

GENERIC VNFM OPENSTACK TACKER

ご紹介

Open Mobile Network Infra Meetup #2

@tj8000rpm

自己紹介

■ ■ な通信会社で NFV 関係のアーキテクツ的ななにかをやっています

辻 広志(@tj8000rpm : Twitter / Qiita / Github)



自称NFV活動家

- OpenStack / Open vSwitch / Ceph / Linux / Ansible
- Python / Golang

本日のお話

- **モチベーション : Generic VNFMである OpenStack Tacker について少しでも興味を持っていただきたい**
- **VNFMの説明・入門（半分以上この話）**
- **Tackerが日系企業を中心に再注目されている背景のご紹介（ニュースリリース、講演資料等に記載の内容の紹介）**

そもそもVNFMってなに？

Virtual Network Function Manager

呼処理ノード等をデプロイするための 仮想化管理コンポーネント

VIM(OpenStack等)とVNF(EPC/5GC等)の中間に位置し**VNFパッケージ**をもとに、VMや仮想NWといった仮想リソースを作成し、VMに設定を投入し、VNFとしてセットアップする。ETSI NFVにて標準化されている。

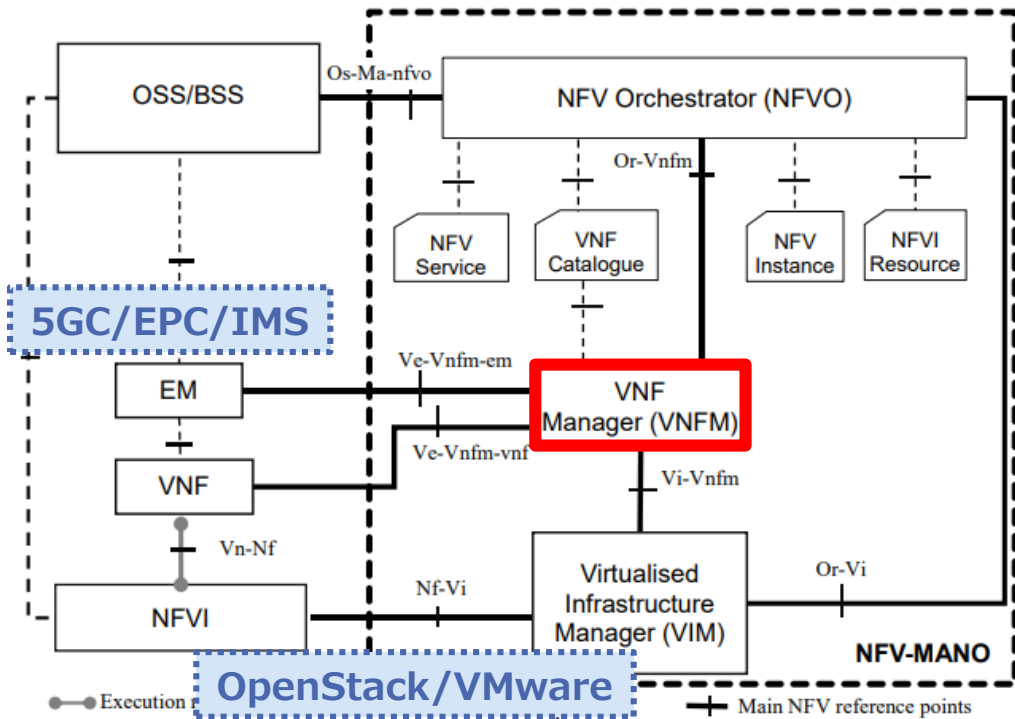
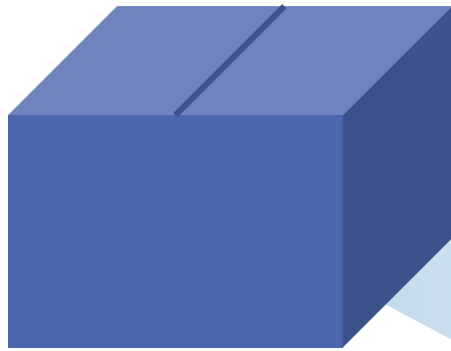


Figure 4.1-1: The NFV-MANO architectural framework with reference points

VNFパッケージ



**CSAR形式
(ZIPアーカイブ)**

- **VNF Descriptor (VNFD)**
VNFの構成情報等を記載
- **VNF 関連資材**
 - VMイメージ
 - 設定ファイル
 - etc

VNFMの役割



Instantiate
VNF
Request

VNFM

VNF
Instantiation

```
my_ims = new IMS(input_params);
```

VNF Instance IMS

HSS

IBCF

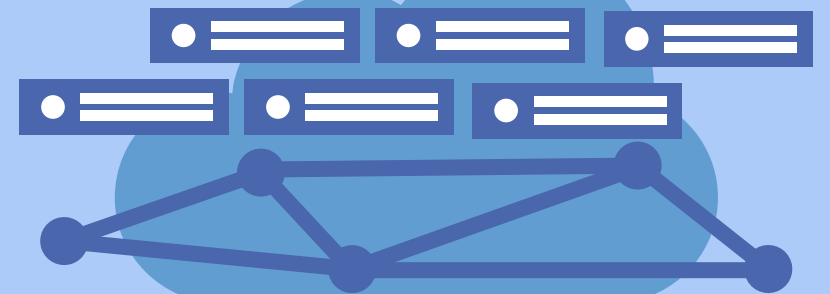
I-CSCF

SLF

P-CSCF

S-CSCF

仮想化環境(VIM/NFVI)



VNFMの役割



Instantiate
VNF
Request

VNFM

VNF
Instantiation

```
my_5gc = new Core5G(input_params);
```

VNF Instance 5GC

UDM

UDR

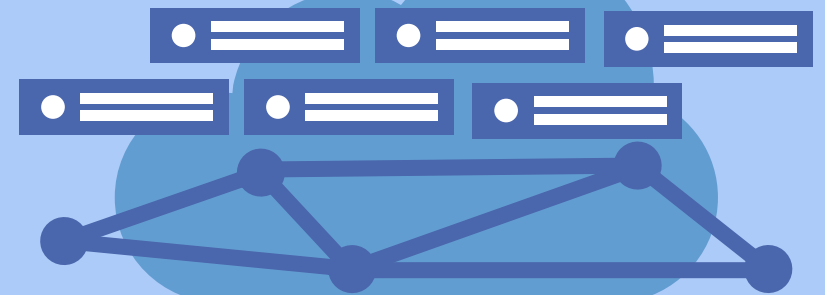
AMF

PCF

SMF

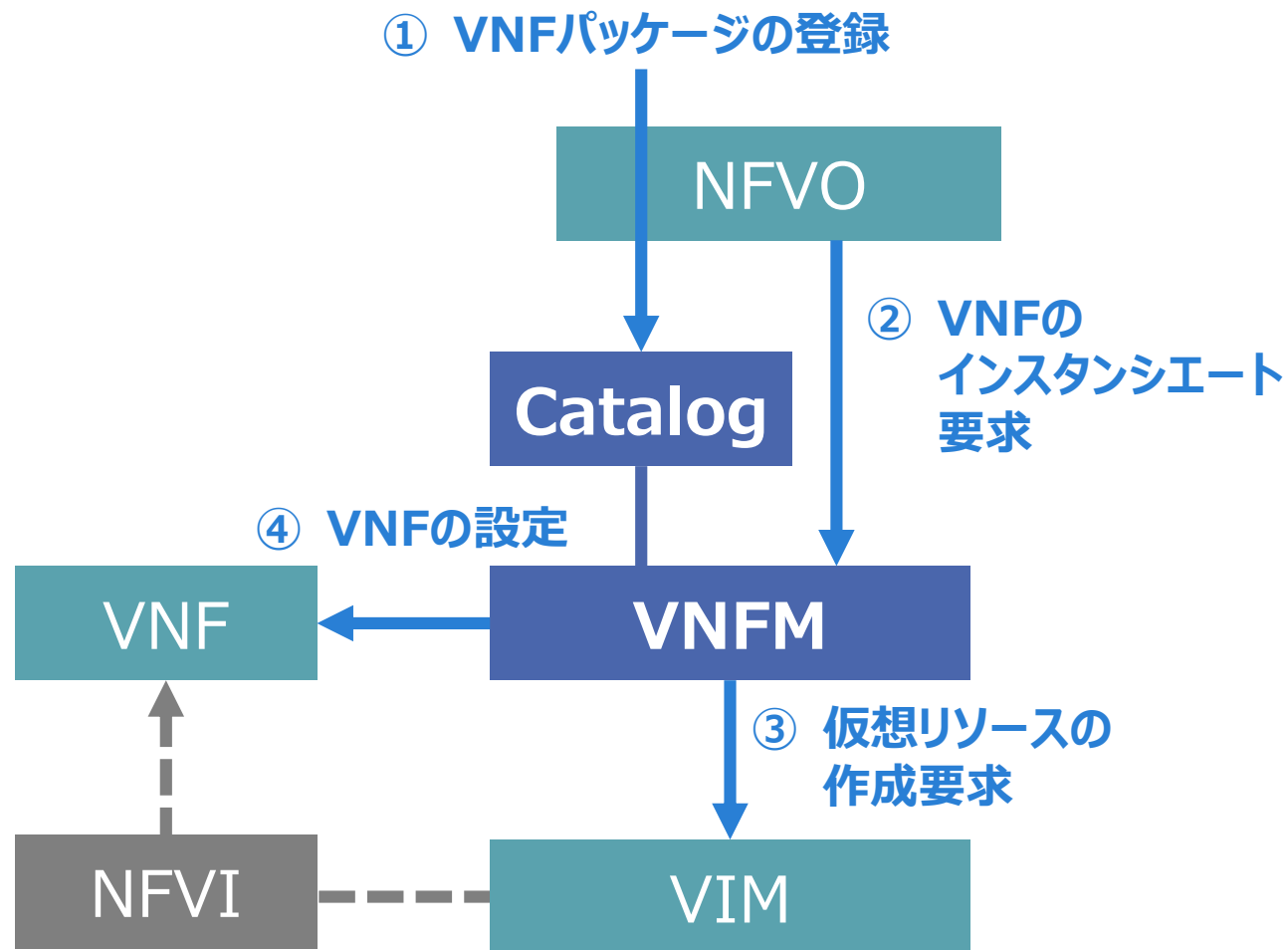
UPF

仮想化環境(VIM/NFVI)



VNFMの動き

Instantiation(Grant/Reservation無し)



- ①**VNFパッケージの登録**
パッケージを事前に登録
- ②**VNFのインスタント要求**
VNFの作成をAPI経由で要求
- ③**仮想リソースの作成要求**
VIM経由でVM等を作成
- ④**VNFの設定**
VIMが作成したVMにVNFMから設定を投入

なぜVNFMが必要になるのか

=やりたいことはテレコ版 IaC

- 複雑でセット単位での増設が多い

例えば東西にひとつずつ配置するなど

- 増設コストを抑えたい

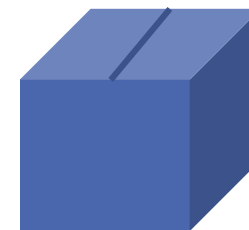
人的・時間的稼働の低減

- オペレーションの自動化をしたい

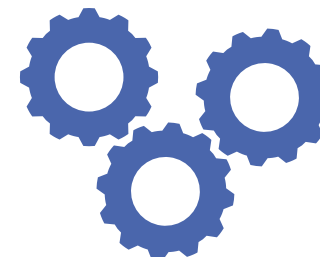
オートヒーリング・スケーリングを定義

- なるべくベンダや特定技術に依存したくない

機能要件やAPIをETSIで標準化



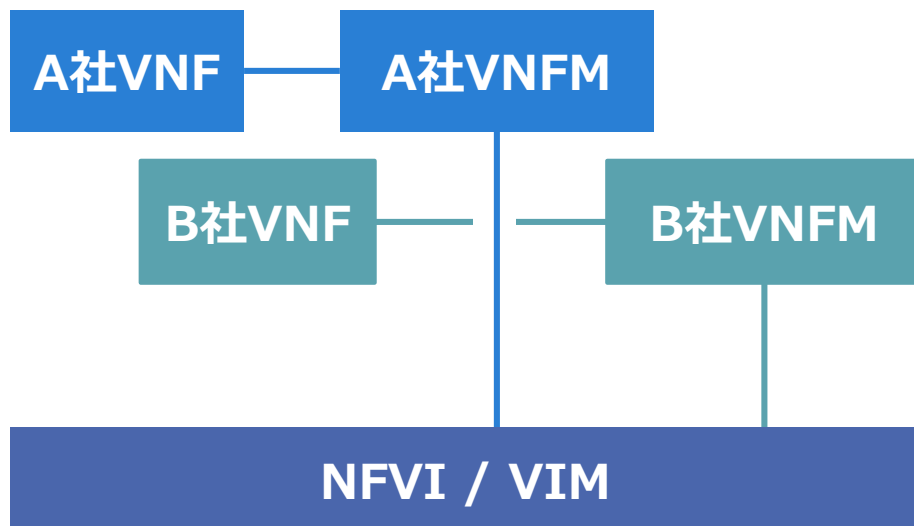
テンプレート化



標準化

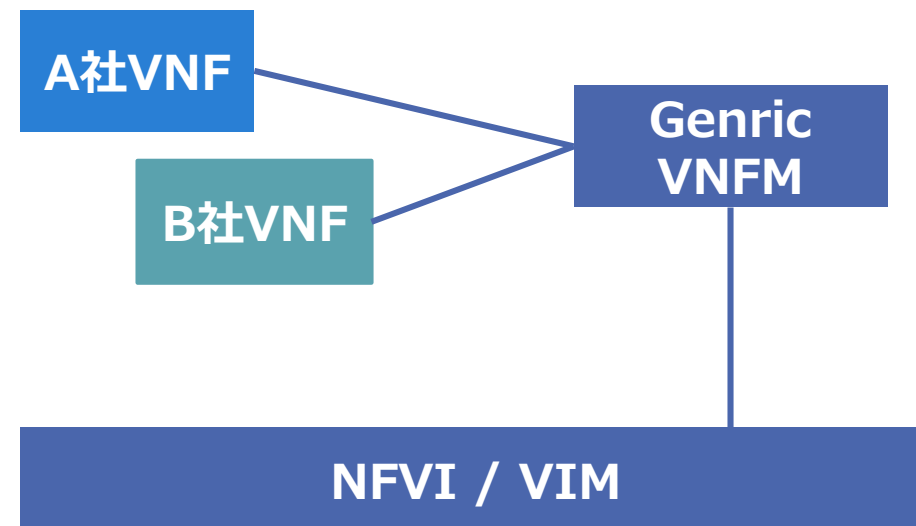
Generic VNFMとは

Generic でない VNFM



- VNF/VNFM間のインテグレーションは比較的容易
- VNFMがVNF（ベンダ）ごとに増加
VNF/VIM間のインテグ・試験コスト増

Generic VNFM



- 構成がシンプル、管理コンポーネントが少ない
- VNF/VNFM間の仕様整合コストが増

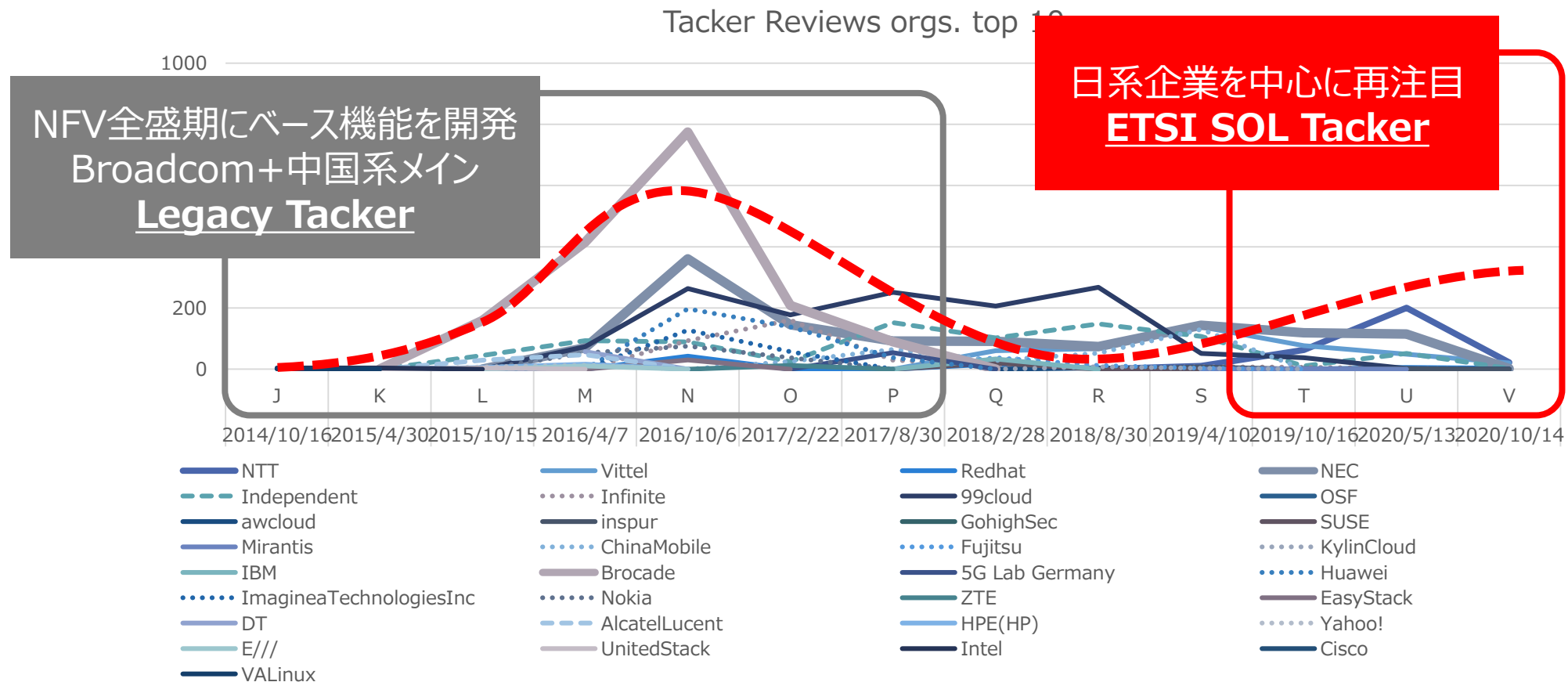
OpenStack Tackerとは



Tacker is an OpenStack service for NFV Orchestration with a general purpose VNF Manager to deploy and operate Virtual Network Functions (VNFs) and Network Services on an NFV Platform.

It is based on ETSI MANO Architectural Framework.

いまさらTacker ?

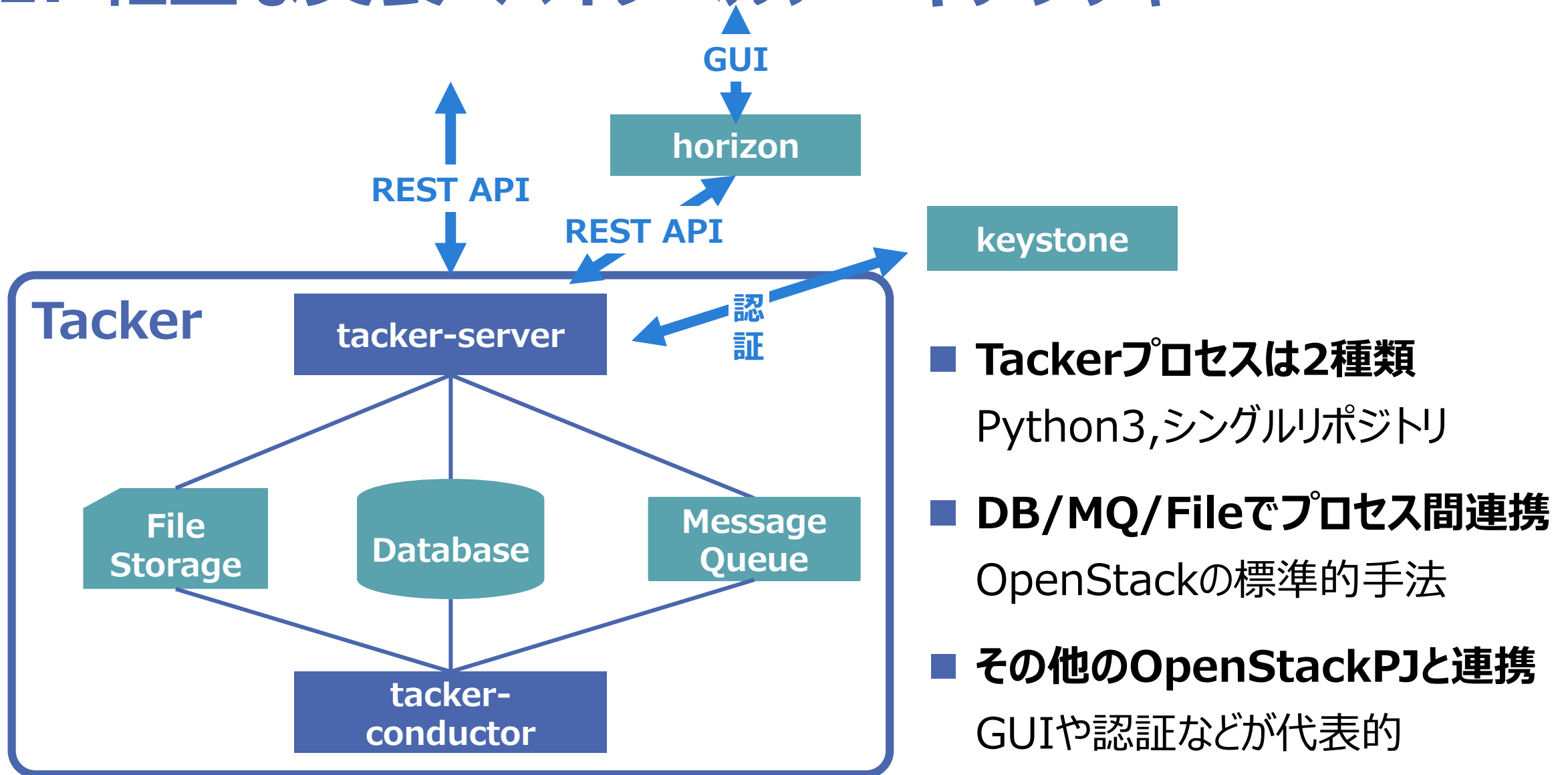


いまTackerが注目されている理由

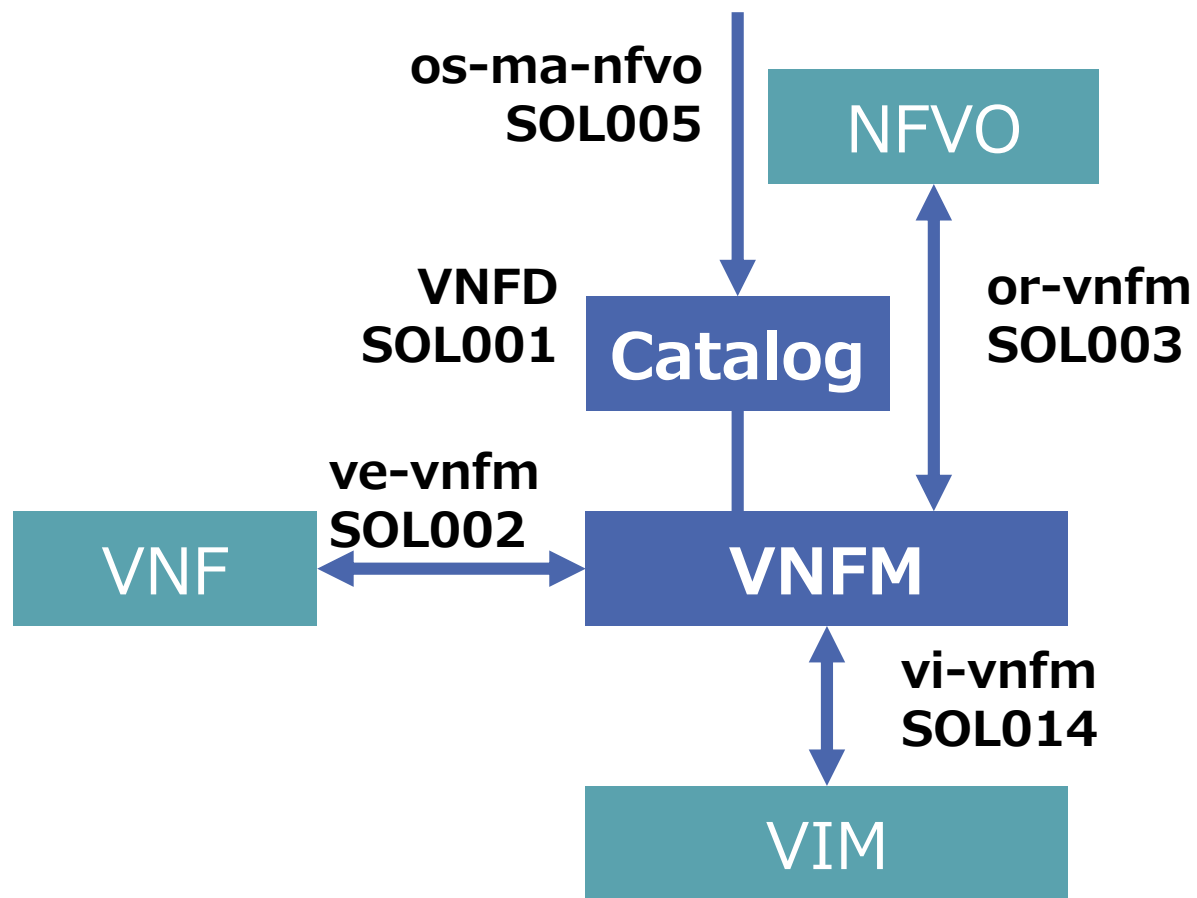
- 1. 軽量な実装
- 2. ETSI NFV 標準準拠
- 3. VNFMインテグレーションの容易さ

NTT、NTTドコモ、NEC、KDDI等が発表している資料※等をもとに

1. 軽量な実装 ハイレベルアーキテクチャ



2. ETSI NFV 標準準拠



■ SOL001

パッケージ形式、TOSCA VNFD
読み込みに対応

■ SOL002/SOL003

VNF LCMに対応

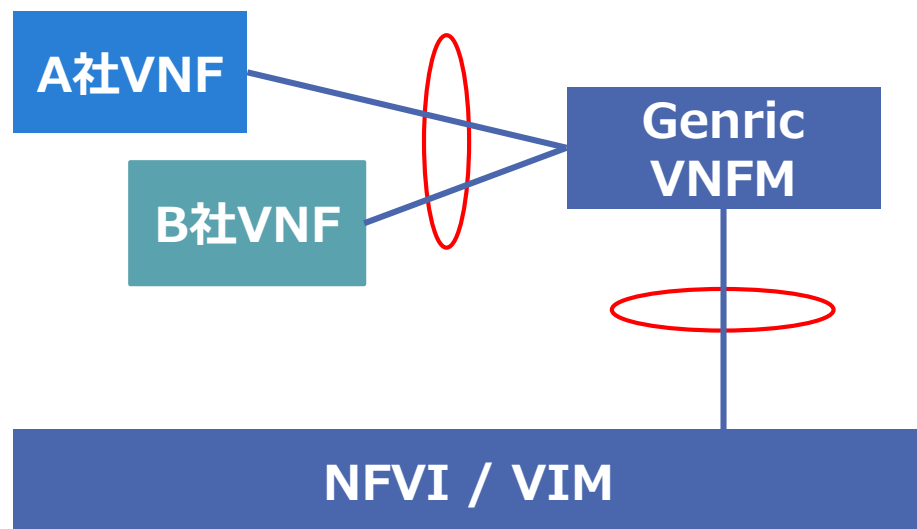
■ SOL005

VNFパッケージ管理の
基本的なCRUD操作に対応

■ SOL014

OpenStack Heatを使用し、
概念に対応（仕様が最近決まったため）

3. VNFMインテグレーションの容易さ インタフェース整合が課題

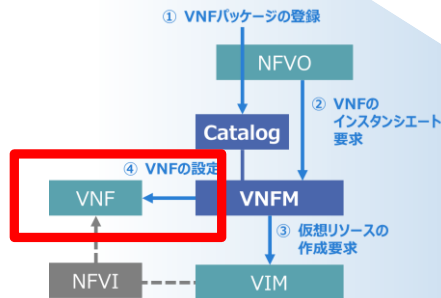


- VNFごとのカスタマイズが避けられない
- VNFから容易に試験できるか

VNFごとに異なる要件の吸収が課題

3. VNFMインテグレーションの容易さ Tackerの解決策

- VNFごとのカスタマイズが避けられない



- VNFから容易に試験できるか

HOTを使用した vi-vnfm
デファクト的なHeatをVNFパッケージ内で
使用することでVNF固有要件をHeatで吸収

ただし、VNF設定機能周りはまだ不十分

オープンソースであるTackerだと
また比較的導入が簡単 (Devstack)
ベンダさんの環境でも利用できる

まとめと課題

TackerはOSSで、軽量なGeneric VNFM

比較的シンプルな構成かつ、少ないソースコード量



軽量でETSI準拠なAPI、柔軟なインタフェースが魅力

複数ベンダ間でのインテグレーションが容易

日系企業が注目し進化している

NTT/ドコモ/NEC/KDDIなどが注目し、実装が進んでいる



VNFへの設定投入機能が弱い

VNFMのもう一つの軸である、VMの設定機能が貧弱、
コミュニティ最新版でもAnsible等での設定を実施するには
相当のカスタマイズが必要

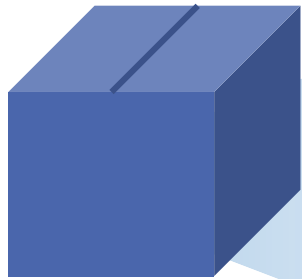
THANK YOU

End of Presentation

参考文献

- **NTT DOCOMO's Operational Challenges of Commercial Multi-vendor NFV System**
<https://object-storage-ca-ymq-1.vexxhost.net/swift/v1/6e4619c416ff4bd19e1c087f27a43eea/www-assets-prod/summits/27/presentations/23901/slides/OISShanghai-NTTDOCOMO-challenge-NFV.pdf>
- **NTT and KDDI Challenges for Sustainable Infrastructure Transformation**
<https://superuser.openstack.org/articles/ntt-and-kddi-challenges-for-sustainable-infrastructure-transformation/>
- **KDDIとNEC、固定通話の事業者間IP接続に向け「OpenStack Tacker」を利用した仮想ネットワーク管理機能を開発**
<https://news.kddi.com/kddi/corporate/newsrelease/2020/10/26/4735.html>

やりたいことはTelcoのIaC



テンプレート

VMやNWの抽象化された情報

ポリシー

自動オペレーションの発動契機

テンプレート化

API等のインターオペラビリティ

異なるコンポーネント間での互換性

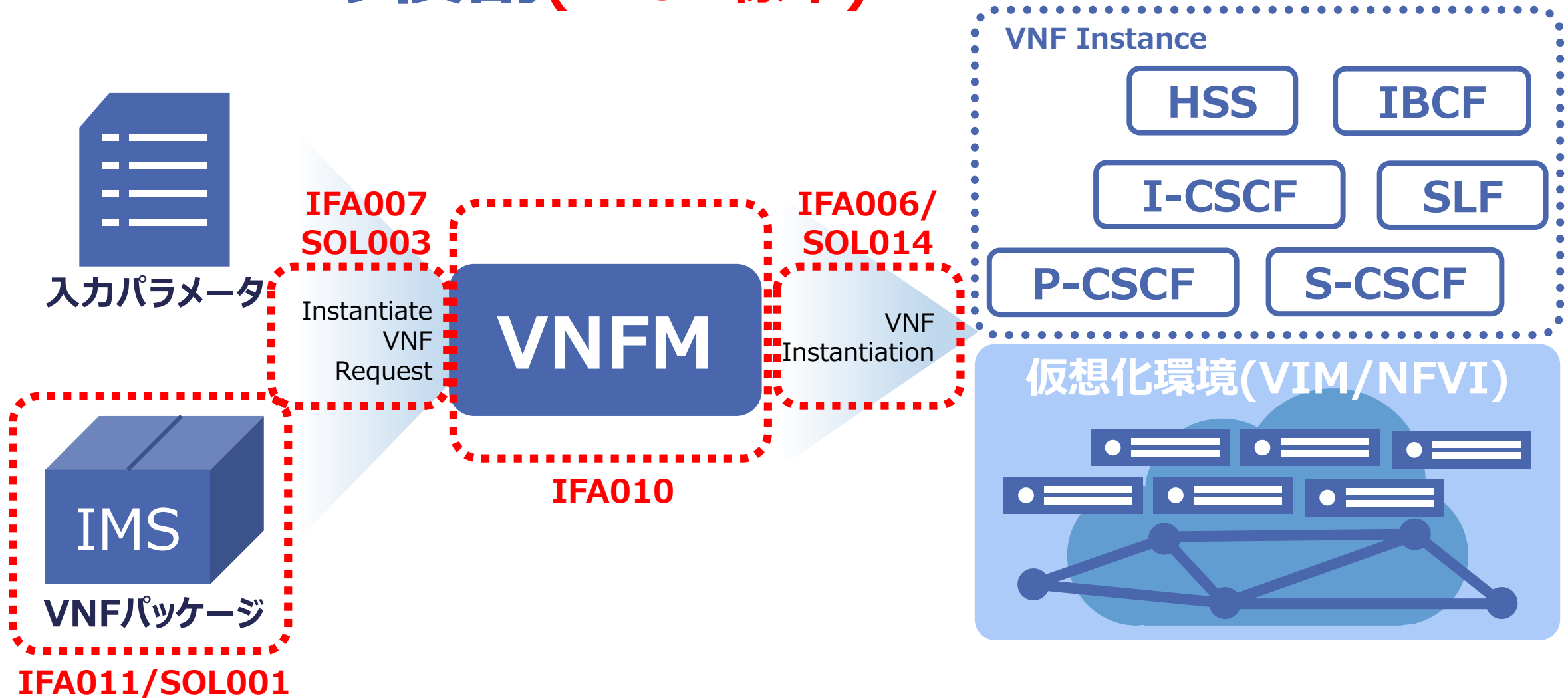
プロセスの統合

標準化されたやり方

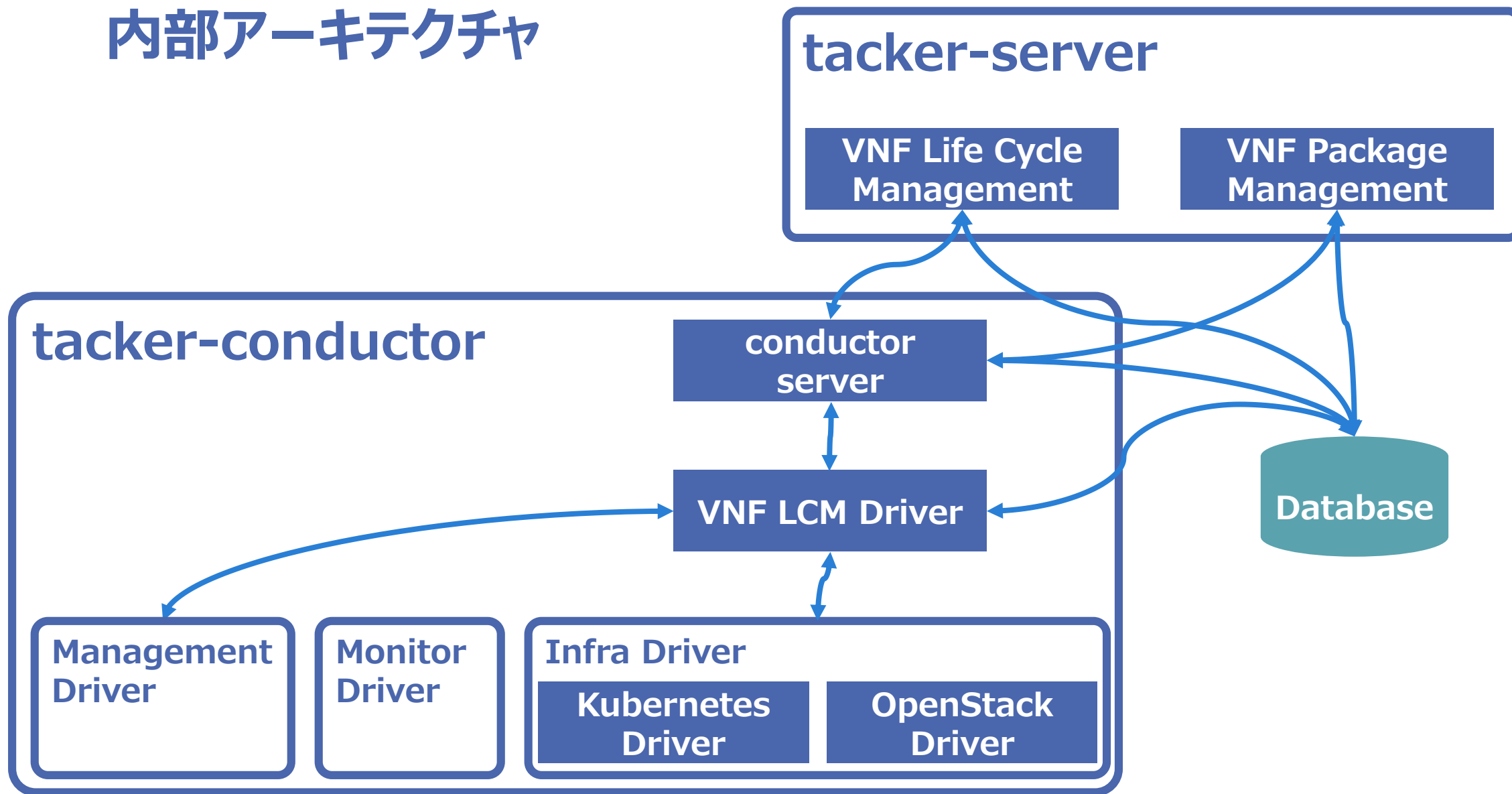


標準化

VNFMの役割(ETSI 標準)



1. 軽量な実装 内部アーキテクチャ



1. 軽量な実装 ハイレベルアーキテクチャ(2)

