

次世代テストベッド運用に向けて

データベースに基づくテストベッドにおける資源
管理と運用可能なテストベッド間連携フレーム
ワークの研究開発

奈良先端科学技術大学院大学
情報科学研究科 助教
櫛山寛章

自己紹介

- **樫山寛章（はぜやまひろあき）**
 - 博士（工学）
 - 専門：ネットワーク運用、ネットワークセキュリティ、大規模ネットワークエミュレーション、など
- **JGN2plus から特別研究員として参加**
 - JGN2plus 時代にはJGN-X IP仮想化サービスのサービス開発に携わる

本日のトピックとキーワード

- **フェデレーション**

- Local Broker Testbed Federation model

- **マネジメント**

- DB oriented Testbed Management

- Resource / Service Template

- **オーケストレーション**

- Testbed API

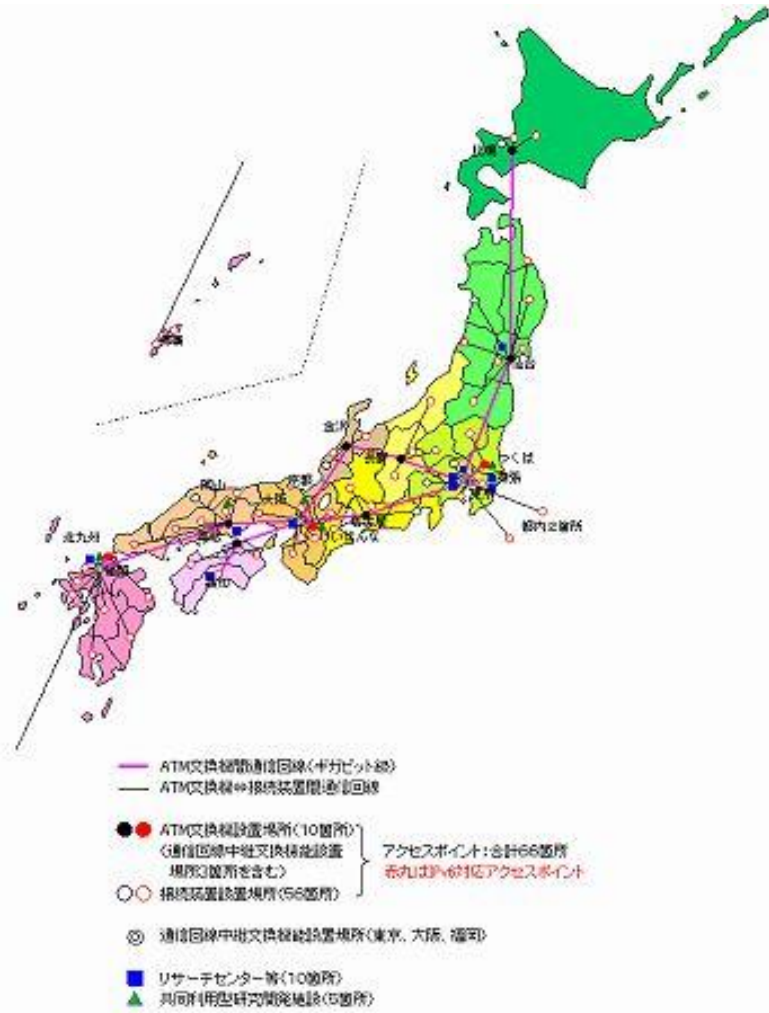
- Modularization

JGN の変遷

- **JGN (1999 - 2004)**
- **JGN2 (2004 - 2008)**
- **JGN2plus (2008 – 2011)**
- **JGN-X (2011 -)**

JGN

- ATM による回線網
- IPv6





構成図でみる変遷

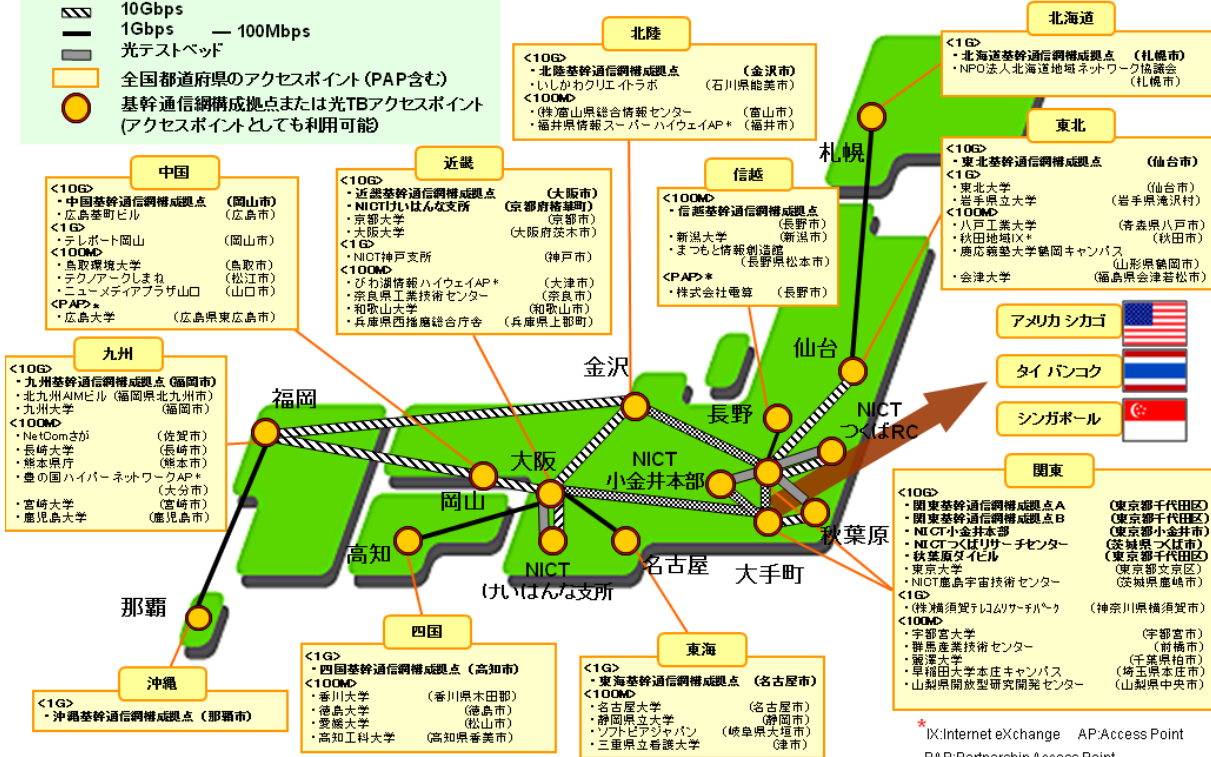
JGN2ネットワーク概要

H19年8月現在

[凡例]

-  20Gbps
-  10Gbps
-  1Gbps
-  100Mbps
-  光テストベッド

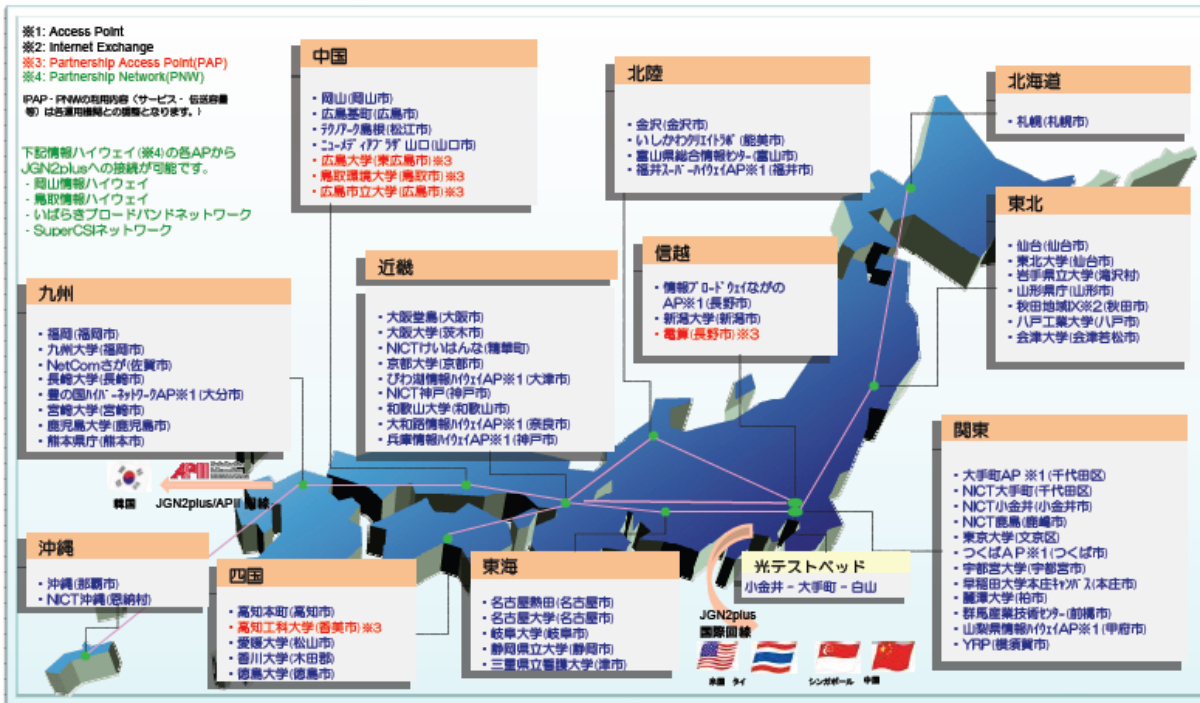
-  全国都道府県のアクセスポイント (PAP含む)
-  基幹通信網構成拠点または光TBアクセスポイント (アクセスポイントとしても利用可能)



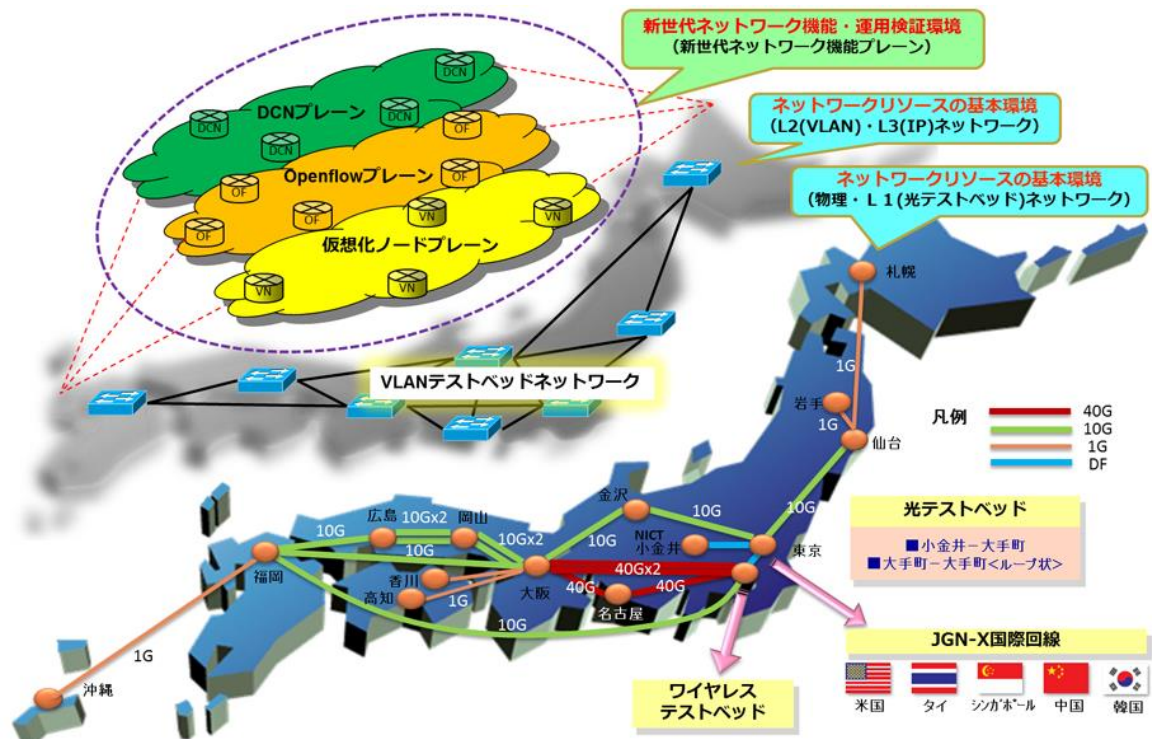
- 広域イーサネット
- IPv6 マルチキャスト

JGN2plus

- 広域イーサネット
- トライアル
 - 仮想化ルータ
 - ネットワークストレージ
 - 仮想マシン
 - PIAX / Planetlab
 - Open Flow
 - PerfSoner
 - DCN

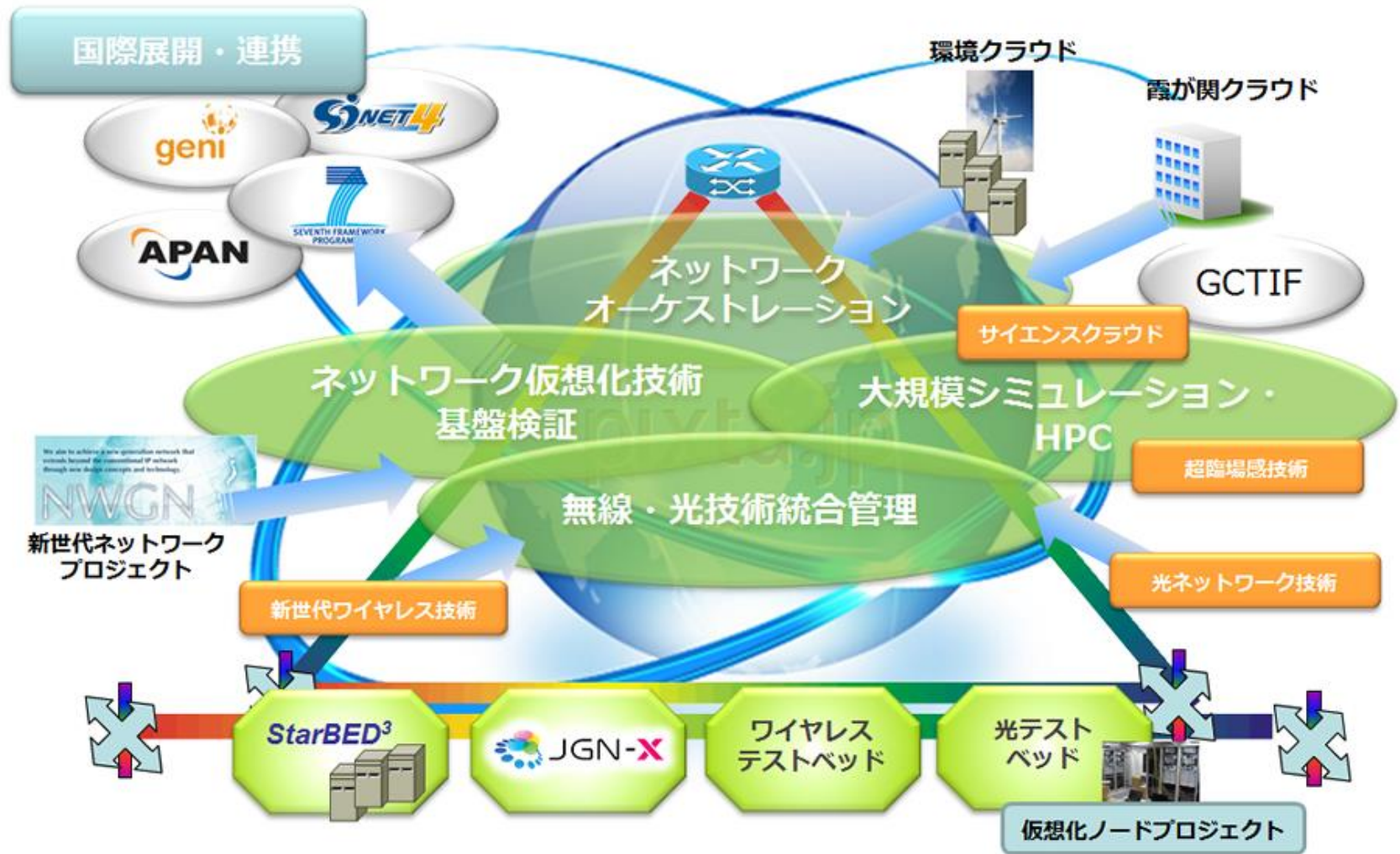


JGN-X



- ・ 標準サービス
 - 広域イーサネット
 - IP仮想化サービス
- ・ サービス化中
 - RISE サービス
 - ✓ Open Flow
 - PIAX サービス
 - DCN サービス
 - 仮想化ノードサービス

JGN-X への多種多様な要求



一方、運用体制は？

- **JGN 時代からほとんど変わっていない**
 - 事務局：利用申請書の受付と実験内容の精査
 - NOC：利用申請書から実際の資源割り当てと設定、トラブル対応
 - 特別研究員：新技術のサービス化
- **資源管理**
 - 申請書
 - 簡易なデータベース
 - ✓ スイッチのポート割り当て管理など
 - トラブルチケット

JGN-X の運用

- 緻密な資源管理とスケジューリングが必須
 - 一定常的な実験(研究開発)と一時的な実験(Super Computer, さっぽろ雪まつりなど)が混在
 - 限られた資源(VLAN、帯域、サーバ、CPU)
 - 商用サービスのように増強・増設は簡単にできない

JGN-X の運用

- サービスの組み合わせ、依存関係が掛け算で複雑になっている
 - 仮想化技術（サーバ、ストレージ、MPLS、VPLS、OpenFlow、仮想ルータ、などなど）
 - サービス（光、L2、L3、IP仮想化、RISE）
 - StarBEDとの連携利用や、異なるプロジェクト間での連携実験の増加

JGN-X の運用

- **現状 はNOC の運用力でカバー**
 - 申請書やトラブルチケットだけでは追い切れない
 - データベースの備考欄や運用wiki への記載が増加
 - ✓ JGN2plus時代のデータベースを継続利用しているが、破綻しかけている
 - 人員が限られているので、スケールしない状態
 - ✓ イベント前などは過密スケジュールになりがち
- **このままでは、次世代テストベッド運用の破たんが見えている！**

委託付き共同研究

- 「テストベッドネットワークにおけるリソース管理および運用連携を実現するアーキテクチャの研究」
 1. 運用連携フレームワーク, 運用方法の整備 (フェデレーション)
 2. 資源管理フレームワーク, 運用方法の整備 (マネジメント)
 3. 環境統合アーキテクチャの整備 (オーケストレーション)

チャレンジ

- (人間力による)サービスクオリティの維持・向上
 - 運用負荷の逡減
 - サービス項目の増大
-
- 上記3項目の最大化

人間力が必要なところ

- 実験利用の承認
- 資源利用における調停や調整、スケジューリング
- ざっくりとした実験要求から、きれいなネットワーク設計に落とし込む叡智
- 新技術のサービス化
- トラブル対応

人間力をICTで補えるところ

- **不整合の検出**
- **ルールに沿った設定投入、
パス設計**
- **割り当てリソースの照会**

Testman-suite

- **TESTbed MANagement SUITE**

- 統合運用フレームワーク

- ✓ フェデレーション、マネジメント、オーケストレーションの3つの課題を運用可能なシステムとして構築するためのフレームワーク

- JGN-X の次世代運用基盤として研究開発中

- ✓ Python、XSD、MySQL、などを用いて開発

Testman-suite

- **フェデレーション**
 - Local Broker Testbed Federation model
- **マネジメント**
 - DB oriented Testbed Management
 - Resource / Service Template
- **オーケストレーション**
 - Testbed API
 - Modularization

Testman-suite

- フェデレーション

- Local Broker Testbed Federation model

- マネジメント

- DB oriented Testbed Management

- Resource / Service Template

- オーケストレーション

- Testbed API

- Modularization

フェデレーション

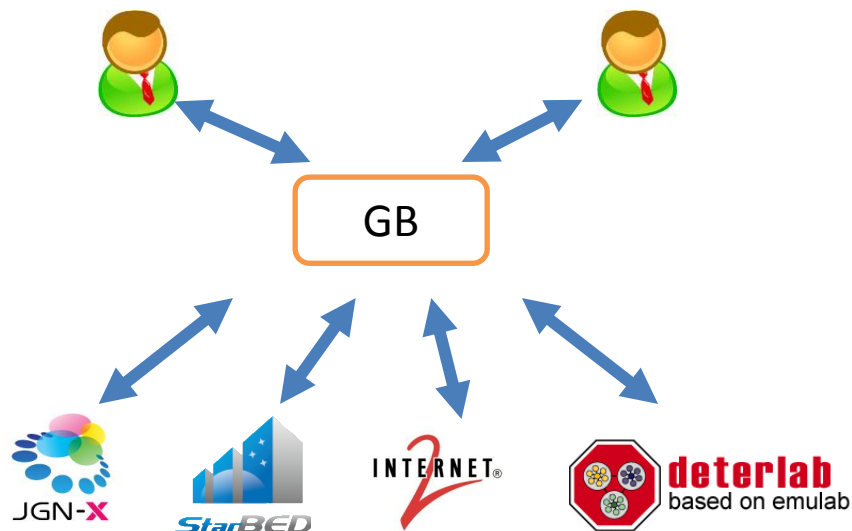
• テストベッド連携

□ 労力が必要な個所

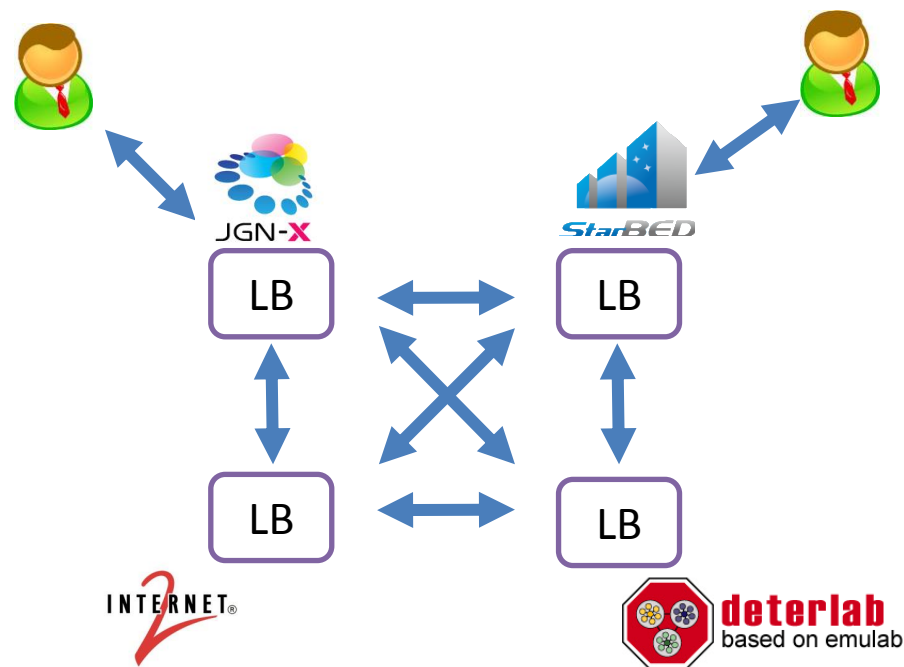
- ユーザ、プロジェクト、リソースの申請
- ユーザ、プロジェクト、リソースの承認
- ユーザ、プロジェクト、リソースの照会
- ユーザ・プロジェクト間のリソース調停
- ユーザ、プロジェクトへの権限付与と
テストベッド間での相互認証

Federation Model

Global Broker Model



Local Broker Model



Pros. & Cons.

- **Global Broker**

- 連携フレームワークを開発しやすい
- アカウンティングしやすい
- プラットフォーム、ポータルが固定されてしまう
- テストベッドごとに異なる機能を持たせにくい

- **Local Broker**

- 各テストベッドのプラットフォーム、ポータルを利用できる
- テストベッドごとに異なる機能を持たせやすい
- 連携フレームワークの開発が大変
- アカウンティング(相互認証)が大変

Local Broker Model で 必要なコンポーネント

- **Testbed API**

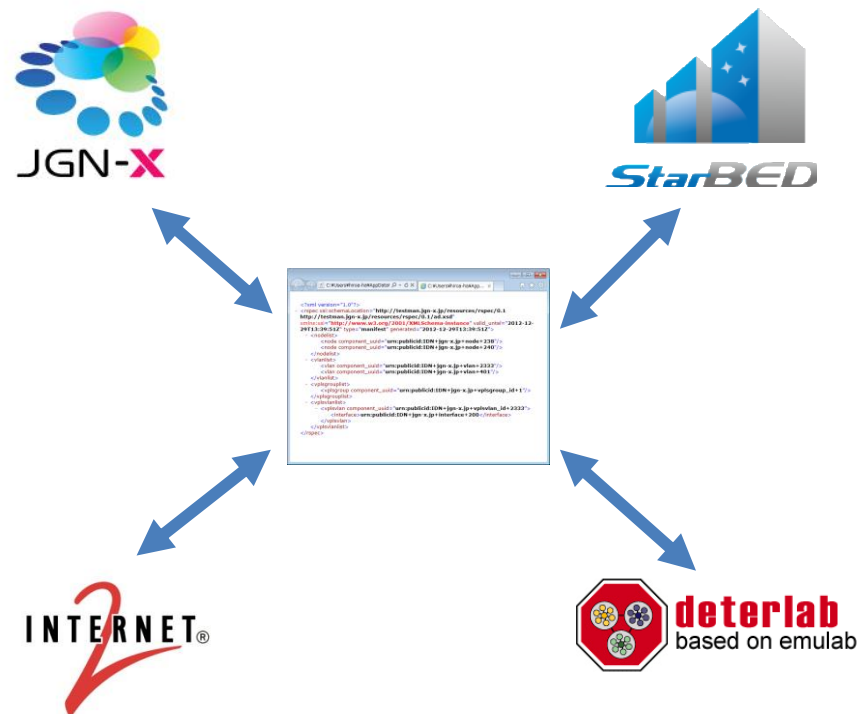
- 何らかのデータベースで資源を管理
- SDN に向けて、何らかのAPI (Testbed API) を整備
- それぞれのテストベッドで独自に開発
 - ✓ StarBED の Spring OS (toolset)、eggrole (API)、K言語
 - ✓ Deterlab の Deterlab tools (API) や topodl

- **共通言語**

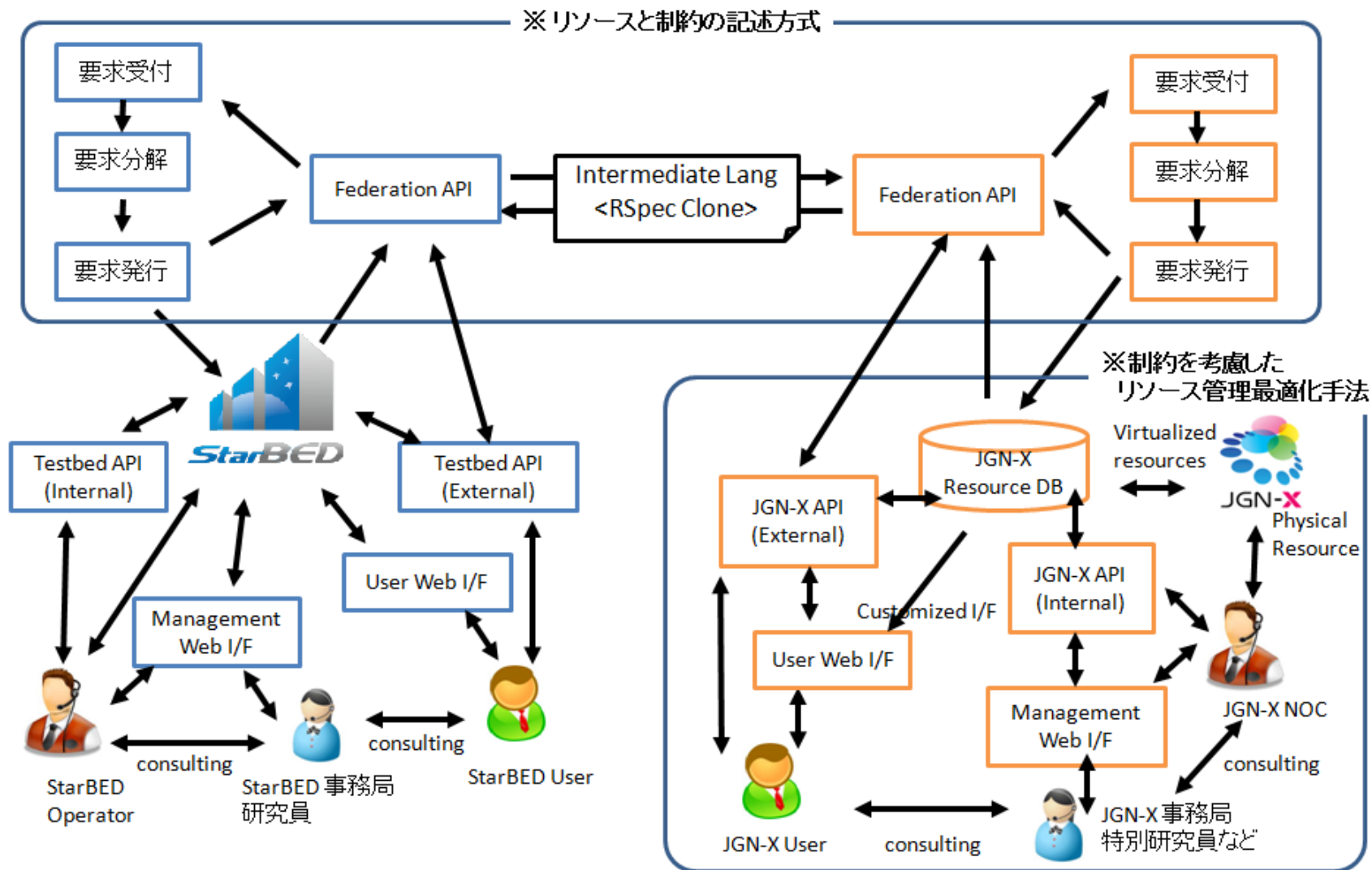
- テストベッドごとに異なるTestbed API を連携、連動させるための共通言語

Testbed API 向け共通言語

- ビジネスの共通言語はひとまず英語を使うのと同じイメージ
 - ローカルの言葉で話せるのであれば、話してもよい
- Testman suite では**GENI** **RSpec**に独自の解釈を加えた利用方法で、予約した資源の確認に焦点を当てて、ひとまず開発を進めている



JGN-X と StarBED による例



Testman-suite

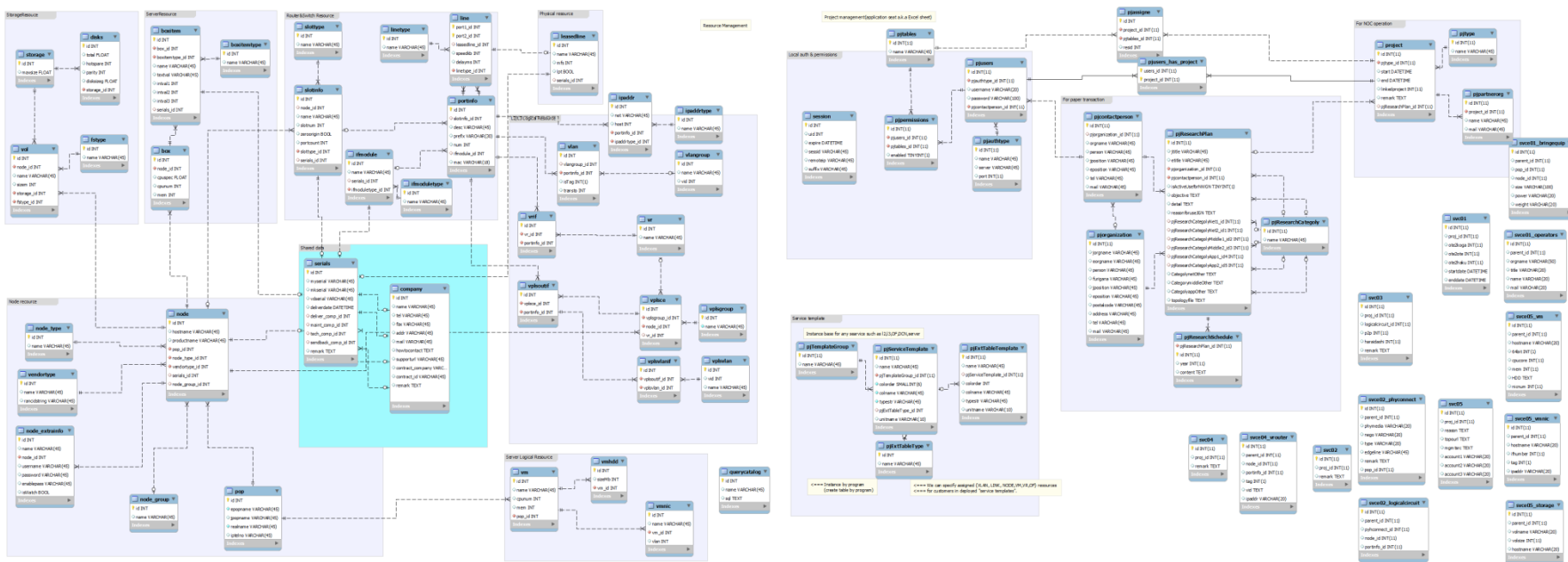
- フェデレーション
 - Local Broker Testbed Federation model
- マネジメント
 - DB oriented Testbed Management
 - Resource / Service Template
- オーケストレーション
 - Testbed API
 - Modularization

DB Oriented Management

- データベースで統合管理しようという流れは、テストベッドだけでなく商用ISP 環境でも進められている
 - 例：NTT コミュニケーションズ MAIKO
- 商用ISP 環境とはやはり違うので、JGN-Xに適したデータベース設計と、運用フレームワークが必要

Testman DB

- 物理リソースだけでなく、仮想リソースも正規化してデータベース登録



作成したJGN-X 資源管理用 MySQL のテーブル

データベースで管理する利点

- 割り当て・未割当、関連プロジェクト、工事の影響範囲プロジェクトの特定などが容易に
- 整合性の確認、不整合の発見が容易に
- UIの拡張やオーケストレーションの実施が容易に
- 外部連携用インターフェースの開発が容易に

Template

- **Resource Template**

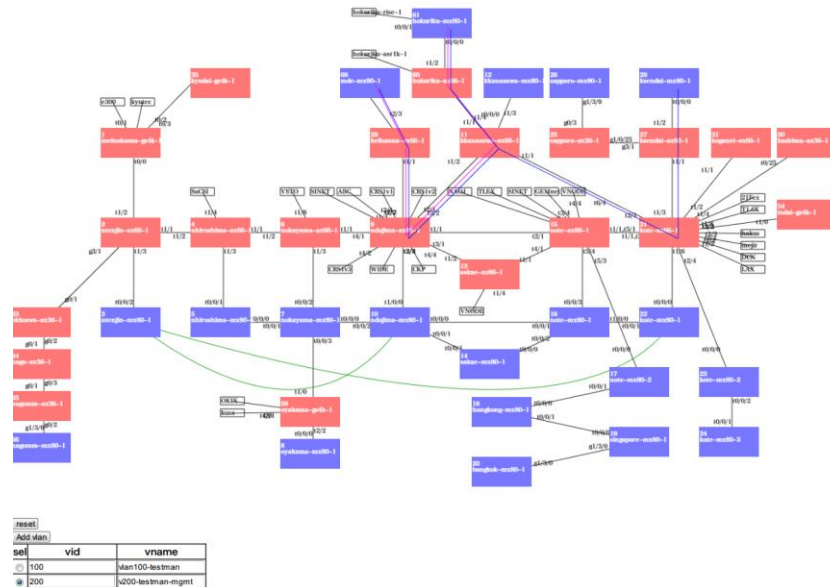
- 新しい機器や技術をリソースデータベースに取り入れるためのテンプレート
- データベースの再設計なしに拡張可能

- **Service Template**

- 新しい機器や技術をサービスとして提供するためのテンプレート
- 利用シーンやユーザの役割に合わせたWeb インターフェースの提供が可能に

Template を用いたWeb UI

- データベースと連動したWeb UI による資源割り当て
- サービステンプレートにより、UIを柔軟に変更可能
 - 利用者用
 - 事務局用
 - NOC用



開発中のVLAN パス割り当てWebUI

Testman-suite

- フェデレーション
 - Local Broker Testbed Federation model
- マネジメント
 - DB oriented Testbed Management
 - Resource / Service Template
- オーケストレーション
 - Testbed API
 - Modularization

オーケストレーション

- **Testbed API**

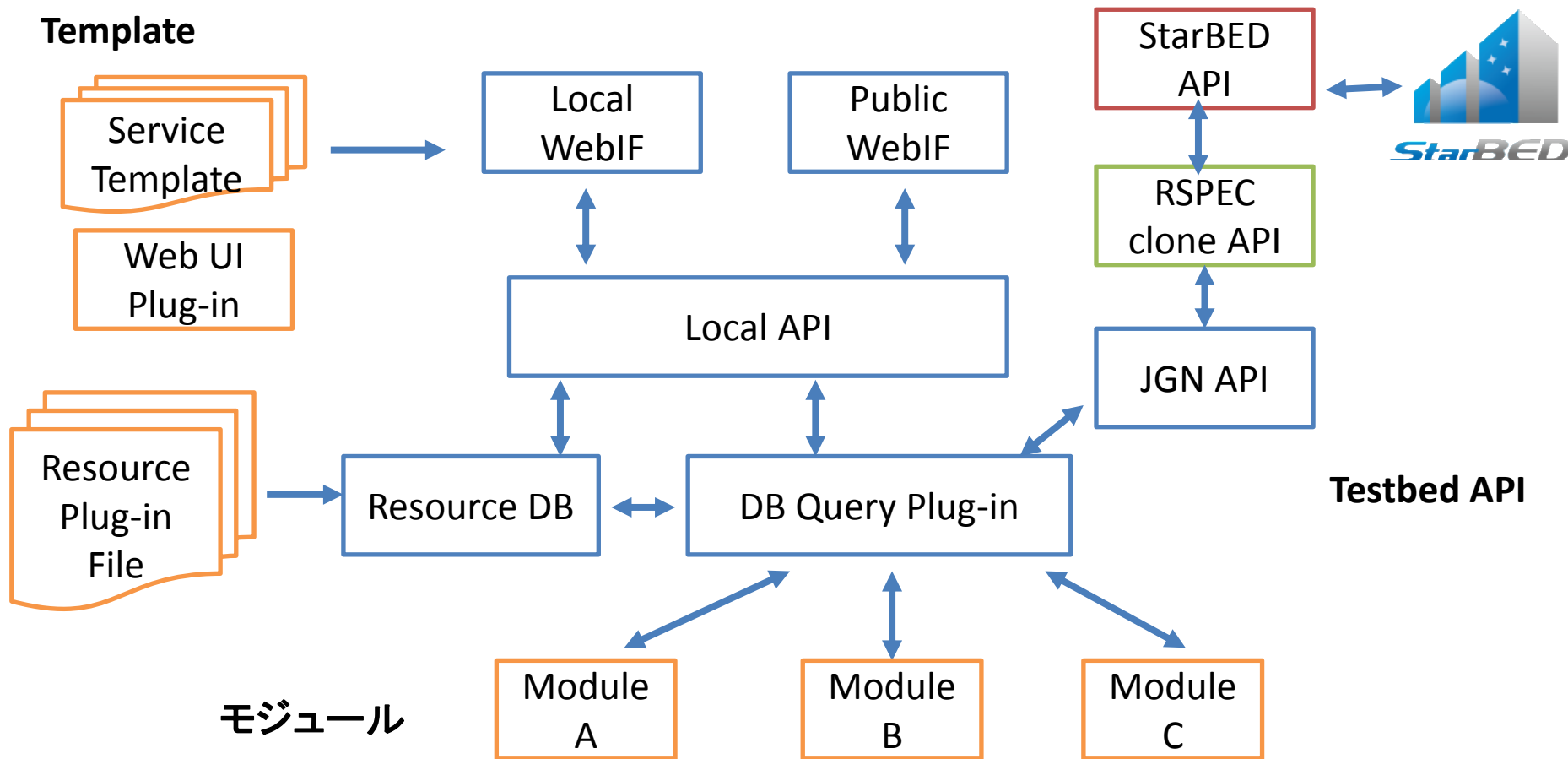
- JGNには今までなかったなので、この機に整備

- ✓ Local APIと連携用 API

- **Modularization**

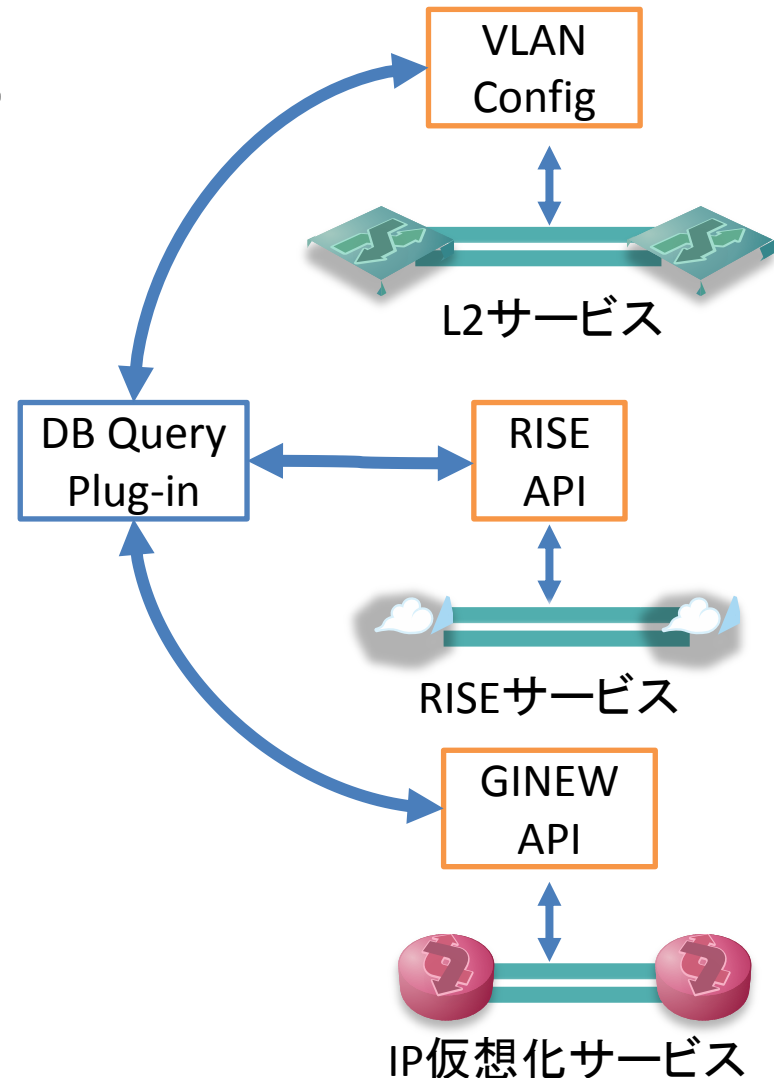
- 運用に必要な自動設定・設定補佐ツールをモジュールとして統合できるようにプラグインテンプレートやAPIを規定

JGN API の全体構成



統合検討中のモジュール

- **RANCID**
 - 設定変更の確認やDBとの不整合の検出
- **VLAN Config**
 - レイヤ2スイッチのVLAN 設定生成ツール
 - レイヤ2サービス設定補佐
- **RISE API**
 - Open Flow North Band API
 - RISE サービス設定補佐
- **GINEW API**
 - JUNOScript / CISCO OnePK 等のNorth Band API
 - IP仮想化サービス設定補佐



Testman suiteの現状

• 鋭意開発中

- 昨年度はプロトタイプを作成
- 今年度はリファクタリングやStarBEDとのテストベッド間認証機能を整備
- 来年度から JGN-X NOC やStarBED Ops を交えた運用実験を実施しながらWeb UI の作りこみ・改修や、他の研究開発プロジェクトとのモジュール連携などを予定

最後に

- **開発中のUIのデモをちょっとだけお見せします**
- **デモの内容**
 - DB から取得したJGN-X のレイヤ2トポロジを Web UI で描画
 - そのWeb UIから、VLAN パスを設定
 - VLAN パス設定から機器へ投入する設定コマンドをWeb UI に出力