

JANOG43

システム構成変更自動化技術Theaterによる ネットワーク構築実験のご紹介

2019年1月24日

日本電気株式会社

システムプラットフォーム研究所

〇八 鋤 豊、福田 達也、黒田 貴之、佐藤 陽一

Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

目次

- サマリ
- 背景
- システム構成変更自動化技術Theater
- 実験内容
- 考察

目次

- サマリ
- 背景
- システム構成変更自動化技術Theater
- 実験内容
- 考察

■ 実施したこと

- 弊社のシステム構成変更自動化技術/ツール “Theater” を用いて EVPN/VXLAN環境を自動構築しました

■ 議論させていただきたい内容

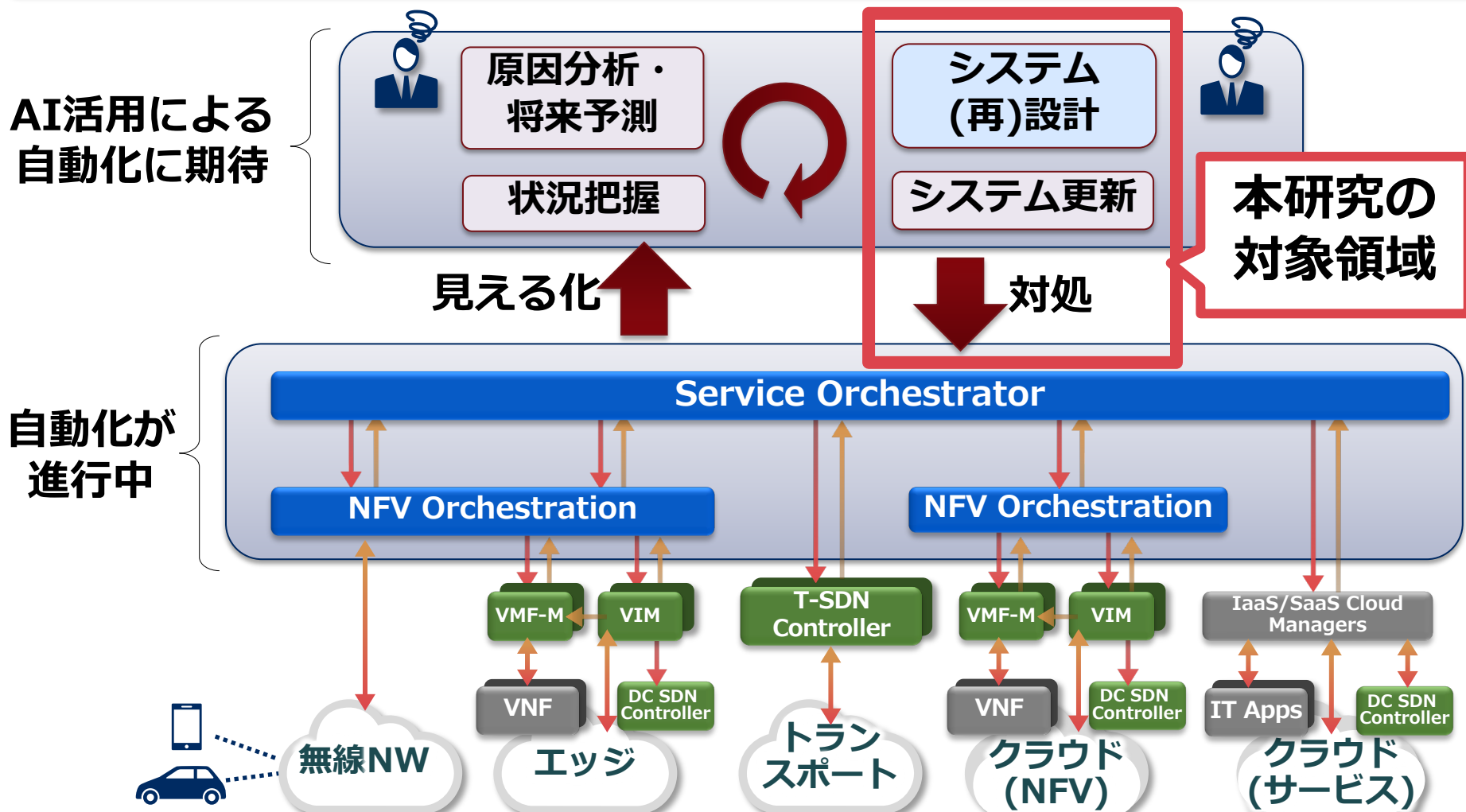
- Theater の有効性
- Theater が使えそうな領域
- Theater を使えるものにするために改善すべき点
- ネットワークの新規構築や変更で困っている点
- 上記困りごとにかかっている工数
- 上記困りごとに対する現在の工夫

目次

- サマリ
- 背景
- システム構成変更自動化技術Theater
- 実験内容
- 考察

仮想NWサービスの運用自動化に向けたチャレンジ

- ・ 仮想化/オーケストレータの効果はインフラの簡素化/統一化まで (知的作業は人力)
- ・ 知的作業のうち“見える化”へのAI活用が進む中、“対処”の自動化が課題



研究ビジョン・ロードマップ

1. [将来ビジョン]

- 利用者が曖昧なビジネス要件を入力すると、その具体化をAIが支援
- NWだけでなくアプリケーションを含めたシステム全体を生成・構築

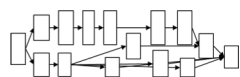
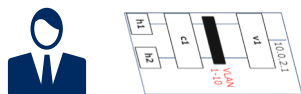
2. [中期ビジョン] NW要件に基づくNWの自動構築

- 専門化が求められるNW要件を明確に記述
- 要件を満足するNW構成を生成・構築

3. [現在の取り組み] NW構成モデルに基づくNWの自動構築

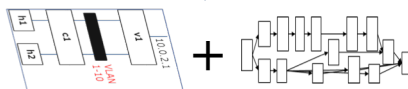
- 専門化が望ましいNW構成や設定を明確に記述
- 入力されたNW構成を実現するための作業Workflowを生成・実行

管理者が構成を与える



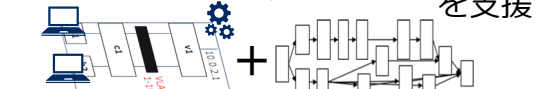
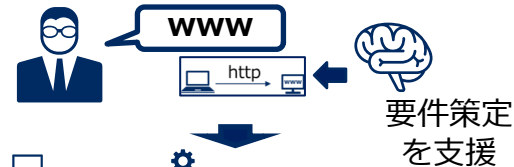
手順を自動生成

管理者が要件を与える



NW構成/手順を自動生成

利用者が要件を与える



システム全体を自動構築

現在の取り組み

中期ビジョン

将来ビジョン

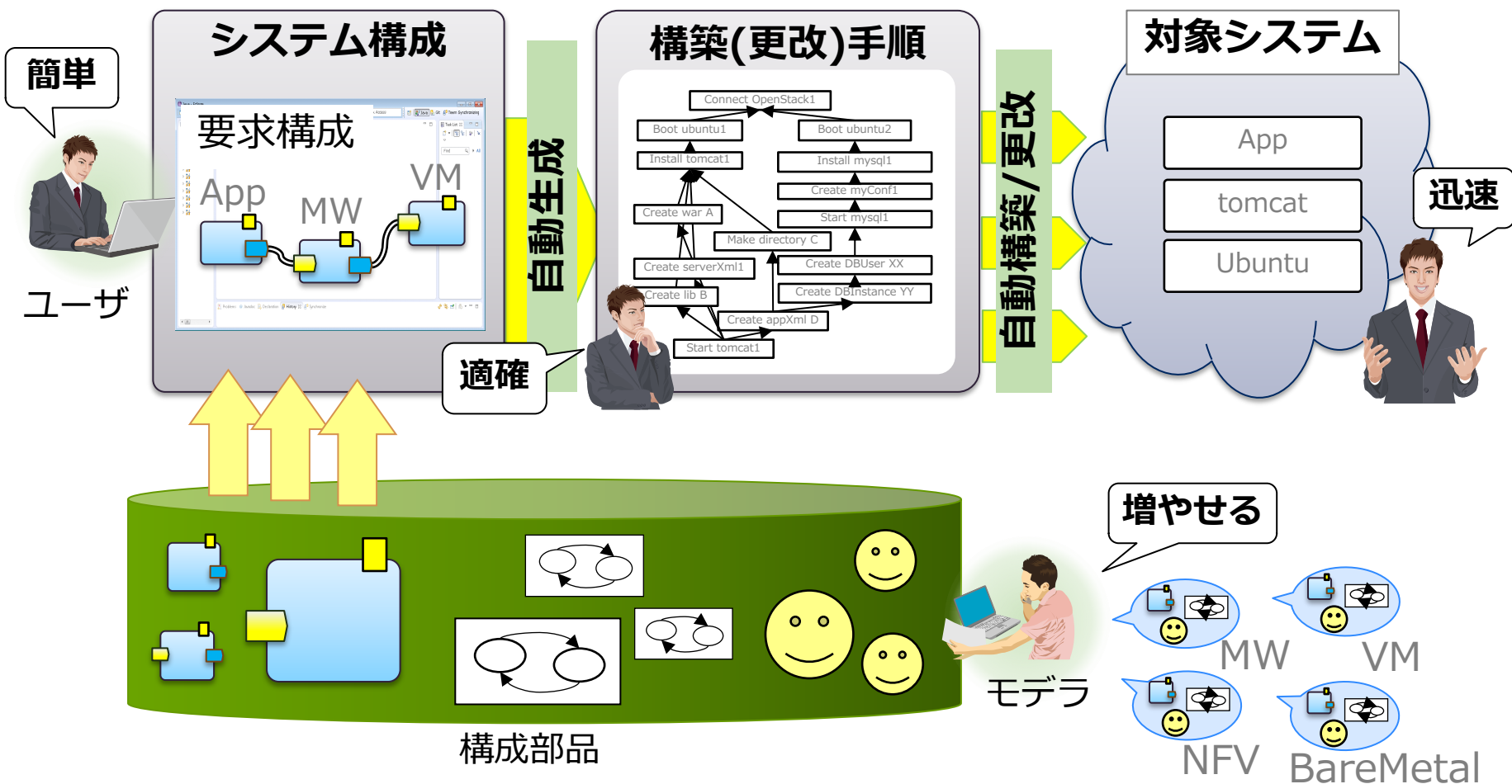
目次

- サマリ
- 背景
- システム構成変更自動化技術Theater
- 実験内容
- 考察

Theater (システム構成変更自動化技術)

所望の構成を入力すると、構築手順が生成され、システムを自動構築できる

- ・ 構成は部品の組み合わせで簡単に定義できる
- ・ 部品を追加して制御対象領域を拡張できる



目次

- サマリ
- 背景
- システム構成変更自動化技術Theater
- 実験内容
- 考察

実施したこと & 目的

■ 実施したこと: TheaterでEVPN/VXLAN環境を構築する

■ 目的

- モデラのモデル部品作成作業の実現性を確認
- ユーザのシステム定義作業の容易さを確認

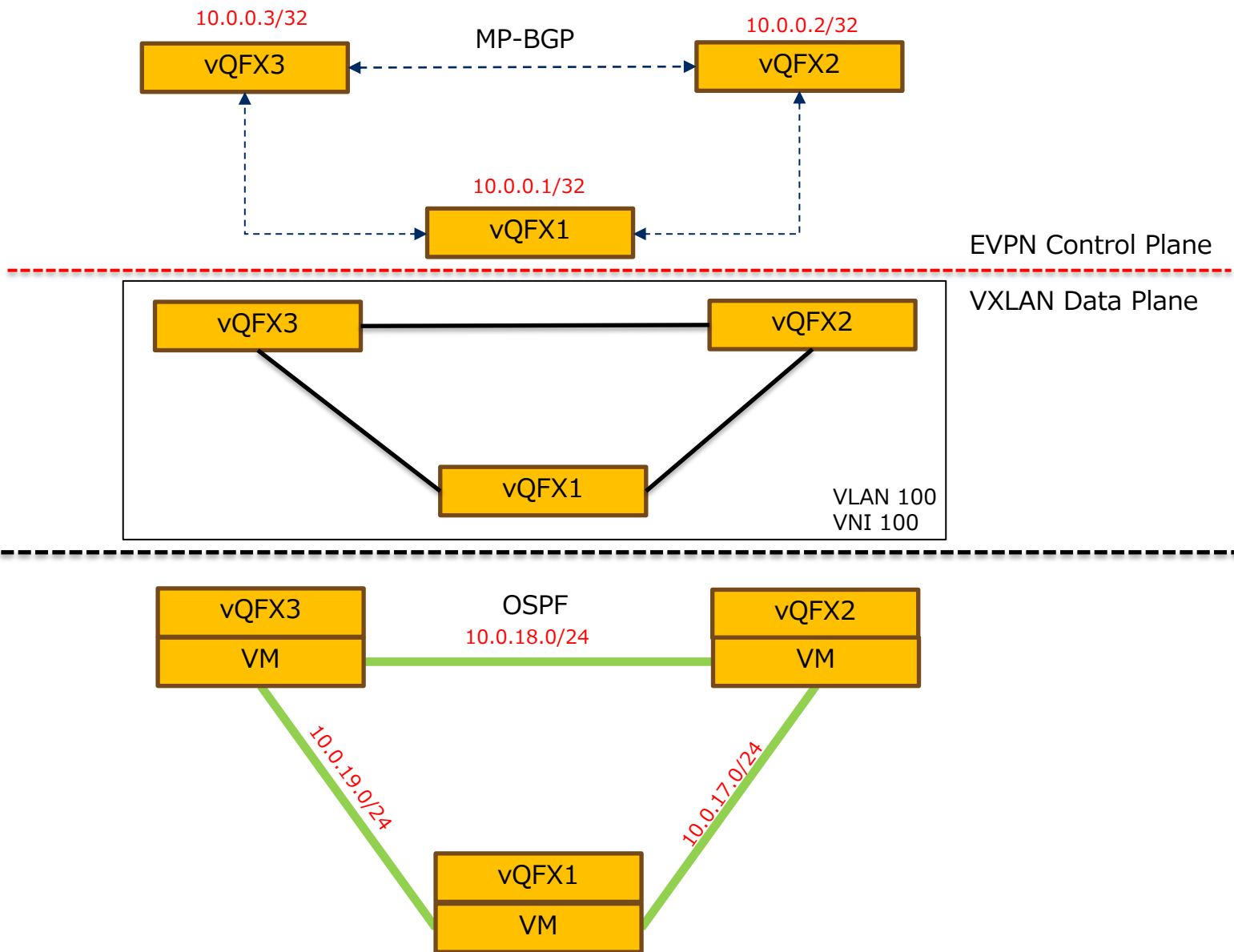
■ 初期構築

- スイッチ本体の設定
- Underlay: OSPF設定
- Overlay: EVPNコントロールプレーン設定
- Overlay: VXLANデータプレーン設定

■ 構成変更

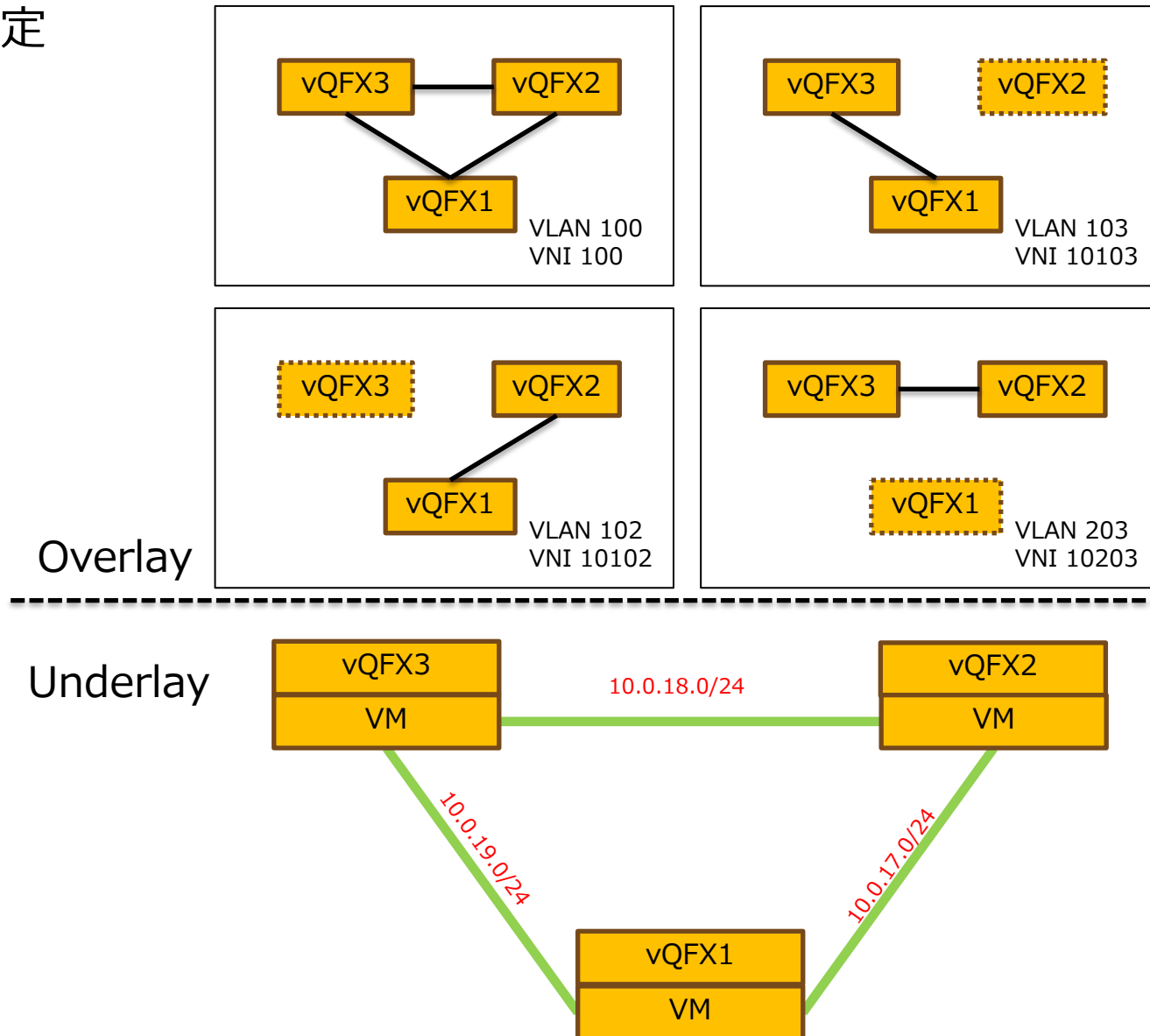
- Overlayへテナントを追加

初期構築したEVPN/VXLAN構成



構成変更: Overlayへテナントを追加

新規VNIの設定



Theaterに関わる作業

■ モデル部品の作成（モデラの作業）

- スイッチ本体設定用の部品
- OSPF設定用の部品
- EVPNコントロールプレーン設定用の部品
- VXLANデータプレーン設定用の部品

■ システム構成の定義（ユーザの作業）

- テナントの初期構成の定義
- テナント構成変更後の定義

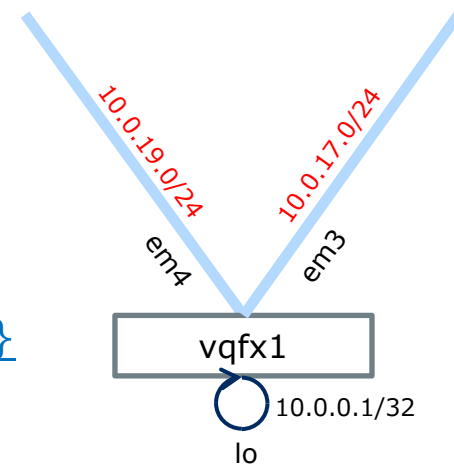
スイッチ本体の設定

ホスト名の設定

- set system host-name {hostname}

インタフェース設定

- set interfaces {interface} unit {num} family {inet} {address}
- アドレスについて
 - ・データプレーンは図の通り
 - ・loは10.0.0.{deviceID}/32
- unit番号:IFの論理番号。物理IFには0を設定するのが通例
 - ・VLAN-IFを利用する場合、VLAN-IDに合わせる



Syslog設定

- set system syslog user * any emergency
- set system syslog file messages any notice
- set system syslog file messages authorization info
- set system syslog file interactive-commands interactive-commands any

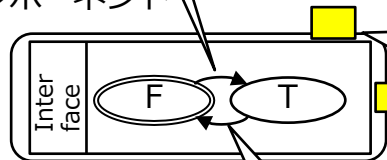
スイッチ本体の設定のTheaterモデル

状態モデル

インタフェースの状態モデル

```
set interfaces {interface} unit {num} family {inet} {address}
```

Interface
コンポーネント



プロパティ:

- ・ インタフェース名
- ・ unit番号
- ・ アドレスファミリ
- ・ IPアドレス

インタフェース情報の交換

```
delete interfaces {interface} unit {num} family {inet} {address}
```

システム設定の状態モデル

```
set system host-name {hostname}  
set system syslog user * any emergency  
set system syslog file messages any notice  
set system syslog file messages authorization info  
set system syslog file interactive-commands  
interactive-commands any
```

System
コンポーネント

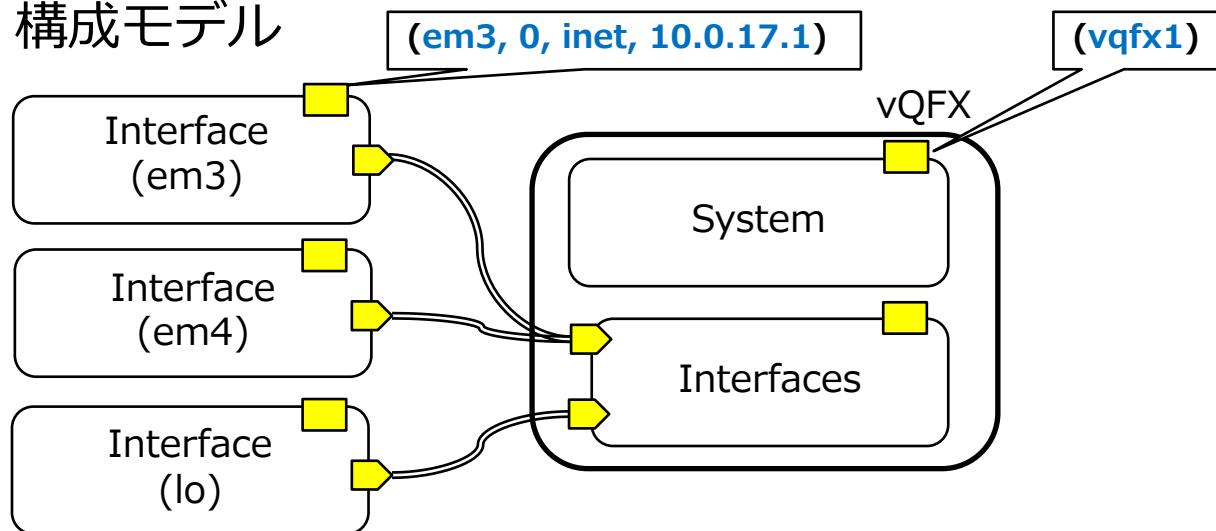


プロパティ:
・ ホスト名

青字: プロパティ

赤字: ポートで他コンポーネントを参照

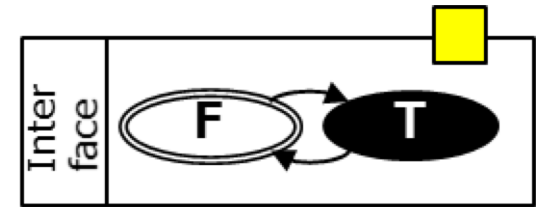
構成モデル



状態要素：システム構成要素の最小単位の振る舞い

-  状態要素 (Element)
 - 変更対象となるシステム構成要素の単位
-  状態 (State)
 - 状態要素が取り得る状態
-  現在状態 (Current State)
 - 状態要素の現在の状態
-  要求状態 (Desired State)
 - 状態要素の変更後に要求される状態
-  状態遷移 (Transition)
 - 状態要素ごとの可能な状態遷移、及びこれを実現するタスク定義
-  プロパティ (Property)
 - 状態遷移タスクに与えるプロパティ

状態要素の図による表記例



あるInterfaceの状態を示す状態要素の例. 取り得る可能な状態と状態遷移が示されている. 現在 F 状態であり、T 状態が要求されている.

OSPF設定

アンダーレイネットワーク

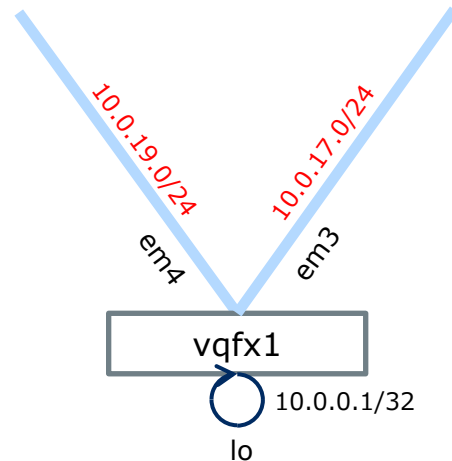
- BGPで使用するループバックアドレスの広告に必要
- Metric:
 - loは1,データプレーンには共通で10を振る

青字: プロパティ

赤字: ポートで他コンポーネントを参照

OSPF設定

- set protocols ospf area 0.0.0.0 interface {interface} metric {metric}



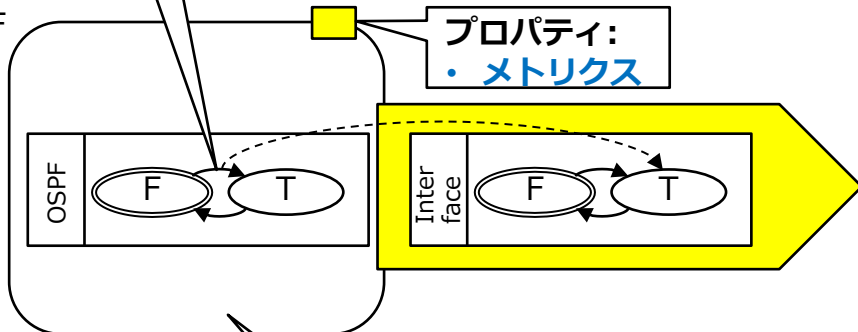
OSPF設定のTheaterモデル

状態モデル

```
set protocols ospf area 0.0.0.0 interface {interface} metric {metric}
```

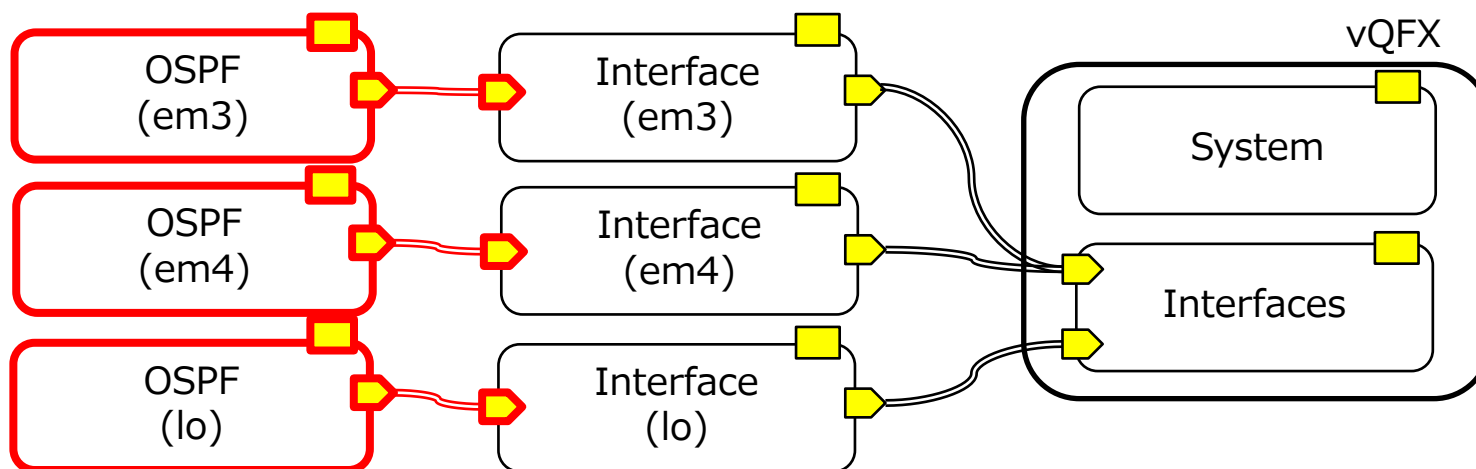
OSPF

プロパティ:
・ メトリクス



```
delete protocols ospf area 0.0.0.0 interface {interface} metric {metric}
```

構成モデル



EVPNコントロールプレーン設定

ルーティング設定

- set routing-options router-id {10.0.0.1}
- set routing-options autonomous-system {65000}

EVPNのMP-BGP設定

- set protocols bgp group {ibgp} type internal
 - set protocols bgp group {ibgp} local-address {10.0.0.1}
 - set protocols bgp group {ibgp} neighbor {10.0.0.2} description {vqfx2}
 - set protocols bgp group {ibgp} neighbor {10.0.0.2} family evpn signaling
 - set protocols bgp group {ibgp} neighbor {10.0.0.3} description {vqfx3}
 - set protocols bgp group {ibgp} neighbor {10.0.0.3} family evpn signaling

EVPNのプロトコル設定

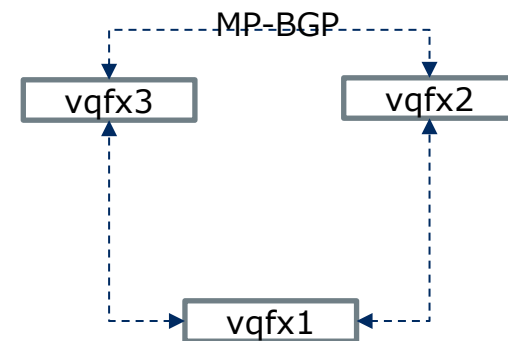
- set protocols evpn encapsulation vxlan
- set protocols evpn extended-vni-list all
- set protocols evpn multicast-mode ingress-replication

EVPNのスイッチ設定

- set switch-options vtep-source-interface lo0.0
- set switch-options route-distinguisher {10.0.0.1}:{65000}
- set switch-options vrf-import {evpn-import}
- set switch-options vrf-target target:{65000}:{65000}

コミュニティ設定

- set policy-options community {vnidefault} members target:{65000}:{65000}
- set policy-options policy-statement {evpn-import} term {vnidefault} from community {vnidefault}
- set policy-options policy-statement {evpn-import} term {vnidefault} then accept
- set policy-options policy-statement {evpn-import} then reject



青字: プロパティ

赤字: ポートで他コンポーネントを参照

EVPNコントロールプレーン設定の状態モデル抜粋

状態モデル

● Routing Option

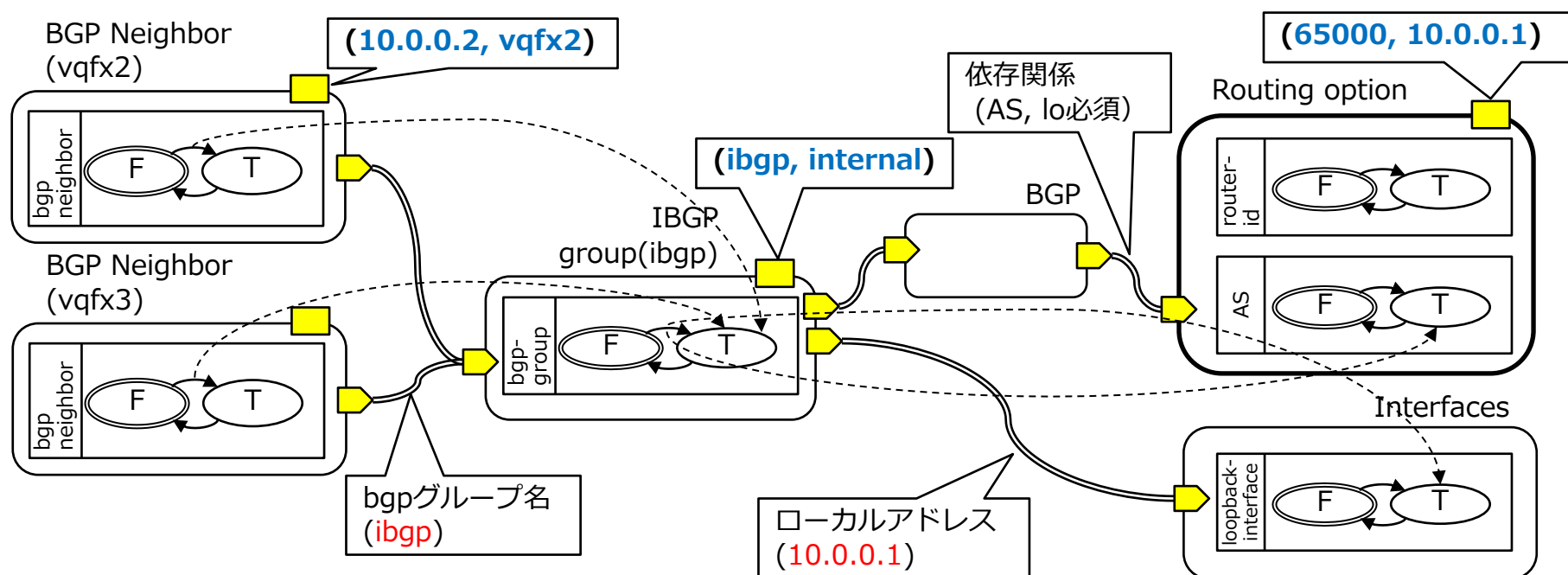
- set routing-options router-id {10.0.0.1}
- set routing-options autonomous-system {65000}

● BGP

- iBGPとRouting optionの橋渡し

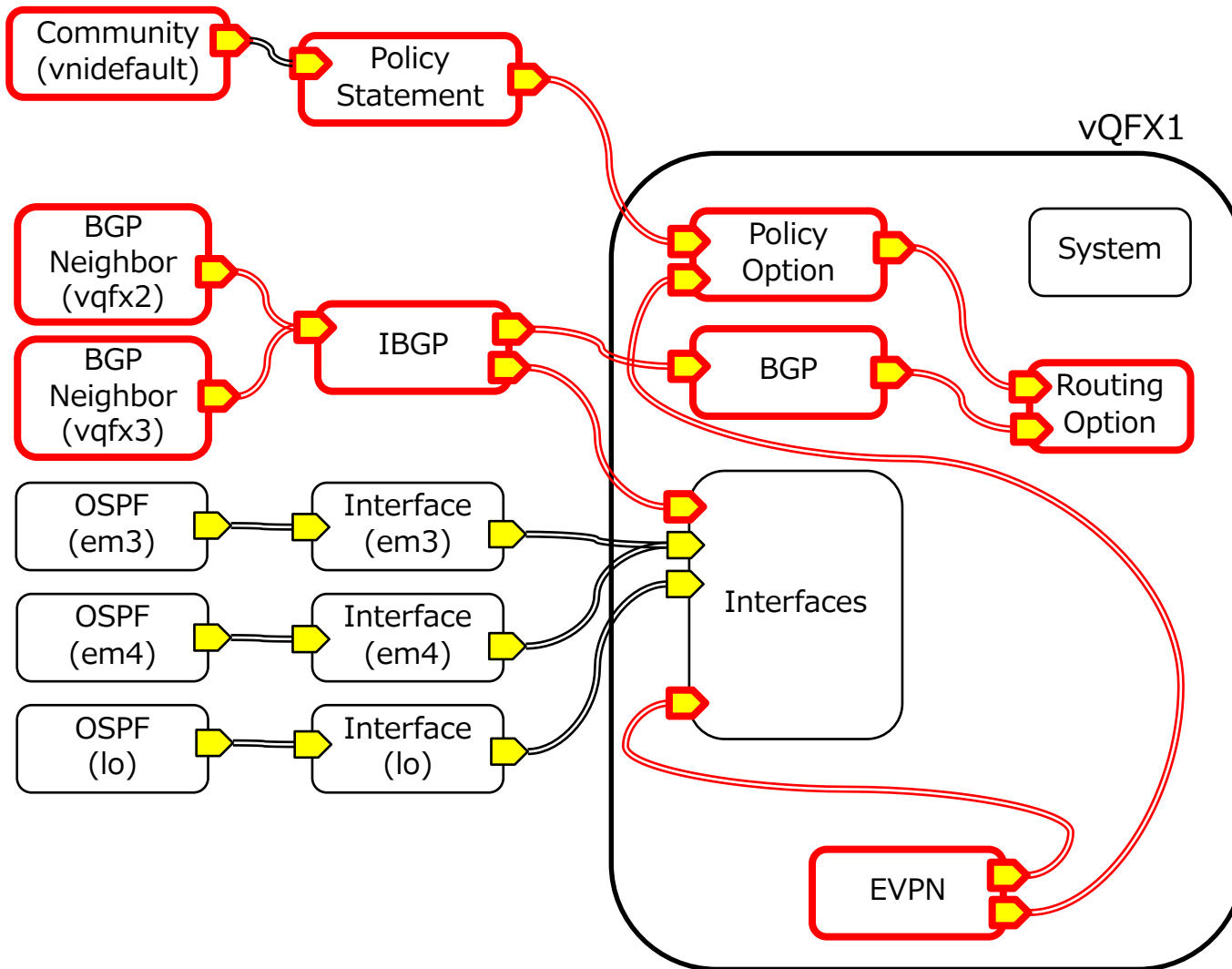
● IBGP

- set protocols bgp group {ibgp} type {internal}
- set protocols bgp group {ibgp} local-address {10.0.0.1}
- BGP Neighbor (Neighbor: vqfx2)
 - set protocols bgp group {ibgp} neighbor {10.0.0.2} description {vqfx2}
 - set protocols bgp group {ibgp} neighbor {10.0.0.2} family evpn signaling



EVPNコントロールプレーン設定の構成モデル

構成モデル



VXLANデータプレーン設定

論理インターフェース設定

- `set interfaces irb unit {vni} family inet {address}`
- `address`
 - `192.168.100.{deviceID}/24`

VLAN設定

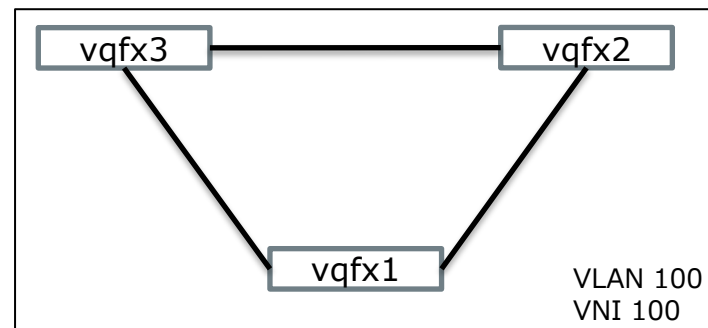
- `set vlans {vlan100} vlan-id {100}`
 - `set vlans {vlan100} l3-interface {irb.100}`
 - `set vlans {vlan100} vxlan vni {100}`

ルートターゲット設定

- `set protocols evpn vni-options vni {vni} vrf-target export target:{ASnum}:{vni}`

コミュニティ設定

- `set policy-options community vni{vni} members target:{ASnum}:{vni}`
- `set policy-options policy-statement evpn-import term vni{vni} from community vni{vni}`
- `set policy-options policy-statement evpn-import term vni{vni} then accept`

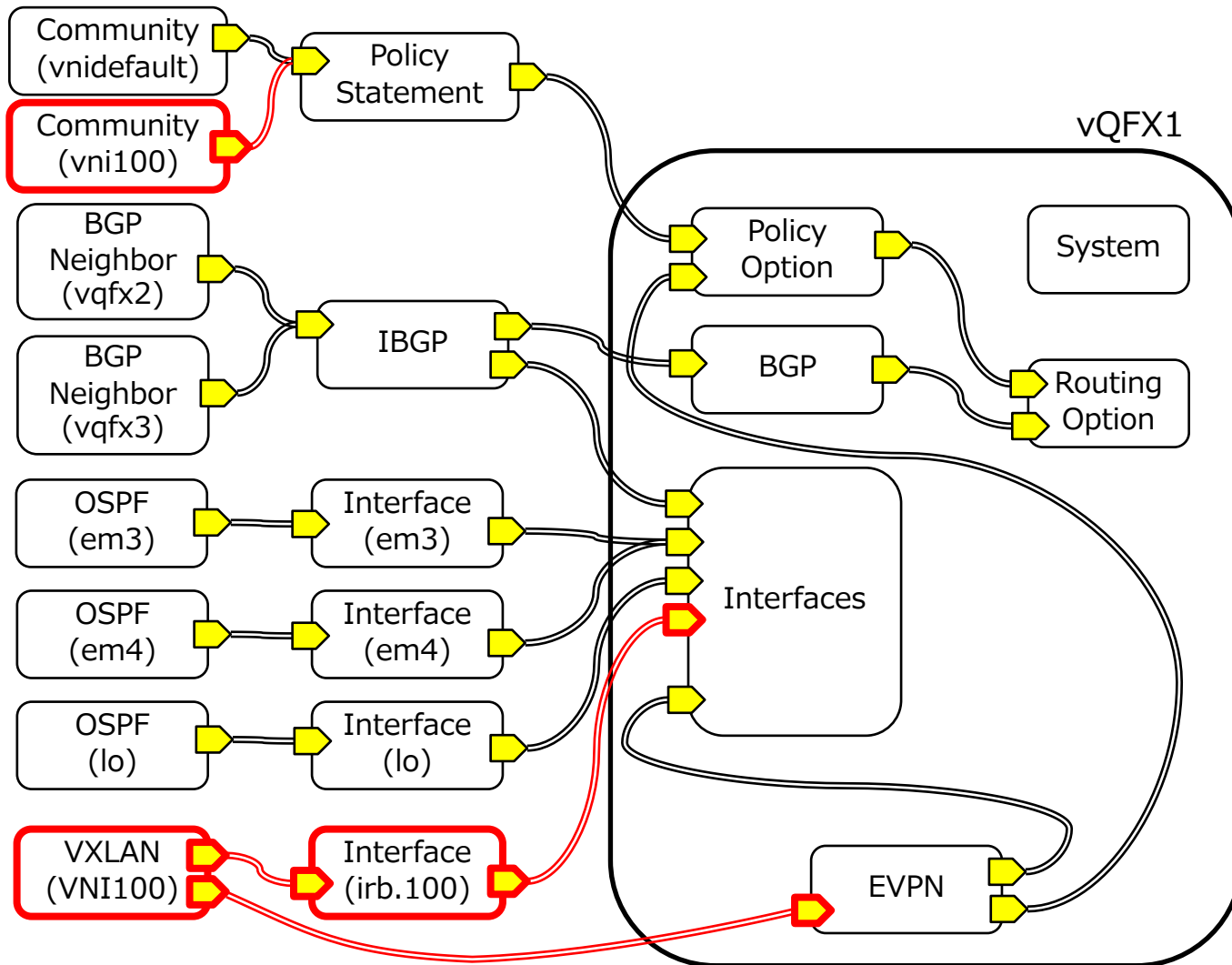


青字: プロパティ

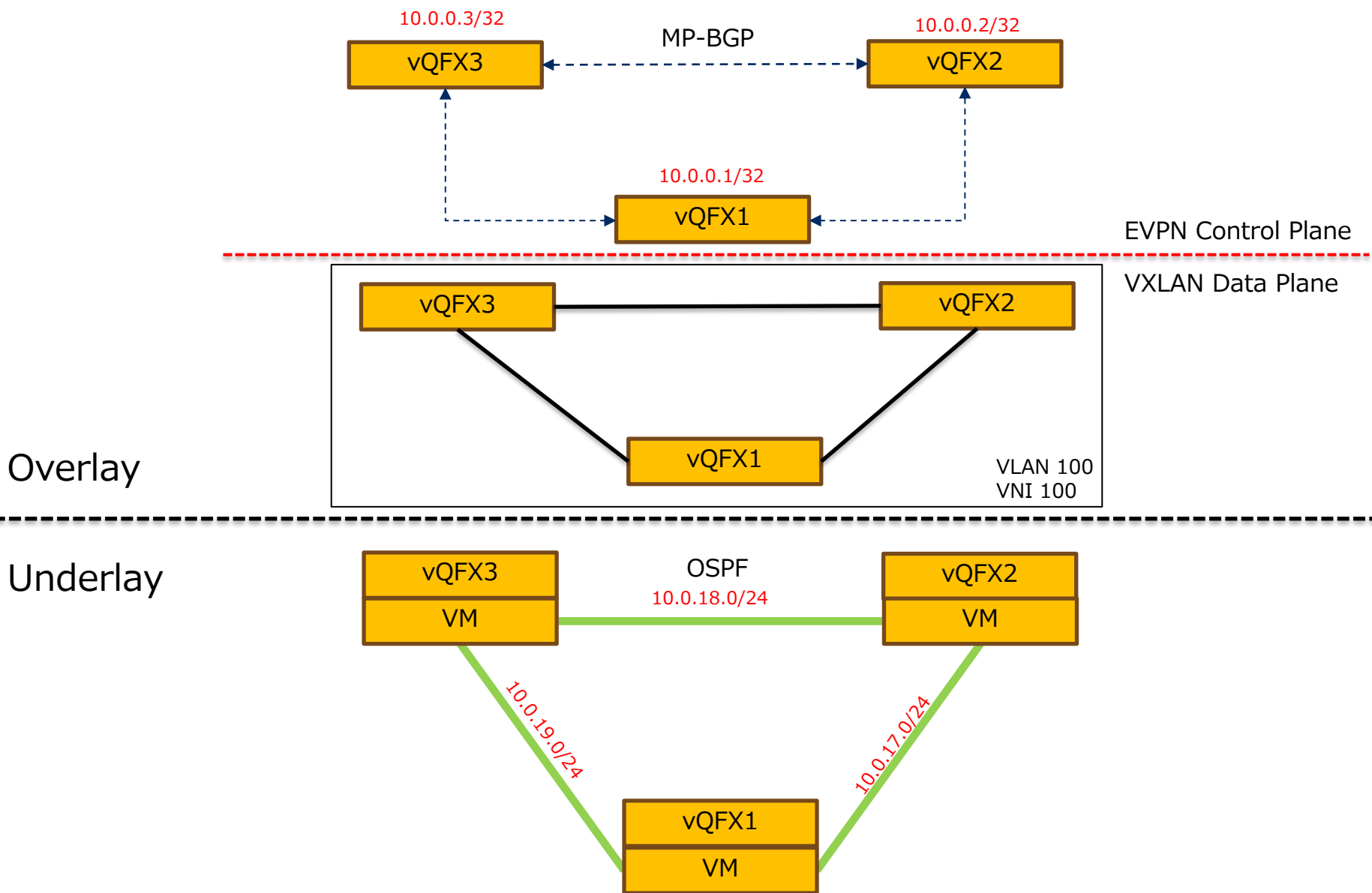
赤字: ポートで他コンポーネントを参照

VXLANデータプレーン設定の構成モデル

最終的な構成モデル

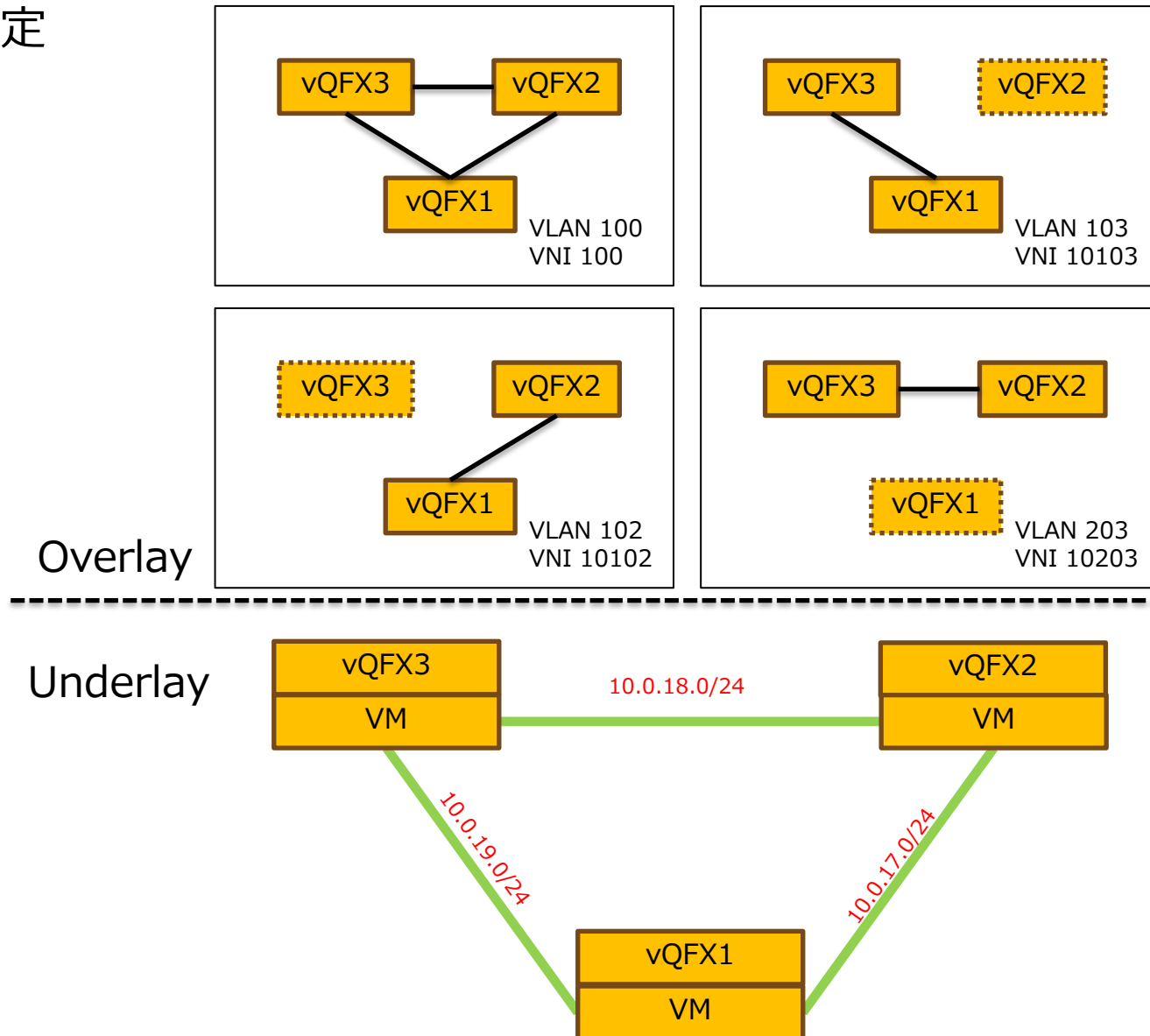


初期構築したEVPN/VXLAN構成（再掲）

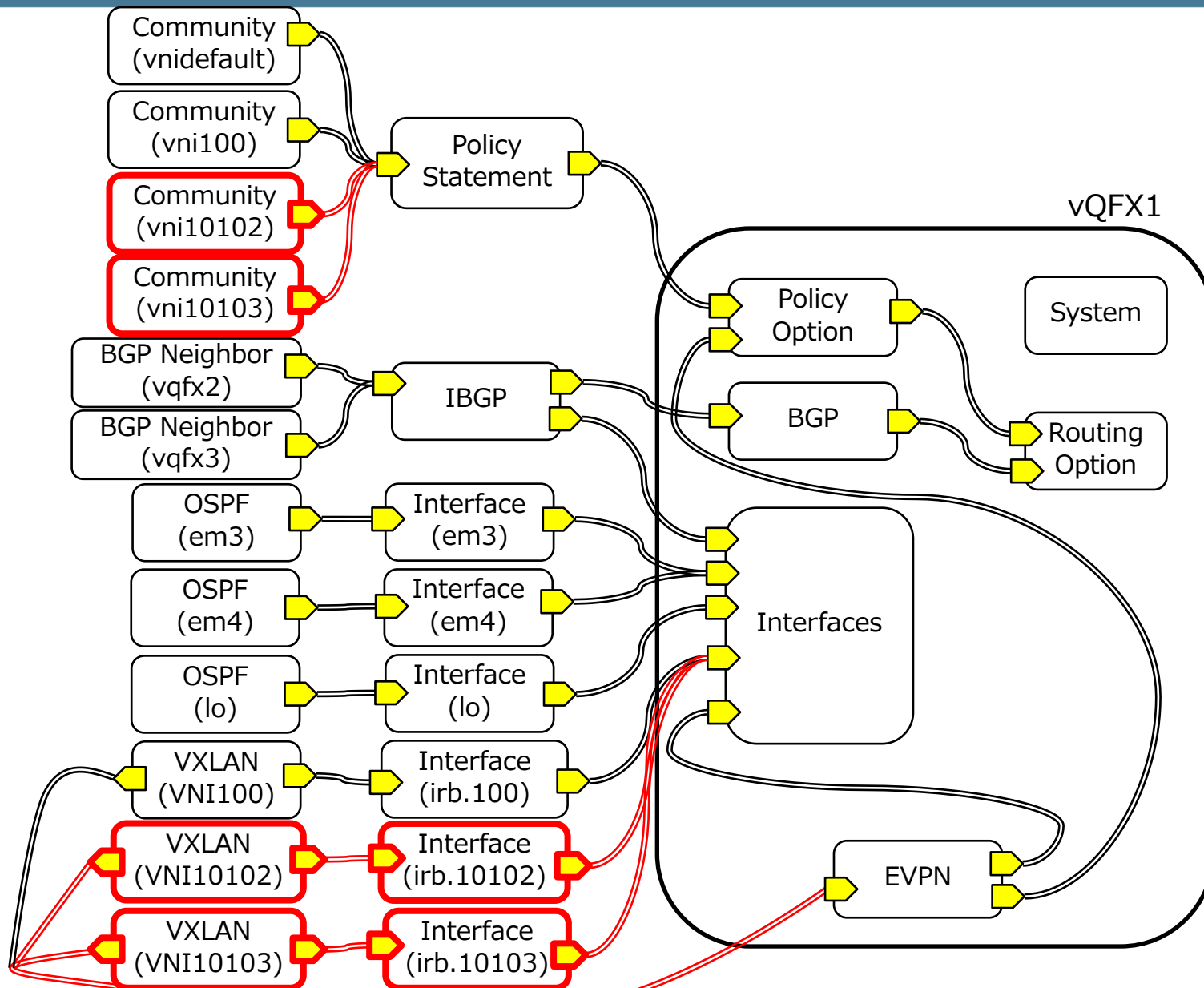


構成変更: Overlayへテナントを追加 (再掲)

新規VNIの設定



Overlayテナントを追加した構成モデル



目次

- サマリ
- 背景
- システム構成変更自動化技術Theater
- 実験内容
- 考察

Theaterの効果

- モデルさえ作成してしまえば、次回以降の変更ではパラメータの具体化だけで済み、コマンドや手順を意識する必要はない
- お試しで変更してすぐ戻すといったことも容易
- 管理用の情報と実機の設定のズレがなくなる

Ansible単独との違い

- スイッチ単体の変更に関しては、おそらく違いはない
 - ・ 最後の一発コミットすれば設定投入順序は無関係なため
- 複数のネットワークノードに対して変更をかける際の順序制御には使えるかも

Theaterの課題

- モデラだけでなくドメインエキスパートの協力が不可欠
 - ・ 依存関係の把握
 - ・ 再利用に向けた部品化の粒度
- 設計者はシステム構成部品の詳細といった観点ではなく、機能（フローなど）や性能（スループットなど）という観点で設計を検討することが多く、それに則した形式のインプットを扱えることが望ましい

サマリ（再掲）

■ 実施したこと

- 弊社のシステム構成変更自動化技術/ツール “Theater” を用いて EVPN/VXLAN環境を自動構築しました

■ 議論させていただきたい内容

- Theater の有効性
- Theater が使えそうな領域
- Theater を使えるものにするために改善すべき点
- ネットワークの新規構築や変更で困っている点
- 上記困りごとにかかっている工数
- 上記困りごとに対する現在の工夫

■ 本研究の一部は、総務省の委託研究「IoT共通基盤技術の確立・実証」プロジェクトの成果である。

 **Orchestrating** a brighter world

NEC