

Next Data Center Networking with SRv6 Control plane

Hirofumi Ichihara
LINE corporation

About Me

- 市原裕史 (Hirofumi Ichihara)
- LINE Corporation
 - ネットワーク開発チーム
- Network Software Developer
 - SDN/NFV
 - OpenStack Neutron
 - Docker
 - Kubernetes

SRv6 コントロールプレーン

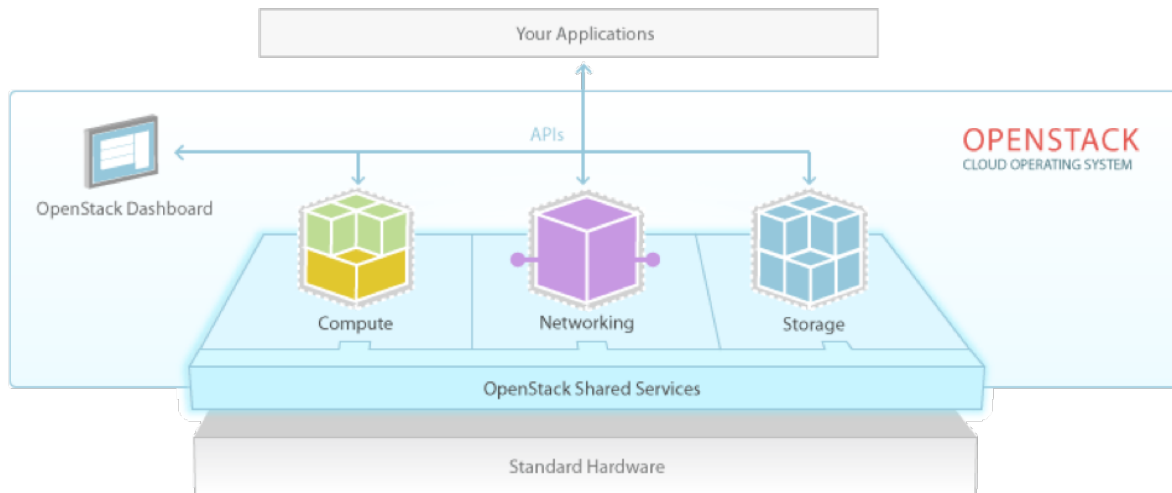
- ISIS
- OSPF
- BGP
- OpenStack Neutron (SDN コントローラ)

OpenStack

- Cloud Operating system
- マルチハイパーバイザサポート
- 様々な SDN コントローラ、ストレージ製品がサポート



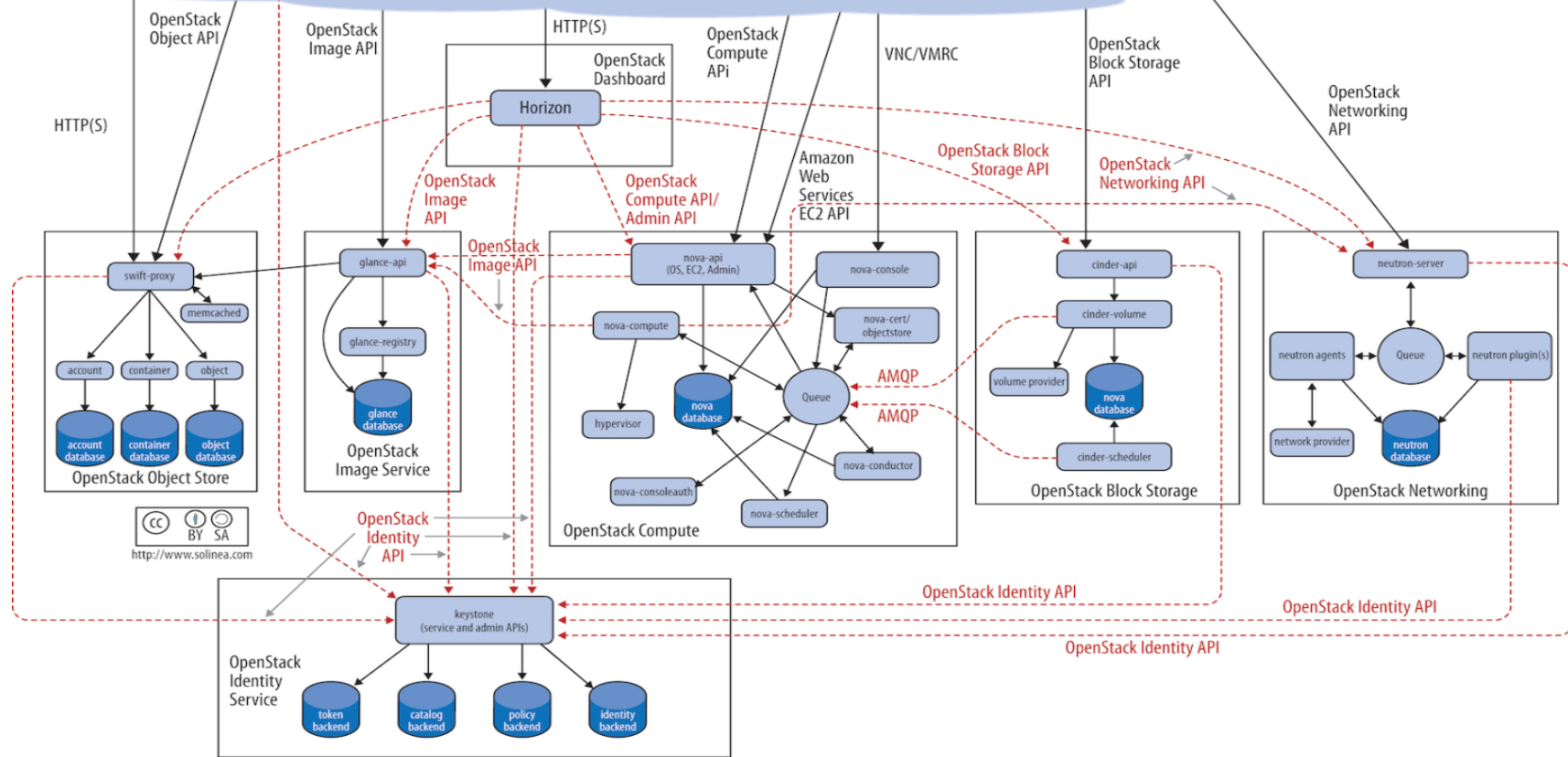
openstack®





- Command-line interfaces (nova, neutron, swift, etc)
- Cloud Management Tools (Rightscale, Enstratus, etc)
- GUI tools (Dashboard, Cyberduck, iPhone client, etc)

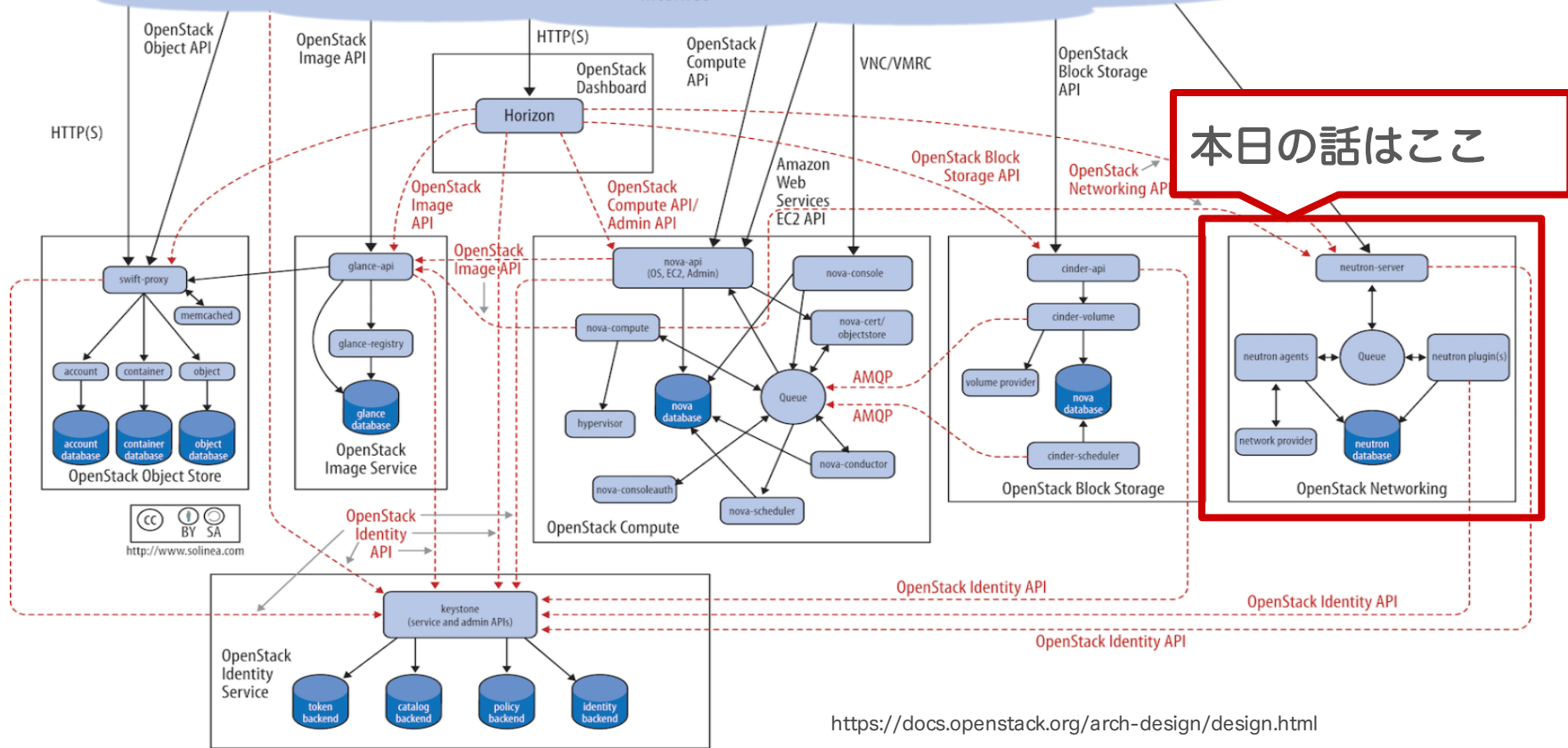
Internet





- Command-line interfaces (nova, neutron, swift, etc)
- Cloud Management Tools (Rightscale, Enstratus, etc)
- GUI tools (Dashboard, Cyberduck, iPhone client, etc)

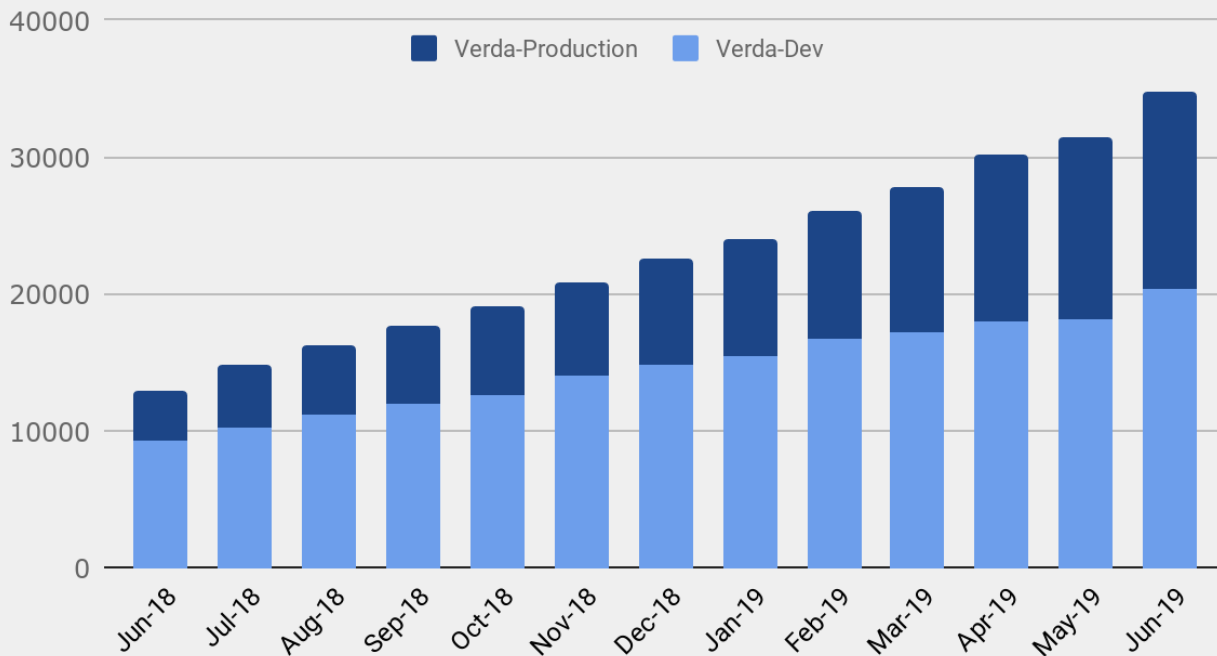
Internet

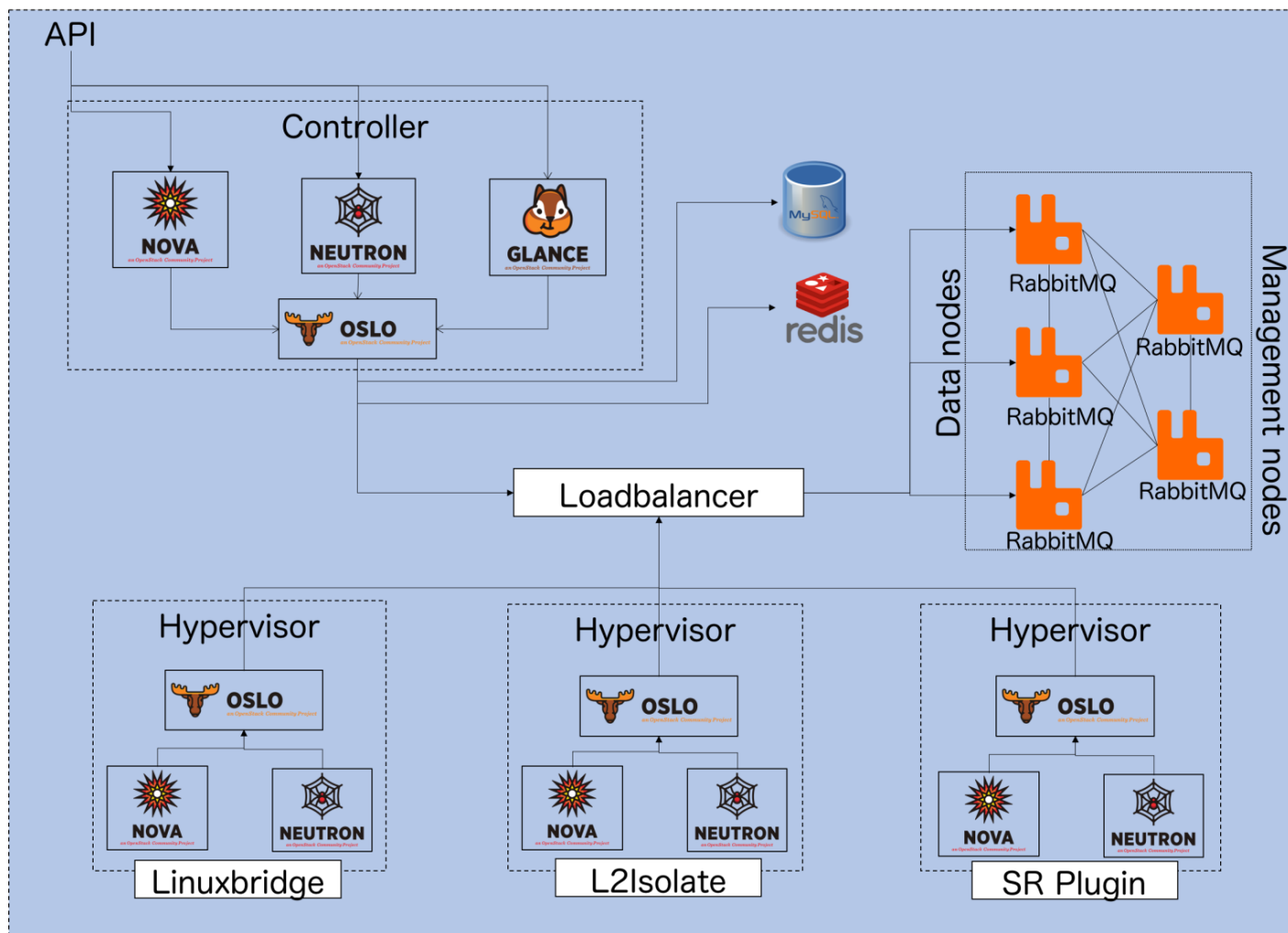


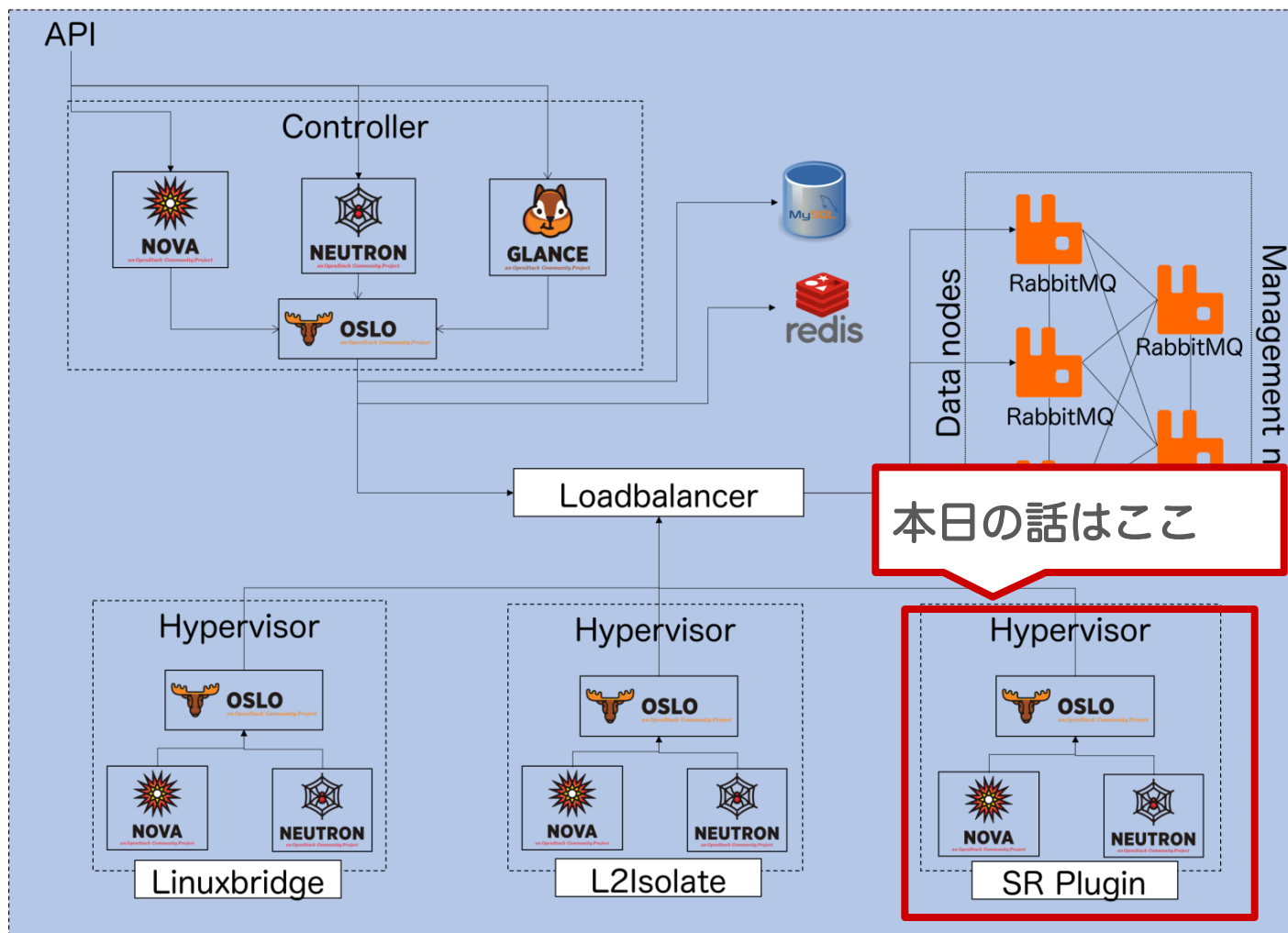


LINE Private IaaS Platform

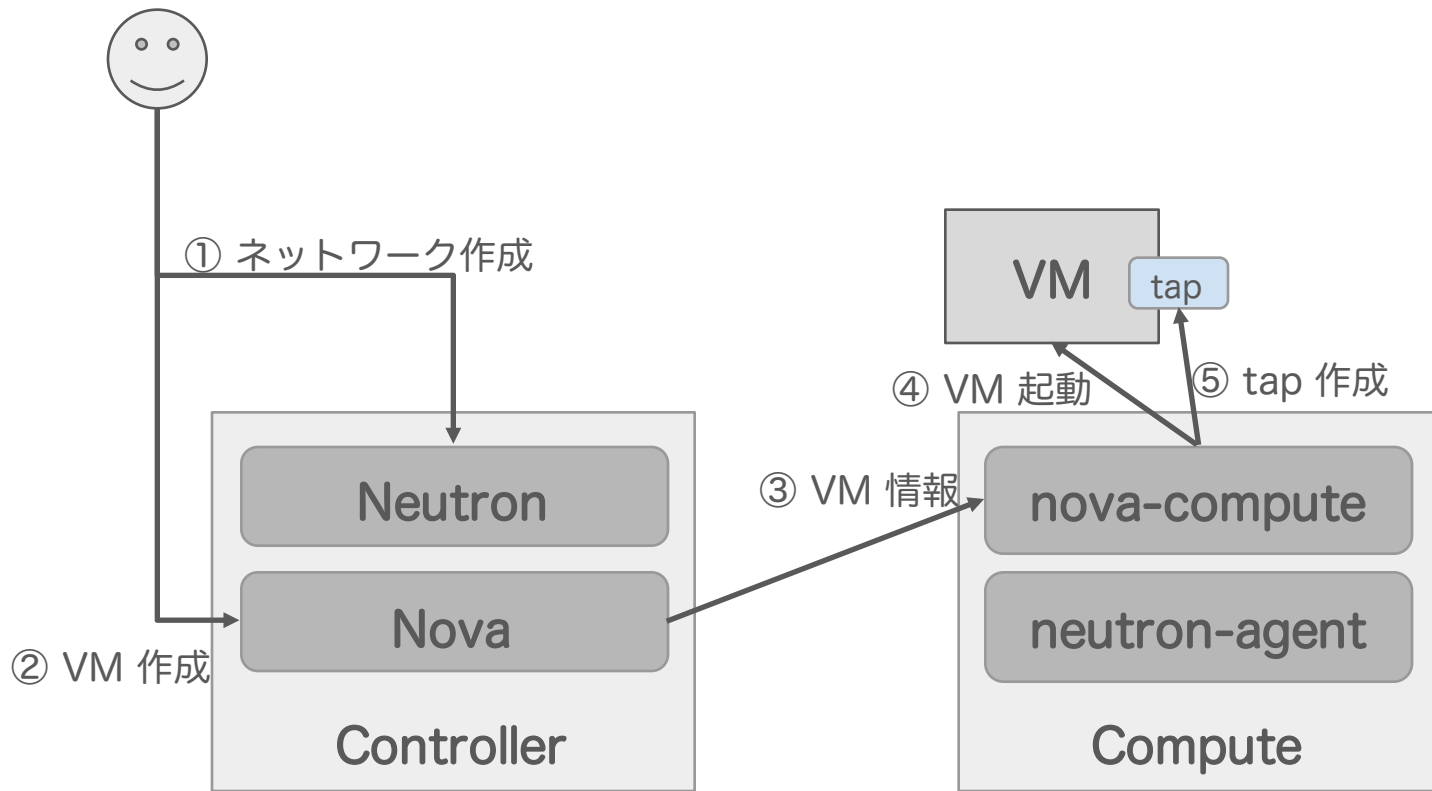
Active VMs transition for Verda-dev & Verda-Production



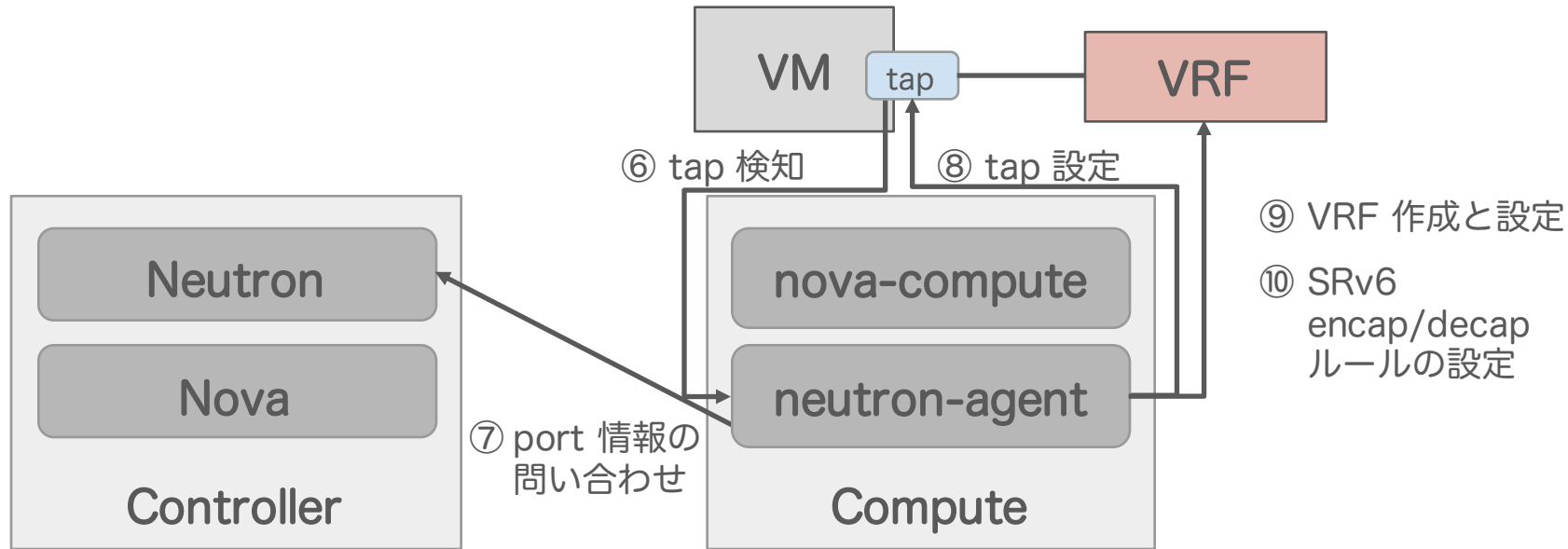




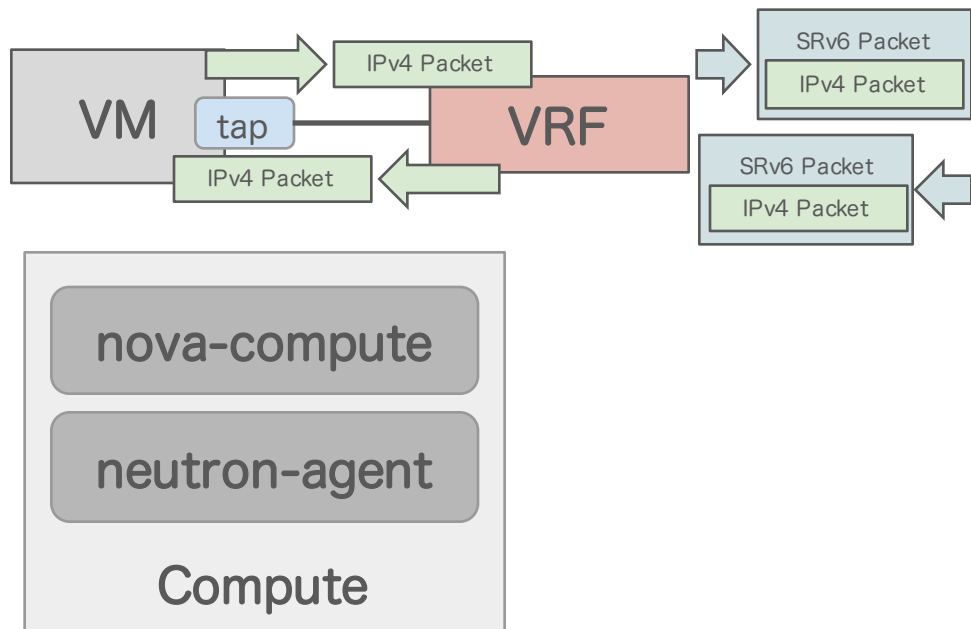
Nova と Neutron の動き VM作成



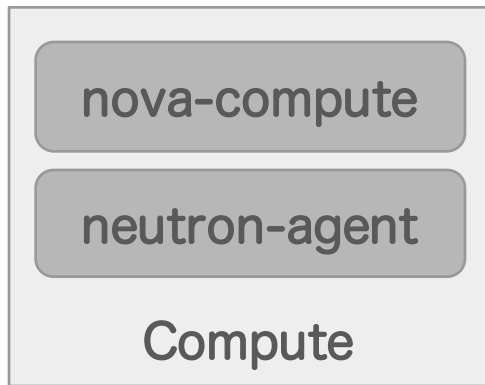
Nova と Neutron の動き NW設定



VM からのパケットは VRF で encap/decap



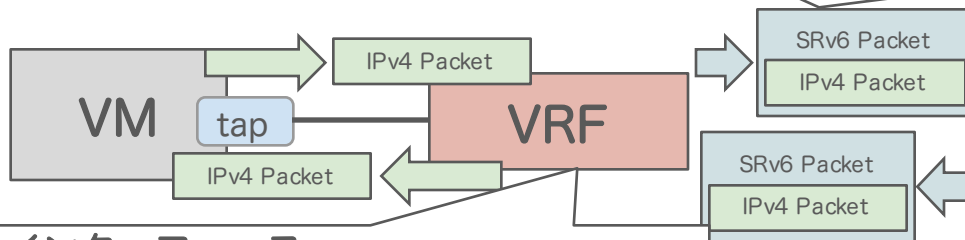
VRF の設定をどうやって取得するか？



SRv6 の設定に必要な情報

Encap ルール

`${送信先のVMのIPv4アドレス} encap seg6 mode encap segs ${送信先のVRFのSID} dev ${VRF Interface 名}`



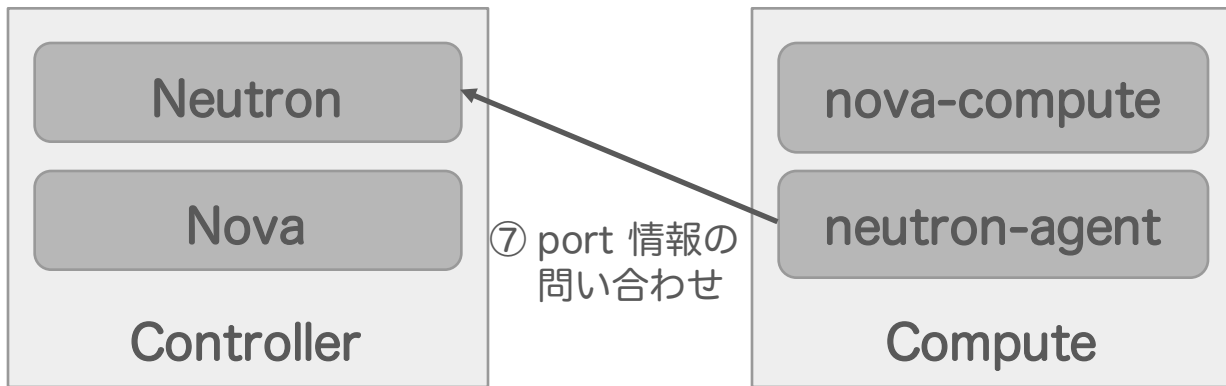
VRF インターフェース

- VRF インターフェース名
- IPv4 アドレス
- VRF の SID (ノードのSID prefix + IPv4 アドレス)

Decap ルール

`local ${受信先のVRFのSID} encap seg6local action End.DX4 nh4 ${受信先のVRFのIPv4} dev ${VRF Interface 名}`

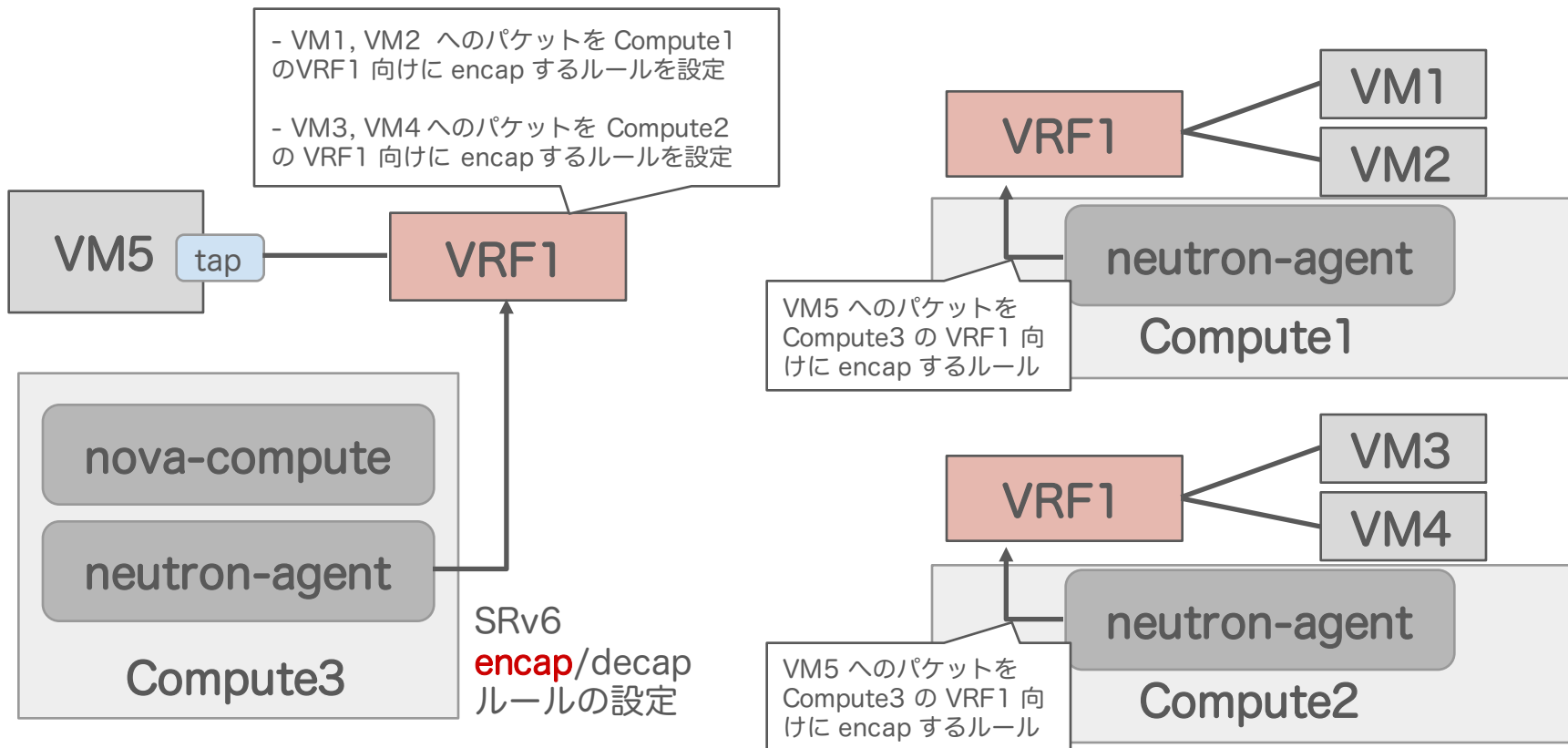
port 情報に VRF 情報を保持



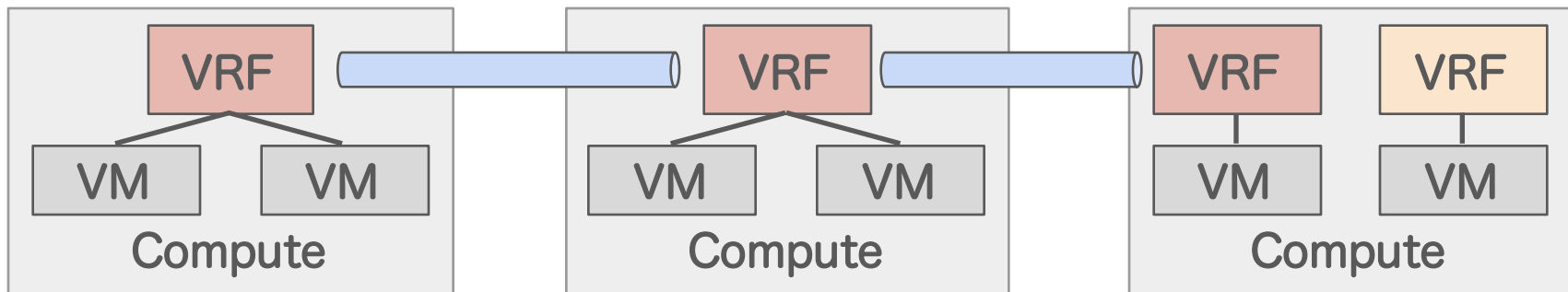
Neutron port の binding:profile を利用

```
{  
  "port":{  
    "binding:profile": {  
      "segment_node_id": "2400:dcc0::a7a:4d8e", # この port を持つ VM が起動しているノードの SID prefix  
      "vrf": "vrf644606a29039", # この port が接続する VRF のインターフェース名 "vrf" + tenant_id + network_id  
      "vrf_cidr": "169.254.1.0/24", # この port が接続する VRF の IP CIDR  
      "vrf_ip": "169.254.1.44" # この port が接続する VRF の IP アドレス  
    }  
  }  
}
```

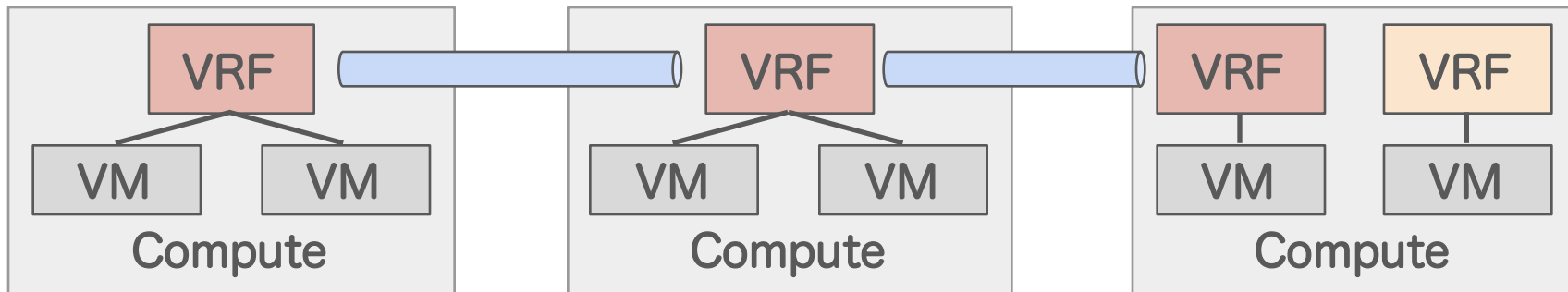
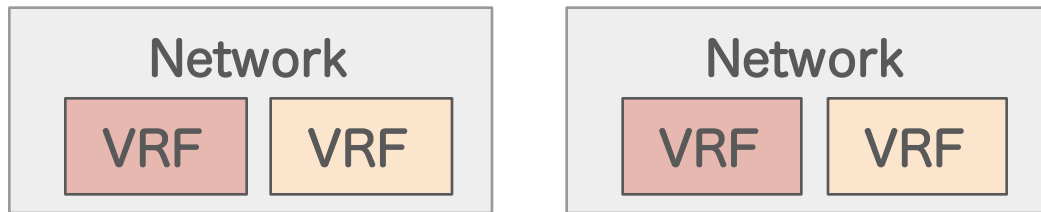

各 VM の port 情報から encap ルールを設定



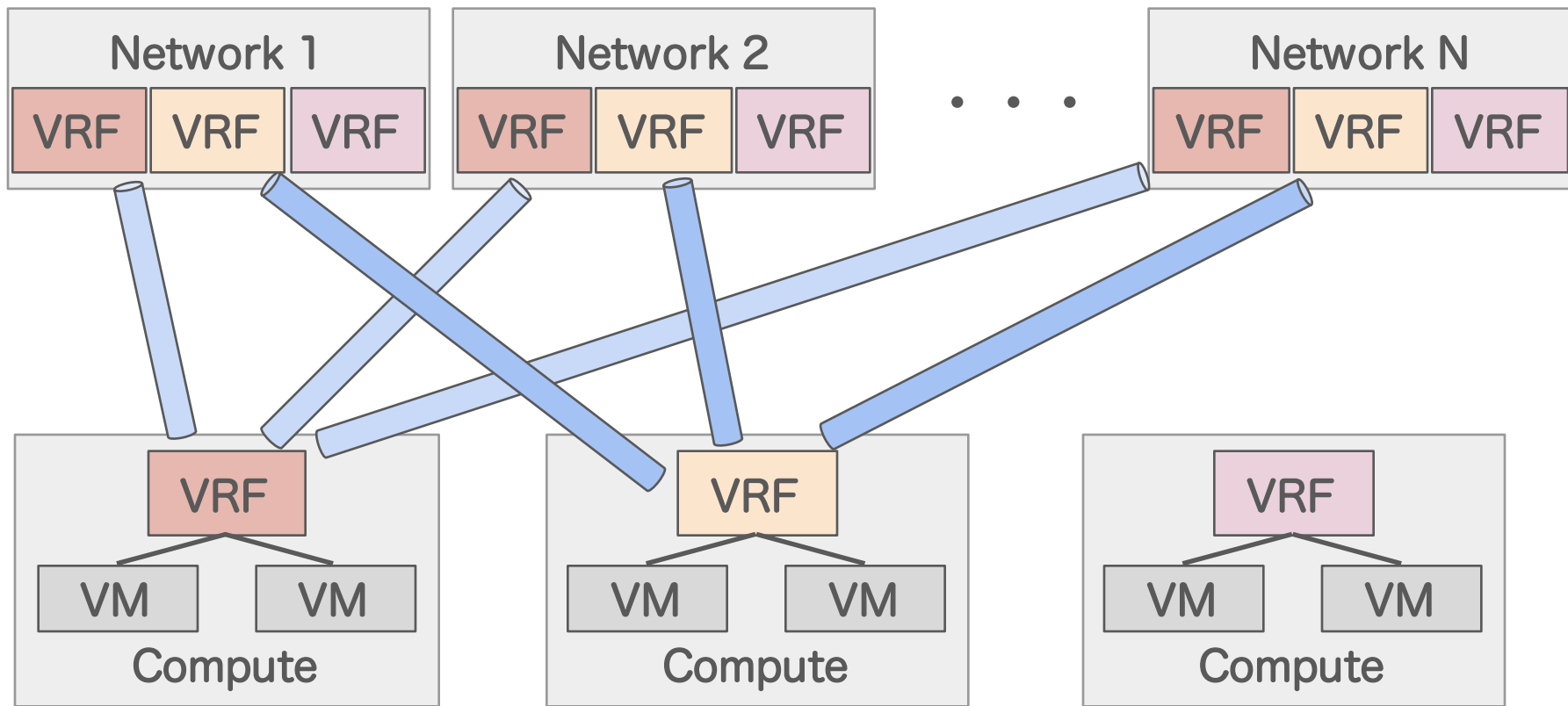
ここまでで VM 間 SRv6 通信が実現



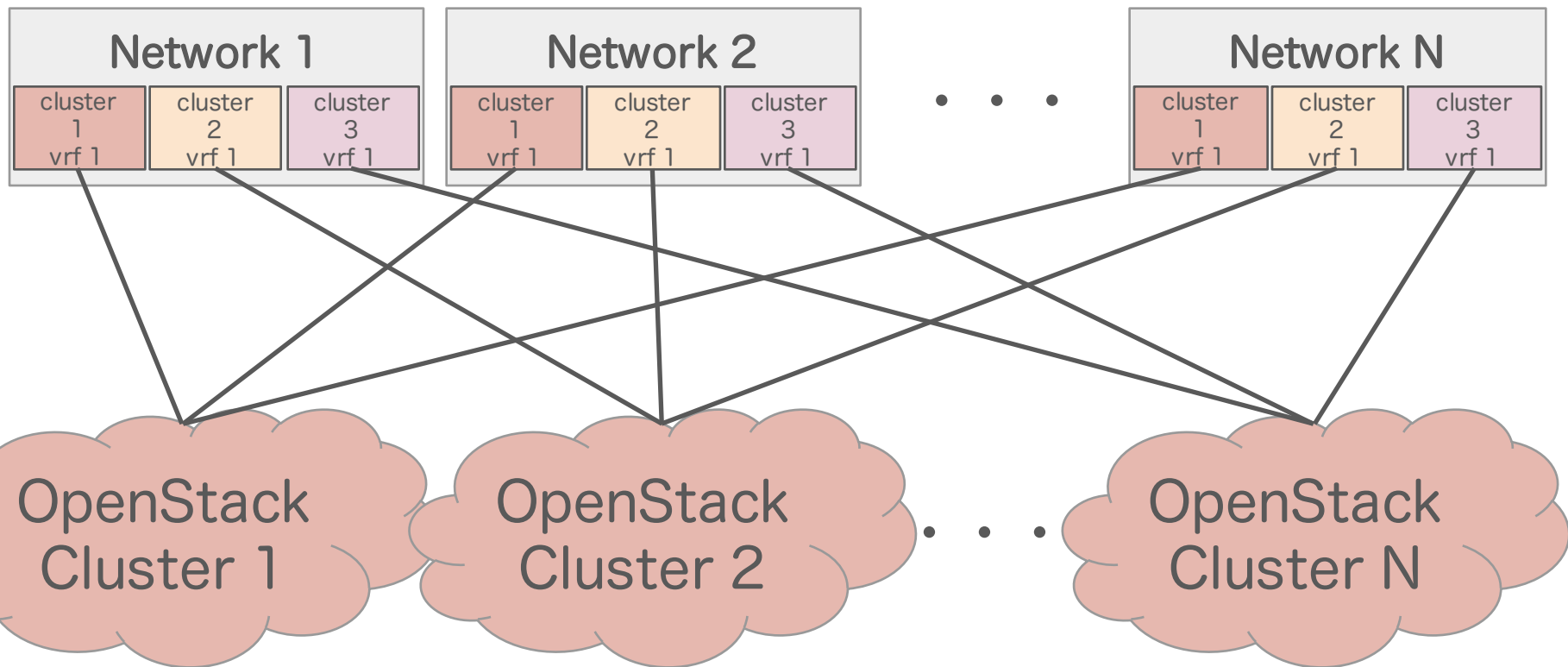
VM ⇔ 外部 間の通信のためのネットワークノード



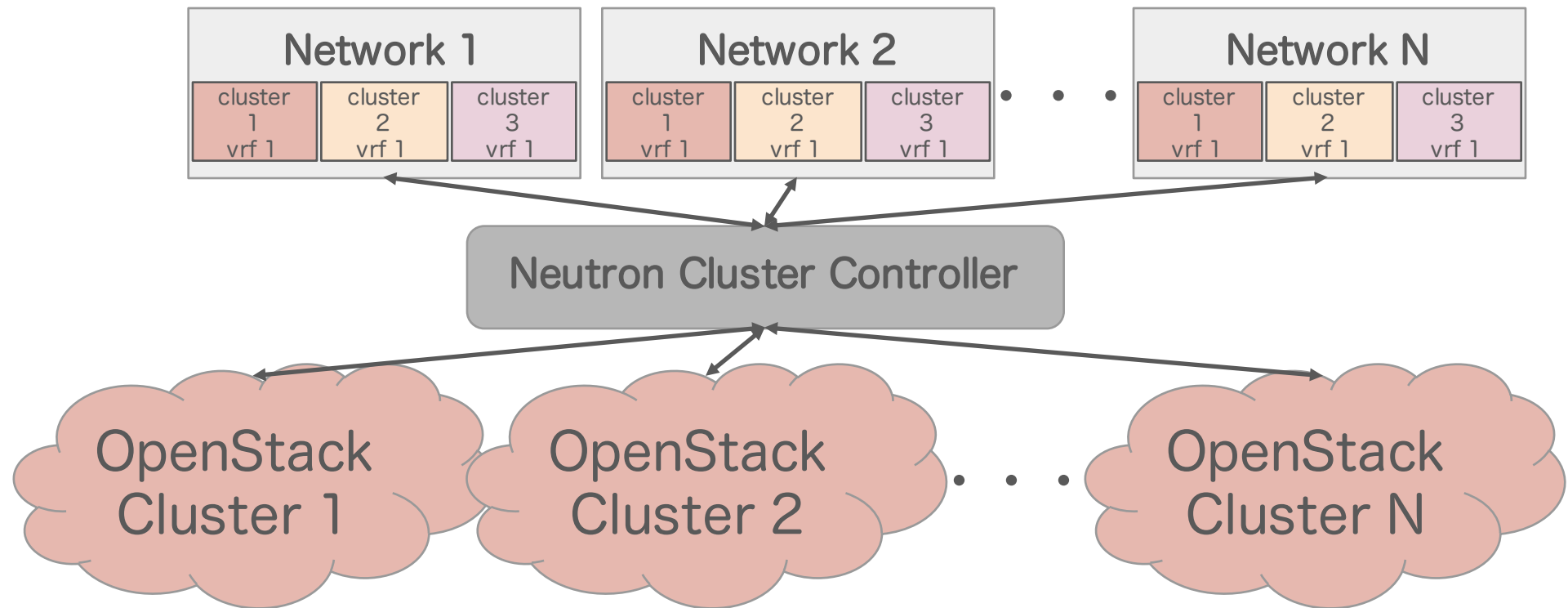
ネットワークノードの要件: スケール



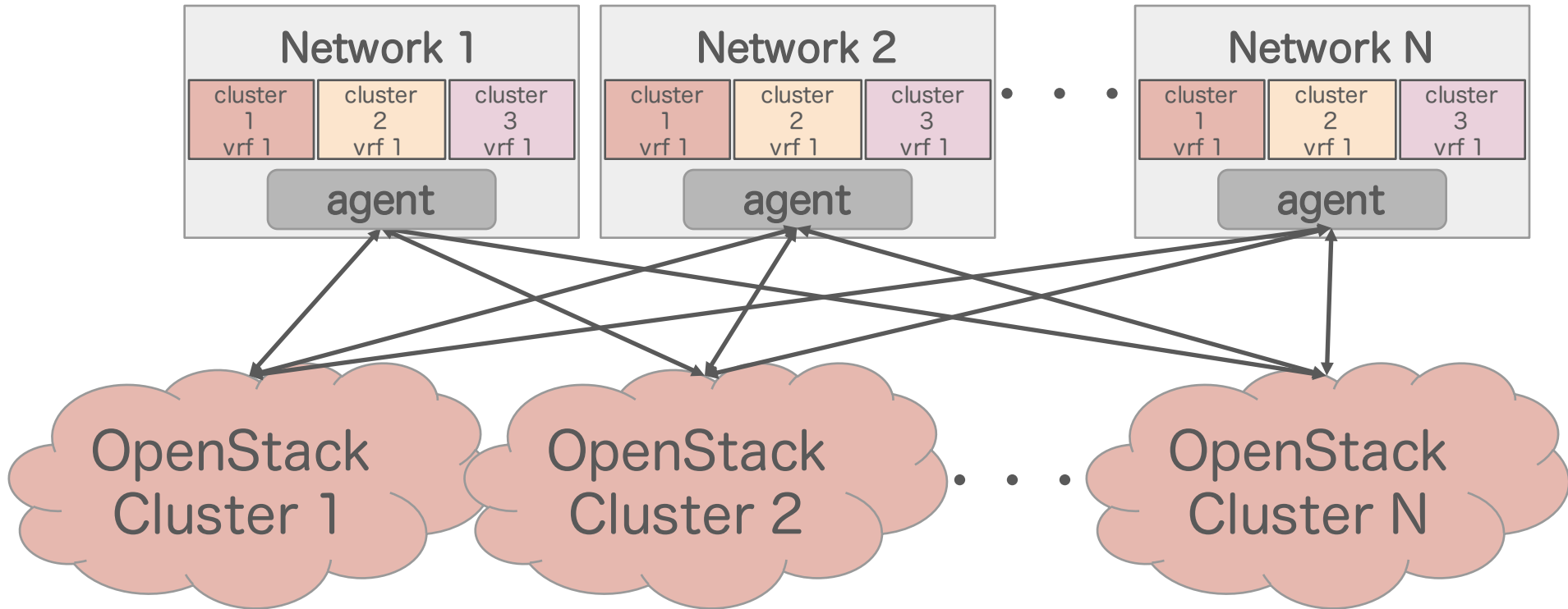
ネットワークノードの要件: マルチクラスタ



Centralized vs Distributed



Centralized vs Distributed



Centralized vs Distributed

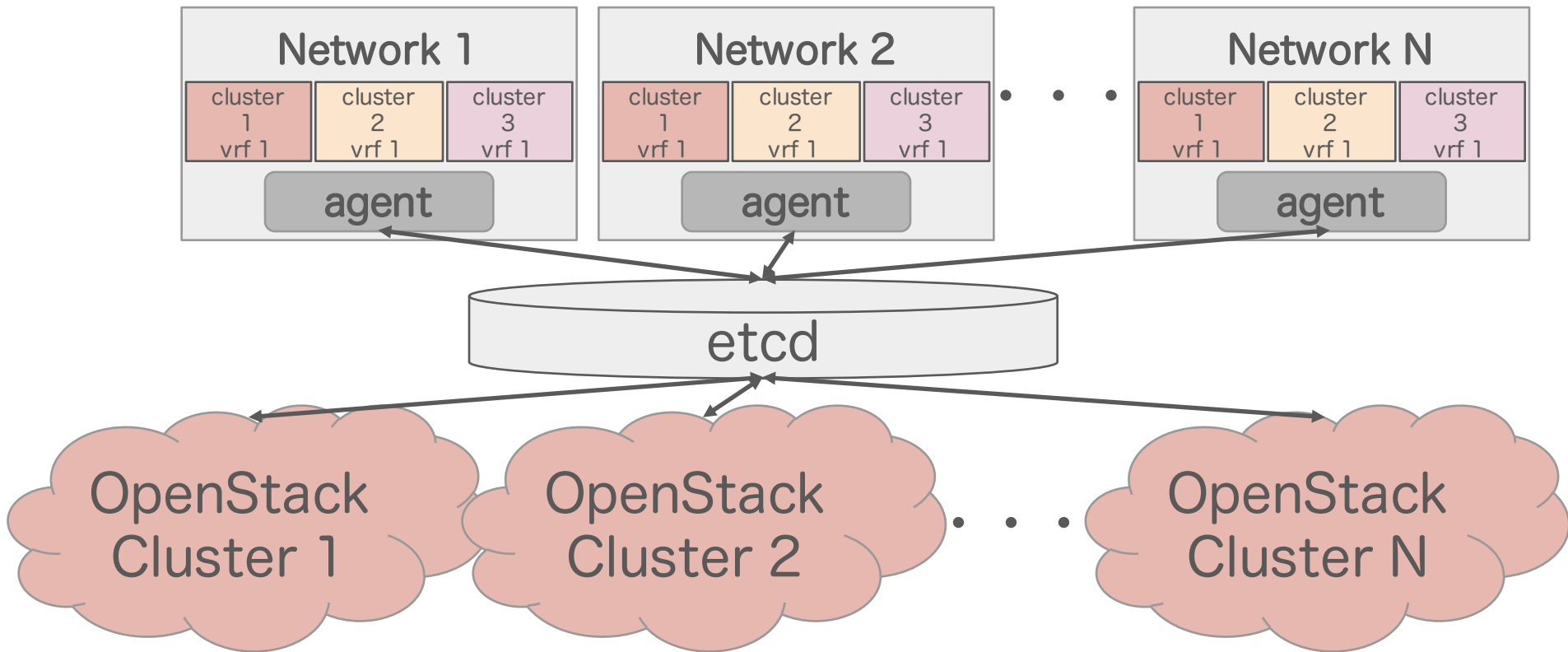
- Centralized

- 各 OpenStack Cluster の Neutron を集約する Neutron Cluster Controller がすべてを管理する
- Cons: この Controller が SPoF になる可能性がある

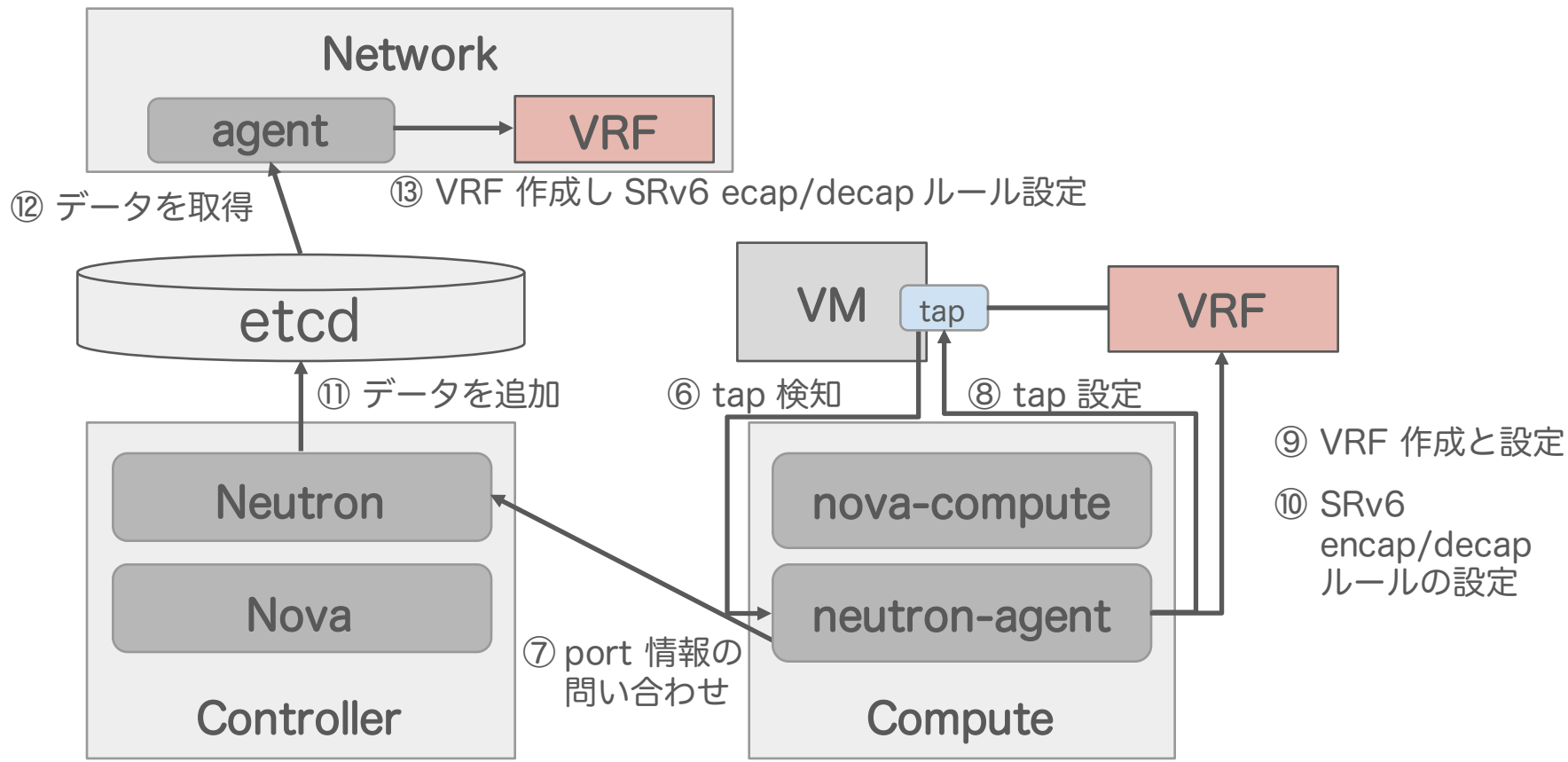
- Distributed

- 各 Network ノード上の agent が各 OpenStack Cluster の Neutron の情報をポーリングして必要な設定をする
- Cons: クラスタ追加時に agent は動的にその新しいクラスタを検知する必要がある

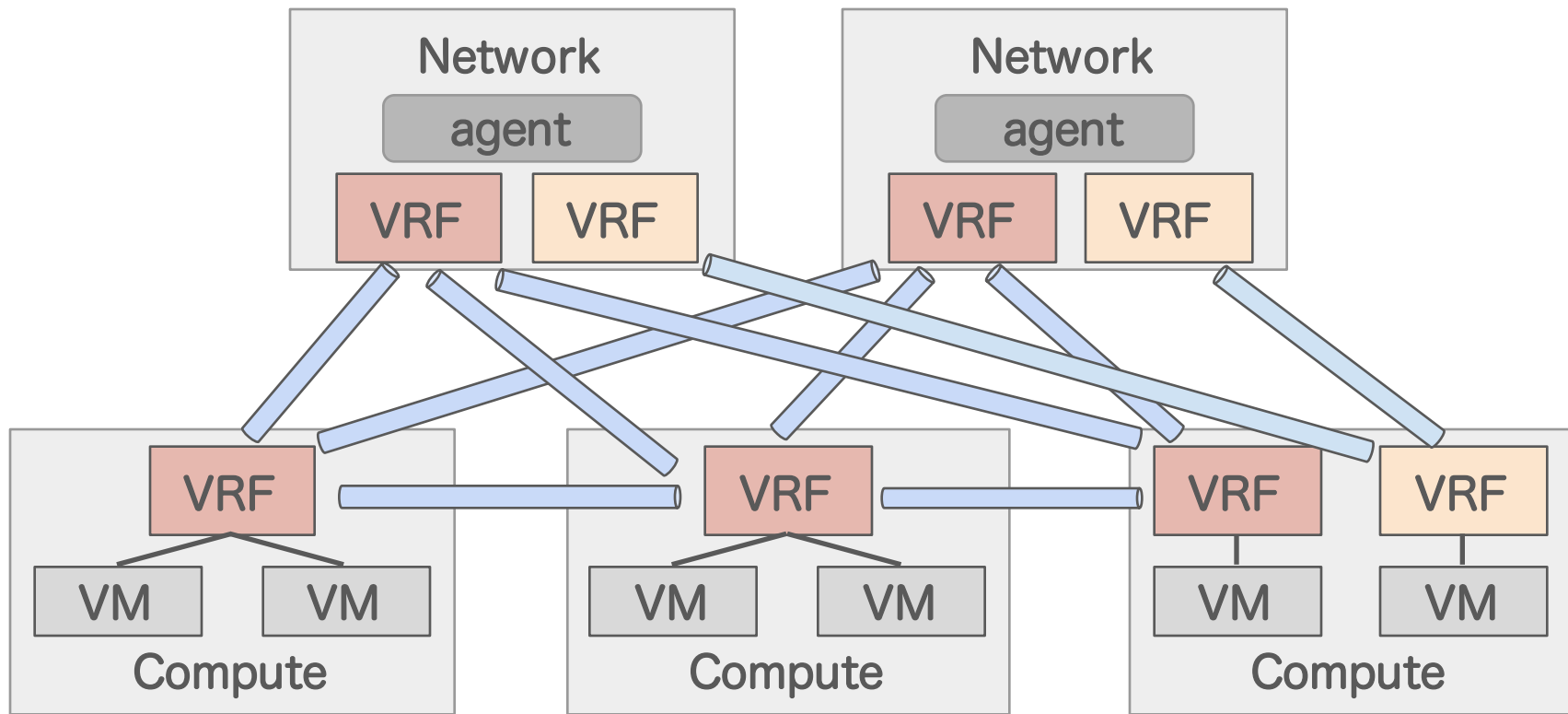
etcd + agent モデル



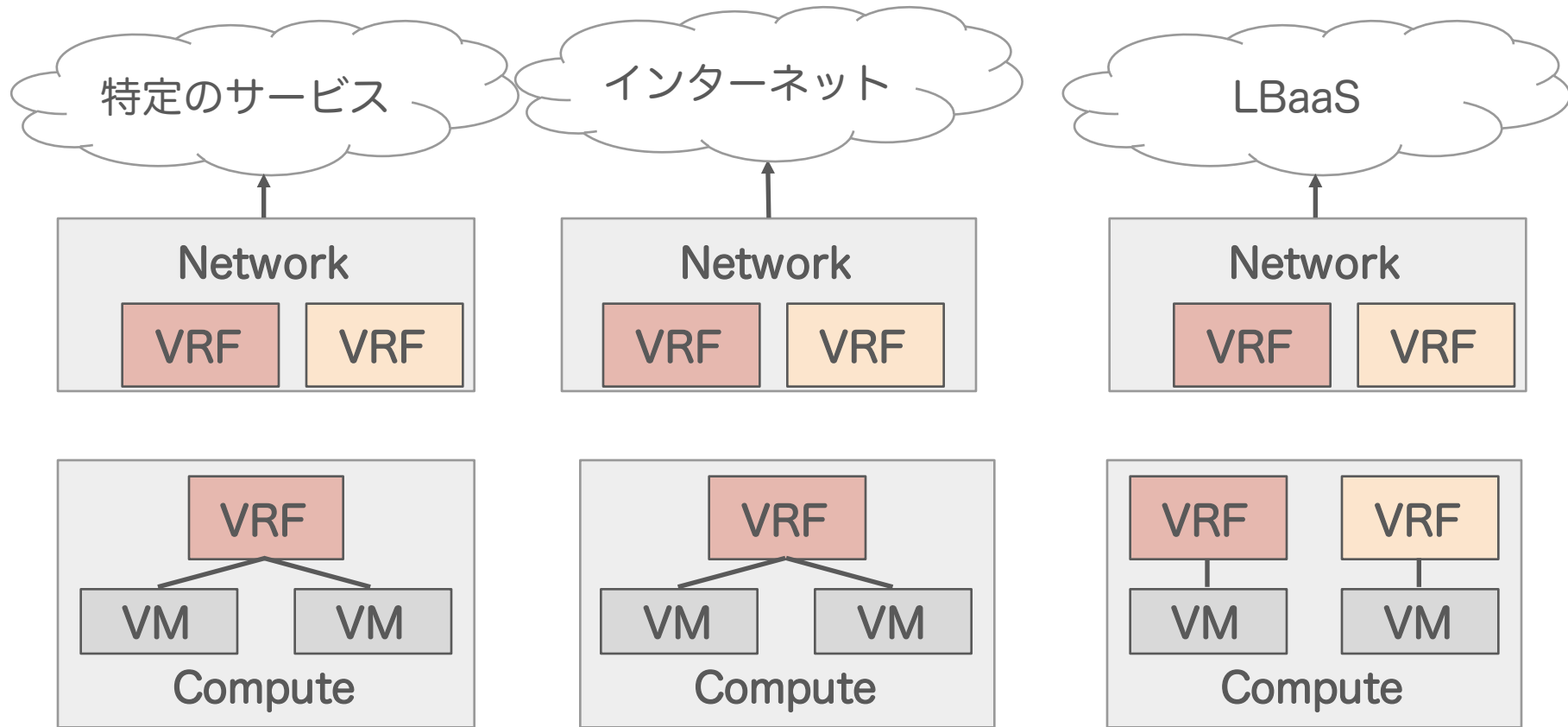
etcd を経由して必要な情報を渡す



VM 間、VM - 外部間 SRv6 通信が実現



VM - *間 SRv6 通信が必要



ユースケース

- 特定のサービス
 - 社内 DNS サーバへの接続
 - OpenStack API サーバへの接続
- LBaaS
 - ユーザが仮想ロードバランサを作成した際に生成される VIP アドレスで仮想ロードバランサへの接続
 - 仮想ロードバランサおよび VIP は生成したユーザが所属するプロジェクト内でのみ共有（他プロジェクトのユーザの VM からは接続できない）



ユースケース

- 特定のサービス
 - 社内 DNS サーバへの接続
 - OpenStack API サーバへの接続
- LBaaS
 - ユーザが仮想ロードバランサを作成した際に生成される VIP アドレスで仮想ロードバランサへの接続
 - 仮想ロードバランサおよび VIP は生成したユーザが所属するプロジェクト内でのみ共有（他プロジェクトのユーザの VM からは接続できない）



srv6_encap_network API

SRv6 Encap Network

SRv6 Encap Network

The **srv6_encap_network** extension lists, creates, shows information for, and updates srv6_encap_network resource.

GET

/v2.0/srv6_encap_networks
List srv6 encap networks

POST

/v2.0/srv6_encap_networks
Create srv6 encap network

GET

/v2.0/srv6_encap_networks/ {srv6_encap_network_id}
Show srv6 encap network

PUT

/v2.0/srv6_encap_networks/ {srv6_encap_network_id}
Update srv6 encap network

DELETE

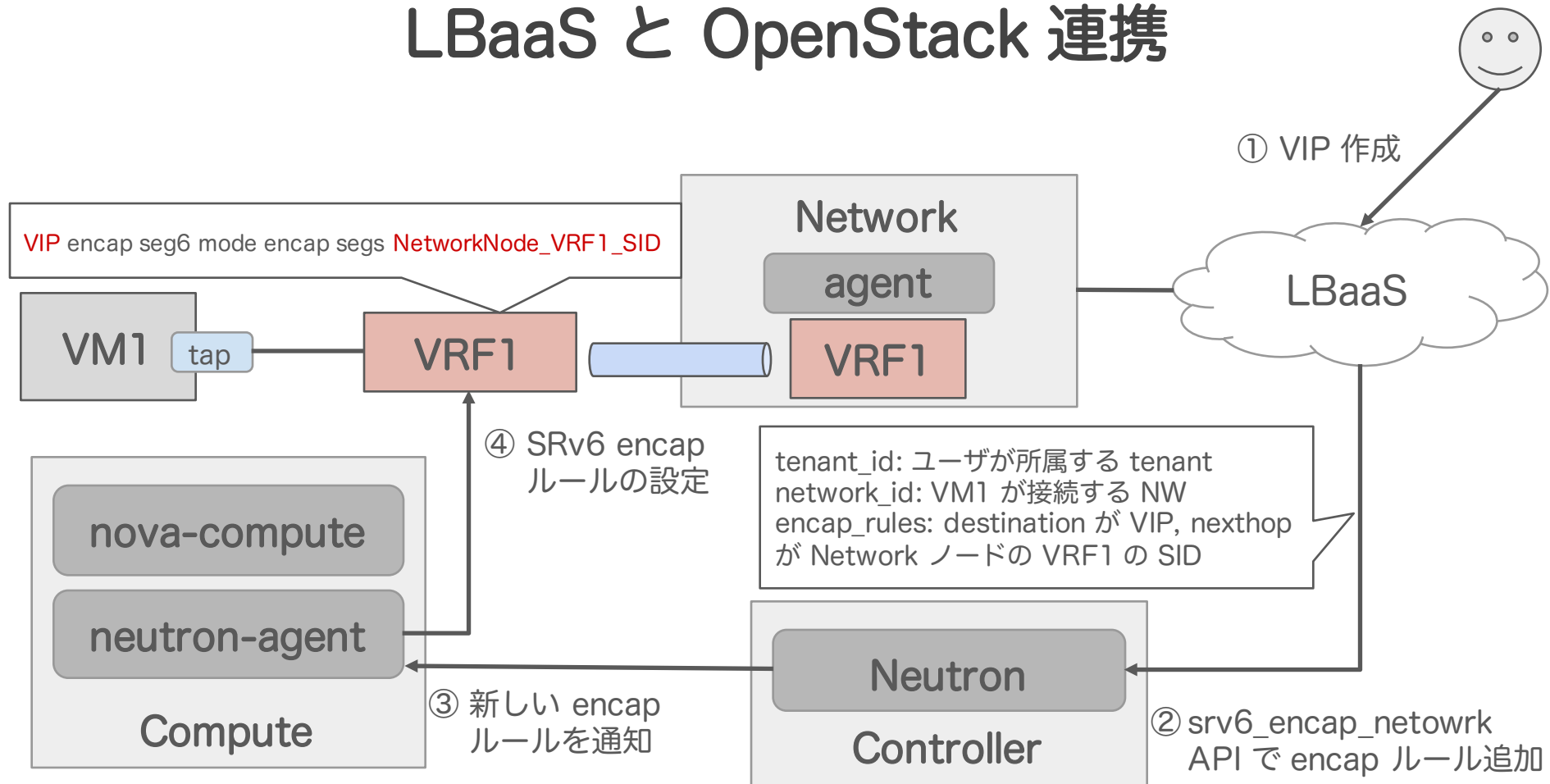
/v2.0/srv6_encap_networks/ {srv6_encap_network_id}
Delete srv6 encap network

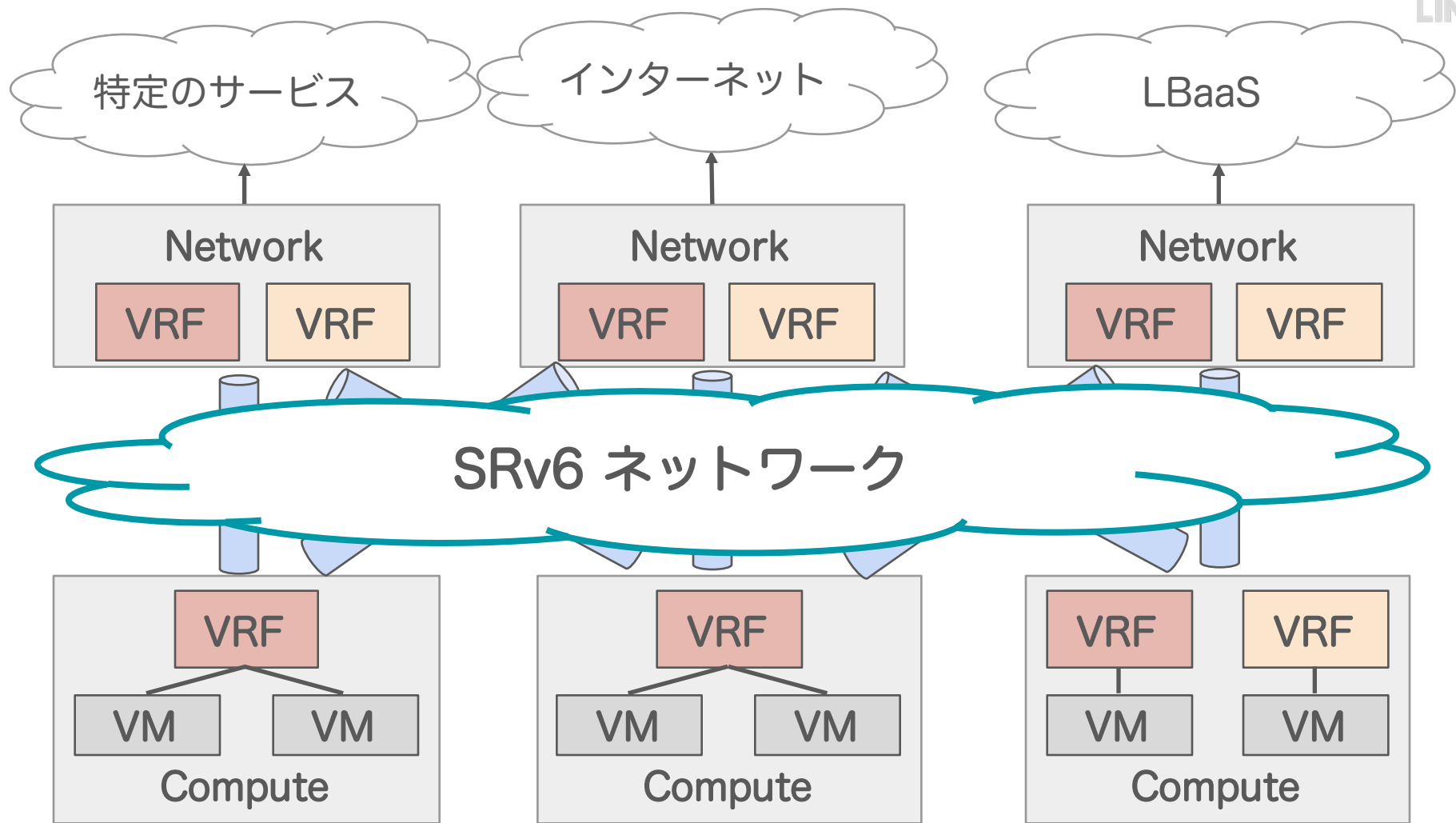
```
{
  "srv6_encap_networks": [
    {
      "network_id": "fbc5f08e-0cb0-4b5c-a5ce-ac7032f50c7b",
      "tenant_id": "d988b205c6e142669a290dd80010587c",
      "project_id": "d988b205c6e142669a290dd80010587c",
      "encap_rules": [
        {
          "nexthop": "fc00:17::a00:fa",
          "destination": "10.0.201.200"
        }
      ],
      "id": "43938fab-ce22-442f-b537-24f2768de773"
    },
    {
      "network_id": "d76c20be-5c2a-40c5-bbd5-0b192fa3ff9c",
      "tenant_id": "aac15739c8034f60b2e8278e84563919",
      "project_id": "aac15739c8034f60b2e8278e84563919",
      "encap_rules": [
        {
          "nexthop": "fc00:17::a00:fa",
          "destination": "10.0.201.201"
        }
      ],
      "id": "70989e81-eae4-490c-b016-f665e6bc872f"
    }
  ]
}
```

srv6_encap_network リソース

- id: リソースを識別する ID
- tenant_id/project_id: リソースのプロジェクトの ID
- network_id: リソースがアサインされている ID
- encap_rules: SRv6 の encap ルールリスト
 - destination: 宛先の IPv4 アドレス
 - nexthop: SRv6 encap に使用する SID

LBaaS と OpenStack 連携





コントロールプレーン設計で大変だったこと

- SRv6 の情報をどのように保持するか、どのタイミングで、どのように設定をするか
- みんなコントロールプレーンで何でもできると思ってる
 - OpenStack と全く関係のない世界の情報を元に設定を要求
 - まだ存在しないリソースを元に設定することを要求
 - 複雑なロジックをかけていい感じに制御してくれると思ってる
- C-plane と D-plane の境界付近で問題が続出した
 - VM 起動時の DHCP や cloud-init、vrf + NW filter 問題

コントロールプレーン設計で気をつけたこと

- OpenStack の設計・思想に則った設計にする
- ユースケースにとらわれず抽象度を上げて対応する
- ハイパースケールな作りにするが、諦める所は諦める
- 複雑なロジックを入れずに出来る限りシンプルに作る
- コントロールプレーンとデータプレーンは疎結合にする

今後やりたいこと

- OSS として公開
- networking-sfc 対応
- Multiple SR driver 対応
 - Kernel
 - VPP
 - OVS
- SRv6 CNI

議論したいポイント

- データセンターの仮想化基盤上で SRv6 を利用することについてどう思いましたか？
- SIDの設計方針やSIDの設定方法について、自分はこうしたら良いと思うという意見ありますか？
- 今後 SRv6 に期待している点やこういう世界になるべき、といった意見ありますか？