ONAP
OPEN NETWORK AUTOMATION PLATFORM

Representation and Identification of a cloud region in ONAP

Bin Yang (Wind River)

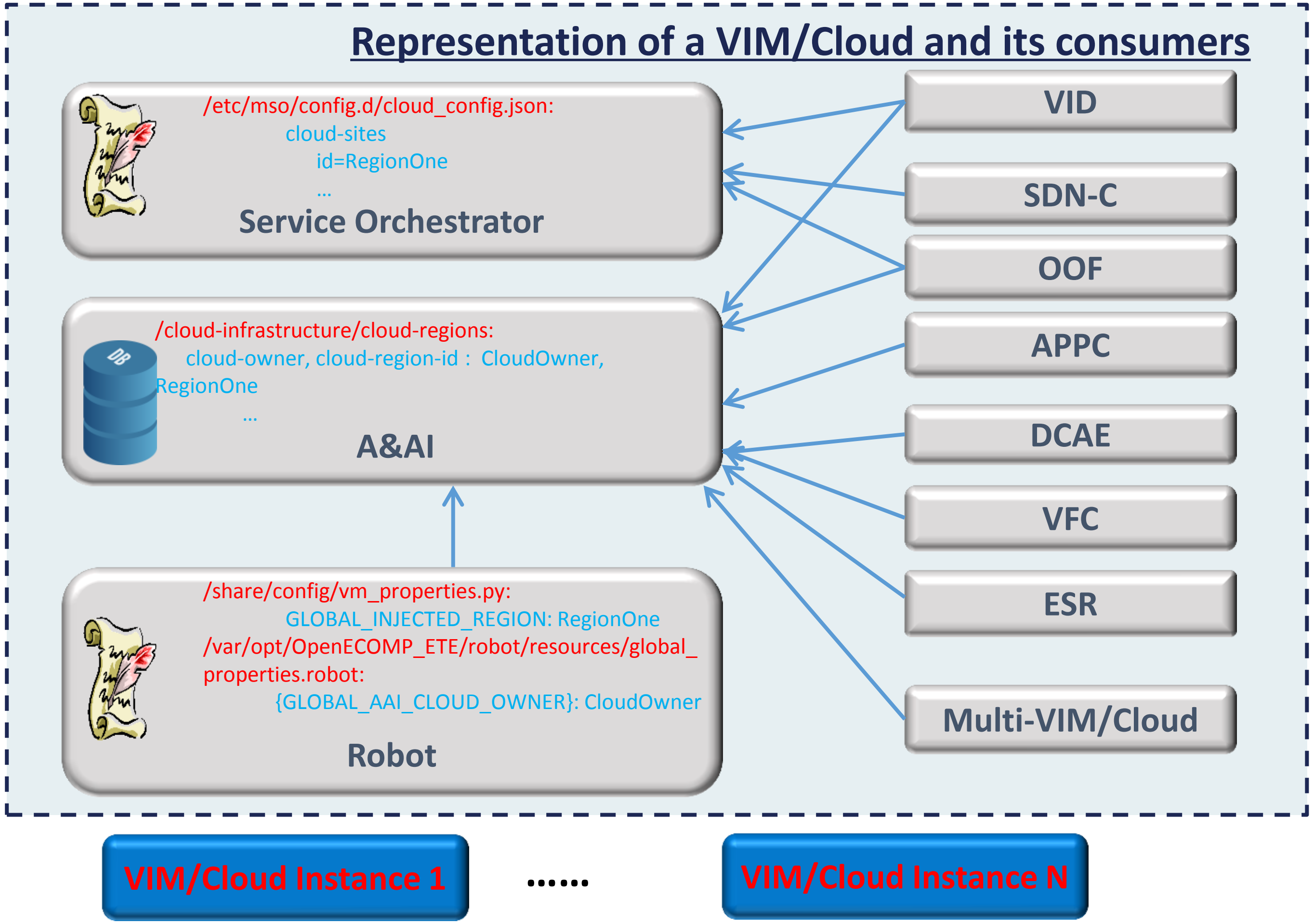
March , 2018

Agenda

- Status quo: tricky approach to onboard a Cloud Region
- Discrepancy: the assumptions and the constraint
- Proposed solution
- Suggested action plan

Status quo: Tricky approach to onboard a Cloud Region

- Representation of a cloud region
  - 3 representations for a single cloud region
  - To add a new Cloud Region for ONAP orchestration
    - Add a Cloud Region Object into AAI
    - Add a cloud-site in SO
    - Launch another Robot VM
      - Provision with new VIM/Cloud information via heat env file
      - It will dedicatedly manage the new cloud region
      - Not sure if it really works since it seems the cloud owner/region id are hard-coded in Robot when it talks to A&AI
  - wiki for detail instructions https://wiki.onap.org/pages/viewpage.action?pageId=25431491
- Identification of a cloud region
  - **'cloud-region-id'** used in VID/SO/SDNC
  - **'cloud-owner' + 'cloud-region-id'** used by AAI and its consumers
  - **'vim-id' = {'cloud-owner'}_{'cloud-region-id'}** used by MultiCloud and its consumers



Representation of a VIM/Cloud and its consumers

```
/etc/mso/config.d/cloud_config.json:
        cloud-sites
            id=RegionOne
            …
```

Service Orchestrator

```
/cloud-infrastructure/cloud-regions:
    cloud-owner, cloud-region-id :  CloudOwner, RegionOne
            …
```

A&AI

```
/share/config/vm_properties.py:
        GLOBAL_INJECTED_REGION: RegionOne
/var/opt/OpenECOMP_ETE/robot/resources/global_properties.robot:
        {GLOBAL_AAI_CLOUD_OWNER}: CloudOwner
```

Robot

Discrepancy: the assumptions and the constraint

- Robot/VID/SDNC assume
  - assume that “cloud-owner” is hard-coded as “CloudOwner”
  - “cloud-region-id” is used to identify a VIM/Cloud instance so it is expected to be unique globally

- Robot assumes
  - “cloud-region-id” being the “region id” of underlying OpenStack
  - Robot use this cloud-region-id to make request to underlying VIM/Cloud instance.

- Constraint resulted by assumptions above:
  - No more than 1 VIM/Cloud instance with same “region id” is not allowed to be represented in ONAP
  - Cloud Owner cannot be value other than “CloudOwner”.

Proposed solution

- Single representation
  - Leverage cloud region representation in AAI
  - Depreciate all other representations in other ONAP modules,e.g. SO, Robot

- Consistent Identification of a cloud region
  - Use composed keys: {cloud-owner} + {cloud-region-id} to identify a cloud region
  - Depreciate the usage of {cloud-region-id} only
  - Depreciate the usage of {vim-id}
- Impacted projects
  - Multiple ONAP projects are involved
    - VID, SO, SDNC, OOF, VFC, MultiCloud, Integration, ESR
  - Some consumers of MultiCloud might be not involved
    - APPC, DCAEgen2, etc.

Suggested action plan

- **ARC subcommittee**
  - In C Release, finalize the architecture design to consistently represent and identify a cloud region
- **UseCase subcommittee**
  - In C Release, drive the implementation of the design above through a specific use case, either stick to a existing use case or a special one.
- **VID/SO/SDNC**
  - In C Release, design the APIs between each other, uses composed keys {cloud-owner} + {cloud-region-id} instead of just {cloud-region-id} to specify a cloud region
  - In C Release, SO leverage AAI for representation of a cloud region, hence depreciate the internal one (cloud-config.json)
- **MultiCloud and its consumer**
  - In C Release, uses composed keys {cloud-owner} + {cloud-region-id} instead of {vim-id} to specify a cloud region
- **Robot**
  - In C Release, allow users to specify composed keys {cloud-owner} + {cloud-region-id} whenever executing scripts
  - In C Release, leverages A&AI for representation of a cloud region, hence depreciate the usage of internal representation of a cloud region (vm_properties.py)

Suggestions?

ONAP

OPEN NETWORK AUTOMATION PLATFORM

谢谢

ONAP
OPEN NETWORK AUTOMATION PLATFORM

BACK UP slides