Cloud Interconnect

SCS VP04 - Lot 3

Angel Kafazov, Bozhidar Ignatov

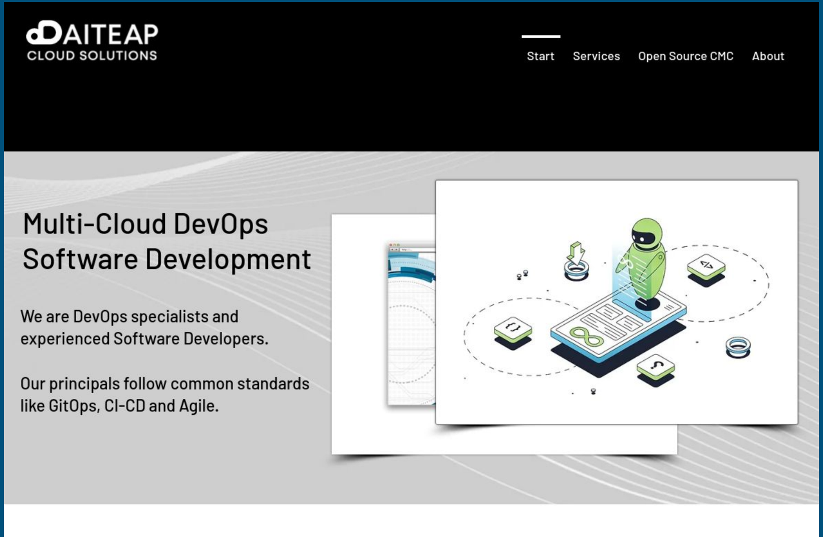
Open Operations Meetup
22 April 2024

About us

Angel Kafazov

Bozhidar Ignatov

https://www.daiteap.com/



Multi-Cloud DevOps
Software Development

We are DevOps specialists and experienced Software Developers.

Our principals follow common standards like GitOps, CI-CD and Agile.

Motivation

Goals

- Provide SCS community and CSPs possibility to interconnect clouds
- use open-source software
- enable integration with GAIA-X
- be less invasive
- flexibility (extend to VMs, containers, across clouds and data centers)
- scalability
- meet the datacenter trends (100% L3 underlay)

Out of scope

- Identity Federation
- DC VPNs

L3 Networks in the Data Center

L2 networks at DC

- Typically data centers route packets at upper layers and the lower layers as L2 network
- Issues: slow convergence time, large failure domains, large broadcast domains

L3 DCs route packets at the lowest levels of the infrastructure (ToR and on the CN)

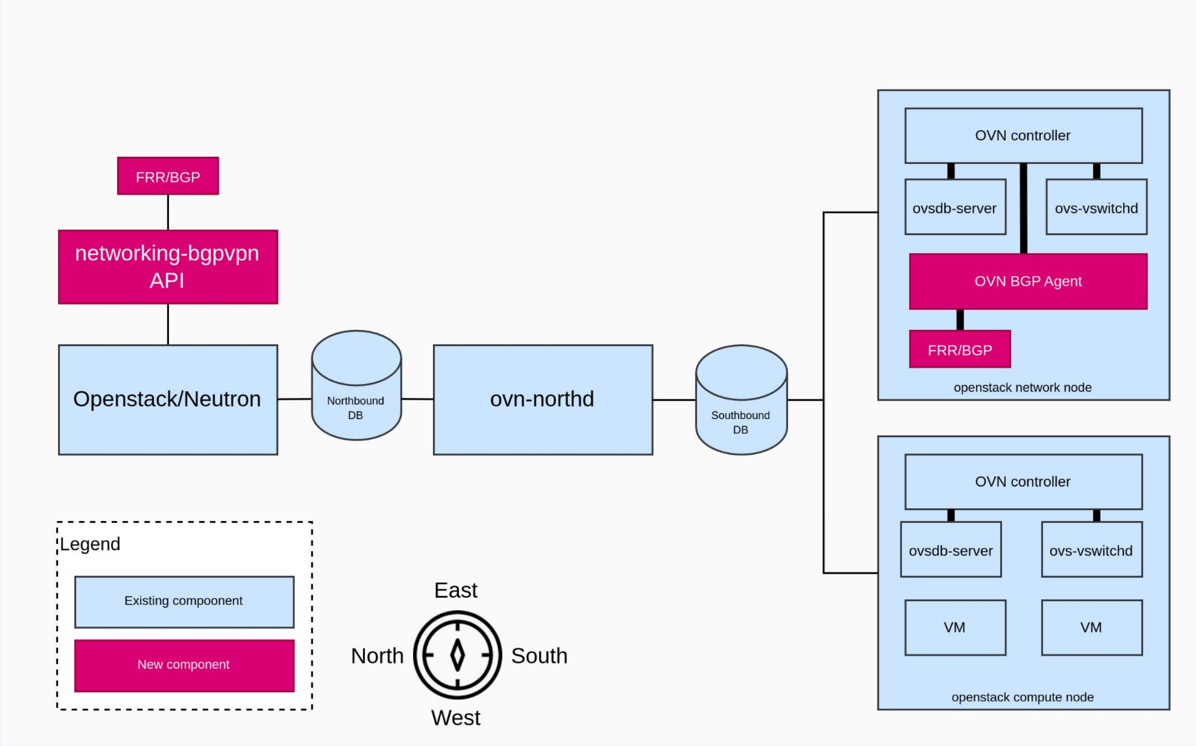
- Smaller L2 broadcast domains
- Smaller L2 failure domains
- Faster convergence
- Avoids static configuration in the fabric
- Network Protocols: BGP, BFD, ECMP

Why L3 networks?

The main functionality of the ovn-bgp-agent is to map the routes from OVN's logical routing constructs into BGP announcements, which inherently assumes a Layer 3 capability in the underlay or at least at the boundary where BGP becomes relevant.

Openstack Networking

- Neutron - main networking in Openstack
- Supports pluggable backends for SDN control-plane
- ML2 driver
- OVN/OVS implement SDN functionality
- OVN-BGP-Agent - extends Openstack networking with cloud interconnect features
- networking-bgpvpn - VPN connectivity



Interconnect OpenStack and BGP/MPLS VPNs

BGP/MPLS VPNs:

- a key building block for backbone network engineering
- the foundation for operators VPN services
- conrolible functionality and quality of interconnection services
- what are BGP/MPLS VPNs:

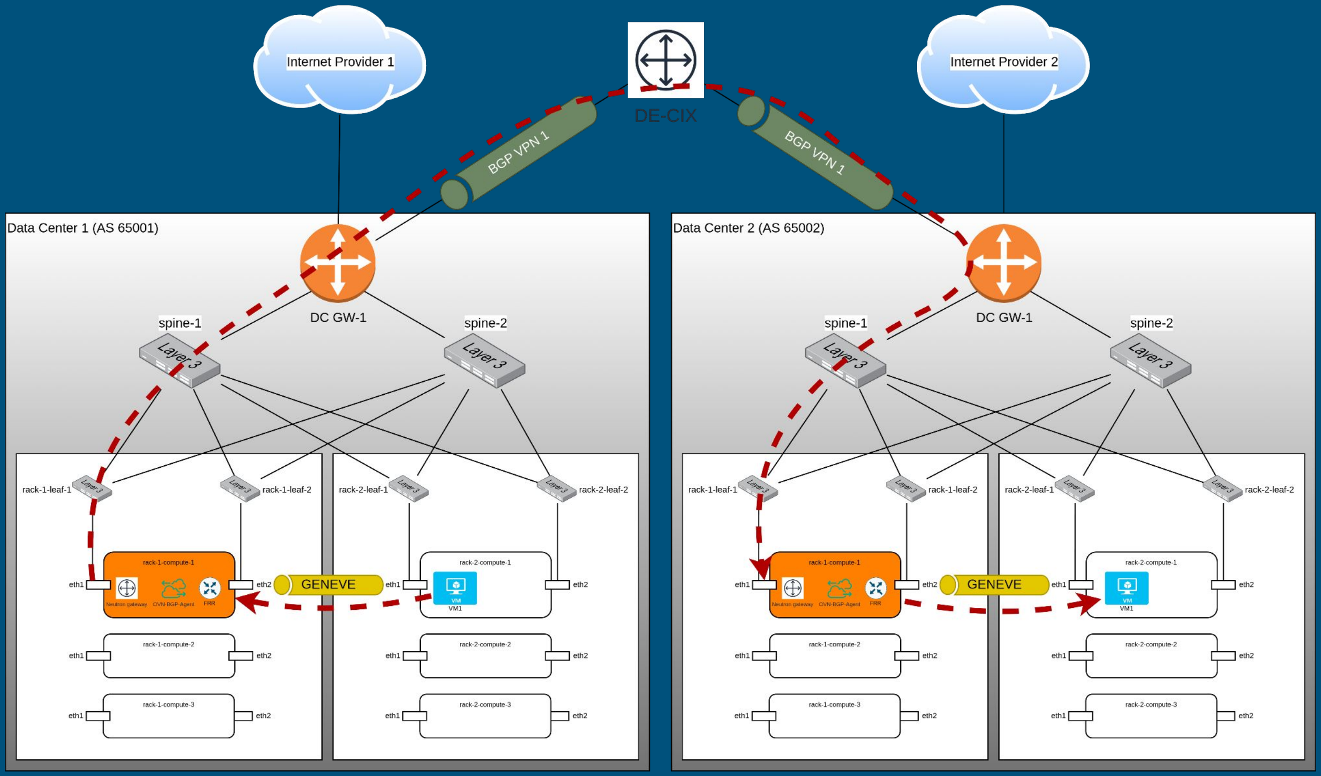
  - use MPLS to isolate the traffic of different VPNs

  - use the BGP routing protocol to indicate where/how to send packets: advertise routes, VPN “identifiers” (Route Target)
- Solution is transparent in regard to the kind of VPN connectivity

Clouds interconnection with BGPVPN

BGP or BGP/MPLS VPN connections

- connecting VMs of one cloud to VMs of the distant cloud
- OVN connects VMs within the clouds (E/W)
- BGP with Fabric to connect beyond the clouds (N/S)



Components, How it works

BGP (FRR)

- running BGP speaker on each node connected to ToR or DC GW
- advertisement of directly connected routes

OVN/OVS

- controllers on different racks/networks
- Loopback IP configuration
- FRR and agent deployment and configuration

OVN BGP Agent

- read from OVN DB
- FRR to advertise host routes to workloads
- configures local vTEP devices for EVPN mode
- redirect traffic to/from the OVN overlay

networking-bgpvpn plugin (API)

BGP (FRR)

- BGP
  - BGP is a dynamic routing protocol: AS (eBGP/iBGP), BGP Unnumbered, ECMP, announce a default route(Loopback IP), VRF support (L3 isolation, routing table)
- FRR is the choice of BGP implementation and deployment
  - a Linux Foundation project
  - VTYSH is a shell for FRR daemons
- BGP Advertisement by triggering from OVN BGP Agent
  - VRF is created, by default with name bgp-vrf
  - FRR is configured to leak routes for a new VRF
  - dummy interface (default name bgp-nic), associated to the VRF device
  - ARP/NDP is enabled at OVS provider bridges

To expose the VMs/LB IPs as they are created, since the FRR configuration has the redistribute connected option enabled, the only action needed to expose it is to add it from the bgp-nic dummy interface.

BGP (FRR)

```
BGP configuration

router bgp 65001
  bgp router-id 172.30.1.1
  bgp graceful-shutdown
  no bgp default ipv4-unicast
  no bgp ebgp-requires-policy

  neighbor uplink peer-group
  neighbor uplink remote-as internal
  neighbor uplink ttl-security hops 1
  neighbor uplink bfd
  neighbor uplink bfd profile 3pleo
  neighbor enp3s0 interface peer-group uplink

  address-family ipv4 unicast
    redistribute connected
    neighbor uplink activate
  exit-aadress-family

  address-family ipv6 unicast
    redistribute connected
    neighbor uplink activate
  exit-aadress-family

bfd
  profile 3pleo
    detect-multiplier 10
    transmit-interval 500
    receive-interval 500
```

OVN BGP Agent

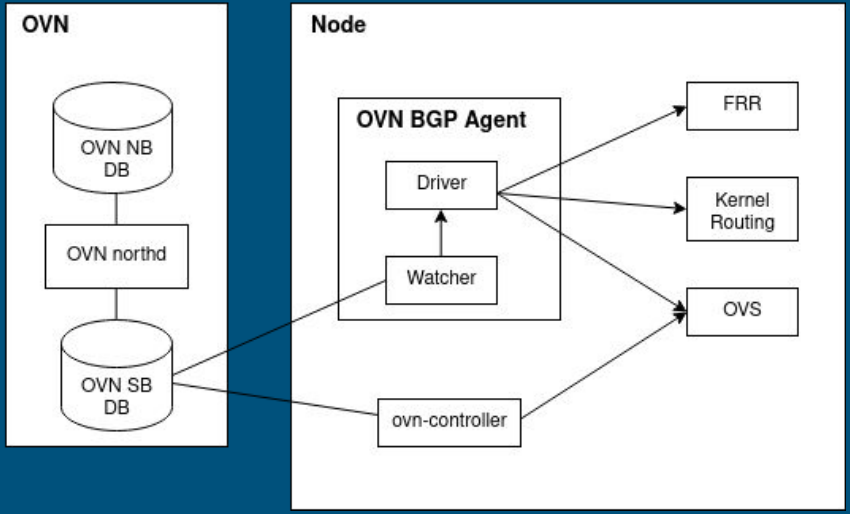
Overview OVN BGP Agent

- Python based daemon running on OpenStack nodes
- Reads OVN SB db events to trigger the actions
- Leverages FRR/BGP to announce relevant IPs (VMs and/or LBs)
- Leverages kernel networking capabilities to redirect traffic to OVN overlay
- It needs:
    - BGP (FRR) to advertise directly connected routes
    - Node to be connected to BGP peer(s) (leafs or DC GW)
    - ARP/NDP proxy enabled

OVN BGP Agent

- No modifications to Core OVN or Neutron
- Different drivers:
  - BGP:
    - has to be installed on every node
    - no API, all VMs/LBs are exposed
  - **EVPN:**
    - advertise tenant networks (on a VxLAN id)
    - installed only on network gateway node(s)
    - API to select the neworks to expose
- Different watchers:
  - triggering the actions in response to OVN SB DB **Port_Binding table** events
  - different actions depending on the driver
- Other drivers can be integrated: doing different actions depending on **Port_Binding table** events
- We will focus on the EVPN mode/driver

OVN BGP Agent Architecture Diagram



OVN BGP Agent

EVPN Driver

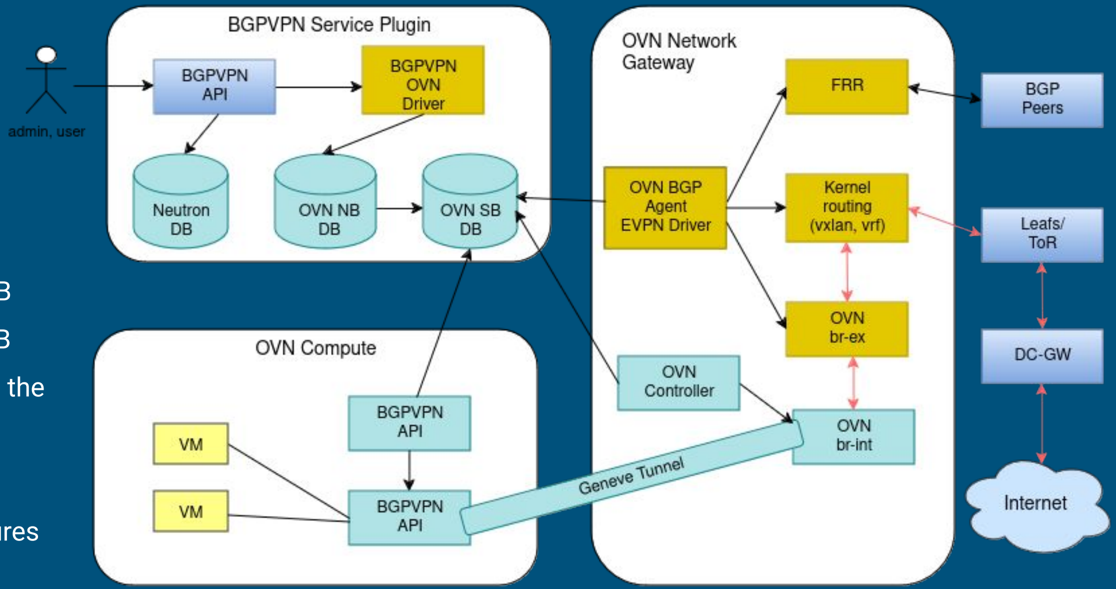
- Advertise tenant networks
- API (networking-bgpvpn) to select networks to expose
- Traffic needs to go through the networker node, that hosts cr-lrp port
- Interconnects OpenStack clusters inside same or across DCs

OVN BGP Agent

EVPN Driver schema / how it works

- networking-bgpvpn as the API
- BGPVPN driver interacts with OVN Database:
  - add VNI/VXLAN ID and AS info into OVN NB DB
  - automatically translate the info into OVN SB DB
  - Agent detects the event (watcher) and triggers the needed actions (driver)
- The OVN BGP Agent then wires the network and configures the BGP daemon

BGPVPN Service Plugin and OVN BGP Agen Diagram with EVPN Driver



OVN BGP Agent

EVPN Driver

Network exposed:

- traffic between nodes (VRF/VXLAN)

  - Create VRF, bridge, VXLAN and dummy device

  - Veth-pair to connect VRF to OVS (provider) bridge

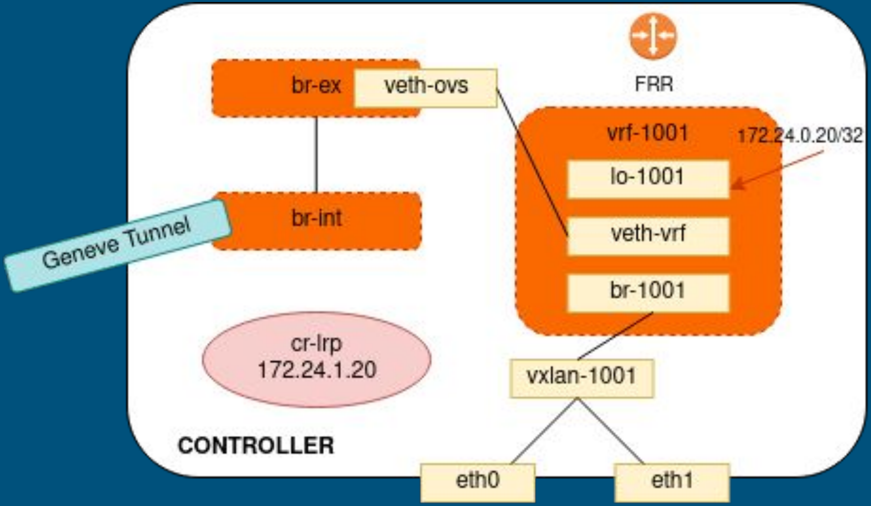
  - Reconfigure FRR with VRF to EVPN

  - Add ip routes to the VRF routing table to redirect the subnet CIDR to br-ex

  - Add OVS flows to redirect traffic back from OVN to VRF

- BGP Advertisement (FRR)

  - Add VM IP to the dummy device

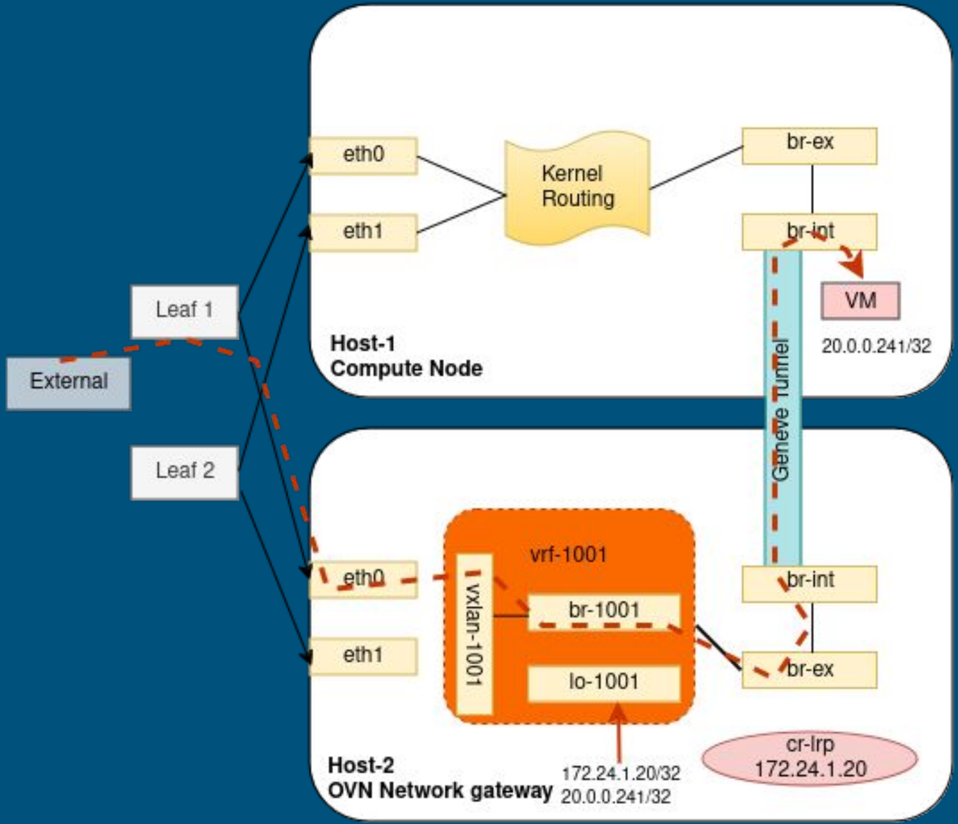


OVN BGP Agent

EVPN Driver
Traffic flow to the tenant network:

- VM IP can be advertised in a node where the traffic could be injected into OVN overlay
- Once the traffic reaches the specific node, the traffic is redirected to the OVN overlay

Traffic Flow - External to Tenant



networking-bgpvpn plugin API

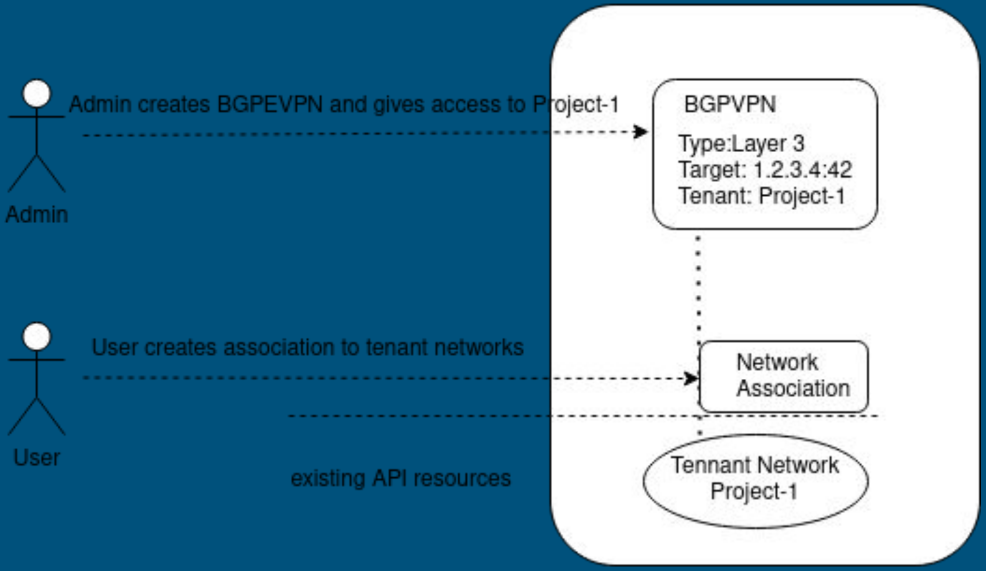
API:

- To expose VMs on a tenant network:

  - create the BGPVPN resource and associate it to a user

  - associate it to either a router or a network
- Admin creates BGPVPN resource
- User associates network/router to BGPVPN

Admin and User Interactions to API


Admin creates BGPEVPN and gives access to Project-1
Admin
BGPVPN
Type:Layer 3
Target: 1.2.3.4:42
Tenant: Project-1
User creates association to tenant networks
User
Network Association
existing API resources
Tennant Network Project-1

Integration into SCS

Activities:

- Integration of ovn-bgp-agent and networking-bgpvpn plugin into kolla-ansible
- Exposing configuration parameters to OSISM and kolla-ansible
- Provide documentation for SCS cloud interconnect
- Current scope: Openstack based clouds

Outcomes:

- Enable cloud interconnectivity for operators and users
- Support enterprise level BGP VPNs and MPLS VPNs
- Leverage open-source software to make SCS more attractive

Integration with GAIA-X and IXPs

Tasks:

- Standardization of GAIA-X purchasing of VPN links between cloud providers
- Control and configuration of purchased connectivity in SCS
- Support for BGP VPNs and MPLS VPNs via IXPs (DE-CIX)
- Integration with TELLUS Project

Roadmap:

- Support of user managed IPSEC and Wireguard VPNs
- Integration with other GAIA-X based dataspaces

Q&A