

# ONAP Beijing: Understanding the vFWCL use-case mechanism

Author: Badr Eddine Aouled Yahia - Orange

Reviewed by: Jose Manuel Sanchez Vilchez, Christian Destré - Orange

This page provides a deep investigation on the mechanisms of deployment and closed loop for the vFW use-case on ONAP Beijing release.



ONAP Beijing vFW...CL Analysis.pptx

## Sections

1. General overview
  - a. vFWCL service overview
  - b. Service deployment in ONAP
  - c. SO overview
  - d. Closed loop
2. vFW deployment mechanism
  - a. Service deployment overview
  - b. Service instance creation
  - c. VNF creation
  - d. VF-module creation
  - e. Analysis
3. vFW closed loop mechanism
  - a. vFW VES measurements reporting
  - b. Closed loop execution mechanism
  - c. Analysis

## General overview

### vFWCL service overview

The vFW service is composed of the following components:

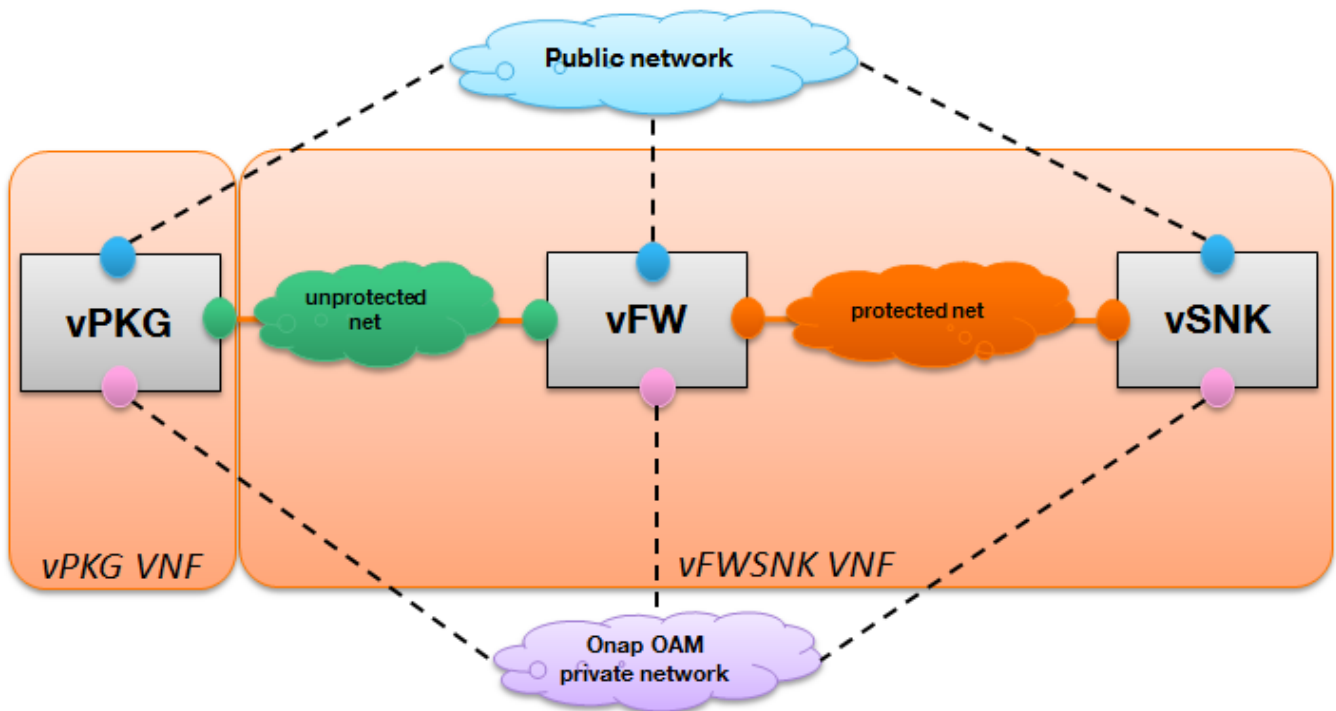
**vPKG VNF:**

–Packet generator: sends periodically different volumes of traffic to the sink through the firewall (vFw)

**vFWSNK VNF:**

–Firewall: reports the volume of traffic passing through to the DCAE collector (VES collector). The vFW has no real firewalling functionality.

–Traffic sink: provides a graphical representation (bar charts) showing the volume of incoming traffic.



The **vFW close loop scenario** consists in applying policy rules that aim to re-adjust the traffic volume when high threshold (700 packets/10s) or low threshold (300 packets/10s) are crossed. In fact, the DCAE collects events from the vFW, applies analytics (Threshold Crossing Analytics: TCA microservice) and publishes events to DMAap. When detecting the triggering event, the policy engine executes the operational policy via the APP-C that modifies vPKG application configurations in order to adjust the traffic volume to 500 packet per 10 seconds.

## Service deployment in ONAP

After onboarding and distributing a service, ONAP execution-time provides tools and processes to deploy a service.

From an ONAP user perspective, the operation of service deployment in ONAP consists in an ordered set of actions performed through scripts or using the VID portal to trigger instantiation in SO.

In general, a service in ONAP is the composition of several elements that consists in VNFs, Networks, VF-modules and volume groups.

In order to fully deploy a service via VID portal, three levels of instantiation are identified:

- Service instantiation
- VNF and/or network instantiation
- VF-module and/or volume group instantiation

Each step of instantiation corresponds to a set of orchestration tasks performed by the SO component. In fact, when receiving a request to create a service instance, the SO selects a process that defines a workflow of actions in order to accomplish the service instantiation. The process includes series of tasks to be executed by ONAP execution-time components.

## SO overview

[blocked URL](#)

Service deployment involves the following main SO components:

- API Handler
  - The SO exposes a restful interface to northbound clients (VID in this case) that handles service-level and infrastructure requests (APIs for service, vnf, network, vf-module ... creation/deletion ...).
  - When receiving a request, the API Handler stores a record in the request DB and maps the requests with the appropriate orchestration recipe (a BPMN flow) based on request attributes (service-type, action ...) or based on service model and action. And then, it forwards data to BPMN flows.
- BPMN Execution Engine
  - This component is responsible for execution BPMN flows. It sequences orchestration tasks and allows interaction with other ONAP components in order to update the inventory, manages cloud resources via openstack adaptors and configures network resources via SDN-C adapter.

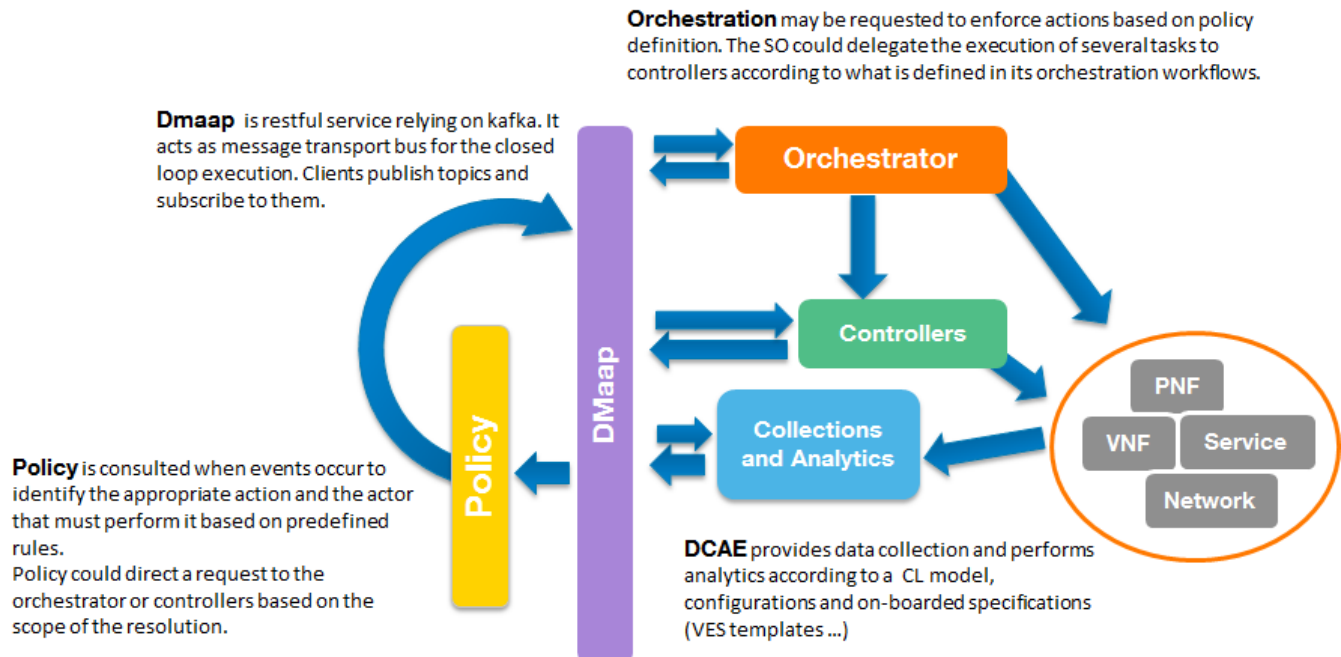
- Resource/Controller adapters  
These interfaces allow the SO to interact with infrastructure and ONAP controllers. This layer hides infrastructure complex details and supports the execution of BPMN tasks by providing SOAP and REST APIs.

## Closed loop

The closed loop concept in ONAP allows an automatic recovery of faults reported by traps or alarms and provides automated capacity management when performance thresholds are crossed. It uses feedbacks to control and optimize the behavior of monitored elements in order to automatically respond to service conditions without human intervention.

The execution of a closed loop involves several components. In fact, the DCAE is responsible for collecting and analysing collected data in order to publish events to DMAap. The policy engine allows to trigger a response according to configuration policies when detecting critical events. Finally, an operational policy is executed by ONAP orchestrator or controllers.

The closed loop design is provided by CLAMP that allows the creation and distribution of a CL workflow.

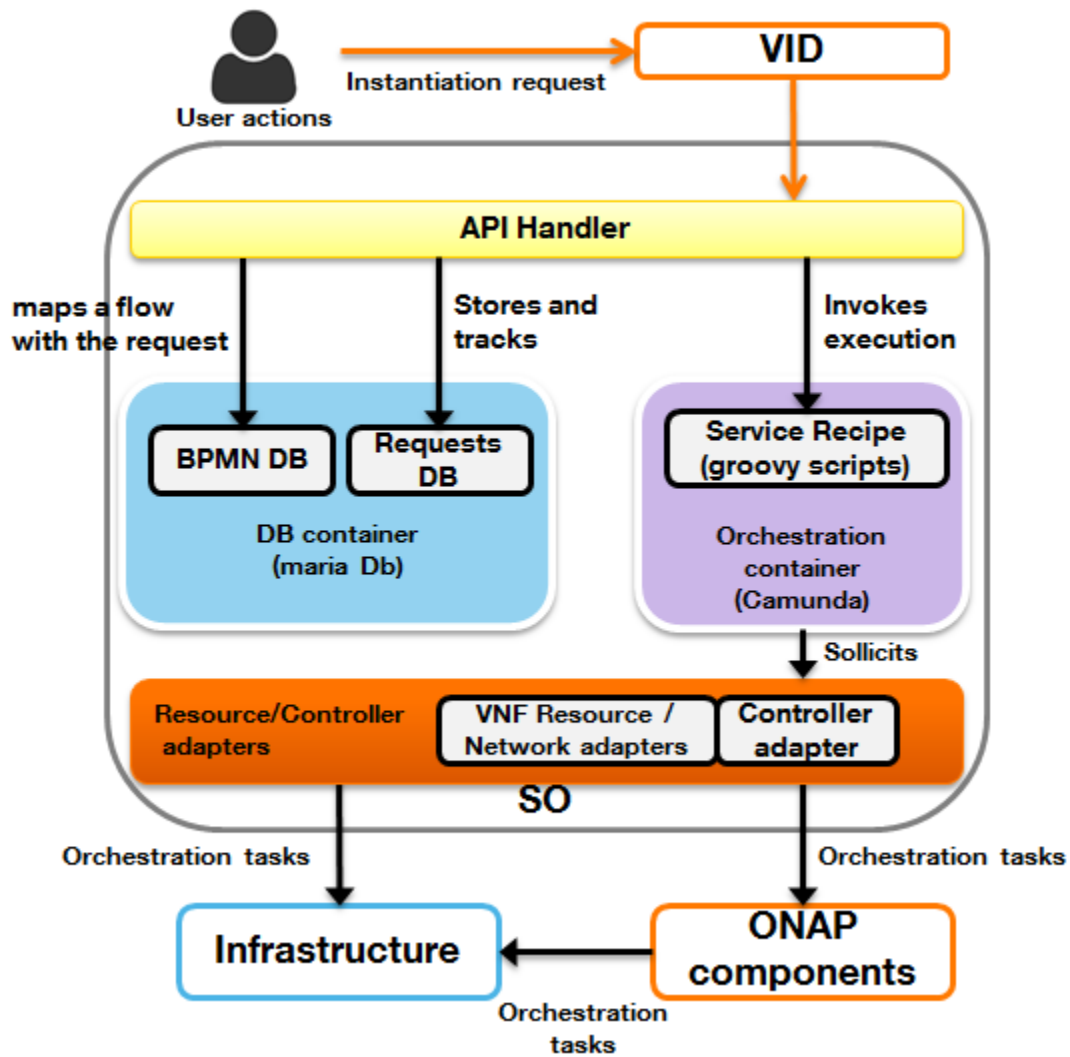


## vFW deployment mechanism

### vFW service deployment overview

After design and distribution stages, the vFW service is ready to be deployed. Three main user actions are performed in VID portal in order to fully deploy the service. These actions correspond to three HTTP requests sent to SO.

1. Instantiate service Service instance creation request
2. Add node VNF instance creation request
3. Add VF-module VF-module creation request



From a high level perspective, each VID request towards SO triggers the execution of an orchestration workflow. In fact, When receiving a request, the API Handler stores an orchestration request in request\_db and selects a BPMN workflow from Camunda\_db to be executed by the orchestration engine (Camunda). The workflow specifies a sequence of orchestration tasks to be performed by ONAP execution-time components. The process includes recursive calls to other orchestration sub-processes.

To each BPMN corresponds a groovy script executed by the camunda engine that allows to perform actions or invoke other ONAP components to execute an action via Resource and Controller adapters.

## Service instance creation

In the VID portal, the demo user deploys the service:

Deploy	84909c11-1a62-47b3-9aed-59c2ec5c891d	46c0d35f-4a34-4f5b-9ee3-fd65f6f74ac2	vfw_svc	2.0	service	DISTRIBUTION_COMPLETE_OK
--------	--------------------------------------	--------------------------------------	---------	-----	---------	--------------------------

The user specifies the input data and confirms:

User action in VID portal:

# Create Service Instance -- a la carte

Service Name: vfw\_svc  
Service Invariant UUID: 46c0d35f-4a34-4f5b-9ee3-fd65f6f74ac2  
Service Version: 2.0  
Service UUID: 84909c11-1a62-47b3-9aed-59c2ec5c891d  
Service Description: vfw svc with 2 vnfs  
Service Category: Network L4+  
Service Type:  
Service Role:

User Provided Data ( \* indicates required field)

Instance Name: *	<input type="text" value="vfwcl_svc_1607"/>
Subscriber Name: *	<input type="text" value="Demonstration"/>
Service Type: *	<input type="text" value="vFWCL"/>
Project:	<input type="text" value="Project-Demonstration"/>
Owning Entity: *	<input type="text" value="OE-Demonstration"/>
Suppress Rollback on Failure:	<input type="text" value="false"/>

The user action corresponds to an HTTP POST request from VID towards SO to trigger service instance creation: [POST /ecomp/mso/infra/serviceInstances/v6](#)

Enter Data and **Confirm** to  
Create **Service Instance**  
**Cancel** to Return to Previous Page.  
Data entered will be lost

#### Http POST request: Json payload for service instance creation

```
POST /ecompmso/infra/serviceInstances/v6 HTTP/1.1
Authorization: Basic SW5mcmFQb3J0YWxDbGllbnQ6cGFzc3dvcmQxJA==
X-FromAppId: VID
X-ECOMP-RequestID: dffdfdbb-b42e-43b6-b2e0-199181141abd
Content-Type: application/json
User-Agent: Jersey/2.23.1 (HttpURLConnection 1.8.0_171)
Cache-Control: no-cache
Pragma: no-cache
Host: vml.mso.simplesdemo.onap.org:8080
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```

```
{
  "requestDetails":{
    "requestInfo":{
      "instanceName":"vfw_svc_1607",
      "source":"VID",
      "suppressRollback":false,
      "requestorId":"demo"
    },
    "modelInfo":{
      "modelType":"service",
      "modelInvariantId":"46c0d35f-4a34-4f5b-9ee3-fd65f6f74ac2",
      "modelVersionId":"84909c11-1a62-47b3-9aed-59c2ec5c891d",
      "modelName":"vfw_svc",
      "modelVersion":"2.0"
    },
    "requestParameters":{
      "userParams":[

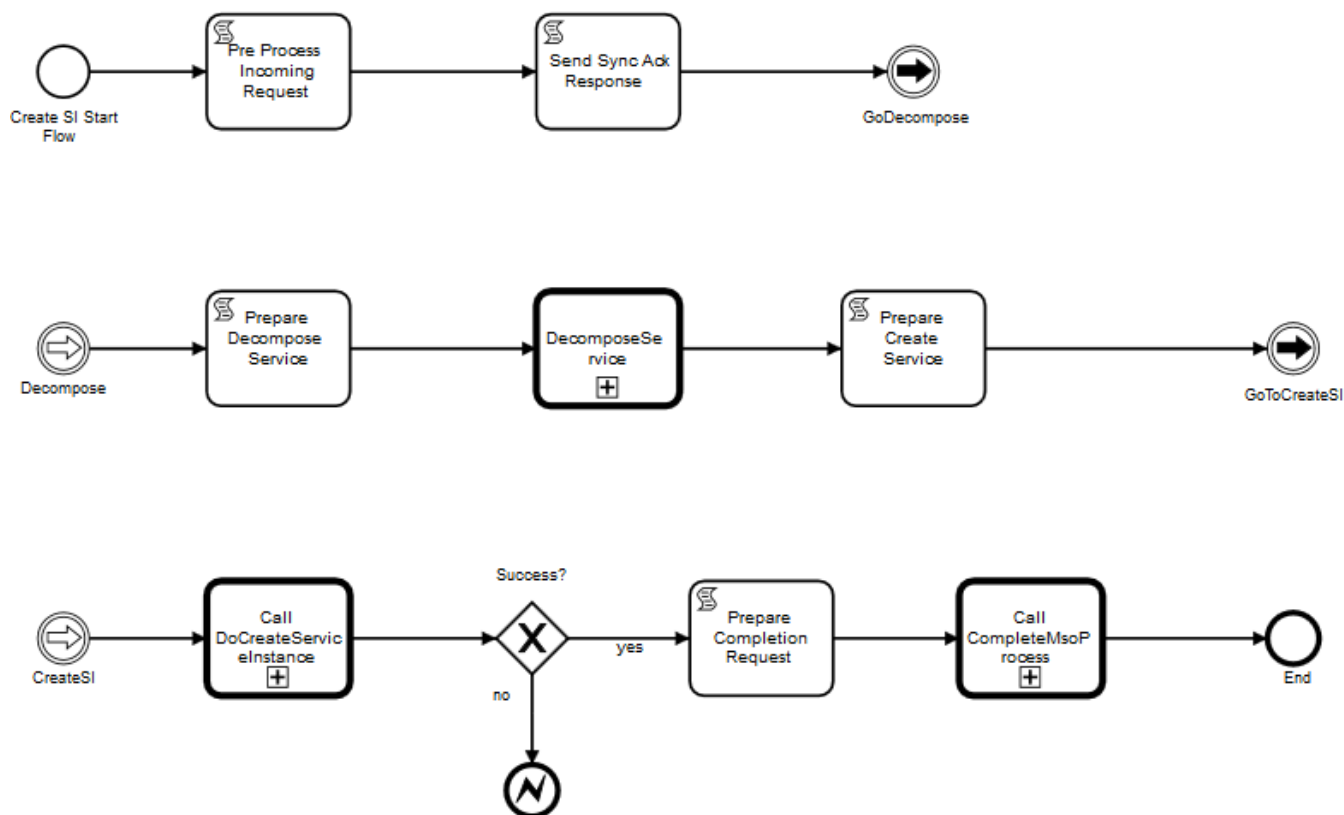
      ],
      "subscriptionServiceType":"vFWCL",
      "aLaCarte":true
    },
    "subscriberInfo":{
      "globalSubscriberId":"Demonstration",
      "subscriberName":"Demonstration"
    },
    "project":{
      "projectName":"Project-Demonstration"
    },
    "owningEntity":{
      "owningEntityId":"75c646bf-6b09-4759-b65d-ae366a40da91",
      "owningEntityName":"OE-Demonstration"
    }
  }
}
```

#### SO response

HTTP/1.1 202 Accepted

```
{
  "requestReferences":{
    "instanceId":"d39d0833-ea8a-45be-8863-76715697cdb3",
    "requestId":"08390a82-88f8-429f-9dd9-fa673bf5ed38"
  }
}
```

The VID request triggers the execution of **CreateGenericALaCarteServiceInstance** BPMN process.



You could find all BPMN processes and follow calls in Camunda Cockpit dashboard: [http://<so\\_ip>:8080/camunda/app/cockpit/default#/processes](http://<so_ip>:8080/camunda/app/cockpit/default#/processes)

This BPMN process calls respectively two sub-processes: **DecomposeService** and **DoCreateServiceInstance**

the DoCreateServiceInstance BPMN process includes **GenericPutService** process that allows to put service instance in the inventory via HTTP PUT request sent by Camunda engine.

PUT /business/customers/customer/{global-customer-id}/service-subscriptions/service-subscription/{service-type}/service-instances/service-instance/{service-instance-id}



Exchanges between SO and A&AI components are encrypted.

We can verify that service is created in the inventory: GET [https://<aai\\_ip>:8443/aai/v13/business/customers/customer/Demonstration/service-subscriptions/service-subscription/vFWCL/service-instances/service-instance/ce035b11-77f5-41ff-a6e8-e38fc8d4ccd4](https://<aai_ip>:8443/aai/v13/business/customers/customer/Demonstration/service-subscriptions/service-subscription/vFWCL/service-instances/service-instance/ce035b11-77f5-41ff-a6e8-e38fc8d4ccd4)

The VID follows the service orchestration request status by sending GET requests to SO:

#### Get orchestration request status

```
GET /ecom/mso/infra/orchestrationRequests/v6/08390a82-88f8-429f-9dd9-fa673bf5ed38 HTTP/1.1
Authorization: Basic SW5mcmFQb3J0YWxDbGllbnQ6cGFzc3dvcmQxJA==
X-FromAppId: VID
X-ECOMP-RequestID: 2ea804d4-9d5f-4ece-bf1e-95bc963c22ae
User-Agent: Jersey/2.23.1 (HttpClientConnection 1.8.0_171)
Cache-Control: no-cache
Pragma: no-cache
Host: vml.mso.simplesdemo.onap.org:8080
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```

## SO response

```
HTTP/1.1 200 OK
Expires: 0
Cache-Control: no-cache, no-store, must-revalidate
X-Powered-By: Undertow/1
Server: WildFly/10
Pragma: no-cache
Date: Thu, 19 Jul 2018 14:50:05 GMT
Connection: keep-alive
Content-Type: application/json
Content-Length: 1133

{
  "request": {
    "requestId": "08390a82-88f8-429f-9dd9-fa673bf5ed38",
    "startTime": "Thu, 19 Jul 2018 14:49:55 GMT",
    "requestScope": "service",
    "requestType": "createInstance",
    "requestDetails": {
      "modelInfo": {
        "modelInvariantId": "46c0d35f-4a34-4f5b-9ee3-fd65f6f74ac2",
        "modelType": "service",
        "modelName": "vfw_svc",
        "modelVersion": "2.0",
        "modelVersionId": "84909c11-1a62-47b3-9aed-59c2ec5c891d"
      },
      "requestInfo": {
        "source": "VID",
        "instanceName": "vfw_svc_1607",
        "suppressRollback": false,
        "requestorId": "demo"
      },
      "subscriberInfo": {
        "globalSubscriberId": "Demonstration",
        "subscriberName": "Demonstration"
      },
      "requestParameters": {
        "alaCarte": true,
        "subscriptionServiceType": "vFWCL",
        "aLaCarte": true
      },
      "project": {
        "projectName": "Project-Demonstration"
      },
      "owningEntity": {
        "owningEntityId": "75c646bf-6b09-4759-b65d-ae366a40da91",
        "owningEntityName": "OE-Demonstration"
      }
    },
    "instanceReferences": {
      "serviceInstanceId": "d39d0833-ea8a-45be-8863-76715697cdb3",
      "serviceInstanceName": "vfw_svc_1607",
      "requestorId": "demo"
    },
    "requestStatus": {
      "requestState": "COMPLETE",
      "statusMessage": "Service Instance was created successfully.",
      "percentProgress": 100,
      "finishTime": "Thu, 19 Jul 2018 14:49:56 GMT"
    }
  }
}
```





**Mechanism analysis will be explained only for the vFWSNK vnf and its module**

The vFW service includes two vnfs, the mechanism is the same for the two vnfs and their vf-module. So we will explain it for the vFWSNK vnf only.

## VNF creation (vFWSNK vnf)

In the VID portal, the user adds a VNF node instance:

### Create Virtual Network Function -- a la carte

**Service Name:** vfw\_svc  
**Subscriber Name:** Demonstration  
**Service Instance Name:** vfwcl\_svc\_1607  
**Model Name:** vsp\_vfwskn\_1607  
**Model Invariant UUID:** c7582c77-5a11-471c-a24f-5347ec811cdd  
**Model Version:** 2.0  
**Model UUID:** db0fbb27-fd0d-40eb-8491-0f4a48adaefe  
**Model Customization UUID:** 5365f464-3e2e-40be-9dc2-480c916bcf10  
**Resource Name:** vsp\_vfwskn\_1607 0  
**NF Type:**  
**NF Role:**  
**NF Function:**  
**NF Naming Code:**

User Provided Data ( \* indicates required field)

<b>Instance Name: *</b>	<input type="text" value="vfw-vf"/>
<b>Product Family: *</b>	<input type="text" value="vFWCL"/>
<b>LCP Region: *</b>	<input type="text" value="fr2"/>
<b>Tenant: *</b>	<input type="text" value="0750179787_ITNSchool"/>
<b>Line Of Business:</b>	<input type="text" value="LOB-Demonstration"/>
<b>Platform: *</b>	<input type="text" value="Platform-Demonstration"/>
<b>Suppress Rollback on Failure:</b>	<input type="text" value="false"/>

Enter Data and **Confirm** to  
Create **Virtual Network Function**

**Cancel** to Return to Previous Page.  
Data entered will be lost

Confirm

Cancel

The user action corresponds to an HTTP POST request from VID towards SO to trigger vnf instance creation: [POST /ecomp/mso/infra/serviceInstances/v6/{service\\_instance\\_id}/vnfs](#)

## Http POST request: Json payload for vnf instance creation

```
POST /ecompmso/infra/serviceInstances/v6/d39d0833-ea8a-45be-8863-76715697cdb3/vnfs HTTP/1.1
Authorization: Basic SW5mcmFQb3J0YWxDbGllbnQ6cGFzc3dvcmQxJA==
X-FromAppId: VID
X-ECOMP-RequestID: 41dc6f28-b847-4190-9573-a762cb77ab30
Content-Type: application/json
User-Agent: Jersey/2.23.1 (HttpURLConnection 1.8.0_171)
Cache-Control: no-cache
Pragma: no-cache
Host: vml.mso.simplesdemo.onap.org:8080
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-Length: 1018
```

```
{
  "requestDetails":{
    "requestInfo":{
      "instanceName":"vfw-vf",
      "source":"VID",
      "suppressRollback":false,
      "requestorId":"demo",
      "productFamilyId":"8eba27fe-89eb-4c32-9ee6-aa5428c7d962"
    },
    "modelInfo":{
      "modelType":"vnf",
      "modelInvariantId":"c7582c77-5a11-471c-a24f-5347ec811cdd",
      "modelVersionId":"db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
      "modelName":"vsp_vfwsk_1607",
      "modelVersion":"2.0",
      "modelCustomizationId":"5365f464-3e2e-40be-9dc2-480c916bcf10",
      "modelCustomizationName":"vsp_vfwsk_1607 0"
    },
    "requestParameters":{
      "userParams":[

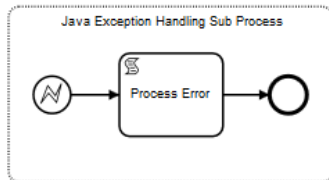
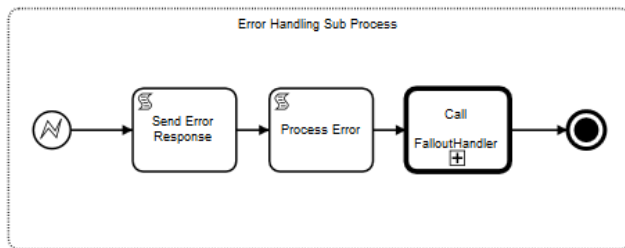
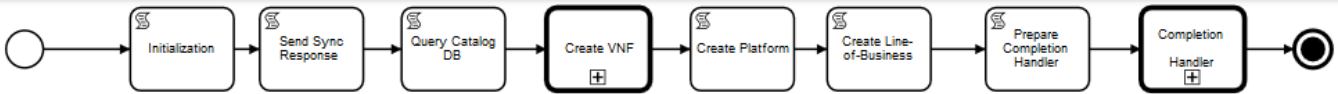
      ]
    },
    "cloudConfiguration":{
      "lcpCloudRegionId":"fr2",
      "tenantId":"324b90de6e9a4ad88e93a100c2cedd5d"
    },
    "lineOfBusiness":{
      "lineOfBusinessName":"LOB-Demonstration"
    },
    "platform":{
      "platformName":"Platform-Demonstration"
    },
    "relatedInstanceList":[
      {
        "relatedInstance":{
          "instanceId":"d39d0833-ea8a-45be-8863-76715697cdb3",
          "modelInfo":{
            "modelType":"service",
            "modelName":"vfw_svc",
            "modelInvariantId":"46c0d35f-4a34-4f5b-9ee3-fd65f6f74ac2",
            "modelVersion":"2.0",
            "modelVersionId":"84909c11-1a62-47b3-9aed-59c2ec5c891d"
          }
        }
      }
    ]
  }
}
```

## SO response

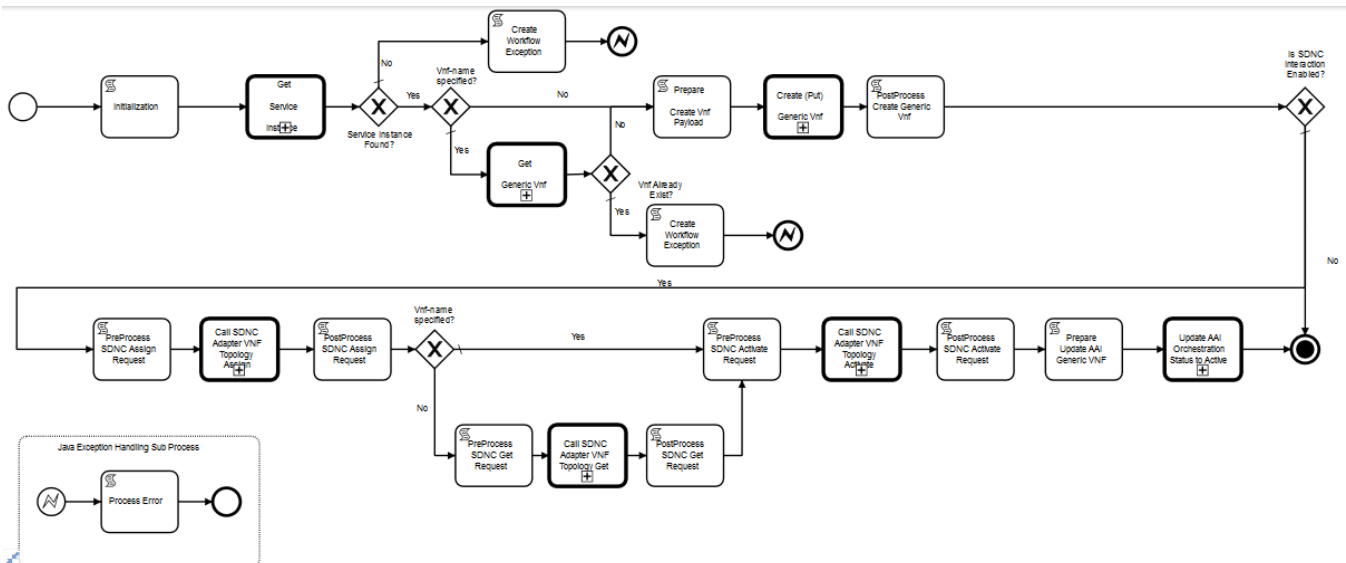
HTTP/1.1 202 Accepted

```
{ "requestReferences": { "instanceId": "415932c3-f295-462a-b130-bdf5f33ced06", "requestId": "e4c03a55-5d64-4867-9177-ee2cf9146dc4" } }
```

The VID request triggers the execution of **CreateVnfInfra** BPMN process:



The main actions of this process are defined in the sub process **DoCreateVnf**.



As specified in this sub process building blocks, the VNF instance creation invokes the following main actions:

- Populate A&AI with a VNF instance data: **Create (Put) Generic Vnf (GenericPutVnf sub-process)**
- Call SDN-C adapter VNF topology assign: **Call SDNC Adapter VNF Topology Assign**
- Call SDN-C adapter VNF topology activate: **Call SDNC Adapter VNF Topology Activate**
- Update orchestration status to active in A&AI: **Update AAI Orchestration status to Active**

As SDNC interaction is not enabled, only the first action (Create Generic Vnf) is performed.

**GenericPutVnf** sub-process populates the inventory with the vnf instance data via Http PUT request (PUT /network/generic-vnfs/generic-vnf/{vnf-id}) and establishes the relation between the vnf instance and service instance.

We can verify that vnf instances and relationship with service instance were created in the inventory: [https://<aai1\\_ip>:8443/aai/v13/network/generic-vnfs/generic-vnf/e1a9ac1b-b9a0-4424-8659-69426c8bdc8](https://<aai1_ip>:8443/aai/v13/network/generic-vnfs/generic-vnf/e1a9ac1b-b9a0-4424-8659-69426c8bdc8)

The VID follows the vnf orchestration request status by sending GET requests to SO:

#### Get orchestration request status

```
GET /ecomp/mso/infra/orchestrationRequests/v6/cfeb170d-7b0f-44db-81e3-ff440fce43ea HTTP/1.1
Authorization: Basic SW5mcmFQb3J0YWxDbGllbnQ6cGFzc3dvcmQxJA==
X-FromAppId: VID
X-ECOMP-RequestID: 0eaa4f8f-b6f0-4183-8f6b-bb1ba847efc8
User-Agent: Jersey/2.23.1 (HttpURLConnection 1.8.0_171)
Cache-Control: no-cache
Pragma: no-cache
Host: vml.mso.simplesdemo.onap.org:8080
Accept: text/html, image/gif, image/jpeg, */*; q=.2, */*; q=.2
Connection: keep-aliv
```

#### SO response

```
HTTP/1.1 200 OK
Expires: 0
Cache-Control: no-cache, no-store, must-revalidate
X-Powered-By: Undertow/1
Server: WildFly/10
Pragma: no-cache
Date: Thu, 19 Jul 2018 14:50:50 GMT
Connection: keep-alive
Content-Type: application/json
Content-Length: 1511

{
  "request": {
    "requestId": "cfeb170d-7b0f-44db-81e3-ff440fce43ea",
    "startTime": "Thu, 19 Jul 2018 14:50:40 GMT",
    "requestScope": "vnf",
    "requestType": "createInstance",
    "requestDetails": {
      "modelInfo": {
        "modelCustomizationName": "vsp_vfwsnk_1607 0",
        "modelInvariantId": "c7582c77-5a11-471c-a24f-5347ec811cdd",
        "modelType": "vnf",
        "modelName": "vsp_vfwsnk_1607",
        "modelVersion": "2.0",
        "modelVersionId": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
        "modelCustomizationId": "5365f464-3e2e-40be-9dc2-480c916bcf10"
      },
      "requestInfo": {
        "productFamilyId": "8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
        "source": "VID",
        "instanceName": "vfw-vf",
        "suppressRollback": false,
        "requestorId": "demo"
      }
    },
    "relatedInstanceList": [
      {
        "relatedInstance": {
          "instanceId": "d39d0833-ea8a-45be-8863-76715697cdb3",
          "modelInfo": {
            "modelInvariantId": "46c0d35f-4a34-4f5b-9ee3-fd65f6f74ac2",
            "modelType": "service",
            "modelName": "vfw_svc",
            "modelVersion": "2.0",
            "modelVersionId": "84909c11-1a62-47b3-9aed-59c2ec5c891d"
          }
        }
      }
    ]
  }
}
```

```

    }
  },
  "cloudConfiguration": {
    "tenantId": "324b90de6e9a4ad88e93a100c2cedd5d",
    "lcpCloudRegionId": "fr2"
  },
  "requestParameters": {
    "platform": {
      "platformName": "Platform-Demonstration"
    },
    "lineOfBusiness": {
      "lineOfBusinessName": "LOB-Demonstration"
    }
  },
  "instanceReferences": {
    "serviceInstanceId": "d39d0833-ea8a-45be-8863-76715697cdb3",
    "vnfInstanceId": "e1a9ac1b-b9a0-4424-8659-69426c8bfdc8",
    "vnfInstanceName": "vfw-vf",
    "requestorId": "demo"
  },
  "requestStatus": {
    "requestState": "COMPLETE",
    "statusMessage": "Vnf has been created successfully.",
    "percentProgress": 100,
    "finishTime": "Thu, 19 Jul 2018 14:50:41 GMT"
  }
}
}

```

## VF-module creation (vFWSNK VF-module)

In the vFW use-case, before creating the vf-module two steps should be performed: Create vnf profile (in sdnc portal) and preload sdnc with vnf data (using sdnc api [http://<sdnc\\_ip>:8282/apidoc/explorer/index.html](http://<sdnc_ip>:8282/apidoc/explorer/index.html) ).

In the VID portal, user demo adds then vf-module:

## Create VF Module -- a la carte

**Service Name:** vfw\_svc  
**Subscriber Name:** Demonstration  
**Service Instance Name:** vfwcl\_svc\_1607  
**Model Name:** VspVfwsnk1607..base\_vfw\_module-0  
**Model Invariant UUID:** 08fa9f0d-4fb5-41ae-8cc3-8282333ff6aa  
**Model Version:** 2  
**Model UUID:** f8d43f3f-8189-4eac-a122-b5a5e06ad3ab  
**Model Customization UUID:** bb084cb1-e8eb-4030-a2b7-931a3ae1c91f

User Provided Data (\* indicates required field)

<b>Instance Name: *</b>	<input type="text" value="vfw-vf-module"/>
<b>LCP Region: *</b>	<input type="text" value="fr2"/>
<b>Tenant: *</b>	<input type="text" value="0750179787_ITNSchool"/>
<b>Suppress Rollback on Failure:</b>	<input type="text" value="false"/>
<b>SDN-C Pre-Load:</b>	<input checked="" type="checkbox"/>
<b>Upload Supplementary Data file:</b>	<input type="checkbox"/>

Enter Data and **Confirm** to

Create **VF Module**

**Cancel** to Return to Previous Page.

Data entered will be lost

Confirm

Cancel

The VID triggers vf-module creation via Http POST request towards SO: [POST /ecomp/mso/infra/serviceInstances/v6/{service\\_instance\\_id}/vnfs/{vnf\\_id}/vfModules](#)

Http Post request: Json payload for vf-module creation (vFWSNK module)

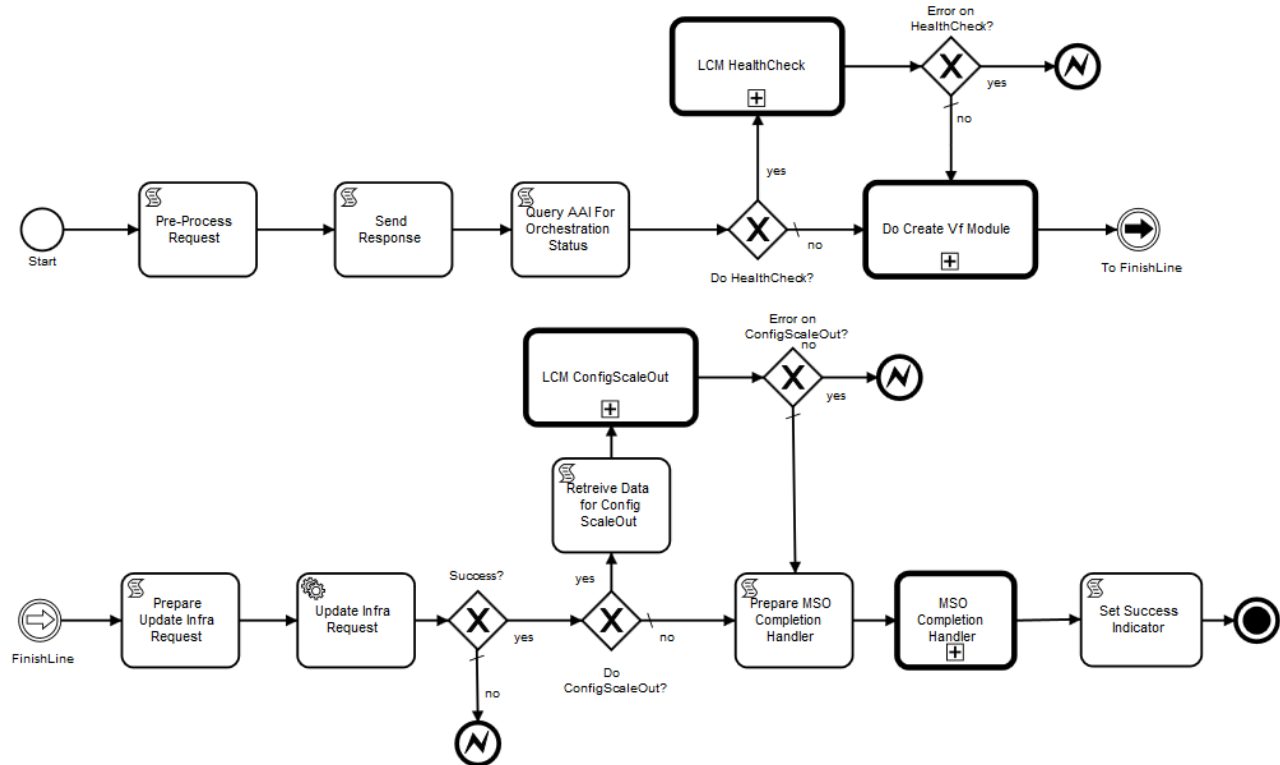
POST /ecompmso/infra/serviceInstances/v6/d39d0833-ea8a-45be-8863-76715697cdb3/vnfs/ela9ac1b-b9a0-4424-8659-69426c8bfcdc8/vfModules HTTP/1.1  
Authorization: Basic SW5mcmFQb3J0YWxDbGllbnQ6cGFzc3dvcmQxJA==  
X-FromAppId: VID  
X-ECOMP-RequestID: 329b557c-3e3f-42c6-8b41-241a9bc0eb27  
Content-Type: application/json  
User-Agent: Jersey/2.23.1 (HttpURLConnection 1.8.0\_171)  
Cache-Control: no-cache  
Pragma: no-cache  
Host: vml.mso.simplesdemo.onap.org:8080  
Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2  
Connection: keep-alive  
Content-Length: 1272

```
{
  "requestDetails": {
    "requestInfo": {
      "instanceName": "vfw-vf-module",
      "source": "VID",
      "suppressRollback": false,
      "requestorId": "demo"
    },
    "modelInfo": {
      "modelType": "vfModule",
      "modelInvariantId": "08fa9f0d-4fb5-41ae-8cc3-8282333ff6aa",
      "modelVersionId": "f8d43f3f-8189-4eac-a122-b5a5e06ad3ab",
      "modelName": "VspVfwsnk1607..base_vfw..module-0",
      "modelVersion": "2",
      "modelCustomizationId": "bb084cb1-e8eb-4030-a2b7-931a3ae1c91f",
      "modelCustomizationName": "VspVfwsnk1607..base_vfw..module-0"
    },
    "requestParameters": {
      "usePreload": true
    },
    "cloudConfiguration": {
      "lcpCloudRegionId": "fr2",
      "tenantId": "324b90de6e9a4ad88e93a100c2cedd5d"
    },
    "relatedInstanceList": [
      {
        "relatedInstance": {
          "instanceId": "d39d0833-ea8a-45be-8863-76715697cdb3",
          "modelInfo": {
            "modelType": "service",
            "modelName": "vfw_svc",
            "modelInvariantId": "46c0d35f-4a34-4f5b-9ee3-fd65f6f74ac2",
            "modelVersion": "2.0",
            "modelVersionId": "84909c11-1a62-47b3-9aed-59c2ec5c891d"
          }
        }
      },
      {
        "relatedInstance": {
          "instanceId": "ela9ac1b-b9a0-4424-8659-69426c8bfcdc8",
          "modelInfo": {
            "modelType": "vnf",
            "modelName": "vsp_vfwsnk_1607",
            "modelInvariantId": "c7582c77-5a11-471c-a24f-5347ec811cdd",
            "modelVersion": "2.0",
            "modelVersionId": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
            "modelCustomizationId": "5365f464-3e2e-40be-9dc2-480c916bcf10",
            "modelCustomizationName": "vsp_vfwsnk_1607 0"
          }
        }
      }
    ]
  }
}
```

## SO response

```
HTTP/1.1 202 Accepted
{"requestReferences":{"instanceId":"be0ae48e-029a-4efc-af46-d11bcf58c403","requestId":"7b1b5758-d274-4305-ae4c-289f1e614d7d"}}
```

The API handler selects a BPMN workflow called **CreateVfModuleInfra**.



In addition to vf-module creation, this BPMN process could perform scale out operation for a vf-module instance but it is not the case for the vFW.

The **DoCreateVfModule** is the main subprocess that defines steps for creating the vf-module. It includes the following actions that are performed in the vFW use-case deployment:

1. Query AAI for orchestration status
2. CreateAAIVfModule: subprocess that creates a vf-module instance object in the inventory:HTTP PUT /network/generic-vnfs/generic-vnf/{vnf-id}/vf-modules/vf-module/{vf-module-id}

We can verify the vf-module instance in the inventory with GET request.

GET [https://<aai1\\_ip>:8443/aai/v13/network/generic-vnfs/generic-vnf/e1a9ac1b-b9a0-4424-8659-69426c8bfdc8/vf-modules/](https://<aai1_ip>:8443/aai/v13/network/generic-vnfs/generic-vnf/e1a9ac1b-b9a0-4424-8659-69426c8bfdc8/vf-modules/)

3. Call SDNC adapter: Vf-module topology assign

To perform this action, Camunda engine calls SDNC adapter via Http request:POST /restconf/operations/VNF-API:vnf-topology-operation



## SO Http POST request towards SDNC to trigger vf-module topology assign

```
<?xml version="1.0" encoding="UTF-8"?>
<input xmlns="org:onap:sdnctl:vnf">
  <sdnc-request-header>
    <svc-request-id>7b1b5758-d274-4305-aefc-289f1e614d7d-1532011988473</svc-request-id>
    <svc-action>assign</svc-action>
    <svc-notification-url>http://cl.vml.mso.simplesdemo.openecomp.org:8080/adapters/rest/SDNCNotify</svc-
notification-url>
  </sdnc-request-header>
  <request-information>
    <request-id>7b1b5758-d274-4305-aefc-289f1e614d7d</request-id>
    <request-action>VNFActivateRequest</request-action>
    <source>VID</source>
    <notification-url />
  </request-information>
  <service-information>
    <service-id>null</service-id>
    <service-type>null</service-type>
    <service-instance-id>d39d0833-ea8a-45be-8863-76715697cdb3</service-instance-id>
    <subscriber-name>notsurewecare</subscriber-name>
  </service-information>
  <vnf-request-information>
    <vnf-id>be0ae48e-029a-4efc-af46-d11bcf58c403</vnf-id>
    <vnf-type>VspVfwsnk1607..base_vfw..module-0</vnf-type>
    <vnf-name>vfw-vf-module</vnf-name>
    <generic-vnf-id>ela9ac1b-b9a0-4424-8659-69426c8bfdc8</generic-vnf-id>
    <generic-vnf-name>vfw-vf</generic-vnf-name>
    <generic-vnf-type>vfw_svc/vsp_vfwsnk_1607_0</generic-vnf-type>
    <aic-cloud-region>fr2</aic-cloud-region>
    <tenant>324b90de6e9a4ad88e93a100c2cedd5d</tenant>
    <use-preload>Y</use-preload>
  </vnf-request-information>
</input>
```

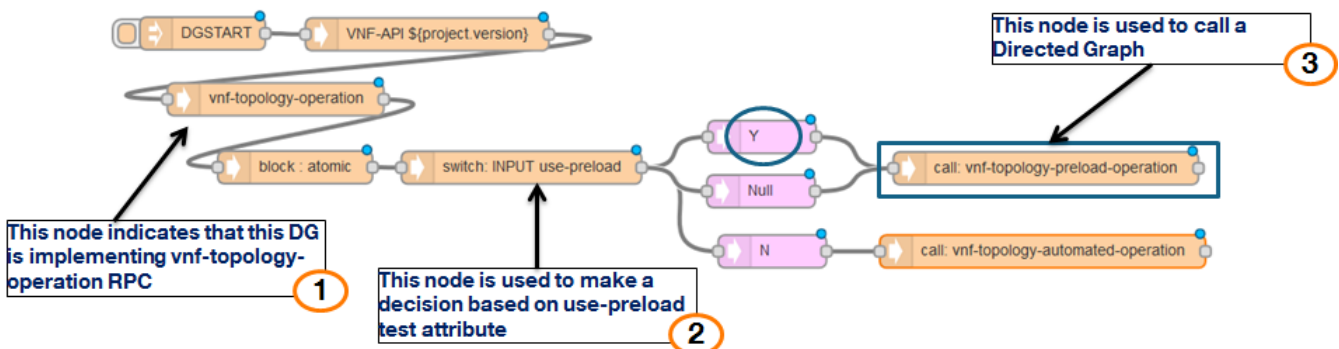
This request triggers the execution of a set of **Directed Graphs**



Git repository of VNF-API DG json: <https://git.onap.org/sdnctl/oam/tree/platform-logic/vnfapi/src/main/json?h=beijing>

You can see Directed graphs by importing Json (to clipboard) in the SDNC- DG dashboard [http://<sdnc\\_ip>:3000](http://<sdnc_ip>:3000)

### ◦ vnf-topology-operation DG



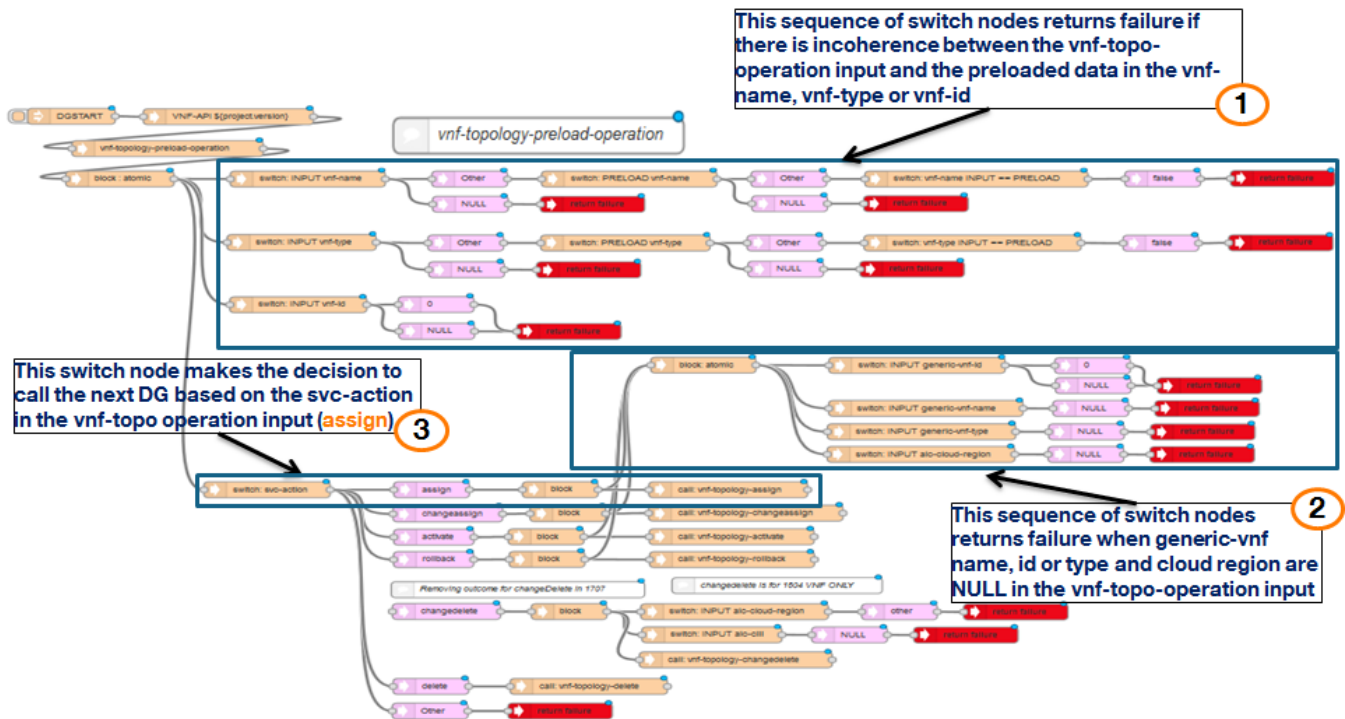
A DG implements a method that initiates a RPC when executed.

By parsing the request body fields, this DG calls a specific second DG based on the use-preload value

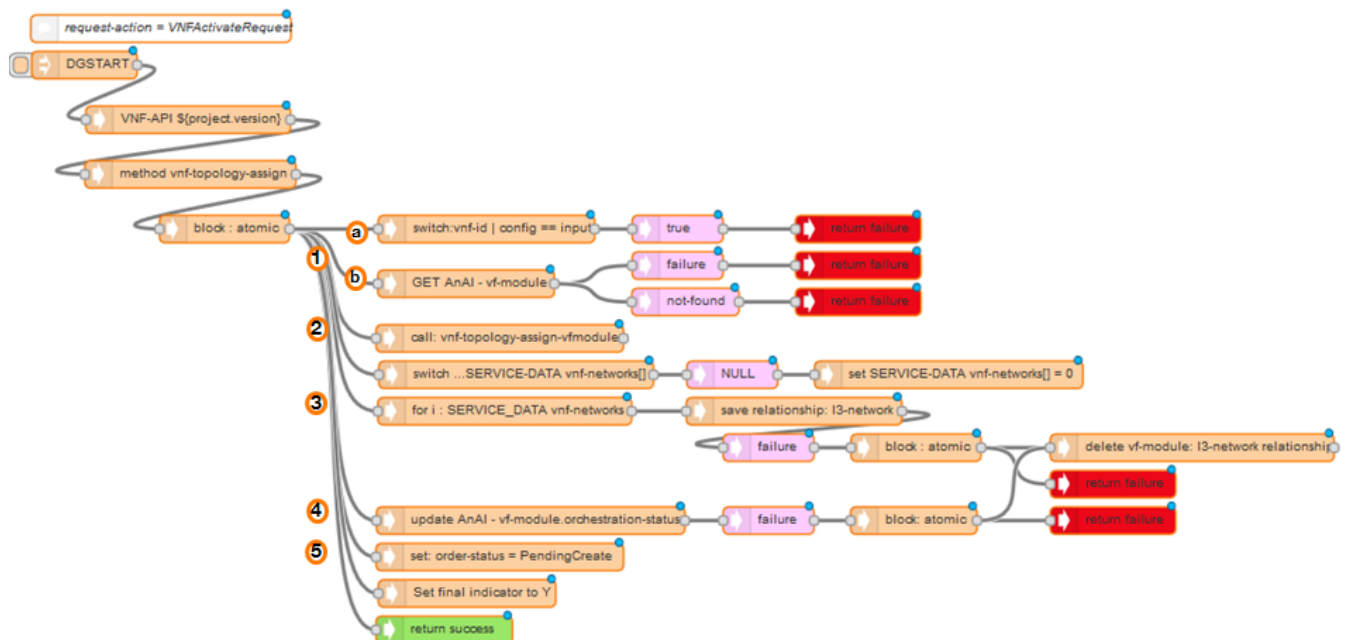
This condition figures out in the POST request body with Y as value as we have manually preloaded SDNC with vnf data.

This DG calls a second called vnf-topology-preload-operation

### ◦ vnf-topology-preload-operation DG



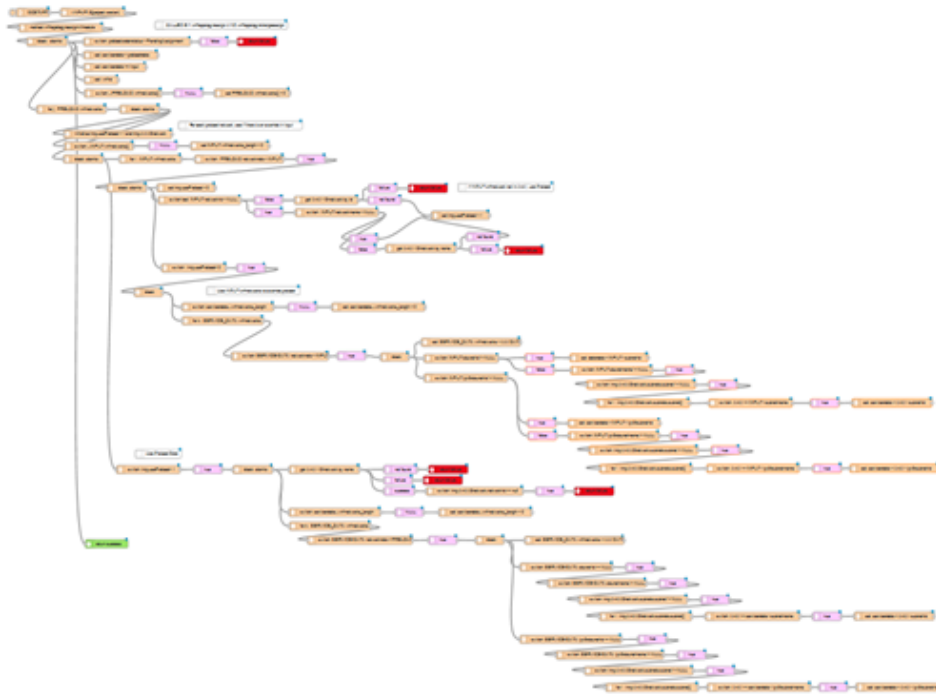
#### vnf-topology-assign DG



This DG vnf-topology-assign includes a sequence of nodes that execute respectively these actions:

- Return failure if:
  - VF-module is already assigned
  - The VF-module does not exist in A&AI
- Call **vnf-topology-assign-vfmodule** DG: this DG constructs a data structure called service-data to be stored in its internal DB and potentially used to perform other actions
- Establish relationships between vf-module and networks in A&AI (if there are networks specified in service data)
- Update the vf-module orchestration status in A&AI (value = PendingCreate)
- Update order status in service data in SDN-C

#### vnf-topology-assign-vfmodule



This DG begins with constructing service-data structure by concatenating preload input and vnf-topology-operation input

Then the service-data is consolidated by information related to I3-networks (id, name, neutron network id, contrail network fqdn, ipv4/6 subnets)

The DG logic:

For network in preload data

- See if there is a network with the same role in the input vnf-topology-operation
- YES, and this network exists in AAI Set these information in service-data
- NO --> Use preload data to be set in service-data



#### Summary of the vf-module topology assign step

The SO request towards SDN-C (POST /restconf/operations/VNF-API:vnf-topology-operation) triggers the execution of a sequence of Directed graphs.

The main goals of this operation are:

- Construct the service-data structure in the SDN-C
- Establish relationship between VF-module and networks using service-data
- Update the vf-module orchestration status in the inventory
- Update the order status in service-data

In the vFWCL use-case, no networks were defined in the preload data or in vnf-topology-operation input

- **The SDN-C actions at this stage are limited to updates of vf-module orchestration status in AAI**

Communications with AAI are done through sdn-c aai-service plugin

No relationships were established between vf-module and I3-networks in the vFWCL use-case.

4. Update vf-module orchestration-status in the inventory to assigned

5. Query SDNC for vf-module topology

In order to get vf-module topology, the SO requests SDNC via its controller adapter: GET /restconf/config/VNF-API:vnfs/vnf-list/be0ae48e-029a-4efc-af46-d11bcf58c403

#### Http GET Vf-module topology response (vFWSNK vnf)

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONID=1czkavxlg5yq715qw2lx7wcbfq; Path=/restconf
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: rememberMe=deleteMe; Path=/restconf; Max-Age=0; Expires=Wed, 18-Jul-2018 14:53:16 GMT
Content-Type: application/yang.data+xml
```

Vary: Accept-Encoding, User-Agent  
Transfer-Encoding: chunked

```
<?xml version="1.0" encoding="UTF-8"?>
<vnf-list xmlns="org:onap:sdnctl:vnf">
  <vnf-id>be0ae48e-029a-4efc-af46-d11bcf58c403</vnf-id>
  <service-data>
    <vnf-request-information>
      <vnf-type>VspVfwsnk1607..base_vfw..module-0</vnf-type>
      <vnf-id>be0ae48e-029a-4efc-af46-d11bcf58c403</vnf-id>
      <generic-vnf-name>vfw-vf</generic-vnf-name>
      <use-preload>Y</use-preload>
      <generic-vnf-type>vfw_svc/vsp_vfwsnk_1607_0</generic-vnf-type>
      <tenant>324b90de6e9a4ad88e93a100c2cedd5d</tenant>
      <vnf-name>vfw-vf-module</vnf-name>
      <aic-cloud-region>fr2</aic-cloud-region>
      <generic-vnf-id>ela9ac1b-b9a0-4424-8659-69426c8bfdc8</generic-vnf-id>
    </vnf-request-information>
    <vnf-topology-information>
      <vnf-parameters>
        <vnf-parameter-name>unprotected_private_net_id</vnf-parameter-name>
        <vnf-parameter-value>unprotected_net</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>protected_private_net_id</vnf-parameter-name>
        <vnf-parameter-value>protected_net</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>ext_private_net_id</vnf-parameter-name>
        <vnf-parameter-value>external_net</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>protected_private_net_cidr</vnf-parameter-name>
        <vnf-parameter-value>192.168.20.0/24</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>key_name</vnf-parameter-name>
        <vnf-parameter-value>vfw_key</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>vfw_flavor_name</vnf-parameter-name>
        <vnf-parameter-value>n2.cw.standard-1</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>cloud_env</vnf-parameter-name>
        <vnf-parameter-value>openstack</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>ext_private_net_cidr</vnf-parameter-name>
        <vnf-parameter-value>192.168.30.0/24</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>vsname_0</vnf-parameter-name>
        <vnf-parameter-value>vsname</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>onap_private_net_id</vnf-parameter-name>
        <vnf-parameter-value>oam_onap_tr30</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>unprotected_private_subnet_id</vnf-parameter-name>
        <vnf-parameter-value>unprotected_subnet</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>repo_url_artifacts</vnf-parameter-name>
        <vnf-parameter-value>https://nexus.onap.org/content/repositories/releases</vnf-parameter-value>
      </vnf-parameters>
      <vnf-parameters>
        <vnf-parameter-name>vfw_name_0</vnf-parameter-name>
        <vnf-parameter-value>vfw-vf</vnf-parameter-value>
      </vnf-parameters>
    </vnf-topology-information>
  </service-data>
</vnf-list>
```

```
<vnf-parameters>
  <vnf-parameter-name>repo_url_blob</vnf-parameter-name>
  <vnf-parameter-value>https://nexus.onap.org/content/sites/raw</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>dcae_collector_port</vnf-parameter-name>
  <vnf-parameter-value>8081</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>public_net_id</vnf-parameter-name>
  <vnf-parameter-value>public</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vf_module_id</vnf-parameter-name>
  <vnf-parameter-value>vFirewallCL</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>onap_private_net_cidr</vnf-parameter-name>
  <vnf-parameter-value>10.0.0.0/16</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vnf_id</vnf-parameter-name>
  <vnf-parameter-value>vFirewall_demo_app</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vfw_image_name</vnf-parameter-name>
  <vnf-parameter-value>Ubuntu 14.04</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>onap_private_subnet_id</vnf-parameter-name>
  <vnf-parameter-value>oam_onap_tR30</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>install_script_version</vnf-parameter-name>
  <vnf-parameter-value>1.2.1</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vpg_private_ip_0</vnf-parameter-name>
  <vnf-parameter-value>192.168.10.200</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vfw_private_ip_3</vnf-parameter-name>
  <vnf-parameter-value>192.168.30.101</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vfw_private_ip_2</vnf-parameter-name>
  <vnf-parameter-value>10.0.100.1</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>demo_artifacts_version</vnf-parameter-name>
  <vnf-parameter-value>1.2.1</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vfw_private_ip_1</vnf-parameter-name>
  <vnf-parameter-value>192.168.20.100</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vfw_private_ip_0</vnf-parameter-name>
  <vnf-parameter-value>192.168.10.100</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vs_n_private_ip_2</vnf-parameter-name>
  <vnf-parameter-value>192.168.30.102</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vs_n_private_ip_1</vnf-parameter-name>
  <vnf-parameter-value>10.0.100.3</vnf-parameter-value>
</vnf-parameters>
<vnf-parameters>
  <vnf-parameter-name>vs_n_private_ip_0</vnf-parameter-name>
  <vnf-parameter-value>192.168.20.250</vnf-parameter-value>
```

```

    </vnf-parameters>
    <vnf-parameters>
      <vnf-parameter-name>pub_key</vnf-parameter-name>
      <vnf-parameter-value>ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDYI15cdBmIs2XOpe4EiFCsaY6bmUmK
/GysMoLl4UG5lJcFJwvwoWCoA+6mDIbymZxhxq9IGxilp/yTA6WQ9s
/5pBaglcUMJmFuda9PjOkXl04jgqh5tR6I+GZ97AvCg93KAECis5ubSqw1xOCj4utfEUtPoF1OuzqM
/lE5mY4N6VKXn+fT7pCD6cifBEs6JHhVNvs5OLLp
/tO8Pa3kKYQodyS0xc3rh+t2lrzvKUSWGZbX+dLiFiEpjsUL3tDqzkEMNUn4pdv690JuzWHCXRWPfdrY9Wg0j3mJesP29EBht+w+EC9
/kBKq+1VKdmsXUXAcjEvjovVL8l1BrX3BY0R8D</vnf-parameter-value>
    </vnf-parameters>
    <vnf-parameters>
      <vnf-parameter-name>ext_private_subnet_id</vnf-parameter-name>
      <vnf-parameter-value>external_subnet</vnf-parameter-value>
    </vnf-parameters>
    <vnf-parameters>
      <vnf-parameter-name>dcae_collector_ip</vnf-parameter-name>
      <vnf-parameter-value>84.39.39.157</vnf-parameter-value>
    </vnf-parameters>
    <vnf-parameters>
      <vnf-parameter-name>protected_private_subnet_id</vnf-parameter-name>
      <vnf-parameter-value>protected_subnet</vnf-parameter-value>
    </vnf-parameters>
    <vnf-parameters>
      <vnf-parameter-name>unprotected_private_net_cidr</vnf-parameter-name>
      <vnf-parameter-value>192.168.10.0/24</vnf-parameter-value>
    </vnf-parameters>
    <vnf-topology-identifier>
      <service-type>d39d0833-ea8a-45be-8863-76715697cdb3</service-type>
      <vnf-type>VspVfwsnk1607..base_vfw..module-0</vnf-type>
      <generic-vnf-name>vfw-vf</generic-vnf-name>
      <generic-vnf-type>vsp_vfwsnk_1607_0</generic-vnf-type>
      <vnf-name>vfw-vf-module</vnf-name>
    </vnf-topology-identifier>
  </vnf-topology-information>
  <service-information>
    <service-id>null</service-id>
    <service-type>null</service-type>
    <service-instance-id>d39d0833-ea8a-45be-8863-76715697cdb3</service-instance-id>
    <subscriber-name>notsurewecare</subscriber-name>
  </service-information>
  <vnf-id>be0ae48e-029a-4efc-af46-d11bcf58c403</vnf-id>
  <sdnc-request-header>
    <svc-notification-url>http://c1.vml.mso.simplifiedemo.openecomp.org:8080/adapters/rest/SDNCNotify</svc-
notification-url>
    <svc-request-id>7b1b5758-d274-4305-aefc-289f1e614d7d-1532011988473</svc-request-id>
    <svc-action>assign</svc-action>
  </sdnc-request-header>
  <oper-status>
    <order-status>PendingCreate</order-status>
  </oper-status>
  <request-information>
    <request-action>VNFActivateRequest</request-action>
    <source>VID</source>
    <request-id>7b1b5758-d274-4305-aefc-289f1e614d7d</request-id>
  </request-information>
  </service-data>
  <service-status>
    <final-indicator>Y</final-indicator>
    <request-status>synccomplete</request-status>
    <response-code>200</response-code>
    <vnfsdn-action>VNFActivateRequest</vnfsdn-action>
    <rpc-name>vnf-topology-operation</rpc-name>
    <rpc-action>assign</rpc-action>
    <response-timestamp>2018-07-19T14:53:09.027Z</response-timestamp>
  </service-status>
</vnf-list>

```

## 6. Call VNF adapter to create vf-module

This step of the DoCreateVfModule subprocess corresponds to the allocation of the service related virtual resources and the deployment of the stacks. In fact, the SO invokes the infrastructure orchestration services (Heat in the case of Openstack-based infrastructure) via its VNF-adapters. The stack is created using heat template available in SO catalog.

7. Update AAI vf-module

8. Call SDNC adapter: Vf-module topology activate

Same as the second step of this BPMN subprocess, this step corresponds to the execution of a set of Directed Graphs.

The SO HTTP Post request triggers the execution of **vnf-topology-operation DG**

#### SO Http POST request to trigger vf-module topology activate (vFWSNK vnf)

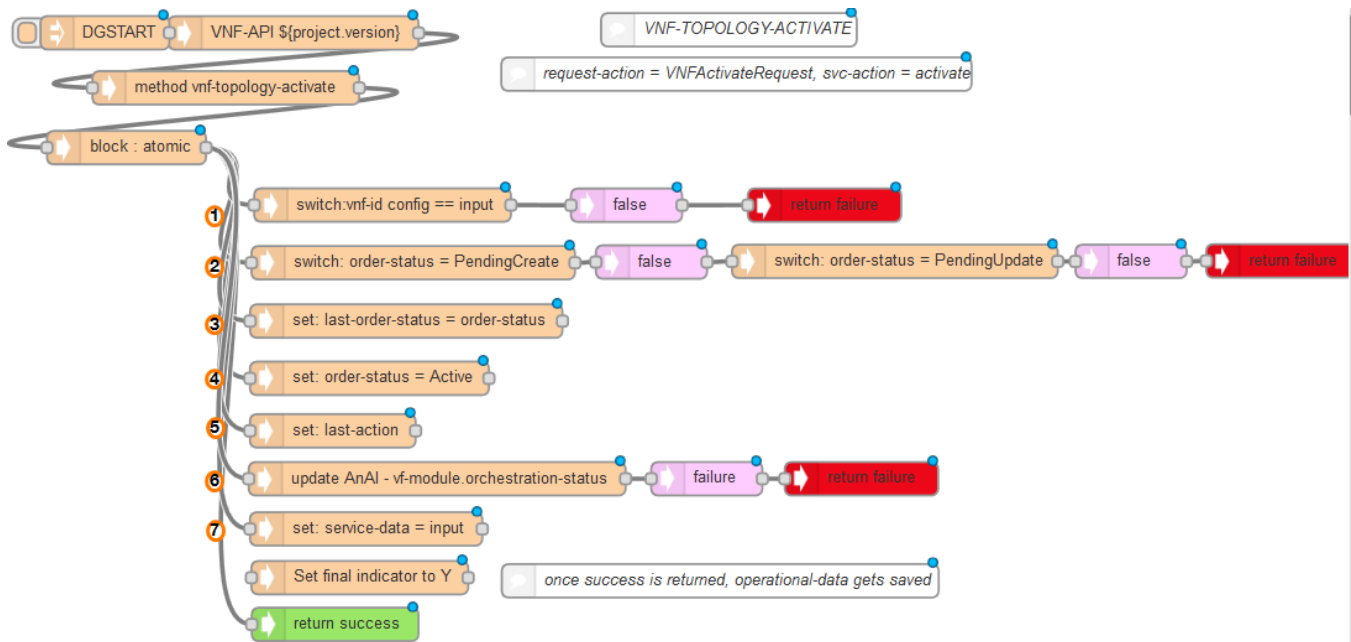
```
POST /restconf/operations/VNF-API:vnf-topology-operation HTTP/1.1
Accept: application/yang.data+xml
Authorization: Basic YWRtaW46S3A4Yko0Ulhzek0wVlhsaGFrM2VIbGNzZTJnQXc4NHZhb0dHbUp2VXkyVQ==
Content-type: application/xml
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/1.8.0_171
Host: cl.vml.sdnc.simplesdemo.openecomp.org:8282
Connection: keep-alive
Content-Length: 1272

<?xml version="1.0" encoding="UTF-8"?>
<input xmlns="org:onap:sdnctl:vnf">
  <sdnc-request-header>
    <svc-request-id>7b1b5758-d274-4305-aefc-289f1e614d7d-1532012022299</svc-request-id>
    <svc-action>activate</svc-action>
    <svc-notification-url>http://cl.vml.mso.simplesdemo.openecomp.org:8080/adapters/rest/SDNCNotify</svc-notification-url>
  </sdnc-request-header>
  <request-information>
    <request-id>7b1b5758-d274-4305-aefc-289f1e614d7d</request-id>
    <request-action>VNFActivateRequest</request-action>
    <source>VID</source>
    <notification-url />
  </request-information>
  <service-information>
    <service-id>null</service-id>
    <service-type>null</service-type>
    <service-instance-id>d39d0833-ea8a-45be-8863-76715697cdb3</service-instance-id>
    <subscriber-name>notsurewecare</subscriber-name>
  </service-information>
  <vnf-request-information>
    <vnf-id>be0ae48e-029a-4efc-af46-d11bcf58c403</vnf-id>
    <vnf-type>VspVfwsnk1607..base_vfw..module-0</vnf-type>
    <vnf-name>vfw-vf-module</vnf-name>
    <generic-vnf-id>ela9ac1b-b9a0-4424-8659-69426c8bfdc8</generic-vnf-id>
    <generic-vnf-name>vfw-vf</generic-vnf-name>
    <generic-vnf-type>vfw_svc/vsp_vfwsnk_1607_0</generic-vnf-type>
    <aic-cloud-region>fr2</aic-cloud-region>
    <tenant>324b90de6e9a4ad88e93a100c2cedd5d</tenant>
    <use-preload>Y</use-preload>
  </vnf-request-information>
</input>
```

Based on the POST request yang payload, (use-preload = Yes, svc-action = activate):

**vnf-topology-operation DG vnf-topology-preload-operation DG vnf-topology-activate DG.**

- **vnf-topology-activate DG**



This DG "vnf-topology-activate" includes a sequence of nodes that execute the following actions:

- 1.Return failure if the vf-module is not assigned
- 2.Return failure if the order status is not PendingCreate or PendingUpdate
- 3.Set last order status in service-data as with the actual value of order status (PendingCreate or PendingUpdate)
- 4.Set order status in service-data as active
- 5.Set last action in service-data as VNFActivateRequest
- 6.Update the vf-module orchestration status in A&AI (value= active)
- 7.Set service-data with the actual vnf-topology-operation input

Finally, the VID Gets the orchestration status of the vf-module creation operation

#### Get orchestration request status

```

GET /ecomp/mso/infra/orchestrationRequests/v6/18c536e3-9208-46e5-8c7f-131f9e645439 HTTP/1.1
Authorization: Basic SW5mcmFQb3J0YWxDbGllbnQ6cGFzc3dvcmQxJA==
X-FromAppId: VID
X-ECOMP-RequestID: 1a14556b-0e01-49db-a6e4-e000505d6116
User-Agent: Jersey/2.23.1 (HttpURLConnection 1.8.0_171)
Cache-Control: no-cache
Pragma: no-cache
Host: vml.mso.simplesdemo.onap.org:8080
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
  
```

#### Get orchestration request status

```

HTTP/1.1 200 OK
Expires: 0
Cache-Control: no-cache, no-store, must-revalidate
X-Powered-By: Undertow/1
Server: WildFly/10
Pragma: no-cache
Date: Thu, 19 Jul 2018 14:53:48 GMT
Connection: keep-alive
Content-Type: application/json
Content-Length: 1863

{
  "request":{
    "requestId":"7b1b5758-d274-4305-aefc-289f1e614d7d",
    "startTime":"Thu, 19 Jul 2018 14:53:08 GMT",
    "requestScope":"vfModule",
  
```



```

"requestType": "createInstance",
"requestDetails": {
  "modelInfo": {
    "modelCustomizationName": "VspVfwsnk1607..base_vfw..module-0",
    "modelInvariantId": "08fa9f0d-4fb5-41ae-8cc3-8282333ff6aa",
    "modelType": "vfModule",
    "modelName": "VspVfwsnk1607..base_vfw..module-0",
    "modelVersion": "2",
    "modelVersionId": "f8d43f3f-8189-4eac-a122-b5a5e06ad3ab",
    "modelCustomizationId": "bb084cb1-e8eb-4030-a2b7-931a3ae1c91f"
  },
  "requestInfo": {
    "source": "VID",
    "instanceName": "vfw-vf-module",
    "suppressRollback": false,
    "requestorId": "demo"
  },
  "relatedInstanceList": [
    {
      "relatedInstance": {
        "instanceId": "d39d0833-ea8a-45be-8863-76715697cdb3",
        "modelInfo": {
          "modelInvariantId": "46c0d35f-4a34-4f5b-9ee3-fd65f6f74ac2",
          "modelType": "service",
          "modelName": "vfw_svc",
          "modelVersion": "2.0",
          "modelVersionId": "84909c11-1a62-47b3-9aed-59c2ec5c891d"
        }
      }
    },
    {
      "relatedInstance": {
        "instanceId": "ela9ac1b-b9a0-4424-8659-69426c8bfdc8",
        "modelInfo": {
          "modelCustomizationName": "vsp_vfwsnk_1607_0",
          "modelInvariantId": "c7582c77-5a11-471c-a24f-5347ec811cdd",
          "modelType": "vnf",
          "modelName": "vsp_vfwsnk_1607",
          "modelVersion": "2.0",
          "modelVersionId": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
          "modelCustomizationId": "5365f464-3e2e-40be-9dc2-480c916bcf10"
        }
      }
    }
  ],
  "cloudConfiguration": {
    "tenantId": "324b90de6e9a4ad88e93a100c2cedd5d",
    "lcpCloudRegionId": "fr2"
  },
  "requestParameters": {
    "usePreload": true
  }
},
"instanceReferences": {
  "serviceInstanceId": "d39d0833-ea8a-45be-8863-76715697cdb3",
  "vnfInstanceId": "ela9ac1b-b9a0-4424-8659-69426c8bfdc8",
  "vfModuleInstanceId": "be0ae48e-029a-4efc-af46-d11bcf58c403",
  "vfModuleInstanceName": "vfw-vf-module",
  "requestorId": "demo"
},
"requestStatus": {
  "requestState": "COMPLETE",
  "statusMessage": "Vf Module has been created successfully.",
  "percentProgress": 100,
  "finishTime": "Thu, 19 Jul 2018 14:53:43 GMT"
}
}

```

## Analysis

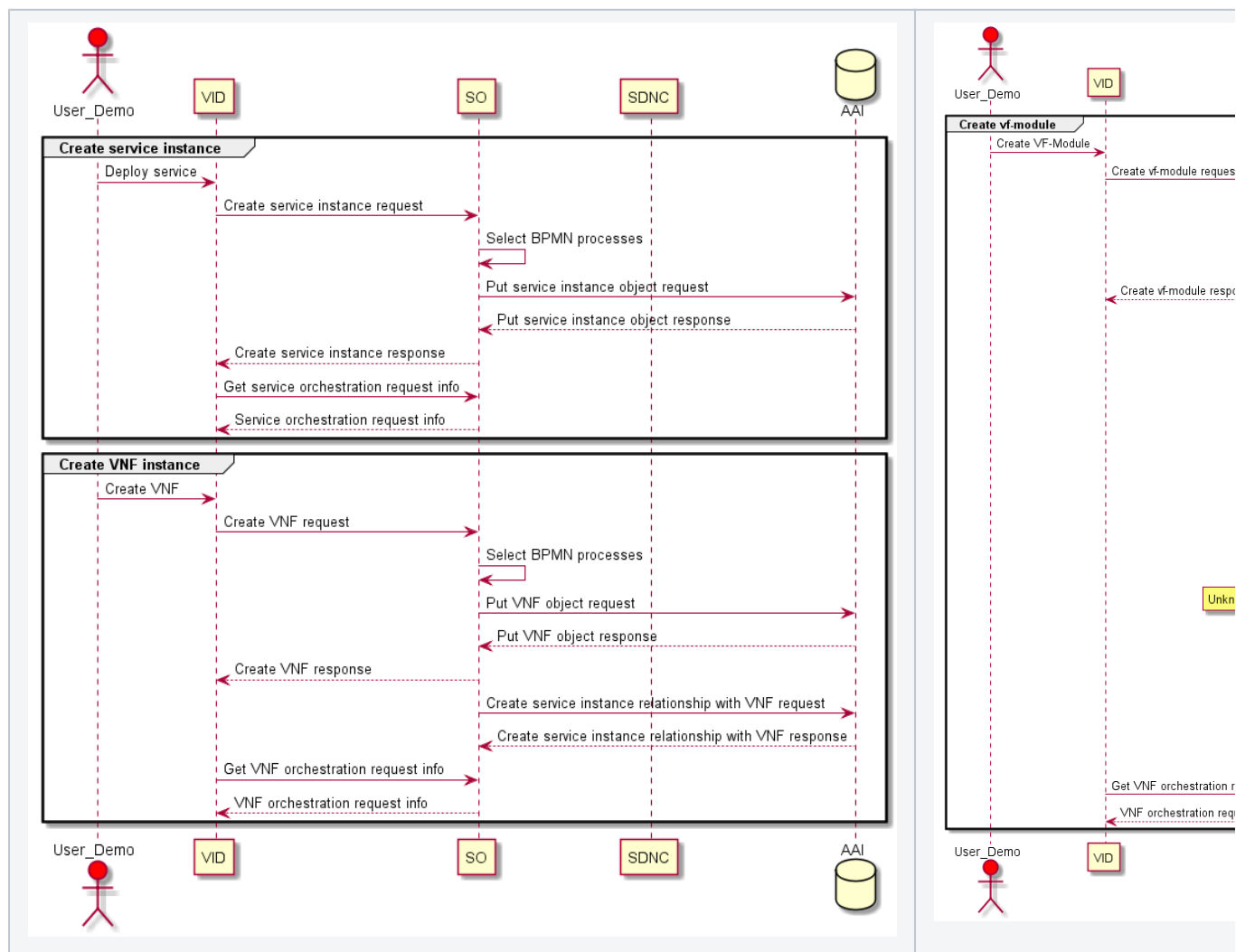
The vFW service deployment corresponds to three user operations on VID portal. These user actions trigger respectively service, vnf node and vf-module instantiation by the SO.

Service and vnf instantiation corresponds to the creation of an object instance in the inventory.

The vf-module creation invokes several ONAP components in order to create the vf-module object instance in the inventory and deploy service virtualized resources in the infrastructure.

The SO BPMN processes invoked in the vFW deployment are generic and service creation is based on requests parameters. ( **a la carte** )

The SDNC role in the vFW deployment is limited to the creation of service related records (service-data) in the SDNC local data base. This is due to the vFW service itself because it doesn't include networks defined within the service models.

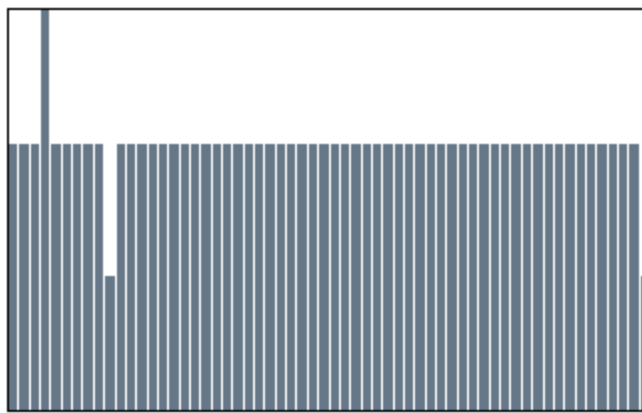


## vFW closed loop mechanism

### vFW VES measurements reporting

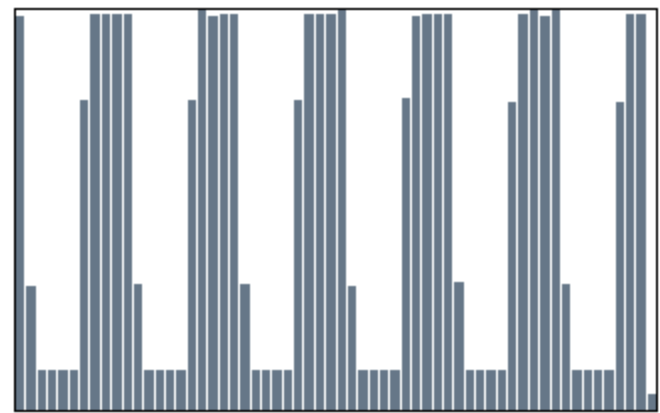
Before running the close loop, let's investigate what happens on the vFW service side and how the vFW component interacts with DCAE.

The packet generator sends UDP packets to the sink through the vfirewall. The sink shows in its graphs periodical traffic volumes:



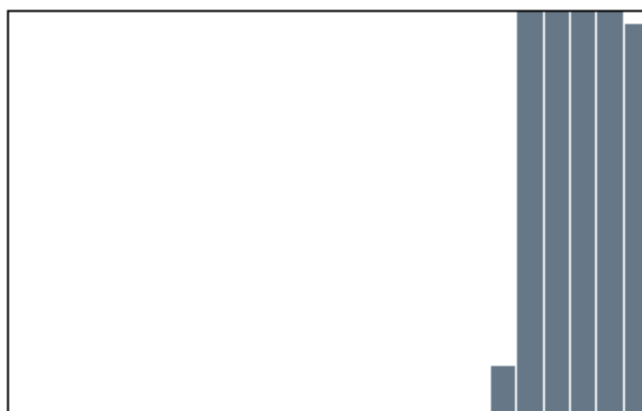
in ■ min: 1.4 KB/s, avg: 2.8 KB/s, max: 4.2 KB/s  
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s

**last 60 seconds**



in ■ min: 0.1 KB/s, avg: 1.5 KB/s, max: 2.9 KB/s  
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s

**last 60 minutes**



in ■ min: 0.2 KB/s, avg: 0.3 KB/s, max: 1.6 KB/s  
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s

**last 24 hours**



in ■ min: 0.3 KB/s, avg: 0.0 KB/s, max: 0.3 KB/s  
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s

**last 31 days**

Meanwhile, the vFW component includes a VES agent that reports measurements to DCAE collector (VES or VNF Event Stream) on port 8081.

Example 1: High traffic ~ 1000 packets (100 packets x 10 streams)

#### VES agent reporting

```
POST /eventListener/v5 HTTP/1.1
Authorization: Basic Og==
User-Agent: libcurl-agent/1.0
Host: 84.39.39.157:8081
Accept: */*
Content-type: application/json
Content-Length: 1020
```

```
{
  "event": {
    "commonEventHeader": {
      "domain": "measurementsForVfScaling",
      "eventId": "TrafficStats_1.2.3.4",
      "eventName": "vFirewallBroadcastPackets",
      "lastEpochMicrosec": 1532032611090627,
      "priority": "Normal",
      "reportingEntityName": "fwll",
      "sequence": 2012,
      "sourceName": "vfw-vf",
      "startEpochMicrosec": 1532032600902328,
      "version": 3,
      "eventType": "HTTP request rate",
      "reportingEntityId": "No UUID available",
```

```

        "sourceId": "4087b997-db38-42c9-929b-10d223388159",
        "nfcNamingCode": "vVNF",
        "nfNamingCode": "vVNF"
    },
    "measurementsForVfScalingFields": {
        "measurementInterval": 10,
        "cpuUsageArray": [
            {
                "cpuIdentifier": "cpu1",
                "cpuIdle": 100.000000,
                "cpuUsageSystem": 0.000000,
                "cpuUsageUser": 0.000000,
                "percentUsage": 0.000000
            }
        ],
        "requestRate": 9862,
        "vNicUsageArray": [
            {
                "receivedOctetsDelta": 43731.000000,
                "receivedTotalPacketsDelta": 1016.000000,
                "transmittedOctetsDelta": 0.000000,
                "transmittedTotalPacketsDelta": 0.000000,
                "valuesAreSuspect": "true",
                "vNicIdentifier": "eth0"
            }
        ],
        "measurementsForVfScalingVersion": 2.1
    }
}

```

Another reported measurements

Example 2: Low traffic ~ 100 packets (100 packets x 1 stream)

#### VES agent reporting

```

POST /eventListener/v5 HTTP/1.1
Authorization: Basic Og==
User-Agent: libcurl-agent/1.0
Host: 84.39.39.157:8081
Accept: */*
Content-type: application/json
Content-Length: 1018

```

```

{
  "event": {
    "commonEventHeader": {
      "domain": "measurementsForVfScaling",
      "eventId": "TrafficStats_1.2.3.4",
      "eventName": "vFirewallBroadcastPackets",
      "lastEpochMicrosec": 1532032906557136,
      "priority": "Normal",
      "reportingEntityName": "fw11",
      "sequence": 2041,
      "sourceName": "vfw-vf",
      "startEpochMicrosec": 1532032896368945,
      "version": 3,
      "eventType": "HTTP request rate",
      "reportingEntityId": "No UUID available",
      "sourceId": "4087b997-db38-42c9-929b-10d223388159",
      "nfcNamingCode": "vVNF",
      "nfNamingCode": "vVNF"
    },
    "measurementsForVfScalingFields": {
      "measurementInterval": 10,
      "cpuUsageArray": [
        {
          "cpuIdentifier": "cpu1",

```

```

        "cpuIdle":100.000000,
        "cpuUsageSystem":0.000000,
        "cpuUsageUser":0.000000,
        "percentUsage":0.000000
    }
],
"requestRate":9992,
"vNicUsageArray":[
    {
        "receivedOctetsDelta":4343.000000,
        "receivedTotalPacketsDelta":102.000000,
        "transmittedOctetsDelta":0.000000,
        "transmittedTotalPacketsDelta":0.000000,
        "valuesAreSuspect":"true",
        "vNicIdentifier":"eth0"
    }
],
"measurementsForVfScalingVersion":2.1
}
}
}

```

VES agent measurements concern cpuUsage and vNicUsage of the vFW host. Measurement fields include "receivedTotalPacketsDelta" attribute that indicates the number of packets being forwarded from the packet generator to sink. The close loop is based on the value of this metric.

When receiving measurements, the VES collector (located in the DCAE) publishes these measurements on DMAap (Message router).

This published measurements matches with the example 1.

#### VES publishing measurements received from vFW host

```

POST /events/unauthenticated.VES_MEASUREMENT_OUTPUT HTTP/1.1
Content-Type: application/cambria
Accept: application/json
Content-Length: 1001
Host: 10.0.11.1:3904
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.3 (Java/1.8.0_162)
Accept-Encoding: gzip,deflate
14.977.MyPartitionKey
{
  "event":{
    "commonEventHeader":{
      "startEpochMicrosec":1532032600902328,
      "sourceId":"4087b997-db38-42c9-929b-10d223388159",
      "eventId":"TrafficStats_1.2.3.4",
      "nfcNamingCode":"vVNF",
      "reportingEntityId":"No UUID available",
      "internalHeaderFields":{
        "collectorTimeStamp":"Thu, 07 19 2018 08:36:51 GMT"
      },
      "eventType":"HTTP request rate",
      "priority":"Normal",
      "version":3,
      "reportingEntityName":"fwll",
      "sequence":2012,
      "domain":"measurementsForVfScaling",
      "lastEpochMicrosec":1532032611090627,
      "eventName":"vFirewallBroadcastPackets",
      "sourceName":"vfw-vf",
      "nfcNamingCode":"vVNF"
    },
    "measurementsForVfScalingFields":{
      "cpuUsageArray":[
        {
          "percentUsage":0,
          "cpuIdentifier":"cpu1",
          "cpuIdle":100,
          "cpuUsageSystem":0,
          "cpuUsageUser":0
        }
      ]
    }
  }
}

```

```

    }
  ],
  "measurementInterval":10,
  "requestRate":9862,
  "vNicUsageArray":[
    {
      "transmittedOctetsDelta":0,
      "receivedTotalPacketsDelta":1016,
      "vNicIdentifier":"eth0",
      "valuesAreSuspect":"true",
      "transmittedTotalPacketsDelta":0,
      "receivedOctetsDelta":43731
    }
  ],
  "measurementsForVfScalingVersion":2.1
}
}
}

```

## Closed loop execution mechanism

After performing correctly the closed loop steps for the vFW service, we can investigate interactions between ONAP components. In fact, as we have seen in the previous section, the DCAE VES collector sends measurements to DMAap. Now, the TCA microservice gets measurements from DMAap in order to analyse threshold crossing:

### Http GET mesurement from DMAap

```

GET /events/unauthenticated.VES_MEASUREMENT_OUTPUT/OpenDCAE-c12/c12-analytics-tca HTTP/1.1
Host: 10.0.11.1:3904
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.5 (Java 1.5 minimum; Java/1.8.0_131)
Accept-Encoding: gzip,deflate

```

A list of VES measurements is provided in response (The closed loop action is not yet applied)

### Http GET measurements from DMAap response

```

HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 20:36:56 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Expires: 0
Content-Type: application/json
Accept-Encoding: gzip,deflate
breadcrumbId: ID-7473fa227425-48456-1528796902051-0-11576860
User-Agent: Apache-HttpClient/4.5.5 (Java 1.5 minimum; Java/1.8.0_131)
X-CSI-Internal-WriteableRequest: true
Content-Length: 3253
Server: Jetty(9.2.z-SNAPSHOT)
[
  "{ \"event\": { \"commonEventHeader\": { \"startEpochMicrosec\": 1532032580526046, \"sourceId\": \"4087b997-db38-42c9-929b-10d223388159\", \"eventId\": \"TrafficStats_1.2.3.4\", \"nfcNamingCode\": \"vVNF\", \"reportingEntityId\": \"No UUID available\", \"internalHeaderFields\": { \"collectorTimeStamp\": \"Thu, 07 19 2018 08:36:30 GMT\", \"eventType\": \"HTTP request rate\", \"priority\": \"Normal\", \"version\": 3, \"reportingEntityName\": \"fw1\", \"sequence\": 2010, \"domain\": \"measurementsForVfScaling\", \"lastEpochMicrosec\": 1532032590713867, \"eventName\": \"vFirewallBroadcastPackets\", \"sourceName\": \"vfw-vf\", \"nfNamingCode\": \"vVNF\", \"measurementsForVfScalingFields\": { \"cpuUsageArray\": [ { \"percentUsage\": 0, \"cpuIdentifier\": \"cpul\", \"cpuIdle\": 100, \"cpuUsageSystem\": 0, \"cpuUsageUser\": 0 } ], \"measurementInterval\": 10, \"requestRate\": 8001, \"vNicUsageArray\": [ { \"transmittedOctetsDelta\": 0, \"receivedTotalPacketsDelta\": 1020, \"vNicIdentifier\": \"eth0\", \"valuesAreSuspect\": \"true\", \"transmittedTotalPacketsDelta\": 0, \"receivedOctetsDelta\": 43860 } ], \"measurementsForVfScalingVersion\": 2.1 } } }\",
  "{ \"event\": { \"commonEventHeader\": { \"startEpochMicrosec\": 1532032590713867, \"sourceId\": \"4087b997-db38-42c9-929b-10d223388159\", \"eventId\": \"TrafficStats_1.2.3.4\", \"nfcNamingCode\": \"vVNF\", \"reportingEntityId\": \"No UUID available\", \"internalHeaderFields\": { \"collectorTimeStamp\": \"Thu, 07 19 2018 08:36:40 GMT\", \"eventType\": \"HTTP request rate\", \"priority\": \"Normal\", \"version\": 3, \"

```

```
reportingEntityName\":"fwll","\sequence\":"2011","\domain\":"measurementsForVfScaling","\lastEpochMicrosec\":"1532032600902328","\eventName\":"vFirewallBroadcastPackets","\sourceName\":"vfw-vf","\nfNamingCode\":"vVNF\","\measurementsForVfScalingFields\":[{"percentUsage\":"0","\cpuIdentifier\":"cpul","\cpuIdle\":"100","\cpuUsageSystem\":"0","\cpuUsageUser\":"0}],\measurementInterval\":"10","\requestRate\":"7722","\vNicUsageArray\":[{"transmittedOctetsDelta\":"0","\receivedTotalPacketsDelta\":"1019","\vNicIdentifier\":"eth0","\valuesAreSuspect\":"true","\transmittedTotalPacketsDelta\":"0","\receivedOctetsDelta\":"43645}],\measurementsForVfScalingVersion\":"2.1}}",

{"event\":[{"commonEventHeader\":{"startEpochMicrosec\":"1532032600902328","\sourceId\":"4087b997-db38-42c9-929b-10d223388159","\eventId\":"TrafficStats_1.2.3.4","\nfcNamingCode\":"vVNF","\reportingEntityId\":"No UUID available","\internalHeaderFields\":{"collectorTimeStamp\":"Thu, 07 19 2018 08:36:51 GMT","\eventType\":"HTTP request rate","\priority\":"Normal","\version\":"3","\reportingEntityName\":"fwll","\sequence\":"2012","\domain\":"measurementsForVfScaling","\lastEpochMicrosec\":"1532032611090627","\eventName\":"vFirewallBroadcastPackets","\sourceName\":"vfw-vf","\nfNamingCode\":"vVNF\","\measurementsForVfScalingFields\":[{"percentUsage\":"0","\cpuIdentifier\":"cpul","\cpuIdle\":"100","\cpuUsageSystem\":"0","\cpuUsageUser\":"0}],\measurementInterval\":"10","\requestRate\":"9862","\vNicUsageArray\":[{"transmittedOctetsDelta\":"0","\receivedTotalPacketsDelta\":"1016","\vNicIdentifier\":"eth0","\valuesAreSuspect\":"true","\transmittedTotalPacketsDelta\":"0","\receivedOctetsDelta\":"43731}],\measurementsForVfScalingVersion\":"2.1}}}]
```

The last event in the previous response corresponds to VES measurements published by the VES collector as shown in example 1.

Based on measurements (vfw-vf in this case), TCA micro-service selects the appropriate configuration policy in order to analyse threshold crossing. As receivedTotalPacketsDelta > 700, TCA gets vnf source data from the inventory, sets CL event status to ONSET and publishes alarm to DMAap with source VNF data.

#### Http POST, TCA alarms publishing to DMAap

```
POST /events/unauthenticated.DCAE_CL_OUTPUT HTTP/1.1
Content-Length: 3511
Content-Type: application/json; charset=UTF-8
Host: 10.0.11.1:3904
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.5 (Java 1.5 minimum; Java/1.8.0_131)
Accept-Encoding: gzip,deflate
```

```
[
{
  "closedLoopAlarmStart":1532032580526046,
  "closedLoopControlName":"ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a",
  "closedLoopEventClient":"DCAE_INSTANCE_ID.dcae-tca",
  "closedLoopEventStatus":"ONSET",
  "from":"DCAE",
  "policyName":"DCAE.Config_tca-hi-lo",
  "policyScope":"DCAE",
  "policyVersion":"v0.0.1",
  "requestID":"79687ab1-ccf4-495a-b6dd-99eb8d890608",
  "target":"generic-vnf.vnf-name",
  "version":"1.0.2",
  "AAI":{"
    "generic-vnf.vnf-name":"vfw-vf",
    "generic-vnf.vnf-id":"ela9aclb-b9a0-4424-8659-69426c8bfdc8",
    "generic-vnf.vnf-type":"vfw_svc/vsp_vfwsk_1607 0",
    "generic-vnf.service-id":"8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
    "generic-vnf.prov-status":"PROV",
    "generic-vnf.orchestration-status":"Active",
    "generic-vnf.in-maint":false,
    "generic-vnf.is-closed-loop-disabled":false,
    "generic-vnf.resource-version":"1532012526533",
    "generic-vnf.model-invariant-id":"c7582c77-5a11-471c-a24f-5347ec811cdd",
    "generic-vnf.model-version-id":"db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
    "generic-vnf.model-customization-id":"5365f464-3e2e-40be-9dc2-480c916bcf10",
    "generic-vnf.nf-type":"","
    "generic-vnf.nf-function":"","
    "generic-vnf.nf-role":"","
    "generic-vnf.nf-naming-code":""
  },
  "target_type":"VNF"
```

```

},
{
  "closedLoopAlarmStart":1532032590713867,
  "closedLoopControlName":"ControlLoop-vFirewall-d0aldfc6-94f5-4fd4-a5b5-4630b438850a",
  "closedLoopEventClient":"DCAE_INSTANCE_ID.dcae-tca",
  "closedLoopEventStatus":"ONSET",
  "from":"DCAE",
  "policyName":"DCAE.Config_tca-hi-lo",
  "policyScope":"DCAE",
  "policyVersion":"v0.0.1",
  "requestID":"79cf7e2d-a39b-41fc-abe4-e0c58b79af8a",
  "target":"generic-vnf.vnf-name",
  "version":"1.0.2",
  "AAI":{
    "generic-vnf.vnf-name":"vfw-vf",
    "generic-vnf.vnf-id":"ela9aclb-b9a0-4424-8659-69426c8bfdc8",
    "generic-vnf.vnf-type":"vfw_svc/vsp_vfwsk_1607_0",
    "generic-vnf.service-id":"8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
    "generic-vnf.prov-status":"PROV",
    "generic-vnf.orchestration-status":"Active",
    "generic-vnf.in-maint":false,
    "generic-vnf.is-closed-loop-disabled":false,
    "generic-vnf.resource-version":"1532012526533",
    "generic-vnf.model-invariant-id":"c7582c77-5a11-471c-a24f-5347ec811cdd",
    "generic-vnf.model-version-id":"db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
    "generic-vnf.model-customization-id":"5365f464-3e2e-40be-9dc2-480c916bcf10",
    "generic-vnf.nf-type":"",
    "generic-vnf.nf-function":"",
    "generic-vnf.nf-role":"",
    "generic-vnf.nf-naming-code":""
  },
  "target_type":"VNF"
},
{
  "closedLoopAlarmStart":1532032600902328,
  "closedLoopControlName":"ControlLoop-vFirewall-d0aldfc6-94f5-4fd4-a5b5-4630b438850a",
  "closedLoopEventClient":"DCAE_INSTANCE_ID.dcae-tca",
  "closedLoopEventStatus":"ONSET",
  "from":"DCAE",
  "policyName":"DCAE.Config_tca-hi-lo",
  "policyScope":"DCAE",
  "policyVersion":"v0.0.1",
  "requestID":"0460e743-bf88-40e0-9e49-c356c903b0ce",
  "target":"generic-vnf.vnf-name",
  "version":"1.0.2",
  "AAI":{
    "generic-vnf.vnf-name":"vfw-vf",
    "generic-vnf.vnf-id":"ela9aclb-b9a0-4424-8659-69426c8bfdc8",
    "generic-vnf.vnf-type":"vfw_svc/vsp_vfwsk_1607_0",
    "generic-vnf.service-id":"8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
    "generic-vnf.prov-status":"PROV",
    "generic-vnf.orchestration-status":"Active",
    "generic-vnf.in-maint":false,
    "generic-vnf.is-closed-loop-disabled":false,
    "generic-vnf.resource-version":"1532012526533",
    "generic-vnf.model-invariant-id":"c7582c77-5a11-471c-a24f-5347ec811cdd",
    "generic-vnf.model-version-id":"db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
    "generic-vnf.model-customization-id":"5365f464-3e2e-40be-9dc2-480c916bcf10",
    "generic-vnf.nf-type":"",
    "generic-vnf.nf-function":"",
    "generic-vnf.nf-role":"",
    "generic-vnf.nf-naming-code":""
  },
  "target_type":"VNF"
}
]

```

The policy component requests DMAap in order to get published alarms: GET /events/unauthenticated.DCAE\_CL\_OUTPUT/dcae.policy.shared/drools?timeout=15000&limit=100 HTTP/1.1



## GET response

```
HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 20:41:36 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Expires: 0
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip,deflate
breadcrumbId: ID-7473fa227425-48456-1528796902051-0-11578575
limit: 100
timeout: 15000
User-Agent: Apache-HttpClient/4.5.5 (Java/1.8.0_171)
X-CSI-Internal-WriteableRequest: true
Content-Length: 1281
Server: Jetty(9.2.z-SNAPSHOT)
```

```
[
  {
    "closedLoopEventClient\\":\\"DCAE_INSTANCE_ID.dcae-tca\\",\\"policyVersion\\":\\"v0.0.1\\",\\"policyName\\":\\"DCAE.Config_tca-hi-lo\\",\\"policyScope\\":\\"DCAE\\",\\"target_type\\":\\"VNF\\",\\"AAI\\":{
      "generic-vnf.resource-version\\":\\"1532012526533\\",\\"generic-vnf.nf-role\\":\\"\\",\\"generic-vnf.prov-status\\":\\"PROV\\",\\"generic-vnf.orchestration-status\\":\\"Active\\",\\"generic-vnf.is-closed-loop-disabled\\":false,\\"generic-vnf.service-id\\":\\"8eba27fe-89eb-4c32-9ee6-aa5428c7d962\\",\\"generic-vnf.in-maint\\":false,\\"generic-vnf.nf-type\\":\\"\\",\\"generic-vnf.nf-naming-code\\":\\"\\",\\"generic-vnf.vnf-name\\":\\"vfw-vf\\",\\"generic-vnf.model-version-id\\":\\"db0fbb27-fd0d-40eb-8491-0f4a48adaefe\\",\\"generic-vnf.model-customization-id\\":\\"5365f464-3e2e-40be-9dc2-480c916bcf10\\",\\"generic-vnf.nf-function\\":\\"\\",\\"generic-vnf.vnf-type\\":\\"vfw_svc/vsp_vfwsk_1607 0\\",\\"generic-vnf.model-invariant-id\\":\\"c7582c77-5a11-471c-a24f-5347ec811cdd\\",\\"generic-vnf.vnf-id\\":\\"e1a9ac1b-b9a0-4424-8659-69426c8bfdc8\\"},
    "closedLoopAlarmStart\\":1532032875991084,\\"closedLoopEventStatus\\":\\"ONSET\\",
    "closedLoopControlName\\":\\"ControlLoop-vFirewall-d0aldfc6-94f5-4fd4-a5b5-4630b438850a\\",\\"version\\":\\"1.0.2\\",\\"target\\":\\"generic-vnf.vnf-name\\",\\"requestID\\":\\"1a6f980d-4eed-432d-ba58-83bd34659f64\\",\\"from\\":\\"DCAE\\"}
  ]
```

When detecting the published alarm, the policy begins to send notifications to DMAap with different stages of policy processing (POLICY-CL-MGT topic).

It first acknowledges the onset alarm by sending an ACTIVE notification.

## POST POLICY-CL-MGT: ACTIVE notification

```
POST /events/POLICY-CL-MGT HTTP/1.1
Content-Type: application/cambria
Accept: application/json
Content-Length: 1492
Host: vml.mr.simplesdemo.onap.org:3904
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.5 (Java/1.8.0_171)
Accept-Encoding: gzip,deflate
```

```
36.1447.9fb731ea-6449-4194-b0ae-618e50a87125{
  "AAI": {
    "generic-vnf.resource-version": "1532012526533",
    "generic-vnf.nf-role": "",
    "generic-vnf.prov-status": "PROV",
    "generic-vnf.orchestration-status": "Active",
    "generic-vnf.is-closed-loop-disabled": "false",
    "generic-vnf.service-id": "8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
    "generic-vnf.in-maint": "false",
    "generic-vnf.nf-type": "",
    "generic-vnf.nf-naming-code": "",
    "generic-vnf.vnf-name": "vfw-vf",
    "generic-vnf.model-version-id": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
    "generic-vnf.model-customization-id": "5365f464-3e2e-40be-9dc2-480c916bcf10",
    "generic-vnf.nf-function": "",
    "generic-vnf.vnf-type": "vfw_svc/vsp_vfwsk_1607 0",
    "generic-vnf.model-invariant-id": "c7582c77-5a11-471c-a24f-5347ec811cdd",
    "generic-vnf.vnf-id": "e1a9ac1b-b9a0-4424-8659-69426c8bfdc8"
  },
}
```

```

"closedLoopAlarmStart": 1532032875991084,
"closedLoopControlName": "ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a",
"version": "1.0.2",
"requestId": "1a6f980d-4eed-432d-ba58-83bd34659f64",
"closedLoopEventClient": "DCAE_INSTANCE_ID.dcae-tca",
"targetType": "VNF",
"target": "generic-vnf.vnf-name",
"from": "pdp-0001-controller=controlloop:amsterdam",
"policyScope": "com",
"policyName": "com.Config_BRMS_Param_BRMSParamvFirewall.EVENT",
"policyVersion": "5",
"notification": "ACTIVE",
"notificationTime": "2018-07-19 20:41:46.486000+00:00",
"history": []
}

```

In the previous notification, the onset is enriched with inventory data of the vFWSNK vnf. The policy action does apply on this vnf. So, policy queries the inventory to get the targeted entity by the closed loop.

#### Policy request to A&AI to get CL targeted entity

```

[Session org.onap.policy-engine.drools.amsterdam:policy-amsterdam-rules:3.0.0:closedloop-amsterdam]
[OUT|AAI|https://aai.api.simpLEDemo.onap.org:8443/aai/search/named-query|]
{
  "query-parameters": {
    "named-query": {
      "named-query-uuid": "a93ac487-409c-4e8c-9e5f-334ae8f99087"
    }
  },
  "instance-filters": {
    "instance-filter": [
      {
        "generic-vnf": {
          "vnf-id": "e1a9ac1b-b9a0-4424-8659-69426c8bfdc8"
        }
      }
    ]
  }
}

```

#### A&AI response

```

[Session org.onap.policy-engine.drools.amsterdam:policy-amsterdam-rules:3.0.0:closedloop-amsterdam]
[IN|AAI|https://aai.api.simpLEDemo.onap.org:8443/aai/search/named-query|]
{
  "inventory-response-item": [
    {
      "model-name": "vsp_vfwsnk_1607",
      "generic-vnf": {
        "vnf-id": "e1a9ac1b-b9a0-4424-8659-69426c8bfdc8",
        "vnf-name": "vfw-vf",
        "vnf-type": "vfw_svc/vsp_vfwsnk_1607 0",
        "service-id": "8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
        "prov-status": "PROV",
        "orchestration-status": "Active",
        "in-maint": false,
        "is-closed-loop-disabled": false,
        "resource-version": "1532012526533",
        "model-invariant-id": "c7582c77-5a11-471c-a24f-5347ec811cdd",
        "model-version-id": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
        "model-customization-id": "5365f464-3e2e-40be-9dc2-480c916bcf10",
        "nf-type": "",
        "nf-function": "",
        "nf-role": "",
        "nf-naming-code": ""
      }
    }
  ],

```

```
"extra-properties":{
},
"inventory-response-items":{
  "inventory-response-item":[
    {
      "model-name":"vfw_svc",
      "service-instance":{
        "service-instance-id":"d39d0833-ea8a-45be-8863-76715697cdb3",
        "service-instance-name":"vfw_svc_1607",
        "environment-context":"General_Revenue-Bearing",
        "workload-context":"Production",
        "model-invariant-id":"46c0d35f-4a34-4f5b-9ee3-fd65f6f74ac2",
        "model-version-id":"84909c11-1a62-47b3-9aed-59c2ec5c891d",
        "resource-version":"1532011795844",
        "orchestration-status":"Active"
      },
      "extra-properties":{
    },
    "inventory-response-items":{
      "inventory-response-item":[
        {
          "model-name":"vsp_vpg_1607",
          "generic-vnf":{
            "vnf-id":"belc20fe-2646-45a1-b85d-80c4b00832a7",
            "vnf-name":"vpg-vf",
            "vnf-type":"vfw_svc/vsp_vpg_1607_0",
            "service-id":"8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
            "prov-status":"PROV",
            "orchestration-status":"Active",
            "in-maint":false,
            "is-closed-loop-disabled":false,
            "resource-version":"1532012545029",
            "model-invariant-id":"bf2a3891-148d-442f-a136-10782e48a418",
            "model-version-id":"89468b54-0e2c-4307-aac4-4661a53c0aac",
            "model-customization-id":"0ecc8cf9-19d8-41d7-9048-f22da4172d58",
            "nf-type":"","",
            "nf-function":"","",
            "nf-role":"","",
            "nf-naming-code":""
          },
          "extra-properties":{
        }
      ]
    }
  ],
}
}
```

The policy reacts to the published alarm by posting the CL operation to DMAap:

### POST CL operation to DMAap

```
POST /events/APPC-CL HTTP/1.1
Content-Type: application/cambria
Accept: application/json
Content-Length: 810
Host: vml.mr.simplifiedemo.onap.org:3904
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.5 (Java/1.8.0_171)
Accept-Encoding: gzip,deflate

36.766.1cb70869-6a79-4d28-ba72-fe13cff1da14{
```

```

"CommonHeader": {
  "TimeStamp": 1532032906679,
  "APIVer": "1.01",
  "RequestID": "1a6f980d-4eed-432d-ba58-83bd34659f64",
  "SubRequestID": "1",
  "RequestTrack": [],
  "Flags": []
},
"Action": "ModifyConfig",
"Payload": {
  "generic-vnf.vnf-id": "be1c20fe-2646-45a1-b85d-80c4b00832a7",
  "pg-streams": {
    "pg-stream": [
      {
        "id": "fw_udp1",
        "is-enabled": "true"
      },
      {
        "id": "fw_udp2",
        "is-enabled": "true"
      },
      {
        "id": "fw_udp3",
        "is-enabled": "true"
      },
      {
        "id": "fw_udp4",
        "is-enabled": "true"
      },
      {
        "id": "fw_udp5",
        "is-enabled": "true"
      }
    ]
  }
}
}

```

At this stage, the policy posts a set of notifications related to the operation processing (POLICY-CL-MGT topic). In fact, the policy engine queries the GUARD for the ability to apply the operation of the CL. If permitted, it locks the target vnf (vPKG) so that no other similar operation could target the same vnf. It specifies also the actor responsible for performing the operation and the target vnf for the closed loop.

#### Policy operation processing notifications

```

POST /events/POLICY-CL-MGT HTTP/1.1
Content-Type: application/cambria
Accept: application/json
Content-Length: 4765
Host: vml.mr.simplesdemo.onap.org:3904
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.5 (Java/1.8.0_171)
Accept-Encoding: gzip,deflate

```

```

36.1530.9fb731ea-6449-4194-b0ae-618e50a87125{
  "AAI": {
    "generic-vnf.resource-version": "1532012526533",
    "generic-vnf.nf-role": "",
    "generic-vnf.prov-status": "PROV",
    "generic-vnf.orchestration-status": "Active",
    "generic-vnf.is-closed-loop-disabled": "false",
    "generic-vnf.service-id": "8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
    "generic-vnf.in-maint": "false",
    "generic-vnf.nf-type": "",
    "generic-vnf.nf-naming-code": "",
    "generic-vnf.vnf-name": "vfw-vf",
    "generic-vnf.model-version-id": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
    "generic-vnf.model-customization-id": "5365f464-3e2e-40be-9dc2-480c916bcf10",
    "generic-vnf.nf-function": "",
    "generic-vnf.vnf-type": "vfw_svc/vsp_vfwskn_1607 0",
  }
}

```

```

    "generic-vnf.model-invariant-id": "c7582c77-5a11-471c-a24f-5347ec811cdd",
    "generic-vnf.vnf-id": "e1a9ac1b-b9a0-4424-8659-69426c8bfdc8"
  },
  "closedLoopAlarmStart": 1532032875991084,
  "closedLoopControlName": "ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a",
  "version": "1.0.2",
  "requestId": "1a6f980d-4eed-432d-ba58-83bd34659f64",
  "closedLoopEventClient": "DCAE_INSTANCE_ID.dcae-tca",
  "targetType": "VNF",
  "target": "generic-vnf.vnf-name",
  "from": "policy:amsterdam",
  "policyScope": "com",
  "policyName": "com.Config_BRMS_Param_BRMSParamvFirewall.EVENT.MANAGER.OPERATION.LOCKED.
GUARD_NOT_YET_QUERIED",
  "policyVersion": "5",
  "notification": "OPERATION",
  "message": "Sending guard query for APPC ModifyConfig",
  "notificationTime": "2018-07-19 20:41:46.671000+00:00",
  "history": []
}
36.1495.9fb731ea-6449-4194-b0ae-618e50a87125{
  "AAI": {
    "generic-vnf.resource-version": "1532012526533",
    "generic-vnf.nf-role": "",
    "generic-vnf.prov-status": "PROV",
    "generic-vnf.orchestration-status": "Active",
    "generic-vnf.is-closed-loop-disabled": "false",
    "generic-vnf.service-id": "8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
    "generic-vnf.in-maint": "false",
    "generic-vnf.nf-type": "",
    "generic-vnf.nf-naming-code": "",
    "generic-vnf.vnf-name": "vfw-vf",
    "generic-vnf.model-version-id": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
    "generic-vnf.model-customization-id": "5365f464-3e2e-40be-9dc2-480c916bcf10",
    "generic-vnf.nf-function": "",
    "generic-vnf.vnf-type": "vfw_svc/vsp_vfwsk_1607 0",
    "generic-vnf.model-invariant-id": "c7582c77-5a11-471c-a24f-5347ec811cdd",
    "generic-vnf.vnf-id": "e1a9ac1b-b9a0-4424-8659-69426c8bfdc8"
  },
  "closedLoopAlarmStart": 1532032875991084,
  "closedLoopControlName": "ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a",
  "version": "1.0.2",
  "requestId": "1a6f980d-4eed-432d-ba58-83bd34659f64",
  "closedLoopEventClient": "DCAE_INSTANCE_ID.dcae-tca",
  "targetType": "VNF",
  "target": "generic-vnf.vnf-name",
  "from": "policy:amsterdam",
  "policyScope": "com",
  "policyName": "com.Config_BRMS_Param_BRMSParamvFirewall.GUARD.RESPONSE",
  "policyVersion": "5",
  "notification": "OPERATION",
  "message": "Guard result for APPC ModifyConfig is Permit",
  "notificationTime": "2018-07-19 20:41:46.676000+00:00",
  "history": []
}
36.1605.9fb731ea-6449-4194-b0ae-618e50a87125{
  "AAI": {
    "generic-vnf.resource-version": "1532012526533",
    "generic-vnf.nf-role": "",
    "generic-vnf.prov-status": "PROV",
    "generic-vnf.orchestration-status": "Active",
    "generic-vnf.is-closed-loop-disabled": "false",
    "generic-vnf.service-id": "8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
    "generic-vnf.in-maint": "false",
    "generic-vnf.nf-type": "",
    "generic-vnf.nf-naming-code": "",
    "generic-vnf.vnf-name": "vfw-vf",
    "generic-vnf.model-version-id": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
    "generic-vnf.model-customization-id": "5365f464-3e2e-40be-9dc2-480c916bcf10",
    "generic-vnf.nf-function": "",
    "generic-vnf.vnf-type": "vfw_svc/vsp_vfwsk_1607 0",

```

```

    "generic-vnf.model-invariant-id": "c7582c77-5a11-471c-a24f-5347ec811cdd",
    "generic-vnf.vnf-id": "e1a9ac1b-b9a0-4424-8659-69426c8bfdc8"
  },
  "closedLoopAlarmStart": 1532032875991084,
  "closedLoopControlName": "ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a",
  "version": "1.0.2",
  "requestId": "1a6f980d-4eed-432d-ba58-83bd34659f64",
  "closedLoopEventClient": "DCAE_INSTANCE_ID.dcae-tca",
  "targetType": "VNF",
  "target": "generic-vnf.vnf-name",
  "from": "policy:amsterdam",
  "policyScope": "com",
  "policyName": "com.Config_BRMS_Param_BRMSParamvFirewall.EVENT.MANAGER.OPERATION.LOCKED.GUARD_PERMITTED",
  "policyVersion": "5",
  "notification": "OPERATION",
  "message": "actor=APPC,operation=ModifyConfig,target=Target [type=VNF, resourceID=bf2a3891-148d-442f-a136-10782e48a418],subRequestId=1",
  "notificationTime": "2018-07-19 20:41:46.679000+00:00",
  "history": []
}

```

The APPC indicates that operation is accepted through POST to DMAap (APPC-CL topic):

#### APPC POST "operation accepted"

```

POST /events/APPC-CL HTTP/1.1
Content-Type: application/cambria
Content-Length: 643
Host: 10.0.11.1:3904
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.3 (Java 1.5 minimum; Java/1.8.0_171)
Accept-Encoding: gzip,deflate

{
  "Status":{
    "Value":"ACCEPTED",
    "Code":"100"
  },
  "CommonHeader":{
    "OriginatorID":null,
    "SubrequestID":null,
    "RequestID":"1a6f980d-4eed-432d-ba58-83bd34659f64",
    "APIver":"1.01",
    "TimeStamp":"1532032906679"
  },
  "Payload":{
    "generic-vnf.vnf-id":"belc20fe-2646-45a1-b85d-80c4b00832a7",
    "pg-streams":{"\\\\"pg-streams\\\\"": {\\\\"pg-stream\\\\"":[\\\\"id\\\\":\\\\"fw_udp1\\\\" , \\\\"is-enabled\\\\" : \\\\"true\\\\"},{\\\\"id\\\\":\\\\"fw_udp2\\\\" , \\\\"is-enabled\\\\" : \\\\"true\\\\"},{\\\\"id\\\\":\\\\"fw_udp3\\\\" , \\\\"is-enabled\\\\" : \\\\"true\\\\"},{\\\\"id\\\\":\\\\"fw_udp4\\\\" , \\\\"is-enabled\\\\" : \\\\"true\\\\"},{\\\\"id\\\\":\\\\"fw_udp5\\\\" , \\\\"is-enabled\\\\" : \\\\"true\\\\"}]]}
  }
}

```

APPC applies the CL operation:

#### APPC action

```

PUT /restconf/config/network-topology:network-topology/topology/topology-netconf/node/belc20fe-2646-45a1-b85d-80c4b00832a7/yang-ext:mount/sample-plugin:sample-plugin/pg-streams/ HTTP/1.1
Content-type: application/json
Content-Length: 221
Host: 10.0.2.1:8282
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.3 (Java/1.8.0_171)
Accept-Encoding: gzip,deflate
Authorization: Basic YWRtaW46YWRtaW4=

```

```
{
  "pg-streams": {
    "pg-stream": [
      {
        "id": "fw_udp1",
        "is-enabled": "true"
      },
      {
        "id": "fw_udp2",
        "is-enabled": "true"
      },
      {
        "id": "fw_udp3",
        "is-enabled": "true"
      },
      {
        "id": "fw_udp4",
        "is-enabled": "true"
      },
      {
        "id": "fw_udp5",
        "is-enabled": "true"
      }
    ]
  }
}
```

This operation corresponds to the execution of the latest action (ModifyConfig) of **topology-operation-all Directed Graph**



APPC publishes the status of the operation:

#### Operation status published by APPC

```
POST /events/APPC-CL HTTP/1.1
Content-Type: application/cambria
Content-Length: 642
Host: 10.0.11.1:3904
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.3 (Java 1.5 minimum; Java/1.8.0_171)
Accept-Encoding: gzip,deflate
```

```
0.636.{
  "Status":{
```

```
    "Value": "SUCCESS",
    "Code": "400"
  },
  "CommonHeader": {
    "OriginatorID": null,
    "SubrequestID": null,
    "RequestID": "1a6f980d-4eed-432d-ba58-83bd34659f64",
    "APIVer": "1.01",
    "TimeStamp": "1532032906679"
  },
  "Payload": {
    "generic-vnf.vnf-id": "belc20fe-2646-45a1-b85d-80c4b00832a7",
    "pg-streams": "{\\\\"pg-streams\\\\": [{\\\\"pg-stream\\\\":[{\\\\"id\\\\":\\\\"fw_udp1\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}], {\\\\"id\\\\":\\\\"fw_udp2\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}], {\\\\"id\\\\":\\\\"fw_udp3\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}], {\\\\"id\\\\":\\\\"fw_udp4\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}], {\\\\"id\\\\":\\\\"fw_udp5\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}]]}"
  }
}
```

Policy gets the last events published on APPC-CL topic: GET /events/APPC-CL/e14c7c74-be39-492f-b3f4-26b0a0748155/drools?  
timeout=15000&limit=100 HTTP/1.1

#### GET response

```
HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 20:41:47 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Expires: 0
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip,deflate
breadcrumbId: ID-7473fa227425-48456-1528796902051-0-11578639
limit: 100
timeout: 15000
User-Agent: Apache-HttpClient/4.5.5 (Java/1.8.0_171)
X-CSI-Internal-WriteableRequest: true
Content-Length: 854
Server: Jetty(9.2.z-SNAPSHOT)
[
  {
    "Status": {
      "Value": "SUCCESS",
      "Code": "400"
    },
    "CommonHeader": {
      "OriginatorID": null,
      "SubrequestID": null,
      "RequestID": "1a6f980d-4eed-432d-ba58-83bd34659f64",
      "APIVer": "1.01",
      "TimeStamp": "1532032906679"
    },
    "Payload": {
      "generic-vnf.vnf-id": "belc20fe-2646-45a1-b85d-80c4b00832a7",
      "pg-streams": "{\\\\"pg-streams\\\\": [{\\\\"pg-stream\\\\":[{\\\\"id\\\\":\\\\"fw_udp1\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}], {\\\\"id\\\\":\\\\"fw_udp2\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}], {\\\\"id\\\\":\\\\"fw_udp3\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}], {\\\\"id\\\\":\\\\"fw_udp4\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}], {\\\\"id\\\\":\\\\"fw_udp5\\\\"}, {\\\\"is-enabled\\\\":\\\\"true\\\\"}]]}"
    }
  }
]
```

Policy publishes notifications that indicate the final status of the operation

#### Notifications on final operation status

```
POST /events/POLICY-CL-MGT HTTP/1.1
Content-Type: application/cambria
Accept: application/json
Content-Length: 3592
Host: vml.mr.simplesdemo.onap.org:3904
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.5 (Java/1.8.0_171)
Accept-Encoding: gzip,deflate

36.1647.9fb731ea-6449-4194-b0ae-618e50a87125{
  "AAI": {
    "generic-vnf.resource-version": "1532012526533",
```



```

    "generic-vnf.nf-role": "",
    "generic-vnf.prov-status": "PROV",
    "generic-vnf.orchestration-status": "Active",
    "generic-vnf.is-closed-loop-disabled": "false",
    "generic-vnf.service-id": "8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
    "generic-vnf.in-maint": "false",
    "generic-vnf.nf-type": "",
    "generic-vnf.nf-naming-code": "",
    "generic-vnf.vnf-name": "vfw-vf",
    "generic-vnf.model-version-id": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
    "generic-vnf.model-customization-id": "5365f464-3e2e-40be-9dc2-480c916bcf10",
    "generic-vnf.nf-function": "",
    "generic-vnf.vnf-type": "vfw_svc/vsp_vfwsk_1607 0",
    "generic-vnf.model-invariant-id": "c7582c77-5a11-471c-a24f-5347ec811cdd",
    "generic-vnf.vnf-id": "ela9aclb-b9a0-4424-8659-69426c8bfdc8"
  },
  "closedLoopAlarmStart": 1532032875991084,
  "closedLoopControlName": "ControlLoop-vFirewall-d0aldfc6-94f5-4fd4-a5b5-4630b438850a",
  "version": "1.0.2",
  "requestId": "1a6f980d-4eed-432d-ba58-83bd34659f64",
  "closedLoopEventClient": "DCAE_INSTANCE_ID.dcae-tca",
  "targetType": "VNF",
  "target": "generic-vnf.vnf-name",
  "from": "policy:amsterdam",
  "policyScope": "com",
  "policyName": "com.Config_BRMS_Param_BRMSParamvFirewall.APPC.RESPONSE",
  "policyVersion": "5",
  "notification": "OPERATION: SUCCESS",
  "message": "actor=APPC,operation=ModifyConfig,target=Target [type=VNF, resourceID=bf2a3891-148d-442f-a136-10782e48a418],start=2018-07-19T20:41:46.678Z,end=null,subRequestId=1,outcome=null,message=null",
  "notificationTime": "2018-07-19 20:41:47.032000+00:00",
  "history": []
}
36.1855.9fb731ea-6449-4194-b0ae-618e50a87125{
  "AAI": {
    "generic-vnf.resource-version": "1532012526533",
    "generic-vnf.nf-role": "",
    "generic-vnf.prov-status": "PROV",
    "generic-vnf.orchestration-status": "Active",
    "generic-vnf.is-closed-loop-disabled": "false",
    "generic-vnf.service-id": "8eba27fe-89eb-4c32-9ee6-aa5428c7d962",
    "generic-vnf.in-maint": "false",
    "generic-vnf.nf-type": "",
    "generic-vnf.nf-naming-code": "",
    "generic-vnf.vnf-name": "vfw-vf",
    "generic-vnf.model-version-id": "db0fbb27-fd0d-40eb-8491-0f4a48adaefe",
    "generic-vnf.model-customization-id": "5365f464-3e2e-40be-9dc2-480c916bcf10",
    "generic-vnf.nf-function": "",
    "generic-vnf.vnf-type": "vfw_svc/vsp_vfwsk_1607 0",
    "generic-vnf.model-invariant-id": "c7582c77-5a11-471c-a24f-5347ec811cdd",
    "generic-vnf.vnf-id": "ela9aclb-b9a0-4424-8659-69426c8bfdc8"
  },
  "closedLoopAlarmStart": 1532032875991084,
  "closedLoopControlName": "ControlLoop-vFirewall-d0aldfc6-94f5-4fd4-a5b5-4630b438850a",
  "version": "1.0.2",
  "requestId": "1a6f980d-4eed-432d-ba58-83bd34659f64",
  "closedLoopEventClient": "DCAE_INSTANCE_ID.dcae-tca",
  "targetType": "VNF",
  "target": "generic-vnf.vnf-name",
  "from": "policy:amsterdam",
  "policyScope": "com",
  "policyName": "com.Config_BRMS_Param_BRMSParamvFirewall.EVENT.MANAGER",
  "policyVersion": "5",
  "notification": "FINAL: SUCCESS",
  "message": "Exception in processing closed loop",
  "notificationTime": "2018-07-19 20:41:47.036000+00:00",
  "history": [
    {
      "actor": "APPC",
      "operation": "ModifyConfig",
      "target": "Target [type=VNF, resourceID=bf2a3891-148d-442f-a136-10782e48a418]",
    }
  ]
}

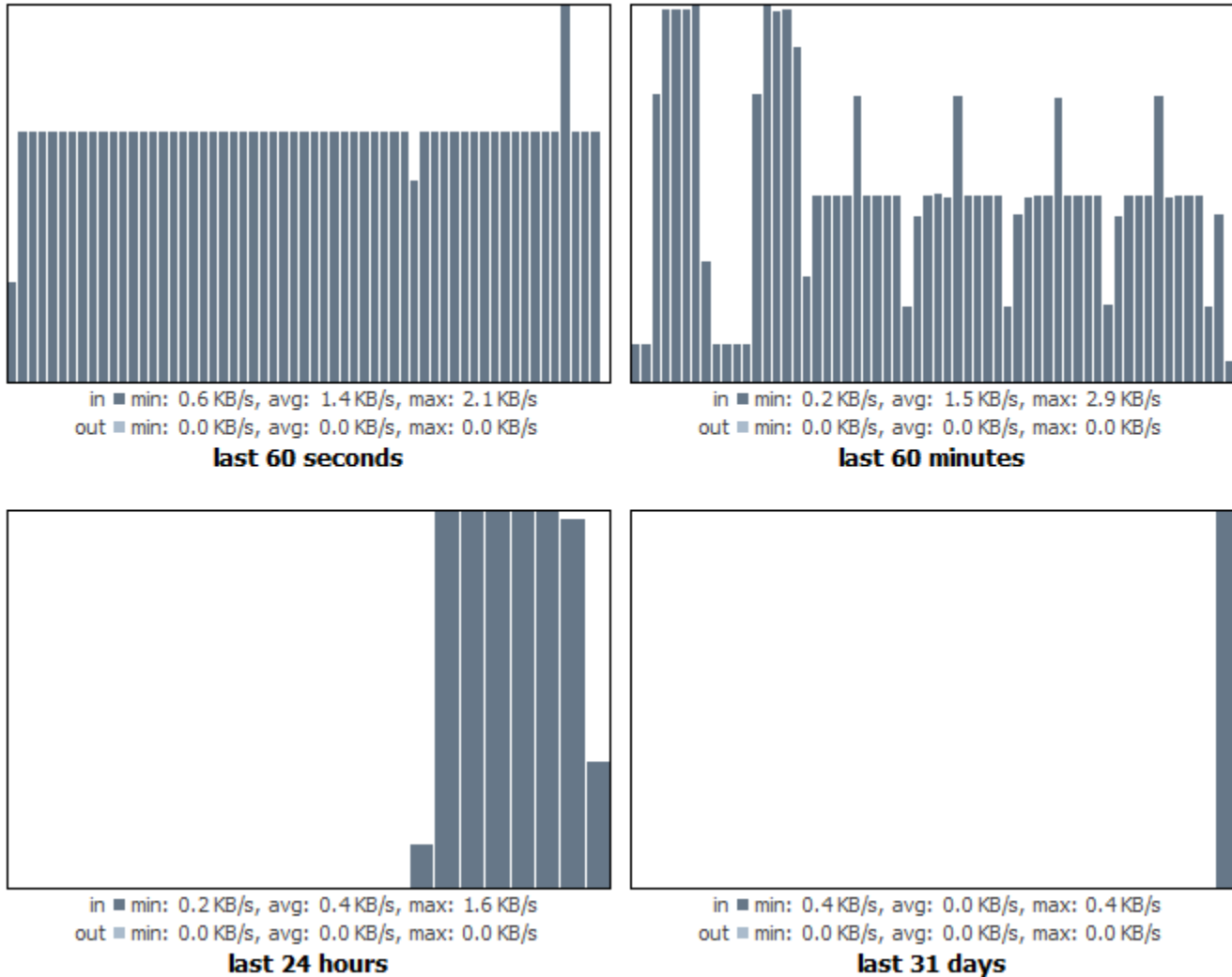
```

```

    "start": 1532032906678,
    "end": 1532032907035,
    "subRequestId": "1",
    "outcome": "Failure_Exception",
    "message": "current operation is not null (an operation is already running)"
  }
]
}

```

At this stage, we can check sink graphs to see if traffic volume has been regulated:



## Analysis

The vFW closed loop is based on VES (VNF Event Stream) reported measurements that TCA analyses and publishes related alarms to DMAap. The policy has the responsibility for triggering ModifyConfig action that APPC executes on the target vnf (vPKG). All closed loop interactions are relying on the Message router (DMAap) by publishing topics and subscribing to them.

The policy component has the biggest responsibility in this mechanism. It takes decision for applying CL actions, requests the inventory for the target vnf and selects CL actor, and follows the execution of the action by posting CL operation notification.

On the APPC side, the CL the action corresponds to the execution of a Directed graph.

