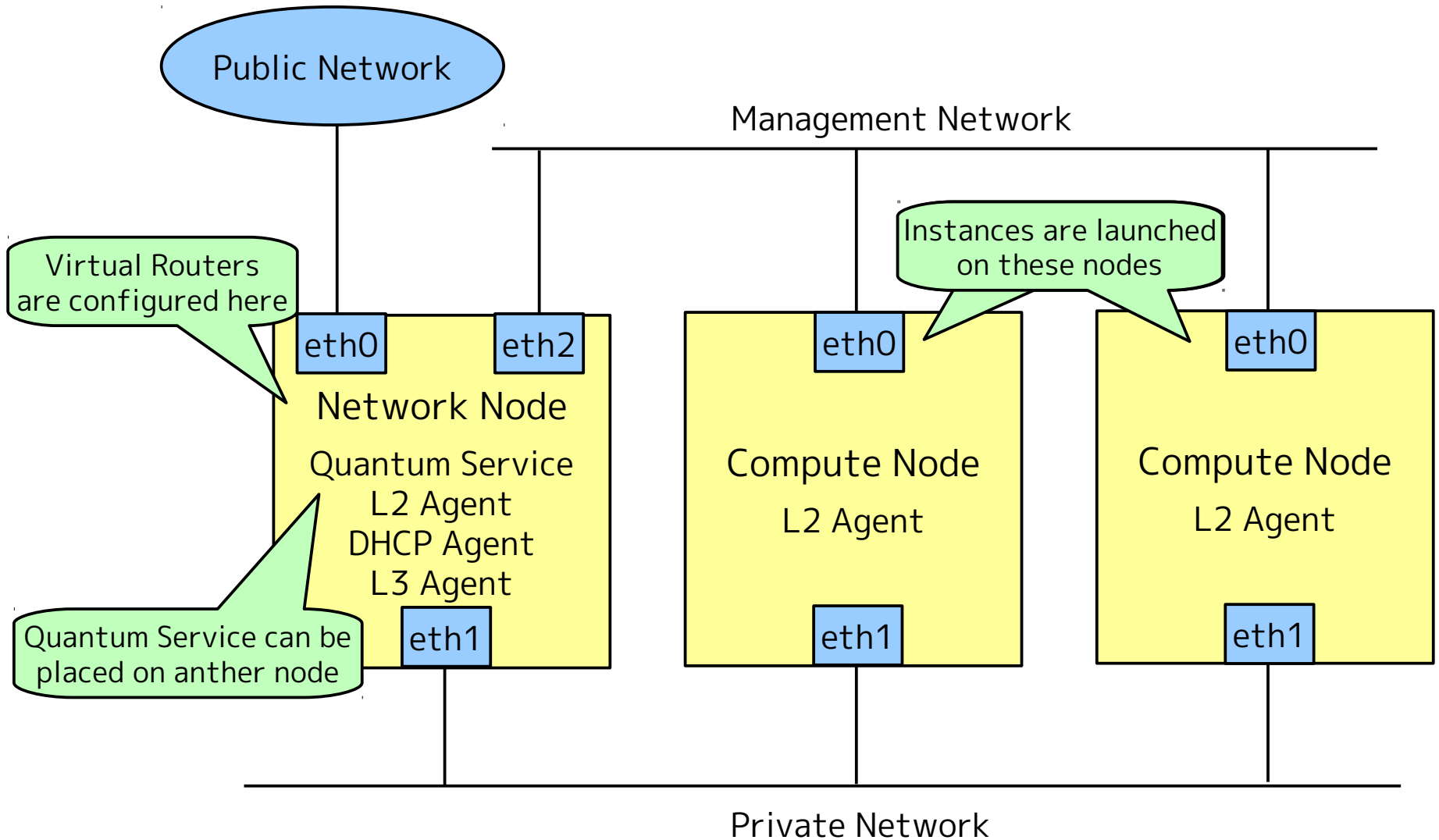


# How Quantum Configures Virtual Networks Under the Hood?

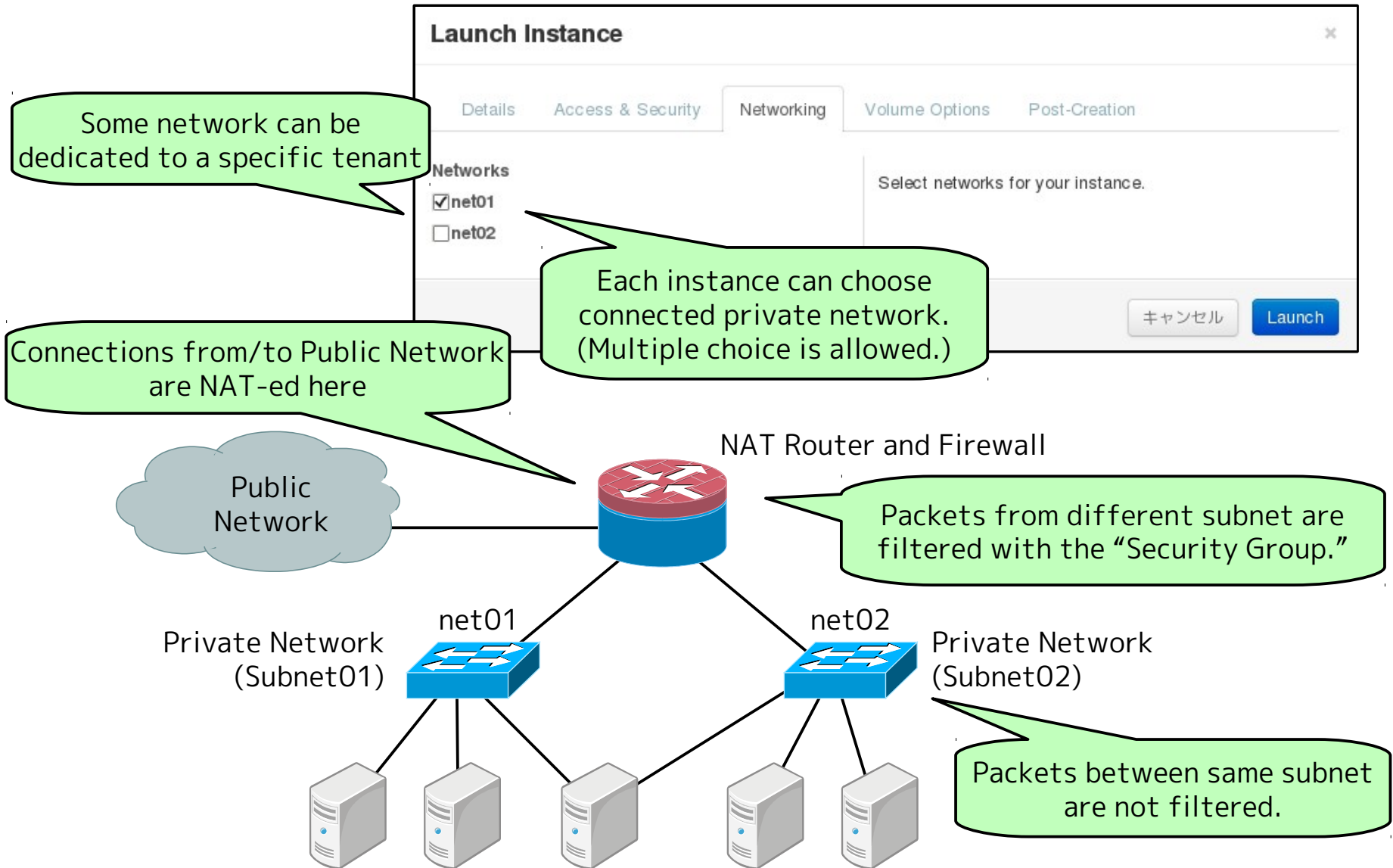


# Physical and Logical Configuration

# Physical Network Connection and Agent Placement

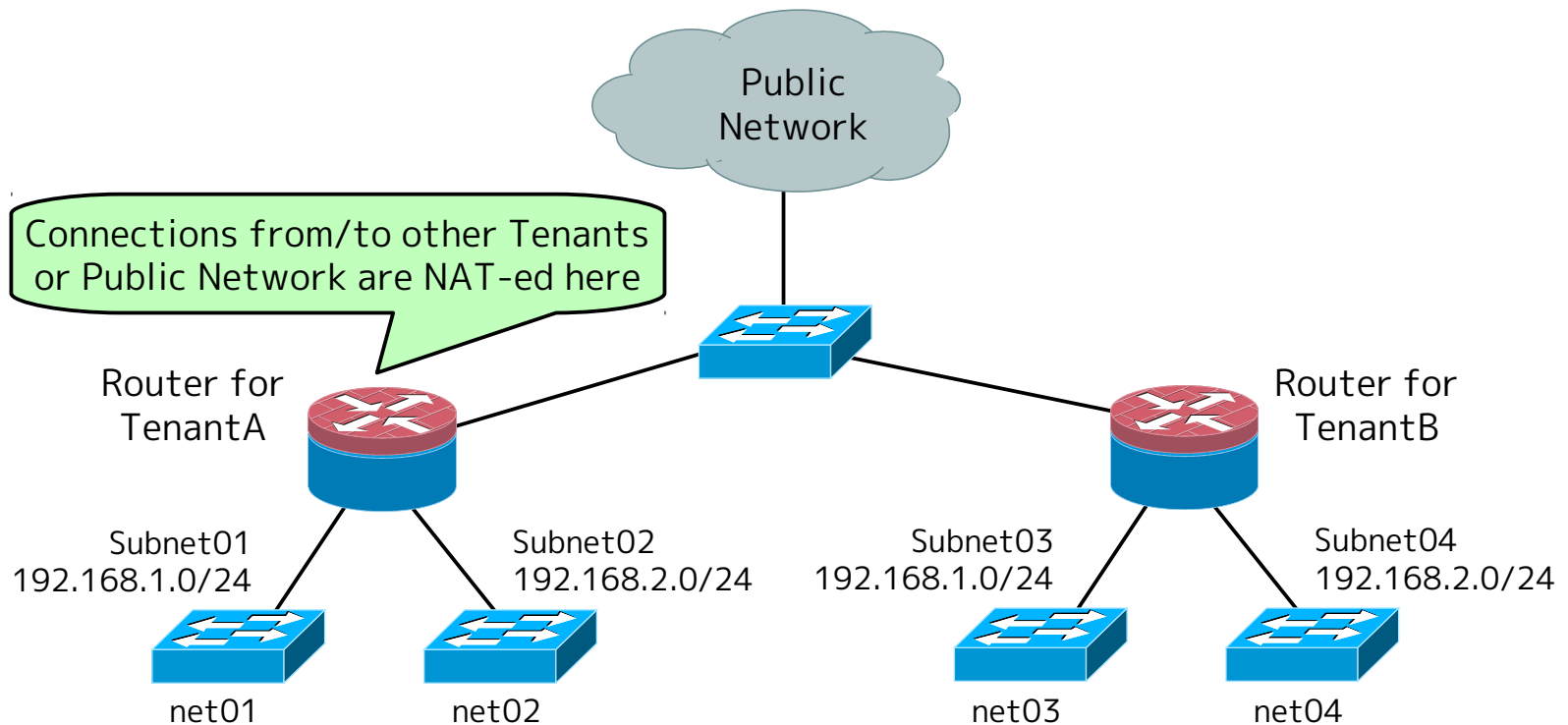


# Logical Configuration of the Virtual Network - Case1



## Logical Configuration of the Virtual Network - Case2

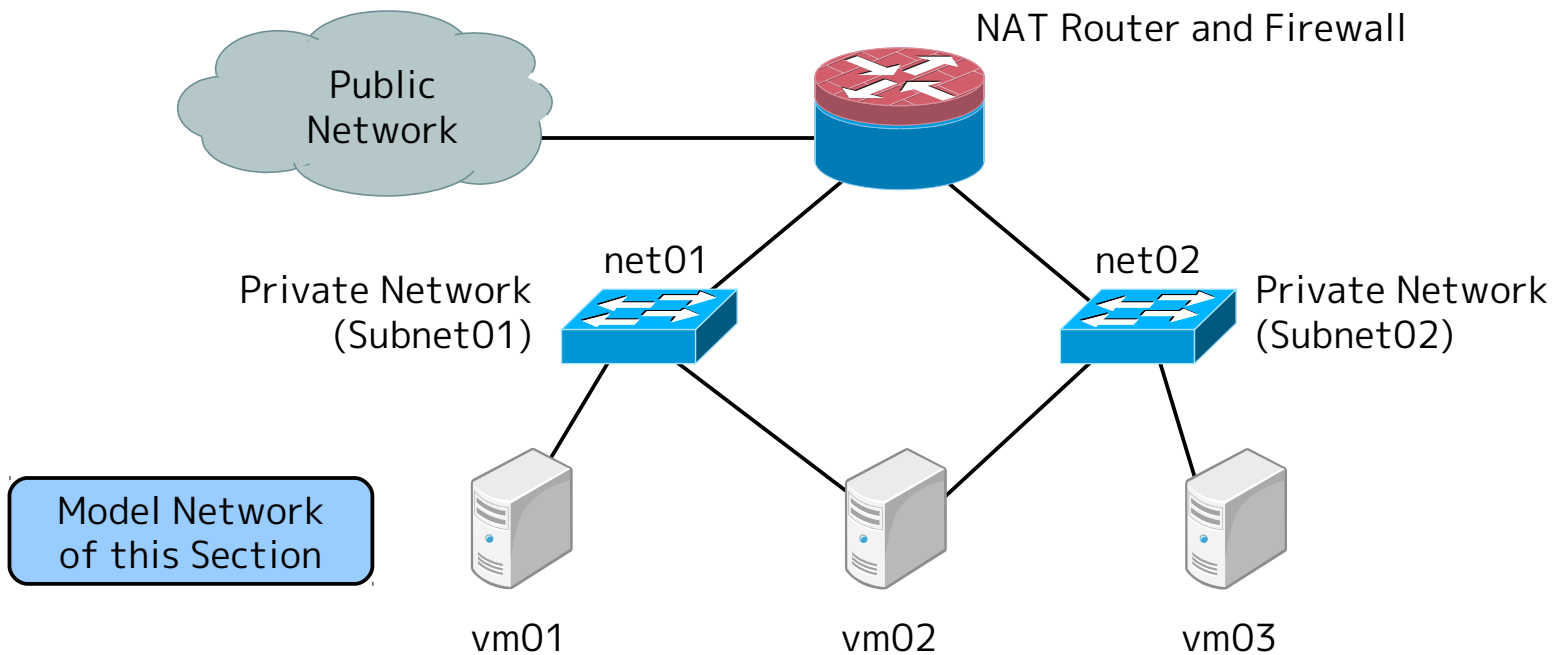
- By assigning a dedicated router for each tenant, each tenant can create its own private networks freely, even overlapping subnets can be used by multiple tenants.



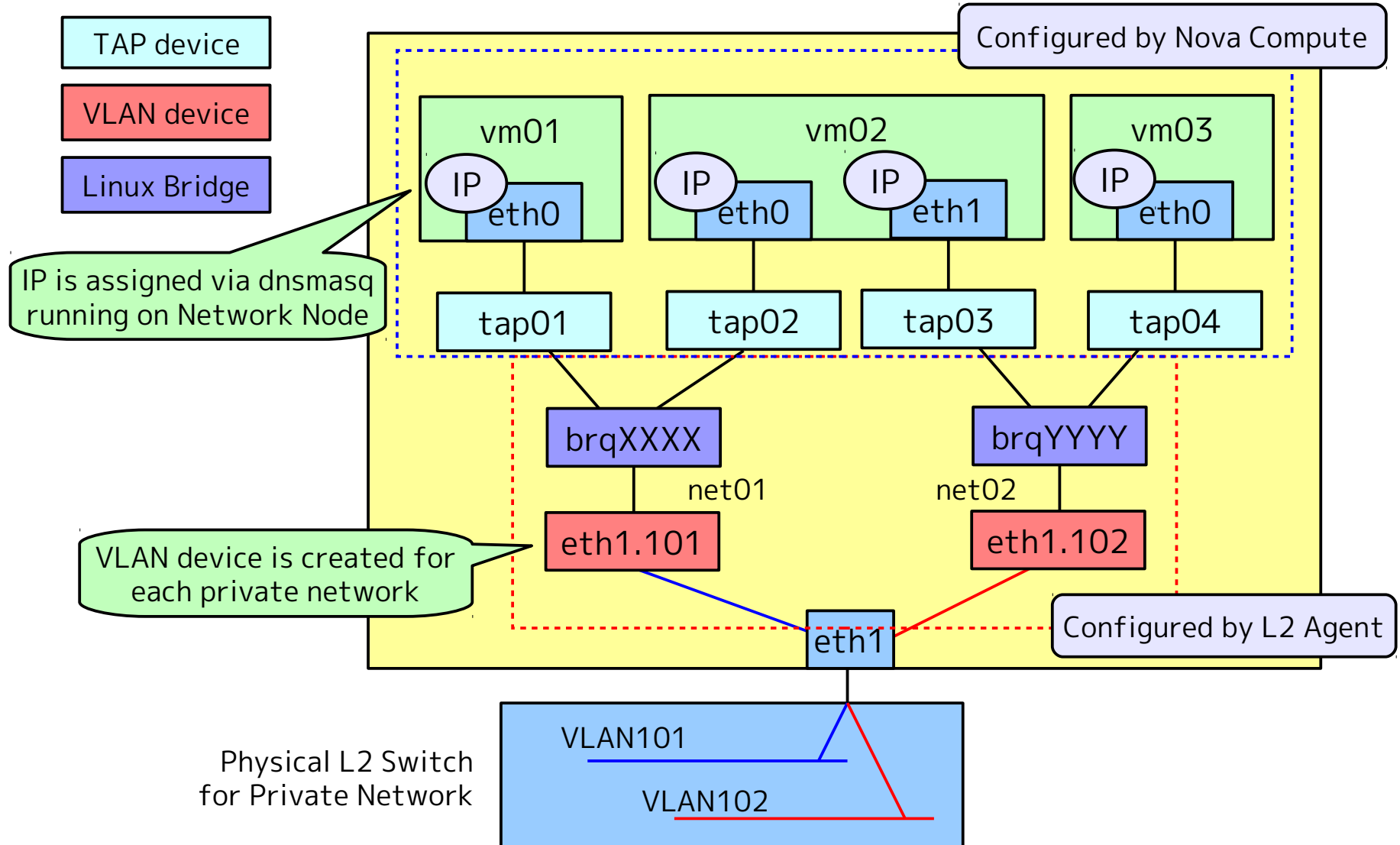
## Notes on the Configuration

- There are various plugin's for L2 Agent. In this document, we describe “LinuxBridge plugin” and “Open vSwitch plugin” which provides basic bridge functionality using Linux bridge / Open vSwitch.
- The single Network Node could be a potential bottle neck and a single point of failure. There is a lot of design discussion about it in the upstream.
- Currently dnsmasq is used by DHCP agent as in the following charts, but different DHCP options are possible by design. Other options may be added in the future.
- Network Namespace support is *\_not\_* mandatory for Case1 configuration. If you want to use multiple routers (and optionally the overlapping IP address feature), however, you have to use Case2 design and Network Namespace support is required there.
- It depends on Linux distributions whether you can use the Network Namespace feature. For example, RHEL6.3/6.4 doesn't support it. Fedora17/18 supports it.

# Network Components configured by LinuxBridge Plugin using VLAN separation - Case1 Configuration -

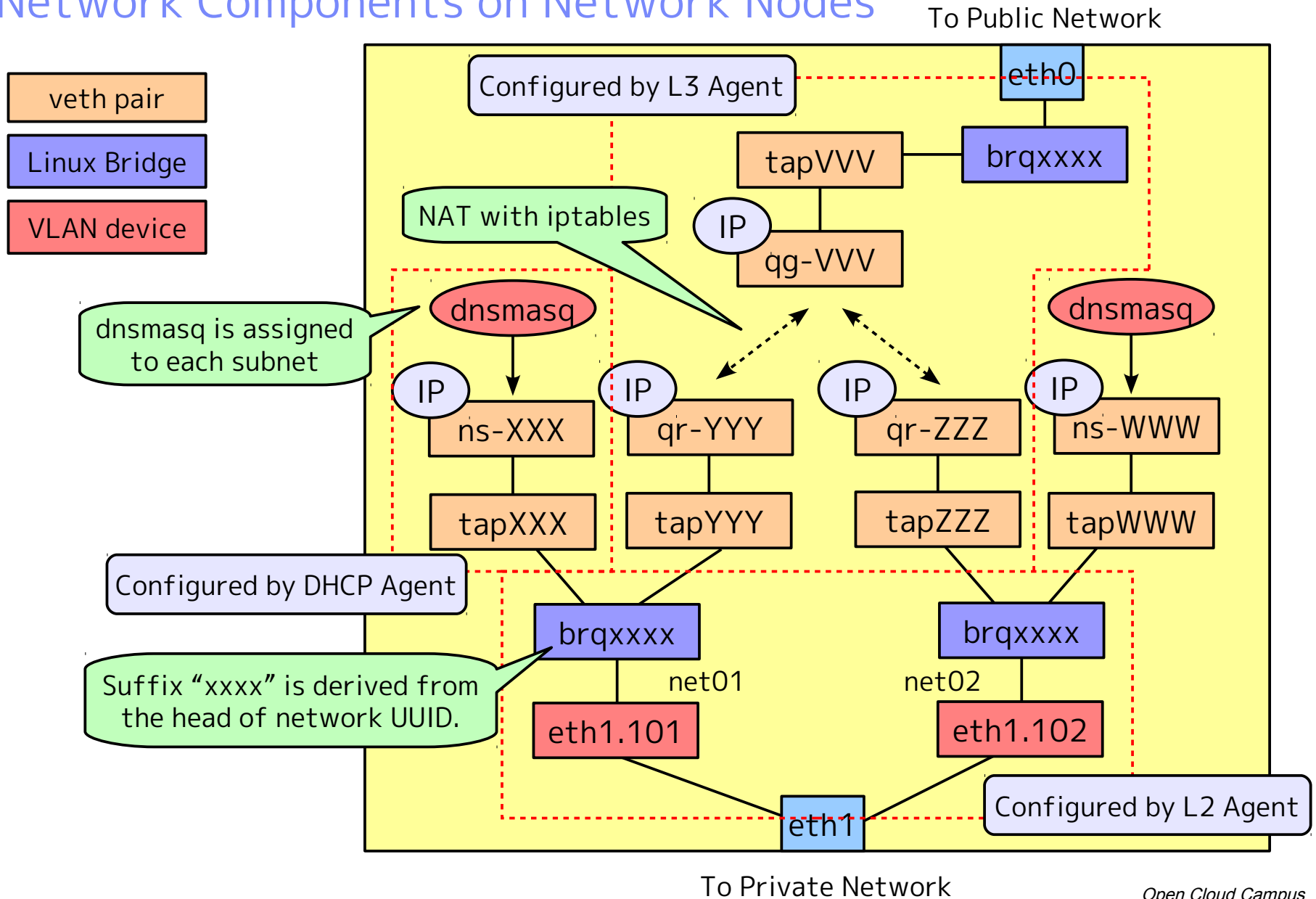


# Network Components on Compute Nodes

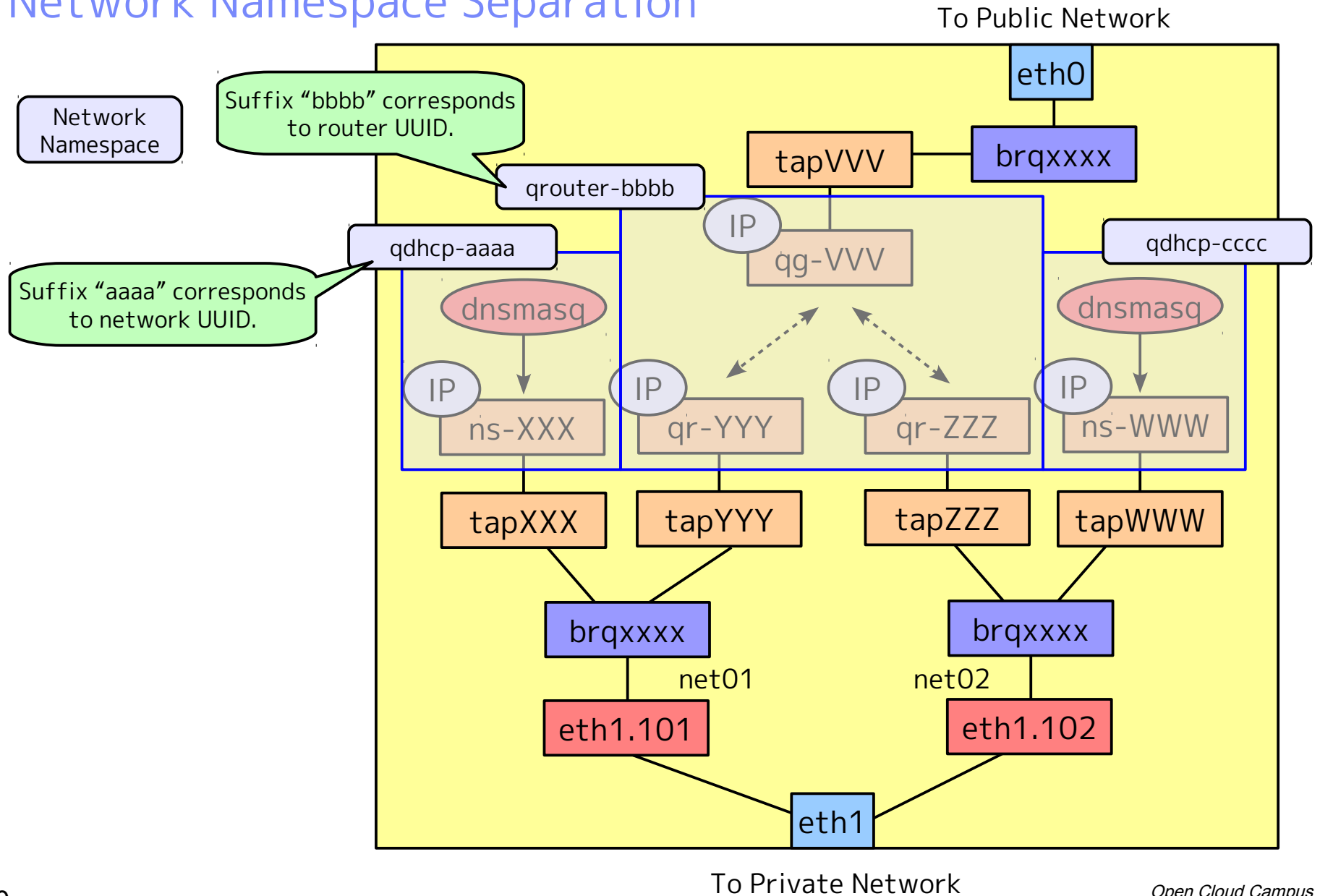




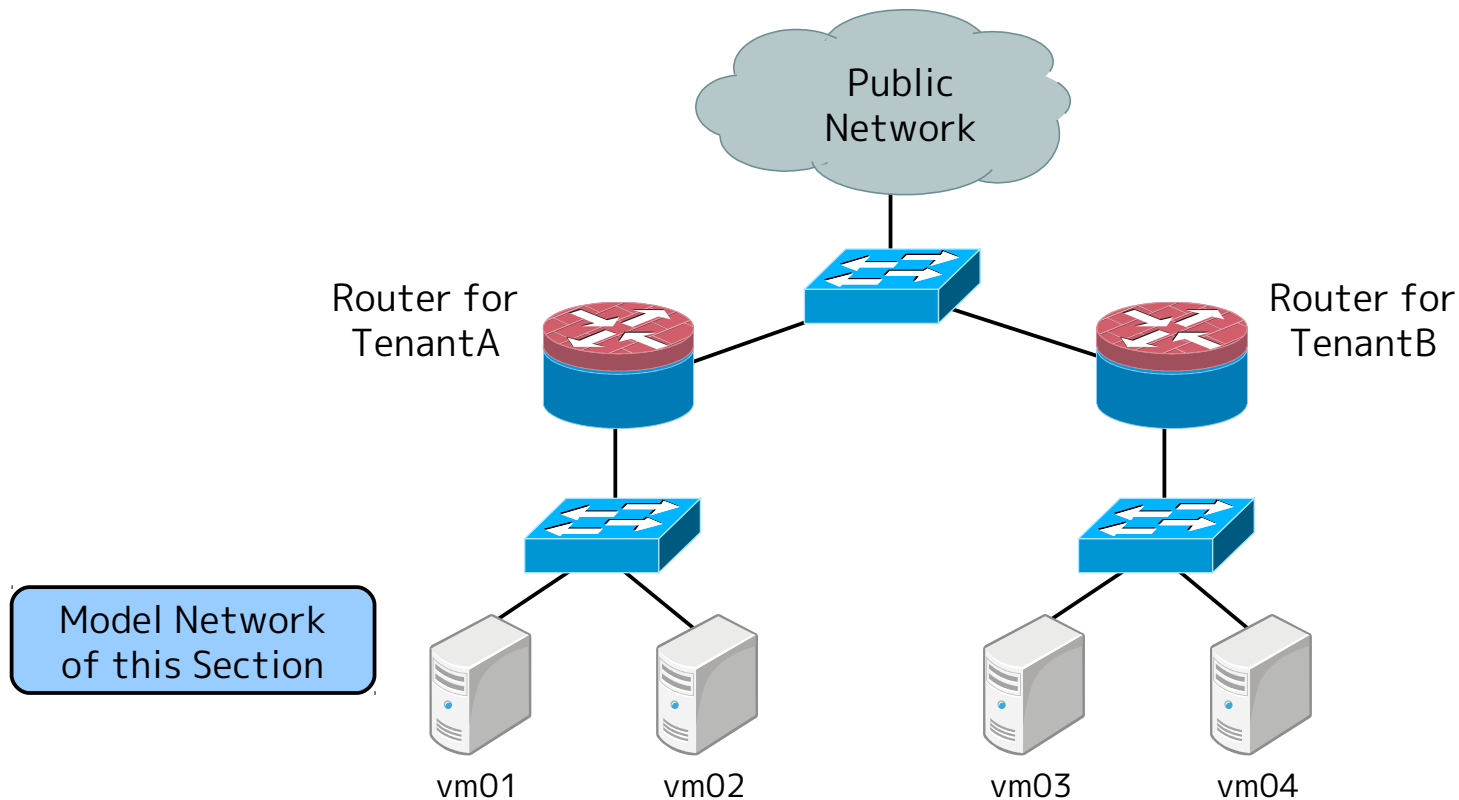
# Network Components on Network Nodes



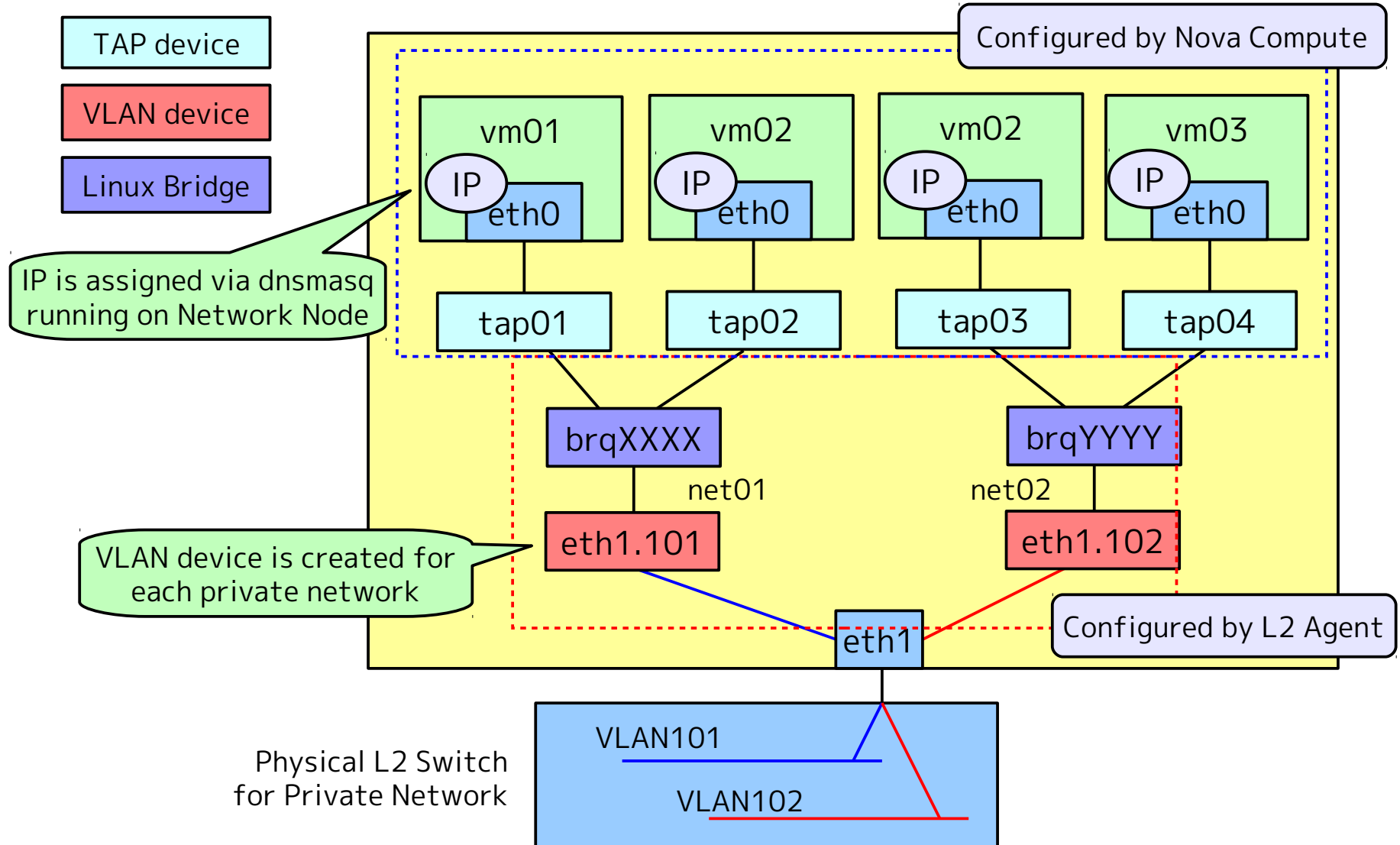
# Network Namespace Separation



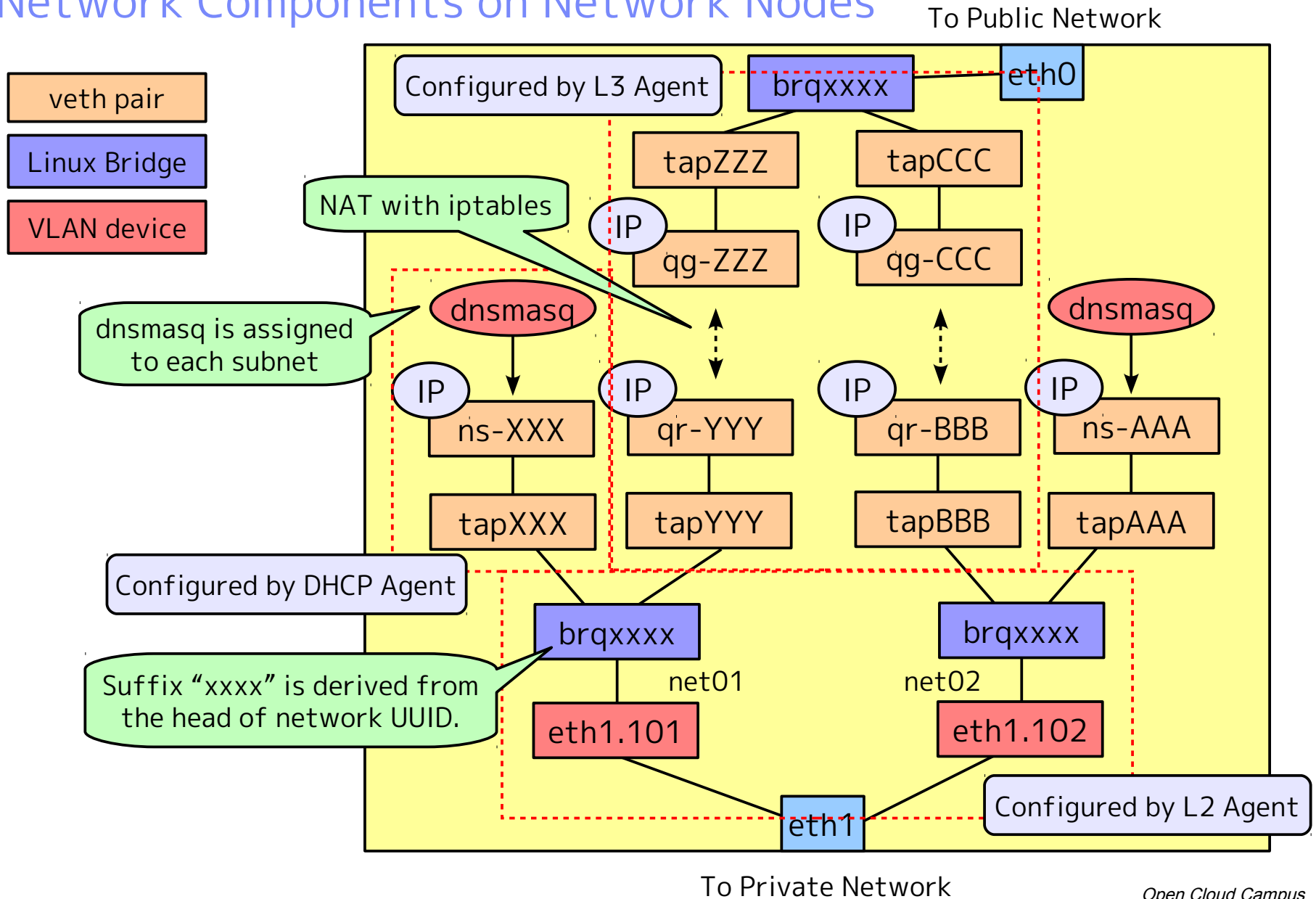
# Network Components configured by LinuxBridge Plugin Plugin using VLAN separation - Case2 Configuration -



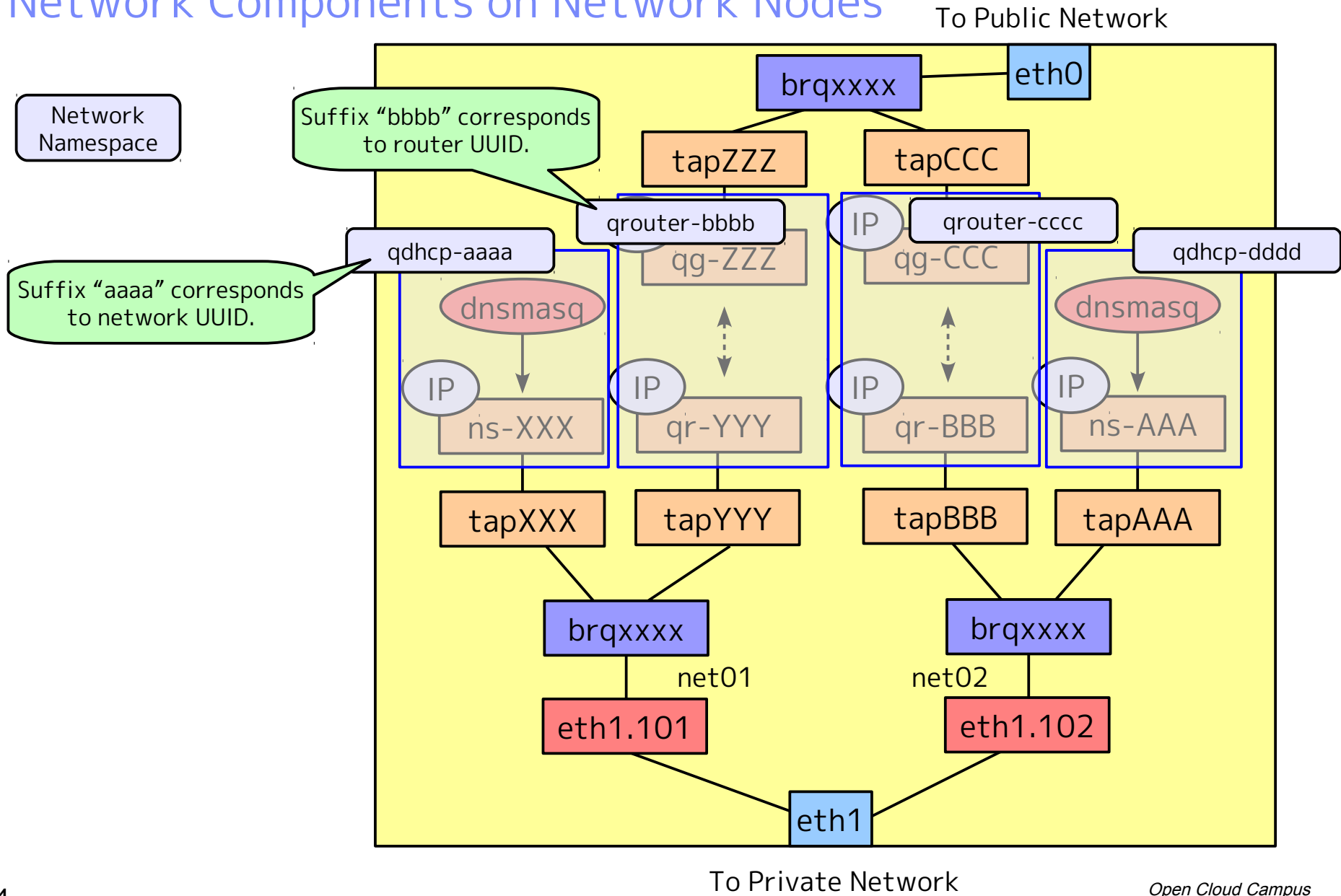
# Network Components on Compute Nodes



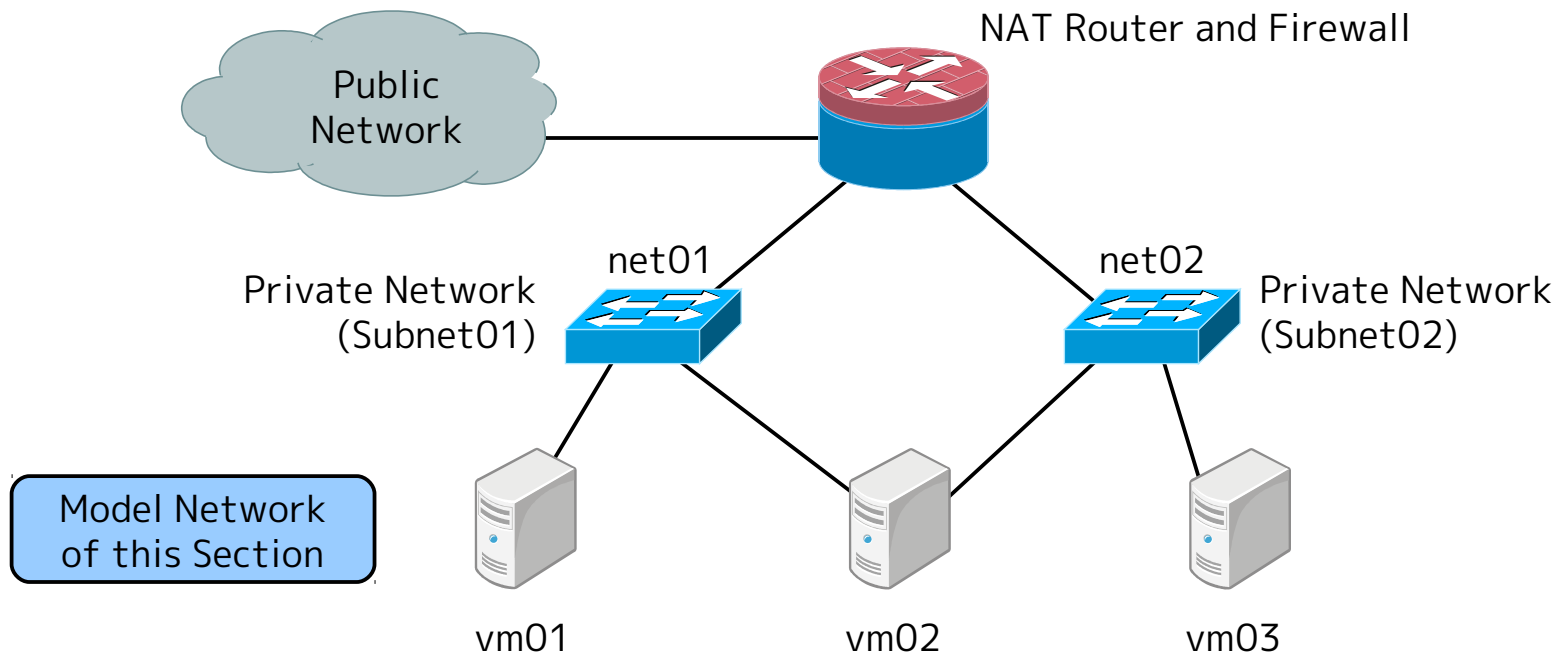
# Network Components on Network Nodes



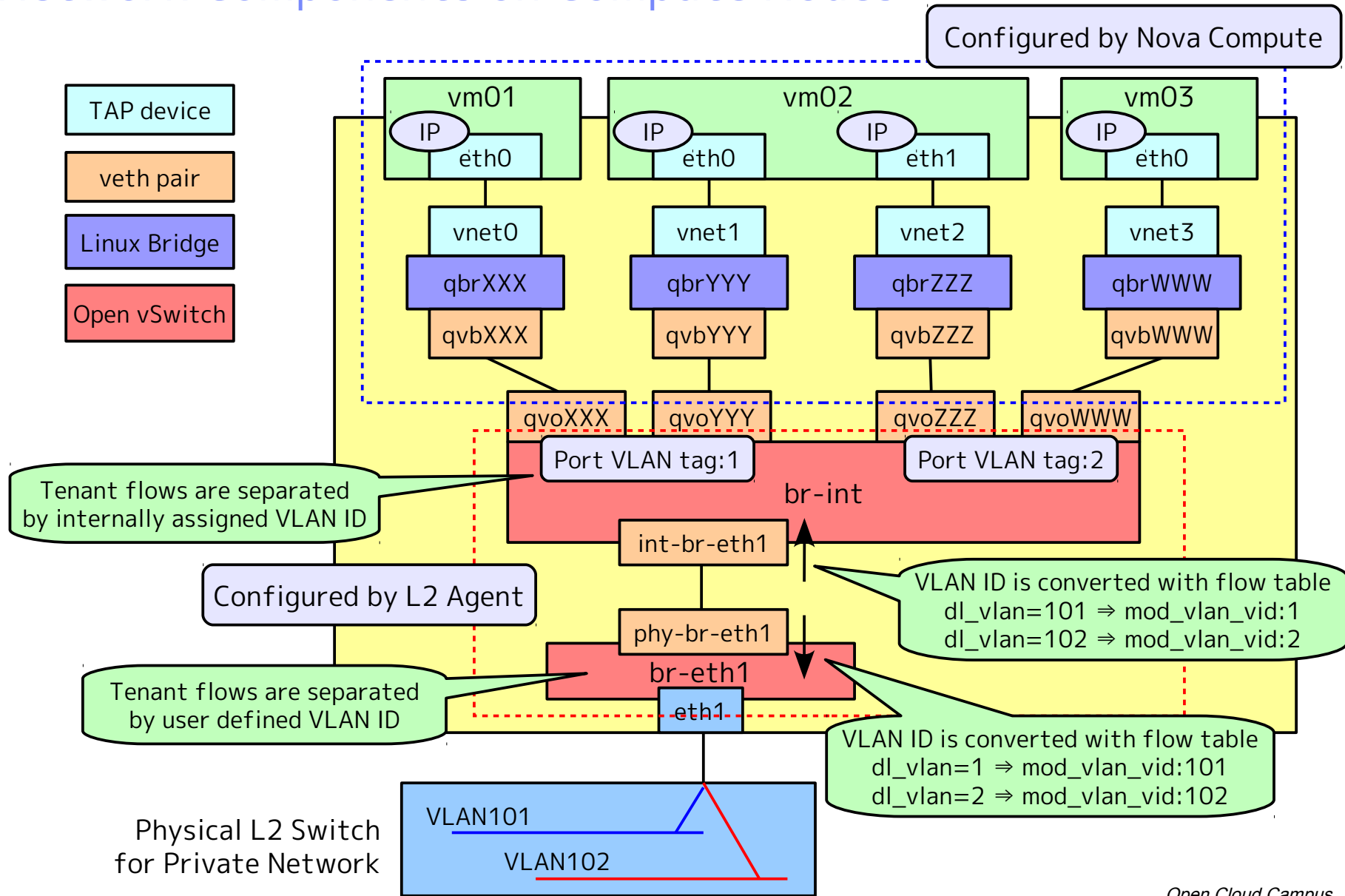
# Network Components on Network Nodes



# Network Components configured by Open vSwitch Plugin using VLAN separation - Case1 Configuration -

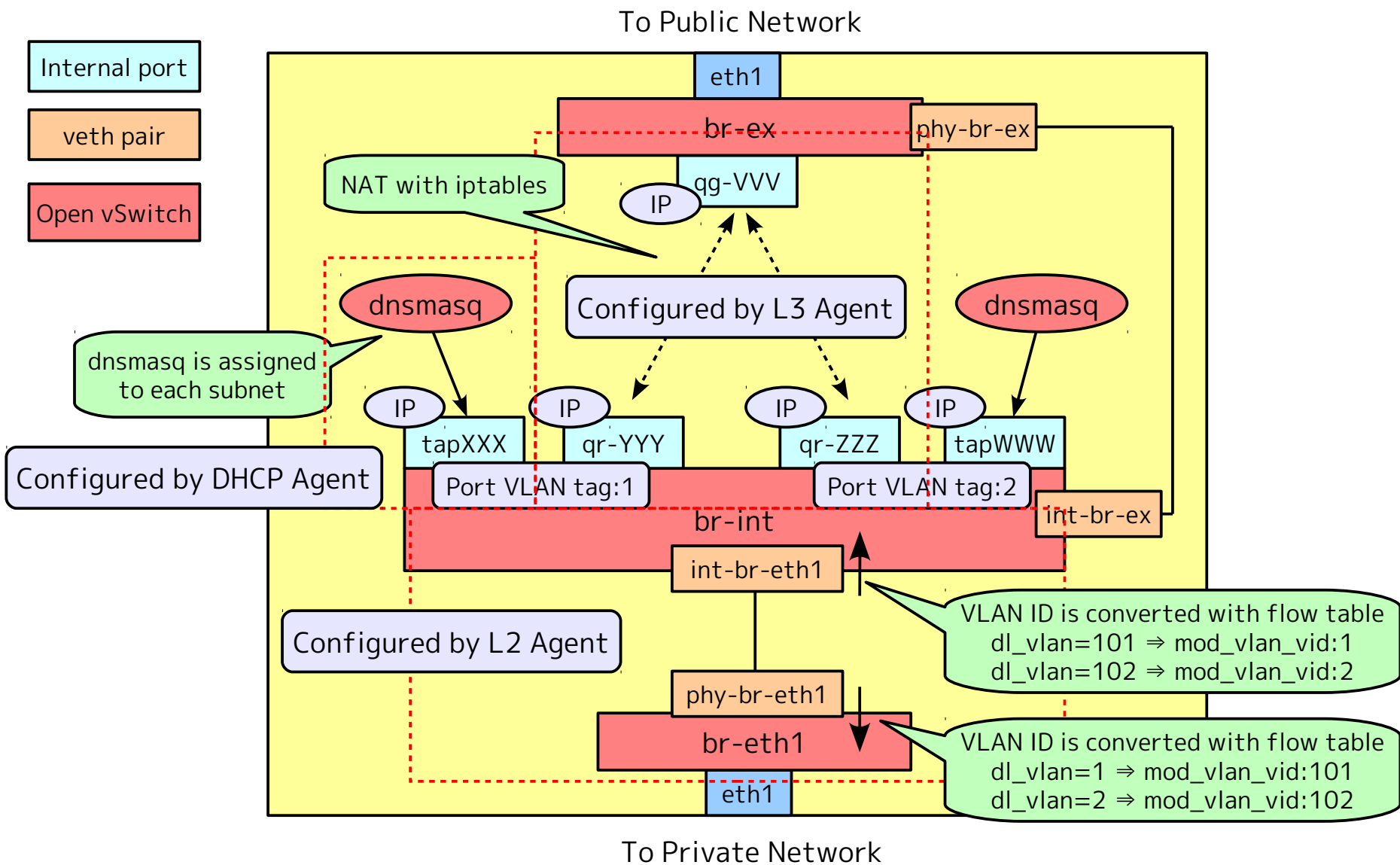


# Network Components on Compute Nodes

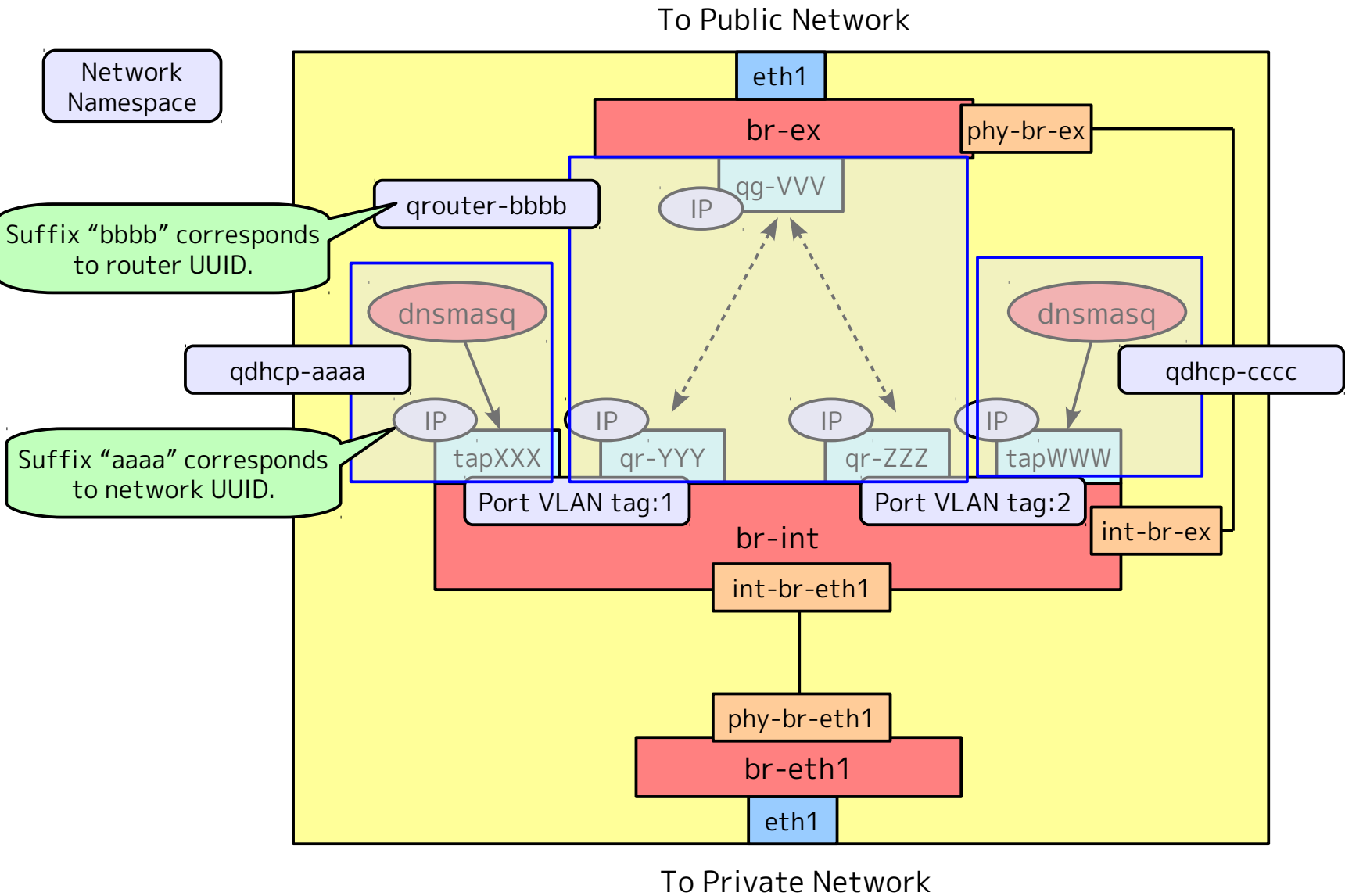




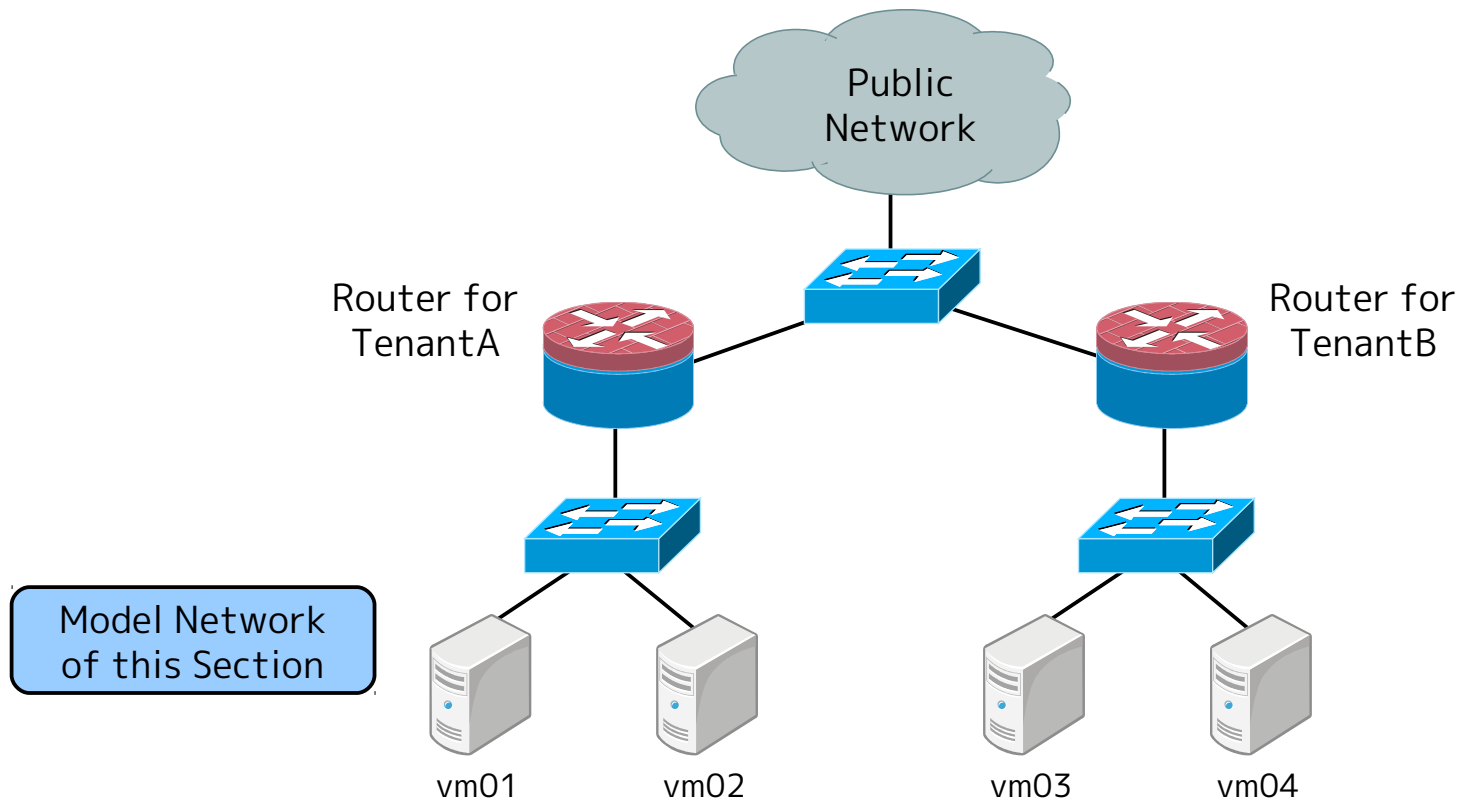
# Network Components on Network Node



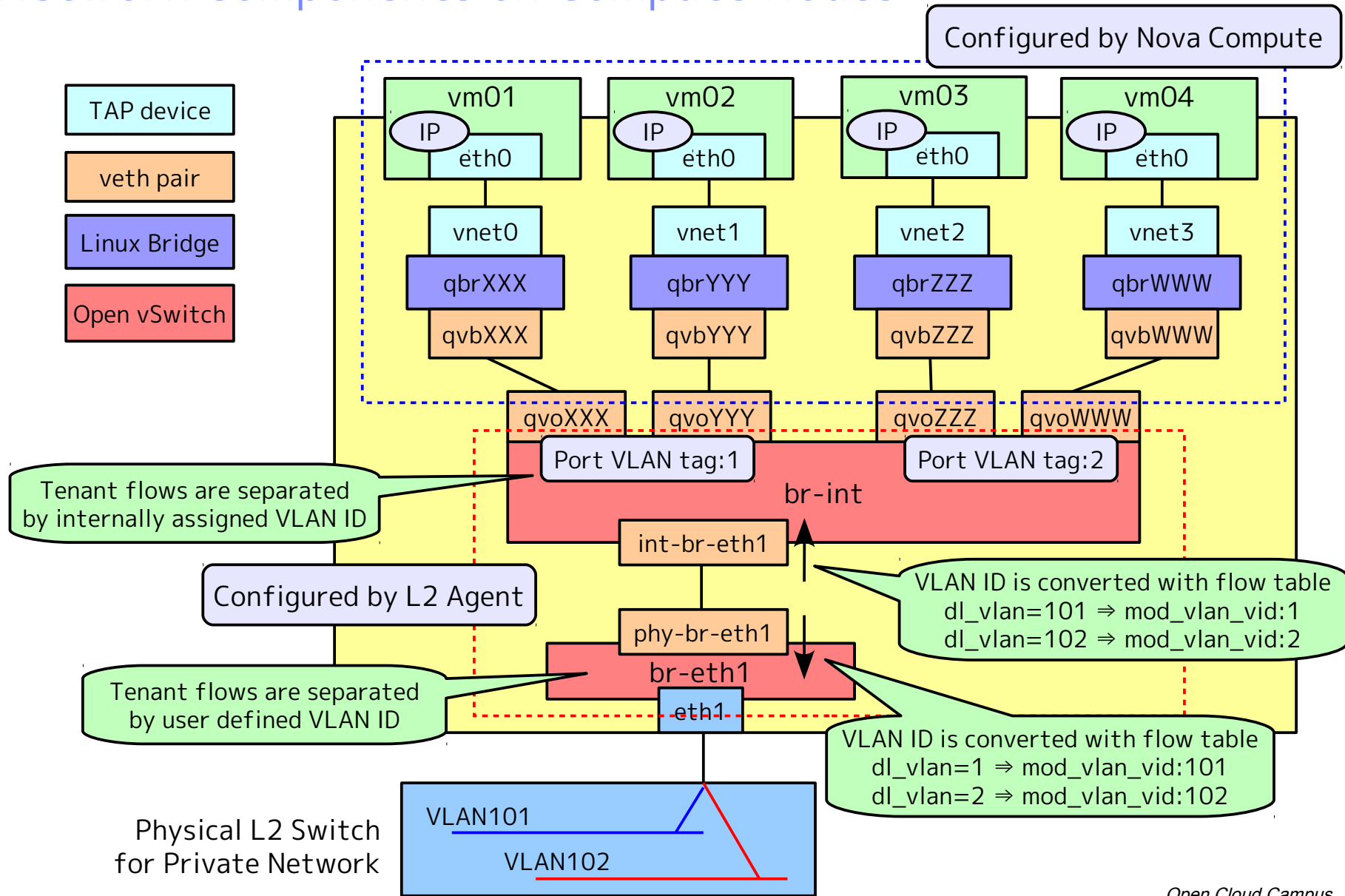
# Network Namespace Separation



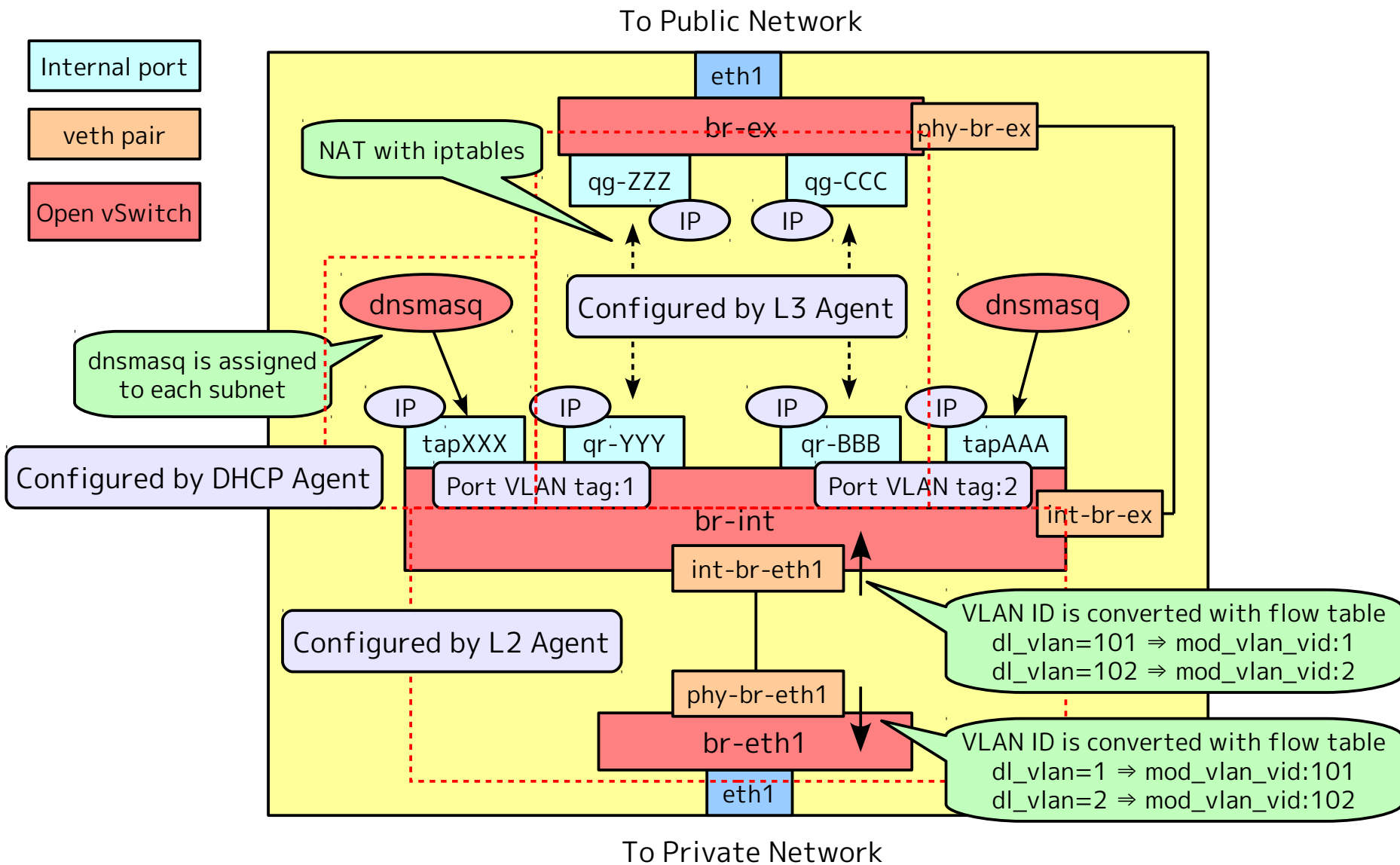
# Network Components configured by Open vSwitch Plugin using VLAN separation - Case2 Configuration -



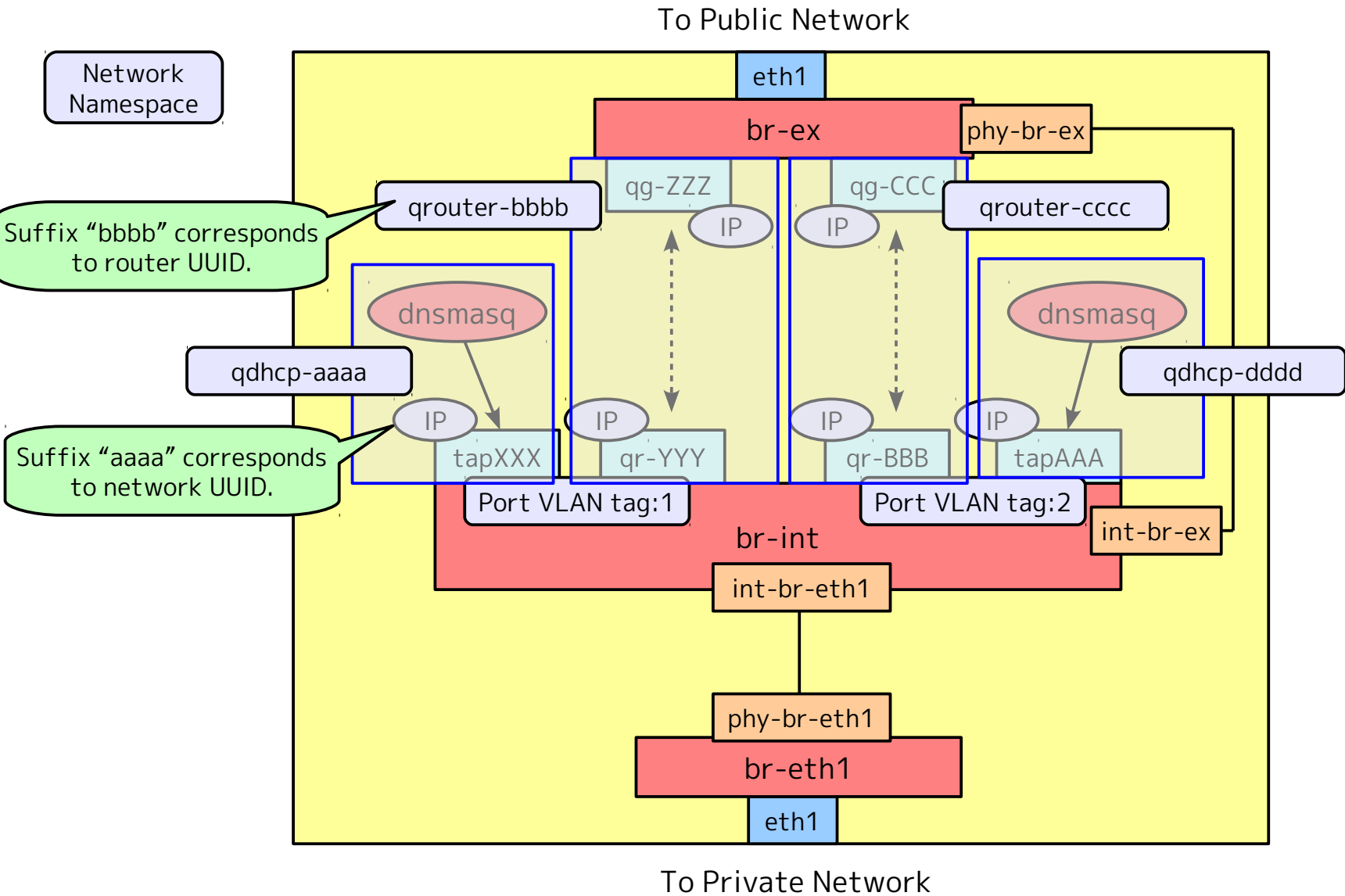
# Network Components on Compute Nodes



# Network Components on Network Node



# Network Namespace Separation



## Notes on the Configuration

- On Compute Nodes, the use of Linux Bridge between the integration switch (br-int) and VM tap devices may look redundant. But It's required for Nova's security group feature to work. They are configured with Nova's "LibvirtHybridOVSBridgeDriver".
- You can choose another driver if you don't need the security group functionality. Then, the configuration will be different.
- The security group feature will be integrated with Quantum in the future, and this would be more simplified.

# Example of Configuration Steps



## Example Configuration Steps for Case1 (1/2)

- Under the “service” tenant, create the shared router, define the public network, and set it as a default gateway of the router.

```
# tenant=$(keystone tenant-list | awk '/service/ {print $2}')
# quantum router-create router01
# quantum net-create --tenant-id $tenant public01 \
    --provider:network_type flat \
    --provider:physical_network physnet1 \
    --router:external=True
# quantum subnet-create --tenant-id $tenant --name public01_subnet01 \
    --gateway 10.64.201.254 public01 10.64.201.0/24 --enable_dhcp False
# quantum router-gateway-set router01 public01
```

- Under the user tenant “redhat”, create the private network “net01” and its subnet, and connect it to the router.

```
# tenant=$(keystone tenant-list|awk '/redhat/ {print $2}')
# quantum net-create --tenant-id $tenant net01 \
    --provider:network_type vlan \
    --provider:physical_network physnet2 \
    --provider:segmentation_id 101
# quantum subnet-create --tenant-id $tenant --name net01_subnet01 net01 192.168.101.0/24
# quantum router-interface-add router01 net01_subnet01
```

## Example Configuration Steps for Case1 (2/2)

- Another network “net02” can be added in the same way.

```
# tenant=$(keystone tenant-list|awk '/redhat/ {print $2}')
# quantum net-create --tenant-id $tenant net02 \
    --provider:network_type vlan \
    --provider:physical_network physnet2 \
    --provider:segmentation_id 102
# quantum subnet-create --tenant-id $tenant --name net02_subnet01 net02 192.168.102.0/24
# quantum router-interface-add router01 net02_subnet01
```

## Example Configuration Steps for Case2 (1/2)

- Under the “service” tenant, define the public network.

```
# tenant=$(keystone tenant-list | awk '/service/ {print $2}')
```

```
# quantum net-create --tenant-id $tenant public01 \  
    --provider:network_type flat \  
    --provider:physical_network physnet1 \  
    --router:external=True
```

```
# quantum subnet-create --tenant-id $tenant --name public01_subnet01 \  
    --gateway 10.64.201.254 public01 10.64.201.0/24 --enable_dhcp False
```

- Under the user tenant “redhat”, create the tenant router and set its gateway for the public network.

```
# tenant=$(keystone tenant-list|awk '/redhat/ {print $2}')
```

```
# quantum router-create --tenant-id $tenant router01
```

```
# quantum router-gateway-set router01 public01
```

- Then, define private network “net01” and its subnet, and connect it to the router.

```
# quantum net-create --tenant-id $tenant net01 \  
    --provider:network_type vlan \  
    --provider:physical_network physnet2 \  
    --provider:segmentation_id 101
```

```
# quantum subnet-create --tenant-id $tenant --name net01_subnet01 net01 192.168.101.0/24
```

```
# quantum router-interface-add router01 net01_subnet01
```

## Example Configuration Steps for Case2 (2/2)

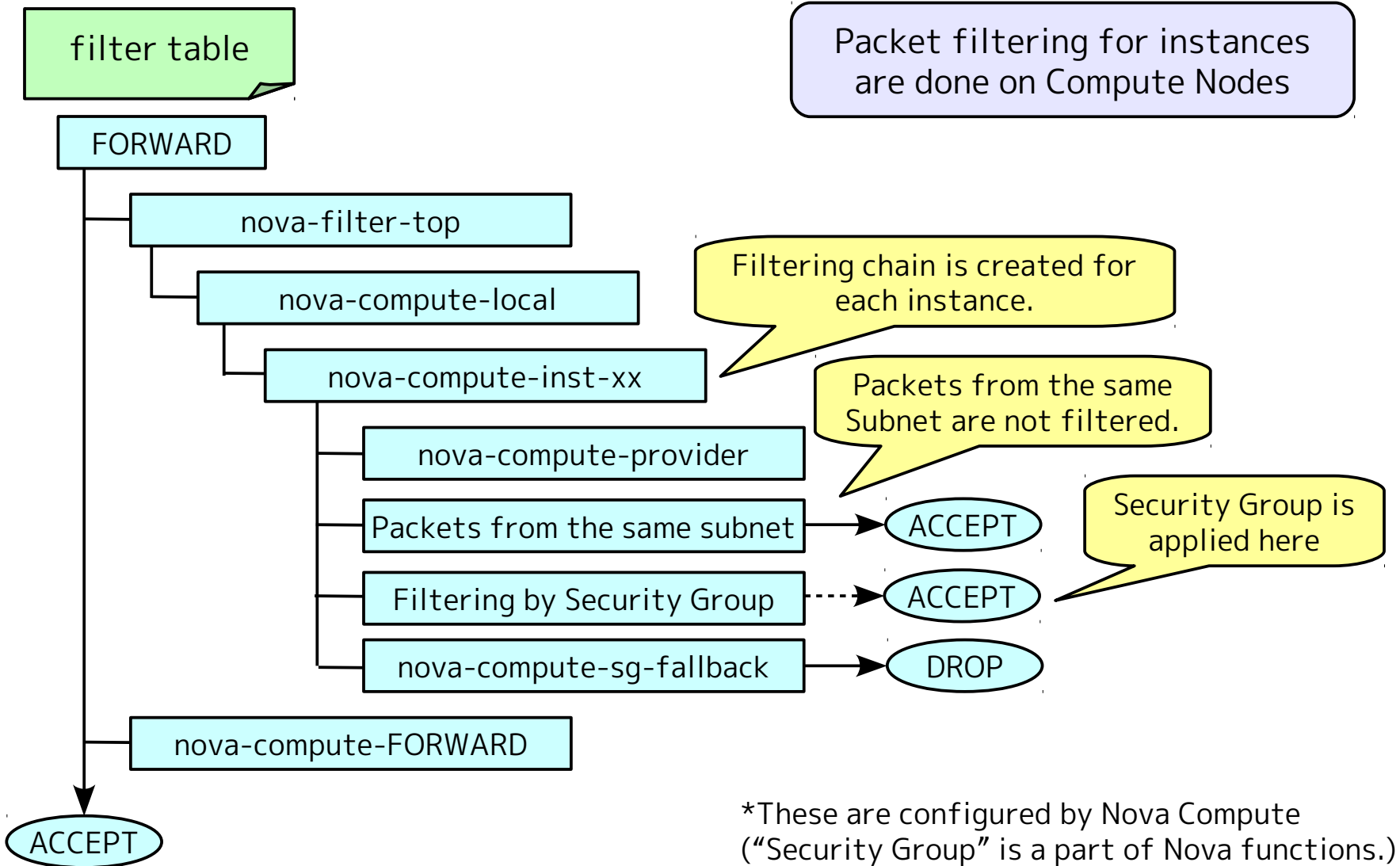
- Additional router and private networks for another tenant “redhat2” can be added in the same way.

```
# tenant=$(keystone tenant-list|awk '/redhat2/ {print $2}')
# quantum router-create --tenant-id $tenant router02
# quantum router-gateway-set router02 public01
```

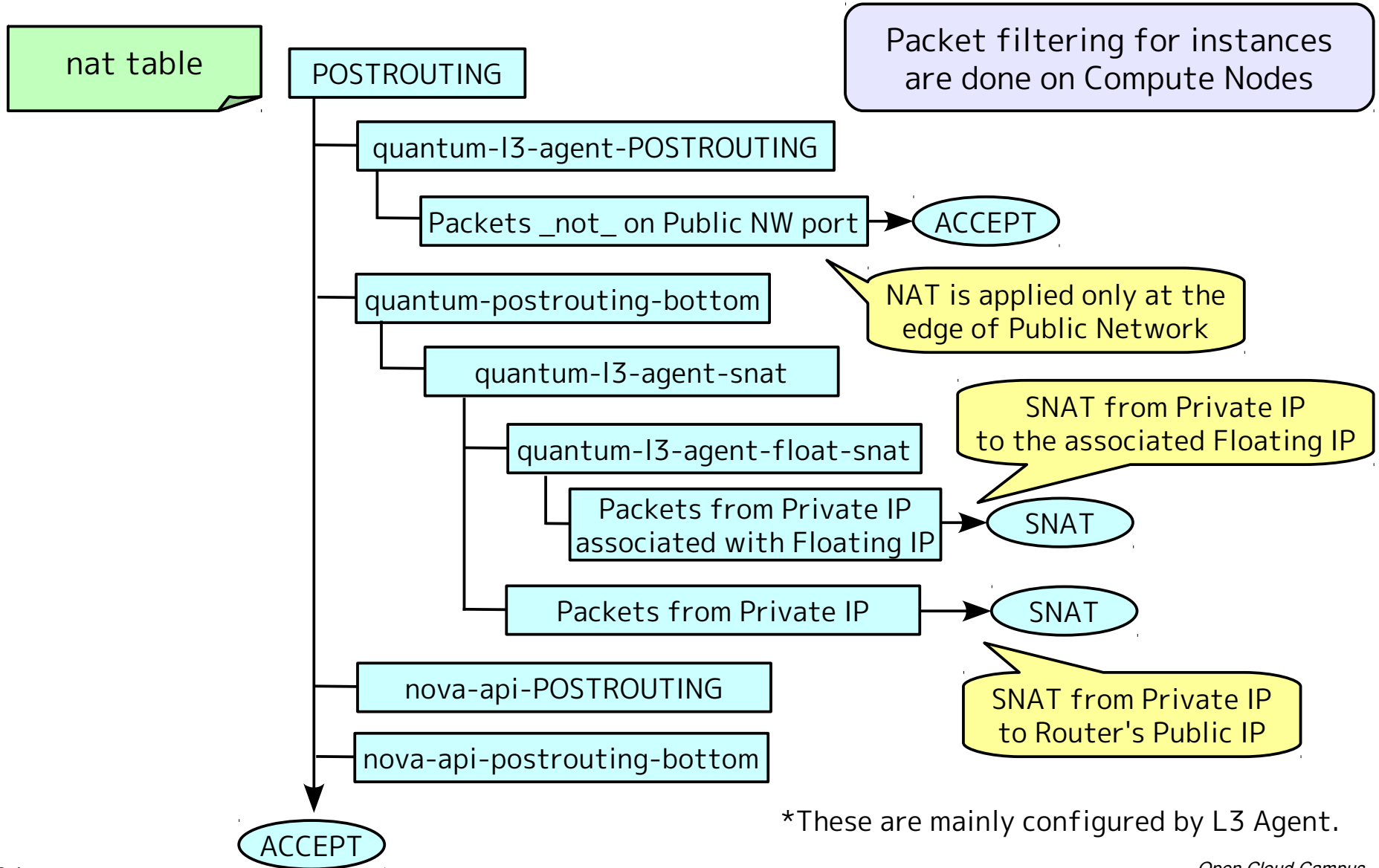
```
# quantum net-create --tenant-id $tenant net02 \
    --provider:network_type vlan \
    --provider:physical_network physnet2 \
    --provider:segmentation_id 102
# quantum subnet-create --tenant-id $tenant --name net02_subnet01 net01 192.168.101.0/24
# quantum router-interface-add router02 net02_subnet01
```

# iptables chains

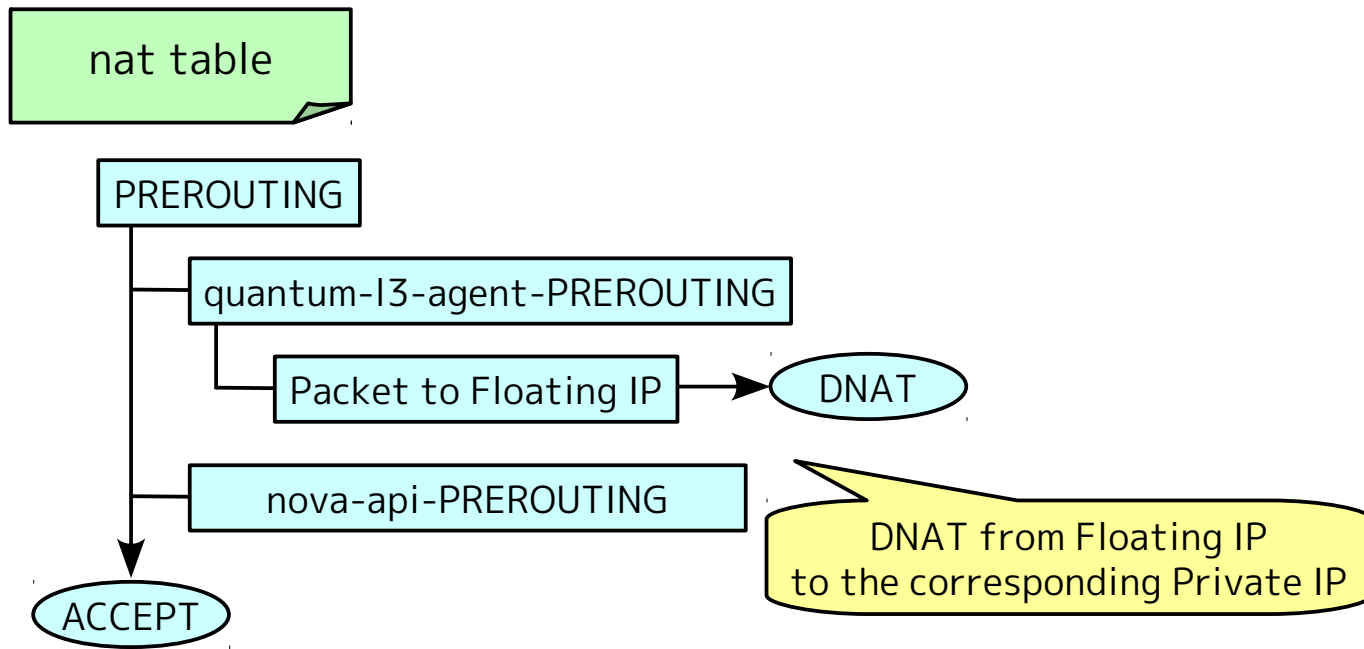
# Packet filtering chains on Compute Nodes



# NAT chains on Network Node (1/2)



## NAT chains on Network Node (2/2)



\*These are mainly configured by L3 Agent.



# References

## References

- OpenStack Network (Quantum) Administration Guide
  - <http://docs.openstack.org/trunk/openstack-network/admin/content/index.html>
- Quantum L2 Linux Bridge Plugin
  - <http://wiki.openstack.org/Quantum-Linux-Bridge-Plugin>
- QuickStart with RHOS(Red Hat OpenStack) Folsom Preview
  - <http://d.hatena.ne.jp/enakai00/20121118/1353226066>

**Enjoy the Life with OpenStack!**



**Etsuji Nakai**  
**Twitter @enakai00**