NFV PoC #43
Towards an efficient Data Plane processing

Jokin Garay <jokin.garay@keynetic.es>

Keynetic Technologies

2017/03/21

PoC #43: Team Members

- Network Operator / Service Provider – OSM & OpenVIM:

- Manufacturer – VNF FlowNAC:

- Manufacturer – NFVI provider:

- Additional members (research):

Keynetic Technologies

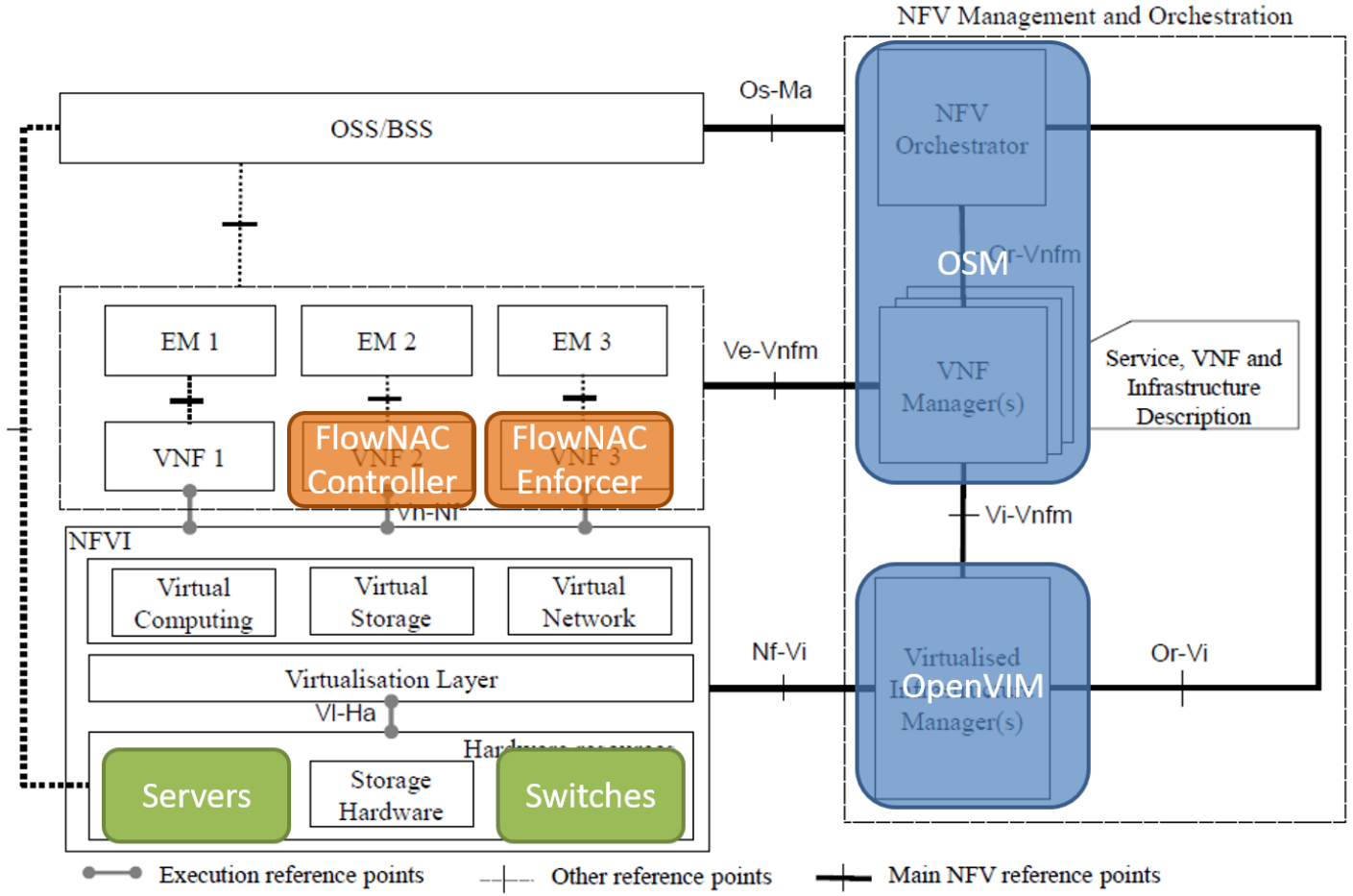
- Recently incorporated spinoff from the UPV/EHU I2T Research Group
- Background in R&D at european level and System Integration & Technology consulting
- Cybersecurity and network virtualization solutions based on SDN and NFV
- Involved in OSM since the kick off…

… and we've been in the Plugtest too

PoC #43: Mapping to NFV ISG Work

Component provided by

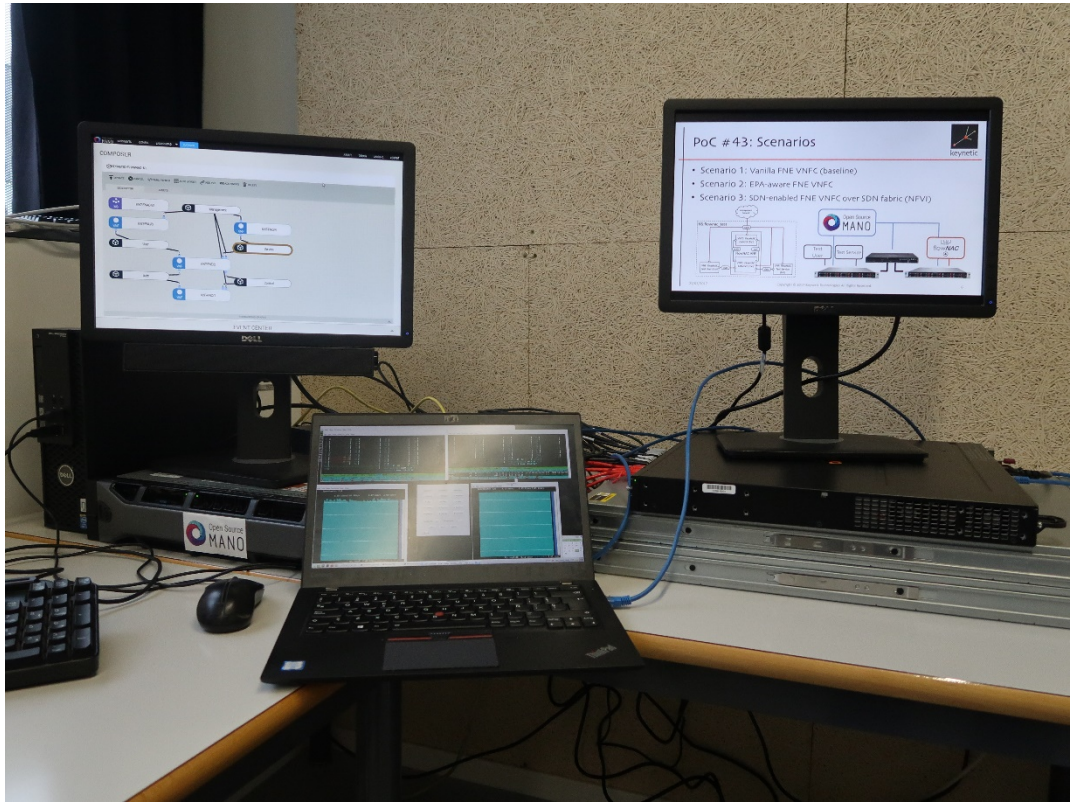
- Telefónica
- Keynetic - UPV/EHU
- HPE

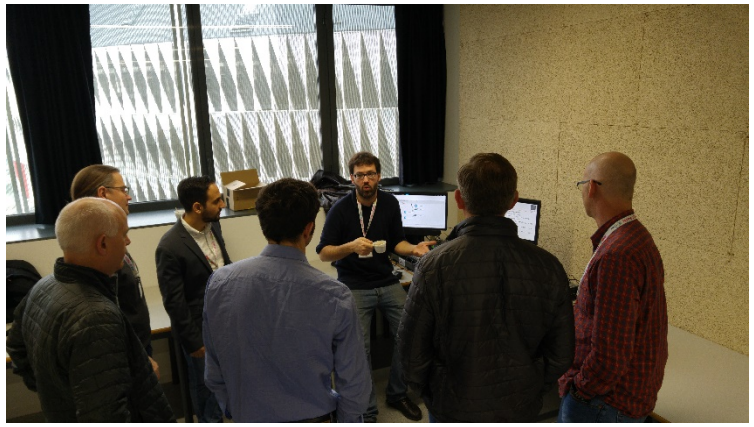


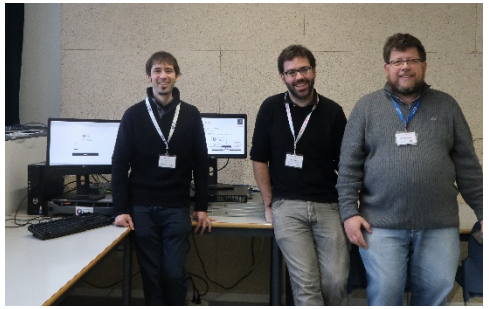
http://nfvwiki.etsi.org/index.php?title=Toward_an_efficient_dataplane_processing

PoC #43: Public demostration

- NFV#17 meeting (22nd - 25th February 2017)



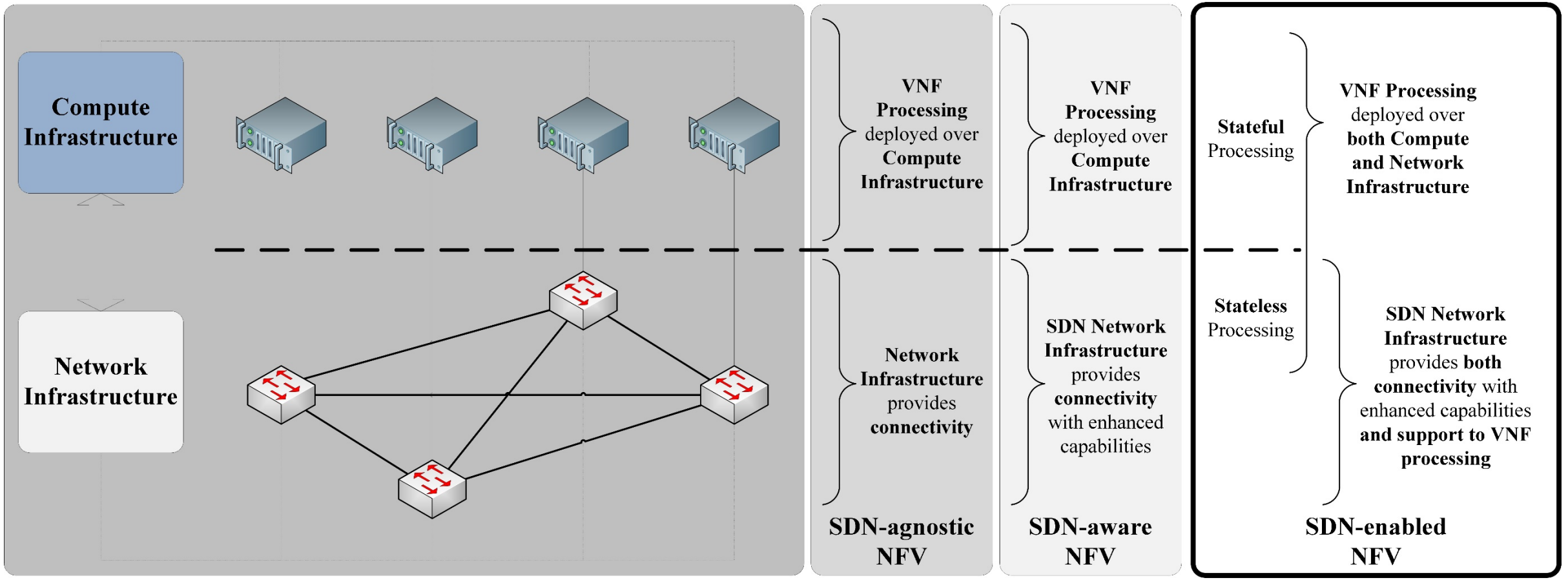




PoC #43: Goals

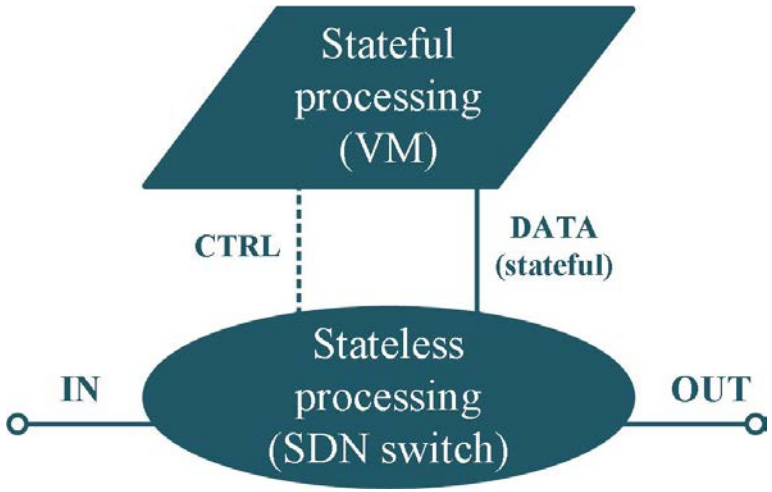
- VNF designed with separate sub-components:
  - Stateful network function components (compute)
    - … and stateless data path processing components (networking)
  - Allow each sub-component to independently scale
- Improvement of VNF data processing efficiency while minimizing the overall NFVI resources used
- Evaluate different alternatives to deploy the stateless data path processing component
  - Propose pointers to ease the integration into the ETSI NFV architectural framework

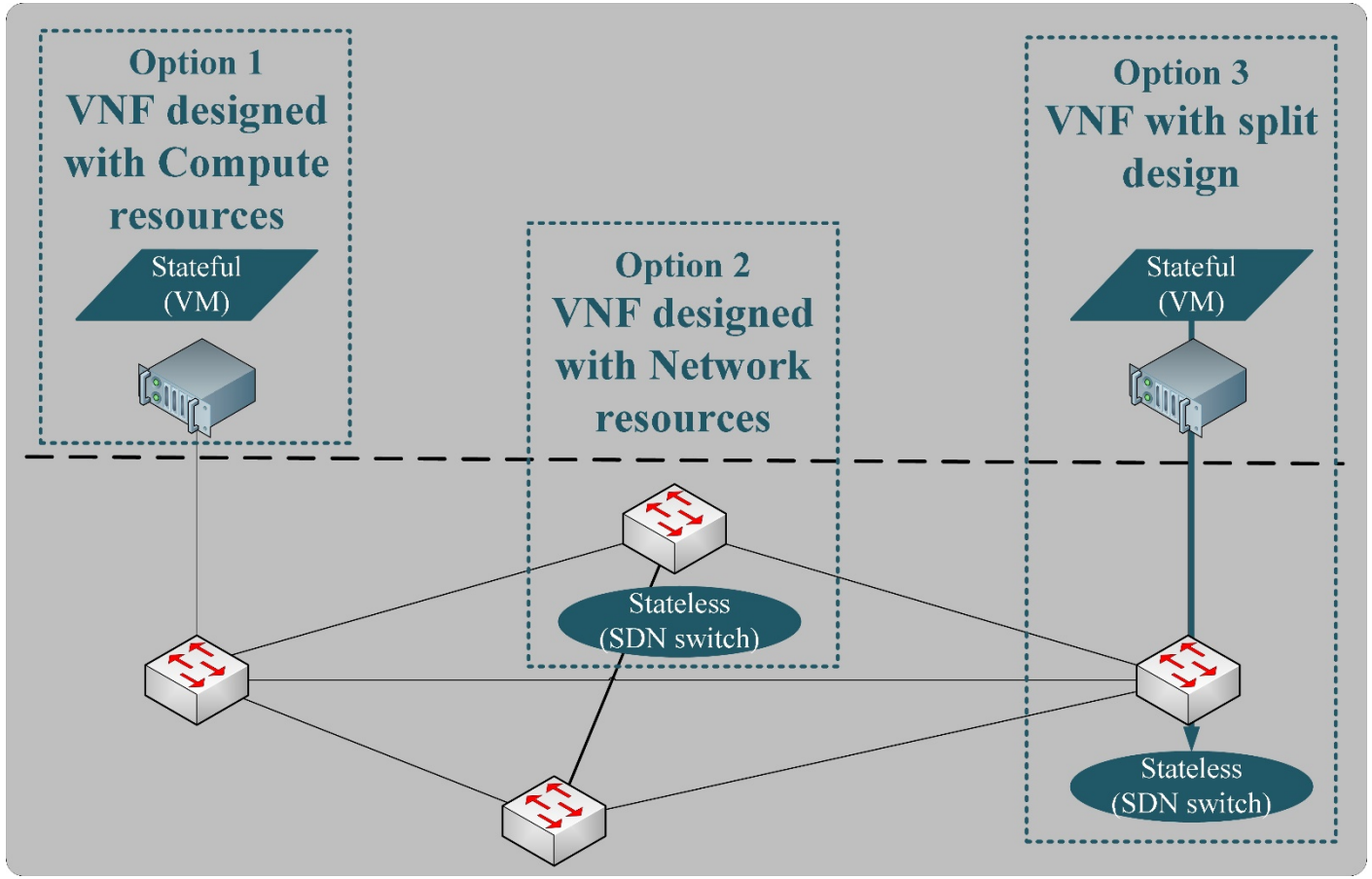
Towards SDN-enabled NFV



Matias, J., Garay, J., Toledo, N., Unzilla, J. and Jacob, E., 2015. Toward an SDN-enabled NFV architecture. *IEEE Communications Magazine*, *53*(4), pp.187-193. https://doi.org/10.1109/MCOM.2015.7081093

Towards SDN-enabled NFV





Matias, J., Garay, J., Toledo, N., Unzilla, J. and Jacob, E., 2015. Toward an SDN-enabled NFV architecture. *IEEE Communications Magazine*, *53*(4), pp.187-193. https://doi.org/10.1109/MCOM.2015.7081093

Towards SDN-enabled NFV

- Challenges arising from
  - VNF must be designed splitting the components to be deployed over compute and network resources
  - the network infrastructure must support a dual role for traffic steering and VNF processing
- Services must be redesigned, determine if the performance gain overcomes the effort involved
- Adds complexity to the optimal placement decision. The NFV framework must now orchestrate an additional type of resource with its own constraints

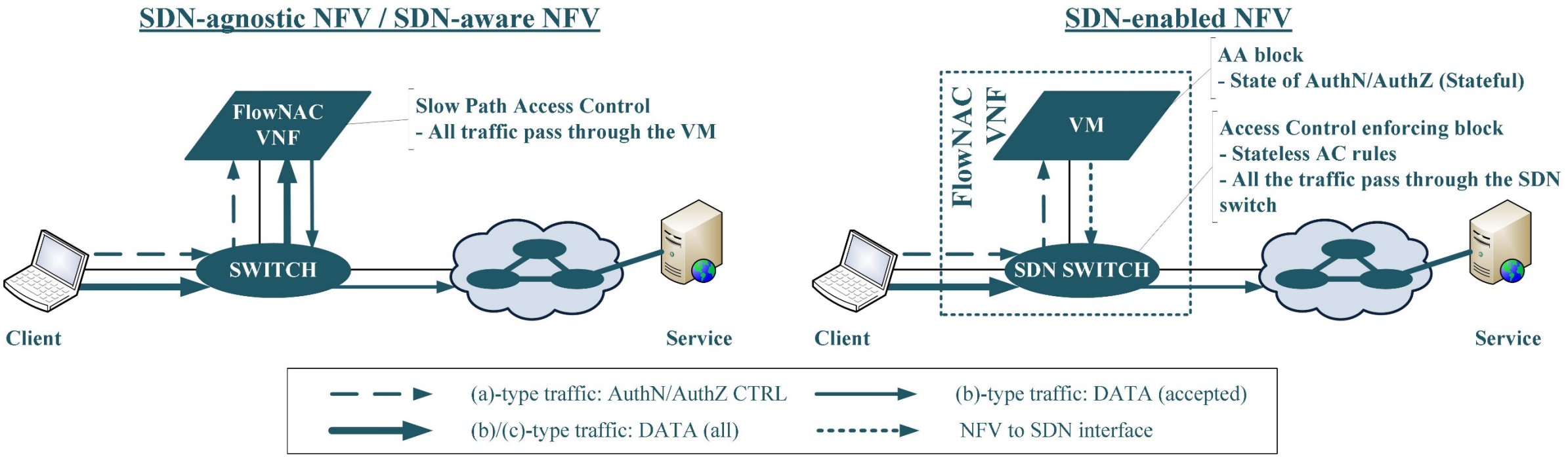
- Bring Network Virtualization to the same level as Compute Virtualization

- Dual use of the infrastructure, for traffic steering and stateless NF processing, requires the underlying network infrastructure to guarantee isolation between both functionalities
- Performance isolation is also required to avoid VNF functionality hindering the correct behavior of the NFV architecture

Matias, J., Garay, J., Toledo, N., Unzilla, J. and Jacob, E., 2015. Toward an SDN-enabled NFV architecture. *IEEE Communications Magazine*, *53*(4), pp.187-193. https://doi.org/10.1109/MCOM.2015.7081093

PoC #43: Technical Details

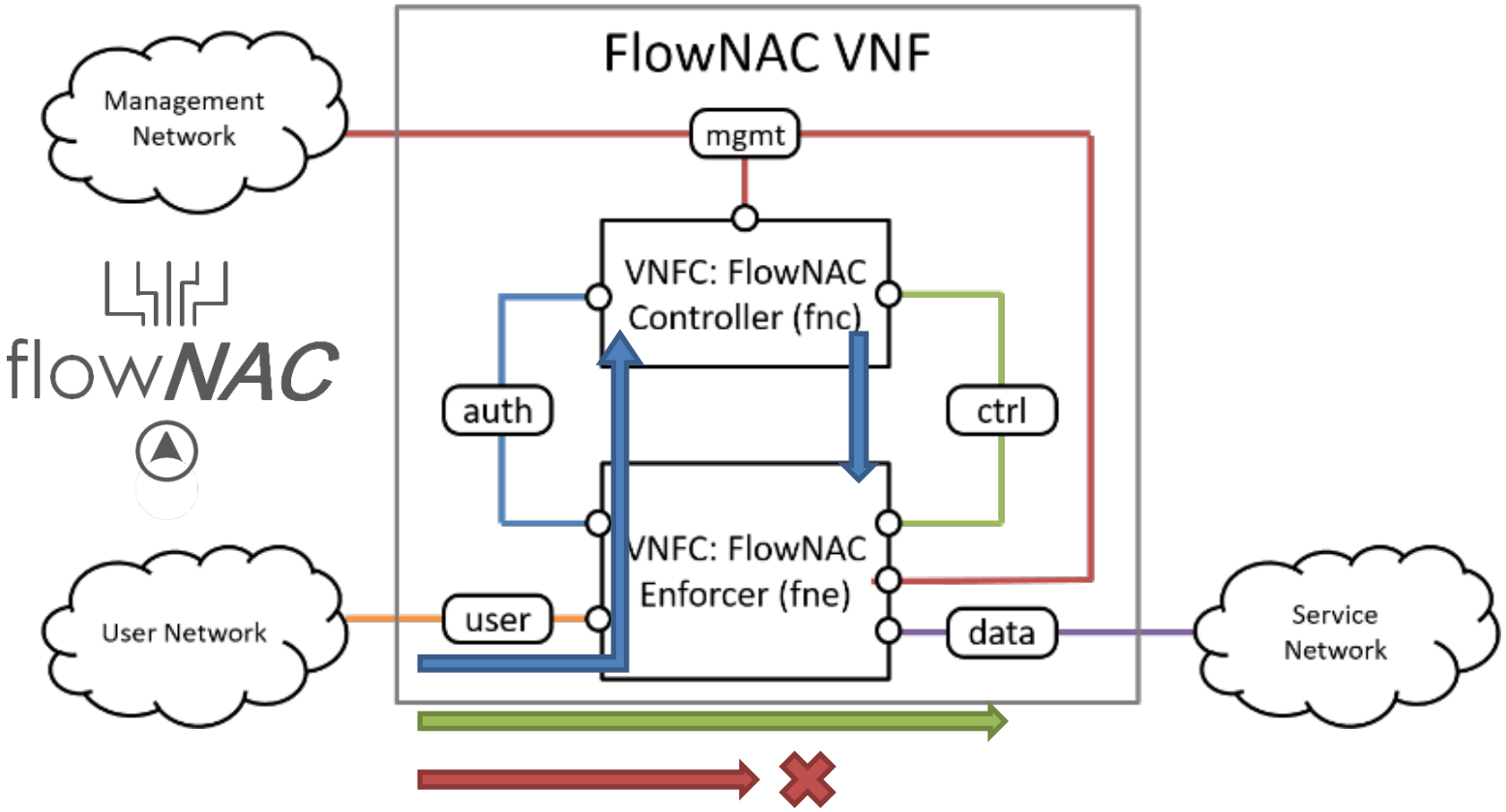
- Flow-based Network Access Control VNF



Matias, J., Garay, J., Toledo, N., Unzilla, J. and Jacob, E., 2015. Toward an SDN-enabled NFV architecture. *IEEE Communications Magazine*, *53*(4), pp.187-193. https://doi.org/10.1109/MCOM.2015.7081093

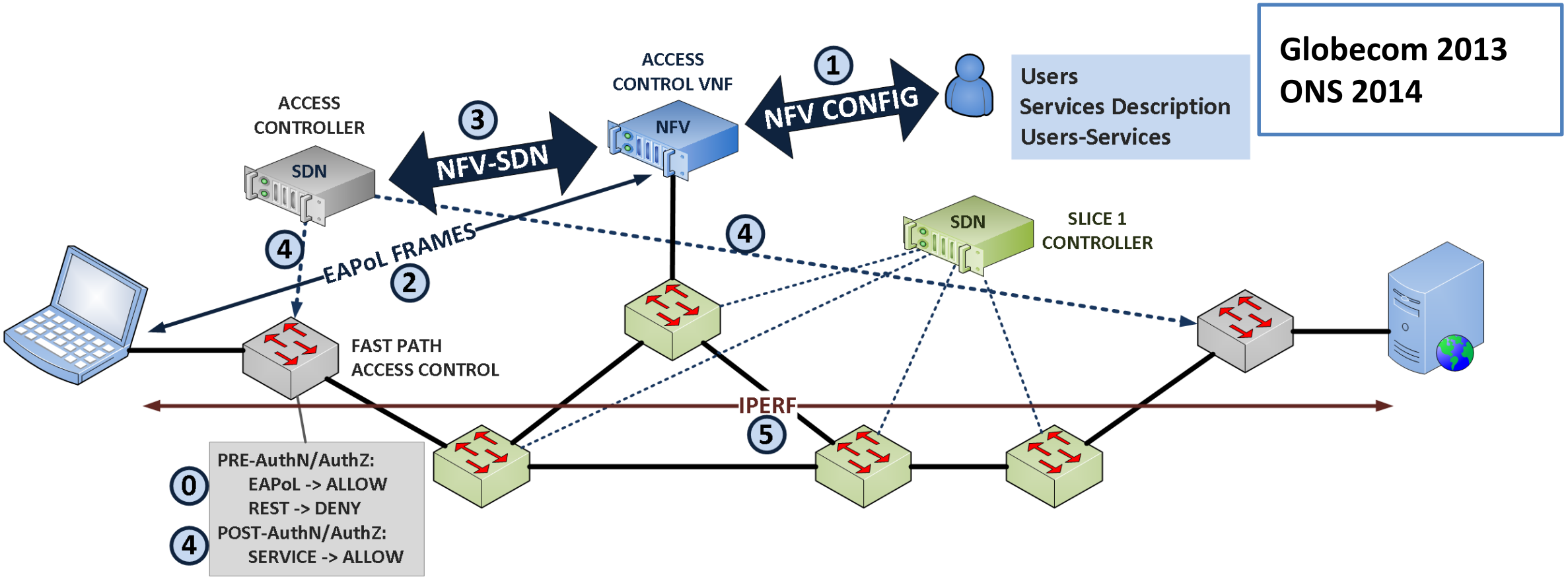
PoC #43: Technical Details

- Flow-based Network Access Control VNF



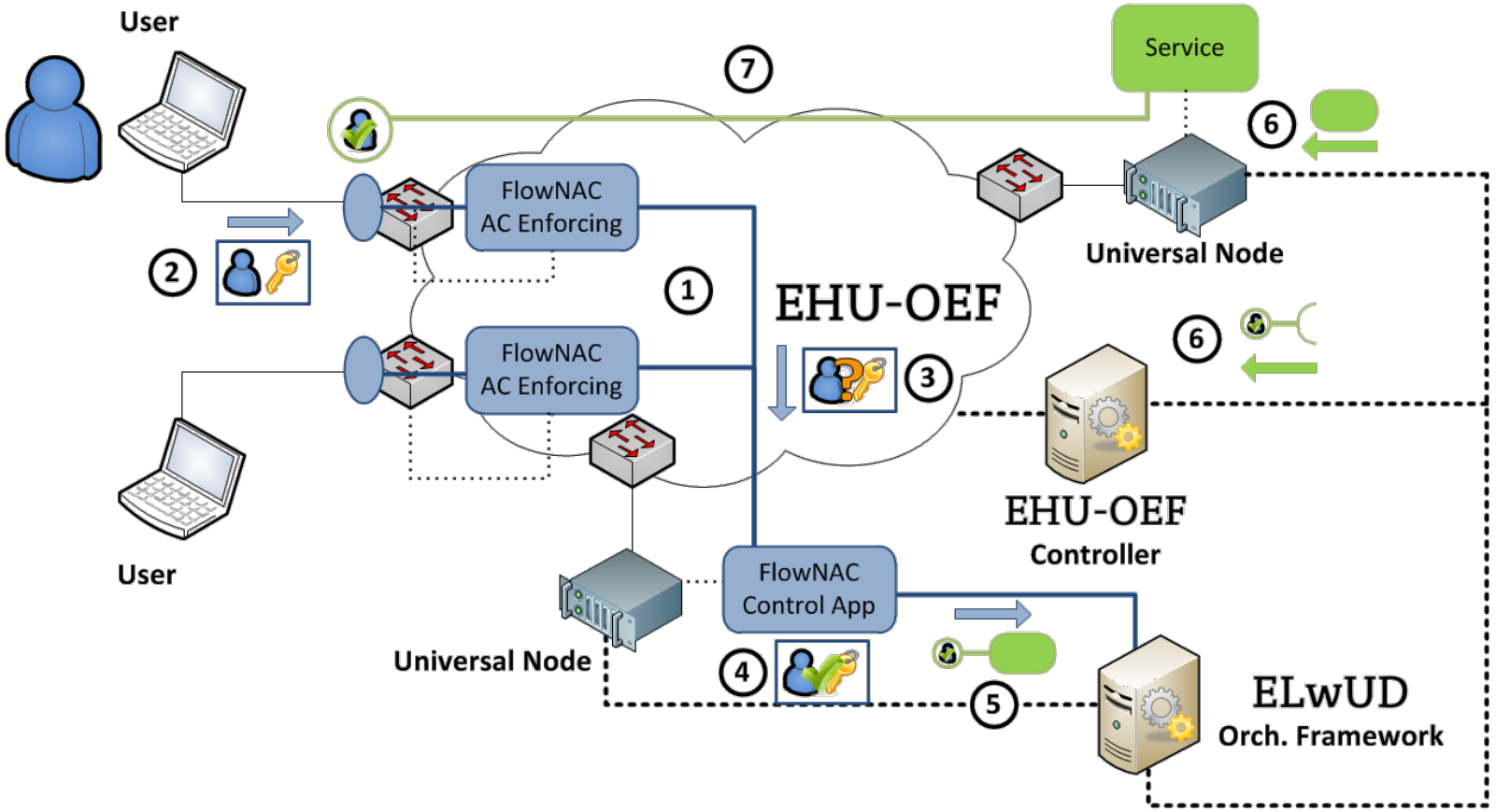
FlowNAC - Demo I

- Access Control VNF use case



FlowNAC - Demo II

- Self-deploying Service Graphs over ELwUD

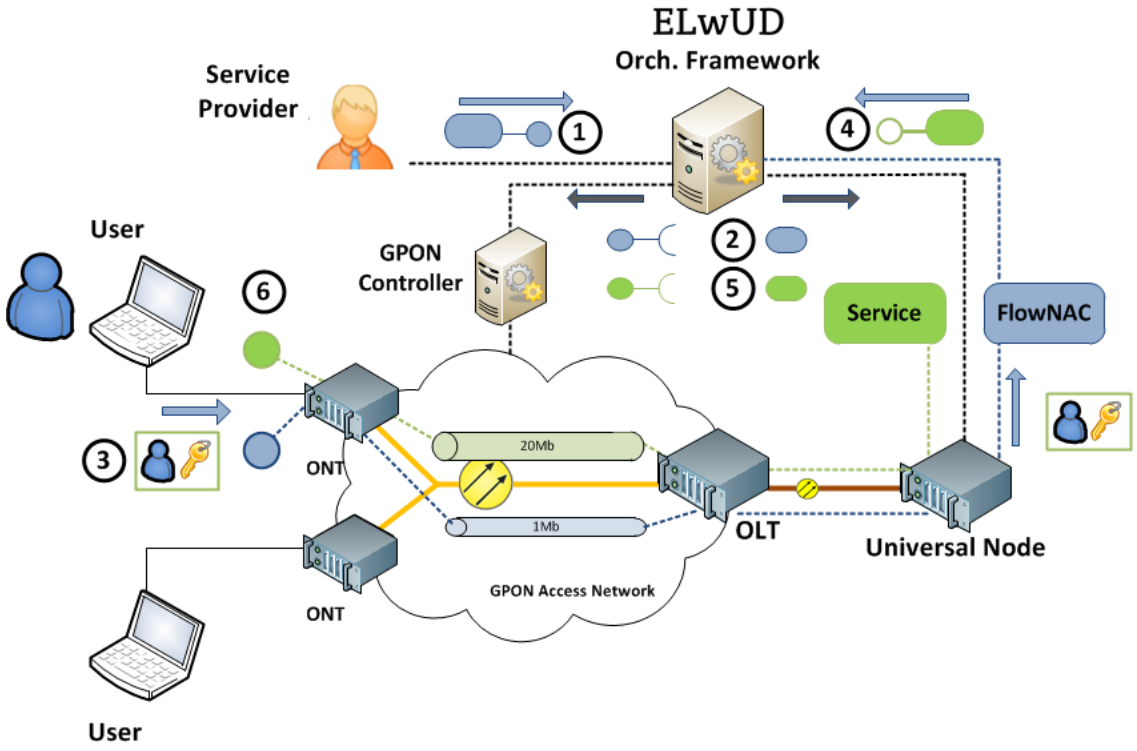


ONS S3 2015
IEEE NetSoft 2015

FlowNAC - Demo III

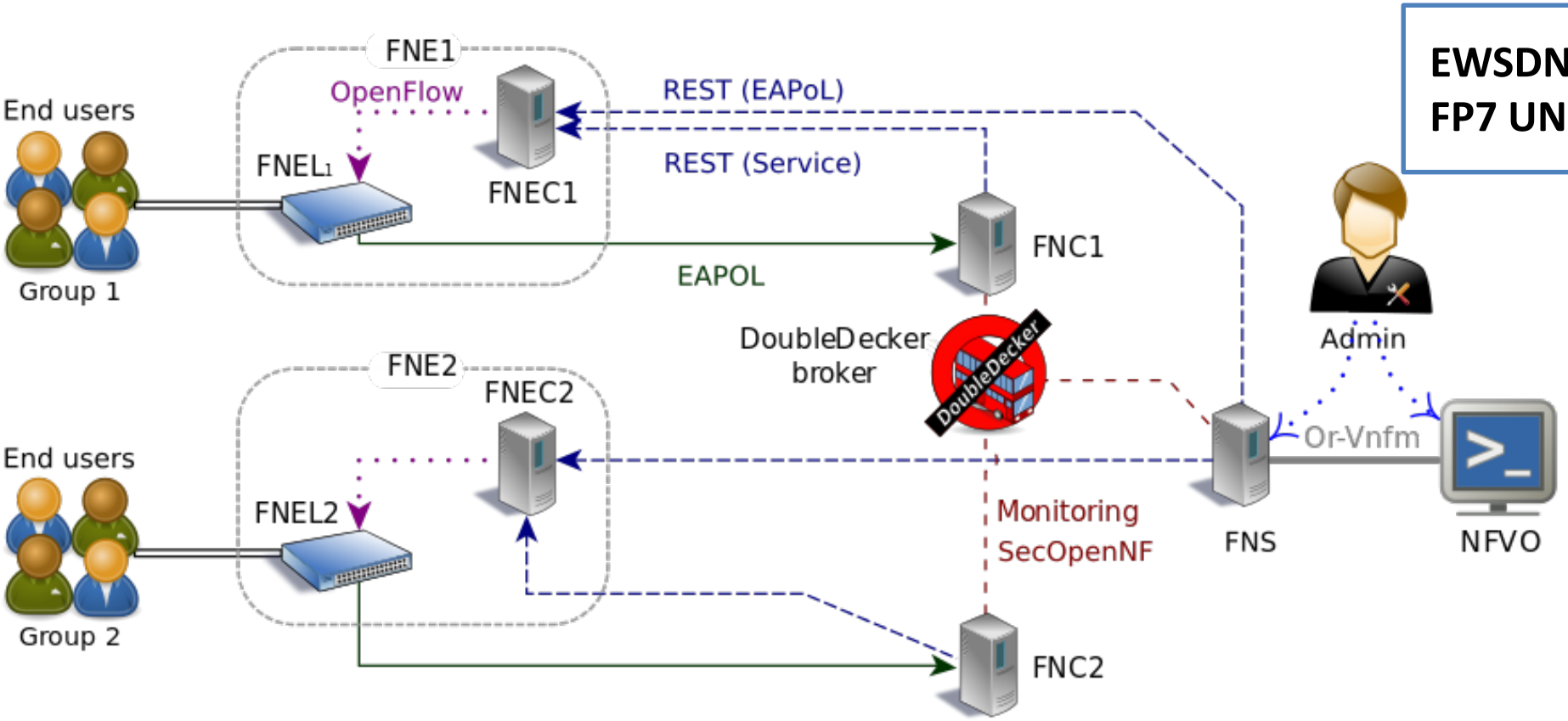
- On demand high speed access over Optical Networks

EWSDN 2015



FlowNAC - Demo IV

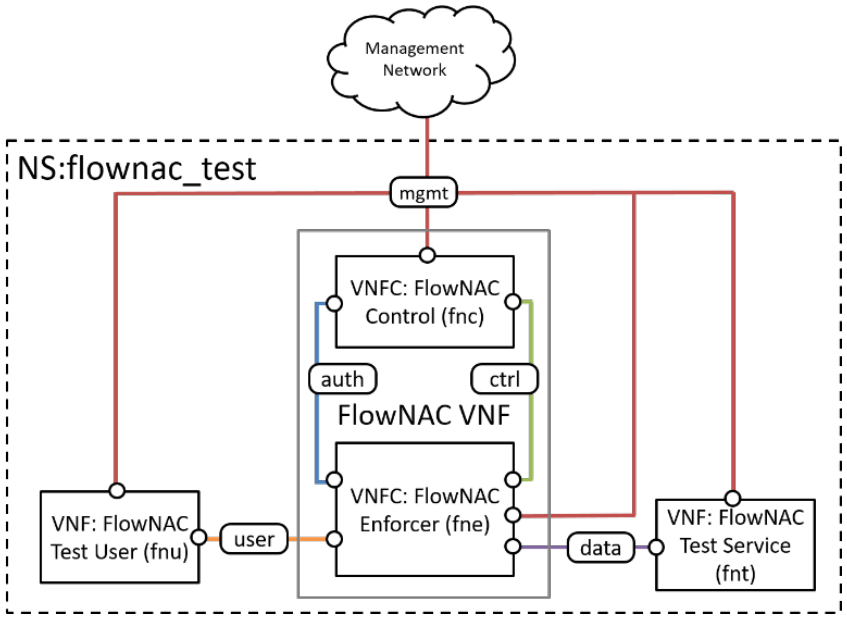
- Autoscable and Resilient Network Service

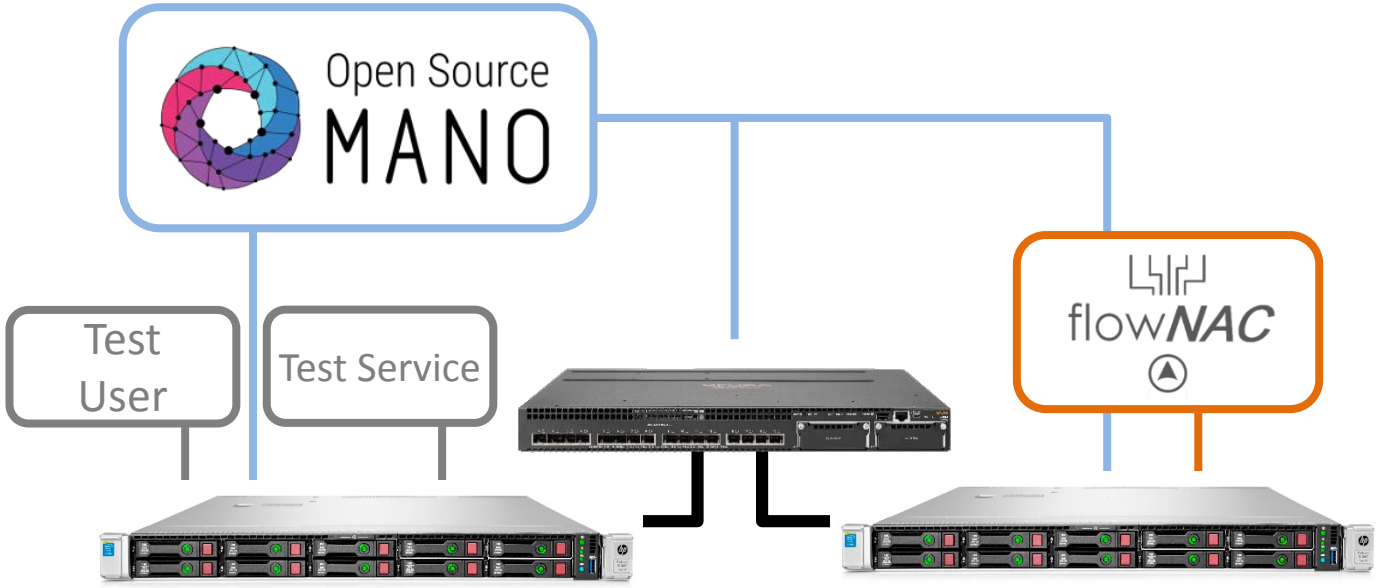


EWSDN 2016
FP7 UNIFY

PoC #43: Scenarios

- Scenario 1: Vanilla FNE VNFC (baseline)
- Scenario 2: EPA-aware FNE VNFC
- Scenario 3: SDN-enabled FNE VNFC over SDN fabric (NFVI)

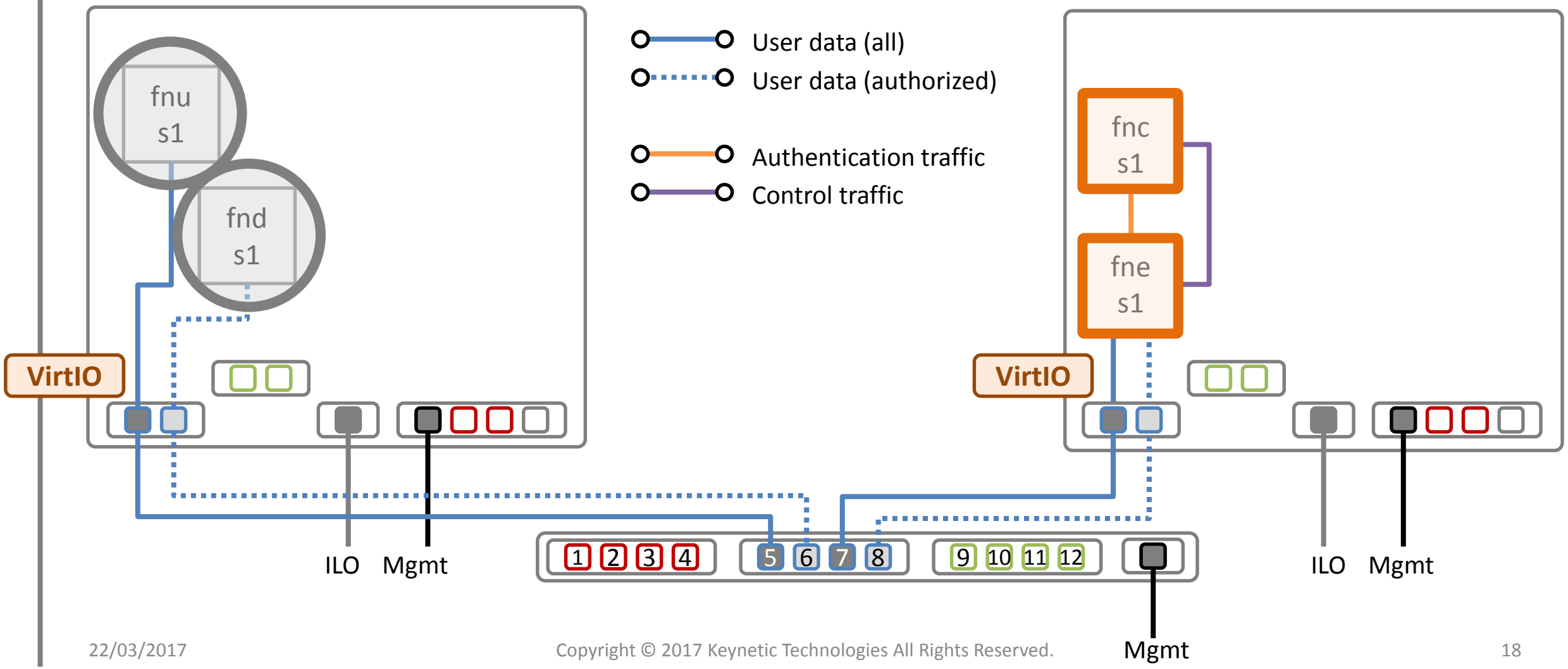




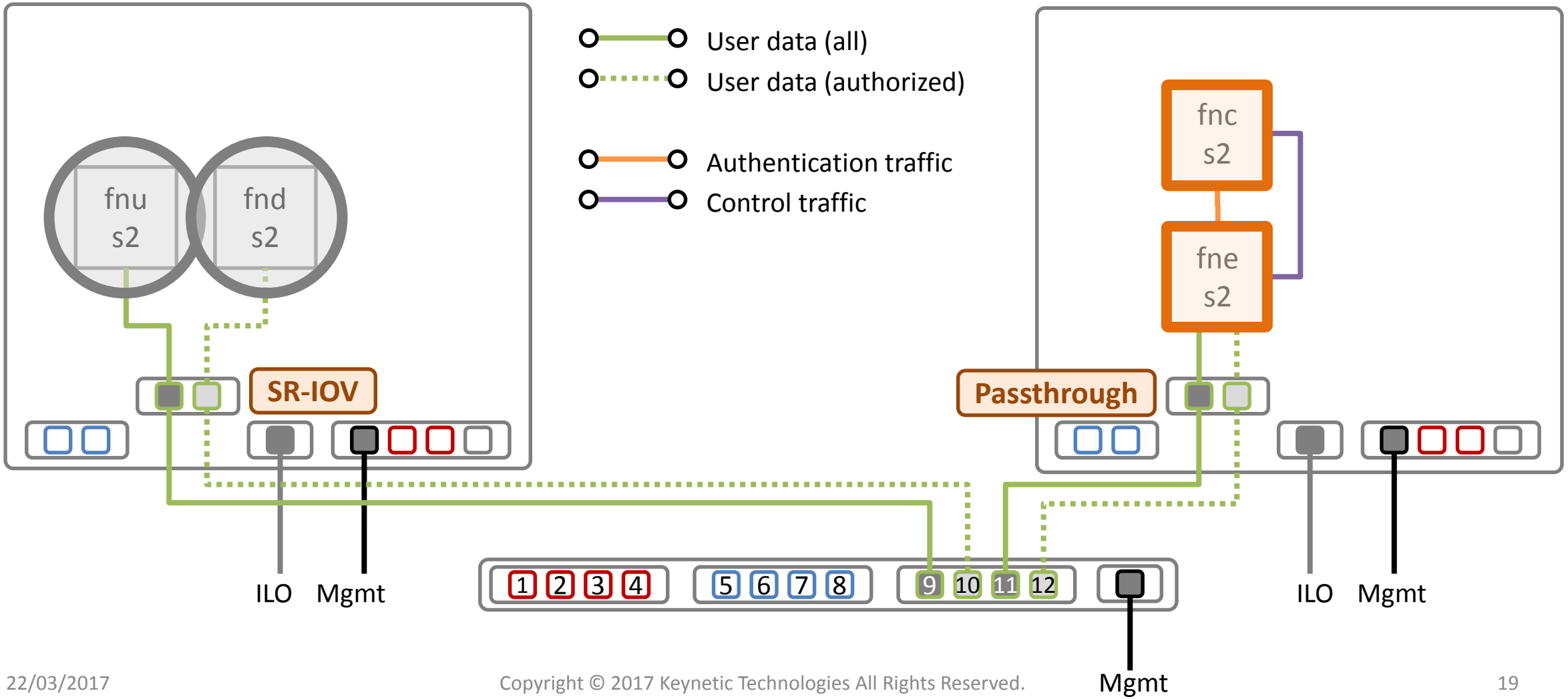
PoC #43: VNF Resources

VNF	Scenario 1 Vanilla	Scenario 2 EPA-aware	Scenario 3 SDN-enabled
FNC	Compute-based, VirtIO networking		
FNE	Compute-based, VirtIO networking	Compute-based, PCI Passthrough networking	SDN-based
FNU FND	Compute-based, VirtIO networking	Compute-based, SR-IOV networking	

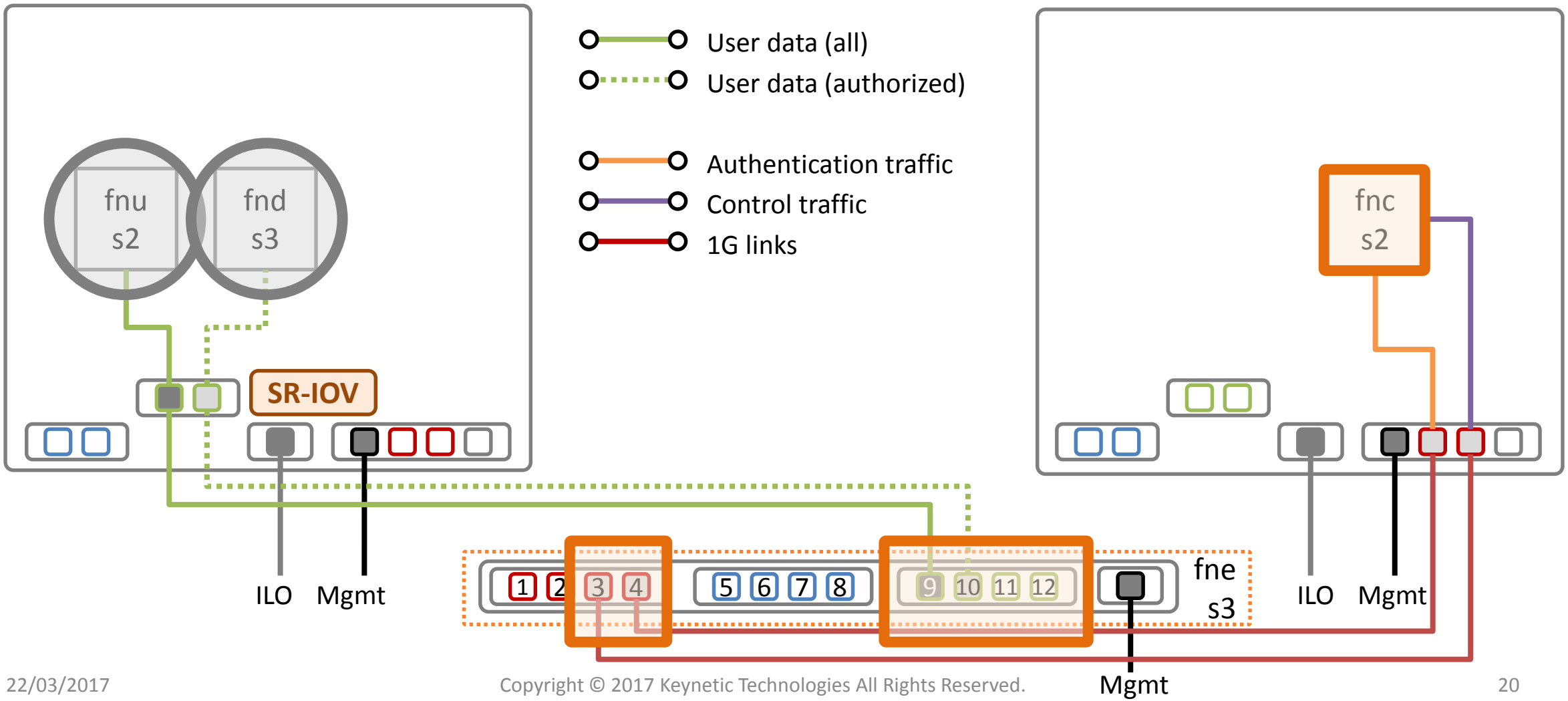
PoC #43: Scenario 1



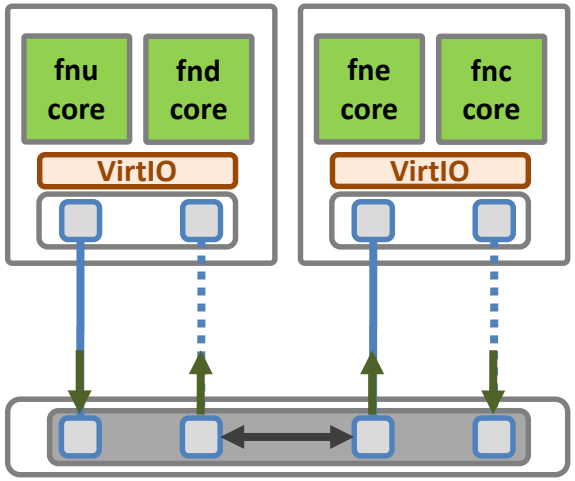
PoC #43: Scenario 2



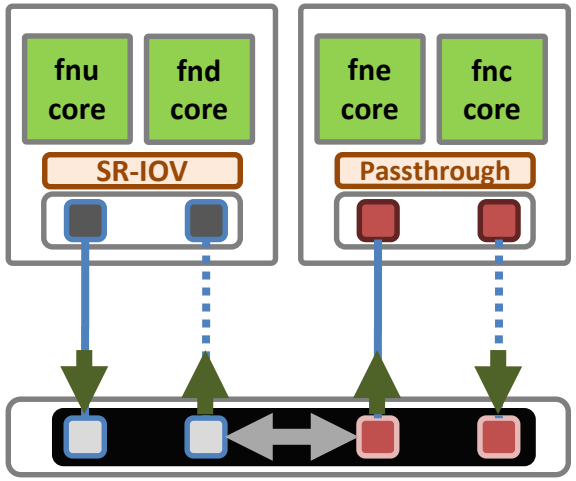
PoC #43: Scenario 3



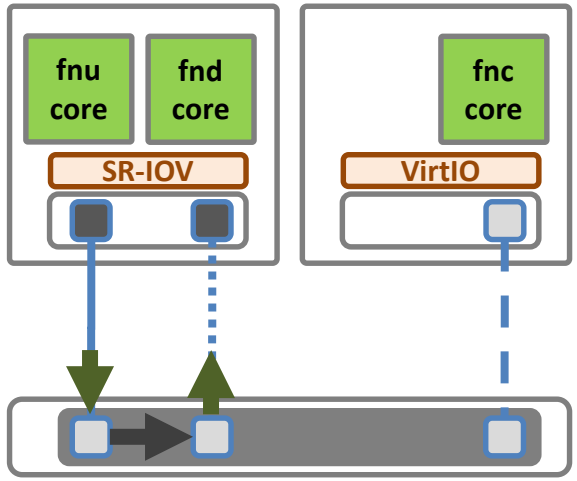
PoC #43: Resources per Scenario



Compute	
Cores	4
Ports	4 (vlan)
Networking	
Ports	4 (vlan)
Traffic	2x **3.5 Gbps** *



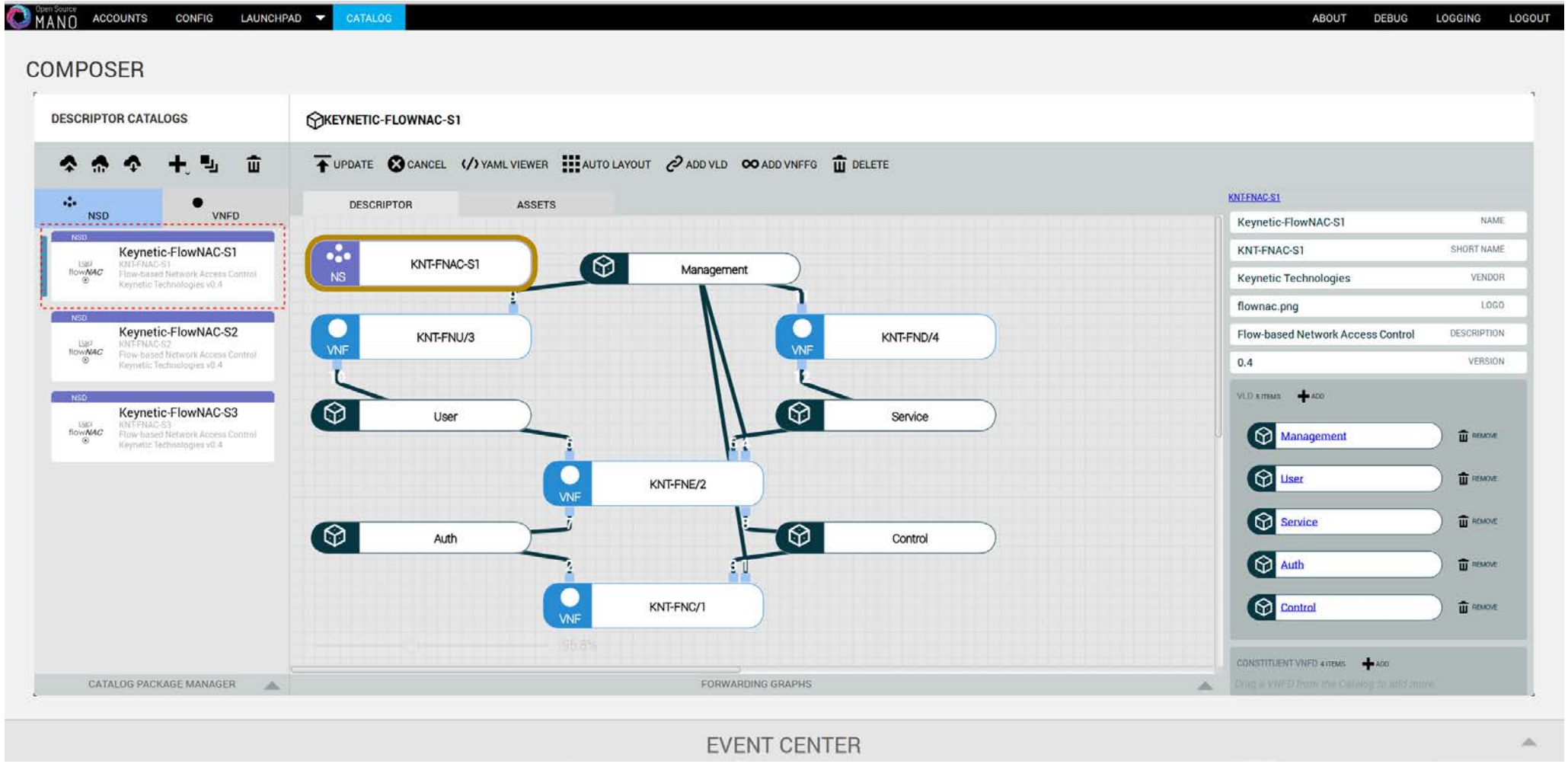
Compute	
Cores	4
Ports	**2 (sr-iov/vlan)** **2 (all)**
Networking	
Ports	2 (vlan) + **2 (all)**
Traffic	2x **9.8 Gbps** *



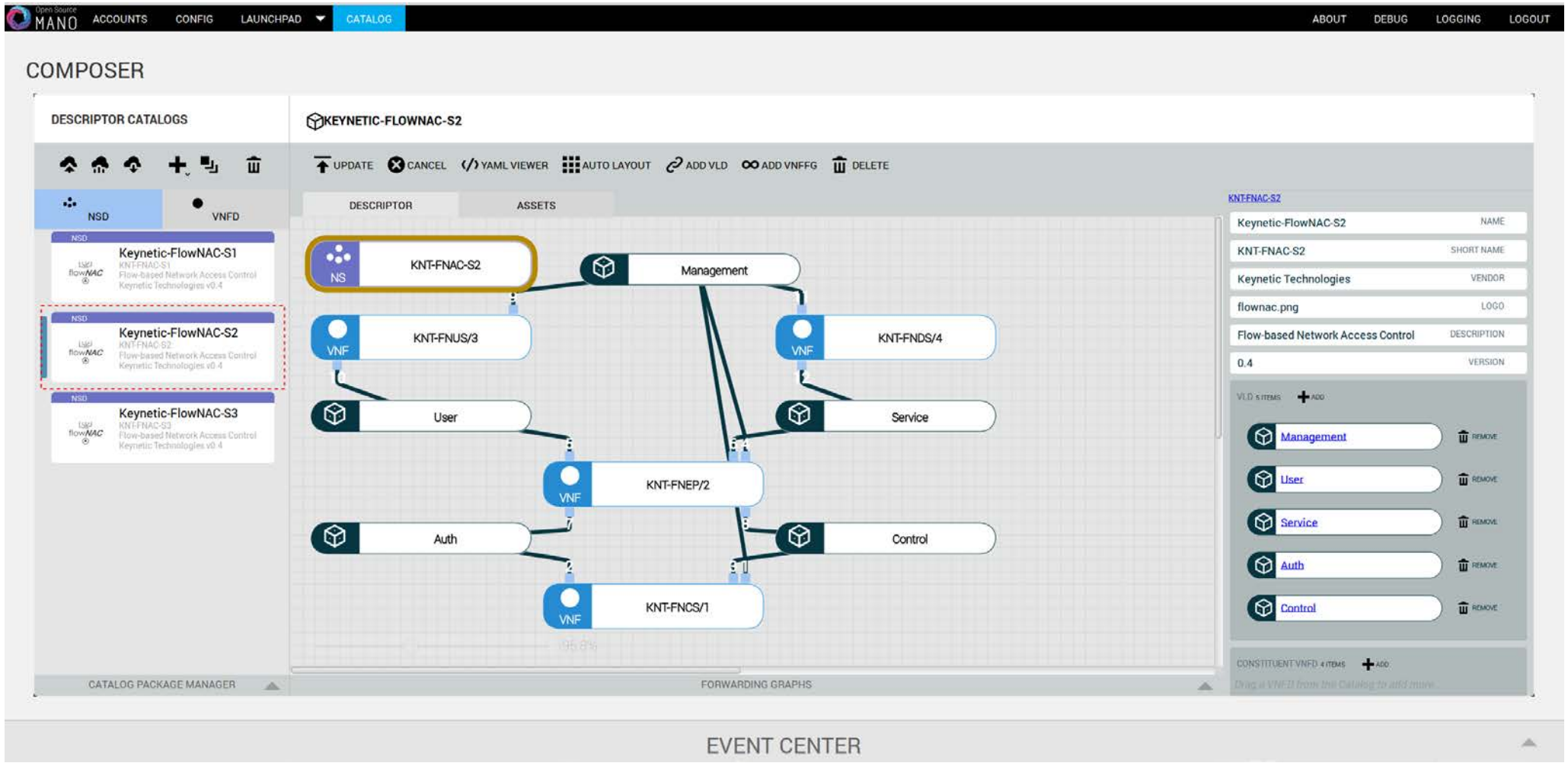
Compute	
Cores	3
Ports	**2 (sr-iov/vlan)** *1 (vlan)*
Networking	
Ports	2 (vlan) *+ 1 (vlan)*
Traffic	1x **9.8 Gbps** *

* 1500 byte frames

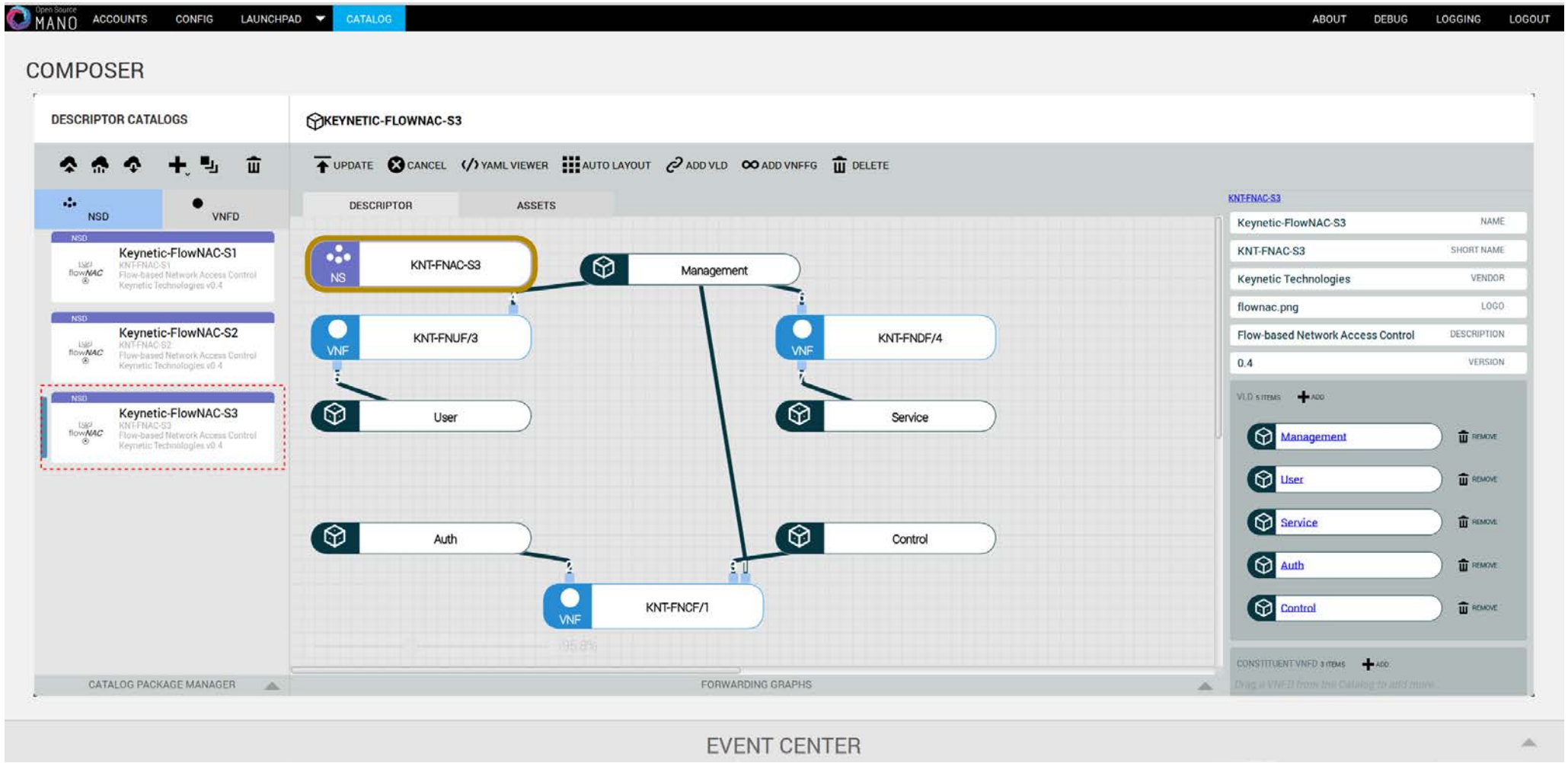
PoC #43: Scenario 1 - NS



PoC #43: Scenario 2 - NS



PoC #43: Scenario 3 - NS



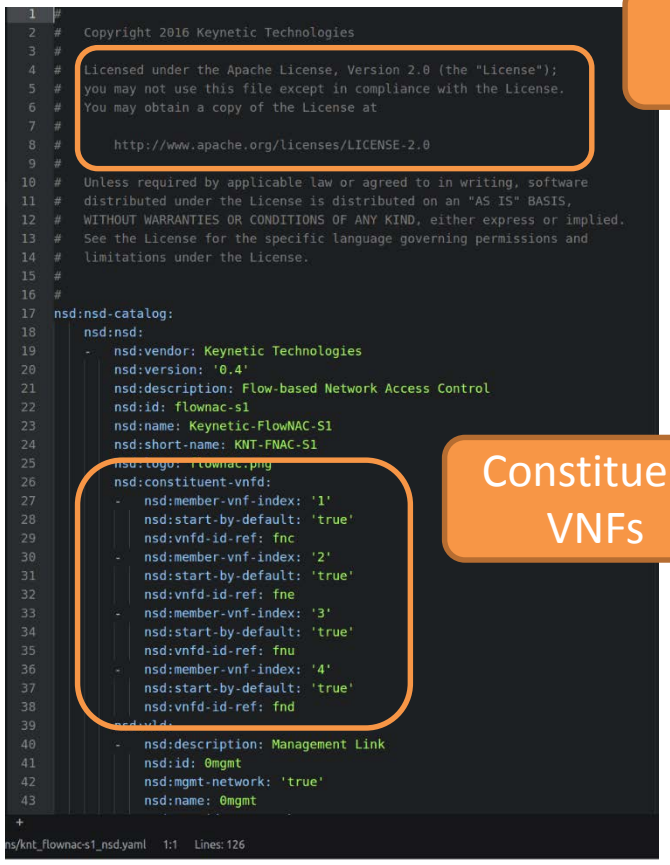
PoC #43: NS Descriptor - Scenario 1

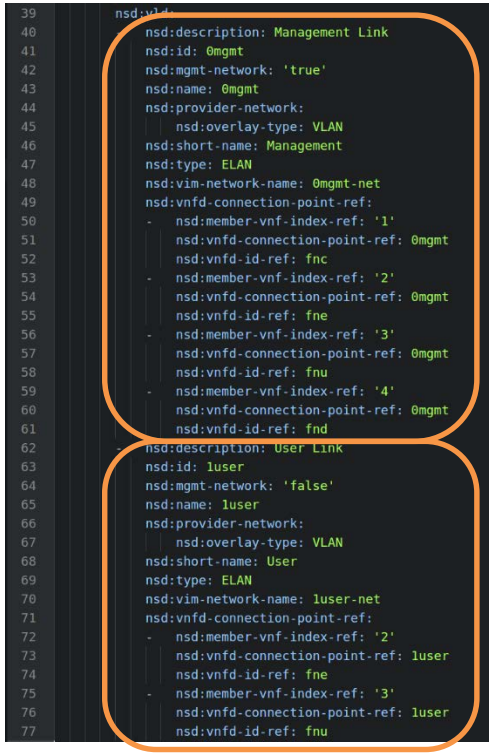
- Mostly a VNF / VL enumeration

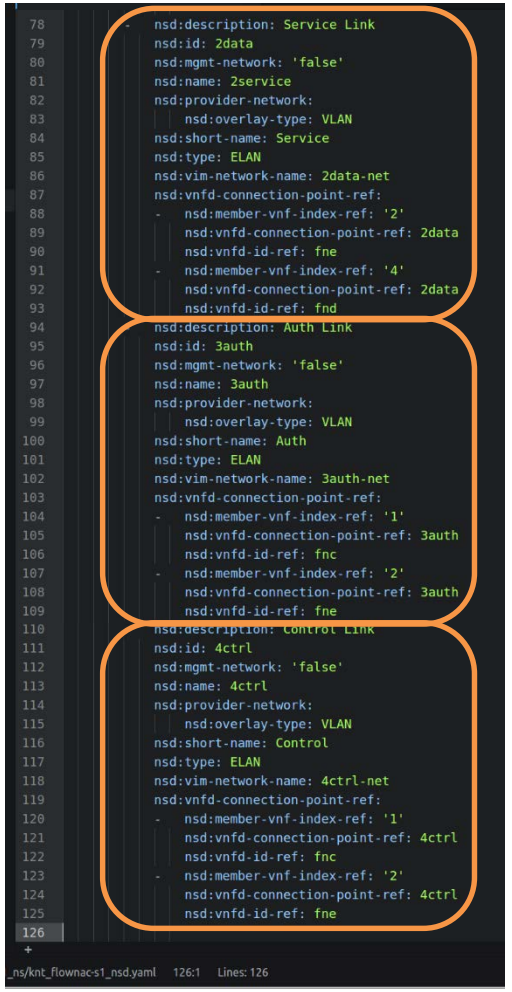
Descriptors in OSM repo under Apache 2 license

Constituent VNFs

Constituent VLs







PoC #43: VNF Descriptors - FNC

- Same descriptor all cases

Header info

Connection points

Interfaces

EPA requirements

Image requirements

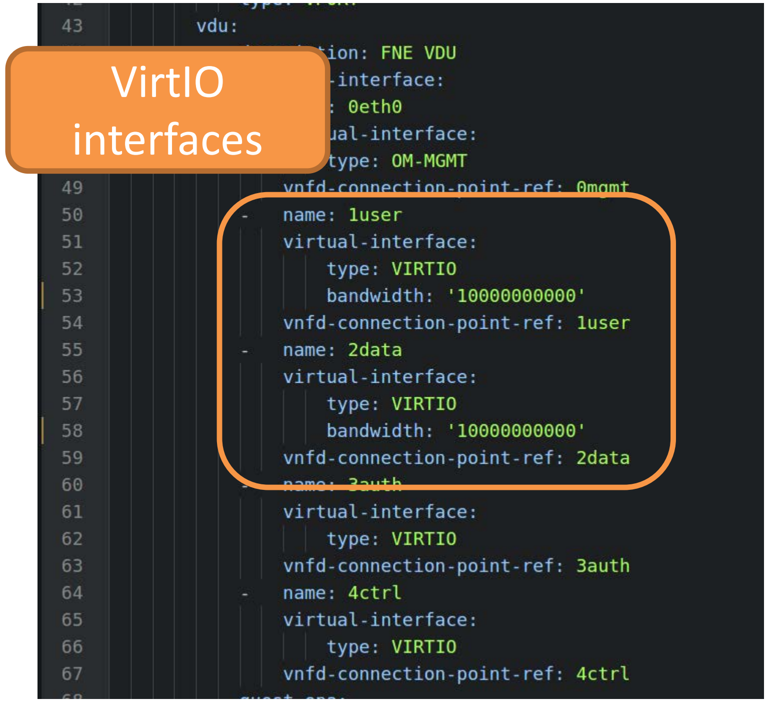
Configuration primitives

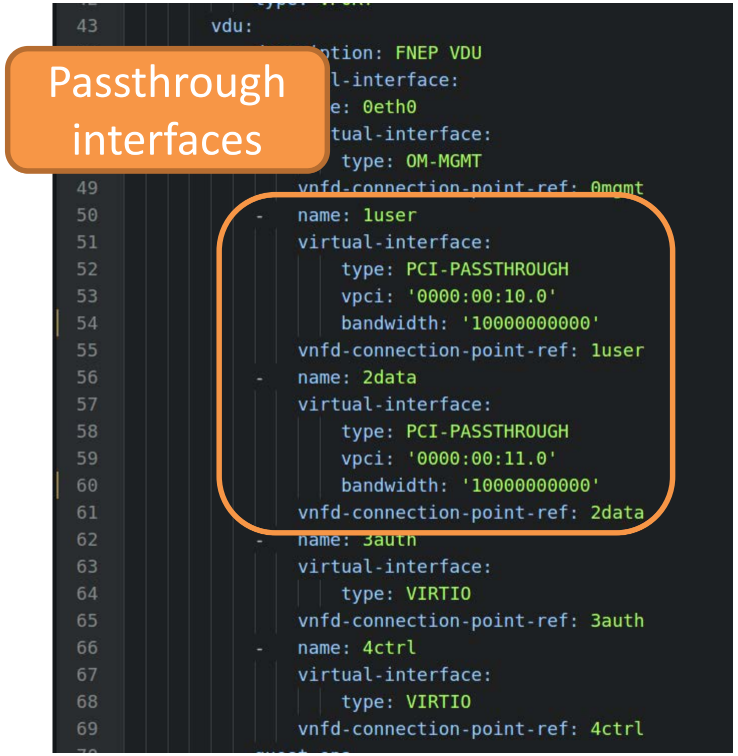
Initial configuration

```
vnfd:vnfd-catalog:
    vnfd:
        vendor: Keynetic Technologies
        version: '0.4'
        description: FlowNAC Control
        id: fnc
        name: Keynetic-FNC
        short-name: KNT-FNC
        logo: flownac.png
        service-function-chain: UNAWARE
        connection-point:
        -   id: 0mgmt
            name: 0mgmt
            type: VPORT
        -   id: 3auth
            name: 3auth
            type: VPORT
        -   id: 4ctrl
            name: 4ctrl
            type: VPORT
        vdu:
        -   description: FNC VDU
            external-interface:
            -   name: 0eth0
                virtual-interface:
                    type: OM-MGMT
                vnfd-connection-point-ref: 0mgmt
            -   name: 3auth
                virtual-interface:
                    type: VIRTIO
                vnfd-connection-point-ref: 3auth
            -   name: 4ctrl
                virtual-interface:
                    type: VIRTIO
                vnfd-connection-point-ref: 4ctrl
            guest-epa:
                cpu-pinning-policy: SHARED
                cpu-thread-pinning-policy: ISOLATE
                mempage-size: LARGE
                numa-node-policy:
                    mem-policy: STRICT
            id: fnc-vdu
            image: fnc
            name: fnc-vdu
            vm-flavor:
                memory-mb: '2048'
                storage-gb: '4'
                vcpu-count: '2'
        vnf-configuration:
            config-attributes:
                config-delay: 10
            service-primitive:
            -   name: config
                parameter:
                -   name: ssh-hostname
                    data-type: STRING
                    mandatory: 'true'
                -   name: ssh-username
                    data-type: STRING
                    mandatory: 'true'
                -   name: ssh-password
                    data-type: STRING
                    mandatory: 'true'
                -   name: mode
                    data-type: STRING
                    read-only: 'true'
                    default-value: fnc
            -   name: start
            -   name: stop
            -   name: restart
            initial-config-primitive:
            -   name: config
                parameter:
                -   name: ssh-hostname
                    value: <rw_mgmt_ip>
                -   name: ssh-username
                    value: flownac
                -   name: ssh-password
                    value: fnac@osm
                -   name: mode
                    value: fnc
                seq: '1'
            -   name: start
                seq: '2'
            juju:
                charm: flownac
```

PoC #43: VNF Descriptors - FNE

- Different descriptors in Scenario 1 and Scenario 2
  - – Change virtual interface type: VirtIO → Passthrough





PoC #43: VNF Descriptors - FN{U|D}

- Descriptors in Scenario 1 and Scenarios 2/3
  - Change virtual interface type: VirtIO → SR-IOV

VirtIO interface

```
vdu:
-   description: FNU VDU
    external-interface:
    -   name: 0eth0
        virtual-interface:
            type: OM-MGMT
        vnfd-connection-point-ref: 0mgmt
        name: 1user
        virtual-interface:
            type: VIRTIO
            bandwidth: '10000000000'
        vnfd-connection-point-ref: 1user
    guest-epa:
        cpu-pinning-policy: SHARED
        cpu-thread-pinning-policy: ISOLATE
        mempage-size: LARGE
        numa-node-policy:
            mem-policy: STRICT
    id: fnu-vdu
    image: fnu
    name: fnu-vdu
    vm-flavor:
        memory-mb: '2048'
        storage-gb: '4'
        vcpu-count: '2'
```

SR-IOV interface

```
vdu:
-   description: FNUS VDU
    external-interface:
    -   name: eth0
        virtual-interface:
            type: OM-MGMT
        vnfd-connection-point-ref: 0mgmt
    -   name: user
        virtual-interface:
            type: SR-IOV
            vpci: '0000:05:10.0'
            bandwidth: '10000000000'
        vnfd-connection-point-ref: 1user
    guest-epa:
        cpu-pinning-policy: SHARED
        cpu-thread-pinning-policy: ISOLATE
        mempage-size: LARGE
        numa-node-policy:
            mem-policy: STRICT
    id: fnus-vdu
    image: fnus
    name: fnus-vdu
    supplemental-boot-data:
        boot-data-drive: 'false'
    vm-flavor:
        memory-mb: '2048'
        storage-gb: '4'
        vcpu-count: '2'
```

PoC #43: Process

Step	Scenario 1 Vanilla	Scenario 2 EPA-aware	Scenario 3 SDN-enabled
Instantiate	Instantiate NS from OSM • Provision VNFs • Provision paths (VLANs) • Initial VNF configuration		Very similar to what's done for the compute-based VNFs
Operate	Launch actions through configuration channel • Check access before authentication • Authenticate • Measure resource usage and performance		

PoC #43: Initial learnings & Next steps

- Resources efficiency
  - Less cores, less backplane, less (dedicated) ports
- Bump-in-the-wire requirements
  - Careful with VIM networking ‘interference’
  - Passthrough (dedicated ports)
- Interfaces not ready for networking resources
  - (Currently, just) vlan pipes
- From an abstract perspective not much difference between provisioning compute and networking resources
  - Still not proper slicing/isolation (both data and control planes)
  - Still not proper abstractions/APIs
- Measure with multiple packet sizes (10 Gbps with small packets)
- Scenario 2 with DPDK
- Measure latency

Keynetic Technologies

<info@keynetic.es>