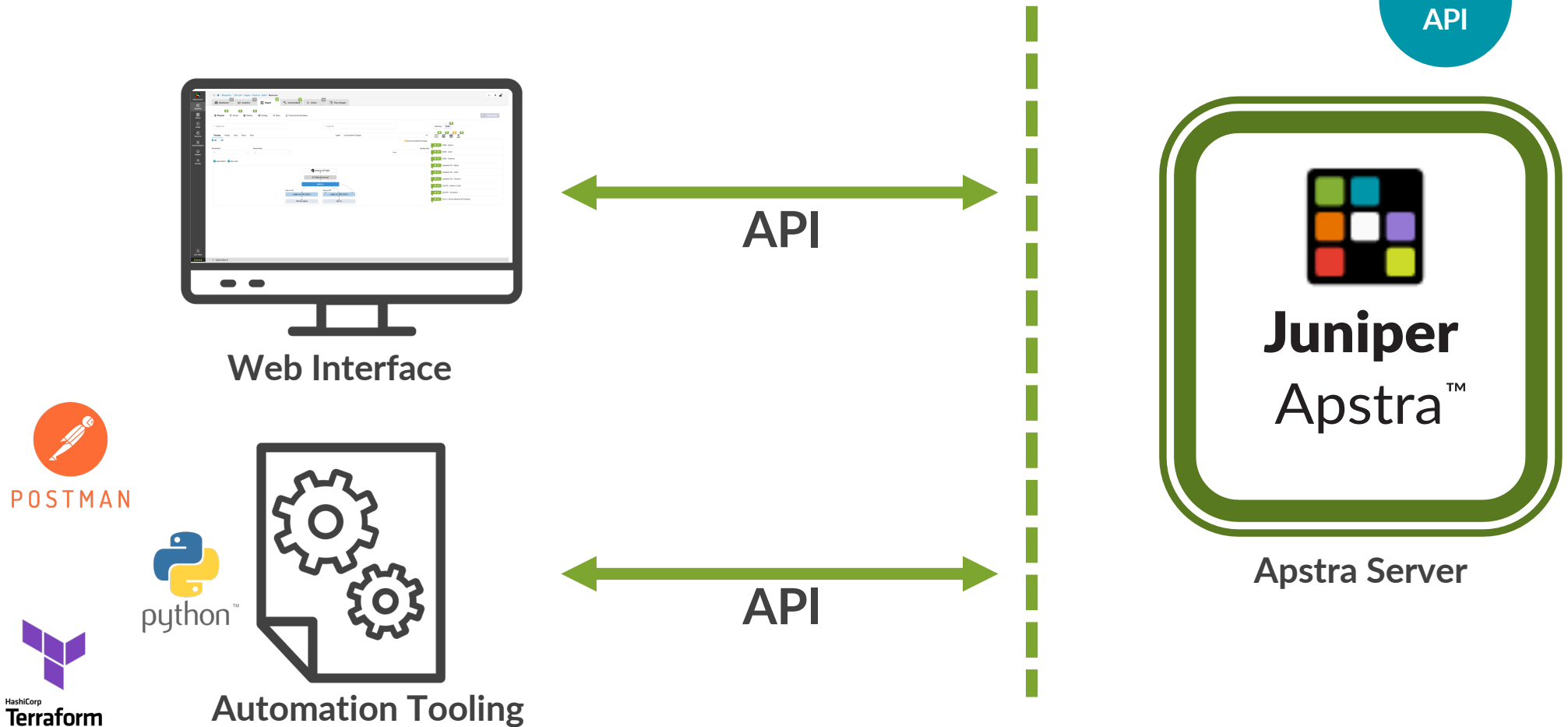
Apstra + Terraform



Juniper Apstra™

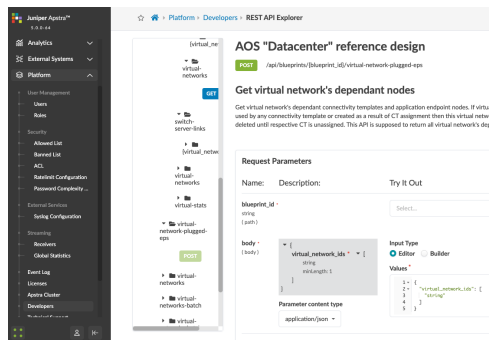
API at the Core : GUIのみでなくAPI活用も

RESTful
API



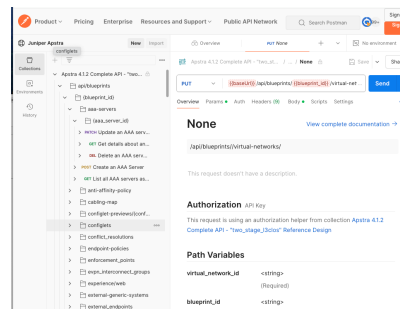
コアとなるAPI

Built-in API
Developer Tool



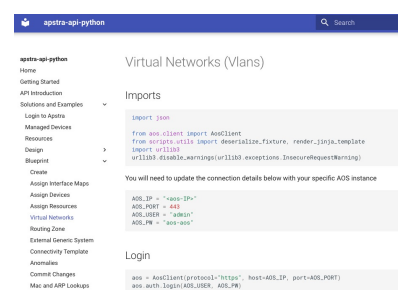
Rest API Explorer
&
Swagger UI

Postmon-API



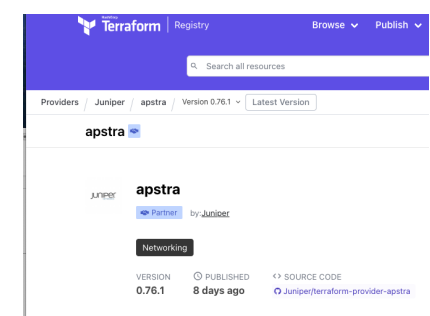
<https://www.postman.com/apstra/>

Python & SDK



<https://apstra-api-python.readthedocs.io/>

Terraform



<https://registry.terraform.io/providers/Juniper/apstra/>

AOS SDK 5.0.0
documentation



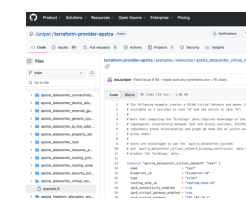
SDK Examples

This chapter contains different examples of usage of SDK client. It does not cover all possible usecases, but it strives to cover most cases met by customer on daily basis.

All examples are logically structured by 3 different days:

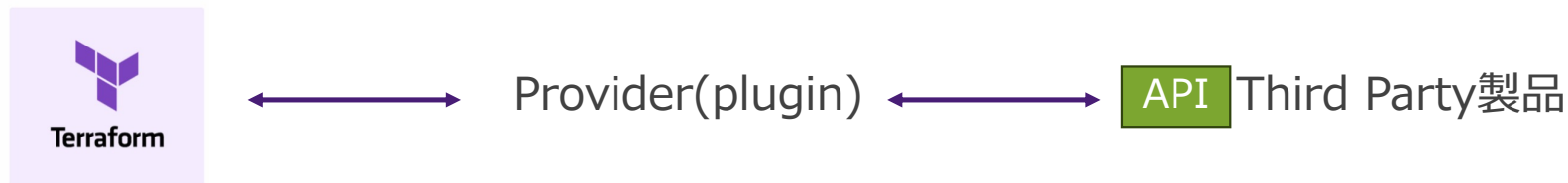
- Day 0
 - Day 1
 - Day 2
- All these examples assume the common setup. This could be achieved getting virtual environment made by following snippet (tested on Python 3.10x):

```
1. python -m venv sdk-venv
2. source sdk-venv/bin/activate
3. python -m pip install fabricate vish
4. python -m pip install -r path/to/aos/sdk-venv
```



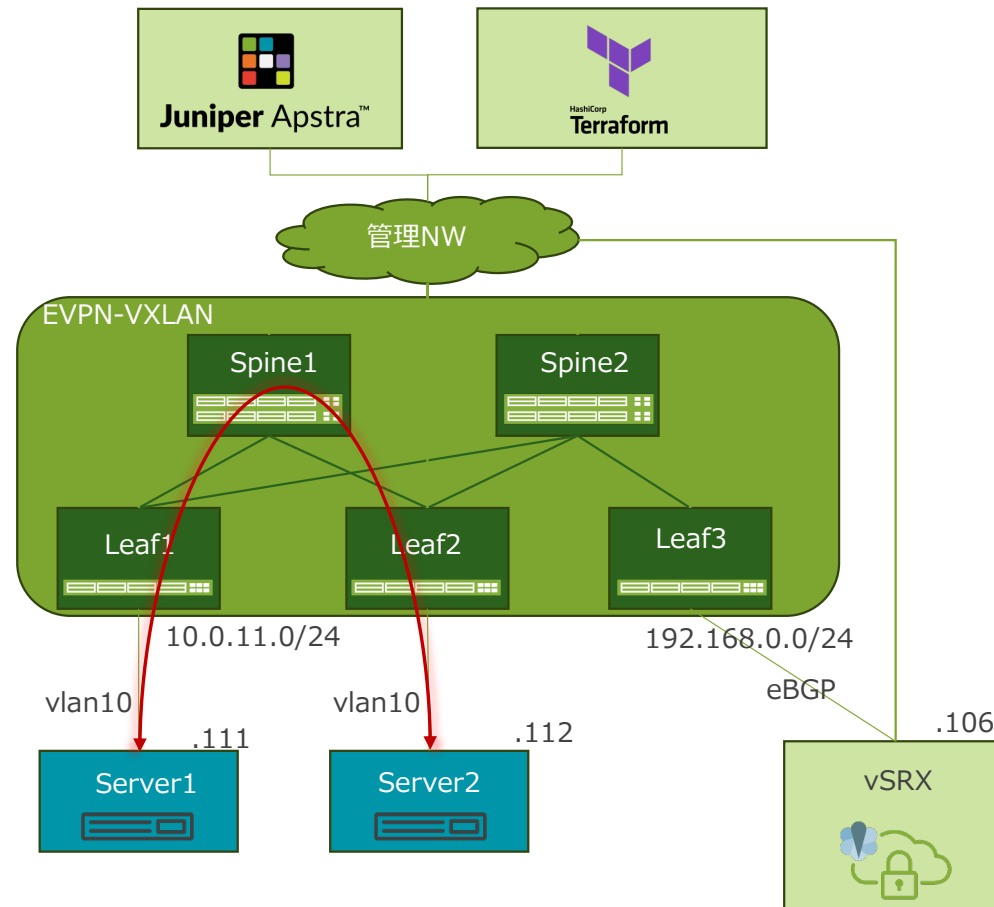
Terraformのメリット

- IaC(Infrastructure as Code)
 - インフラをコードとして管理できるためいつ何が変更されたか把握できる
- ドキュメント量が豊富
 - メジャーな製品でユーザも多く日本語資料やブログも多い
- 柔軟性と拡張性
 - Terraform Providerというプラグインの仕組みにより他製品と連携できる
- 使いやすさ
 - APIを使った自動化に比べると設定記述量が大幅に少なく済む



自動化できる内容

- Terraform + Apstra
 - IP Fabric(underlay)デザイン
 - コンフィグ生成とNW機器へ投入
 - Server間のVXLAN overlay構築
 - Leaf3のBGP設定
 - など



TerraformでApstraを自動化

- GUIで入力必要な情報がそのままコード化されるイメージ

racks.tf抜粋

```
resource "apstra_rack_type" "lab_guide_esi" {  
  name = "apstra-esi"  
  fabric_connectivity_design = "l3clos"  
  leaf_switches = {  
    apstra-esi = {  
      logical_device_id = data.apstra_logical_device.lab_guide_switch.id  
      spine_link_count = 1  
      spine_link_speed = "10G"  
      redundancy_protocol = "esi"  
    }  
  }  
  ...  
}
```

Edit Rack Type

Summary

Name *
apstra-esi

Description

Fabric Connectivity Design *
☒ L3 Clos
Use this option to design rack types used in 3-stage and 5-stage fabric template
☐ L3 Collapsed
Use this option to design rack types used in a collapsed template (spineless)

Configuration

Leafs Access Switches Generic Systems

Leaf

Name *
apstra-esi

Leaf Logical Device *
slice-7x10-1 ✕

Links per spine (7 available) *
1

Link speed *
10 Gbps ✕

Redundancy Protocol
☐ None ☐ MLAG ☒ ESI

参考URL

- Apstra Terraform連携情報
 - <https://registry.terraform.io/providers/Juniper/apstra/latest/docs>
- Apstra情報サイト（日本語）
 - <https://www.juniper.net/jp/ja/local/solution-technical-information/software.html>
- Github（terraformサンプル）
 - <https://github.com/Juniper/terraform-apstra-examples>

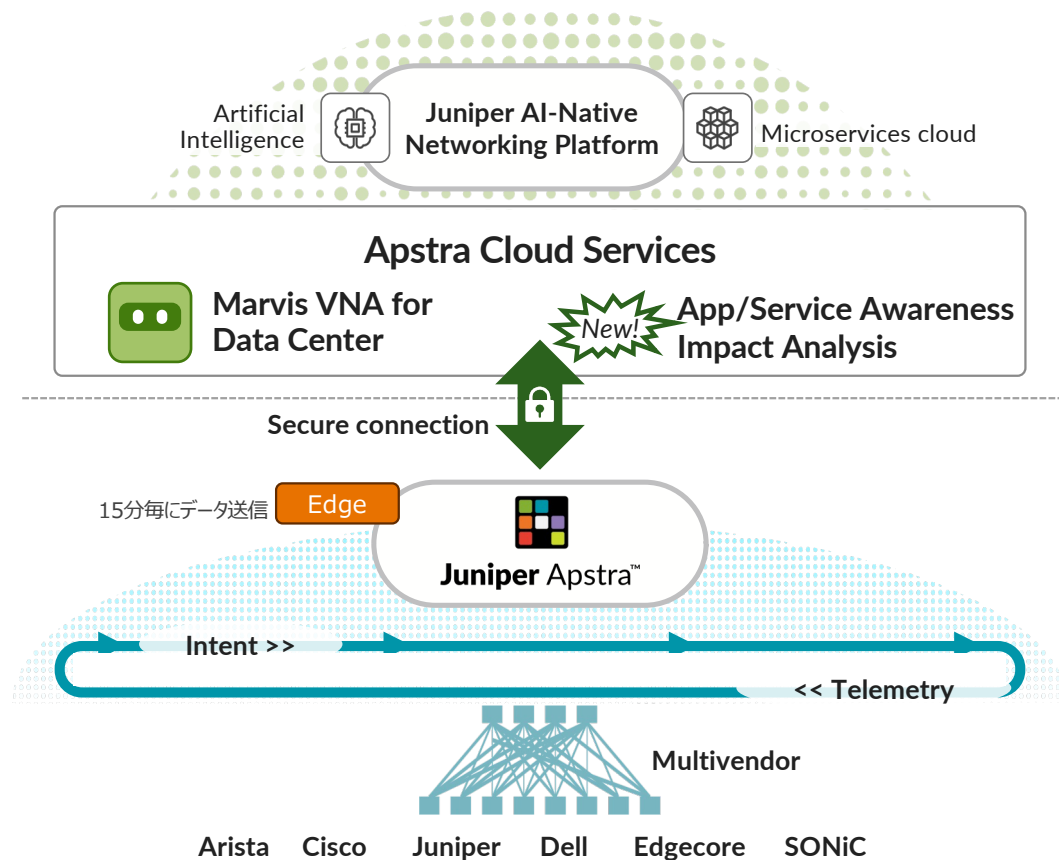
Apstra Cloud Services

Marvis VNA for DC



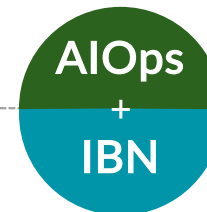
Juniper Apstra™

Data Center AIOps + IBN Architecture



Microservices Cloud environment

- 豊富なAI機能とマイクロサービス
- 10年以上のAIOpsの知見(MIST)

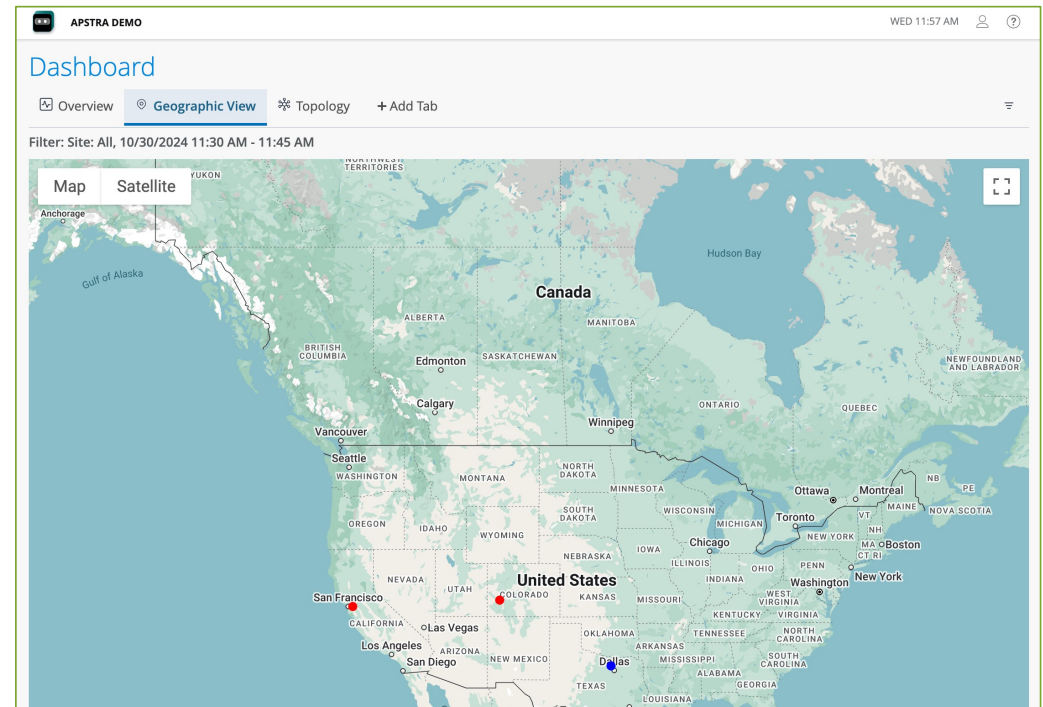
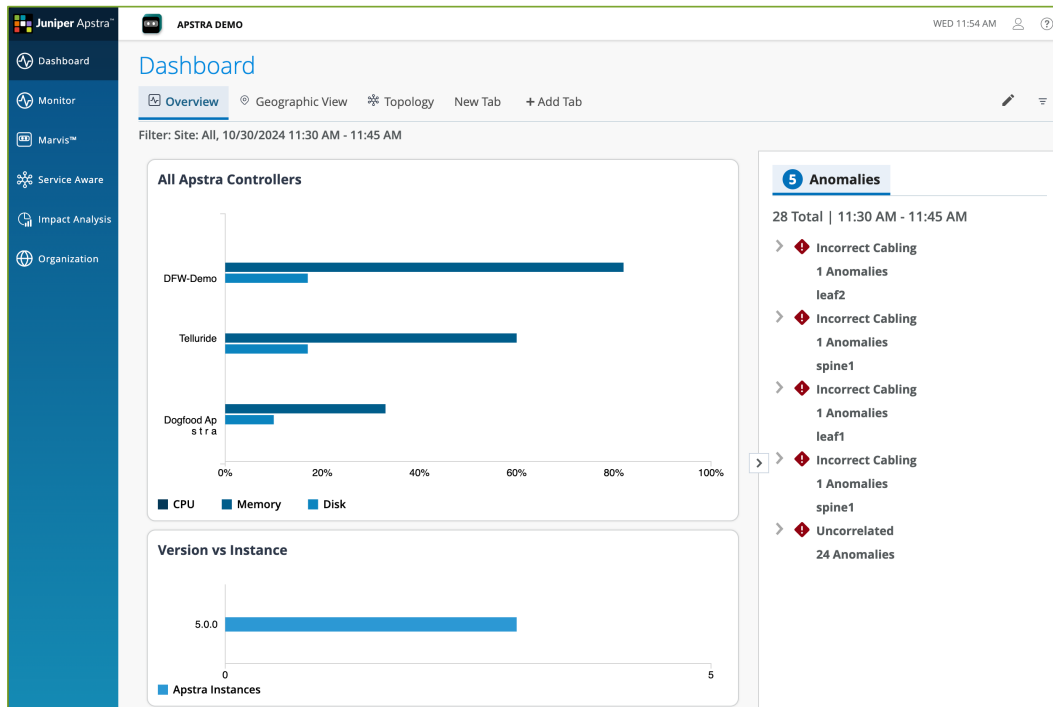


Apstra on-premises environment

- DC networking
- Intent-based design
- Graph DB, telemetry, flow data

Dashboards

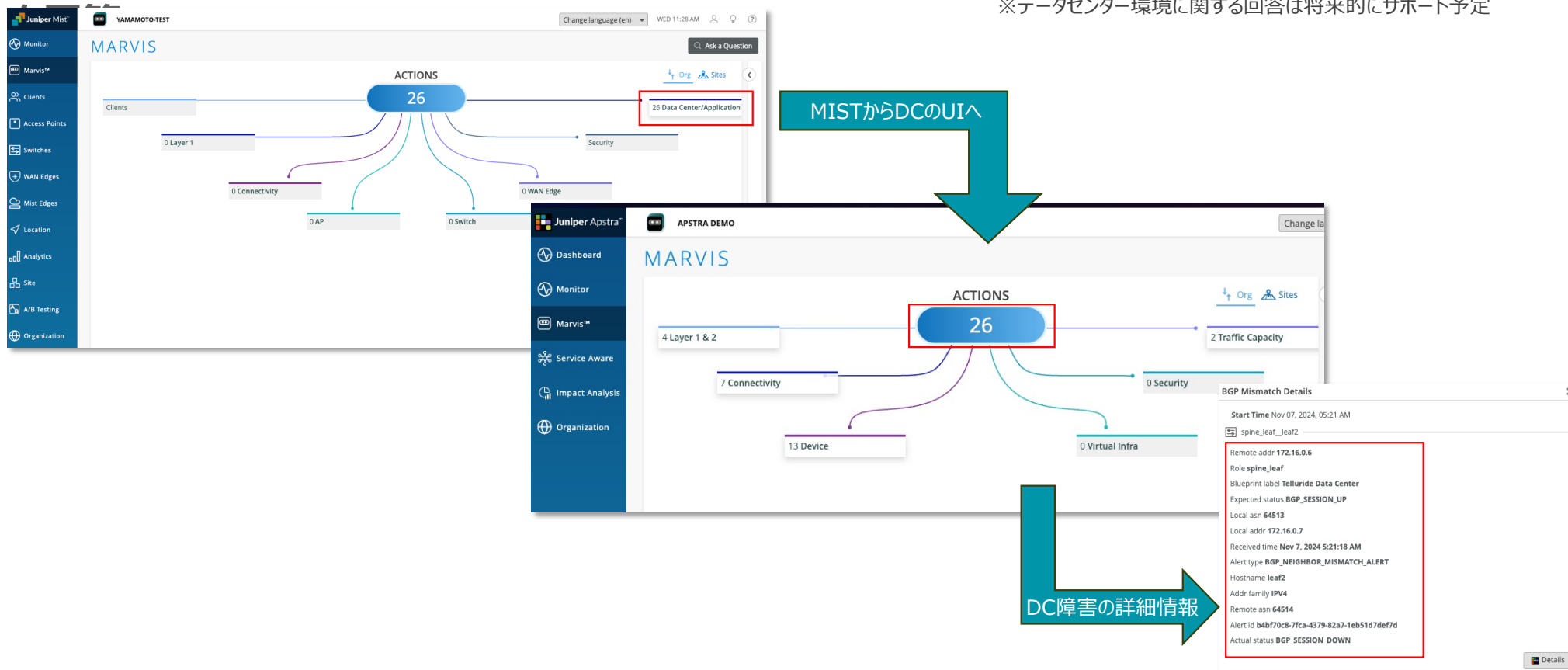
- 複数のApstraをまたいでインベントリやリソースを集中的に可視化
 - 以前にApstra Centralという名前で複数のApstraをまとめて表示する機能がACSでリリース
 - 複数サイトを管理するケースや大規模な場合にApstraコントローラを分ける必要があるときに集中管理が可能



Mist Integration紹介

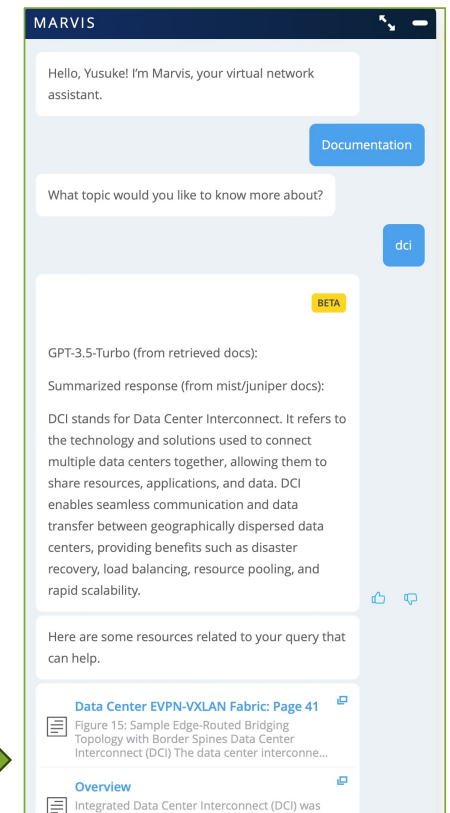
Mist UIからデータセンター障害の把握も可能に。Marvisからドキュメントや障害解決※について

※データセンター環境に関する回答は将来的にサポート予定



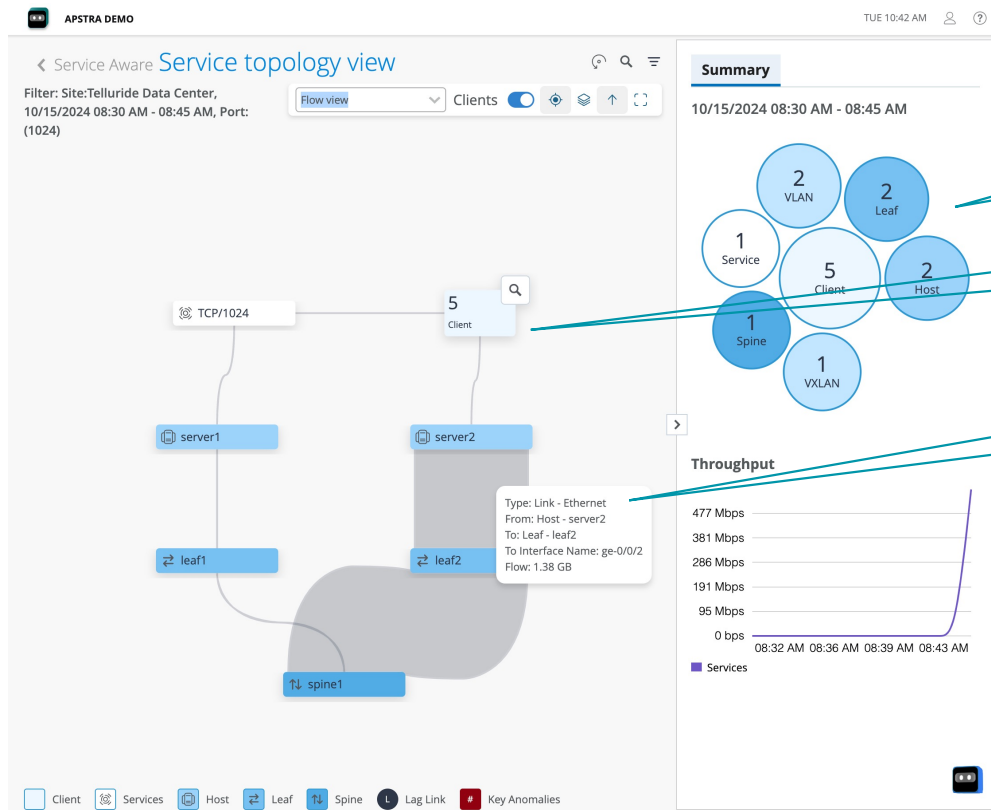
Marvis chatbot for docs

- Marvisからチャット形式でドキュメント検索が可能。ドキュメント以外は将来的に対応予定



Service Awareness

- ネットワークトポロジとアプリケーションの接続を可視化
 - サーバ通信のパスやトラフィック量を見たいという声が多かった。Apstra Flowは情報のソートやフィルターまで



ネットワークに含まれている情報のサマリー表示

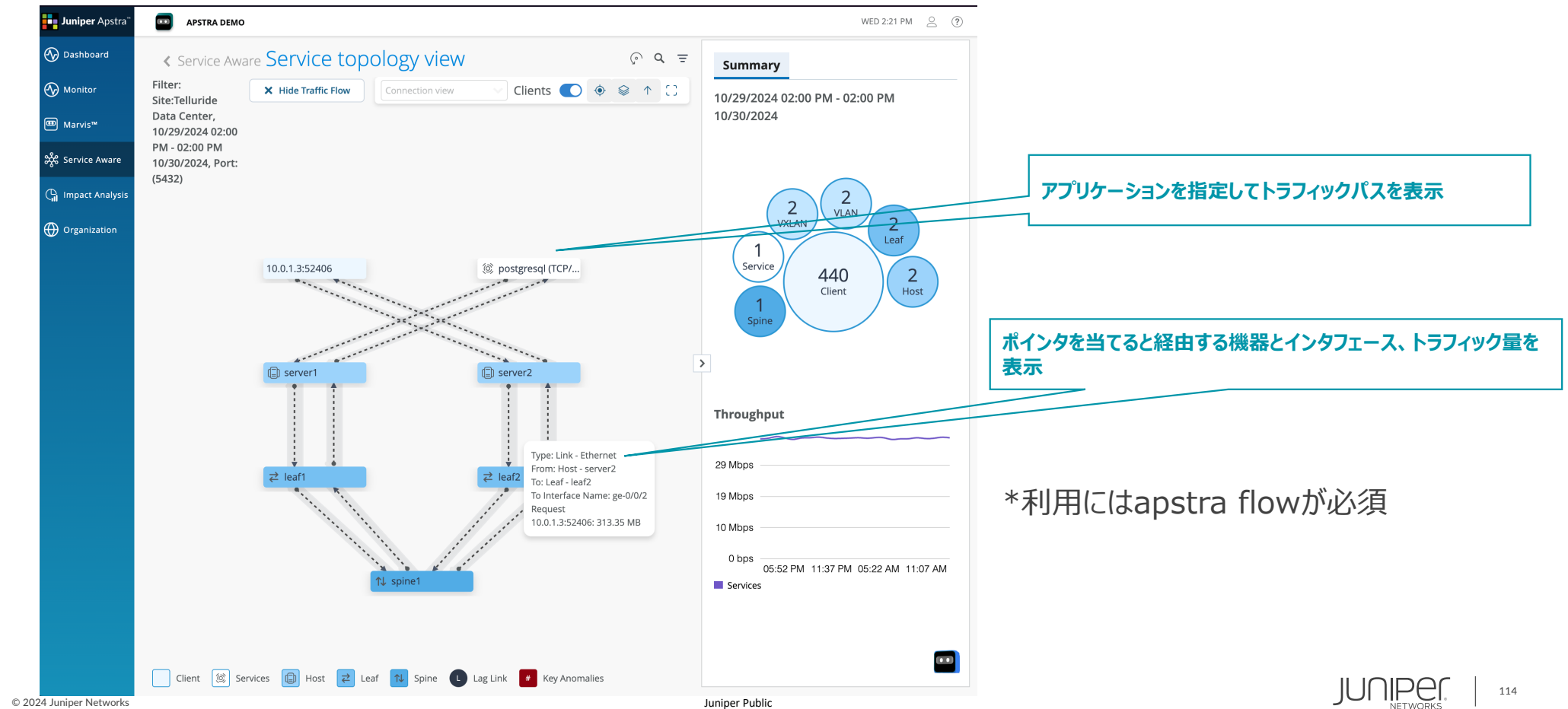
クライアントとサービスを含んだトポロジー表示

Flow情報から通信のパスとスループットを可視化

*利用にはapstra flowが必須

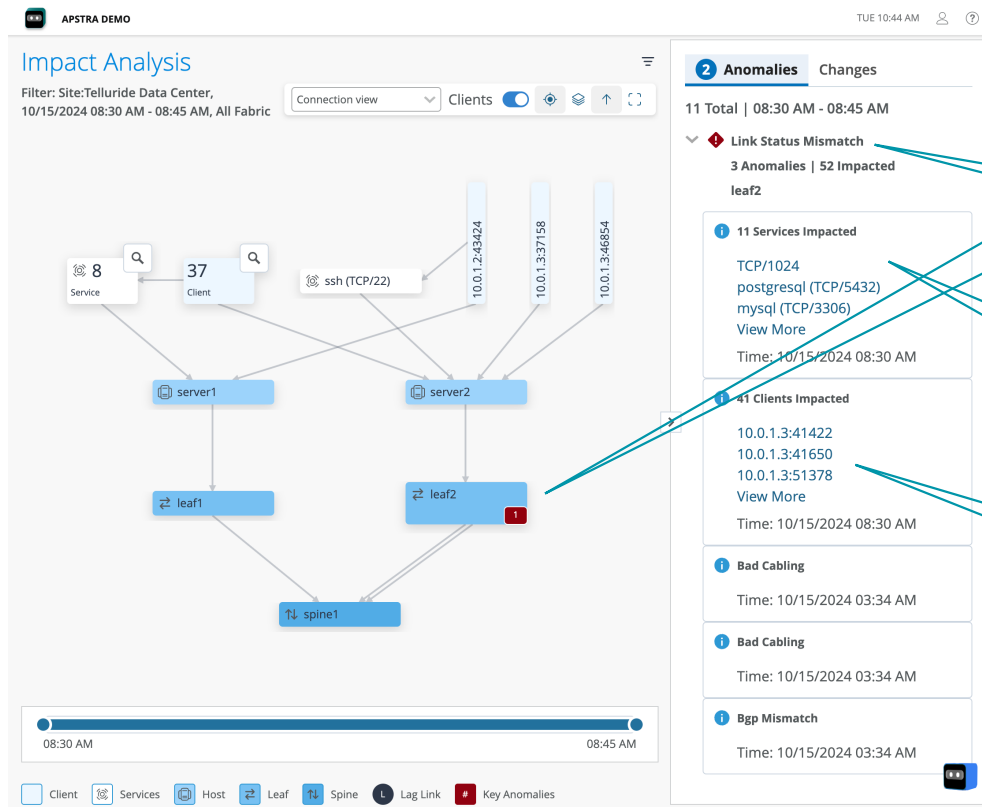
Service Awareness

- アプリケーションが経由しているパスを可視化



Impact Analysis

- 例外エラーとエラーによりインパクトのあるサービスやクライアントを関連して可視化
 - ネットワークの問題かサーバの問題か切り分けに有効



例外エラー発生デバイス

Link Statusミスマッチが発生している(Root Cause)

インパクトがあるサービス一覧

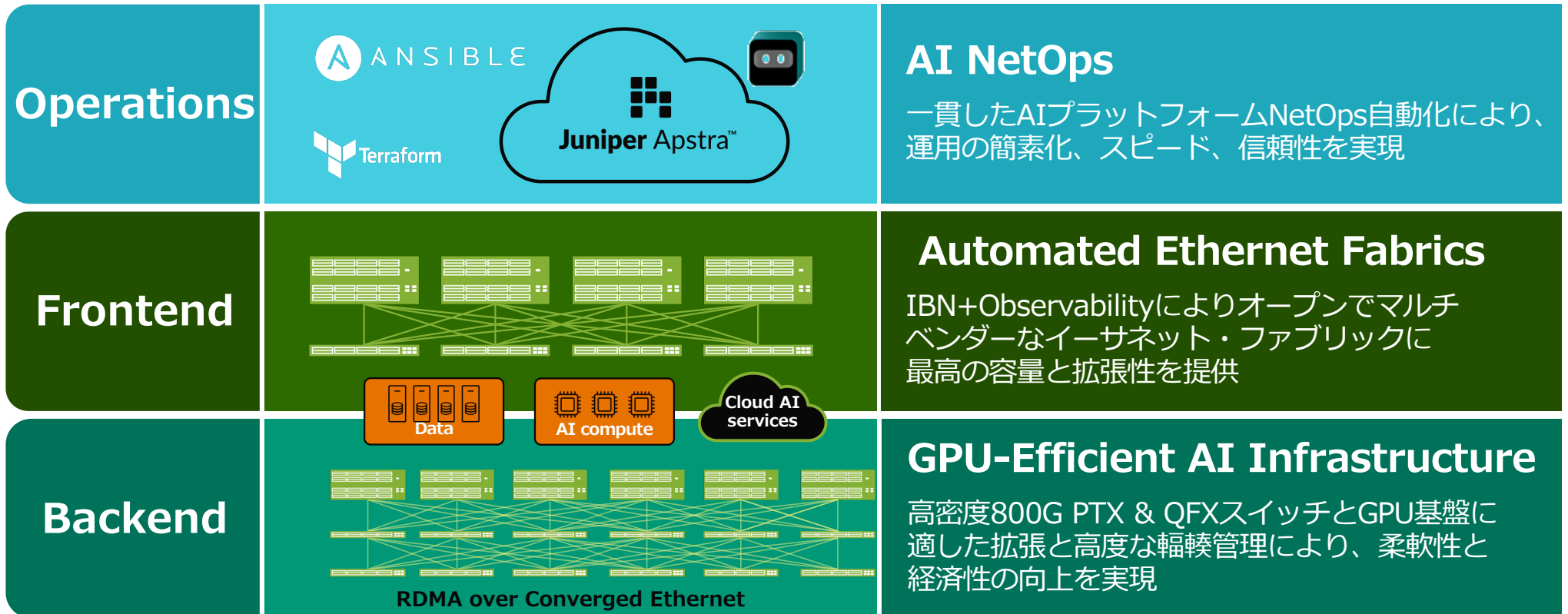
インパクトがあるクライアント一覧

AI Cluster Template



Juniper Apstra™

Juniper's AI DC Networking Solution



Apstra for AI DC Networking Solution

Intent-based Networking for AI DC により、
ベストプラクティスのAI DCネットワークを短期間で利用でき、E2Eかつ詳細な分析と最適化

AI DCネットワークの必要な解決方法

1. 高帯域HW & Rail Optimized Design

2. DCQCN(ECN/PFC) + DLB/GLB

3. 最適化によりEthernetもInfinibandに匹敵

4. Queue/DCQCN/GPU可視化

Apstraにより実現



1. Ops4AI Lab - JVD リファレンスデザイン
Rail Optimized Topology -400G/800G



2. デザインに必要な設定の自動生成

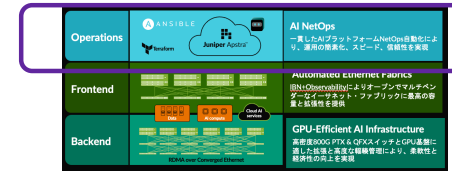


3. 状態把握とAuto Tune App



4. 詳細かつ自動化された可視化

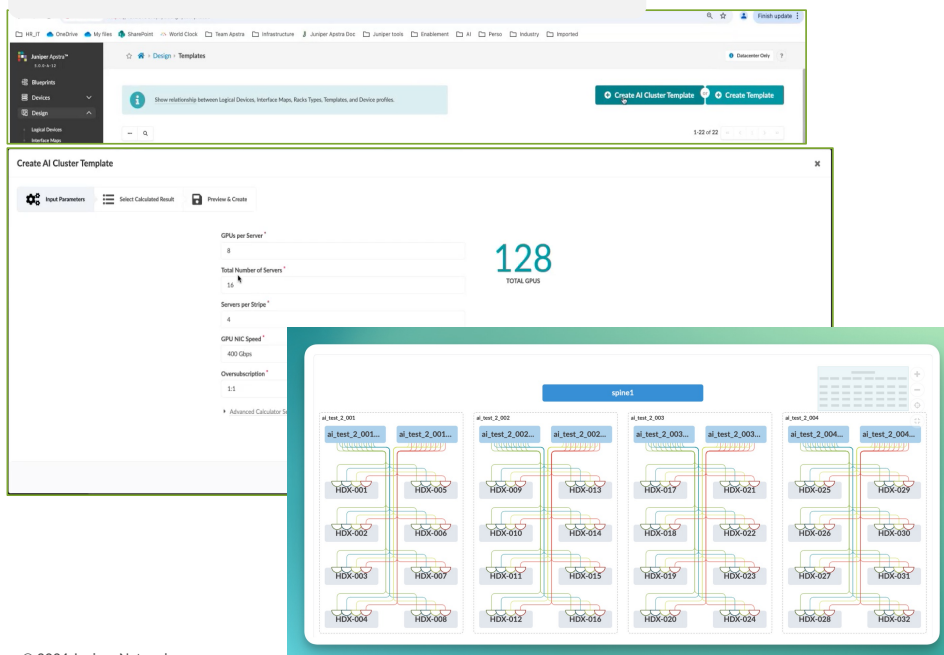
Apstra for AI Clusters



ベストプラクティスの保証された設定と可視化の自動化

1. Intent Based Template– 要件指定だけでRail-Optimizationの保証された設定を自動生成
2. Specialized analytics validations – AI DCに特化した状態把握と分析
3. GPU Agent Collector – コンピュトエージェントによりNIC含めたE2Eの可視化

Simplified Template designer with minimal inputs



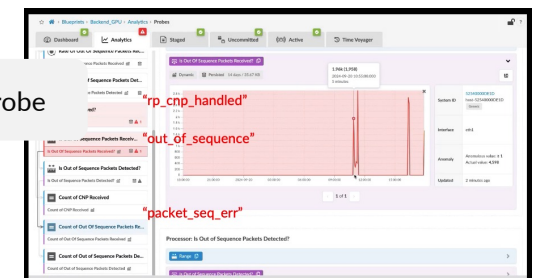
Queue-Level traffic management visibility

Queue Name	Queue ID	Queue Type	Queue Size	Queue Status	Queue Priority	Queue Color
Queue 1	1	1	1	1	1	1
Queue 2	2	2	2	2	2	2
Queue 3	3	3	3	3	3	3
Queue 4	4	4	4	4	4	4
Queue 5	5	5	5	5	5	5
Queue 6	6	6	6	6	6	6
Queue 7	7	7	7	7	7	7
Queue 8	8	8	8	8	8	8
Queue 9	9	9	9	9	9	9
Queue 10	10	10	10	10	10	10
Queue 11	11	11	11	11	11	11
Queue 12	12	12	12	12	12	12
Queue 13	13	13	13	13	13	13
Queue 14	14	14	14	14	14	14
Queue 15	15	15	15	15	15	15
Queue 16	16	16	16	16	16	16
Queue 17	17	17	17	17	17	17
Queue 18	18	18	18	18	18	18
Queue 19	19	19	19	19	19	19
Queue 20	20	20	20	20	20	20
Queue 21	21	21	21	21	21	21
Queue 22	22	22	22	22	22	22
Queue 23	23	23	23	23	23	23
Queue 24	24	24	24	24	24	24
Queue 25	25	25	25	25	25	25
Queue 26	26	26	26	26	26	26
Queue 27	27	27	27	27	27	27
Queue 28	28	28	28	28	28	28
Queue 29	29	29	29	29	29	29
Queue 30	30	30	30	30	30	30
Queue 31	31	31	31	31	31	31
Queue 32	32	32	32	32	32	32

Dashboards Stripe & Rail Traffic



GPU Hardware Traffic Monitoring probe



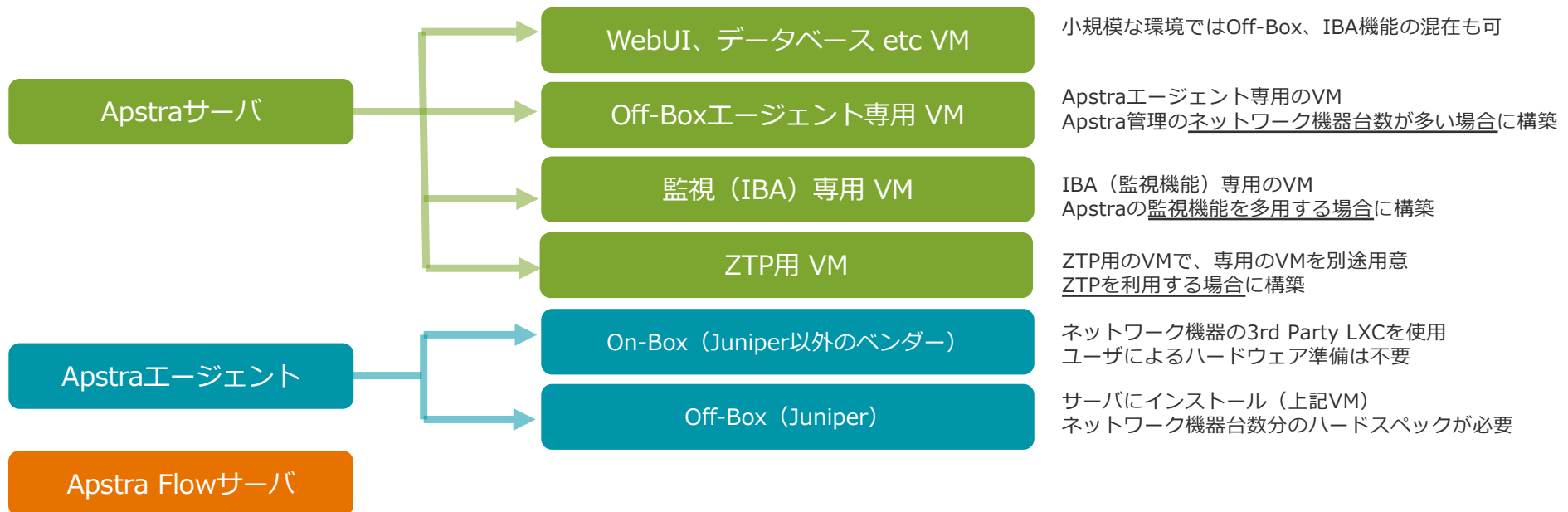
コンポーネント



Juniper Apstra™

Juniper Apstraのコンポーネント

Apstraはサーバとエージェントの2種でサーバはVM、エージェントはコンテナで提供



- VMはユーザが準備するハイパーバイザにインストール
- VM冗長はハイパーバイザ側で実施 (例: vSphere HA)
- サポートするハイパーバイザはこちらを参照ください

<https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-install-upgrade/topics/ref/apstra-server-platforms.html>

Apstraサーバ必須VMのハードウェアスペックはこちらを参照ください

実際のハードウェアスペックは管理するデバイス数などにより変更が必要です。サイジングも可能です。

<https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-install-upgrade/topics/ref/apstra-server-resources.html>

Apstra の構成要素

Apstra Cloud Services

- 複数のApstraを管理
- AIopsやService AwarenessのGUI

Artificial Intelligence



JUNIPER
driven by Mist AI



Microservices cloud

Apstra Cloud Services

Apstra Agents

- Apstra Server上のコンテナ
- Apstra ServerはSpine/LeafをAgents経由で管理(分散処理)
- 管理台数によっては専用VM上で稼働

Apstra Agents
(コンテナ)



Apstra
Server VM

Apstra
Flow VM

Apstra Flow

- Flow用仮想マシン
- Flow GUI/APIを提供

Apstra Server

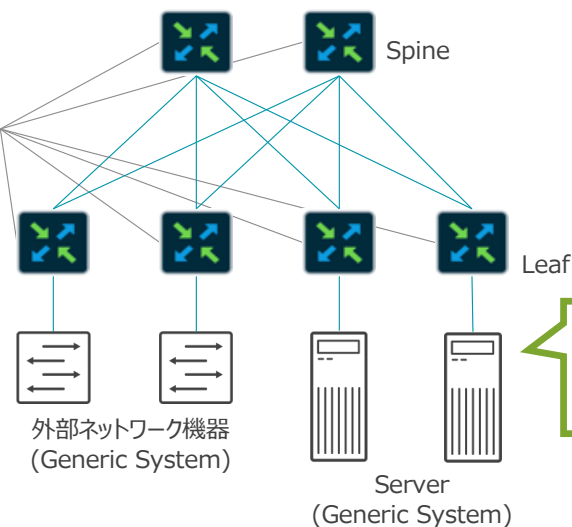
- 仮想マシン
- Apstra GUI/APIを提供

ZTP
Server VM

ZTP Server

- 仮想マシン
- Spine/Leafのゼロタッチプロビジョニング (ZTP)を提供

管理SW



Generic System

- 外部ネットワーク機器やServerをGeneric Systemと呼ぶ

Juniper Apstraの構築

Apstraインストール

VMイメージを起動するだけ



Juniper Apstra™

ESXi / KVM



Welcome to AOS!

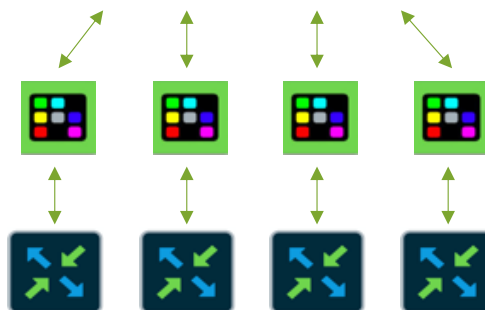


Apstra Agentのインストール

NOS*の設定・監視はAgent経由
分散処理でスケール



Juniper Apstra™



Apstra ZTP

NOSの初期環境構築はZTP
マルチベンダーにも対応



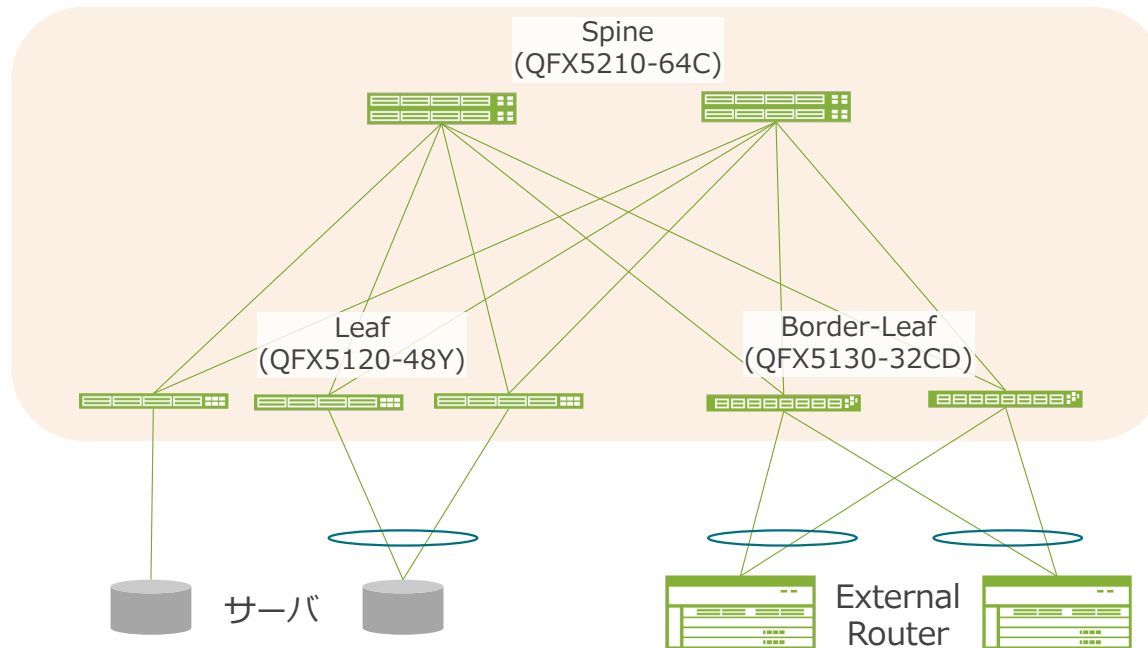
ESXi / KVM / Hyper-V



- DHCP管理IPアドレス
- NOSバージョン変更
- Apstra Agentインストール
- 初期コンフィグ

サンプル構成

- Leaf & Spine構成と推奨ネットワーク機器
- L3処理が必要な場合はLeafにてIRBを利用(Edge Routing Bridge)
- Border Leafからの外部接続



Juniper Apstra™

※バーチャルアプライアンス利用
サーバー、仮想化基盤は別途用意が必要

※Apstraとネットワーク機器の
管理Interface経由で接続
管理ネットワークは別途用意が必要

Juniperネットワーク機器サポートHW

他ベンダ様対応機器や、サポートversionなど最新情報はこちらを参照ください

<https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/topic-map/devices-qualified.html>

Device Operating System	Device Role	Supported Device / Apstra Release 4.2.0
Juniper Junos OS	Access switch	EX4400-24MP, EX4400-48MP, EX4400-24T, EX4400-48T, EX4400-48F, EX4650-48Y
	Spine or Superspine	QFX5200, QFX5210
	Leaf or Spine or Superspine	•QFX5100*, QFX5110**, QFX5120, QFX10002, QFX10008, QFX10016, EX4400-24MP, EX4400-48MP, EX4400-24T, EX4400-48T, EX4400-48F, EX4650-48Y 注意事項：EX4400シリーズはキャンパス向けにデザインされているためスケールの制限があります
Juniper Junos OS Evolved	Spine or Superspine	QFX5220
	Leaf or Spine or Superspine	QFX5130, QFX5700 chassis and 3 line cards, PTX10001-36MR, PTX10004 chassis (7RU, 4-slots), PTX10008 chassis (13RU, 8-slots), PTX10016 chassis (21RU, 16-slots), ACX7100-32C, ACX7100-48L, ACX7024

*Don't use as leaf with Layer 3 VNI.

** Can't be used as a border leaf. It can't route between VXLAN IRB and L3 interface.



Agenda

- 製品コンセプト
- JUNIPER APSTRAのご紹介
 - 製品概要
 - デザインパターン(DC Reference, Freeform)
 - 機能紹介(IBN, ACS, AI cluster etc.)
 - コンポーネント
- オペレーションサンプル
- 購買情報
- お役立ち情報

Juniper Apstraのご紹介

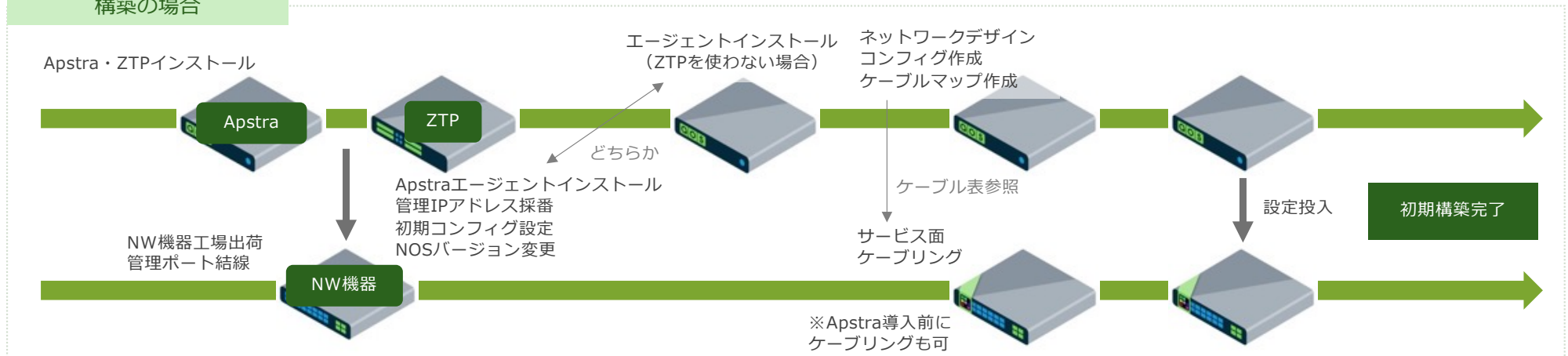
- オペレーションサンプル



Juniper Apstra™

構築・運用の全体概要

構築の場合

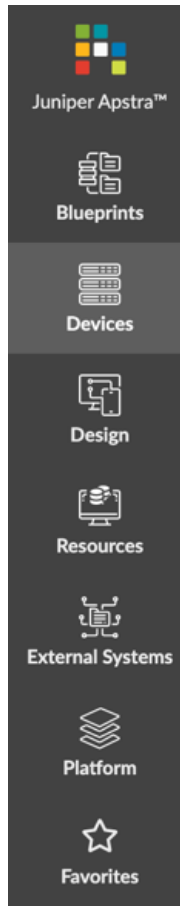


運用の場合



※詳細な内容はマニュアルもご参照ください

Apstra GUIメニュー



テンプレートを用いて実際にデプロイをするタブで、実設定(仮想ネットワーク等)の画面

物理デバイス関連(エージェント、OSイメージ、ZTP etc)の管理画面

論理的なデザインの設定関連(テンプレート、ポート構成、論理デバイス etc)の管理画面

利用するパラメータについて使用する範囲を事前に定義

ファブリックが外のネットワークと通信するための外接ルータの定義

Apstraのプラットフォーム関連の設定

ネットワーク設計: ロジカルデバイス

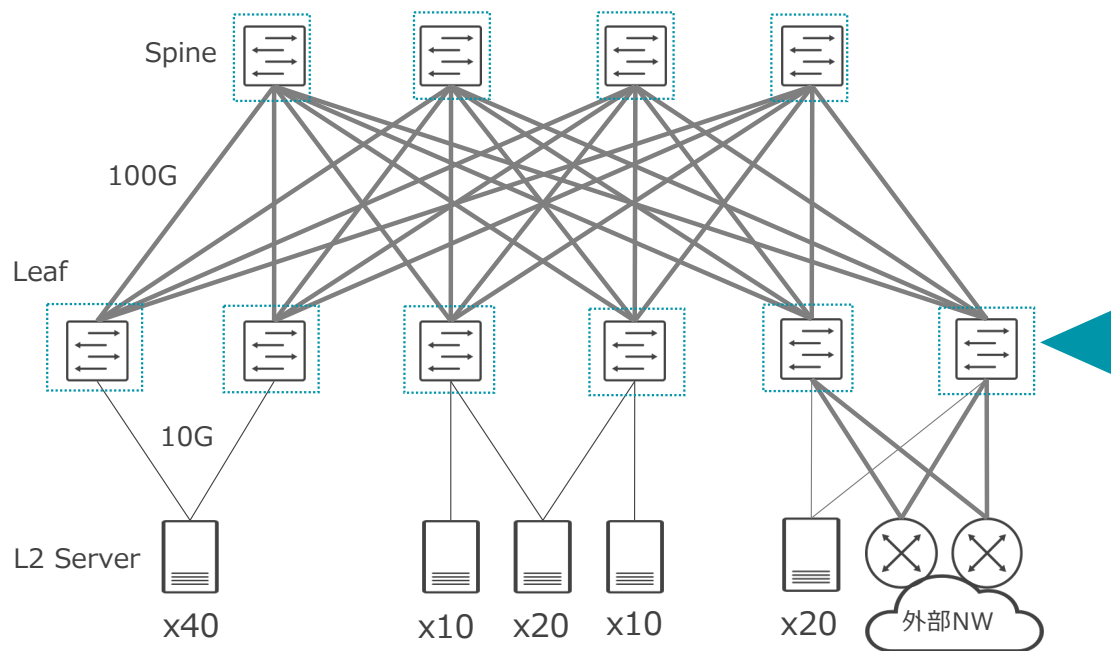
ポート数と役割を定義

ラックタイプ作成

NW構成を作成

パラメータプール

ハードウェア選定



例: Leafのロジカルデバイスを定義

PANEL #1

TOTAL: 48 ports (48 assigned • 0 available)

PORT GROUPS: 48 x 10 Gbps L2 Server

10Gbps x 48Portはサーバ接続用

1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48

PANEL #2

TOTAL: 4 ports (4 assigned • 0 available)

PORT GROUPS: 4 x 100 Gbps Spine

100Gbps x 4PortはSPINE接続用

+ Add Panel

Create Another? Create

ネットワーク設計: ラックタイプ

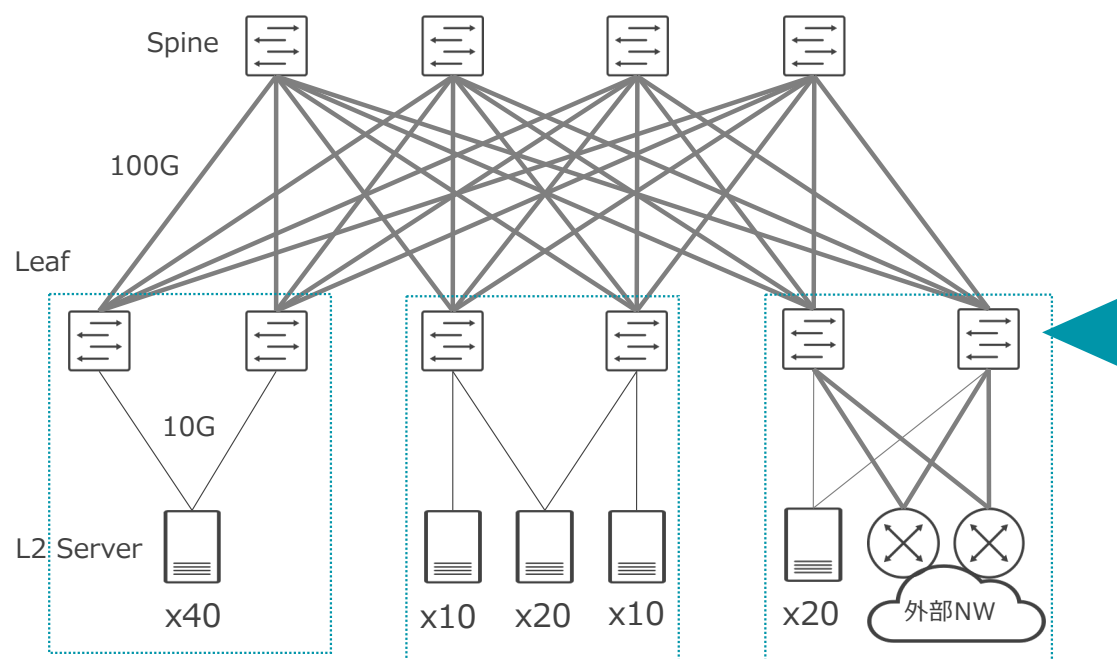
ポート数と役割を定義

ラックタイプ作成

NW構成を作成

パラメータプール

ハードウェア選定



例 : Leaf 2台, Server 40台のラックを定義



ネットワーク設計: テンプレート

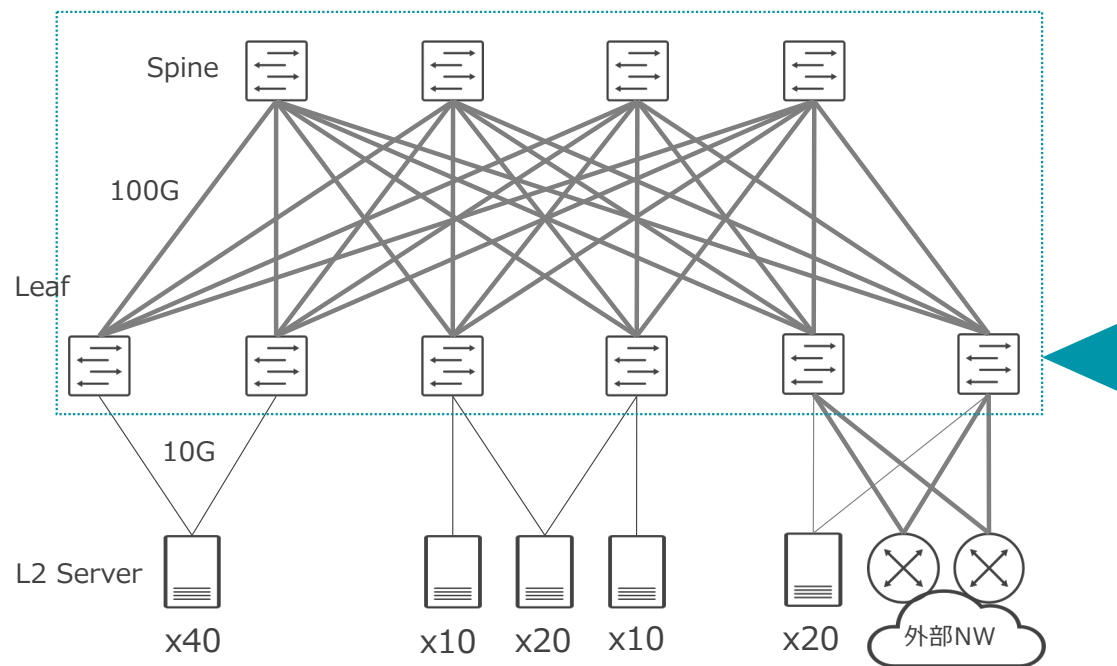
ポート数と役割を定義

ラックタイプ作成

NW構成を作成

パラメータプール

ハードウェア選定



例：15ラックのLeaf, Server構成をSpine 4台の 3 stage closで定義



ネットワーク設計: ネットワークリソース

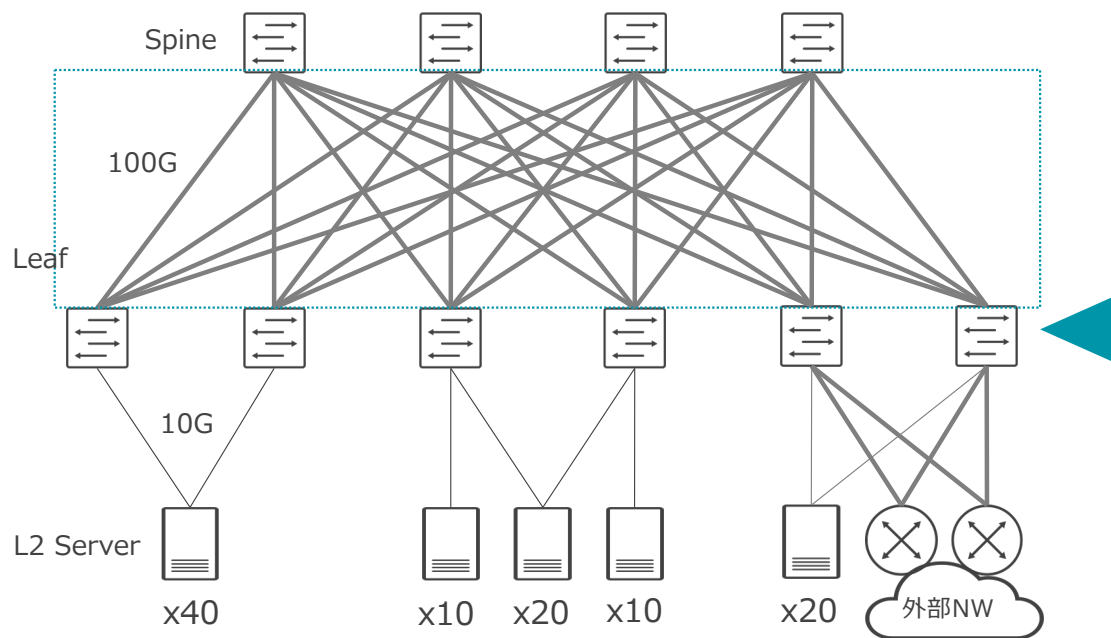
ポート数と役割を定義

ラックタイプ作成

NW構成を作成

パラメータプール

ハードウェア選定



例 : IP Address Pool

Pool Name	Total Usage	Per Subnet Usage
External Device Link	0%	0% 10.0.5.0/24
Loopback	0%	0% 10.0.0.0/24
MLAG Link	0%	0% 10.0.4.0/24
Spine-Leaf Link	0%	0% 10.0.2.0/23
VXLAN VTEP	0%	0% 10.0.1.0/24

例 : AS Num Pool

Pool Name	Total Usage	Range Usage
Private-64512-65534	1.66%	1.66% 64512 - 65534
Private-4200000000-4294967294	0%	0% 4200000000 - 4294967294

例 : VNI Pool

Pool Name	Total Usage	Range Usage
Default-10000-20000	0.02%	0.02% 10000 - 20000

ネットワーク設計: デバイスプロフィール

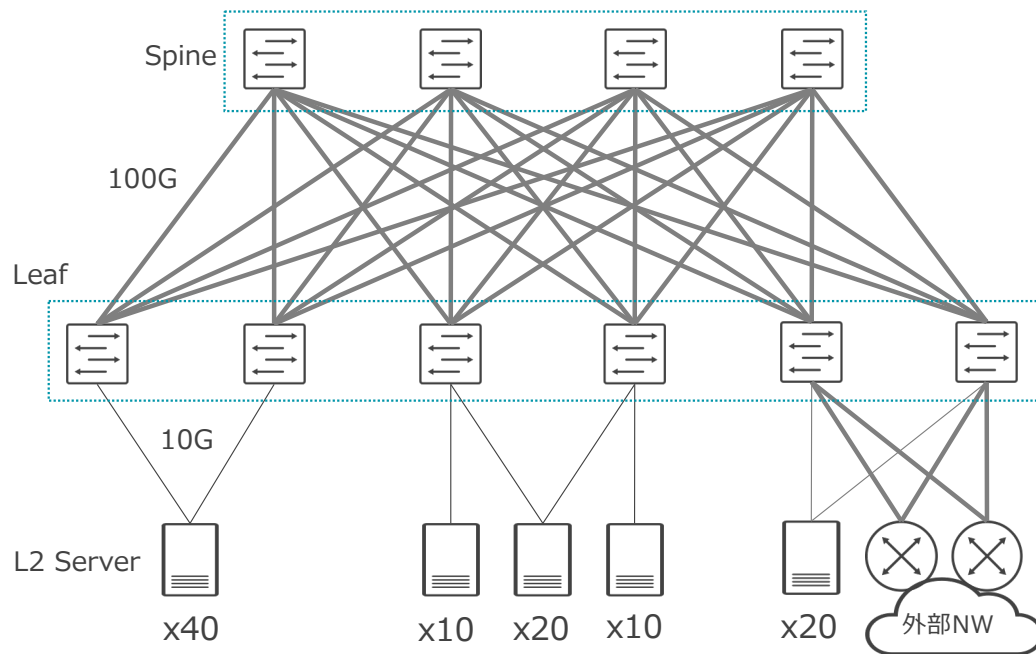
ポート数と役割を定義

ラックタイプ作成

NW構成を作成

パラメータプール

ハードウェア選定



Logical Device
(ベンダに依存しないポート構成)

Device Profile
(実際のハードポート構成)

Interface Map
(IM)

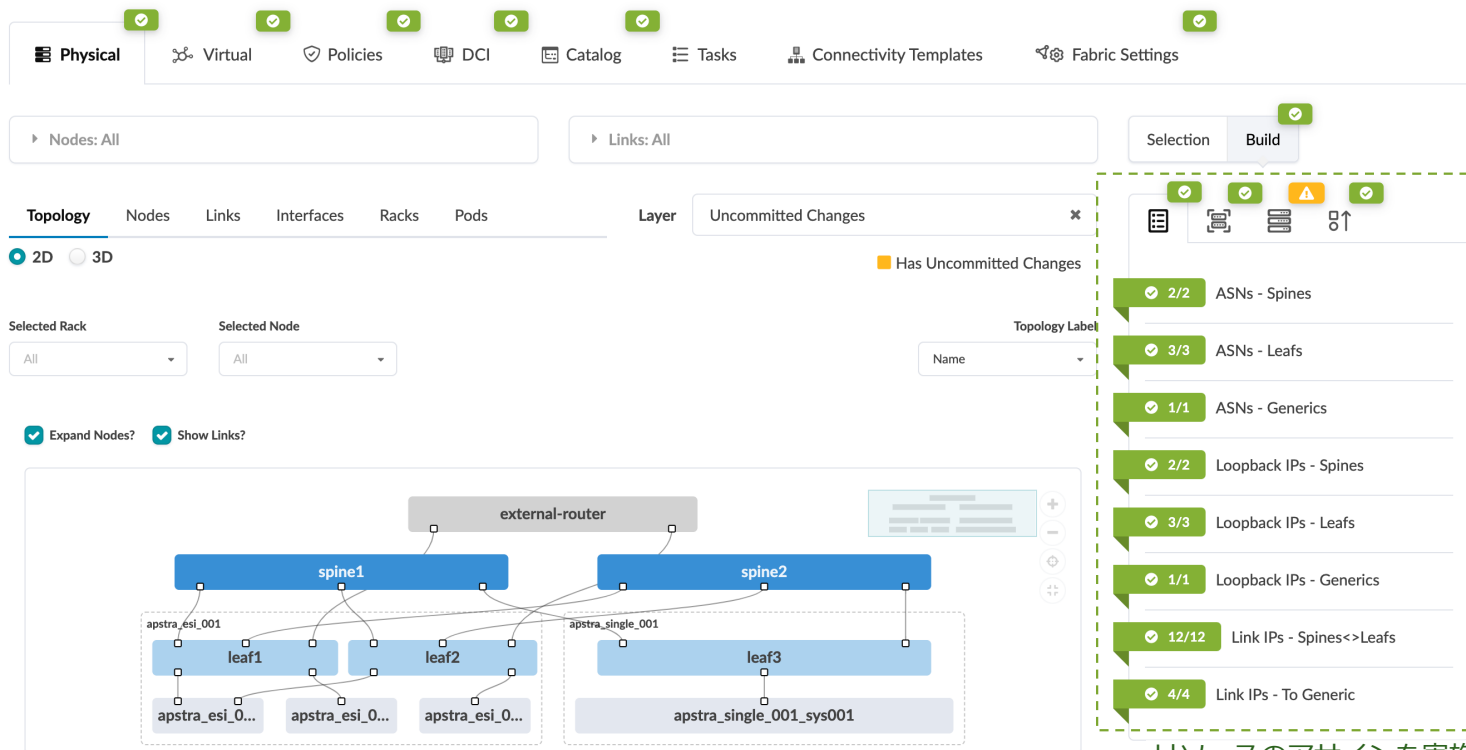
Blueprint

例: Juniper QFX5120-48Y デバイスプロフィール

Panel #1	
INTERFACES CAPACITY	
80 x 25 Gbps	80 x 10 Gbps
48 x 1 Gbps	8 x 40 Gbps
8 x 100 Gbps	
PORTS Click on port to toggle the details	
1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50
51	52
53	54
55	56

ネットワーク構築: ブループリント

事前定義したロジカルデバイス、ラックタイプ、テンプレート、ネットワークリソースをもとに
実ネットワーク構成（ブループリント）を設計し、ネットワークが完成



リソースのアサインを実施し、
Configurationを作成

仮想ネットワーク設定：概要

1. Routing Zone(RZ)の設定

- ルーティングテーブルを分割するVRFの作成

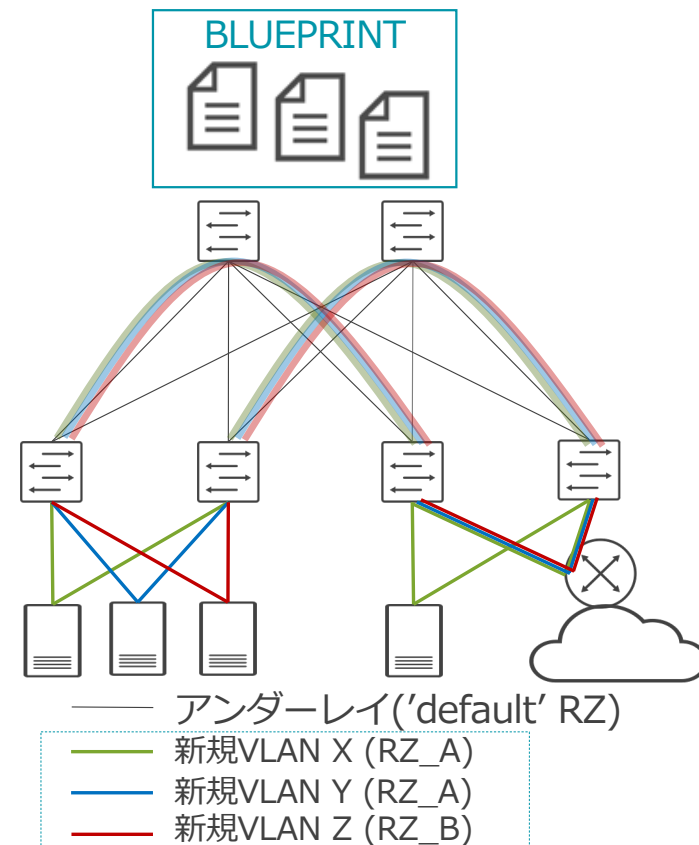
2. Virtual Network(VN)の設定

- 仮想ネットワークと関連オプション設定
- L2 vlan tag、L3アドレスレンジ など

3. Connectivity Template(CT)の設定

- 仮想ネットワークと接続するLeafのアクセスポートを指定
- Connectivity Template利用

4. Commit(機器に設定反映)



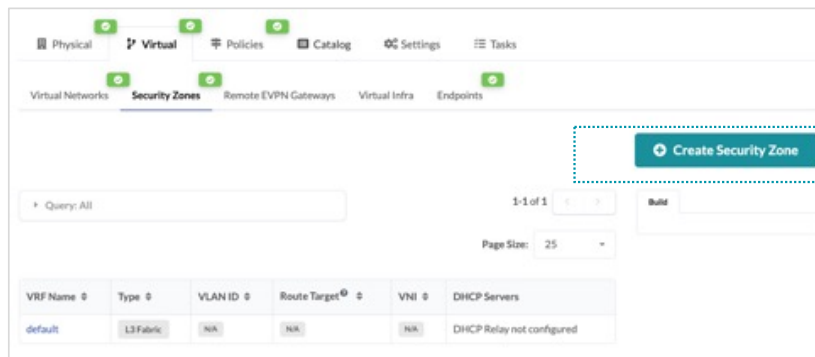
仮想ネットワーク設定 : Routing Zone

Routing Zone(VRF)

Virtual Network (VN)

Connectivity Template(CT)

- Routing Zone(RZ) で各ルーティングテーブルを分割
- アンダーレイは、"default"のRZを利用するためオーバーレイは新規にRZを作成
- RZに必要なオプション(RZ外部から許可する経路等)を設定し、追加することでRZを作成



VRF Name: Finance

VLAN ID: 2

VNI: 5001

Routing Policies

Import Policy: ☒ Default ☐ All ☐ Extra Only

Extra Import Routes

Prefix: 172.16.0.0/16 GE mask: 20 LE mask: 24

Add

仮想ネットワーク設定 : Virtual Network

Routing Zone(VRF)

Virtual Network (VN)

Connectivity Template(CT)

- ・ サーバーなどワークロードが通信に利用する仮想ネットワークを作成し、RZに割り当て

Type
☐ VLAN ☒ VXLAN

① Will create single VXLAN for all selected nodes

Name*
Finance-app1 → VNの名前

Routing Zone
Finance → VRFを選択

VNI(s)
From → VXLAN ID. 未指定は自動アサイン

Set same VLAN ID on all leafs?
☒ → 全リーフに同じVLANをアサインする場合

VLAN ID (on leafs)
101 → 全リーフにアサインするVLAN

Route Target
Not assigned

DHCP Service
☒ Disabled ☐ Enabled

IPv4 Connectivity
☐ Disabled ☒ Enabled

IPv4 Subnet
10.1.101.0/24

Virtual Gateway IPv4 Enabled?
☒

Virtual Gateway IPv4
10.1.101.1

このVNでDHCPリレーを使用

IPv4 Subnetを有効

VNのサブネット

Anycast GW
(GenericのデフォルトGW)を有効

Anycast GW

Build

1/1 Finance: Virtual Network SVI Subnets

1/1 VNI Virtual Network IDs

1-1 of 1

Pool Name
☒ Default-10000-20000

リソースからVNIを割り当て

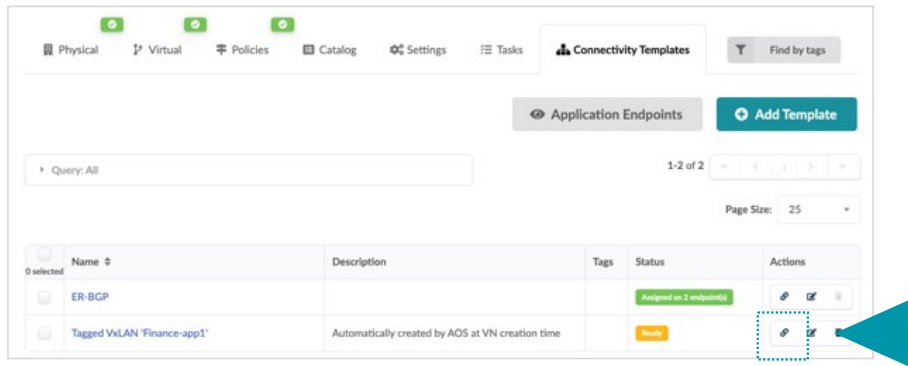
仮想ネットワーク設定 : CT

Routing Zone(VRF)

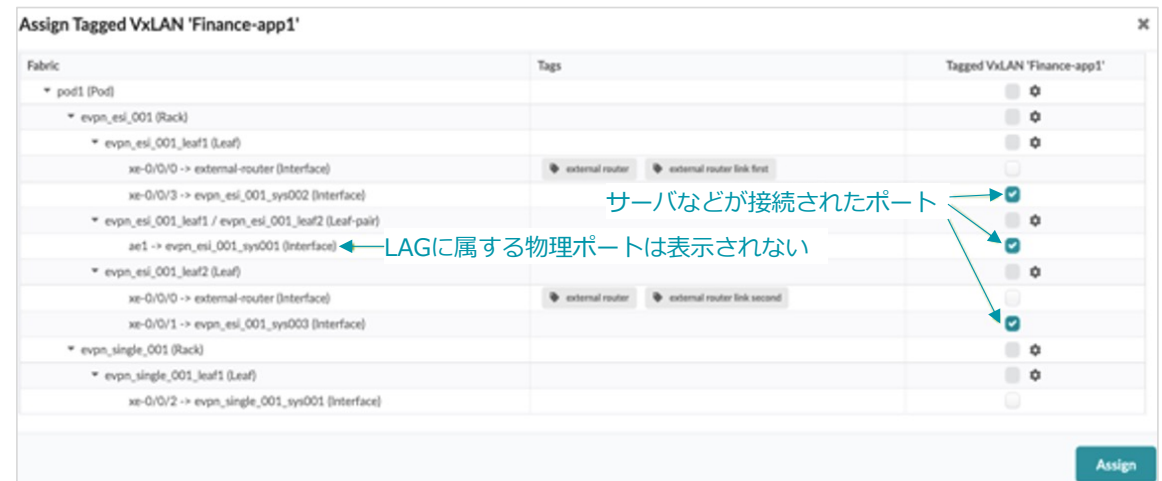
Virtual Network (VN)

Connectivity Template(CT)

- 仮想ネットワークを物理インターフェースに割当



作成した仮想ネットワークをLeafポートへアサイン





Agenda

- 製品コンセプト
- JUNIPER APSTRAのご紹介
 - 製品概要
 - デザインパターン(DC Reference, Freeform)
 - 機能紹介(IBN, ACS, AI cluster etc.)
 - コンポーネント
- オペレーションサンプル
- 購買情報
- お役立ち情報

購買情報



Juniper Apstra™

Apstra ライセンス体系

デバイス毎の 1/3/5 Year Terms ライセンス

Standard

Basic configuration
and operations

**6-month Advanced trial included
with Standard purchase:**

- 1 Blueprint per Instance
- Basic Telemetry/Intent Based Analytics (IBA)
- 3/5-Stage IP Clos Fabric, Collapsed Fabric, Freeform
- Device management
- Platform management

Advanced

Full operation, assurance,
and intent-based analytics

All Standard features plus:

- Up to 3 Blueprints per Instance
- Advanced Intent Based Analytics (IBA)
- Apstra Telemetry Streaming
- Root-Cause Identification (RCI/RCA)
- Data Center Interconnect (OTT DCI)

Premium

Large scale, multi-vendor
and policy control

All Advanced features plus:

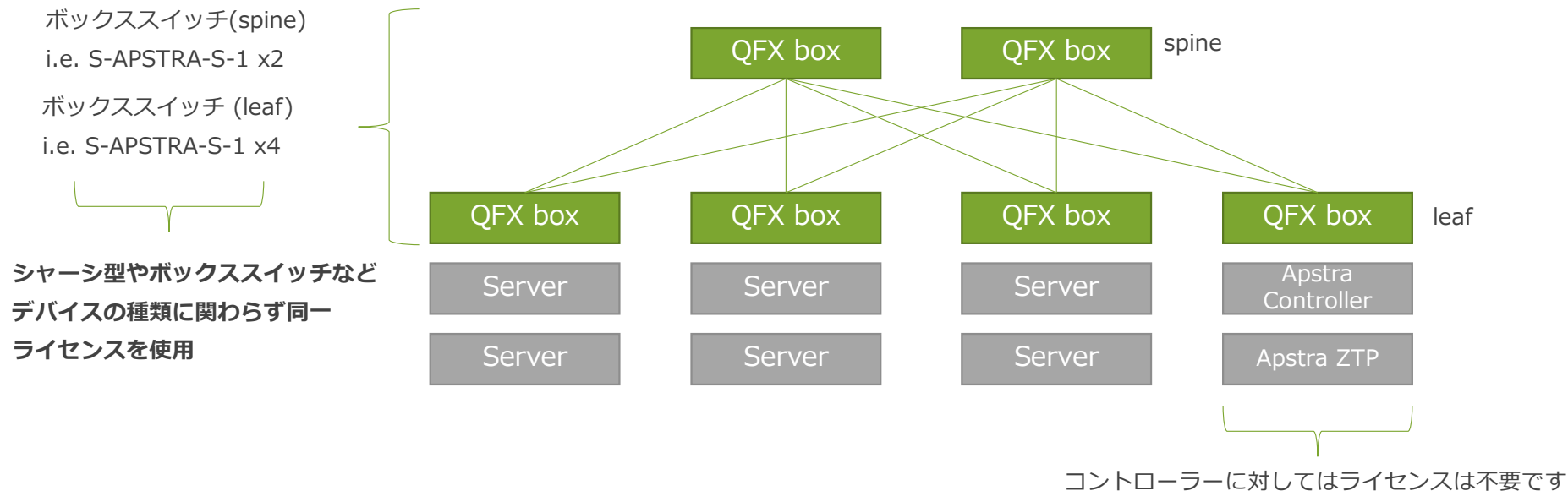
- More than 3 Blueprints
- 3rd Party Vendor Fabrics
- Policy Assurance
- Apstra Flow
- Apstra Cloud Services Service Assurances and Impact Analysis

VMware integration license

ライセンス計算例

合計6台構成の場合

スタンダードライセンスの場合、S-APSTRA-S-1 x 6



Apstra Professional Services

メリット:

- ✓ データセンターの導入をシンプルにし価値実現までの時間短縮
- ✓ Juniper Apstraへの移行をリスク低減し実現

デプロイサービス

1. データセンターへの導入
2. ユースケース例:
 - Data Center Interconnect(DCI)を使った2つのデータセンターへのデプロイ
 - 要件に沿ったカスタマイズデプロイ

移行サービス

1. 既存のレガシーネットワーク(非EVPN VXLAN)からApstraへ移行
2. 置き換え移行: Apstra管理ではないEVPN VXLANネットワークをApstraへ移行
3. 要件に沿ったカスタマイズ移行



高信頼



ビジネス
俊敏性



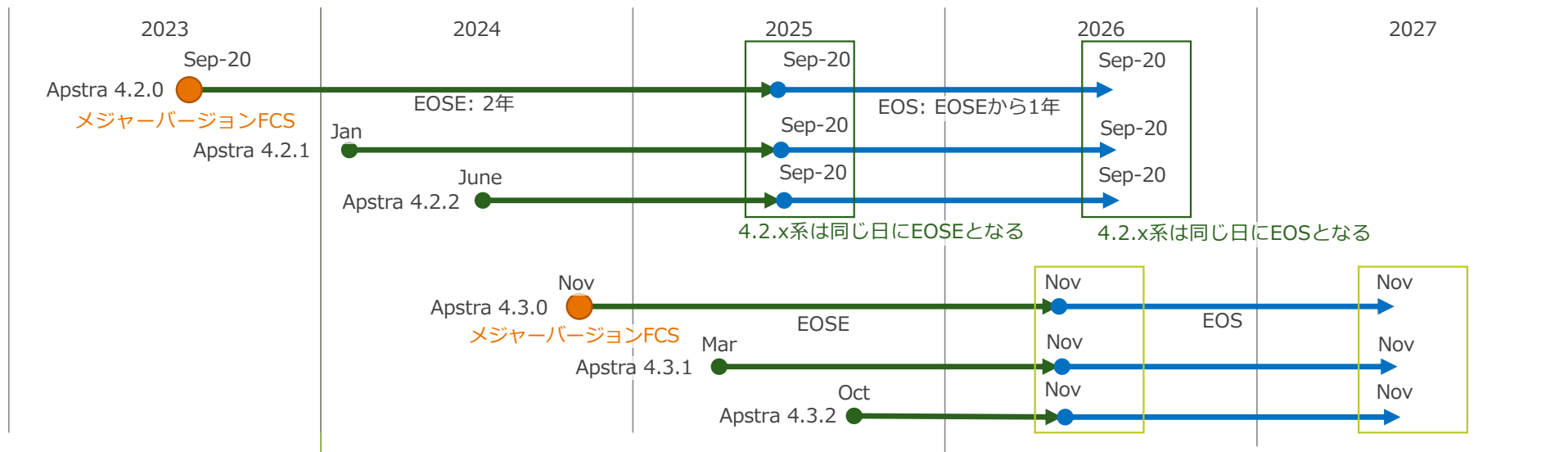
デプロイまで
時間短縮



リスク低減

Apstra ソフトウェア サポートポリシー

- メジャーバージョン First Commercial Ship (FCS)
 - 各メジャーバージョンの最初のリリース日 (Apstra4.2.0であれば、Apstra4.2.0がリリースされた日)
- End of Software Engineering (EOSE)
 - サービス/メンテナンスリリースやパッチなどのエンジニアリングサポート提供は、メジャーバージョンのFCSから2年間
 - FCSリリース後のサービス/パッチリリースはエンジニアリングサポートがFCSのバージョンと同じ日付 (Apstra4.2.1はApstra4.2.0のEOSEと同じ)
- End of Support (EOS)
 - エンジニアリングサポートを除くサポート期間は、メジャーバージョンのFCSから3年間 (EOSE同様にApstra4.2.1はApstra4.2.0のEOSと同じ)



*TSB: [Apstra] Supported Releases of Juniper Apstra
<https://kb.juniper.net/InfoCenter/index?page=content&id=TSB18062>



Agenda

- 製品コンセプト
- JUNIPER APSTRAのご紹介
 - 製品概要
 - デザインパターン(DC Reference, Freeform)
 - 機能紹介(IBN, ACS, AI cluster etc.)
 - コンポーネント
- オペレーションサンプル
- 購買情報
- お役立ち情報

お役立ち情報



Juniper Apstra™

GUI紹介デモサンプル

各種デモなどご相談ください

Day0 インストールデザイン

- Logical Device
- Rack Type
- Template
- Blueprint
- Device Profile
- Interface Map
- Agents / Managed Device
- Resource Pool
- Configlet

Staged Blueprint

Day1 設定運用

- Blueprint Active
- Staged
(Resources, Device Profiles, Devices)
- Configlet / Rendered-config
- Commit / Rollback

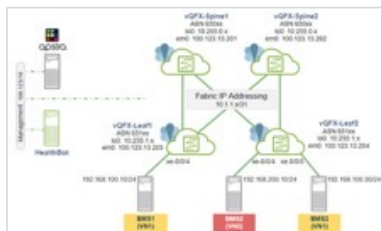
Active Blueprint

Day2+ 変更、可視化

- Config change detection
- Default Telemetry
- IBA(routing table, BGP)
- Apstra Telemetry
- Event log
- Show tech
- Maintenance mode
- vCenter連携
- API, Python
- SSOT

各種リソースがございますので是非Apstraをお試ください

Try It. Right Now.



- ・クラウドベースのラボ環境
- ・オンデマンドでご利用いただけます

<https://vlabs.juniper.net>

Learning Portal



https://learningportal.juniper.net/juniper/user_activity_info.aspx?id=12392

Youtube Playlist



<https://juniper.net/apstra-playlist>

Juniper Japan Apstra公開資料サイト



Apstra

製品概要

- ・Apstra Fabric Conductor ジェネラルプレ
ゼン

動画・デモ

- ・5分でわかる Apstra AOS/AIS でEVPN-VXLANネットワークの構築と運用を自動化
- ・Apstra AOS/AIS デモ
- ・ネットワーク監視の自動化

VLAB(無償:仮想ラボ環境)

- ・vLAB (英語)
- ・vLAB登録・利用ガイド
- ・vLAB - Apstra Fabric Conductor利用ガイド

各種資料もごございますので活用ください

- Apstra日本語簡易マニュアル
- テクニカル資料等 など

<https://www.juniper.net/jp/ja/local/solution-technical-information/software.html>

ハンズオン

Juniper Japanにて実施しておりますのでご相談ください

トレーニング環境	実施方法	対象人数	期間/時間
クラウドラボ (AWS上の仮想環境)	オンライン 事前にJuniper側でクラウドラボ環境を構築 *2週間環境は維持されます。	最大6人程度 (要相談)	都度相談 4時間 (要相談)

トレーニングアジェンダ

1. Apstra Serverインストール (説明のみ)
2. Apstra Agentインストール
3. Device Profile定義
4. Logical Device定義
5. Interface Map定義
6. Rack定義
7. リソース定義
8. 外接Router定義
9. Template定義
10. Blueprint作成
11. Overlay Network作成
12. Configlet
13. Rollback
14. 設定差分チェック
15. 接続Serverの確認

評価、POC方法

不明な点は弊社担当営業までご相談ください

LAB/DEMO環境	利用方法	対象	期間	用途
VLAB(オンライン)	vLABへ登録 (メールアドレス/氏名 等)	どなたでも	最大6時間	マーケティング・セールスツールとして
クラウドラボ (オンライン)	発行者が限られるため Juniper担当者に依頼お願いします (メールアドレス/氏名/社名)	お客様 パートナー	Default 2週間 *延長は要相談	デモ・評価環境として * Apstraはインストール済み
自社LABでの評価用 Apstraダウンロード	JuniperサイトよりDL 案件情報共有が必要	お客様 パートナー *社内はDL可	POC申請期間	インストール評価 オンプレミスでの評価 物理障害などの評価



Juniper Apstra™ - Route to learning

1. Japanソリューション情報サイト

<https://www.juniper.net/jp/ja/local/solution-technical-information/software.html>

f 2. IBN for Dummies

<https://www.juniper.net/content/dam/www/assets/ebooks/us/en/intent-based-networking-for-dummies.pdf>

3. Juniper Apstra Demo Library

www.juniper.net/us/en/dm/apstra-demos.html

- Short videos of Apstra in action
- Click-through demos with scripts

4. Free Trial with Cloudlabs

www.juniper.net/us/en/forms/apstra-free-trial.html

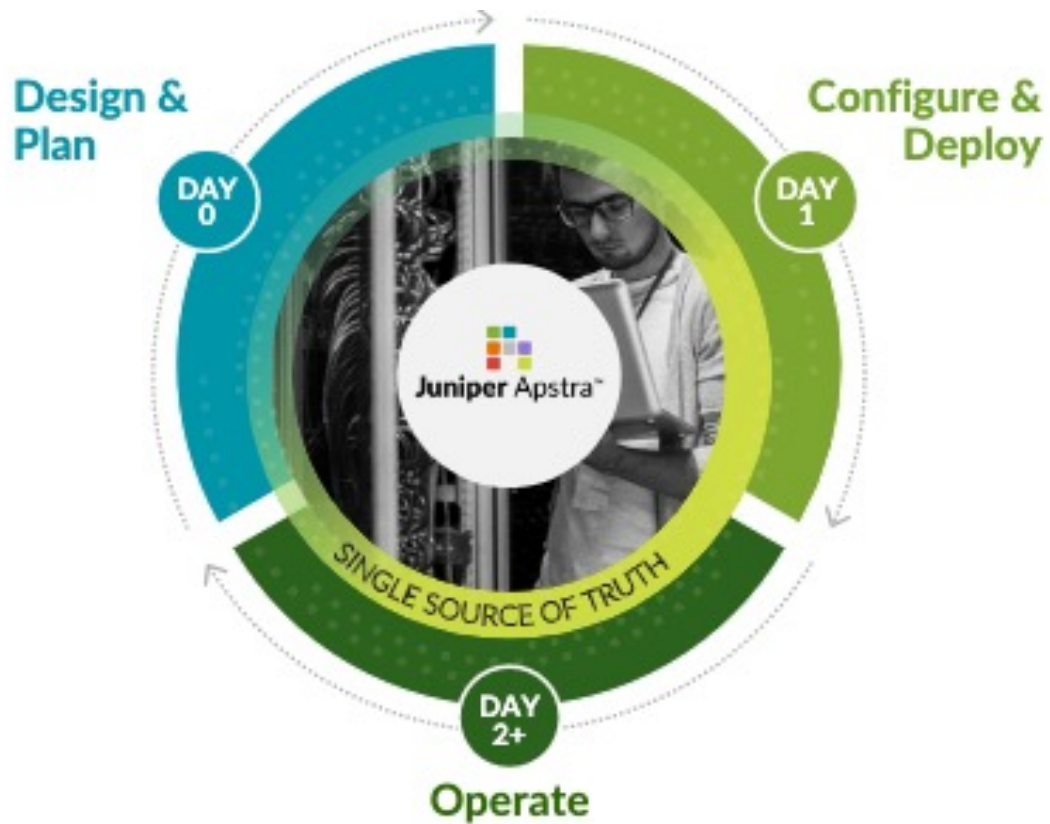
- Cloud-based lab environment (access from anywhere)
- Pre-built or build your own topologies
- Self-paced and user-friendly
- Complete in a short amount of time
- In Focus guides

5. Apstra Instructor Led 5-day Training

https://learningportal.juniper.net/juniper/user_activity_info.aspx?id=12600

This instructor-led course is the comprehensive course to learn managing data center networks with Juniper Apstra.

まとめ



高い信頼性



容易な構築・運用



即時の状態把握



THANK YOU

JUNIPER
NETWORKS®

Driven by
Experience™