

Policy Specification and Retrieval for OOF

Contents

- [Summary](#)
- [Introduction](#)
 - [Policy Platform](#)
 - [Resources Related to Policy](#)
- [OOF Policies \(multiple applications\)](#)
 - [OOF-HAS Policies](#)
 - [OOF-HAS Service Design Policies](#)
 - [OOF-CMSO Policies](#)
- [Modes for Fetching Policies](#)
- [TOSCA Policy Models](#)
- [HAS Policies for Testing](#)

Summary

Introduction

The OOF retrieves applicable constraints and objective functions as policies from the policy platform. These policies are retrieved at runtime, thus allowing an operator to change policies as and when needed. These policies are specified using policy models that are on-boarded during the OOF application deployment-time in the policy platform. The OOF-related policy models are typically derived from the constraints and objective functions of an OOF-application. Currently, for R2, these models will be pushed into the policy platform manually using the OOF policy uploader module.

Policy Platform

The OOF currently relies on the following features of the policy platform:

- Policy specification: Optimization constraints and objective functions can be specified in terms of policies by service providers and operators.
- Policy prioritization: Policies capturing constraints and objectives can be prioritized.
- Policy searching and filtering: OOF policies can be searched and filtered based on different criteria; by scope or by name patterns.

Resources Related to Policy

OOF Policies (multiple applications)

OOF-HAS Policies

OOF-HAS Service Design Policies

Service design policies are typically defined as a part of a service design model and evaluated/enforced prior to the service instantiation phase. For example, Hardware Platform Enablement (HPA) policies are defined in an SDC service model and evaluated/enforced during the homing optimization process in ONAP.

The HPA requirements are captured in a VNF descriptor provided by a vendor. During the service-design phase, these requirements are extracted out from the VNF descriptor and captured in a service model. Some of the HPA requirements cannot be changed during the lifecycle of a service; while other optional requirements can be changed by a service designer or operator over time (e.g. the use of SR-IOV may be preferred by an operator if the SR-IOV capability is optionally provided by a vendor). Once a service model is designed and uploaded into the SDC repository, SDC notifies the policy service about the distribution of a new service model. The policy service then fetches the service model and extracts out the HPA requirements through model decomposition. Next, the policy service creates HPA policies for the homing service (OOF-HAS) by populating the HPA requirements in policy models provided by OOF, and stores the policies in a repository. When the homing service receives a homing request from SO, the homing service asks for the associated HPA policies from the policy platform. Finally, the homing service finds homing solutions based on the evaluation of the received HPA policies and returns the solutions to SO.

As per meeting minutes here: [OOF 2018-02-19 Meeting Notes](#), it was understood that all HPA requirements for a given VM can be put in one policy or across different policies. Also, it was discussed that each HPA capability can be enhanced to include mandatory and score attributes. Keeping this in mind, HPA capabilities are defined here. As indicated, policy can have multiple capabilities.

Template for the HPA policies

```
{
  "service": "{the model name of a policy}",
  "policyName": "{policy scope folder}.{policy name; must be
unique}",
  "description": "{description of a policy}",
  "templateVersion": "{policy model version}",
  "version": "{policy version}",
  "priority": "{priority in 1-10; larger the number higher the
priority is}",
  "riskType": "{the type of risk}",
  "riskLevel": "{the level of risk}",
  "guard": "{True/False flag to indicate whether guard is
applicable or not}",
  "content": {
    "resource": "{resource name}",
    "identity": "{policy identity}",
    "policyScope": ["{a tag representing policy scope}"],
    "policyType": "{policy type}",
    "flavorFeatures": [
      {
        "flavorLabel": "{VM/VFC id}",
        "flavorProperties": [
          {
            "hpa-feature" : "{HPA feature type}",
            "mandatory" : "{mandatoriness of the feature}",
            "score" : "{priority of this feature if the feature
is not mandatory for VM}"
            "architecture": "{supported architecture}",
            "hpa-version": "{version for hpa capability}",
            "hpa-feature-attributes": [
              {
                "hpa-attribute-key": "{attribute name}",
                "hpa-attribute-value": "{attribute value}",
                "operator": "{comparison operator}",
                "unit": "{the unit of an attribute}"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

Template for the HPA policies.

HPA Policy Example

```
{
  "service": "hpaPolicy",
  "policyName": "oofBeijing.hpaPolicy_vGMuxInfra",
  "description": "HPA policy for vGMuxInfra",
  "templateVersion": "0.0.1",
  "version": "1.0",
  "priority": "3",
  "riskType": "test",
  "riskLevel": "2",
  "guard": "False",
  "content": {
    "resources": "vGMuxInfra",
    "identity": "hpaPolicy_vGMuxInfra",
    "policyScope": ["vCPE", "US", "INTERNATIONAL", "ip",
"vGMuxInfra"],
    "policyType": "hpaPolicy",
    "flavorFeatures": [
      {
        "flavorLabel": "flavor_label_vm_01",
```

```

    "flavorProperties": [
      {
        "hpa-feature" : "cpuTopology",
        "mandatory" : "True",
        "architecture": "generic",
        "hpa-version": "v1",
        "hpa-feature-attributes": [
          { "hpa-attribute-key": "numCpuSockets", "hpa-attribute-value": "2", "operator": ">=", "unit": "" },
          { "hpa-attribute-key": "numCpuSockets", "hpa-attribute-value": "4", "operator": "<=", "unit": "" },
          { "hpa-attribute-key": "numCpuCores", "hpa-attribute-value": "2", "operator": ">=", "unit": "" },
          { "hpa-attribute-key": "numCpuCores", "hpa-attribute-value": "4", "operator": "<=", "unit": "" },
          { "hpa-attribute-key": "numCpuThreads", "hpa-attribute-value": "4", "operator": ">=", "unit": "" },
          { "hpa-attribute-key": "numCpuThreads", "hpa-attribute-value": "8", "operator": "<=", "unit": "" }
        ]
      },
      {
        "hpa-feature" : "basicCapabilities",
        "mandatory" : "True",
        "architecture": "generic",
        "hpa-version": "v1",
        "hpa-feature-attributes": [
          { "hpa-attribute-key": "numVirtualCpu", "hpa-attribute-value": "6", "operator": "=", "unit": "" },
          { "hpa-attribute-key": "virtualMemSize", "hpa-attribute-value": "6", "operator": "=", "unit": "GB" }
        ]
      },
      {
        "hpa-feature" : "ovsDpdk",
        "mandatory" : "False",
        "score" : "3",
        "architecture": "generic",
        "hpa-version": "v1",
        "hpa-feature-attributes": [
          { "hpa-attribute-key": "dataProcessingAccelerationLibrary", "hpa-attribute-value": "ovsDpdk_version", "operator": "=", "unit": "" }
        ]
      },
      {
        "hpa-feature" : "cpuInstructionSetExtensions",
        "mandatory" : "True",
        "architecture": "INTEL-64",
        "hpa-version": "v1",
        "hpa-feature-attributes": [
          { "hpa-attribute-key": "instructionSetExtensions", "hpa-attribute-value": ["<CPUINST>", "<CPUINST>"], "operator": "ALL", "unit": "" }
        ]
      }
    ],
    {
      "flavorLabel": "flavor_label_vm_02",
      "flavorProperties": [
        {
          "hpa-feature" : "cpuPinningy",
          "mandatory" : "True",
          "architecture": "generic",
          "hpa-version": "v1",
          "hpa-feature-attributes": [
            { "hpa-attribute-key": "logicalCpuThreadPinningPolicy", "hpa-attribute-value": "<CPUTHREADPOLICY>", "operator": "=", "unit": "" },
            { "hpa-attribute-key": "logicalCpuPinningPolicy",

```

```

    "hpa-attribute-value": "<CPUPOLICY>", "operator": "=", "unit": ""},
  ],
  {
    "hpa-feature" : "basicCapabilities",
    "mandatory" : "True",
    "architecture": "generic",
    "hpa-version": "v1",
    "hpa-feature-attributes": [
      {"hpa-attribute-key": "numVirtualCpu", "hpa-attribute-value": "6", "operator": "=", "unit": ""},
      {"hpa-attribute-key": "virtualMemSize", "hpa-attribute-value": "6", "operator": "=", "unit": "GB"}
    ]
  },
  {
    "hpa-feature" : "localStorage",
    "mandatory" : "False",
    "score" : "5",
    "architecture": "generic",
    "hpa-version": "v1",
    "hpa-feature-attributes": [
      {"hpa-attribute-key": "diskSize", "hpa-attribute-value": "2", "operator": "=", "unit": "GB"},
      {"hpa-attribute-key": "ephemeralDiskSize", "hpa-attribute-value": "2", "operator": "=", "unit": "GB"},
      {"hpa-attribute-key": "swapMemSize", "hpa-attribute-value": "16", "operator": "=", "unit": "MB"},
    ]
  },
  {
    "hpa-feature" : "pcie",
    "mandatory" : "True",
    "architecture": "generic",
    "hpa-version": "v1",
    "hpa-feature-attributes": [
      {"hpa-attribute-key": "pciCount", "hpa-attribute-value": "2", "operator": "=", "unit": ""},
      {"hpa-attribute-key": "pciVendorId", "hpa-attribute-value": "8086", "operator": "=", "unit": ""},
      {"hpa-attribute-key": "pciDeviceId", "hpa-attribute-value": "2", "operator": "=", "unit": ""},
      {"hpa-attribute-key": "functionType", "hpa-attribute-value": "<PCITYPEVALUE>", "operator": "=", "unit": ""}
    ]
  }
],
{
  "flavorLabel": "flavor_label_vm_03",
  "flavorProperties": [
    {
      "hpa-feature" : "numa",
      "mandatory" : "False",
      "score" : "5",
      "architecture": "generic",
      "hpa-version": "v1",
      "hpa-feature-attributes": [
        {"hpa-attribute-key": "numaNodes", "hpa-attribute-value": "2", "operator": "=", "unit": ""},
        {"hpa-attribute-key": "numaCpu-0", "hpa-attribute-value": "2", "operator": "=", "unit": ""},
        {"hpa-attribute-key": "numaMem-0", "hpa-attribute-value": "2048", "operator": "=", "unit": "MB"},
        {"hpa-attribute-key": "numaCpu-1", "hpa-attribute-value": "4", "operator": "=", "unit": ""},
        {"hpa-attribute-key": "numaMem-1", "value": "4096", "operator": "=", "unit": "MB"}
      ]
    }
  ],
  {

```

```

        "hpa-feature" : "basicCapabilities",
        "mandatory" : "True",
        "architecture": "generic",
        "hpa-version": "v1",
        "hpa-feature-attributes": [
            { "hpa-attribute-key": "numVirtualCpu", "hpa-
attribute-value": "6", "operator": "=", "unit": "" },
            { "hpa-attribute-key": "virtualMemSize", "hpa-
attribute-value": "6", "operator": "=", "unit": "GB" }
        ],
        {
            "hpa-feature" : "hugePages",
            "mandatory" : "False",
            "score" : "7",
            "architecture": "generic",
            "hpa-version": "v1",
            "hpa-feature-attributes": [
                { "hpa-attribute-key": "memoryPageSize", "hpa-
attribute-value": "<MEMORYPAGESIZE>", "operator": "=", "unit": "" }
            ]
        }
    ]
}

```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

CPUTHREADPOLICY = prefer, isolate, require

CPUPOLICY = shared, dedicated

PCIETYPEVALUE: SRIOV, PCI-PASSTHROUGH

CPUINST = aes, avx, sha_ni, mpx, adcx, mpx, pclmulqdq, rdrand, sse, etc

MEMORYPAGESIZE = 4KB (unit=KB), 2MB (unit=MB), 1GB (unit=GB), ANY, Other explicit Page size

Distance Policy Example

```
{
  "service": "distancePolicy",
  "policyName": "oofBeijing.distancePolicy_vGMuxInfra",
  "description": "Distance Policy for vGMuxInfra",
  "templateVersion": "0.0.1",
  "version": "oofBeijing",
  "priority": "3",
  "riskType": "test",
  "riskLevel": "2",
  "guard": "False",
  "content": {
    "distanceProperties": {
      "locationInfo": customer_location",
      "distance": { "value": "500", "operator": "<", "unit": "km"
    }
  },
  "resources": ["vGMuxInfra", "vG"],
  "applicableResources": "any",
  "identity": "distance-vGMuxInfra",
  "policyScope": ["vCPE", "US", "INTERNATIONAL", "ip",
    "vGMuxInfra"],
  "policyType": "distancePolicy"
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. However, the policy can be updated over time by operators.

HPA Basic Capabilities Policy Example

```
{
  "hpa-feature" : "basicCapabilities",
  "mandatory" : "False",
  "score" : "5",
  "architecture": "generic",
  "hpa-version": "v1",
  "hpa-feature-attributes": [
    { "hpa-attribute-key": "numVirtualCpu", "hpa-attribute-
      value": "6", "operator": "=", "unit": "" },
    { "hpa-attribute-key": "virtualMemSize", "hpa-attribute-
      value": "6", "operator": "=", "unit": "GB" }
  ]
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

HPA OVS DPDK Policy Example

```
{
  "hpa-feature" : "ovsDpdk",
  "mandatory" : "False",
  "score" : "5",
  "architecture": "INTEL-64",
  "hpa-version": "v1",
  "hpa-feature-attributes": [
    { "hpa-attribute-key": "dataProcessingAccelerationLibrary",
      "hpa-attribute-value": "ovsDpdk_version", "operator": "=",
      "unit": "" }
  ]
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

For this policy others architectures are also applicable.

HPA CPU Pinning Policy Example

```
{
  "hpa-feature" : "cpuPinning",
  "mandatory" : "False",
  "score" : "5",
  "architecture": "generic",
  "hpa-version": "v1",
  "hpa-feature-attributes": [
    { "hpa-attribute-key": "logicalCpuThreadPinningPolicy", "hpa-attribute-value": "<CPUTHREADPOLICY>", "operator": "=", "unit": "" },
    { "hpa-attribute-key": "logicalCpuPinningPolicy", "hpa-attribute-value": "<CPUPOLICY>", "operator": "=", "unit": "" },
  ]
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

CPUTHREADPOLICY = prefer, isolate, require

CPUPOLICY = shared, dedicated

HPA NUMA Policy Example

```
{
  "hpa-feature" : "numa",
  "mandatory" : "False",
  "score" : "5",
  "architecture": "generic",
  "hpa-version": "v1",
  "hpa-feature-attributes": [
    { "hpa-attribute-key": "numaNodes", "hpa-attribute-value":
"2", "operator": "=", "unit": "" },
    { "hpa-attribute-key": "numaCpu-0", "hpa-attribute-value":
2, "operator": "=", "unit": "" },
    { "hpa-attribute-key": "numaMem-0", "hpa-attribute-value":
"2048", "operator": "=", "unit": "MB" },
    { "hpa-attribute-key": "numaCpu-1", "hpa-attribute-value":
4, "operator": "=", "unit": "" },
    { "hpa-attribute-key": "numaMem-1", "hpa-attribute-value":
"4096", "operator": "=", "unit": "MB" }
  ]
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

HPA CPU Topology Policy Example

```
{
  "hpa-feature" : "cpuTopology",
  "mandatory" : "False",
  "score" : "5",
  "architecture": "generic",
  "hpa-version": "v1",
  "hpa-feature-attributes": [
    { "hpa-attribute-key": "numCpuSockets", "hpa-attribute-
value": "2", "operator": ">=", "unit": "" },
    { "hpa-attribute-key": "numCpuSockets", "hpa-attribute-
value": "4", "operator": "<=", "unit": "" },
    { "hpa-attribute-key": "numCpuCores", "hpa-attribute-value":
"2", "operator": ">=", "unit": "" },
    { "hpa-attribute-key": "numCpuCores", "hpa-attribute-value":
"4", "operator": "<=", "unit": "" },
    { "hpa-attribute-key": "numCpuThreads", "hpa-attribute-
value": "4", "operator": ">=", "unit": "" },
    { "hpa-attribute-key": "numCpuThreads", "hpa-attribute-
value": "8", "operator": "<=", "unit": "" }
  ]
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

Affinity Policy Example

```
{
  "service": "affinityPolicy",
  "policyName": "oofBeijing.affinityPolicy_vcpe",
  "description": "Affinity policy for vCPE",
  "templateVersion": "1702.03",
  "version": "oofBeijing",
  "priority": "5",
  "riskType": "test",
  "riskLevel": "2",
  "guard": "False",
  "content": {
    "identity": "affinity_vCPE",
    "policyScope": ["vCPE", "US", "INTERNATIONAL", "ip",
"vGMuxInfra"],
    "affinityProperties": {
      "qualifier": "different",
      "category": "complex"
    },
    "policyType": "affinityPolicy",
    "resources": ["vGMuxInfra", "vG"],
    "applicableResources": "all"
  }
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

Capacity Policy Example

```
{
  "service": "capacityPolicy",
  "policyName": "oofBeijing.capacityPolicy_vGMuxInfra",
  "description": "Capacity policy for vGMuxInfra",
  "templateVersion": "1702.03",
  "version": "oofBeijing",
  "priority": "5",
  "riskType": "test",
  "riskLevel": "2",
  "guard": "False",
  "content": {
    "identity": "capacity_vGMuxInfra",
    "policyScope": ["vCPE", "US", "INTERNATIONAL", "ip",
"vGMuxInfra"],
    "controller": "multicloud",
    "capacityProperties": {
      "cpu": {"value": 2, "operator": ">"},
      "memory": {"value": 4, "operator": ">", "unit": "GB"},
      "storage": {"value": 80, "operator": ">"},
      "unit": "GB"
    },
    "policyType": "vim_fit",
    "resources": ["vGMuxInfra"],
    "applicableResources": "any"
  }
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

Min Guarantee Policy Example

```
{
  "service": "minGuaranteePolicy",
  "policyName": "oofBeijing.minGuarantee_vGMuxInfra",
  "description": "Min guarantee policy for vGMuxInfra",
  "templateVersion": "1702.03",
  "version": "oofBeijing",
  "priority": "5",
  "riskType": "test",
  "riskLevel": "2",
  "guard": "False",
  "content": {
    "identity": "minGuarantee_vGMuxInfra",
    "policyScope": ["vCPE", "US", "INTERNATIONAL", "ip",
"vGMuxInfra"],
    "minGuaranteeProperty": {
      "cpu": "true",
      "memory": "false",
    },
    "type": "minGuaranteePolicy",
    "resources": ["vGMuxInfra"],
    "applicableResources": "any"
  }
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service. A min-guarantee policy can be specified using the HPA policy model.

Optimization Policy Example

```
{
  "service": "PlacementOptimizationPolicy",
  "policyName": "oofBeijing.
PlacementOptimizationPolicy_vGMuxInfra",
  "description": "Placement Optimization Policy for vGMuxInfra",
  "templateVersion": "1702.03",
  "version": "oofBeijing",
  "priority": "5",
  "riskType": "test",
  "riskLevel": "3",
  "guard": "False",
  "content": {
    "objectiveParameter": {
      "parameterAttributes": [
        {
          "resources": ["vGMuxInfra"],
          "customerLocationInfo": "customer_loc",
          "parameter": "distance",
          "weight": "1",
          "operator": "product"=
        },
        {
          "resources": ["vG"],
          "customerLocationInfo": "customer_loc",
          "parameter": "distance",
          "weight": "1",
          "operator": "product"
        }
      ],
      "operator": "sum"
    },
    "identity": "optimization",
    "policyScope": ["vCPE", "US", "INTERNATIONAL", "ip",
"vGMuxInfra"],
    "policyType": "placementOptimization",
    "objective": "minimize"
  }
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

HPA PCIe Policy Example

```
{
  "hpa-feature" : "pcie",
  "mandatory" : "False",
  "score" : "5",
  "architecture": "generic",
  "hpa-version": "v1",
  "hpa-feature-attributes": [
    { "hpa-attribute-key": "pciCount", "hpa-attribute-value":
"2", "operator": "=", "unit": "" },
    { "hpa-attribute-key": "pciVendorId", "hpa-attribute-value": "
8086", "operator": "=", "unit": "" },
    { "hpa-attribute-key": "pciDeviceId", "hpa-attribute-
value": "2", "operator": "=", "unit": "" },
    { "hpa-attribute-key": "functionType", "hpa-attribute-
value": "<PCIETYPEVALUE>", "operator": "=", "unit": "" }
  ]
}
```

PCIETYPEVALUE: SRIOV, PCI-PASSTHROUGH

HPA Local Storage Policy Example

```
{
  "hpa-feature" : "localStorage",
  "mandatory" : "False",
  "score" : "5",
  "architecture": "generic",
  "hpa-version": "v1",
  "hpa-feature-attributes": [
    { "hpa-attribute-key": "diskSize", "hpa-attribute-value":
"2", "operator": "=", "unit": "GB" },
    { "hpa-attribute-key": "ephemeralDiskSize", "hpa-attribute-
value": "2", "operator": "=", "unit": "GB" },
    { "hpa-attribute-key": "swapMemSize", "hpa-attribute-
value": "16", "operator": "=", "unit": "MB" },
  ]
}
```

The fields in this example policy are typically generated from a TOSCA service model specified by VNF vendors or service designers. This policy cannot be changed during the life-cycle of a service.

HPA CPU Instruction Set Extensions Policy Example

```
{
  "hpa-feature" : "cpuInstructionSetExtensions",
  "mandatory" : "False",
  "score" : "5",
  "architecture": "INTEL-64",
  "hpa-version": "v1",
  "hpa-feature-attributes": [
    { "hpa-attribute-key": "instructionSetExtensions", "hpa-
attribute-value": [ "<CPUINST>", "<CPUINST>" ], "operator": "ALL",
"unit": "" }
  ]
}
```

instructionSetExtensions attribute has a list of all instruction set extensions required.

CPUINST = aes, avx, sha_ni, mpx, adcx, mpx, pclmulqdq, rdrand, sse, etc

HPA Huge Pages Policy Example

```
{
  "hpa-feature" : "hugePages",
  "mandatory" : "False",
  "score" : "5",
  "architecture": "generic",
  "hpa-version": "v1",
  "hpa-feature-attributes": [
    { "hpa-attribute-key": "memoryPageSize", "hpa-attribute-
value": "<MEMORYPAGESIZE>", "operator": "=", "unit": ""}
  ]
}
```

MEMORYPAGESIZE = 4KB (unit=KB), 2MB (unit=MB), 1GB
(unit GB), ANY, Other explicit Page size

VNF Policy Example

```
{
  "service": "vnfPolicy",
  "policyName": "oofBeijing.vnfPolicy_vGMuxInfra",
  "description": "vnfPolicy",
  "templateVersion": "1702.03",
  "version": "oofBeijing",
  "priority": "6",
  "riskType": "test",
  "riskLevel": "3",
  "guard": "False",
  "content": {
    "identity": "vnf_vGMuxInfra",
    "policyScope": ["vCPE", "INTERNATIONAL", "ip",
"vGMuxInfra"],
    "policyType": "vnf_policy",
    "resources": ["vGMuxInfra"],
    "applicableResources": "any",
    "vnfProperties": [
      {
        "inventoryProvider": "aai",
        "serviceType": "",
        "inventoryType": "cloudRegionId",
        "customerId": ""
      },
      {
        "inventoryProvider": "multicloud",
        "serviceType": "HNGATEWAY",
        "inventoryType": "serviceInstanceId",
        "customerId": "21014aa2-526b-11e6-beb8-
9e71128cae77"
      }
    ]
  }
}
```

VNF policy captures the location of inventories and customer information.

Subscriber Role Policy Example

```
{
  "service": "SubscriberPolicy",
  "policyName": "oofBeijing.SubscriberPolicy_v1",
  "description": "Subscriber Policy",
  "templateVersion": "0.0.1",
  "version": "oofBeijing",
  "priority": "1",
  "riskType": "test",
  "riskLevel": "3",
  "guard": "False",
  "content": {
    "identity": "subscriber",
    "policyScope": ["vCPE", "subscriber_x", "subscriber_y"],
    "properties": {
      "subscriberName": ["subscriber_x",
"subscriber_y"],
      "subscriberRole": ["Test"],
      "provStatus": ["CAPPED"]
    },
    "policyType": "SubscriberPolicy"
  }
}
```

Subscriber role policy to capture the role of a subscriber and related provisioning states.

OOF-CMSO Policies

TBD

Modes for Fetching Policies

OOF can fetch optimization policies using the getConfig API of the policy system. The policies can be searched and filtered in two different modes; by-name and by-scope from the policy system.

By-name: OOF can send an explicit policy name or a regular expression matching a set of policy names as a part of the getConfig API payload. In return, OOF expects to receive a specific policy or a set of policies with name matching to the regular expression.

By-scope: A scope is the domain to which a policy is applicable. Scope information can be captured as a set of attribute-value pairs, which can be sent as a part of the getConfig API payload. In response, the policy system is expected to return a set of policies with the matching attribute-value pairs. A policy can only be included in the response if all the matching attribute-value pairs exist.

OOF can request prioritization by enabling a prioritization flag in the getConfig API call to the policy system, and expects to receive a single policy with the highest priority policy among the set of policies meeting the search criteria.

TOSCA Policy Models

The following TOSCA policy models need to be uploaded as a dictionary during the deployment-time of an optimization application. Currently, the model uploading process is manual since the policy system does not offer an interface to upload models programmatically. Once the models are uploaded, policy templates are created in the policy portal/GUI using which optimization policies can be created. Alternatively, the policy system offers CRUD REST APIs using which the policies can be managed by the application.

- [HPA Policy](#)
- [Affinity Policy](#)
- [Distance Policy](#)
- [Capacity Policy](#)
- [VNF Policy](#)
- [Optimization Policy](#)
- [Query Policy](#)
- [Subscriber Policy](#)

HAS Policies for Testing



Affinity_vCPE_1.json



Capacity_vG_1.json



Capacity_vGMuxInfra.json



Distance_vG_1.json



Distance_vGMuxInfra_1.json



hpa_policy_vG_1.json



hpa_policy_vGMuxInfra_1.json



Placement_Optimization_1.json



QueryPolicy_vCPE.json



subscriber_policy.json



vnfPolicy_vG.json



vnfPolicy_vGMuxInfra.json