Sovereign Cloud Stack

a project of

OSB Open Source Business ALLIANCE
Bundesverband für digitale Souveränität e.V.

Supported by

Supported by:
Federal Ministry for Economic Affairs and Climate Action
on the basis of a decision by the German Bundestag

Open Networking in the Sovereign Cloud Stack Project

A Community Hyperscaler - built and operated by many, owned by all

Sovereign Cloud Stack

One platform — standardized, built and operated by many.

- Control of data - enabling cloud operations through a decentralized, standardized and federated cloud & container platform
- True digital sovereignty to promote trust in clouds
  - Enabling innovation on a sovereign infrastructure
  - Reducing technical, economic, strategic and legal dependencies
- It's up to you: Find the right provider of your choice or operate your own SCS in a data center of your choice.

Sovereign Cloud Stack

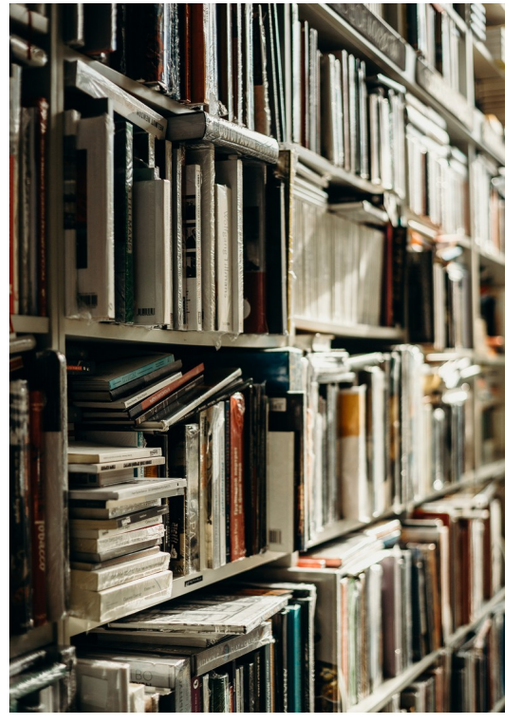
One platform — standardized, built and operated by many.

- We are vital open source project
- Developed, supported and operated by a globally active open source community
- Financed with the kind support of the German Ministry of Economic Affairs (now it its third and final year of public funding)
- We combine proven and robust cloud technologies under a uniform standard
  - Users find the same conditions with different providers
  - Providers use their narrow engineering capacities for the things that make the difference by sharing technologies and having a community of practice

Sovereign Cloud Stack

Certifiable Standards

Modular Open Source Reference Implementation



Operational Knowledge

The motivation for Open Networking

- Many manufacturers offer the functionalities required for SCS, but these are configured and behave in different ways, this is not a nice thing from the point of view of
  - Implementation of Automation and Orchestration
  - Standardization
  - Documentation
- An open design and an open implementation provides
  - the ability to analyze, understand and solve problems in depth
    (by reading the source, debugging techniques, creating temporary and permanent fixes, ...)
  - enables the addition and improvement of needed functions
  - offers the possibility to purchase hardware from different manufacturers but to use it uniformly
  - enables the same technologies to be used at different points in the infrastructure
    (e.g. FRRrouting on nodes and the switches)

The motivation for Open Networking

- Agility and Independence
  - A vital opensource community provides a open eco system of developers, users and support mechanisms
  - Vendor Independence, knowledge about the product itself and the strategy is not gatekeeped anymore
  - Problems / new solutions can be accessible with a temporary fork / flavor of the NOS
  - Strategy of the solution is subject to the needs of the users and is shaped by those who contribute
  - In-house experts with powerful capabilities can be established in a way that would never be conceivable with a proprietary product.
- Open network technologies and standards facilitate adoption of SCS by providing a standardized and well-supported technology stack for users
  - Open technologies avoid vendor lock-in and reduce cost for end users by competition of vendors
  - SCS support for open technologies benefits the wider open-source community

SONiC as Network Operating System

- The attractiveness in the CSP environment
  - a sound alternative to enterprise grade network OS (NOS) – proven track record with Microsoft Azure
  - easier to support and mange within an consuming organization compared to multiple NOS vendors in parallel
  - supports a wide range of vendors, switches and ASICs, flexible usage of devices from different vendors
  - a Debian based system which run containers fosters collaboration, knowledge-transfer and technological exchange between network and Linux experts
- The opportunities for the future
  - Improved support for SONiC in SCS
    - simplified provisioning, configuration-management, powerful and holistic automation of devices
    - Seamless integration into the overall system
  - Improved ML2 plugin – new drivers/update to existing drivers (networking-generic-switch)
  - Leverage SONiC to improve SDN scalability in SCS
  - New SONiC features – Support Geneve for network-isolation, -abstraction, -tunneling and -load-balancing

SONiC Enterprise and Community

- Motivation for building our own SONiC images
  - Get to know the internals better – develop the capability to make bugfixes and enhancements
  - Get our lab hardware working
    (Edgecore 4630-54TE-O-AC-B, 5835-54X-O-AC-B, 7326-56X-O-AC-B, 7726-32X-O-AC-B)
  - Getting to know the interaction with the upstream community
  - Fast integration of the current development status
- Our learnings and takeaways
  - Community and enterprise versions have both issues on our purchased equipment
    - The bug fixes were not particularly complex.
    - It seems to be a challenge to support multiple device types, even from a single vendor)
  - We are enthusiastic supporters, but SONiC in general needs some love in the area of quality assurance
  - SCS decided not to create its own SONiC distro, will use community images
  - New features can be released as packages/containers and installed on top of SONiC
  - Improvements / bugfixes will flow directly upstream

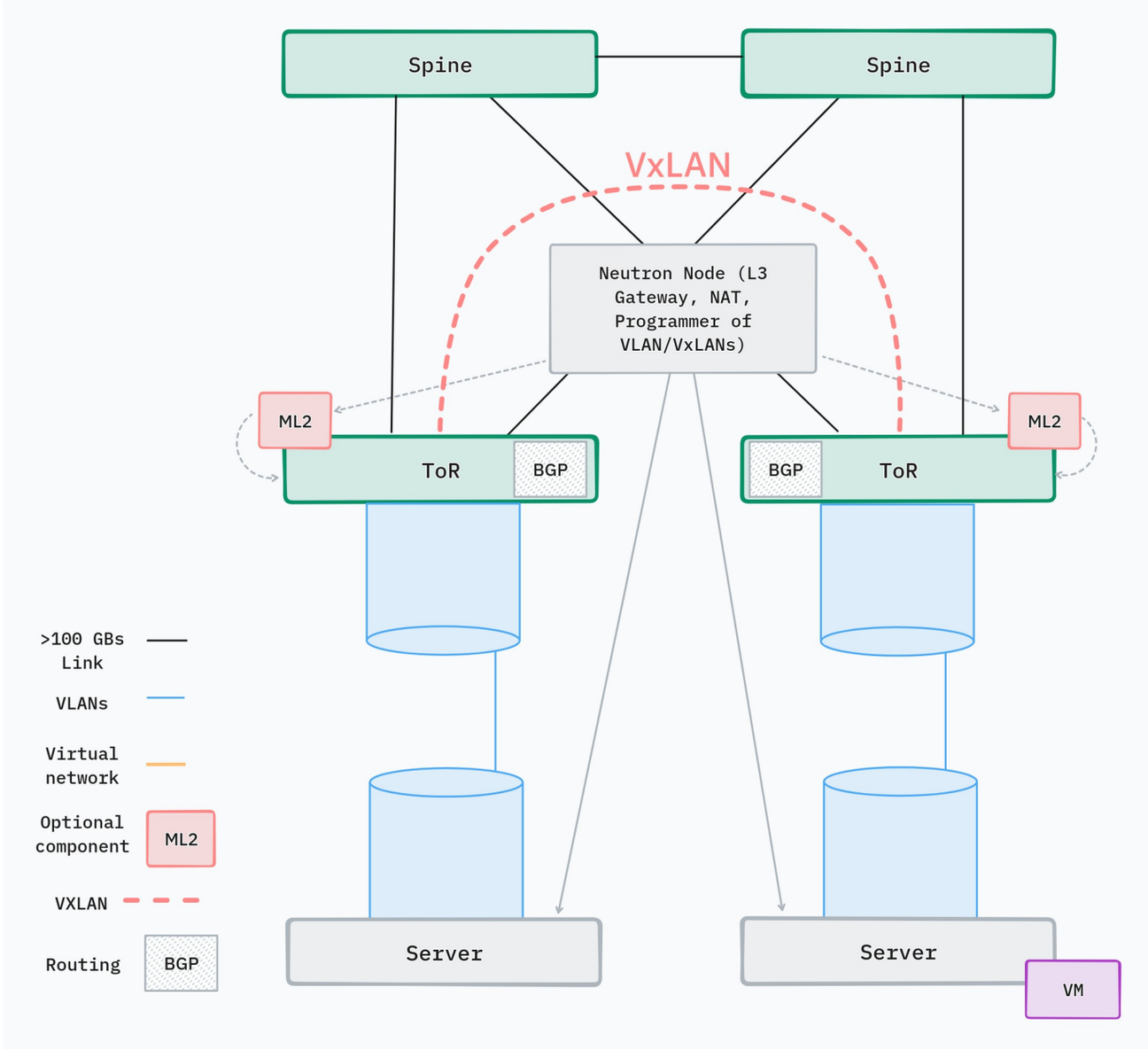
SONiC Enterprise and Community

- Possible drawbacks specific to SONiC Community
  - Hard to support, if not purchasing commercial support and really not want to spend resources to have experts for SONiC in your organization
  - The manufacturers have a deeper understanding of the details of the hardware. When it comes to hardware-related topics, it is much more difficult to provide 3rd party support on a par with the manufacturer
- Possible drawbacks of both variants
  - We are enthusiastic supporters, but SONiC in general needs some love in the area of quality assurance
  - Lacks currently(!) wide adoption in the open-source community

Openstack Networking – Quick Start

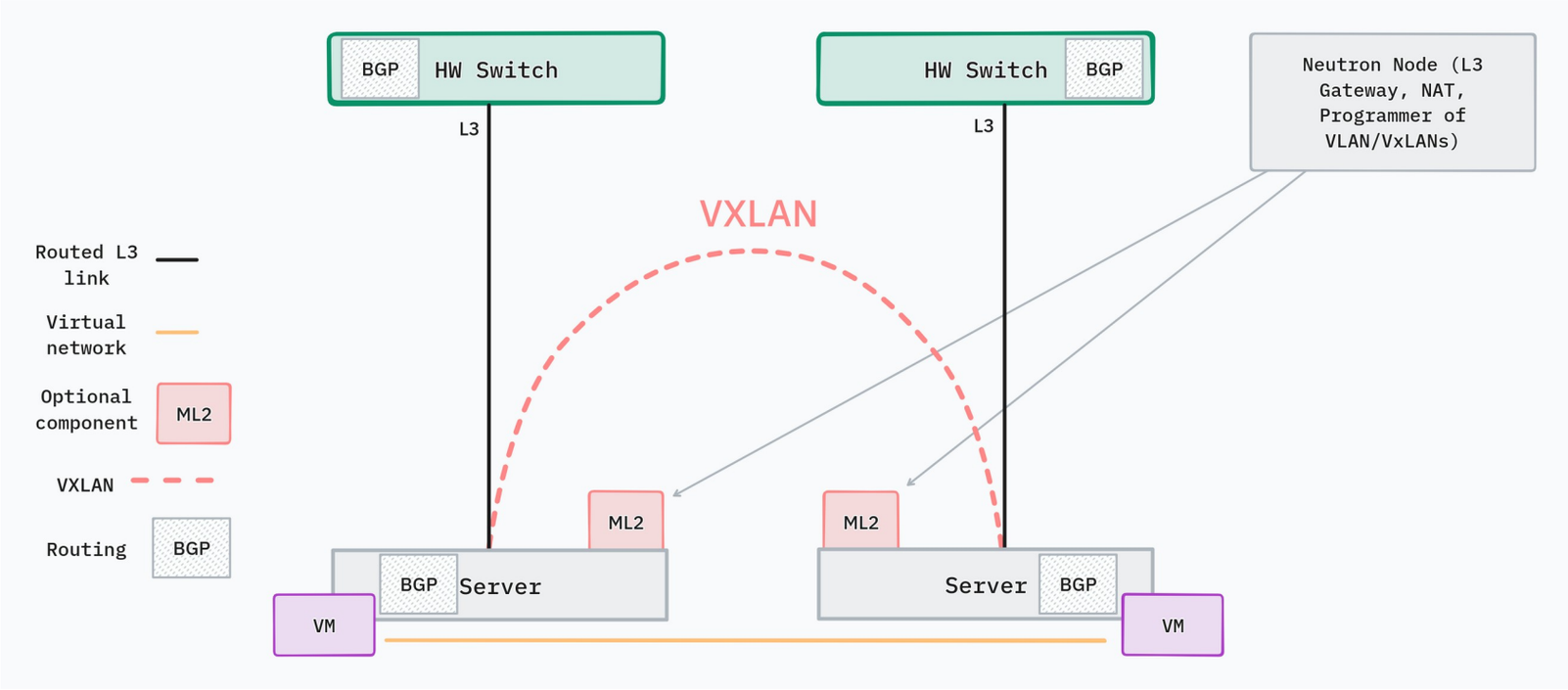
- General key aspects
  - Design the network before you start. Different designs can handle different amount of tenants.
  - Automate everything – especially network configuration for switches.
- Options for scaling Openstack environments
  - Migration to underlay network designs better suited for scalability
    - Layer 3 underlay
    - Use eBGP unnumbered with private 4 byte ASNs
    - Use VXLAN and EVPN for tenant separation / communication
  - SmartNICs and DPUs can be leveraged to improve network stack performance
- The role of ML2
  - Configures network devices automatically from Neutron
  - ML2 Agent runs on each network (and server) device
  - ML2 drivers for different device types and vendors

Network Scaleout - EVPN-VxLAN



- Network centric
- Scalable, resilient
- Works well for 100-1000 tenants
- eBGP on ToR/Spine
- L2/VLAN-Trunk to the Servers
- VXLAN needs EVPN (Openstacks knows only about VXLAN)
- Neutron node can become a choke-point because east/west and north/south traffic is routed through it

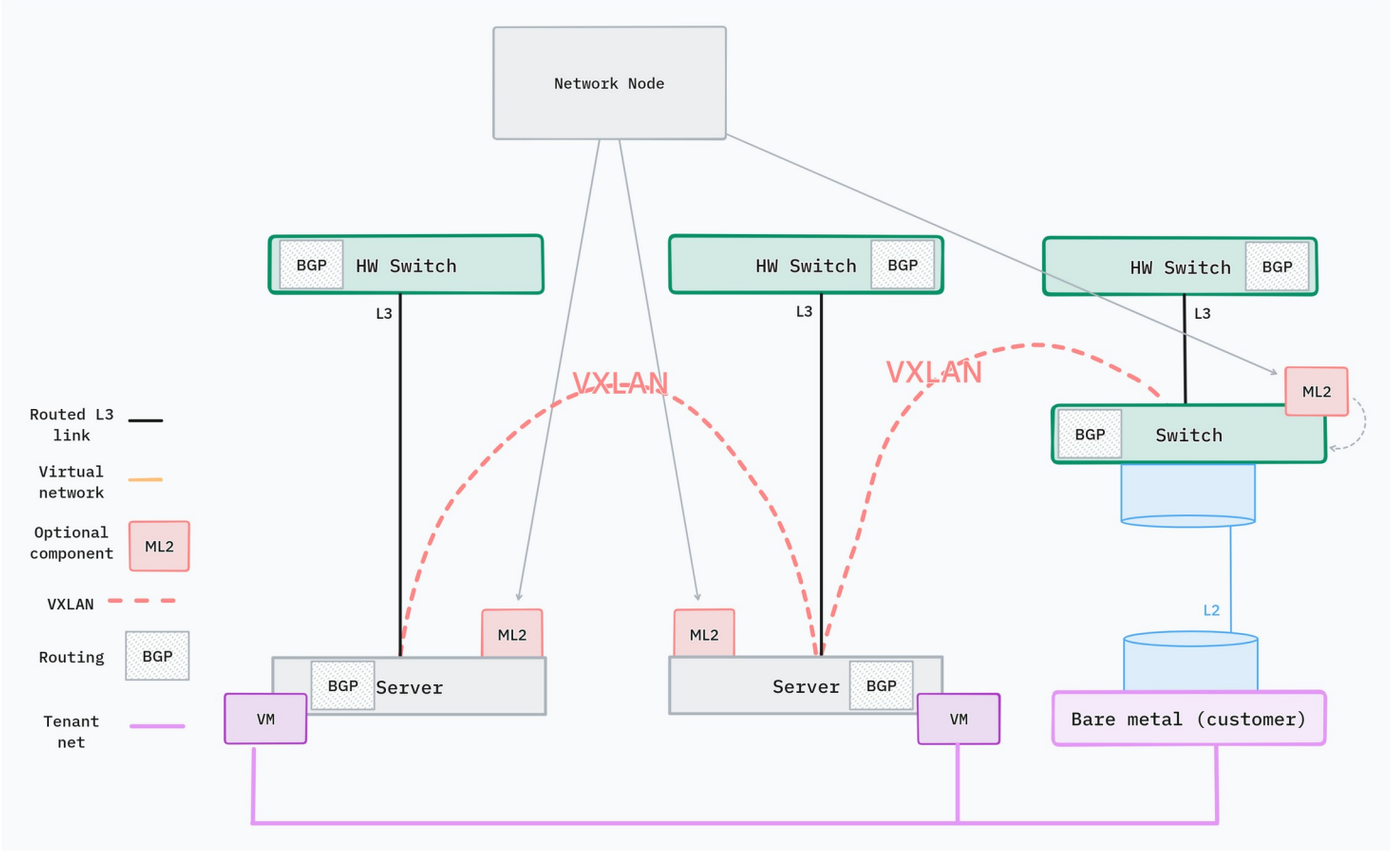
Network Scaleout - VxLAN on Servers



- Most scalable, Simplest network
- More complex servers
- > 1000 tenants
- eBGP on the Servers
- VXLAN needs EVPN

- NIC needs VxLAN offloading
- OpenStack links VxLAN to VM
- Switches does basic L3 routing
- VXLAN tunnel programmed by Neutron
- Efficient data path routing for floating IPs and east/west traffic

Network Scaleout – Combined Designs



- Mix both previous designs
- Allow the integration of bare metal nodes / Customer Nodes
- Bare Metal: Routing is terminated at the switch (DPU)

Future SCS Networking

- The future objectives
  - Provide good SONiC integration and support for SCS deployments (especially Layer 3 underlay)
  - Make network network scalability and performance easily accessible
  - Direct integration of rollout and configuration of the network
  - Improve support for SmartNICs and DPUs
  - Support Bare Metal and hybrid deployments by ML2 and SmartNICs/DPUs
  - BGP improvements related to cloud interconnectivity
- The evolutionary next steps
  1) Automate deployment of SONiC switches in OSISM, Netbox
  2) Dynamic SONiC configuration via Openstack (ML2)
  3) Maybe: Move Openstack network control-plane from Network Node over to SONiC
  4) Improve SmartNIC and DPU Support
  5) Host Orchestration: BGP/EVPN on the host from Openstack Neutron

Upcoming SCS Event

SCS: A Community Hyperscaler - built and operated by many, owned by all