LINE Data Center Networking with SRv6

Hirofumi Ichihara
co-author: Toshiki Tsuchiya
LINE corporation

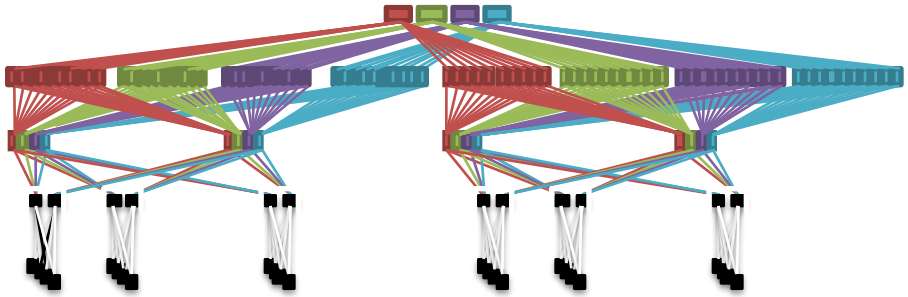
About Me

- Hirofumi Ichihara
- LINE Corporation
  - Network Development Team
- Network Software Developer
  - SDN/NFV
  - OpenStack Neutron
  - Docker
  - Kubernetes

LINE Services and Networks

Full L3 CLOS Network*

- Single tenant network
- LINE message service and related services running



* Excitingly simple multi-path OpenStack networking: LAG-less, L2-less, yet fully redundant
https://www.slideshare.net/linecorp/excitingly-simple-multipath-openstack-networking-lagless-l2less-yet-fully-redundant

Exclusive Network for Services

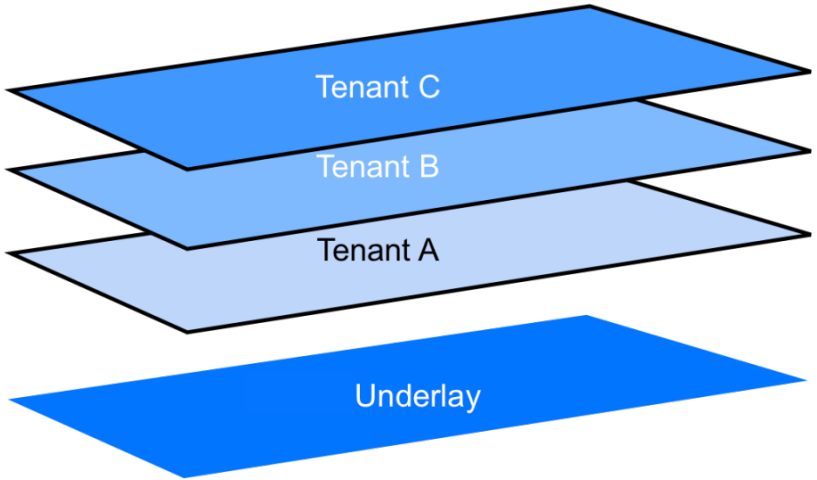
- Service with specific requirements running
- Building specific network for each service

Other: Fintech Business

Many fragment underlay networks
Many works to design and build
Management cost increases

Multi tenant network

- Sharing underlay network decrease management cost
- Achieve policy for each service(tenant) on overly network



Flexibly scale overlay network
Security individual tenant
Service Chaining

Simple L3 underlay network

Multi tenancy

VXLAN

Pros

- More information
- Many network devices support

Cons

- Lose advances of full-L3
- Need additional protocol to achieve service

IPv6 Segment Routing (SRv6)

Pros

- IPv6 forwarding only on underlay
- Support segregation and service chaining with Segment ID

Cons

- No information about DC use case
- No network device support

+ SRv6 future

Adopted SRv6

SRv6

Segment ID (SID)

- 128bit number(IPv6 address)
- **Locator**: Information for routing to SRv6 node(parent node). It must be unique whitin a SR domain
- **Function**: Information to identify the action to be performed on the parent node

Segment Routing Header (SRH)

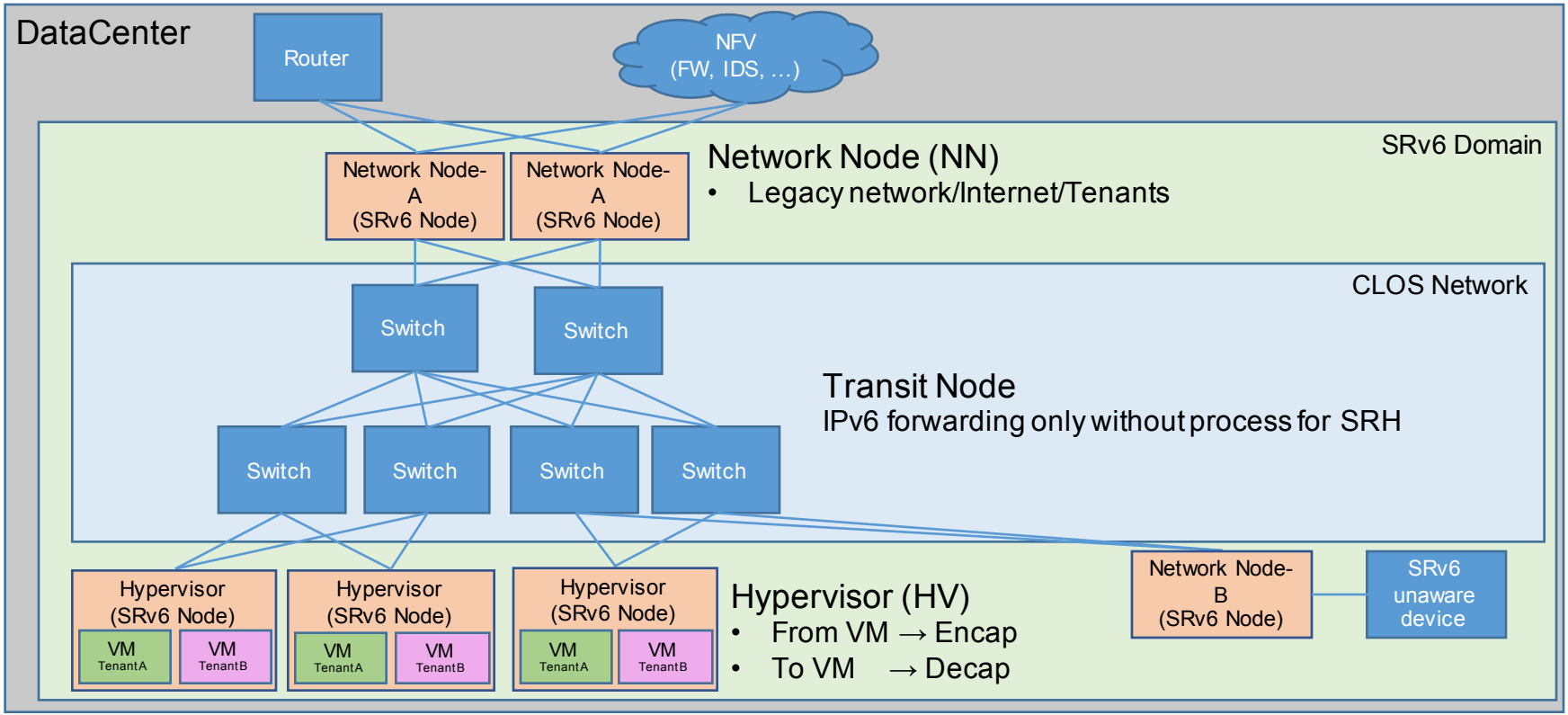
- IPv6 extension header
- Including a Segment List, Segment Left points out current point of Segment List and so on

Function examples

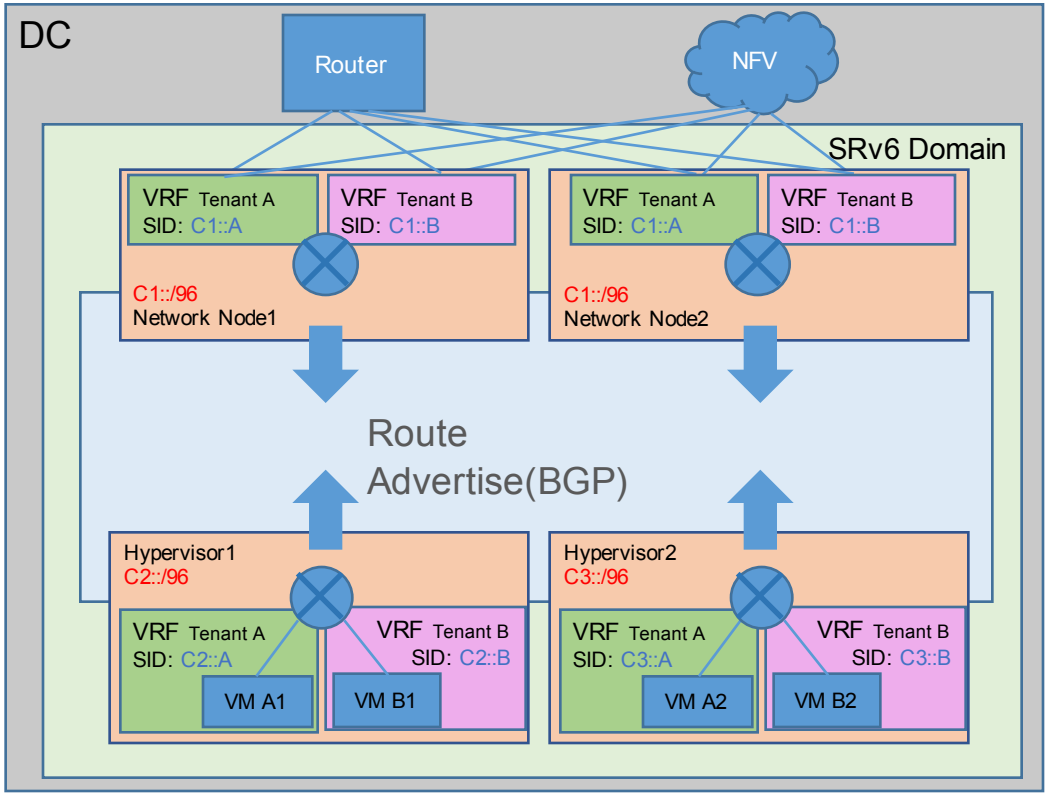
- **T.Encaps(Encap)**: Encapsulation packet with IPv6 header and SRH
- **End.DX4(Decap)**: Remove IPv6 header and SRH from packet and then forward next hop
- **End.DT4(Decap)**: Remove IPv6 header and SRH from packet and then lookup routing table and forward (DT4 is not implemented in Linux Kernel so we used DX4 although DT4 is better)

SRv6 Data Center Network Data Plane

Data Plane - Architecture

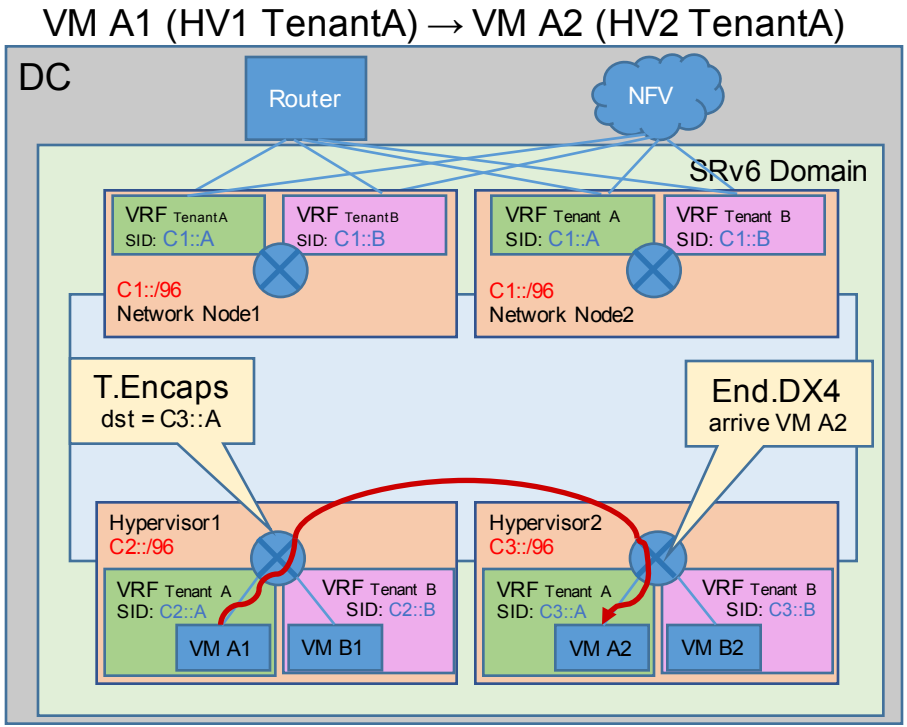


Data Plane - SID, Routing

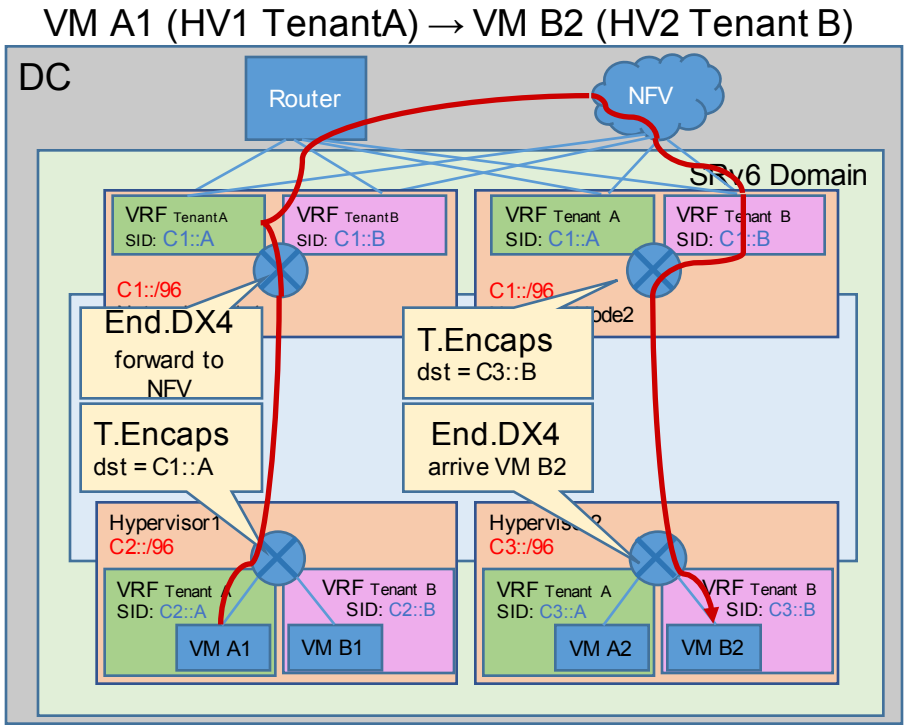


- Create VRF (l3master device) for each tenant on NetworkNode, Hypervisor
- Assign IPv6 address /96 block **(Locator)** to nodes(NetworkNode, Hypervisor)
- Add identifier for each tenant to the Locator as **Function** (LINE uses specific address from 169.254.0.0/16 each tenant)
- Advertise /96 IPv6 address(Locator) via BGP

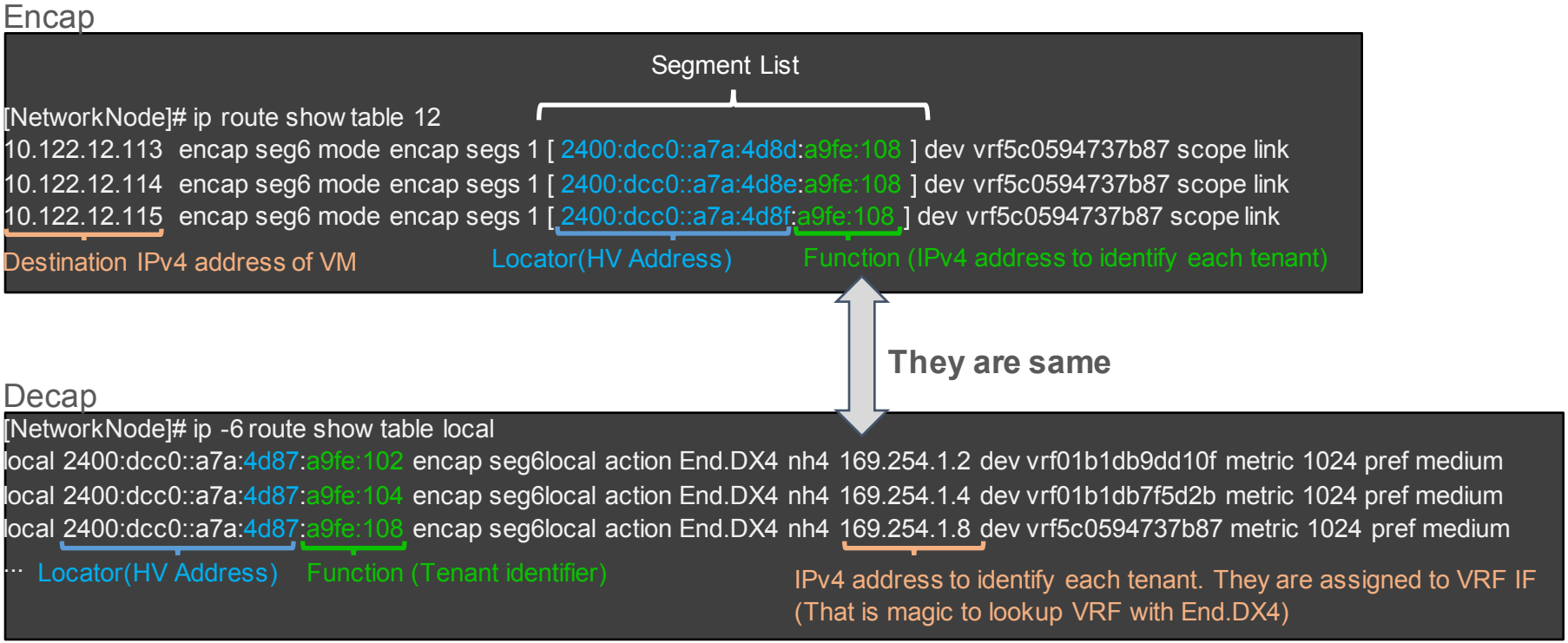
Data Plane - Packet flow in a tenant



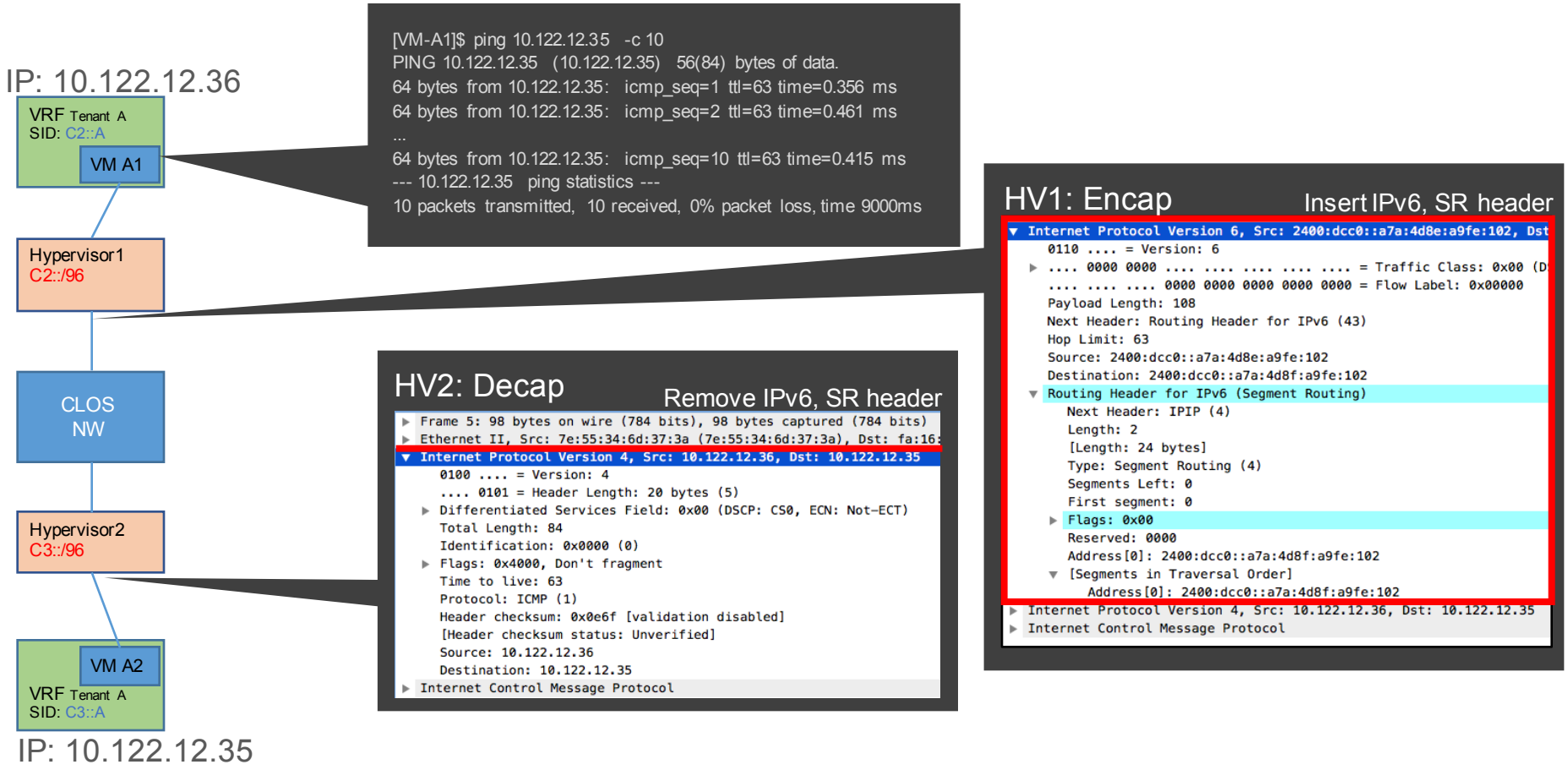
Data Plane - Packet flow between tenants



Data Plane - Real config on Network Node



Data Plane - Real behavior



SRv6 Data Center Network Control Plane

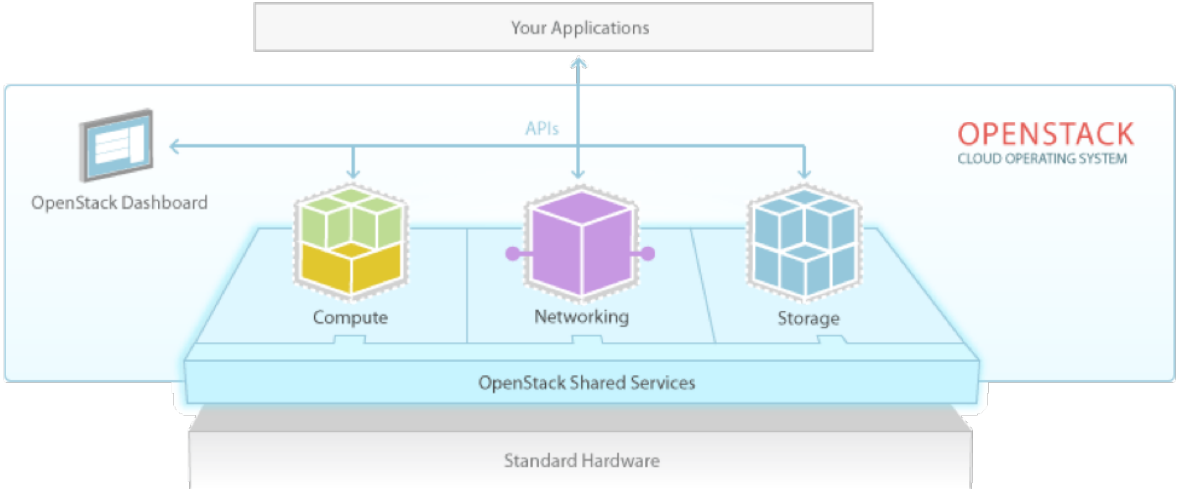
SRv6 Control Plane Choices

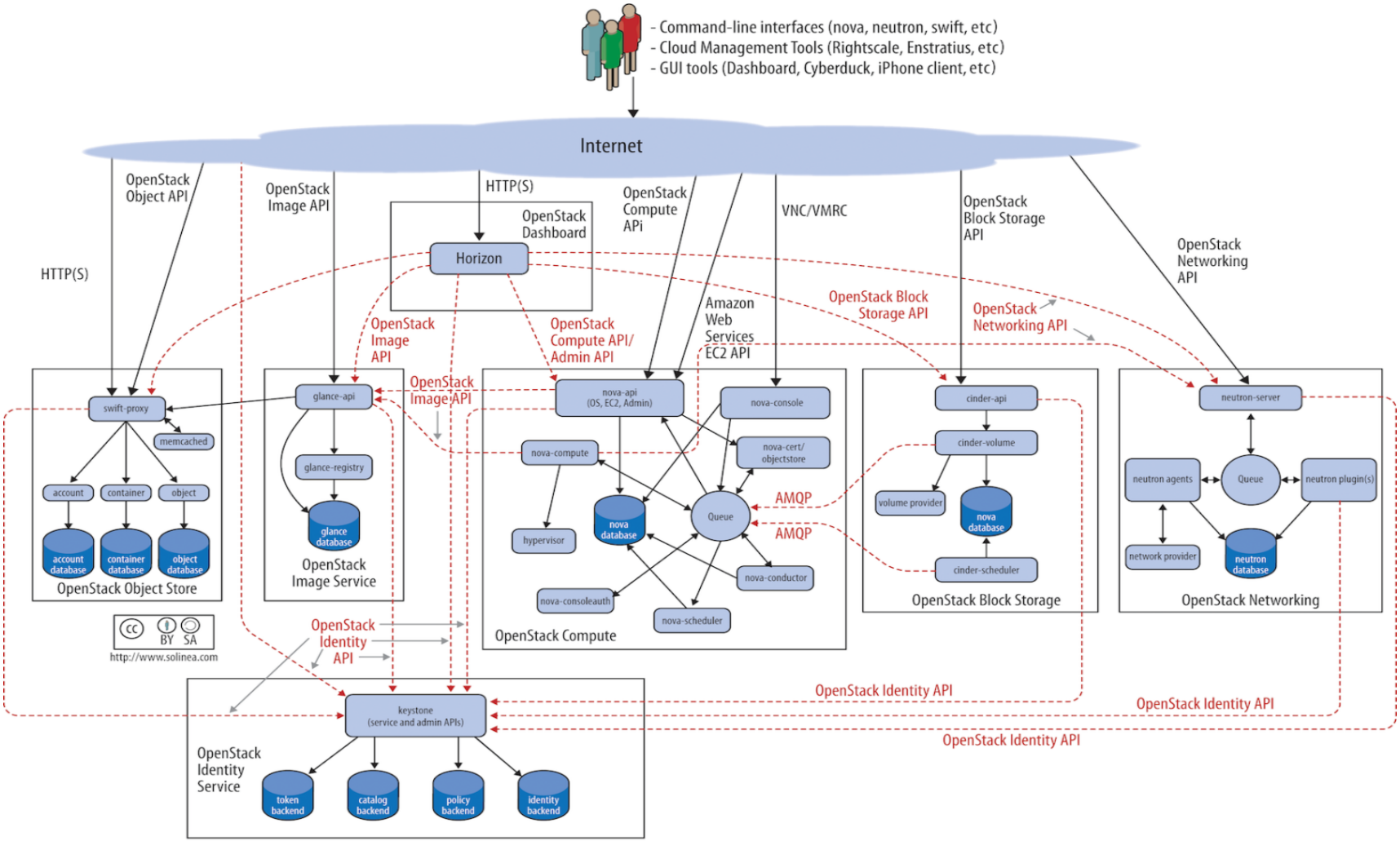
- ISIS
- OSPF
- BGP
- SDN Controller

LINE uses OpenStack as Private Cloud Controller so adopted SDN Controller

OpenStack

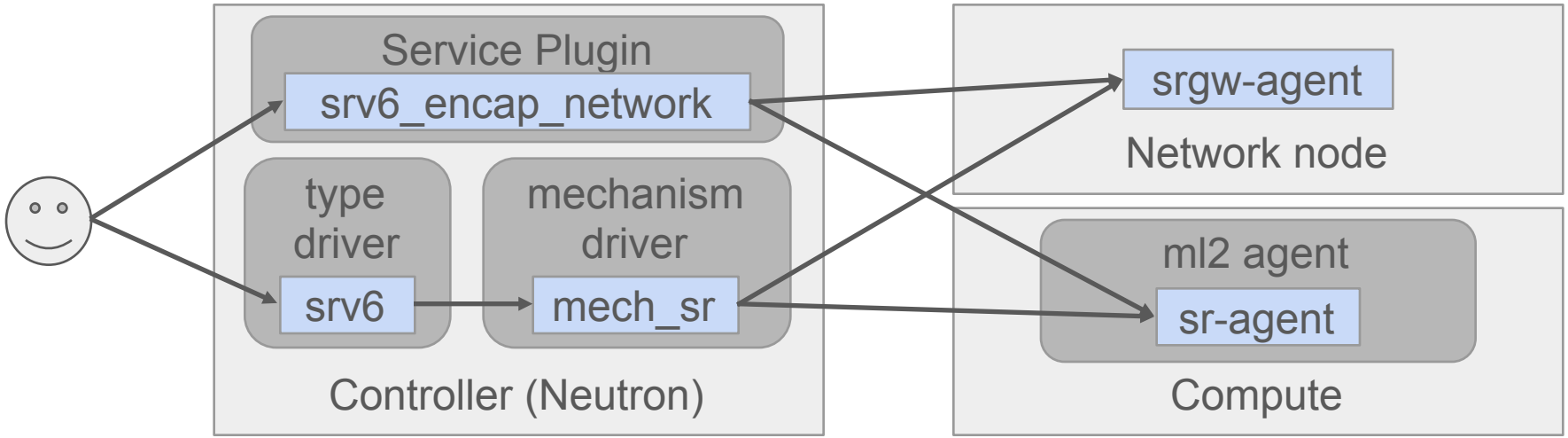
- Cloud Operating system
- Support Multi Hypervisor
- Support various SDN controllers and Storage appliances




- Command-line interfaces (nova, neutron, swift, etc)
- Cloud Management Tools (Rightscale, Enstratius, etc)
- GUI tools (Dashboard, Cyberduck, iPhone client, etc)
Internet
OpenStack Object API
OpenStack Image API
HTTP(S)
OpenStack Dashboard
OpenStack Compute APi
VNC/VMRC
OpenStack Block Storage API
OpenStack Networking API
HTTP(S)
Horizon
OpenStack Image API
OpenStack Compute API/ Admin API
Amazon Web Services EC2 API
OpenStack Block Storage API
OpenStack Networking API
swift-proxy
memcached
account
container
object
account database
container database
object database
OpenStack Object Store
glance-api
glance-registry
glance database
OpenStack Image Service
OpenStack Image API
nova-api (OS, EC2, Admin)
nova-console
nova-compute
nova-cert/ objectstore
hypervisor
nova database
Queue
AMQP
AMQP
nova-conductor
nova-consoleauth
nova-scheduler
OpenStack Compute
cinder-api
cinder-volume
volume provider
nova database
cinder-scheduler
OpenStack Block Storage
neutron-server
neutron agents
Queue
neutron plugin(s)
network provider
neutron database
OpenStack Networking
BY SA
http://www.solinea.com
OpenStack Identity API
keystone (service and admin APIs)
OpenStack Identity Service
token backend
catalog backend
policy backend
identity backend
OpenStack Identity API
OpenStack Identity API
OpenStack Identity API

Neutron SRv6 Plugin - networking-sr

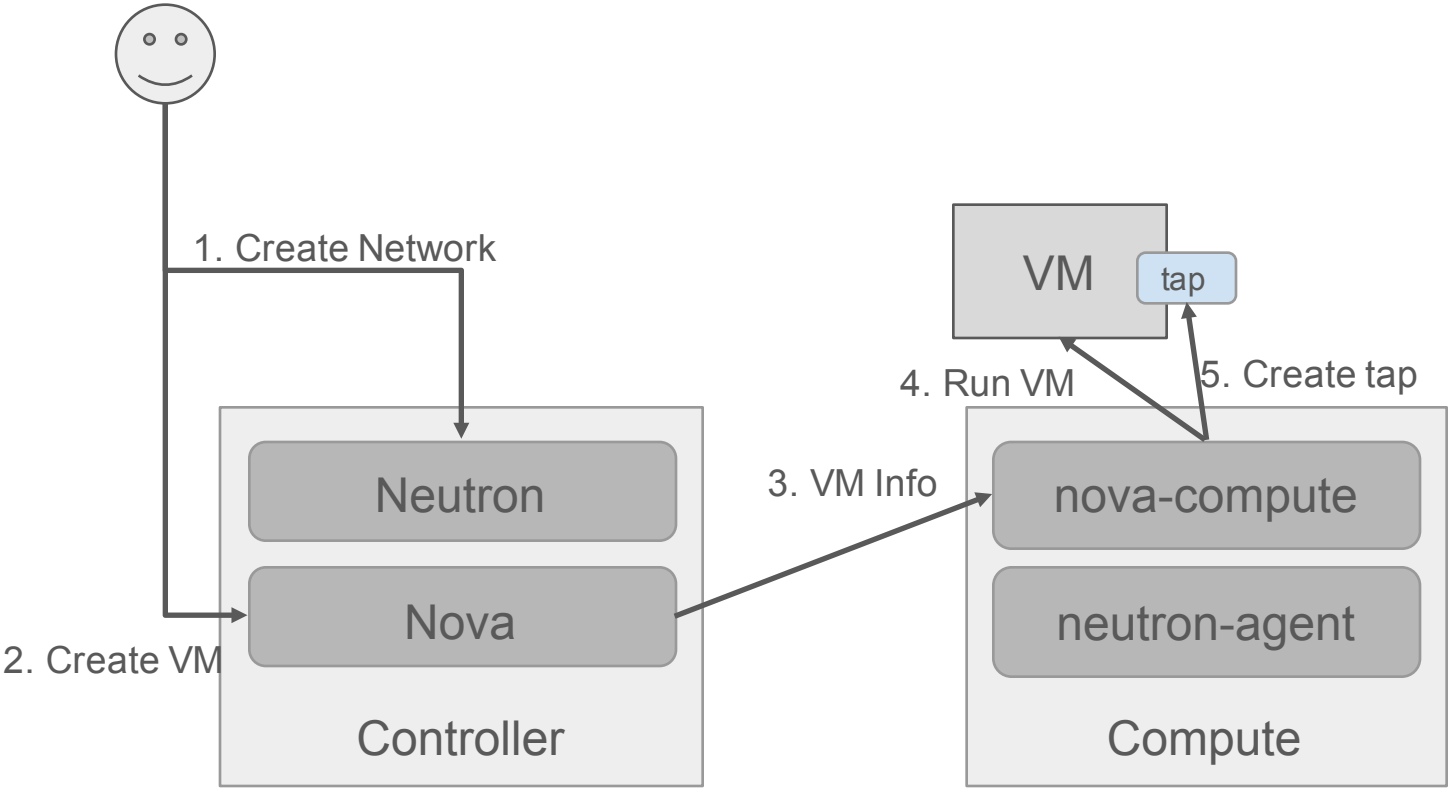
- ML2 mechanism/type driver and agent
- Gateway agent on network nodes
- Service plugin for new API to add SRv6 encap rule



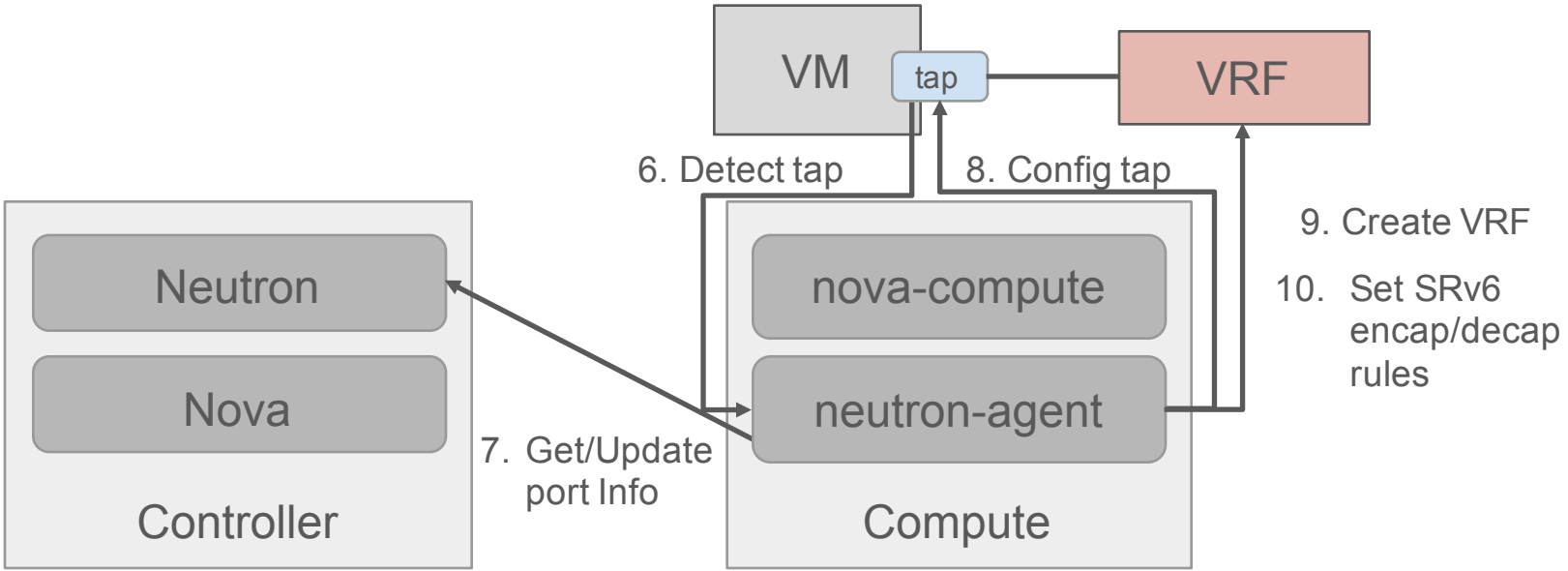
SRv6 Data Center Network Control Plane

ML2 mechanism/type driver and agent

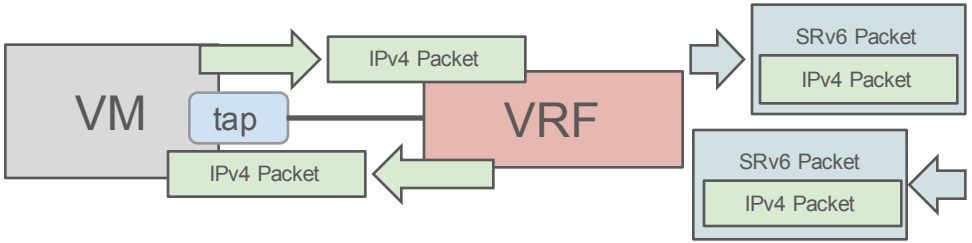
Nova, Neutron Behavior - VM create



Nova, Neutron Behavior - Network configuration



Packets for VM encap/decap on VRF



Neutron

Nova

Controller

nova-compute

neutron-agent

Compute

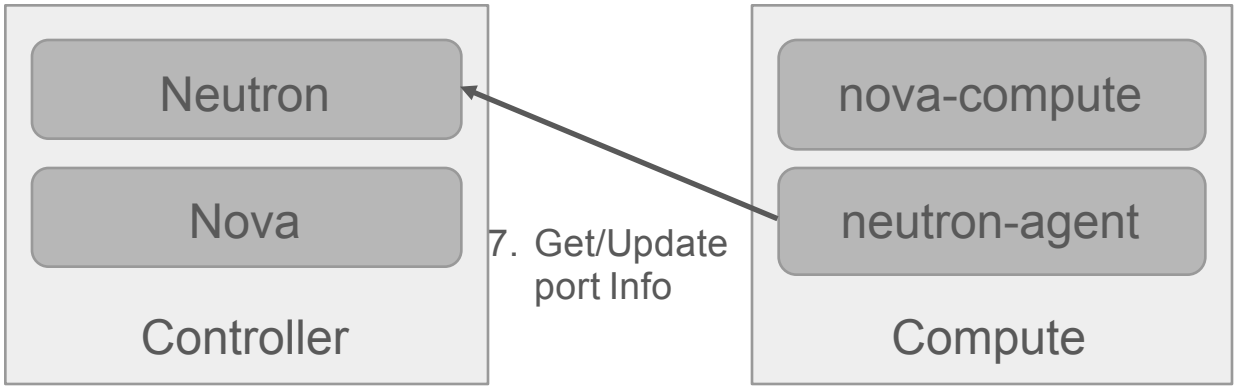
How does sr-agent get VRF info?

Virtual Machine Configuration

1. Create network
2. Create VM
3. Notify VM info
4. Run VM
5. Create tap

Network Configuration

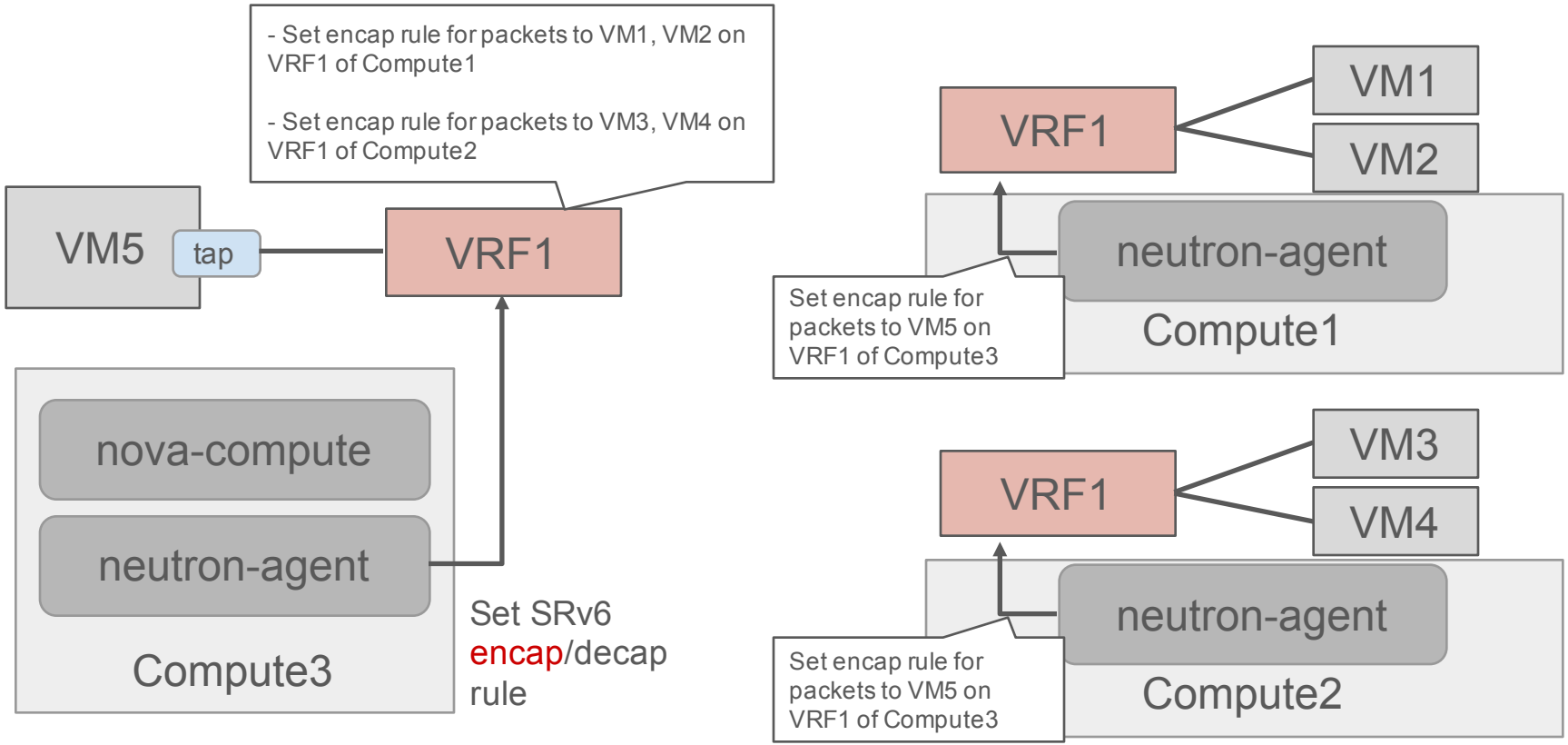
6. Detect tap
7. Update/Get port info
8. Config tap
9. Create VRF
10. Set SRv6 encap/decap rules



VRF info in Port binding:profile

```
{
  "port":{
    "binding:profile":{
      "segment_node_id": "2400:dcc0::a7a:4d8e", # Locator(Hypervisor address) where VM with the port running
      "vrf": "vrf644606a29039", # VRF IF name for the port. The name is combined by "vrf" + tenant_id + network_id
      "vrf_cidr": "169.254.1.0/24", # IP CIDR of VRF for the port
      "vrf_ip": "169.254.1.44" # IP Address of VRF for the port
    }
  }
}
```

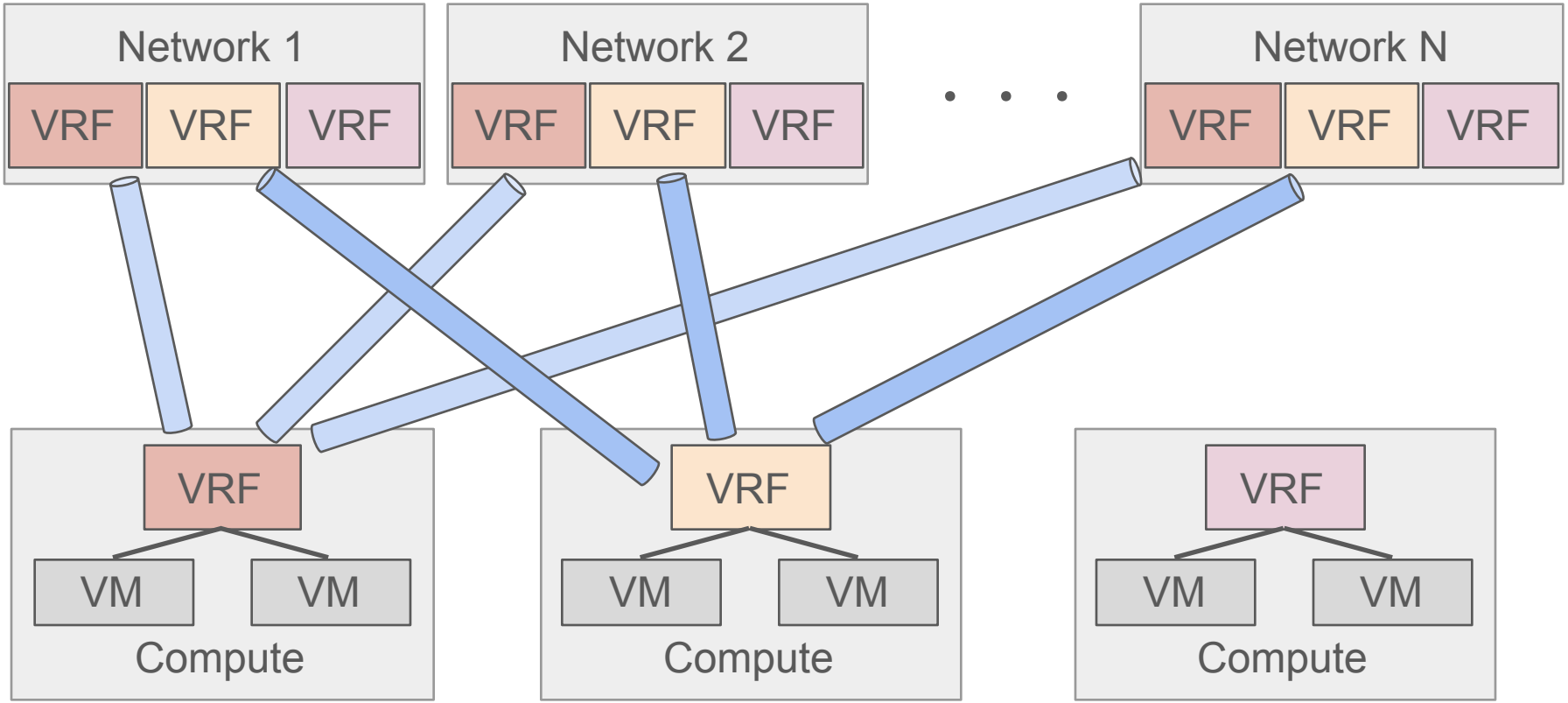
Set encap rule from Port info of each VM



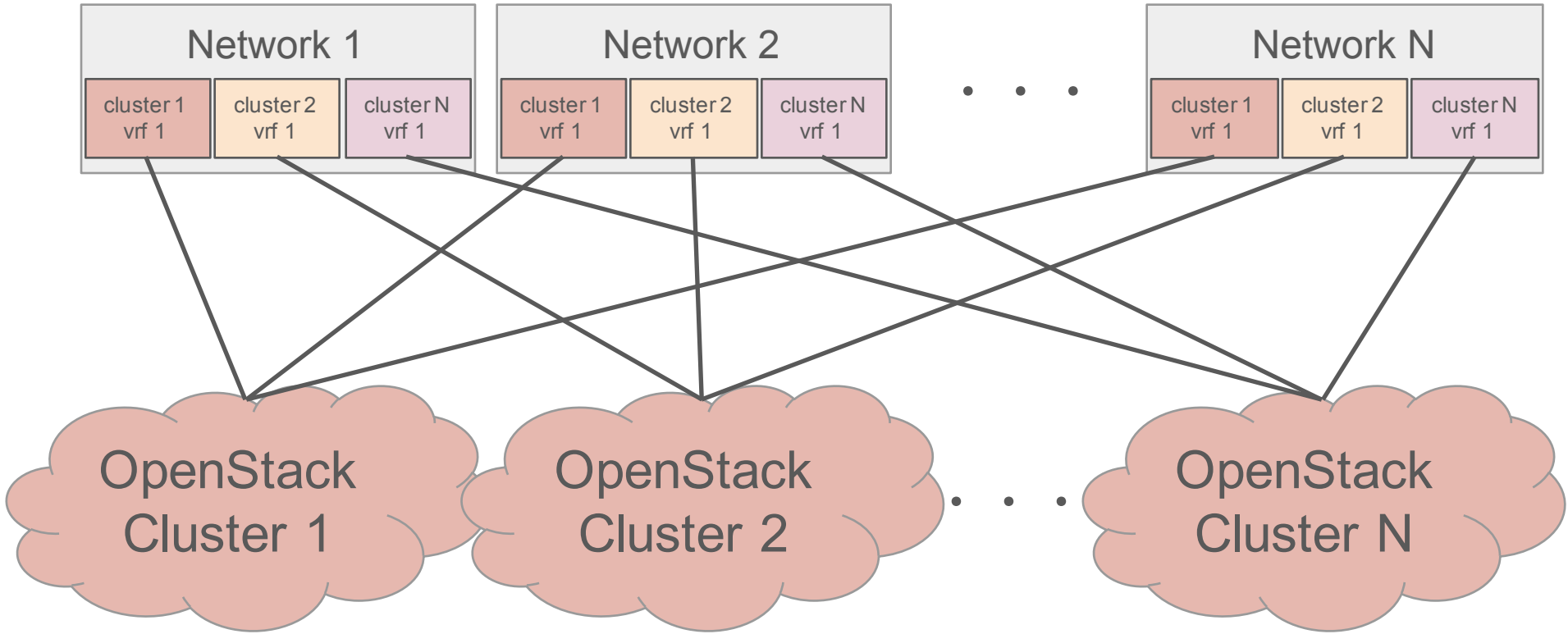
SRv6 Data Center Network Control Plane

Gateway agent on network nodes

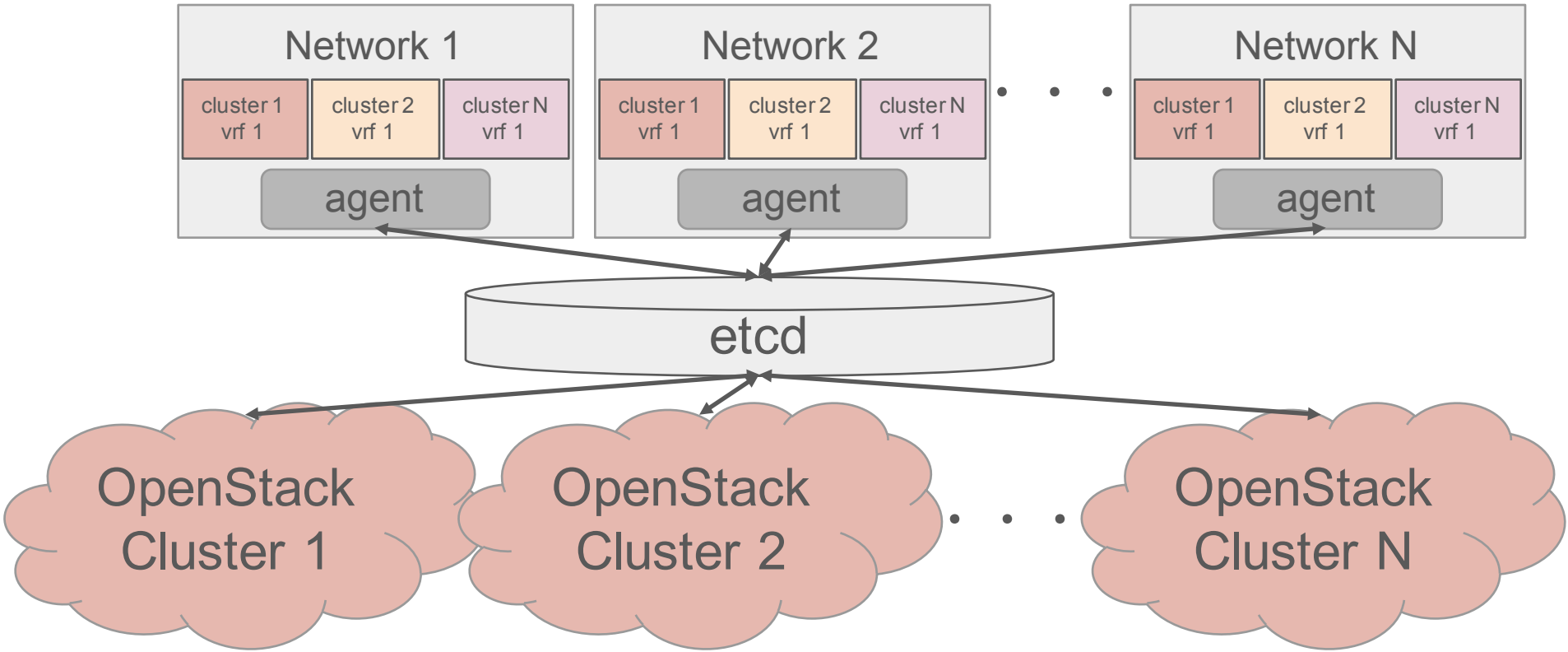
Network Node Requirements: Scale



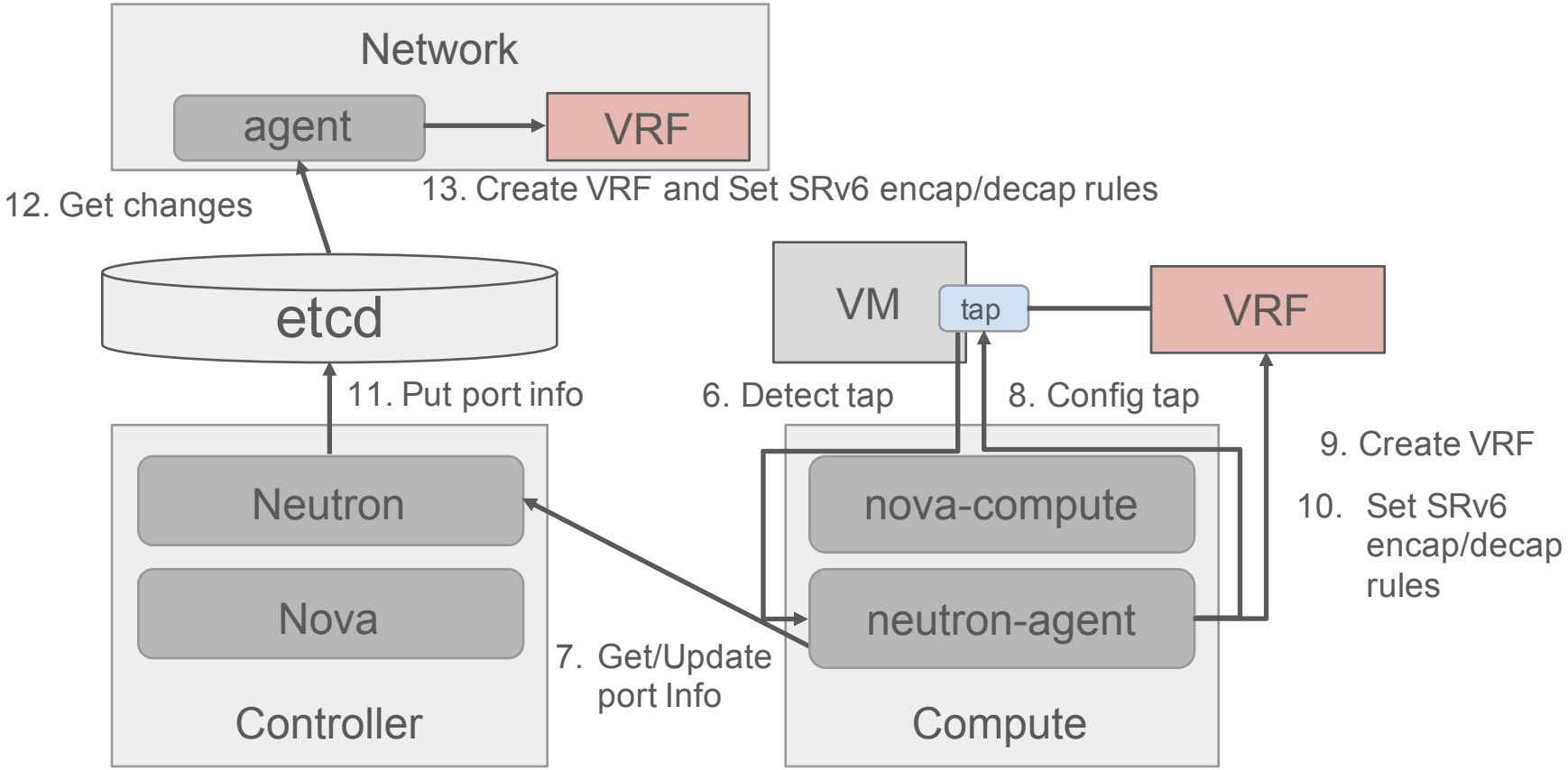
Network Node Requirements: Multi clusters



Etcd + Agent Model



Notify New Encap/Decap Rule via Etcd



SRv6 Data Center Network Control Plane

Service plugin for new API to add SRv6 encap rule

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.

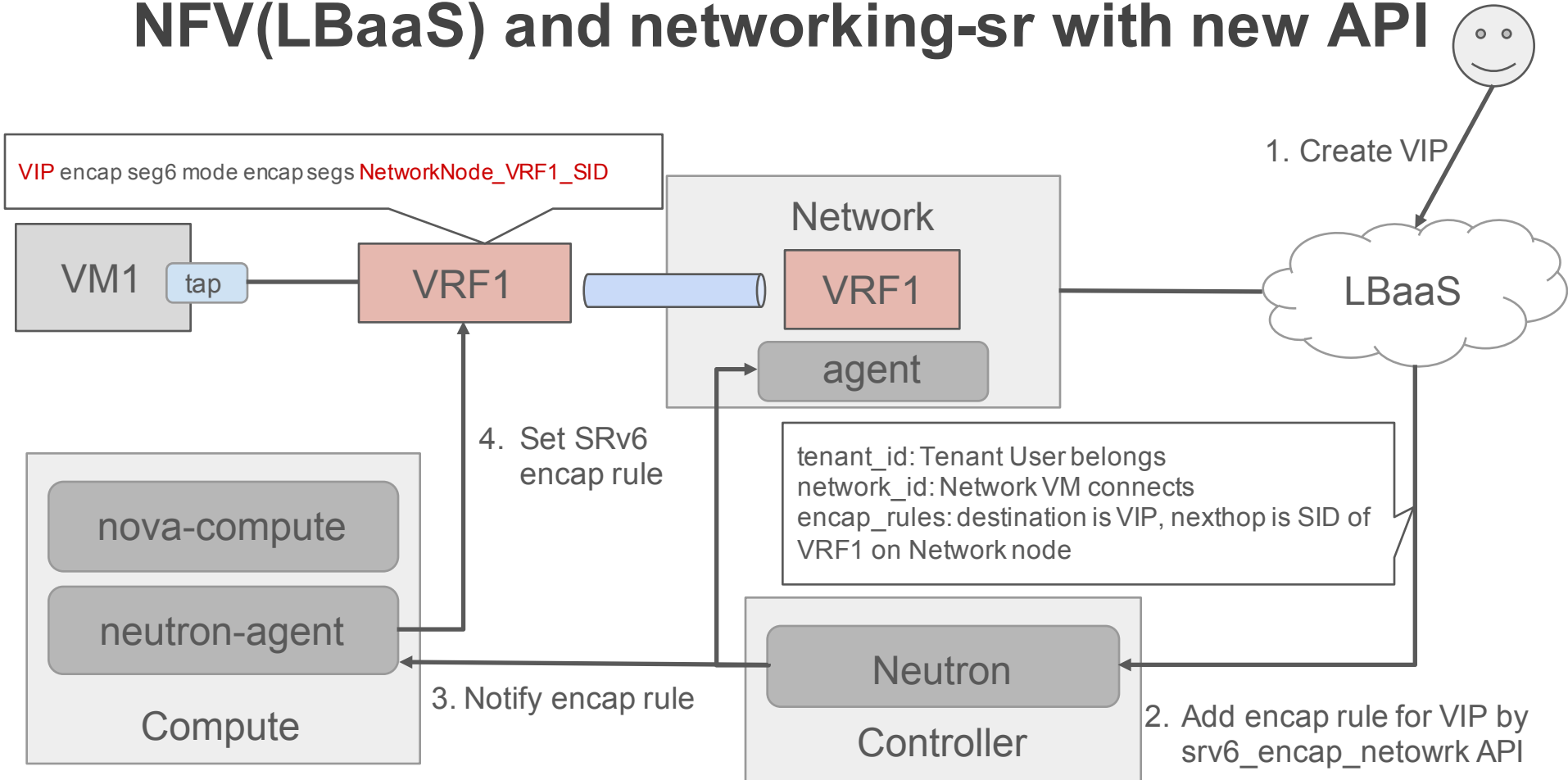
GET	/v2.0/srv6_encap_networks	**List srv6 encap networks**
POST	/v2.0/srv6_encap_networks	**Create srv6 encap network**
GET	/v2.0/srv6_encap_networks/`{srv6_encap_network_id}`	**Show srv6 encap network**
PUT	/v2.0/srv6_encap_networks/`{srv6_encap_network_id}`	**Update srv6 encap network**
DELETE	/v2.0/srv6_encap_networks/`{srv6_encap_network_id}`	**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 resource

- id: Identifier for resource
- tenant_id/project_id: Identifier for project/tenant of resource
- network_id: Identifier of network which resource is assigned
- encap_rules: SRv6 encap rule list
  - destination: IPv4 address for specific destination of packet
  - nexthop: SID packets should be encaped

NFV(LBaaS) and networking-sr with new API



Summary

- SRv6 network for data center use case
  - Multi tenant networks
- Data plane architecture
  - SRv6 Encap/Decap support on Hypervisors and Network nodes
  - End.DX4 + Routing to VRF (Kernel doesn’t have End.DT4)
- Control plane architecture
  - OpenStack Neutron SRv6 plugin networking-sr
  - Gateway agent with etcd for large scale
  - New API to add SRv6 encap rule