

OCNに ネットワークコントローラ 入れてみたよ

NTTコミュニケーションズ株式会社
伊藤 良哉/高橋 春樹

自己紹介



伊藤 良哉

NTTコミュニケーションズ株式会社

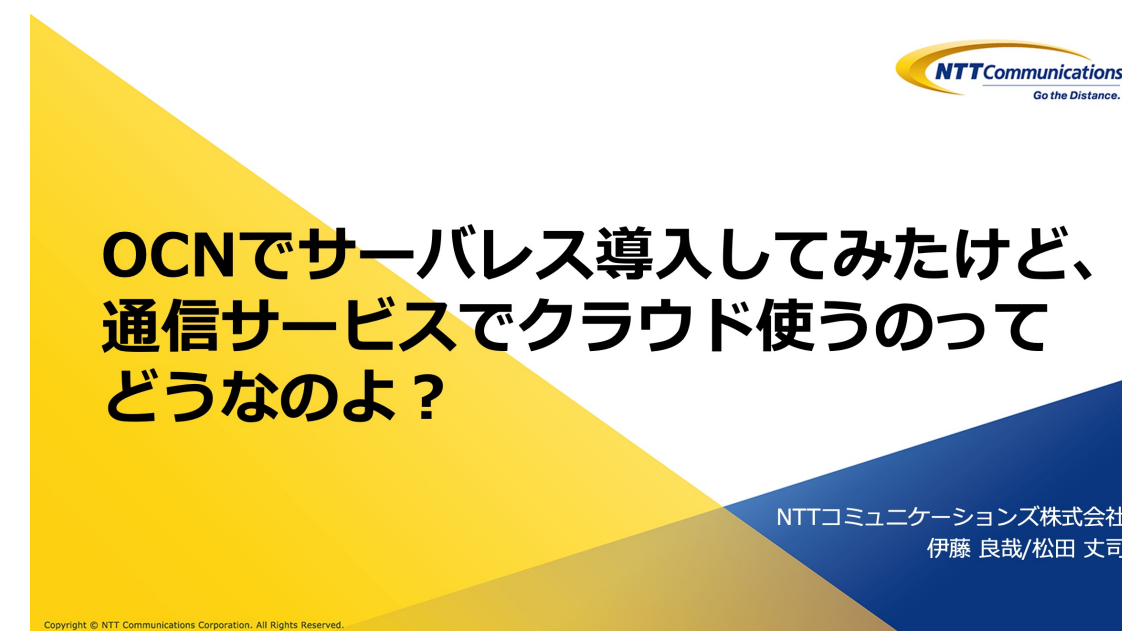
JANOG43



高橋 春樹

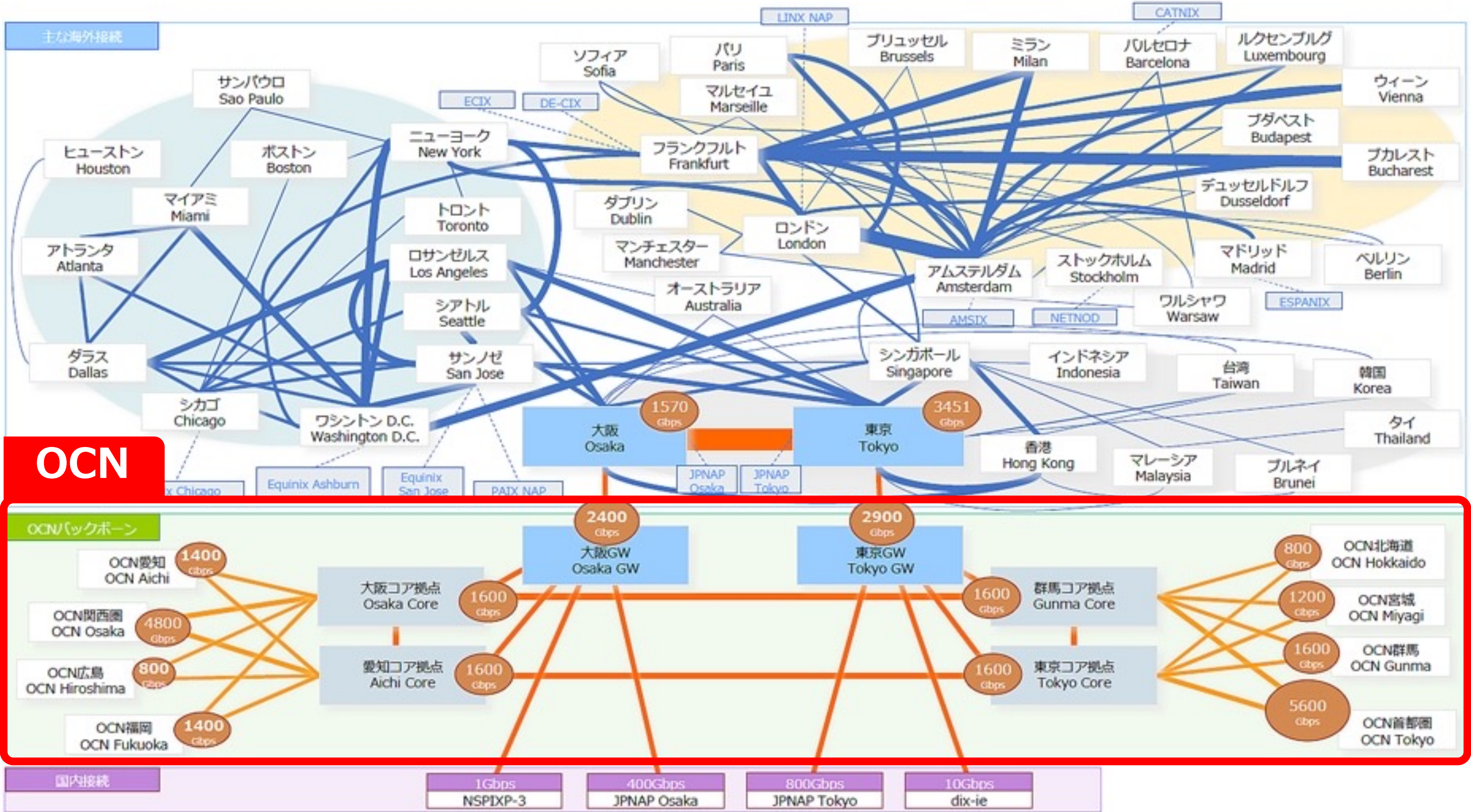
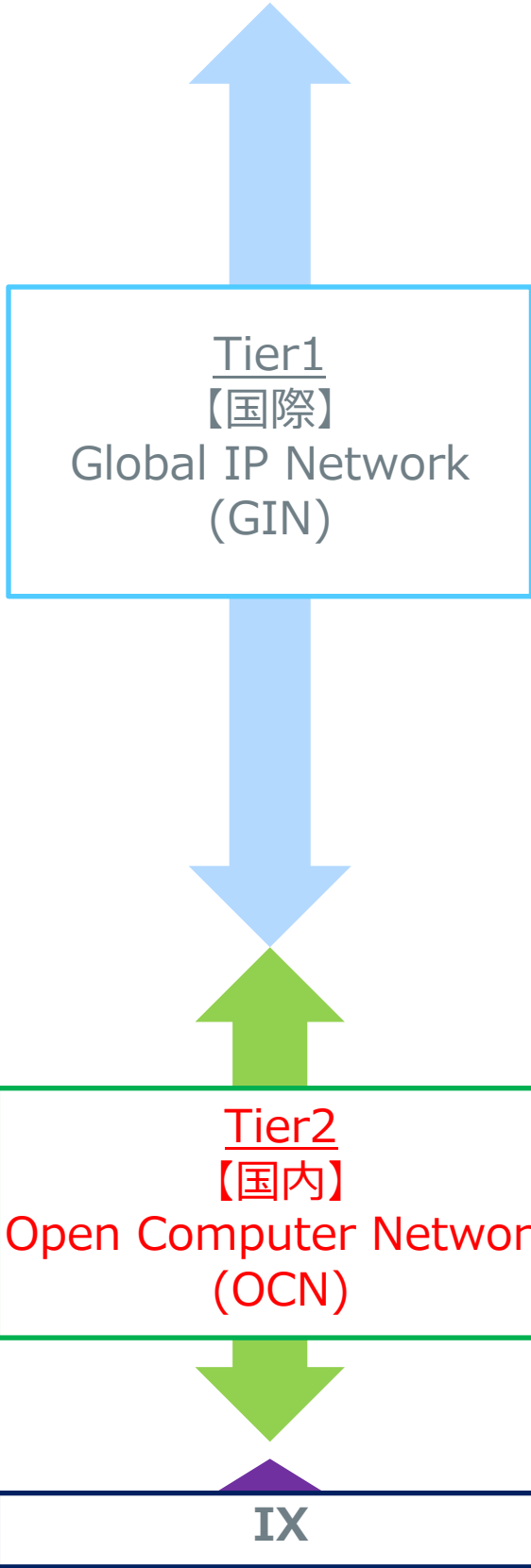
NTTコミュニケーションズ株式会社

JANOG45

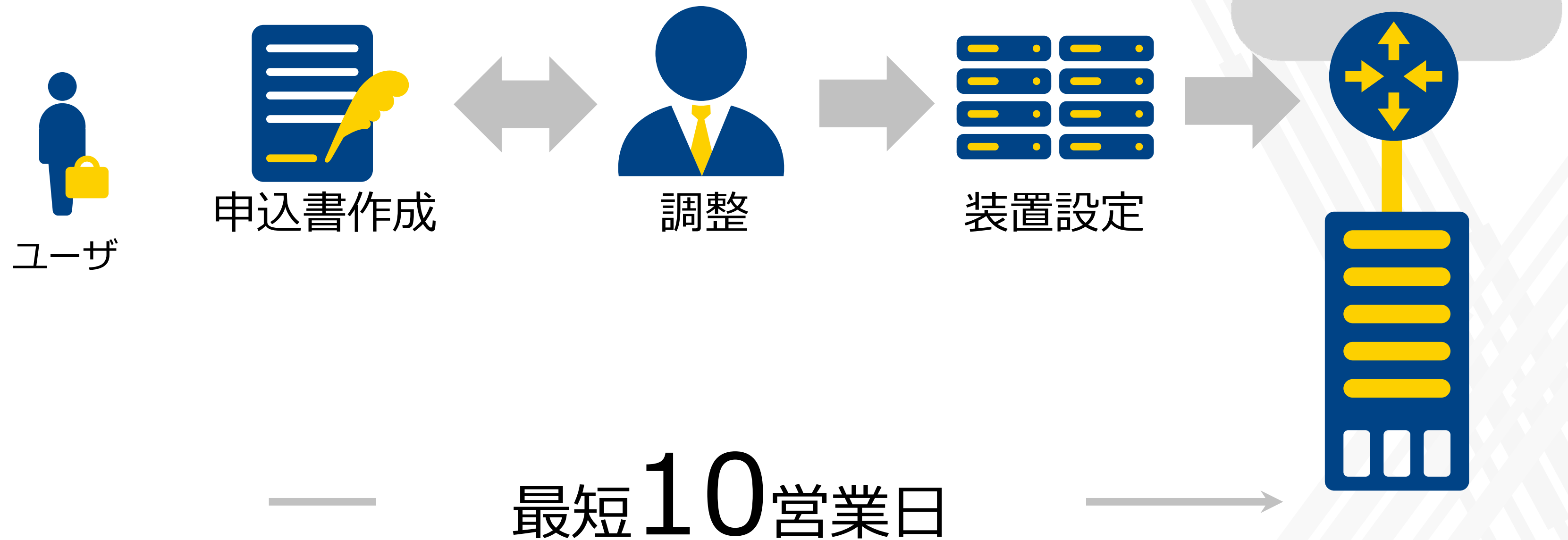


イントロダクション

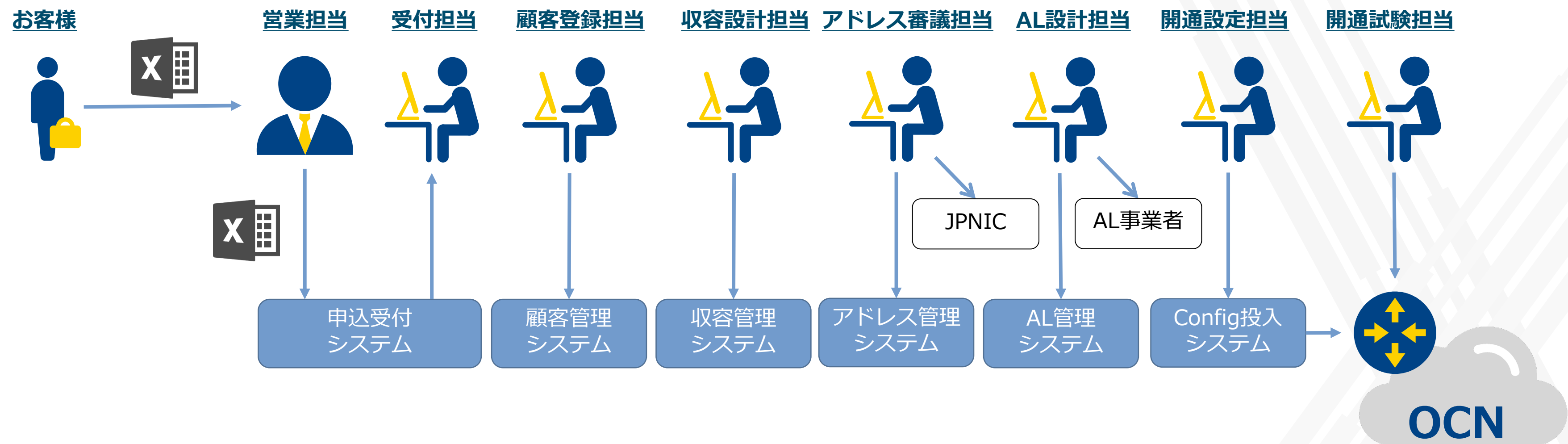
OCNネットワーク概要



お客様にネットワークを提供するまで



お客様にネットワークを提供するまで -詳細-



3つの問題点

1

複数の
システム



2

人間中心の
設計

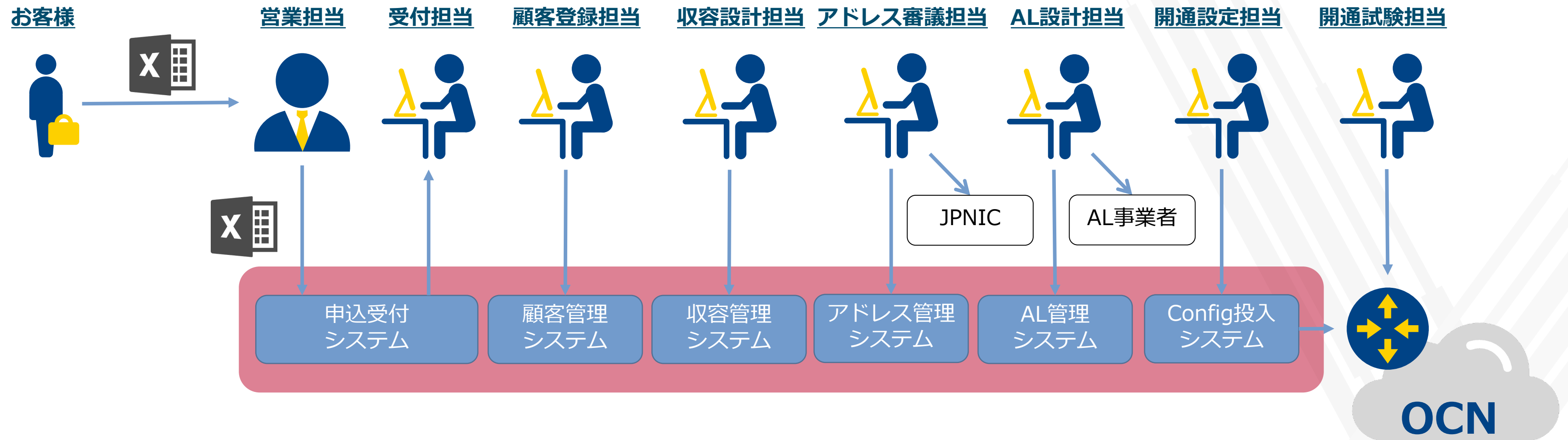


3

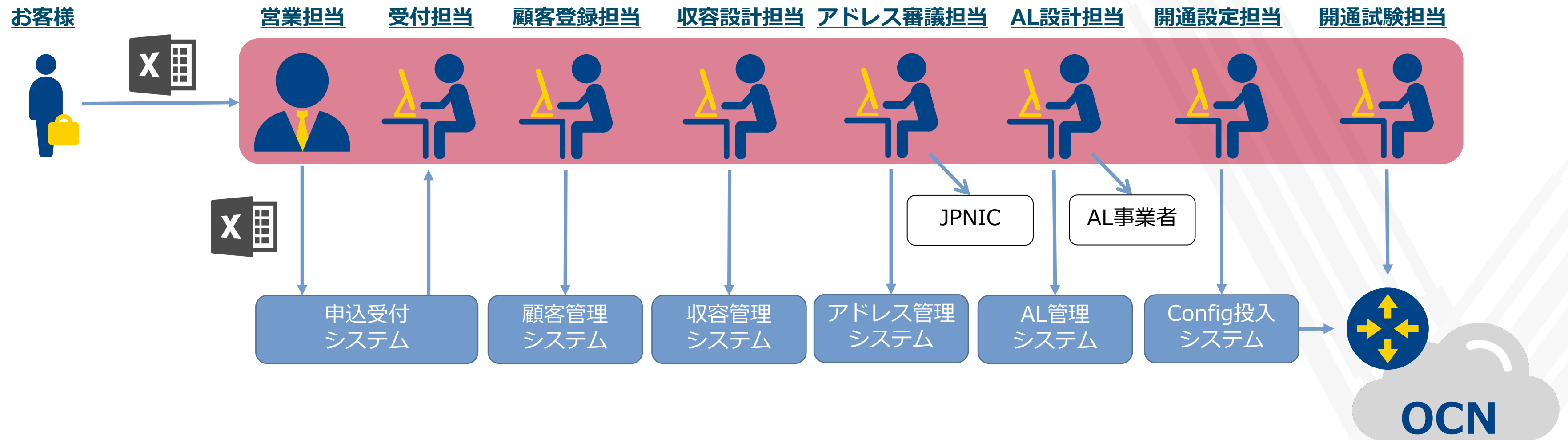
エクセル
オーダー



複数のシステム

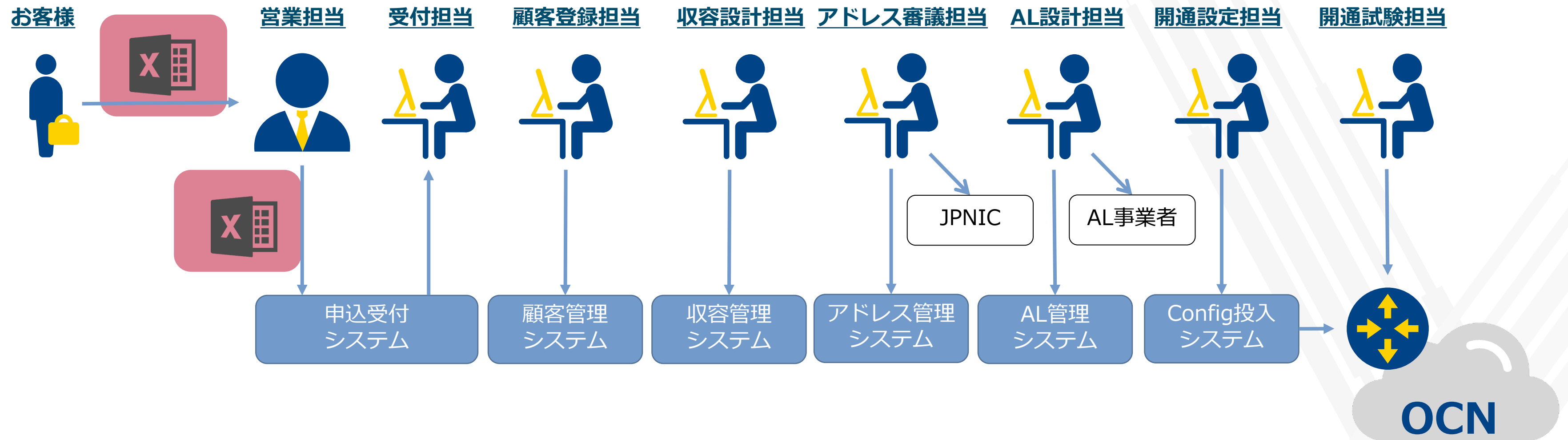


- 担当それぞれシステムを開発
- 新サービスを作るとき、手が入るシステムが膨大なためコスト増



- 人が操作することを前提としたアーキテクチャ
- 潤滑油としての人
- 納期を短縮するのは限界がある

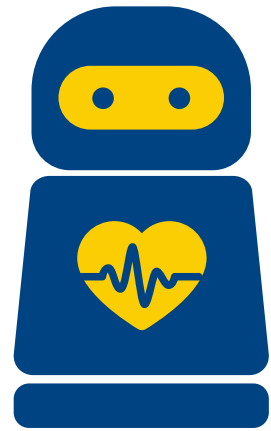
エクセルオーダー



- エクセル故に自由記述が可能
- 人がいることをいいことに手運用の嵐

どのように解決するか？

RPA



根本的解決にならない

オーケストレータを調達



ライセンス費用が高い
習熟コストもかかる

改善を諦める



実は真っ当な選択肢？



NTTCom製の オーケストラタが あるらしい

その名も

Q monus

なぜQmonusを？



調達コストがほぼ0



開発者が近い



デプロイが容易

Qmonusとは？



Qmonus

- ✓ Qmonusは、クラウドネイティブアプリケーションの開発及びデリバリ・運用を高度化するための Platform as a Service

Qmonus SDK

クラウド上でMicroserviceを
高速に開発するためのSDK

Qmonus Value Stream

DevOps環境の柔軟な自動構築、アプリケーションのデリバリ、テスト、運用の自動化を実現する宣言的CI/CDプラットフォーム

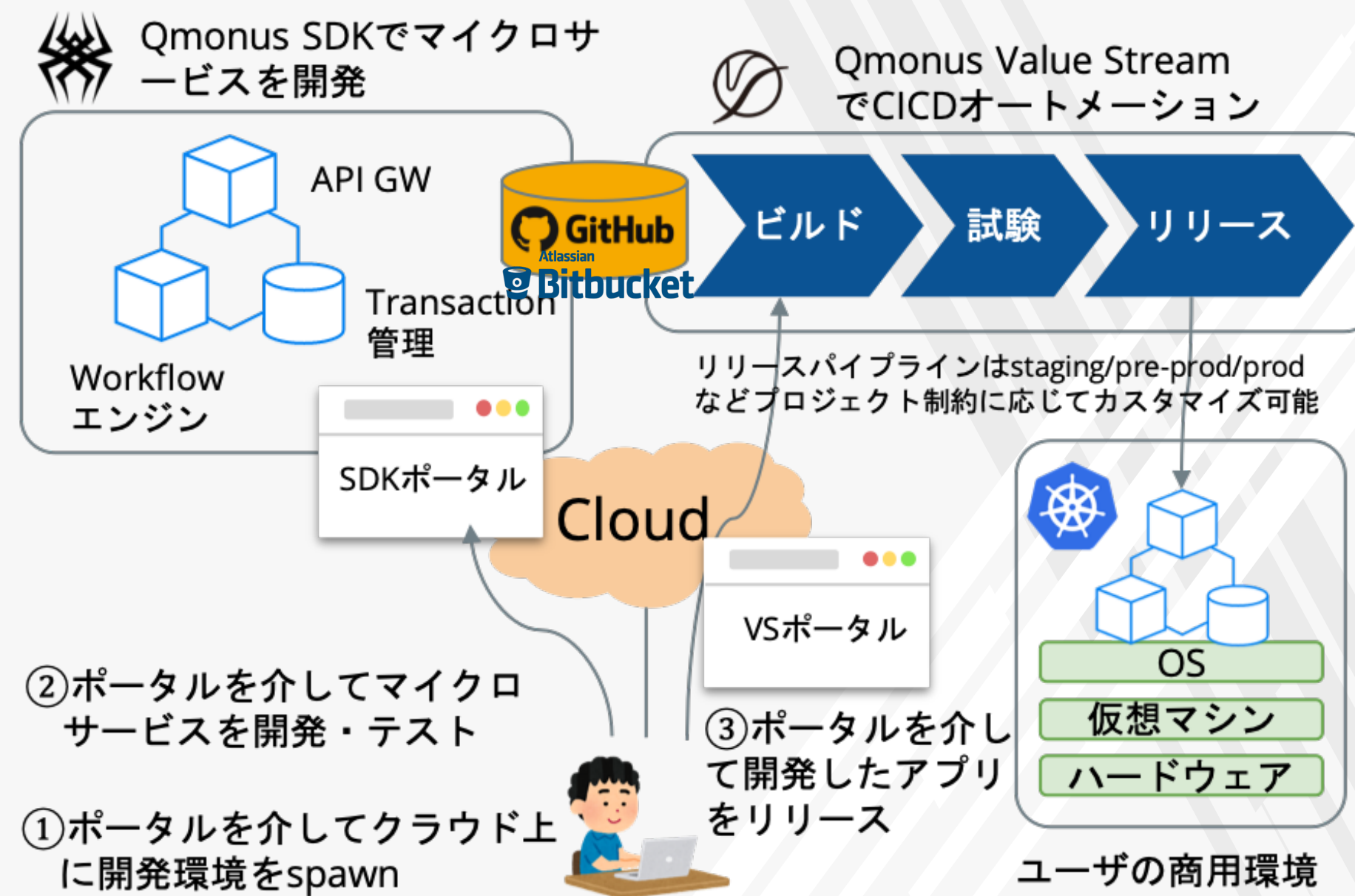
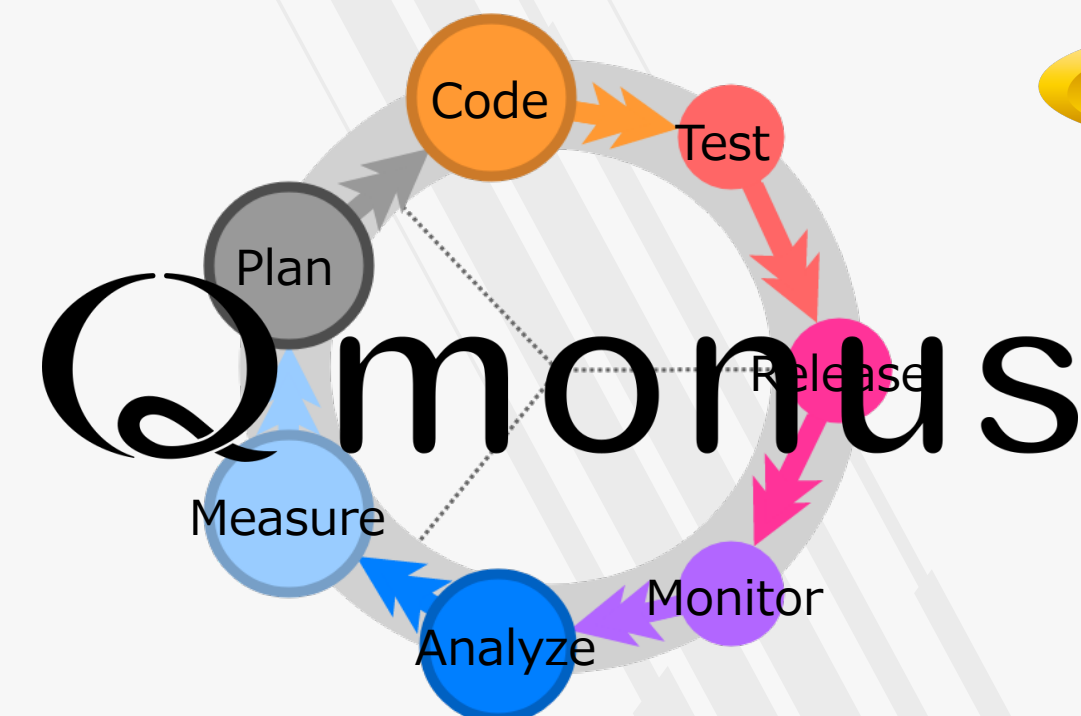
ユーザ体験

Qmonus SDK

SDKポータルより、マイクロサービスを高速に開発、強力なトランザクション管理機構を備えたワークフローやモデルドリブンエンジンを搭載、テスト駆動開発をサポート

Qmonus Value Stream

アプリケーションの構成、設定に加え、デリバリパイプラインをInfrastructure as Codeで管理し、CI/CDを自動化



Qmonus SDK



<https://axis-edge.github.io/qmonus-sdk-programming-guide/>

Qmonus SDK開発経緯

NTTCom内のシステムを組み合わせで新しいサービスを作ろう！

😞 組み合わせるシステムが多い

😞 システムの癖が強い

😞 ロールバック難しい

😞 これ人手でやるのは厳しい

😇 今後もこういったサービス増えるみたい

Qmonus SDK開発経緯



NTTCom内のシステムを組み合わせる新しいサービスを作ろう！

オーケストレータ

😓組み合わせるシステムが多い

つくる

😓システムの癖が強い

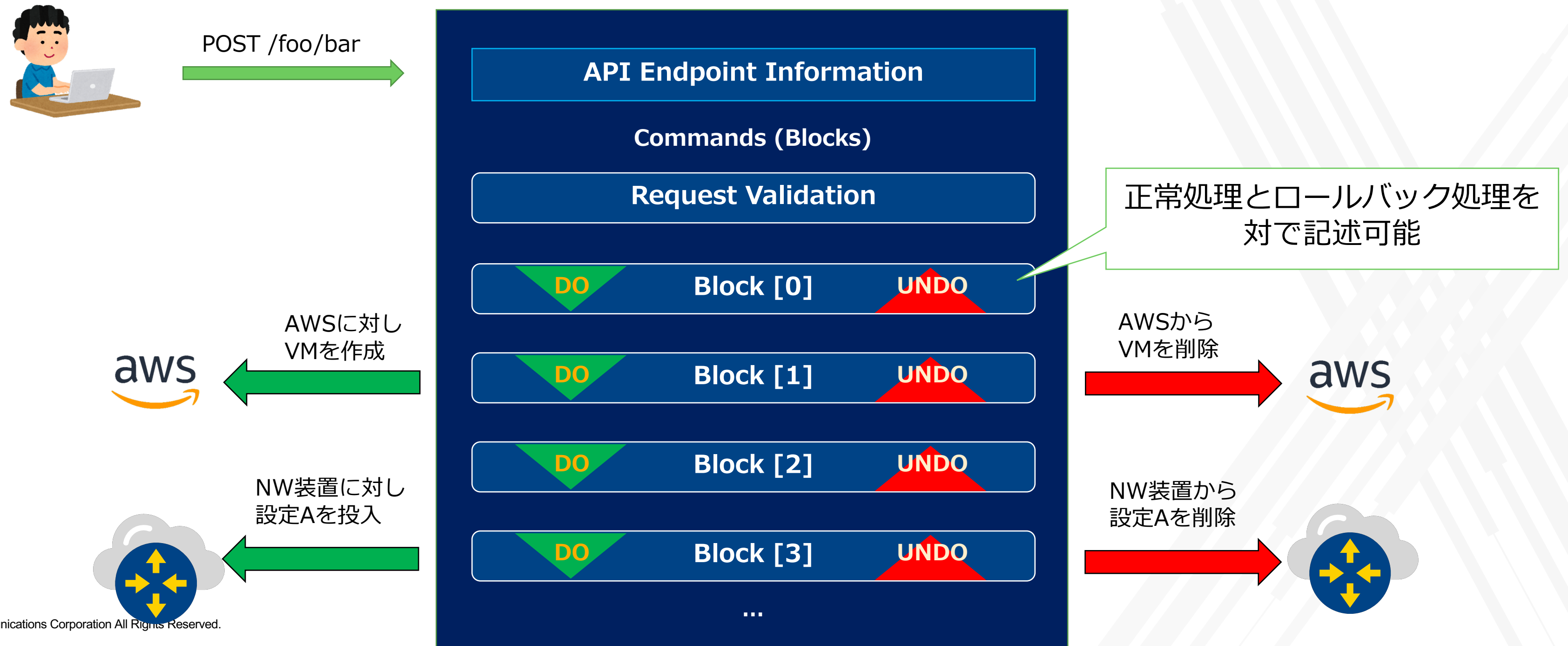
😓ロールバック難しい

😓これ人手でやるのは厳しい

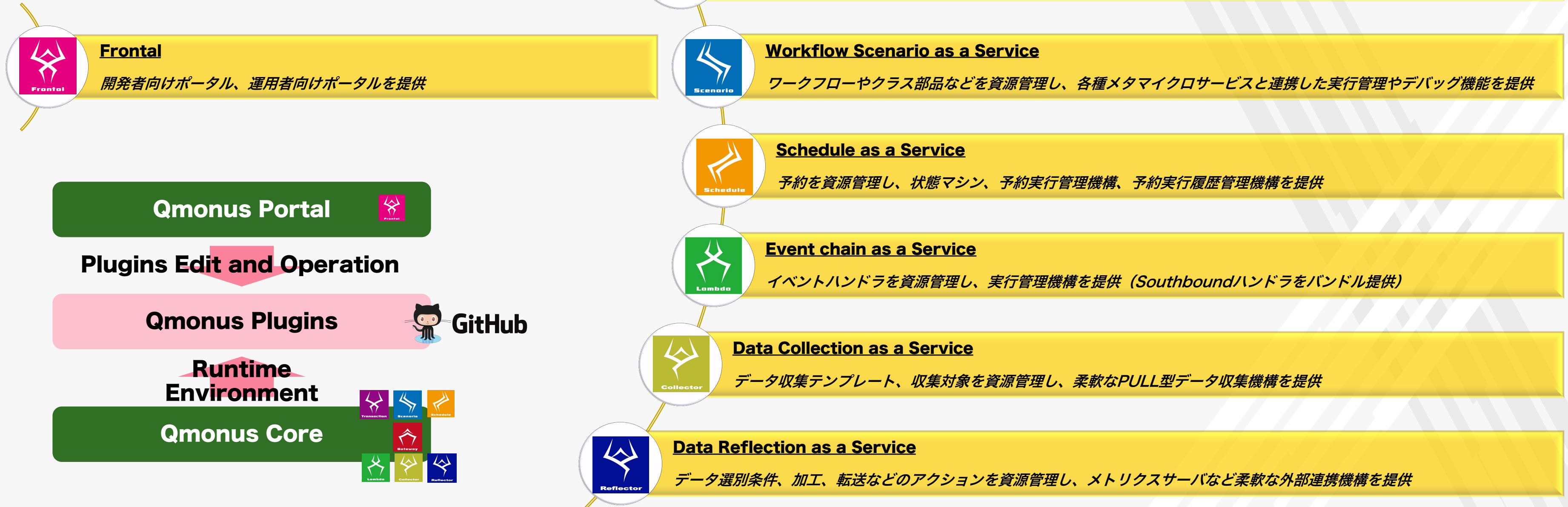
😇今後もこういったサービス増えるみたい

ワークフロー

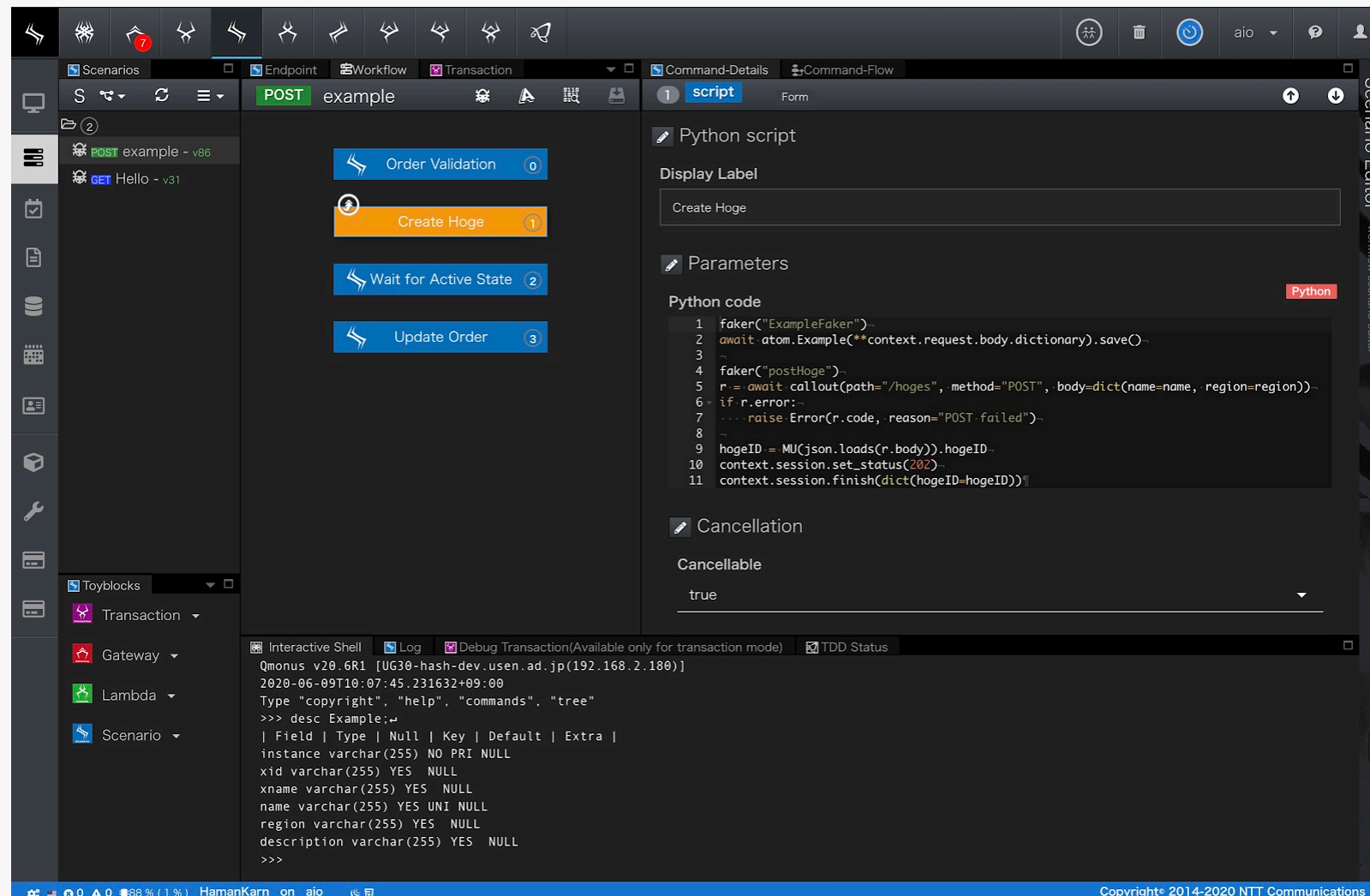
ロジックをブロック毎にPythonコードで記述する
APIでの呼び出しが可能でブロックは順次実行される
失敗した箇所からロールフォワード or ロールバックを選択する



Qmonus SDK コンポーネント一覧



Qmonus SDK Portal



クラウドベースの統合開発環境

- コードエディタ（ワークフローやクラスなど豊富なアプリ部品編集）
- アプリケーション実行、Dry Run、モニタリング
- コードトレース、プロファイリング、カバレッジ計測
- テスト駆動型開発
- ペアプログラミング（ビデオ&ボイスチャット）
- REPL
- GitHubリポジトリ連携
- カスタムダッシュボード開発
- 外形監視（SNMPd、Prometheus exporter）

Qmonus Value Stream

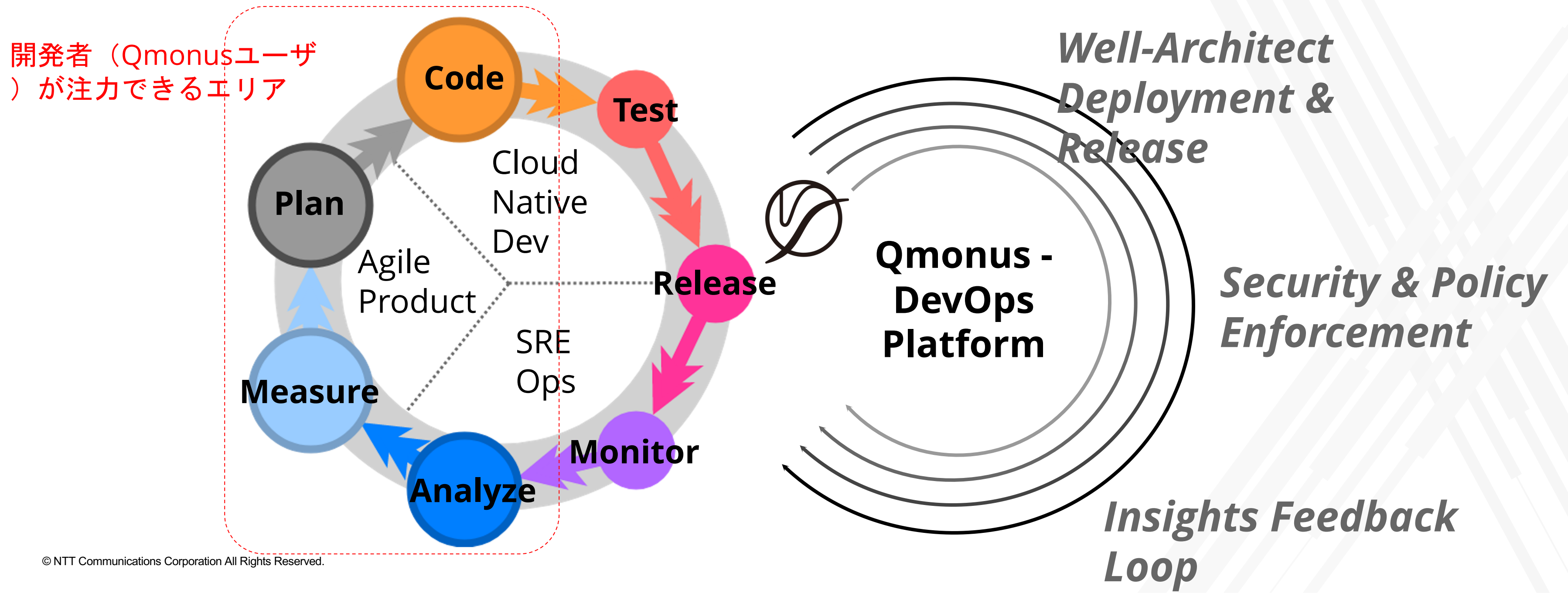
<https://vs-docs.qmonus.axis-dev.io/>



Qmonus Value Stream

アプリをユーザへ提供するまでのバリューストリームを最大化するDevOps基盤

- ・ ビジネスに価値を生む開発工程に注力するために、早いDevOpsのフィードバックループを実現
- ・ 基盤に詳しくなくてもプロのようにCloud Nativeアプリケーションをデリバリ・運用ができる



現状のサービス開発のツラミ

サービスごとにサイロでシステムアーキテクチャ構成の設計、検証、構築、運用を担当するため、アプリ開発に注力できない

サービス開発者

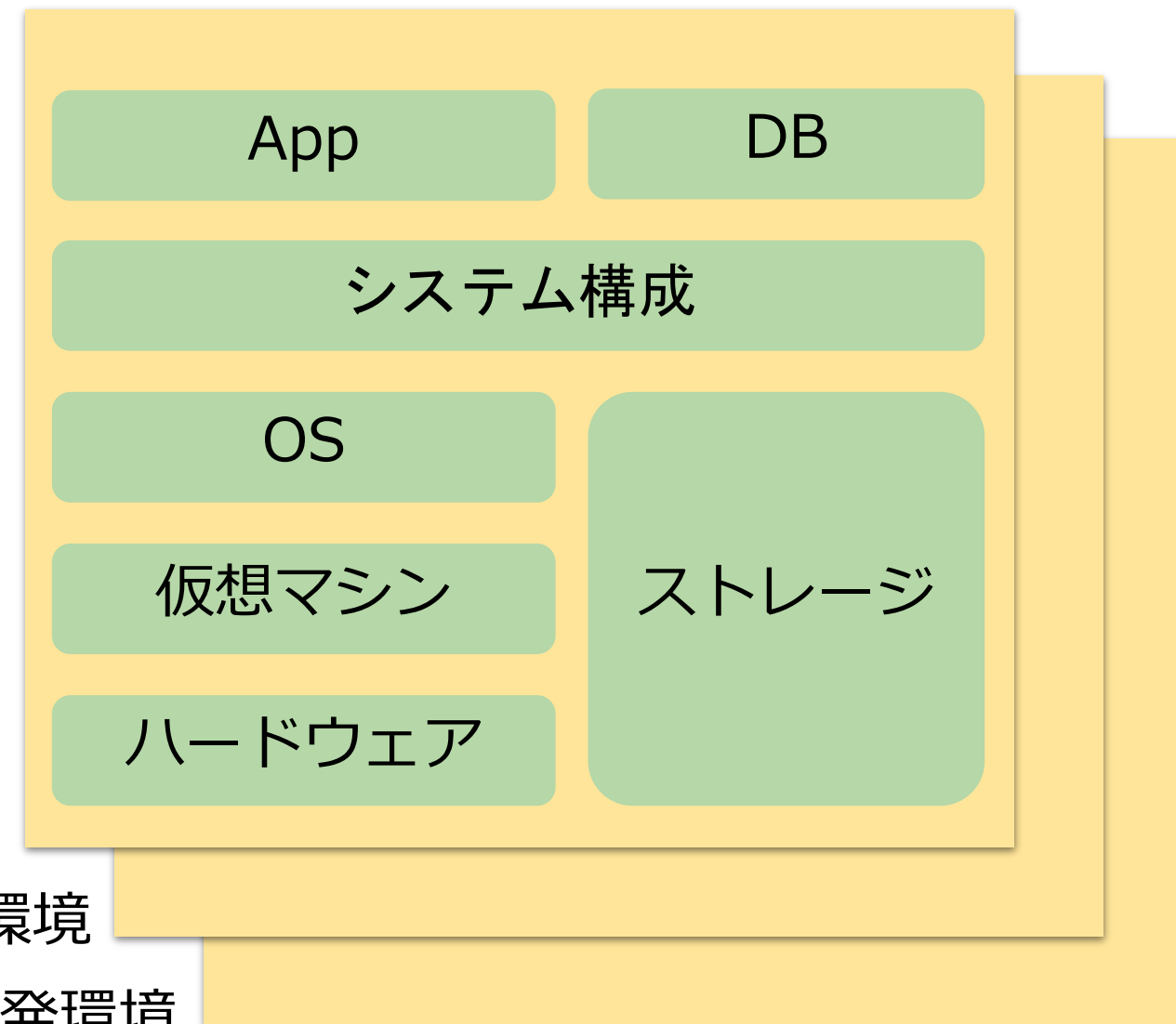


ユーザ側で全ての
環境を構築・運用

プロダクション環境

ステージング環境

開発環境



ユーザ所有・管理

Qmonus VSを導入するとどうなる

ユーザの持ち込みクラウドにパターン化されたシステム構成をワンクリックで構築
アプリ開発からプロダクションへのリリース効率を最大化

サービス開発者

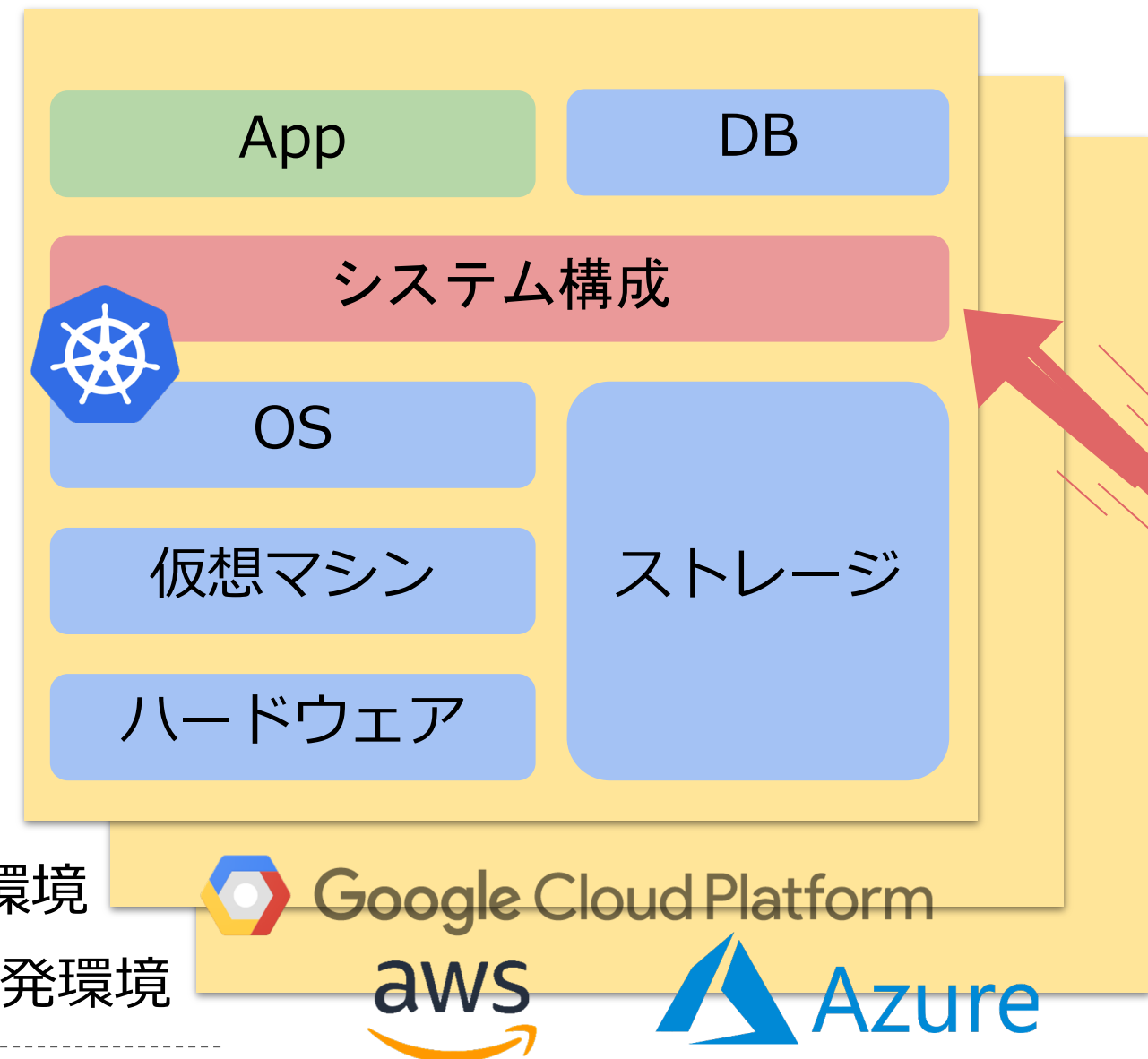


ユーザ
持ち込み
クラウド

プロダクション環境

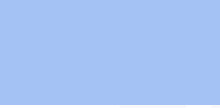
ステージング環境

開発環境



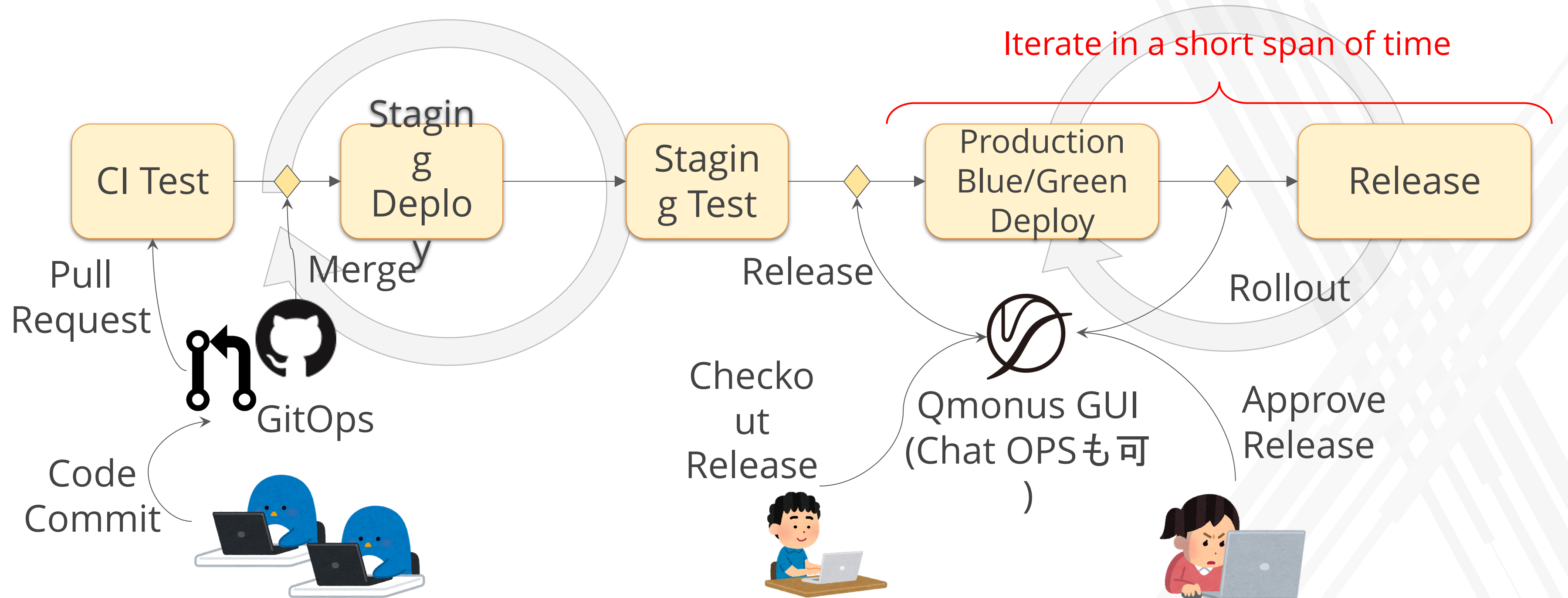
ユーザに代わって
安全に素早くデリバリ



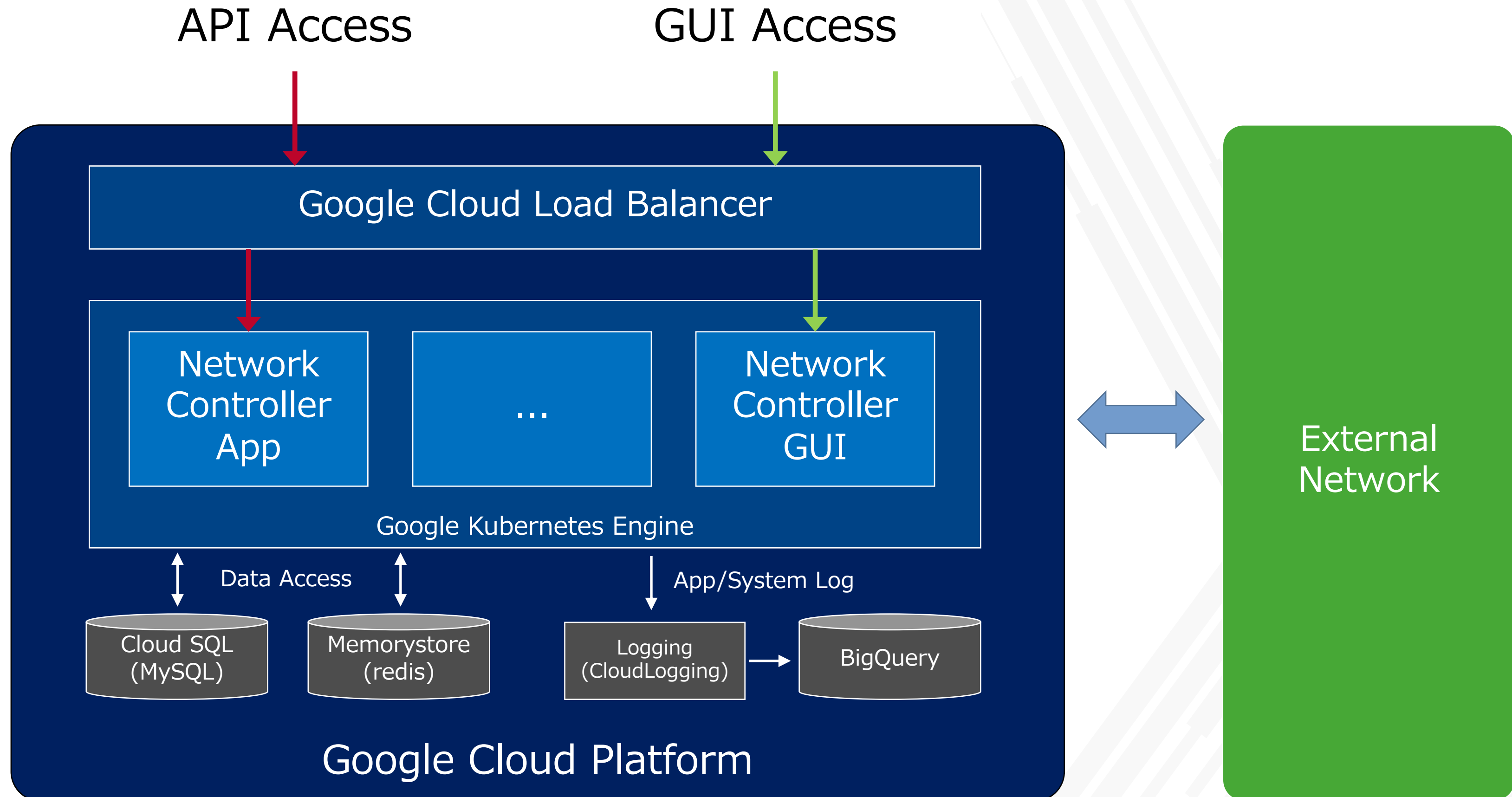
-  Qmonus VSサービス
-  ユーザ所有・管理
-  ユーザ所有 & Qmonus VSでコントロール

Simple Flow

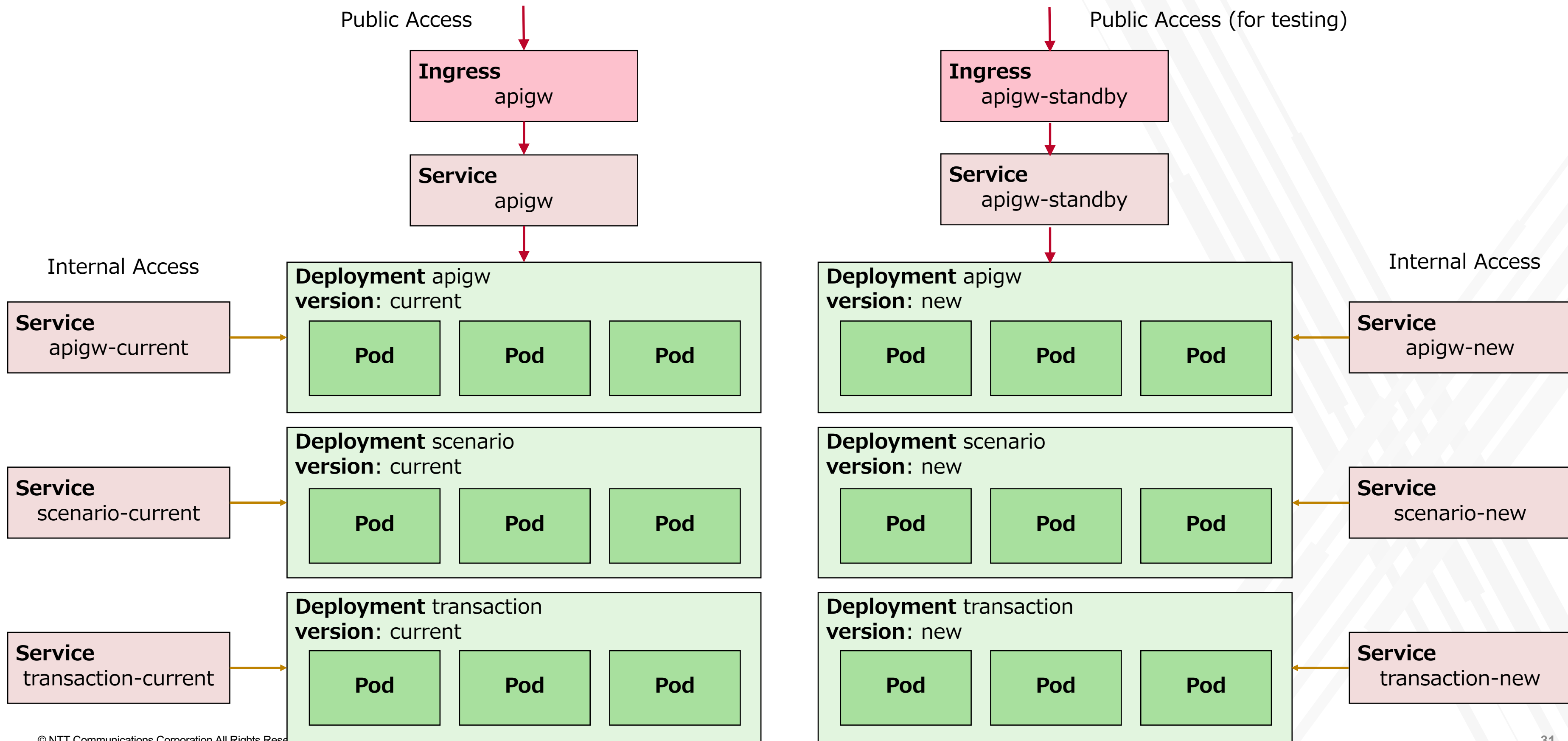
シンプルなインタフェースと簡素化された承認フローで高速かつ短いサイクルでリリースできるようになる



ネットワークコントローラのアーキテクチャ



Kubernetesリソース Overview



Blue/Green デプロイメント

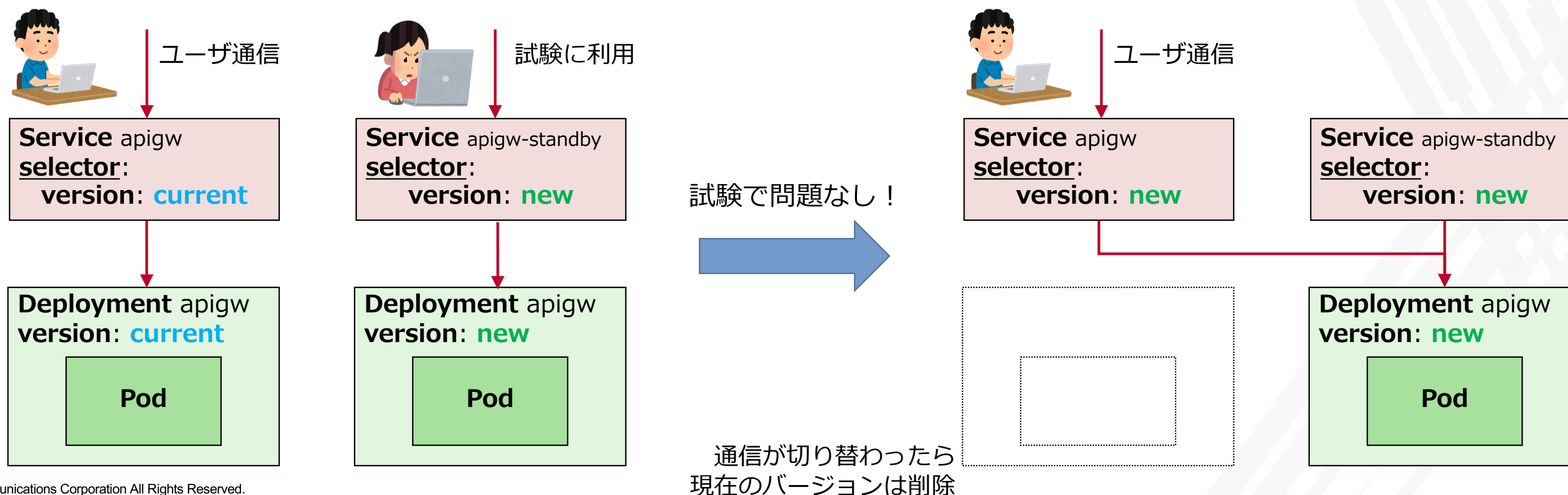
ユーザに極力影響を与えずにバージョンアップを行う手法

現在のバージョンを動かしながら新しいバージョンを立ち上げ問題ないかを確認する

問題なければ通信を新しいバージョンに**ダウンタイム無し**で切り替え、

問題があれば通信はそのまま新しいバージョンを削除する

リリースサイクルを速める仕掛けになっている

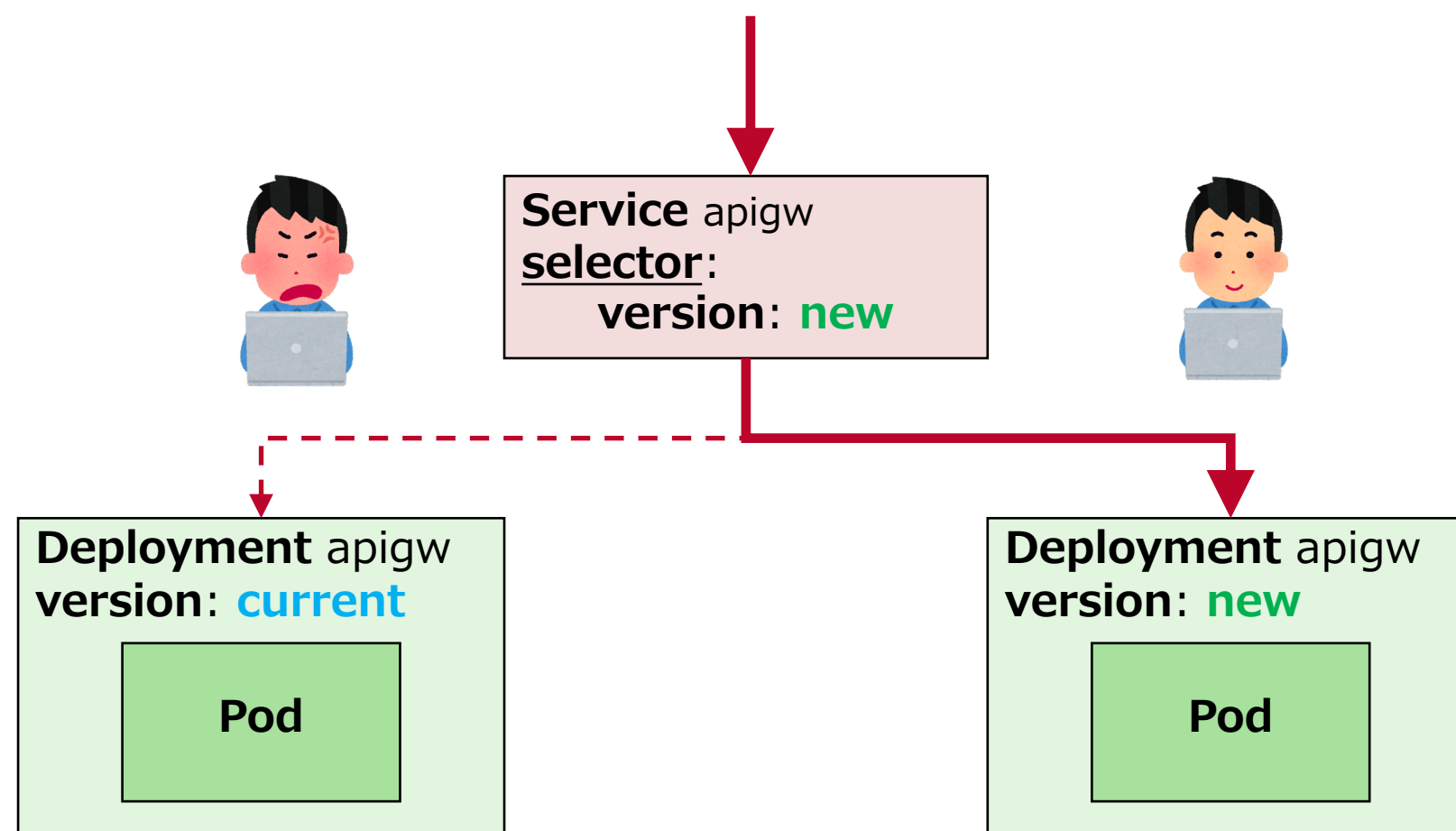


Blue/Green デプロイメントのツラミ

実は通信切り替え時に問題がある

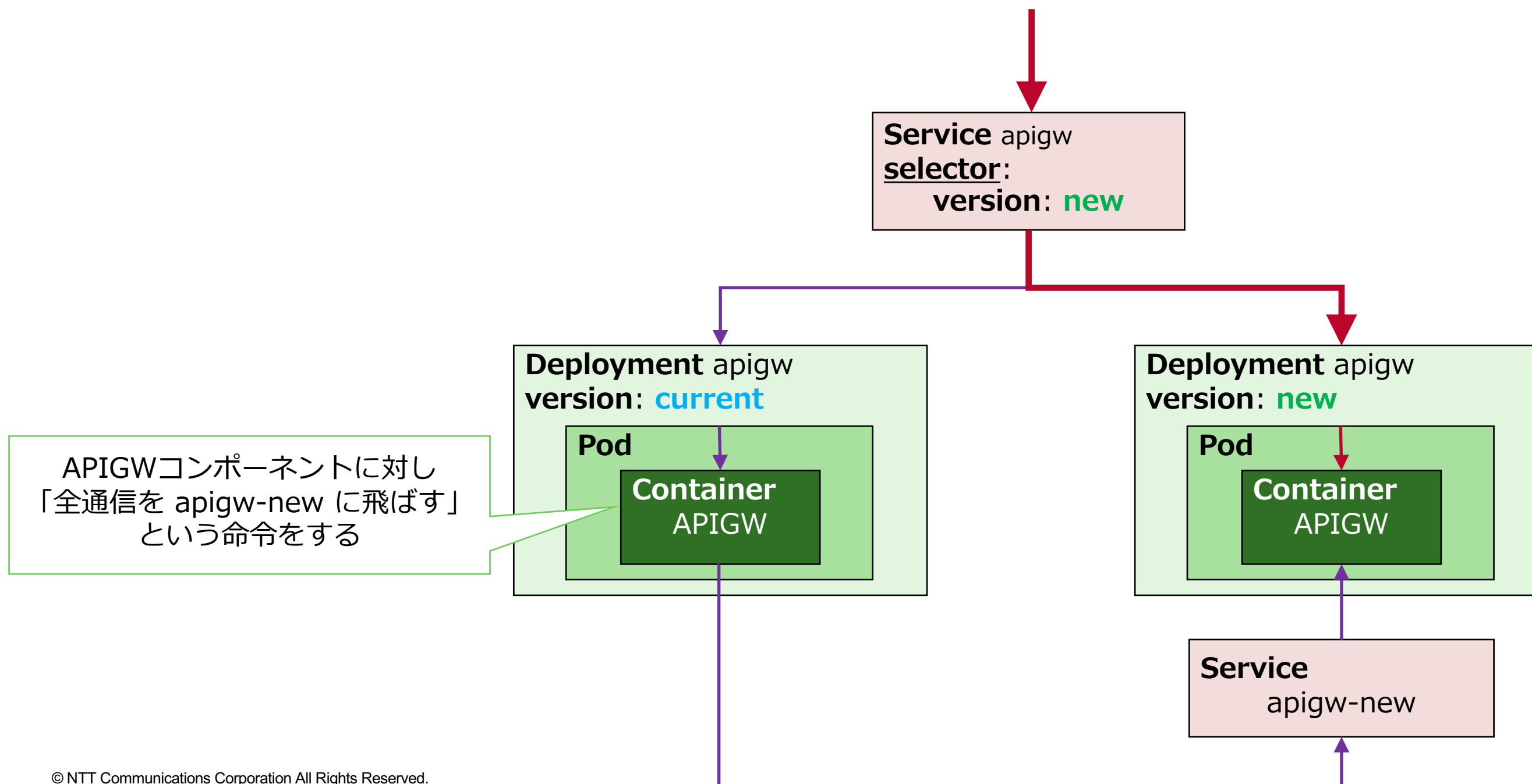
Serviceリソースのselectorを書き換えて通信の切り替えをすると、
ある一定の期間(10分程)少量の通信が書き換え前の系に入ってしまう

あれ、ダウンタイム無しにできないのでは。。



一旦こう解決した

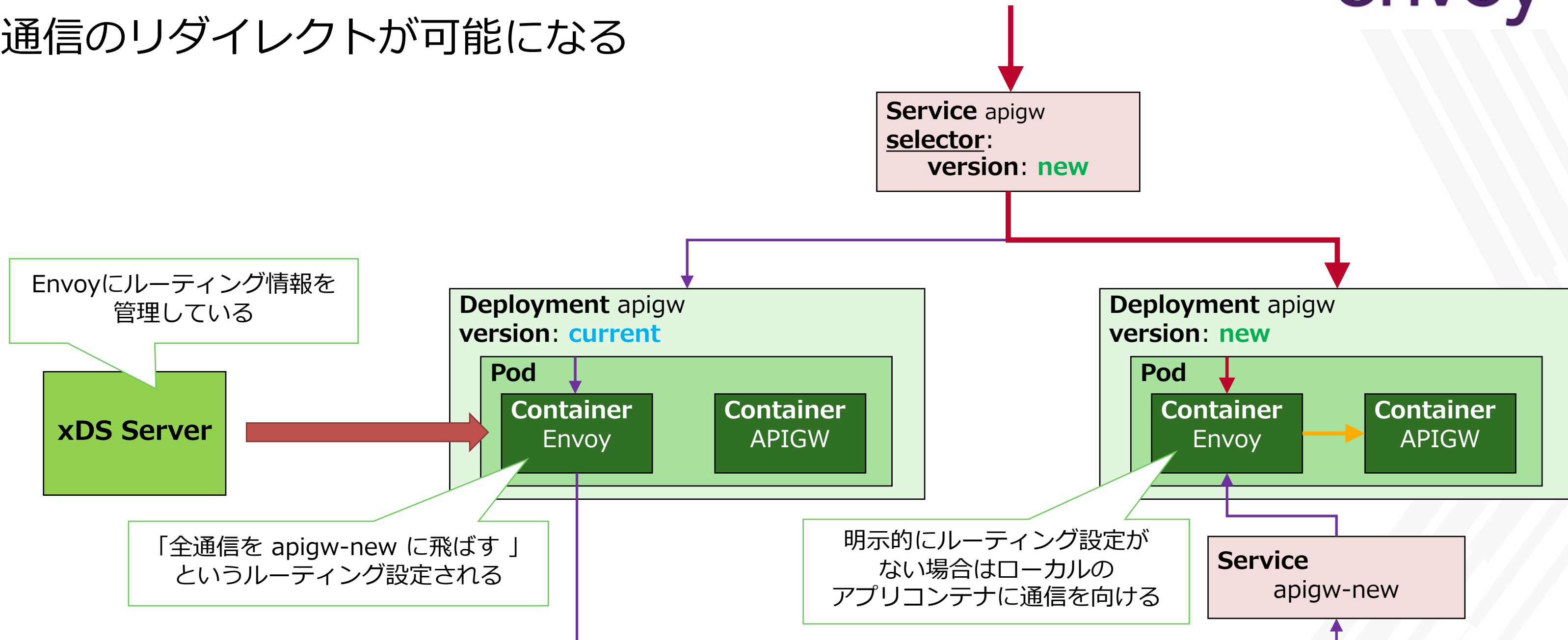
ゴリゴリではあるが、Qmonus SDKのAPIGWコンポーネントで通信のリダイレクトを行う
 ただ他アプリにこのリダイレクト機能があるわけではないので汎用的解決策ではない



更なる解決策

高機能で軽量なL4/L7ロードバランサーである**Envoy**を
Sidecarパターンとしてアプリコンテナとともに立ち上げる

アプリコンテナの通信を全てEnvoyコンテナが中継することにより
通信のリダイレクトが可能になる



良かったこと/大変だったこと

【良かったこと】

- アプリの死活制御をKubernetesに任せることができた
 - アプリが落ちた場合の自動復旧やスケーリングが容易に
- アプリの構成について、 Infrastructure as Code として管理することができた
 - パターンとして抽象化・再利用ができるようになり、様々なプロジェクトで流用することができる
- リリースをパターン化することができ、リリースのための準備・稼働を減らすことができた

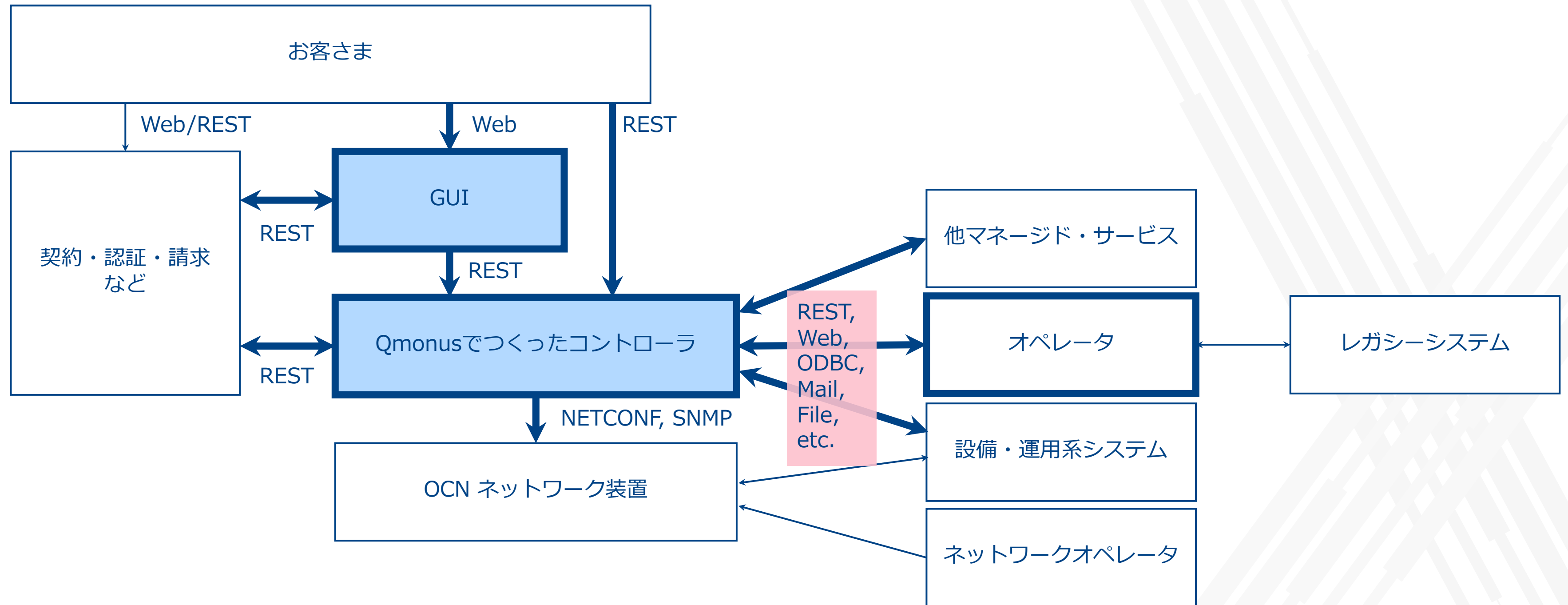
【大変だったこと】

- Kubernetesの世界にどっぷり入らないといけない
 - 通信系はなかなか追えなくて大変だった
- Kubernetes Clusterの運用はしたくなくてその整理が大変だった
 - マネージドに管理してくれるクラウドを使おう
 - ただこの規模でクラウドを使う例が過去になく、社内を説き伏せるのに時間労力がかかった

OCNコントローラの作り方と苦労

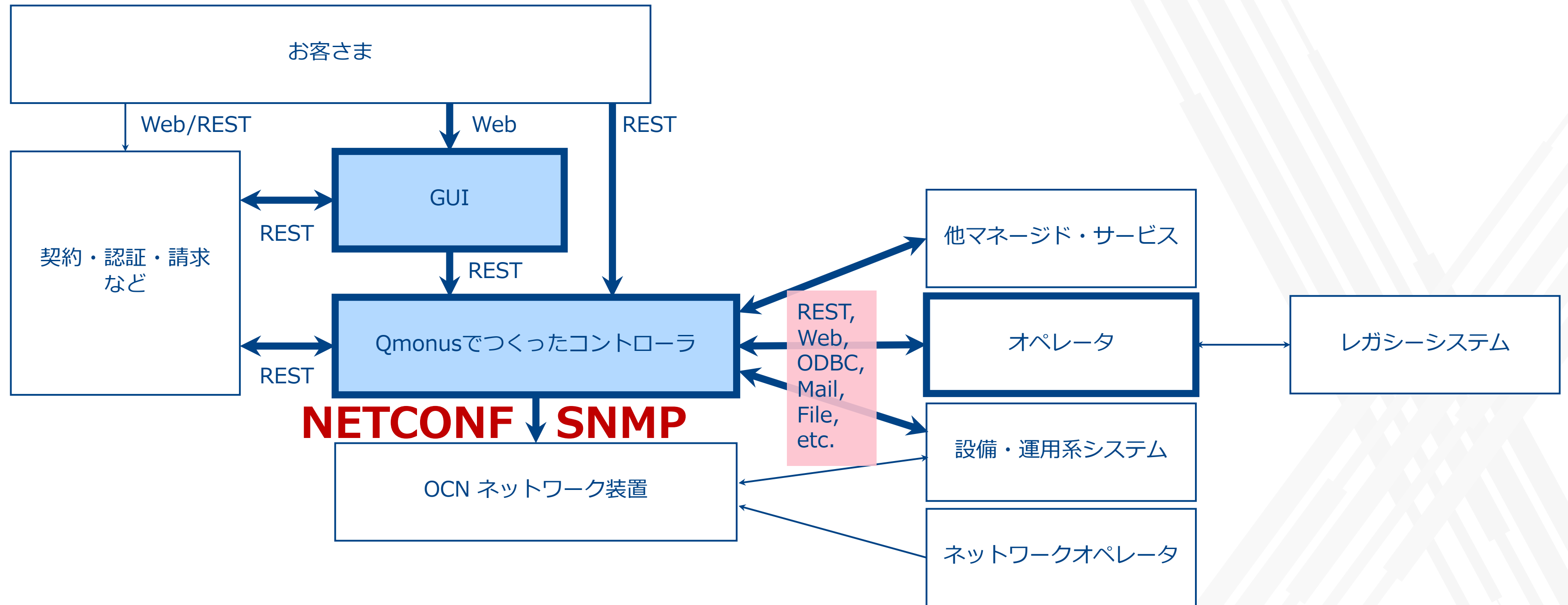
実際に開発した機能アーキテクチャ図

- お客さま向けには一貫した Web UI, API を提供！
- 非APIな茨の道。ヒト interface な場面も…



実際に開発した機能アーキテクチャ図

- お客さま向けには一貫した Web UI, API を提供！
- 非APIな茨の道。ヒト interface な場面も…



SouthboundはNETCONF/SNMP

CLI

- Telnet/SSH
- 出力の変換つらい

NETCONF

- 装置対応済み
- system friendly
- xmlはつらい

RESTCONF

- 装置未対応
- まだバグもありそう

OpenFlow

- 装置未対応
- もはやレガシー？

SNMP

- 伝統
- 装置対応済み
- 取得が多いと重い

Telemetry

- マルチベンダなコレクタがない
- 今後期待

JANOG43 「OCNネットワークアーキテクチャの変遷と将来」 から引用

- 現在のOCNポリシー
 - ・ サービス毎にPEを用意しない
 - ・ PEルータは可能な限り集約

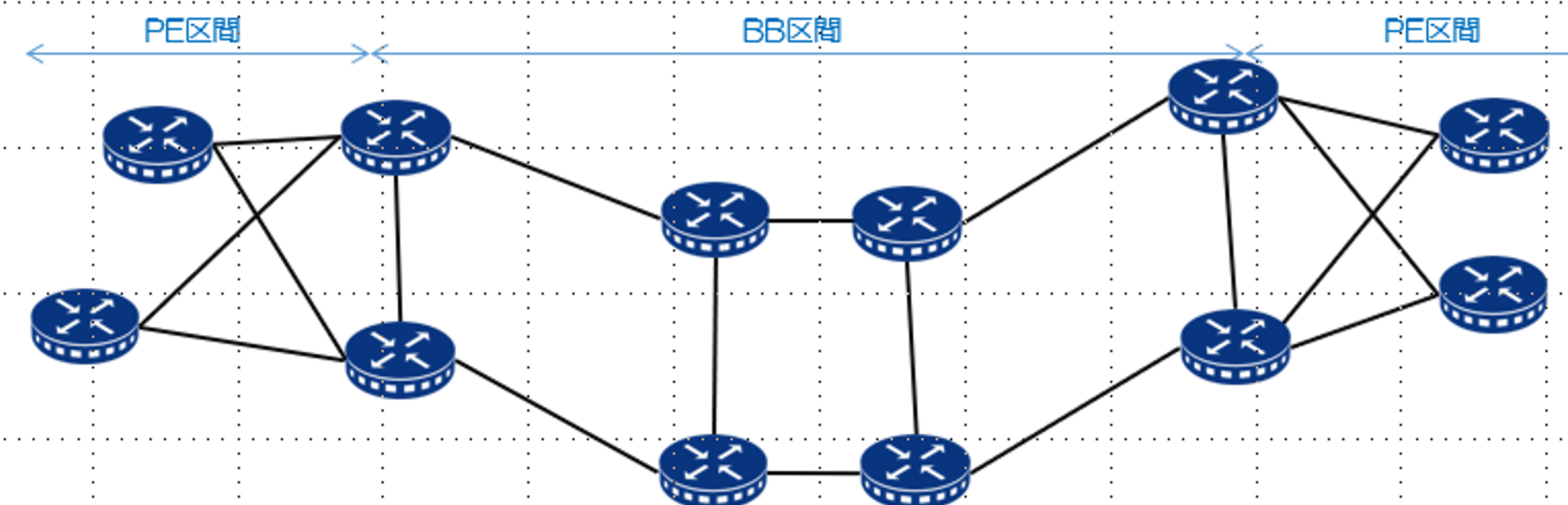
現在のアーキテクチャ概要

■ 現在のOCN基盤NWの特徴

- ・ はしご構成
- ・ OSPF/BGP/GRE/LDP/RSVP-TE/VPLS/VPWS
- ・ 1台あたり1000以上のユーザを収容
- ・ 1台あたり100以上のネットワークサービスを提供
- ・ すべてルータ
- ・ 複数リンクでLAGを利用
- ・ 複数のネットワークサービスを同一PEに収容
- ・ 同一アクセスラインでOCN/GIN/JPNAPが提供可能

これが苦勞の元

すべて3ホップ

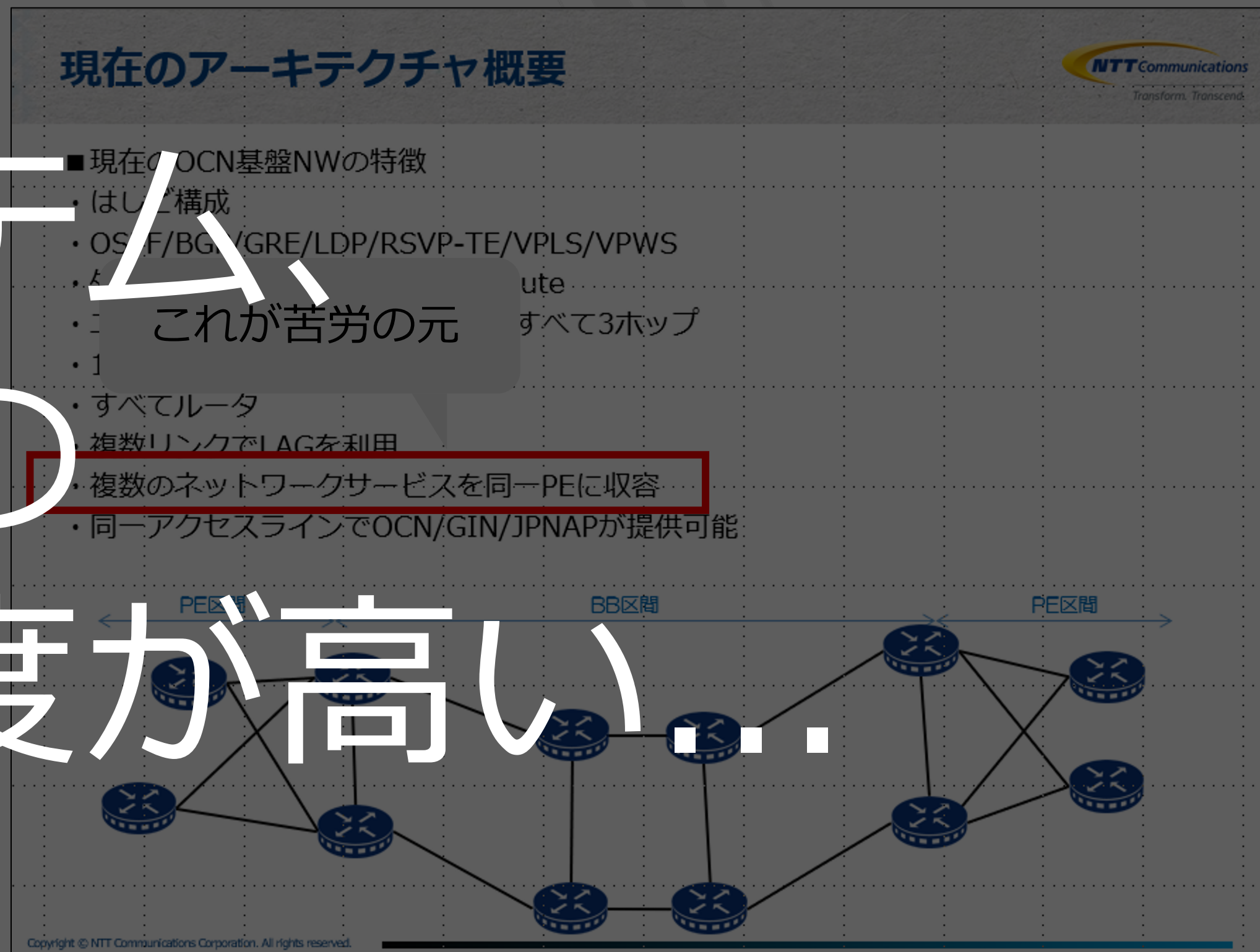


Copyright © NTT Communications Corporation. All rights reserved.

JANOG43 「OCNネットワークアーキテクチャの変遷と将来」 から引用

旧システム、 運用者の 触る頻度が高い

- # 運用者



新システムの開発指針 - 3つの『変えない』もの

■ ネットワーク設計ポリシー

- サービス横断の設備共用
(設備コスト・運用コストの集約)



ネットワーク装置は共用・ラインカード分割

■ 装置保守

- 安定している保守運用はそのままに
- バックボーンのメンテナンスなど、オペレータ作業の完全排除は不可



- Config 投入時は排他制御モードに
- 奪い合いのときのためのリトライ機能
- 保守者ログイン中は config 投入しない

■ 既存システム

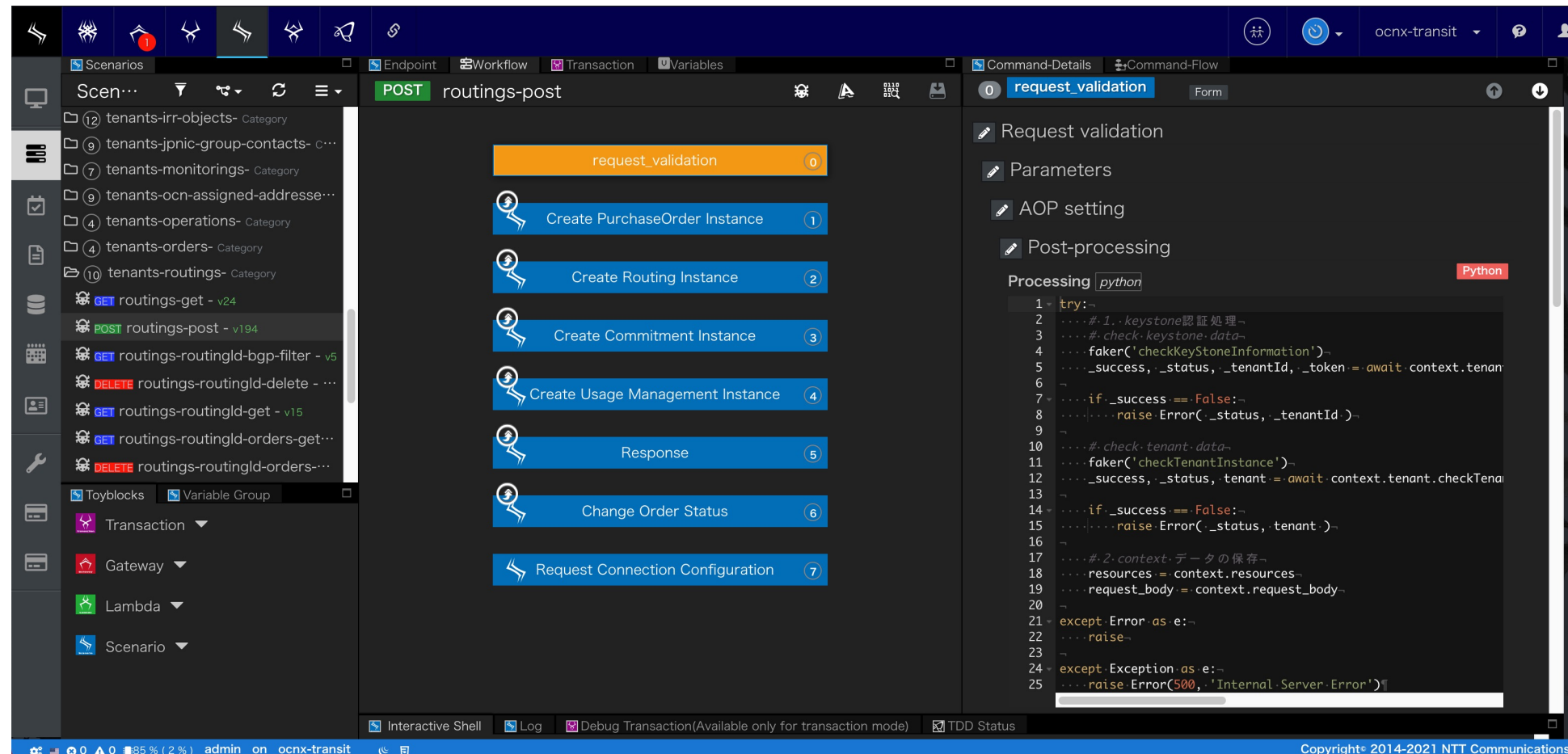
- 影響範囲の広さなど、メンテコストが高い



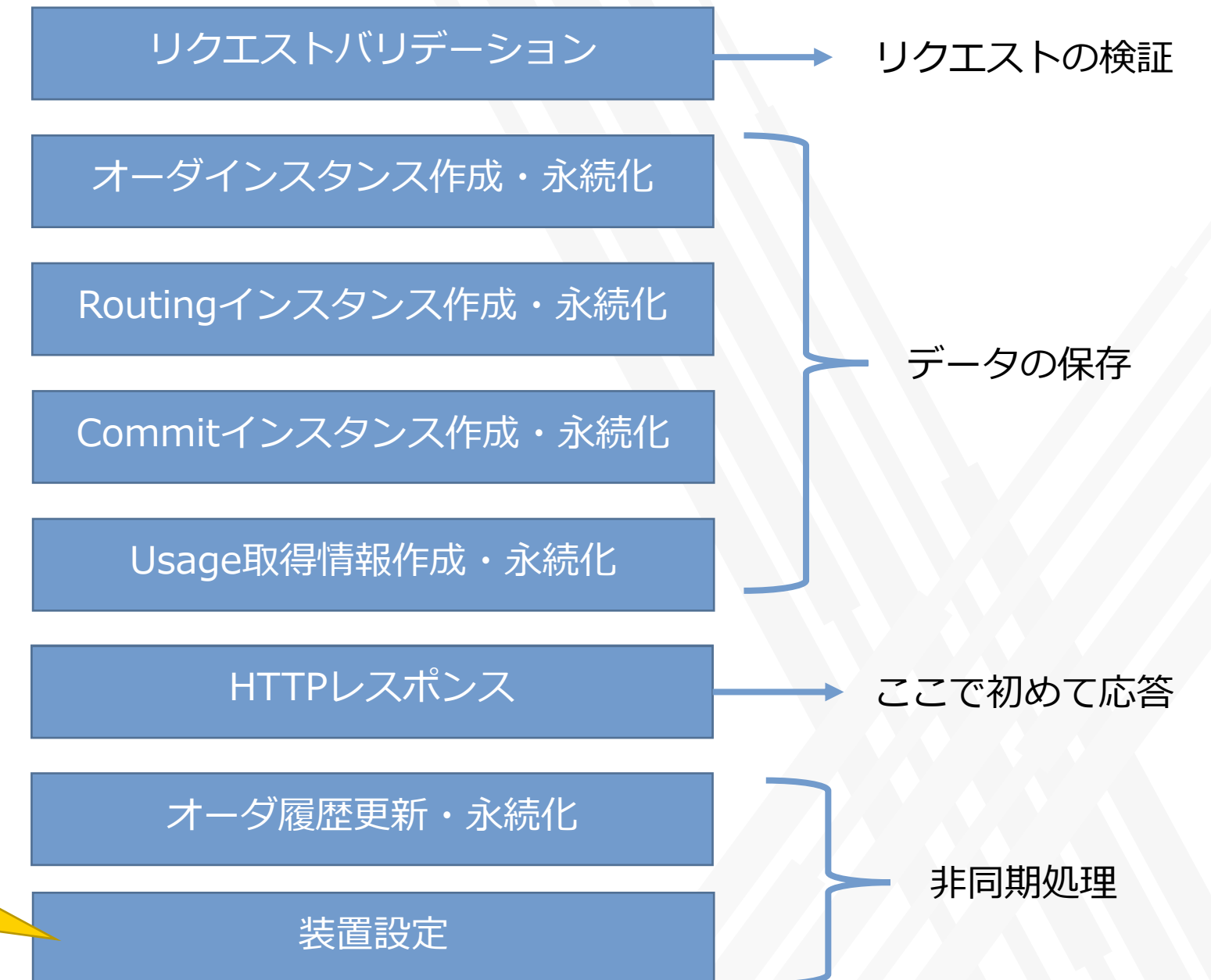
- 新サービスとしてシステムを開発
- 旧システムとの連携を極力排除
- 機能は徐々に拡充

『変える』ものを限定し、徐々に変えていく

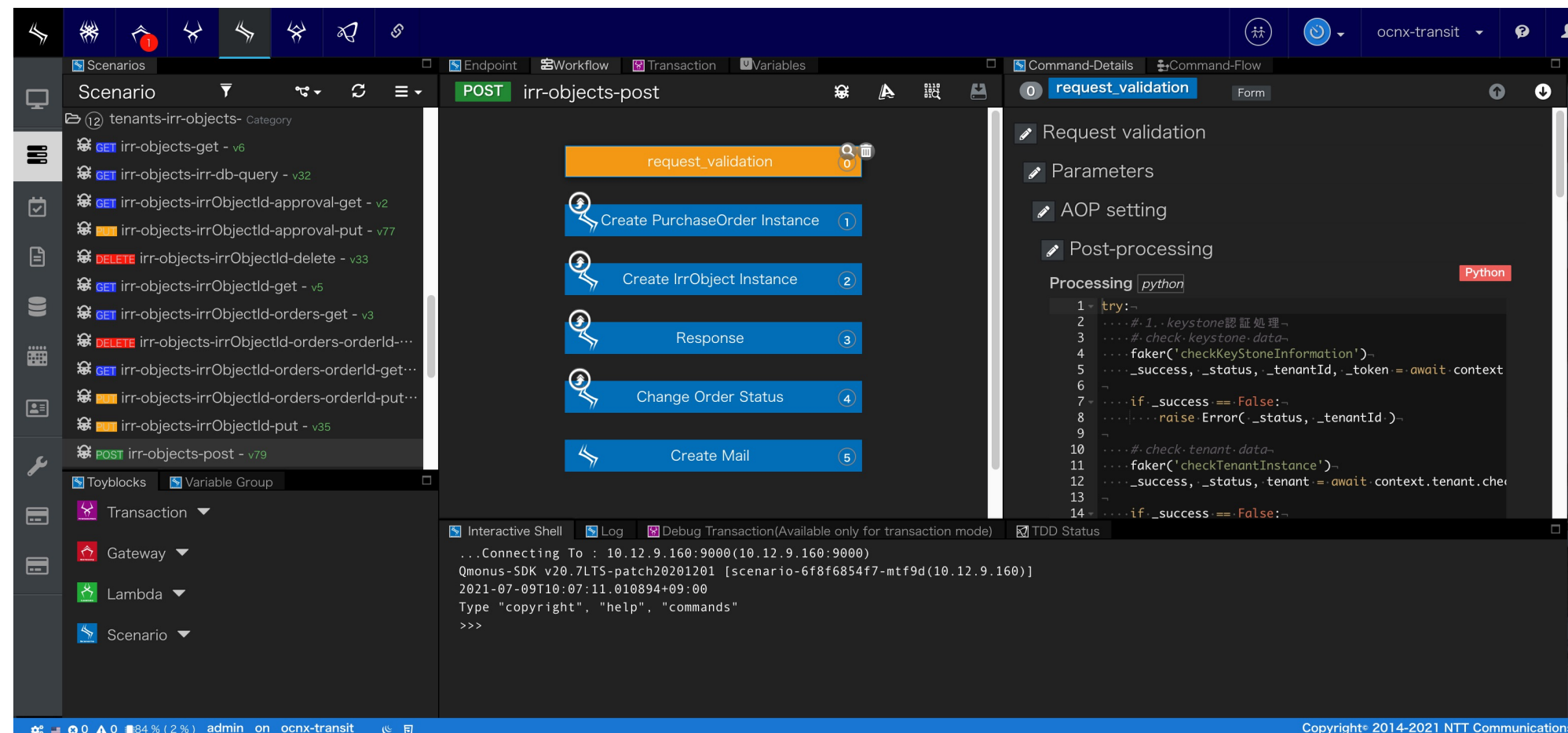
Routing作成のロジック



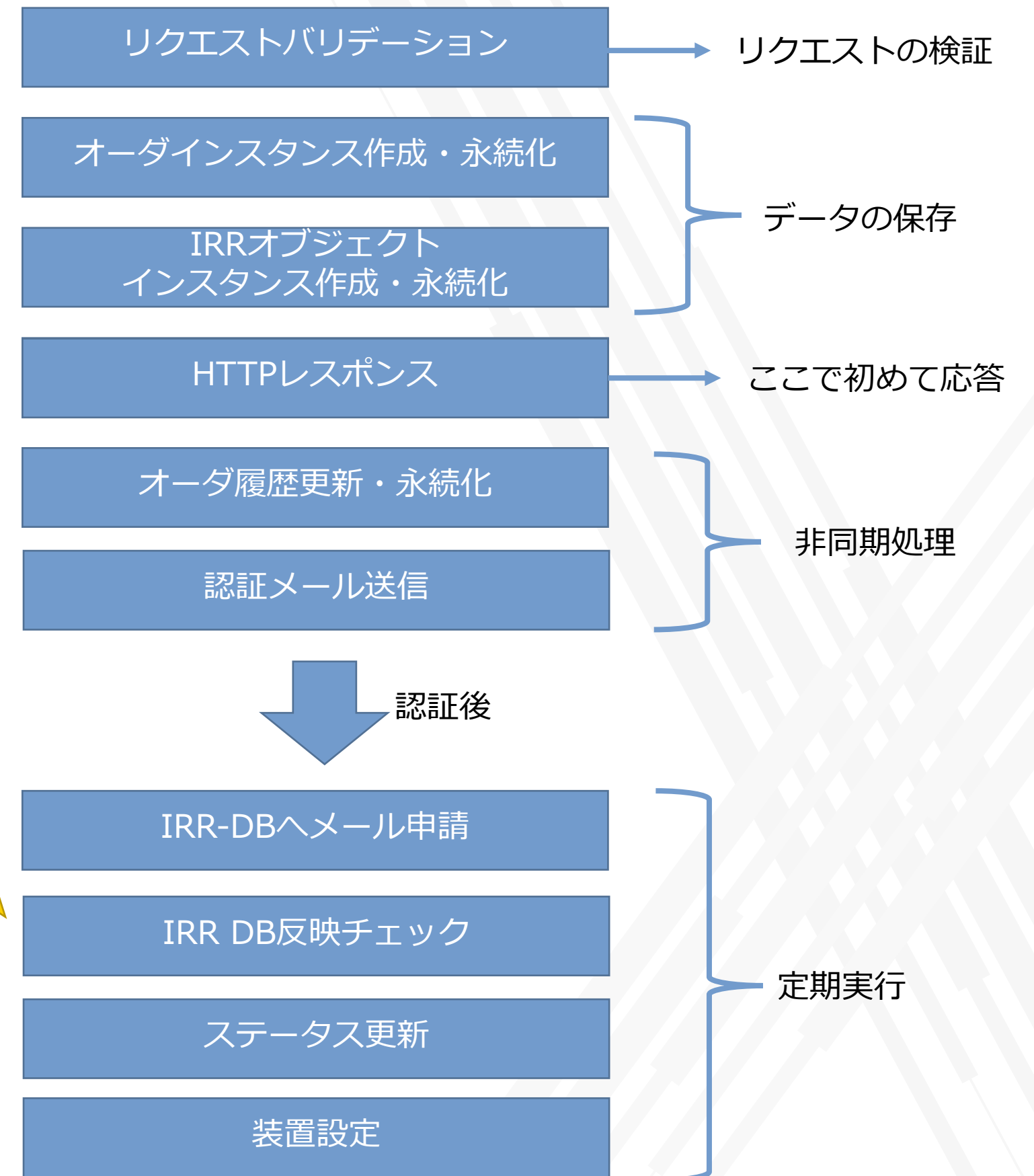
- config生成はqmonus SDK Template(jinja2)
- 成功するまで繰り返す
- 何度も失敗するようならアラーム



IRR連携のロジック



- 3つのIRR-DBへ、as-set member追加メール
- 失敗したら再送
- 申請した内容が正しく反映されているかチェック
- 前日の内容と差分があれば、config投入



■ IPアドレスのWhois登録ってどうやってるか知ってますか？

■ IPアドレスのWhois登録ってどうやってるか知ってますか？

- JPNICが用意しているWebで申請します
 - IPアドレス管理指定事業者専用のWeb申請システム
<https://iphostmaster.nic.ad.jp/jpnic/dispmemberlogin.do>
- つらみ
 - 電子証明書をインストールしたブラウザのみアクセス可能
 - システム連携といえな**APIがありません**

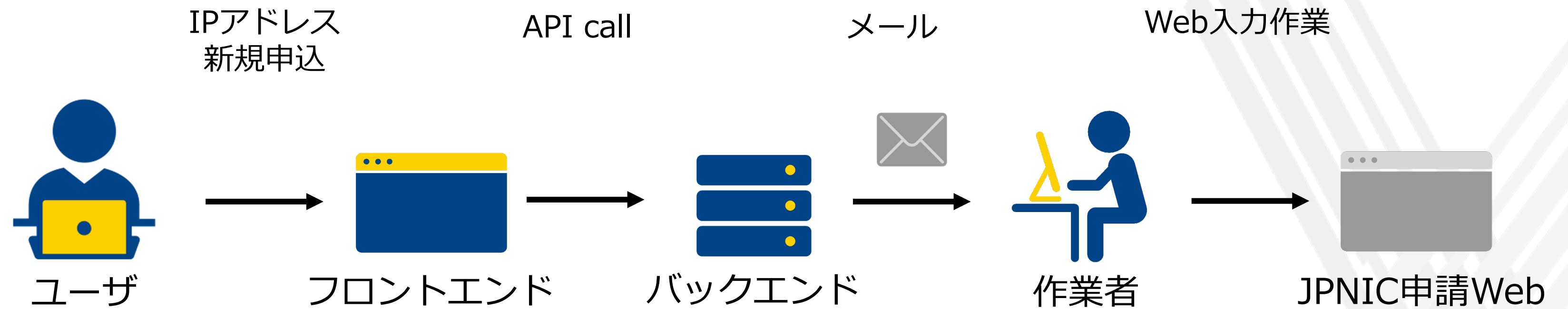
どうしたか？

人です。

人。

人が操作しています。

IPアドレス割り当ての流れ



JPNICさんに聞いてみたい

やっぱりAPI化は
難しい？
RDAP待ちですかね？

3つの問題点

1

複数の
システム



2

人間中心の
設計



3

エクセル
オーダー



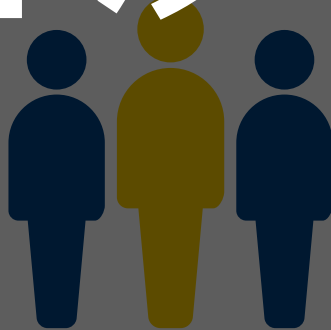
1

複数のシステム



2

人間中心の設計



3

エクセルオーダー



Whoisを除き、
すべて解決



議論したいこと

- ネットワークコントローラ導入で一番苦労したのはどこですか？
- 装置負荷観点で、snmpつらくないですか？
- アクセスラインって人ありきですよ？
どうオーケストレーションするといい？
- システムの開発・リリース工程で自動化ツールやCI/CDを取り入れてる？
- Whois登録つらいですよ？