impact of the updated PDP API on policy-handling by DCAE-Controller

see Re: Policy Design and API Flow for Model Driven Control Loop - Draft for the PDP API changes in Dublin R4

Intentions and requirements

1. make API on PAP/PDP to use the consistent structures on all operations
2. simplify the usage of PAP/PDP API
3. for transition period of about 1 year both the old and the new APIs to be maintained by PAP/PDP
4. CLAMP to use only the new API of PAP in Dublin R4
5. DCAE-Controller is requested to use both the new and the old APIs of PDP for the transition period to avoid breaking the existing components

---

PDP API change that impacts DCAE-Controller

1. Design of the new API on PAP/PDP is in progress
2. Only a **subset** of the new PAP/PDP API is designed and is in the draft status.
3. The new API changes the following aspects (**everything**)
   a. url
   b. request structures are totally redesigned. New field names, new placements, many fields are removed, no obvious mapping between the old and the new fields.
   c. response structures are totally redesigned. New field names, new placements, many fields are removed, no obvious mapping between the old and the new fields.
   d. the structure of the policy config that contains the actual policy is changed to match the policy-model.

The impact to DCAE-Controller is MAJOR and the PDP API design is far from being usable

---

replacement of `(POST /getConfig)` with the new Policy Decision API - Getting Policy Decisions `(POST /decision/v1/) for a single policy_id`

The current version of **`/decision/v1/`** only provides a single policy request-response for the single **`policy_id`** (= onap.vfirewall.tca):

/decision/v1

```
http:{url}:{port}/decision/v1/ POST

Content-Type: application/yaml
Accepts: application/json
subject: DCAE
action: configure
resource:
    policy_id: onap.vfirewall.tca

#
# Return Output
#

HTTP/1.1 200 OK
Content-Type: application/json
{
    "onap.vfirewall.tca": {
        "type": "onap.policy.monitoring.cdap.tca.hi.lo.app",
        "properties": {
            "domain": "measurementsForVfScaling",...
}}}
```

Getting multiple policies and varying number of policies with the new PDP API - is not provided

This feature is heavily used by and needed to DCAE-Controller because it has to go from multiple policies in PDP to multiple policies attached to multiple components in DCAE in as few http queries as possible

---

Comparison with the **`POST /getConfig`**

1. **`policy_id`** is the new field in PDP API that was not used before. DCAE-Controller had to introduce the **`policy_id`** term to identify the single policy regardless of its version. No complaints here
2. only a **single string value** is provided in the new PDP API for the **`policy_id`**
    - a. the old PDP API had a way to use the **regular expression** in the **`policyName`** field to request multiple policies at a time.
        - i. *This feature of the **regular expression** matching is **heavily** used by DCAE-Controller to send a **single** or as few requests to PDP as possible when **querying for the latest policies that are attached to the components in DCAE***
3. response contains a dictionary from **`policy_id`** value to the **config** structure that contains the actual policy instance. No header information is provided in the response that affects
    - a. it is not obvious from looking at the response to explain what is the key and what is the value in the dictionary.
    - b. it is not clear what the response would look like if **no policy is found** for the policy_id.
        - i. Preferably we would need to avoid relying on the absence of data in response to identify the absence of the record since the absence might be caused by other reasons as well like database not available at all or any other error conditions, etc.
    - c. **no `policyVersion`** is provided in response header for the **`policy_id`**.
        - i. The **`policyVersion`** is used by many places in DCAE-Controller to figure out
            1. whether the policy is changed in PDP
            2. whether the wave of changes has to be propagated through the DCAE-Controller,
            3. whether the policy record has to be updated in consul-kv
            4. whether the component needs to be push-notified about the update.
        - ii. Not having the **`policyVersion`** in the response makes it unclear how
    - d. there are **no fields** that were provided before that were used for **matching** of the policy to the policy-filter(s) optionally attached to the DCAE component. The policy-filter currently is a copy-paste of the /getConfig request.
4. Therefore using the new PDP API as it is design at the moment will break all the behavior of DCAE-Controller and the components

---

High level view on the changes of the DCAE-Controller to accommodate the new API of PDP

It seems that to selectively consume both the new and the old PDP APIs by DCAE-Controller the **new node types** need to be introduced to the blueprint of the components. Suggested node types could be as follows

1. `dcae.nodes.policy.`**`v2`** can be derived from the current `dcae.nodes.policy`
    - a. having the new name with v2 of the node type will indicate to use the new PDP API to retrieve policy from the PDP. This way DCAE-Controller can slowly migrate the components from the old PDP API to the new PDP API over time
    - b. the new type will be mutually exclusive in the blueprint to avoid the confusion for having the same policy in two formats.
        - i. We have to enforce the mutual exclusiveness during the blueprint generation

        ii. if blueprint would still contain the mix
            1. The suggestion is to use only the new policy structures (v2) on all the policies attached to the component as soon as at least one policy node type is of v2.
            2. However, that would require some ugly coding and re-retrieving of policies on gathering the policies to the component...
    c. the v2 node type can contain both the `policy_id` and the `policy_model_id` since this is the new type anyway
2. `dcae.nodes.policies.`**`v2`** can be derived from the current `dcae.nodes.policies`
    a. it is not clear how this type should be modeled since there is no matching PDP API

It seems that **all the chain of platform components** (policy-handler, deployment-handler, policy-plugin, onap-dcae-dcaepolicy-lib) in DCAE-Controller will have to be **heavily changed** to accommodate the new node types and use the new PDP API along with the old PDP API.

Besides, **each service component** will have to be changed as well to **consume the new policy structures**.

Open issues

1. It is not clear on whether and how the web-socket API is changing
    a. the web-socket API contains the policyName, policyVersion, as well as the matchingConditions used for matching to the policy-filters
    b. it is not clear how the policy-update notification can be matched to the new PDP API
