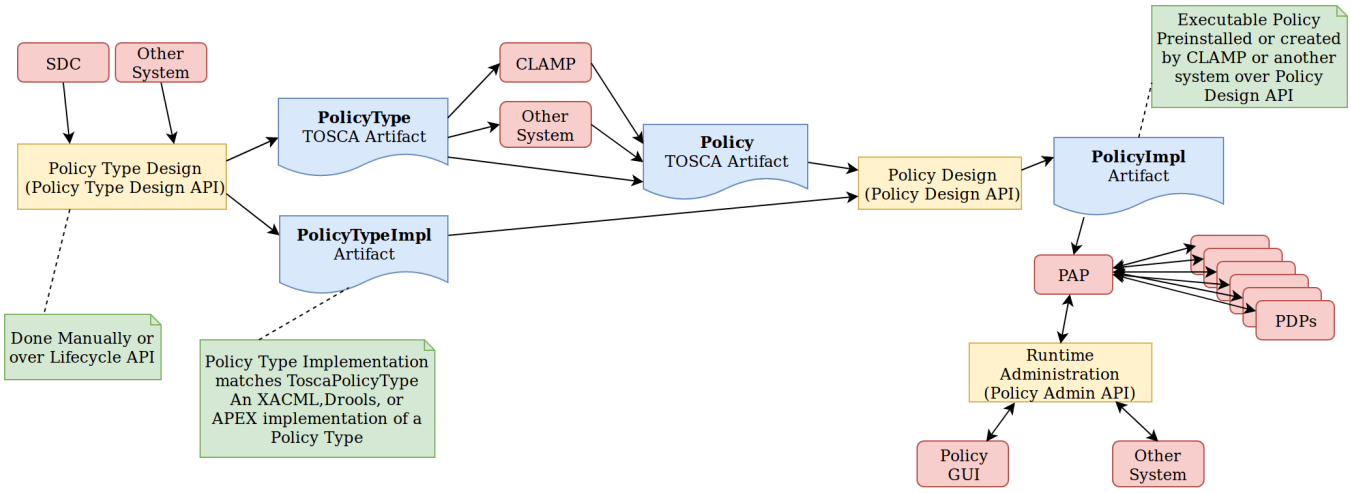
TO BE DELETED - refer to Dublin Documentation

Official Documentation for Dublin

https://docs.onap.org/en/dublin/submodules/policy/parent.git/docs/design/design.html

This page shows how the Policy Design and API Flow to/from the PAP and PDPs works to support Model Driven Control Loops in Dublin.

The figure below shows the Artifacts (Blue) in the ONAP Policy Framework, the Activities (Yellow) that manipulate them, and important components (Pink) that interact with them.



Please see the TOSCA Policy Primer page for an introduction to TOSCA policy concepts.

TOSCA defines a *PolicyType*, the definition of a type of policy that can be applied to a service. It also defines a *Policy*, the definition of an instance of a *PolicyType*. In the Policy Framework, we must handle and manage these TOSCA definitions and tie them to real implementations of policies that can run on PDPs.

The diagram above outlines how this is achieved. Each TOSCA *PolicyType* must have a corresponding *PolicyTypeImpl* in the Policy Framework. The TOSCA *PolicyType* definition can be used to create a TOSCA *Policy* definition, either directly by the Policy Framework, by CLAMP, or by some other system. Once the *Policy* artifact exists, it can be used together with the *PolicyTypeImpl* artifact to create a *PolicyImpl* artifact. A *PolicyImpl* artifact is an executable policy implementation that can run on a PDP.

The TOSCA *PolicyType* artifact defines the external characteristics of the policy; defining its properties, the types of entities it acts on, and its triggers. A *PolicyTypeImpl* artifact is an XACML, Drools, or APEX implementation of that policy definition. *PolicyType* and *PolicyTypeImpl* artifacts may be preloaded, may be loaded manually, or may be created using the Lifecycle API. Alternatively, *PolicyType* definitions may be loaded over the Lifecycle API for preloaded *PolicyTypeImpl* artifacts. A TOSCA *PolicyType* artifact can be used by clients (such as CLAMP or CLI tools) to create, parse, serialize, and /or deserialize an actual Policy.

The TOSCA *Policy* artifact is used internally by the Policy Framework, or is input by CLAMP or other systems. This artifact specifies the values of the properties for the policy and specifies the specific entities the policy acts on. Policy Design uses the TOSCA *Policy* artifact and the *PolicyTypeImpl* artifact to create an executable *PolicyImpl* artifact.

1 Policy Types

Policy Type Design manages TOSCA *PolicyType* artifacts and their *PolicyTypeImpl* implementations.

TOSCA PolicyType may ultimately be defined by the modeling team but for now are defined by the Policy Framework project. Various editors and GUIs are available for creating *PolicyTypeImpl* implementations. However, systematic integration of *PolicyTypeImpl* implementation is outside the scope of the ONAP Dublin release.

The *PolicyType* definitions and implementations listed below are preloaded and are always available for use in the Policy Framework.

Policy Type	Description
onap.policies.Monitoring	Overarching model that supports Policy driven DCAE microservice components used in a Control Loops
onap.policies.controlloop.Operational	Used to support actor/action operational policies for control loops
onap.policies.controlloop.Guard	Control Loop guard policies for policing control loops
onap.policies.controlloop.Coordination	Control Loop Coordination policies to assist in coordinating multiple control loops at runtime

1.1 onap.policies.Monitoring Policy Type

This is a base Policy Type that supports Policy driven DCAE microservice components used in a Control Loops. The implementation of this Policy Type is developed using the XACML PDP to support question/answer Policy Decisions during runtime for the DCAE Policy Handler.

Base Policy Type definition for onap.policies.Monitoring

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
policy_types:
  - onap.policies.Monitoring:
      derived_from: tosca.policies.Root
      version: 1.0.0
      description: a base policy type for all policies that govern monitoring provision
```

The *PolicyTypeImpl* implementation of the *onap.policies.Montoring* Policy Type is generic to support definition of TOSCA *PolicyType* artifacts in the Policy Framework using the Policy Type Design API. Therefore many TOSCA *PolicyType* artifacts will use the same *PolicyTypeImpl* implementation with different property types and towards different targets. This allows dynamically generated DCAE microservice component Policy Types to be created at Design Time.

DCAE microservice components can generate their own TOSCA *PolicyType* using TOSCA-Lab Control Loop guard policies in SDC (Stretch Goal) or can do so manually. See How to generate artefacts for SDC catalog using Tosca Lab Tool for details on TOSCA-LAB in SDC. For Dublin, the DCAE team is defining the manual steps required to build policy models Onboarding steps for DCAE MS through SDC/Policy/CLAMP (Dublin).

NOTE: For Dublin, mS Policy Types will be pre-loaded into the SDC platform and be available as a Normative. The policy framework will pre-load support for those mS Monitoring policy types.

PolicyType onap.policies.monitoring.MyDCAEComponent derived from onap.policies.Monitoring

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
policy_types:
  - onap.policies.Monitoring:
      derived_from: tosca.policies.Root
      version: 1.0.0
      description: a base policy type for all policies that govern monitoring provision
  - onap.policies.monitoring.MyDCAEComponent:
      derived_from: onap.policies.Monitoring
      version: 1.0.0
      properties:
        mydcaecomponent_policy:
          type: map
          description: The Policy Body I need
          entry_schema:
            type: onap.datatypes.monitoring.mydatatype
data_types:
  - onap.datatypes.monitoring.MyDataType:
        derived_from: tosca.datatypes.Root
        properties:
          my_property_1:
            type: string
            description: A description of this property
            constraints:
              - valid_values:
                - value example 1
                - value example 2
```

TCA Example - Please note that the official version of this will be located in the SDC repository.

Example TCA DCAE microservice

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
policy_types:
    onap.policies.Monitoring:
        derived_from: tosca.policies.Root
        description: a base policy type for all policies that governs monitoring provisioning
    onap.policy.monitoring.cdap.tca.hi.lo.app:
        derived_from: onap.policies.Monitoring
        version: 1.0.0
        properties:
            tca_policy:
                type: map
                description: TCA Policy JSON
```

```
                entry_schema:
                    type: onap.datatypes.monitoring.tca_policy
data_types:
    onap.datatypes.monitoring.metricsPerEventName:
        derived_from: tosca.datatypes.Root
        properties:
            controlLoopSchemaType:
                type: string
                required: true
                description: Specifies Control Loop Schema Type for the event Name e.g. VNF, VM
                constraints:
                  - valid_values:
                      - VM
                      - VNF
            eventName:
                type: string
                required: true
                description: Event name to which thresholds need to be applied
            policyName:
                type: string
                required: true
                description: TCA Policy Scope Name
            policyScope:
                type: string
                required: true
                description: TCA Policy Scope
            policyVersion:
                type: string
                required: true
                description: TCA Policy Scope Version
            thresholds:
                type: list
                required: true
                description: Thresholds associated with eventName
                entry_schema:
                    type: onap.datatypes.monitoring.thresholds
    onap.datatypes.monitoring.tca_policy:
        derived_from: tosca.datatypes.Root
        properties:
            domain:
                type: string
                required: true
                description: Domain name to which TCA needs to be applied
                default: measurementsForVfScaling
                constraints:
                  - equal: measurementsForVfScaling
            metricsPerEventName:
                type: list
                required: true
                description: Contains eventName and threshold details that need to be applied to given eventName
                entry_schema:
                    type: onap.datatypes.monitoring.metricsPerEventName
    onap.datatypes.monitoring.thresholds:
        derived_from: tosca.datatypes.Root
        properties:
            closedLoopControlName:
                type: string
                required: true
                description: Closed Loop Control Name associated with the threshold
            closedLoopEventStatus:
                type: string
                required: true
                description: Closed Loop Event Status of the threshold
                constraints:
                  - valid_values:
                      - ONSET
                      - ABATED
            direction:
                type: string
                required: true
                description: Direction of the threshold
```

```
                constraints:
                  - valid_values:
                      - LESS
                      - LESS_OR_EQUAL
                      - GREATER
                      - GREATER_OR_EQUAL
                      - EQUAL
            fieldPath:
                type: string
                required: true
                description: Json field Path as per CEF message which needs to be analyzed for TCA
                constraints:
                  - valid_values:
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].receivedTotalPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].receivedOctetsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedUnicastPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedMulticastPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedBroadcastPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedDiscardedPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].receivedErrorPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedTotalPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].receivedOctetsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedUnicastPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedMulticastPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedBroadcastPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedDiscardedPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedErrorPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedTotalPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].transmittedOctetsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedUnicastPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedMulticastPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedBroadcastPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedDiscardedPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedErrorPacketsDelta
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedTotalPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedOctetsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedUnicastPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedMulticastPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedBroadcastPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedDiscardedPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
transmittedErrorPacketsAccumulated
                      - $.event.measurementsForVfScalingFields.cpuUsageArray[*].cpuIdle
                      - $.event.measurementsForVfScalingFields.cpuUsageArray[*].cpuUsageInterrupt
                      - $.event.measurementsForVfScalingFields.cpuUsageArray[*].cpuUsageNice
                      - $.event.measurementsForVfScalingFields.cpuUsageArray[*].cpuUsageSoftIrq
                      - $.event.measurementsForVfScalingFields.cpuUsageArray[*].cpuUsageSteal
                      - $.event.measurementsForVfScalingFields.cpuUsageArray[*].cpuUsageSystem
                      - $.event.measurementsForVfScalingFields.cpuUsageArray[*].cpuWait
```

```
                - $.event.measurementsForVfScalingFields.cpuUsageArray[*].percentUsage
                - $.event.measurementsForVfScalingFields.meanRequestLatency
                - $.event.measurementsForVfScalingFields.memoryUsageArray[*].memoryBuffered
                - $.event.measurementsForVfScalingFields.memoryUsageArray[*].memoryCached
                - $.event.measurementsForVfScalingFields.memoryUsageArray[*].memoryConfigured
                - $.event.measurementsForVfScalingFields.memoryUsageArray[*].memoryFree
                - $.event.measurementsForVfScalingFields.memoryUsageArray[*].memoryUsed
                - $.event.measurementsForVfScalingFields.additionalMeasurements[*].arrayOfFields[0].value
            severity:
                type: string
                required: true
                description: Threshold Event Severity
                constraints:
                  - valid_values:
                      - CRITICAL
                      - MAJOR
                      - MINOR
                      - WARNING
                      - NORMAL
            thresholdValue:
                type: integer
                required: true
                description: Threshold value for the field Path inside CEF message
            version:
                type: string
                required: true
                description: Version number associated with the threshold
```

1.2 onap.policies.controlloop.Operational Policy Type

This policy type is used to support actor/action operational policies for control loops. There are two types of implementations for this policy type

1. Existing Drools implementations that supports runtime Control Loop actions taken on components such as SO/APPC/VFC/SDNC/SDNR
2. New implementations using APEX to support Control Loops.

For Dublin, this policy type will ONLY be used for the Policy Framework to distinguish the policy type as operational. The contents are still TBD for El Alto.

Base Policy type definition for onap.policies.controlloop.Operational

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
policy_types:
  onap.policies.controlloop.Operational:
    derived_from: tosca.policies.Root
    version: 1.0.0
    description: Operational Policy for Control Loops
```

Applications should use the following Content-Type when creating onap.policies.controlloop.Operational policies:

Content-Type: "application/yaml; vnd.onap.operational"

1.2.1 Operational Policy Type Schema for Drools

For Dublin Drools will still support the Casablanca YAML definition of an Operational Policy for Control Loops.

Please use the Casablanca version of the YAML Operational Policy format defined https://git.onap.org/policy/drools-applications/tree/controlloop/common/policy-yaml/README-v2.0.0.md.

1.2.3 Operational Policy Type Schema for APEX (El Alto proposal)

The operational Policy Type schema for for APEX will extend the base operational Policy Type schema. This Policy Type allows parameters specific to the APEX PDP to be specified as a TOSCA policy.

Operational Policy Model Parameter Schema for APEX

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
# Note: The full APEX PolicyType definition will be developed during the Dublin
# timeframe of the ONAP project
policy_types:
  onap.policies.controlloop.Operational:
    derived_from: tosca.policies.Root
    version: 1.0.0
    description: Operational Policy for Control Loops

 onap.policies.controloop.operational.Apex:
    derived_from: onap.policies.controlloop.Operational
    version: 1.0.0
    description: Operational Policy for Control Loops using the APEX PDP
    properties:
      # Some of these properties may be redundant in a Kubernetes deployment
      engine_service:
        type: onap.datatypes.policies.controlloop.operational.apex.EngineService
        description: APEX Engine Service Parameters
      inputs:
        type: map
        description: Inputs for handling events coming into the APEX engine
        entry_schema:
          type: onap.datatypes.policies.controlloop.operational.apex.EventHandler
      outputs:
        type: map
        description: Outputs for handling events going out of the APEX engine
        entry_schema:
          type: onap.datatypes.policies.controlloop.operational.apex.EventHandler
      environment:
        type: list
        description: Envioronmental parameters for the APEX engine
        entry_schema:
          type: onap.datatypes.policies.controlloop.operational.apex.Environment

data_types:
  onap.datatypes.policies.controlloop.operational.apex.EngineService:
  derived_from: tosca.datatypes.Root
  properties:
    name:
      type: string
      description: Specifies the engine name
      required: false
      default: "ApexEngineService"
    version:
      type: string
      description: Specifies the engine version in double dotted format
      required: false
      default: "1.0.0"
    id:
      type: int
      description: Specifies the engine id
      required: true
    instance_count:
      type: int
      description: Specifies the number of engine threads that should be run
      required: true
    deployment_port:
      type: int
      description: Specifies the port to connect to for engine administration
      required: false
      default: 1
    policy_model_file_name:
      type: string
      description: The name of the file from which to read the APEX policy model
```

```
    required: false
    default: ""
  policy_type_impl:
    type: string
    description: The policy type implementation from which to read the APEX policy model
    required: false
    default: ""
  periodic_event_period:
    type: string
    description: The time interval in milliseconds for the periodic scanning
     event, 0 means "don't scan"
    required: false
    default: 0
  engine:
    type: onap.datatypes.policies.controlloop.operational.apex.engineservice.Engine
    description: The parameters for all engines in the APEX engine service
    required: true

onap.datatypes.policies.controlloop.operational.apex.EventHandler:
derived_from: tosca.datatypes.Root
properties:
  name:
    type: string
    description: Specifies the event handler name, if not specified this is set to
     the key name
    required: false
  carrier_technology:
    type: onap.datatypes.policies.controlloop.operational.apex.CarrierTechnology
    description: Specifies the carrier technology of the event handler (such
     as REST/Web Socket/Kafka)
    required: true
  event_protocol:
    type: onap.datatypes.policies.controlloop.operational.apex.EventProtocol
    description: Specifies the event protocol of events for the event handler
     (such as Yaml/JSON/XML/POJO)
    required: true
  event_name:
    type: string
    description: Specifies the event name for events on this event handler, if
     not specified, the event name is read from or written to the event being
     received or sent
    required: false
  event_name_filter:
    type: string
    description: Specifies a filter as a regular expression, events that do
     not match the filter are dropped, the default is to let all events
     through
    required: false
  synchronous_mode:
    type: bool
    description: Specifies the event handler is syncronous (receive event and
     send response)
    required: false
    default: false
  synchronous_peer:
    type: string
    description: The peer event handler (output for input or input for output)
     of this event handler in synchronous mode, this parameter is mandatory if
     the event handler is in synchronous mode
    required: false
    default: ""
  synchronous_timeout:
    type: int
    description: The timeout in milliseconds for responses to be issued by
     APEX torequests, this parameter is mandatory if the event handler is in
     synchronous mode
    required: false
    default: ""
  requestor_mode:
    type: bool
    description: Specifies the event handler is in requestor mode (send event
```

```
     and wait for response mode)
    required: false
    default: false
  requestor_peer:
    type: string
    description: The peer event handler (output for input or input for output)
     of this event handler in requestor mode, this parameter is mandatory if
     the event handler is in requestor mode
    required: false
    default: ""
  requestor_timeout:
    type: int
    description: The timeout in milliseconds for wait for responses to
     requests, this parameter is mandatory if the event handler is in
     requestor mode
    required: false
    default: ""

onap.datatypes.policies.controlloop.operational.apex.CarrierTechnology:
derived_from: tosca.datatypes.Root
properties:
  label:
    type: string
    description: The label (name) of the carrier technology (such as REST,
     Kafka, WebSocket)
    required: true
  plugin_parameter_class_name:
    type: string
    description: The class name of the class that overrides default handling
      of event input or output for this carrier technology, defaults to the supplied
      input or output class
    required: false

onap.datatypes.policies.controlloop.operational.apex.EventProtocol:
derived_from: tosca.datatypes.Root
properties:
  label:
    type: string
    description: The label (name) of the event protocol (such as Yaml,
     JSON, XML, or POJO)
    required: true
  event_protocol_plugin_class:
    type: string
    description: The class name of the class that overrides default handling
      of the event protocol for this carrier technology, defaults to the
      supplied event protocol class
    required: false

onap.datatypes.policies.controlloop.operational.apex.Environmental:
derived_from: tosca.datatypes.Root
properties:
  name:
    type: string
    description: The name of the environment variable
    required: true
  value:
    type: string
    description: The value of the environment variable
    required: true

onap.datatypes.policies.controlloop.operational.apex.engineservice.Engine:
derived_from: tosca.datatypes.Root
properties:
  context:
    type: onap.datatypes.policies.controlloop.operational.apex.engineservice.engine.Context
    description: The properties for handling context in APEX engines,
     defaults to using Java maps for context
    required: false
  executors:
    type: map
    description: The plugins for policy executors used in engines such as
```

```
      javascript, MVEL, Jython
    required: true
    entry_schema:
      description: The plugin class path for this policy executor
      type: string

onap.datatypes.policies.controlloop.operational.apex.engineservice.engine.Context:
derived_from: tosca.datatypes.Root
properties:
  distributor:
    type: onap.datatypes.policies.controlloop.operational.apex.Plugin
    description: The plugin to be used for distributing context between
     APEX PDPs at runtime
    required: false
  schemas:
    type: map
    description: The plugins for context schemas available in APEX PDPs
     such as Java and Avro
    required: false
    entry_schema:
      type: onap.datatypes.policies.controlloop.operational.apex.Plugin
  locking:
    type: onap.datatypes.policies.controlloop.operational.apex.plugin
    description: The plugin to be used for locking context in and
     between APEX PDPs at runtime
    required: false
  persistence:
    type: onap.datatypes.policies.controlloop.operational.apex.Plugin
    description: The plugin to be used for persisting context for APEX PDPs
     at runtime
    required: false

onap.datatypes.policies.controlloop.operational.apex.Plugin:
derived_from: tosca.datatypes.Root
properties:
  name:
    type: string
    description: The name of the executor such as Javascript, Jython or MVEL
    required: true
  plugin_class_name:
    type: string
    description: The class path of the plugin class for this executor
```

1.3 onap.policies.controlloop.Guard Policy Type

This policy type is the the type definition for Control Loop guard policies for frequency limiting, blacklisting and min/max guards to help protect runtime Control Loop Actions from doing harm to the network. This policy type is developed using the XACML PDP to support question/answer Policy Decisions during runtime for the Drools and APEX onap.controlloop.Operational policy type implementations.

The base schema is defined as below:

Base Policy type definition for onap.policies.controlloop.Guard

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
policy_types:
  - onap.policies.controlloop.Guard:
      derived_from: tosca.policies.Root
      version: 1.0.0
      description: Guard Policies for Control Loop Operational Policies
```

As with *onap.policies.Monitoring* policy type, the *PolicyTypeImpl* implementation of the *onap.policies.controlloop.Guard* Policy Type is generic to support definition of TOSCA *PolicyType* artifacts in the Policy Framework using the Policy Type Design API.

For Dublin, only the following derived Policy Type definitions below are preloaded in the Policy Framework. However, the creation of policies will still support the payload from Casablanca.

Casablanca Guard Payload

```
ContentType: "application/json; vnd.onap.guard"
Accepts: "application/json"


#
# Request BODY
#
{
  "policy-id" : "guard.frequency.scaleout",
  "contents" : {
            "actor": "SO",
            "recipe": "scaleOut",
            "targets": ".*",
            "clname": "ControlLoop-vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3",
            "limit": "1",
            "timeWindow": "10",
            "timeUnits": "minute",
            "guardActiveStart": "00:00:01-05:00",
            "guardActiveEnd": "23:59:59-05:00"
   }
}

#
# Request RESPONSE
#
{
    "guard.frequency.scaleout": {
        "type": "onap.policies.controlloop.guard.FrequencyLimiter",
        "version": "1.0.0",
        "metadata": {
            "policy-id": "guard.frequency.scaleout",
            "policy-version": 1
        }
    }
}
```

1.3.1 onap.policies.controlloop.guard.FrequencyLimiter Policy Type

This is WIP for El Alto for the proposed policy type.

Policy Typefor Frequency Limiter Guard Policy

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
policy_types:
  - onap.policies.controlloop.Guard:
      derived_from: tosca.policies.Root
      version: 1.0.0
      description: Guard Policies for Control Loop Operational Policies
  - onap.policies.controlloop.guard.FrequencyLimiter:
      derived_from: onap.policies.controlloop.Guard
      version: 1.0.0
      description: Supports limiting the frequency of actions being taken by a Actor.
      properties:
        frequency_policy:
          type: map
          description:
          entry_schema:
            type: onap.datatypes.guard.FrequencyLimiter
data_types:
  - onap.datatypes.guard.FrequencyLimiter:
      derived_from: tosca.datatypes.Root
      properties:
        actor:
          type: string
          description: Specifies the Actor
          required: true
        recipe:
          type: string
          description: Specified the Recipe
          required: true
        time_window:
          type: scalar-unit.time
          description: The time window to count the actions against.
          required: true
        limit:
          type: integer
          description: The limit
          required: true
          constraints:
            - greater_than: 0
        time_range:
          type: tosca.datatypes.TimeInterval
          description: An optional range of time during the day the frequency is valid for.
          required: false
        controlLoopName:
          type: string
          description: An optional specific control loop to apply this guard to.
          required: false
        target:
          type: string
          description: An optional specific VNF to apply this guard to.
          required: false
```

1.3.2 onap.policies.controlloop.guard.Blacklist Policy Type

Policy Type for Blacklist Guard Policies

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
policy_types:
 - onap.policies.controlloop.Guard:
     derived_from: tosca.policies.Root
     version: 1.0.0
     description: Guard Policies for Control Loop Operational Policies
  - onap.policies.controlloop.guard.Blacklist:
      derived_from: onap.policies.controlloop.Guard
      version: 1.0.0
      description: Supports blacklist of VNF's from performing control loop actions on.
      properties:
        blacklist_policy:
          type: map
          description:
          entry_schema:
            type: onap.datatypes.guard.Blacklist
data_types:
  - onap.datatypes.guard.Blacklist:
      derived_from: tosca.datatypes.Root
      properties:
        actor:
          type: string
          description: Specifies the Actor
          required: true
        recipe:
          type: string
          description: Specified the Recipe
          required: true
        time_range:
          type: tosca.datatypes.TimeInterval
          description: An optional range of time during the day the blacklist is valid for.
          required: false
        controlLoopName:
          type: string
          description: An optional specific control loop to apply this guard to.
          required: false
        blacklist:
          type: list
          description: List of VNF's
          required: true
```

1.3.3 onap.policies.controlloop.guard.MinMax Policy Type

Policy Type for Min/Max VF Module Policies

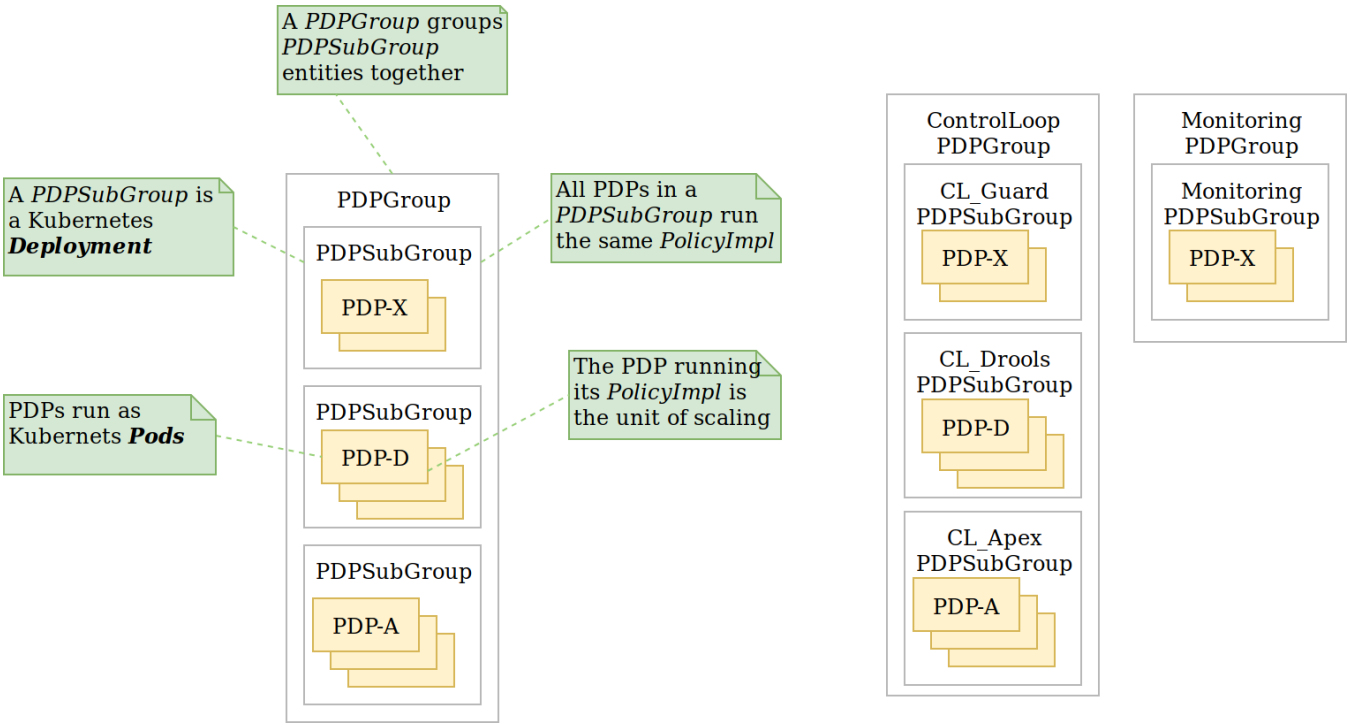
```
policy_types:
  - onap.policies.controlloop.Guard:
      derived_from: tosca.policies.Root
      version: 1.0.0
      description: Guard Policies for Control Loop Operational Policies
  - onap.policies.controlloop.guard.MinMax:
      derived_from: onap.policies.controlloop.Guard
      version: 1.0.0
      description: Supports Min/Max number of VF Modules
      properties:
        minmax_policy:
          type: map
          description:
          entry_schema:
            type: onap.datatypes.guard.MinMax
data_types:
  - onap.datatypes.guard.MinMax:
      derived_from: tosca.datatypes.Root
      properties:
        actor:
          type: string
          description: Specifies the Actor
          required: true
        recipe:
          type: string
          description: Specified the Recipe
          required: true
        time_range:
          type: tosca.datatypes.TimeInterval
          description: An optional range of time during the day the Min/Max limit is valid for.
          required: false
        controlLoopName:
          type: string
          description: An optional specific control loop to apply this guard to.
          required: false
        min_vf_module_instances:
          type: integer
          required: true
          description: The minimum instances of this VF-Module

        max_vf_module_instances:
          type: integer
          required: false
          description: The maximum instances of this VF-Module
```

1.3.4 onap.policies.controlloop.Coordination Policy Type (STRETCH)

This policy type defines Control Loop Coordination policies to assist in coordinating multiple control loops during runtime. This policy type is developed using XACML PDP to support question/answer policy decisions at runtime for the onap.policies.controlloop.operational policy types.

2 PDP Deployment and Registration with PAP

The unit of execution and scaling in the Policy Framework is a *PolicyImpl* entity. A *PolicyImpl* entity runs on a PDP. As is explained above a *PolicyImpl* entity is a *PolicyTypeImpl* implementation parameterized with a TOSCA *Policy*.



In order to achieve horizontal scalability, we group the PDPs running instances of a given *PolicyImpl* entity logically together into a *PDPSubGroup*. The number of PDPs in a *PDPSubGroup* can then be scaled up and down using Kubernetes. In other words, all PDPs in a subgroup run the same *PolicyImpl*, that is the same policy template implementation (in XACML, Drools, or APEX) with the same parameters.

The figure above shows the layout of *PDPGroup* and *PDPSubGroup* entities. The figure shows examples of PDP groups for Control Loop and Monitoring policies on the right.

The health of PDPs is monitored by the PAP in order to alert operations teams managing policy. The PAP manages the life cycle of policies running on PDPs.

The table below shows the methods in which *PolicyImpl* entities can be deployed to PDP Subgroups

Method	Description	Advantages	Disadvantages
Cold Deployment	The *PolicyImpl (PolicyTypeImpl* and TOSCA *Policy)* are predeployed on the PDP. The PDP is fully configured and ready to execute when started. PDPs register with the PAP when they start, providing the *PolicyImpl* they have been predeployed with.	No run time configuration required and run time administration is simple.	Very restrictive, no run time configuration of PDPs is possible.
Warm Deployment	The *PolicyTypeImpl* entity is predeployed on the PDP. A TOSCA *Policy* may be loaded at startup. The PDP may be configured or reconfigured with a new or updated TOSCA *Policy* at run time. PDPs register with the PAP when they start, providing the *PolicyImpl* they have been predeployed with if any. The PAP may update the TOSCA *Policy* on a PDP at any time after registration.	The configuration, parameters, and PDP group of PDPs may be changed at run time by loading or updating a TOSCA *Policy* into the PDP. Lifecycle management of TOSCA *Policy* entities is supported, allowing features such as *PolicyImpl* Safe Mode and *Policy*Impl retirement.	Administration and management is required. The configuration and life cycle of the TOSCA policies can change at run time and must be administered and managed.
Hot Deployment	The *PolicyImpl (PolicyTypeImpl* and TOSCA *Policy)* are deployed at run time. The *PolicyImpl (PolicyTypeImpl* and TOSCA *Policy)* may be loaded at startup. The PDP may be configured or reconfigured with a new or updated *PolicyTypeImpl* and/or TOSCA *Policy* at run time. PDPs register with the PAP when they start, providing the *PolicyImpl* they have been predeployed with if any. The PAP may update the TOSCA *Policy* and *PolicyTypeImpl* on a PDP at any time after registration.	The policy logic, rules, configuration, parameters, and PDP group of PDPs may be changed at run time by loading or updating a TOSCA *Policy* and *PolicyTypeImpl* into the PDP. Lifecycle management of TOSCA *Policy* entities and *PolicyTypeImpl* entites is supported, allowing features such as *PolicyImpl* Safe Mode and *Policy*Impl retirement.	Administration and management is more complex. The *PolicyImpl* itself and its configuration and life cycle as well as the life cycle of the TOSCA policies can change at run time and must be administered and managed.

3. Public APIs

The Policy Framework supports the APIs documented in the subsections below. The APIs in this section are supported for use by external components.

3.1 Policy Type Design API for TOSCA Policy Types

The purpose of this API is to support CRUD of TOSCA *PolicyType* entities. This API is provided by the *PolicyDevelopment* component of the Policy Framework, see The ONAP Policy Framework architecture.

The API allows applications to create, update, delete, and query *PolicyType* entities so that they become available for use in ONAP by applications such as CLAMP. Some Policy Type entities are preloaded in the Policy Framework. The TOSCA fields below are valid on API calls:

Field	GET	POST	DELETE	Comment
(name)	M	M	M	The definition of the reference to the Policy Type, GET allows ranges to be specified
version	O	M	C	GET allows ranges to be specified, must be specified if more than one version of the Policy Type exists
description	R	O	N/A	Desciption of the Policy Type
derived_from	R	C	N/A	Must be specified when a Policy Type is derived from another Policy Type such as in the case of derived Monitoring Policy Types
metadata	R	O	N/A	Metadata for the Policy Type
properties	R	M	N/A	This field holds the specification of the specific Policy Type in ONAP
targets	R	O	N/A	A list of node types and/or group types to which the Policy Type can be applied
triggers	R	O	N/A	Specification of policy triggers, not currently supported in ONAP

Note: On this and subsequent tables, use the following legend: M-Mandatory, O-Optional, R-Read-only, C-Conditional. Conditional means the field is mandatory when some other field is present.
Note: Preloaded policy types may only be queried over this API, modification or deletion of preloaded policy type implementations is disabled.
Note: Policy types that are in use (referenced by defined Policies) may not be deleted
Note: The group types of targets in TOSCA are groups of TOSCA nodes, not PDP groups; the *target* concept in TOSCA is equivalent to the Policy Enforcement Point (PEP) concept

3.1.1 Policy Type query

The API allows applications (such as CLAMP and Integration) to query the *PolicyType* entities that are available for *Policy* creation using a GET operation.

https:{url}:{port}/policy/api/v1/policytypes GET

Policy Type Query - When system comes up before any mS are onboarded

```
policy_types:
  - onap.policies.Monitoring:
      version: 1.0.0
      description: A base policy type for all policies that govern monitoring provision
      derived_from: tosca.policies.Root
      properties:
        # Omitted for brevity, see Section 1

 - onap.policies.controlloop.Operational:
      version: 1.0.0
      description: Operational Policy for Control Loops
      derived_from: tosca.policies.Root
      properties:
        # Omitted for brevity, see Section 1

  - onap.policies.controloop.operational.Drools:
      version: 1.0.0
      description: Operational Policy for Control Loops using the Drools PDP
      derived_from: onap.policies.controlloop.Operational
      properties:
        # Omitted for brevity, see Section 1

  - onap.policies.controloop.operational.Apex:
      version: 1.0.0
      description: Operational Policy for Control Loops using the APEX PDP
      derived_from: onap.policies.controlloop.Operational
      properties:
        # Omitted for brevity, see Section 1

 - onap.policies.controlloop.Guard:
      version: 1.0.0
      description: Operational Policy for Control Loops
      derived_from: tosca.policies.Root
      properties:
        # Omitted for brevity, see Section 1

  - onap.policies.controlloop.guard.FrequencyLimiter:
      version: 1.0.0
      description: Supports limiting the frequency of actions being taken by a Actor.
      derived_from: onap.policies.controlloop.Guard
      properties:
        # Omitted for brevity, see Section 1

  - onap.policies.controlloop.guard.Blacklist:
      version: 1.0.0
      description: Supports blacklist of VNF's from performing control loop actions on.
      derived_from: onap.policies.controlloop.Guard
      properties:
        # Omitted for brevity, see Section 1

  - onap.policies.controlloop.guard.MinMax:
      version: 1.0.0
      description: Supports Min/Max number of VF Modules
      derived_from: onap.policies.controlloop.Guard
      properties:
        # Omitted for brevity, see Section 1

  - onap.policies.controlloop.coordination.TBD: (STRETCH GOALS)
      version: 1.0.0
      description: Control Loop Coordination policy types
      derived_from: onap.policies.controlloop.Coordination
      properties:
        # Omitted for brevity, see Section 1

data_types:
  # Any bespoke data types referenced by policy type definitions
```

The table below shows some more examples of GET operations

Example	Description
https:{url}:{port}/policy/api/v1/policytypes	Get all Policy Type entities in the system
https:{url}:{port}/policy/api/v1/policytypes/{policy type id} *eg.* *https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app*	Get a specific policy type and all the available versions.
https:{url}:{port}/policy/api/v1/policytypes/{policy type id}/versions/{version id} *eg.* *https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app* */versions/1.0.0*	Get the specific Policy Type with the specified name and version

3.1.2 Policy Type Create/Update

The API allows applications and users (such as a DCAE microservice component developer) to create or update a Policy Type using a POST operation. This API allows new Policy Types to be created or existing Policy Types to be modified. POST operations with a new Policy Type name or a new version of an existing Policy Type name are used to create a new Policy Type. POST operations with an existing Policy Type name and version are used to update an existing Policy Type. Many Policy Types can be created or updated in a single POST operation by specifying more than one Policy Type on the TOSCA *policy_types* list.

For example, the POST operation below with the TOSCA body below is used to create a new Policy type for a DCAE microservice.

https:{url}:{port}/policy/api/v1/policytypes POST

Create a new Policy Type for a DCAE microservice

```
policy_types:
  - onap.policies.monitoring.cdap.tca.hi.lo.app:
      version: 1.0.0
      derived_from: onap.policies.Monitoring
      description: A DCAE TCA high/low policy type
      properties:
        tca_policy:
          type: map
          description: TCA Policy JSON
          default:'{<JSON omitted for brevity>}'
          entry_schema:
            type: onap.datatypes.monitoring.tca_policy

data_types:
  <omitted for brevity>
```

Following creation of a DCAE TCA policy type operation, the GET call for Monitoring policies will list the new policy type.

https:{url}:{port}/policy/api/v1/policytypes GET

Policy Type Query after DCAE TCA mS Policy Type is created

```
policy_types:
  - onap.policies.Monitoring:
      version: 1.0.0
      derived_from: tosca.policies.Root
      description: A base policy type for all policies that govern monitoring provision

  - onap.policies.monitoring.cdap.tca.hi.lo.app:
      version: 1.0.0
      derived_from: onap.policies.Monitoring
      description: A DCAE TCA high/low policy type

  - onap.policies.controlloop.Operational:
      version: 1.0.0
      description: Operational Policy for Control Loops
      derived_from: tosca.policies.Root

  - onap.policies.controloop.operational.Drools:
      version: 1.0.0
      description: Operational Policy for Control Loops using the Drools PDP
      derived_from: onap.policies.controlloop.Operational

  - onap.policies.controloop.operational.Apex:
      version: 1.0.0
      description: Operational Policy for Control Loops using the APEX PDP
      derived_from: onap.policies.controlloop.Operational

 - onap.policies.controlloop.Guard:
      version: 1.0.0
      description: Operational Policy for Control Loops
      derived_from: tosca.policies.Root

  - onap.policies.controlloop.guard.FrequencyLimiter:
      version: 1.0.0
      description: Supports limiting the frequency of actions being taken by a Actor.
      derived_from: onap.policies.controlloop.Guard

  - onap.policies.controlloop.guard.Blacklist:
      version: 1.0.0
      description: Supports blacklist of VNF's from performing control loop actions on.
      derived_from: onap.policies.controlloop.Guard

  - onap.policies.controlloop.guard.MinMax:
      version: 1.0.0
      description: Supports Min/Max number of VF Modules
      derived_from: onap.policies.controlloop.Guard

  - onap.policies.controlloop.coordination.TBD: (STRETCH GOALS)
      version: 1.0.0
      description: Control Loop Coordination policy types
      derived_from: onap.policies.controlloop.Coordination
```

Now the *onap.policies.Monitoring.cdap.tca.hi.lo.app* Policy Type is available to CLAMP for creating concrete policies. See the Yaml contribution on the Model driven Control Loop Design page for a full listing of the DCAE TCA policy type used in the example above.

3.1.3 Policy Type Delete

The API also allows Policy Types to be deleted with a DELETE operation. The format of the delete operation is as below:

https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.0.0 DELETE

Note: Predefined policy types cannot be deleted
Note: Policy types that are in use (Parameterized by a TOSCA Policy) may not be deleted, the parameterizing TOSCA policies must be deleted first
Note: The *version* parameter may be omitted on the DELETE operation if there is only one version of the policy type in the system

3.2 Policy Design API

The purpose of this API is to support CRUD of TOSCA *Policy* entities from TOSCA compliant *PolicyType* definitions. TOSCA *Policy* entities become the parameters for *PolicyTypeImpl* entities, producing *PolicyImpl* entities that can run on PDPs. This API is provided by the *PolicyDevelopment* component of the Policy Framework, see The ONAP Policy Framework architecture.

This API allows applications (such as CLAMP and Integration) to create, update, delete, and query *Policy* entities. The TOSCA fields below are valid on API calls:

Field	GET	POST	DELETE	Comment
(name)	M	M	M	The definition of the reference to the Policy, GET allows ranges to be specified
type	O	M	O	The Policy Type of the policy, see section 3.1
description	R	O	O	
metadata	R	O	N/A	
properties	R	M	N/A	This field holds the specification of the specific Policy in ONAP
targets	R	O	N/A	A list of nodes and/or groups to which the Policy can be applied

Note: Policies that are deployed (used on deployed *PolicyImpl* entities) may not be deleted
Note: This API is NOT used by DCAE for a decision on what policy the DCAE PolicyHandler should retrieve and enforce
Note: The groups of targets in TOSCA are groups of TOSCA nodes, not PDP groups; the *target* concept in TOSCA is equivalent to the Policy Enforcement Point (PEP) concept

YAML is used for illustrative purposes in the examples in this section. JSON (application/json) will be used as the content type in the implementation of this API.

3.2.1 Policy query

The API allows applications (such as CLAMP and Integration) to query the *Policy* entities that are available for deployment using a GET operation.

Note: This operation simply returns TOSCA policies that are defined in the Policy Framework, it does NOT make a decision.

The table below shows some more examples of GET operations

Example	Description
https:{url}:{port}/policy/api/v1/policytypes/{policy type id}/versions/{versions}/policies *eg.* *https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.0.0 /policies*	Get all Policies for a specific Policy Type and version
https://{url}:{port}/policy/api/v1/policytypes/{policy type id}/versions/{version}/policies/{policy name} /versions/{version} *eg.* *https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.0.0 /policies/onap.scaleout.tca/versions/1.0.0 GET*	Gets a specific Policy version
https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.0.0 /policies/onap.scaleout.tca/versions/latest GET	Returns the latest version of a Policy
https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.0.0 /policies/onap.scaleout.tca/deployed GET	Returns the version of the Policy that has been deployed on one or more PDP groups.
https://{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.2.3 /policies/CL-LBAL-LOW-TRAFFIC-SIG-FB480F95-A453-6F24-B767-FD703241AB1A/versions/1.0.2 GET	Returns a specific version of a monitoring policy

3.2.2 Policy Create/Update

The API allows applications and users (such as CLAMP and Integration) to create or update a Policy using a POST operation. This API allows new Policies to be created or existing Policies to be modified. POST operations with a new Policy name are used to create a new Policy. POST operations with an existing Policy name are used to update an existing Policy. Many Policies can be created or updated in a single POST operation by specifying more than one Policy on the TOSCA *policies* list.

3.2.2.1 Monitoring Policy Create/Update

While designing a control loop using CLAMP, a Control Loop Designer uses the Policy Type for a specific DCAE mS component (See Section 3.1.1) to create a specific Policy. CLAMP then uses this API operation to submit the Policy to the Policy Framework.

For example, the POST operation below with the TOSCA body below is used to create a new scaleout Policy for the *onap.policies.monitoring.cdap.tca.hi.lo. app* microservice. The name of the policy "onap.scaleout.tca" is up to the user to determine themselves.

https:{url}:{port}/policy/api/v1/policytypes/onap.policies.Monitoring.cdap.tca.hi.lo.app/versions/1.0.0/policies POST

TOSCA Body of a new TCA High/Low Policy

```
https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.0.0
/policies POST
Content-Type: application/yaml
Accept: application/yaml


#Request Body
policies:
  -
    onap.scaleout.tca:
        type: onap.policies.monitoring.cdap.tca.hi.lo.app
        version: 1.0.0
        metadata:
          policy-id: onap.scaleout.tca # SHOULD MATCH THE TOSCA policy-name field above. DCAE needs this -
convenience.
          description: The scaleout policy for vDNS # GOOD FOR CLAMP GUI
        properties:
          domain: measurementsForVfScaling
          metricsPerEventName:
            -
              eventName: vLoadBalancer
              controlLoopSchemaType: VNF
              policyScope: "type=configuration"
              policyName: "onap.scaleout.tca"
              policyVersion: "v0.0.1"
              thresholds:
                - closedLoopControlName: "CL-LBAL-LOW-TRAFFIC-SIG-FB480F95-A453-6F24-B767-FD703241AB1A"
                  closedLoopEventStatus: ONSET
                  version: "1.0.2"
                  fieldPath: "$.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedBroadcastPacketsAccumulated"
                  thresholdValue: 500
                  direction: LESS_OR_EQUAL
                  severity: MAJOR
                -
                  closedLoopControlName: "CL-LBAL-LOW-TRAFFIC-SIG-0C5920A6-B564-8035-C878-0E814352BC2B"
                  closedLoopEventStatus: ONSET
                  version: "1.0.2"
                  fieldPath: "$.event.measurementsForVfScalingFields.vNicPerformanceArray[*].
receivedBroadcastPacketsAccumulated"
                  thresholdValue: 5000
                  direction: GREATER_OR_EQUAL
                  severity: CRITICAL


#Response Body
policies:
  - onap.scaleout.tca:
      type: onap.policies.monitoring.cdap.tca.hi.lo.app
      version: 1.0.0
      metadata:
        #

        # version is managed by Policy Lifecycle and returned
        # back to the caller.
        #
        policy-version: 1
        #
        # These were passed in, and should not be changed. Will
        # be passed back.
        #
        policy-id: onap.scaleout.tca
      properties:
        domain: measurementsForVfScaling
        metricsPerEventName:
          -
            eventName: vLoadBalancer
            controlLoopSchemaType: VNF
            policyScope: "type=configuration"
            <OMITTED FOR BREVITY>
```

Given a return code of success and a "metadata" section that indicates versioning information. The "metadata" section conforms exactly to how SDC implements lifecycle management versioning for first class normatives in the TOSCA Models. The policy platform will implement lifecycle identically to SDC to ensure conformity for policy creation. The new metadata fields return versioning details.

The following new policy will be listed and will have a "metadata" section as shown below:

https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.0.0/policies GET

Policy with Metadata section for lifecycle management

```
policies:
  - onap.scaleout.tca:
      type: onap.policies.monitoring.cdap.tca.hi.lo.app
      version: 1.0.0
      metadata:
        policy-id: onap.scaleout.tca

        policy-version: 1
  - my.other.policy:
      type: onap.policies.monitoring.cdap.tca.hi.lo.app
      version: 1.0.0
      metadata:
        invariantUUID: 20ad46cc-6b16-4404-9895-93d2baaa8d25
        UUID: 4f715117-08b9-4221-9d63-f3fa86919742
        version: 5
        name: my.other.policy
        scope: foo=bar;field2=value2
        description: The policy for some other use case
  - yet.another.policy:
      type: onap.policies.monitoring.cdap.tca.hi.lo.app
      version: 1.0.0
      metadata:
        invariantUUID: 20ad46cc-6b16-4404-9895-93d2baaa8d25
        UUID: 4f715117-08b9-4221-9d63-f3fa86919742
        version: 3
        name: yet.another.policy
        scope: foo=bar;
        description: The policy for yet another use case
```

The contents of the new policy can be retrieved using the ID:

https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.0.0/policies/onap.scaleout.tca GET

Query on a new TCA High/Low Policy

```
policies:
 -
    onap.scaleout.tca:
      type: onap.policies.monitoring.cdap.tca.hi.lo.app
      version: 1.0.0
      metadata:
        invariantUUID: 20ad46cc-6b16-4404-9895-93d2baaa8d25
        UUID: 4f715117-08b9-4221-9d63-f3fa86919742
        version: 1
        name: onap.scaleout.tca
        scope: foo=bar;
        description: The scaleout policy for vDNS
      properties:
        domain: measurementsForVfScaling
        <OMMITTED FOR BREVITY>
```

3.2.2.2 Operational Policy Create/Update

While designing an operational policy, the designer uses the Policy Type for the operational policy (See Section 3.1.1) to create a specific Policy and submits the Policy to the Policy Framework.

This URL will be fixed for CLAMP in Dublin and the payload will match updated version of Casablanca YAML that supports VFModules.

https:{url}:{port}/policy/api/v1/policytypes/onap.policies.controloop.operational/versions/1.0.0/policies POST

Content-Type: application/yaml; legacy-version

FUTURE: Content-Type: application/yaml; tosca

NOTE: The controlLoopName will be assumed to be the policy-id

Create an Operational Policy

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
topology_template:
   policies:
      -
        operational.scaleout:
           type: onap.policies.controlloop.Operational
           version: 1.0.0
           metadata:
             policy-id: operational.scaleout
           properties:
           controlLoop:
             version: 2.0.0
             controlLoopName: ControlLoop-vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3
             trigger_policy: unique-policy-id-1-scale-up
             timeout: 1200
             abatement: false
           policies:
             - id: unique-policy-id-1-scale-up
               name: Create a new VF Module
               description:
               actor: SO
               recipe: VF Module Create
               target:
                 type: VNF
               payload:
                 requestParameters: '{"usePreload":true,"userParams":[]}'
                 configurationParameters: '[{"ip-addr":"$.vf-module-topology.vf-module-parameters.param[9]","
oam-ip-addr":"$.vf-module-topology.vf-module-parameters.param[16]","enabled":"$.vf-module-topology.vf-module-
parameters.param[23]"}]'
               retry: 0
               timeout: 1200
               success: final_success
               failure: final_failure
               failure_timeout: final_failure_timeout
               failure_retries: final_failure_retries
               failure_exception: final_failure_exception
               failure_guard: final_failure_guard
```

Response from creating Operational Policy

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
topology_template:
   policies:
      -
        operational.scaleout:
           type: onap.policies.controlloop.Operational
           version: 1.0.0
           metadata:
             policy-id: operational.scaleout
             policy-version: 1
           properties:
           controlLoop:
             version: 2.0.0
             controlLoopName: ControlLoop-vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3
             trigger_policy: unique-policy-id-1-scale-up
             timeout: 1200
             abatement: false
           policies:
             - id: unique-policy-id-1-scale-up
               name: Create a new VF Module
               description:
               actor: SO
               recipe: VF Module Create
               target:
                 type: VNF
               payload:
                 requestParameters: '{"usePreload":true,"userParams":[]}'
                 configurationParameters: '[{"ip-addr":"$.vf-module-topology.vf-module-parameters.param[9]","
oam-ip-addr":"$.vf-module-topology.vf-module-parameters.param[16]","enabled":"$.vf-module-topology.vf-module-
parameters.param[23]"}]'
               retry: 0
               timeout: 1200
               success: final_success
               failure: final_failure
               failure_timeout: final_failure_timeout
               failure_retries: final_failure_retries
               failure_exception: final_failure_exception
               failure_guard: final_failure_guard
```

3.2.2.2.1 Drools Operational Policy Create/Update

TBD Jorge Hernandez

3.2.2.2.2 APEX Operational Policy Create/Update

The POST operation below with the TOSCA body below is used to create a new Sample Domain test polict for the APEX Sample Domain operational policy type.

https:{url}:{port}/policy/api/v1/policytypes/onap.policies.controloop.operational.apex/versions/1.0.0/policies POST

Create an APEX Policy for a Sample Domain

```
policies:
  - onap.policy.operational.apex.sampledomain.Test:
      type: onap.policies.controloop.operational.Apex
      properties:
        engine_service:
          name: "MyApexEngine"
          version: "0.0.1"
          id: 45
          instance_count: 4
          deployment_port: 12561
          policy_type_impl: "onap.policies.controlloop.operational.apex.sampledomain.Impl"
          engine:
            executors:
                JAVASCRIPT: "org.onap.policy.apex.plugins.executor.javascript.JavascriptExecutorParameters"

        inputs:
          first_consumer:
            carrier_technology:
              label: "RESTCLIENT",
              plugin_parameter_class_name: "org.onap.policy.apex.plugins.event.carrier.restclient.
RestClientCarrierTechnologyParameters",
              parameters:
                url: "https://localhost:32801/EventGenerator/GetEvents"
            event_protocol:
              label: "JSON"

        outputs:
          first_producer:
            carrier_technology:
              label: "RESTCLIENT",
              plugin_parameter_class_name: "org.onap.policy.apex.plugins.event.carrier.restclient.
RestClientCarrierTechnologyParameters",
              parameters:
                url: "https://localhost:32801/EventGenerator/PostEvent"
            event_protocol:
              label: "JSON"
```

3.2.2.3 Guard Policy Create/Update

TBD Pamela Dragosh Similar to Operational Policies

3.2.2.4 Policy Lifecycle API - Creating Coordination Policies

TBD Similar to Operational Policies, stretch for Dublin

3.2.3 Policy Delete

The API also allows Policies to be deleted with a DELETE operation. The format of the delete operation is as below:

Example	Description
https:{url}:{port}/policy/api/v1/policytypes/onap.policies.monitoring.cdap.tca.hi.lo.app/versions/1.0.0/policies/onap.scaleout.tca DELETE	Deletes a Policy - all versions will be deleted. NOTE: The API call will fail if the policy has been deployed in one or more PDP Group. They must be undeployed first from all PDP Groups.

3.3 Policy Administration API

The purpose of this API is to support CRUD of PDP groups and subgroups and to support the deployment and life cycles of *PolicyImpl* entities (TOSCA *Policy* and *PolicyTypeImpl* entities) on PDP sub groups and PDPs. See Section 2 for details on policy deployment on PDP groups and subgroups. This API is provided by the *PolicyAdministration* component (PAP) of the Policy Framework, see The ONAP Policy Framework architecture.

PDP groups and subgroups may be prefedined in the system. Predefined groups and subgroups may not be modified or deleted over this API. However, the policies running on predefined groups or subgroups as well as the instance counts and properties may be modified.

A PDP may be preconfigured with its PDP group, PDP subgroup, and policies. The PDP sends this information to the PAP when it starts. If the PDP group, subgroup, or any policy is unknown to the PAP, the PAP locks the PDP in state PASSIVE.

The fields below are valid on API calls:

Field	GET	POST	DELETE	Comment
name	M	M	M	The name of the PDP group
version	O	M	C	The version of the PDP group
state	R	N/A	N/A	The administrative state of the PDP group: PASSIVE, SAFE, TEST, or ACTIVE
description	R	O	N/A	The PDP group description
properties	R	O	N/A	Specific properties for a PDP group
pdp_subgroups	R	M	N/A	A list of PDP subgroups for a PDP group
pdp_type	R	M	N/A	The PDP type of this PDP subgroup, currently xacml, drools, or apex
supported_policy_types	R	N/A	N/A	A list of the policy types supported by the PDPs in this PDP subgroup
policies	R	M	N/A	The list of policies running on the PDPs in this PDP subgroup
(name)	R	M	N/A	The name of a TOSCA policy running in this PDP subgroup
policy_type	R	N/A	N/A	The TOSCA policy type of the policy
policy_type_version	R	N/A	N/A	The version of the TOSCA policy type of the policy
policy_type_impl	R	C	N/A	The policy type implementation (XACML, Drools Rules, or APEX Model) that implements the policy
instance_count	R	N/A	N/A	The number of PDP instances running in a PDP subgroup
min_instance_count	O	N/A	N/A	The minumum number of PDP instances to run in a PDP subgroup
properties	O	N/A	N/A	Deployment configuration or other properties for the PDP subgroup
deployment_info	R	N/A	N/A	Information on the deployment for a PDP subgroup
instances	R	N/A	N/A	A list of PDP instances running in a PDP subgroup
instance	R	N/A	N/A	The instance ID of a PDP running in a Kuberenetes Pod
state	R	N/A	N/A	The administrative state of the PDP: PASSIVE, SAFE, TEST, or ACTIVE
healthy	R	N/A	N/A	The result of the latest health check on the PDP: HEALTHY/NOT_HEALTHY /TEST_IN_PROGRESS
message	O	N/A	N/A	A status message for the PDP if any
deployment_instance_info	R	N/A	N/A	Information on the node running the PDP

Note: In the Dublin release, the *policy_type_impl* of all policy types in a PDP subgroup must be the same.

YAML is used for illustrative purposes in the examples in this section. JSON (application/json) will be used as the content type in the implementation of this API.

3.3.1 PDP Group Query

This operation allows the PDP groups and subgroups to be listed together with the policies that are deployed on each PDP group and subgroup.

https:{url}:{port}/policy/pap/v1/pdps GET

PDP Group query for all PDP groups and Subgroups

```
pdp_groups:
  - name: onap.pdpgroup.controlloop.Operational
    version: 1.0.0
    state: active
    description: ONAP Control Loop Operational and Guard policies
    properties:
      # PDP group level properties if any
    pdp_subgroups:
      pdp_type: drools
        supported_policy_types:
          - onap.controllloop.operational.drools.vCPE
          - onap.controllloop.operational.drools.vFW
        policies:
```

```
    - onap.controllloop.operational.drools.vCPE.eastRegion:
        policy_type: onap.controllloop.operational.drools.vCPE
        policy_type_version: 1.0.0
        policy_type_impl: onap.controllloop.operational.drools.impl
    - onap.controllloop.operational.drools.vFW.eastRegion:
        policy_type: onap.controllloop.operational.drools.vFW
        policy_type_version: 1.0.0
        policy_type_impl: onap.controllloop.operational.drools.impl
  min_instance_count: 3
  instance_count: 3
  properties:
    # The properties below are for illustration only
    instance_spawn_load_threshold: 70%
    instance_kill_load_threshold: 50%
    instance_geo_redundancy: true
 deployment_info:
    service_endpoint: https://<the drools service endpoint for this PDP group>
    deployment: A deployment identifier
      # Other deployment info
  instances:
    - instance: drools_1
      state: active
      healthy: yes
      deployment_instance_info:
        node_address: drools_1_pod
        # Other deployment instance info
    - instance: drools_2
      state: active
      healthy: yes
      deployment_instance_info:
        node_address: drools_2_pod
        # Other deployment instance info
    - instance: drools_3
      state: active
      healthy: yes
      deployment_instance_info:
        node_address: drools_3_pod
        # Other deployment instance info

- pdp_type: apex
    supported_policy_types:
      - onap.controllloop.operational.apex.BBS
      - onap.controllloop.operational.apex.SampleDomain
    policies:
      - onap.controllloop.operational.apex.BBS.eastRegion:
          policy_type: onap.controllloop.operational.apex.BBS
          policy_type_version: 1.0.0
          policy_type_impl: onap.controllloop.operational.apex.impl
      - onap.controllloop.operational.apex.sampledomain.eastRegion:
          policy_type: onap.controllloop.operational.apex.SampleDomain
          policy_type_version: 1.0.0
          policy_type_impl: onap.controllloop.operational.apex.impl
    min_instance_count: 2
    instance_count: 3
    properties:
      # The properties below are for illustration only
      instance_spawn_load_threshold: 80%
      instance_kill_load_threshold: 60%
      instance_geo_redundancy: true
    deployment_info:
      service_endpoint: https://<the apex service endpoint for this PDP group>
      deployment: A deployment identifier
        # Other deployment info
    instances:
      - instance: apex_1
        state: active
        healthy: yes
        deployment_instance_info:
          node_address: apex_1_podgroup
          # Other deployment instance info
      - instance: apex_2
```

```
            deployment_instance_info:
              node_address: apex_2_pod
              # Other deployment instance infoCreation
          - instance: apex_3
            state: active
            healthy: yes
            deployment_instance_info:
              node_address: apex_3_pod
              # Other deployment instance info

    - pdp_type: xacml
        supported_policy_types:
          - onap.policies.controlloop.guard.FrequencyLimiter
          - onap.policies.controlloop.guard.BlackList
          - onap.policies.controlloop.guard.MinMax
        policies:
          - onap.policies.controlloop.guard.frequencylimiter.EastRegion:
              policy_type: onap.policies.controlloop.guard.FrequencyLimiter
              policy_type_version: 1.0.0
              policy_type_impl: onap.controllloop.guard.impl
          - onap.policies.controlloop.guard.blackList.EastRegion:
              policy_type: onap.policies.controlloop.guard.BlackList
              policy_type_version: 1.0.0
              policy_type_impl: onap.controllloop.guard.impl
          - onap.policies.controlloop.Guard.MinMax.EastRegion:
              policy_type: onap.policies.controlloop.guard.MinMax
              policy_type_version: 1.0.0
              policy_type_impl: onap.controllloop.guard.impl
        min_instance_count: 2
        instance_count: 2
        properties:
          # The properties below are for illustration only
          instance_geo_redundancy: true
        deployment_info:
          service_endpoint: https://<the XACML service endpoint for this PDP group>
          deployment: A deployment identifier
            # Other deployment info
        instances:
          - instance: xacml_1
            state: active
            healthy: yes
            deployment_instance_info:
              node_address: xacml_1_pod
              # Other deployment instance info
          - instance: xacml_2
            state: active
            healthy: yes
            deployment_instance_info:
              node_address: xacml_2_pod
              # Other deployment instance info

- name: onap.pdpgroup.monitoring
  version: 2.1.3
  state: active
  description: DCAE mS Configuration Policies
  properties:
    # PDP group level properties if any
  pdp_subgroups:
    - pdp_type: xacml
        supported_policy_types:
          - onap.policies.monitoring.cdap.tca.hi.lo.app
         policies:
          - onap.scaleout.tca:
              policy_type: onap.policies.monitoring.cdap.tca.hi.lo.app
              policy_type_version: 1.0.0
              policy_type_impl: onap.policies.monitoring.impl
        min_instance_count: 2
        instance_count: 2
        properties:
          # The properties below are for illustration only
          instance_geo_redundancy: true
```

```
          deployment_info:
            service_endpoint: https://<the XACML service endpoint for this PDP group>
            deployment: A deployment identifier
              # Other deployment info
          instances:
            - instance: xacml_1
              state: active
              healthy: yes
              deployment_instance_info:
                node_address: xacml_1_pod
                # Other deployment instance info
            - instance: xacml_2
              state: active
              healthy: yes
              deployment_instance_info:
                node_address: xacml_2_pod
                # Other deployment instance info
```

The table below shows some more examples of GET operations

Example	Description
https:{url}:{port}/policy/pap/v1/pdps	Get all PDP Groups and subgroups in the system
https:{url}:{port}/policy/pap/v1/pdps/groups/onap.pdpgroup.controlloop	Get PDP Groups and subgroups that match the supplied name filter
https:{url}:{port}/policy/pap/v1/pdps/groups/onap.pdpgroup.monitoring/subgroups /xacml	Get the PDP subgroup informtation for the specified subgroup

3.3.2 PDP Group Deployment

This operation allows the PDP groups and subgroups to be created. A POST operation is used to create a new PDP group name. A POST operation is also used to update an existing PDP group. Many PDP groups can be created or updated in a single POST operation by specifying more than one PDP group in the POST operation body.

https:{url}:{port}/policy/pap/v1/pdps POST

POST body to deploy or update PDP groups

```
pdp_groups:
  - name: onap.pdpgroup.controlloop.operational
    description: ONAP Control Loop Operational and Guard policies
    pdp_subgroups:
      - pdp_type: drools
          supportedPolicyTypes:
            - onap.controllloop.operational.drools.vcpe.EastRegion
                              version: 1.2.3
                        - onap.controllloop.operational.drools.vfw.EastRegion
                              version: 1.2.3
          min_instance_count: 3group
          properties:
            # The properties below are for illustration only
            instance_spawn_load_threshold: 70%
            instance_kill_load_threshold: 50%
            instance_geo_redundancy: true

      - pdp_type: apex
          policies:
            - onap.controllloop.operational.apex.bbs.EastRegion
                              version: 1.2.3
            - onap.controllloop.operational.apex.sampledomain.EastRegion
                              version: 1.2.3
          min_instance_count: 2
          properties:
            # The properties below are for illustration only
            instance_spawn_load_threshold: 80%
            instance_kill_load_threshold: 60%
            instance_geo_redundancy: true

      - pdp_type: xacml
          policies:
            - onap.policies.controlloop.guard.frequencylimiter.EastRegion
                              version: 1.2.3
            - onap.policies.controlloop.guard.blacklist.EastRegion
                              version: 1.2.3
            - onap.policies.controlloop.guard.minmax.EastRegion
                              version: 1.2.3
          min_instance_count: 2
          properties:
            # The properties below are for illustration only
            instance_geo_redundancy: true

  - name: onap.pdpgroup.monitoring
    description: DCAE mS Configuration Policies
    properties:
      # PDP group level properties if any
    pdp_subgroups:
      - pdp_type: xacml
          policies:
            - onap.scaleout.tca
                              version: 1.2.3
          min_instance_count: 2
          properties:
            # The properties below are for illustration only
            instance_geo_redundancy: true
```

Other systems such as CLAMP can use this API to deploy policies using a POST operation with the body below where only mandatory fields are specified.

https:{url}:{port}/policy/pap/v1/pdps POST

POST body to deploy or update PDP groups

```
pdp_groups:
  - name: onap.pdpgroup.Monitoring
    description: DCAE mS Configuration Policies
    pdp_subgroups:
      - pdp_type: xacml
          policies:
            - onap.scaleout.tca
```

Simple API for CLAMP to deploy one or more policy-id's with optional policy-version.

https:{url}:{port}/policy/pap/v1/pdps/policies POST

Content-Type: application/json

```
{
  "policies" : [
    {
      "policy-id": "onap.scaleout.tca",
      "policy-version": 1
    },
    {
      "policy-id": "ControlLoop-vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3"
    },
    {
      "policy-id": "guard.frequency.ControlLoop-vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3"
    },
    {
      "policy-id": "guard.minmax.ControlLoop-vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3"
    }
    ]
}
```

HTTP status code indicates success or failure.{

```
        "errorDetails": "some error message"
}
```

Simple API for CLAMP to undeploy a policy-id with optional policy-version.

https:{url}:{port}/policy/pap/v1/pdps/policies{policy-id} DELETE

https:{url}:{port}/policy/pap/v1/pdps/policies{policy-id}/versions/{policy-version} DELETE

HTTP status code indicates success or failure.

```
{
        "errorDetails": "some error message"
}
```

3.3.3 PDP Group Delete

The API also allows PDP groups to be deleted with a DELETE operation. DELETE operations are only permitted on PDP groups in PASSIVE state. The format of the delete operation is as below:

https:{url}:{port}/policy/pap/v1/pdps/groups/onap.pdpgroup.monitoring DELETE

3.3.4 PDP Group State Management

The state of PDP groups is managed by the API. PDP groups can be in states PASSIVE, TEST, SAFE, or ACTIVE. For a full description of PDP group states, see The ONAP Policy Framework architecture page. The state of a PDP group is changed with a PUT operation.

The following PUT operation changes a PDP group to ACTIVE:

https:{url}:{port}/policy/pap/v1/pdps/groups/onap.pdpgroup.monitoring/state=active

There are a number of rules for state management:

1. Only one version of a PDP group may be ACTIVE at any time
2. If a PDP group with a certain version is ACTIVE and a later version of the same PDP group is activated, then the system upgrades the PDP group
3. If a PDP group with a certain version is ACTIVE and an earlier version of the same PDP group is activated, then the system downgrades the PDP group
4. There is no restriction on the number of PASSIVE versions of a PDP group that can exist in the system
5. <Rules on SAFE/TEST> ? Pamela Dragosh

3.3.5 PDP Group Statistics

This operation allows statistics for PDP groups, PDP subgroups, and individual PDPs to be retrieved.

https:{url}:{port}/policy/pap/v1/pdps/statistics GET

Draft Example statistics returned for a PDP Group

```
report_timestamp: 2019-02-11T15:23:50+00:00
pdp_group_count: 2
pdp_groups:
  - name: onap.pdpgroup.controlloop.Operational
    state: active
    create_timestamp: 2019-02-11T15:23:50+00:00
    update_timestamp: 2019-02-12T15:23:50+00:00
    state_change_timestamp: 2019-02-13T15:23:50+00:00
    pdp_subgroups:
      - pdp_type: drools
        instance_count: 3
        deployed_policy_count: 2
        policy_execution_count: 123
        policy_execution_ok_count: 121
        policy_execution_fail_count: 2
        instances:
          - instance: drools_1
              start_timestamp: 2019-02-13T15:23:50+00:00
              policy_execution_count: 50
              policy_execution_ok_count: 49
              policy_execution_fail_count: 1
          - instance: drools_2
              start_timestamp: 2019-02-13T15:30:50+00:00
              policy_execution_count: 50
              policy_execution_ok_count: 49
              policy_execution_fail_count: 1
          - instance: drools_3
              start_timestamp: 2019-02-13T15:33:50+00:00
              policy_execution_count: 23
              policy_execution_ok_count: 23
              policy_execution_fail_count: 0
```

The table below shows some more examples of GET operations for statistics

Example	Description
https:{url}:{port}/policy/pap/v1/pdps/statistics	Get statistics for all PDP Groups and subgroups in the system
https:{url}:{port}/policy/pap/v1/pdps/groups/onap.pdpgroup.controlloop/statistics	Get statistics for all PDP Groups and subgroups that match the supplied name filter
https:{url}:{port}/policy/pap/v1/pdps/groups/onap.pdpgroup.monitoring/subgroups/xacml/statistics	Get statistics for the specified subgroup

3.3.6 PDP Group Health Check

A PDP group health check allows ordering of health checks on PDP groups and on individual PDPs. As health checks may be long lived operations, Health checks are scheduled for execution by this operation. Users check the result of a health check test by issuing a PDP Group Query operation (see Section 3.3.1) and checking the *healthy* field of PDPs.

https:{url}:{port}/policy/pap/v1/pdps/healthcheck PUT

The operation returns a HTTP status code of 202: Accepted if the health check request has been accepted by the PAP. The PAP then orders execution of the health check on the PDPs. The health check result is retrieved with a subsequent GET operation.

The table below shows some more examples of PUT operations for ordering health checks

Example	Description
https:{url}:{port}/policy/pap/v1/pdps/healthcheck PUT	Order a health check on all PDP Groups and subgroups in the system
https:{url}:{port}/policy/pap/v1/pdps/groups/onap.pdpgroup.controlloop /healthcheck PUT	Order a health check on all PDP Groups and subgroups that match the supplied name filter
https:{url}:{port}/policy/pap/v1/pdps/groups/onap.pdpgroup.monitoring /subgroups/xacml/healthcheck PUT	Order a health check on the specified subgroup

3.4 Policy Decision API - Getting Policy Decisions

Policy decisions are required by ONAP components to support the policy-driven ONAP architecture. Policy Decisions are implemented using the XACML PDP. The calling application must provide attributes in order for the XACML PDP to return a correct decision.

3.4.1 Decision API Schema

The schema for the decision API is defined below.

3.4.2 Decision API Queries

Decision API queries are implemented with a POST operation with a JSON body that specifies the filter for the policies to be returned. The JSON body must comply with the schema sepcified in Section 3.4.1.

https:{url}:{port}/decision/v1/ POST

Description of the JSON Payload for the decision API Call

Field	R/O	Type	Description
ONAPName	R	String	Name of the ONAP Project that is making the request.
ONAPComponent	O	String	Name of the ONAP Project component that is making the request.
ONAPInstance	O	String	Optional instance identification for that ONAP component.
action	R	String	The action that the ONAP component is performing on a resource. eg. "configure" DCAE uS onap.Monitoring policy Decisions to configure uS "naming" "placement" "guard"
These sub metadata structures are used to refine which resource the ONAP component is performing an action upon. At least one is required in order for Policy to return a Decision. Multiple structures may be utilized to help refine a Decision.			
policy-type-name		String	The policy type name. This may be a regular expression.
policy-id		String	The policy id. This may be a regular expression or an exact value.

This example below shows the JSON body of a query for a specify policy-id

Decision API Call - Policy ID

```
{
  "ONAPName": "DCAE",
  "ONAPComponent": "PolicyHandler",
  "ONAPInstance": "622431a4-9dea-4eae-b443-3b2164639c64",
  "action": "configure",
  "resource": {
      "policy-id": "onap.scaleout.tca"
  }
}
```

Decision Response - Single Policy ID query


```
{
        "policies": {
                "onap.scaleout.tca": {
                        "type": "onap.policies.monitoring.cdap.tca.hi.lo.app",
                        "version": "1.0.0",
                        "metadata": {
                                "policy-id": "onap.scaleout.tca",
                                "policy-version": 1
                        },
                        "properties": {
                                "tca_policy": {
                                        "domain": "measurementsForVfScaling",
                                        "metricsPerEventName": [
                                                {
                                                        "eventName": "vLoadBalancer",
                                                        "controlLoopSchemaType": "VNF",
                                                        "policyScope": "type=configuration",
                                                        "policyName": "onap.scaleout.tca",
                                                        "policyVersion": "v0.0.1",
                                                        "thresholds": [
                                                                {
                                                                        "closedLoopControlName": "ControlLoop-
vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3",
                                                                        "closedLoopEventStatus": "ONSET",
                                                                        "version": "1.0.2",
                                                                        "fieldPath": "$.event.
measurementsForVfScalingFields.vNicPerformanceArray[*].receivedBroadcastPacketsAccumulated",
                                                                        "thresholdValue": 500,
                                                                        "direction": "LESS_OR_EQUAL",
                                                                        "severity": "MAJOR"
                                                                },
                                                                {
                                                                        "closedLoopControlName": "ControlLoop-
vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3",
                                                                        "closedLoopEventStatus": "ONSET",
                                                                        "version": "1.0.2",
                                                                        "fieldPath": "$.event.
measurementsForVfScalingFields.vNicPerformanceArray[*].receivedBroadcastPacketsAccumulated",
                                                                        "thresholdValue": 5000,
                                                                        "direction": "GREATER_OR_EQUAL",
                                                                        "severity": "CRITICAL"
                                                                }
                                                        ]
                                                }
                                        ]
                                }
                        }
                }
        }
}
```


This example below shows the JSON body of a query for a multiple policy-id's

Decision API Call - Policy ID

```
{
  "ONAPName": "DCAE",
  "ONAPComponent": "PolicyHandler",
  "ONAPInstance": "622431a4-9dea-4eae-b443-3b2164639c64",
  "action": "configure",
  "resource": {
      "policy-id": [
                "onap.scaleout.tca",
                "onap.restart.tca"
                ]
  }
}
```

The following is the response object:

Decision Response - Single Policy ID query

```
{
        "policies": {
                "onap.scaleout.tca": {
                        "type": "onap.policies.monitoring.cdap.tca.hi.lo.app",
                        "version": "1.0.0",
                        "metadata": {
                                "policy-id": "onap.scaleout.tca"
                        },
                        "properties": {
                                "tca_policy": {
                                        "domain": "measurementsForVfScaling",
                                        "metricsPerEventName": [
                                                {
                                                        "eventName": "vLoadBalancer",
                                                        "controlLoopSchemaType": "VNF",
                                                        "policyScope": "type=configuration",
                                                        "policyName": "onap.scaleout.tca",
                                                        "policyVersion": "v0.0.1",
                                                        "thresholds": [
                                                                {
                                                                        "closedLoopControlName": "ControlLoop-
vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3",
                                                                        "closedLoopEventStatus": "ONSET",
                                                                        "version": "1.0.2",
                                                                        "fieldPath": "$.event.
measurementsForVfScalingFields.vNicPerformanceArray[*].receivedBroadcastPacketsAccumulated",
                                                                        "thresholdValue": 500,
                                                                        "direction": "LESS_OR_EQUAL",
                                                                        "severity": "MAJOR"
                                                                },
                                                                {
                                                                        "closedLoopControlName": "ControlLoop-
vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3",
                                                                        "closedLoopEventStatus": "ONSET",
                                                                        "version": "1.0.2",
                                                                        "fieldPath": "$.event.
measurementsForVfScalingFields.vNicPerformanceArray[*].receivedBroadcastPacketsAccumulated",
                                                                        "thresholdValue": 5000,
                                                                        "direction": "GREATER_OR_EQUAL",
                                                                        "severity": "CRITICAL"
                                                                }
                                                        ]
                                                }
                                        ]
                                }
```

```
                        }
                },
                "onap.restart.tca": {
                        "type": "onap.policies.monitoring.cdap.tca.hi.lo.app",
                        "version": "1.0.0",
                        "metadata": {
                                "policy-id": "onap.restart.tca",
                                "policy-version": 1
                        },
                        "properties": {
                                "tca_policy": {
                                        "domain": "measurementsForVfScaling",
                                        "metricsPerEventName": [
                                                {
                                                        "eventName": "Measurement_vGMUX",
                                                        "controlLoopSchemaType": "VNF",
                                                        "policyScope": "DCAE",
                                                        "policyName": "DCAE.Config_tca-hi-lo",
                                                        "policyVersion": "v0.0.1",
                                                        "thresholds": [
                                                                {
                                                                        "closedLoopControlName": "ControlLoop-
vCPE-48f0c2c3-a172-4192-9ae3-052274181b6e",
                                                                        "version": "1.0.2",
                                                                        "fieldPath": "$.event.
measurementsForVfScalingFields.additionalMeasurements[*].arrayOfFields[0].value",
                                                                        "thresholdValue": 0,
                                                                        "direction": "EQUAL",
                                                                        "severity": "MAJOR",
                                                                        "closedLoopEventStatus": "ABATED"
                                                                },
                                                                {
                                                                        "closedLoopControlName": "ControlLoop-
vCPE-48f0c2c3-a172-4192-9ae3-052274181b6e",
                                                                        "version": "1.0.2",
                                                                        "fieldPath": "$.event.
measurementsForVfScalingFields.additionalMeasurements[*].arrayOfFields[0].value",
                                                                        "thresholdValue": 0,
                                                                        "direction": "GREATER",
                                                                        "severity": "CRITICAL",
                                                                        "closedLoopEventStatus": "ONSET"
                                                                }
                                                        ]
                                                }
                                        ]
                                }
                        }
                }
        }
}
```

The simple draft example below shows the JSON body of a query in which all the deployed policies for a specific policy type are returned.

```
{
  "ONAPName": "DCAE",
  "ONAPComponent": "PolicyHandler",
  "ONAPInstance": "622431a4-9dea-4eae-b443-3b2164639c64",
  "action": "configure",
  "resource": {
     "policy-type": "onap.policies.monitoring.cdap.tca.hi.lo.app"
  }
}
```

The query above gives a response similar to the example shown below.


```
{
        "policies": {
                "onap.scaleout.tca": {
                        "type": "onap.policies.monitoring.cdap.tca.hi.lo.app",
                        "version": "1.0.0",
                        "metadata": {
                                "policy-id": "onap.scaleout.tca",
                                "policy-version": 1,
                         },
                        "properties": {
                                "tca_policy": {
                                        "domain": "measurementsForVfScaling",
                                        "metricsPerEventName": [
                                                {
                                                        "eventName": "vLoadBalancer",
                                                        "controlLoopSchemaType": "VNF",
                                                        "policyScope": "type=configuration",
                                                        "policyName": "onap.scaleout.tca",
                                                        "policyVersion": "v0.0.1",
                                                        "thresholds": [
                                                                {
                                                                        "closedLoopControlName": "ControlLoop-
vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3",
                                                                        "closedLoopEventStatus": "ONSET",
                                                                        "version": "1.0.2",
                                                                        "fieldPath": "$.event.
measurementsForVfScalingFields.vNicPerformanceArray[*].receivedBroadcastPacketsAccumulated",
                                                                        "thresholdValue": 500,
                                                                        "direction": "LESS_OR_EQUAL",
                                                                        "severity": "MAJOR"
                                                                },
                                                                {
                                                                        "closedLoopControlName": "ControlLoop-
vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3",
                                                                        "closedLoopEventStatus": "ONSET",
                                                                        "version": "1.0.2",
                                                                        "fieldPath": "$.event.
measurementsForVfScalingFields.vNicPerformanceArray[*].receivedBroadcastPacketsAccumulated",
                                                                        "thresholdValue": 5000,
                                                                        "direction": "GREATER_OR_EQUAL",
                                                                        "severity": "CRITICAL"
                                                                }
                                                        ]
                                                }
                                        ]
                                }
                        }
                },
                "onap.restart.tca": {
                        "type": "onap.policies.monitoring.cdap.tca.hi.lo.app",
                        "version": "1.0.0",
                        "metadata": {
                                "policy-id": "onap.restart.tca",
                                "policy-version": 1
                        },
                        "properties": {
                                "tca_policy": {
                                        "domain": "measurementsForVfScaling",
                                        "metricsPerEventName": [
                                                {
                                                        "eventName": "Measurement_vGMUX",
                                                        "controlLoopSchemaType": "VNF",
                                                        "policyScope": "DCAE",
                                                        "policyName": "DCAE.Config_tca-hi-lo",
                                                        "policyVersion": "v0.0.1",
                                                        "thresholds": [
```

```
                                                        {
                                                                "closedLoopControlName": "ControlLoop-
vCPE-48f0c2c3-a172-4192-9ae3-052274181b6e",
                                                                "version": "1.0.2",
                                                                "fieldPath": "$.event.
measurementsForVfScalingFields.additionalMeasurements[*].arrayOfFields[0].value",
                                                                "thresholdValue": 0,
                                                                "direction": "EQUAL",
                                                                "severity": "MAJOR",
                                                                "closedLoopEventStatus": "ABATED"
                                                        },
                                                        {
                                                                "closedLoopControlName": "ControlLoop-
vCPE-48f0c2c3-a172-4192-9ae3-052274181b6e",
                                                                "version": "1.0.2",
                                                                "fieldPath": "$.event.
measurementsForVfScalingFields.additionalMeasurements[*].arrayOfFields[0].value",
                                                                "thresholdValue": 0,
                                                                "direction": "GREATER",
                                                                "severity": "CRITICAL",
                                                                "closedLoopEventStatus": "ONSET"
                                                        }
                                                ]
                                        }
                                ]
                        }
                }
        },
        "onap.vfirewall.tca": {
                "type": "onap.policy.monitoring.cdap.tca.hi.lo.app",
                "version": "1.0.0",
                "metadata": {
                        "policy-id": "onap.vfirewall.tca",
                        "policy-version": 1
                },
                "properties": {
                        "tca_policy": {
                                "domain": "measurementsForVfScaling",
                                "metricsPerEventName": [
                                        {
                                                "eventName": "vLoadBalancer",
                                                "controlLoopSchemaType": "VNF",
                                                "policyScope": "resource=vLoadBalancer;
type=configuration",
                                                "policyName": "onap.vfirewall.tca",
                                                "policyVersion": "v0.0.1",
                                                "thresholds": [
                                                        {
                                                                "closedLoopControlName": "ControlLoop-
vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a",
                                                                "closedLoopEventStatus": "ONSET",
                                                                "version": "1.0.2",
                                                                "fieldPath": "$.event.
measurementsForVfScalingFields.vNicPerformanceArray[*].receivedBroadcastPacketsAccumulated",
                                                                "thresholdValue": 500,
                                                                "direction": "LESS_OR_EQUAL",
                                                                "severity": "MAJOR"
                                                        },
                                                        {
                                                                "closedLoopControlName": "ControlLoop-
vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a",
                                                                "closedLoopEventStatus": "ONSET",
                                                                "version": "1.0.2",
                                                                "fieldPath": "$.event.
measurementsForVfScalingFields.vNicPerformanceArray[*].receivedBroadcastPacketsAccumulated",
                                                                "thresholdValue": 5000,
                                                                "direction": "GREATER_OR_EQUAL",
                                                                "severity": "CRITICAL"
                                                        }
                                                ]
                                        }
```

```
                                        ]
                                    }
                                }
                            }
                        }
                    }
}
```

4. Policy Framework Internal APIs

The Policy Framework uses the internal APIs documented in the subsections below. The APIs in this section are used for internal communication in the Policy Framework. The APIs are NOT supported for use by components outside the Policy Framework and are subject to revision and change at any time.

4.1 PAP to PDP API

This section describes the API between the PAP and PDPs. The APIs in this section are implemented using DMaaP API messaging. There are four messages on the API:

1. PDP_STATUS: PDPPAP, used by PDPs to report to the PAP
2. PDP_UPDATE: PAPPDP, used by the PAP to update the policies running on PDPs, triggers a PDP_STATUS message with the result of the PDP_UPDATE operation
3. PDP_STATE_CHANGE: PAPPDP, used by the PAP to change the state of PDPs, triggers a PDP_STATUS message with the result of the PDP_STATE_CHANGE operation
4. PDP_HEALTH_CHECK: PAPPDP, used by the PAP to order a heakth check on PDPs, triggers a PDP_STATUS message with the result of the PDP_HEALTH_CHECK operation

The fields below are valid on API calls:

Field	PDP STATUS	PDP UPDATE	PDP STATE CHANGE	PDP HEALTH CHECK	Comment
(message_name)	M	M	M	M	pdp_status, pdp_update, pdp_state_change, or pdp_health_check
name	M	M	C	C	The name of the PDP, for state changes and health checks, the PDP group and subgroup can be used to specify the scope of the operation
version	M	N/A	N/A	N/A	The version of the PDP
pdp_type	M	M	N/A	N/A	The type of the PDP, currently xacml, drools, or apex
state	M	N/A	M	N/A	The administrative state of the PDP group: PASSIVE, SAFE, TEST, ACTIVE, or TERMINATED
healthy	M	N/A	N/A	N/A	The result of the latest health check on the PDP: HEALTHY/NOT_HEALTHY /TEST_IN_PROGRESS
description	O	O	N/A	N/A	The description of the PDP
pdp_group	O	M	C	C	The PDP group to which the PDP belongs, the PDP group and subgroup can be used to specify the scope of the operation
pdp_subgroup	O	M	C	C	The PDP subgroup to which the PDP belongs, the PDP group and subgroup can be used to specify the scope of the operation
supported_policy_types	M	N/A	N/A	N/A	A list of the policy types supported by the PDP
policies	O	M	N/A	N/A	The list of policies running on the PDP
(name)	O	M	N/A	N/A	The name of a TOSCA policy running on the PDP
policy_type	O	M	N/A	N/A	The TOSCA policy type of the policyWhen a PDP starts, it commences periodic sending of *PDP_STATUS* messages on DMaaP. The PAP receives these messages and acts in whatever manner is appropriate.
policy_type_version	O	M	N/A	N/A	The version of the TOSCA policy type of the policy
properties	O	M	N/A	N/A	The properties of the policy for the XACML, Drools, or APEX PDP, see section 3.2 for details
instance	M	N/A	N/A	N/A	The instance ID of the PDP running in a Kuberenetes Pod
deployment_instance_info	M	N/A	N/A	N/A	Information on the node running the PDP
properties	O	O	N/A	N/A	Other properties specific to the PDP

statistics		M	N/A	N/A	N/A	Statistics on policy execution in the PDP
	policy_download_count	M	N/A	N/A	N/A	The number of policies downloaded into the PDP
	policy_download_success_count	M	N/A	N/A	N/A	The number of policies successfully downloaded into the PDP
	policy_download_fail_count	M	N/A	N/A	N/A	The number of policies downloaded into the PDP where the download failed
	policy_executed_count	M	N/A	N/A	N/A	The number of policy executions on the PDP
	policy_executed_success_count	M	N/A	N/A	N/A	The number of policy executions on the PDP that completed successfully
	policy_executed_fail_count	M	N/A	N/A	N/A	The number of policy executions on the PDP that failed
response		O	N/A	N/A	N/A	The response to the last operation that the PAP executed on the PDP
	response_to	M	N/A	N/A	N/A	The PAP to PDP message to which this is a response
	response_status	M	N/A	N/A	N/A	SUCCESS or FAIL
	response_message	O	N/A	N/A	N/A	Message giving further information on the successful or failed operation

YAML is used for illustrative purposes in the examples in this section. JSON (application/json) is used as the content type in the implementation of this API.

Note: The PAP checks that the set of policy types supported in all PDPs in a PDP subgroup are identical and will not add a PDP to a PDP subgroup that has a different set of supported policy types
Note: The PA checks that the set of policy loaded on all PDPs in a PDP subgroup are are identical and will not add a PDP to a PDP subgroup that has a different set of loaded policies

4.1.1 PAP API for PDPs

The purpose of this API is for PDPs to provide heartbeat, status. health, and statistical information to Policy Administration. There is a single *PDP_STATUS* message on this API. PDPs send this message to the PAP using the *POLICY_PDP_PAP* DMaaP topic. The PAP listens on this topic for messages.

When a PDP starts, it commences periodic sending of *PDP_STATUS* messages on DMaaP. The PAP receives these messages and acts in whatever manner is appropriate. *PDP_UPDATE*, *PDP_STATE_CHANGE*, and *PDP_HEALTH_CHECK* operations trigger a *PDP_STATUS* message as a response.

The *PDP_STATUS* message is used for PDP heartbeat monitoring. A PDP sends a *PDP_STATUS* message with a state of *TERMINATED* when it terminates normally. If a *PDP_STATUS* message is not received from a PDP in a certain configurable time, then the PAP assumes the PDP has failed.

A PDP may be preconfigured with its PDP group, PDP subgroup, and policies. If the PDP group, subgroup, or any policy sent to the PAP in a *PDP_STATUS* message is unknown to the PAP, the PAP locks the PDP in state PASSIVE.

PDP_STATUS message from an XACML PDP running control loop policies

```
pdp_status:
  name: xacml_1
  version: 1.2.3
  pdp_type: xacml
  state: active
  healthy: true
  description: XACML PDP running control loop policies
  pdp_group: onap.pdpgroup.controlloop.operational
  pdp_subgroup: xacml
  supported_policy_types:
    - onap.policies.controlloop.guard.FrequencyLimiter
    - onap.policies.controlloop.guard.BlackList
    - onap.policies.controlloop.guard.MinMax
  policies:
    - onap.policies.controlloop.guard.frequencylimiter.EastRegion:
        policy_type: onap.policies.controlloop.guard.FrequencyLimiter
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
   - onap.policies.controlloop.guard.blacklist.eastRegion:
        policy_type: onap.policies.controlloop.guard.BlackList
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
    - onap.policies.controlloop.guard.minmax.eastRegion:
        policy_type: onap.policies.controlloop.guard.MinMax
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
  instance: xacml_1
  deployment_instance_info:
    node_address: xacml_1_pod
    # Other deployment instance info
  statistics:
    policy_download_count: 0
    policy_download_success_count: 0
    policy_download_fail_count: 0
    policy_executed_count: 123
    policy_executed_success_count: 122
    policy_executed_fail_count: 1
```

PDP_STATUS message from a Drools PDP running control loop policies

```
pdp_status:
  name: drools_2
  version: 2.3.4
  pdp_type: drools
  state: safe
  healthy: true
  description: Drools PDP running control loop policies
  pdp_group: onap.pdpgroup.controlloop.operational
  pdp_subgroup: drools
  supported_policy_types:
    - onap.controlloop.operational.drools.vCPE
    - onap.controlloop.operational.drools.vFW
  policies:
    - onap.controlloop.operational.drools.vcpe.EastRegion:
        policy_type: onap.controlloop.operational.drools.vCPE
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
    - onap.controlloop.operational.drools.vfw.EastRegion:
        policy_type: onap.controlloop.operational.drools.vFW
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
  instance: drools_2
  deployment_instance_info:
    node_address: drools_2_pod
    # Other deployment instance info
  statistics:
    policy_download_count: 3
    policy_download_success_count: 3
    policy_download_fail_count: 0
    policy_executed_count: 123
    policy_executed_success_count: 122
    policy_executed_fail_count: 1
  response:
    response_to: PDP_HEALTH_CHECK
    response_status: SUCCESS
```

PDP_STATUS message from an APEX PDP running control loop policies

```
pdp_status:
  name: apex_3
  version: 2.2.1
  pdp_type: apex
  state: test
  healthy: true
  description: APEX PDP running control loop policies
  pdp_group: onap.pdpgroup.controlloop.operational
  pdp_subgroup: apex
  supported_policy_types:
    - onap.controlloop.operational.apex.BBS
    - onap.controlloop.operational.apex.SampleDomain
   policies:
    - onap.controlloop.operational.apex.bbs.EastRegion:
        policy_type: onap.controlloop.operational.apex.BBS
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
    - onap.controlloop.operational.apex.sampledomain.EastRegion:
        policy_type: onap.controlloop.operational.apex.SampleDomain
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
  instance: apex_3
  deployment_instance_info:node_address
    node_address: apex_3_pod
    # Other deployment instance info
  statistics:
    policy_download_count: 2
    policy_download_success_count: 2
    policy_download_fail_count: 0
    policy_executed_count: 123
    policy_executed_success_count: 122
    policy_executed_fail_count: 1
  response:
    response_to: PDP_UPDATE
    response_status: FAIL
    response_message: policies specified in update message incompatible with running policy state
```

PDP_STATUS message from an XACML PDP running monitoring policies

```
pdp_status:
  name: xacml_1
  version: 1.2.3
  pdp_type: xacml
  state: active
  healthy: true
  description: XACML PDP running monitoring policies
  pdp_group: onap.pdpgroup.Monitoring
  pdp_subgroup: xacml
  supported_policy_types:
    - onap.monitoring.cdap.tca.hi.lo.app
   policies:
    - onap.scaleout.tca:message
        policy_type: onap.policies.monitoring.cdap.tca.hi.lo.app
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
  instance: xacml_1
  deployment_instance_info:
    node_address: xacml_1_pod
    # Other deployment instance info
  statistics:
    policy_download_count: 0
    policy_download_success_count: 0
    policy_download_fail_count: 0
    policy_executed_count: 123
    policy_executed_success_count: 122
    policy_executed_fail_count: 1
```

4.1.2 PDP API for PAPs

The purpose of this API is for the PAP to load and update policies on PDPs and to change the state of PDPs. It also allows the PAP to order health checks to run on PDPs. The PAP sends *PDP_UPDATE*, *PDP_* STATE_CHANGE, and *PDP_HEALTH_CHECK* messages to PDPs using the *POLICY_PAP_PDP* DMaaP topic. PDPs listens on this topic for messages.

The PAP can set the scope of STATE_CHANGE, and *PDP_HEALTH_CHECK* messages:

- PDP Group: If a PDP group is specified in a message, then the PDPs in that PDP group respond to the message and all other PDPs ignore it.
- PDP Group and subgroup: If a PDP group and subgroup are specified in a message, then only the PDPs of that subgroup in the PDP group respond to the message and all other PDPs ignore it.
- Single PDP: If the name of a PDP is specified in a message, then only that PDP responds to the message and all other PDPs ignore it.

Note: *PDP_UPDATE* messages must be issued individually to PDPs because the *PDP_UPDATE* operation can change the PDP group to which a PDP belongs.

4.1.2.1 PDP Update

The *PDP_UPDATE* operation allows the PAP to modify the PDP group to which a PDP belongs and the policies in a PDP. Only PDPs in state PASSIVE accept this operation. The PAP must change the state of PDPs in state ACTIVE, TEST, or SAFE to state PASSIVE before issuing a *PDP_UPDATE* operation on a PDP.

The following examples illustrate how the operation is used.

PDP_UPDATE message to upgrade XACML PDP control loop policies to versino 1.0.1

```
pdp_update:
  name: xacml_1
  pdp_type: xacml
  description: XACML PDP running control loop policies, Upgraded
  pdp_group: onap.pdpgroup.controlloop.operational
  pdp_subgroup: xacml
  policies:
    - onap.policies.controlloop.guard.frequencylimiter.EastRegion:
        policy_type: onap.policies.controlloop.guard.FrequencyLimiter
        policy_type_version: 1.0.1
        properties:
          # Omitted for brevity, see Section 3.2
   - onap.policies.controlloop.guard.blackList.EastRegion:
        policy_type: onap.policies.controlloop.guard.BlackList
        policy_type_version: 1.0.1
        properties:
          # Omitted for brevity, see Section 3.2
    - onap.policies.controlloop.guard.minmax.EastRegion:
        policy_type: onap.policies.controlloop.guard.MinMax
        policy_type_version: 1.0.1
        properties:
          # Omitted for brevity, see Section 3.2
```

PDP_UPDATE message to a Drools PDP to add an extra control loop policy

```
pdp_update:
  name: drools_2
  pdp_type: drools
  description: Drools PDP running control loop policies, extra policy added
  pdp_group: onap.pdpgroup.controlloop.operational
  pdp_subgroup: drools
  policies:
    - onap.controllloop.operational.drools.vcpe.EastRegion:
        policy_type: onap.controllloop.operational.drools.vCPE
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
    - onap.controllloop.operational.drools.vfw.EastRegion:
        policy_type: onap.controllloop.operational.drools.vFW
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
    - onap.controllloop.operational.drools.vfw.WestRegion:
        policy_type: onap.controllloop.operational.drools.vFW
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
```

PDP_UPDATE message to an APEX PDP to remove a control loop policy

```
pdp_update:
  name: apex_3
  pdp_type: apex
  description: APEX PDP updated to remove a control loop policy
  pdp_group: onap.pdpgroup.controlloop.operational
  pdp_subgroup: apex
  policies:
    - onap.controllloop.operational.apex.bbs.EastRegion:
        policy_type: onap.controllloop.operational.apex.BBS
        policy_type_version: 1.0.0
        properties:
          # Omitted for brevity, see Section 3.2
```

4.1.2.2 PDP State Change

The *PDP_STATE_CHANGE* operation allows the PAP to order state changes on PDPs in PDP groups and subgroups. The following examples illustrate how the operation is used.

Change the state of all control loop Drools PDPs to ACTIVE

```
pdp_state_change:
  state: active
  pdp_group: onap.pdpgroup.controlloop.Operational
  pdp_subgroup: drools
```

Change the state of all monitoring PDPs to SAFE

```
pdp_state_change:
  state: safe
  pdp_group: onap.pdpgroup.Monitoring
```

Change the state of a single APEX PDP to TEST

```
pdp_state_change:
  state: test
  name: apex_3
```

4.1.2.3 PDP Health Check

The *PDP_HEALTH_CHECK* operation allows the PAP to order health checks on PDPs in PDP groups and subgroups. The following examples illustrate how the operation is used.

Perform a health check on all control loop Drools PDPs

```
pdp_health_check:
  pdp_group: onap.pdpgroup.controlloop.Operational
  pdp_subgroup: drools
```

perform a health check on all monitoring PDPs

```
pdp_health_check:
  pdp_group: onap.pdpgroup.Monitoring
```

Perform a health check on a single APEX PDP

```
pdp_health_check:
  name: apex_3
```

4.2 Policy Type Implementations (Native Policies)

The policy Framework must have implementations for all Policy Type entities that may be specified in TOSCA. Policy type implementations are native policies for the various PDPs supported in the Policy Framework. They may be predefined and preloaded into the Policy Framework. In addition, they may also be added, modified, queried, or deleted using this API during runtime.

The API supports CRUD of *PolicyTypeImpl* policy type implementations, where the XACML, Drools, and APEX policy type implementations are supplied as strings. This API is provided by the *PolicyDevelopment* component of the Policy Framework, see The ONAP Policy Framework architecture.

Note that client-side editing support for TOSCA *PolicyType* definitions or for *PolicyTypeImpl* implementations in XACML, Drools, or APEX is outside the current scope of the API.
Note: Preloaded policy type implementations may only be queried over this API, modification or deletion of preloaded policy type implementations is disabled.
Note: Policy type implementations that are in use (referenced by defined Policies) may not be deleted.

The fields below are valid on API calls:

Field	GET	POST	DELETE	Comment
name	M	M	M	The name of the Policy Type implementation
version	O	M	C	The version of the Policy Type implementation
policy_type	R	M	N/A	The TOSCA policy type that this policy type implementation implements
pdp_type	R	M	N/A	The PDP type of this policy type implementation, currently xacml, drools, or apex
description	R	O	N/A	The description of the policy type implementation
writable	R	N/A	N/A	Writable flag, false for predefined policy type implementations, true for policy type implementations defined over the API
policy_body	R	M	N/A	The body (source) of the policy type implementation
properties	R	O	N/A	Specific properties for the policy type implementation

4.2.1 Policy Type Implementation Query

This operation allows the PDP groups and subgroups to be listed together with the policies that are deployed on each PDP group and subgroup.

https:{url}:{port}/policy/api/v1/native/onap.policies.controlloop.operational/impls GET

Policy Type Implementation Query Result

```
policy_type_impls:
  - name: onap.policies.controlloop.operational.drools.Impl
    version: 1.0.0
    policy_type: onap.policies.controlloop.Operational
    pdp_type: drools
    description: Implementation of the drools control loop policies
    writable: false

  - name: onap.policies.controlloop.operational.apex.bbs.Impl
    version: 1.0.0
    policy_type: onap.policies.controlloop.operational.Apex
    pdp_type: apex
    description: Implementation of the APEX BBS control loop policy
    writable: true
    policy_body: "<policy body>"

  - name: onap.policies.controlloop.operational.apex.sampledomain.Impl
    version: 1.0.0
    policy_type: onap.policies.controlloop.operational.Apex
    pdp_type: apex
    description: Implementation of the SampleDomain test APEX policy
    writable: true
    policy_body: "<policy body>"
```

The table below shows some more examples of GET operations

Example	Description
https:{url}:{port}/policy/api/v1/native/{policy type id}/impls *eg.* *https:{url}:{port}/policy/api/v1/native/onap.policies.monitoring/impls* *https:{url}:{port}/policy/api/v1/native/onap.policies.controlloop.operational.apex/impls*	Get all Policy Type implementations for the given policy type
https:{url}:{port}/policy/api/v1/native/{policy type id}/impls/{policy type impl id} *eg.* *https:{url}:{port}/policy/api/v1/native/onap.policies.controlloop.operational/impls/onap.policies.controlloop.operational.drools.impl* *https:{url}:{port}/policy/api/v1/native/onap.policies.controlloop.operational.apex/impls/onap.policies.controlloop.operational.apex.sampledomain.impl*	Get all Policy Type implementation versions that match the policy type and policy type implementation IDs specified
https:{url}:{port}/policy/api/v1/native/{policy type id}/impls/{policy type impl id}/versions/{version id} *eg.* *https:{url}:{port}/policy/api/v1/native/onap.policies.controlloop.operational/impls/onap.policies.controlloop.operational.drools.impl/versions/1.2.3* *https:{url}:{port}/policy/api/v1/native/onap.policies.controlloop.operational.apex/impls/onap.policies.controlloop.operational.apex.sampledomain.impl/versions/latest*	Get the specific Policy Type implementation with the specified name and version, if the version ID is specified a *latest*, the latest version is returned

4.2.2 Policy Type Implementation Create/Update

The API allows users (such as a policy editor or DevOps system) to create or update a Policy Type implementation using a POST operation. This API allows new Policy Type implementations to be created or existing Policy Type implementations to be modified. POST operations with a new name or a new version of an existing name are used to create a new Policy Type implementation. POST operations with an existing name and version are used to update an existing Policy Type implementations. Many implementations can be created or updated in a single POST operation by specifying more than one Policy Type implementation on the *policy_type_impls* list.

For example, the POST operation below with the YAML body below is used to create a new APEX Policy type implementation.

https:{url}:{port}/policy/api/v1/native/onap.policies.controlloop.operational.apex/impls POST

Create a new Policy Type Implementation

```
policy_type_impls:
  - onap.policies.controlloop.operational.apex.bbs.Impl:
      version: 1.0.0
      policy_type: onap.policies.controlloop.operational.Apex
      pdp_type: apex
      description: Implementation of the APEX BBS control loop policy
      policy_body: "<policy body>"
  - onap.policies.controlloop.operational.apex.sampledomain.Impl:
      version: 1.0.0
      policy_type: onap.policies.controlloop.operational.Apex
      pdp_type: apex
      description: Implementation of the APEX SampleDomain control loop policy
      policy_body: "<policy body>
```

Once this call is made, the Policy Type query in Section 3.1.2.1 returns a result with the new Policy Type implementation defined.

4.2.3 Policy Type Implementation Delete

The API also allows Policy Type implementations to be deleted with a DELETE operation. The format of the delete operation is as below:

https:{url}:{port}/api/v1/native/onap.policies.controlloop.operational.apex/impls/onap.policies.apex.bbs.impl/versions/1.0.0 DELETE

Note: Predefined policy type implementations cannot be deleted
Note: Policy type implementations that are in use (Parameterized by a TOSCA Policy) may not be deleted, the parameterizing TOSCA policies must be deleted first
Note: The *version* parameter may be omitted on the DELETE operation if there is only one version of the policy type implementation in the system