L3 ROUTING TO HYPERVISOR

VINCENT BERNAT — EXOSCALE

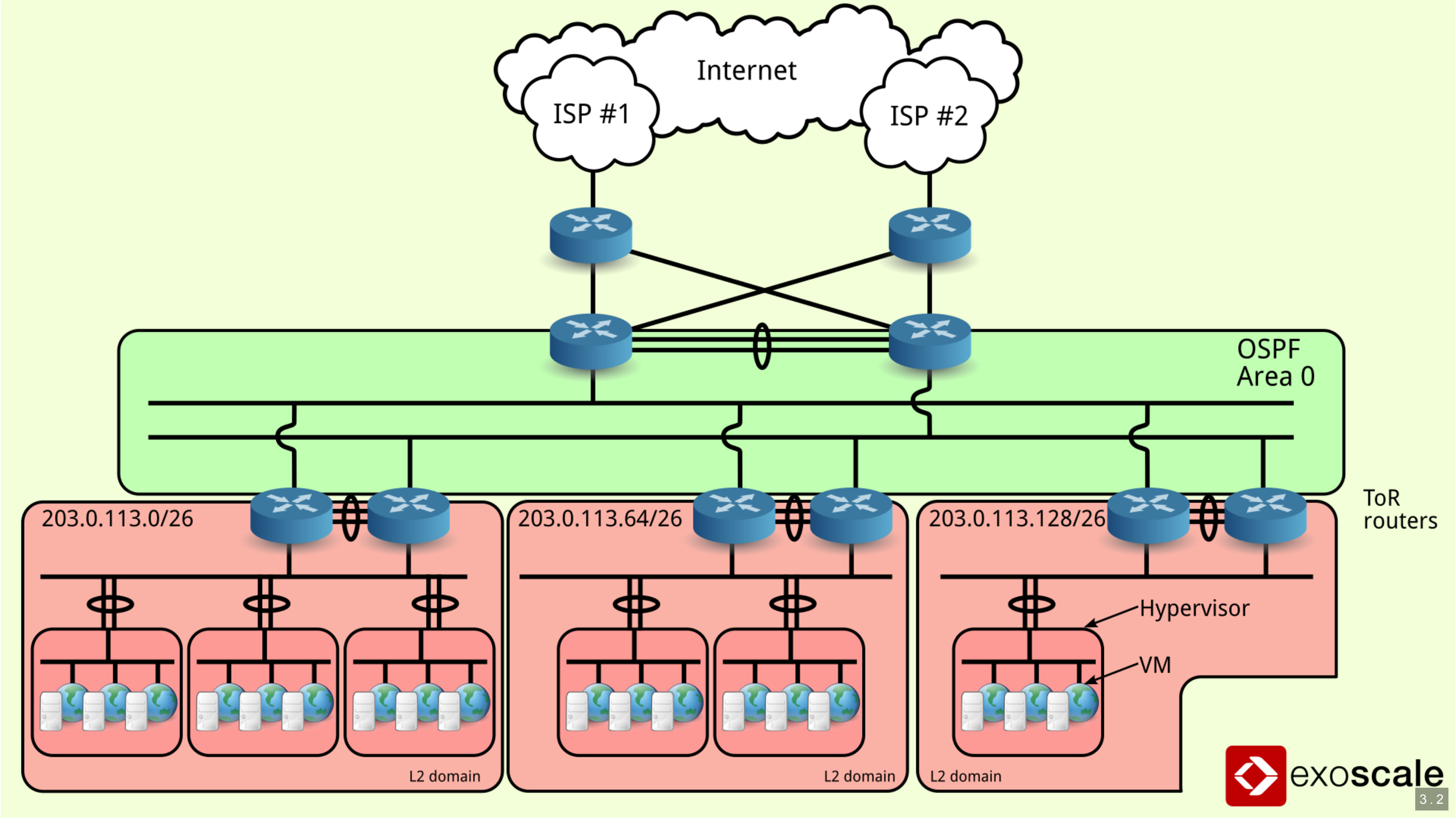
SWINOG 31 — BERN — 2017-05-30

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 with **SWINOG31** coupon code

LEGACY NETWORK

Initially, our network used a very classic design.

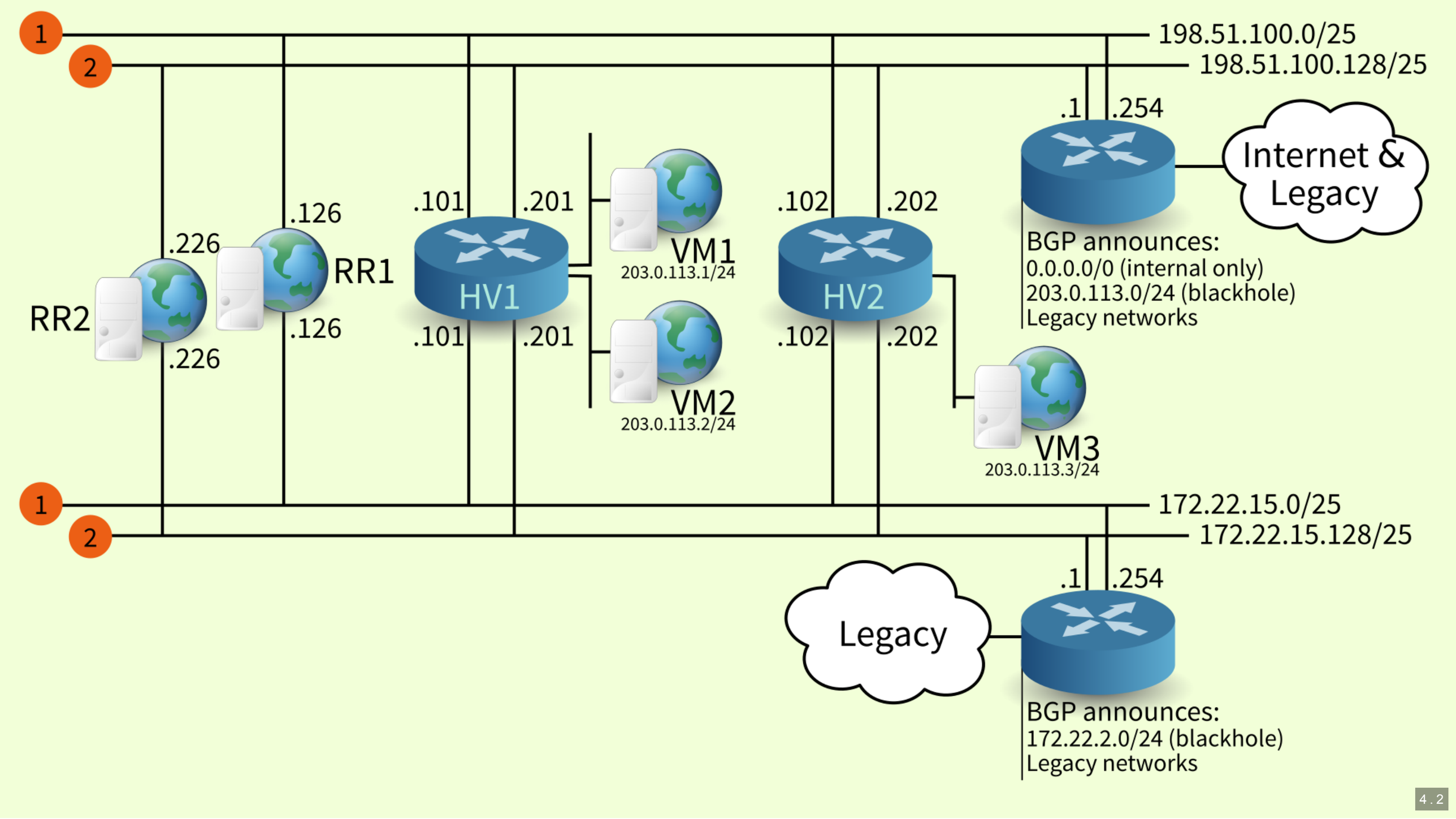

Internet
ISP #1
ISP #2
OSPF
Area 0
ToR
routers
203.0.113.0/26
203.0.113.64/26
203.0.113.128/26
Hypervisor
VM
L2 domain
L2 domain
L2 domain
exoscale

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

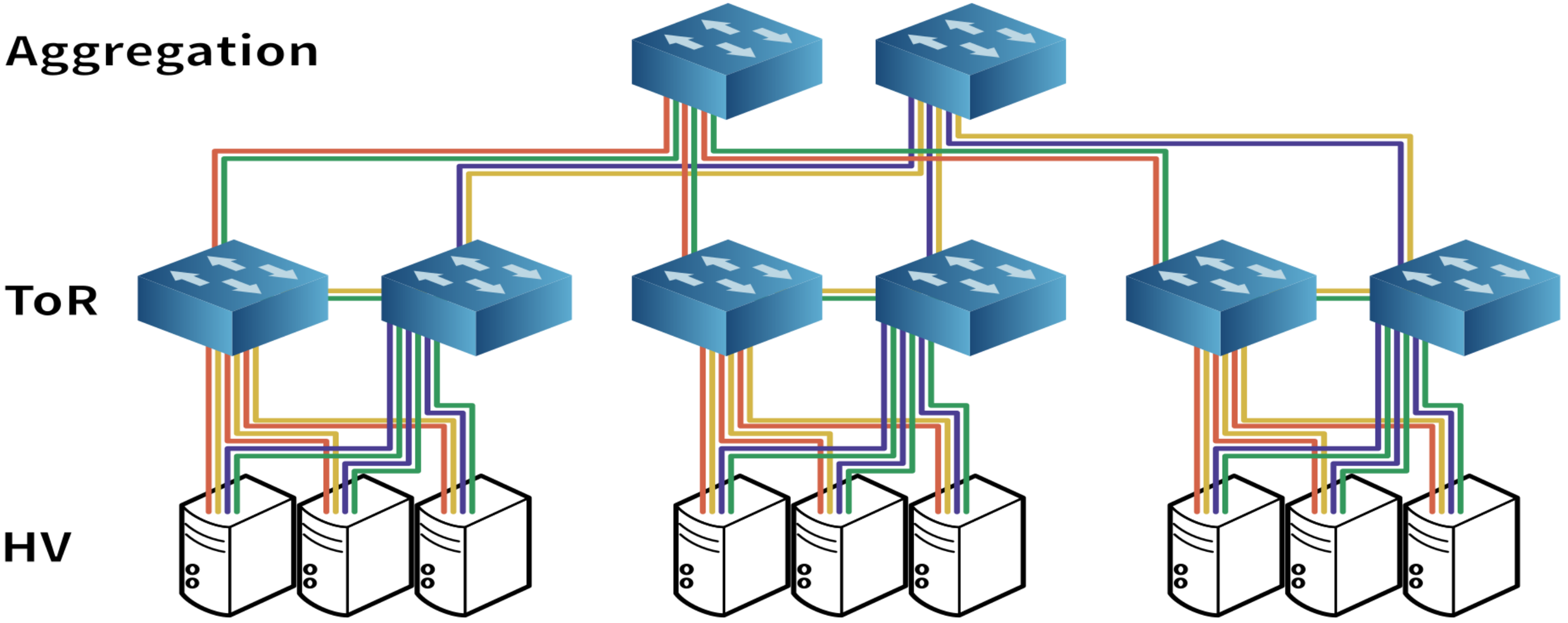
OUR SOLUTION

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


1
2
198.51.100.0/25
198.51.100.128/25
.1
.254
Internet &
Legacy
BGP announces:
0.0.0.0/0 (internal only)
203.0.113.0/24 (blackhole)
Legacy networks
.126
.226
RR1
RR2
.226
.126
.101
.201
HV1
.101
.201
VM1
203.0.113.1/24
VM2
203.0.113.2/24
.102
.202
HV2
.102
.202
VM3
203.0.113.3/24
1
2
172.22.15.0/25
172.22.15.128/25
.1
.254
Legacy
BGP announces:
172.22.2.0/24 (blackhole)
Legacy networks

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

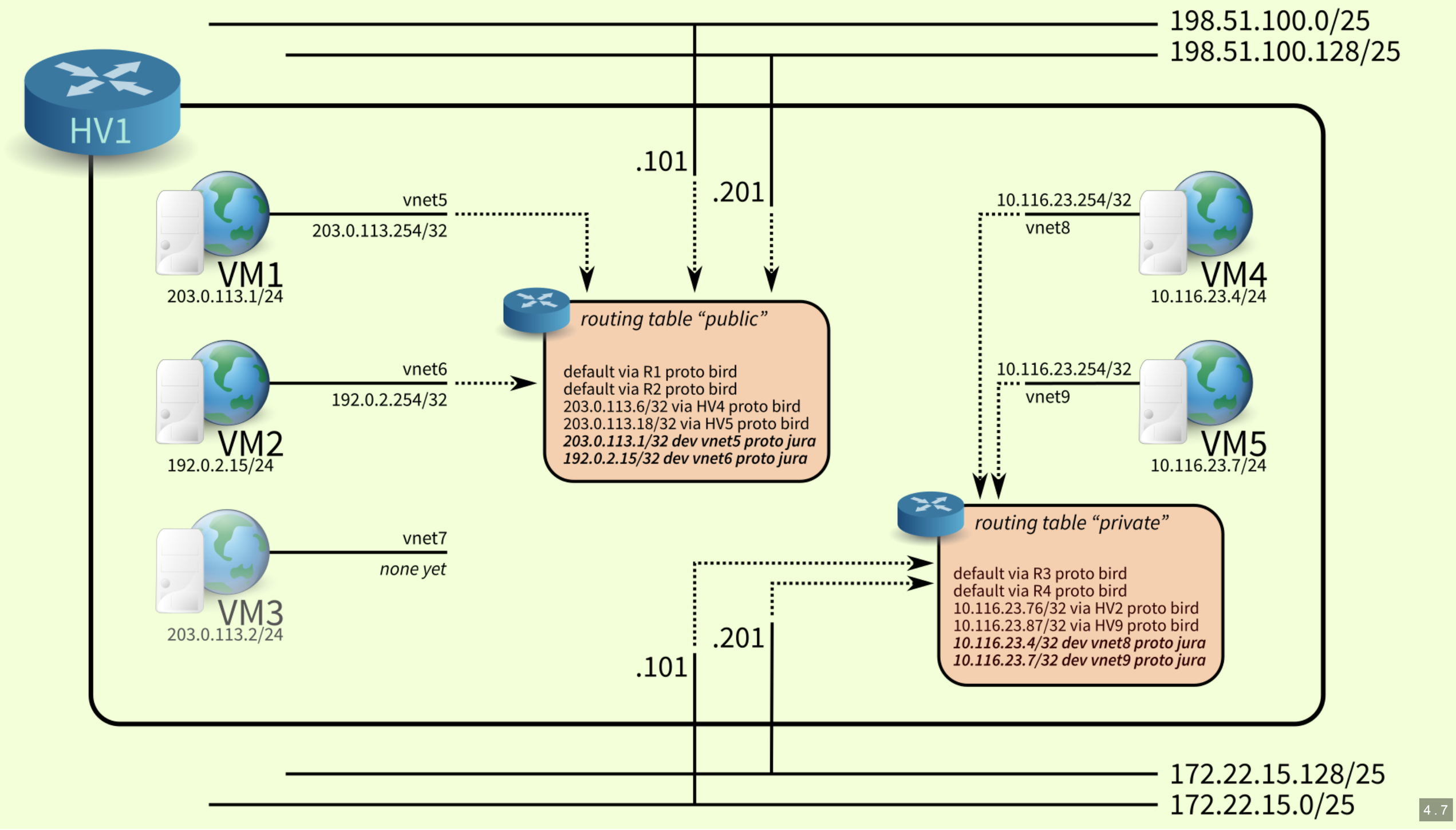

Aggregation
ToR
HV

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


198.51.100.0/25
198.51.100.128/25
HV1
.101
.201
vnet5
203.0.113.254/32
VM1
203.0.113.1/24
routing table "public"
default via R1 proto bird
default via R2 proto bird
203.0.113.6/32 via HV4 proto bird
203.0.113.18/32 via HV5 proto bird
203.0.113.1/32 dev vnet5 proto jura
192.0.2.15/32 dev vnet6 proto jura
vnet6
192.0.2.254/32
VM2
192.0.2.15/24
vnet7
none yet
VM3
203.0.113.2/24
10.116.23.254/32
vnet8
VM4
10.116.23.4/24
10.116.23.254/32
vnet9
VM5
10.116.23.7/24
routing table "private"
default via R3 proto bird
default via R4 proto bird
10.116.23.76/32 via HV2 proto bird
10.116.23.87/32 via HV9 proto bird
10.116.23.4/32 dev vnet8 proto jura
10.116.23.7/32 dev vnet9 proto jura
.201
.101
172.22.15.128/25
172.22.15.0/25

HYPERVISOR

- **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 (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?