

JANOG 53 - Day3 2024年1月19日(金) 13:30~14:30

# 次世代OpenStack NFV基盤の検証

---

KDDI株式会社 技術統括本部 ネットワーク開発本部

2024/01/19



# アジェンダ

1. 登壇者紹介
2. 背景/次世代NFV基盤（辻）
  1. 背景 / IPv6化・基盤共通化・ラージスケール
3. IPv6苦労点（高橋さん）
  1. IPv6 アドレス設計
  2. DHCPv6/PXE
  3. BGP+RA
4. BGP苦労点（松本さん）
  1. L2が得意なDCNソリューションとの相性問題
  2. RouterID (ipv6) /ASNどうやって決めるのか？
  3. マルチNICでの経路制御
  4. マルチパスとIP ANYCAST
5. Large scale + 製品版OSSの苦労点（横山さん）
  1. Configの複雑化
  2. 設定のタイムアウト
6. ディスカッションタイム（15分程度）

# 登壇者紹介



**辻 広志**

**所属:**  
 技術統括本部  
 ネットワーク開発本部

**担当業務:**  
 IaaSチーム テックリード的立場

**一言:**  
 ・ 本日はプロデューサー業です



**高橋 宣行**

**所属:**  
 技術統括本部  
 ネットワーク開発本部  
 共通プラットフォーム技術部

**担当業務:**  
 IaaSチーム OpenStack 担当

**一言:**  
 初JANOG・初福岡  
 よろしくお願ひします！



**松本 良輔**

**所属:**  
 技術統括本部  
 ネットワーク開発本部  
 共通プラットフォーム技術部

**担当業務:**  
 IaaSチーム Ceph(SDS) 担当

**一言:**  
 Ceph触る人増えたらいいな



**横山 周太**

**所属:**  
 技術統括本部  
 ネットワーク開発本部  
 共通プラットフォーム技術部

**担当業務:**  
 IaaSチーム OpenStack 担当

**一言:**  
 ソフトウェアエンジニア (自称)

# 背景/What is 次世代OpenStack NFV基盤

JANOG49 – Day3 SAIGOU 16:45-17:45

**KDDI固定電話ネットワーク、NFV化の5年間の道のり**  
KDDI株式会社 次世代自動化開発本部 コアネットワーク部  
 2022/1/28

臥新管胆

- 今回作った基盤で将来にわたる拡張性を見込むことが厳しい
- 2018年から次期基盤の検討を開始（並行開発）
  - ・ ところあるVNFの構築に合わせて実施
  - ・ より一般的なOpenStackの機能を使用
    - Cinder / Swift / NeutronのL3機能etc..
- 反省を生かして共通NFV基盤として企画
  - ・ 共通NFV基盤には明確にポリシーを定める
  - ・ 要件が合わない部分はVNF側が合わせる

**JANOG 49**

## 第1世代（運用中）

KDDI固定電話ネットワーク、NFV化の5年間の道のり @ JANOG49(2022/02)

OpenStack NFV基盤のバージョンアップ  
 運用改善を内製対応した話  
KDDI株式会社 米村 淳、松本 良輔  
 2022/5/31

仮想化基盤について

**OSSを活用したベンダニュートラルな仮想化基盤を構築**

- ・ OpenStack + Cephで仮想化基盤を構築
  - ・ Red Hat OpenStack Platform 16.1
  - ・ Red Hat Ceph Storage 4

様々な音声サービスを提供するVMを構築。顧客商用サービス開始。

VA M社 M社 ... VB M社 M社 ... VC M社 M社 ...

Volume Volume Volume ...

OpenStack Ceph

HW

KDDI IP電話網 他社網 移動体網

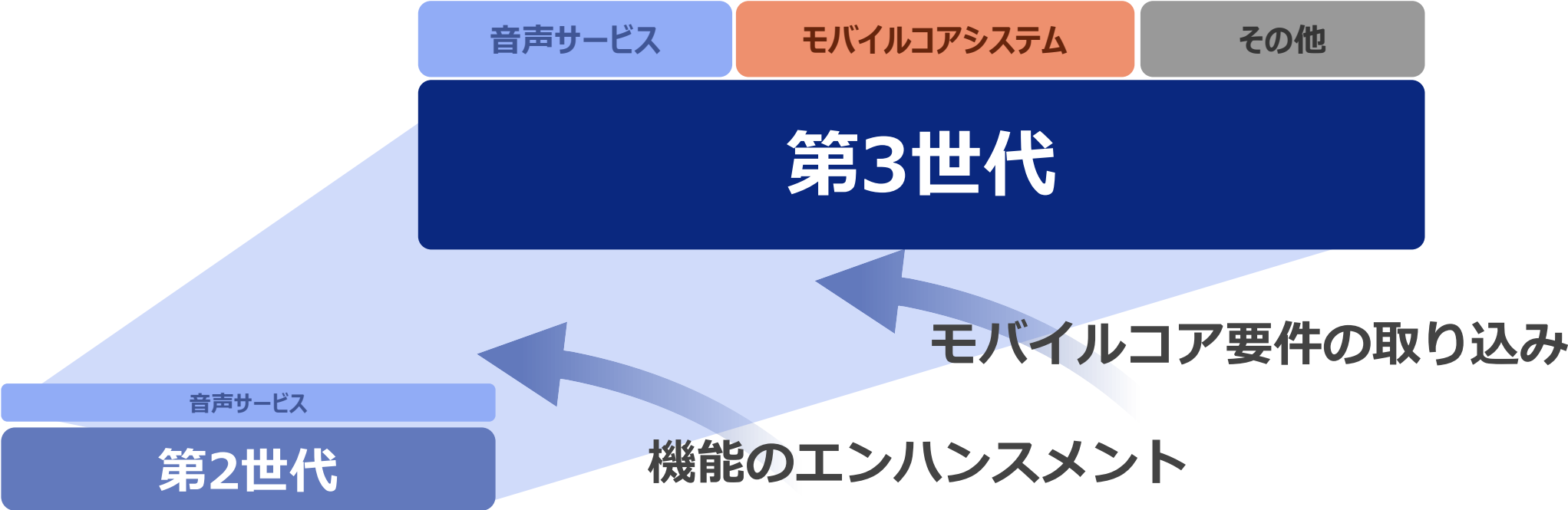
**CODT2022**

## 第2世代（運用中）

OpenStack NFV基盤のバージョンアップと運用改善を内製対応した話（KDDI編） @CODT2022(2022/07)

# 今回はさらに進んだ第3世代(構築中)の話です

どうやら (KDDI編) の看板を下ろす時が来たようです



# 背景/Concept of 次世代OpenStack NFV基盤

## 脱「仮想化基盤」、「プライベートクラウド」化



利用者

- ✓ システム開発の効率化
- ✓ スキルやアセットの共有
- ✓ 効率的なファシリティ利用
- ✓ パブリッククラウドとの連携



プライベートクラウドをKDDIで広く利用していくことで生じるメリット

- ✗ Excelの申請書
- ✗ 管理者側の作業待ち・承認待ち
- ✓ AZごとの独立性を高め、障害影響を局所化

- 共通的なインフラ・プラットフォームとして利便性を向上させセルフサービスやポータルによる利用者自身での自己完結。
- クラウドっぽい考え方によるアベイラビリティの向上



## IPv6

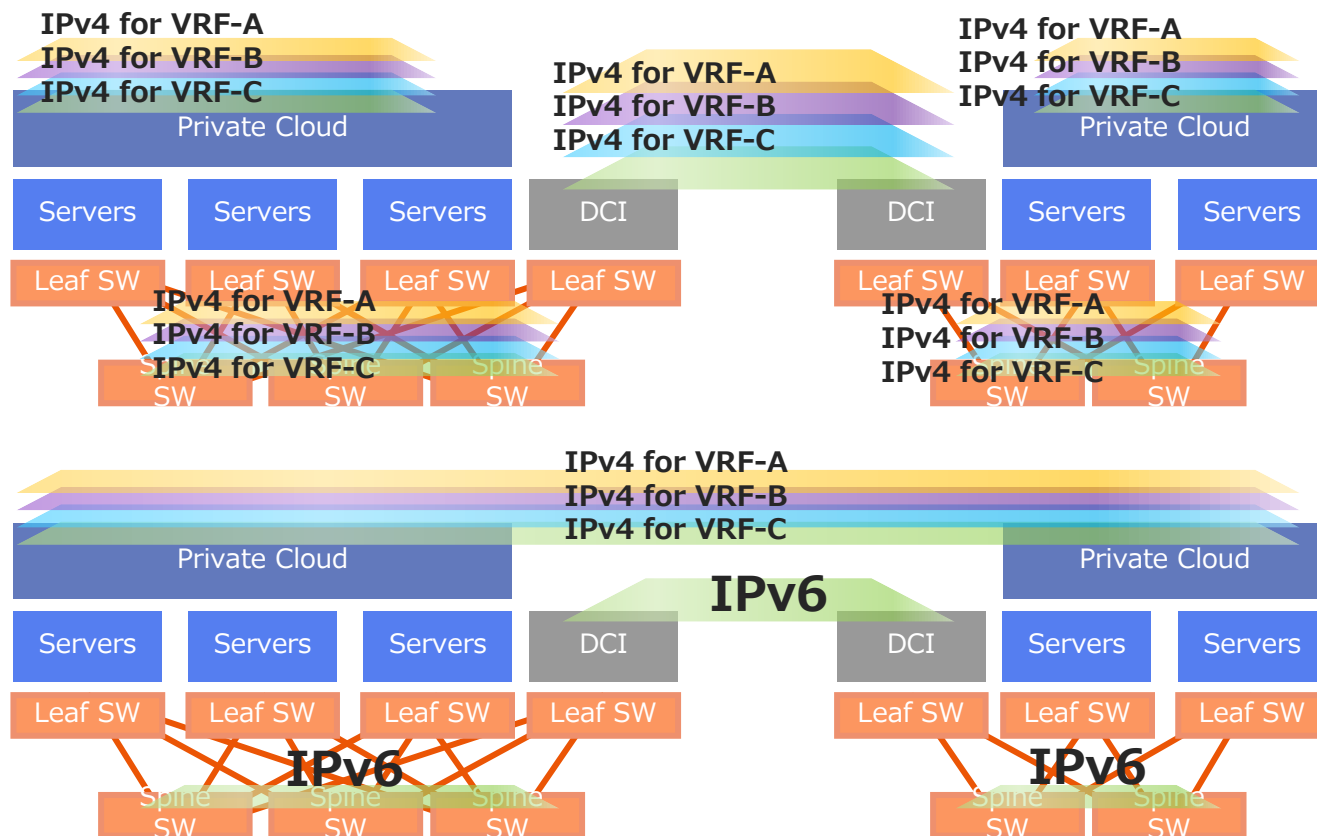
社内NWを共通化する上での課題を解決

## Routing on the host

脱・L2冗長化とロードバランシングの両立

他にも色々あるのですが、また別の機会に…

歴史的経緯により社内に多数存在するIPv4面  
様々な設備やシステム間での統合上の足かせになりがち。  
「誰かがこれをやらねばならぬ 期待の人が俺達ならば♪」



- ✗ IPv4のアドレス空間が被っており VRFの設定が各所で必要
- ✗ VRF統合は不可能
- ✗ 様々なレイヤーで複雑な処理を実施
- ✓ 基本的な統合先をIPv6にする
- ✓ 物理的なハードウェアやネットワークを極力シンプル化
- ✓ 複雑さはすべてクラウド上(ソフトウェア)に寄せる





## 400G Break out + 100G NIC

ServerのUplinkでの効率的なネットワーク帯域の利用(Active/Active)  
Active/Backupで100Gbps余らせるのはもったいなすぎる

## 広帯域化 + ECMPでの負荷分散

共通化に伴うネットワークトラフィックの増大や柔軟なNW構成の変更への対応

## 脱・L2冗長化ソリューション

ベンダーニュートラルでかつ楽な方法の模索  
**誰かがこれをやらねばならぬ 期待の人 (ry**



## 2000+

共通化していくとなると今まで見ていたドメイン以上の規模に

## 安定性

どの大きさまで行けるか？適切なサイジングは何か？



# あらためて本日のアジェンダ

1. 登壇者紹介
2. 背景/次世代NFV基盤（辻）
  1. 背景 / IPv6化・基盤共通化・ラージスケール
3. IPv6苦労点（高橋さん）
  1. IPv6 アドレス設計
  2. DHCPv6/PXE
  3. BGP+RA
4. BGP苦労点（松本さん）
  1. L2が得意なDCNソリューションとの相性問題
  2. RouterID (ipv6) /ASNどうやって決めるのか？
  3. マルチNICでの経路制御
  4. マルチパスとIP ANYCAST
5. 製品版OSSの苦労 + Large scale編（横山さん）
  1. Configの複雑化
  2. 設定のタイムアウト
6. ディスカッションタイム



# IPv6の苦労編

# IPv6苦労点 アドレス設計/Prefix/Back born

## ■ 地域別、用途別で分けることで経路数を削減

### ● 地域別：経路数をいたずらに増やさない

- 細かな場所の指定 (e.g.DCフロア等) も検討したが、今後の柔軟性 (上流ルータの収容変更等) も考慮してざっくりとしたエリアに限定
- 地域ごとに経路を集約
- IP Anycast用の「メタ地域」も設定

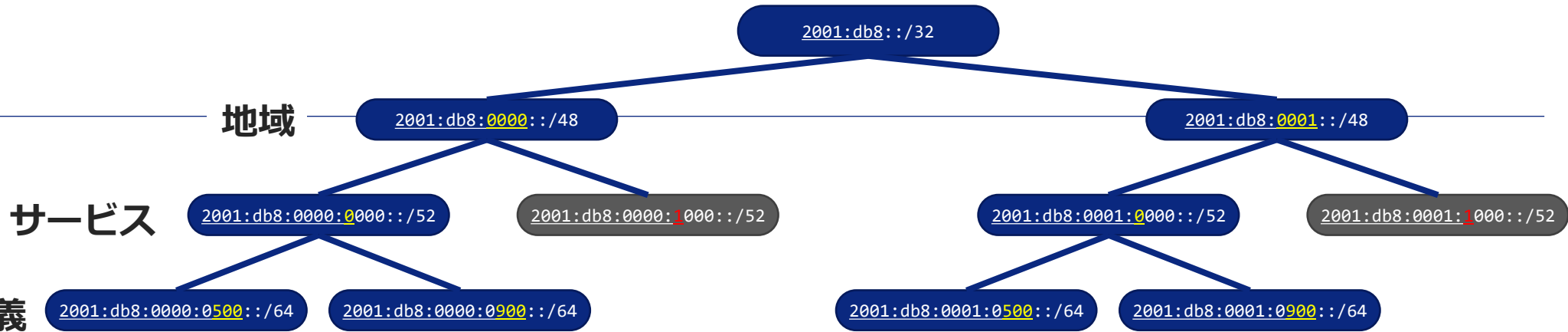
### ● サービス別：

- ビットを見てどのサービスか判断

バックボーン

DC内部

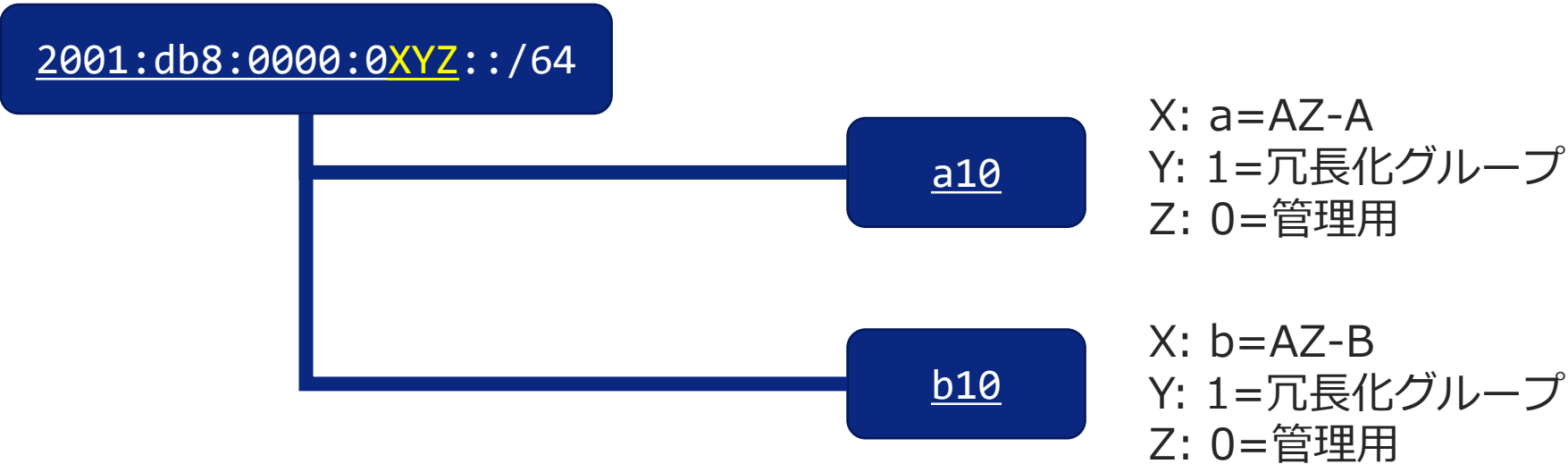
サービス毎に定義



# IPv6苦労点 アドレス設計/Prefix/サービス

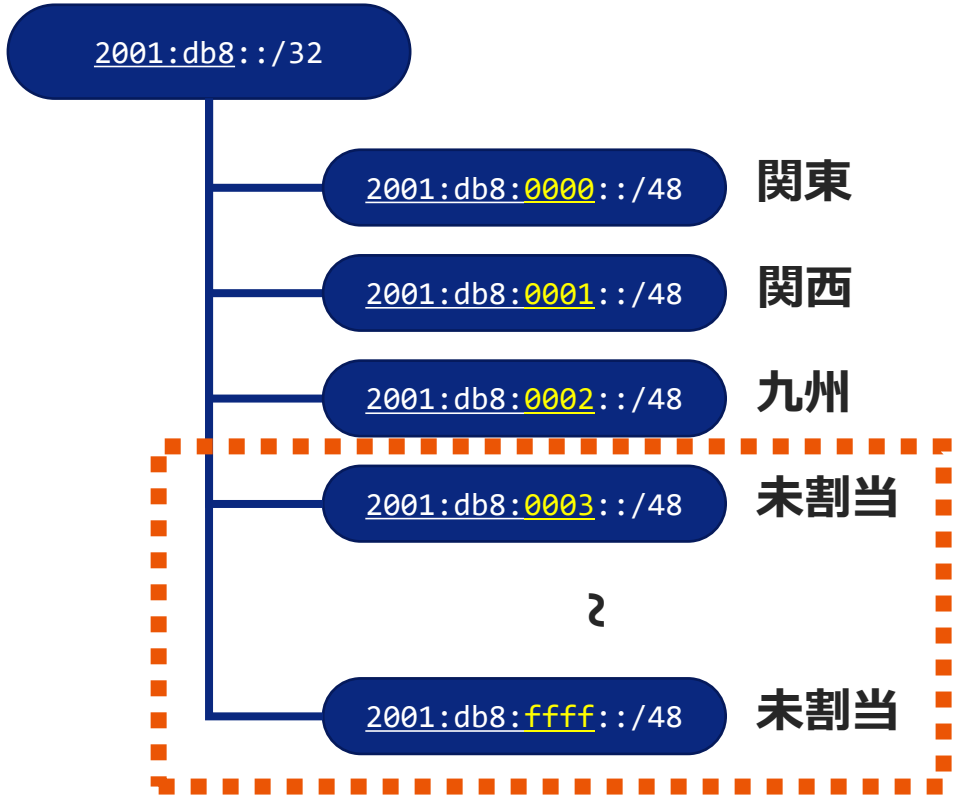
## ■ サービスごとにポリシーを作成

- 3バイト(24 bit/16,777,216プレフィックス)でサービスでの利用を定義
  - ・ 用途・AZ・AZ内でのリーフスイッチ等の物理リソース
- IaaSの場合ほとんどがユーザ利用用の空間



# IPv6苦労点 アドレス設計/Prefix/余白

- ほとんどの領域は未割当（将来拡張用）
  - ガチガチのルールではなく頭からひとまず現状ルールでアサイン
  - 使い勝手が悪くなれば将来ルール変更も考える



# IPv6苦労点 アドレス設計/Host

- 用途別等で決めて適当に採番
  - 64bitもあるし、アルファベット(a-fもつかえるので)これも法則を決めやすい
- IPv6シングルスタックなのでIPv4アドレスを気にしないで済む

2001:db8:0000:0a10::aNNN/64

aNNN: 運用系サーバ

2001:db8:0000:0a10::bNNN/64

bNNN: ハイパーバイザー

2001:db8:0000:0a10:10:0:0:5/64

今回はIPv6シングルスタックなので  
IPv4とのデュアルとしてこういったことを  
しなくても済む



# IPv6苦労点 アドレス設計/やってみて

- 法則を覚えれば以外とIPアドレスを覚えやすい(IPアドレス固定の場合)
  - アドレス見てどのリージョン・どのAZかわかる
  - アドレス見てどのサーバタイプかわかる
- サービスのアサインを後ろに持ってくるとACL等が面倒

IaaS管理系だけがアクセス可能なフィルタを書きたい・・・

- ワイルドカードマスクで  
かけないことが多い  
(OpenStack Security groupとか)

`2001:db8:0000:0a10::/64`

`2001:db8:0000:0a20::/64`

`2001:db8:0001:0a10::/64`

# IPv6苦労点 DHCPv6(サマリ)

## ■ ほぼほぼDHCPv6およびそれに付随する技術で発生

### ● デフォルトゲートウェイの扱いが特にハマりやすし

- ゲートウェイやスタティックルートに対する扱い・RA/DHCPv6その他のアドレス自動設定
- ARP/ICMPv6 . . . .

### ● PXE/iPXE

- プレフィックス64bit固定実装: 検証のために細切れでサブネット当てたらGW設定がされない
- DHCPv6ステートフル固定実装: PXEシーケンス時にプロトコル固定
- DHCPv6 Relay agentが未対応
- PXEでデフォグ設定ができないGRUBのバグ

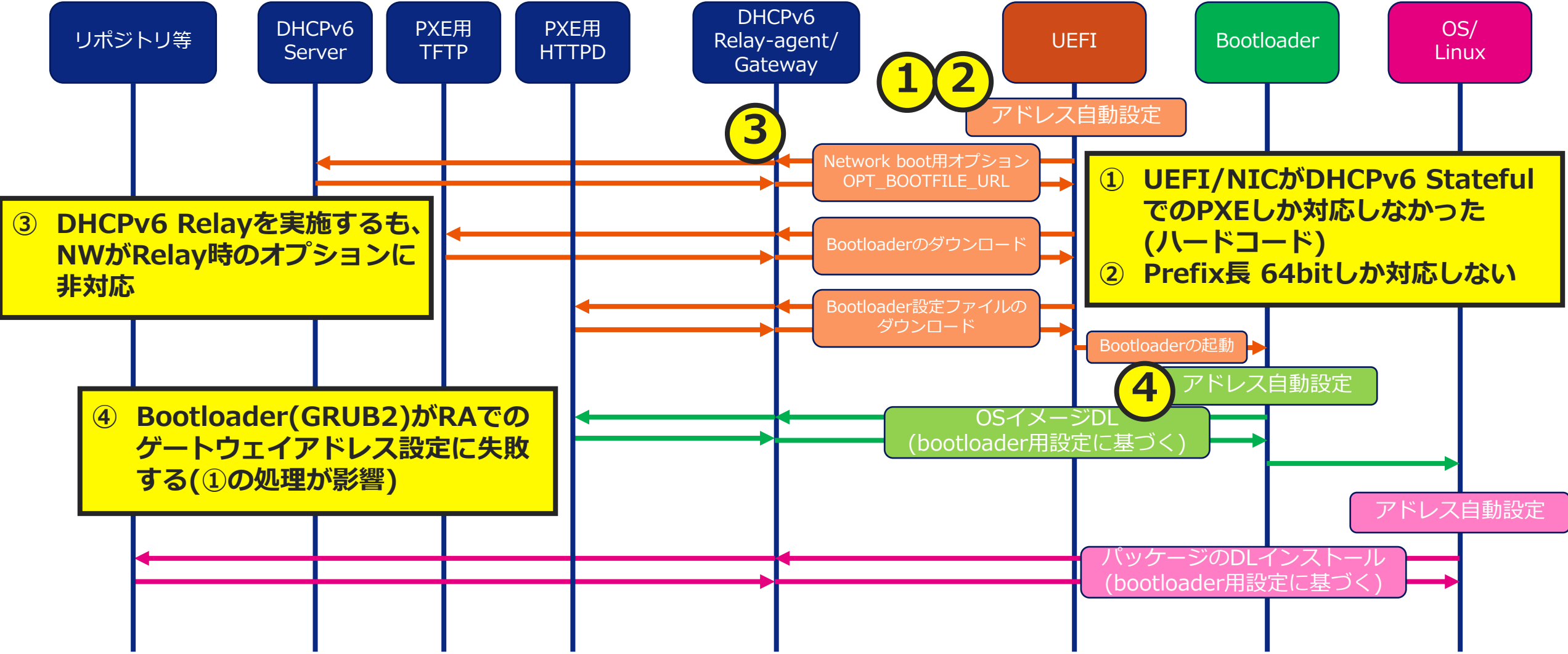
### ● BGP + RA

- BGPのセットアップまでのデフォグの扱い、BGPがUpした後のデフォグの扱い

<p>枚数足りなく本日は せなかった苦労</p>	<ul style="list-style-type: none"> <li>• <del>DHCPv6</del> <ul style="list-style-type: none"> <li>• <del>ゲストOSごとに挙動が違う</del></li> <li>• <del>サポートしていないアプライアンス(サブネット上でアドレス設定モードの混在不可)</del></li> <li>• <del>VMのブリッジインタフェースでデバイスIDが変わってしまう問題</del></li> </ul> </li> <li>• <del>OpenStack(ML2/OVN)がMetadata agent未対応</del></li> </ul>
------------------------------	--

# IPv6苦労点 DHCPv6(PXE全体像)

## ■ 当初意図したベアメタルサーバのインストール自動化で利用



# 補足:DHCPv6 (1/3)

## ■ 2つのモードがある (Stateful/Stateless)

- RAのオプションをみて「クライアント」で動作モードを制御

オプション		内容
Mフラグ		RFC4861: 1-bit "Managed address configuration" flag. When set, it indicates that addresses are available via Dynamic Host Configuration Protocol [DHCPv6]. (意訳: Trueの場合DHCPv6でアドレスを設定する)
Oフラグ		RFC4861: 1-bit "Other configuration" flag. When set, it indicates that other configuration information is available via DHCPv6. Examples of such information are DNS-related information or information on other servers within the network. (意訳: Trueの場合その他の設定、例えばDNSとかをDHCPv6から取得する)
プレフィックス通知	Aフラグ	RFC4861: 1-bit autonomous address-configuration flag. When set indicates that this prefix can be used for stateless address configuration as specified in [ADDRCONF]. (意訳: Trueの場合アドレスはDHCPv6を使わないステートレス設定方法でアドレス設定をする)

# 補足:DHCPv6 (2/3)

## SLAAC

ルータ  
など

fe80::1

オプション	値
Mフラグ	False
Oフラグ	False
プレフィックス通知	Aフラグ <b>True</b>
	プレフィックス 2001:db8::/64

fe80::2

サーバなど

- ゲートウェイは fe80::1
- アドレスはプレフィックス通知から自動生成

## DHCPv6 Stateless

ルータ  
など

fe80::1

DHCP  
サーバ

オプション	値
Mフラグ	False
Oフラグ	<b>True</b>
プレフィックス通知	Aフラグ <b>True</b>
	プレフィックス 2001:db8::/64

fe80::2

サーバなど

- ゲートウェイは fe80::1
- アドレスはプレフィックス通知から自動生成
- その他の情報があるっぽいのでDHCPサーバから取得

## DHCPv6 Stateful

ルータ  
など

fe80::1

DHCP  
サーバ

オプション	値
Mフラグ	<b>True</b>
Oフラグ	False or True
プレフィックス通知	Aフラグ False
	プレフィックス 2001:db8::/64

fe80::2

サーバなど

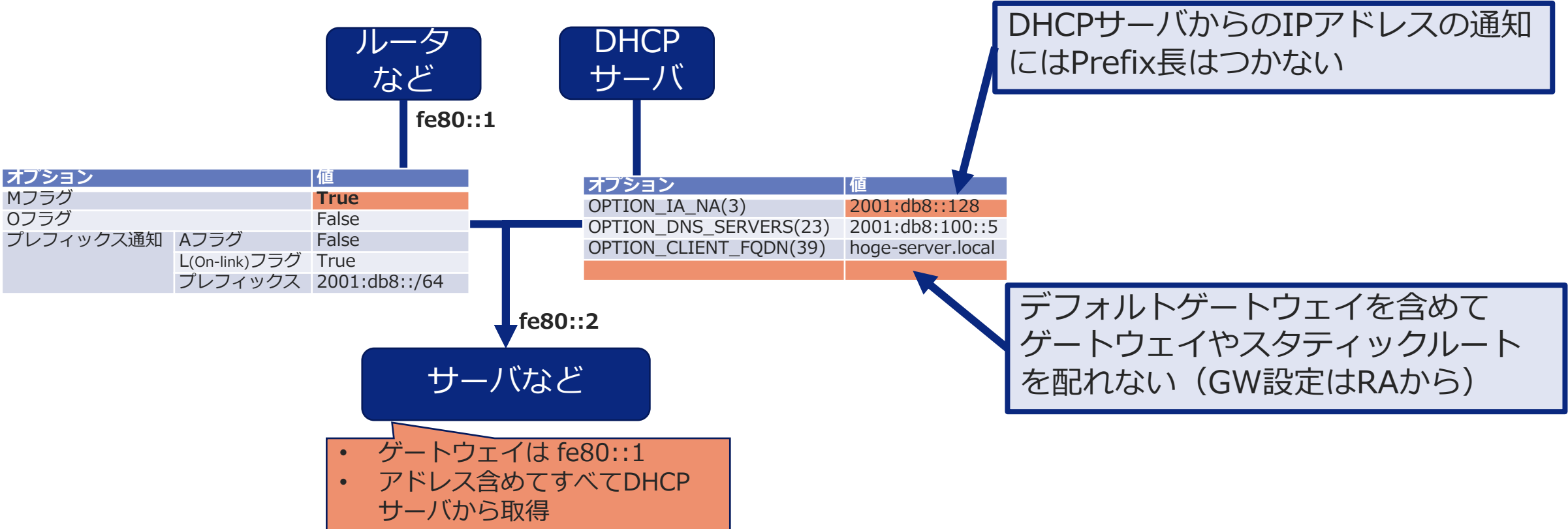
- ゲートウェイは fe80::1
- アドレス含めてすべてDHCPサーバから取得

# 補足:DHCPv6 (3/3)

## ■ 概ねon-linkのPrefix長の扱い方とゲートウェイ設定方法の扱い方ではまる

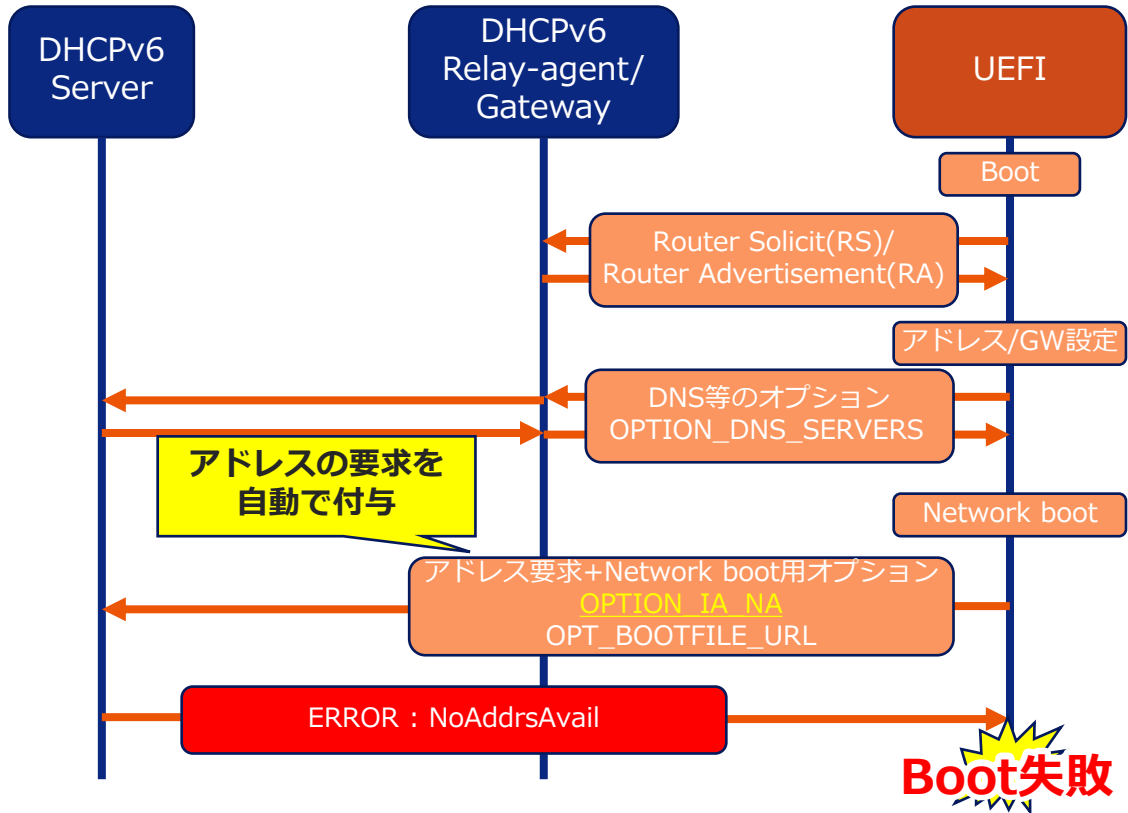
- クライアントによってこのあたりの動作が違ったりもする

### DHCPv6 Stateful



# IPv6苦労点 DHCPv6(PXE①)

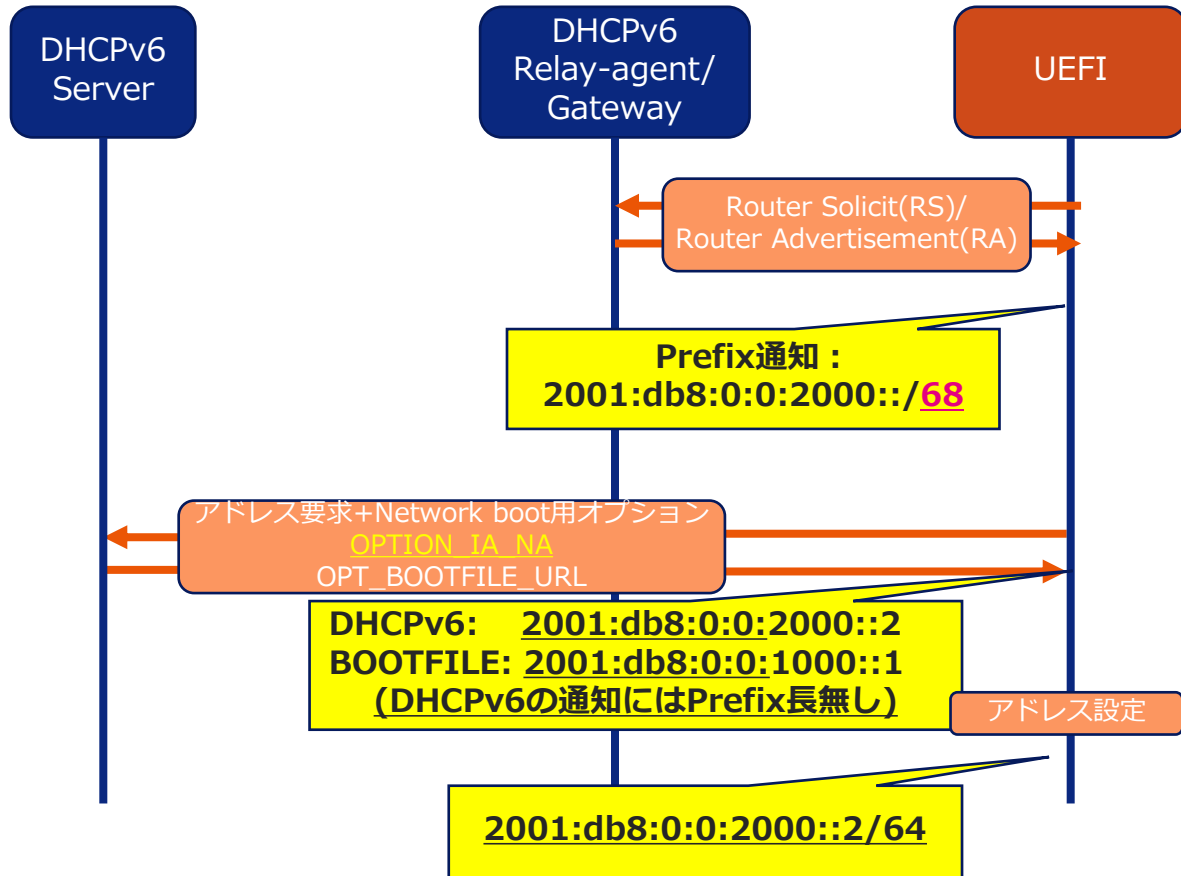
## ■ DHCPv6-“Stateless”を選択しようとしたがUEFI/NICがDHCPv6 Stateful でのPXEしか対応しなかった(ハードコード)



- UEFI/NICファームの実装上の問題
- 自分たちの力だけでは変更が難しいため、そのままDHCPv6 Statefulで実装を進める

# IPv6苦労点 DHCPv6(PXE②)

## ■ Prefix長が 64 bitしか対応しない (ハードコード)

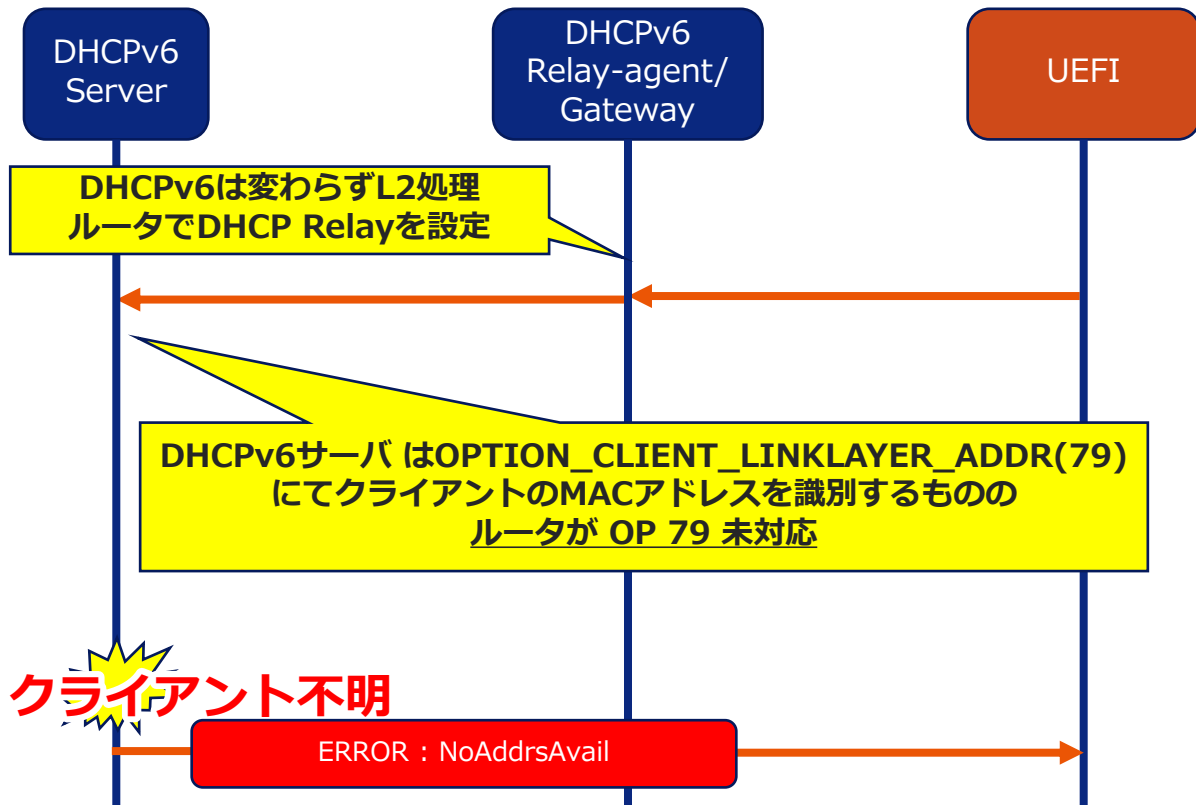


- NW側への設定追加等をサボって/64のサブネットを分割して利用しようとしたところアドレスアサインに失敗
- UEFI/NICファームの実装上の問題で/64以外のアドレスが入らない (64 bit決め打ち)
- /64を分割して/68で設定していたところ、L2解決しようとして失敗 (HTTPでイメージを取得できず)
- **RFC上も64 bitなので/64でNW含めて再設定**



# IPv6苦労点 DHCPv6(PXE③)

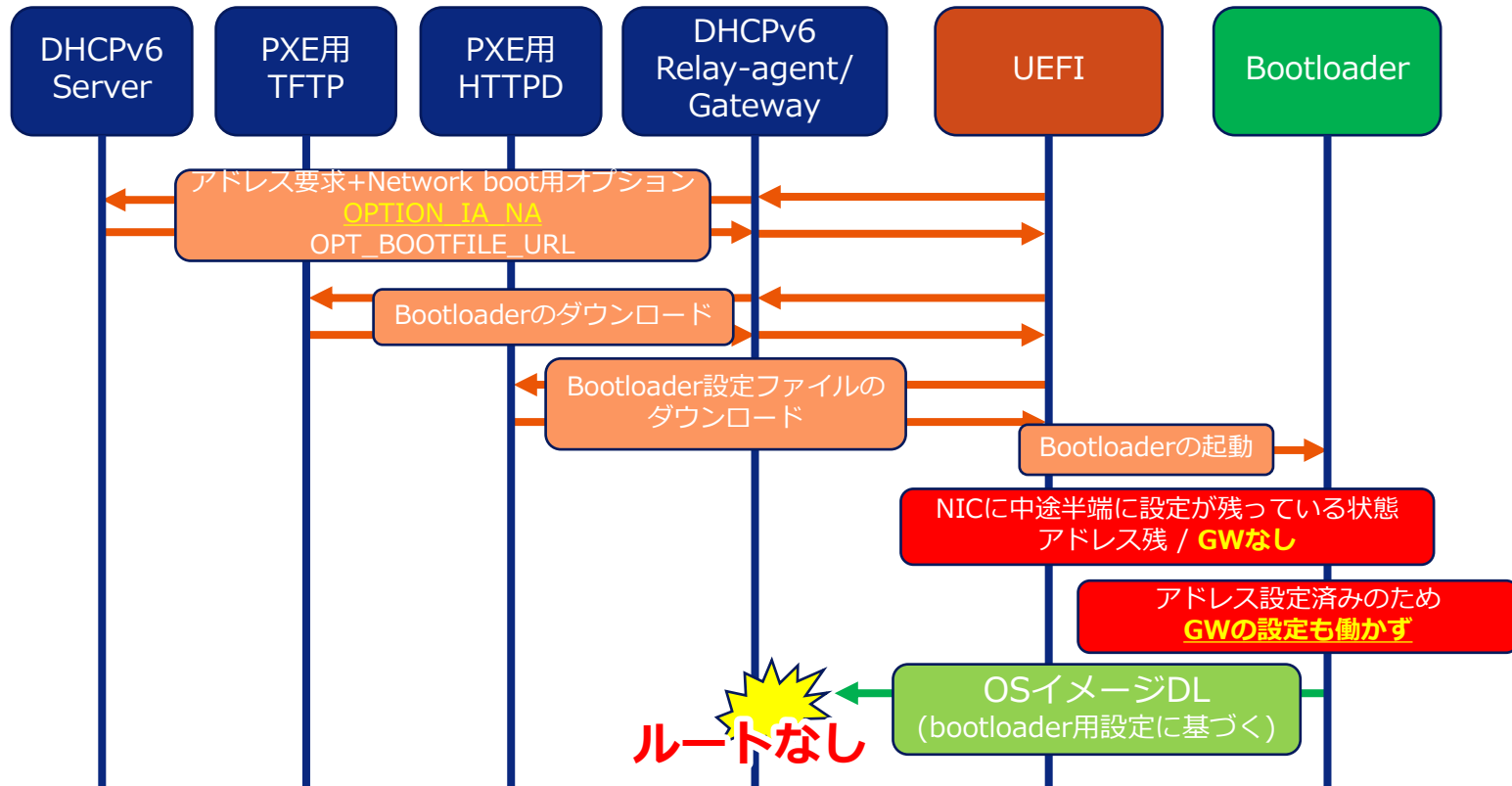
## ■ DHCP relay設定をしたルータがRelayの一部オプションに未対応



- DHCPv6サーバはクライアントを識別する上においてMACアドレスでIPアドレスや返すオプションを規定
- DHCP relay時にRelay agentを設定したルータがこのオプション (OPTION\_CLIENT\_LINKLAYER\_ADDR(79))に未対応
- 同一セグメントにDHCP RelayのためだけのSONICをVMで建てて回避 (ルータは最新バージョンではOp79に対応可能/とはいえそんなに古いバージョンではなかった…)

# IPv6苦労点 DHCPv6(PXE④)

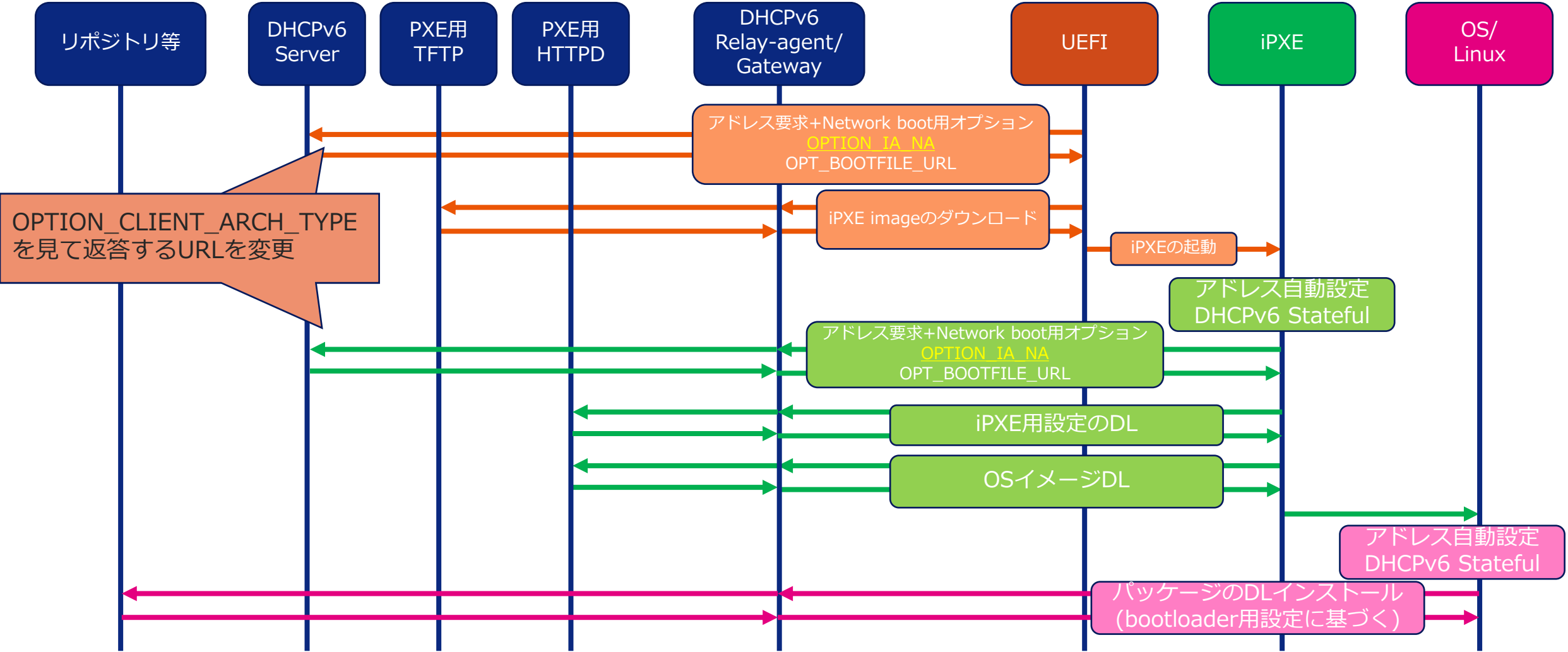
## ■ Bootloader(GRUB2)がRAでのゲートウェイアドレス設定に失敗する (UEFIでのアドレス設定の処理が影響)



- UEFI/NICファーム+ GRUB2の実装上の問題で GRUB 2 起動後にNICにアドレス設定が残存
- RAでのゲートウェイの設定が働かずルートなしでHTTPのファイル取得に失敗
- よりインテリジェントな iPXEを利用し HTTPブートを採用

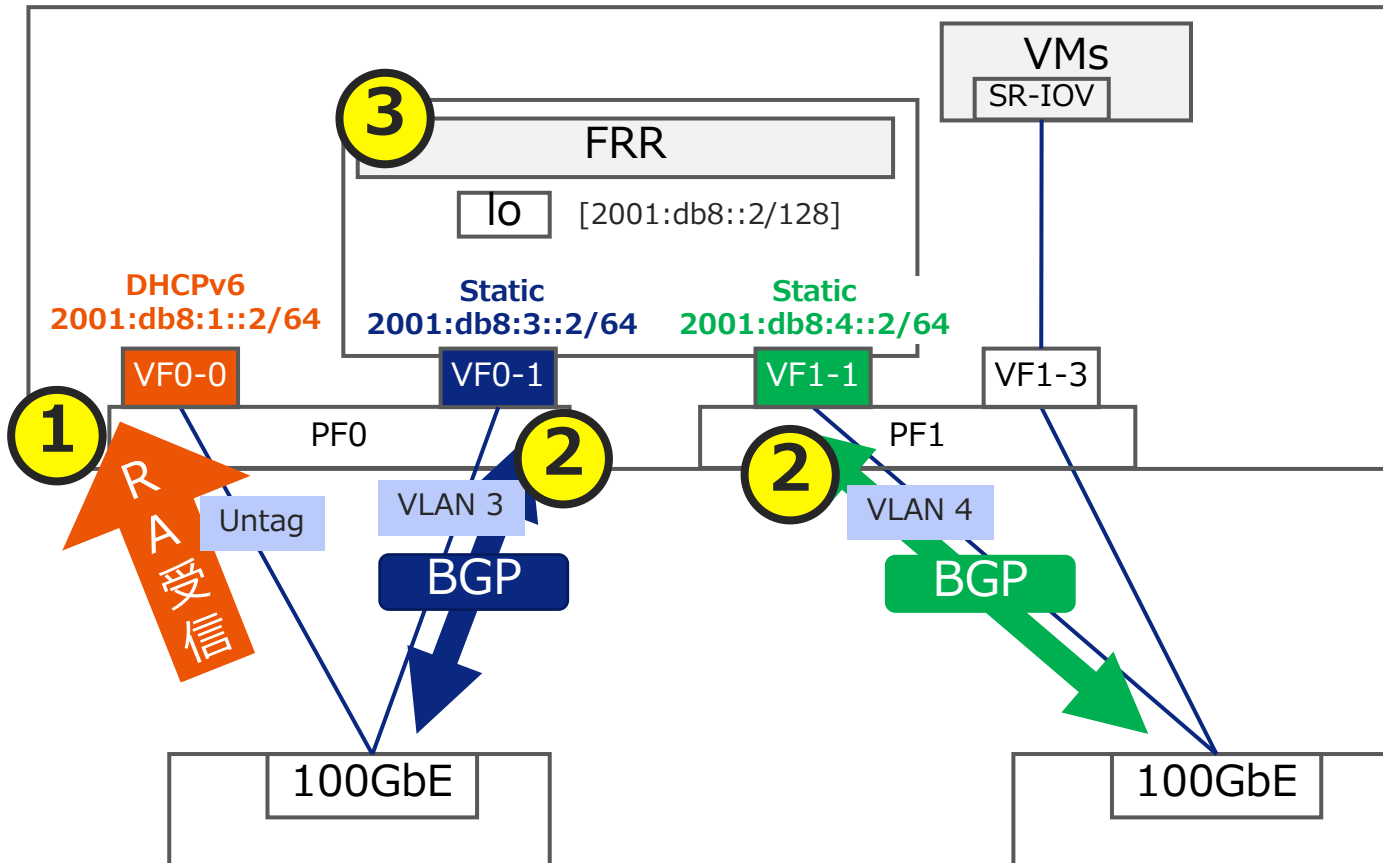
# 補足: iPXE/HTTPブート

## Clientのタイプを見分けてDHCPサーバで2段階のPXEブートを制御



# IPv6苦労点 BGPとRA

- BGP設定を行ったところRAでのデフォゲが残存し、FRRで勝てない
  - マルチNIC/マルチVLAN構成での問題



- ① サーバの初期SetupのためにNative VLANによるPXEブート+SSH経由でのAnsibleでのセットアップ
- ② FRRをインストールしBGP化
- ③ FRRでBGP Up後もRAのデフォゲが残存。FRRはカーネルのルート情報がAD値=0で固定、FRRで得た経路は絶対に勝てないためBGP経路がテーブルに乗らない(Native VLANなので未アサインのSR-IOVのVF等でも検知)

**適切なタイミングでのRA受信設定の停止(カーネルパラメータ等)が必要**



# BGPの苦勞編

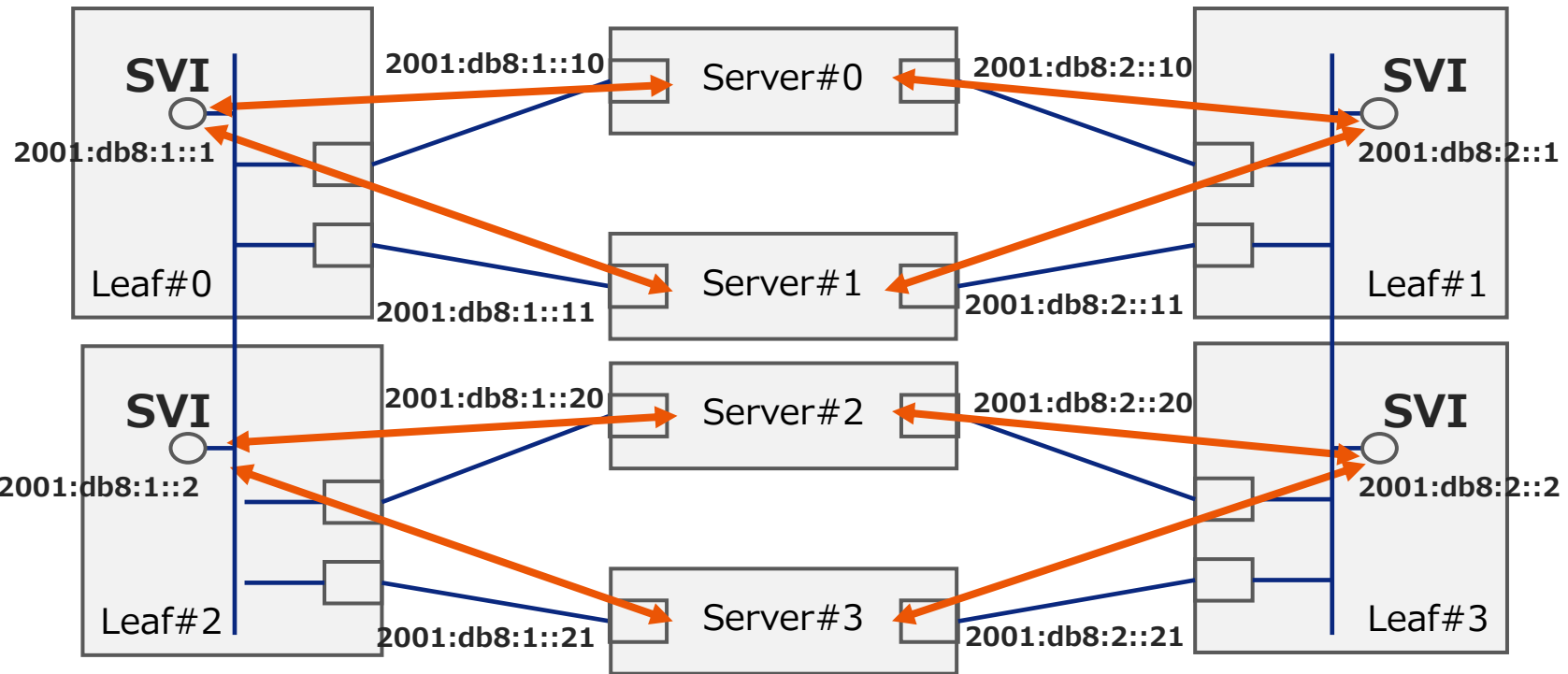


- BGP Unnumbered が使えない
- L2ソリューションをテナント向けには残さないといけないこと両立
- RouterIDが32bit / ASN(32bit/4Byte)どうするか?
- マルチNICでの経路制御
- ANYCAST/マルチパスでの偏り

# BGPで苦労点(BGP Unnumbered が使えない)

## ■ 採用したデータセンタNWのソリューションが未対応

- 後述するNFVの要件を満たすためにはL2やインテリジェントなNW機能が必要
- 某有名なデータセンタNWソリューションを導入
- BGP Unnumbered などのL3などに対しては機能が実装されておらず、設定の簡素化がNW側サーバ側共に難しく、各リンクごとのIPアドレス付与が必要

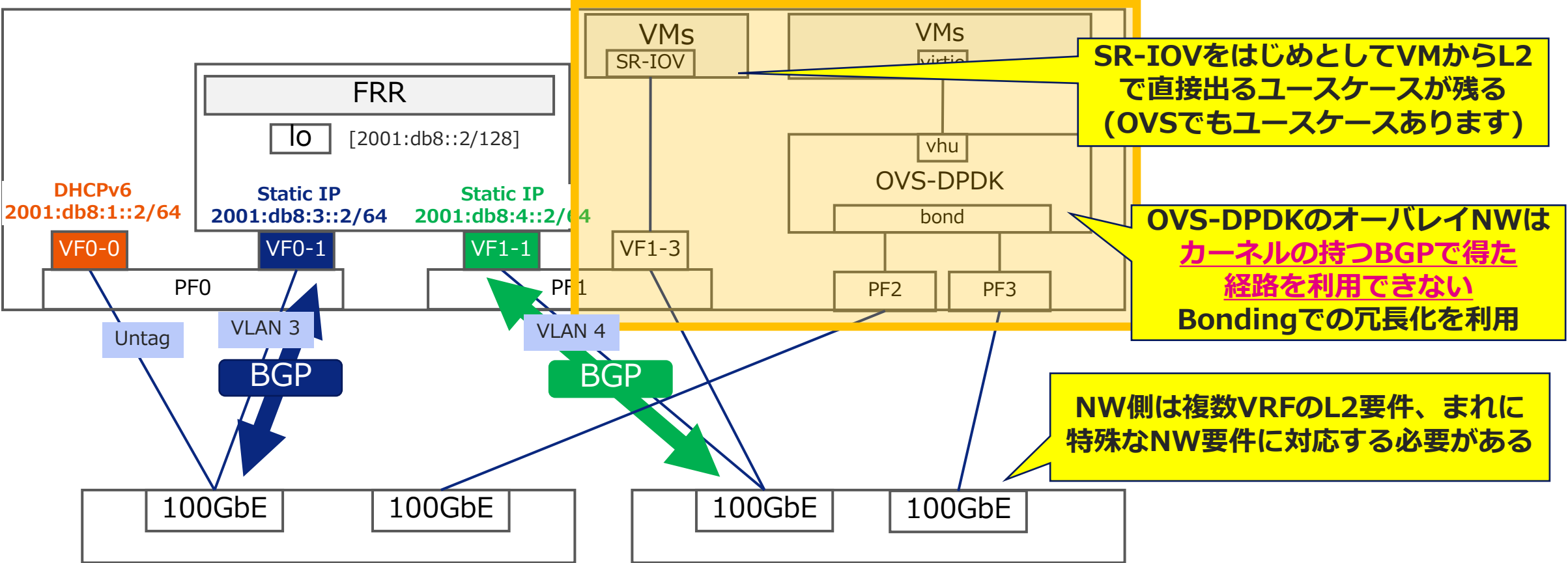


- Linkごとの設定投入はNW側に膨大な設定投入が必要なため **SVIを定義したバス接続の構成**
- サーバ側はある程度のサイズごとに**数台のLeafスイッチ単位等でVLANを分割しブロードキャストドメインを分割**
- **サーバ間の詳細経路は不要なため同一ブロードキャストドメイン内は同一ASN**
- IPv6前提だったのでその点は助かった (IPv4アドレス枯渇)

# BGPで苦労点(テナント向けL2サービスとの両立)

## ■ NFVといえばL2。OVS-DPDKやSR-IOVなどでL2要素を残す必要有り

- フルL3のシンプルなDCNWを組めない・採用できない
- 複数VRFでのL2NWを作る必要がある



SR-IOVをはじめとしてVMからL2で直接出るユースケースが残る (OVSでもユースケースあります)

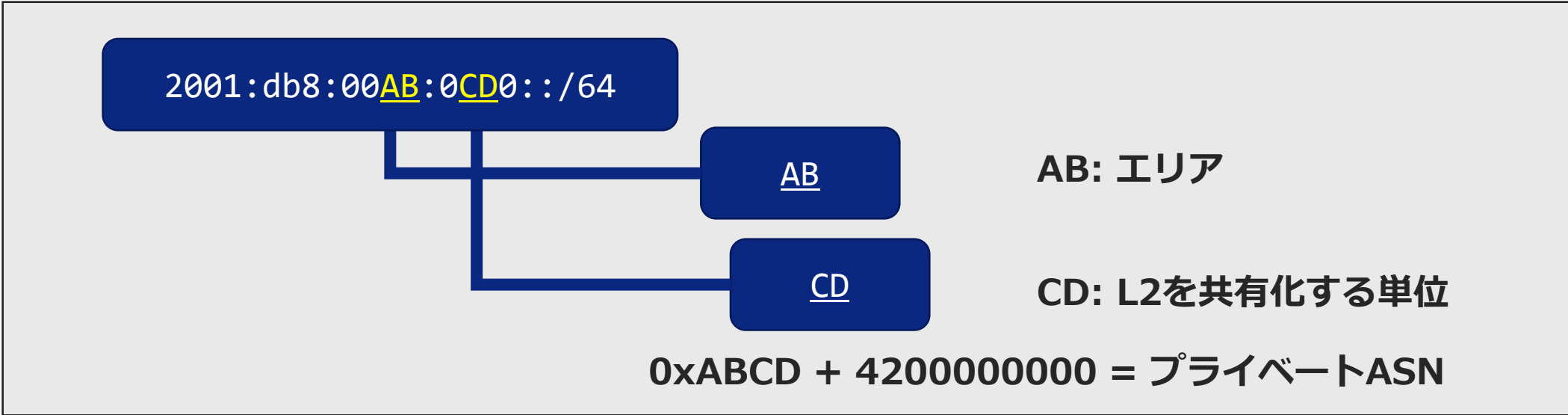
OVS-DPDKのオーバーレイNWはカーネルの持つBGPで得た経路を利用できない Bondingでの冗長化を利用

NW側は複数VRFのL2要件、まれに特殊なNW要件に対応する必要がある



# BGPで苦労点(RouterID/ASNどうするか?)

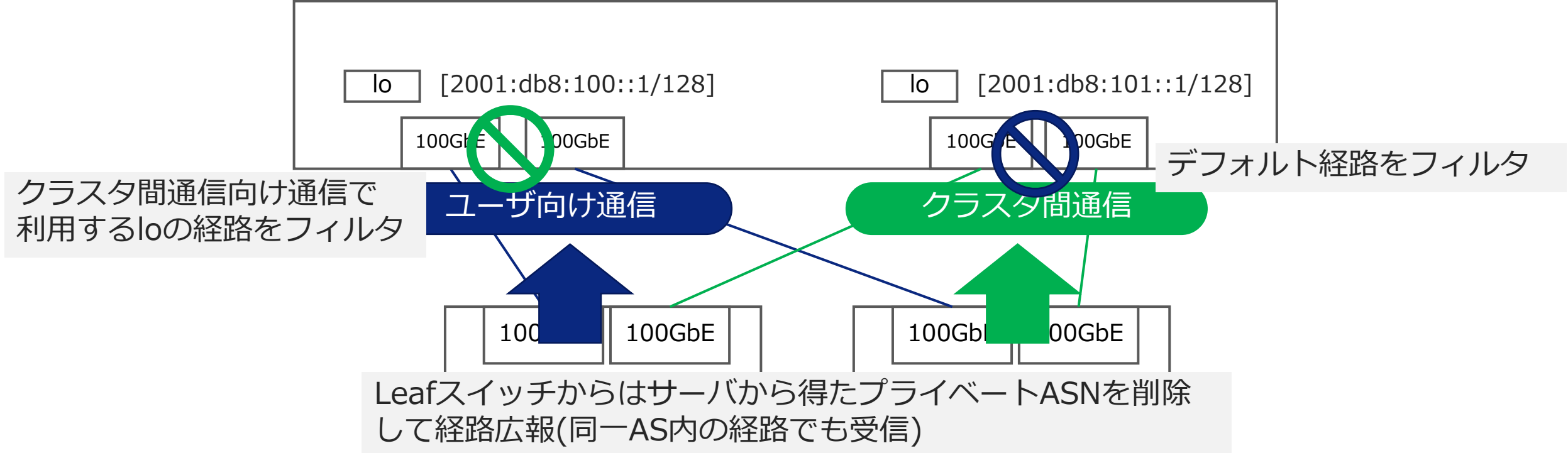
- RouterIDは隣接同士が同じじゃなければ大丈夫そうなので適当に設定
- ASNはかぶらないようにIPアドレスアサインルールから独自のアルゴリズムでIPv6アドレスのネットワークパートから生成



AB/CD に対して 適当な4Byte プライベートASNの値をオフセットして付与

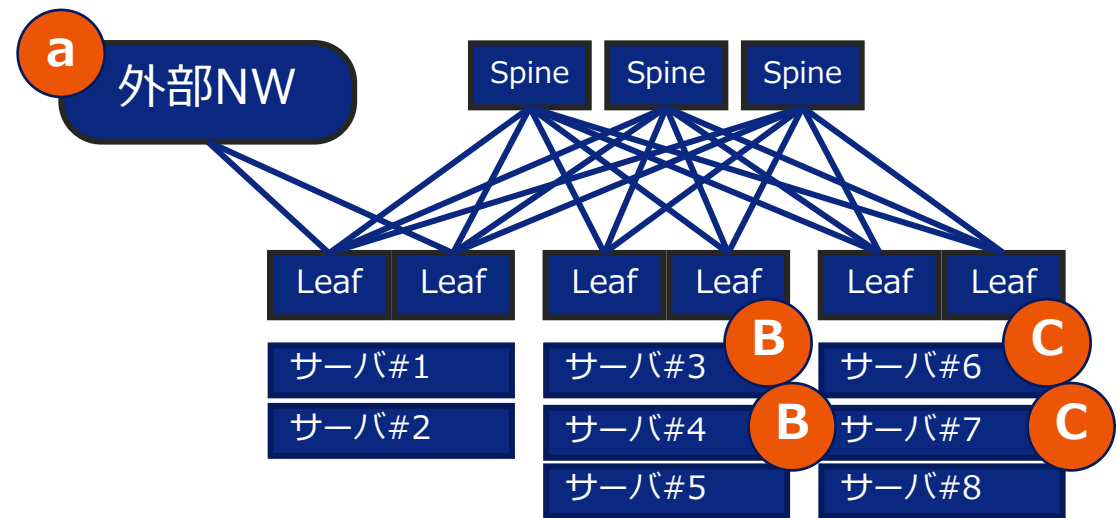
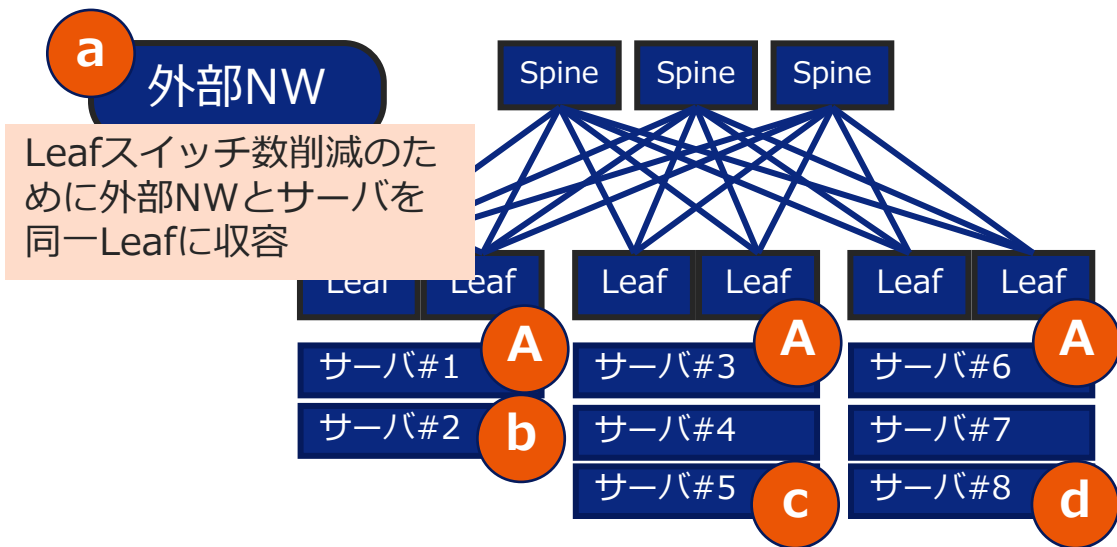
# BGPで苦労点(マルチNICでの経路制御)

- 今回のサーバでは100Gポートx4を採用
- Cephはクライアント向けトラフィックとクラスター間向けトラフィックが存在し、それぞれ独立して管理したいために4パスをうまく使い分けたい
  - 前述の通り同一Leafスイッチ間等ではASNが同一なので詳細経路が入らない



# BGPで苦労点(ANYCAST/マルチパスでの偏り)

- S3互換などのオブジェクトストレージにCephを利用、水平スケールのためにマルチパス+IP Anycastによる負荷分散を利用しようとしたところうまく負荷分散がされない事象
- Leaf内でのトラフィックは最もルーティングコストが少ないLeaf内のみで負荷分散
- NWの収容とDNSラウンドロビンを組み合わせて負荷分散を計る



- a → VIP:A向け通信は全てサーバ#1に
- b → VIP:A向け通信は全てサーバ#1に
- c → VIP:A向け通信は全てサーバ#3に
- d → VIP:A向け通信は全てサーバ#6に

- DNSラウンドロビンでVIP:B/VIP:Cを確率的に選択
- 外部NWの収容Leafにはオブジェクトストレージサービスを置かない

# 製品版OSSの苦勞 + Large scale編



# OSSってそもそも～

## ■ OSSを使うこと

- OSS = オープンソースソフトウェアライセンスで定義されているもの
- OSS : 自己責任のもと利用や改変の自由が許されている

## ■ 製品版OSS?

- OSSにサポートサービス付与したりして売ってるもの
  - Red Hatさんとか etc

	OSS-Upstream版	OSS製品版	プロプライエタリ製品
構成・使い方	組み合わせや使い方自由	<u>ある程度組合せや使い方が自由</u> ある程度パターンを制限	パターンを制限
サポート	コミュニティサポート、セルフサポート	ベンダーサポート	ベンダーサポート
修正反映	Upstreamで実施、フォークして自ら実施	<b>ベンダーで実施</b> (原則Upstreamファースト)	ベンダーで実施
お値段	-		 ※雰囲気での比較です



## ■ KDDIではビジネス的な判断でコアな機能については製品版OSSを選択

### 品質の要件

キャリアグレード、総務省報告対象

### NEPとの相互接続性

5G-Coreを提供するようなNEPベンダが動作をサポートしている

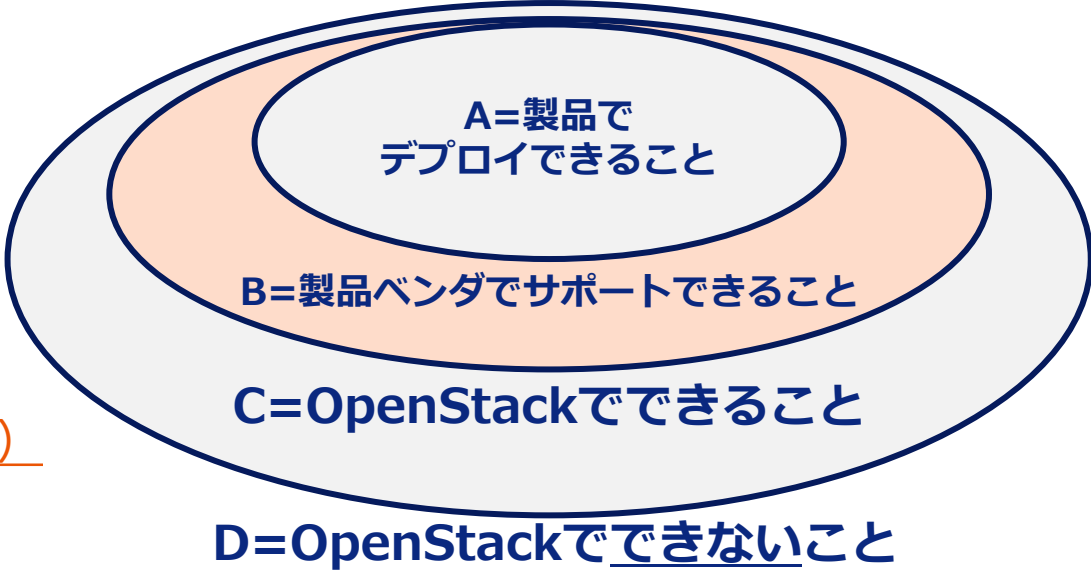
### 体制面の制限

OSS開発できるようなソフトウェアエンジニアを多く抱えていない

# 製品版OSSでの苦労点

- OpenStack(オープンソースソフトウェア)としてはできることでも、製品版のプロダクト(ダウンストリーム)ではサポートできることは限られる
- さらに製品版でサポートできるが、製品の持つ標準インストールツールからそれらが完全に適用できるわけではない

- ▶ またOSSでできないことは製品版ではできない(=D)
- ▶ 製品版はOSSに機能が実装されてから実装される
  - ▶ Upstream firstという考え方
- ▶ OSSのダウンストリームとアップストリームの間には、どうしてもタイムラグも発生する (テストや安定化のため)



製品版を使う上においても Upstreamをきちんと把握し、何ができてできないか、必要に応じて機能実装の依頼をしていく必要がある (そうしないとずっと機能が入らない)

# 製品版OSSでの苦労点（郷に入っては郷に従え①）

## ■ 安定しているML2/OVSではなくML2/OVNの選択

- OSSベンダ側が開発リソースをK8sともプロダクトを共通化できる**OVN**にシフト
- ML2/OVSの機能で十分満足していたが…**渋々ML2/OVN**を採用

OVN (Open Virtual Network) とは

- 複数ハイパーバイザ/k8sノード上のOVSにまたがった仮想ネットワークを作る仕組み
- OVS (Open vSwitch) のサブプロジェクトとして、2015年に始動
  - 最初のリリース: 27 Sep 2016 (OVS v2.6)
  - OpenStack Neutron Plugin (networking-ovn) の最初のリリース: 06 Oct 2016 (Newton)
  - OVS v2.11からリポジトリが分離 <https://github.com/ovn-org>
- オーバーレイネットワークを論理ネットワークとして抽象化

物理ネットワーク

論理ネットワーク

Red Hat



自社都合だけでは技術選択ができない+無理に選択するとVerupやサポートで禍根を残す

<https://speakerdeck.com/orimanabu/ovn-kubernetes-introduction-ja-2023-01-27?slide=5>



# 製品版OSSでの苦労点（郷に入っては郷に従え②）

## ■ OSS製品が対応する方法に常に従うこと

- 自己流のカスタマイズをすると予期せぬ設定ファイルの書き換えなどが起こる

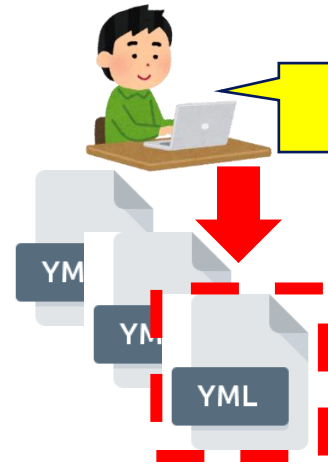
初期インストール



インストーラ

①インストーラが Config生成

インストーラも OSSソフトウェア



②手動で編集

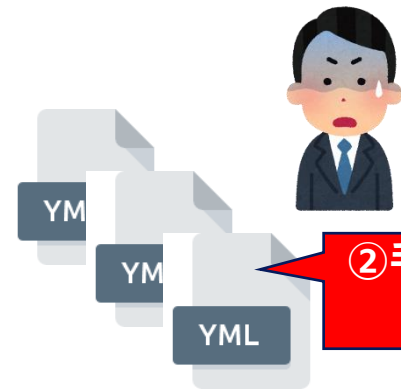
しばらくたって...

Upgradeするなど



インストーラ

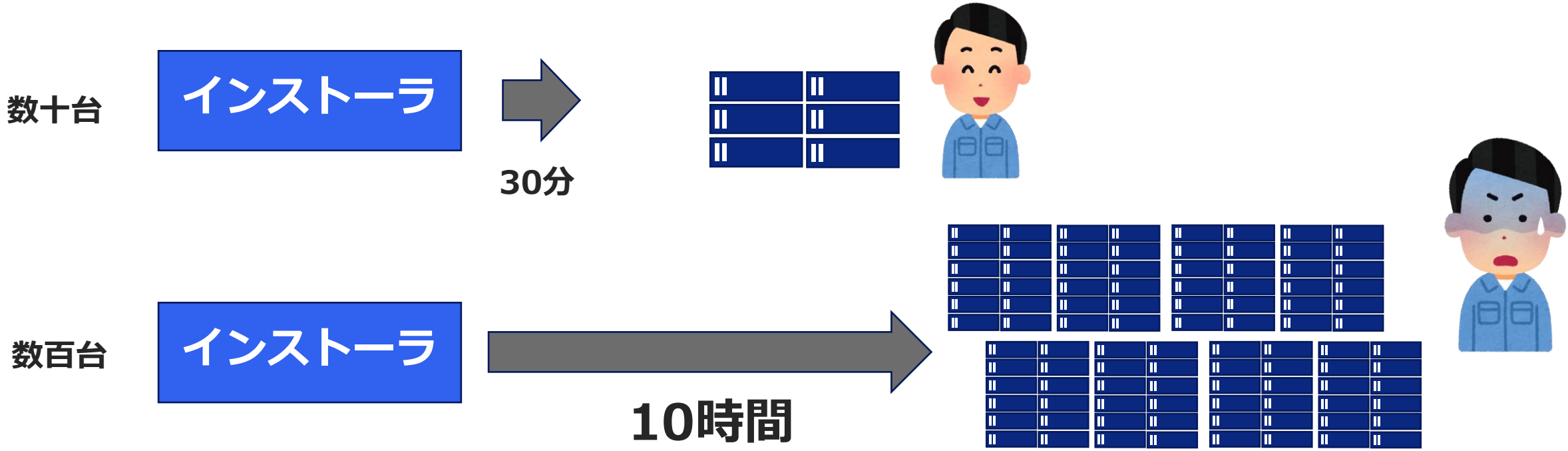
設定ファイルの変更等も インストーラを経由すべし



②手動で編集したファイルが先祖返りする

# 製品版OSSでの苦労点（郷に入っては郷に従え③）

- ラージスケールでのデプロイに時間がかかる
  - 標準的な構成でかつ台数が少ない場合はすぐ終わる処理も・・・
  - 諸々の細かなパラメータの取り込みのために複雑化すると・・・
- 複雑な自動化が組み込まれており独自に改変することがむづかしい





## 製品版OSSでの取り組み

- **積極的に自分たちの要望、使い方などの声を届ける**
  - どういう使い方をしているか
  - どういうことを目指しているのか
  - どういう機能を追加してほしいのか
- **足りないと思ったときにきちんとインプットしておく**
  - 要望が無いと実装されない（自分たちでUpstreamやらない限り）
- **技術的にナンセンスなことはできるだけ言わない**
  - 相手も技術者・そのソフトウェアのプロ
- **Upstreamに直接行く（まだまだそこまでできてませんが…）**
  - Upstreamに取り込まれた内容はDownstream入るので…

# ディスカッションタイム

## ■ IPv6 苦労

- IPアドレスの決め方とかどうしてますか（JANOG50でのLINE様の発表は非常に参考になりました）  
「LINE」をIPv4/IPv6 dual stack環境に変更した話  
[https://www.janog.gr.jp/meeting/janog50/line\\_dualstack/](https://www.janog.gr.jp/meeting/janog50/line_dualstack/)
- IPv6だとそれできないの？この実装されてないの？このOSとこのOSで動きが違う
  - ・（やっぱりRAでしかルーティング設定できないのとか、やっぱり不便じゃないですか…DHCPv6で入れたい…）
- IPv6を契機とした脱IP直打ち運用
- IPv6面はいくつも作りたくないから1つに寄せたいけどセキュリティや運用上の制約 etc
- (IPv6関係ないけど)L3 fabric上でのIP anycastは思っていた以上に難しかった（考えることが多かった）

## ■ NF向けのリッチなソリューションとNWシンプル化アプローチが相性悪い現実

- JANOG52でのソフトバンク様の発表は悩みが近そう
  - ・ キャリアバックボーンからIP Clos Fabricへの移行による5GSAネットワークの進化  
<https://www.janog.gr.jp/meeting/janog52/cf5g/>
- 私たちは極力オーバーレイに寄せようとしてたりします（JANOG51でのvRTの件など）

## ■ OSSをフォークするのではなく市販製品

- OSSをフォークするだけの体力がない我々はどう付き合っていくべきか

## ■ その他：人材育成 or そもそも内製化

「つなぐチカラ」を進化させ、  
誰もが思いを実現できる社会をつくる。

# KDDI VISION 2030

