

# L3 ROUTING TO HYPERVISOR

VINCENT BERNAT — EXOSCALE

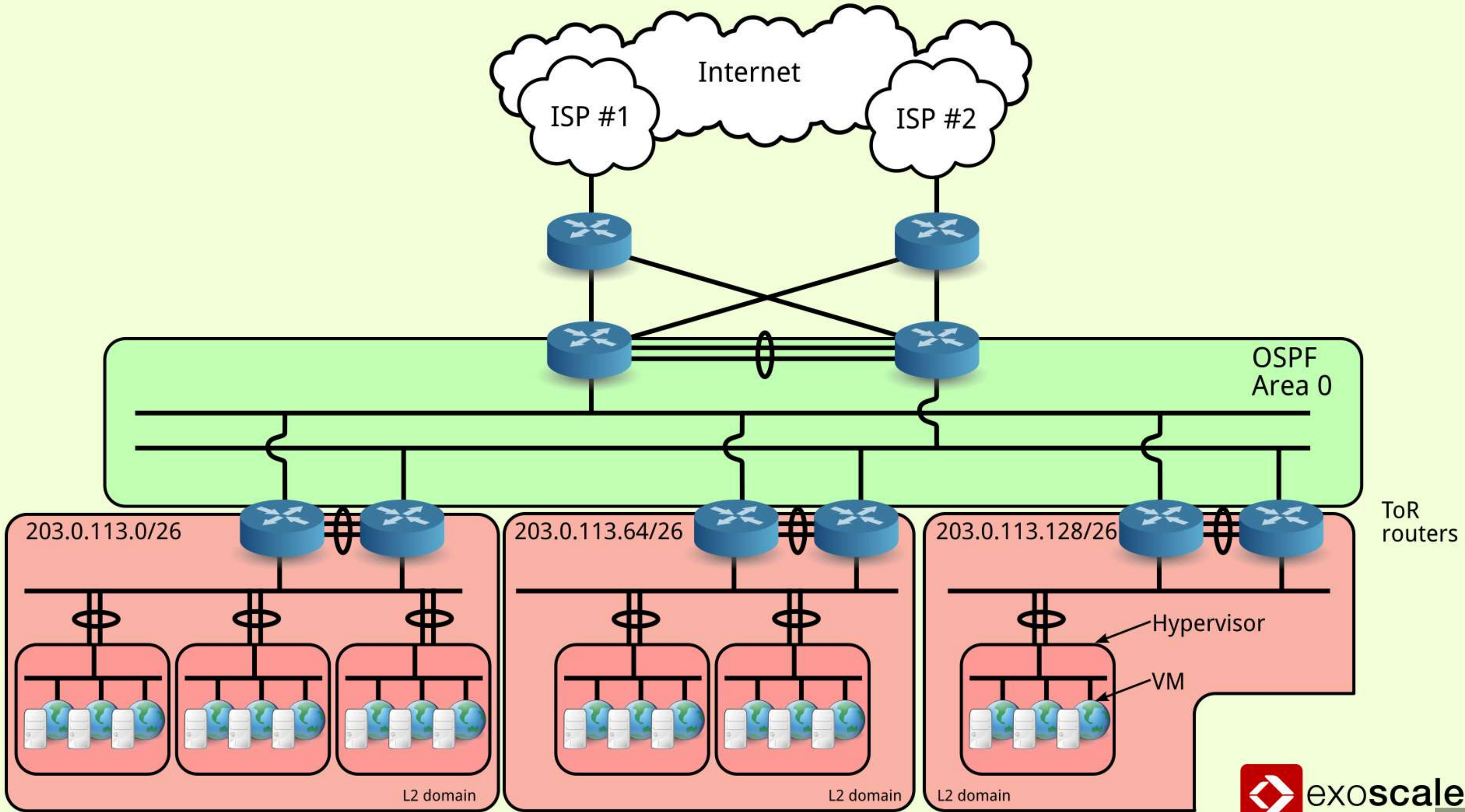
SWINOG 31 — BERN — 2017-05-30

# EXOSCALE

- “Simple cloud computing for cloud native teams”
- Infrastructure provider based on CloudStack
- Two datacenters in Switzerland (Geneva and Zurich)
- Try yourself at [www.exoscale.ch/register](http://www.exoscale.ch/register) with **SWINOG31** coupon code

# LEGACY NETWORK

Initially, our network used a very classic design.

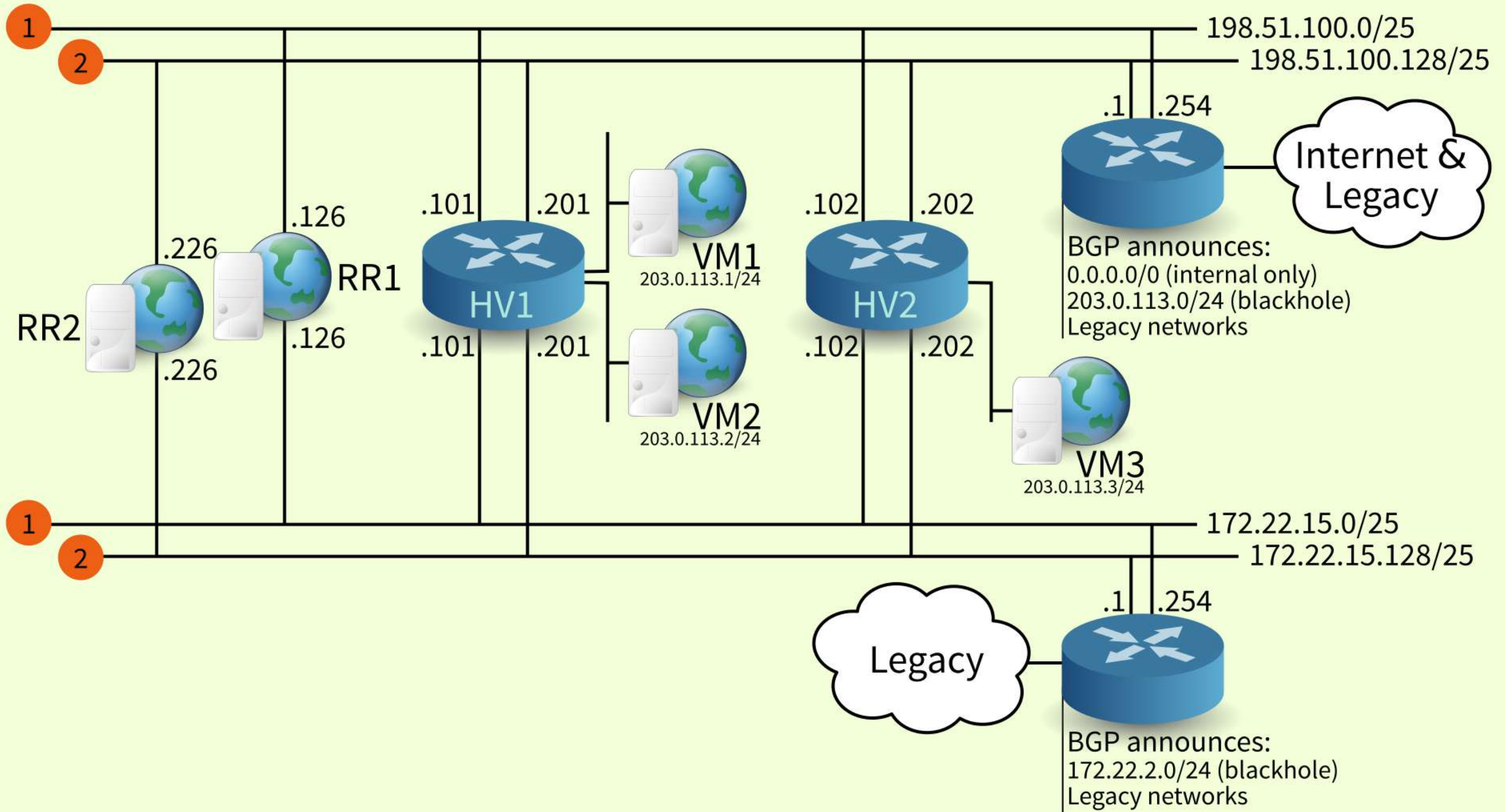


# PROBLEMS

- An **L2 failure** impacts a whole rack (1000+ VMs)
- Many **IPs unused** in each rack
- Cannot **migrate** a VM from one rack to another

# OUR SOLUTION

- ~~Datacenter wide L2 network~~
- L3 routing to the hypervisor with BGP as a control plane



## L2 “LEAF AND SPINE FABRIC”

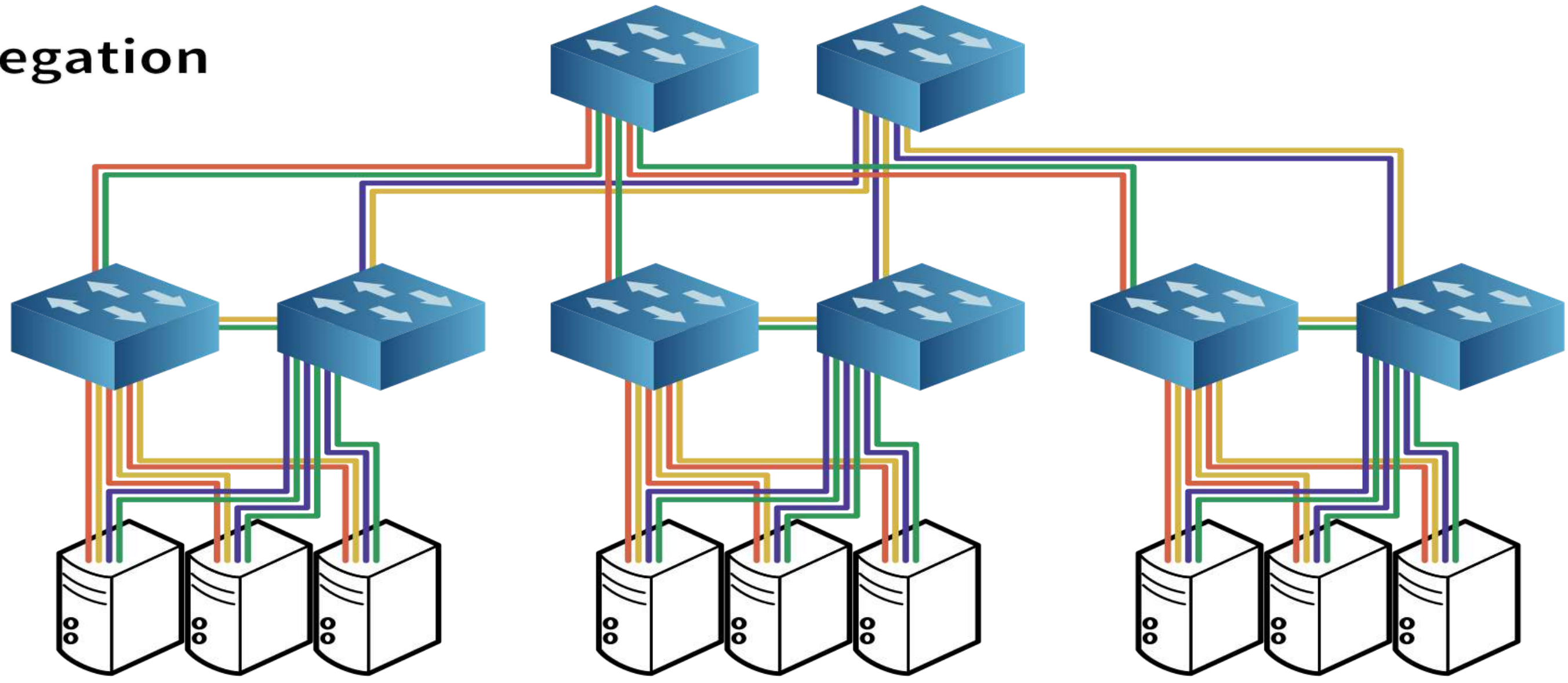
- Distinct L2 networks
- Loop-free
- Simple
- Low cost (“dumb switches”)
- Nonetheless scalable
- Actual design involves using some VLANs to leverage interconnection between ToR switches of the same rack and sustain loss of several links



# Aggregation

ToR

HV

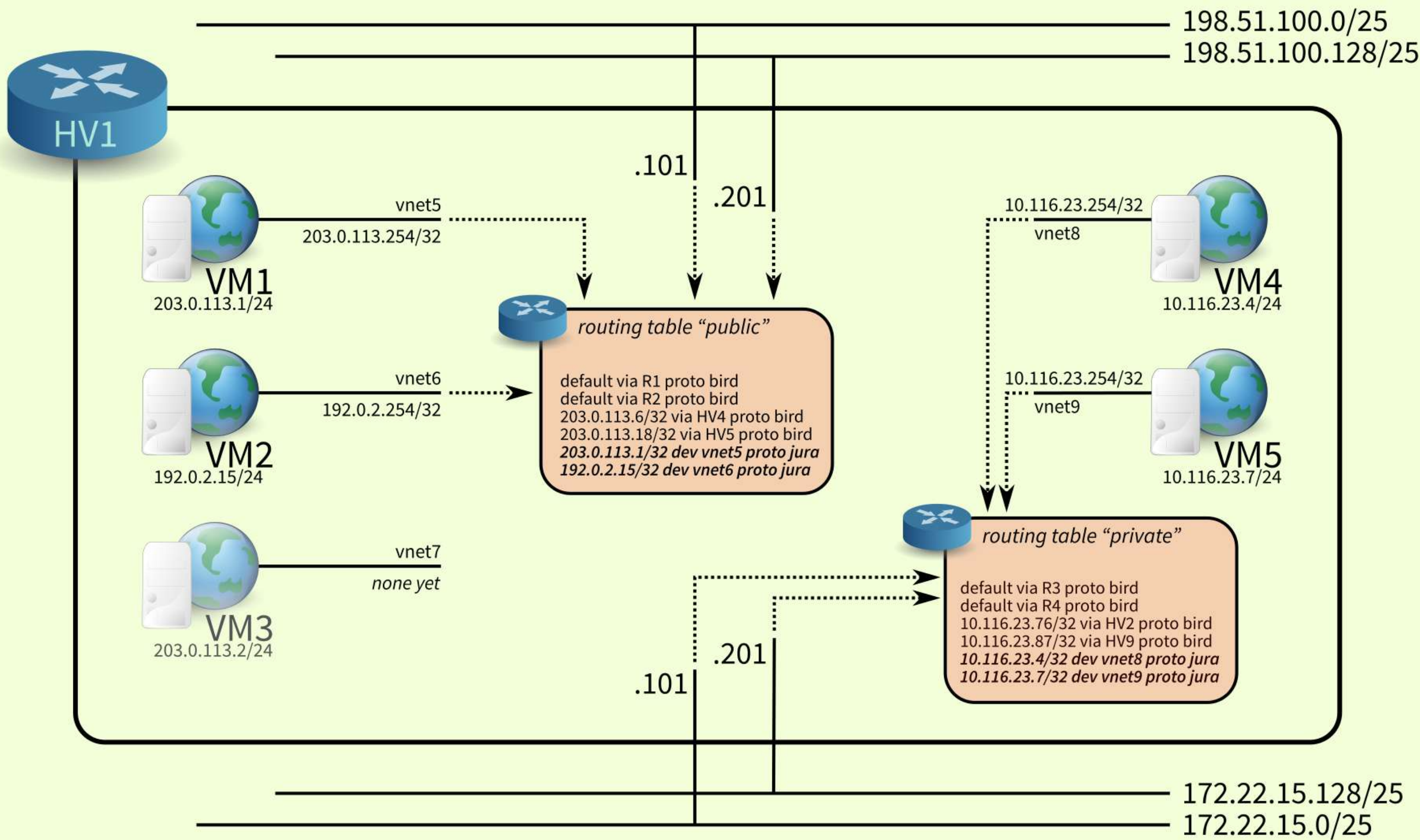


# ROUTE REFLECTORS

- Some **ToR switches** (Juniper QFX 5100) act as route reflectors too (BIRD is an option too)
- Use of virtual router feature to get **one route reflector per VLAN**: the best path from the RR point of view may be unavailable to an hypervisor due to an outage!
- Each hypervisor establishes BGP sessions with the nearest route reflectors (at least one for each path)
- Using BFD to quickly detect dead path

## L2 IN DISGUISE

- VMs expect to be on a classic L2 network (subnet, default router, friendly neighbors, DHCP)
- Hypervisors act as a **DHCP server** (radvd for IPv6 RA)
- In-kernel **ARP proxying** is enabled (ndppd for IPv6 ND proxying)
- **MAC spoofing** is useless (point-to-point connection with the hypervisor, no bridging)
- In-kernel **RP filtering** is enabled: no IP spoofing possible



# HYPERVERSOR

- **BIRD** for BGP (BFD, multiple routing tables)
- Routing tables with **local** routes and **BGP** routes
- Local routes are inserted by our orchestrator
- Routing table selection is done with IP rules:

```
11:    from all iif lo lookup main
12:    from all iif lo lookup private
13:    from all iif lo unreachable
20:    from all iif vlanX lookup public
21:    from all iif vlanX blackhole
20:    from all iif vlanY lookup private
21:    from all iif vlanY blackhole
200:   from all iif vnet5 lookup public
201:   from all iif vnet5 blackhole
200:   from all iif vnet8 lookup private
201:   from all iif vnet8 blackhole
32766: from all lookup main
```

# SCALABILITY

- **BIRD** can handle 2 million routes with 256 MB
- **Linux** can handle 2 million /32 with 256 MB
- **Route lookup** is  $O(1)$  when routing tree is dense
- QFX should be able to handle 4000+ connected hypervisors
- QFX handle BFD sessions in hardware (limit? 🚧)
- QFX can learn 4 million routes (2 GB)
- Route reflectors can also be scaled horizontally (and we can introduce other implementations)

# IP ADDRESSING

- Each hypervisor needs **several public IP addresses** for interconnect
- **BGP unnumbered** is not widely supported
- BGP + **OSPF unnumbered** doesn't scale (notably, BFD sessions)
- Use of **100.64/10**: they look like regular IP addresses

## OTHER FEATURES

- We can provide **elastic IPs** (IPs that can be freely assigned to any host)
- We can provide **anycast IPs** (in fact, no, JunOS has a terribly limited BGP add-path implementation 😞)



# DEMO

- Complete setup on GitHub: [github.com/vincentbernat/network-lab/tree/master/lab-l3-hyperv](https://github.com/vincentbernat/network-lab/tree/master/lab-l3-hyperv) (many additional information in README .md)

# FUTURE

- IPv6 deployment still in progress
- Some customers want an L2 network, we will use BGP EVPN with VXLAN for that
- From Linux 4.4, replace use of many IP rules with VRF

**QUESTIONS?**