JANOG57 Meeting in Osaka

構成管理はどこまでやるべきか？

ネットワーク運用の未来を見据えた検討と実践

IIJ
Internet Initiative Japan

2026年2月12日

株式会社インターネットイニシアティブ

- 背景/目的
- これまでのIIJにおける構成管理と課題
- 理想のネットワーク運用とそれを実現するための構成管理の在り方
- Peering Managerを用いた構成管理PoC
- まとめと今後の展望
- ディスカッション

竹﨑　友哉
（Tomoya Takezaki）
ネットワーク技術部　ネットワーク技術1課
takez@iij.ad.jp

岸　祐斗
（Yuto Kishi）
ネットワーク技術部　企画開発課
y-kishi@iij.ad.jp

竹﨑 友哉(Tomoya ”takez” Takezaki)

ネットワークサービス事業本部 基盤エンジニアリング本部 ネットワーク技術部 ネットワーク技術1課
(NSU-INF-NE-NE1)

現在の業務

IP backboneの設計・構築・運用
AS2497 Peering coordinator

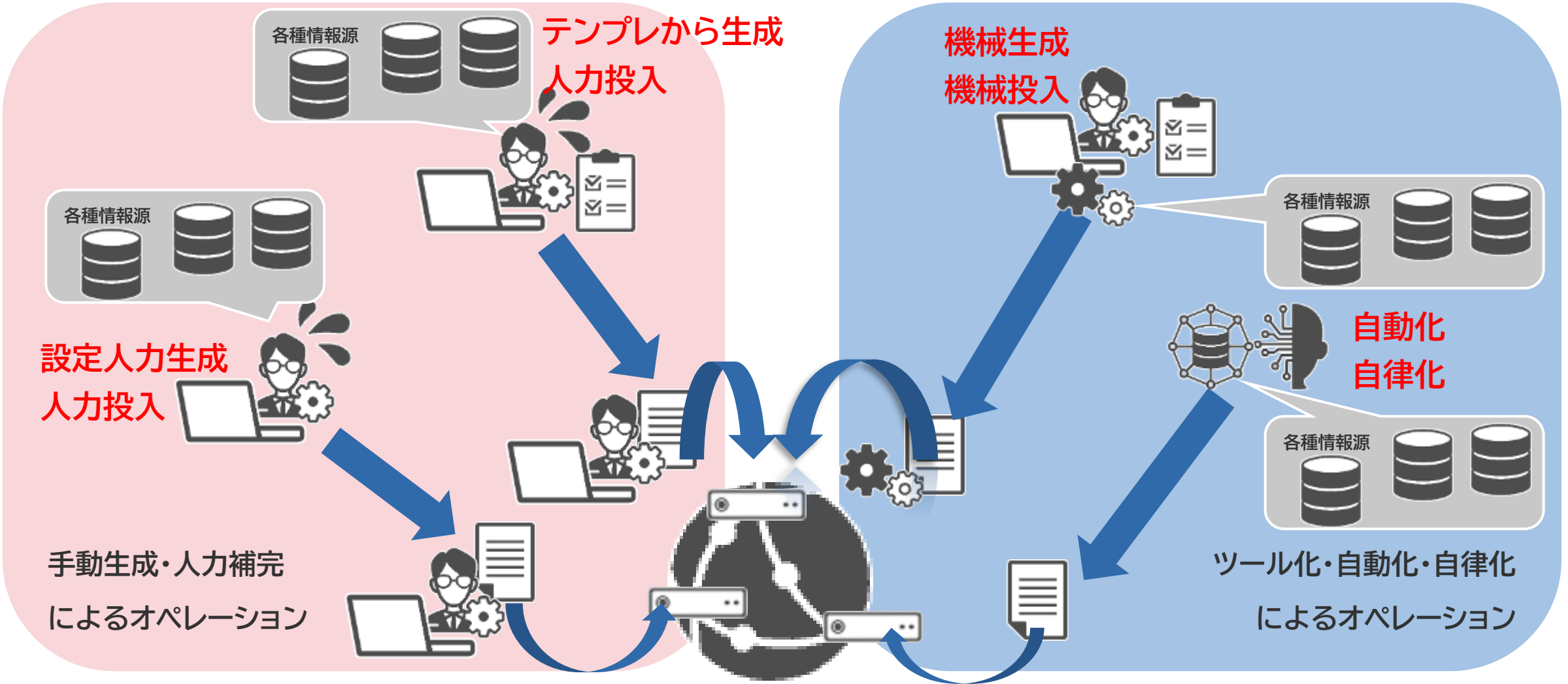
経歴

IIJ/NTT Com/JPNAP 400GbE共同検証
400G-ZR導入

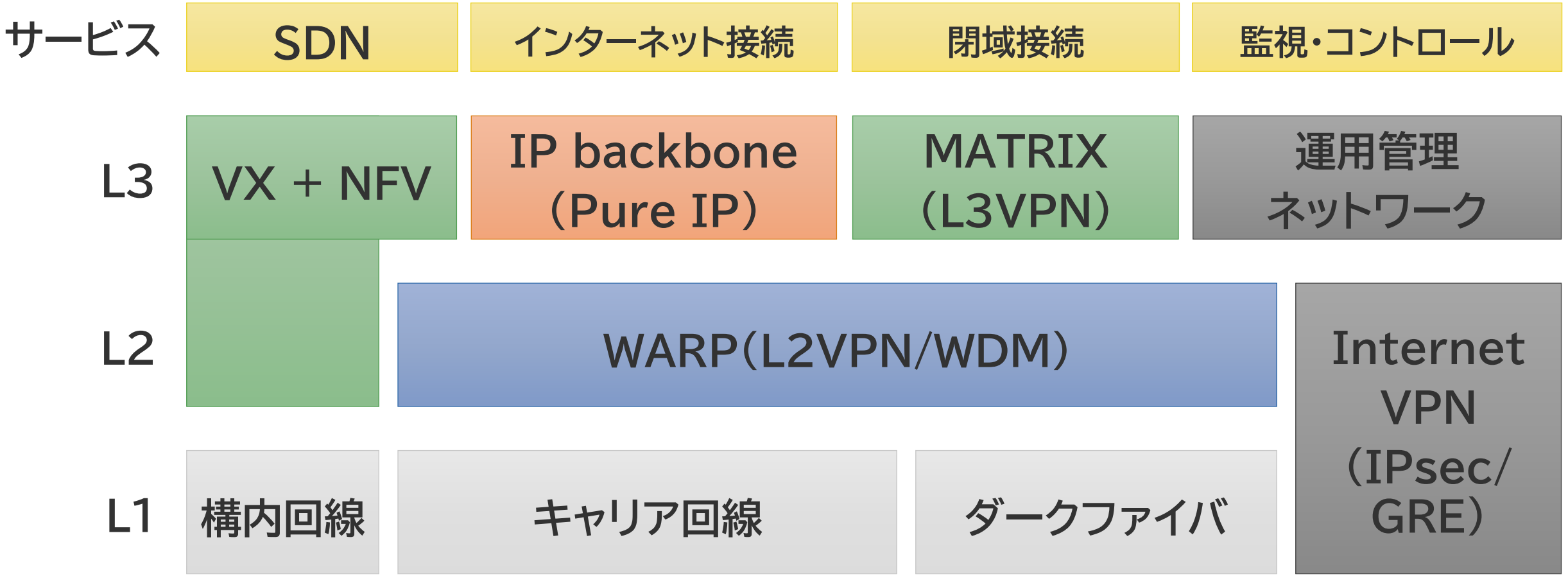
趣味

野球観戦
マンホール探索

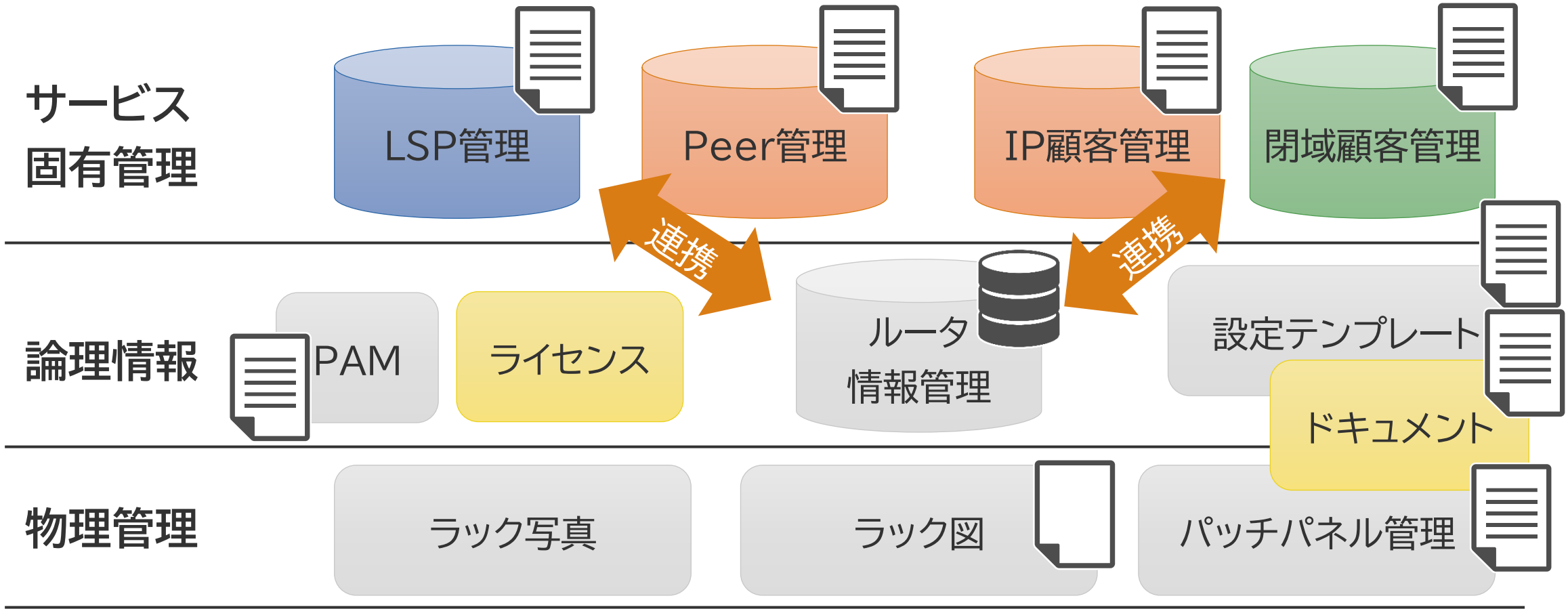
ネットワーク要件の多様化により手動運用は限界に達しており、自動化に向けて構成情報を適切なデータ形式で管理する必要がある。



要件に合わせた4つのネットワークを構築・運用
WARP, IP backboneは6カ国に展開

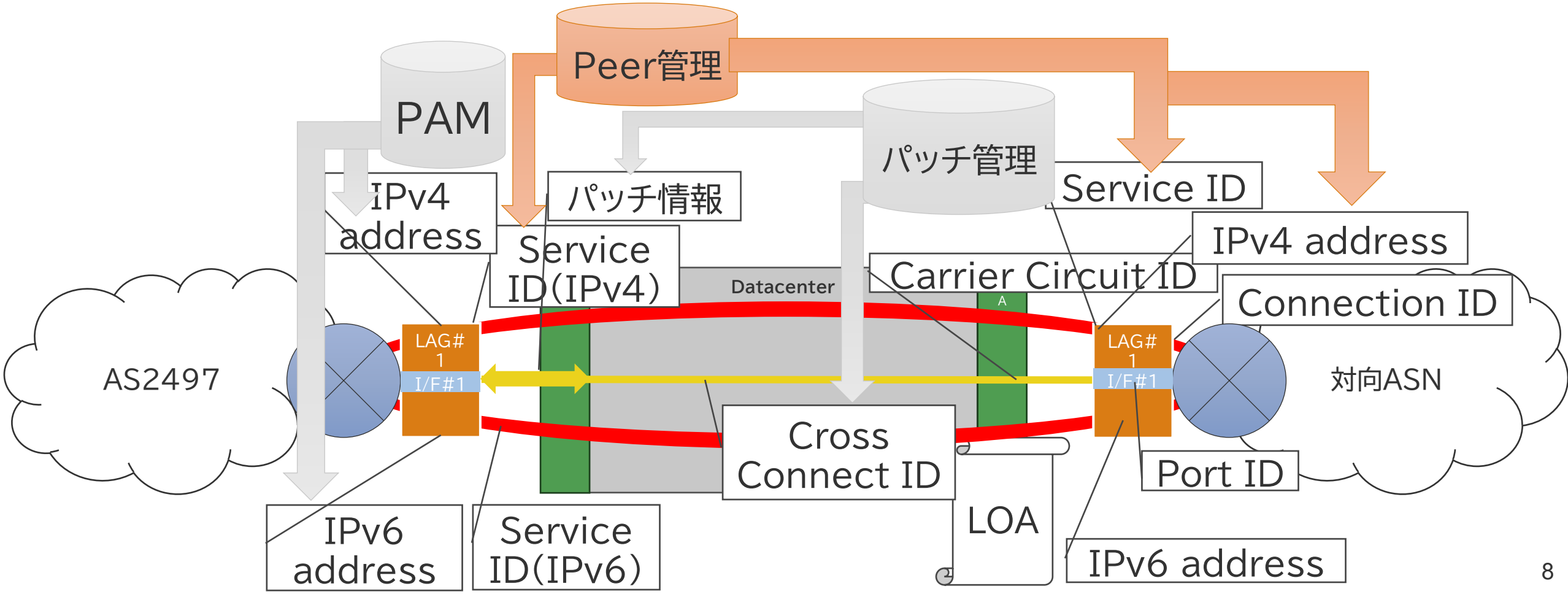


- **漠然と多くの情報を取得・管理**
  - 運用上大きな問題とはなっていないが、しかし…
- **一部は連携し動作しているが、各種情報を人力で相互に照らし合わせて妥当性・整合性を担保している場合もある。**

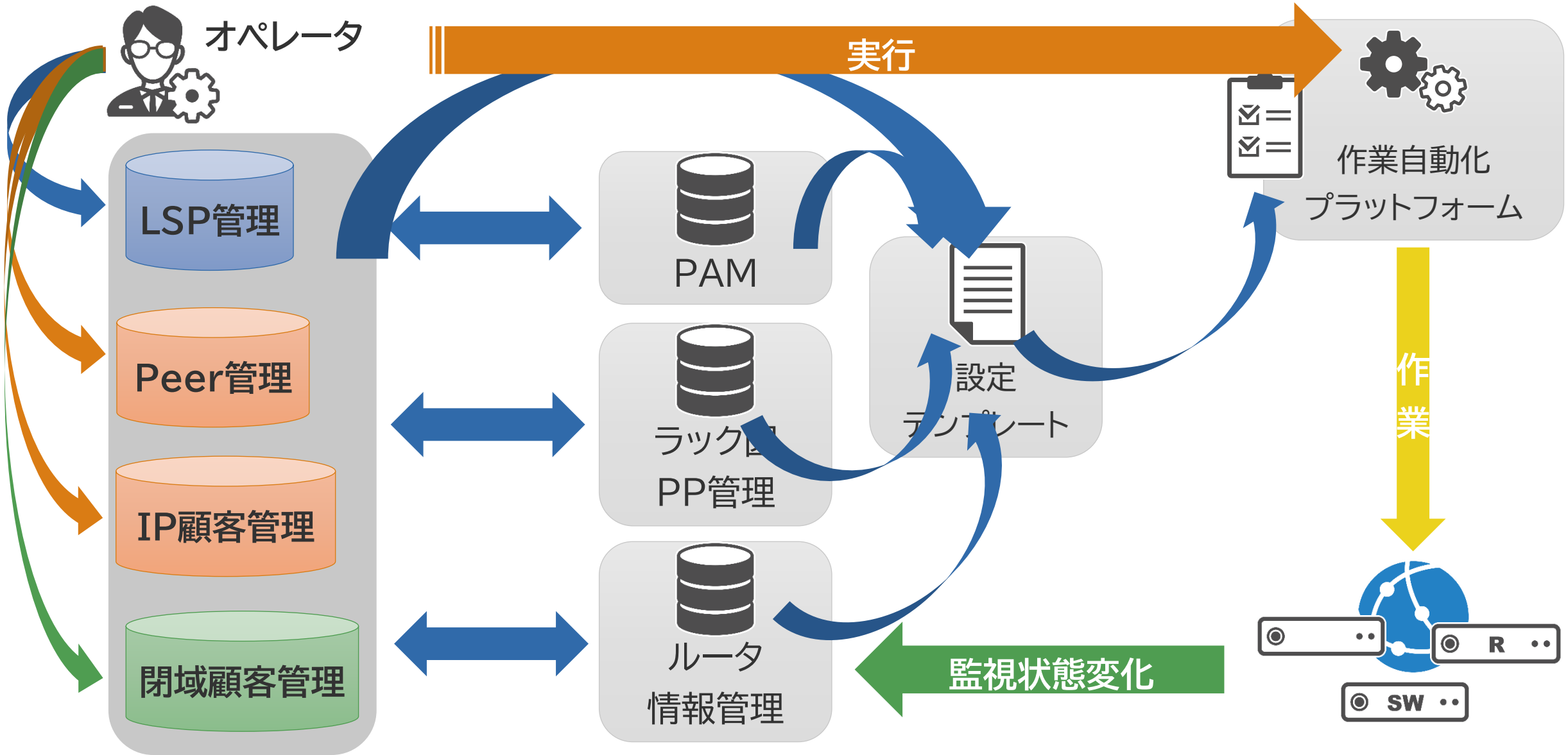


BGP 1接続に対して多い場合は以下の管理情報が発生する。

必要なデータは人が加工すれば扱えるものの、システムからは取り出しにくく、システム間連携やデータの活用が進めにくい。



理想は人手を介在せず、システム間が連携してネットワーク運用が完結する世界



統合データベース管理

1つのデータベースで全NW領域の情報を管理

#	コアルータ		接続先		事業者/顧客
	ホスト名	Port	ホスト名	Port	
1	コアルータA	1	ピアルータA	1	事業者A
2	コアルータA	1	ピアルータA	2	事業者B
3	コアルータA	2	ピアルータB	1	事業者C
4	コアルータA	2	ピアルータB	2	事業者D
5	コアルータA	3	エッジルータA	1	顧客A
6	コアルータA	3	エッジルータA	2	顧客B
7	コアルータA	4	コアルータB	2	-
8	コアルータB	1	ピアルータC	1	事業者E
9	コアルータB	2	コアルータA	4	-

分散データベース管理

NW領域ごとにデータベースを分けて情報を管理

#	コアルータ		接続先
	ホスト名	Port	ホスト名
1	コアルータA	1	ピアルータA
2	コアル		
3	コアル		
4	コアル		
5	コアル		
6	コアル		

#	ピアルータ		事業者
	ホスト名	Port	
1	ピアルータA	1	事業者A
2	ピアルータA	2	事業者B
3	ピア		
4	ピア		
5	ピア		

#	エッジルータ		顧客
	ホスト名	Port	
1	エッジルータA	1	顧客A
2	エッジルータA	2	顧客B

- 下表の観点から「**分散データベース管理 ＋ API連携**」方式を選定
- NW領域ごとに独立してデータを管理し、必要な時に情報を紐づける

比較観点	統合データベース管理	分散データベース管理
柔軟性	領域ごとの差異を吸収しづらい	領域ごとに最適なデータモデルを採用可能
変更影響範囲	全体へ波及し、大規模改修が必要	局所化され改修が容易
開発スピード	大規模化し、追加開発が困難	迅速に機能追加・改修可能
拡張・将来性	構造が複雑化し、APIが扱いにくい	API連携が前提で、自動化との親和性が高い

統合化（全体最適）ではなく、「分散（局所最適） ＋ 連携化」により、自動化親和性とメンテナンス性の両立を目指す

- **目的**
  分散データベース管理 + API連携による、ネットワーク運用自動化の実現性
  および OSS(オープンソースソフトウェア)の実用性と開発効率を評価する。

- **実施内容**
  - APIによる外部ツール連携
  - Peering Manager(PM)を用いたピアリング情報の登録・更新
  - 当社要件との整合性確認

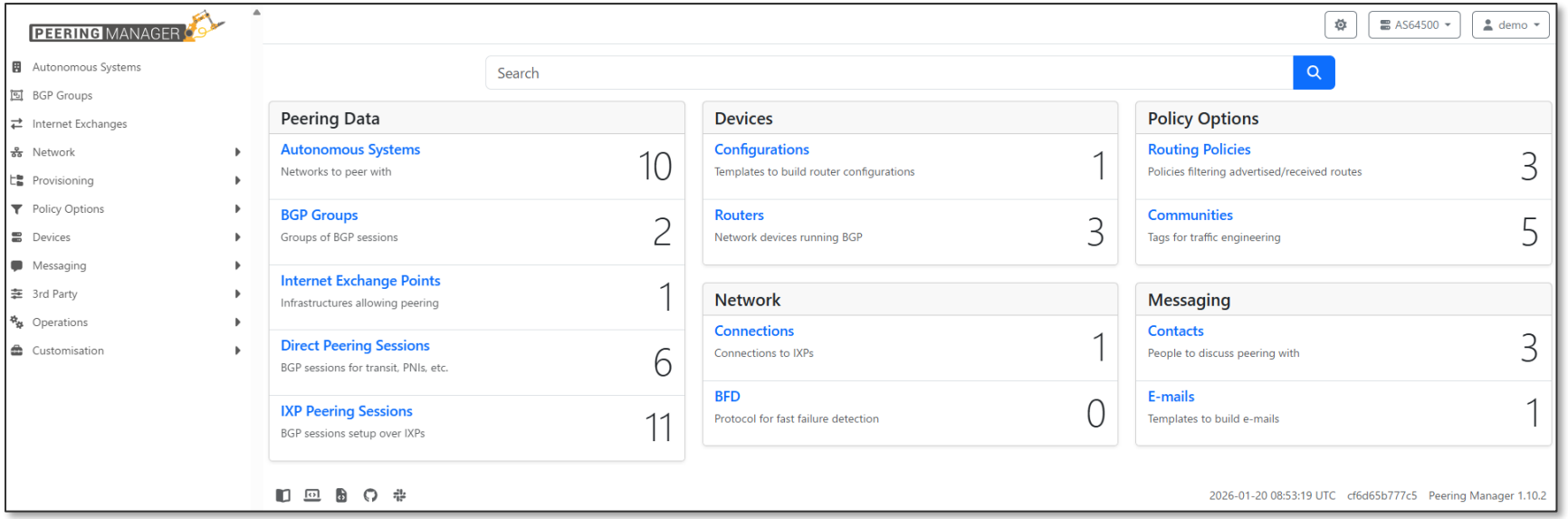
評価軸	内容
自動化親和性	API品質・外部システムとの連携性
データ管理範囲	必要なピアリング情報を十分に表現できるか
拡張性	当社要件に合わせて項目追加・変更が容易か

● 概要

- BGP セッションやピアリング情報の管理に特化した OSS
- Python/Django で実装され、Web UI と REST API を標準提供
- 構成管理データベースとして利用可能

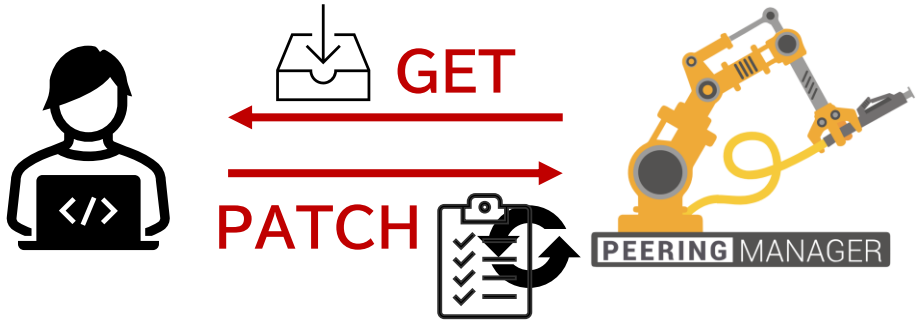
● 選定理由

- OSSのため、**無償**かつ**拡張性が高い**
- **APIが充実**しており、自動化との親和性が高い

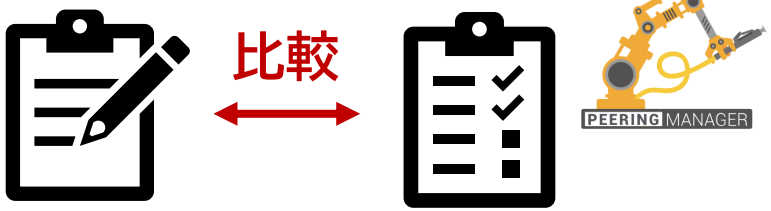


引用: https://demo.peering-manager.net/

1. API による情報取得・更新



2.管理すべき情報の整理



既存管理情報　PM標準管理項目

3. 管理項目の追加拡張

不足項目

必要な全ての情報管理

- データ管理 ＋ API連携により**ネットワーク運用自動化が実現可能**
- ピア領域において、PM活用により、開発スピードやコストを抑えながら当社要件に適合したデータ管理が可能

評価軸	評価結果
自動化親和性	API経由で情報取得・更新が可能であり、ネットワーク運用自動化の実現が可能
データ管理範囲	PM標準提供機能だけでは、不足している管理項目が存在するため、要件に合わせた拡張が必要
拡張性	Python/Django ベースであり、一般的な開発スキルで容易にカスタマイズ可能

取り出したい形式で、API経由での情報取得・更新が可能であり、ネットワーク運用自動化が実現可能。

API経由での情報取得

Pythonツール


```
回線IDを入力してください: TST001
=== Circuit Information ===
Service Code          : TST001
Status                : enabled
IXP                   : TEST1
IPv6 Address          : 2001:db8:1::1/64
IPv4 Address          : 192.0.2.1/24
Router                : PR1
Interface             : LAG1
Physical Interfaces   : ['IF-0', 'IF-1', 'IF-2']
============================
```

API経由での情報更新

Pythonツール

Service code	Internet exchange point	Status	IPv6	IPv4	Router	Interface	Physical interfaces
TST001	TEST1	Enabled	2001:db8:1::1/64	192.0.2.1/24	PR1	LAG1	IF-0, IF-1, IF-2
TST002	TEST1	Enabled	2001:db8:2::1/64	192.0.3.1/24	PR1	LAG2	IF-3, IF-4
TST003	TEST2	Enabled	2001:db8:3::1/64	192.0.4.1/24	PR2	—	IF-5

```
回線IDを入力してください: TST001
新しいIPv6アドレスを入力してください: 2001:db8:100::1/64
Updated Connection TST001 -> IPv6=2001:db8:100::1/64
```

Service code	Internet exchange point	Status	IPv6	IPv4	Router	Interface	Physical interfaces
TST001	TEST1	Enabled	2001:db8:100::1/64	192.0.2.1/24	PR1	LAG1	IF-0, IF-1, IF-2
TST002	TEST1	Enabled	2001:db8:2::1/64	192.0.3.1/24	PR1	LAG2	IF-3, IF-4
TST003	TEST2	Enabled	2001:db8:3::1/64	192.0.4.1/24	PR2	—	IF-5

当社要件と PM 標準項目を突合した結果、以下の項目が不足。

不足項目	内容
回線ID	PMの自動採番ルールが当社の採番ポリシーと不一致
複数インターフェース（LAG）	単一インターフェースは登録可能だが、 LAGに紐づく複数インターフェースをまとめて管理できない
BGPパス制御用フラグ	当社独自の運用パラメータであり、標準項目に存在しない

回線IDはシステム側で整数を自動採番

単一インターフェースのみ登録可能

ID	Internet exchange point	Status	IPv6	IPv4	Router	Interface
33	TEST1	Enabled	2001:db8:1::1/64	192.0.2.1/24	PR1	LAG1
34	TEST1	Enabled	2001:db8:2::1/64	192.0.3.1/24	PR1	LAG2
35	TEST2	Enabled	2001:db8:3::1/64	192.0.4.1/24	PR2	IF-5

一般的な Python/Django スキルがあれば、当社固有の要件に合わせたデータモデル拡張が十分可能。

標準提供(Before)

ID	Internet exchange point	Status	IPv6	IPv4	Router	Interface
33	TEST1	Enabled	2001:db8:1::1/64	192.0.2.1/24	PR1	LAG1
34	TEST1	Enabled	2001:db8:2::1/64	192.0.3.1/24	PR1	LAG2
35	TEST2	Enabled	2001:db8:3::1/64	192.0.4.1/24	PR2	IF-5

物理インターフェース(リスト型)の管理項目を追加

追加拡張後(After)

Service code	Internet exchange point	Status	IPv6	IPv4	Router	Interface	Physical interfaces
TST001	TEST1	Enabled	2001:db8:100::1/64	192.0.2.1/24	PR1	LAG1	IF-0, IF-1, IF-2
TST002	TEST		1/64	192.0.3.1/24	PR1	LAG2	IF-3, IF-4
TST003	TEST		1/64	192.0.4.1/24	PR2	—	IF-5

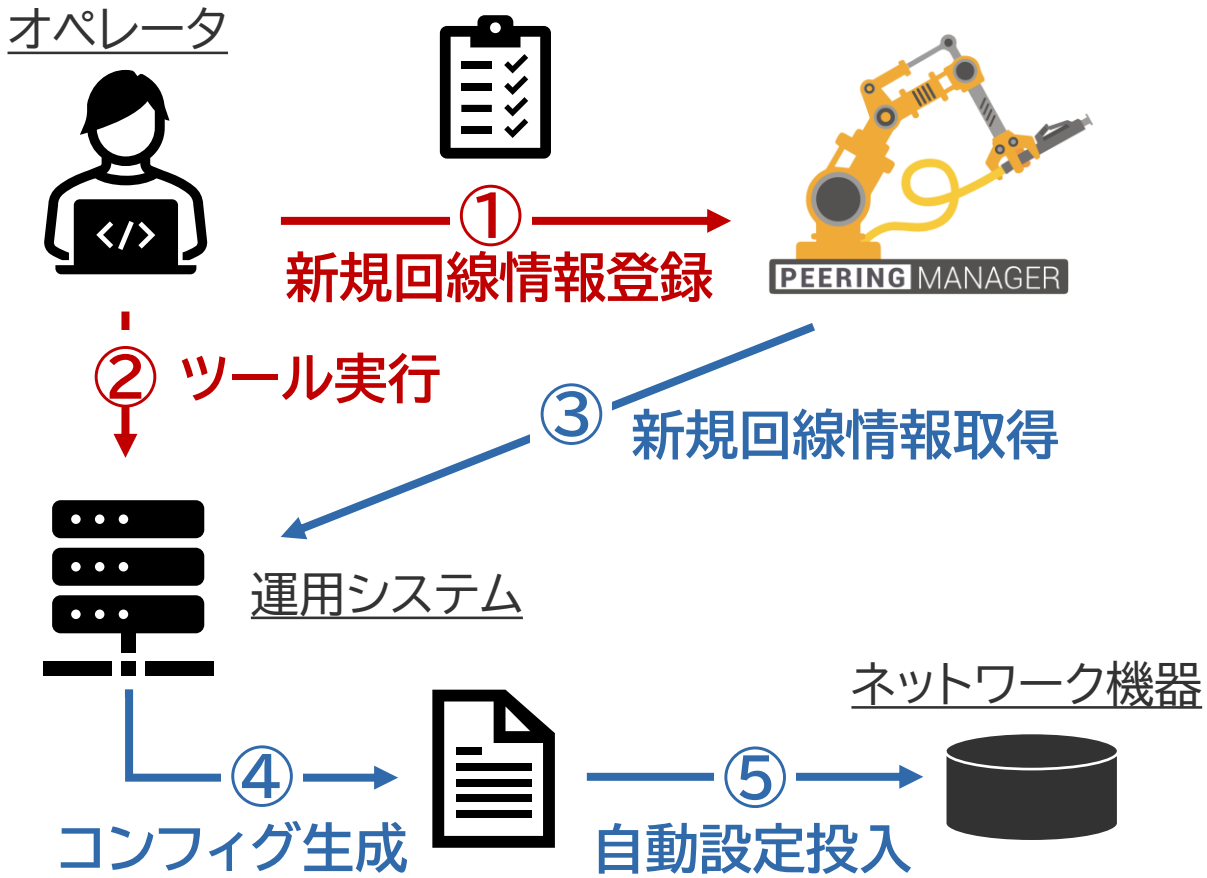
当社採番ポリシーに沿って登録可能な管理項目へ変更

回線ID

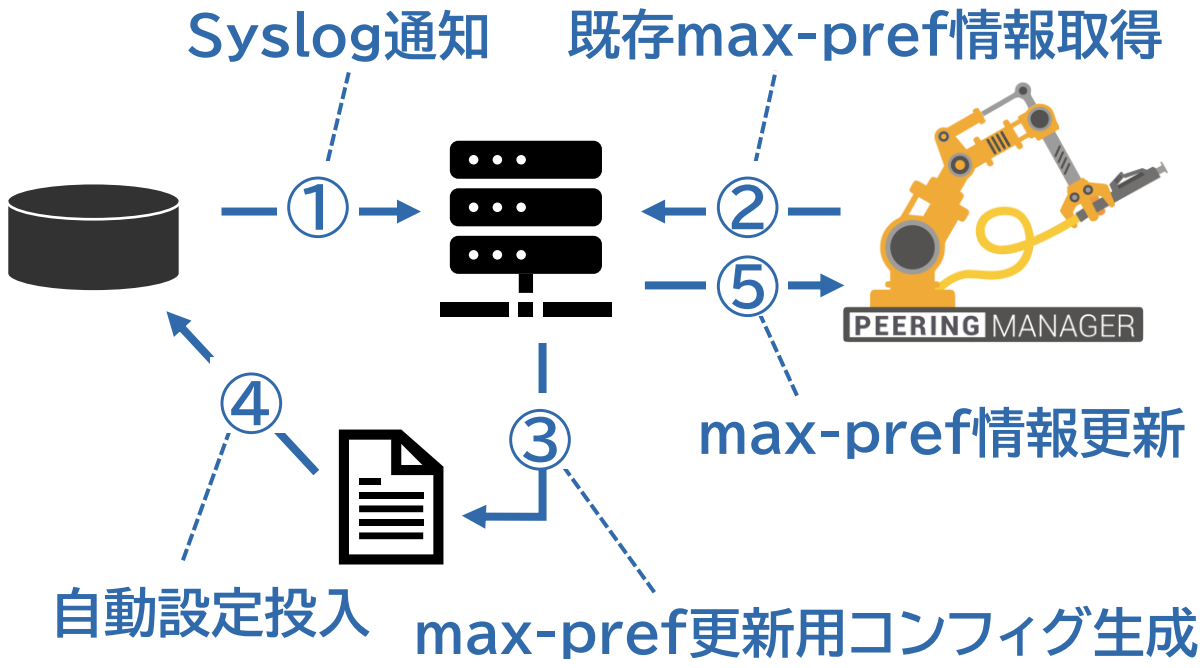
複数インターフェース

BGPセッション管理や関連作業を効率化し、業務負荷を低減させる

新規回線の自動設定



BGP maximum prefixの自動更新



凡例 → 手動 → 自動

まとめ

IIJにおける理想の構成管理手法を検討し、PoCを行った結果、
自動化親和性の高い構成管理アーキテクチャを実現できた。

今後の展望

ピア領域の情報管理システムとして、
Peering Manager を商用導入予定。

将来的には外部システムと連携して、
デリバリ・メンテナンス作業の自動化・自律化を目指す。

- OSSは継続性・安定性の面でリスクがあるため、
  長期運用する場合、内製開発/ベンダ製品利用の選択肢もある
- コストや運用負荷を考慮しながら、
  管理範囲の適切な落とし所を検討する必要がある

項目	OSS活用	内製開発	ベンダ製品
初期コスト・導入スピード	◎	△	△
自社要件適合度	○	◎	△
保守・運用負荷	△	△	◎
継続性・安定性	△	◎	◎

1. ネットワーク構成情報について、現在どの範囲まで管理できているのか？
   また、今後どこまで管理すべきなのか？

2. 構成管理のために、どのようなツールや仕組みを活用しているのか？

3. 組織として、構成管理の標準化・共通化はどの程度進んでいるのか？
   （部署ごとの管理に留まっているのか、会社全体で一元的に管理できているのか？）

4. 構成管理と運用自動化の連携は、どこまで実現できているのか？

5. 人の介在を前提としないシステムを選定・検証する際、どのようなハードルがあるか？

日本のインターネットは1992年、IIJとともにはじまりました。以来、IIJグループはネットワーク社会の基盤をつくり、技術力でその発展を支えてきました。インターネットの未来を想い、新たなイノベーションに挑戦し続けていく。それは、つねに先駆者としてインターネットの可能性を切り拓いてきたIIJの、これからも変わることのない姿勢です。IIJの真ん中のIはイニシアティブ――IIJはいつもはじまりであり、未来です。