

クラウド基盤の仮想化NWを利用した構築自動化

2019年07月25日

中原一彦

日本電気株式会社

高塚修吾

NECソリューションイノベータ株式会社

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

- はじめに
- JANOG43のおさらい
- 物理機器はプール化して払い出したい
 - ✓ 在庫
 - ✓ 払い出し
- デモ
- 参考
 - ✓ 仮想化NWの検証
 - ✓ デモ用の初期設定

はじめに

今回の発表について

JANOG43において、

「クラウド基盤の仮想化NWの設計と課題」と題して物理リソース(HaaSレイヤ)のあるべき姿の課題をあげた。

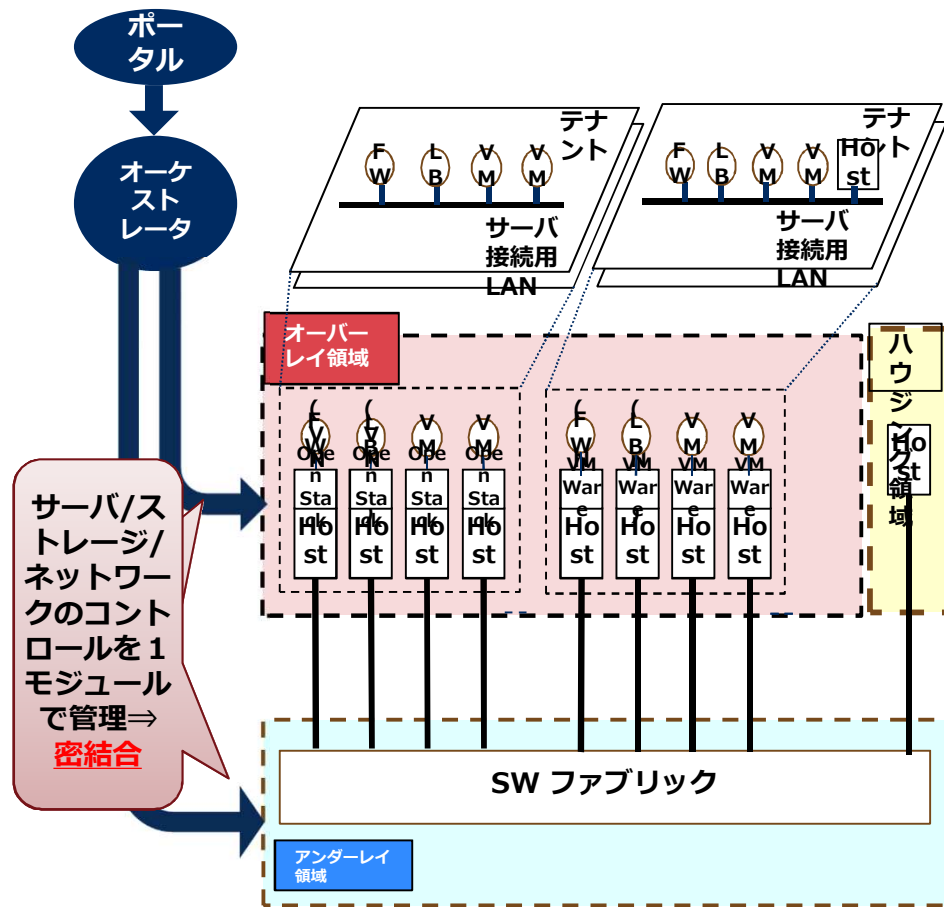
解はないものの、PoC環境を構築して、なるべく物理結線に変更をくわないで済む設計はないかを検証してみました。

→ HaaSレイヤにおけるその後の取り組みを共有します。

JANOG43のおさらい

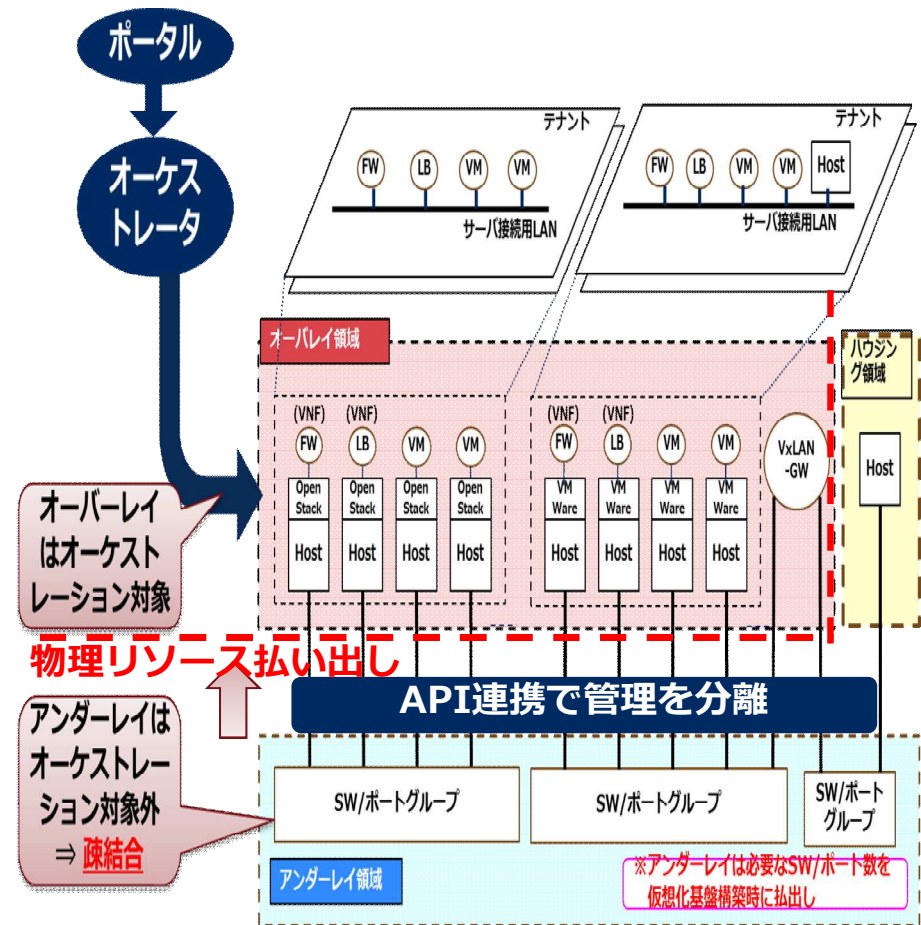
いままでのアーキテクチャ

<密結合> オーバーレイとアンダーレイをオーケストレーション対象とする。

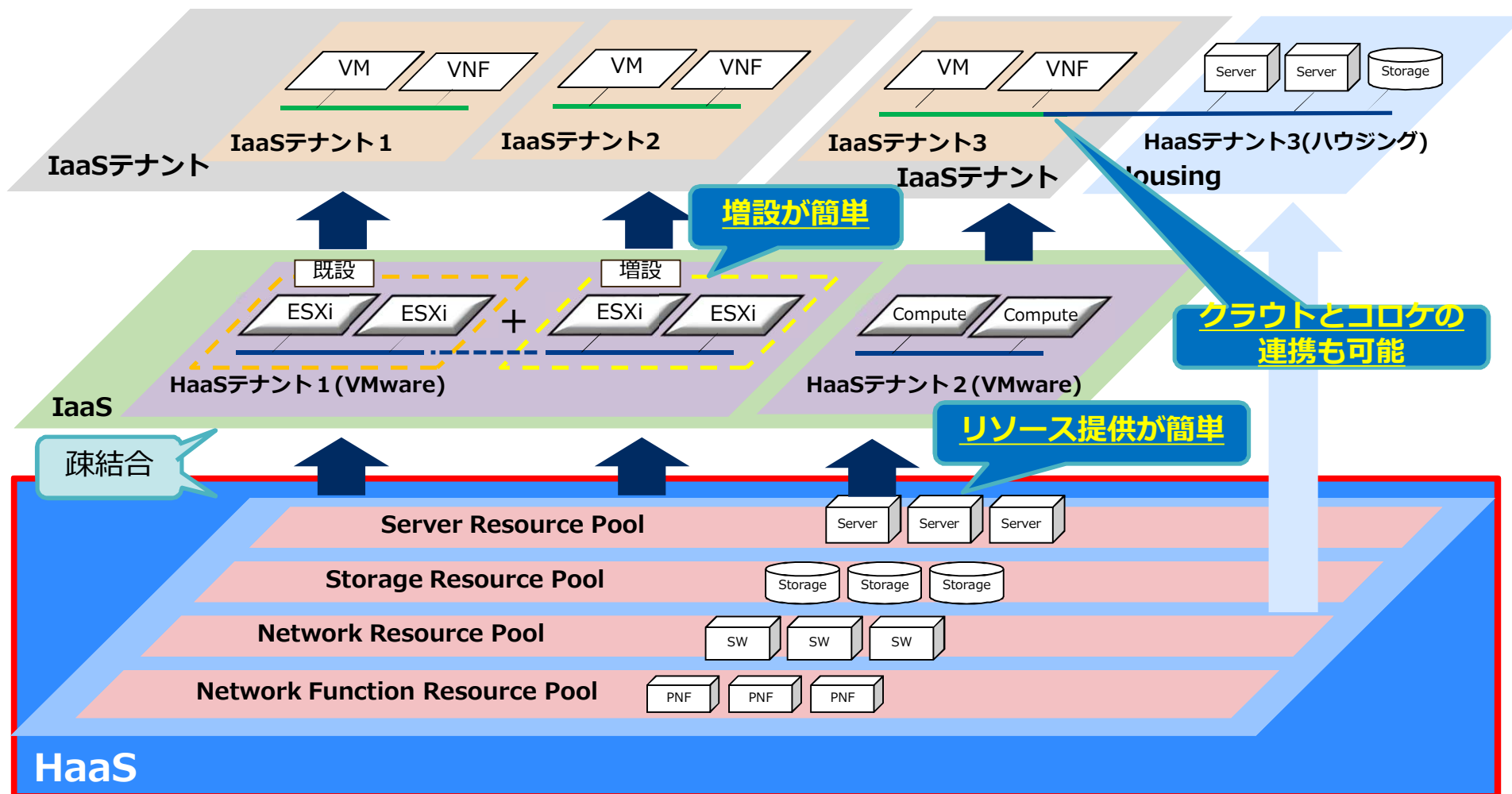


これから実施のアーキテクチャ

<疎結合> オーバーレイはオーケストレーション対象。アンダーレイは、HaaSとして別



HaaSレイヤを定義し、オンデマンドにリソースの提供を実現
— IaaSレイヤでは物理リソースの受給管理から解放 —



物理機器による構築やメンテを仮想機器のサービスと分離

HaaS – Hardware as a Service

- **物理機器を提供**すること

IaaS – Infrastructure as a Service

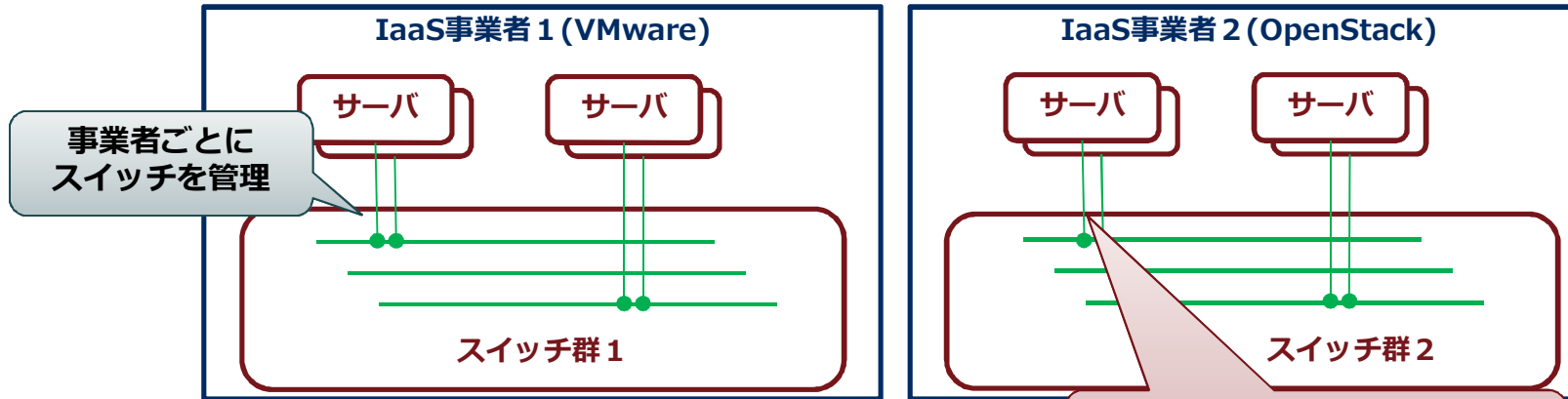
- **物理機器を使う環境を提供**すること
- 具体的には物理機器を他機器と連携する事と、物理機器上へ構築した仮想化機器や、それを複数組み合わせた環境を提供

HaaS / IaaSを別物と考える

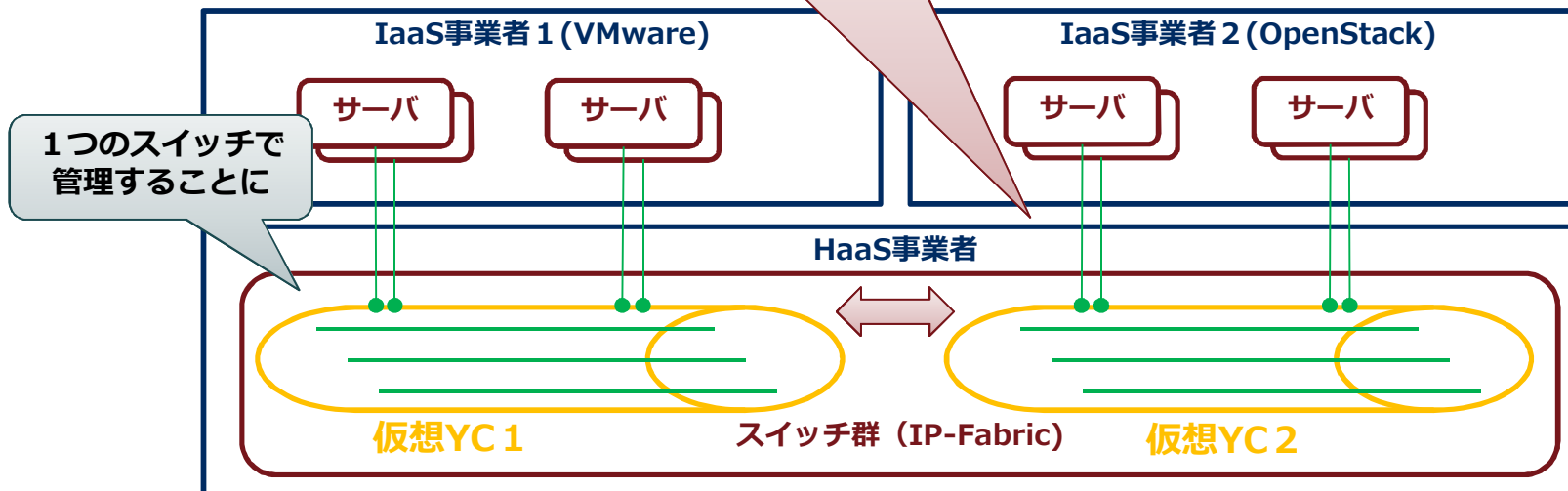
- HaaSは物理機器管理する事と、物理的に結線する事。構築、配線、故障に対応。
→ **HaaS事業者**
- IaaSは物理機器の構築と配線とをしない。物理結線はHaaSの役目。故障したら切り離して別の機器を使う事とし、面倒見ない。
→ **IaaS事業者**

論物分離のイメージ

HaaSレイヤがない場合



理想



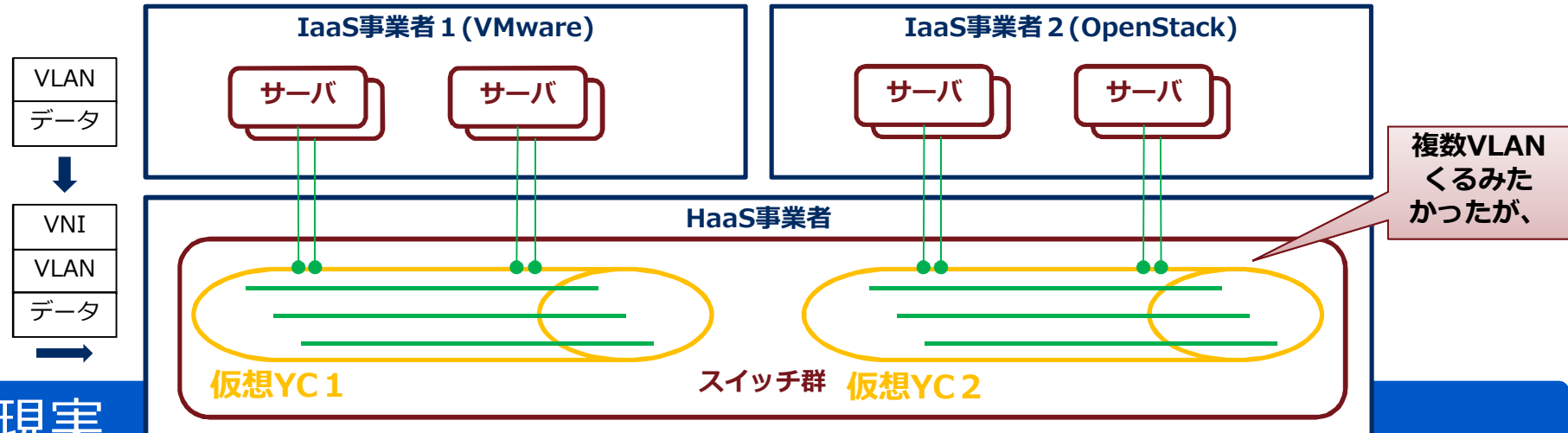
オーバーレイにEVPN/VxLANを採用（仮想YCを考えた）

- VLAN-Based Service
仮想ネットワーク（物理サーバポートからでてくるVLAN）の収容数が4Kに限られる。
- VLAN Bundle Service
仮想ネットワーク（物理サーバポートからでてくるVLAN）の収容数が4K x 4K可能であるが、MAC重複は許されない
- **VLAN-Aware Bundle Service**
仮想ネットワーク（物理サーバポートからでてくるVLAN）の収容数が4K x 4K可能でMAC重複も可能

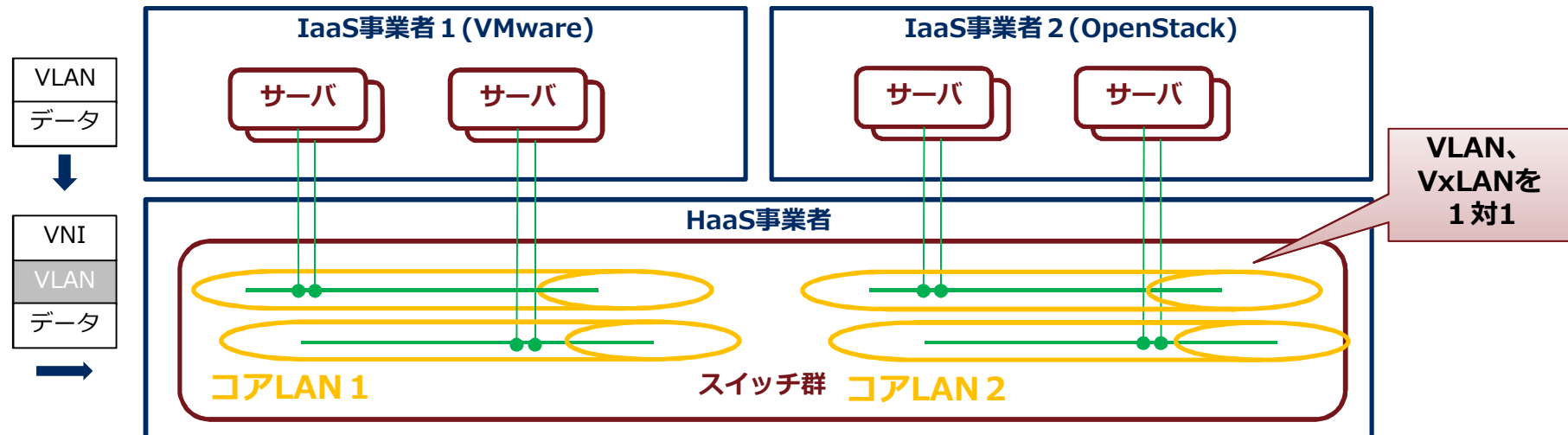
	VLAN-Based Service	VLAN Bundle Service	VLAN-Aware Bundle Service
EVPN instance (EVI)・Bridge domain (BD)	1BD対1EVI 1VLAN対1BD ×EVI数の消費量が膨大	1BD対1EVI 多VLAN対1BD ×EVI数の消費は抑えられるが、1BDに収容するためにBUMフラiddiingの影響範囲が大きい	多BD対1EVI 1VLAN対1BD ○EVI数の消費も抑えられ、BUMフラiddiingの影響範囲も対象VLANにしか与えない
EVPN-VXLAN対応	1BDに1VLAN⇔1VXLANマッピングする形になる ○	1BDに複数VLAN⇔1VXLANマッピングする形になる ×複数VLANをマッピングする場合はQ-in-Qを利用することになるが、Q-in-QではMAC重複ができないため	1BDに 1VLAN⇔1VXLANマッピング する形になる ○

論物分離の理想と現実

理想



現実



目標

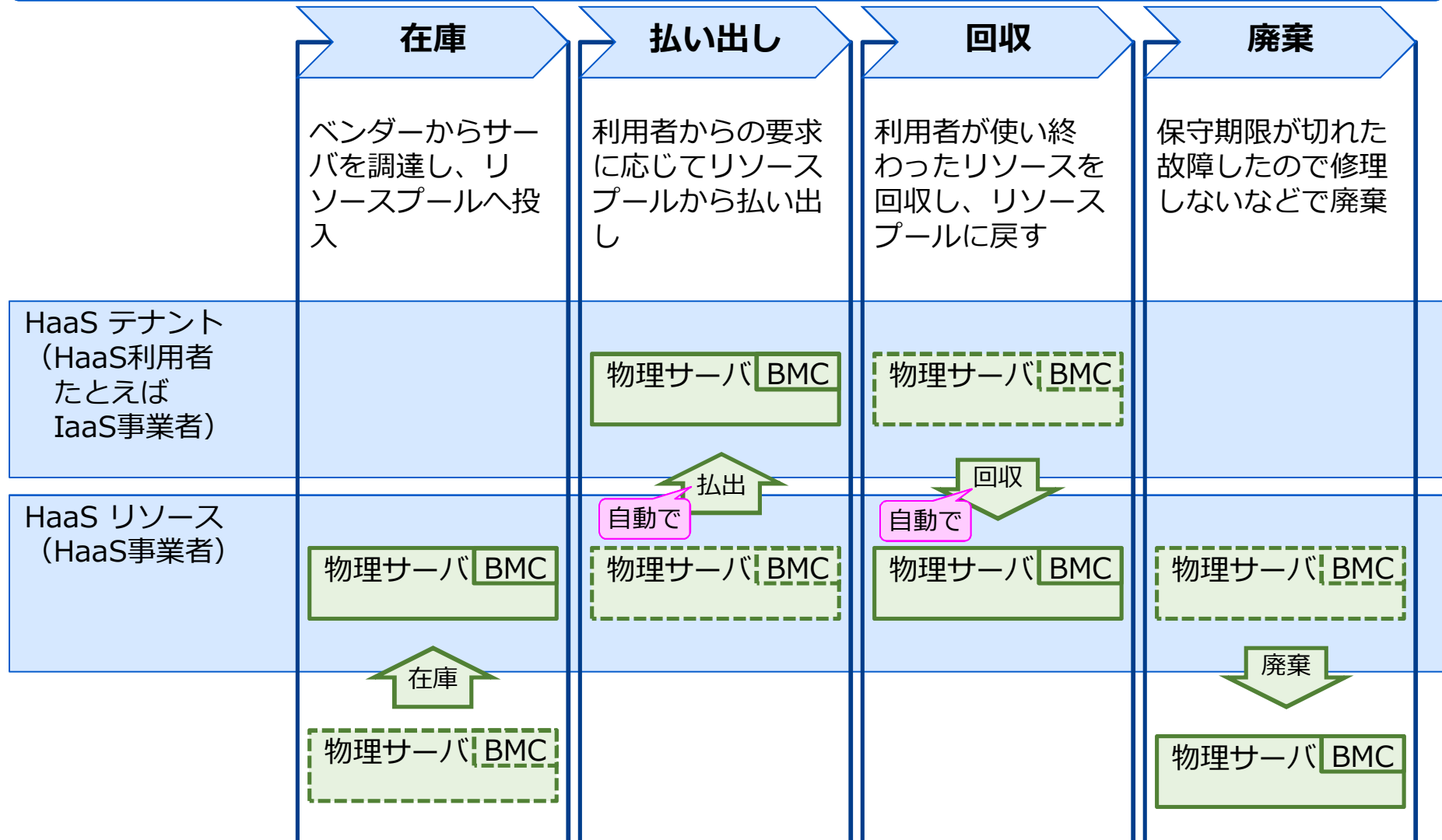
- HaaSレイヤで物理SVをボタンひとつで上位レイヤへ払い出す。
また、HaaSテナントを準備して、それぞれサービス運用が始まったらHaaSテナントでアイソレートしたNWを提供する
- IaaSサービス管理者（オーバレイNW）とHaaSサービス管理者（アンダーレイNW)間で密な連携を不要にする
- これまで出来ていたサービスの継続。
(つまりマルチハイパーバイザ対応、ハウジング連携)

LCM

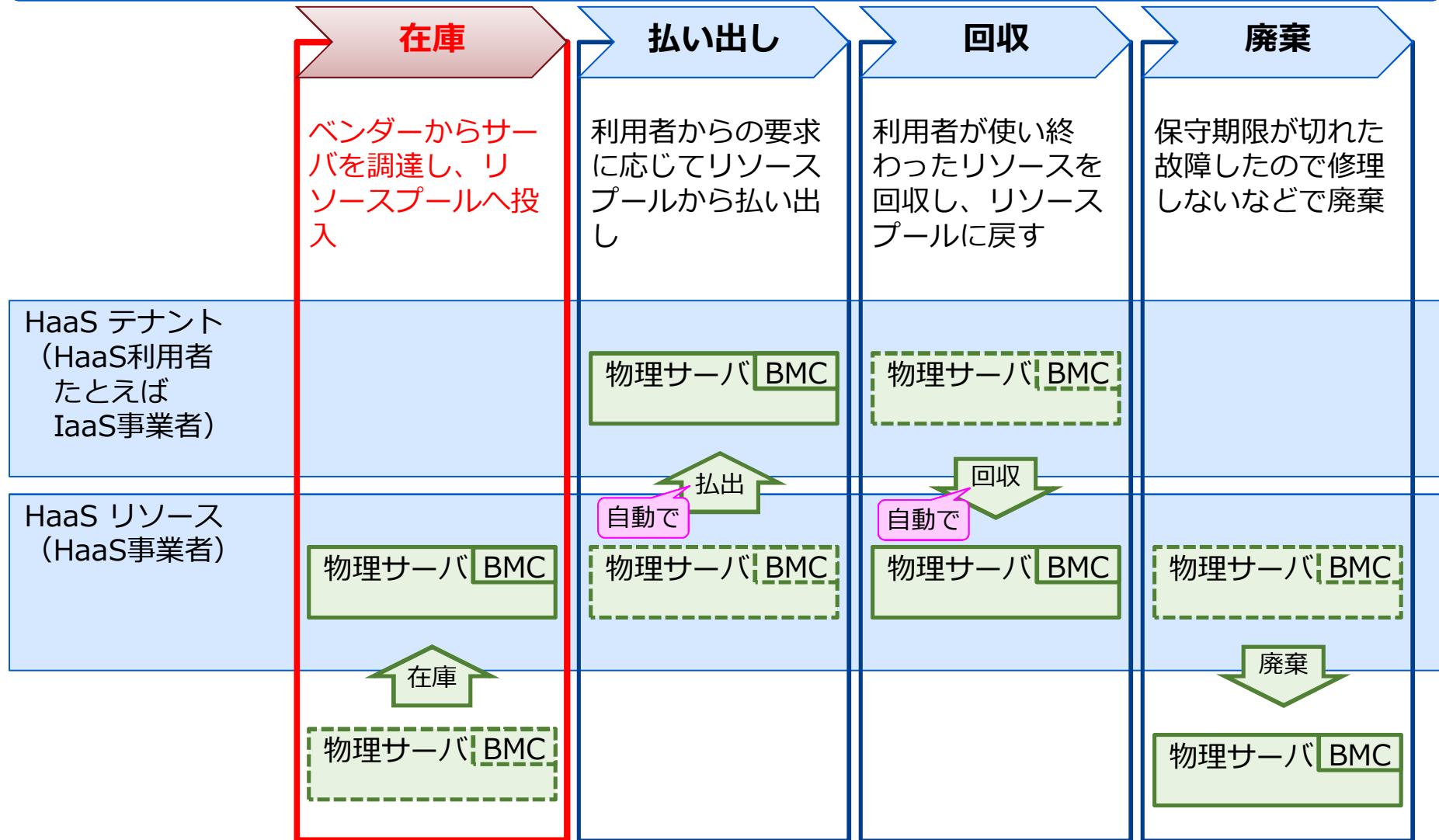
- 物理SVをラッキングしたら物理結線の変更をなくしたい。（構築コスト最適化）
- 故障しても定期メンテまでの長期間縮退運転で対応したい。（運用コスト最適化:計画可能）
- 物理SV機器は、自由に**プール化してHaaSテナントに払い出したい。**（リソースの効率化）

物理機器はプール化して払い出したい

このライフサイクルで自動化を実現したい (SV払い出し)

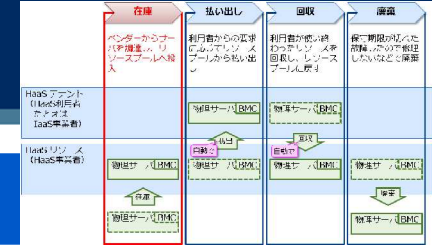


このライフサイクルで自動化を実現したい (SV払い出し)



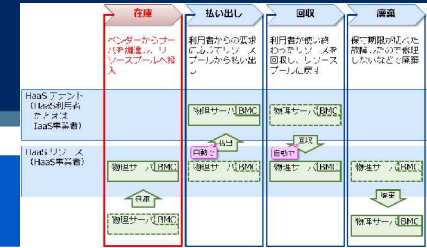
はじめに着手したこと

情報の電子化と共通化



管理項目	管理内容	いままで	これから
ホスト名	ホスト名	<ul style="list-style-type: none"> Excel管理 PJが命名規則を管理し、各チームが命名 	<ul style="list-style-type: none"> DB管理 PJが命名規則を管理し、各チームが命名 ⇒一元管理
HW構成パーツ	CPUやメモリなど	<ul style="list-style-type: none"> Excel管理 各チームが設計 	<ul style="list-style-type: none"> DB管理 PJが設計 ⇒HWの共通化
機器構成	機器の搭載場所や用途など	<ul style="list-style-type: none"> Excel管理 インフラチームが設計 	<ul style="list-style-type: none"> DB管理 PJが設計 ⇒機器構成の共通化
NW結線	結線情報やケーブル情報(線種や長さ)など	<ul style="list-style-type: none"> Excel管理 各チームが収容要件(構成情報とI/F情報)を設計し、インフラチームが詳細設計 	<ul style="list-style-type: none"> DB管理 PJが設計、各チームが詳細設計 ⇒結線と設定の分離

情報の電子化と共通化のメリット[1/2]



ラック設計の自動化

ラックA

ラックB

ラックC



情報の電子化と共通化のメリット[2/2]

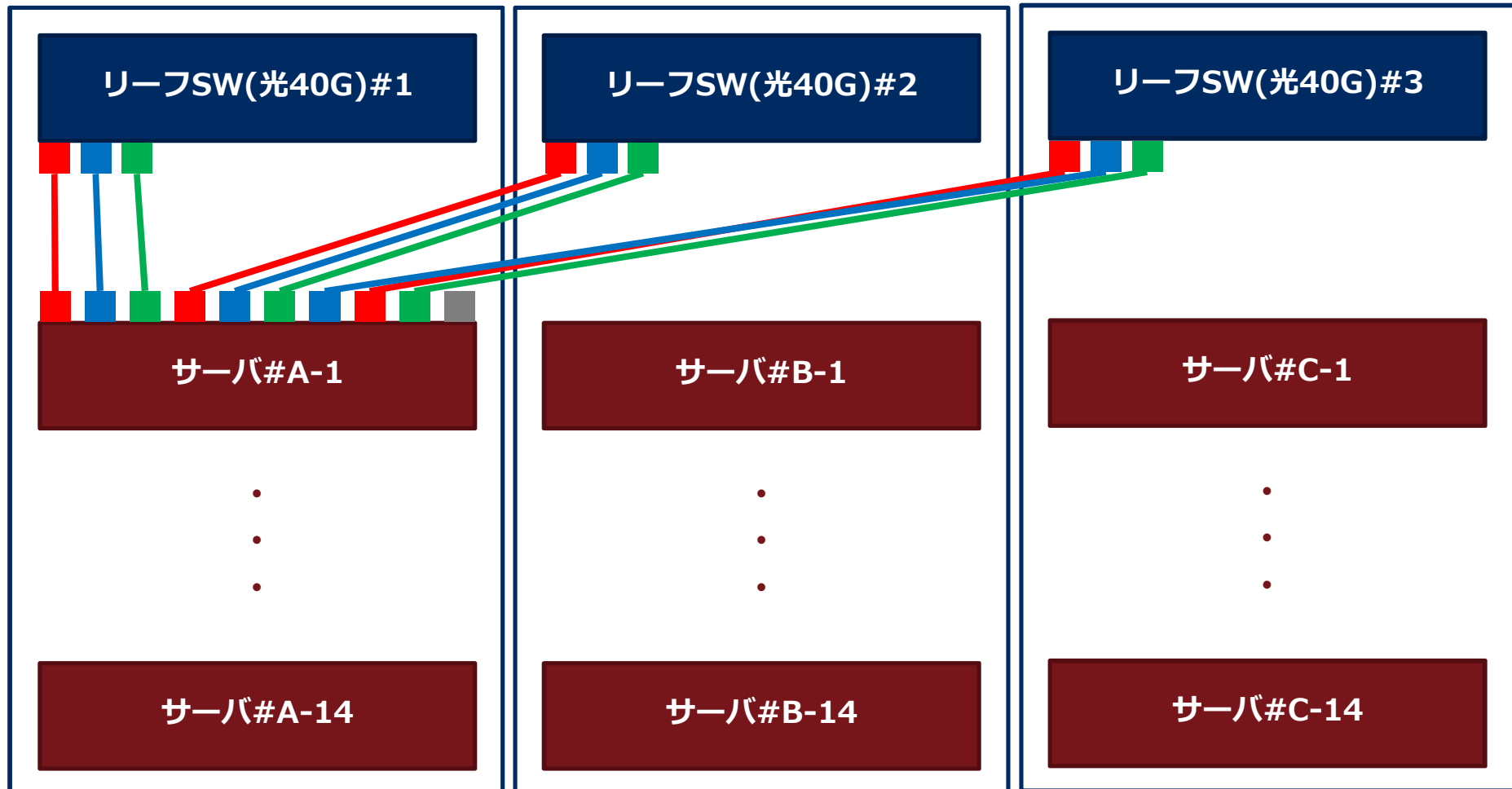
	登録	払い出し	回収	廃品
Handyフロント (1405利用名 から変更 1405準拠)	ベンダーからのデータ連携のシナリオへの登録	利用書からの取崩しをシステムから取り出し、システムから取り出し	利用書が破損した場合はシステムから取り出し、システムから取り出し	破損が確認された利用書の廃棄処理
Handyリンク (1405準拠)	管理サーバに接続	管理サーバに接続	管理サーバに接続	管理サーバに接続

物理ポート収容設計の自動化

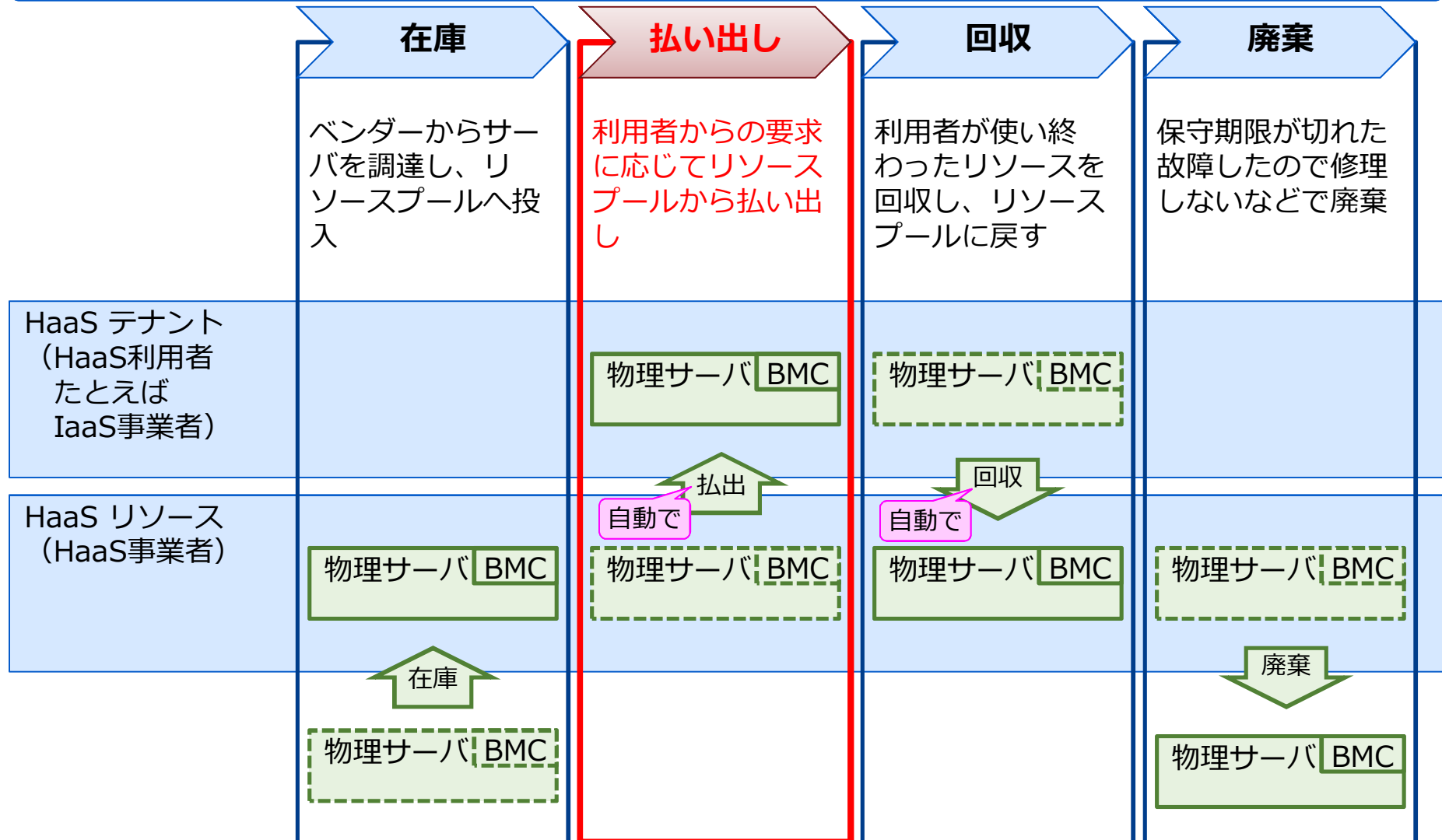
ラックA

ラックB

ラックC



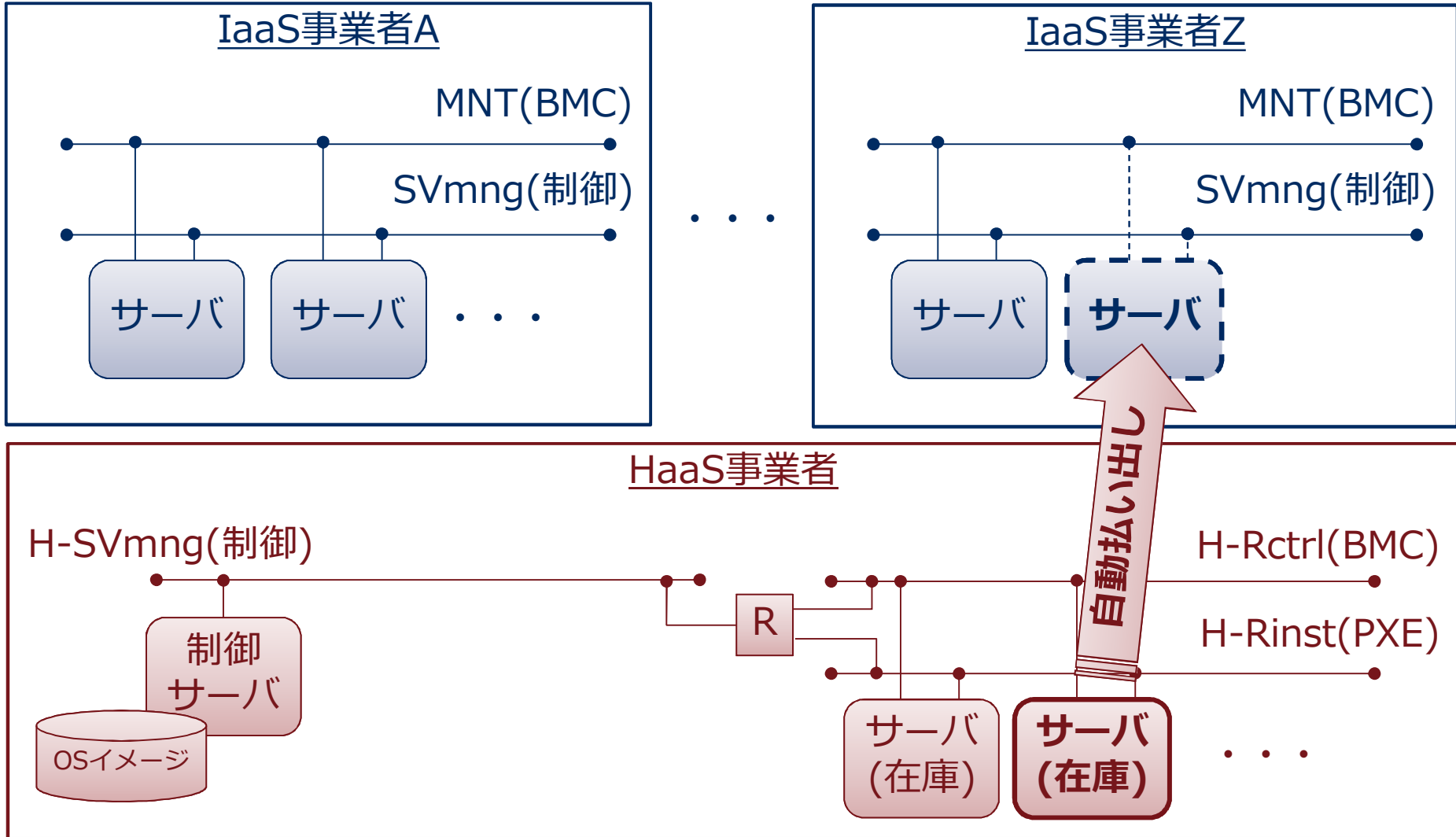
このライフサイクルで自動化を実現したい (SV払い出し)



自動払い出しするために

	在庫	払い出し	回収	廃棄
ベンダー/メーカーの承認なしにシステムをインストール		利用者の要求に応じてシステムをインストールし、システムから払い出し	利用者の要求に応じてシステムを回収し、システムから払い出し	利用者の要求に応じてシステムを廃棄し、システムから払い出し
HaaS デバイス (HaaS利用者のHaaS事業者)		物理リソース	物理リソース	
HaaS リソース (HaaS事業者)	物理リソース	物理リソース	物理リソース	物理リソース

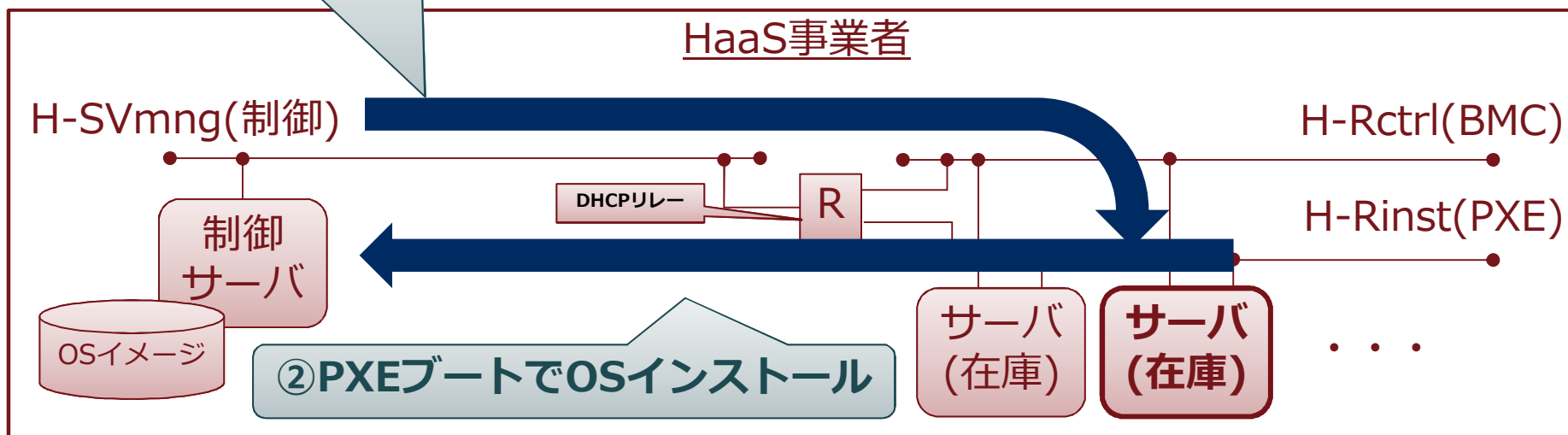
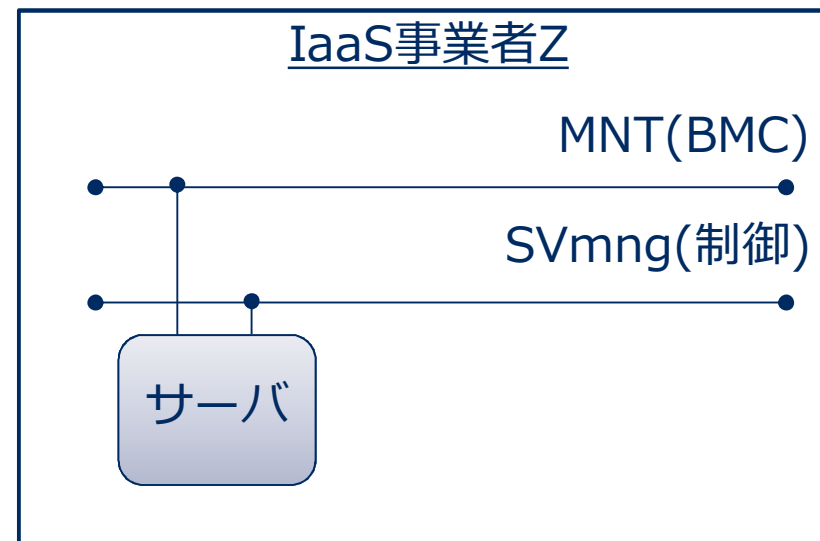
初期状態



自動払い出しするために

サーバ(在庫)の初期設定

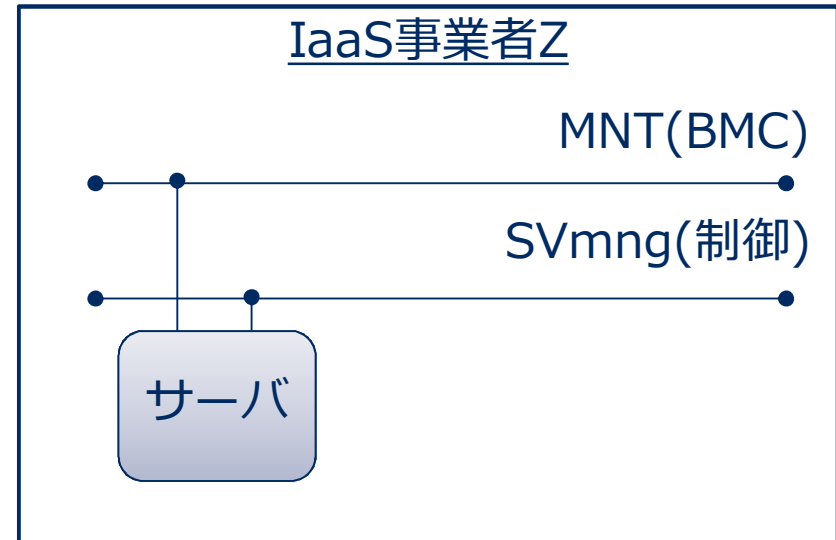
	在庫	払い出し	回収	廃棄
ベンダー/メーカー の承認を受け、シ ステムイメージを インストール		利用者の要求 に応じて、シ ステムイメージを インストール	利用者の要求 に応じて、シ ステムイメージを インストール	利用者の要求 に応じて、シ ステムイメージを インストール
HaaSマシント (HaaS制御サ ーバ)の稼働		稼働	稼働	稼働
HaaSリソース (HaaSサーバ)	稼働	稼働	稼働	稼働



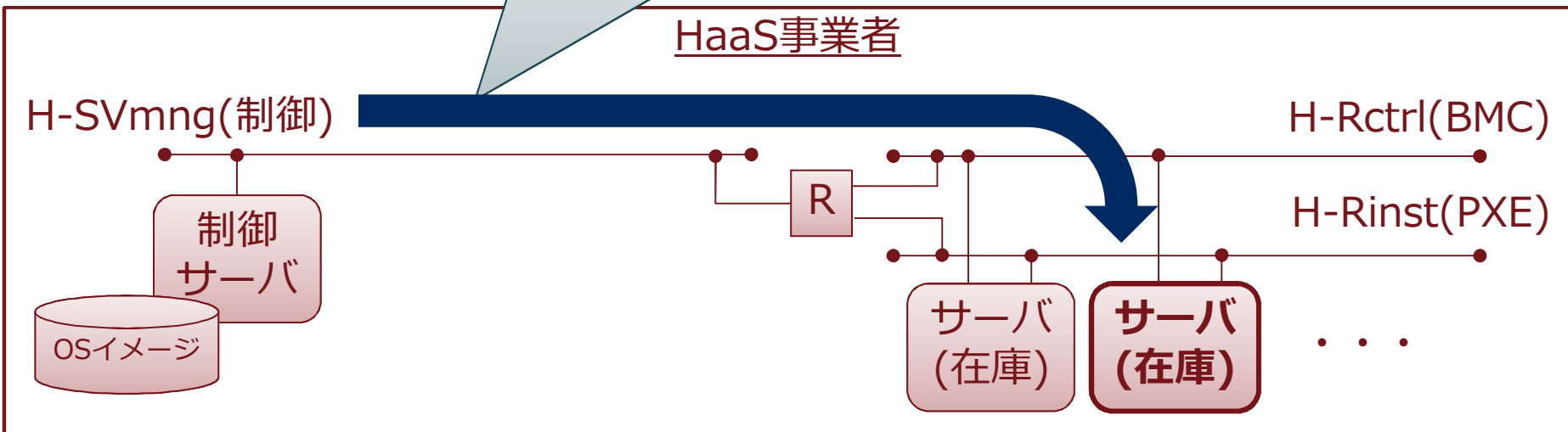
自動払い出しするために

サーバ(在庫)のNW設定変更

	在庫	払い出し	回収	廃棄
ベンダーからのサーバの受取、システムインストール		利用者の要求に応じてシステムインストール	利用者の要求に応じてシステムインストール	利用者の要求に応じてシステムインストール
HaaSマシント (HaaS制御) または HaaSサーバ		管理UI / 監視UI	管理UI / 監視UI	
HaaSリソース (HaaSサーバ群)	管理UI / 監視UI	管理UI / 監視UI	管理UI / 監視UI	管理UI / 監視UI
	上機		同機	廃棄
	管理UI / 監視UI		管理UI / 監視UI	管理UI / 監視UI



① redfishを利用して
VLAN ID・IPアドレスを変更

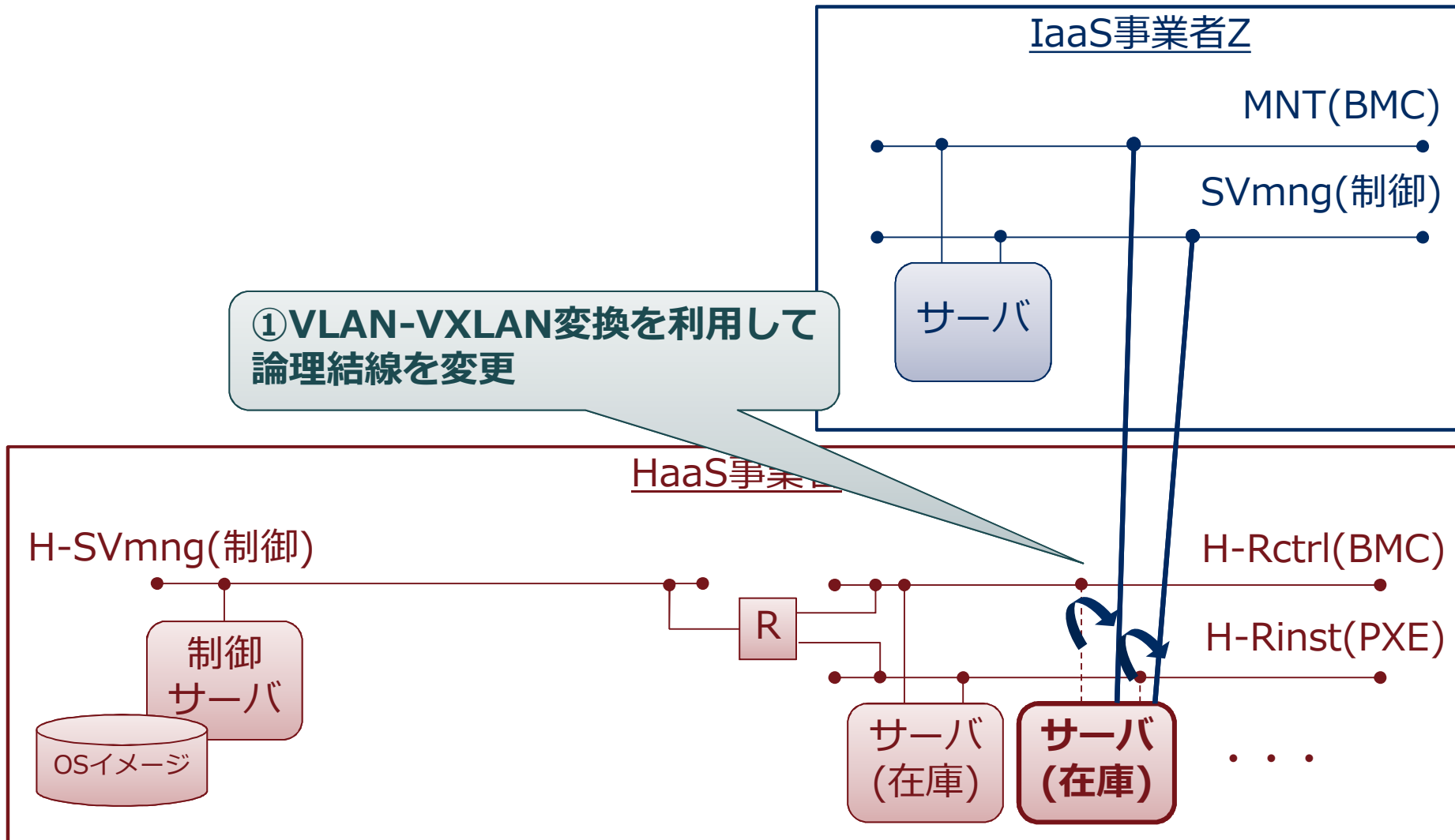


自動払い出しするために

	在庫	払い出し	回収	廃棄
ベンダー/ホス タの管理/レ リース管理/レ リース終了の 処理		利用開始の通知 /リース開始の 通知/リース 終了の通知	利用開始の通知 /リース開始の 通知/リース 終了の通知	リース終了の 通知/リース 終了の通知
HaaSマシント (HaaS制御サ ーバ)の管理 (HaaS事業者)		物理リソース の管理	物理リソース の管理	物理リソース の管理
HaaSリソース (HaaS事業者)	物理リソース の管理	物理リソース の管理	物理リソース の管理	物理リソース の管理

サーバ(在庫)の論理結線変更

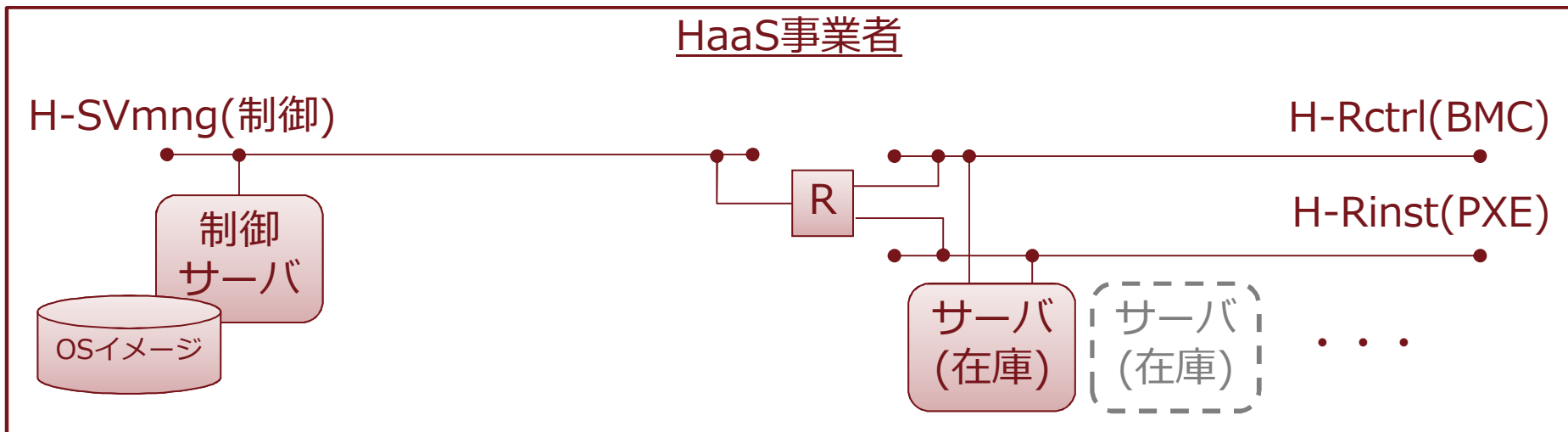
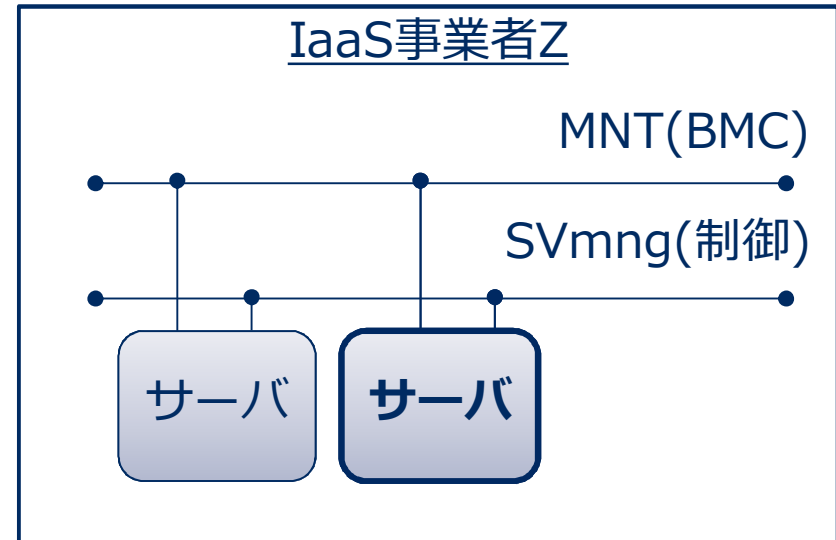
① VLAN-VXLAN変換を利用して
論理結線を変更



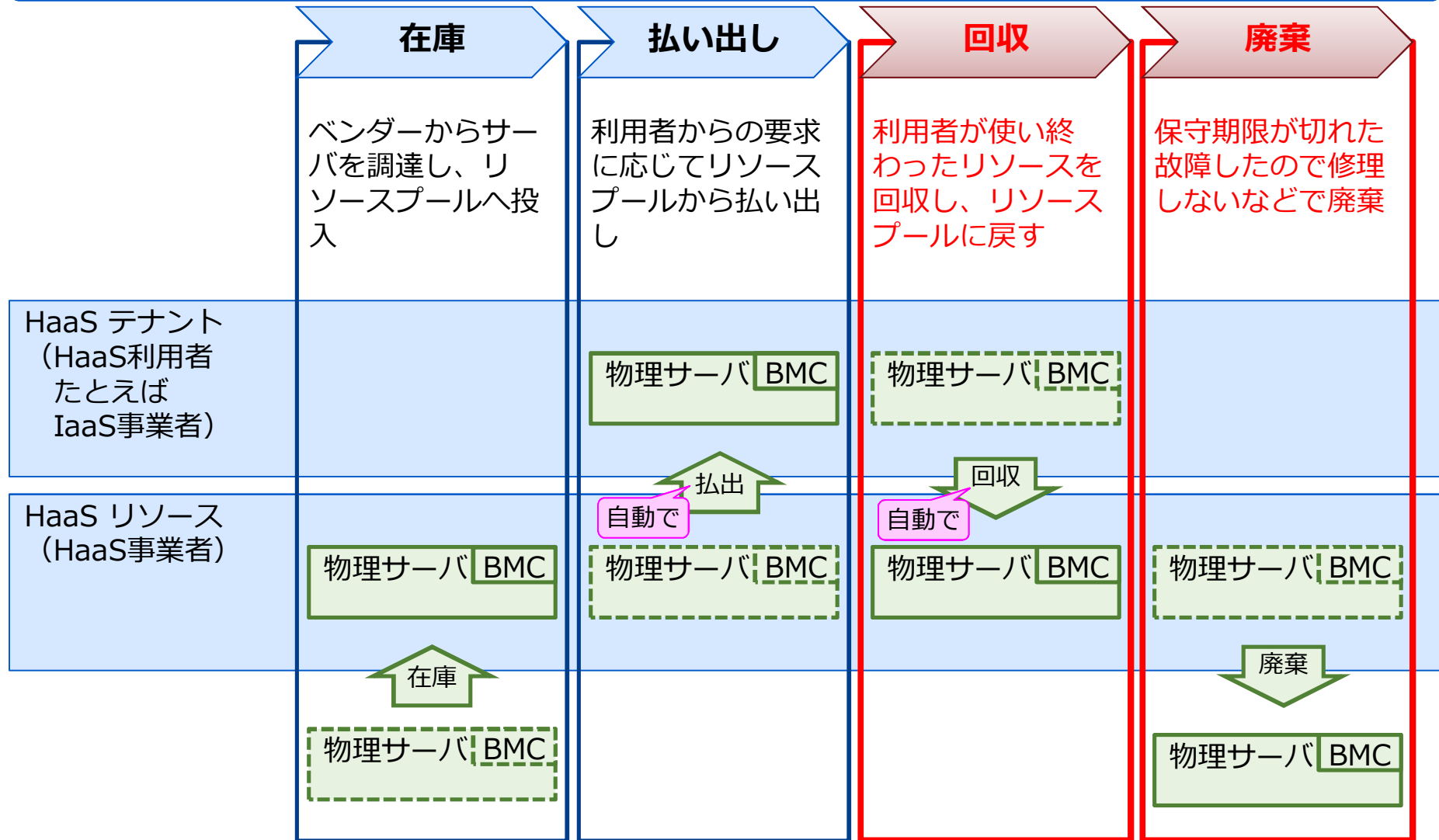
自動払い出しするために

払い出し完了

	在庫	払い出し	回収	廃棄
ベンダー/ホス タの管理/レ ジューア ド/システム のインストール		利用者の要求 に応じて、シ ステムのインストール が完了するまで	利用者の要求 に応じて、シ ステムのインストール が完了するまで	利用者の要求 に応じて、シ ステムのインストール が完了するまで
HaaS マニファ スト (HaaS利用者 またはHaaS事業者)		物理リソース	物理リソース	
HaaS リソース (HaaS事業者)	物理リソース	物理リソース	物理リソース	物理リソース



このライフサイクルで自動化を実現したい (SV払い出し)



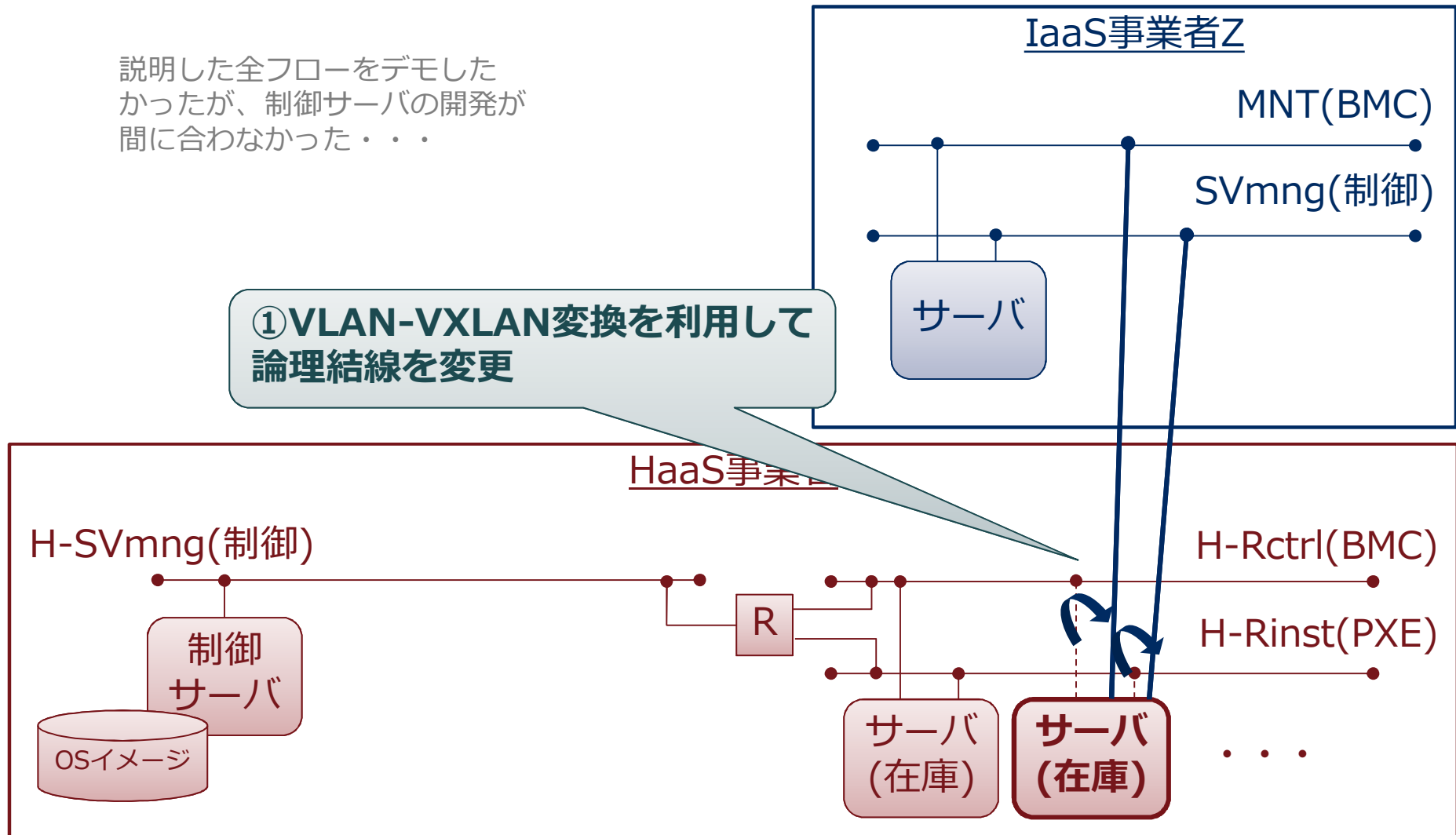
デモ

本日のデモ(録画ですが・・・)

サーバ(在庫)の論理結線変更

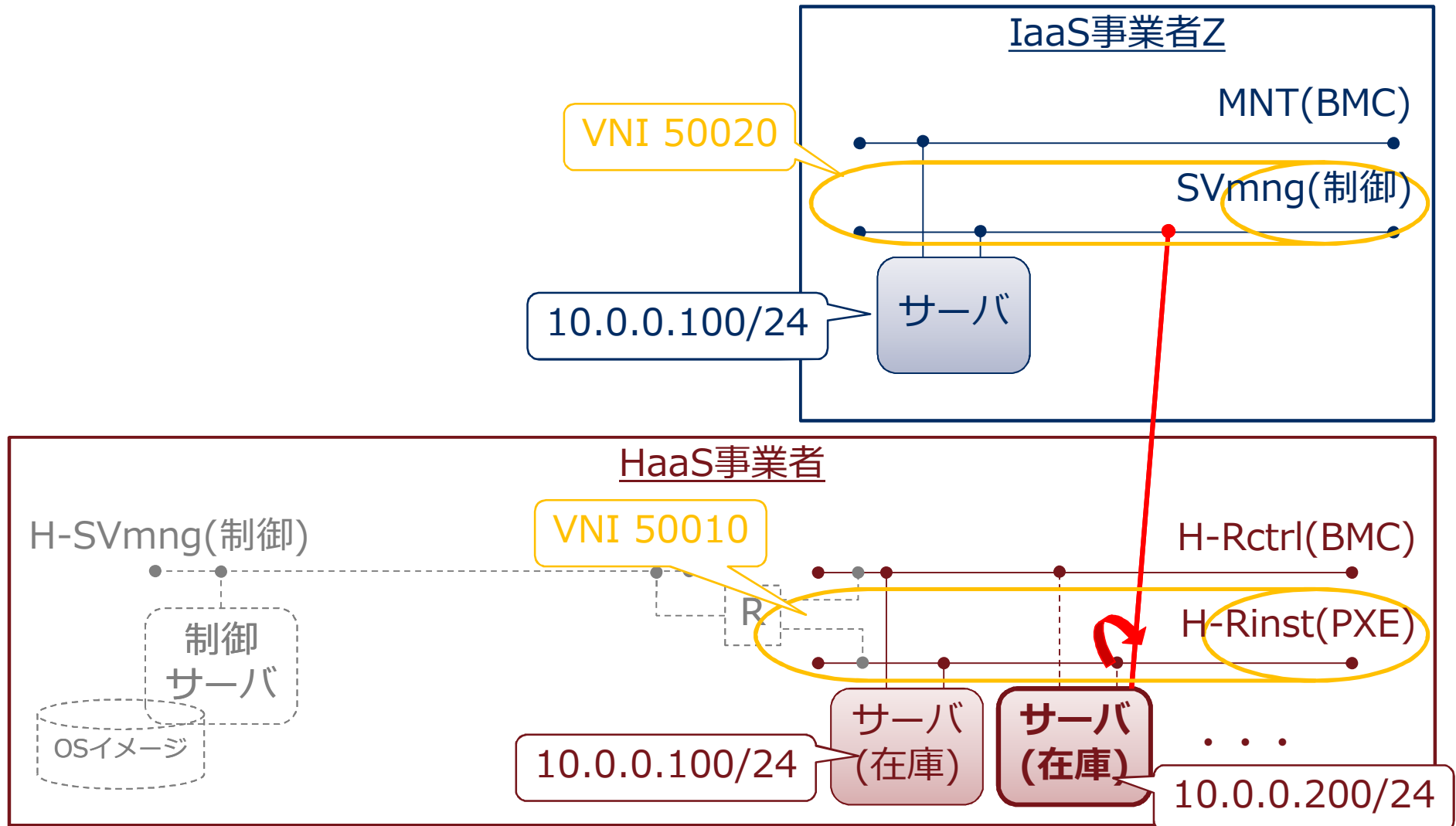
説明した全フローをデモした
かったが、制御サーバの開発が
間に合わなかった・・・

① VLAN-VXLAN変換を利用して
論理結線を変更



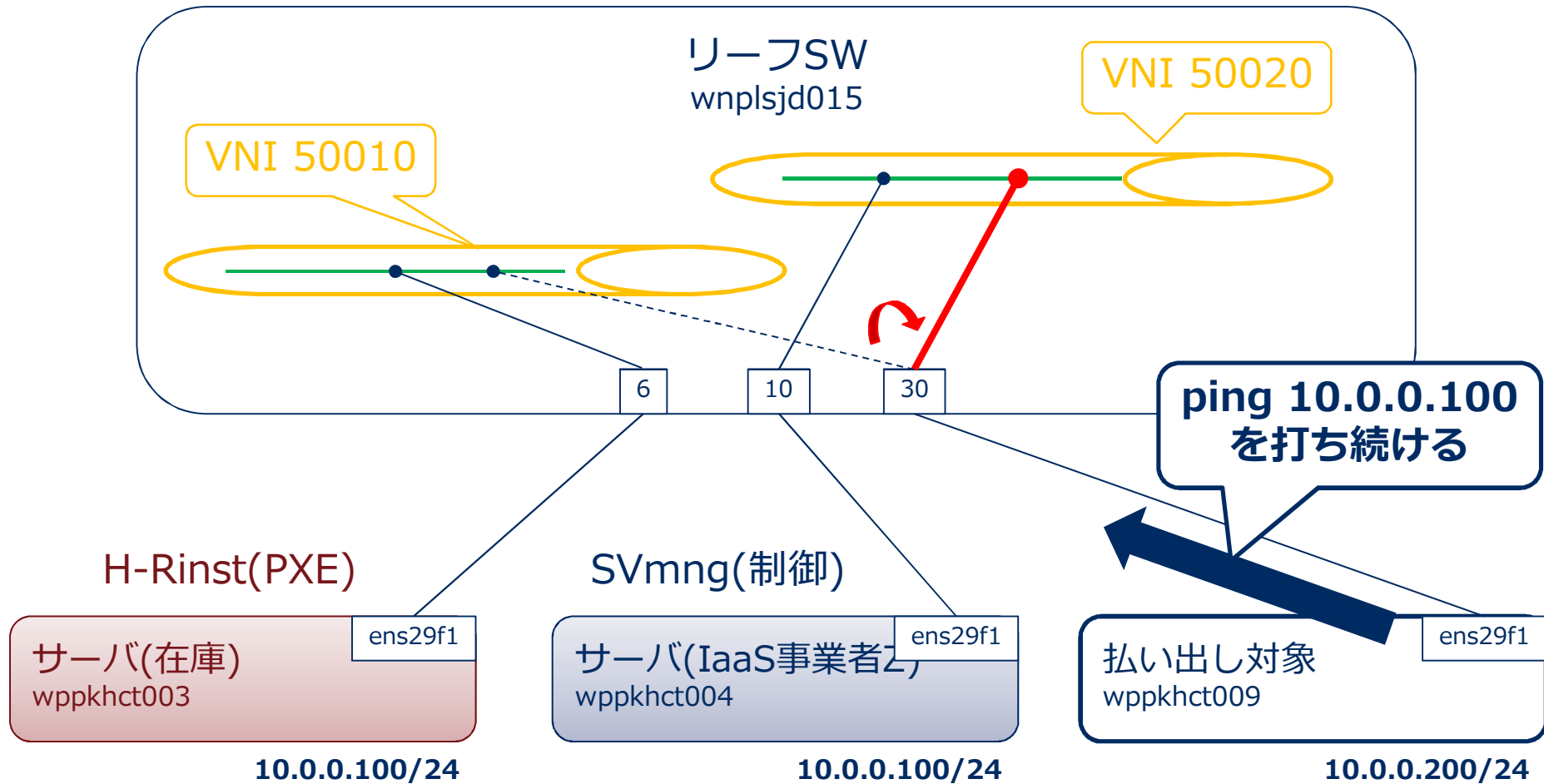
本日のデモ(録画ですが・・・)

論理構成



本日のデモ(録画ですが・・・)

物理構成



デモ画面

デモの画面配置と実行コマンド

リーフSW wnplsjd015

```
del interfaces xe-0/0/30
del vlans VLAN_0000005_10 interface xe-0/0/30.10

set interfaces xe-0/0/30 flexible-vlan-tagging
set interfaces xe-0/0/30 native-vlan-id 20
set interfaces xe-0/0/30 mtu 9216
set interfaces xe-0/0/30 encapsulation extended-vlan-bridge
set interfaces xe-0/0/30 unit 20 vlan-id 20
set vlans VLAN_0000005_20 interface xe-0/0/30.20
set vlans VLAN_0000005_20 vxlan vni 50020
set vlans VLAN_0000005_20 vxlan ingress-node-replication
set policy-options community 0000005_50020 members target:64601:50020
set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50020 from community 0000005_50020
set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50020 then accept
set protocols evpn vni-options vni 50020 vrf-target target:64601:50020
```

サーバ(在庫) wppkhct003

```
tcpdump -nt -i ens29f1
```

サーバ wppkhct004

```
tcpdump -nt -i ens29f1
```

払い出し対象 wppkhct009

```
ping 10.0.0.100 -n
```

デモ(録画 : 1分17秒)

The image shows a network configuration demo. The main window is a Tera Term VT terminal with the following configuration commands:

```
root@wplsjd015# load set terminal
[Type ^D at a new line to end input]

del interfaces xe-0/0/30
del vlans VLAN_0000005_10 interface xe-0/0/30.10

set interfaces xe-0/0/30 flexible-vlan-tagging
set interfaces xe-0/0/30 native-vlan-id 20
set interfaces xe-0/0/30 mtu 9216
set interfaces xe-0/0/30 encapsulation extended-vlan-bridge
set interfaces xe-0/0/30 unit 20 vlan-id 20
set vlans VLAN_0000005_20 interface xe-0/0/30.20
set vlans VLAN_0000005_20 vxlan vni 50020
set vlans VLAN_0000005_20 vxlan ingress-node-replication
set policy-options community 0000005_50020 members target:64601:50020
set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50020 from community 0000005_50020
set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50020 then accept
set protocols evpn vni-options vni 50020 vrf-target target:64601:50020
```

Below the main window are three smaller terminal windows:

- Left window: localhost - root@wppkhct003 - VT. Command: `[root@wppkhct003 ~]# tcpdump -nt -i ens29f1`
- Middle window: localhost - root@wppkhct004 - VT. Command: `[root@wppkhct004 ~]# tcpdump -nt -i ens29f1`
- Right window: localhost - root@wppkhct009 - VT. Command: `[root@wppkhct009 ~]# ping 10.0.0.100 -r`

物理サーバをプール化して
HaaSテナントに払い出せた

しかし . . .

課題を残した部分

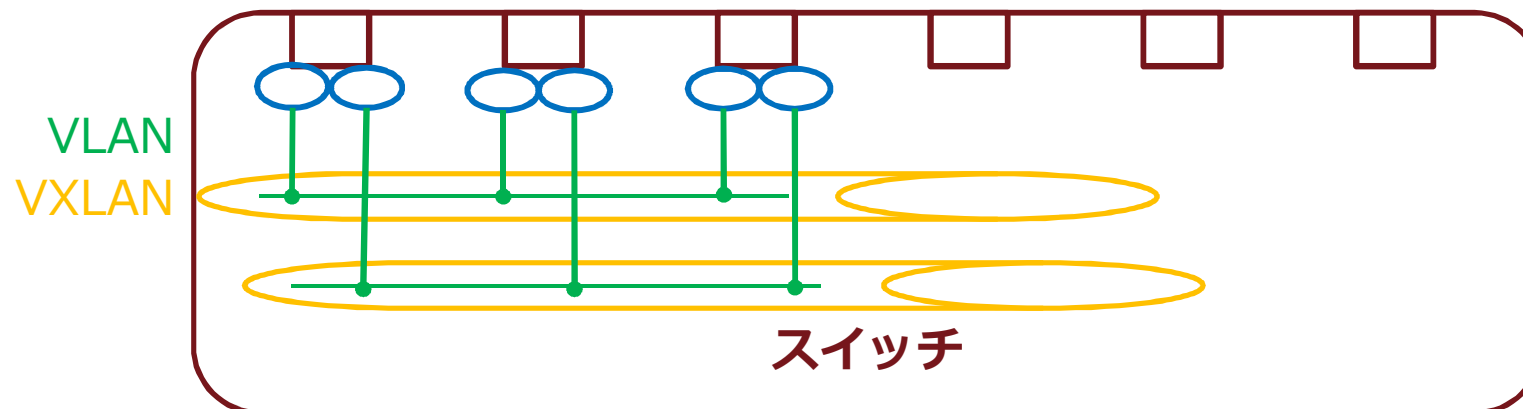
VLAN設計問題

概要

- 論理結線変更としてEVPN-VXLANを利用するようにしたが、結局は1VLAN⇔1VXLANマッピングするため、VLANに縛られてしまう

対策

- 物理ポート単位でVLAN4K利用できるので、HaaS事業者とIaaS事業者でVLAN空間を分けてVLAN設計する



意図的に忘れた部分（もしかしたらねじ伏せた？）

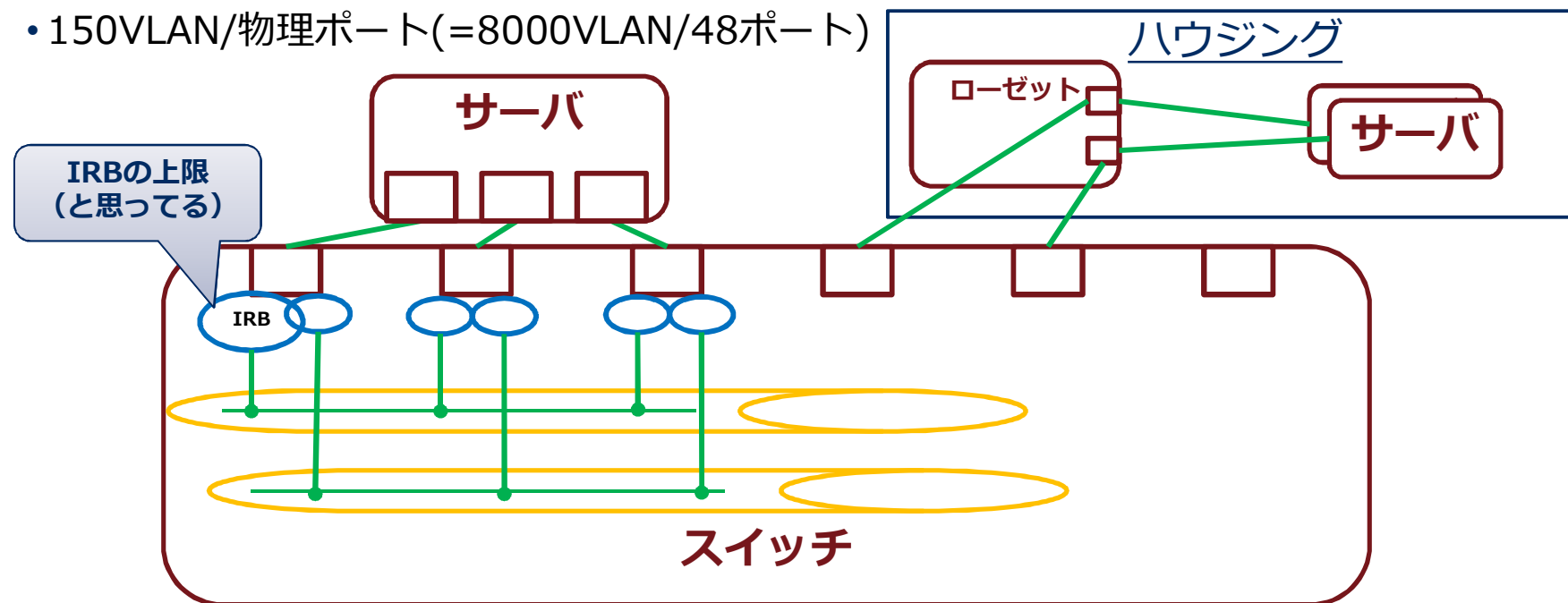
VLAN8K問題

概要

- Tomahawk 2(BroadCom)のCAM(Content Addressable Memory)容量による
エントリ数の上限が8K個に制限される

対策

- 物理ポートあたりに使用できるVLAN数を制限する
 - 150VLAN/物理ポート(=8000VLAN/48ポート)



VLAN設計問題

- VLAN設計はどうすれば良いのだろうか？

VLAN 8 K問題

- スイッチで8KしかVLANを扱えない対策は？

VLAN設計問題

- VLAN設計はどうすれば良いのだろうか？
→ VLAN=V x LANをPort単位で管理するか。。。

VLAN 8 K問題

- スイッチで8KしかVLANを扱えない対策は？
→ 1ポート当たりのVLAN数を制限するしかないか。。。

参考

論物分離の検証
デモ用初期設定

仮想化NWの検証

仮想化NWは、どのように実現するのか

検証したNW機器とNW技術

NW機器

- スパイン
 - (QFX5100-48S×2台)×1セット
- リーフ
 - (QFX5100-48S×2台 + QFX5100-48T×1台)×6セット

アンダーレイNW

- EBGP(External Border Gateway Protocol)
 - 機器毎にASを分け、フルメッシュEBGPネイバーを確立する。

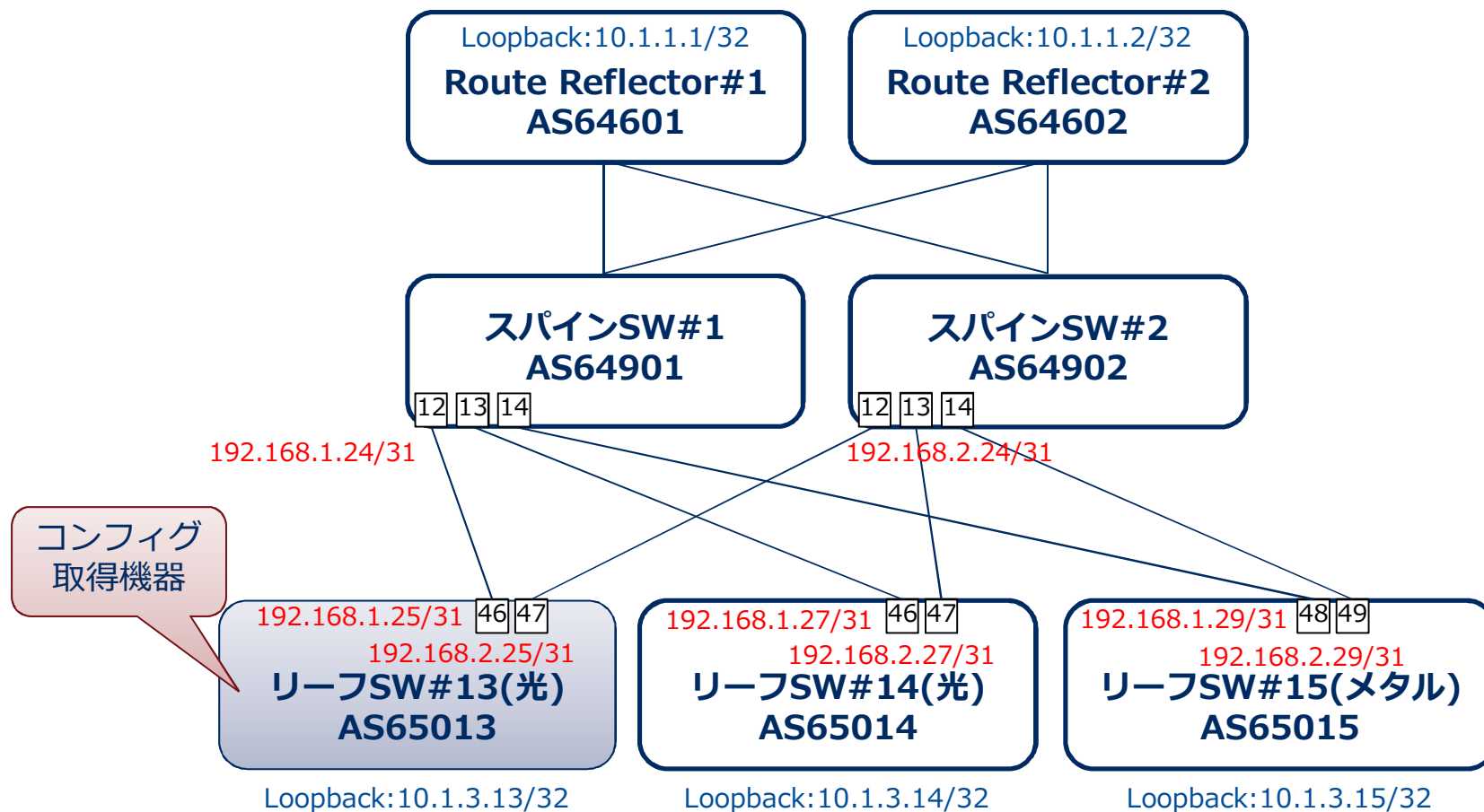
オーバーレイNW

- EVPN(Ethernet VPN)
 - MP-IBGP(Multi Protocol Internal BGP)
 - RRを配置することで、MP-IBGPネイバー数を削減する。
 - VXLAN(Virtual eXtensible Local Area Network)

検証構成

スパインSW
QFX5100-48S
リーフSW
QFX5100-48S or QFX5100-48T

物理構成図



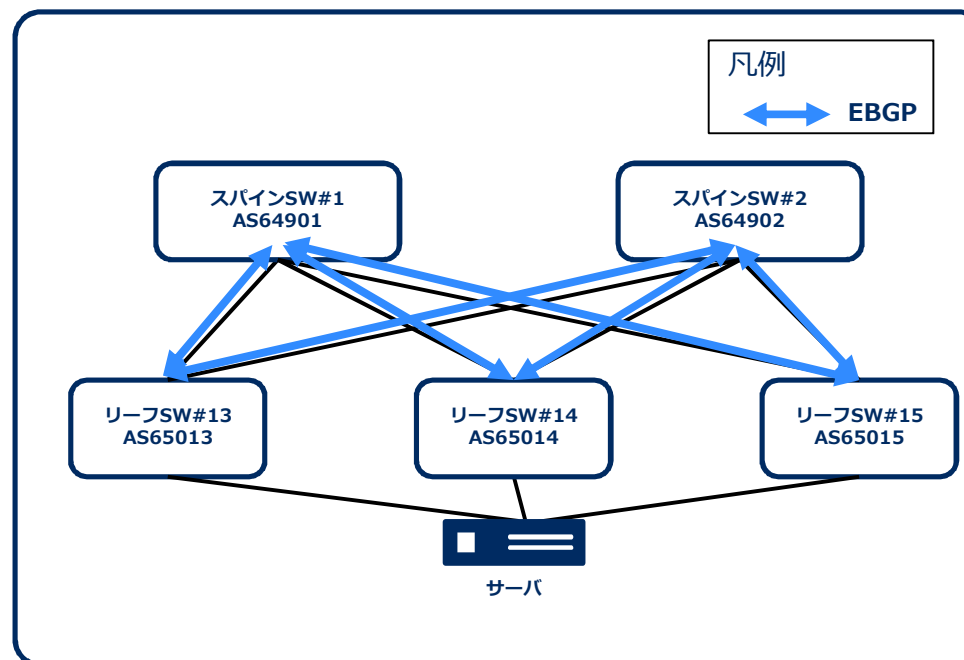
IPファブリック構成

EBGPによるフルメッシュ構成

- 機器毎にASを分け、フルメッシュEBGPネイバーを確立する。

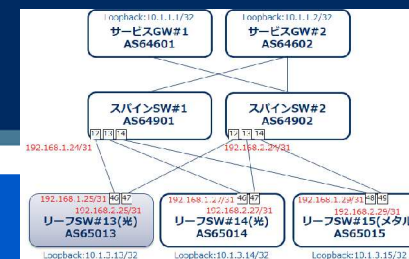
BFD(Bidirectional Forwarding Detection)による高速切替

- 障害発生時の切り替わり時間を考慮し、EBGPに対してBFDを動作させる。



アンダーレイNW[1/2]

IPファブリックのコンフィグ



1. set interfaces **xe-0/0/0** description zwsa1zd001_P3
2. set interfaces **xe-0/0/0** flexible-vlan-tagging
3. set interfaces **xe-0/0/0** mtu 9216
4. set interfaces **xe-0/0/0** encapsulation extended-vlan-bridge
5. set interfaces **xe-0/0/0** unit 120 vlan-id 120
6. set interfaces **xe-0/0/46** description To-wnpssjd001-0/0/12
7. set interfaces **xe-0/0/46** mtu 9216
8. set interfaces **xe-0/0/46** unit 0 family inet address **192.168.1.25/31**
9. set interfaces **xe-0/0/47** description To-wnpssjd002-0/0/12
10. set interfaces **xe-0/0/47** mtu 9216
11. set interfaces **xe-0/0/47** unit 0 family inet address **192.168.2.25/31**
12. set interfaces em0 unit 0 family inet address **10.38.0.244/16**

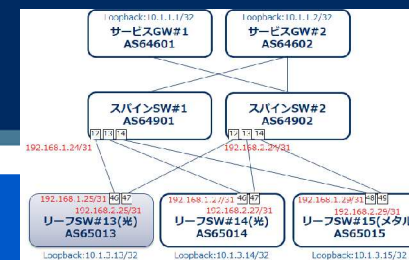
I/F設定

13. set interfaces lo0 unit 0 family inet address **10.1.3.13/32**
14. set routing-options forwarding-table export load-balance
15. set routing-options forwarding-table ecmp-fast-reroute
16. set routing-options router-id **10.1.3.13**

BGP設定

アンダーレイNW[2/2]

IPファブリックのコンフィグ



1. set protocols bgp group IP-Fabric-EBGP type external
2. set protocols bgp group IP-Fabric-EBGP mtu-discovery
3. set protocols bgp group IP-Fabric-EBGP export send-direct
4. set protocols bgp group IP-Fabric-EBGP local-as 65013
5. set protocols bgp group IP-Fabric-EBGP multipath multiple-as
6. set protocols bgp group IP-Fabric-EBGP neighbor 192.168.1.24 peer-as 64901
7. set protocols bgp group IP-Fabric-EBGP neighbor 192.168.2.24 peer-as 64902

EBGP設定

8. set protocols bgp log-updown
9. set protocols bgp graceful-restart
10. set protocols bgp bfd-liveness-detection minimum-interval 1000
11. set protocols bgp bfd-liveness-detection multiplier 3
12. set protocols bgp bfd-liveness-detection session-mode automatic

BFD設定

EVPN/VXLAN構成

EVPN/VXLANによるコントロールプレーンとデータプレーンの分離

- コントロールプレーンでMAC学習させて、BUMトラフィックを最小化する。

MP-IBGPによるMACアドレス学習

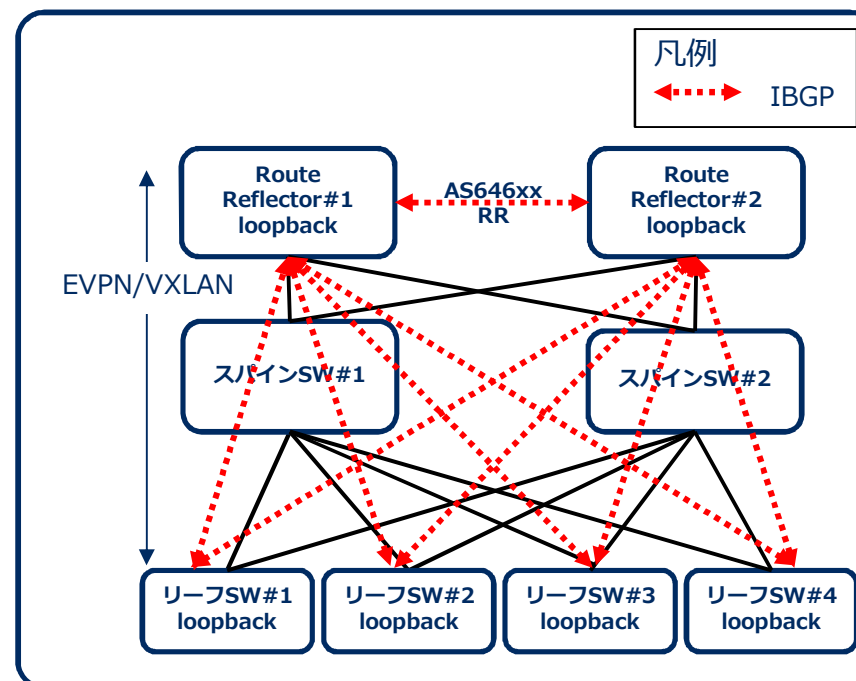
- EVPNはIPファブリック上で構築し、コントロールプレーンとしてMP-IBGPを利用する。

MP-IBGPのRR(Route Reflector)によるネイバー数の削減

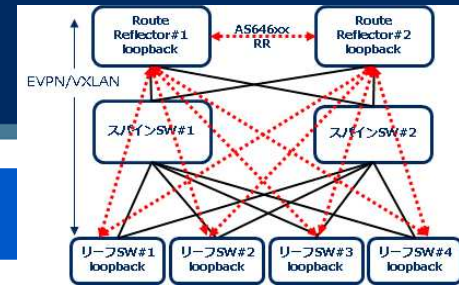
- RRを配置することで、MP-IBGPネイバー数を削減する。

BFDによる高速切替

- 障害発生時の切り替わり時間を考慮し、MP-IBGPでBFDを動作させる。



EVPN/VXLANのコンフィグ



EVPN-IBGP設定

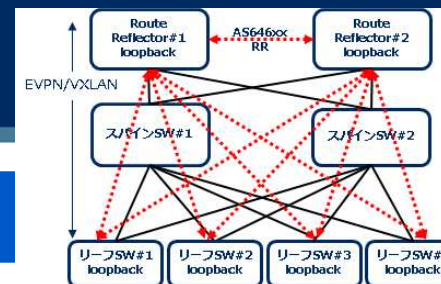
1. set protocols bgp group EVPN-IBGP type internal
2. set protocols bgp group EVPN-IBGP local-address 10.1.3.13
3. set protocols bgp group EVPN-IBGP family evpn signaling
4. set protocols bgp group EVPN-IBGP local-as 64601
5. set protocols bgp group EVPN-IBGP multipath
6. set protocols bgp group EVPN-IBGP bfd-liveness-detection minimum-interval 350
7. set protocols bgp group EVPN-IBGP bfd-liveness-detection multiplier 3
8. set protocols bgp group EVPN-IBGP bfd-liveness-detection transmit-interval minimum-interval 350
9. set protocols bgp group EVPN-IBGP neighbor 10.1.1.1
10. set protocols bgp group EVPN-IBGP neighbor 10.1.1.2

EVPN-VXLAN設定

11. set protocols evpn vni-options vni 10120 vrf-target target:64601:10120
12. set protocols evpn encapsulation vxlan
13. set protocols evpn multicast-mode ingress-replication
14. set protocols evpn extended-vni-list all
15. set protocols lldp interface all
16. set switch-options vtep-source-interface lo0.0
17. set switch-options route-distinguisher 10.1.3.13:1
18. set switch-options vrf-import EVPN-VXLAN_IMPORT
19. set switch-options vrf-target target:1:1

コンフィグ確認

BGPの設定やステータス



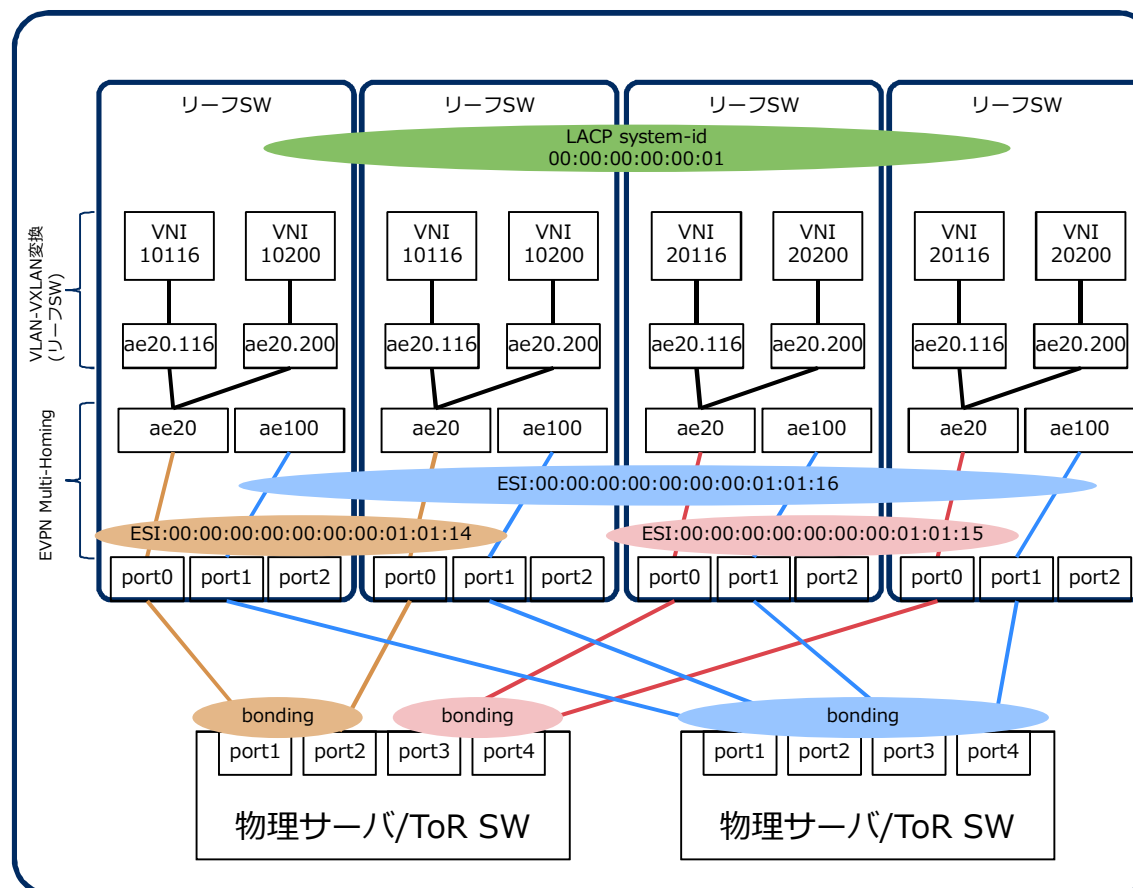
```
1. root@leaf013> show bgp summary | no-more
2. Threading mode: BGP I/O
3. Groups: 2 Peers: 4 Down peers: 0
4. Table      Tot Paths  Act Paths Suppressed  History Damp State  Pending
5. bgp.evpn.0
6.           422      211      0      0      0      0
7. inet.0
8.           162      112      0      0      0      0
9. Peer          AS   InPkt  OutPkt  OutQ  Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
10. 10.1.1.1      64601  783205 100505  0    27 4w3d 17:37:03 Establ オーバーレイNW
11.  __default_evpn__.evpn.0: 0/0/0/0
12.  bgp.evpn.0: 211/211/211/0
13.  default-switch.evpn.0: 211/211/211/0
14. 10.1.1.2      64601  780520 100121  0    28 4w3d 14:42:58 Establ
15.  __default_evpn__.evpn.0: 0/0/0/0
16.  bgp.evpn.0: 0/211/211/0
17.  default-switch.evpn.0: 0/211/211/0
18. 192.168.1.24  64901  130434 133094  0    0 5w6d 14:39:46 Establ アンダーレイNW
19.  inet.0: 56/81/81/0
20. 192.168.2.24  64902  130472 133082  0    0 5w6d 14:39:43 Establ
21.  inet.0: 56/81/81/0
```

VLAN-VXLAN変換(リーフSW)

- リーフSWでは、各VLAN IDごとにサブインターフェイスを作成し、VLANをユーザ・VLAN IDの組み合わせごとに一意のVNIに変換する。

Link Aggregation 方式

- EVPN Multi-Homingを利用する。
- Ethernet-Segmentを割り当てたLAG構成のリーフSWを冗長モード(active-active)で動作させて、トラフィックを物理サーバ/ToR SW~リーフSW間で冗長/ロードバランシングさせる。



物理結線の仮想化[1/3]

VLANタグ有り、LAGなし

1. set interfaces `xe-0/0/0` flexible-vlan-tagging
2. set interfaces `xe-0/0/0` mtu 9216
3. set interfaces `xe-0/0/0` encapsulation extended-vlan-bridge
4. set interfaces `xe-0/0/0` unit `116` vlan-id `116`
5. set vlans `VLAN_000001_116` interface `xe-0/0/0.116`
6. set vlans `VLAN_000001_116` vxlan vni `10116`
7. set vlans `VLAN_000001_116` vxlan ingress-node-replication

VLAN-VXLAN変換

VLANタグ有り、Static LAG

1. set interfaces `xe-0/0/0` ether-options 802.3ad `ae20`
2. set interfaces `ae20` flexible-vlan-tagging
3. set interfaces `ae20` mtu 9216
4. set interfaces `ae20` encapsulation extended-vlan-bridge
5. set interfaces `ae20` esi `00:00:00:00:00:00:00:01:01:14`
6. set interfaces `ae20` esi all-active
7. set interfaces `ae20` unit `116` vlan-id `116`
8. set vlans `VLAN_000001_116` interface `ae20.116`
9. set vlans `VLAN_000001_116` vxlan vni `10116`
10. set vlans `VLAN_000001_116` vxlan ingress-node-replication

LAG設定

物理結線の仮想化[2/3]

VLANタグ有り、LACP(Link Aggregation Control Protocol)

1. set interfaces `xe-0/0/0` ether-options 802.3ad `ae20`
2. set interfaces `ae20` flexible-vlan-tagging
3. set interfaces `ae20` mtu 9216
4. set interfaces `ae20` encapsulation extended-vlan-bridge
5. set interfaces `ae20` esi `00:00:00:00:00:00:01:01:14`
6. set interfaces `ae20` esi all-active
7. set interfaces `ae20` aggregated-ether-options lacp active
8. set interfaces `ae20` aggregated-ether-options lacp periodic fast
9. set interfaces `ae20` aggregated-ether-options lacp system-id `00:00:00:00:00:01`
10. set interfaces `ae20` unit `116` vlan-id `116`
11. set vlans `VLAN_000001_116` interface `ae20.116`
12. set vlans `VLAN_000001_116` vxlan vni `10116`
13. set vlans `VLAN_000001_116` vxlan ingress-node-replication

LAG設定

LACP設定

物理結線の仮想化[3/3]

VLANタグ無し、LACP(Link Aggregation Control Protocol)

1. set interfaces `xe-0/0/0` ether-options 802.3ad `ae20`
2. set interfaces `ae20` flexible-vlan-tagging
3. set interfaces `ae20` mtu 9216
4. set interfaces `ae20` encapsulation extended-vlan-bridge
5. set interfaces `ae20` native-vlan-id `116`
6. set interfaces `ae20` esi `00:00:00:00:00:00:00:01:01:14`
7. set interfaces `ae20` esi all-active
8. set interfaces `ae20` aggregated-ether-options lacp active
9. set interfaces `ae20` aggregated-ether-options lacp periodic fast
10. set interfaces `ae20` aggregated-ether-options lacp system-id `00:00:00:00:00:01`
11. set interfaces `ae20` unit `116` vlan-id `116`
12. set vlans `VLAN_000001_116` interface `ae20.116`
13. set vlans `VLAN_000001_116` vxlan vni `10116`
14. set vlans `VLAN_000001_116` vxlan ingress-node-replication

タグ無し設定

LAG設定

LACP設定

デモ用の初期設定

デモ用初期設定(リーフSW)

xe-0/0/6(対wppkhct003)

1. set interfaces xe-0/0/6 flexible-vlan-tagging
2. set interfaces xe-0/0/6 native-vlan-id **10**
3. set interfaces xe-0/0/6 mtu 9216
4. set interfaces xe-0/0/6 encapsulation extended-vlan-bridge
5. set interfaces xe-0/0/6 unit 10 vlan-id 10
6. set vlans VLAN_0000005_10 interface xe-0/0/6.10
7. set vlans VLAN_0000005_10 vxlan vni **50010**
8. set vlans VLAN_0000005_10 vxlan ingress-node-replication
9. set policy-options community 0000005_50010 members target:64601:50010
10. set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50010 from community 0000005_50010
11. set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50010 then accept
12. set protocols evpn vni-options vni 50010 vrf-target target:64601:50010

デモ用初期設定(リーフSW)

xe-0/0/10(対wppkhct004)

1. set interfaces xe-0/0/10 flexible-vlan-tagging
2. set interfaces xe-0/0/10 native-vlan-id **20**
3. set interfaces xe-0/0/10 mtu 9216
4. set interfaces xe-0/0/10 encapsulation extended-vlan-bridge
5. set interfaces xe-0/0/10 unit 20 vlan-id 20
6. set vlans VLAN_0000005_20 interface xe-0/0/10.20
7. set vlans VLAN_0000005_20 vxlan vni **50020**
8. set vlans VLAN_0000005_20 vxlan ingress-node-replication
9. set policy-options community 0000005_50020 members target:64601:50020
10. set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50020 from community 0000005_50020
11. set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50020 then accept
12. set protocols evpn vni-options vni 50020 vrf-target target:64601:50020

デモ用初期設定(リーフSW)

xe-0/0/30(対wppkhct009)

1. set interfaces xe-0/0/30 flexible-vlan-tagging
2. set interfaces xe-0/0/30 native-vlan-id **10**
3. set interfaces xe-0/0/30 mtu 9216
4. set interfaces xe-0/0/30 encapsulation extended-vlan-bridge
5. set interfaces xe-0/0/30 unit 10 vlan-id 10
6. set vlans VLAN_0000005_10 interface xe-0/0/30.10
7. set vlans VLAN_0000005_10 vxlan vni **50010**
8. set vlans VLAN_0000005_10 vxlan ingress-node-replication
9. set policy-options community 0000005_50010 members target:64601:50010
10. set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50010 from community 0000005_50010
11. set policy-options policy-statement EVPN-VXLAN_IMPORT term 0000005_50010 then accept
12. set protocols evpn vni-options vni 50010 vrf-target target:64601:50010

デモ用初期設定(サーバ)

IPアドレスとVLAN付与

wppkhct003

- ens29f1に10.0.0.100/24を設定
 - コマンド : ip addr add 192.168.251.100/24 dev ens29f1
 - コマンド : ip route add default via 192.168.251.254

wppkhct004

- ens29f1に10.1.1.100/24を設定
 - コマンド : ip addr add 10.36.15.100/24 dev ens29f1

wppkhct009

- ens29f1に10.0.0.150/24を設定
 - コマンド : ip addr del 10.36.15.150/24 dev ens29f1
 - コマンド : ip route add default via 10.36.15.253
 - コマンド : ip link add link ens29f1 name ens29f1.114 type vlan id 114
 - コマンド : ip link add link ens29f1 name ens29f1.111 type vlan id 111
 - コマンド : ip addr add 10.36.15.150/24 dev ens29f1.111
 - コマンド : ip addr add 192.168.251.150/24 dev ens29f1.114

 **Orchestrating** a brighter world

NEC