T3 できる網設計
自社網とクラウド接続の実装例と課題

吉野純平

XFLAG

ケタハズレな冒険を。

クラウド接続のモチベーション

- **クラウドへのマイグレーションに**
- **オンプレへのマイグレーションに**
- **設備メンテのスワップ領域に**

スライド内の言葉定義

テナント
- 一つの事業の単位

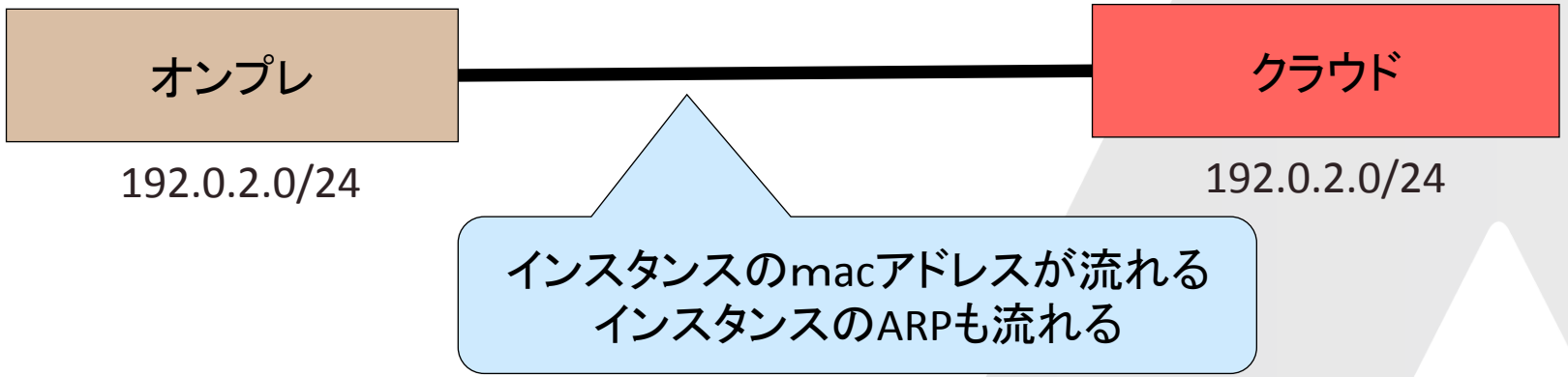
インスタンス
- オンプレのサーバ
- クラウドのインスタンス

クラウドとどう繋がるか(レイヤ2)

クラウドとオンプレで同じセグメントを共有

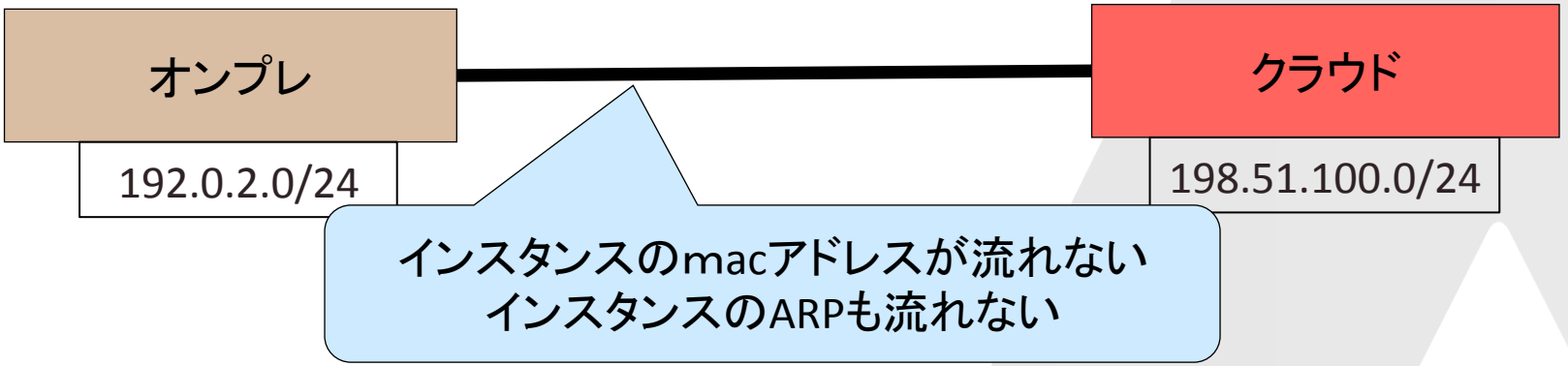
現時点では多分無理

- macとarpのテーブル管理が厳しい
- evpnが解決するかも？
- 運用上厳しいはず



クラウドとどう繋がるか(レイヤ3)

クラウドとオンプレで異なるセグメント
よくある接続方法
IPsecと物理接続が多い
static routeか、bgpでのルーティングが多い



クラウドとどう繋がるか(レイヤ7)

インターネットだけ繋がればよい
APIやwebでアクセスできればよい
DB等のアクセスをしない・接続制限で良いとする
ある意味一番よい

クラウド間接続の実装例

今回はレイヤ3での接続の話

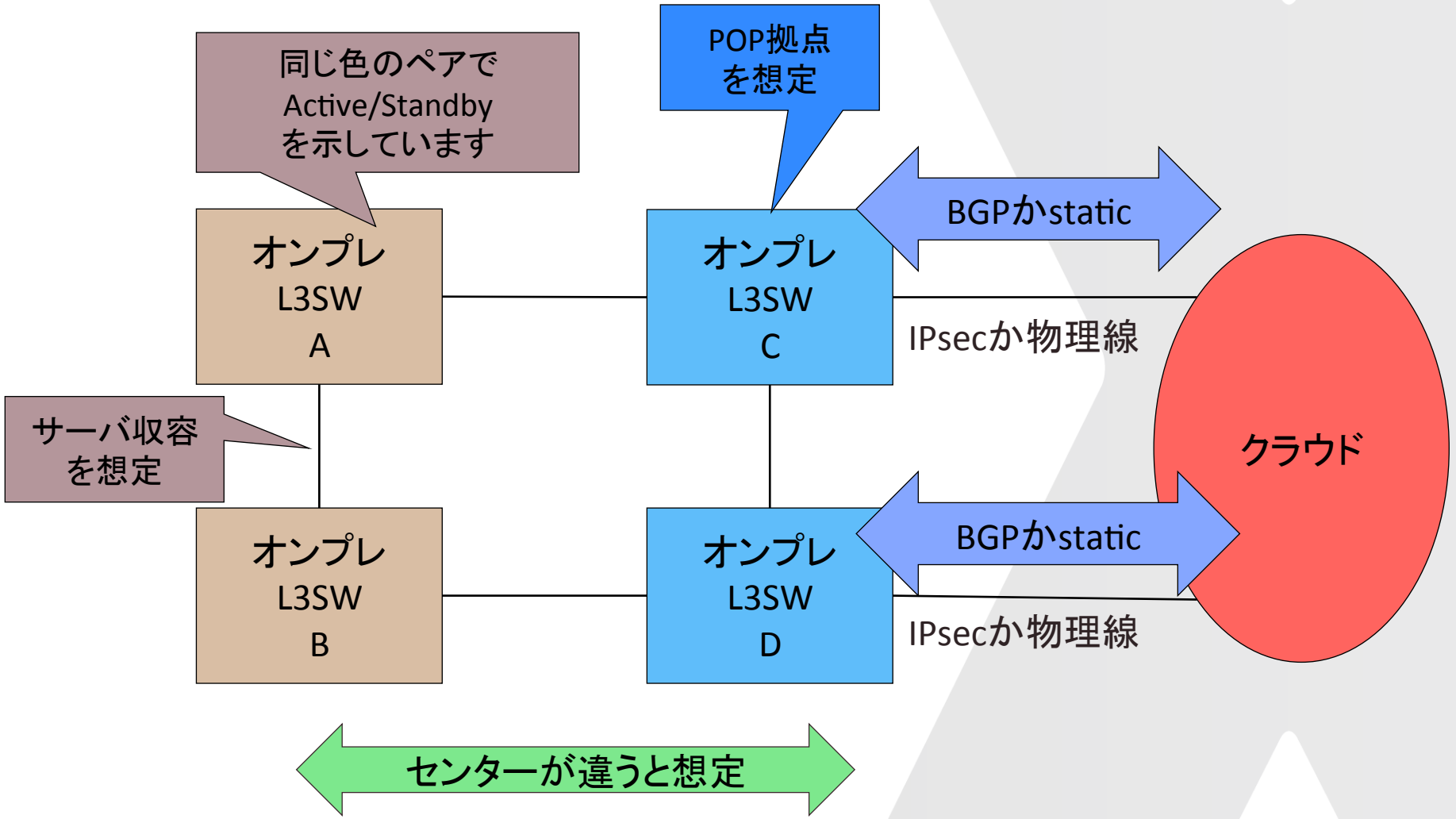
ミクシィ社 第一世代

- 1つのルーティングテーブル
- BGP+OSPF

ミクシィ社 第二世代

- 複数のルーティングテーブル
- VRF+BGP+OPSF+MPLS(LDP)

クラウド間を考える際の概念図



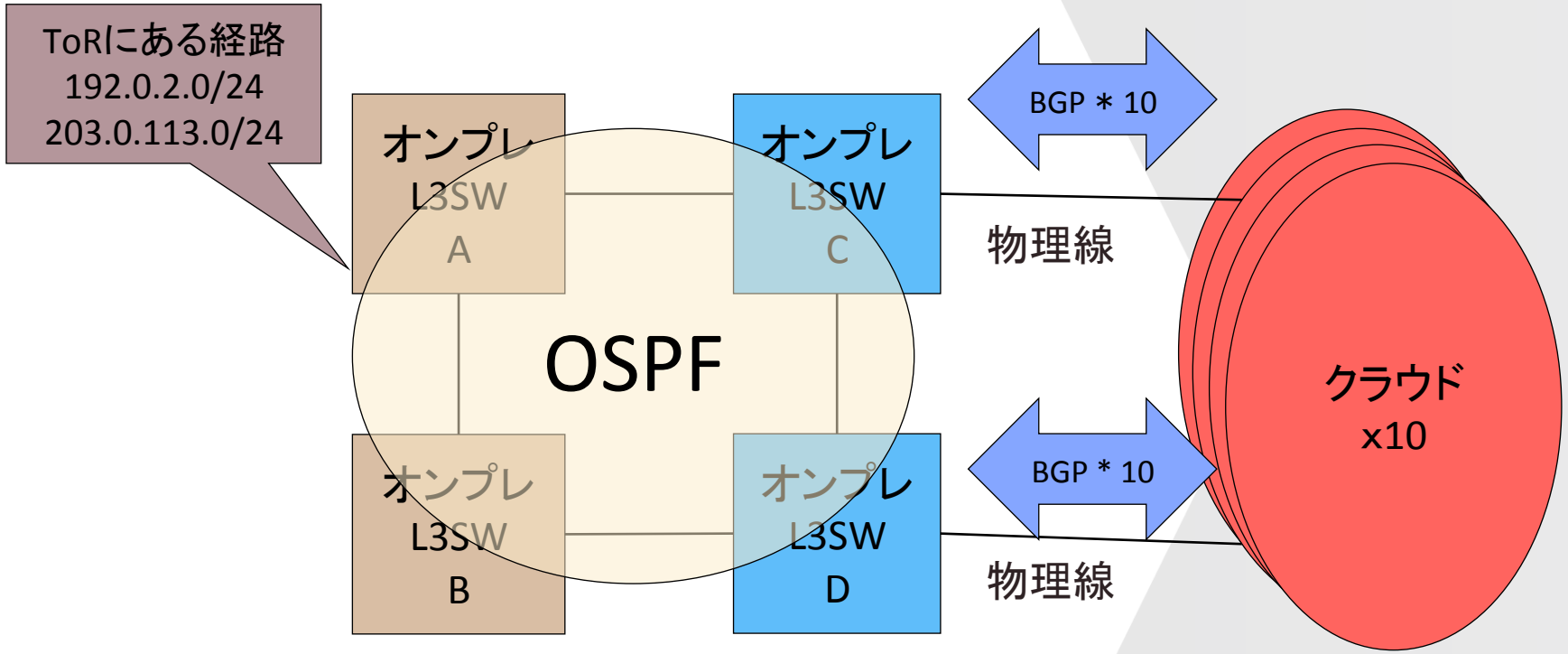
第一世代の例の前提

over 10 テナント

接続制限が欲しい

- 相互に通信したい環境
- 相互に通信したくない環境

構成概要

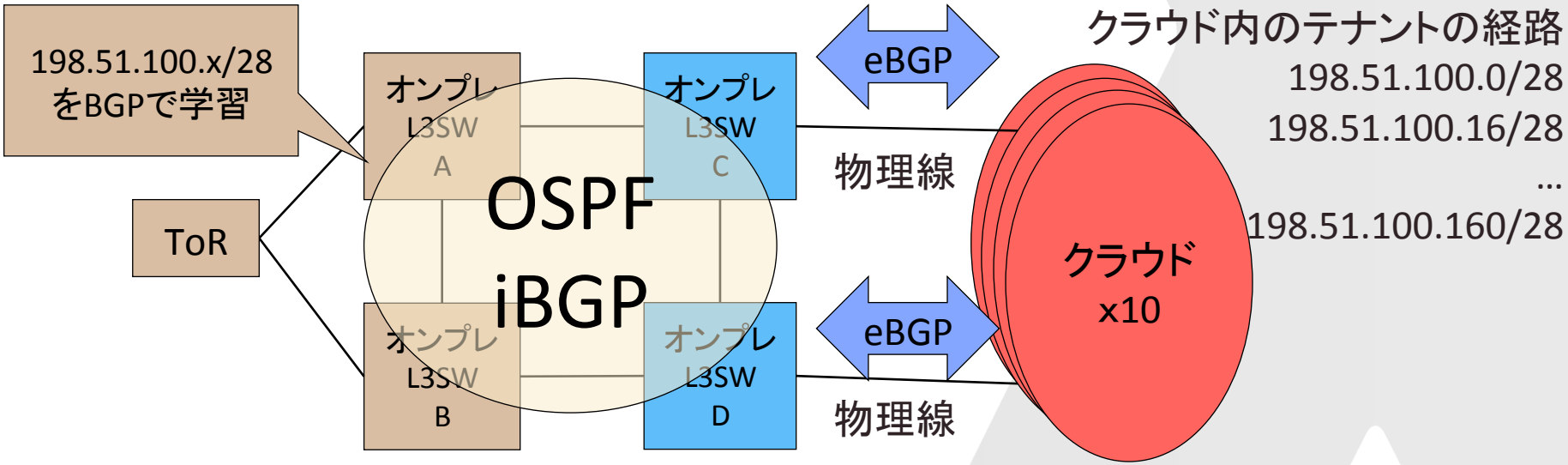


クラウド内のテナントの経路
198.51.100.0/28
198.51.100.16/28
…
198.51.100.160/28

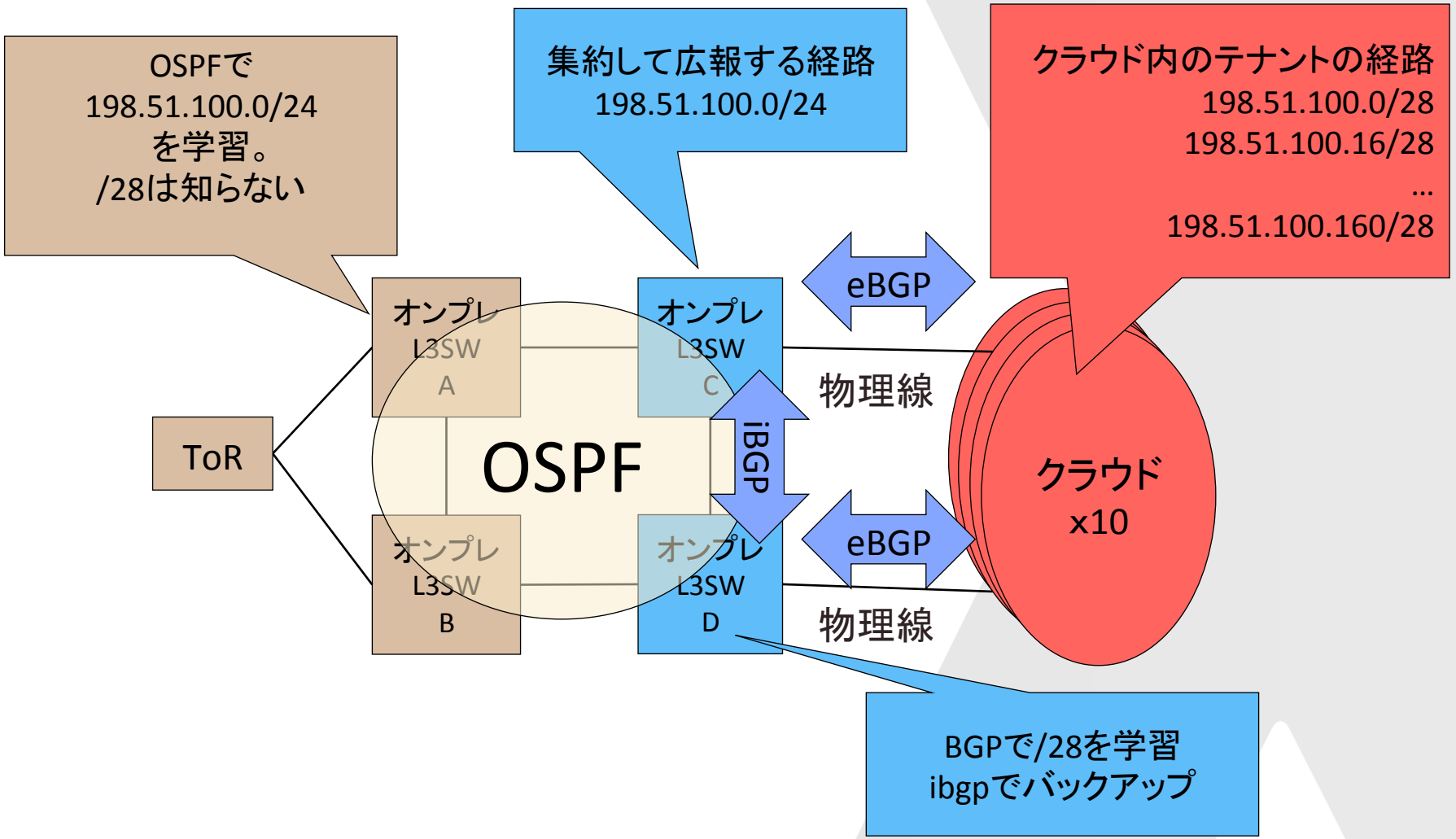
クラウドへの到達性の確保方法の論点

内部網を全てBGPを動かせるか？

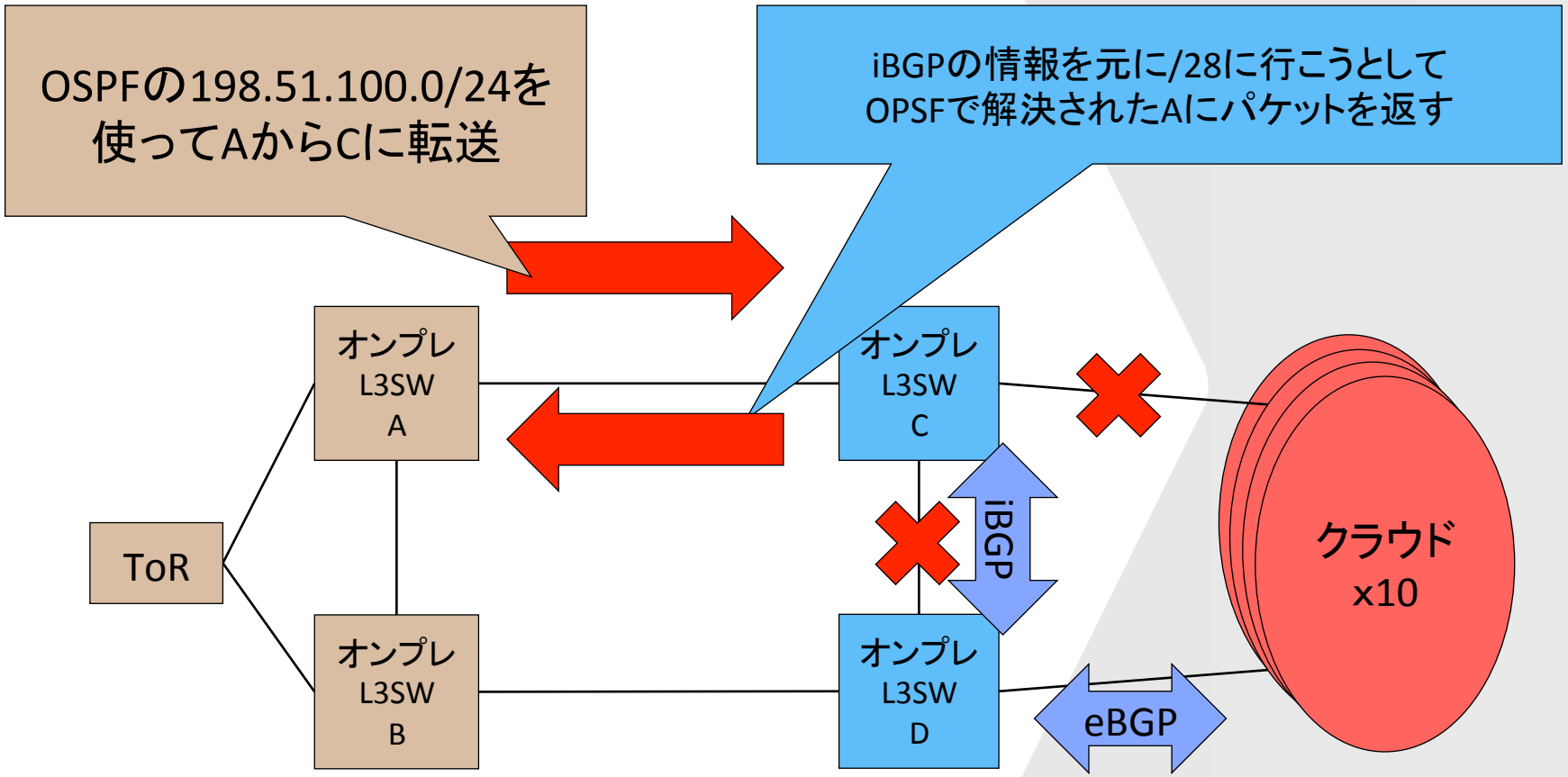
- ライセンスや導入期間で厳しい場合もある
- 障害耐性等で考えると良いとおもう



クラウド接続部分にだけBGPを使う



デメリット：特定の故障パターンがダメ

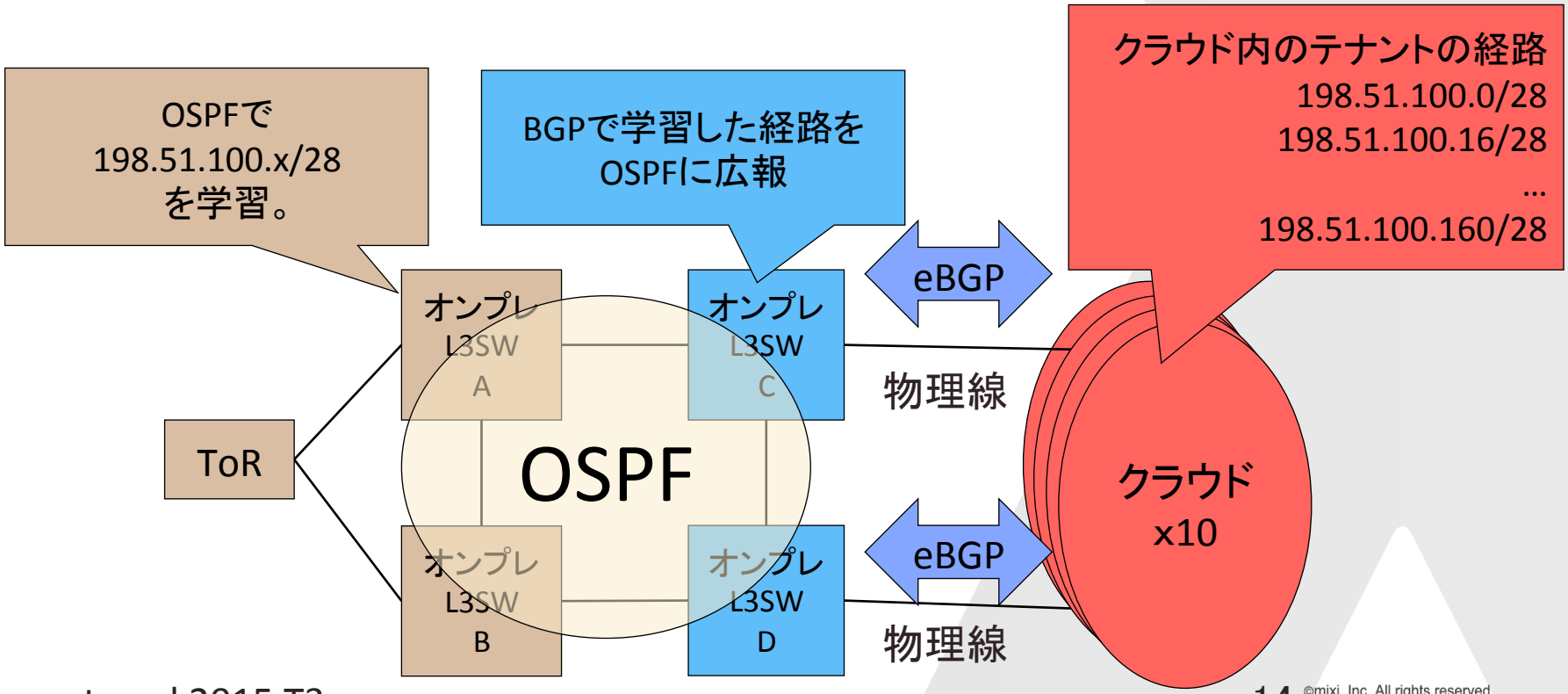


青のスイッチ部分でこのような故障パターンがないように設計する
例えば、バーチャルシャーシ・スタック系技術で守る

別のパターン：再広報

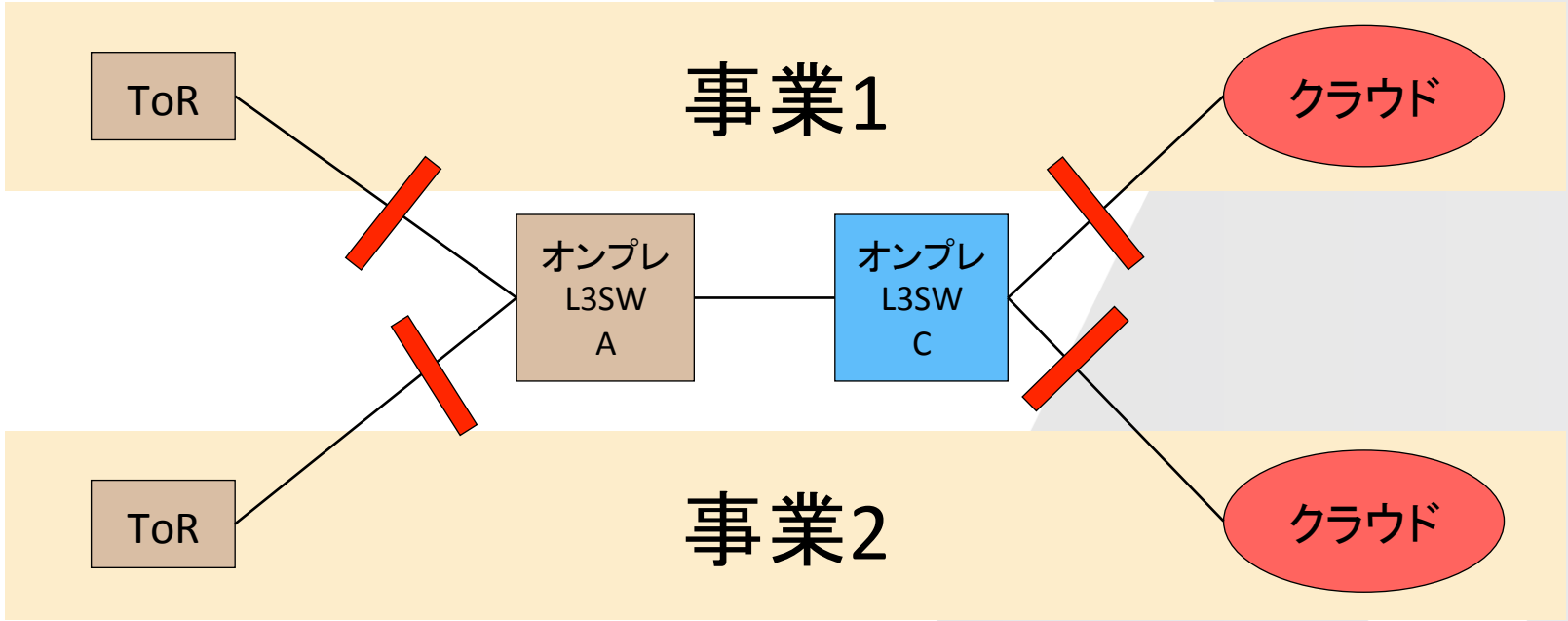
設定例

```
router ospf x
  redistribute bgp 65000 subnets
```



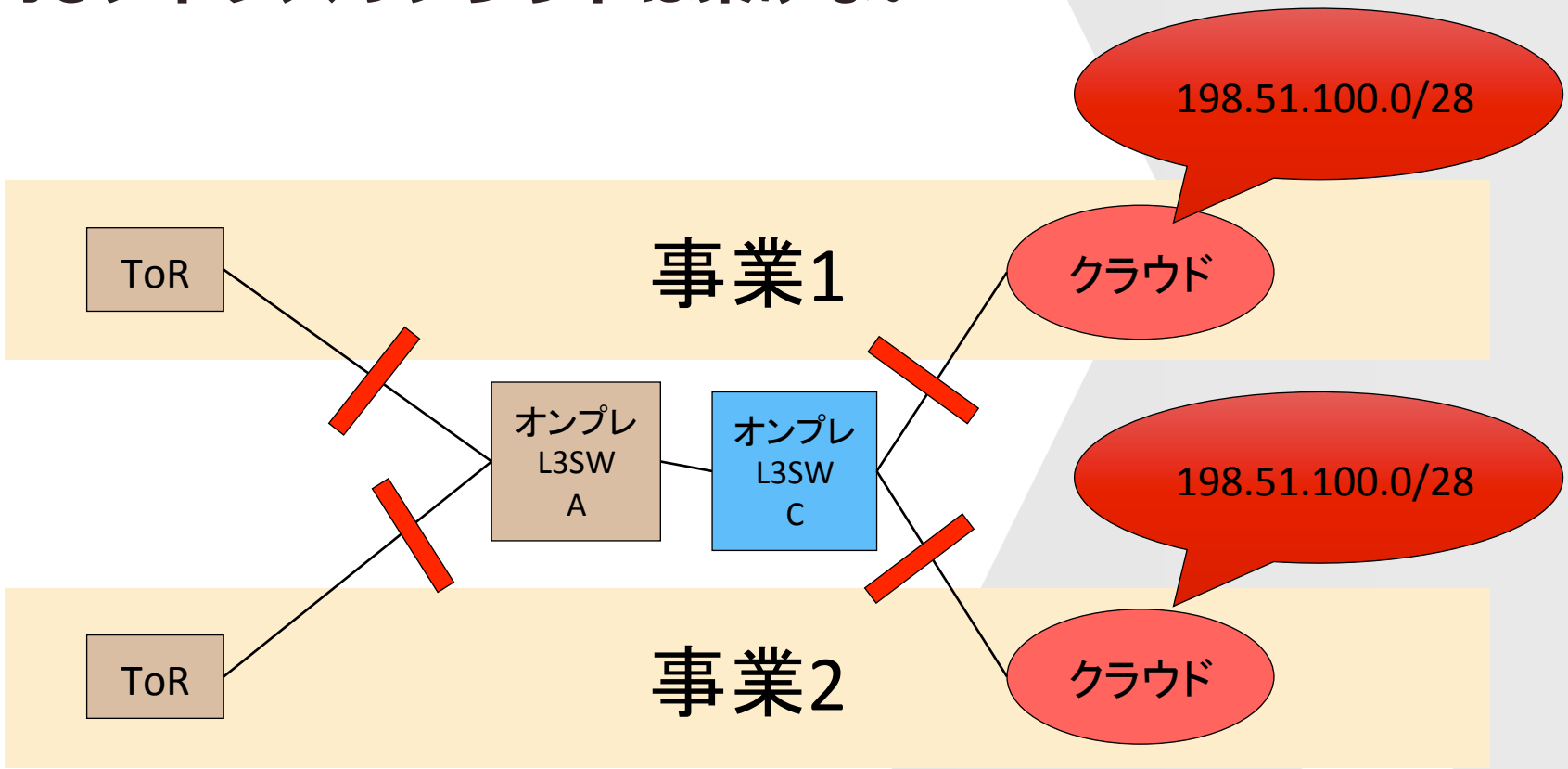
第一世代の問題点1

接続先を制限しようとすると難解なACL設定が発生する



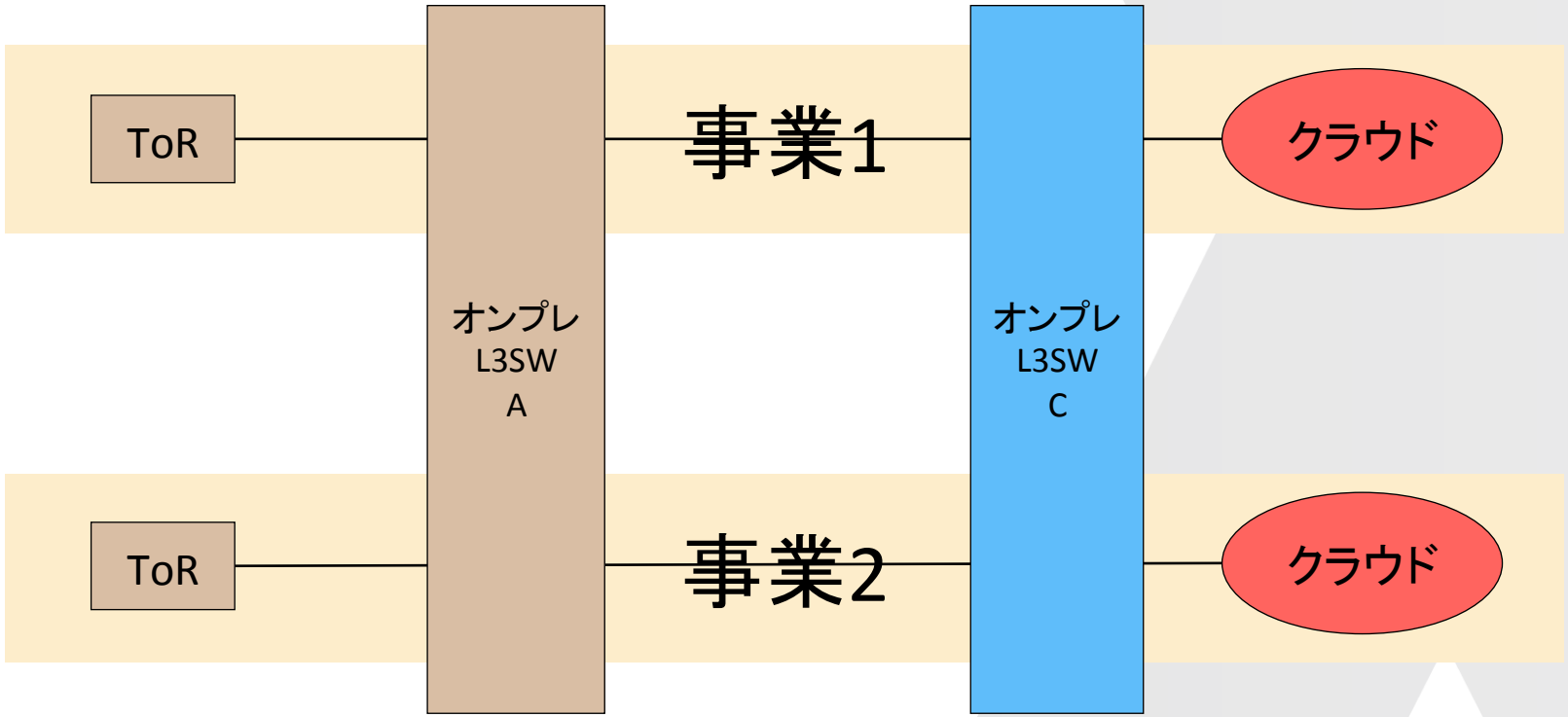
第一世代の問題点2

同じアドレスのクラウドは繋げない



第二世代

必要な範囲でルーティングテーブルを分ける



実装手段

vrf-lite,virtual-router等の機能使う

- シンプルな分離

vrfを使う

- mp-BGPで経路情報を送受信可能
- ルーティングテーブル数の増加と動くプロセスの数が比例して増えない

vrfで実装した理由

切り替わり時間短縮のため

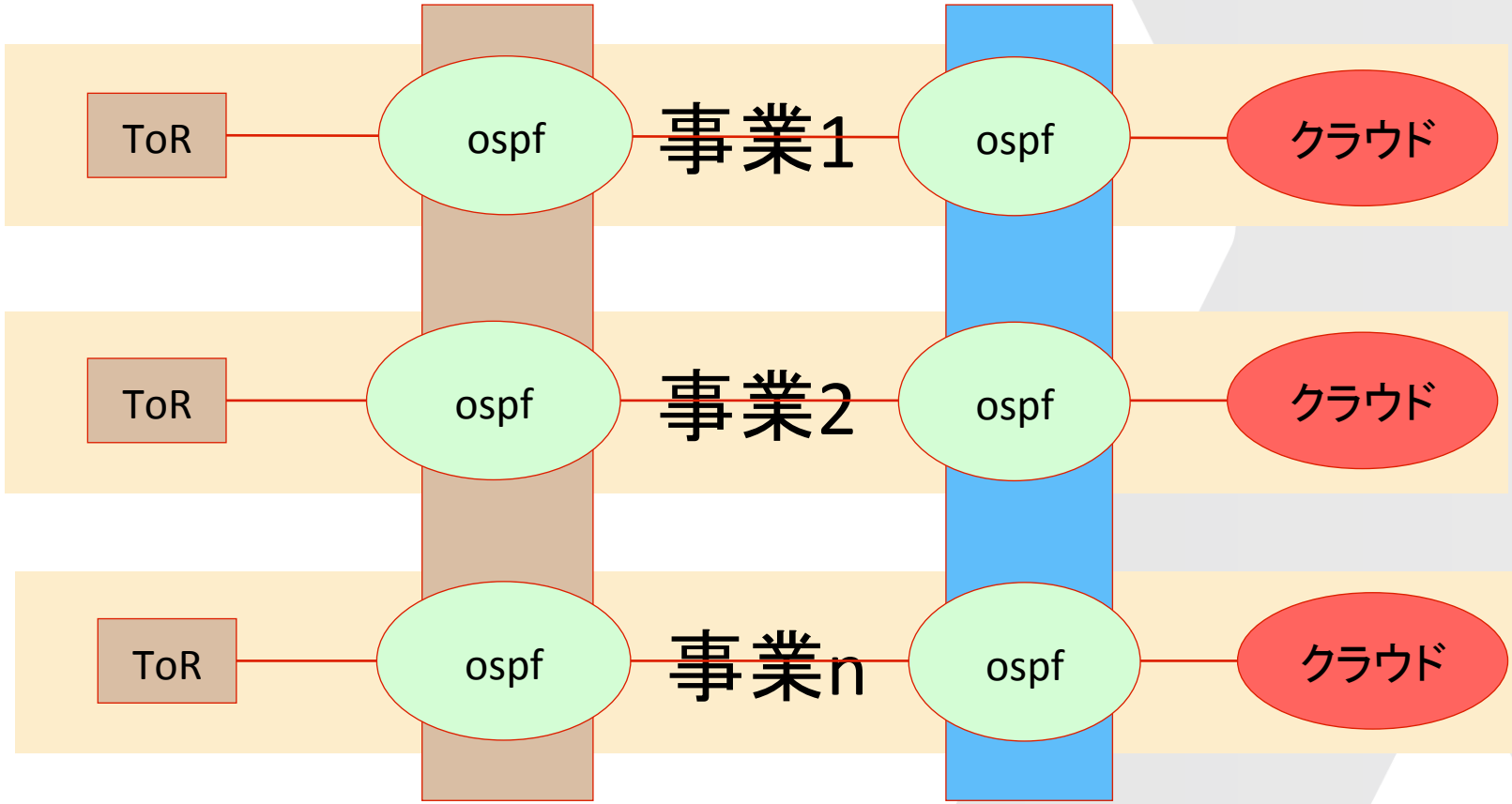
- プロセス数を抑えたい
- bfd等を考えるとneighbor数も抑えたい

メンテ時の迂回作業量を減らしたい

テナントを増やす作業を楽したい

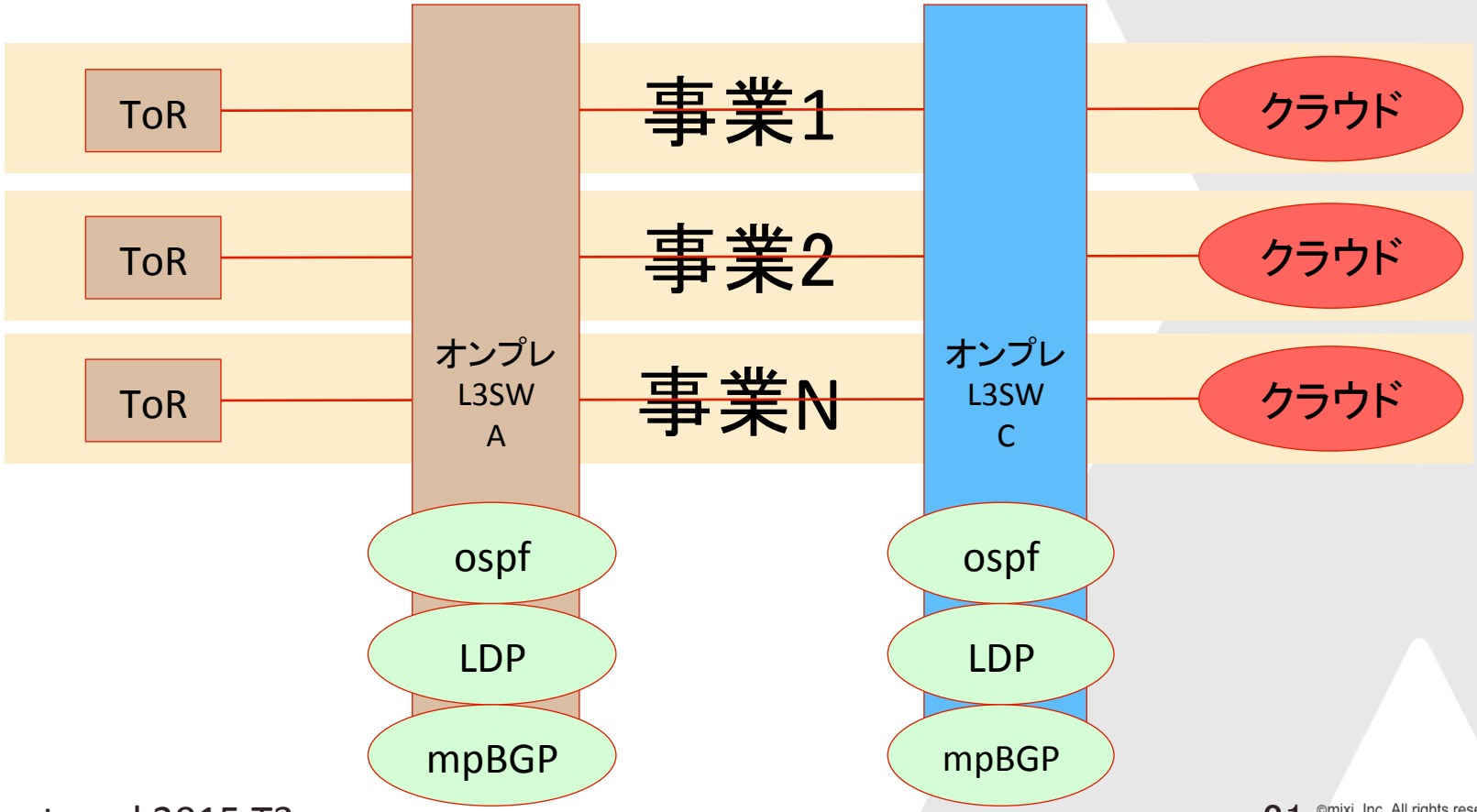
vrfを使わない場合

プロセスの数が比例して増える



vrfを使う場合

案件の数にプロセス数が連動しない



どちらが良いかは前提次第

vrf-lite/virtual router

- 少ない案件数とNW規模
- メンテナンスが多いと辛い

vrf

- 案件数が多い時に効果が大きい
- 新しい技術チャレンジは必要
- 経路迂回も1つのトポロジだけ触ればよい

MPLSの活用の実装例

今回の紹介の前提

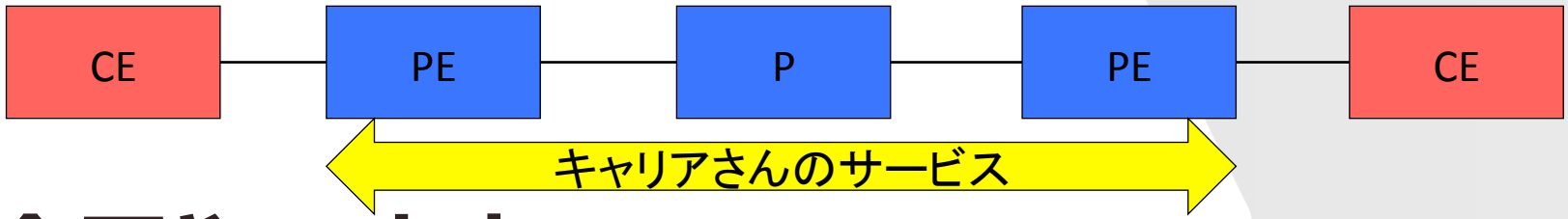
ラベルの配布はLDPを使う

- TrafficEngineering系の技術を使わない
- FastReRoute系の技術を使わない
- シンプルな実装を目指す

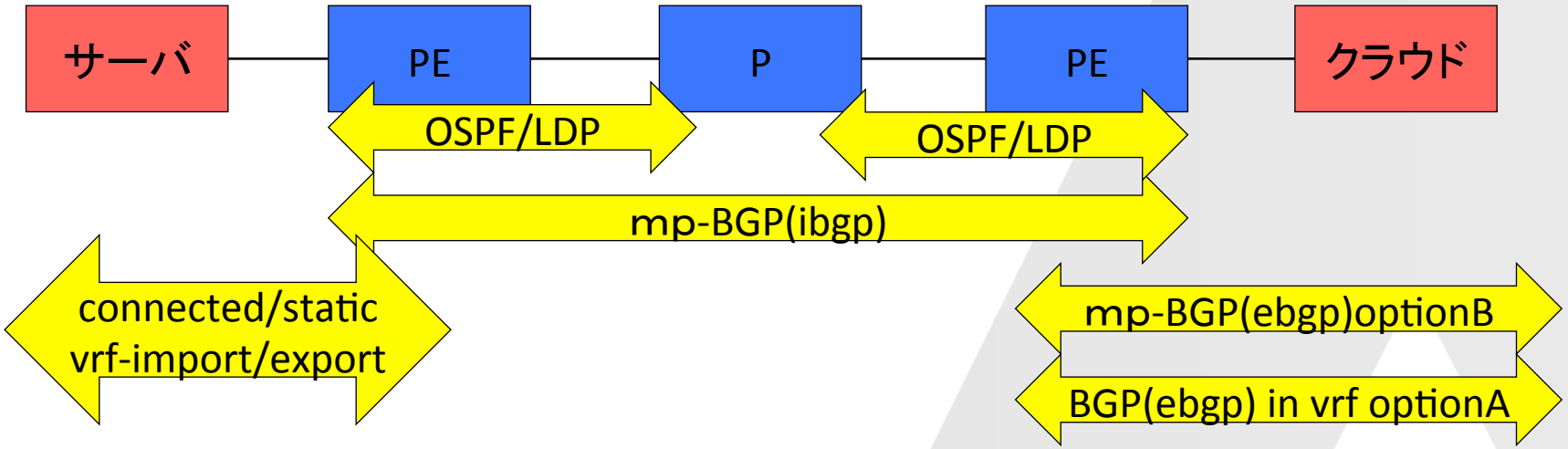
VPN経路はMP-BGP

作るもののイメージ

よく見るL3VPNのモデル



今回やる内容



作り方

OSPFでIGPを作る

LDPでラベルを配布する

- OSPFのメトリックと同期する
- 転送ラベルを解決できるようになる

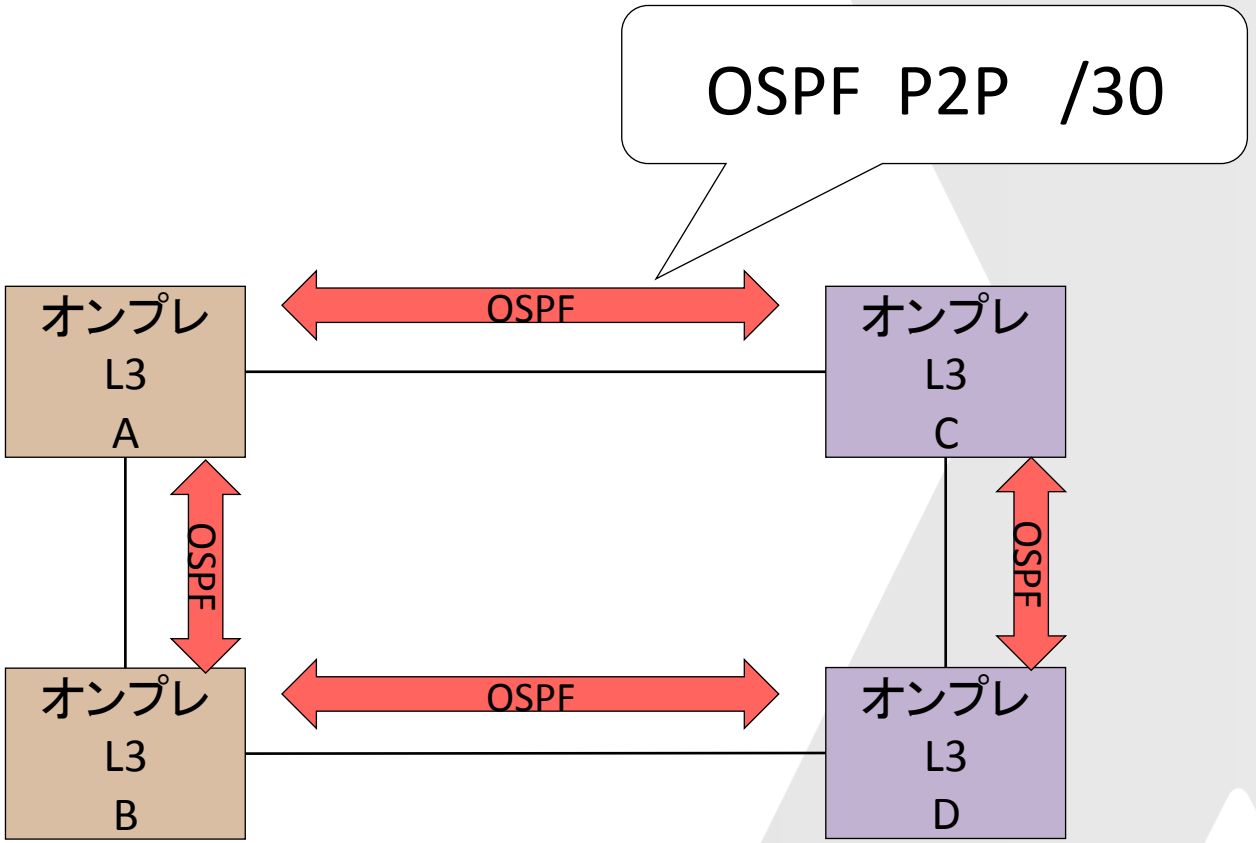
mp-BGPでVPNラベルを配布する

- 転送ラベルの内側に付き、どのテナントか識別

作り方1:IGP

- **IGPが動いているネットワークを作る**
  - 日本で多いのはOSPFらしい
- **neighborは/30でp2pモードに**
- **メーカーごとの差を埋めた方がベター**
  - ospfの各種タイマー
    - CをJに合わせるなら timers throttle spf 200 5000 5000
  - router idになっているloopbackのコスト
    - JをCに合わせるなら metric 1
- **L2スイッチ等を挟む場合はbfdの利用を検討**

図で描くなら



cisco ios-xrなら

```
interface hoge
        mtu 9000
        ipv4 mtu 1600
        ipv4 address x.x.x.1 255.255.255.252

router ospf 1
        router-id x.x.x.a
        timers throttle spf 200 5000 5000
        area 0
        interface hoge
        cost 10
        network point-to-point
```

後ほど紹介する注意点1

タイマー設定
Juniperに合わせる

p2pで設定
後ほどLDPを設定する

Juniperなら

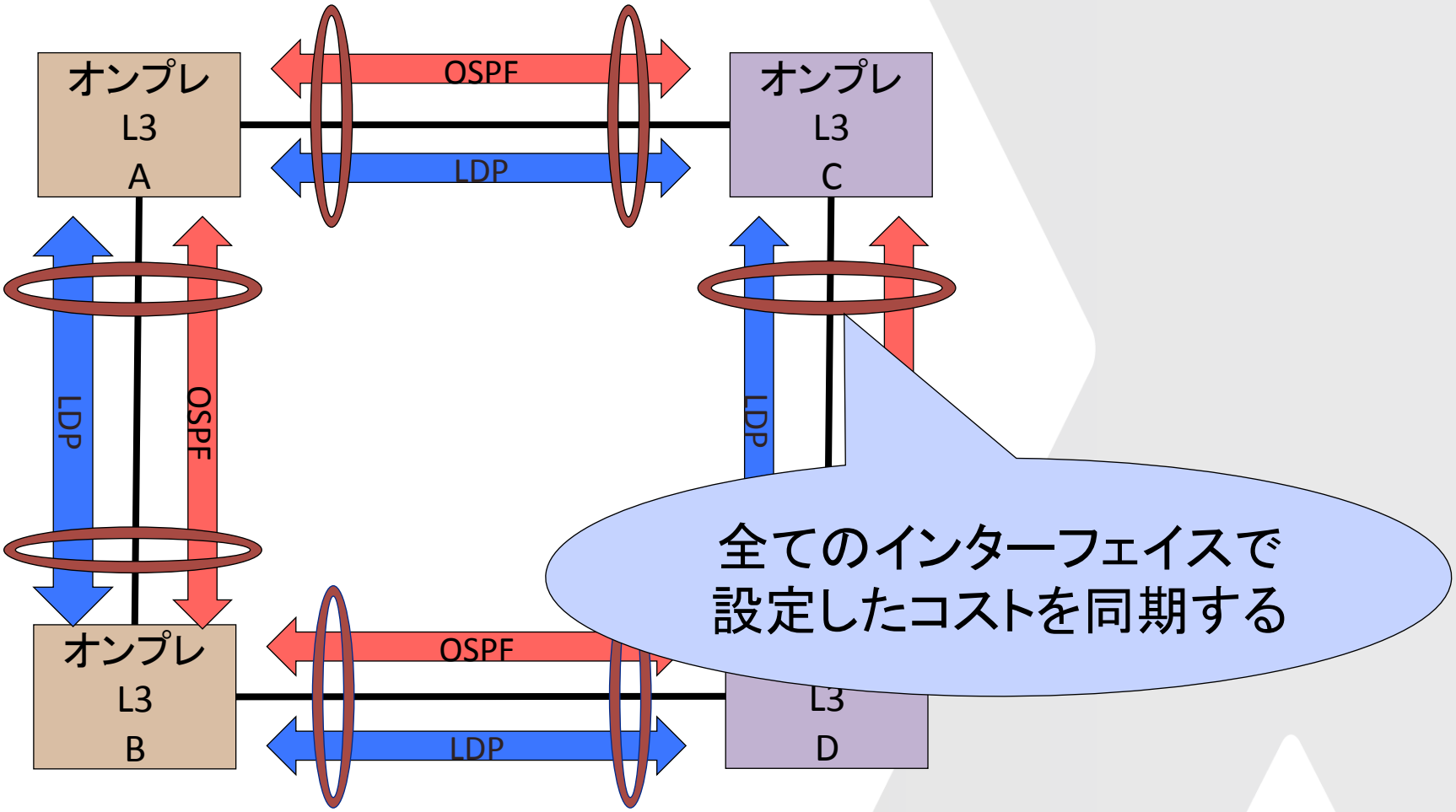
```
interfaces hoge {
    mtu 9000;
    unit 0 {
        family {
            inet {
                mtu 1600;
                address x.x.x.1/30;
            }
            mpls;
        }
    }
}
```

```
protocols {
    ospf {
        area 0.0.0.0 {
            interface hoge.0 {
                interface-type p2p;
                metric 10;
            }
        }
    }
}
```

作り方2:LDP

- **OSPFが動くインターフェイスで動かす**
- **コストもOSPFとsyncする**
- **ラベルの交換中に転送されることを防ぐためにdelayを入れる**
- **ping等でLSPを確認する場合にはping mpls等を使う**

LDPの図



Cisco ios-xrなら

```
router ospf 1
    mpls ldp sync
    mpls ldp auto-config
mpls oam
mpls ldp
    igp sync delay on-proc-restart 60
    igp sync delay on-session-up 60
    router-id x.x.x.a
    address-family ipv4
        label
            local
                advertise
                    disable
                    for label-out
     interface hoge
```

auto-config設定で
ospfのインターフェイスに自動設定

網内のRouter idだけが広報されるようにフィルタリング
Juniperのデフォルトに寄せた

junosなら

```
protocols {
    ospf {
        area 0.0.0.0 {
            interface hoge.0 {
                ldp-synchronization {
                    hold-time 60;
                }
            }
        }
    }
}
```

```
Protocols {
  ldp {
        track-igp-metric;
        interface hoge.0;
        igp-synchronization {
            holddown-interval 60;
        }
  }
}
```

作り方3：mp-bgpの実装例 iBGP

bgpでの経路交換にaddress familyを追加

- cisco なら vpnv4 unicast
  - neighbor-group iBGP
    - address-family vpnv4 unicast
  - mplsで転送するインターフェイス指定も必要
- juniperなら inet-vpn unicast
  - set protocols bgp group iBGP family inet-vpn unicast
  - 条件次第で全iBGPがリセットされるので注意（JANOG35.5参照）

ciscoとjuniperの違い

juniper

- VPNラベルはrouting-instances単位
- VPNラベルでテーブルを識別したあとに再度lookup

Cisco

- VPNラベルの粒度はいろいろ設定可能
  - CE単位等でまとめることも可能
- CEFを活用して1回にしている？（未確認）

mp-bgpの実装例 eBGP

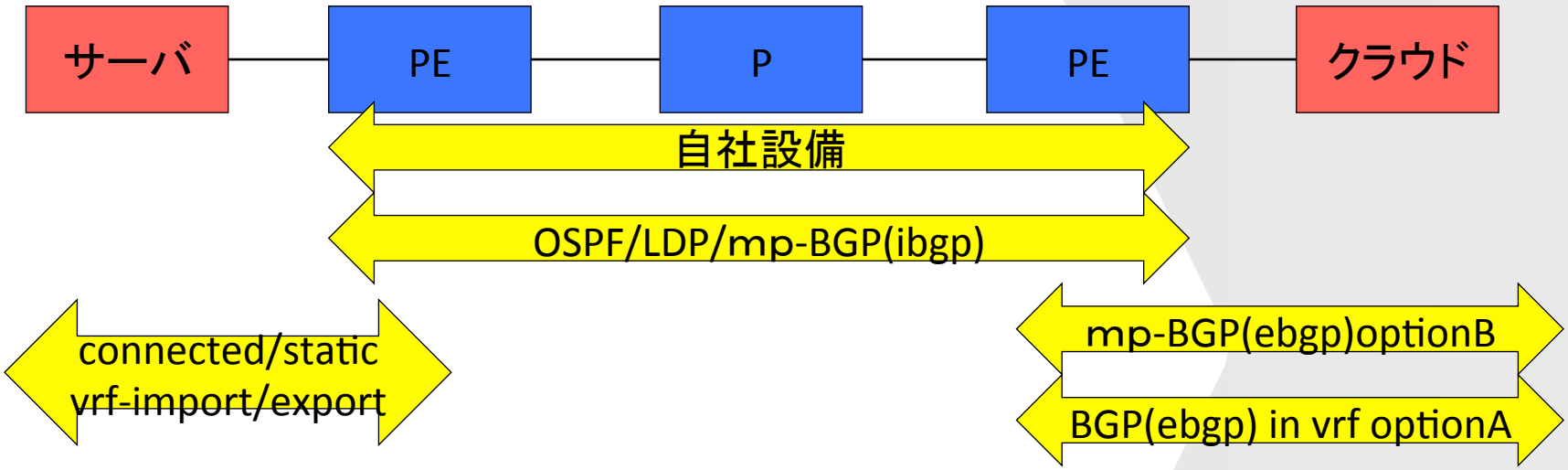
Option A

- 通常のeBGPピアと同様

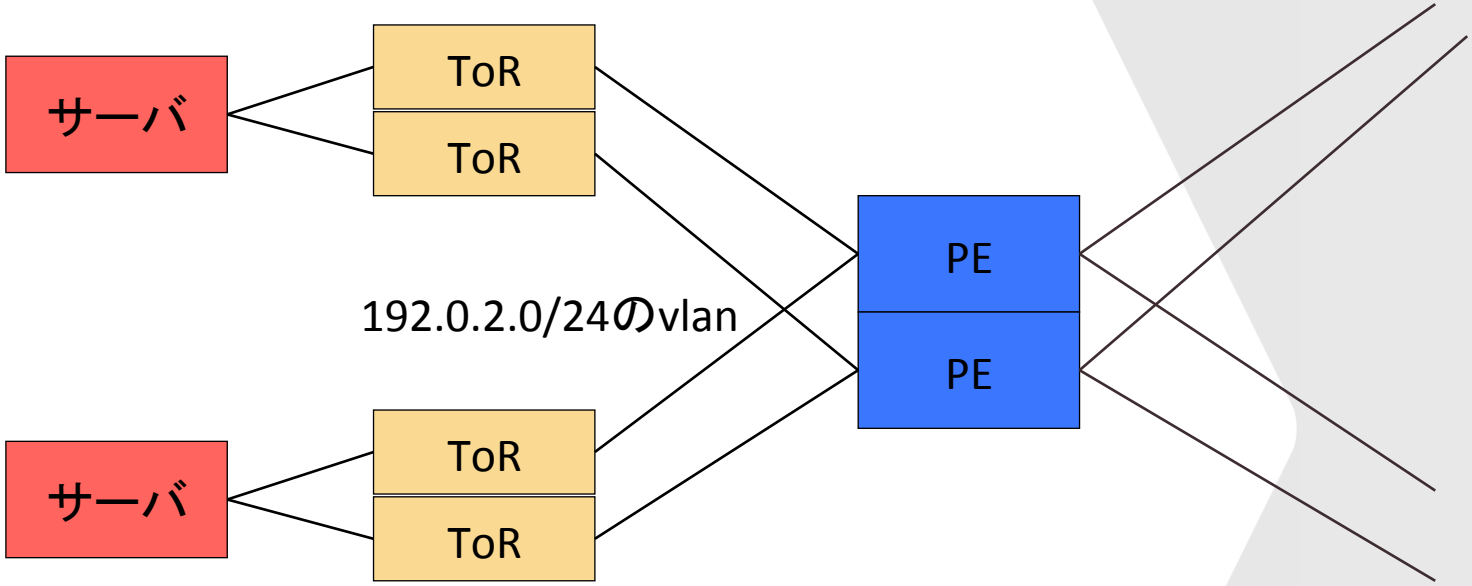
Option B

- JANOG35.5 でJuniperでの例を紹介
  - **「インタークラウドをアプリケーションとしたオープンな閉域網接続の実現に向けて」の井上さんの資料**
- Cisco IOS-XRの場合は以下のworkaroundが必要
  - router static
    - address-family ipv4　unicast
      - x.x.x.x/32 インターフェイス名
  - bgp nexthopの/32をインターフェイス指定でstaticを書く

作るもののイメージ（再掲）



PE部分を詳しく



PEでconnectedやstatic routeを他のPEに広報

vrfの定義 juniper

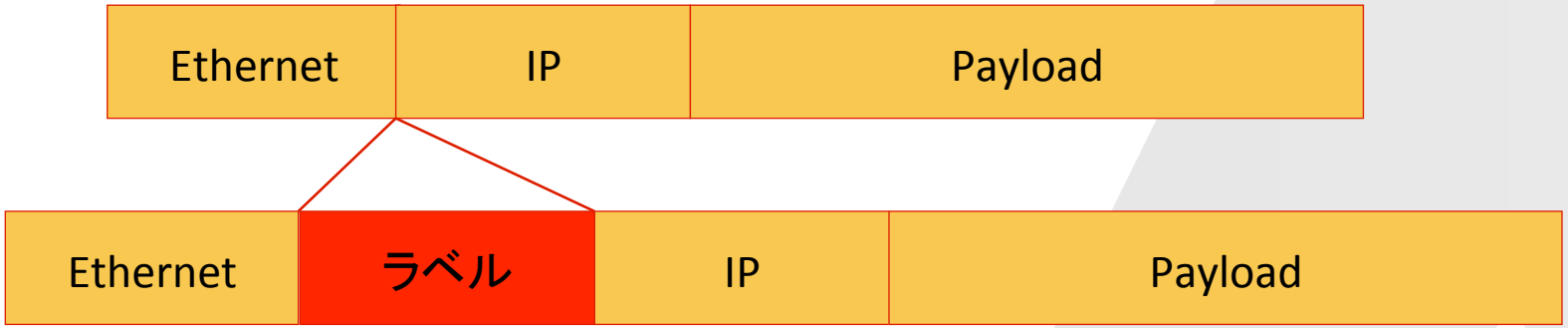
```
routing-instances piyo {
    description 名前;
    instance-type vrf;
    interface irb.xxx;
    vrf-import インポートポリシー;
    vrf-export エクスポートポリシー;
    vrf-table-label;
}
```

vrfの定義 cisco

```
vrf piyo
 address-family ipv4 unicast
  import route-target
   yyyyy:xx
  !
  export route-target
   yyyyy:xx
  !
interface hoge
    vrf piyo
    ipv4 address x.x.x.x 255.255.255.0
```

注意点1

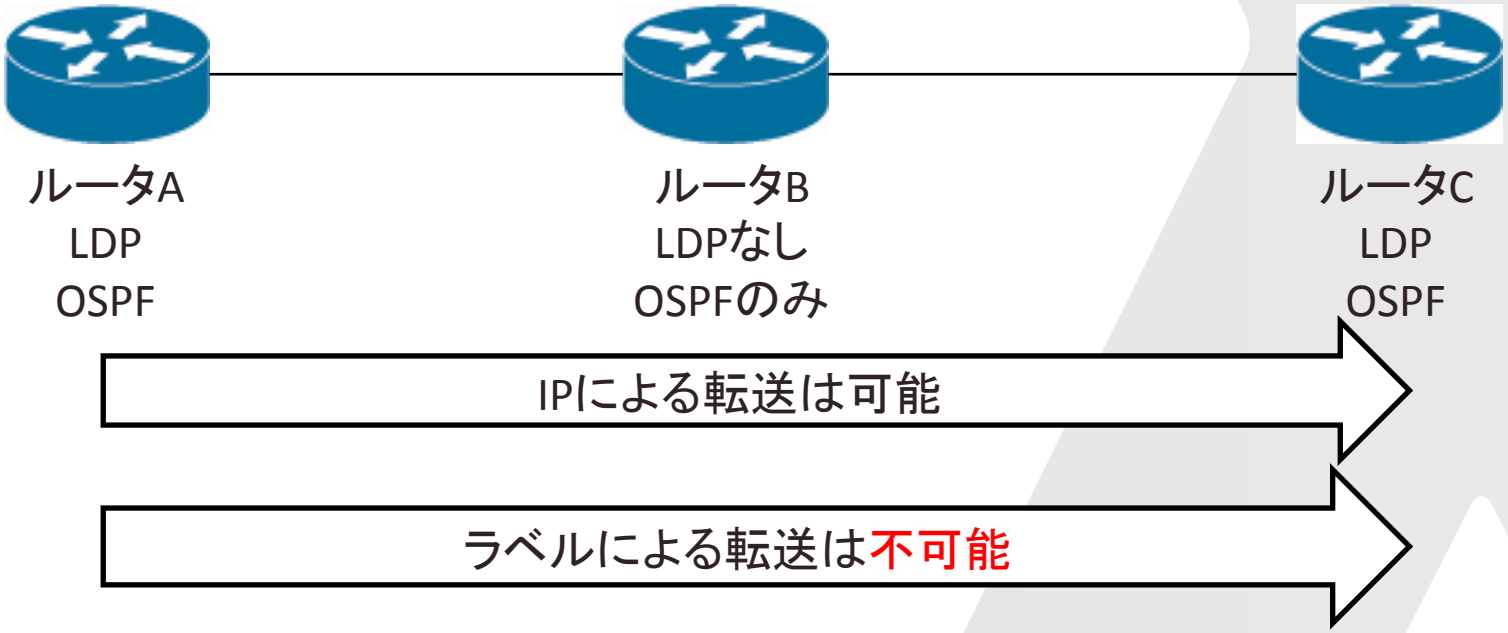
- **ラベル分のサイズ増を考慮**
- **MTUに気をつけましょう**
  - スイッチのMTU設定
  - 専用線のMTU



注意点2

LSPが切れないようにしましょう

- ラベルの転送が必要ないものは問題ない
- L3VPNは転送できない



注意点3

ciscoで下の設定をしてもLDPパケットが出る

- mpls ldpでauto config
- ospfでinterfaceにpassive指定

ospfのpassiveでLDPも止まると勘違いした

- ピア用アドレス等を網に広報する際に注意
- ldp側でinterface指定でdisableを

今のうちにやりたいこと

専用線のEthernet MTUサイズ確認　目指せ9000

IP MTU 1600以上

- IP上にトンネル入れてもフラグメントしないこと

MPLSの段階的導入検討

- 派手な機能を使わなければほとんど変わらない

まとめ

実装例を紹介しました

- 1つのルーティングテーブル
- 複数のルーティングテーブル
  - mplsでの実装

今回の発表はあくまでも実装例です

- 正しくない表現等が含まれる可能性があります
- 十分に検証等いただけますようお願いします


XFLAG