July, 2018

vFWCL Analysis
ONAP Beijing

Badr Eddine Aouled Yahia

Orange

vFW service composition

The vFW service is composed of the following components:

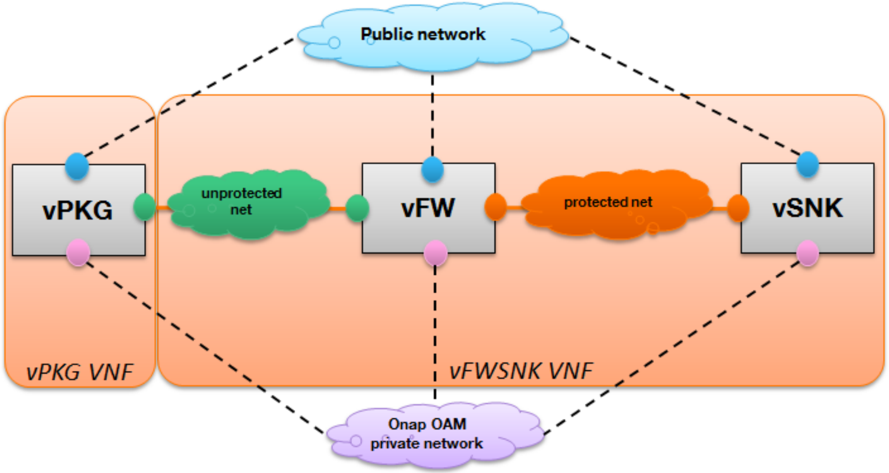
vFWSNK VNF:

– Packet generator: **sends periodically different volumes of traffic to the sink through the firewall**

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

vPKG VNF:

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



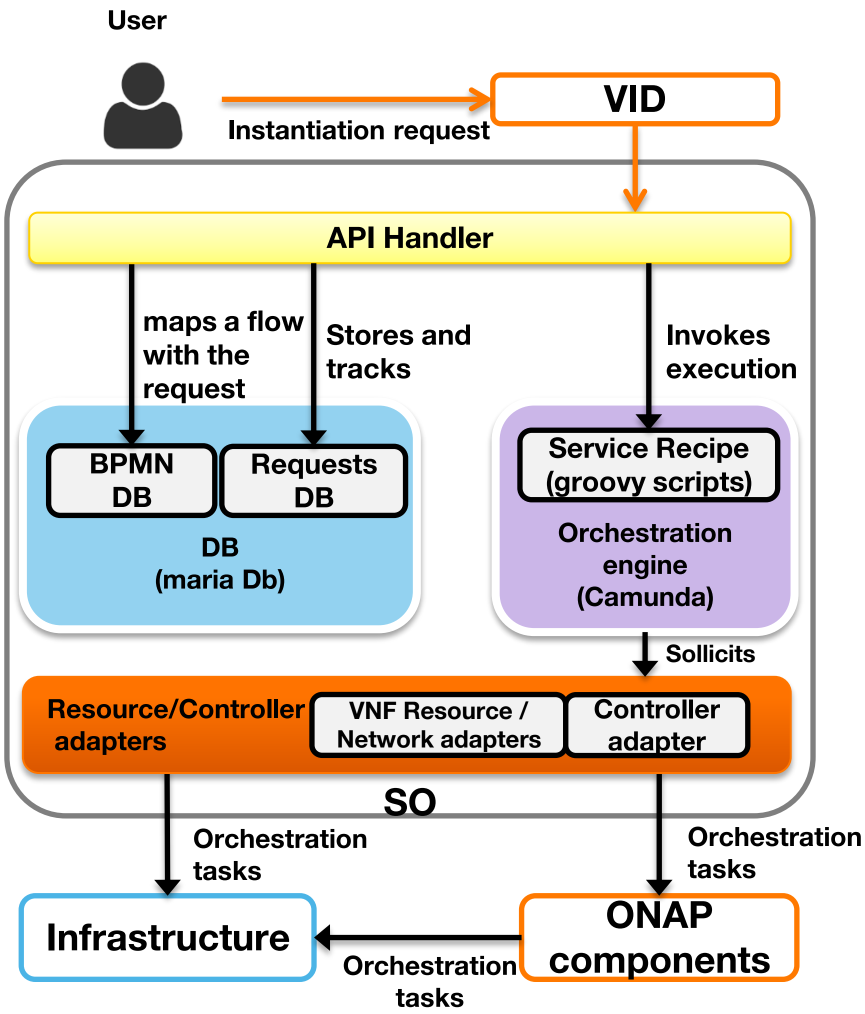
vFW deployment

The vFW service deployment includes the following steps which correspond to VID requests to trigger instantiation in SO:

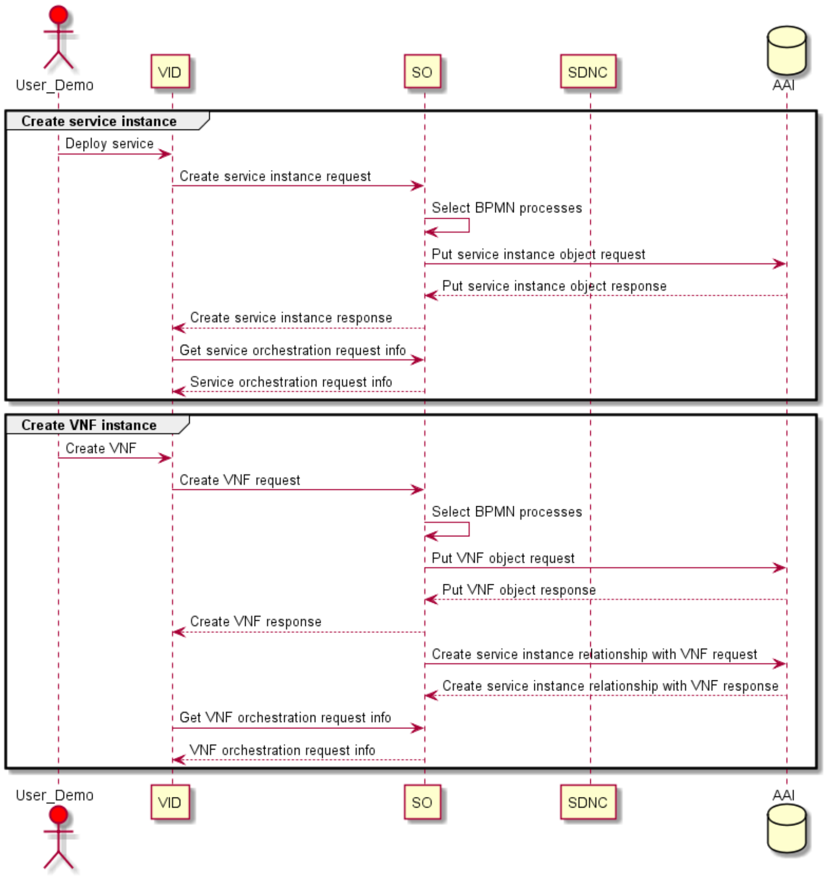
1. Create service instance: Create service instance object in the inventory
2. Create VNF instance: Create vnf instance object and relationship in the inventory
3. Create VF module: Create vf-module in the inventory and deploy the stack

These steps correspond to HTTP requests sent by the VID to SO

Orchestration engine executes BPMN workflows to create instances



vFW deployment
Flow diagram



User_Demo VID SO SDNC AAI

Create service instance

Deploy service

Create service instance request

Select BPMN processes

Put service instance object request

Put service instance object response

Create service instance response

Get service orchestration request info

Service orchestration request info

Create VNF instance

Create VNF

Create VNF request

Select BPMN processes

Put VNF object request

Put VNF object response

Create VNF response

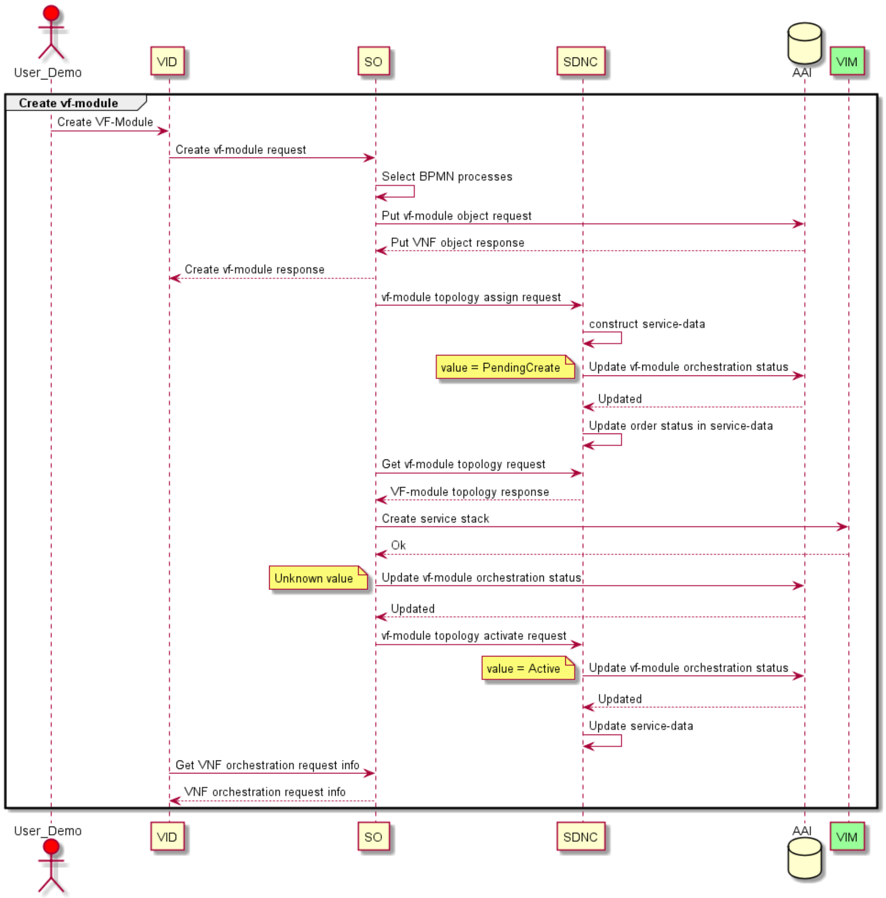
Create service instance relationship with VNF request

Create service instance relationship with VNF response

Get VNF orchestration request info

VNF orchestration request info

User_Demo VID SO SDNC AAI



User_Demo VID SO SDNC AAI VIM

Create vf-module

Create VF-Module

Create vf-module request

Select BPMN processes

Put vf-module object request

Put VNF object response

Create vf-module response

vf-module topology assign request

construct service-data

value = PendingCreate

Update vf-module orchestration status

Updated

Update order status in service-data

Get vf-module topology request

VF-module topology response

Create service stack

Ok

Unknown value

Update vf-module orchestration status

Updated

vf-module topology activate request

value = Active

Update vf-module orchestration status

Updated

Update service-data

Get VNF orchestration request info

VNF orchestration request info

User_Demo VID SO SDNC AAI VIM

vFW deployment
Analysis

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

Instantiation is based on BPMN workflow that orchestration engine executes.

Service and vnf instantiation correspond 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 based on heat templates.

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

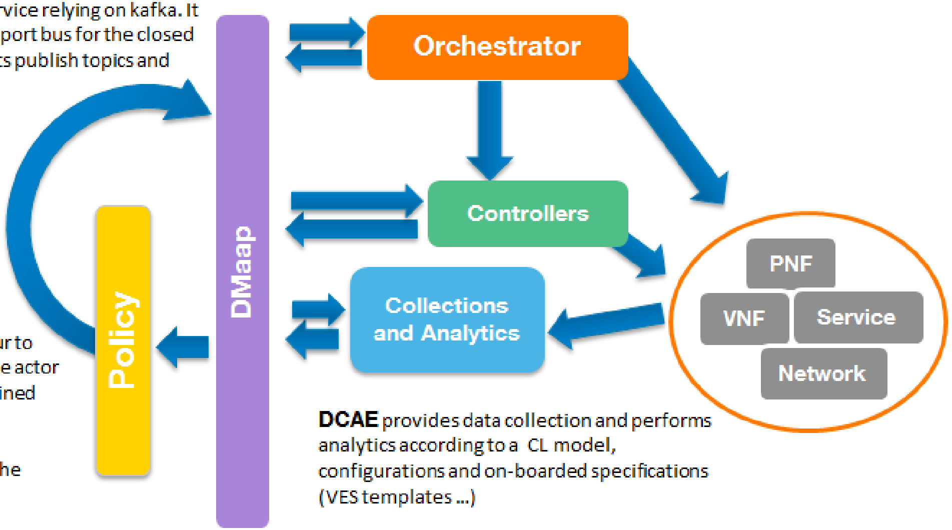
The vFW close loop scenario consists in applying policy rules that aims to re-adjust the traffic volume when high threshold (700 packets) or low threshold (300 packets) are crossed.

DCAE collects events from the vFW, applies analytics (TCA) and publishes events to DMaap.

When detecting the triggering event, the policy engine executes the operational policy (ModifyConfig) via APP-C component that adjusts the traffic volume to 500 packet per 10 seconds.

Orchestration may be requested to enforce actions based on policy definition. The SO could delegate the execution of several tasks to controllers according to what is defined in its orchestration workflows.

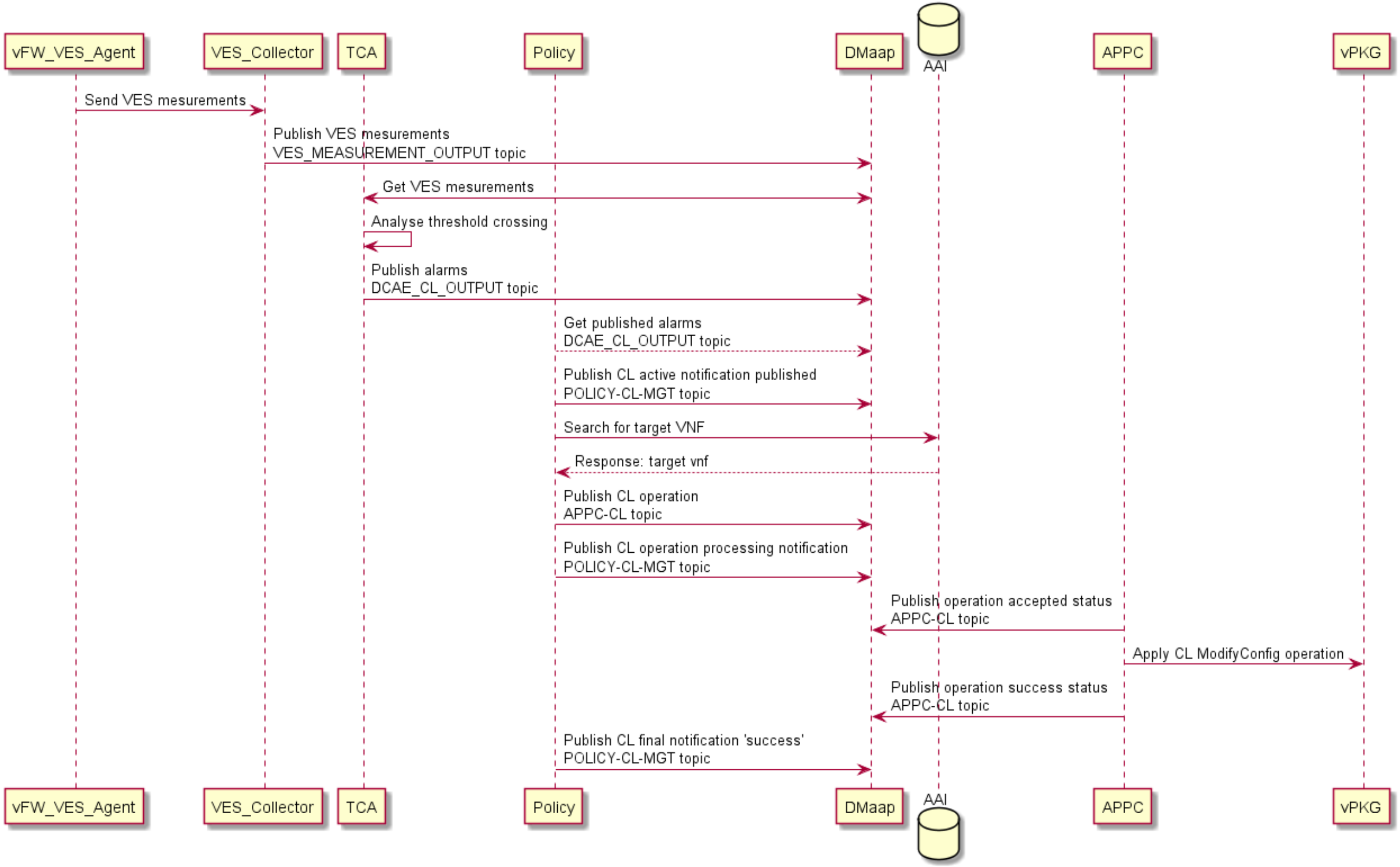
Dmaap is restful service relying on kafka. It acts as message transport bus for the closed loop execution. Clients publish topics and subscribe to them.



Policy is consulted when events occur to identify the appropriate action and the actor that must perform it based on predefined rules.
Policy could direct a request to the orchestrator or controllers based on the scope of the resolution.

DCAE provides data collection and performs analytics according to a CL model, configurations and on-boarded specifications (VES templates ...)

vFW closed loop
Flow diagram



vFW_VES_Agent

VES_Collector

TCA

Policy

DMaap

AAI

APPC

vPKG

Send VES mesurements

Publish VES mesurements
VES_MEASUREMENT_OUTPUT topic

Get VES mesurements

Analyse threshold crossing

Publish alarms
DCAE_CL_OUTPUT topic

Get published alarms
DCAE_CL_OUTPUT topic

Publish CL active notification published
POLICY-CL-MGT topic

Search for target VNF

Response: target vnf

Publish CL operation
APPC-CL topic

Publish CL operation processing notification
POLICY-CL-MGT topic

Publish operation accepted status
APPC-CL topic

Apply CL ModifyConfig operation

Publish operation success status
APPC-CL topic

Publish CL final notification 'success'
POLICY-CL-MGT topic

vFW_VES_Agent

VES_Collector

TCA

Policy

DMaap

AAI

APPC

vPKG

Analysis

The vFW closed loop is based on VES reported measurements that TCA analyses and publishes its related alarms to DMaap.

The VES collector publishes measurements received from VES agent in the vFW host, and TCA gets measurements and applies analytics.

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, it also follows the execution of actions by posting CL operation notification.

All closed loop interactions are relying on the Message router (DMaap) by publishing topics and subscribing to them.

On the APPC side, the CL the action corresponds to the execution of a Directed graph called topology-operation-all that implements an action for modifying configuration.

Thanks

Contacts:

badreddine.aouledyahia@gmail.com

christian.destre@orange.com