



Deep Dive into OpenStack Networking

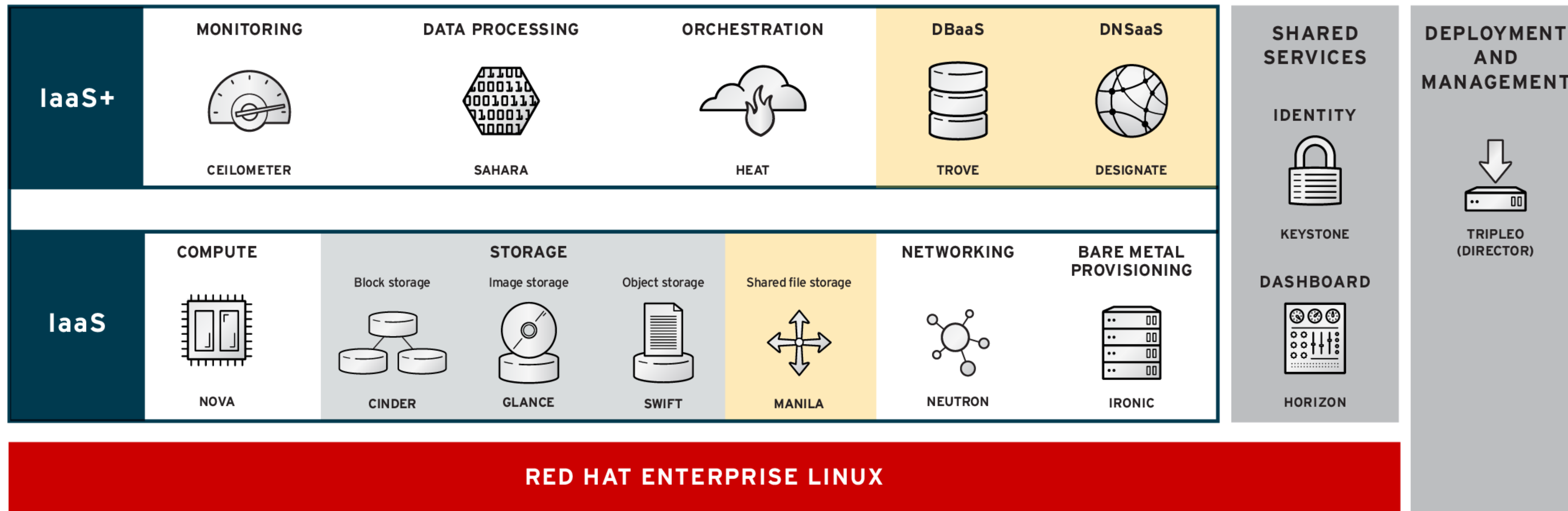
Damrongsak Reetanon <= RHCA RHCDS RHCE RHCSA RHCI =>

Solutions Architect

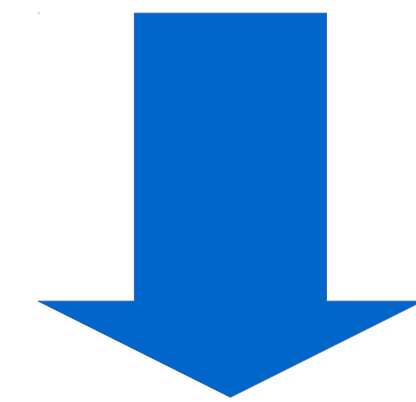
Red Hat (Thailand)

September 22, 2016

Deep Dive into OpenStack Networking



Deep Dive into OpenStack Networking



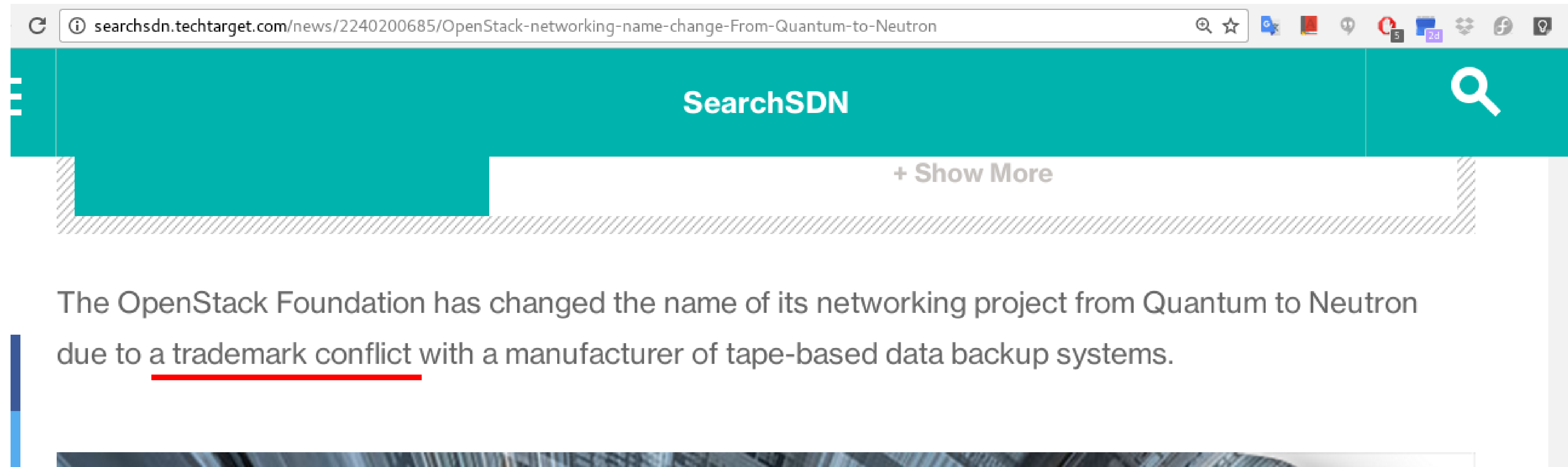
NEUTRON 

Network/neutron-renaming

This is our whiteboard for handling the Quantum -> Neutron transition.

Summary: Starting with the Havana release, the OpenStack Networking project's code name is Neutron. Quantum is no longer used

<https://wiki.openstack.org/wiki/Network/neutron-renaming>



What is Neutron?


Neutron is an OpenStack project to provide "networking as a service" between interface devices (e.g., vNICs) managed by other Openstack services (e.g., nova).

Starting in the Folsom release, Neutron is a core and supported part of the OpenStack platform (for Essex, we were an "incubated" project, which means use is suggested only for those who really know what they're doing with Neutron).

Core Services (6 Results)

NOVA

Compute



Manages the lifecycle of compute instances in an OpenStack environment. Responsibilities include spawning, scheduling and decommissioning of machines on demand.

93 %

Adoption

8 OF 8

Maturity


6 YRS

Age

MORE DETAILS

NEUTRON

Networking



Enables network connectivity as a service for other OpenStack services, such as OpenStack Compute. Provides an API for users to define networks and the attachments into them. Has a pluggable architecture that supports many popular networking vendors and technologies.

84 %

Adoption

8 OF 8

Maturity


4 YRS

Age

MORE DETAILS

SWIFT

Object Storage



Stores and retrieves arbitrary unstructured data objects via a RESTful, HTTP based API. It is highly fault tolerant with its data replication and scale out architecture. Its implementation is not like a file server with mountable directories.

52 %

Adoption

7 OF 8

Maturity


6 YRS

Age

MORE DETAILS

CINDER

Block Storage



Provides persistent block storage to running instances. Its pluggable driver architecture facilitates the creation and management of block storage devices.

81 %

Adoption

7 OF 8

Maturity


4 YRS

Age

MORE DETAILS

KEYSTONE

Identity



Provides an authentication and authorization service for other OpenStack services. Provides a catalog of endpoints for all OpenStack services.

88 %

Adoption

7 OF 8

Maturity


5 YRS

Age

MORE DETAILS

GLANCE

Image Service



Stores and retrieves virtual machine disk images. OpenStack Compute makes use of this during instance provisioning.

87 %

Adoption

7 OF 8

Maturity

6 YRS

Age

MORE DETAILS

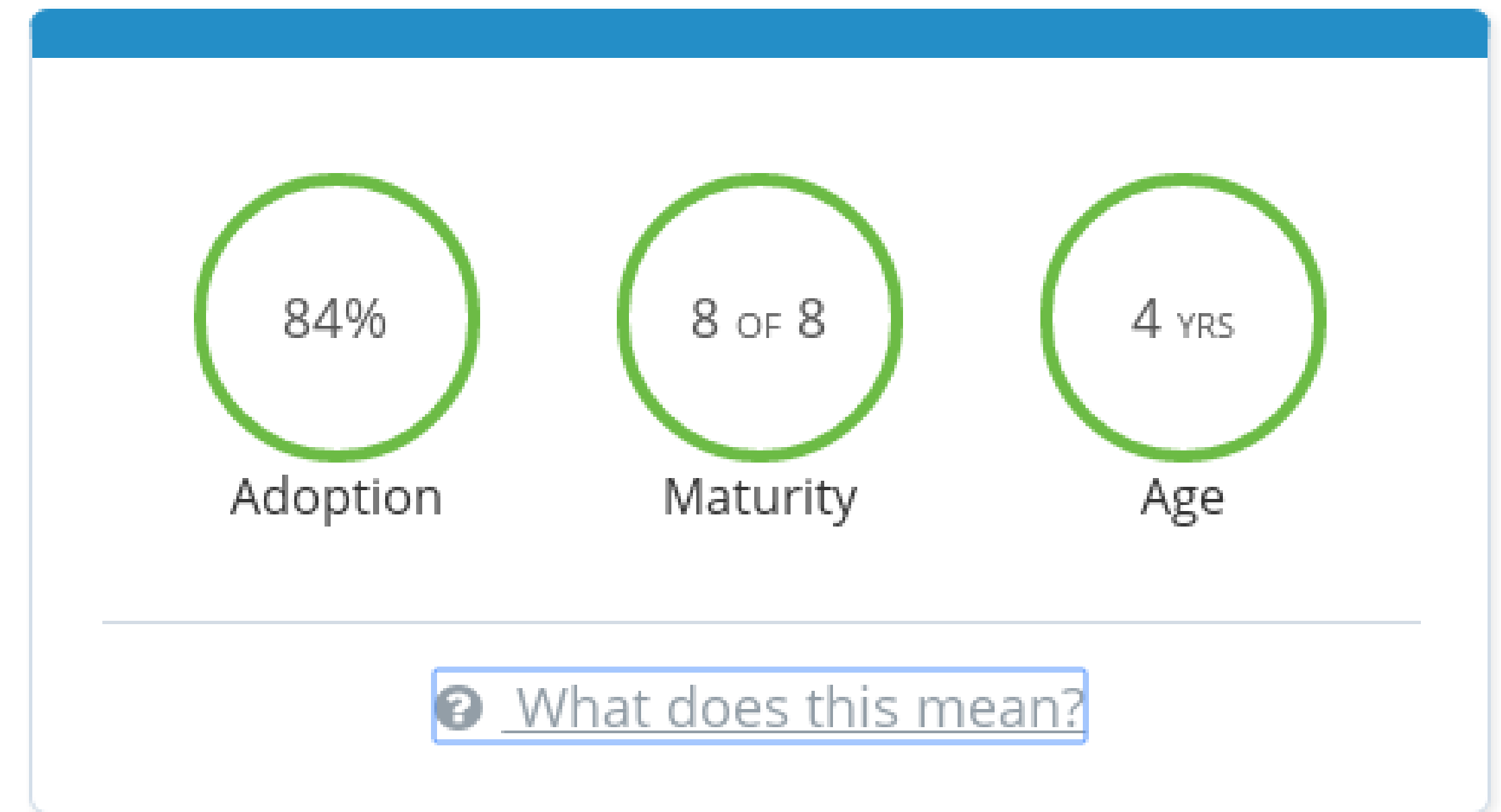
NEUTRON

Networking

[Project wiki page](#)

[View the install guide](#)

[Find this service in the Marketplace](#)



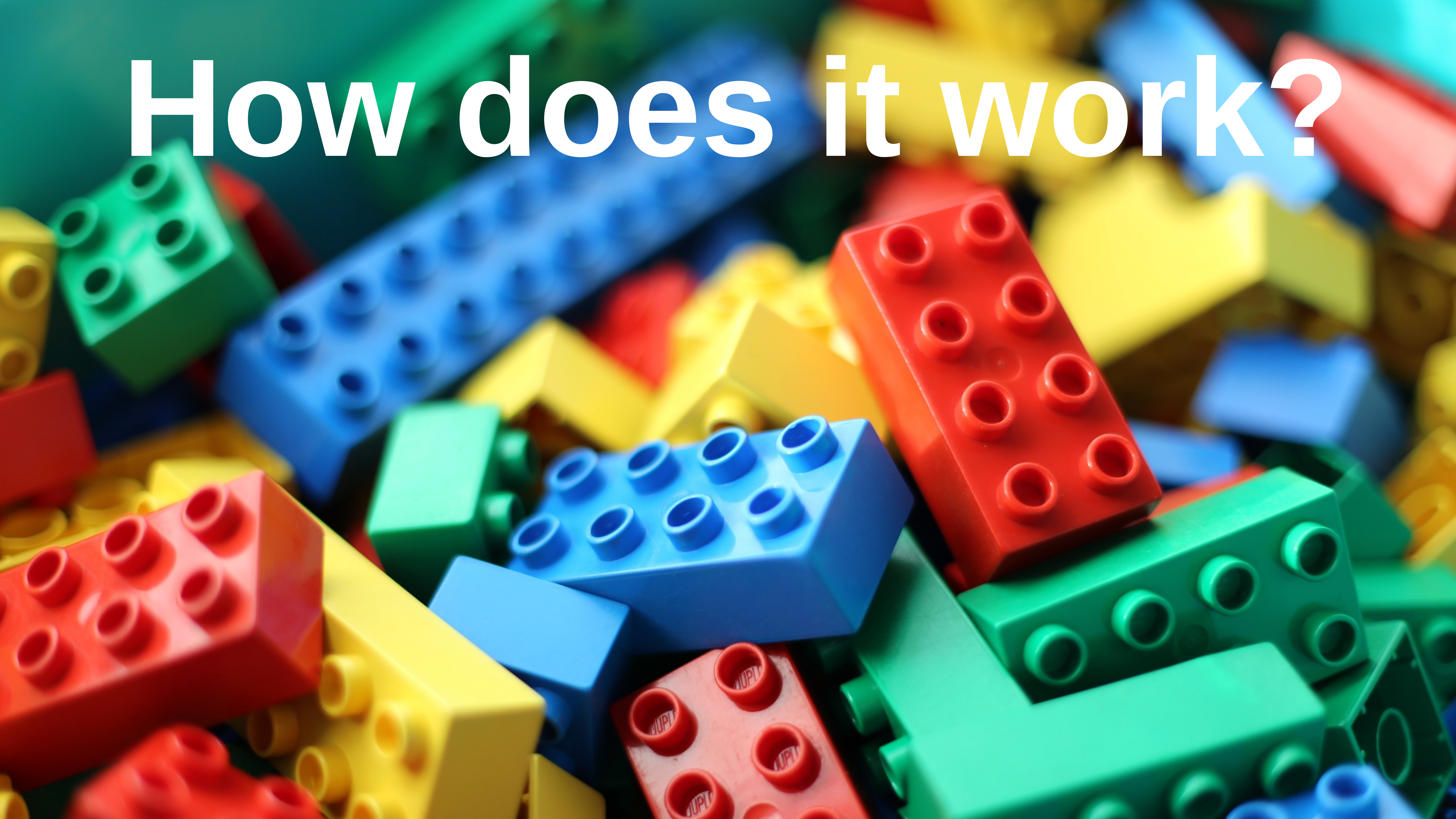
About this project

Enables network connectivity as a service for other OpenStack services, such as OpenStack Compute. Provides an API for users to define networks and the attachments into them. Has a pluggable architecture that supports many popular networking vendors and technologies.



Neutron Components

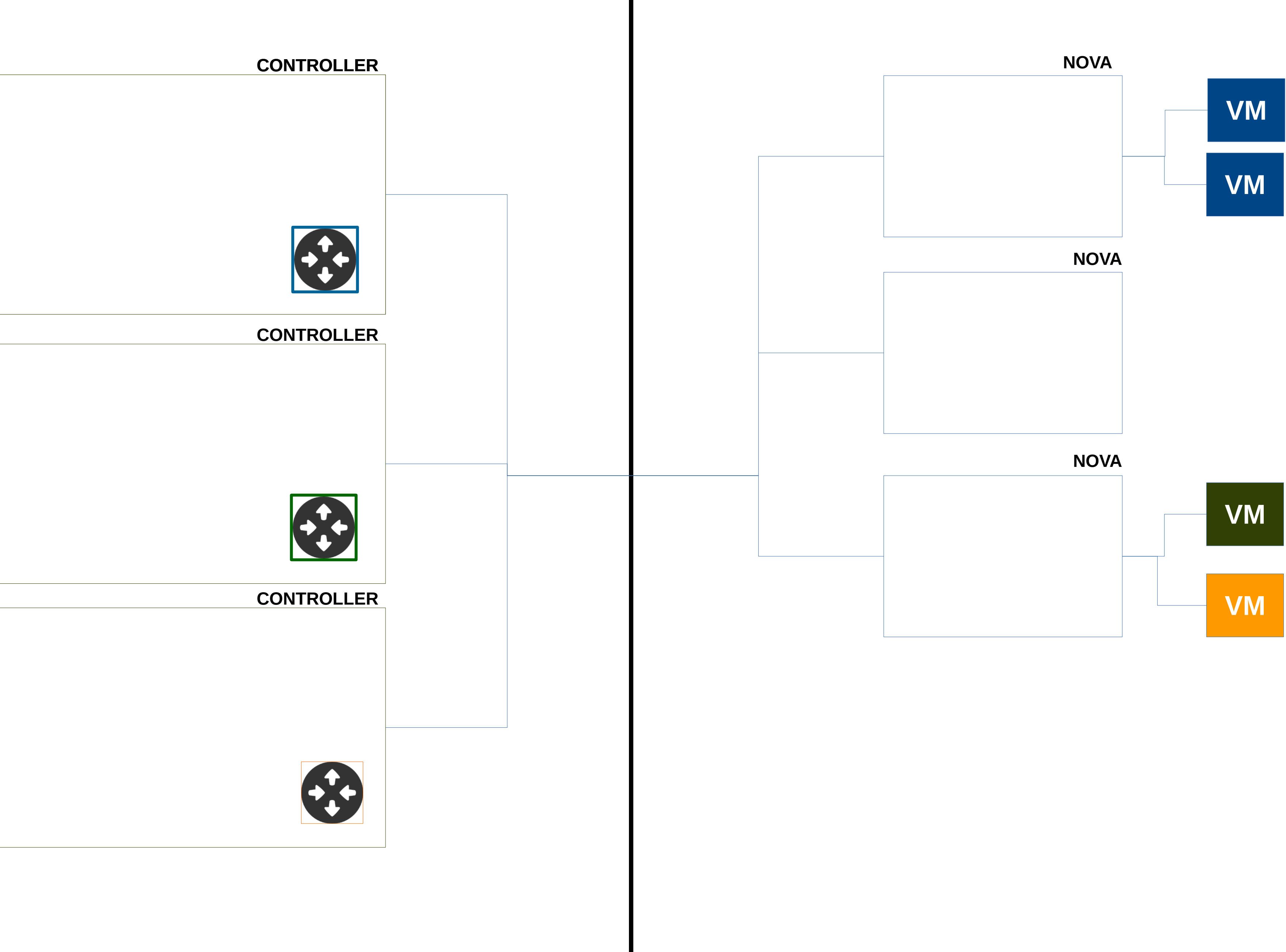
How does it work?

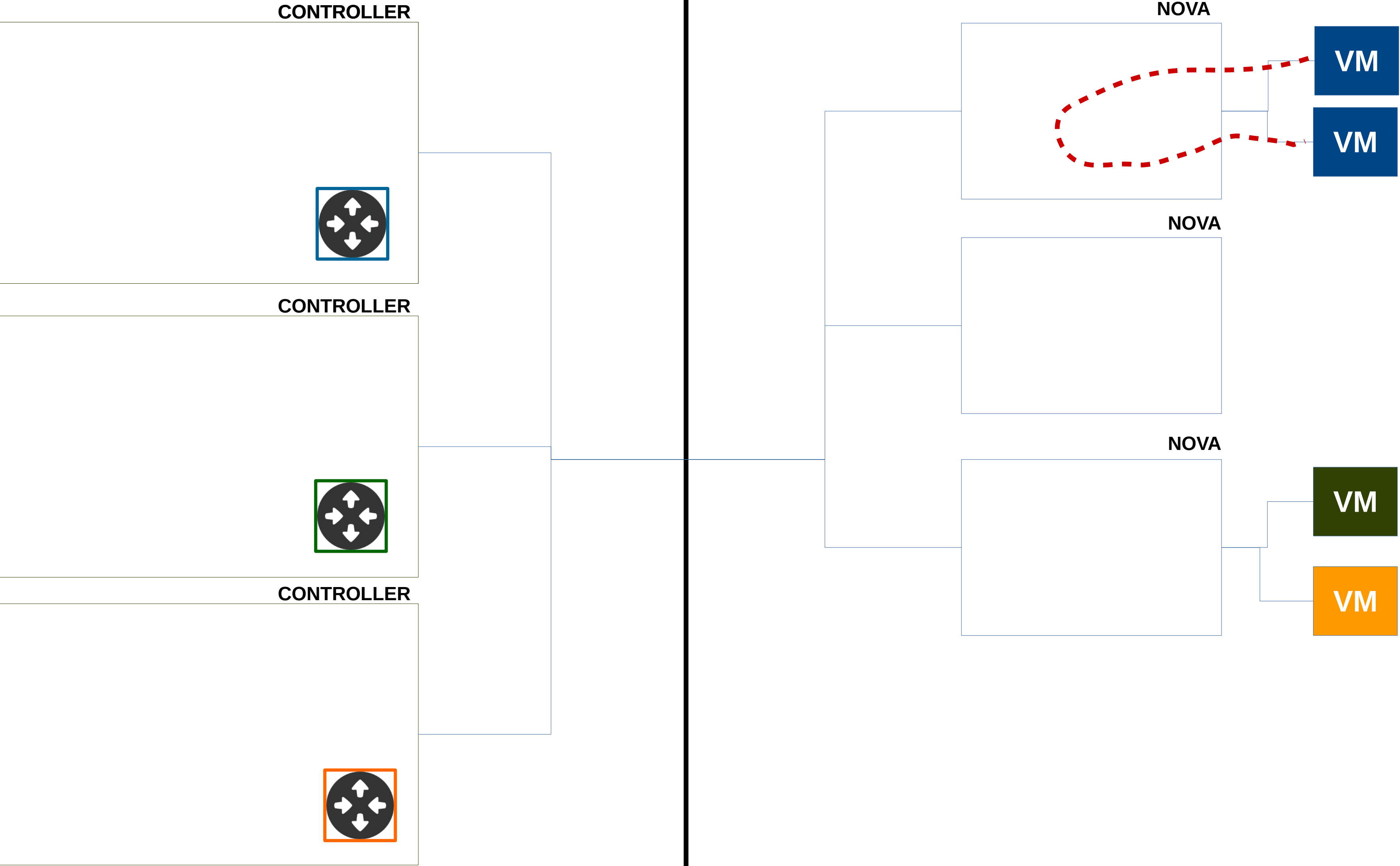


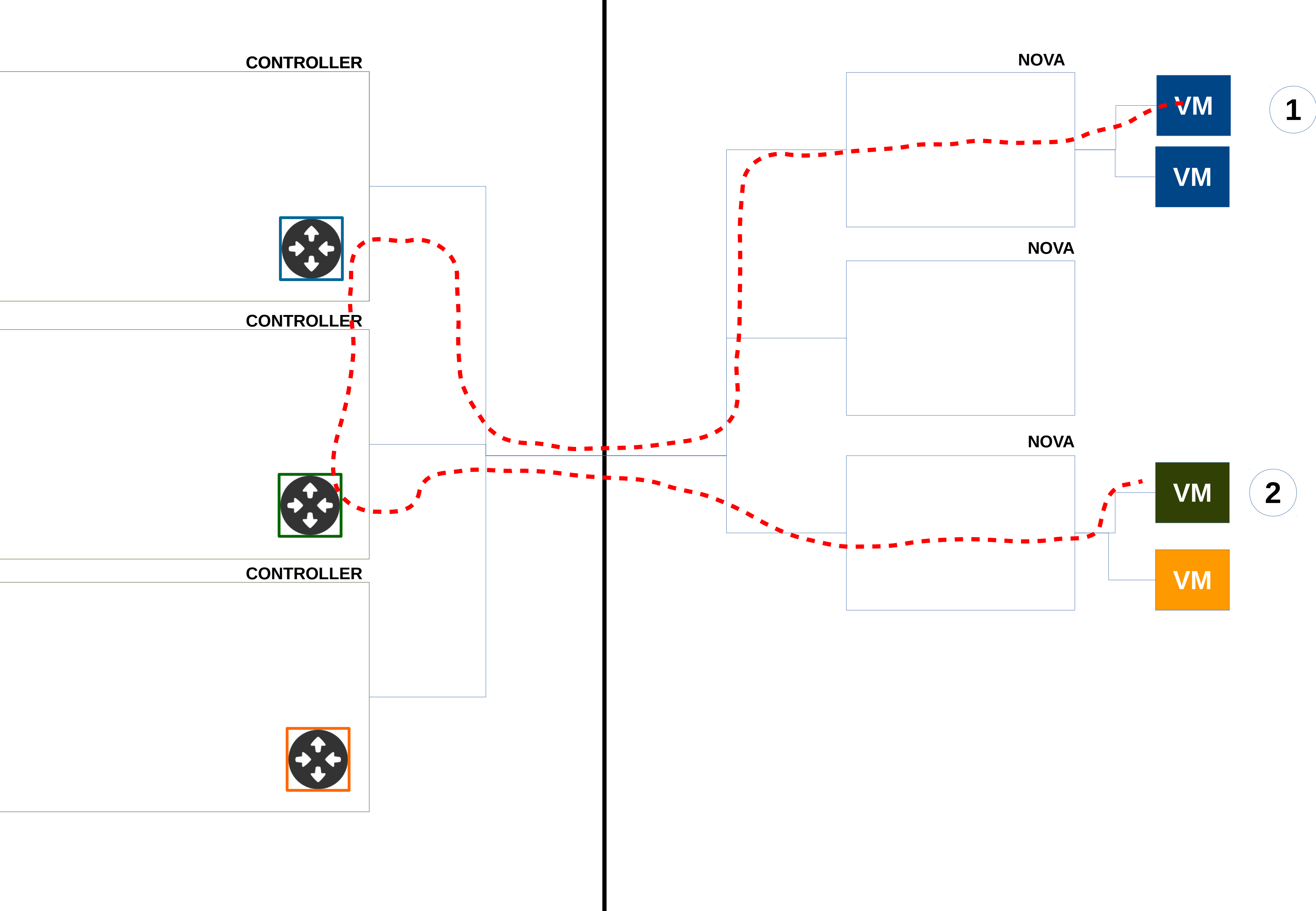


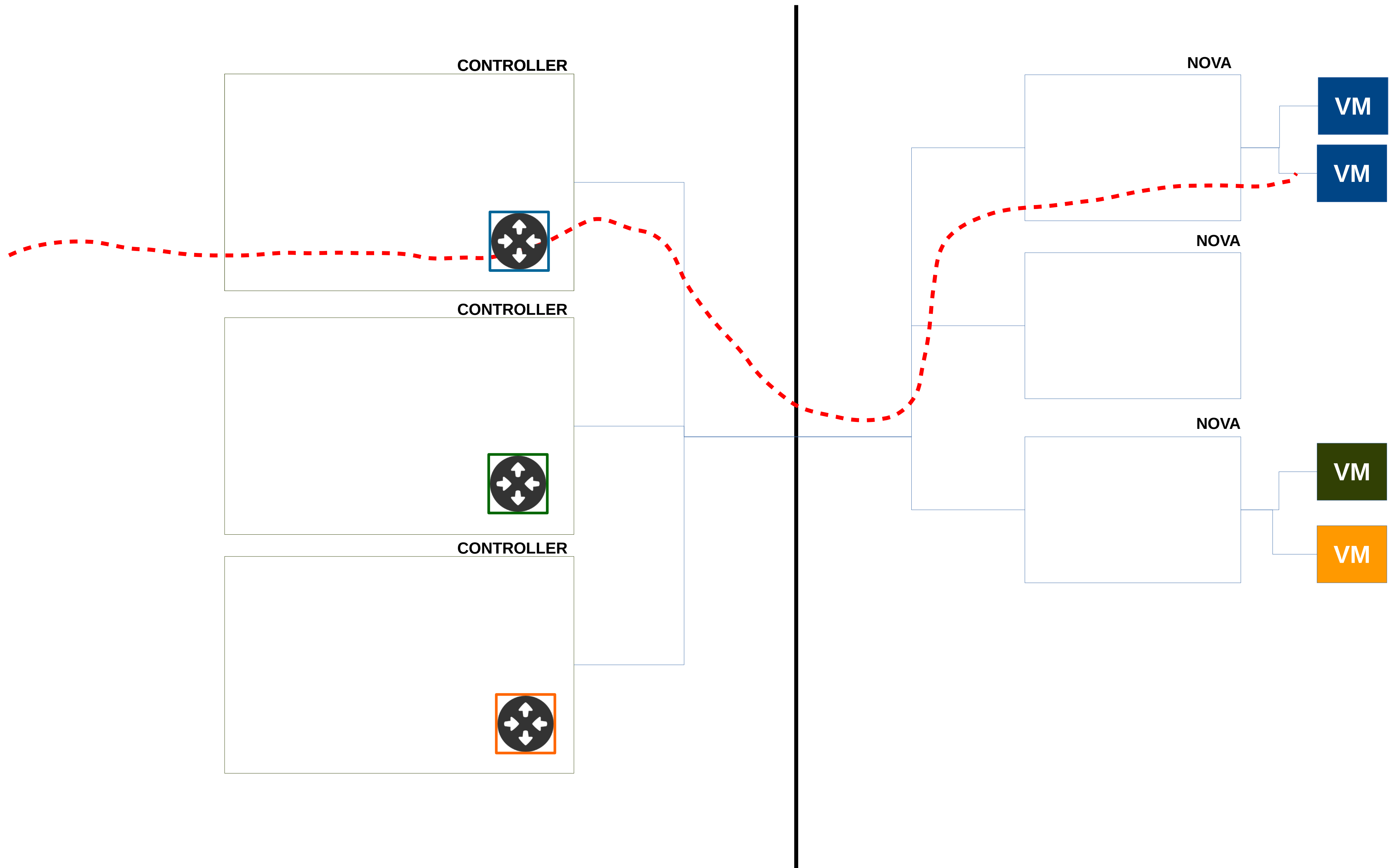
All in One - Controller

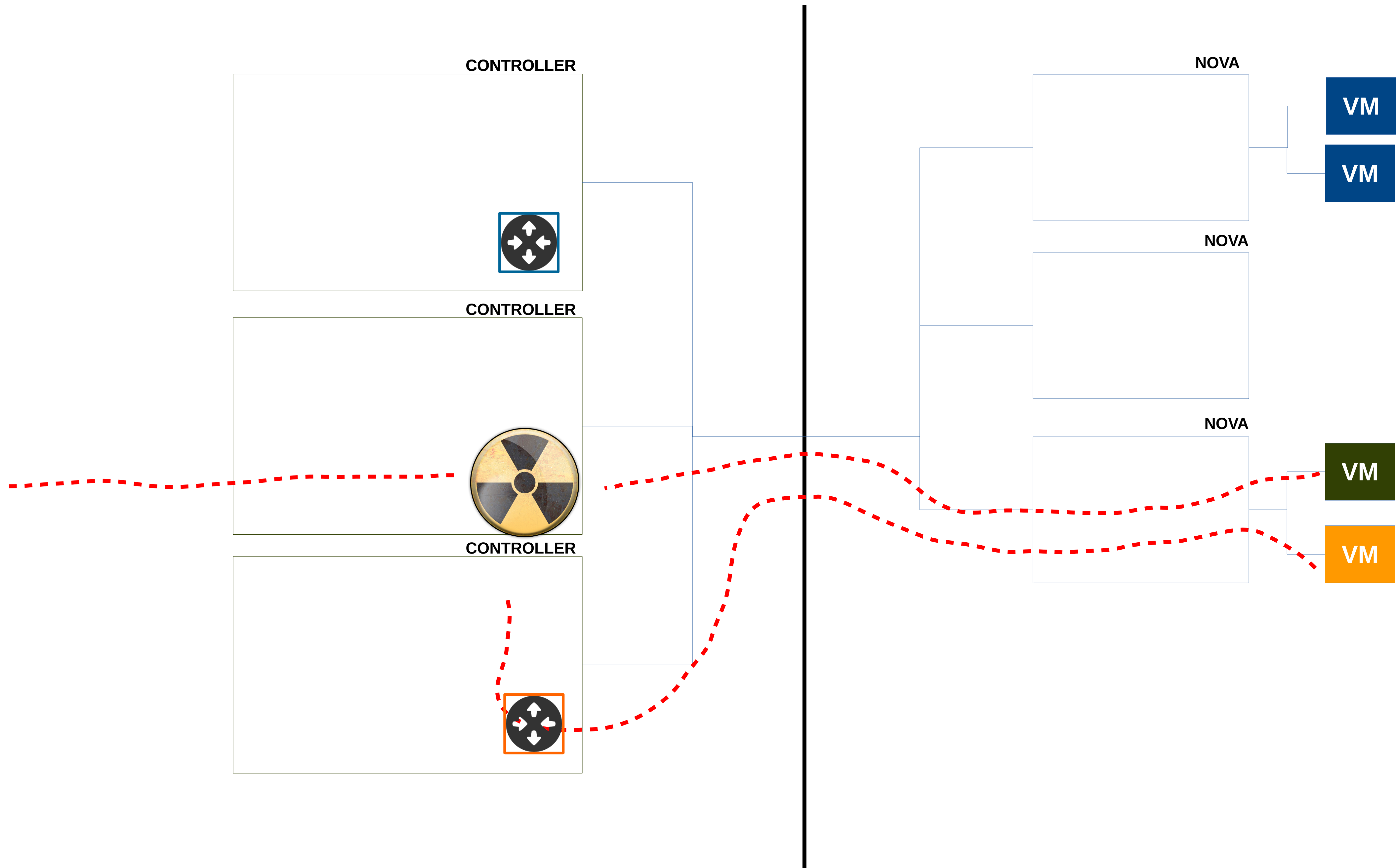
1

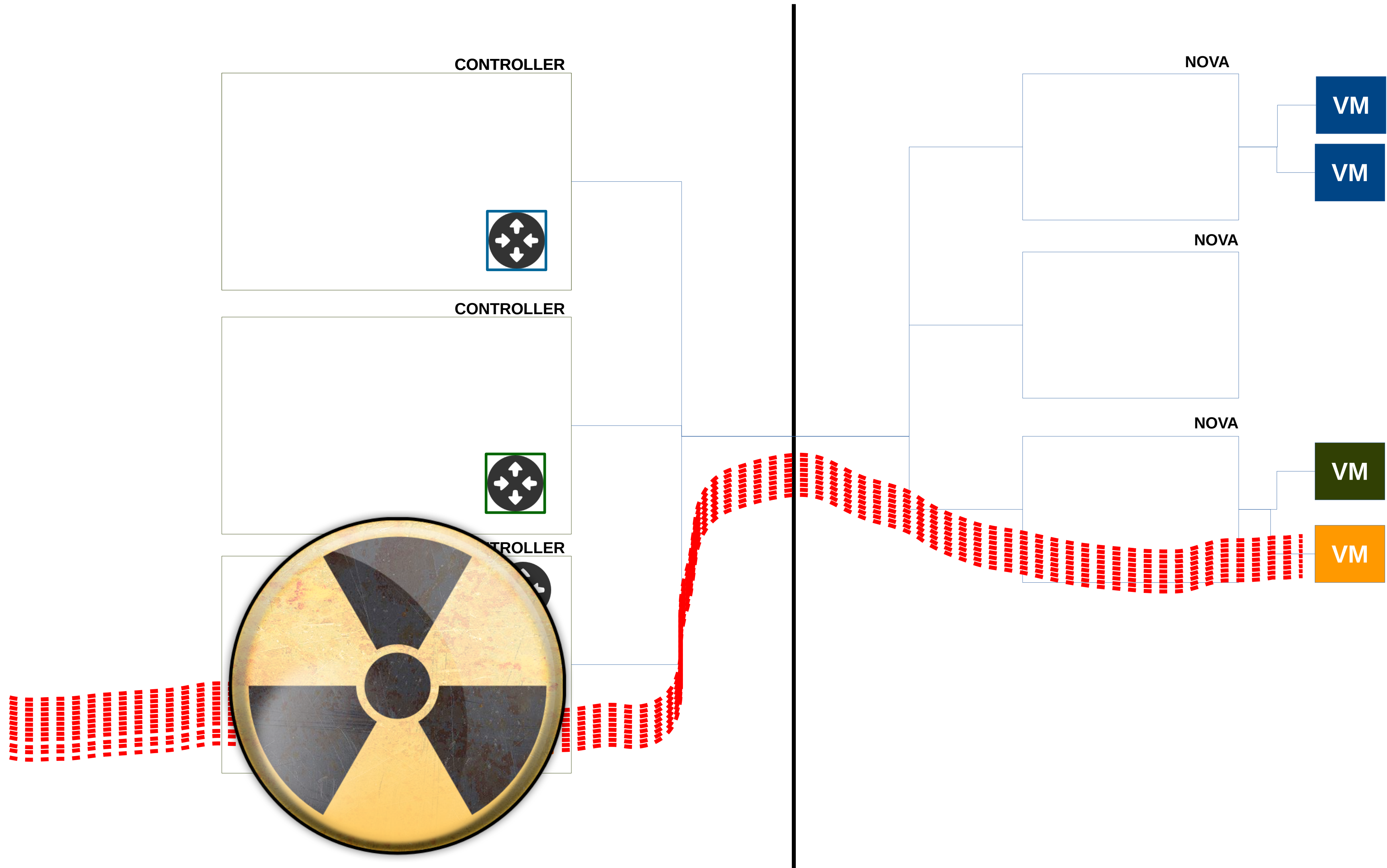


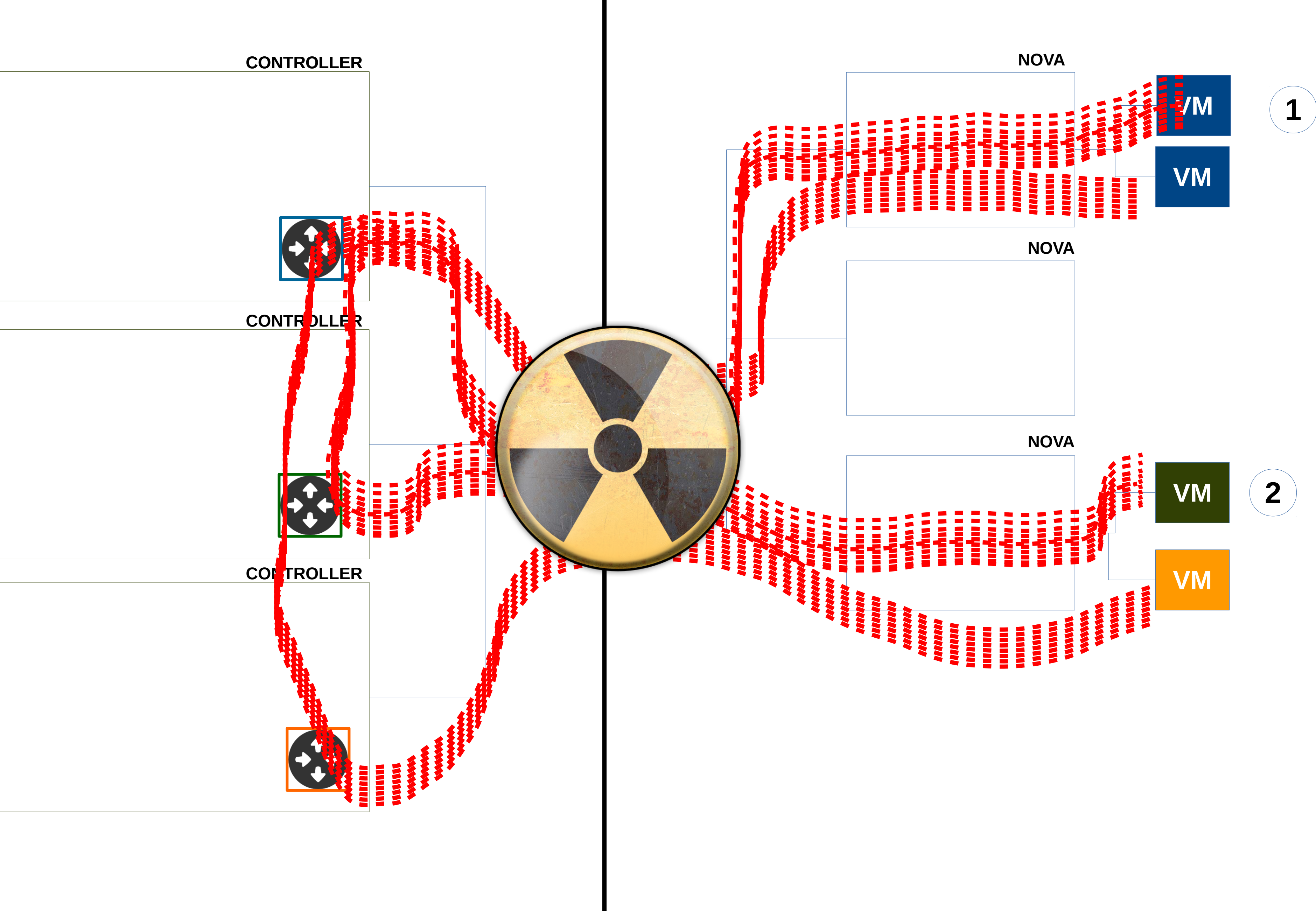








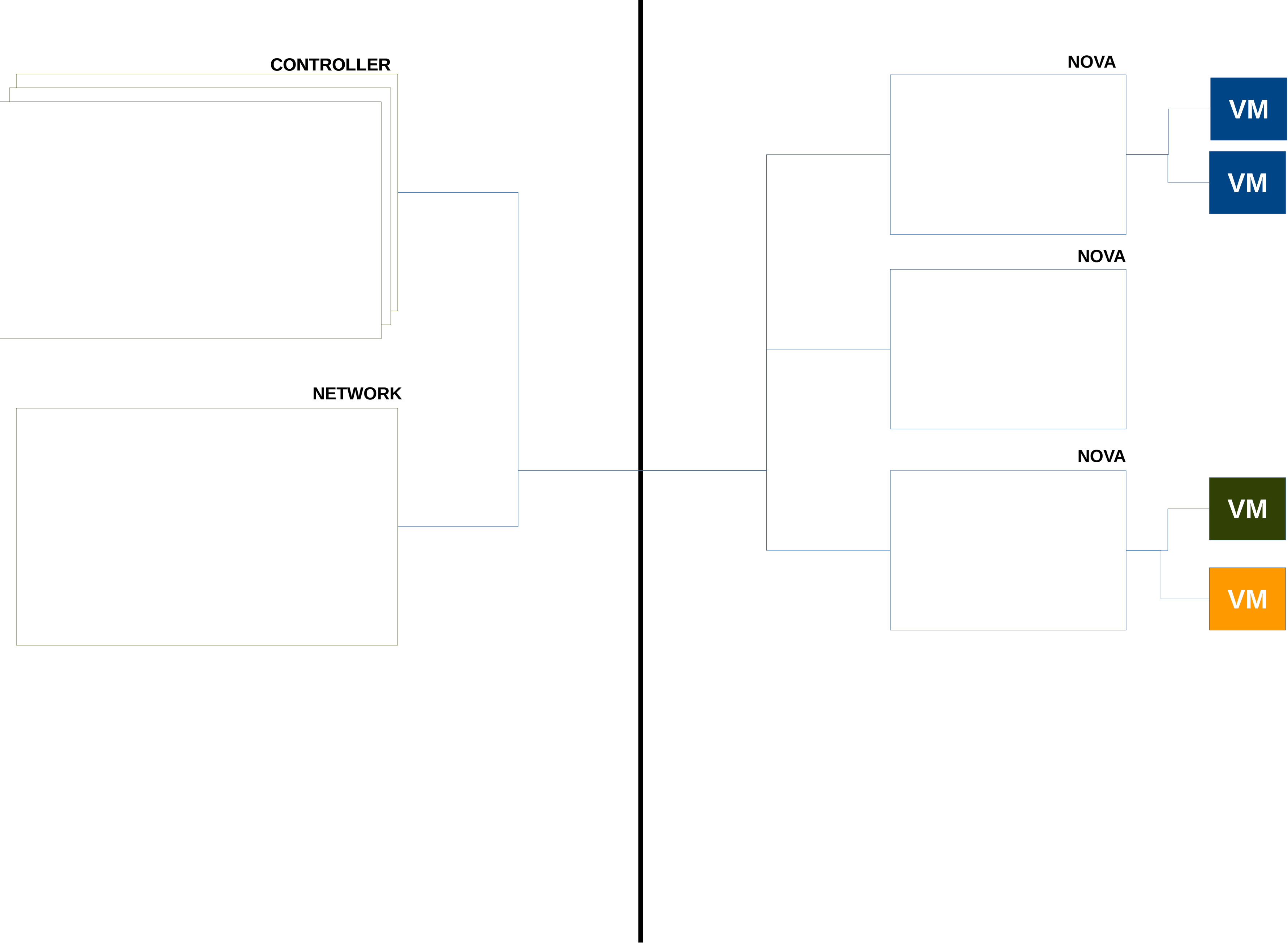


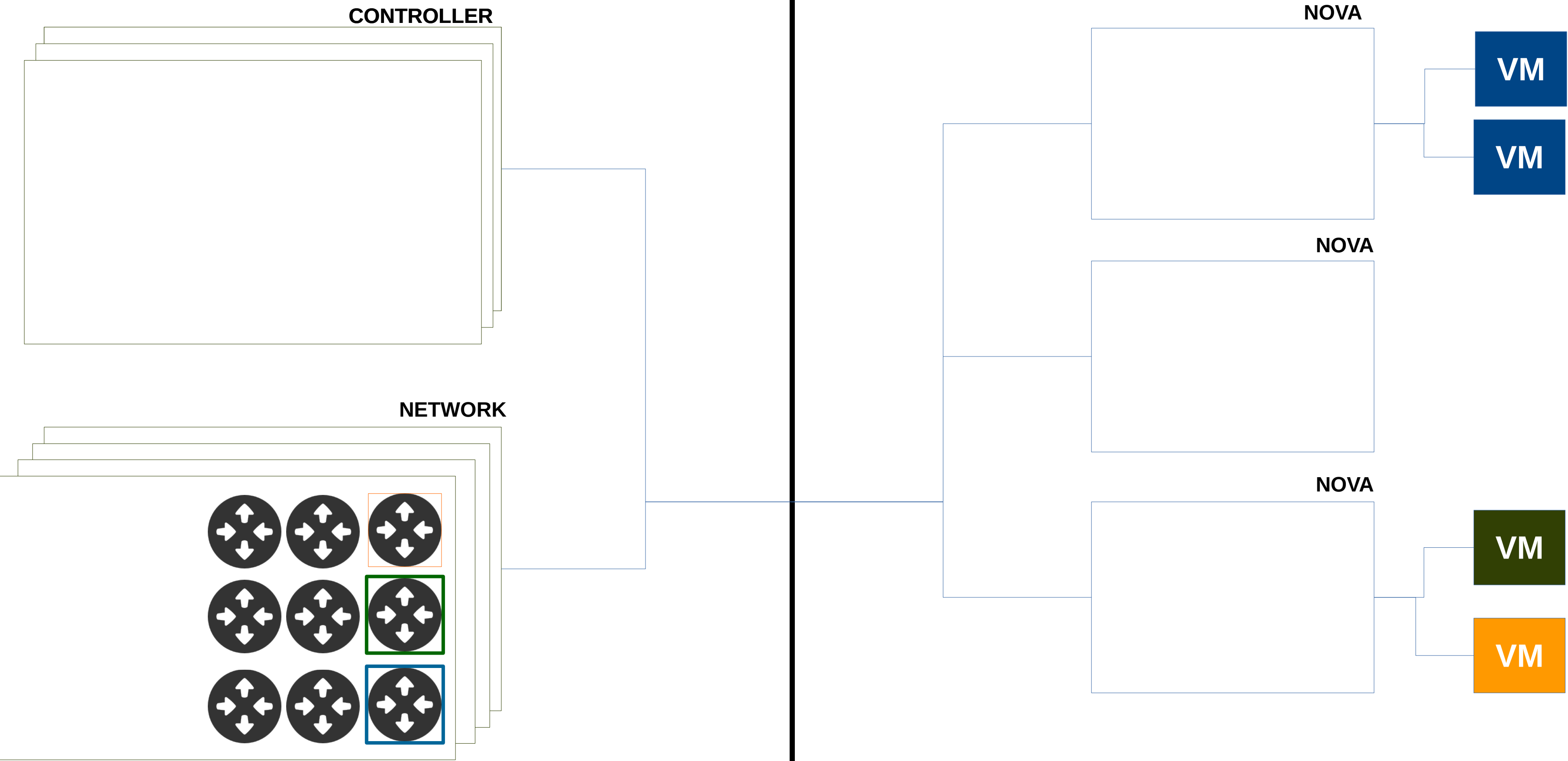


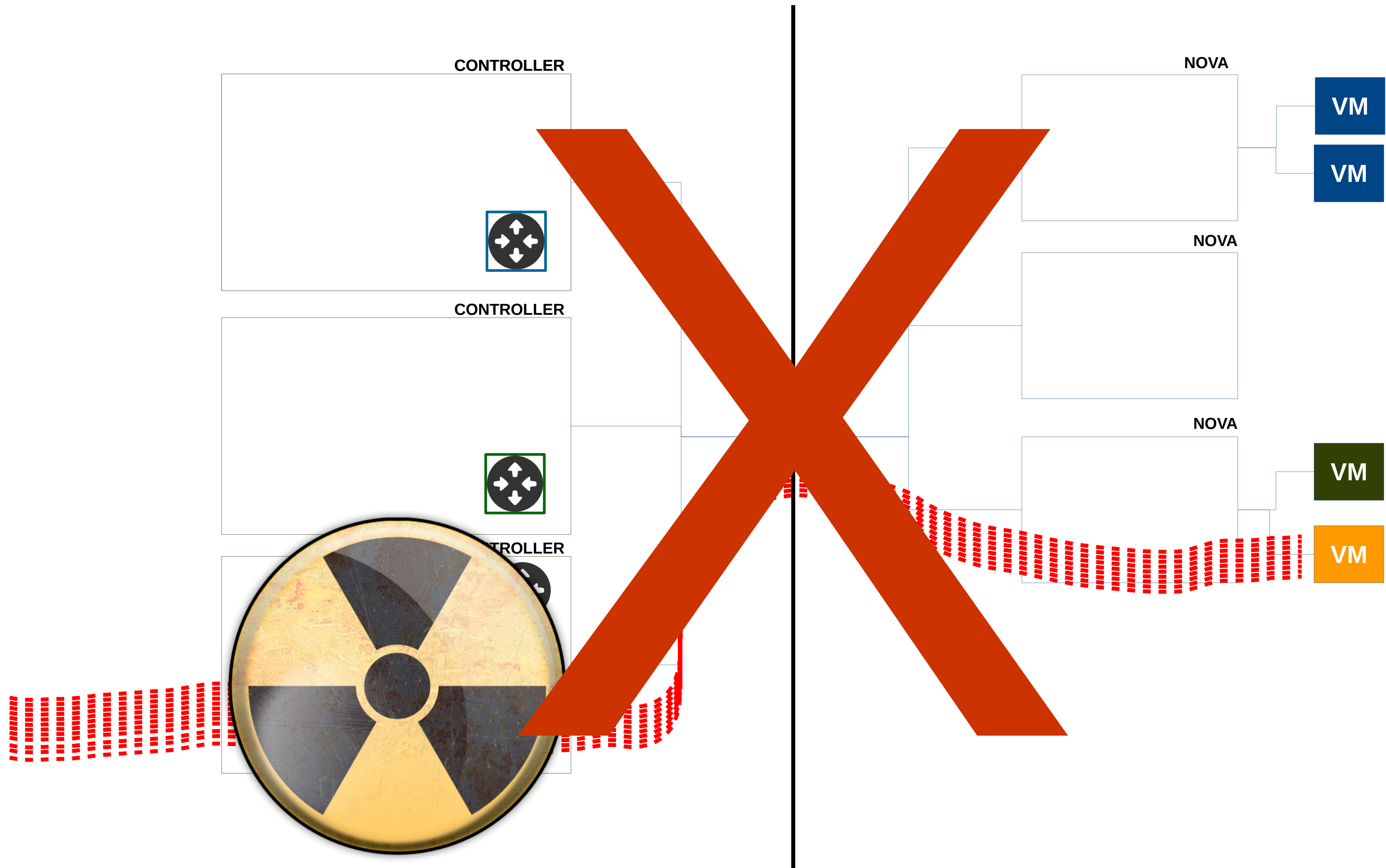


Multiple Network Node

2



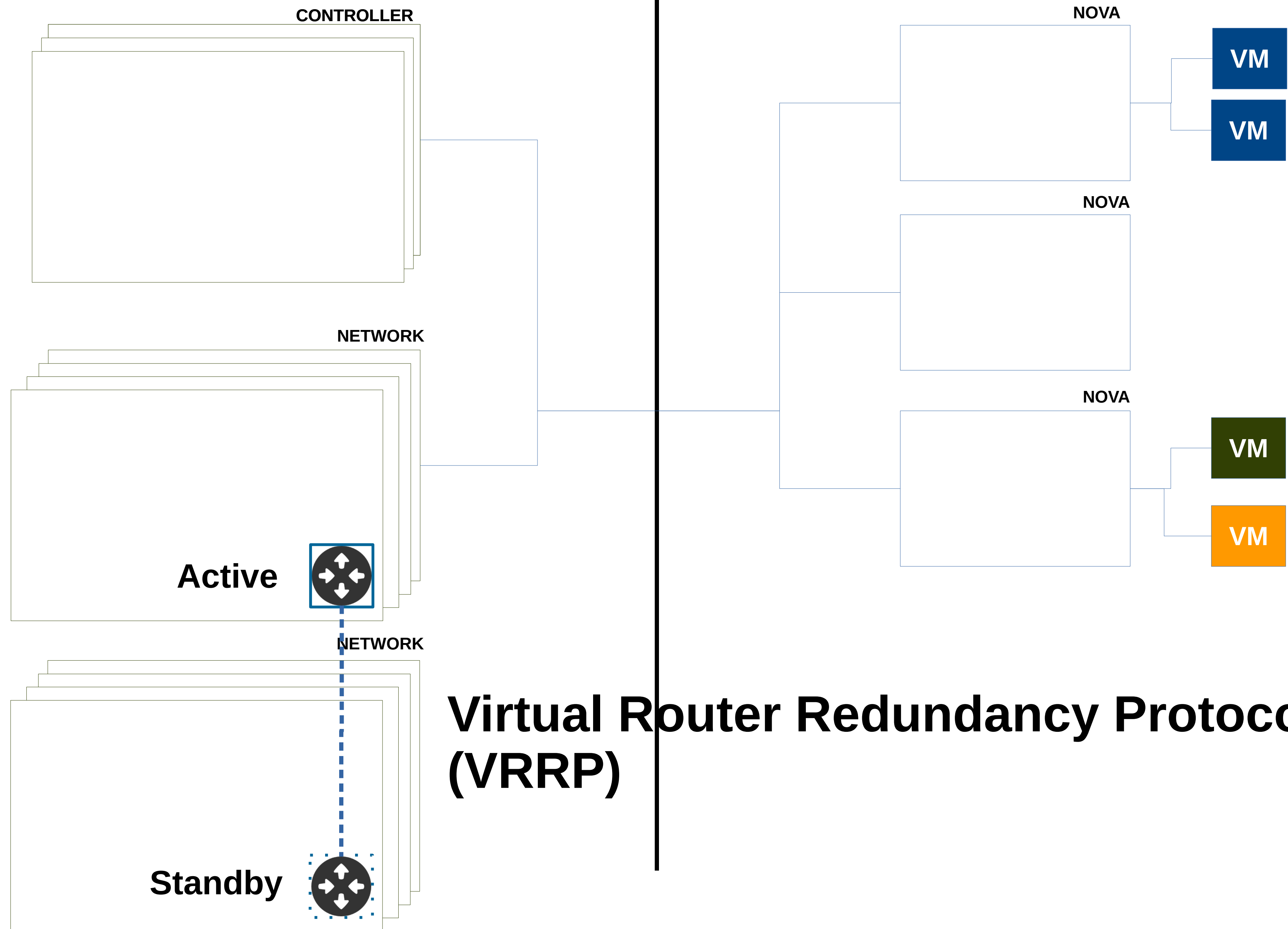




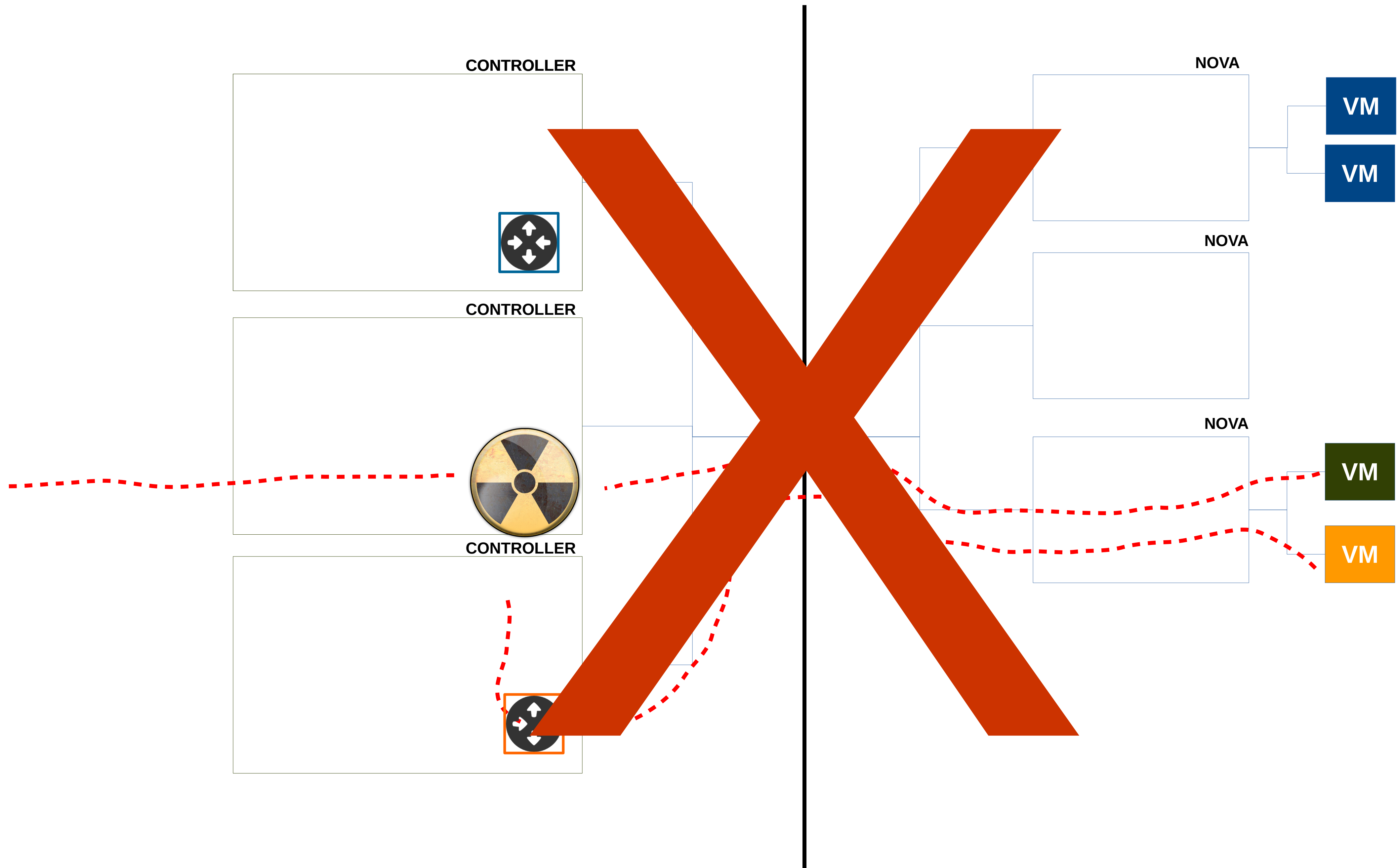


L3-High Availability

3



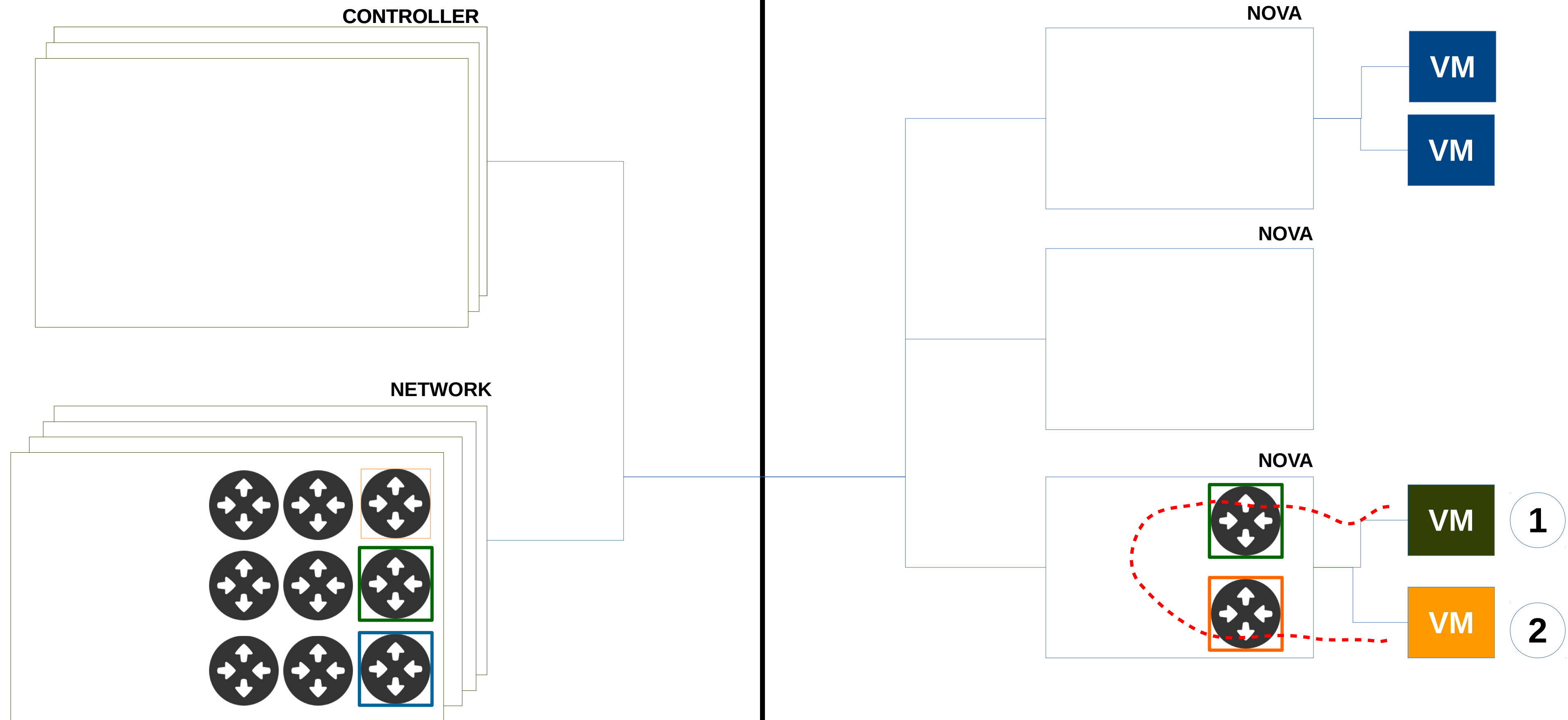
Virtual Router Redundancy Protocol (VRRP)

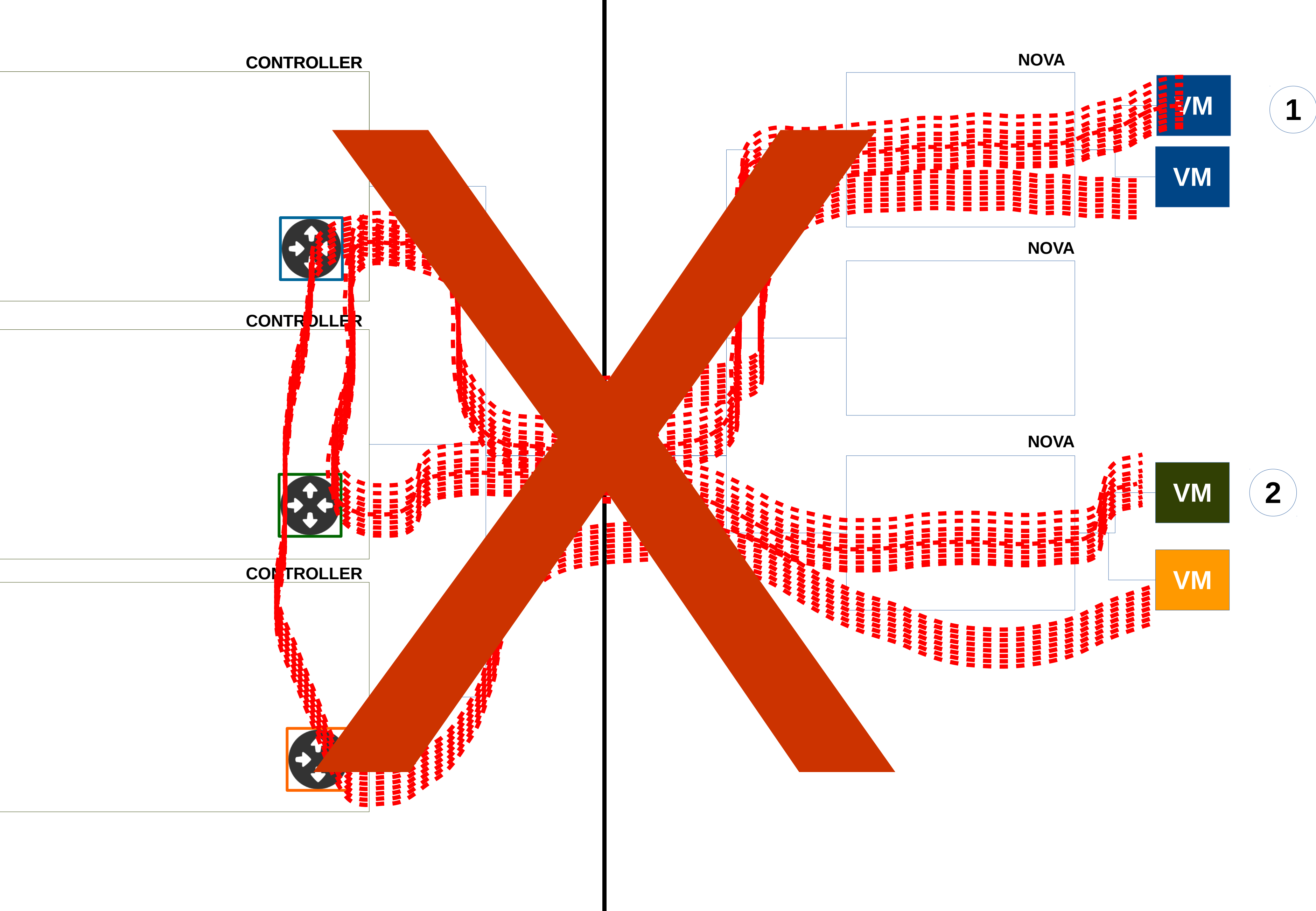




Distributed Virtual Router (DVR)

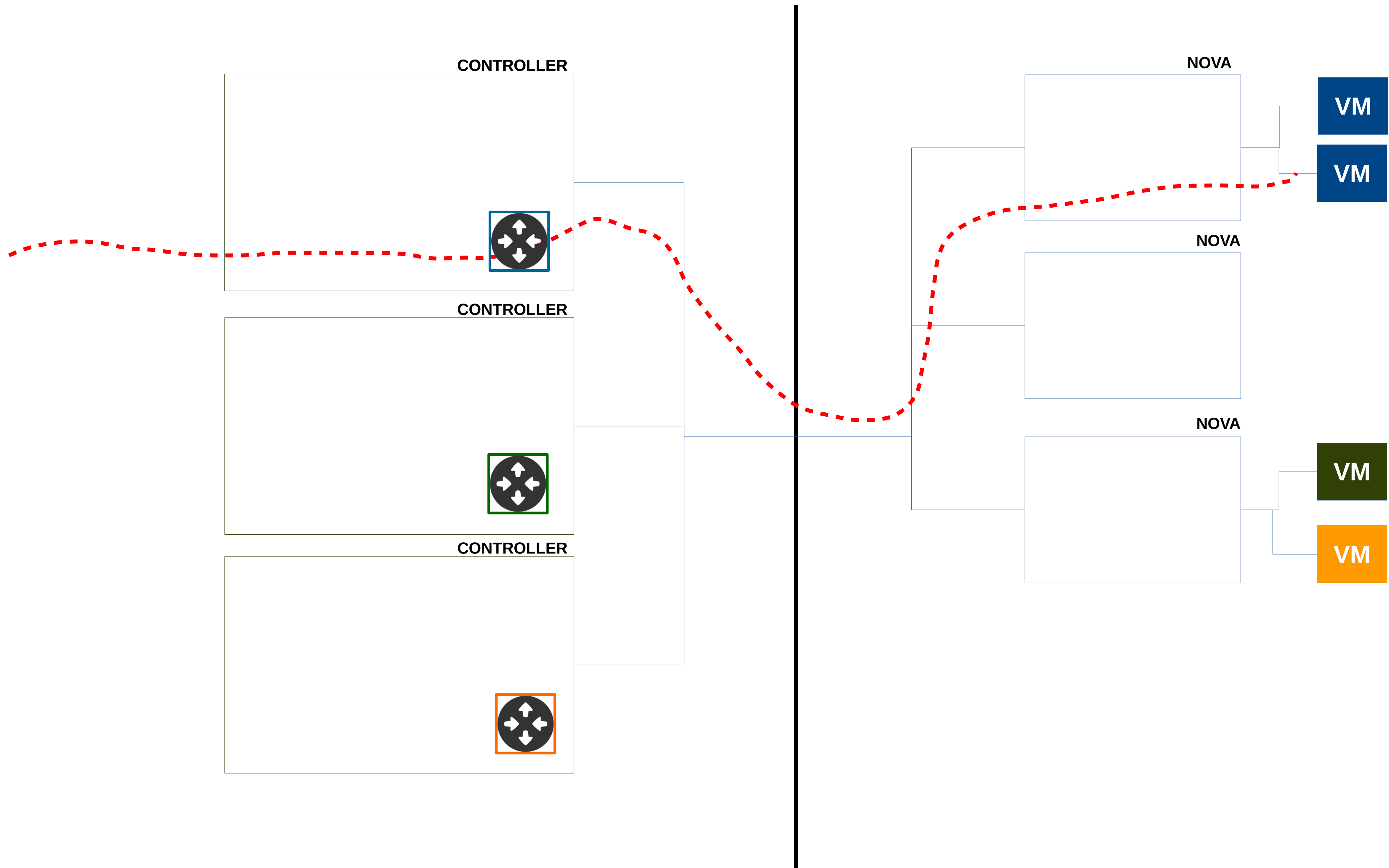
3

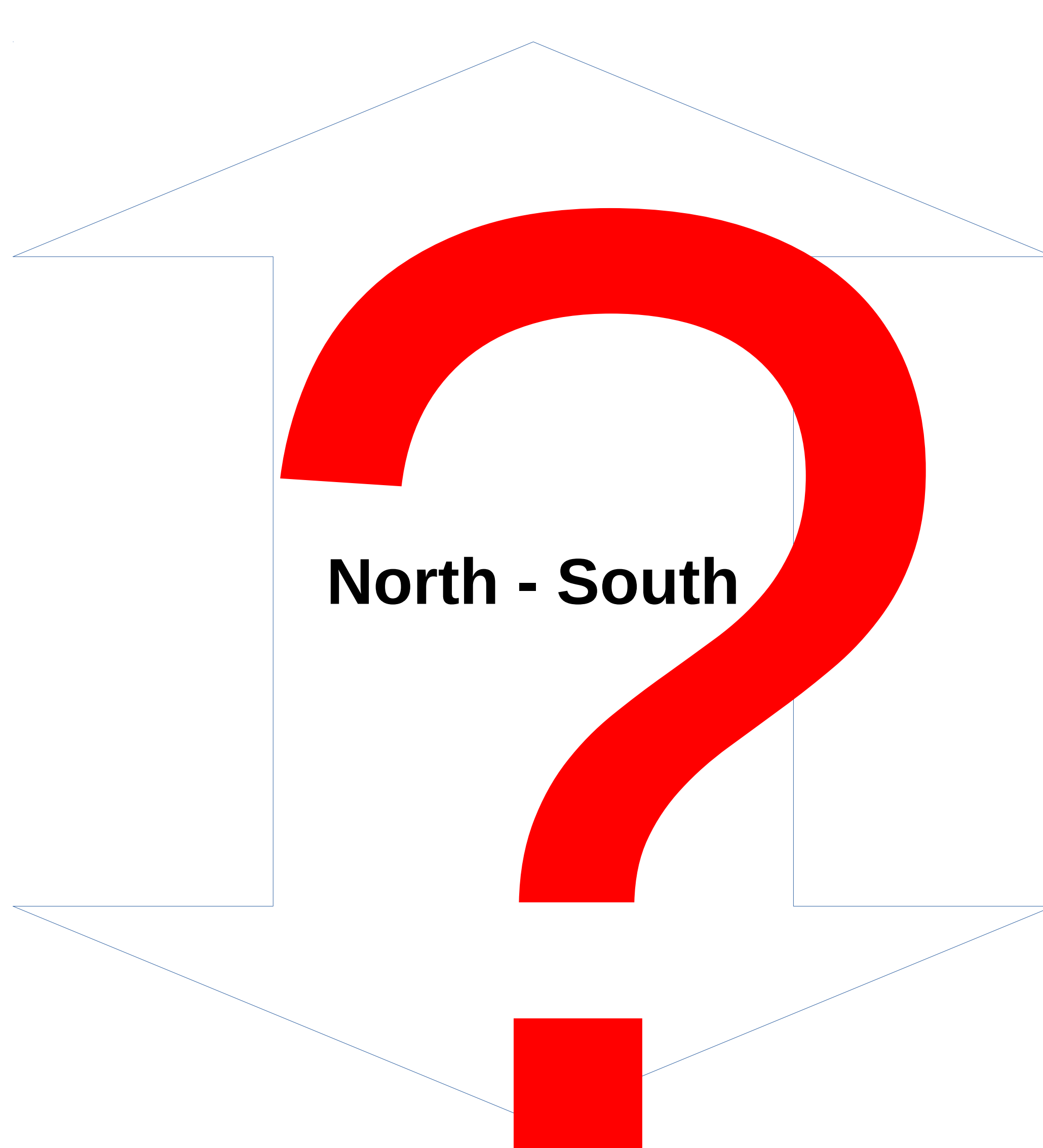






EAST - WEST Traffic



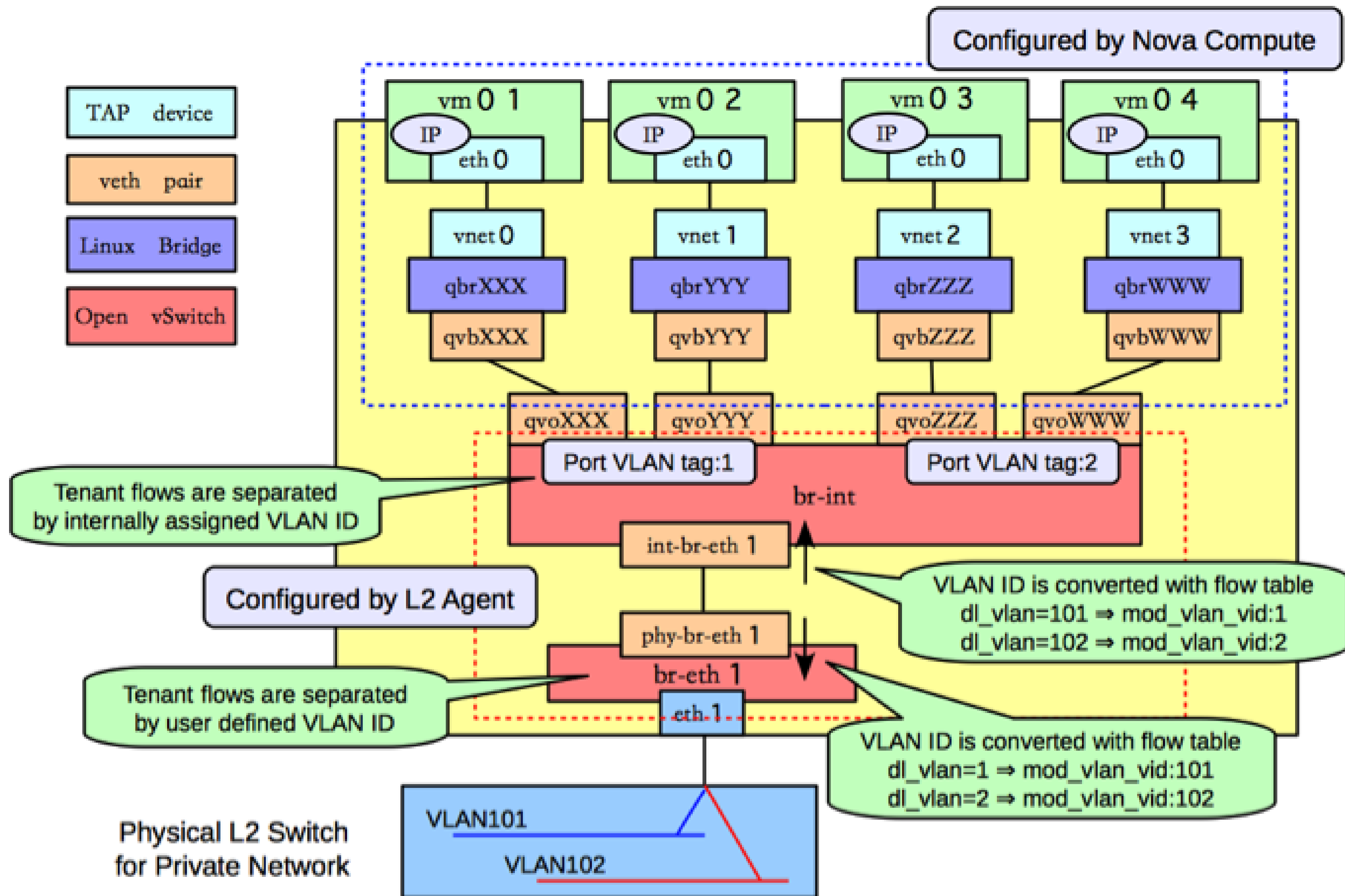


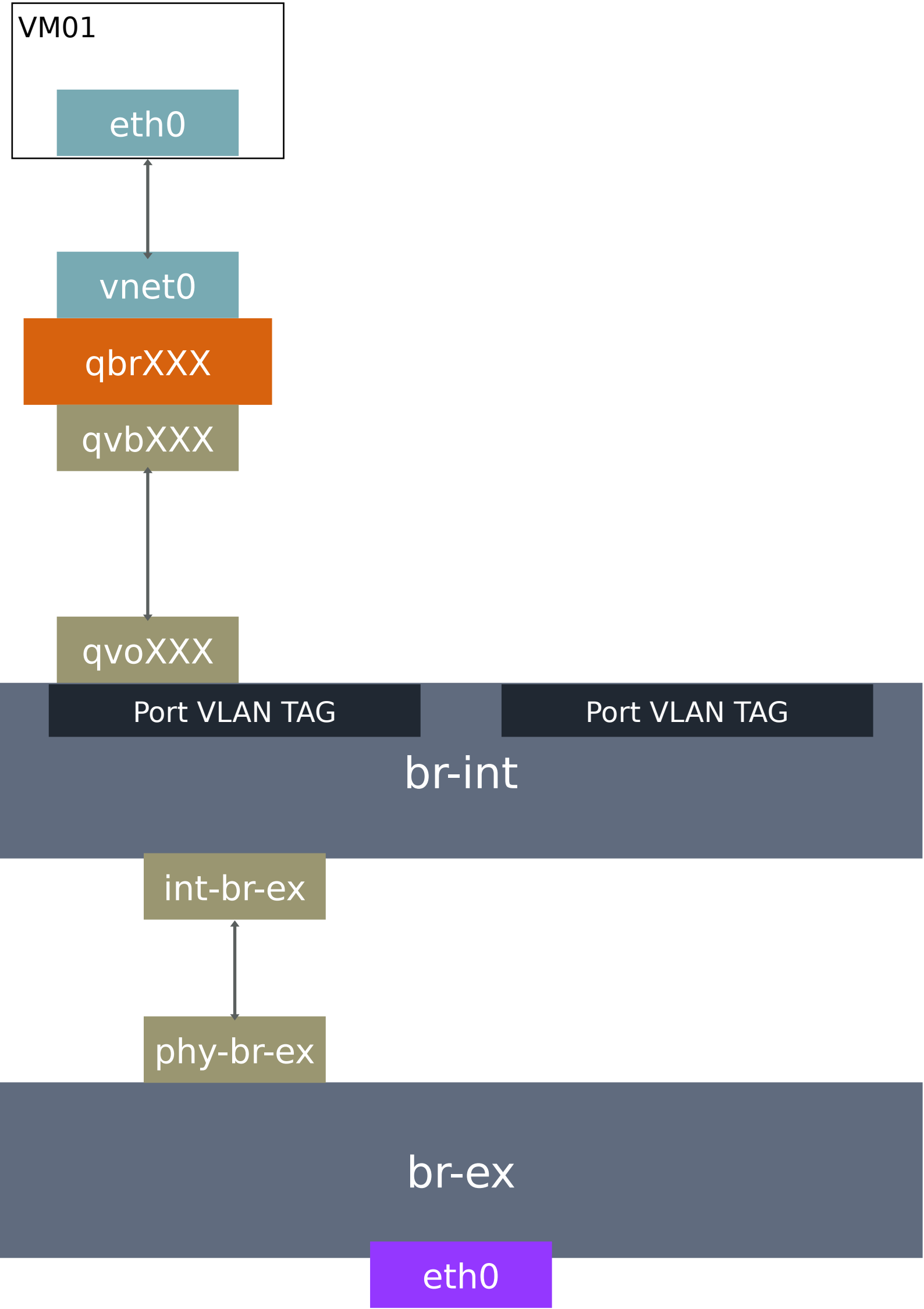
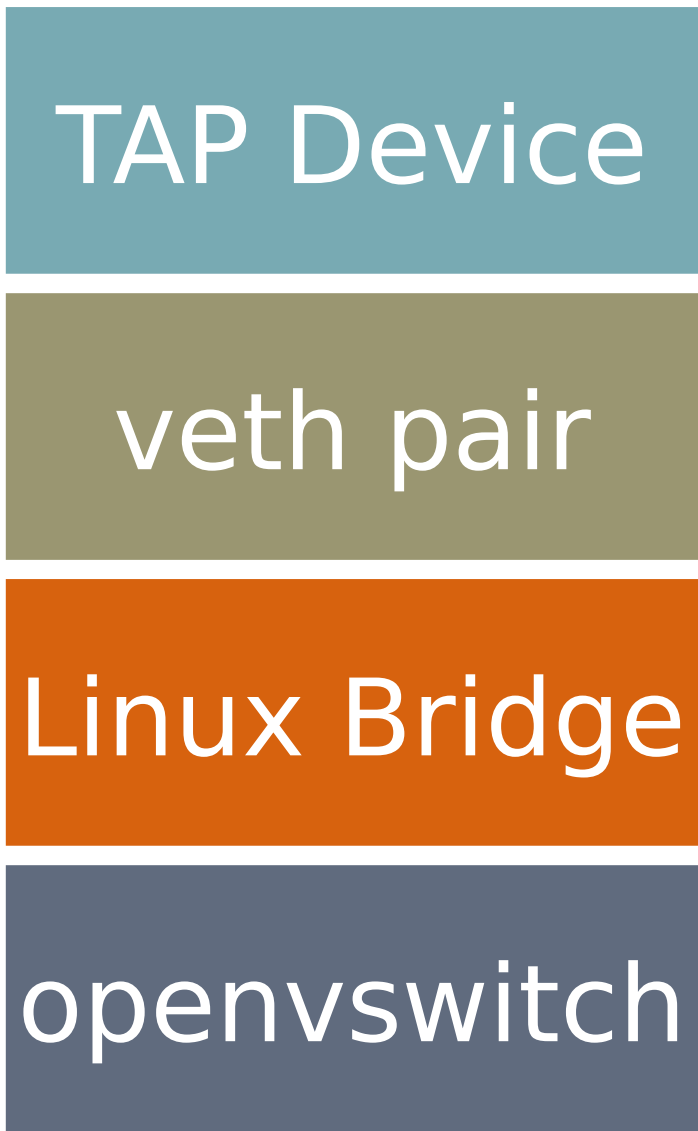


DEEP DIVE into NEUTRON

instance







TAP Device

veth pair

Linux Bridge

openvswitch

TUN (namely network TUNnel) simulates a network layer device and it operates with layer 3 packets like IP packets. **TAP** (namely network tap) simulates a **link layer** device and it operates with layer 2 packets like Ethernet frames. TUN is used with routing, while TAP is used for creating a **network bridge**.

A TAP device, such as vnet0 is how hypervisors such as KVM and Xen implement a virtual network interface card (typically called a VIF or vNIC). An Ethernet frame sent to a TAP device is received by the guest operating system.

TAP Device

veth pair

Linux Bridge

openvswitch

A veth pair is a pair of directly connected virtual network interfaces. An Ethernet frame sent to one end of a veth pair is received by the other end of a veth pair. Networking uses veth pairs as **virtual patch cables** to make connections between virtual bridges

TAP Device

veth pair

Linux Bridge

openvswitch

A Linux bridge behaves like a simple **MAC learning switch**: you can connect multiple (physical or virtual) network interfaces devices to a Linux bridge. The Linux bridge uses a MAC caching table to record which interface on the bridge is used to communicate with a host on the link. For any Ethernet frames that come in from one interface attached to the bridge, the host MAC address and port on which the frame was received is recorded in the MAC caching table for a limited time. When the bridge needs to forward a frame, it will check to see if the frame's destination **MAC address is recorded** in the table. If so, the Linux bridge **forwards the frame through only that port. If not**, the frame is **flooded** to all network ports in the bridge, with the exception of the port where the frame was received.

TAP Device

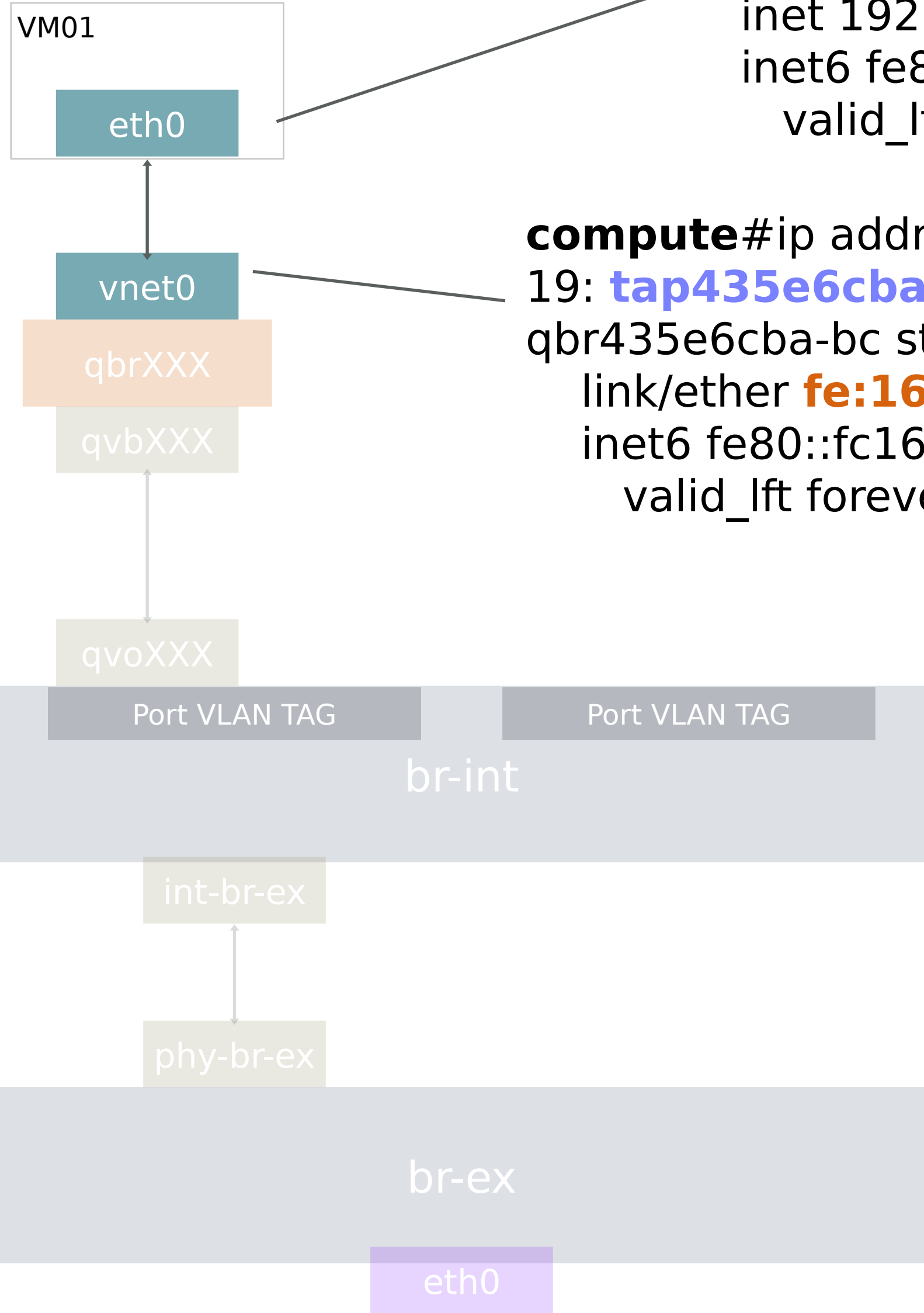
veth pair

Linux Bridge

openvswitch

An Open vSwitch bridge behaves like a **virtual switch:**
network interface devices connect to Open vSwitch
bridge's ports, and the ports can be configured much like
a physical switch's ports, including VLAN configurations.

TAP Device

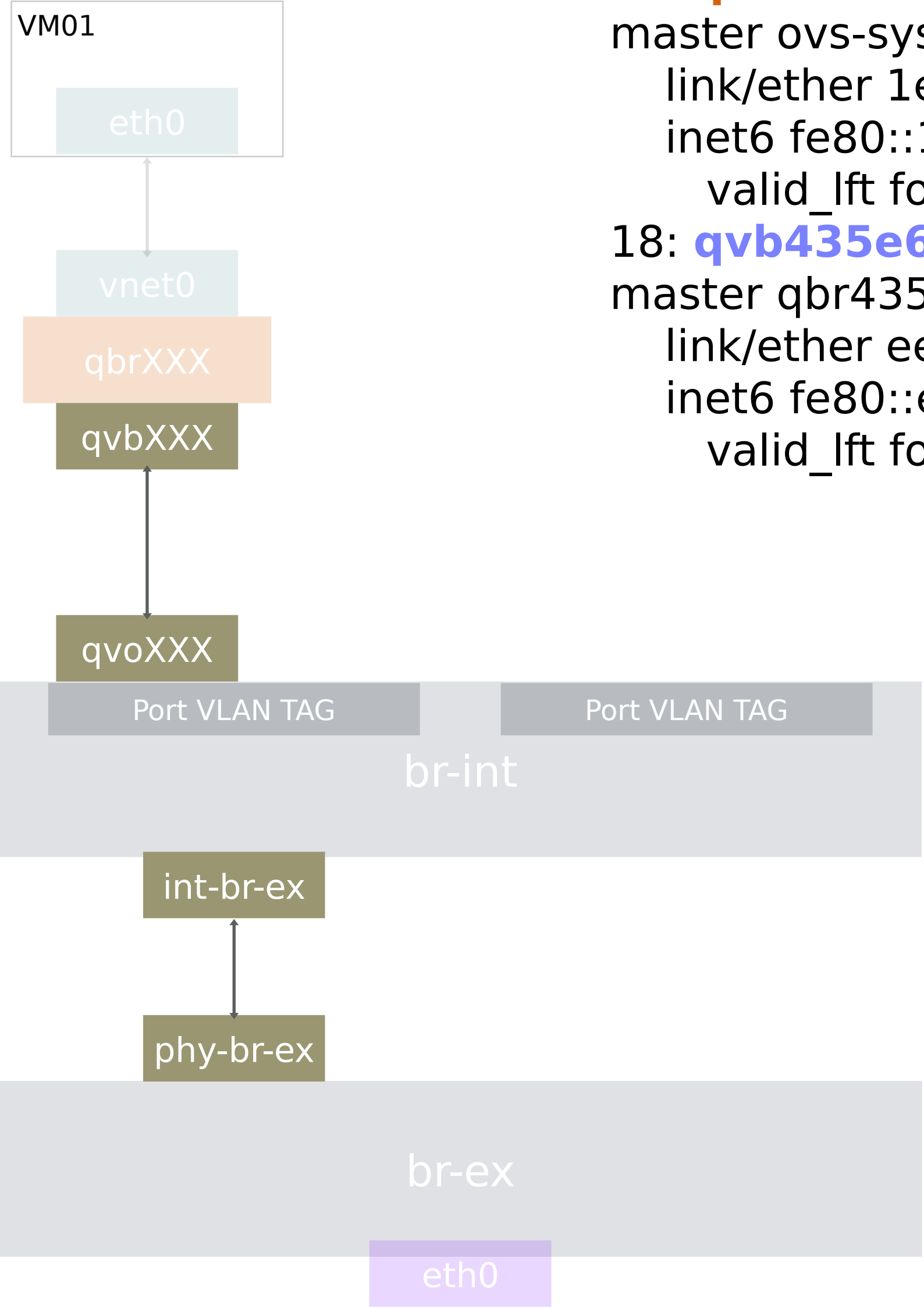


instance#ip addr
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
link/ether **fa:16:3e:42:11:db** brd ff:ff:ff:ff:ff:ff
inet 192.168.64.2/24 brd 192.168.64.255 scope global eth0
inet6 fe80::f816:3eff:fe42:11db/64 scope link
valid_lft forever preferred_lft forever

compute#ip addr
19: **tap435e6cba-bc**: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master
qbr435e6cba-bc state UNKNOWN qlen 500
link/ether **fe:16:3e:42:11:db** brd ff:ff:ff:ff:ff:ff
inet6 fe80::fc16:3eff:fe42:11db/64 scope link
valid_lft forever preferred_lft forever

compute#cat /etc/libvirt/qemu/instance-00000001.xml
.
.
.
<interface type='bridge'>
 <mac address='fa:16:3e:42:11:db'/>
 <source bridge='qbr435e6cba-bc'/>
 <target dev='tap435e6cba-bc'/>
 <model type='virtio'/>
 <driver name='qemu'/>
 <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
function='0x0'/>
</interface>
.
.
.

veth pair

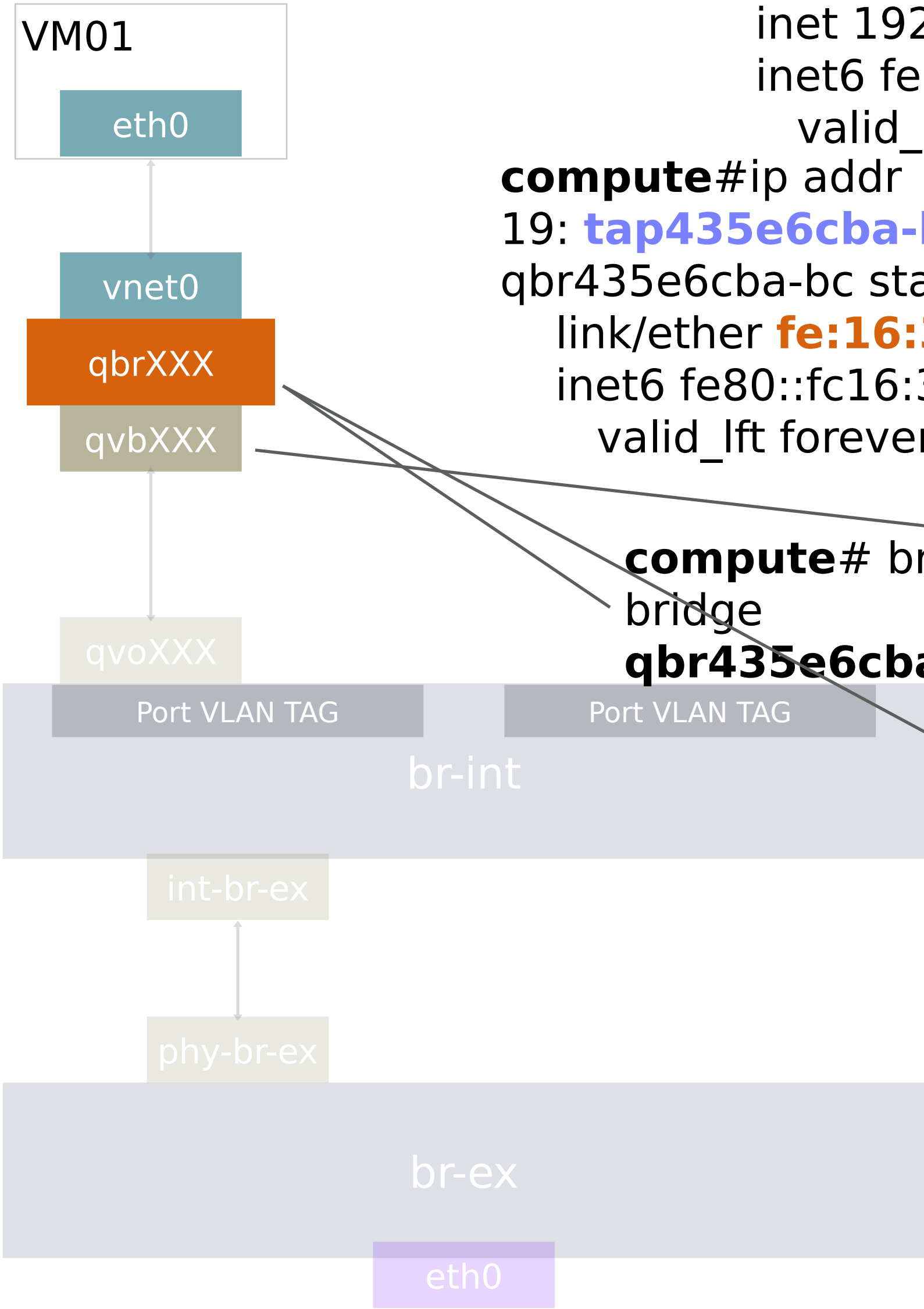


```
compute#ip addr
17: qvo435e6cba-bc: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master ovs-system state UP qlen 1000
    link/ether 1e:8d:cf:23:97:5a brd ff:ff:ff:ff:ff:ff
    inet6 fe80::1c8d:cfff:fe23:975a/64 scope link
        valid_lft forever preferred_lft forever
18: qvb435e6cba-bc: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master qbr435e6cba-bc state UP qlen 1000
    link/ether ee:64:c7:31:17:69 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::ec64:c7ff:fe31:1769/64 scope link
        valid_lft forever preferred_lft forever
```

```
compute# ethtool -S qvo435e6cba-bc
NIC statistics:
    peer_ifindex: 18
```

```
compute# ethtool -S qvb435e6cba-bc
NIC statistics:
    peer_ifindex: 17
```

Linux Bridge



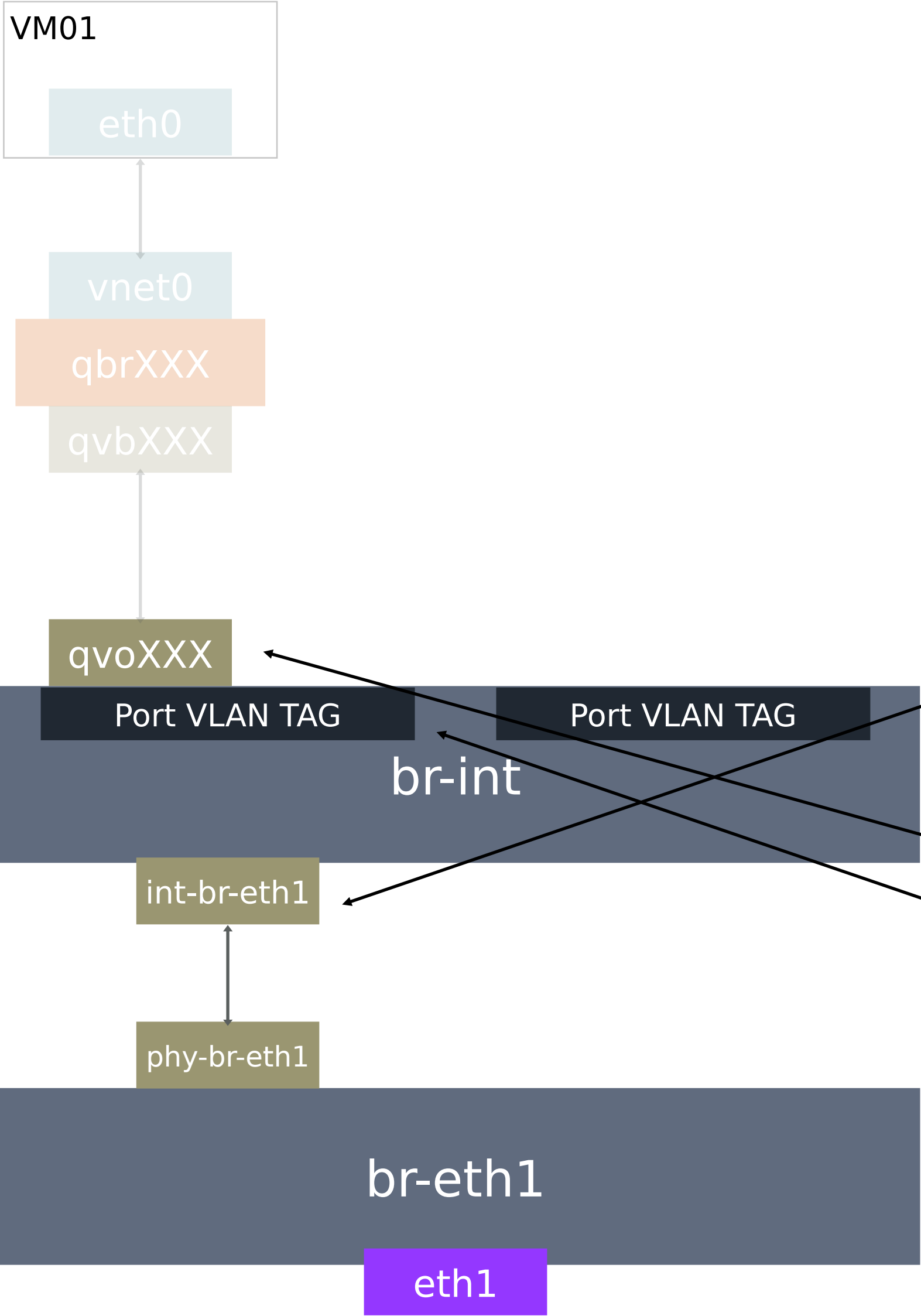
```
instance#ip addr
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether fa:16:3e:42:11:db brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.2/24 brd 192.168.64.255 scope global eth0
    inet6 fe80::f816:3eff:fe42:11db/64 scope link
    valid_lft forever preferred_lft forever
```

```
compute#ip addr
19: tap435e6cba-bc: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master
qbr435e6cba-bc state UNKNOWN qlen 500
    link/ether fe:16:3e:42:11:db brd ff:ff:ff:ff:ff:ff
    inet6 fe80::fc16:3eff:fe42:11db/64 scope link
    valid_lft forever preferred_lft forever
```

```
compute# brctl show
bridge                name      bridge id        STP enabled  interfaces
qbr435e6cba-bc        8000.ee64c7311769 no            qvb435e6cba-bc
tap435e6cba-bc
```

```
compute#cat /etc/libvirt/qemu/instance-000000001.xml
.
.
.
<interface type='bridge'>
  <mac address='fa:16:3e:42:11:db'/>
  <source bridge='qbr435e6cba-bc'/>
  <target dev='tap435e6cba-bc'/>
  <model type='virtio'/>
  <driver name='qemu'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
function='0x0'/>
</interface>
```

openvswitch



```
compute# ovs-vsctl show
d766ab40-b367-407d-ac98-a733be43c3cc
Bridge "br-eth1"
  Port "phy-br-eth1"
    Interface "phy-br-eth1"
  Port "br-eth1"
    Interface "br-eth1"
    type: internal
Bridge br-int
  fail_mode: secure
  Port br-int
    Interface br-int
    type: internal
  Port "int-br-eth1"
    Interface "int-br-eth1"
  Port patch-tun
    Interface patch-tun
    type: patch
    options: {peer=patch-int}
  Port "tapbf360721-11"
    tag: 1
    Interface "tapbf360721-11"
    type: internal
  Port "qvo435e6cba-bc"
    tag: 1
    Interface "qvo435e6cba-bc"
  Port "qr-8a679f69-7d"
    tag: 1
    Interface "qr-8a679f69-7d"
    type: internal
```

```
# ip netns
qdhcp-6af79125-63cb-47cc-b8af-7981e860a73d
qrouter-4cfd7137-86c2-4855-a81a-8a5cf37a85e7
```

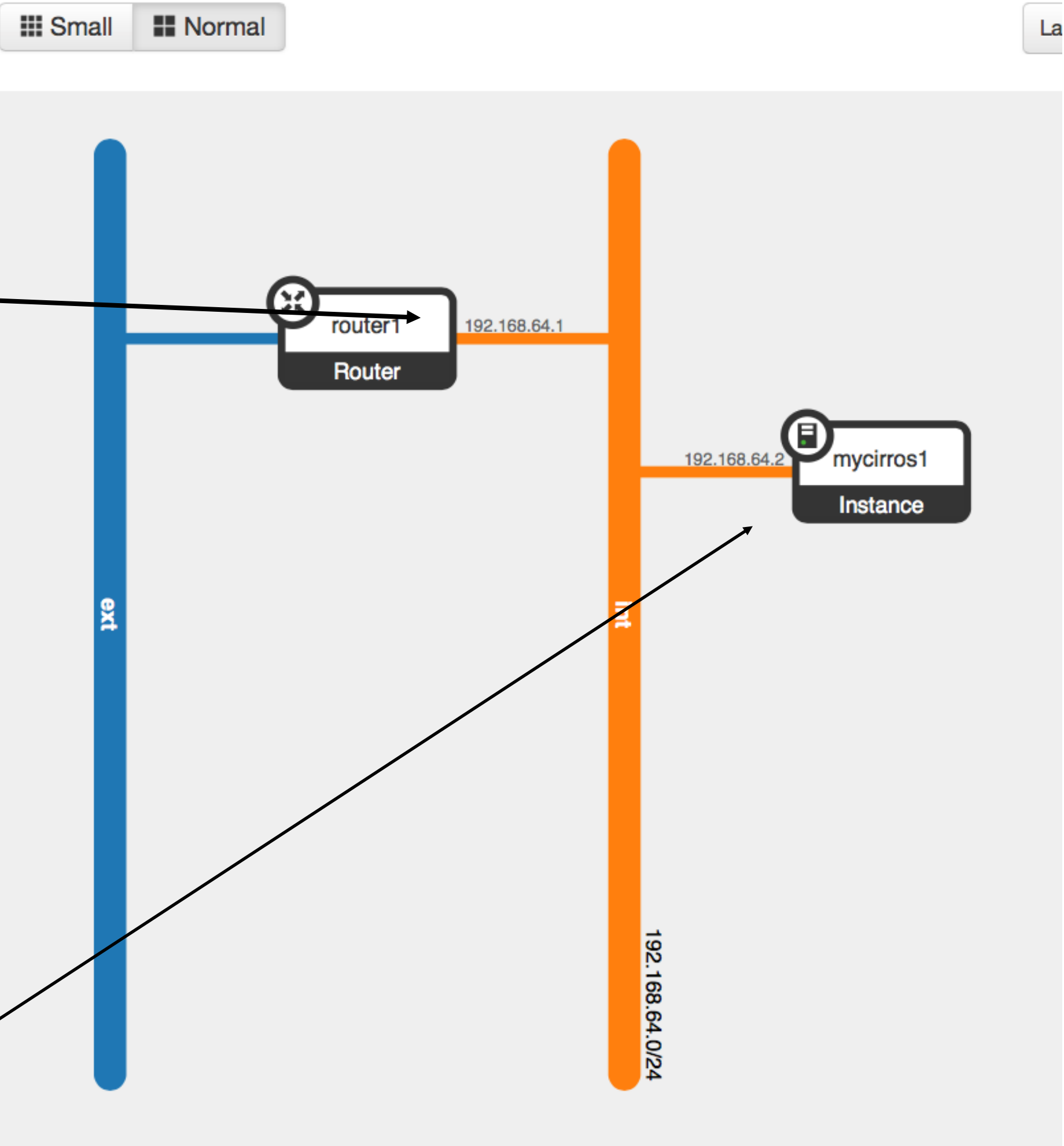
```
# ip netns exec qrouter-4cfd7137-86c2-4855-a81a-8a5cf37a85e7 ip addr
```

```
13: qr-8a679f69-7d: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:91:60:f5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.1/24 brd 192.168.64.255 scope global qr-8a679f69-7d
        valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe91:60f5/64 scope link
        valid_lft forever preferred_lft forever
14: qg-951729dd-88: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:8e:2f:76 brd ff:ff:ff:ff:ff:ff
    inet 192.168.32.50/24 brd 192.168.32.255 scope global qg-951729dd-88
        valid_lft forever preferred_lft forever
    inet 192.168.32.51/32 brd 192.168.32.51 scope global qg-951729dd-88
        valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe8e:2f76/64 scope link
        valid_lft forever preferred_lft forever
```

```
compute# ping 192.168.64.2
PING 192.168.64.2 (192.168.64.2) 56(84) bytes of data.
^C
--- 192.168.64.2 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 3999ms
```

```
compute# ip netns exec qrouter-4cfd7137-86c2-4855-a81a-8a5cf37a85e7 ping 192.168.64.2
PING 192.168.64.2 (192.168.64.2) 56(84) bytes of data.
64 bytes from 192.168.64.2: icmp_seq=1 ttl=64 time=7.95 ms
64 bytes from 192.168.64.2: icmp_seq=2 ttl=64 time=1.30 ms
64 bytes from 192.168.64.2: icmp_seq=3 ttl=64 time=0.984 ms
^C
--- 192.168.64.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.984/3.414/7.955/3.213 ms
```

Network Topology



Floating IPs

	IP Address	Instance	Floating IP Pool
<input type="checkbox"/>	192.168.32.51	mycirros1	ext

THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos