Neutron Deep Dive

Attila Szlovencsák
Component Soft Ltd.

attila@componentsoft.eu

Openstack CEE Day 2015

June 8, 2015

Outline

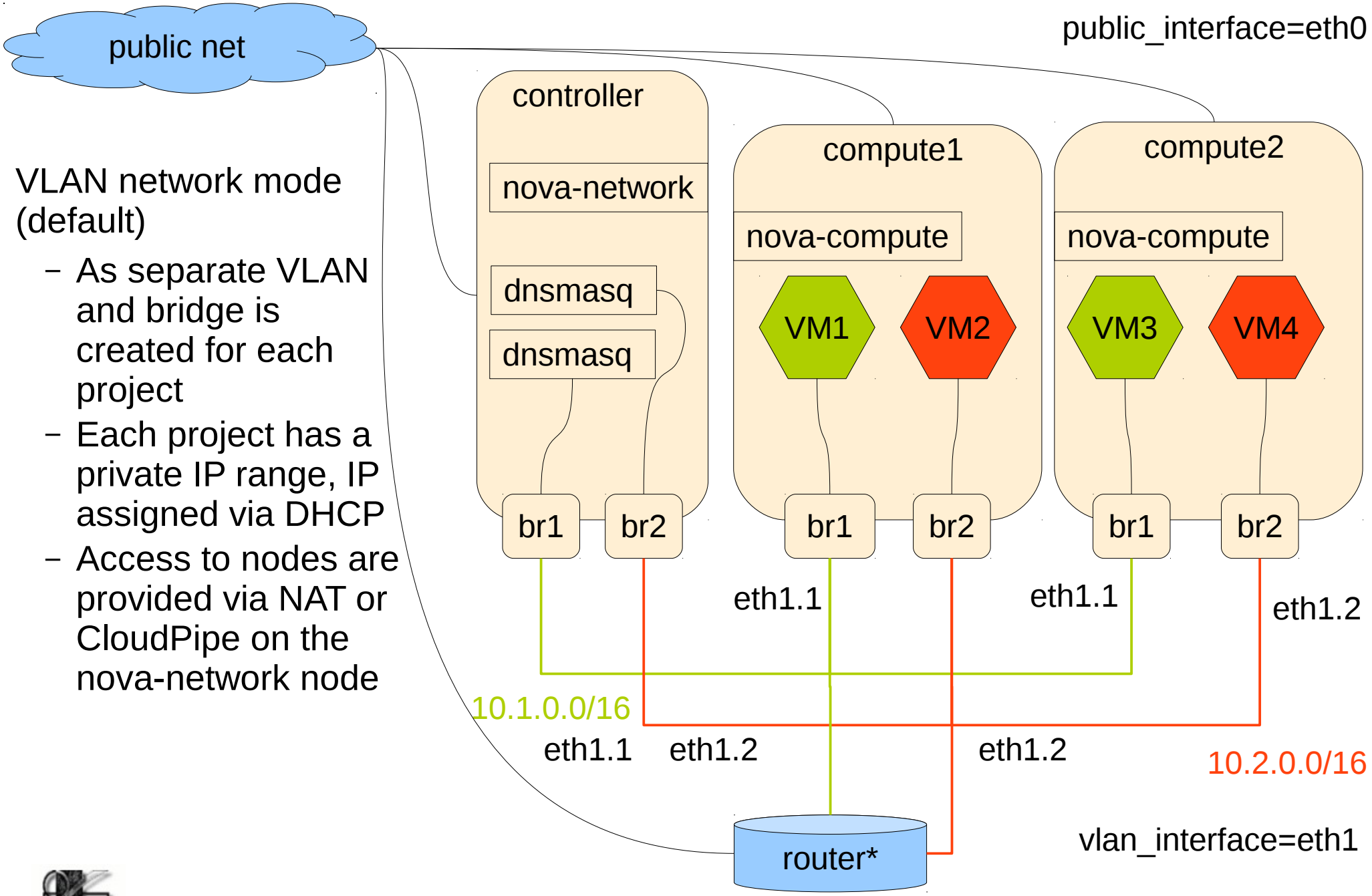
- Openstack networking terms
- Pre-Grizzly networking
- Networking with Neutron
  - LinuxBridge/OVSPlugin
  - ML2Plugin
- ML2 features
  - L2Population
  - L3 HA, DVR
- Use-cases
  - Virtual IPs
  - Ipv6

OpenStack Networking Concepts

- Network:
  - an isolated L2 segment dedicated for tenant (or shared between tenants)
  - subnet: a block of ipv4/6 addresses. More than one allowed for a network
  - external flag (neutron only): “public net”
- Port:
  - A connection point for attaching a single device (NIC of a virtual server) to a virtual network. Port has
    - a fixed address (via DHCP or injected), MAC
    - has an associated security group
    - may have an associated floating address
- Security group:
  - Group of L4 filter rules (no access granted with the “default” group)
- Floating address:
  - an IP address from one 'external' network, associated with a port(fixed address)
  - could be moved between VMs
- Router (neutron only):
  - L3 device that connects the 'external' and the tenant networks.

History – nova-network (pre-Grizzly)

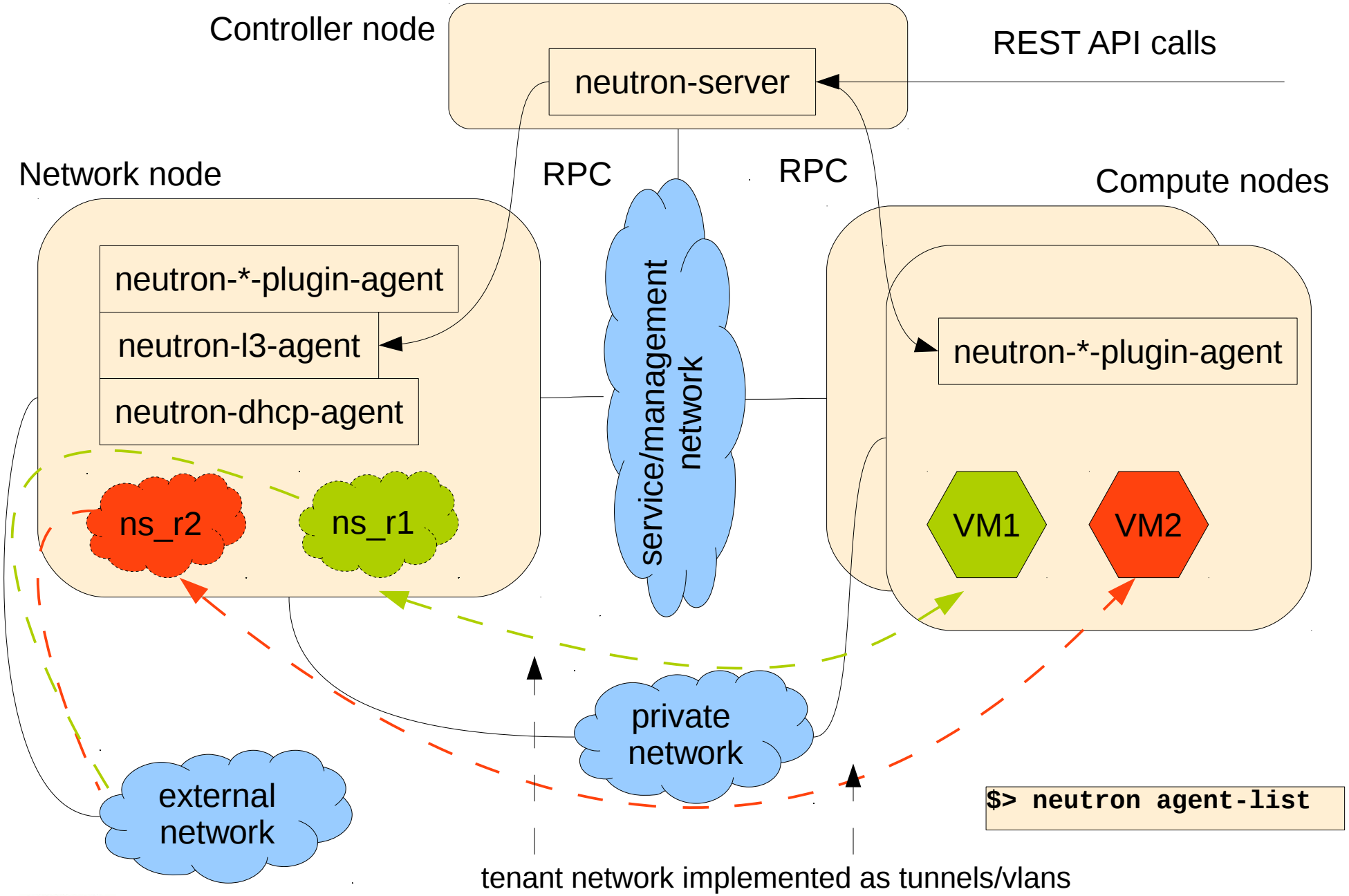
- VLAN network mode (default)
  - As separate VLAN and bridge is created for each project
  - Each project has a private IP range, IP assigned via DHCP
  - Access to nodes are provided via NAT or CloudPipe on the nova-network node



Why neutron? (quantum)

- problems
  - out of VLAN IDs (12 bit,~4096)
    - double VLAN might help
  - for VLAN Compute nodes need to be on the same L2 segment
    - use VLAN trunking with multiple switches
  - Two tenants cannot use the same address space for private net
    - NAT, routing might be complicated
- solution
  - have a separate node (network node), that acts as
    - DHCP server, router for all tenant networks
    - host for floating IPs
    - host for security groups
  - allows
    - Other tenant network isolations (GRE, VxVLAN (with ML2plugin))
    - Complex tenant networks (SDN, overlapping networks)
    - Backward compatibility (with LinuxBridge plugin)

Networking with Neutron

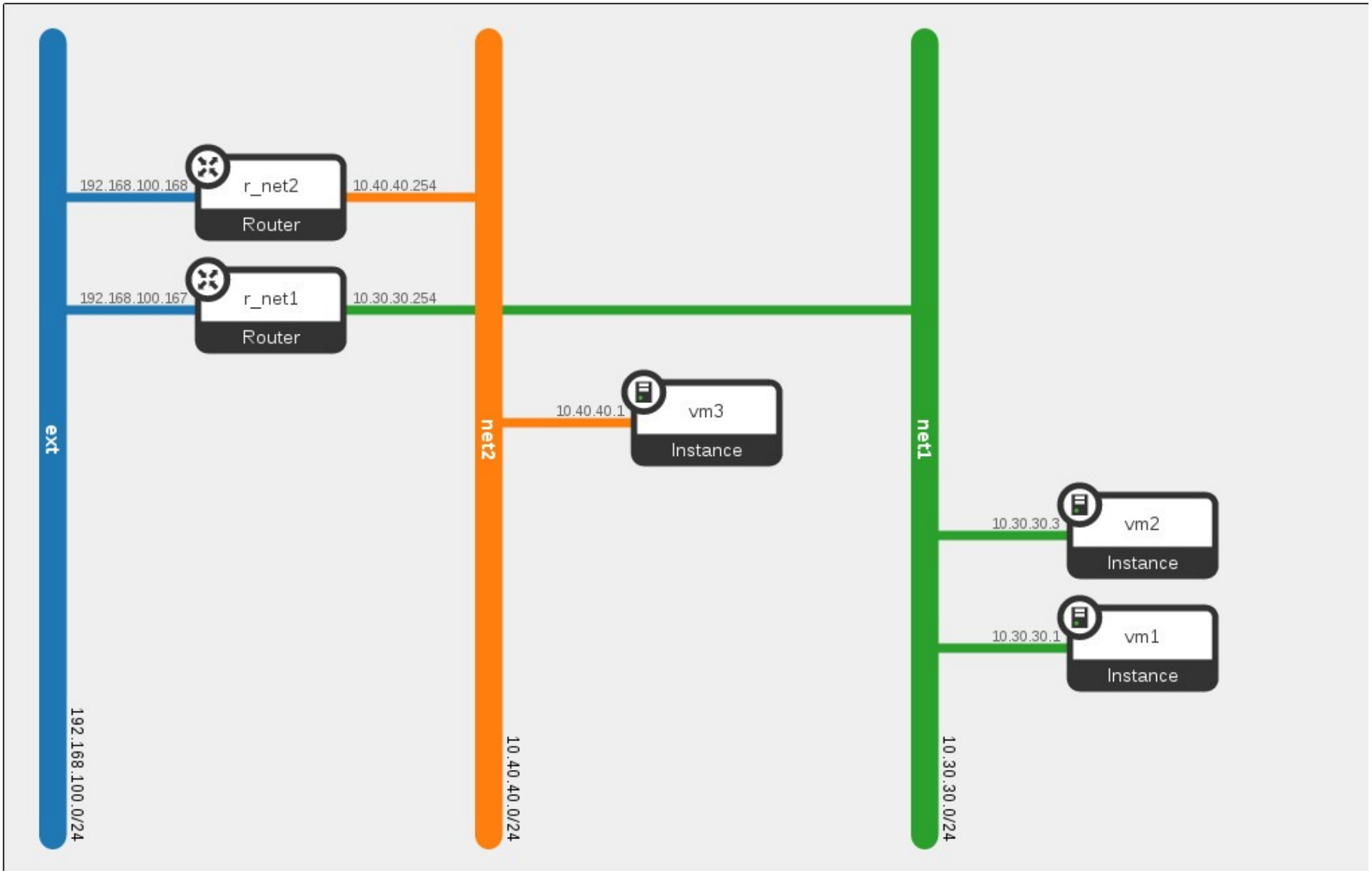


The ML2plugin

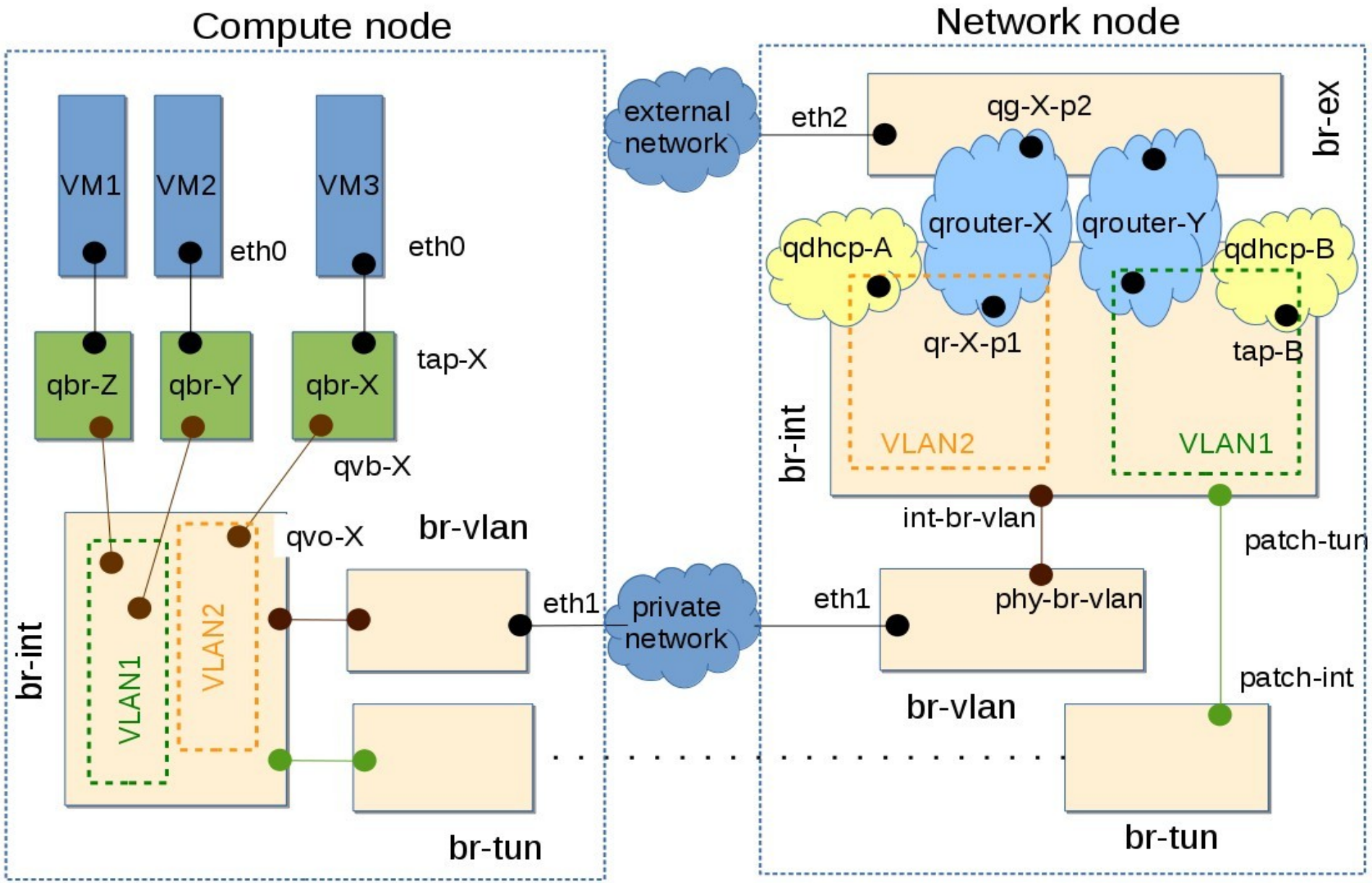
Why?

- Replace the monolithic plugins
  - Eliminate redundant code when developing a new plugin
- Drivers per functionality
  - type drivers: the “network isolation” type
    - Flat, GRE,VLAN, VXLAN
  - mechanism drivers:The way to manage your networks
    - reuse existing plugins: OVSPlugin, LinuxBridgePlugin
      - going to deprecate -> Modular Agent: combine OVS and LinuxBridge functionality
    - optional features
      - L2Population: avoids broadcast flooding via APR responder
    - support for heterogenous deployments (use OVS for a set of compute nodes, LinuxBridge for others)
    - l3router : moved to a service plugin (such as other *aaS plugins)

OVSNeutronPlugin – Example topology

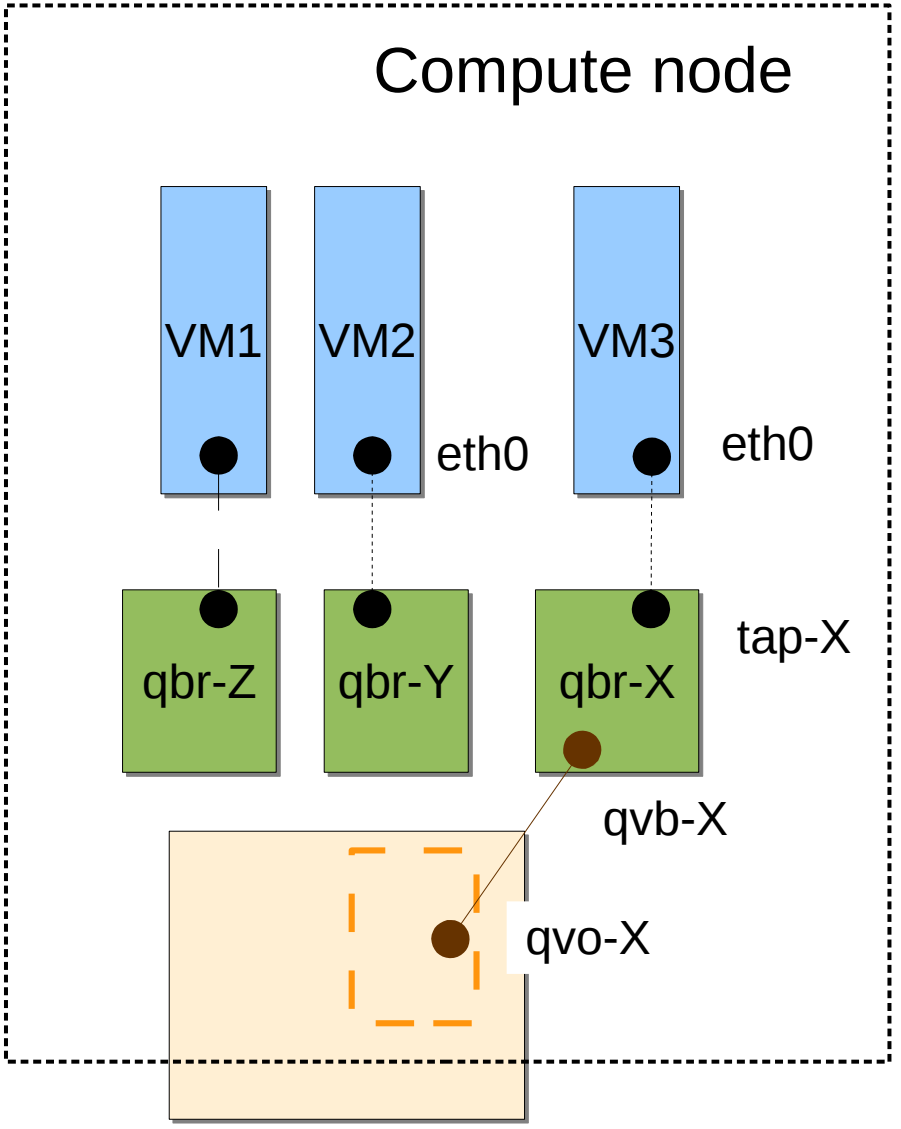


OVSNeutronPlugin – Physical layout



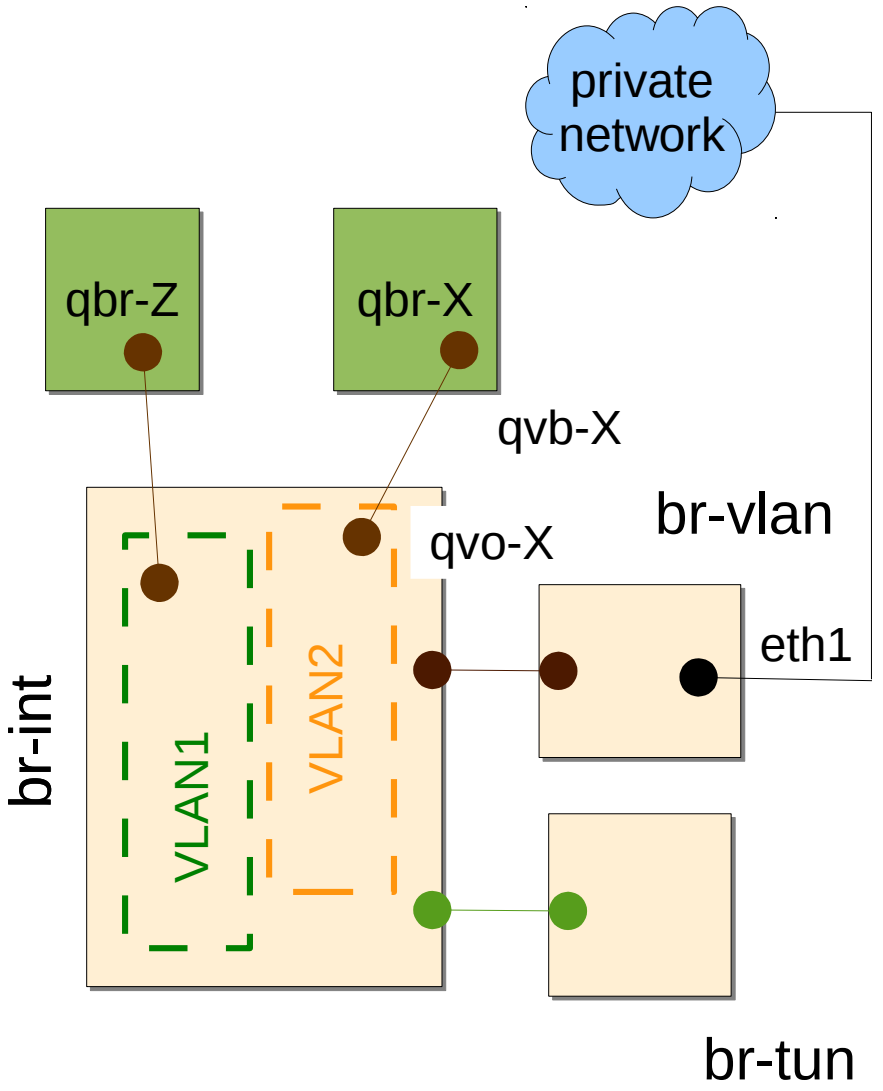
OVS layout - Compute node

- The interface of the VM is plugged into a dedicated Linux bridge
  - needed for security groups
    - goo.gl/UyC5UL
  - might be removed in the future
- The dedicated bridge is connected to the integration bridge
  - veth pair (qvb-X, qvo-X)
  - qvo-X is tagged for a tenant-specific virtual LAN



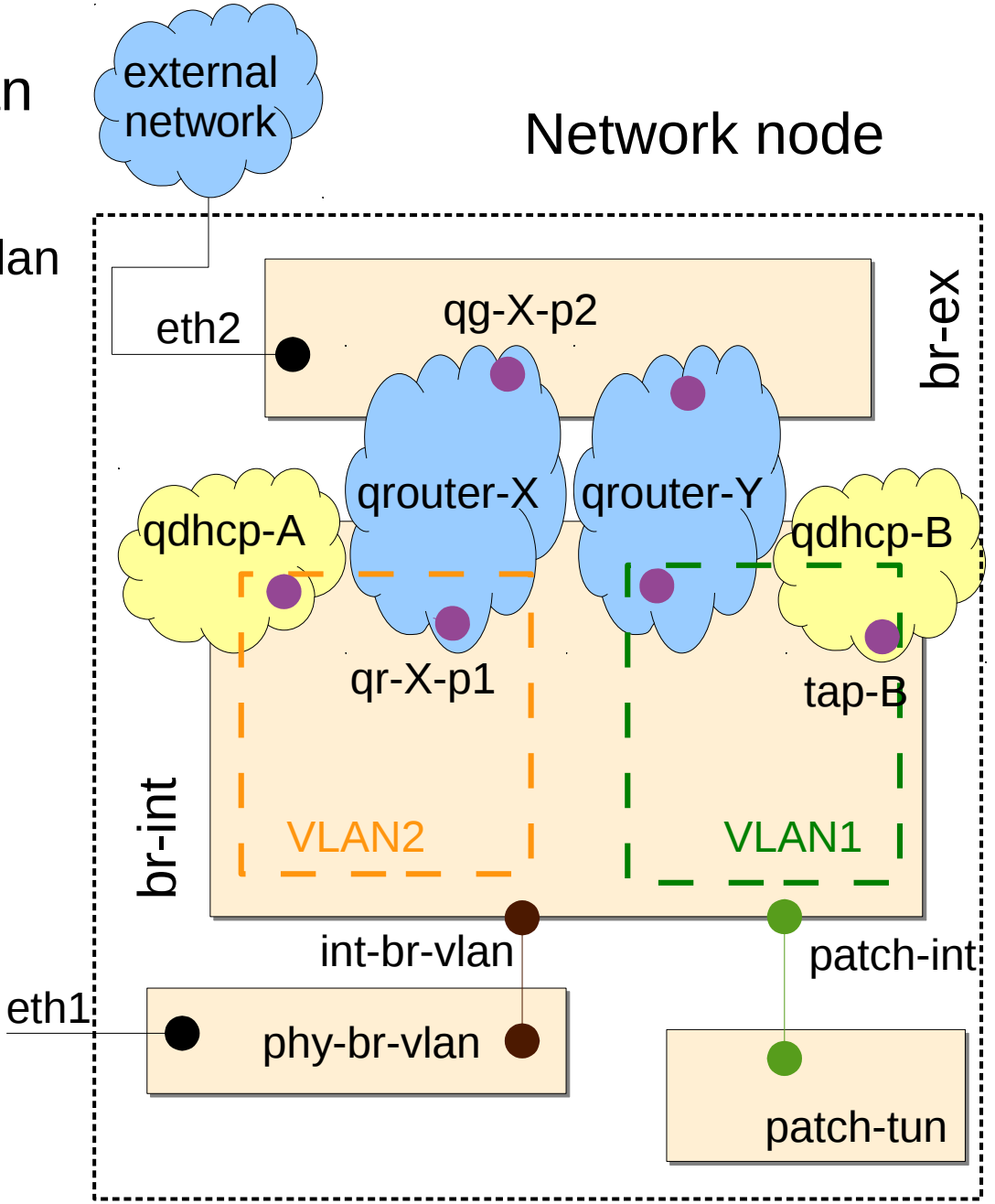
OVS layout - Compute node (2)

- Separated virtual LAN for each tenant network
  - allows VMs to communicate directly without leaving the hypervisor
  - VLAN tags are independent of the segmentation_id
- Depending on the network type
  - For GRE
    - packet enters into br-tun
    - packets are encapsulated and sent to the desired peer
    - broadcast is sent as unicast
  - For 802.1Q (VLAN)
    - the temporary tag is removed
    - the tenant network specific tag is applied
    - packet leaves on eth1



OVS layout - Network node

- Traffic enters into br-tun or br-vlan
- Forwarded to br-int
  - interfaces are tagged to the same vlan
  - router internal interface is on br-int
- Traffic enters into the router namespace (qrouter-X)
  - packets forwarded towards the external interface
  - address translation
    - with or without floating IP
- Packets leave the router
  - the router external interface is on br-ex
  - br-ex has the external physical interface (eth2)



OVSPlugin configuration - compute-node

ifcfg-eth1

```
DEVICE=eth1
ONBOOT=yes
OVS_BRIDGE=br-eth1
DEVICETYPE=ovs
TYPE="OVSPort"
```

ifcfg-br-eth1

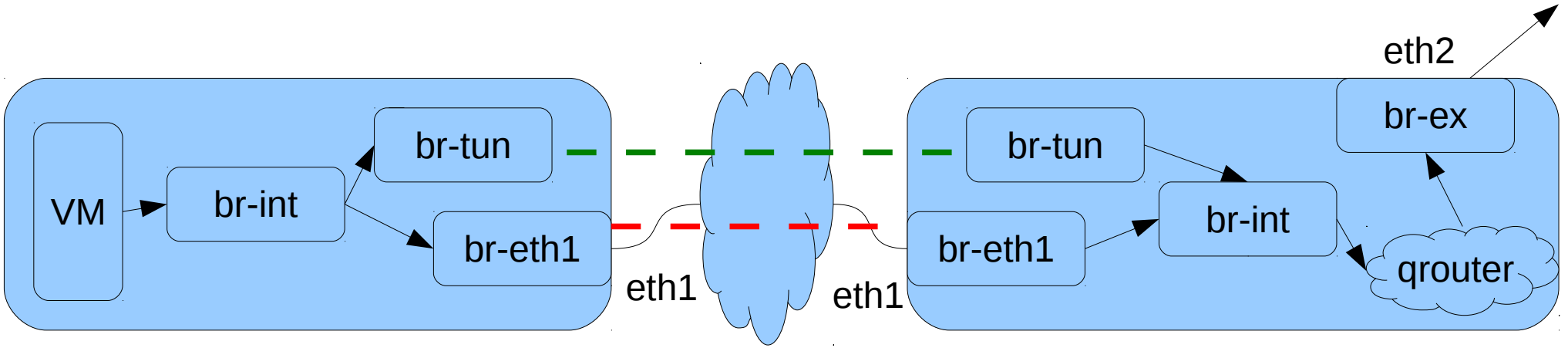
```
DEVICE=br-eth1
IPADDR=10.20.20.53
NETMASK=255.255.255.0
ONBOOT=yes
DEVICETYPE=ovs
TYPE="OVSBridge"
```

/etc/neutron/plugin.ini

```
local_ip=10.20.20.53
integration_bridge=br-int
tunnel_bridge=br-tun
bridge_mappings=vlan_if:br-eth1
```

/etc/neutron/neutron.conf

```
core_plugin=neutron.plugins.openvswitch
.ovs_neutron_plugin.OVSNeutronPluginV2
service_plugins=
```



OVS configuration – network node

ifcfg-eth2

```
DEVICE=eth2
ONBOOT=yes
OVS_BRIDGE=br-ex
DEVICETYPE=ovs
TYPE="OVSPort"
```

ifcfg-eth1

```
DEVICE=eth1
ONBOOT=yes
OVS_BRIDGE=br-eth1
DEVICETYPE=ovs
TYPE="OVSPort"
```

ifcfg-br-eth1

```
DEVICE=br-eth1
IPADDR=10.20.20.52
NETMASK=255.255.255.0
ONBOOT=yes
DEVICETYPE=ovs
TYPE="OVSBridge"
```

ifcfg-br-ex

```
DEVICE=br-ex
IPADDR=192.168.100.52
NETMASK=255.255.255.0
ONBOOT=yes
DEVICETYPE=ovs
TYPE="OVSBridge"
```

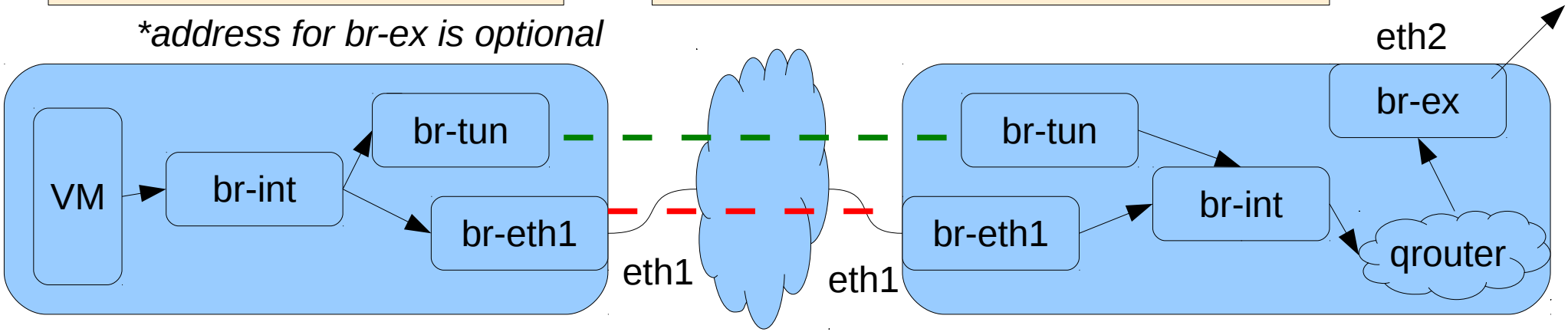
/etc/neutron/plugin.ini

```
local_ip=10.20.20.52
integration_bridge=br-int
tunnel_bridge=br-tun
bridge_mappings=vlan_if:br-eth1
```

/etc/neutron/l3_agent.ini

```
external_network_bridge = br-ex
```

**address for br-ex is optional*



Configuration – OVS+ ML2 (Icehouse)

/etc/neutron/neutron.conf

```
core_plugin =neutron.plugins.ml2.plugin.Ml2Plugin
service_plugins =neutron.services.l3_router.l3_router_plugin.L3RouterPlugin
```

/etc/neutron//etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini

```
[ovs]
enable_tunneling=True
local_ip=10.20.20.53
integration_bridge=br-int
tunnel_bridge=br-tun
bridge_mappings=vlan_if:br-eth1
[agent]
tunnel_type=gre
l2_population=True
```

/etc/neutron/plugins/ml2/ml2_conf.ini

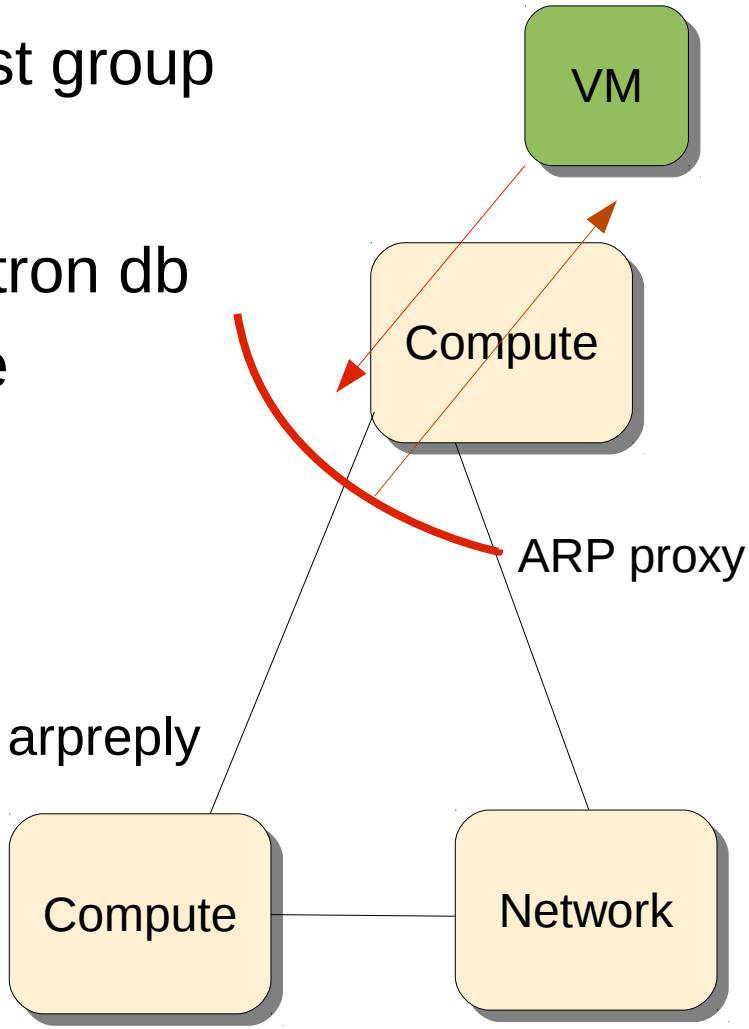
```
[ml2]
type_drivers = gre,vlan,vxlan
tenant_network_types = gre,vlan,vxlan
mechanism_drivers =openvswitch
[ml2_type_flat]
#flat_networks = physnet1,physnet2
[ml2_type_vlan]
network_vlan_ranges =vlan_if:100:200
[ml2_type_gre]
tunnel_id_ranges =100:200
[ml2_type_vxlan]
# vni_ranges =
[securitygroup]
enable_security_group = True
```

ML2 features – L2 population (Icehouse)

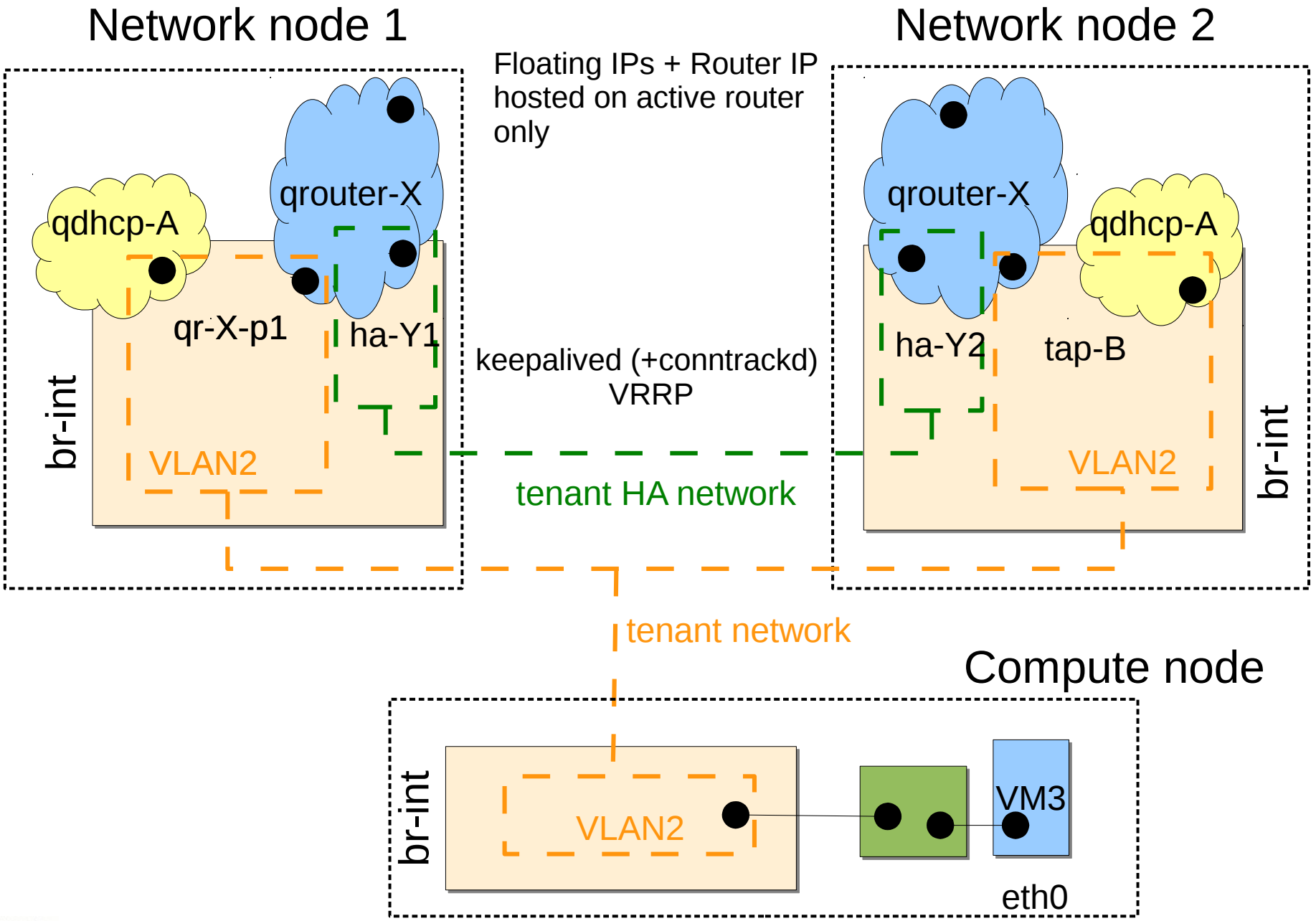
- Network overhead with broadcast/multicast
  - GRE sends an unicast on each tunnel
  - VxVLAN sends to a common multicast group
- avoid ARP broadcasts
  - IP/MAC associations known from neutron db
  - setup a local ARP proxy on each node
    - LinuxBrigde

      ip neighbour add ... permanent
      bridge fdb add
    - OVS

      ebtables -A PREROUTING -t nat ... -j arpreply



Juno features – L3 HA



goo.gl/z4flP3

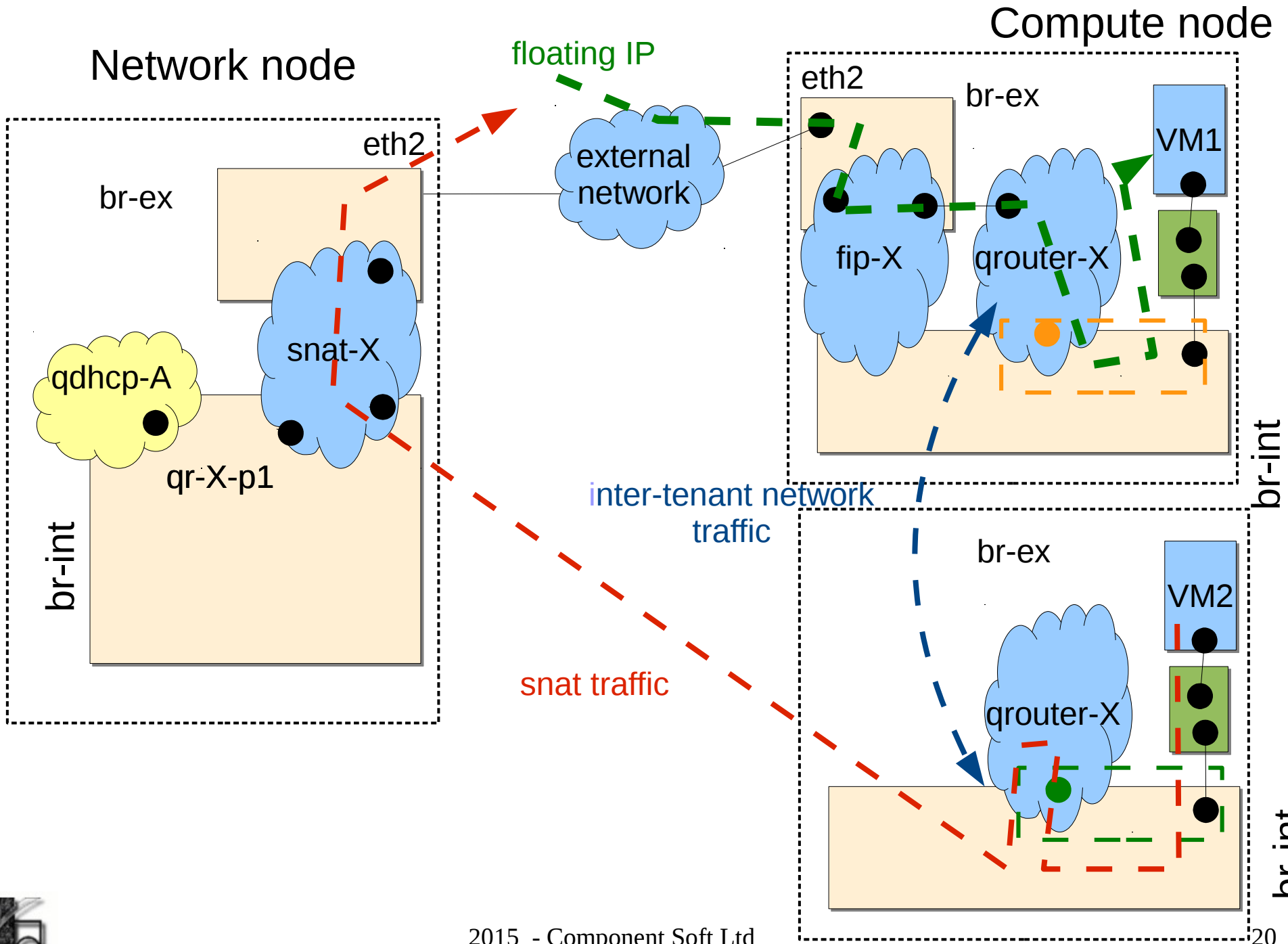
L3 HA facts

- Keepalived
  - processes talk VRRP on a separated network
    - assigned to a virtual tenant
  - VRRP election process
    - hello packet go on multicast 224.0.0.18
    - One byte for VRID in VRRP headers
      - only 255 virtual routers per tenant
    - on 3 missed hellos an election process is started
      - the lower router_id wins
      - failback is done if the original active comes back
    - new active instance sends a gratuitous ARP request
      - switches learn the new port for MAC/IP
- Other *aaS
    - HA for FWaas, LBaaS scheduled to be implemented in Kilo version
    - OS::Neutron::Router supports option 'ha' since Kilo

Juno feature – Distributed Virtual Routing

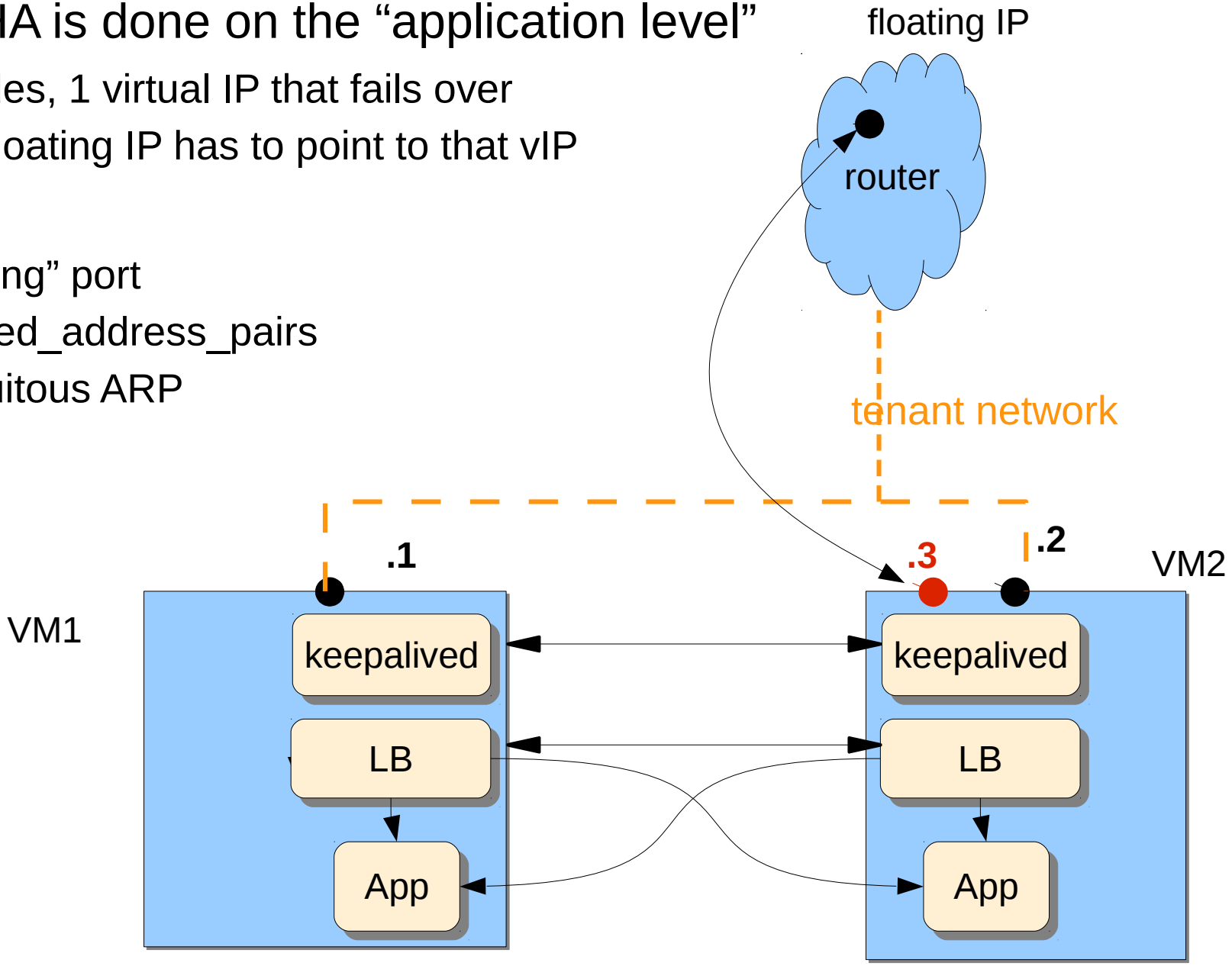
- Motivation
  - Too many network nodes
    - North-South traffic (SNAT/DNAT) has to go throught the a network node
      - We might want to utilize the Compute for DNAT
      - SNAT is still done by the Network node
    - East-West traffic (between two tenant networks)
      - When
        router/firewall can forward between private nets
        VM1 access VM2 via a floating IP
      - We want direct Compute-Compute communcation
- Implementation
  - compute needs access to the external net (br-ex)
  - l3-agent on compute
    - agent-mode=dvr-snat on Network
    - agent-mode=dvr on Compute
    - new namespaces (snat-X on Network, fip-X, qrouter-X on Compute)

Juno features - DVR



Use-case : Virtual IP

- When HA is done on the “application level”
  - 2 nodes, 1 virtual IP that fails over
  - The floating IP has to point to that vIP
- Tricks
  - “floating” port
  - allowed_address_pairs
  - gratuitous ARP



Juno feature – IPv6 support

- IPv6 subnet support, new attributes
  - ipv6-ra-mode, ipv6_address_mode
    - Defines the way how IPv6 addresses are created
      - learn from external routers (not managed by Openstack)
      - learn from radvd and dnsmasq running on the Network node.
    - Values
      - slaac
      - dhcpv6-stateful
      - dhcpv6-stateless
    - Check valid combinations at http://goo.gl/5ObMEx
  - ipv6 subnets are special
    - if a network has both v4 and v6 subnets, a VM port will pick up an address from both
  - Heat support
    - OS::Neutron::Subnet supports ipv6-*-mode since Kilo

Thank you!