

Intent-based Service Function Chaining on ETSI NFV Platforms

Davide Borsatti, Gianluca Davoli,
Walter Cerroni, Franco Callegati

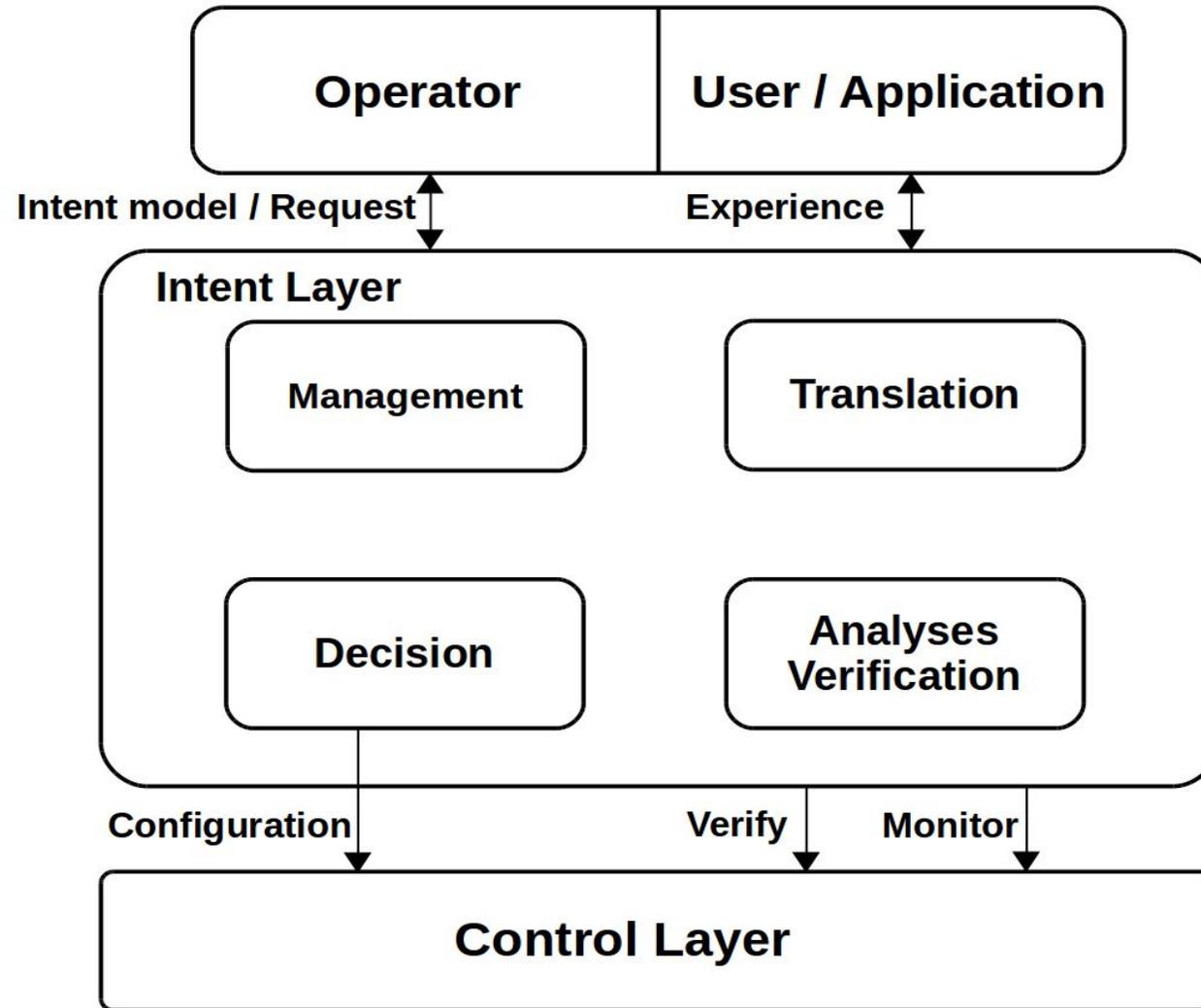
Department of Electrical, Electronic and IT Engineering
University of Bologna, Italy



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

10th International Conference on Networks of the Future (NoF)
NMRG Demo – Rome, 4th October 2019

Intent Framework



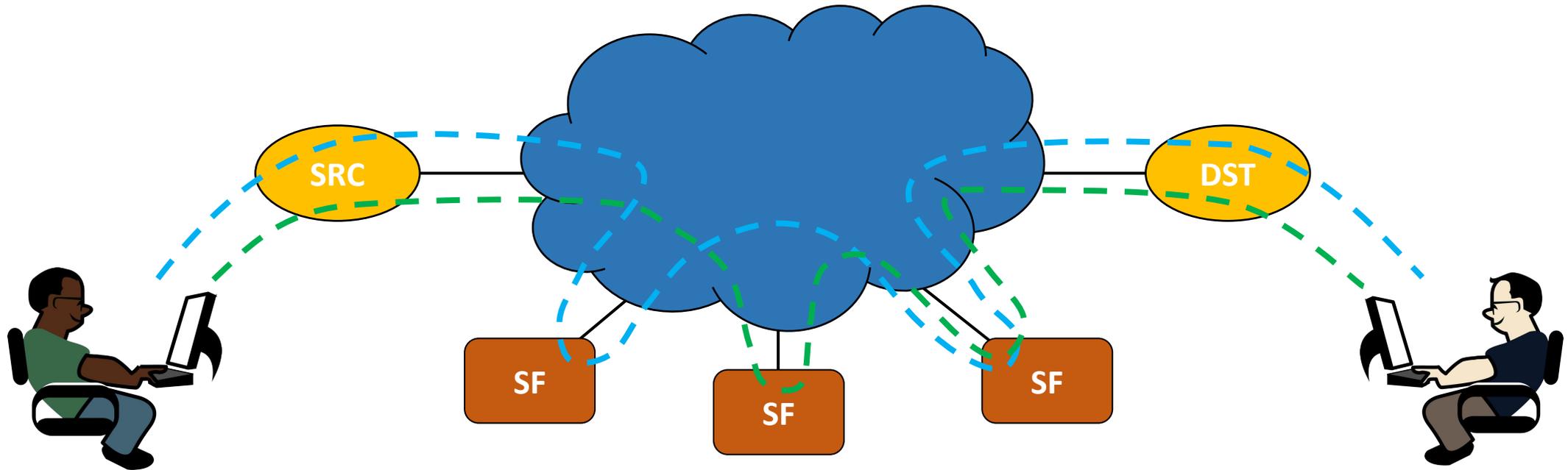
From:
draft-sun-nmrg-intent-framework-00

A couple words on Service Function Chaining (SFC)



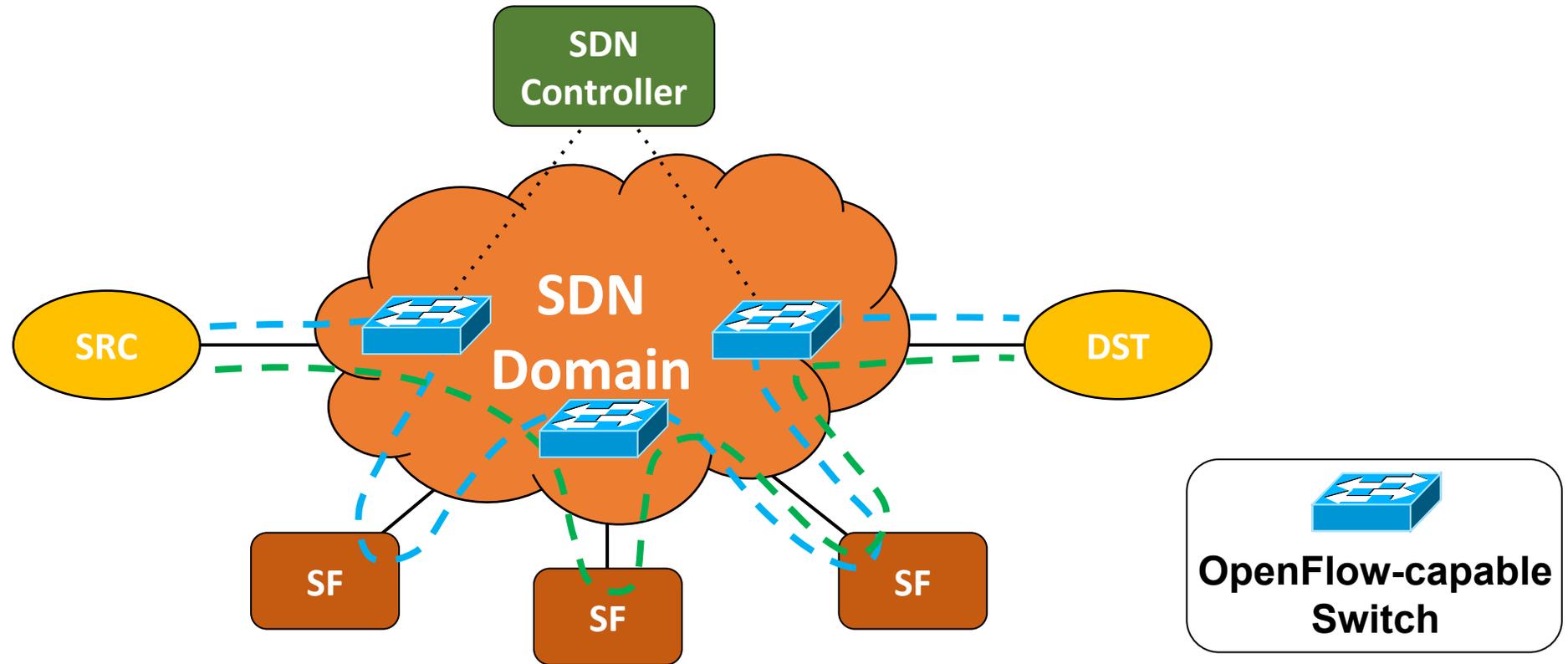
A couple words on Service Function Chaining (SFC)

Concatenation (*chaining*) of basic services or (virtualized) network/service functions...



...in order to obtain a composite service, spanning over the whole network domain

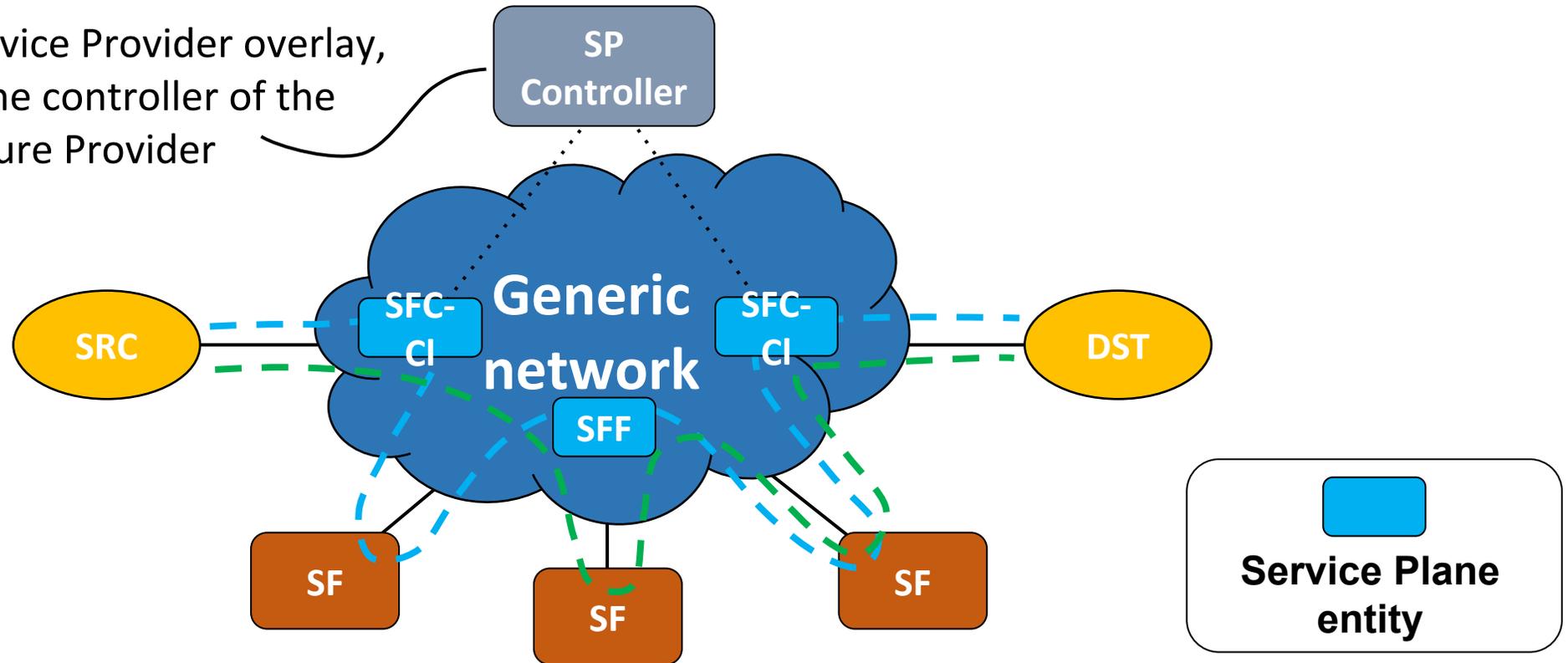
Service Function Chaining over an SDN Domain



Traffic steering is handled by OpenFlow-capable switches, controlled by a SDN Controller

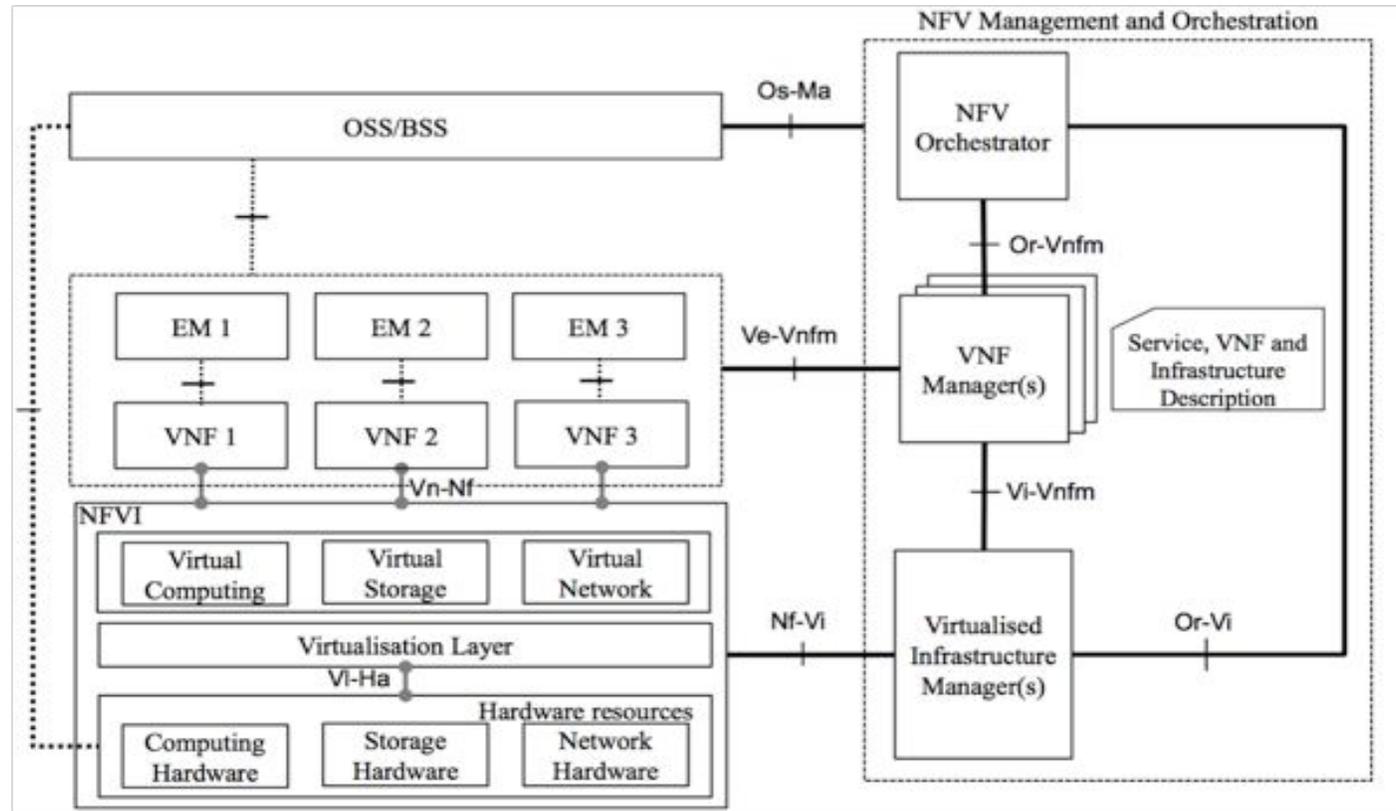
Service Function Chaining over a generic network

Controller of the Service Provider overlay,
independent of the controller of the
Infrastructure Provider

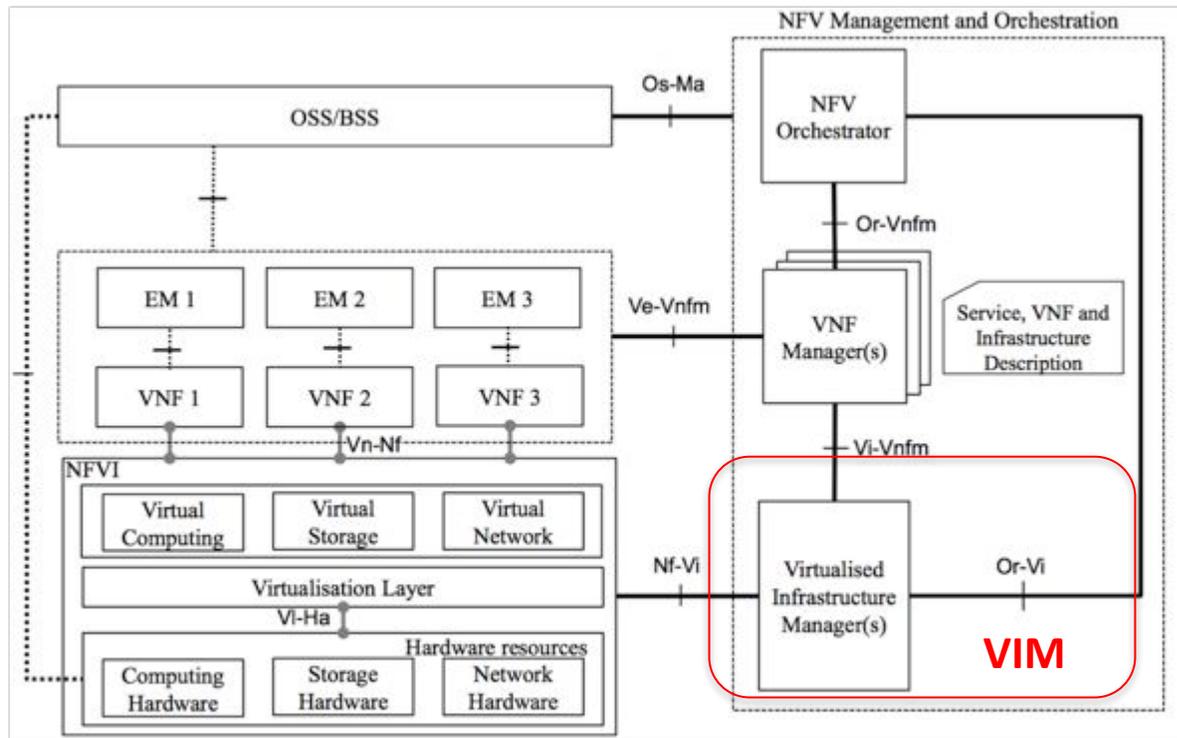


Traffic steering for SFC is handled by Service Plane entities [IETF, 2016]

ETSI NFV-MANO architecture



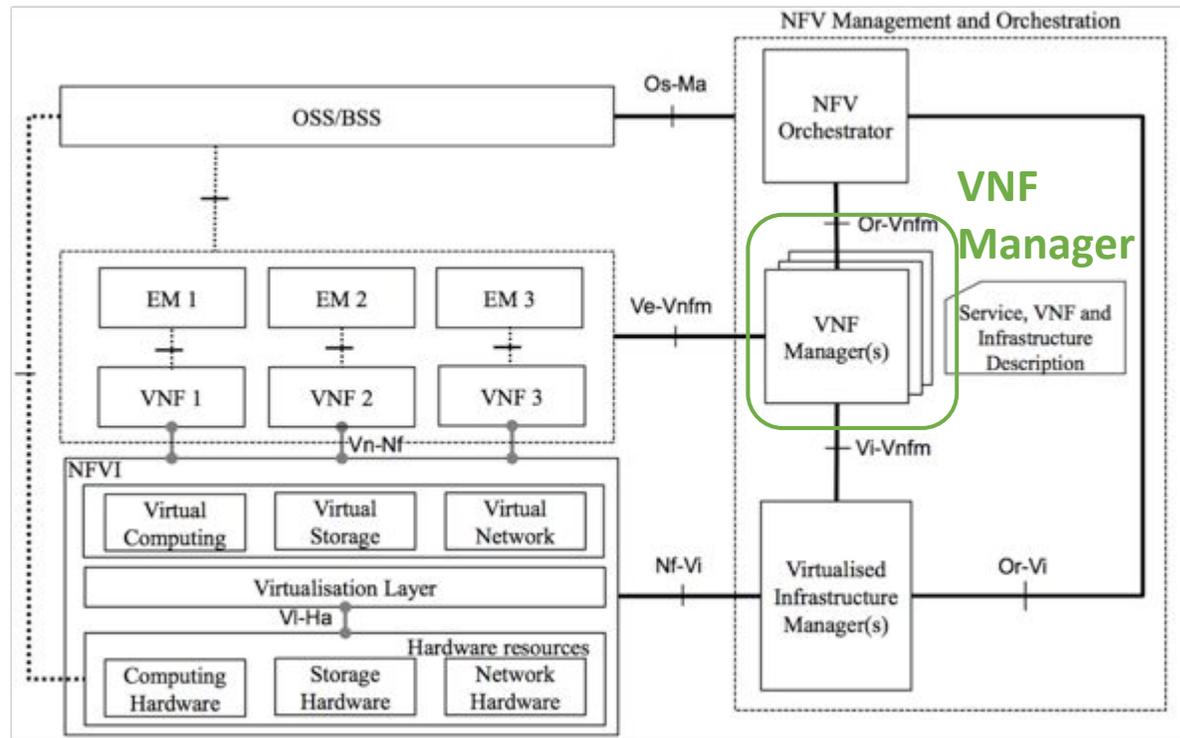
VIM: Virtualized Infrastructure Manager



Oversees the lifecycle of the virtual resources:

- VMs;
- Storage;
- Networking;
- ...

VNF Manager

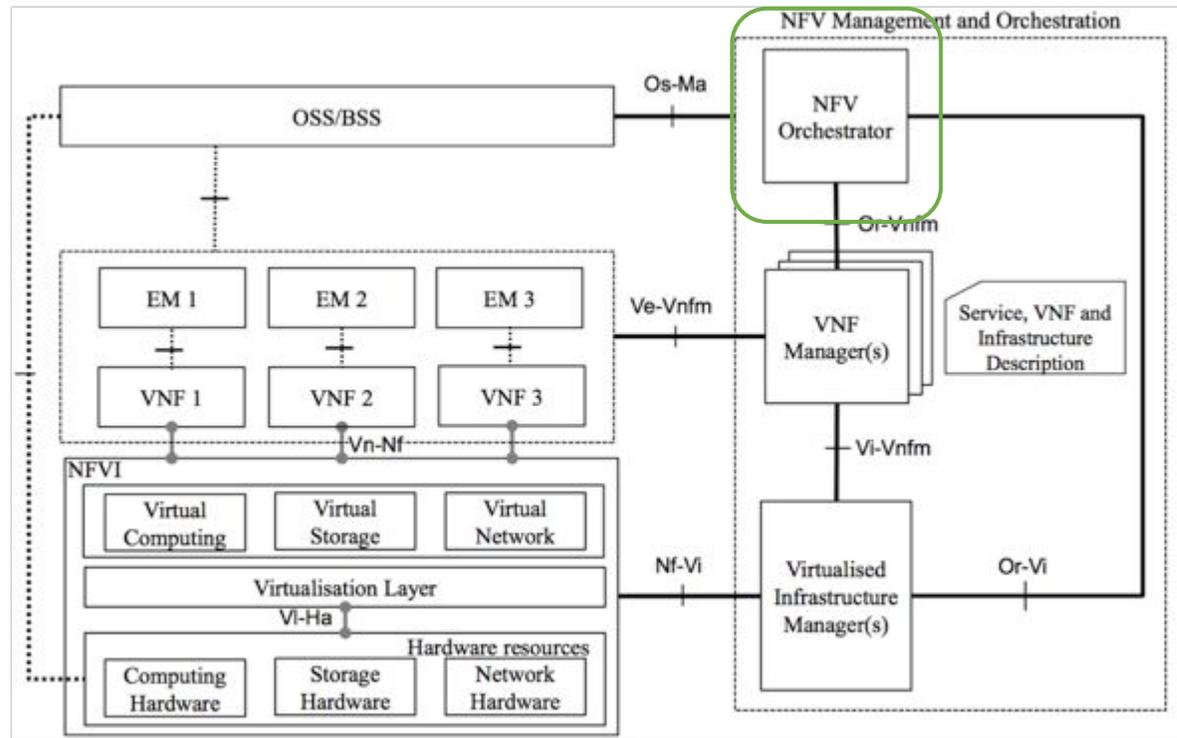


Oversees the lifecycle of the VNFs:

- Creates/Destroys VNF
- Modifies the configuration of the VNF (Day 1 and Day 0 configurations)

NFV Orchestrator

NFV Orchestrator



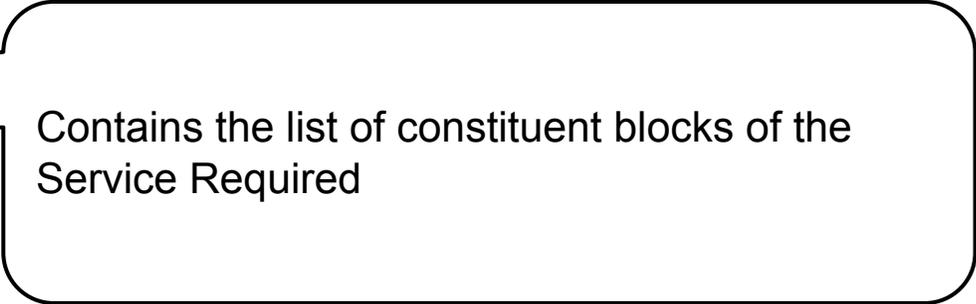
Supervises the whole system, monitoring the set of resources provided by the underlying infrastructures, and controlling the VNF lifecycle through relevant interfaces towards VIMs and VNFMs.

Intent composition

```
{
  "name": "intent_name",
  "service_blocks": [
    {
      "block": "service_block_name",
      "managed": Bool:Management_Required,
      "order": int:Order_inside_SFP ,
      "symmetric": Bool:Block_On_Reverse_Path
    },
    ...
  ],
  "service_requirements": {
    ...
  }
}
```

Intent composition

```
{
  "name": "intent_name",
  "service_blocks": [
    {
      "block": "service_block_name",
      "managed": Bool:Management_Required,
      "order": int:Order_inside_SFP ,
      "symmetric": Bool:Block_On_Reverse_Path
    },
    ...
  ],
  "service_requirements": {
    ...
  }
}
```



Contains the list of constituent blocks of the Service Required

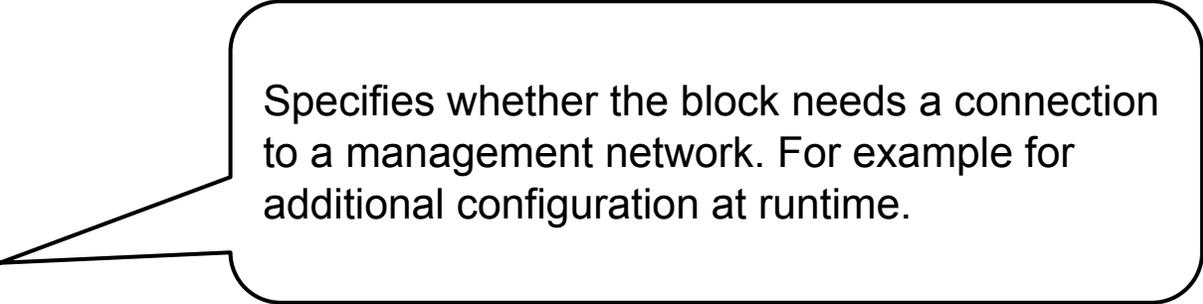
Intent composition

```
{
  "name": "intent_name",
  "service_blocks": [
    {
      "block": "service_block_name",
      "managed": Bool:Management_Required,
      "order": int:Order_inside_SFP ,
      "symmetric": Bool:Block_On_Reverse_Path
    },
    ...
  ],
  "service_requirements": {
    ...
  }
}
```

Name of the block. Used to identify the VNF (or list of VNFs) corresponding to the block

Intent composition

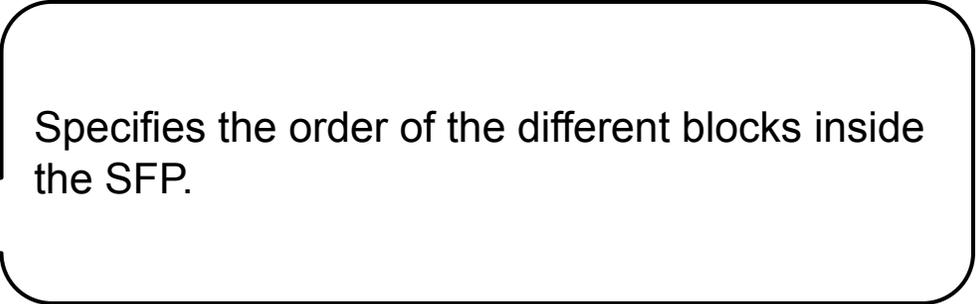
```
{
  "name": "intent_name",
  "service_blocks": [
    {
      "block": "service_block_name",
      "managed": Bool:Management_Required,
      "order": int:Order_inside_SFP ,
      "symmetric": Bool:Block_On_Reverse_Path
    },
    ...
  ],
  "service_requirements": {
    ...
  }
}
```



Specifies whether the block needs a connection to a management network. For example for additional configuration at runtime.

Intent composition

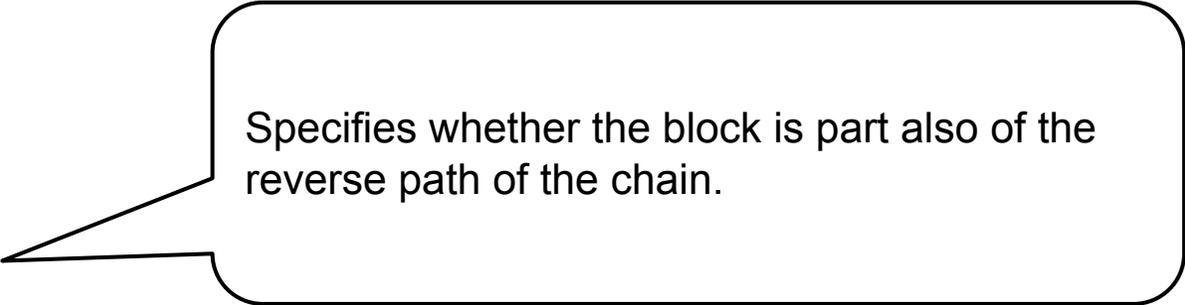
```
{
  "name": "intent_name",
  "service_blocks": [
    {
      "block": "service_block_name",
      "managed": Bool:Management_Required,
      "order": int:Order_inside_SFP ,
      "symmetric": Bool:Block_On_Reverse_Path
    },
    ...
  ],
  "service_requirements": {
    ...
  }
}
```



Specifies the order of the different blocks inside the SFP.

Intent composition

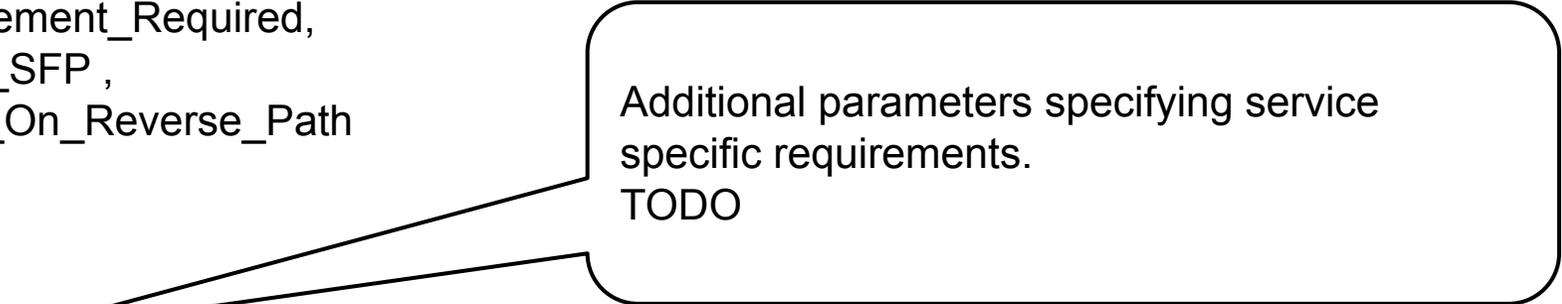
```
{
  "name": "intent_name",
  "service_blocks": [
    {
      "block": "service_block_name",
      "managed": Bool:Management_Required,
      "order": int:Order_inside_SFP ,
      "symmetric": Bool:Block_On_Reverse_Path
    },
    ...
  ],
  "service_requirements": {
    ...
  }
}
```



Specifies whether the block is part also of the reverse path of the chain.

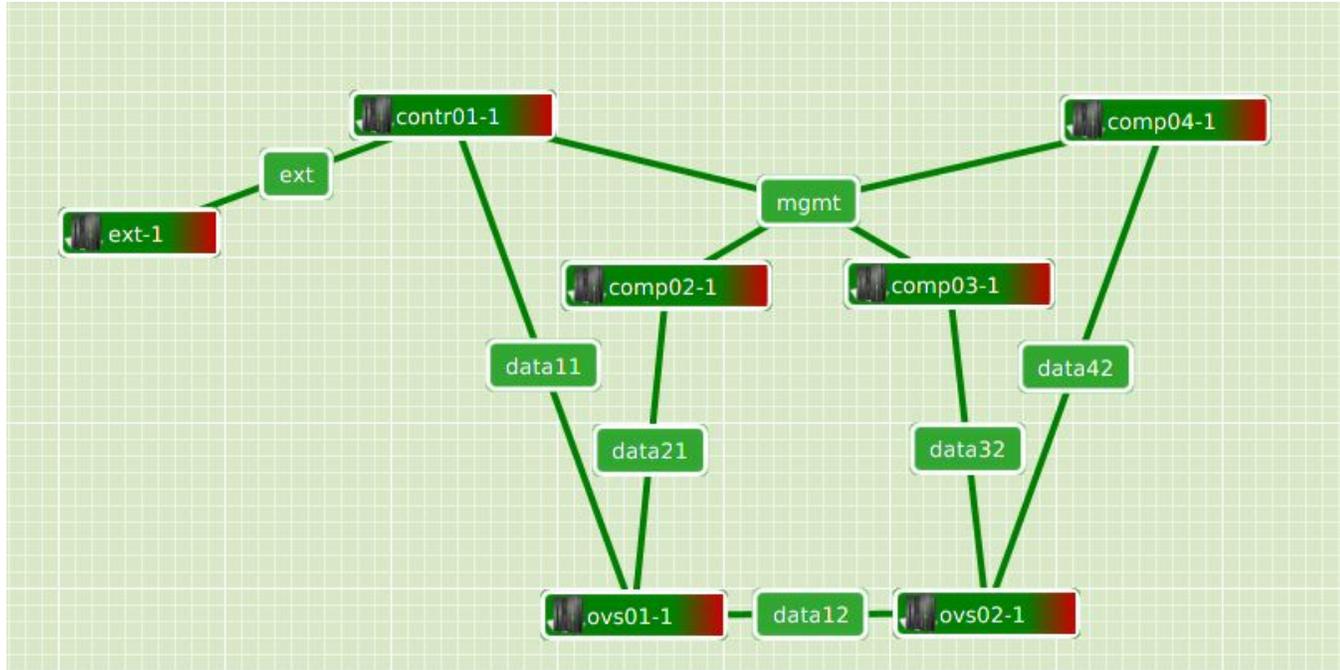
Intent composition

```
{  
  "name": "intent_name",  
  "service_blocks": [  
    {  
      "block": "service_block_name",  
      "managed": Bool:Management_Required,  
      "order": int:Order_inside_SFP ,  
      "symmetric": Bool:Block_On_Reverse_Path  
    },  
    ...  
  ],  
  "service_requirements": {  
    ...  
  }  
}
```

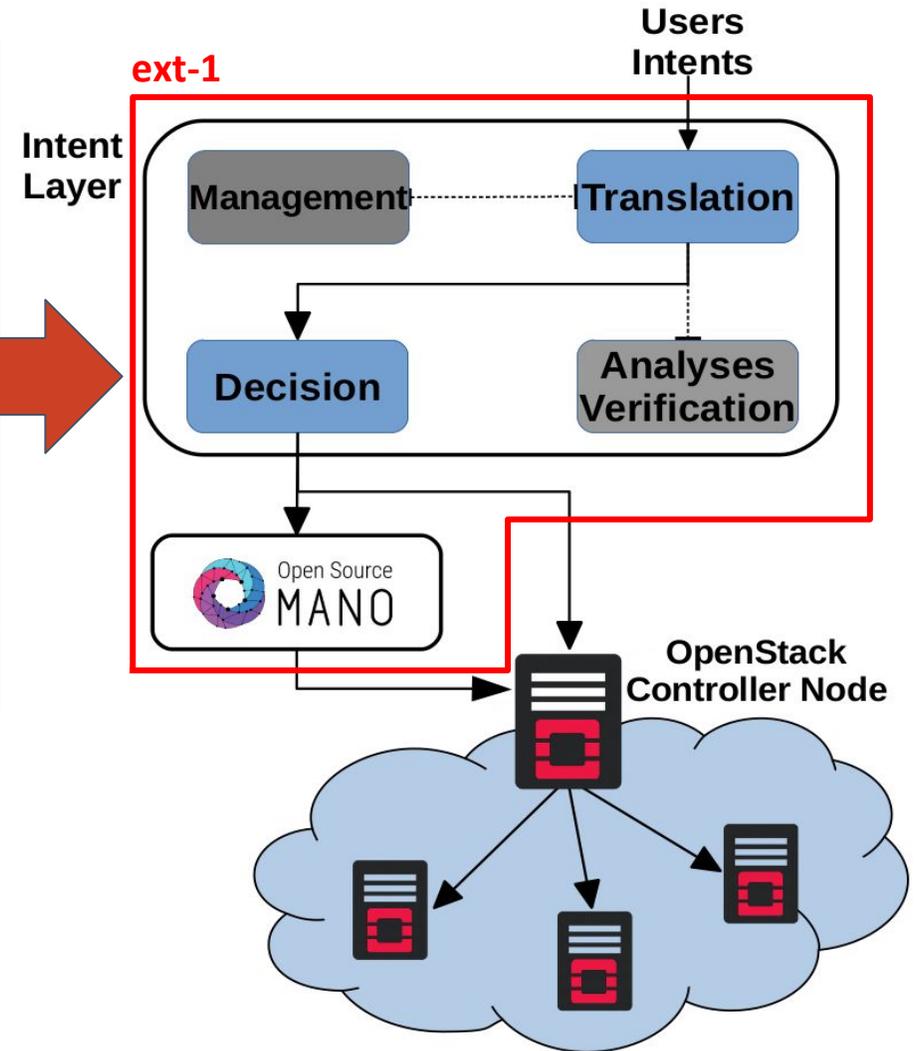
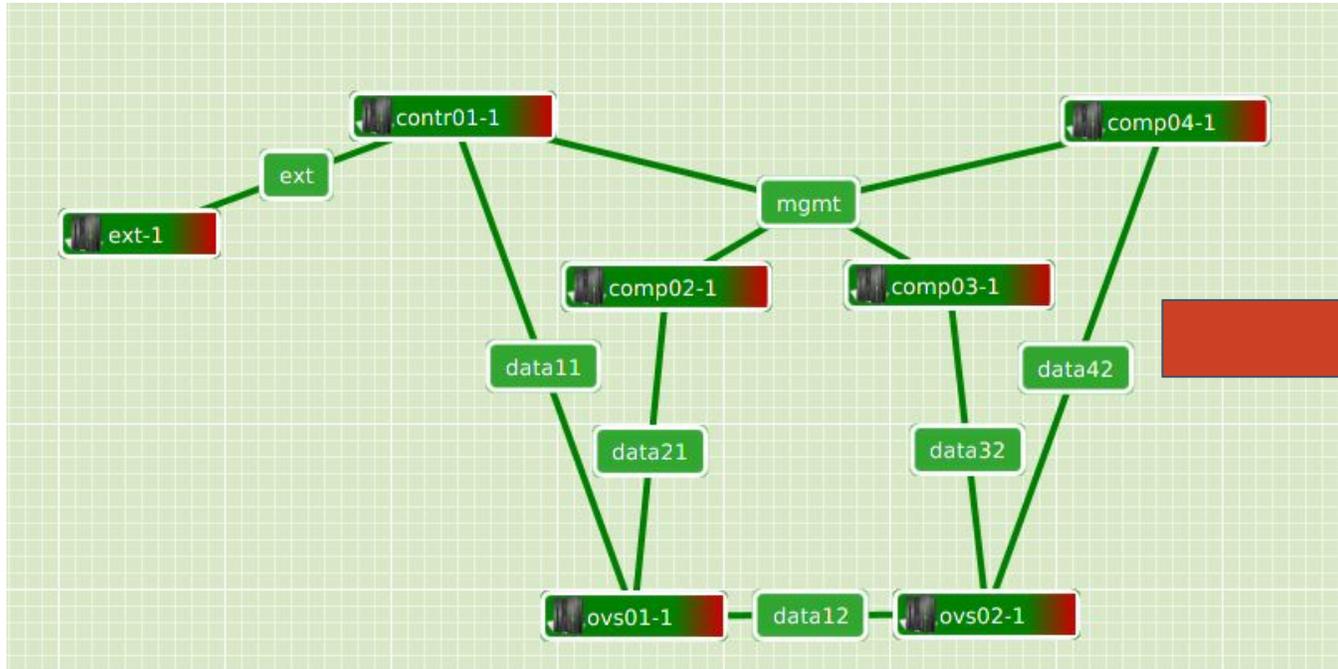


Additional parameters specifying service specific requirements.
TODO

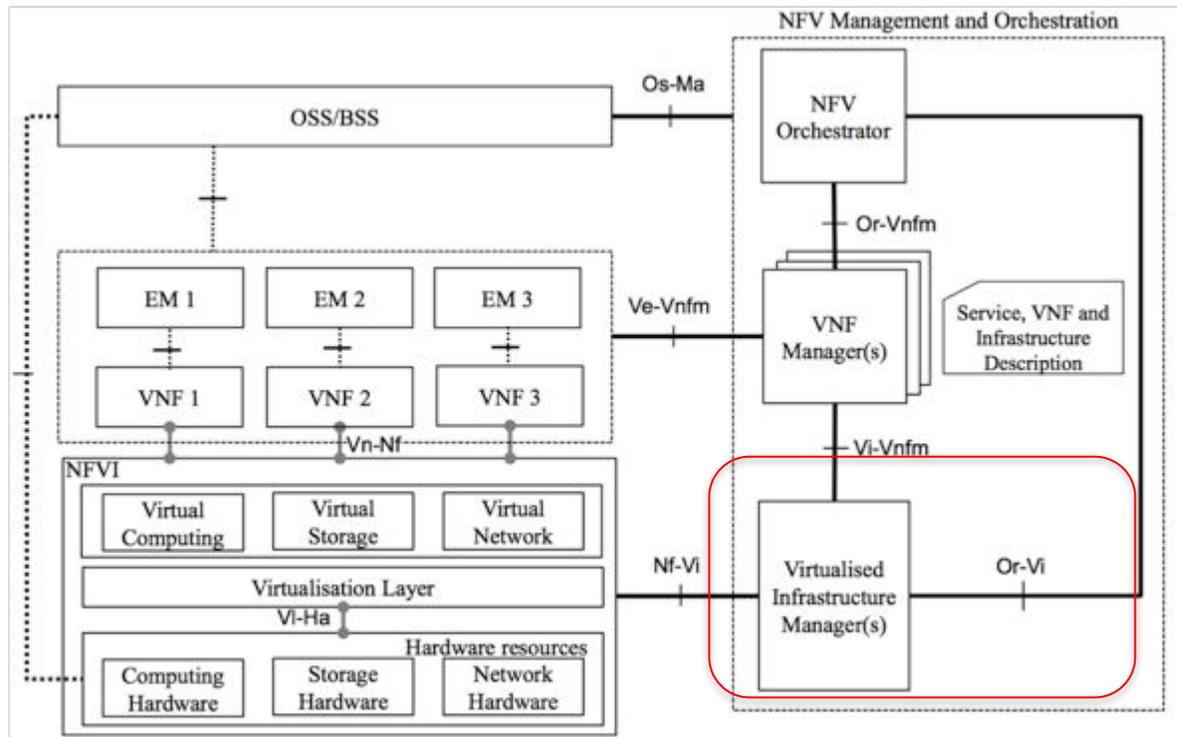
Test-Bed Image



Test-Bed Image



Test-Bed - Blocks Component



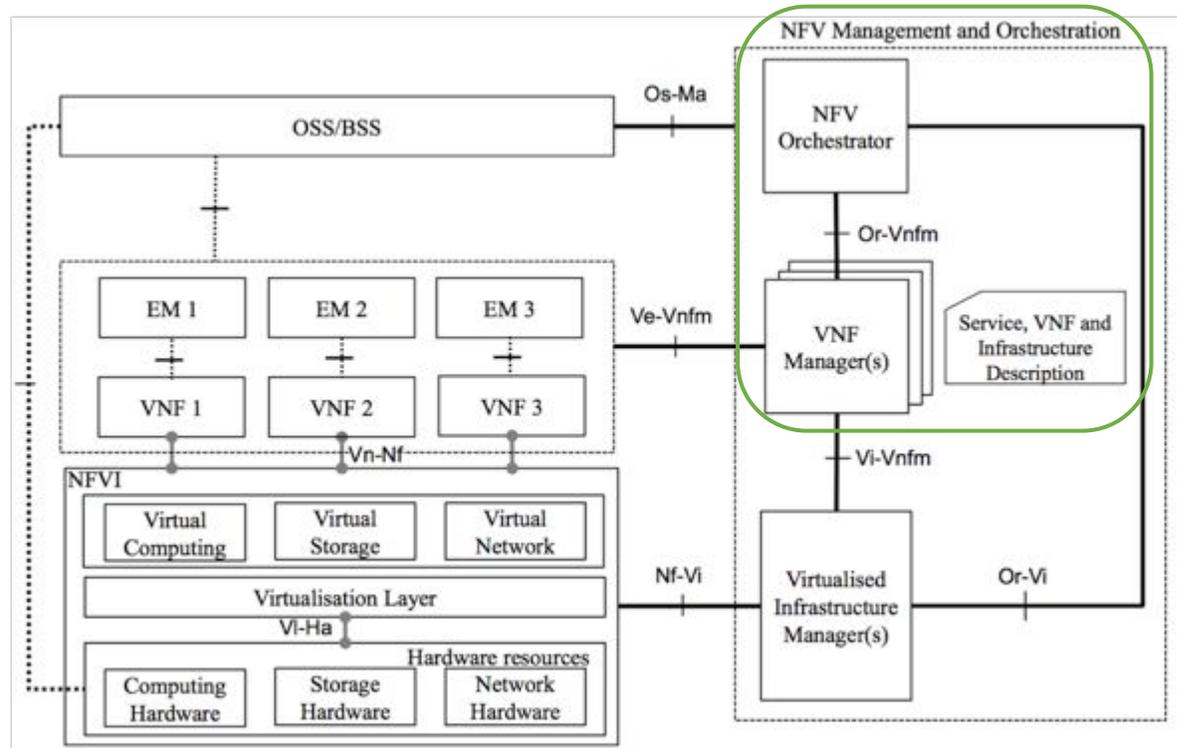
One of the most famous IaaS (Infrastructure as a Service) cloud platform. Used in different scenarios, from research to commercial.

It provides different types of services, such as:

- Computing capabilities;
- Virtualized Networking;
- Storage;



Test-Bed - Blocks Component



Open source solution compliant with the ETSI MANO framework.

OSM Terminology

- **NSD:** Describes in a YAML format the composition of the Network Service in terms of interconnection of VNFD. It is composed by:
 - **Constituent VNFD:** A list containing the VNFs that compose the network service;
 - **VLD:** A Virtual Link Descriptor identifies a network of the Network Service. Each VNF can be connected to it thanks to a **Connection Point (CP)**, a logical entity defined in the VNFD that links the VLD to one of its interfaces.
-

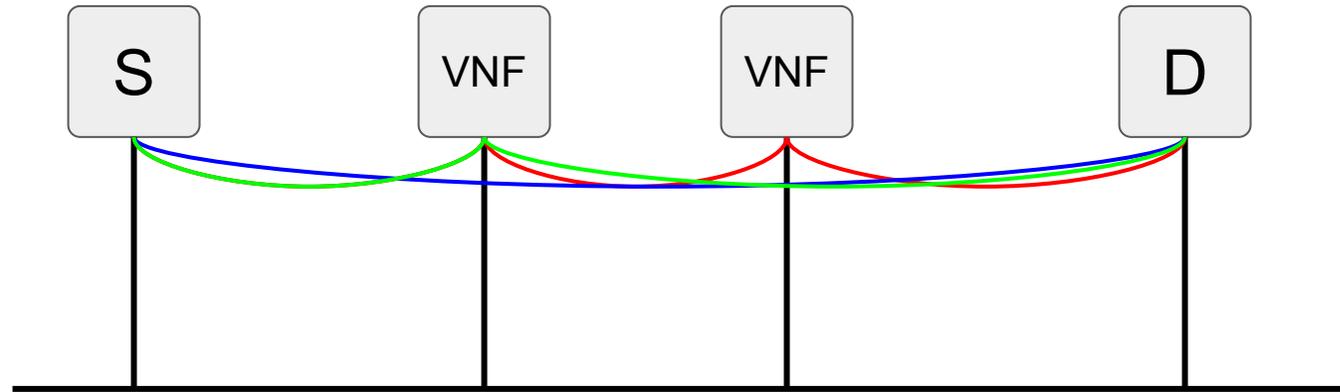
SFC-extension for OpenStack

The extension allows for the creation of SFPs, it natively supports interaction with Open vSwitch (OvS) and it implements a flow classification mechanism.

It is composed by four main entities:

1. Flow Classifier
 2. Port Pair
 3. Port Pair Group
 4. Port Chain
-

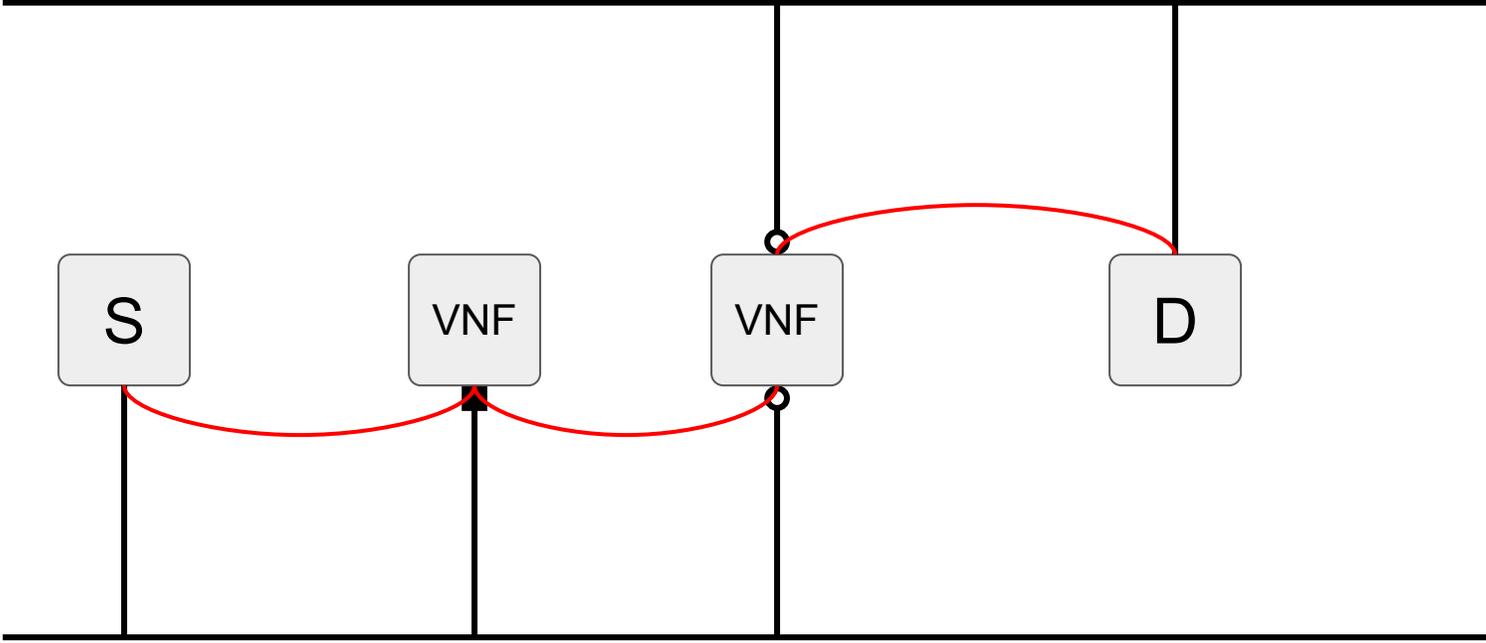
SFC-extension – Flow Classifier



Flow Classifier:

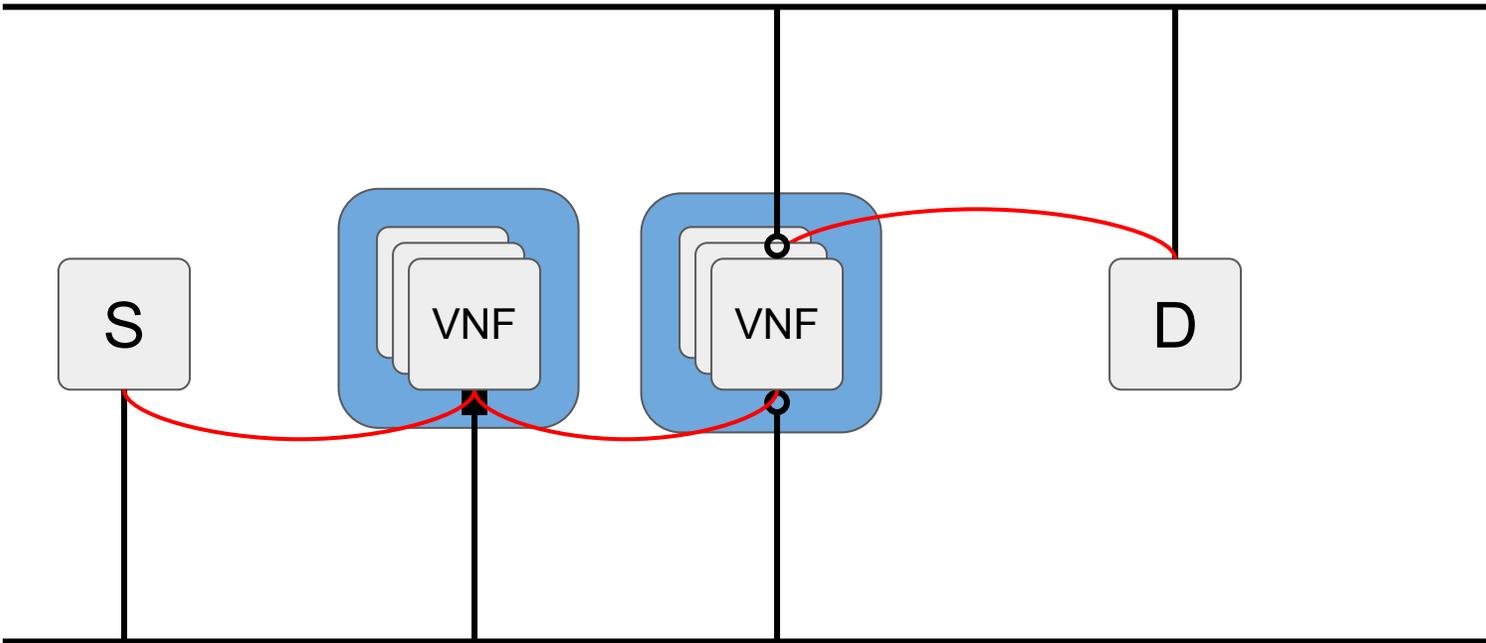
	UDP traffic
	HTTP traffic (TCP port 80)
	Other packets

SFC-extension – Port Pair



Port Pair: —■ Same ingress and egress port
 —○ Different ingress and egress port

SFC-extension – Port Pair Group

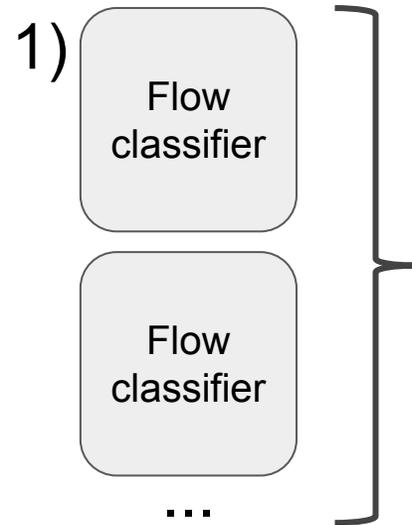


Port Pair Group :

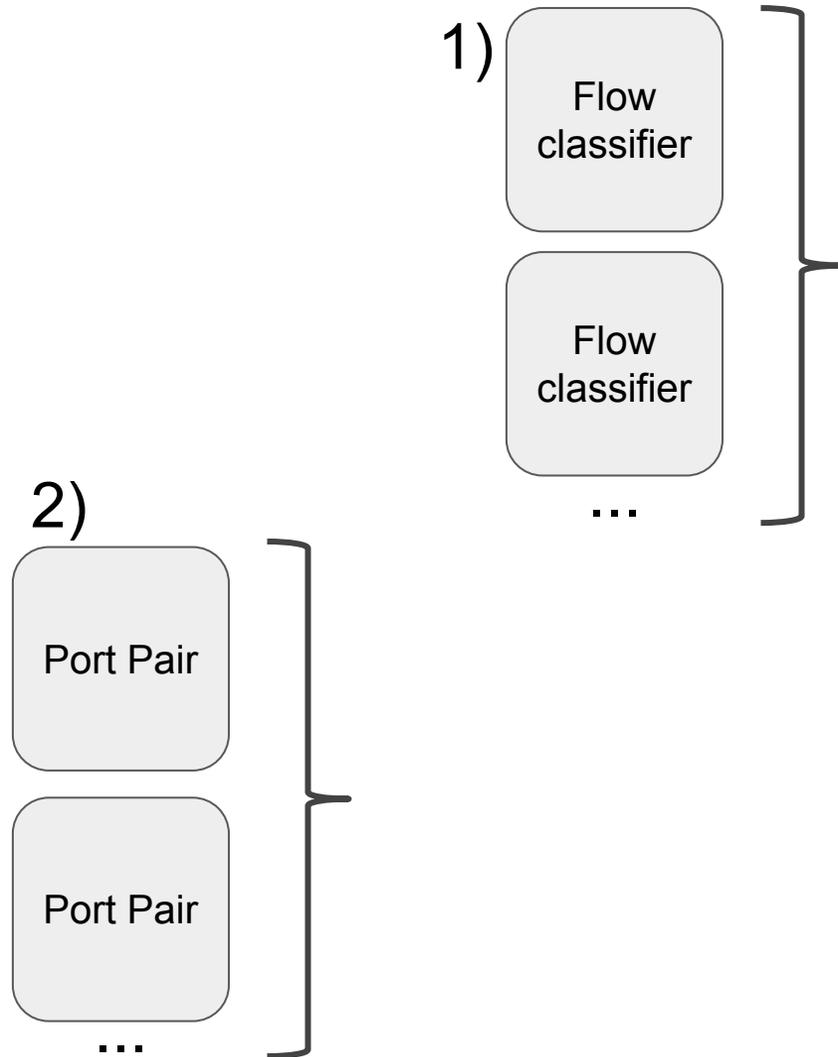


Collection of Port Pairs with same functionalities.

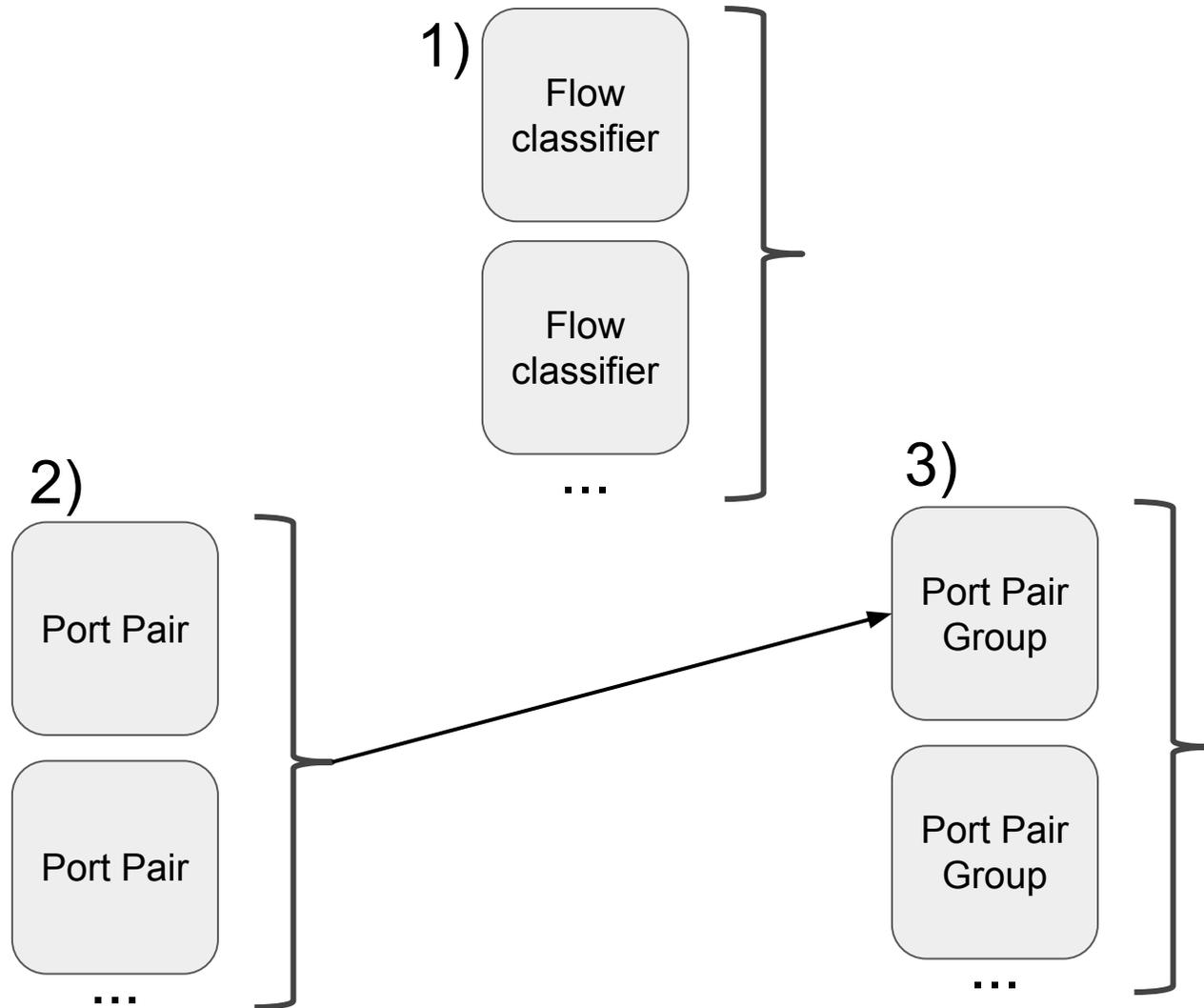
Port Chain creation with SFC-extension



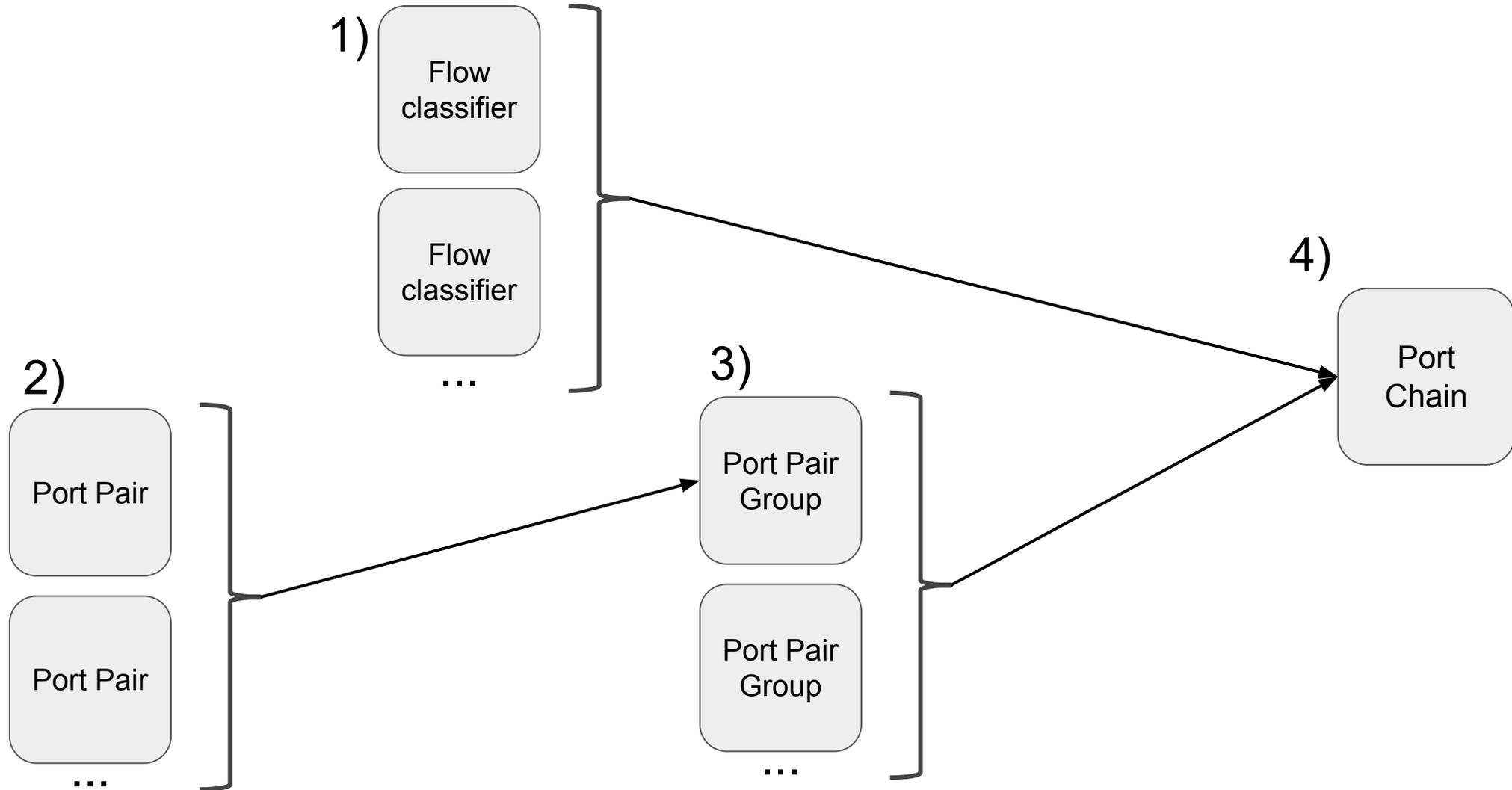
Port Chain creation with SFC-extension



Port Chain creation with SFC-extension

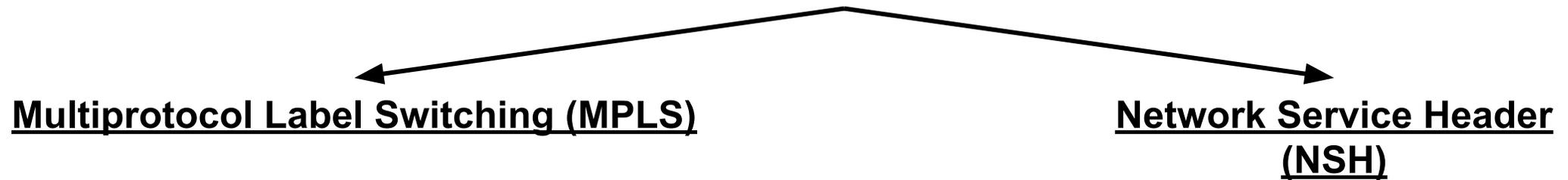


Port Chain creation with SFC-extension

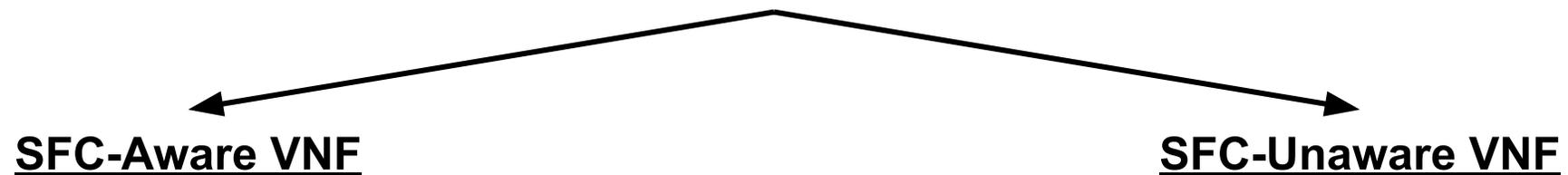


SFC-extension

The SFC-extension support two types of encapsulation mechanism to separate traffic belonging to different Port Chain:

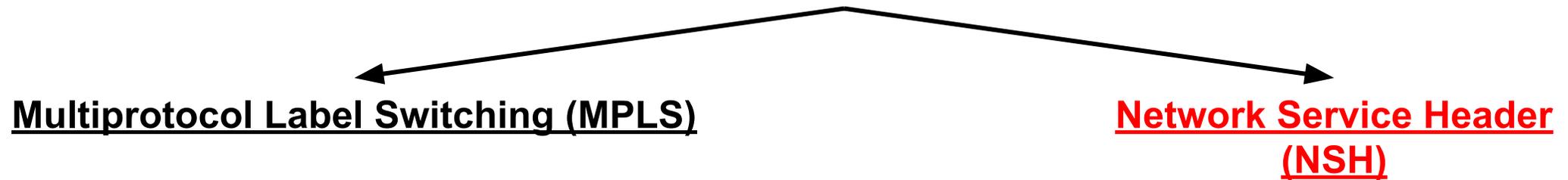


The SFC-extension support two types VNF:

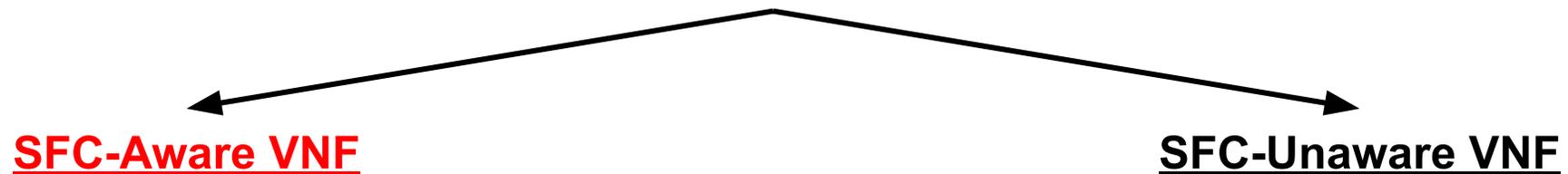


SFC-extension - OSM limitations

The SFC-extension support two types of encapsulation mechanism to separate traffic belonging to different Port Chain:



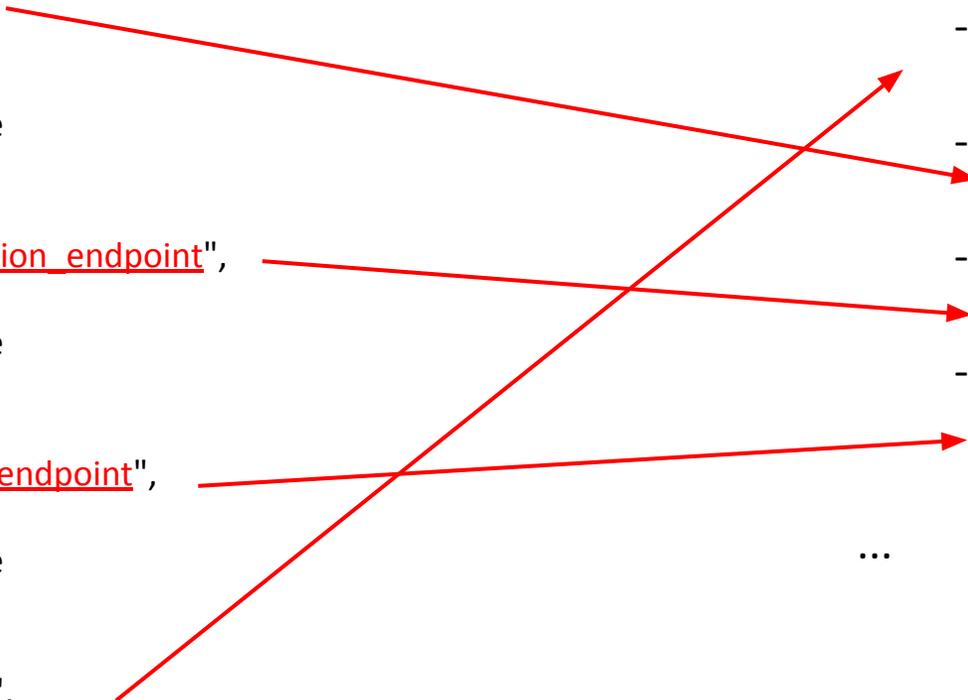
The SFC-extension support two types VNF:



Intent Translation - Example

```
{ "name": "serviceA",  
  "service_blocks": [  
    {  
      "block": "dpi",  
      "managed": true,  
      "order": 1,  
      "symmetric": true  
    },  
    {  
      "block": "destination_endpoint",  
      "managed": true,  
      "symmetric": true  
    },  
    {  
      "block": "source_endpoint",  
      "managed": true,  
      "symmetric": true  
    },  
    {  
      "block": "firewall",  
      "managed": false,  
      "order": 0,  
      "symmetric": true } ] }
```

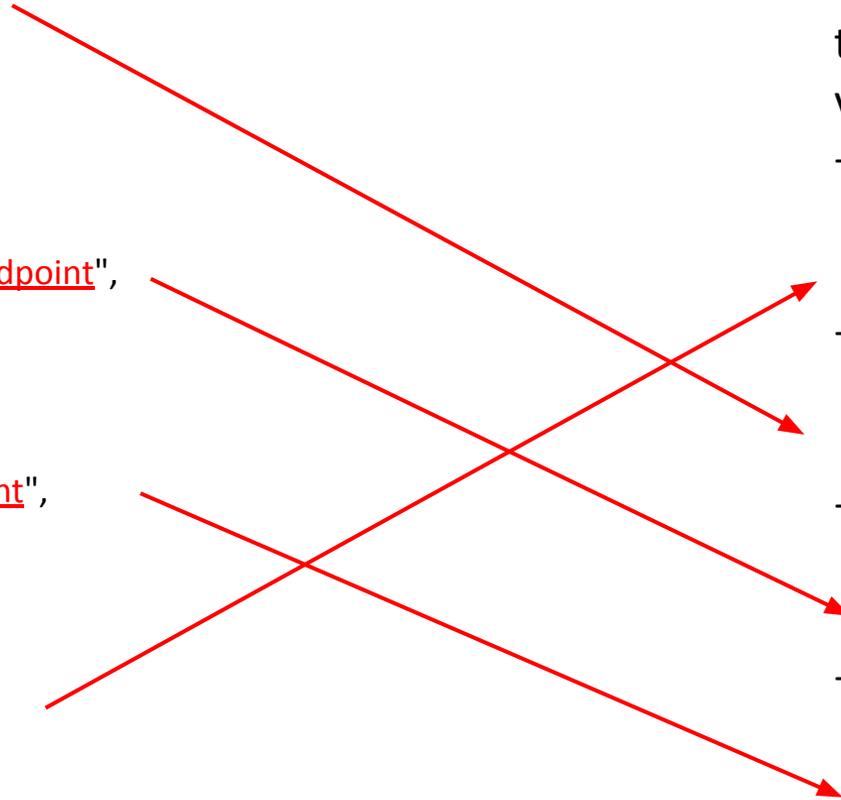
```
nsd:nsd-catalog:  
  nsd:  
    - constituent-vnfd:  
      - member-vnf-index: 0  
        vnfd-id-ref: firewall_vnfd  
      - member-vnf-index: 1  
        vnfd-id-ref: dpi_vnfd  
      - member-vnf-index: 11  
        vnfd-id-ref: destination_endpoint_vnfd  
      - member-vnf-index: 10  
        vnfd-id-ref: source_endpoint_vnfd  
      ...
```



Intent Translation - Example

```
{ "name": "serviceA",  
  "service_blocks": [  
    {  
      "block": "dpi",  
      "managed": true,  
      "order": 1,  
      "symmetric": true  
    },  
    {  
      "block": "destination_endpoint",  
      "managed": true,  
      "symmetric": true  
    },  
    {  
      "block": "source_endpoint",  
      "managed": true,  
      "symmetric": true  
    },  
    {  
      "block": "firewall",  
      "managed": false,  
      "order": 0,  
      "symmetric": true } ] }
```

```
...  
vld:  
- id: dataNet  
  type: ELAN  
  vnf-d-connection-point-ref:  
  - member-vnf-index-ref: 0  
    vnf-d-connection-point-ref: vnf-cp-data  
    vnf-d-id-ref: firewall_vnfd  
  - member-vnf-index-ref: 1  
    vnf-d-connection-point-ref: vnf-cp-data  
    vnf-d-id-ref: dpi_vnfd  
  - member-vnf-index-ref: 11  
    vnf-d-connection-point-ref: vnf-cp-data  
    vnf-d-id-ref: destination_endpoint_vnfd  
  - member-vnf-index-ref: 10  
    vnf-d-connection-point-ref: vnf-cp-data  
    vnf-d-id-ref: source_endpoint_vnfd  
...
```



Intent Translation - Example

```
{ "name": "serviceA",  
  "service_blocks": [  
    {  
      "block": "dpi",  
      "managed": true,  
      "order": 1,  
      "symmetric": true  
    },  
    {  
      "block": "destination_endpoint",  
      "managed": true,  
      "symmetric": true  
    },  
    {  
      "block": "source_endpoint",  
      "managed": true,  
      "symmetric": true  
    },  
    {  
      "block": "firewall",  
      "managed": false,  
      "order": 0,  
      "symmetric": true } ] }
```

```
...  
vld:  
...  
- id: mgmtNet  
  type: ELAN  
  vnfd-connection-point-ref:  
  - member-vnf-index-ref: 1  
    vnfd-connection-point-ref: vnf-cp-mgmt  
    vnfd-id-ref: dpi_vnfd  
  - member-vnf-index-ref: 11  
    vnfd-connection-point-ref: vnf-cp-mgmt  
    vnfd-id-ref: destination_endpoint_vnfd  
  - member-vnf-index-ref: 10  
    vnfd-connection-point-ref: vnf-cp-mgmt  
    vnfd-id-ref: source_endpoint_vnfd  
...
```