



Dynamic SFC from Tacker to incept specific traffic of VM

Yong Sheng Gong

99cloud

Dimitrios Markou

Intracom Telecom

Yan Xing'an

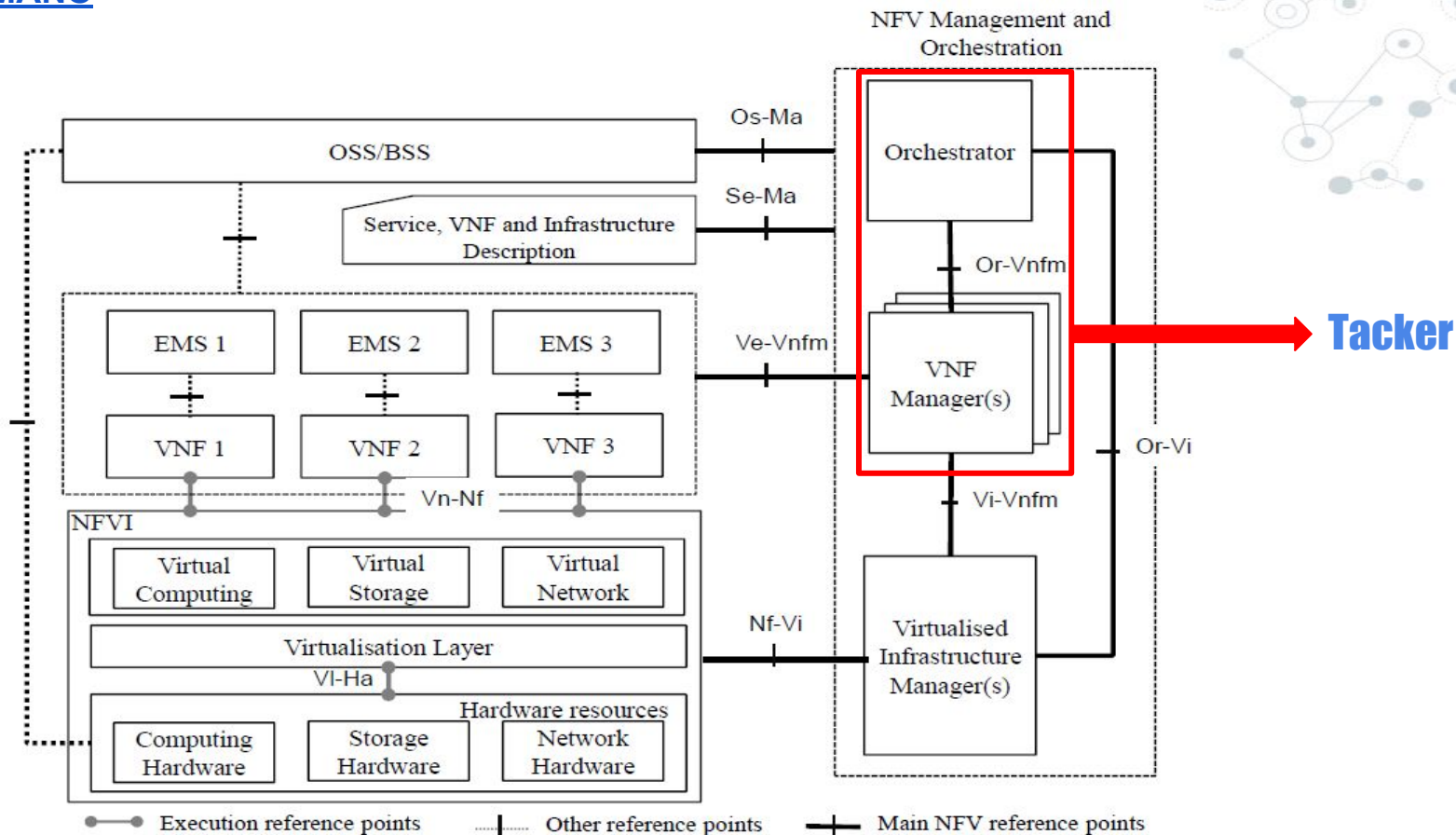
China Mobile



What is Tacker ?


- ◎ Is an official OpenStack project
- ◎ Is a VNFM and a NFVO
- ◎ Is based on ETSI MANO architecture

NFV MANO





Small Overview

- ◎ VNFFG stands for VNF Forwarding Graph
 - ◎ Orchestrates and Manages traffic through VNFs
 - ◎ VNFFG consists of one or multiple NFPs (Network Forwarding Paths)
 - ◎ Ability to implement SFC
- 

What is a NFP?

- ⦿ NFP stands for Network Forwarding Path
- ⦿ Describes the path of the actual traffic flows on the virtual links
- ⦿ Consists of one or many classifiers and a path of one or many VNFs.
- ⦿ NFP is a subset of a VNF-FG

E.g :

Control Traffic NFPs:

Handles the control traffic which flows in a datacenter.

User Traffic NFPs:

Handles the User traffic which flows in a datacenter

What is a VNF-FG

- VNF-FG stands for VNF Forwarding Graph
- Is a superset of the NFP
- Can contain one or more NFPs
- Describes in a more abstract level use cases in a Telco environments

E.g:

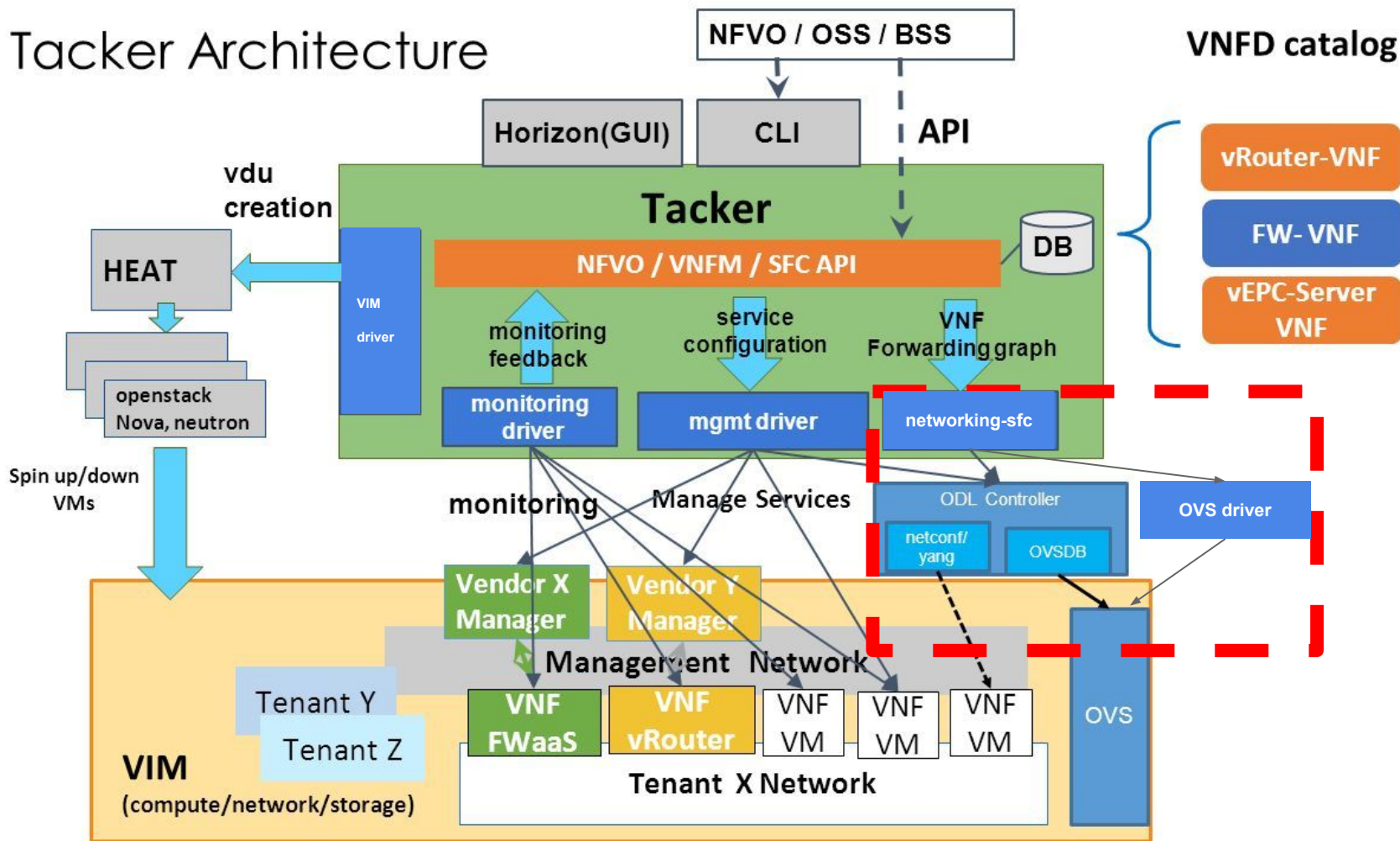
Control Traffic VNFFG:

Consists of all the NFPs which are handling the control traffic flows in a datacenter.

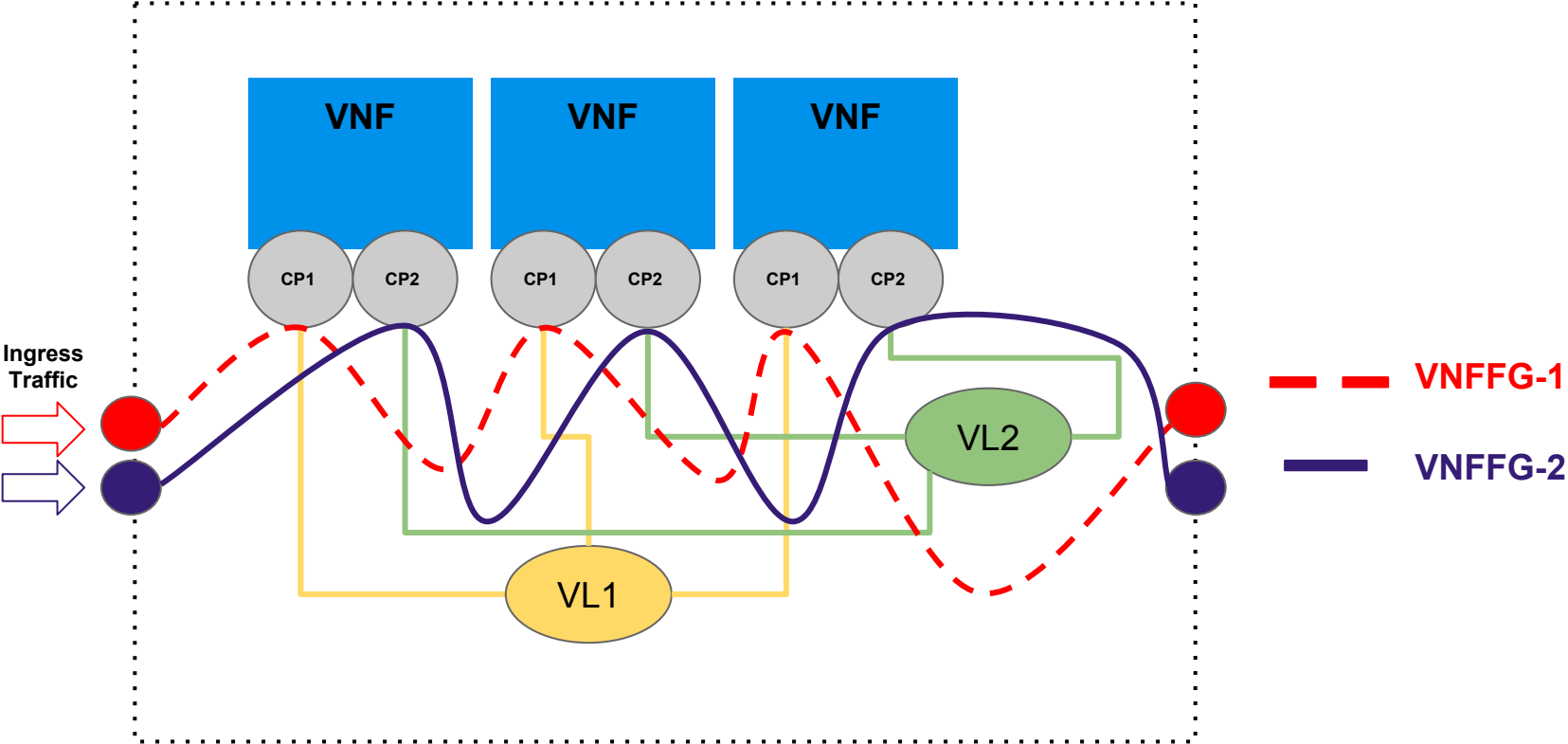
User Traffic VNFFG:

Consists of all the NFPs which are handling the user traffic flows in a Datacenter.

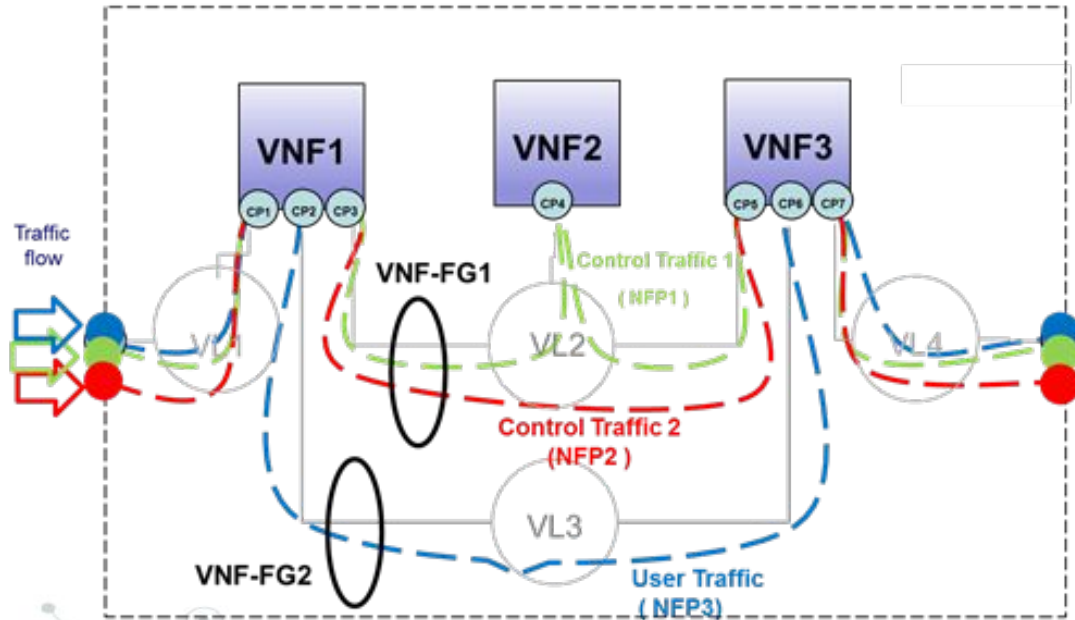
Tacker Architecture



Example Topology



Example Topology



VNF-FG1 contains:

VNFs: VNF1, VNF2, VNF3

CPs: CP1, CP3, CP4, CP5, CP7

VLs: VL1, VL2, VL4

NFPs: NFP1, NFP2

VNF-FG2 contains:

VNFs: VNF1, VNF3

CPs: CP1, CP2, CP6, CP7

VLs: VL1, VL3, VL4

NFPs: NFP3

How to create VNFFG

Prepare a VNFFGD using your favorite editor

E.g. **vim demo-vnffgd.yml**

upload the VNFFGD to tacker catalogue using openstack tacker command

E.g. **openstack vnf graph descriptor create demo-vnffgd --vnffgd-file demo-vnffgd.yml**

start a VNFFG using the VNFFGD

E.g. **openstack vnf graph create demo-vnffg --vnffgd-name demo-vnffgd**



What does a VNFFGD look like?

VNFFG Descriptor

```
tosca_definitions_version: tosca_simple_profile_for_nfv_1_0_0

description: Sample VNFFG template

topology_template:
  description: Sample VNFFG template

  node_templates:

    Forwarding_path1:
      type: tosca.nodes.nfv.FP.TackerV2
      description: creates path (CP12->CP22)
      properties:
        id: 51
        policy:
          type: ACL
          criteria:
            - name: block_tcp
              classifier:
                network_src_port_id: 640dfd77-c92b-45a3-b8fc-22712de480e1
                destination_port_range: 80-1024
                ip_proto: 6
                ip_dst_prefix: 192.168.1.2/24
            - name: block_udp
              classifier:
                network_src_port_id: 640dfd77-c92b-45a3-b8fc-22712de480eda
                destination_port_range: 80-1024
                ip_proto: 17
                ip_dst_prefix: 192.168.2.2/24
      path:
        - forwarder: VNFD1
          capability: CP12
        - forwarder: VNFD2
          capability: CP22

  groups:
    VNFFG1:
      type: tosca.groups.nfv.VNFFG
      description: HTTP to Corporate Net
      properties:
        vendor: tacker
        version: 1.0
        number_of_endpoints: 2
        dependent_virtual_link: [VL12,VL22]
        connection_point: [CP12,CP22]
        constituent_vnfs: [VNFD1,VNFD2]
      members: [Forwarding_path1]
```

Classifiers

NFP

Chain

VNFFG classifier-chain

Classifier

```
policy:
  type: ACL
  criteria:
    - name: block_tcp
      classifier:
        network_src_port_id: 640dfd77-c92b-45a3-b8fc-22712de480e1
        destination_port_range: 80-1024
        ip_proto: 6
        ip_dst_prefix: 192.168.1.2/24
    - name: block_udp
      classifier:
        network_src_port_id: 640dfd77-c92b-45a3-b8fc-22712de480eda
        destination_port_range: 80-1024
        ip_proto: 17
        ip_dst_prefix: 192.168.2.2/24
```

Chain

```
path:
  - forwarder: VNFD1
    capability: CP12
  - forwarder: VNFD2
    capability: CP22
```



VNF-FG Properties

VNF-FG Properties

groups:

VNFFG1:

type: `tosca.groups.nfv.VNFFG`

description: HTTP to Corporate Net

properties:

vendor: `tacker`

version: `1.0`

number_of_endpoints: `2`

dependent_virtual_link: `[VL12,VL22]`

connection_point: `[CP12,CP22]`

constituent_vnfs: `[VNFD1,VNFD2]`

members: `[Forwarding_path1]`

A decorative network diagram at the bottom of the slide, featuring a series of interconnected nodes and lines, representing a network topology. The nodes are represented by circles of varying sizes, some with internal details, and the lines represent connections between them.

Networking-sfc OvS driver

- OvS driver communicates with the OvS agents to configure the switches
- OvS agents install flows to steer chain traffic to the SF instances.
- The classification of traffic takes place in the Integration bridge and classifies traffic which is coming from a VM port.
- The SF forwarding takes place in the Tunnel bridge which forwards the service chain packets to the next hop Compute node via Tunnels or to the next SF port on the same Compute Node.
- MPLS headers will be used for the transport of the chain path identifier and chain hop index.



Networking-sfc ODL driver

- ODL driver is responsible to configure the Integration Bridge of OvS
- Opendaylight install flows to the OvS to steer the traffic to the SF instances
- The classifier classifies the traffic by using ACLs and encapsulates the packet with NSH header
- NSH header consists of a NSP -> RSP id and a NSIndex -> Next Hop
- The SF Forwarder which is responsible to read the NSH header and send the packet to the appropriate SF
- For the above approach NSH aware SFs are required.



Until Pike

- Create/Delete VNFFG
- One NFP
- One Classifier per Chain
- No support for Update VNFFG

In Queens

- Update VNFFG support
- Multiple Classifiers per Chain
- Empty Classifier Support



DEMO

Demo setup instructions: <https://github.com/dangtrinhnt/DynamicSFCDemo>



Demo components

- ⦿ Zabbix monitor

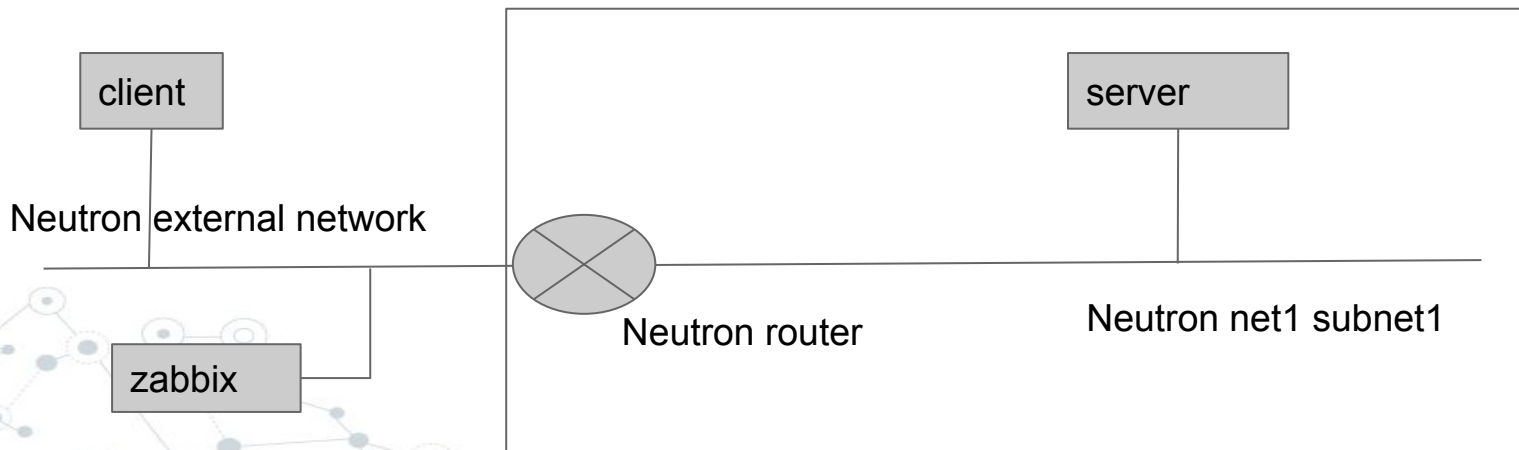
Zabbix is software that monitors numerous parameters of a network and the health and integrity of servers.

- ⦿ Client traffic generator

Client, ICMP and HTTP traffic generator, can be bare metal server or virtual machine

- ⦿ Server VM server provider

Server VM, running simple HTTP server



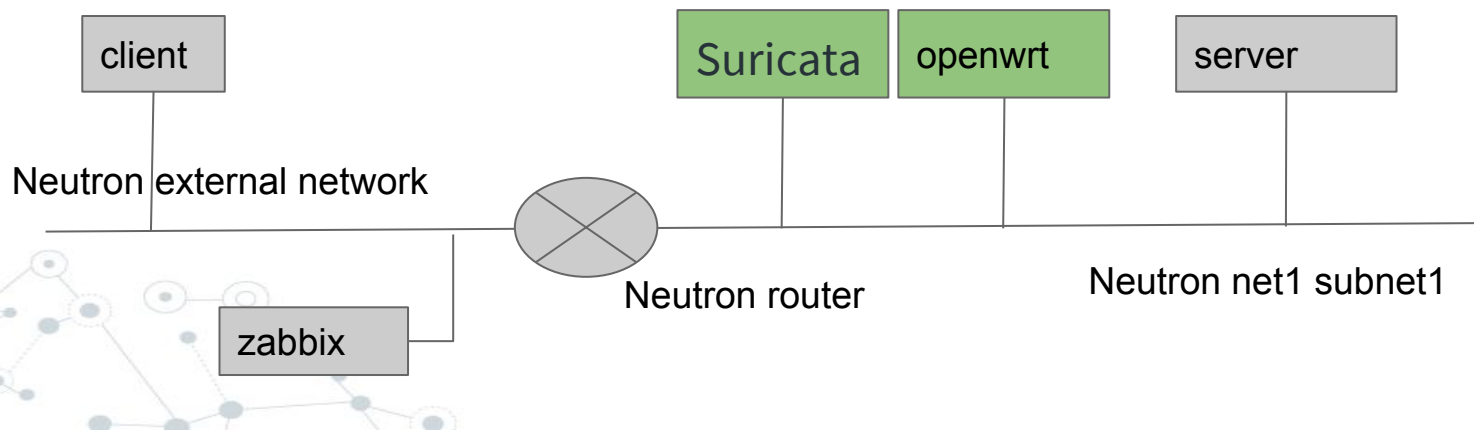
Demo components

- Suricata VNF

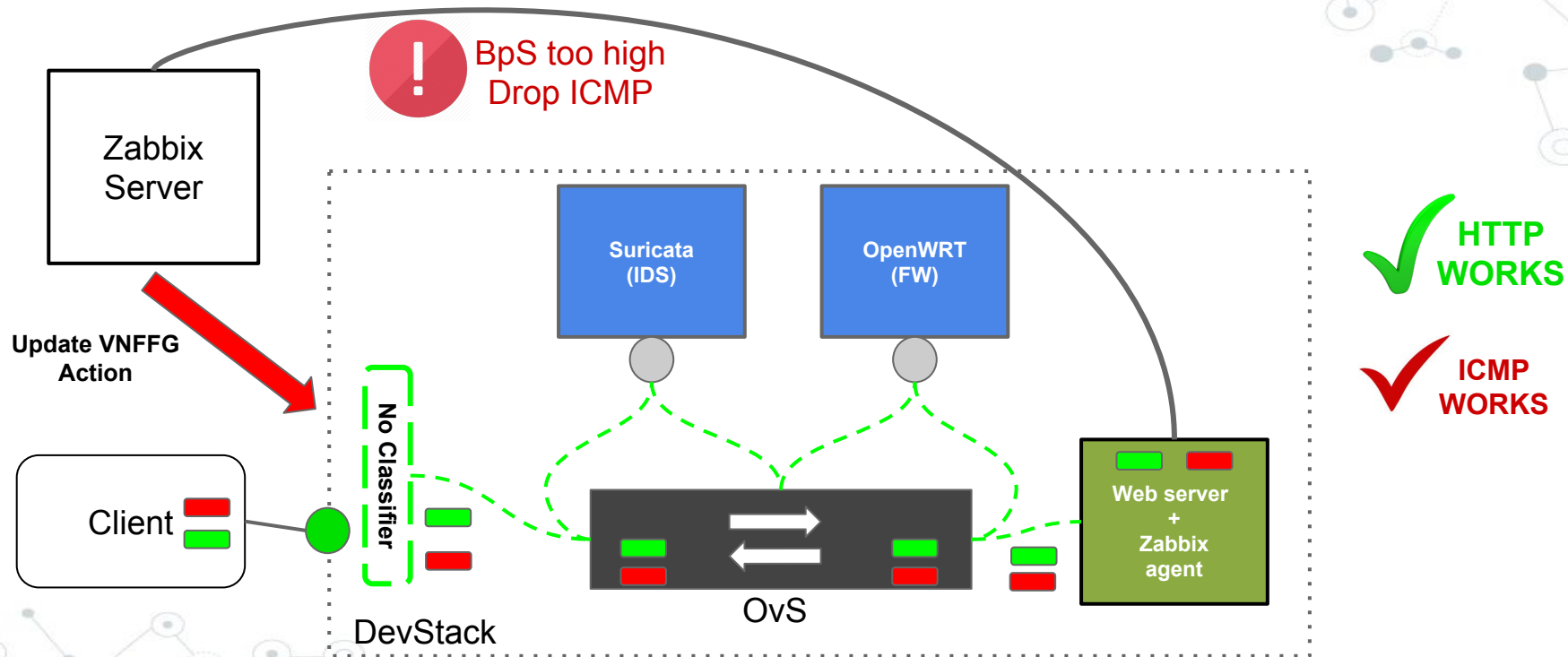
Capable of real time intrusion detection (IDS), inline intrusion prevention (IPS), and network security monitoring (NSM)

- OpenWRT VNF

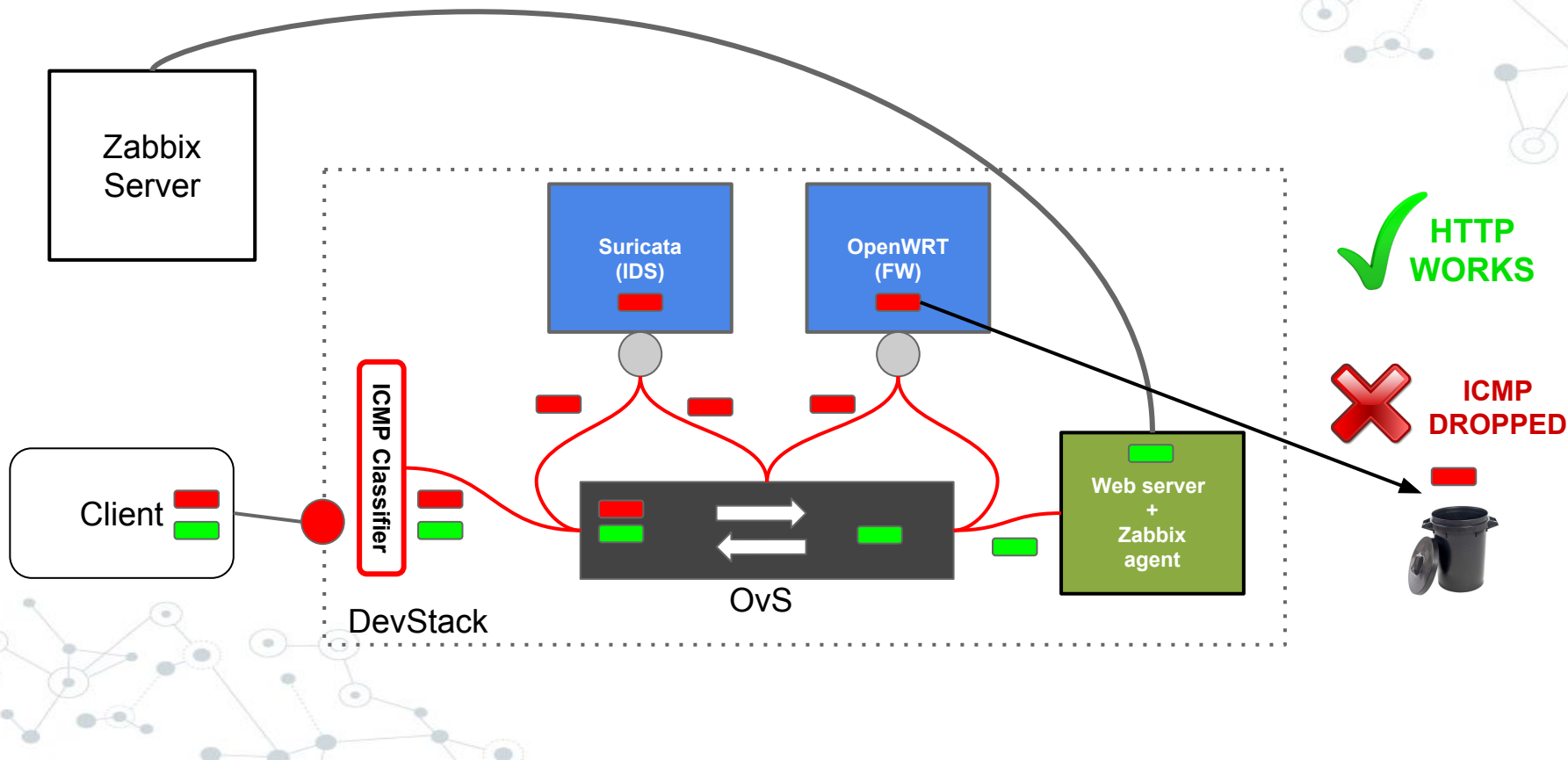
Open source project for embedded operating system based on Linux, primarily used on embedded devices to route network traffic (we are going to use it as FW).



Demo overview



Demo overview



Tacker Update VNFFG command

```
/usr/local/bin/tacker --os-username admin --os-password devstack --os-project-name admin  
--os-user-domain-name default --os-project-domain-name default --os-project-domain-id default  
--os-auth-url http://192.168.122.113/identity/v3 --os-region-name RegionOne vnffg-update  
--vnffgd-template /home/ubuntu/vnffg_block_icmp.yaml block_icmp
```

A decorative network diagram in the top right corner, featuring a cluster of interconnected nodes. Some nodes are represented by solid grey circles, while others are open circles with a grey outline. The nodes are connected by thin grey lines, forming a complex web-like structure.

Thank You

A decorative network diagram in the bottom left corner, similar to the one in the top right. It shows a smaller cluster of interconnected nodes, with some solid grey circles and some open circles with grey outlines, connected by thin grey lines.