

# 通信事業者におけるSDN/SDIの取り組みと 実現に対する課題について

NTTコミュニケーションズ株式会社  
技術開発部 栗原 良尚



# Agenda

- はじめに
  - 自己紹介
  - これまでのSDNの取り組み
- 背景
  - 我々が実現したいSDN/SDI
  - AMPPのユースケースの紹介
- AMPPについて
  - AMPPアーキテクチャ
  - AMPPの各種コントローラの概要紹介
  - Operation Operation GUI
- 将来的な取り組みについて
- さいごに

# 自己紹介

## ■名前

栗原 良尚 (Yoshinao Kurihara)

## ■経歴

2010/04	<b>NTTコミュニケーションズ株式会社</b>	入社
2010/08	<b>インターネットマルチフィード株式会社</b> IX/AS7521運用保守・開発業務	出向
2013/07	<b>NTTコミュニケーションズ株式会社</b> SDN担当/NFV担当	異動

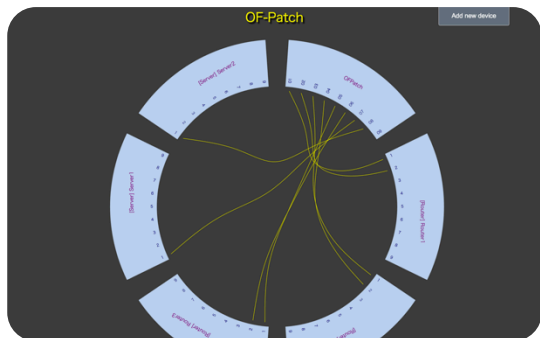
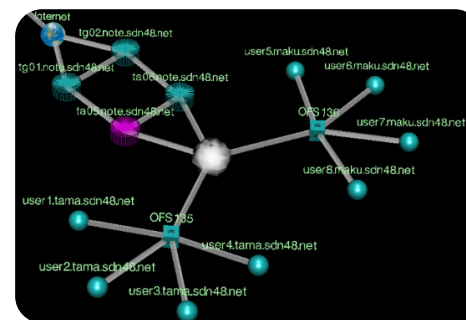
# これまでのSDNに関する取り組み

- SDNを利用したネットワーク運用システムPoC開発
  - 運用の自動化
  - 検証ネットワーク構成変更

## Programing Contest Winner

~2013/12 in 沖縄オープンラボ~

参考：<https://www.facebook.com/okinawaopenlabs>



## OpenFlowPatchPanel 2<sup>nd</sup> Winner

~ONS Hackathon 2014/3~

参考：<http://sdnhub.org/events/sdn-hackathon-ons-2014/>

## INTEROP 2014 Finalist

~SDI ShowCase AMPP 2014/6 ~

参考：<http://www.rbbtoday.com/article/2014/06/12/120756.html/>



# 開発背景

# SDNに期待すること

## Time-to-Marketの短縮

- ・ 自社に必要な機能を必要な時に開発してスピーディに導入可能

## サービスの差異化

- ・ 市中製品の仕様に縛られず、ユーザーの意図に沿ったアプリケーションを実装可能
- ・ 動的で即応可能なサービス提供によるユーザエクスペリエンス向上

## コストの削減（CAPEX/OPEX削減）

- ・ 運用の自動化、標準API、マルチベンダ対応によるOPEX削減
- ・ コモディティ化した安価なSW利用によるCAPEX削減
- ・ リソース利用の最適化によるCAPEX削減

# 我々が実現したいSDN/SDI

## ■ 通信設備まで含めたSDN/SDI化

- 現在のSDN化ターゲットの主流
  - ✓ Cloud連携
  - ✓ DC間オーバーレイネットワーク
- 我々のターゲット
  - ✓ Internet接続、DC拠点間通信、DC/Cloudまでコントロール

## ■ 既存NW機器も含めたSDN/SDI化

- 大容量トラフィックを扱うにはOpenFlow-SWだけでは困難
- 既存のNW機器もコントロールする必要

**今回はココに特にフォーカスします**



### 運用の自動化

- OPEXの削減や提供リードタイムの削減

### ユーザエクスペリエンスの向上

- 通信基盤まで連携する事でより細かなNW制御

# 今回のユースケース：社内検証網

部内開発チームや他部署で検証を行うため、ネットワーク、クラウドリソースを提供する、NTT Comのサービスネットワークを凝縮したもの

## オープンネットワーク

- AS38639

BGP, OSPF, Static, VRRP

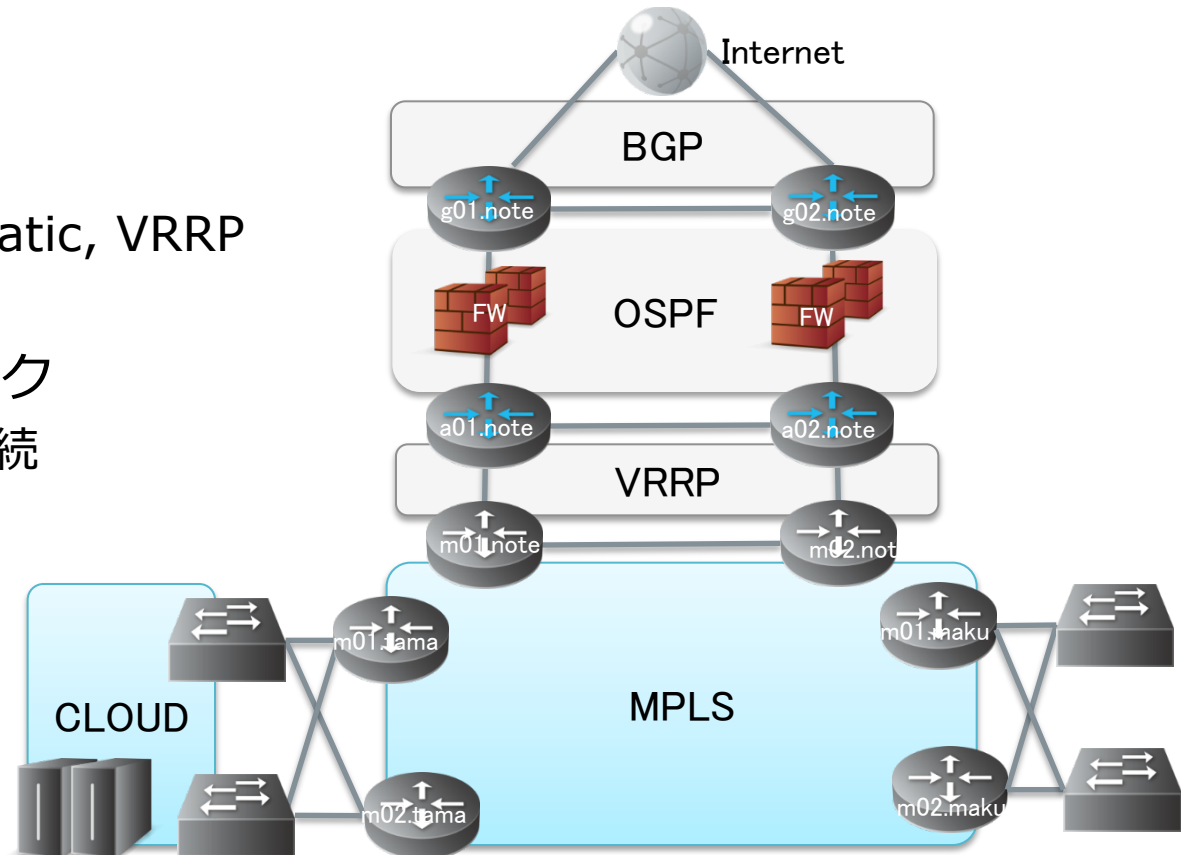
- 海外への拠点展開

## クローズドネットワーク

- 多拠点をMPLS接続

## クラウドサービス

- VMWare
- OpenStack

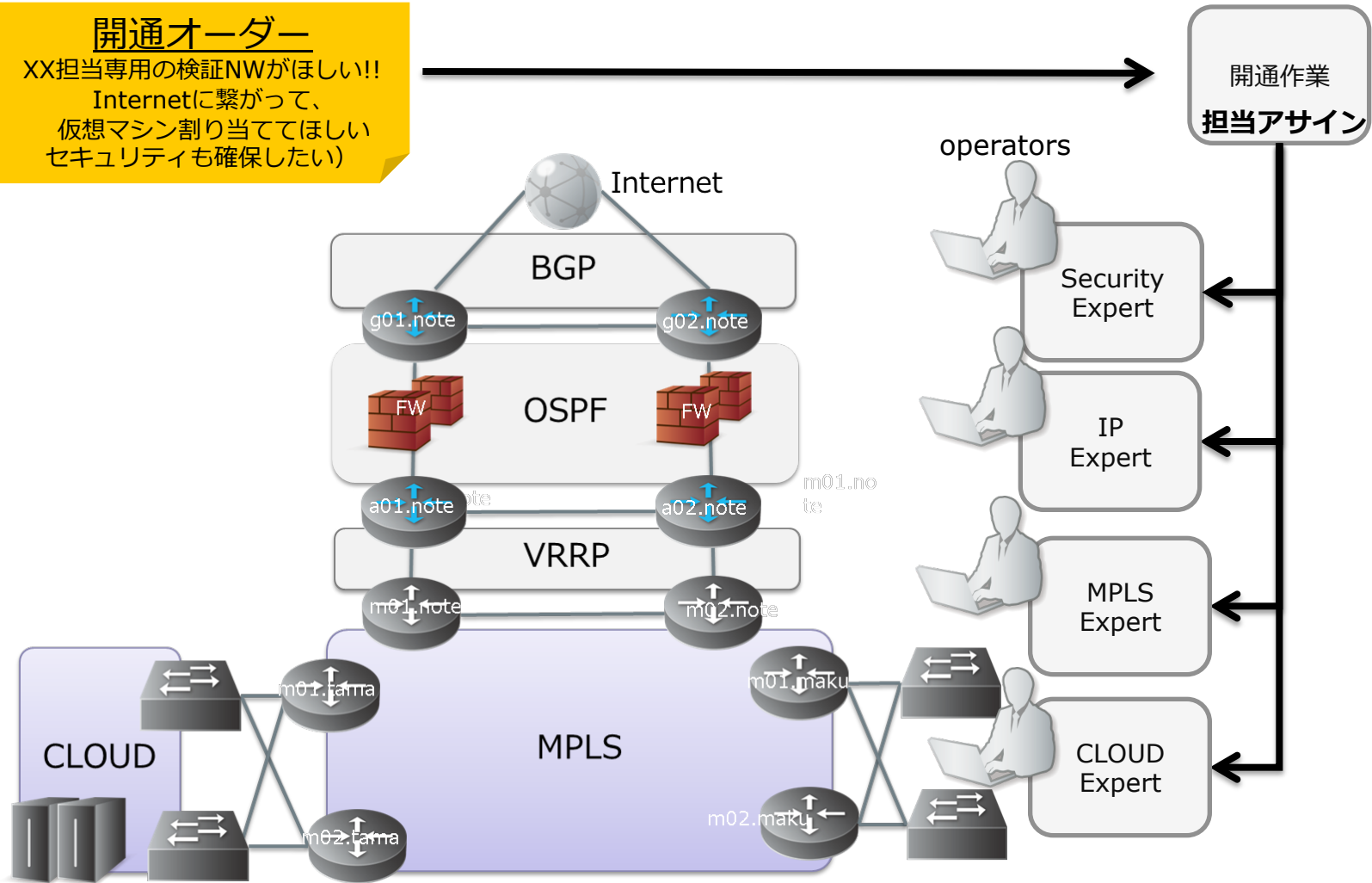




# 現状のオペレーション

## 開通オーダー

XX担当専用の検証NWがほしい!!  
Internetに繋がって、  
仮想マシン割り当ててほしい  
(セキュリティも確保したい)



# サービスオーダーの課題

## 開通オーダー

XX担当専用の検証NWがほしい!!  
Internetに繋がって、  
仮想マシン割り当ててほしい  
(セキュリティを確保したい)

開通作業  
担当アサイン

operators

## 開通業務での課題

多種類の装置を複数開通グループが人手で設定する必要があるため、

- ・ 開通に時間がかかる
- ・ 担当者毎にConfigにムラが発生
- ・ ヒューマンエラーのリスクがある

# 保守の課題

トラブルチケット  
XXX担当専用の検証NW、仮想  
マシンがつかない！！

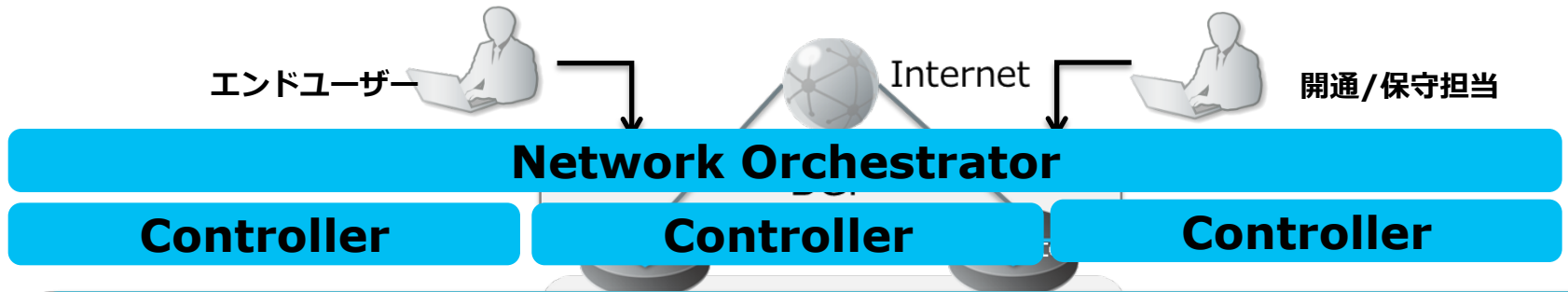
切り分け  
・復旧作業  
担当アサイン

## 保守業務での課題

複数のグループが設定した多種種類の装置を手手で切り分ける必要があるため、

- ・ 全体像の把握に時間がかかる
- ・ 各開通担当者毎のConfigのムラを読み解くスキルが必要
- ・ ヒューマンエラーのリスクがある

# 課題解決に向けたSDN化のアプローチ



**Happy Operator**



**Happy User**

## ■様々なサービス横断で設定自動化

- ✓ Routing
- ✓ FW
- ✓ MPLS
- ✓ Cloud

## ■多くのSDN Readyではない既存網の存在

- ✓ 各HWのばらばらなインタフェースを抽象化して統一
- ✓ OpenFlow/CLIを両方対応

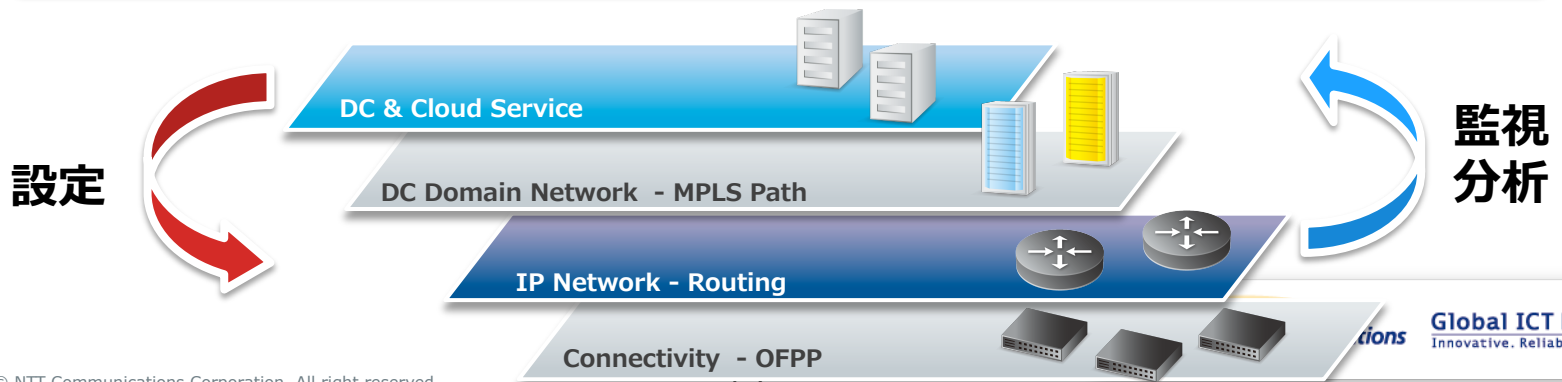
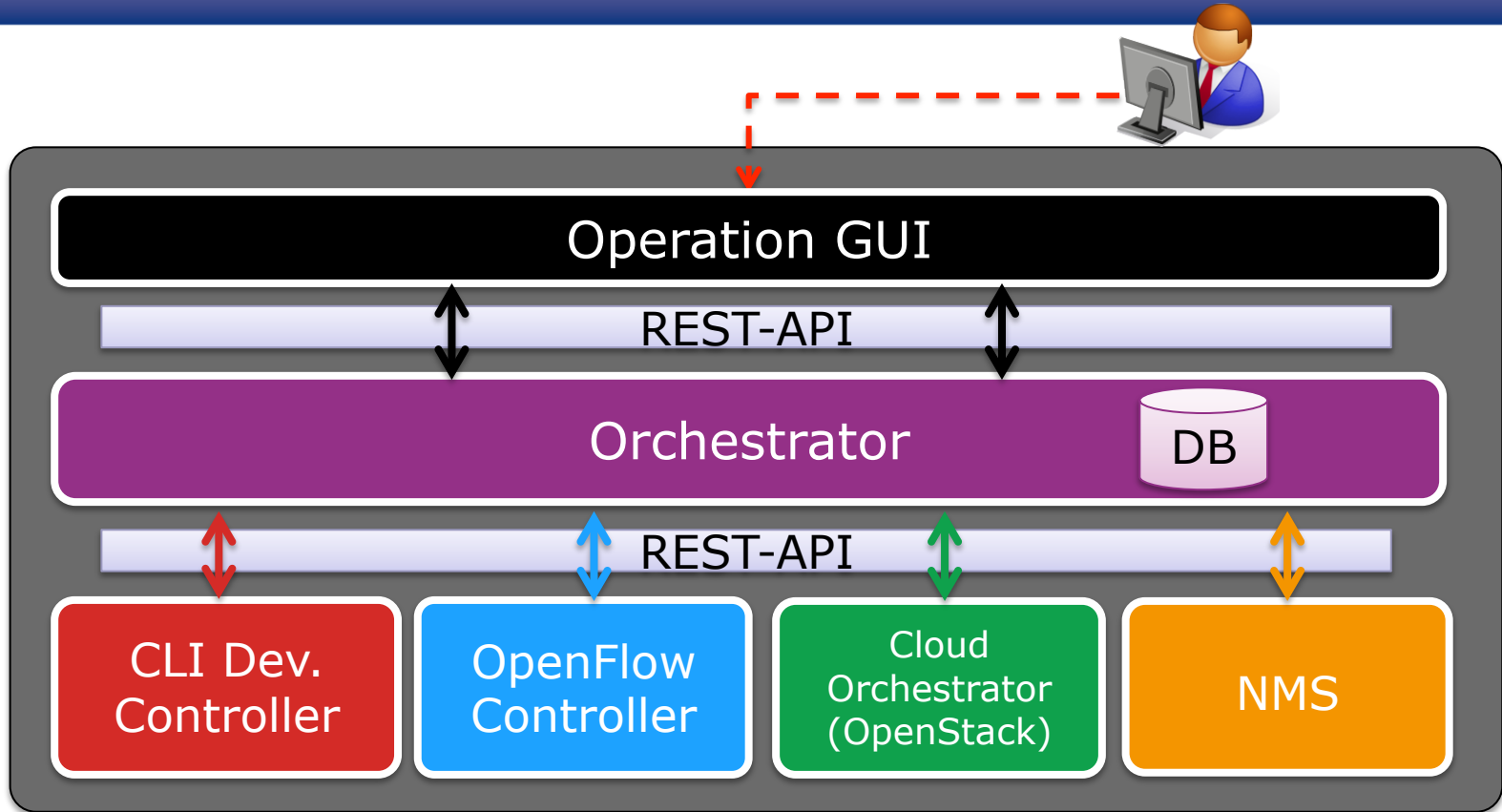
## ■システムの柔軟性

- ✓ 疎結合なAPI
- ✓ 汎用性の高いAPI
- ✓ 既存システム、市販製品など良いものは適宜導入可能

# AMPP :

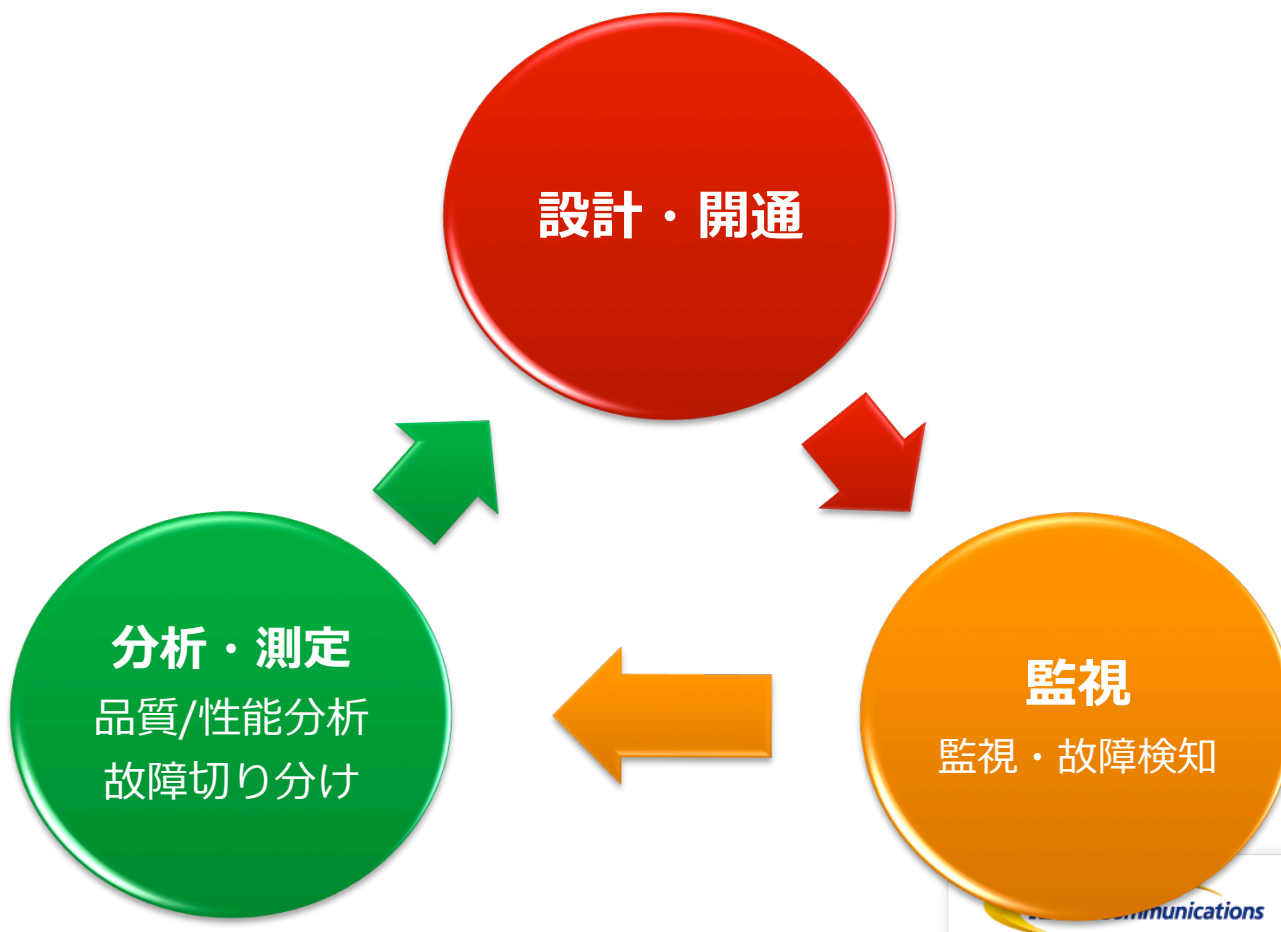
A Multiple southbound interface control Platform and Portal system

# AMPP 構成およびアーキテクチャ



# オペレーションサイクル統合

開通～監視保守のキャリア設備運用業務を一元的に実施  
オペレータがオペレーションに集中することのできるポータル



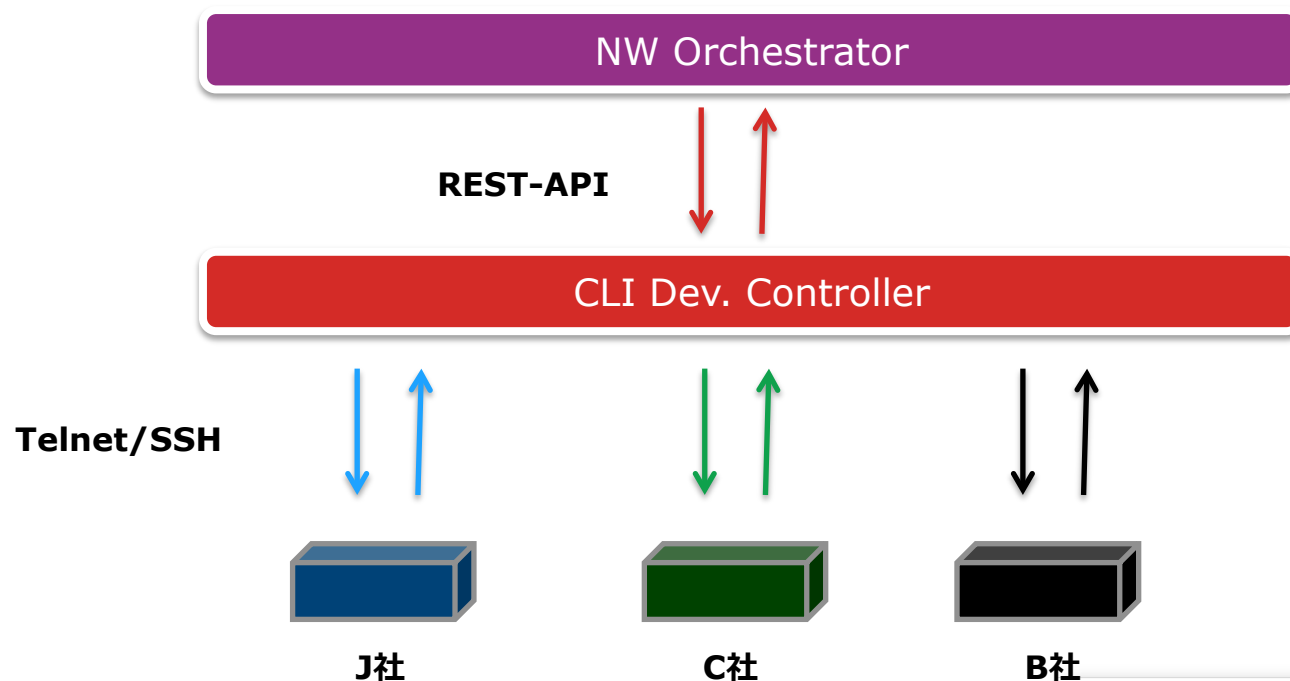


# CLI Dev. Controller概要

# CLI Dev. Controller 機能概要①

## ■ ベンダ依存コマンドの隠蔽

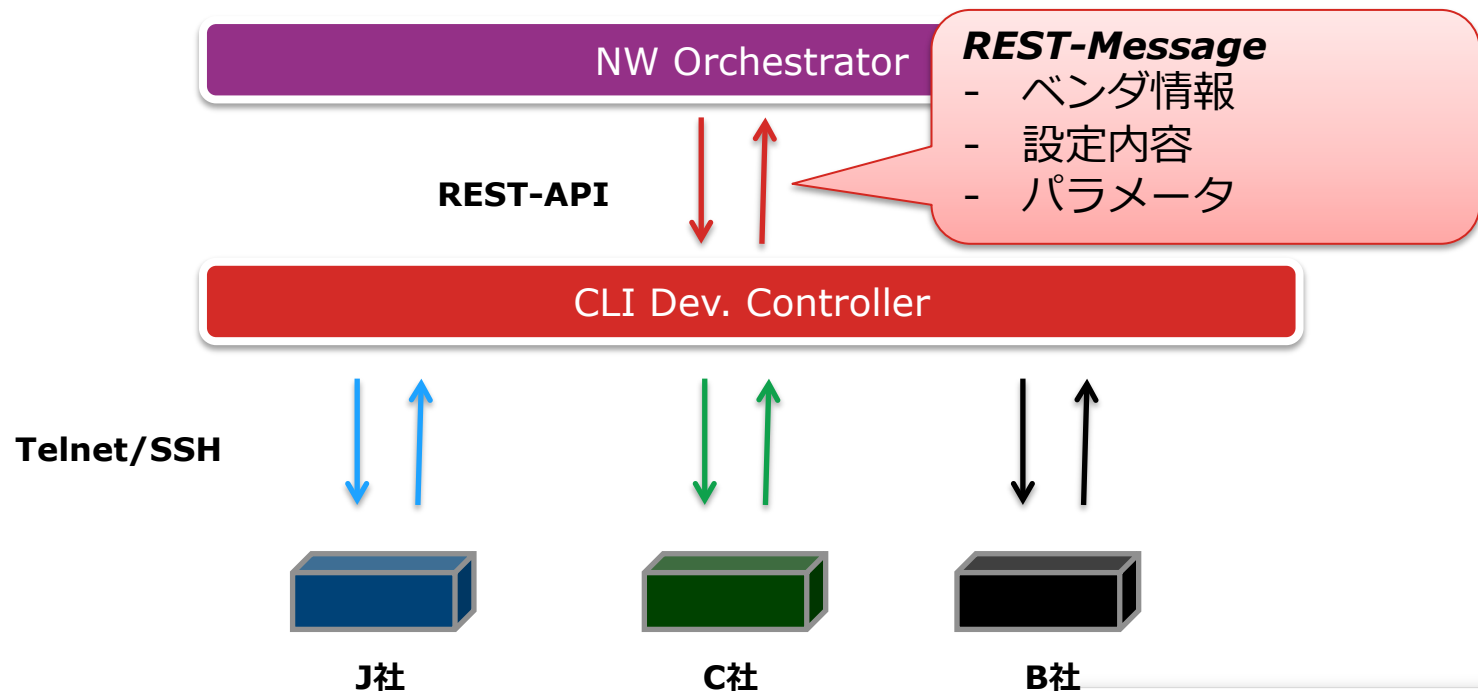
- オーケストレータ側にベンダコマンド、バージョン差異を隠蔽したREST-APIを提供
- Config設定時のエラーハンドリングとロールバック機能



# CLI Dev. Controller 機能概要①

## ■ ベンダ依存コマンドの隠蔽

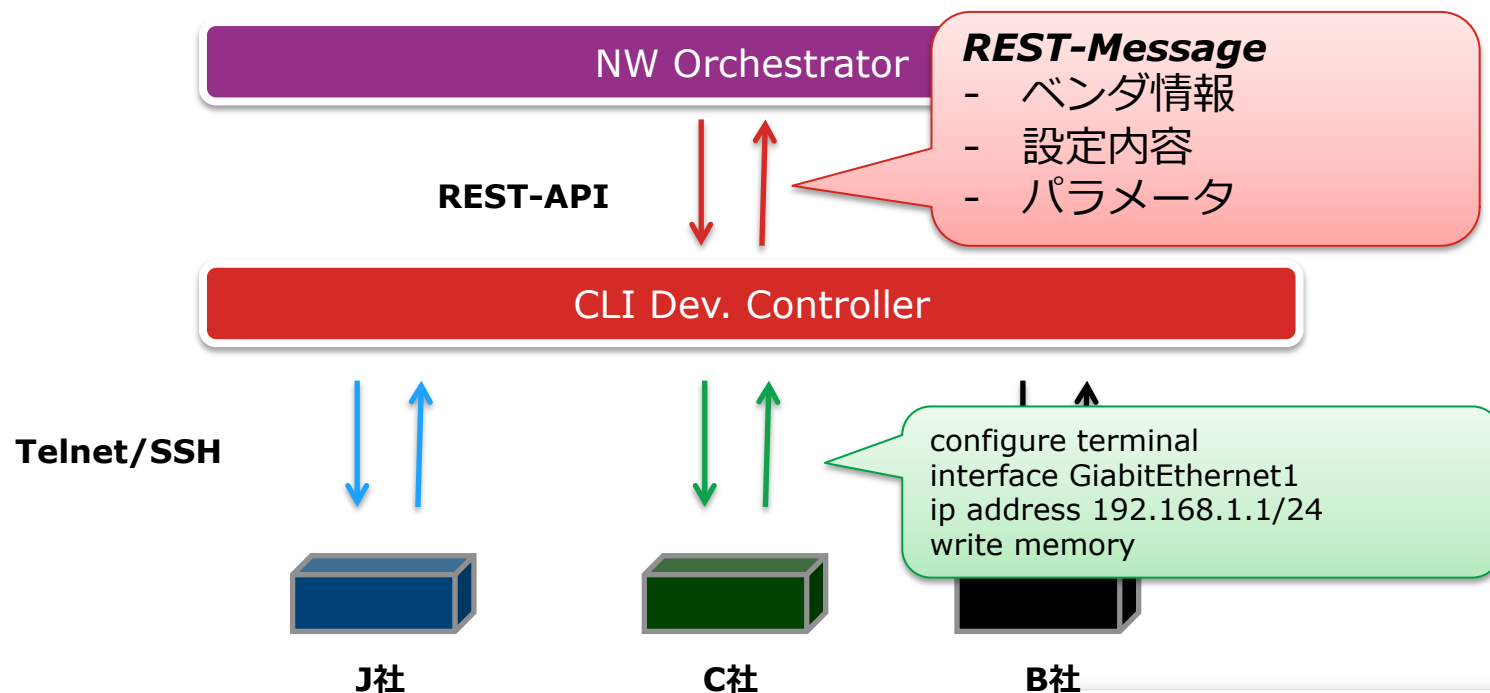
- オークストレータ側にベンダコマンド、バージョン差異を隠蔽したREST-APIを提供
- Config設定時のエラーハンドリングとロールバック機能



# CLI Dev. Controller 機能概要①

## ■ ベンダ依存コマンドの隠蔽

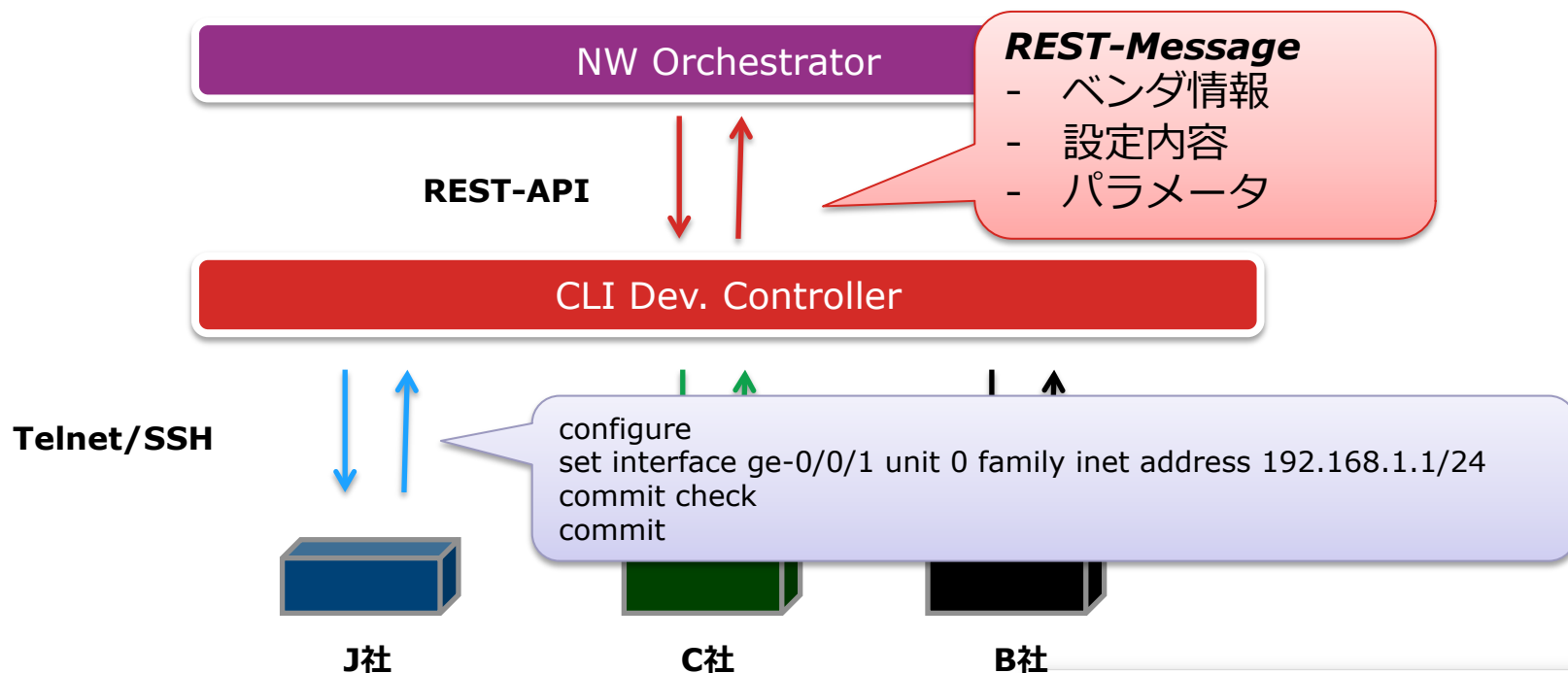
- オークストレータ側にベンダコマンド、バージョン差異を隠蔽したREST-APIを提供
- Config設定時のエラーハンドリングとロールバック機能



# CLI Dev. Controller 機能概要①

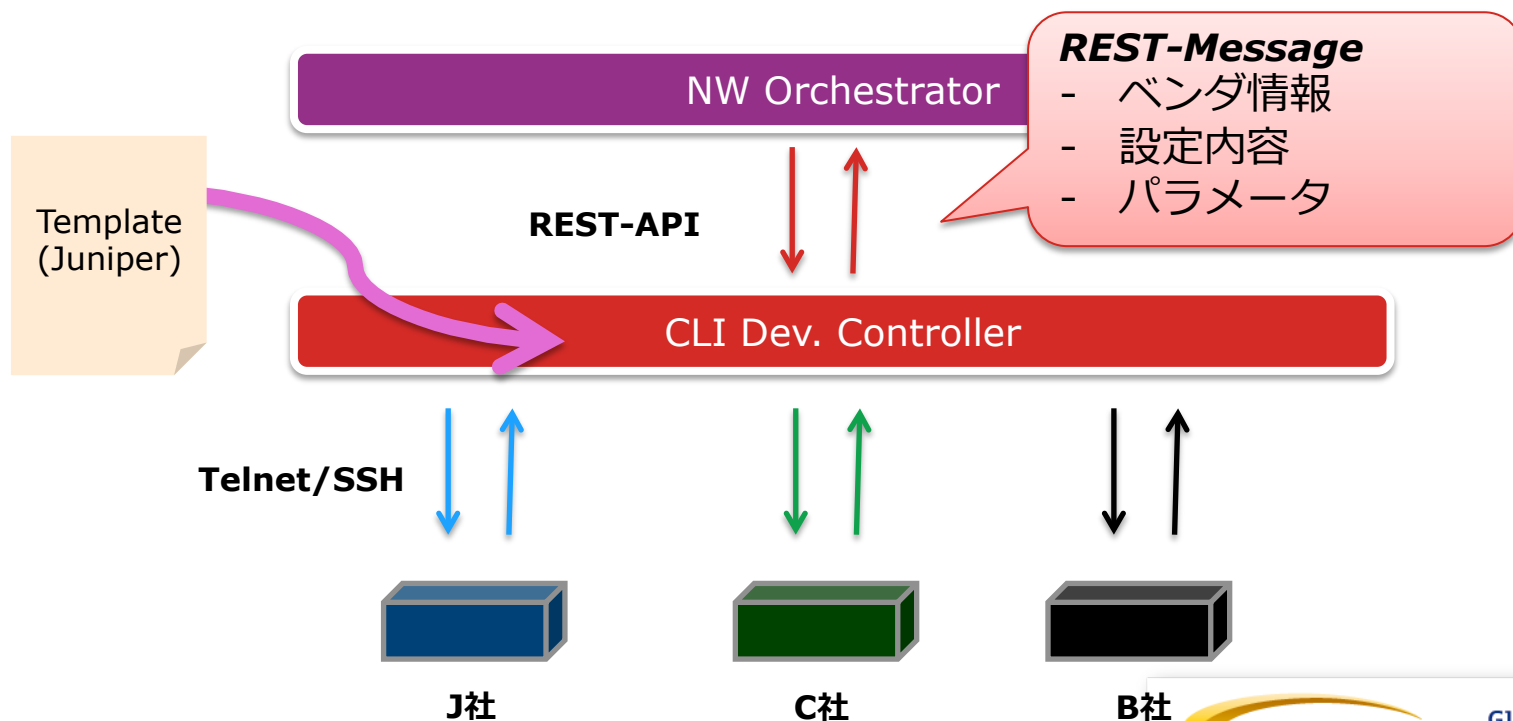
## ■ ベンダ依存コマンドの隠蔽

- オークストレータ側にベンダコマンド、バージョン差異を隠蔽したREST-APIを提供
- Config設定時のエラーハンドリングとロールバック機能



## CLI Dev. Controller 機能概要②

- 容易にREST-API追加、組み合わせ拡張が可能
  - 機能追加する場合は、Config Templateファイルを作成(プログラムの知識は必要なし)
  - 複数のAPIを組み合わせることでSO用のAPI作成可能

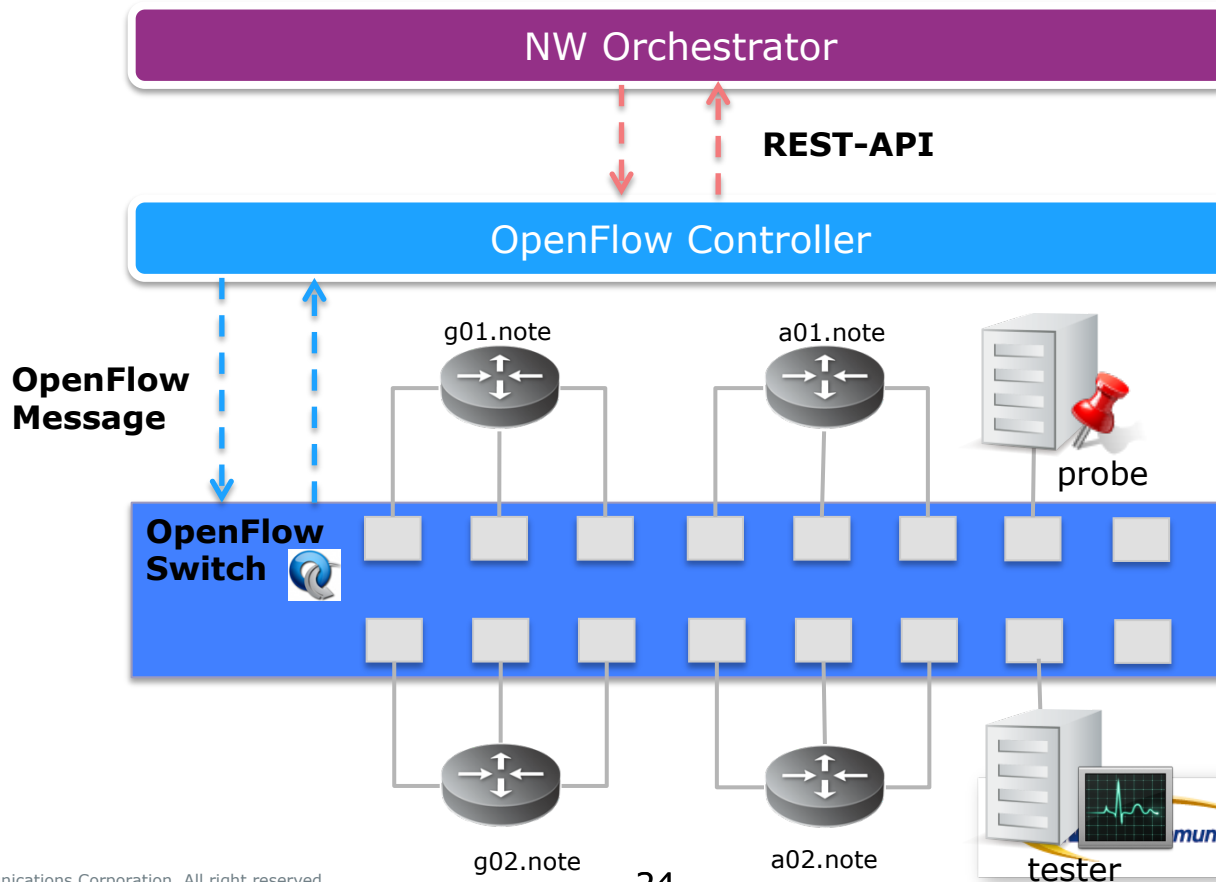


# OpenFlow Controller概要

# OpenFlow PatchPanel 機能概要①

## ■ フレキシブルなトポロジ変更

- OpenFlow-SWにRouter、SWのポートを先行配線
- OpenFlowエントリでトポロジを柔軟に設定変更

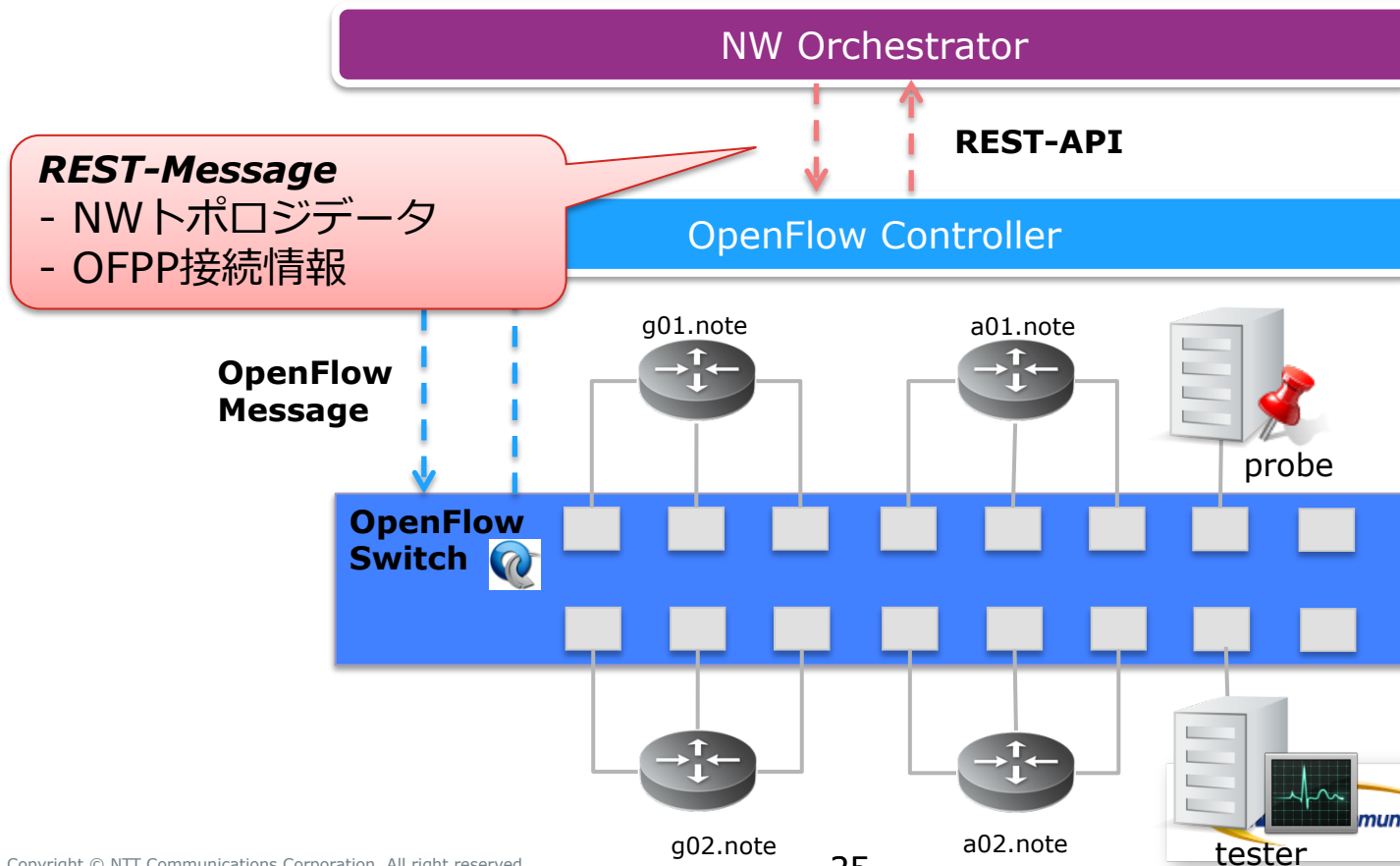




# OpenFlow PatchPanel 機能概要①

## ■ フレキシブルなトポロジ変更

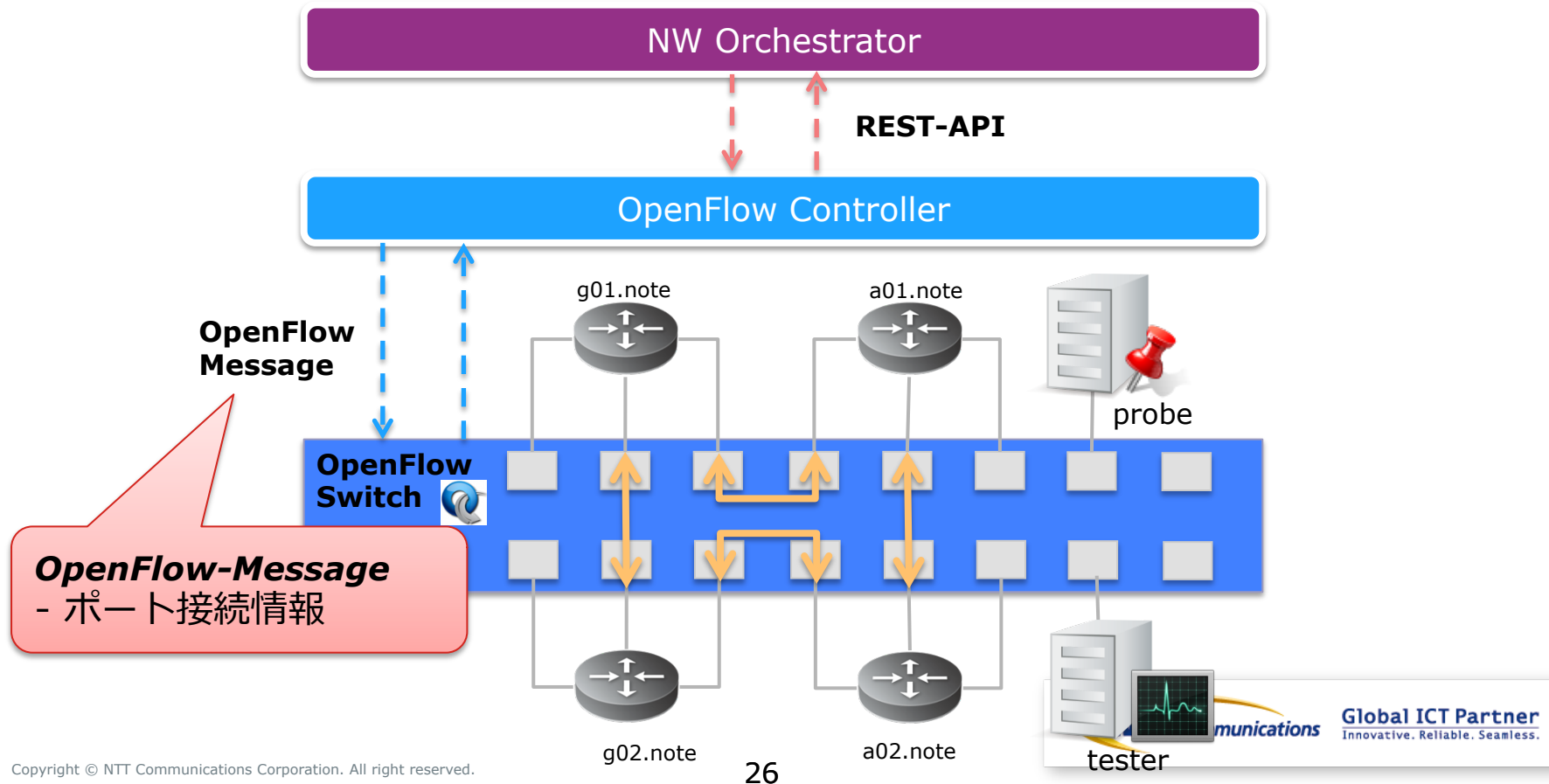
- OpenFlow-SWにRouter、SWのポートを先行配線
- OpenFlowエントリでトポロジを柔軟に設定変更



# OpenFlow PatchPanel 機能概要①

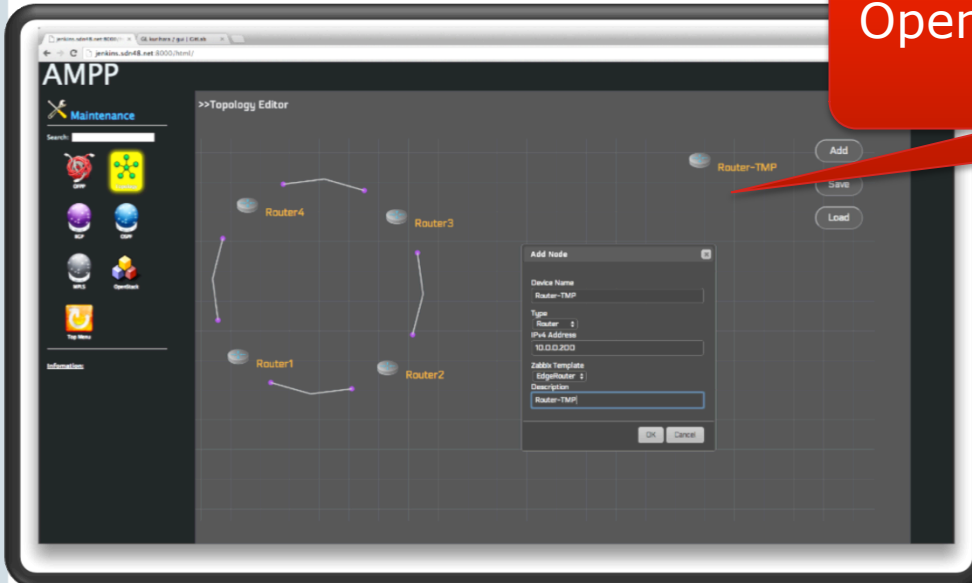
## ■ フレキシブルなトポロジ変更

- OpenFlow-SWにRouter、SWのポートを先行配線
- OpenFlowエントリでトポロジを柔軟に設定変更



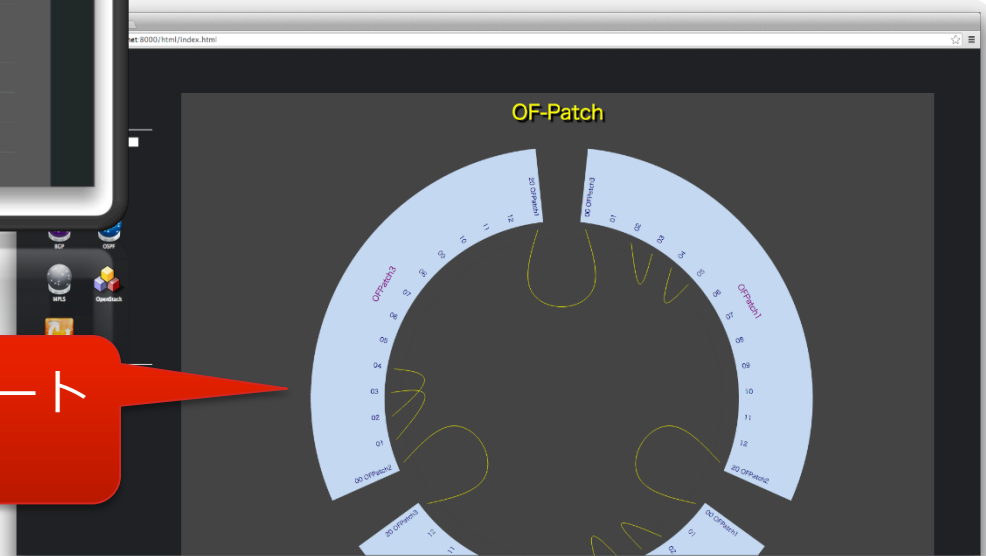
# OpenFlow PatchPanel 機能概要①

## NWトポロジデザイン画面



OpenFlowパッチパネルに登録した機器を任意のトポロジで接続

## OpenFlowパッチパネル管理画面

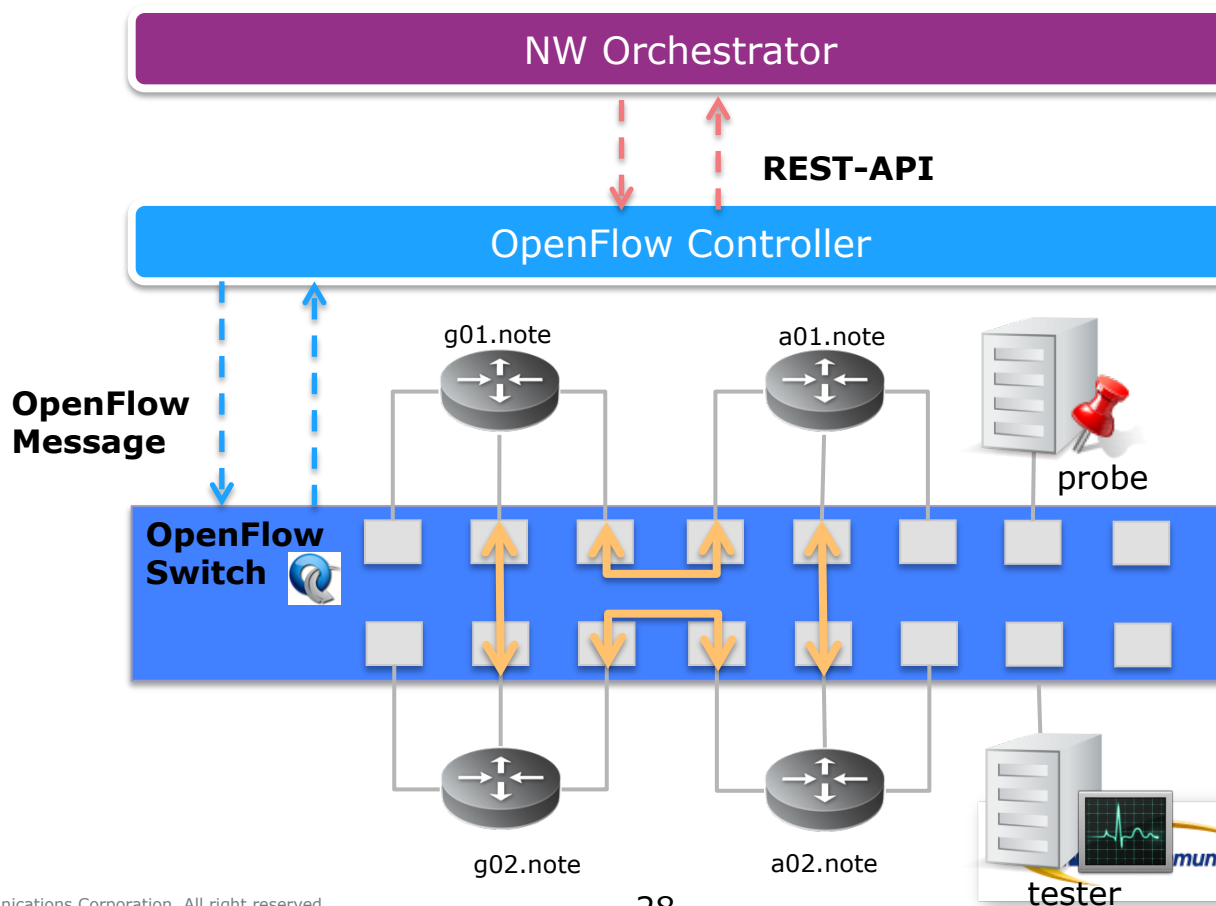


OpenFlowPatchPanelでどのポートが接続されているか管理

# OpenFlow PatchPanel 機能概要②

## ■ Probe & Filtering機能

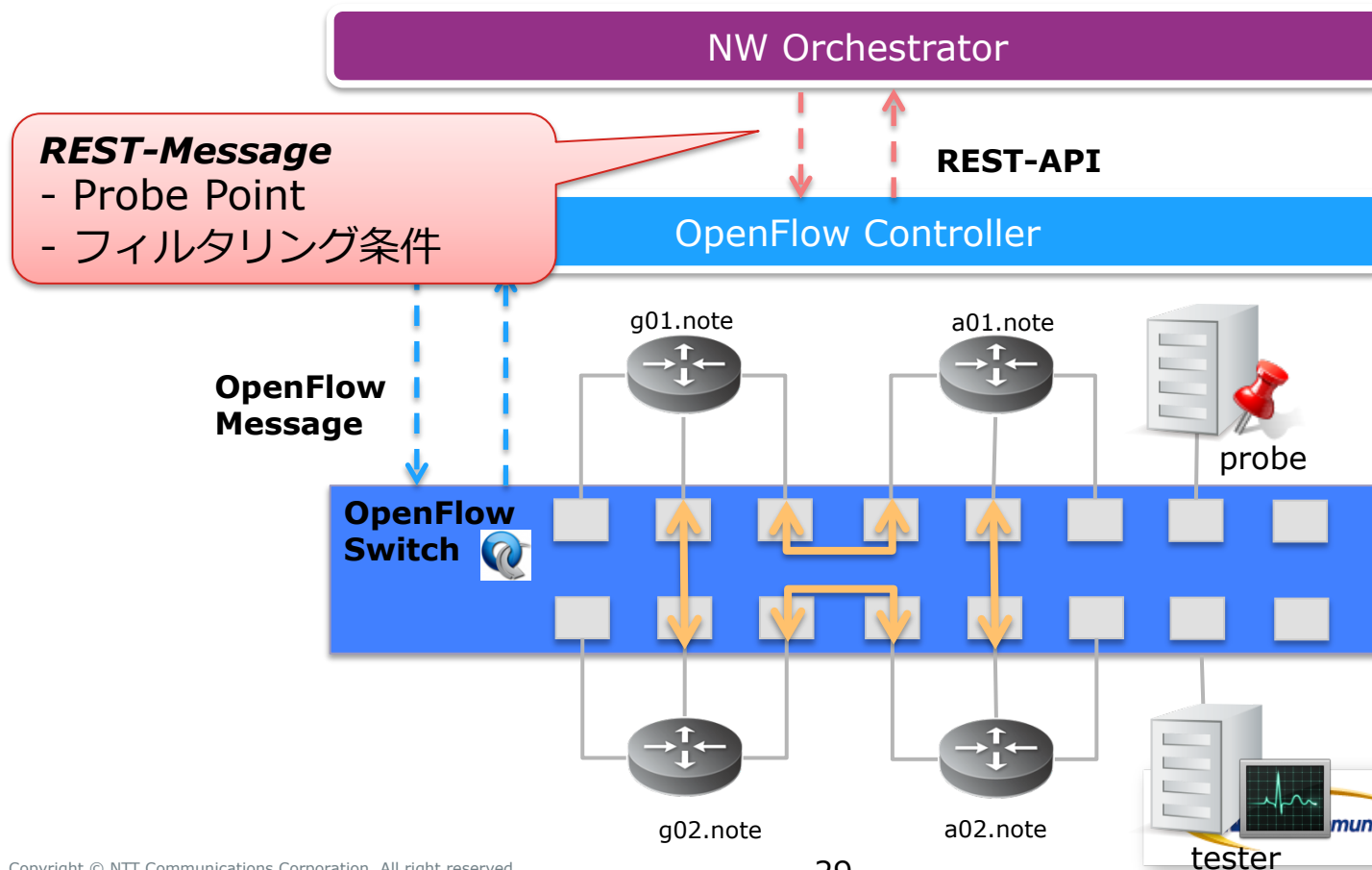
- 任意の条件(ポート,VLAN,IP) でフィルターしたトラフィックをミラーリングやフィルタリングする事でProbeサーバで解析



# OpenFlow PatchPanel 機能概要②

## ■ Probe & Filtering機能

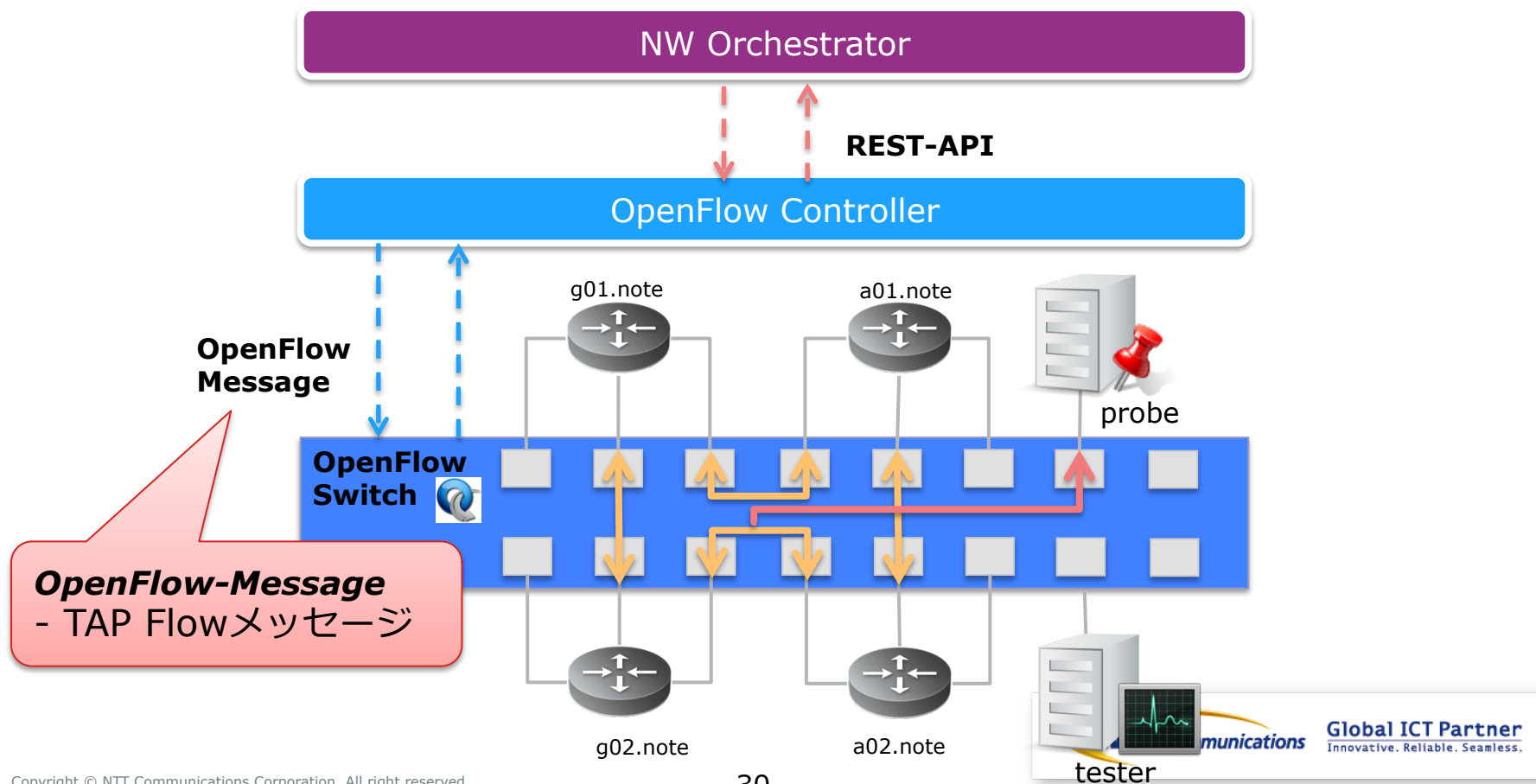
- 任意の条件(ポート,VLAN,IP) でフィルターしたトラフィックをミラーリングやフィルタリングする事でProbeサーバで解析



# OpenFlow PatchPanel 機能概要②

## ■ Probe & Filtering機能

- 任意の条件(ポート,VLAN,IP) でフィルターしたトラフィックをミラーリングやフィルタリングする事でProbeサーバで解析

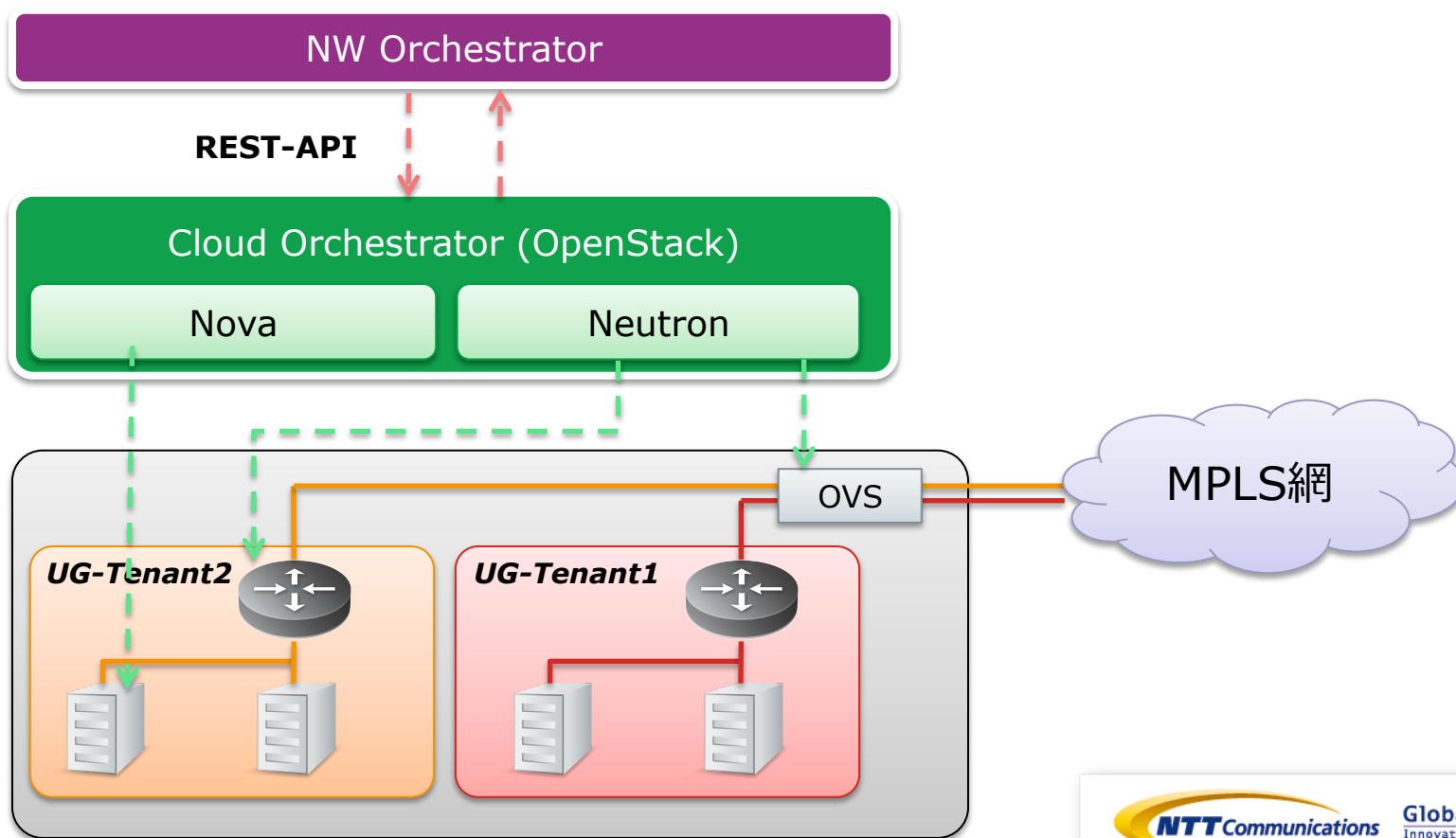


# Cloud Orchestrator概要

# Cloud Orchstrator 機能概要

## ■ Cloud OrchstratorはOpenStack-API利用

- UG-Tenantを作成
- MPLS網に接続
- 監視用VMの作成

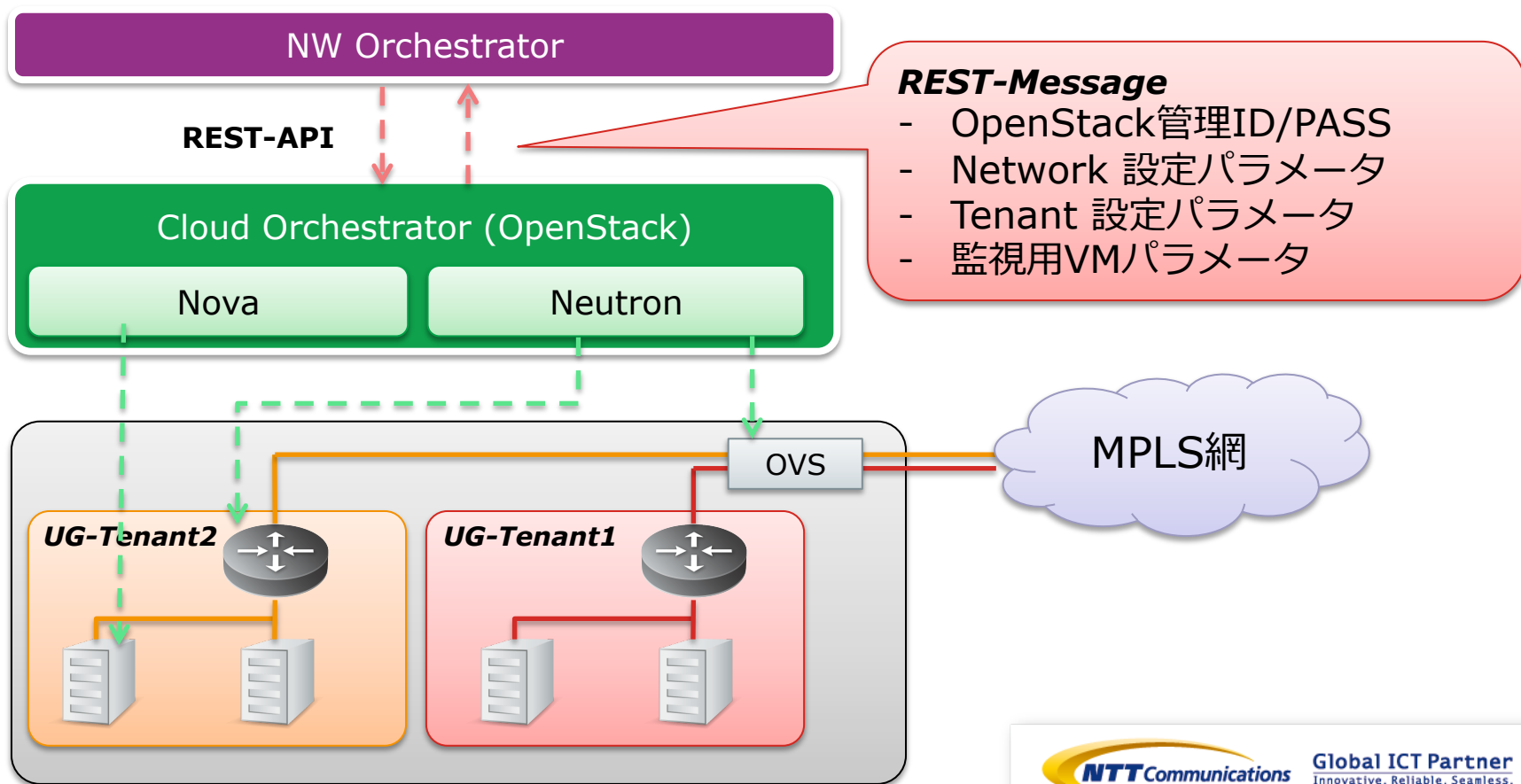




# Cloud Orchstrator 機能概要

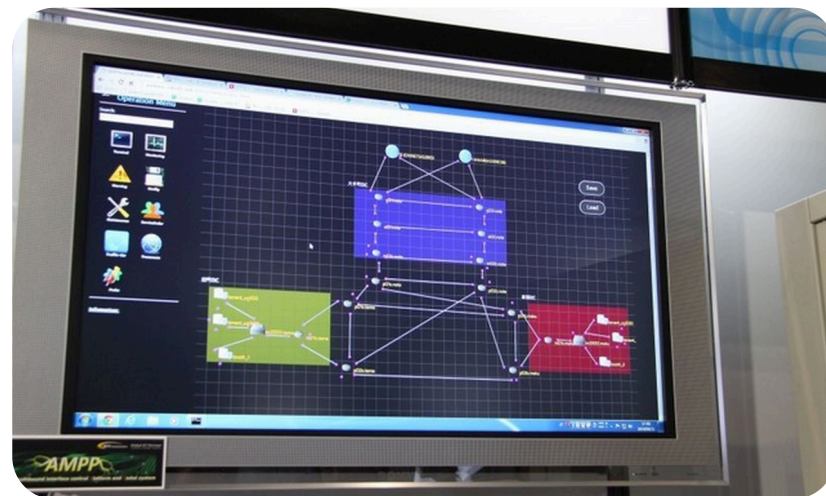
## ■ Cloud OrchstratorはOpenStack-API利用

- UG-Tenantを作成
- MPLS網に接続
- 監視用VMの作成

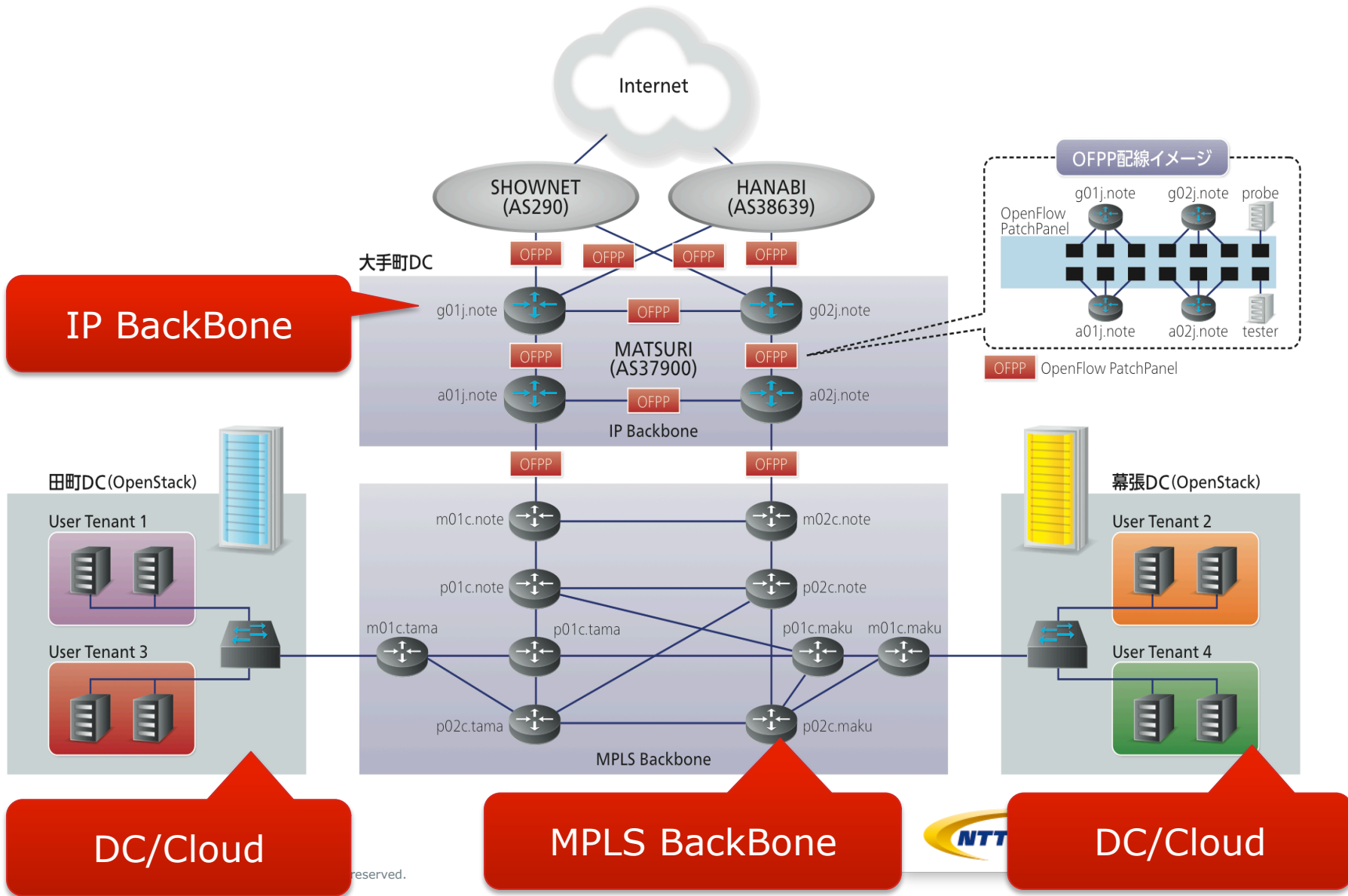


# AMPP Operation GUI

# INTEROP2014 SDI Show Case



# キャリア網オペレーションを支える SDNオーケストレータ+コントローラ AMPP デモンW構成図



# AMPP Top Menu

The screenshot displays the AMPP (Automated Management and Provisioning Platform) interface. At the top, the browser address bar shows the URL `jenkins.sdn48.net:8000/html/index.html`. The main header features the AMPP logo and a navigation menu with the following items: Terminal, Monitoring, Warning, Config, Maintenance, ServiceOrder, Traffic-Ctr, Traceroute, and Probe. The central area is titled '>>Topology Viewer' and contains a network diagram. The diagram illustrates three data centers: 大手町DC (Otemachi DC), 田町DC (Tamachi DC), and 幕張DC (Makuhari DC). 大手町DC is highlighted in blue and contains nodes g01f.note, g02.note, a01f.note, a02.note, m01c.note, and m02c.note. 田町DC is highlighted in green and contains nodes tenant\_ug100, tenant\_ug300, booth\_1, ex2200.tama, and m01c.tama. 幕張DC is highlighted in red and contains nodes tenant\_ug200, tenant\_ug400, booth\_2, m01c.maku, ex2200.maku, p01c.maku, and p02c.maku. Core nodes p01c.note, p02c.note, p01c.tama, p02c.tama, p01c.maku, and p02c.maku are also visible. Two buttons, 'Save' and 'Load', are located on the right side of the topology viewer. The interface also includes a search bar and an 'Information:' section at the bottom left.

# AMPPによるサービスオーダイメージ

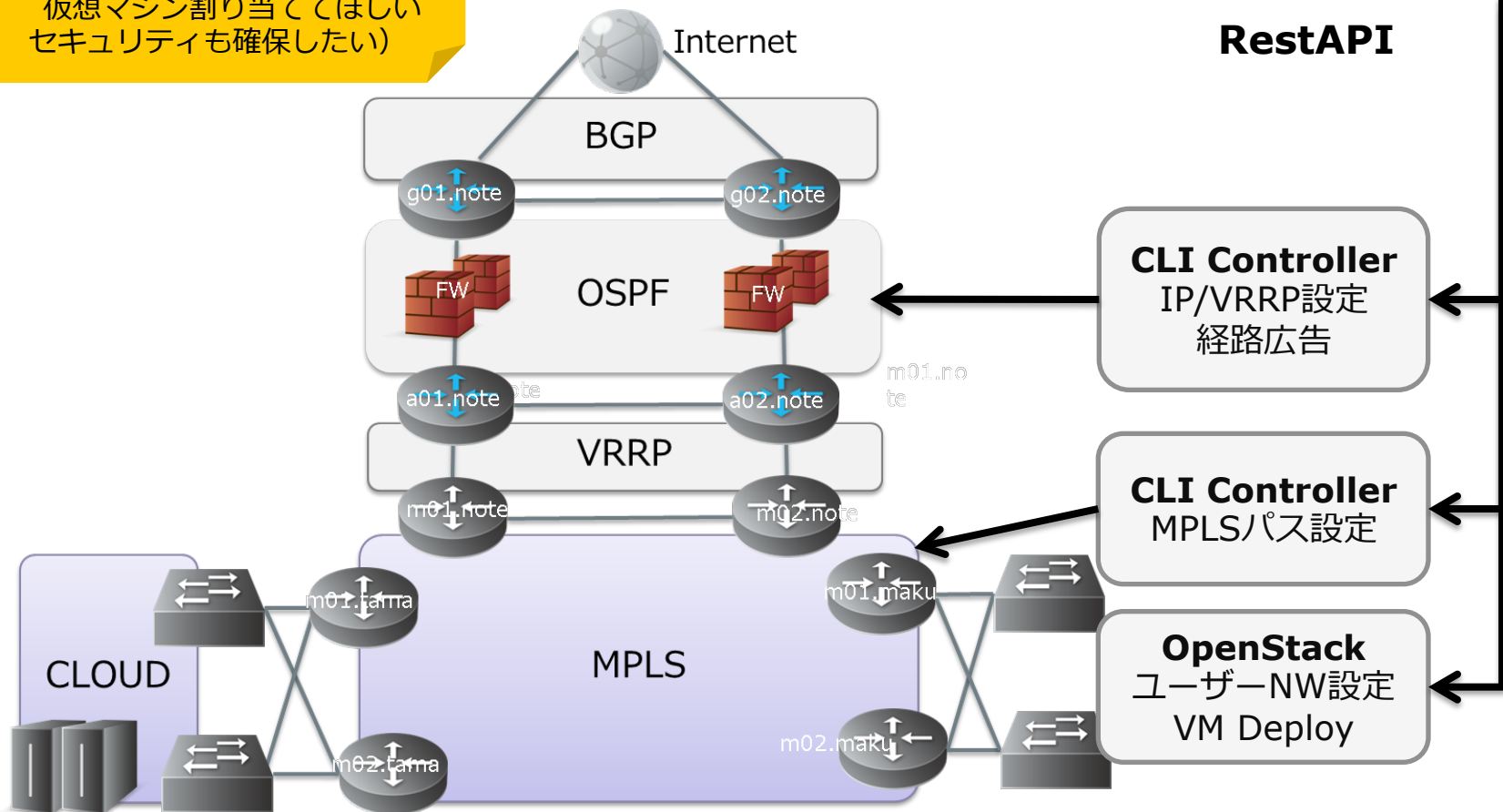
## 開通オーダー

XX担当専用の検証NWがほしい!!  
Internetに繋がって、  
仮想マシン割り当てしてほしい  
セキュリティも確保したい)

GUI

NW  
Orchestrator

RestAPI



# 開通・廃止 設定GUI

jenkins.sdn48.net:8000/html/index.html

## AMPP

ServiceOrder

Search:

NewOrder
 UserGroup

IP-Routing
 MPLS-Path

DC-Tenant
 Top Menu

Information:

### >>Topology Viewer

#### UserGroup Information

id	UserGroup	VLAN	VRRP IP Address:	a01j.note IP Address:	a01j.note Priority	a02j.note IP Address:	a02j.note Priority
1	UG100	100	202.4.245.161	202.4.245.162/29	100	202.4.245.163/29	200
2	UG200	200	202.4.245.169				
3	UG300	300	202.4.245.177				
4	UG400	400	202.4.245.185				

#### IP Backbone Setting

Please enter the required information in the following items.

UG Information

UG100

a01j.note IP Address:  
202.4.245.162/29

a01j.note VRRP Priority  
100

a02j.note IP Address:  
202.4.245.163/29

a02j.note VRRP Priority  
200

VRRP Address:  
202.4.245.161

OK Cancel

Save

Load

# AMPPデモ 保守

## NWリソース監視

CPU/メモリ情報  
MPLSパス  
トポロジー可視化

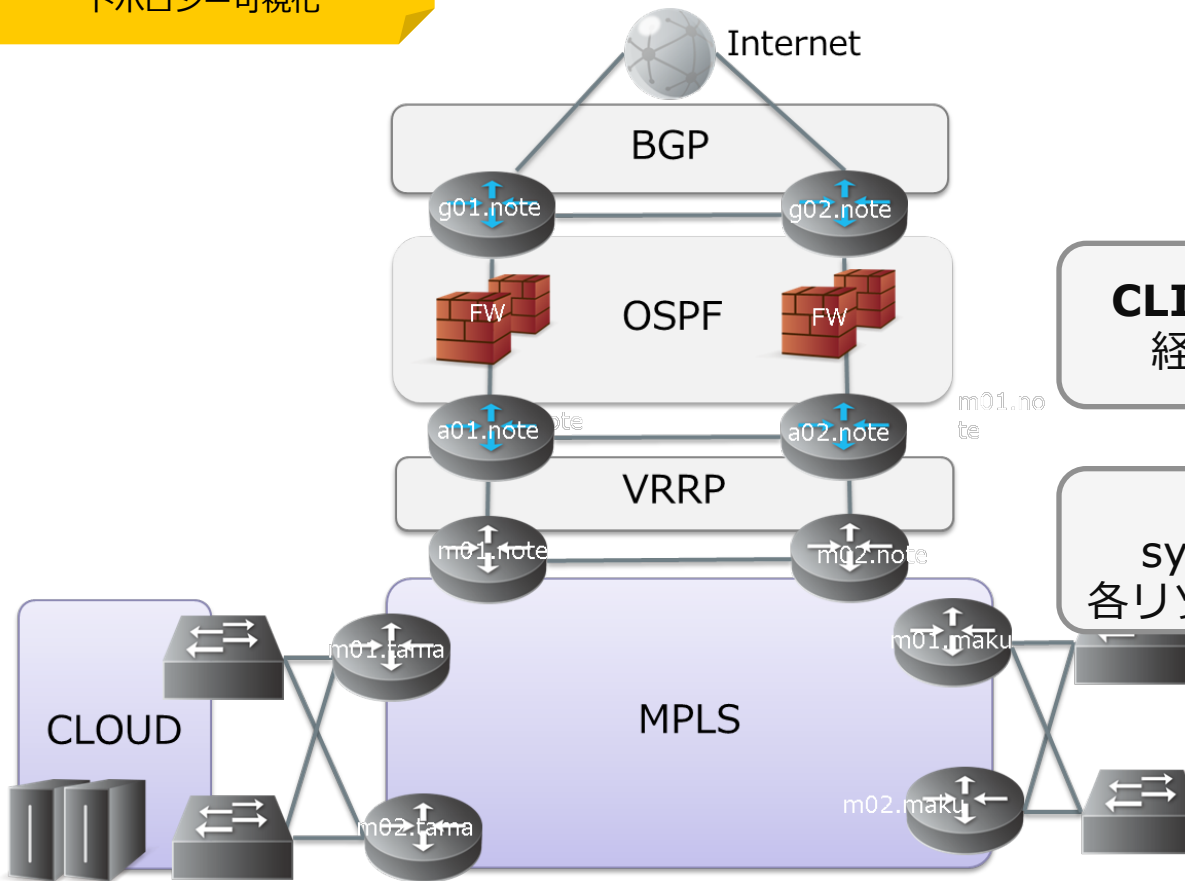
GUI

NW  
Orchestra  
tor

RestAPI

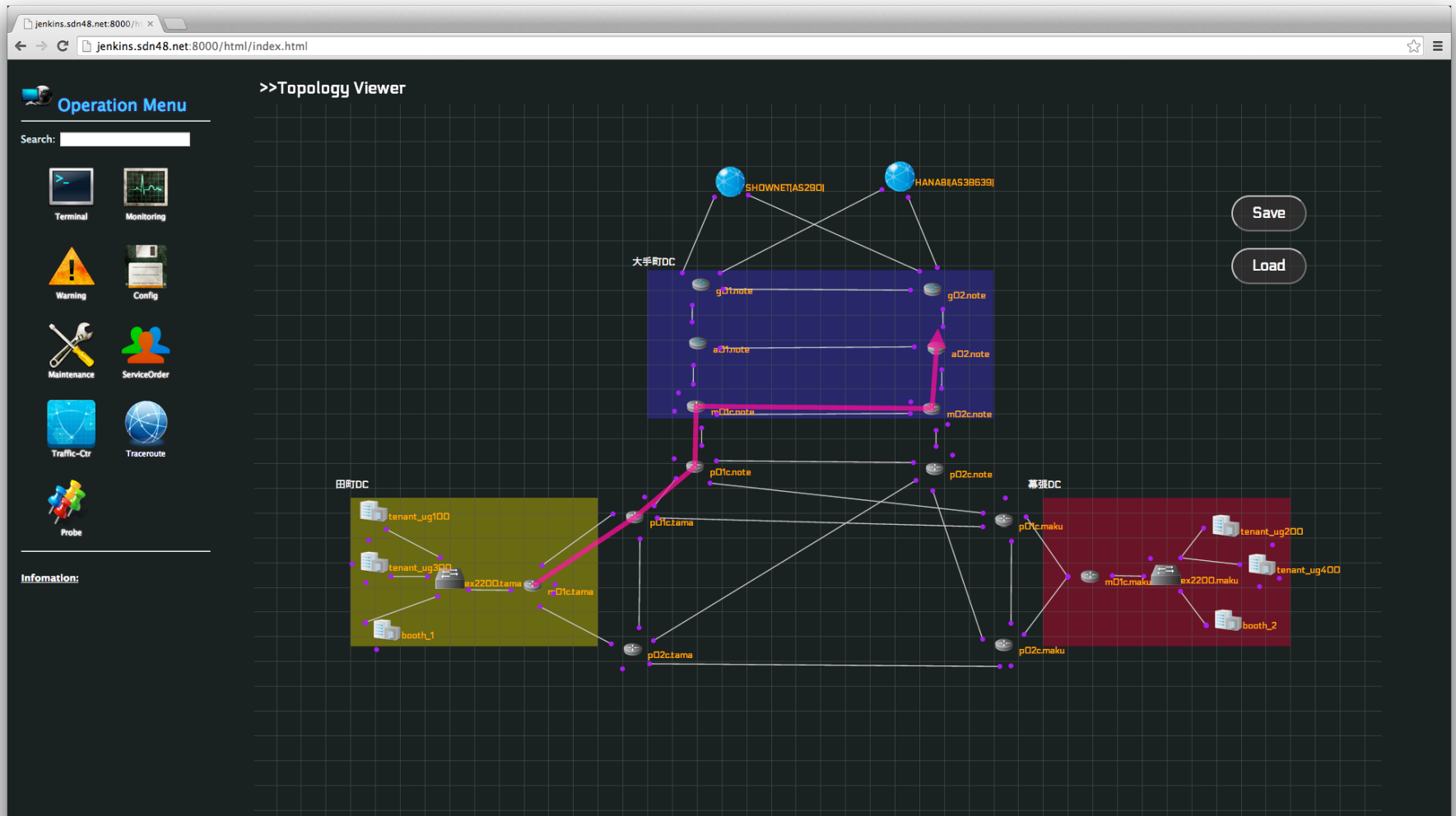
CLI Controller  
経路情報取得

NMS  
syslog/SNMP  
各リソース情報取得





# 通信の確認



# 監視画面

AMPP

>>Topology Viewer

Top Menu

Search: \_\_\_\_\_

Terminal Monitoring

Warning Config

Maintenance ServiceOrder

Traffic-Ctr Traceroute

Probe

Information:

SHOWNETIAS2901 HANABI(A538639)

大手町DC

g01.note

g01.note

CPU Graph

g01.note.sdn48.net - Load Average

processes in the run queue

14:00 16:00 18:00 20:00 22:00 00:00 02:00 04:00 06:00 08:00 10:00 12:00

1 Minute Average Current: 0.22

Save Load

田町DC

tenant\_ug100

tenant\_ug300

ex2200.tama

m01c.tama

booth\_1

boozc.tama

Memory Graph

g01.note.sdn48.net - Memory Usage

16 G

14 G

12 G

10 G

8 G

6 G

4 G

2 G

0

14:00 16:00 18:00 20:00 22:00 00:00 02:00 04:00 06:00 08:00 10:00 12:00

Free Current: 7.14 G Average: 7.48 G Maximum: 7.83 G

Swap Current: 8.59 G Average: 8.59 G Maximum: 8.59 G

OK

m01c.maku

ex2200.maku

tenant\_ug200

tenant\_ug400

booth\_2

# AMPPを開発して得た課題

## ■ AMPPで実現できた事

- サービス横断で設定自動化を実現  
Routing, MPLS-PATH, Cloud
- OpenFlow, 既存NW機器の双方のメリットを活かしたNW制御

## ■ 今後の課題

- 既存機器でもAPI対応してきたが機能が限定的
- 本当の意味でマルチベンダ対応なControllerはあまりない
  - 特定ベンダであればある程度安定して使えるが
- 商用製品との連携
  - マルチベンダのバージョンアップ、新機能に対応していくにはすべて内製開発では限界がある

定常的に利用し実用に耐えうる自動化SDN基盤の  
確立を進める

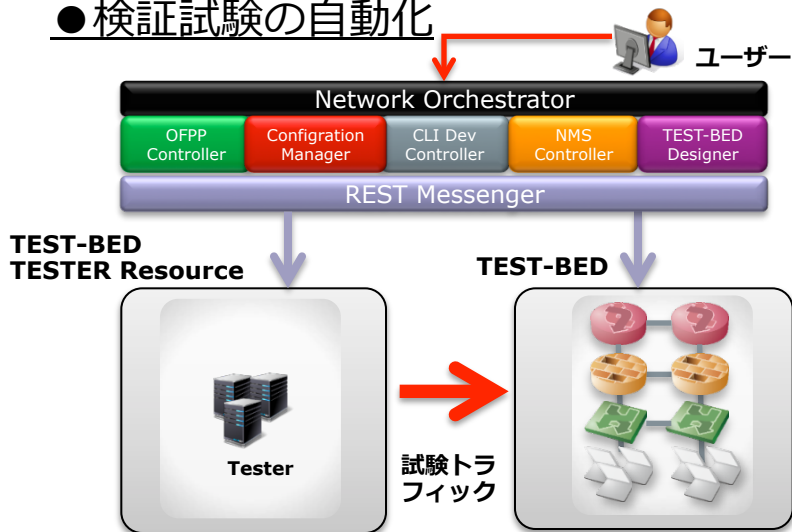
# 将来的な取り組みについて

# 検証網におけるネットワークDevOpsツール利用イメージ

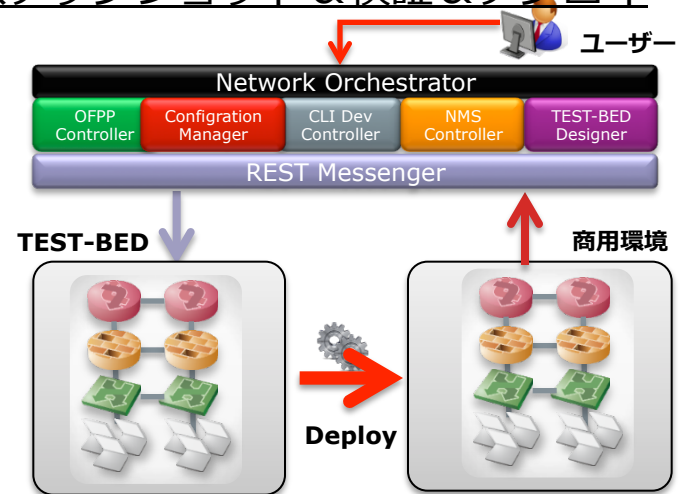
## ● 検証環境のデプロイ



## ● 検証試験の自動化



## ● スナップショット&検証&デプロイ



## 更なる活用へ

### ■ お客さまNW導入試験

- 新NWサービスのお客さま導入時の検証
- 検証環境設定を保持し、実環境への展開・実環境での故障再現試験、次回検証時に再利用
- Test as a Service



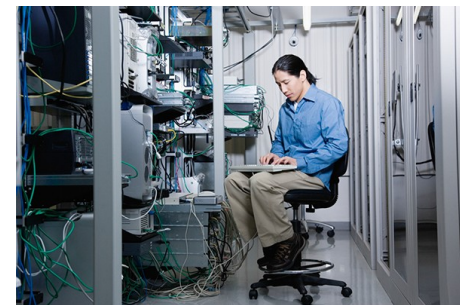
### ■ サービス運用手順試験

- 新NWサービスの運用手順(開通・監視・故障切分)作成・検証
- 設定ツールとして実運用に切り出し



### ■ 設備工事検証

- 設備更改、OSバージョンアップ実施時の検証・移行手順策定
- 新規サービス立ち上げ時検証

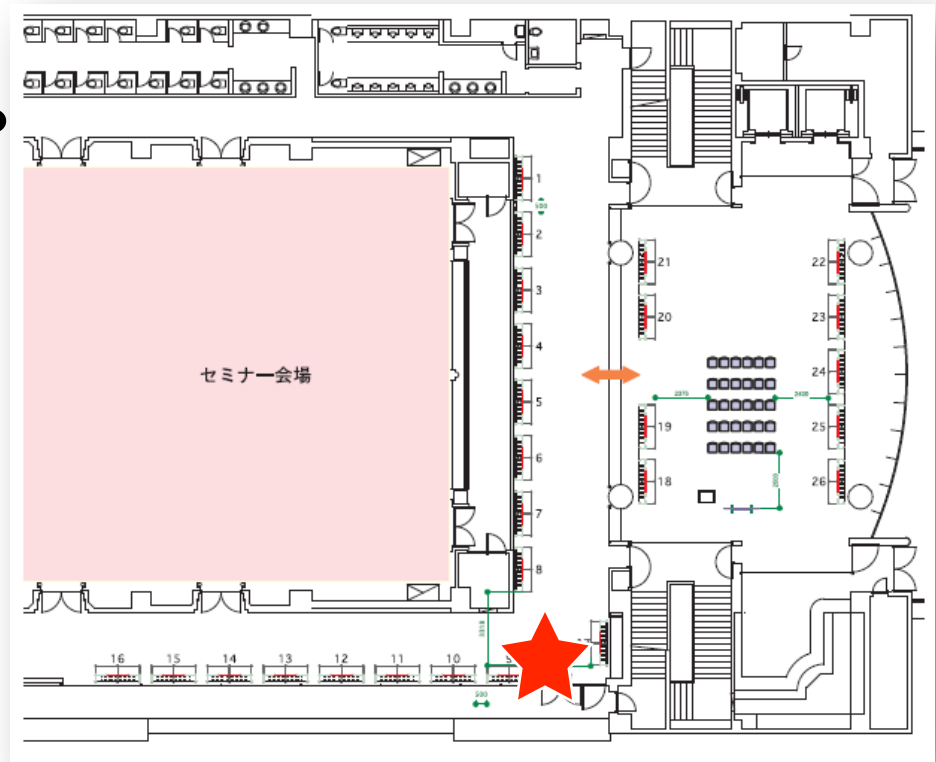


## SDNによるNetwork DevOpsの実現へ

# ご清聴 ありがとうございました。

本日、12:45より  
NTTコミュニケーションズにおける  
「**Transport SDNの取り組み**」  
をお話致します。

展示エリアでもTransport SDN  
のデモをご覧になれます。  
ぜひ展示エリアにお越しください。



★ 印の場所に展示しています。