

VES event enrichment for DCAE mS

- [Enrichment Key](#)
- [CL Event Structure \(From DCAE mS to Policy\)](#)
- [A&AI Enrichment APIs](#)
- [General enrichment guidelines for CL flow](#)
- [VES Event Flow - VNF Enrichment example](#)
- [VES Event Flow - VM Enrichment example](#)
- [DCAE TCA design for Enrichment](#)
 - [Enrichment flow \(VM enrichment is stretch goal for R1\)](#)
- [Reference](#)

For supporting R1 usecases, DCAE expects VES events to be received from both VNF and VM through the VES Collector. Enrichment for these events are necessary for TCA, Holmes (both under DCAE platform) and Policy to support the close-loop flows.

For R1 – enrichment for only VNF and VM are being discussed in this scope; although same rules can be extended for PNF in future release.

Enrichment Key

VES specification requires the sender (vnf/vm) to populate `sourceName` – the actual entity generating the event to DCAE (note this value can be different from `reportingEntityName` incase when EMS is involved). This `sourceName` is expected under **`event.commonEventHeader.sourceName`** in the VES payload. This field is mandatory field in VES `commonEventHeader` and will be used as key for A&AI enrichment lookup

CL Event Structure (From DCAE mS to Policy)

Fields	Required	Description
AAI	Yes	Contains the A&AI Node-Attribute list; dependent on successful enrichment of VNF/VM type
closedLoopAlarmEnd	Yes - only for ABATED	When the alarm was cleared. This field need only be present in the ABATED message.
closedLoopAlarmStart	Yes	When the alarm was first detected.
closedLoopControlName	Yes	This is the unique ID for the Control Loop. It is created by the CLAMP platform during Control Loop design. The DCAE Micro service that publishes this event structure MUST include this ID.
closedLoopEventClient	No	For monitoring/logging/auditing purposes, if there is an instance ID of the DCAE micro service this field should be populated with it.
closedLoopEventStatus	Yes	This is the status of the closedLoopControlName/requestID pair. It can either ONSET or ABATED.
from	Yes	The ONAP platform component publishing this message. If DCAE, then it should be 'DCAE'.
policyName		The name of the Policy driving the DCAE micro service. Should be a part of the configuration policy setup by CLAMP and passed by DCAE controller
policyScope		The scope of the Policy driving the DCAE micro service. Should be a part of the configuration policy setup by CLAMP and passed by DCAE controller
policyVersion		The version of the Policy driving the DCAE Micro service. Should be a part of the configuration policy setup by CLAMP and passed by DCAE controller
requestID	Yes	For the control loop, when an instance of the Control Loop occurs, this unique ID must be created. The same ID must be forwarded for both the ONSET and the ABATED control loop messages.
target	Yes	This is the name of the field within the A&AI sub-tag that indicates the actual entity Node details. There should be a matching node field within the A&AI subtag holding this value.
target_type	Yes	The type of the target: VM or VNF. Future PNF

version	Yes	The version of the Control Loop event message. Should be '1.0.2'
---------	-----	--

A&AI Enrichment APIs

VNF

Query generic-vnf object based on **vnf-name** (maps to event.commonEventHeader.sourceName in VES payload)

<https://{aai}/aai/v11/network/generic-vnfs/generic-vnf?vnf-name={vnf-name}>

Example : <https://aai.onap:8443/aai/v11/network/generic-vnfs/generic-vnf?vnf-name=demofw01fwl>

Object Returned: [v11-generic-vnf-object.docx](#)

VM

Requires two step A&AI query

- 1) Perform node based query to identify the vserver object resource-link using **vserver-name** as parameter to below api (vserver-name maps to event.commonEventHeader.sourceName in VES payload)

<https://{aai}/aai/v11/search/nodes-query?search-node-type=vserver&filter=vserver-name:EQUALS:{vserver-name}>

On a successful match this would return following object containing resource-type and resource-link

result-data: *object*

resource-type: *string*

The specific type of node in the A&AI graph

resource-link: *string*

The URL to the specific resource

- 2) Use the "resource-link" returned to query A&AI to fetch the vserver specific details. (see VM examples below)

Object Returned: [v11-vserver-object.docx](#)

Based on the query parameters, the A&AI returned object can be received as json or xml. For JSON object – include '-H "Accept: application/json"' in the request (default response is xml)

General enrichment guidelines for CL flow

1. If there is indication from VES event (based on A&AI naming convention) or control loop parameters for mS – then appropriate A&AI api's can be invoked directly. In the absence of information, mS will be required to call more than single API to determine the correct object to query (depending on VNF or VM – different API's call are invoked as noted above).
2. On **Successful enrichment**, CL event to Policy will be updated to set based on A&AI object

For VNF, override below fields in CL event to Policy

VNF CL Event Format

```
"target_type": "VNF",
"target": "generic-vnf.vnf-name",
"AAI": {
  "generic-vnf.in-maint": value,
  "generic-vnf.is-closed-loop-disabled": value,
  "generic-vnf.orchestration-status": value,
  "generic-vnf.prov-status": value,
  "generic-vnf.resource-version": value,
  "generic-vnf.service-id": value,
  "generic-vnf.vnf-id": value,
  "generic-vnf.vnf-name": value,
  "generic-vnf.vnf-type": value
}
```

For VM, override below fields in CL event to Policy

VM CL Event Format

```
"target_type": "VM",
"target": "vserver.vserver-name",
"AAI": {
  "vserver.in-maint": value,
  "vserver.is-closed-loop-disabled": value,
  "vserver.prov-status": value,
  "vserver.resource-version": value
  "vserver.vserver-id": value,
  "vserver.vserver-name": value,
  "vserver.vserver-name2": value
  "vserver.vserver-selflink": value
}
```

3. If **A&AI enrichment lookup fails**, DCAE mS can identify the target CL structure based on configuration model defined per Closed loop.

For VNF, override below fields in CL event to Policy

Unenriched VNF CL Event Format

```
"target_type": "VNF",
"target": "generic-vnf.vnf-name",
"AAI": {
  "generic-vnf.vnf-name": $event.commonEventHeader.sourceName
}
```

For VM, override below fields in CL event to Policy

Unenriched VM CL Event Format

```
"target_type": "VM",  
"target": "vserver.vserver-name",  
"AAI": {  
  "vserver.vserver-name": $event.commonEventHeader.sourceName  
}
```

Note: Policy would use lack of other A&AI fields as an indicator for failed DCAE enrichment and attempts to query A&AI directly if necessary to perform the CL flow

4. The "requestId" field between Onset and abatement CL event should match for Policy team.

5. No enrichment necessary for Abatement event (per Pam/policy team)

VES Event Flow - VNF Enrichment example

VNF input

```
{
  "event":{
    "commonEventHeader":{
      "startEpochMicrosec":1500584201765465,
      "sourceId":"Dummy VM UUID - No Metadata available",
      "eventId":"11",
      "reportingEntityId":"No UUID available",
      "internalHeaderFields":{
        "collectorTimeStamp":"Thu, 07 20 2017 08:56:52 GMT"
      },
      "eventType":"HTTP request rate",
      "priority":"Normal",
      "version":1.2,
      "reportingEntityName":"fw11",
      "sequence":11,
      "domain":"measurementsForVfScaling",
      "lastEpochMicrosec":1500584212017216,
      "eventName":"vnfScalingMeasurement",
      "sourceName":"vCPEInfraVNF13"
    },
    "measurementsForVfScalingFields":{
      "cpuUsageArray":[
        {
          "percentUsage":0,
          "cpuIdentifier":"cpu1",
          "cpuIdle":100,
          "cpuUsageSystem":0,
          "cpuUsageUser":0
        }
      ],
      "measurementInterval":10,
      "requestRate":7832,
      "vNicUsageArray":[
        {
          "transmittedOctetsDelta":5680,
          "receivedTotalPacketsDelta":14,
          "vNicIdentifier":"eth0",
          "valuesAreSuspect":"true",
          "transmittedTotalPacketsDelta":31,
          "receivedOctetsDelta":1332
        }
      ],
      "measurementsForVfScalingVersion":2.1
    }
  }
}
```

A&AI query and output sample

```
curl --header "Content-Type: application/json" --header 'X-FromAppId: vv-temp' --header "X-TransactionId: vv-temp" -H "Accept: application/json" -u DCAE:DCAE https://209.61.160.97:8443/aai/v11/network/generic-vnfs/generic-vnf?vnf-name=vCPEInfraVNF13 -k | python -m json.tool
```

```
{
  "in-maint":false,
  "is-closed-loop-disabled":false,
  "orchestration-status":"Created",
  "prov-status":"PREPROV",
  "relationship-list":{
    "relationship":[
      {
        "related-link":"/aai/v11/business/customers/customer/Demonstration3/service-subscriptions/service-subscription/vCPE/service-instances/service-instance/e8feceb6-28ae-480a-bfbc-1985ce333526",
        "related-to":"service-instance",
        "related-to-property":[
          {
            "property-key":"service-instance.service-instance-name",
            "property-value":"vCPEInfraSI13"
          }
        ],
        "relationship-data":[
          {
            "relationship-key":"customer.global-customer-id",
            "relationship-value":"Demonstration3"
          },
          {
            "relationship-key":"service-subscription.service-type",
            "relationship-value":"vCPE"
          },
          {
            "relationship-key":"service-instance.service-instance-id",
            "relationship-value":"e8feceb6-28ae-480a-bfbc-1985ce333526"
          }
        ]
      }
    ]
  },
  "resource-version":"1504896046185",
  "service-id":"e8cb8968-5411-478b-906a-f28747de72cd",
  "vnf-id":"63b31229-9a3a-444f-9159-04ce2dca3be9",
  "vnf-name":"vCPEInfraVNF13",
  "vnf-type":"vCPEInfraService10/vCPEInfraService10 0"
}
```

CL Event Output

```
{
  "closedLoopEventClient": "DCAE_INSTANCE_ID.dcae-tca",
  "policyVersion": "1.0.0.5",
  "policyName": "vFirewall",
  "policyScope": "resource=SampleResource,service=SampleService,type=SampleType,
closedLoopControlName=SampleClosedLoop",
  "target_type": "VNF",
  "AAI": {
    "generic-vnf.in-maint": false,
    "generic-vnf.is-closed-loop-disabled": false,
    "generic-vnf.orchestration-status": "Created",
    "generic-vnf.prov-status": "PREPROV",
    "generic-vnf.resource-version": "1504896046185",
    "generic-vnf.service-id": "e8cb8968-5411-478b-906a-f28747de72cd",
    "generic-vnf.vnf-id": "63b31229-9a3a-444f-9159-04ce2dca3be9",
    "generic-vnf.vnf-name": "vCPEInfraVNF13",
    "generic-vnf.vnf-type": "vCPEInfraService10/vCPEInfraService10 0"
  },
  "closedLoopAlarmStart": 1484855291527925,
  "closedLoopEventStatus": "ONSET",
  "closedLoopControlName": "CL-FRWL-LOW-TRAFFIC-SIG-d925ed73-8231-4d02-9545-db4e101f88f8",
  "version": "1.0.2",
  "target": "generic-vnf.vnf-id",
  "requestID": "8c1b8bd8-06f7-493f-8ed7-daaa4cc481bc",
  "from": "DCAE"
}
```

VES Event Flow - VM Enrichment example

VES Input

```
"event": {
  "commonEventHeader": {
    "startEpochMicrosec": 1500584201765465,
    "sourceId": "Dummy VM UUID - No Metadata available",
    "eventId": "11",
    "reportingEntityId": "No UUID available",
    "internalHeaderFields": {
      "collectorTimeStamp": "Thu, 07 20 2017 08:56:52 GMT"
    },
    "eventType": "HTTP request rate",
    "priority": "Normal",
    "version": 1.2,
    "reportingEntityName": "vLB",
    "sequence": 10,
    "domain": "measurementsForVfScaling",
    "lastEpochMicrosec": 1500584212017216,
    "eventName": "vLoadBalancer",
    "sourceName": "vserver-name-val-2"
  },
  "measurementsForVfScalingFields": {
    "cpuUsageArray": [{
      "percentUsage": 0,
      "cpuIdentifier": "cpu1",
      "cpuIdle": 100,
      "cpuUsageSystem": 0,
      "cpuUsageUser": 0
    }],
    "measurementInterval": 10,
    "requestRate": 7832,
    "vNicUsageArray": [{
      "transmittedOctetsDelta": 5680,
      "receivedTotalPacketsDelta": 14,
      "vNicIdentifier": "eth0",
      "valuesAreSuspect": "true",
      "transmittedTotalPacketsDelta": 31,
      "receivedOctetsDelta": 1332
    }],
    "measurementsForVfScalingVersion": 2.1
  }
}
```

A&AI query and output sample

Query-1 - To identify vserver object resource-link based on vserver-name

```
curl -k -X GET --user DCAE:DCAE -H "X-FromAppId: testUser" -H "X-TransactionId: testUserTrans" -H "Content-Type: application/json" -H "Accept: application/json" -H "Real-Time: true" "https://209.61.160.97:8443/aai/v11/search/nodes-query?search-node-type=vserver&filter=vserver-name:EQUALS:example-vserver-name-val-2" | python -m json.tool
```

```
{
  "result-data": [
    {
      "resource-link": "/aai/v11/cloud-infrastructure/cloud-regions/cloud-region/Rackspace/DFW/tenants/tenant/1031120/vservers/vserver/example-vserver-id-val-2",
      "resource-type": "vserver"
    }
  ]
}
```

Query-2 - Get the vserver object for enrichment

```
$ curl -k -X GET --user DCAE:DCAE -H "X-FromAppId: testUser" -H "X-TransactionId: testUserTrans" -H "Content-Type: application/json" -H "Accept: application/json" -H "Real-Time: true" "https://209.61.160.97:8443/aai/v11/cloud-infrastructure/cloud-regions/cloud-region/Rackspace/DFW/tenants/tenant/1031120/vservers/vserver/example-vserver-id-val-2" | python -m json.tool
```

```
{
  "in-maint": true,
  "is-closed-loop-disabled": true,
  "prov-status": "example-prov-status-val-2",
  "resource-version": "1504912891060",
  "vserver-id": "example-vserver-id-val-2",
  "vserver-name": "example-vserver-name-val-2",
  "vserver-name2": "example-vserver-name2-val-2",
  "vserver-selflink": "example-vserver-selflink-val-2"
}
```

CL Event Output

```
{
  "closedLoopEventClient": "DCAE_INSTANCE_ID.dcae-tca",
  "policyVersion": "1.0.0.5",
  "policyName": "vLoadBalancer",
  "policyScope": "resource=SampleResource,service=SampleService,type=SampleType",
  "closedLoopControlName=SampleClosedLoop",
  "target_type": "VM",
  "AAI": {
    "vserver.in-maint": true,
    "vserver.is-closed-loop-disabled": true,
    "vserver.prov-status": "example-prov-status-val-2",
    "vserver.resource-version": "1504912891060",
    "vserver.vserver-id": "example-vserver-id-val-2",
    "vserver.vserver-name": "example-vserver-name-val-2",
    "vserver.vserver-name2": "example-vserver-name2-val-2",
    "vserver.vserver-selflink": "example-vserver-selflink-val-2"
  },
  "closedLoopAlarmStart": 1484855291527925,
  "closedLoopEventStatus": "ONSET",
  "closedLoopControlName": "CL-LB-LOW-TRAFFIC-SIG-d925ed73-8231-4d02-9545-db4e101f88f8",
  "version": "1.0.2",
  "target": "generic-vnf.vnf-id",
  "requestID": "8c1b8bd8-06f7-493f-8ed7-daaa4cc481bc",
  "from": "DCAE"
}
```

DCAE TCA design for Enrichment

Besides configuration for TCA processing, TCA configuration model will be enhanced to support following new configuration

- RESTProxyHost, RESTProxyPort, (this will be defined part of TCA app_config to indicate A&AI host/port (or MSB host/port) to be used for enrichment. This value will be preset into blueprint for demo setup)
- AppId, AppPwd (to match as provisioned in A&AI)
- Configurable cache time (to avoid repeated A&AI query)
- Per CL flow
 - controlLoopSchemaType = VM|VNF
 - closedLoopEventStatus= ONSET | ABATED

Enrichment flow (VM enrichment is stretch goal for R1)

- When new CL condition is met
- invoke A&AI to check vnf (**generic-vnf** object lookup)
 - If generic-vnf object lookup is successful
 - Set/override fields based on new CL structure and set target-type and AAI setting based on **generic-vnf** object
 - If no match from A&AI – proceed to vm based check
- Invoke A&AI to check vserver (involves two additional AAI API call 1) get vserver link 2) query **vserver** object)
 - If vserver object lookup is successful
 - Set/override fields based on new CL structure and set target-type and AAI setting based on **vserver** object
 - If no match from A&AI – proceed to configuration model based setting.
- Use configuration model identified "**controlLoopSchemaType**" to set CL event (structure as we have currently)

Reference

Policy - <https://wiki.onap.org/display/DW/Control+Loop+Operational+Policy?focusedCommentId=10782165#comment-10782165>

A&AI - <https://wiki.onap.org/pages/viewpage.action?pageId=13598793>