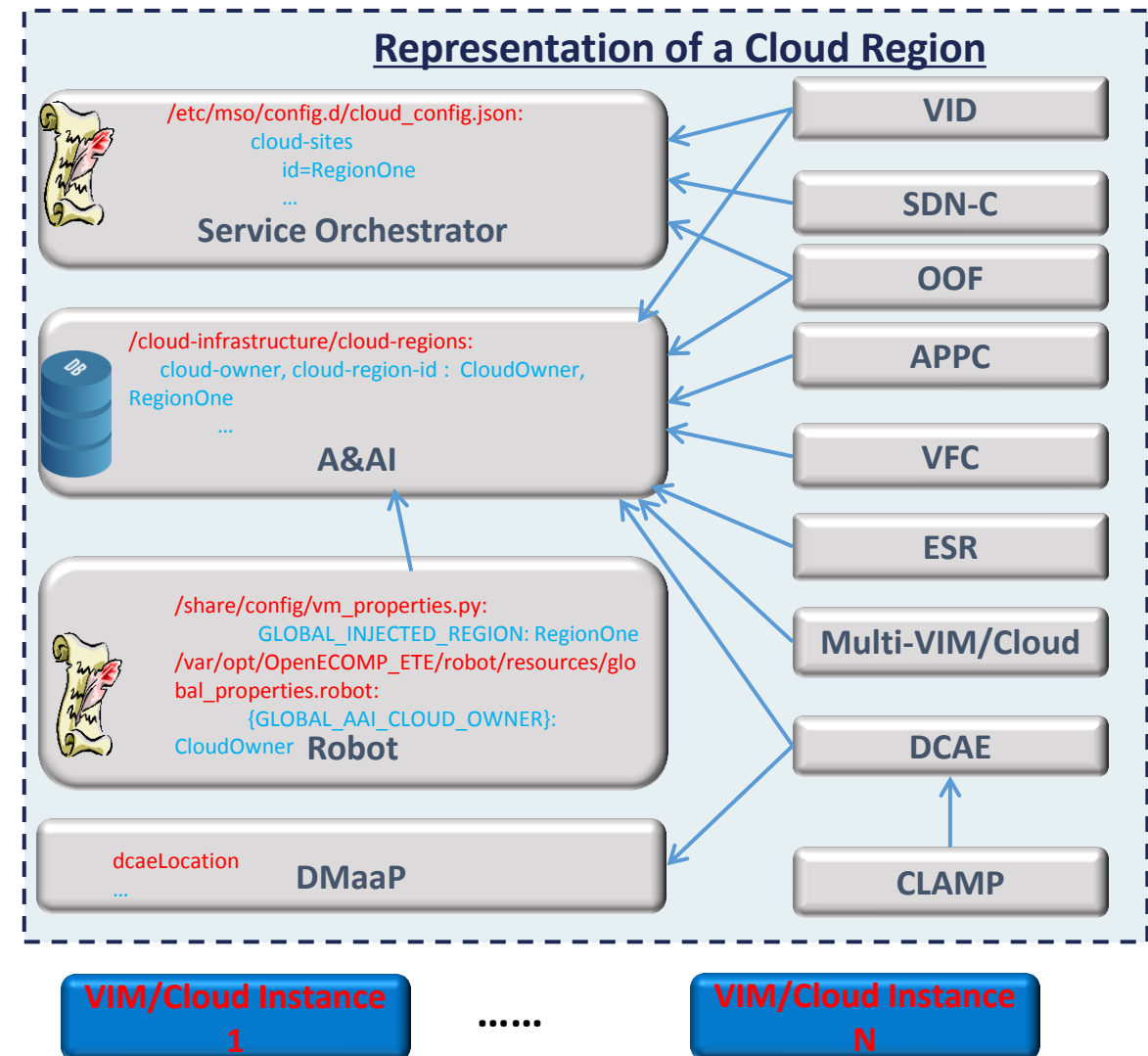


# ONAP Casablanca Requirement: Centralized representation and consistent identification of cloud regions in ONAP

Bin Yang (Wind River)  
June 29, 2018

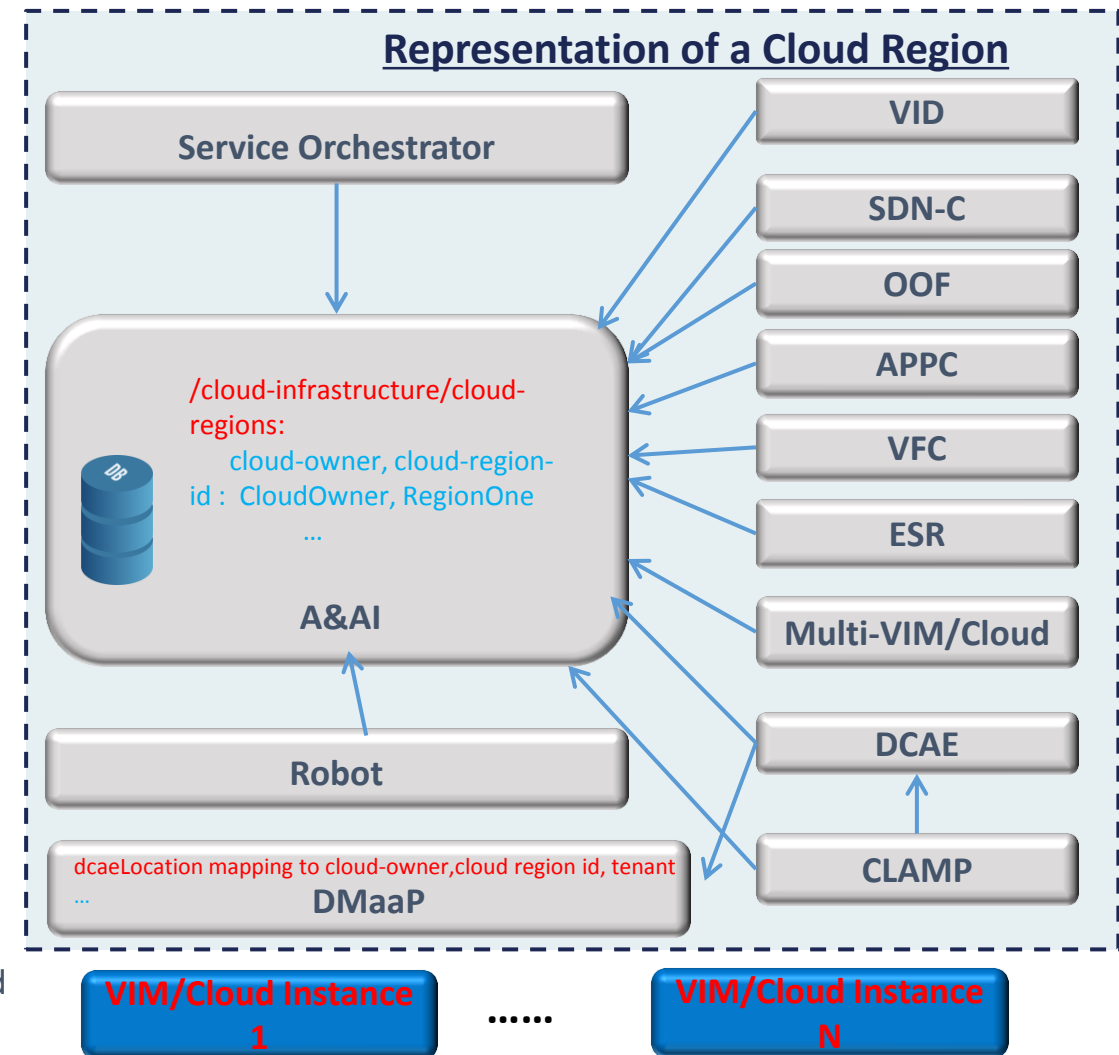
# Status quo: multiple representation for a Cloud Region

- The facts in ONAP A (and B) releases:
  - 4 representations for a single cloud region
    - Cloud region in AAI
    - Cloud-sites in SO
    - Vm\_propertise in Robot VM
    - dcaeLocation in DMaaP
  - Inconsistent identification of a single 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
    - dcaeLocation used by DCAE and DMaaP
- Make it is complicated/tricky to
  - on-board a new Cloud Region for ONAP
    - Add a Cloud Region Object into AAI
    - Add a cloud-site in SO
    - Launch another Robot VM to represent new cloud region
  - impose unnecessary constraint to cloud regions
    - No more than 1 cloud region with the same cloud region id can be on-boarded into ONAP



# Proposed solution

- Single/Centralized representation of a cloud region
  - All ONAP modules leverage cloud region representation in AAI
  - Depreciate all other representations in other ONAP modules, e.g. cloud-sites in SO, vm\_properties in Robot
- Consistent Identification of a cloud region
  - Use composed keys: {cloud-owner} + {cloud-region-id} to identify a cloud region
  - Depreciate the usage of the {cloud-region-id} only
  - Depreciate the usage of {vim-id}
  - dcaeLocation is equivalent to a tenant under a cloud region, suggest dcaeLocation can be mapped to a cloud region + tenant
- Impacted projects
  - Multiple ONAP projects are involved
    - VID, SO, SDNC, OOF, VFC, MultiCloud, UI, DCAE, DMaaP, Integration
  - Some consumers of MultiCloud need to be scrutinized
    - APPC, etc.
    - They are using the “vim-id” but in an transparent way: fetch it from AAI and pass it to MultiCloud
- Notes:
  - AAI's cloud region is the representation of VIM/Cloud instance for VNF orchestration, while it seems dcaeLocation is the representation of VIM/Cloud instance for ONAP components distributed deployment.
  - It is not clear yet if dcaeLocation should be correlated/aligned to AAI's cloud region.



# Suggested action plan A

- **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 standalone one: <https://wiki.onap.org/display/DW/Multiple+Cloud+Regions+On-boarding+and+Decommission+in+ONAP>
- **VID/SO/SDNC**
  - In C Release, design the APIs between each other, to use 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)
- **CLAMP/DCAE/DMaaP**
  - In case that dcaeLocation should be correlated/aligned to AAI's cloud region
  - In C Release, make sure dcaeLocation is derived from a cloud region + tenant, and can be translate to {cloud-owner } + {cloud-region-id} + {tenant} for retrieving cloud region information from AAI
- **MultiCloud and its consumers (OOF, VFC,UUI)**
  - In C Release, design the APIs between each other to use composed keys {cloud-owner} + {cloud-region-id} instead of {vim-id} to specify a cloud region
- **Integration (Robot scripts)**
  - 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)

# Suggested action plan B

- **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 standalone one: <https://wiki.onap.org/display/DW/Multiple+Cloud+Regions+On-boarding+and+Decommission+in+ONAP>

- **Break into 3 Phases**

- Align to MVP principle,
- Make it is possible to realize the complete solution by multiple ONAP releases
- Phase 1: Centralize the representation of cloud regions
- Phase 2: Consistent ID of a cloud region
- Phase 3: Correlate and align dcaeLocation to AAI's cloud region ( **in case that dcaeLocation should be correlated/aligned to AAI's cloud region**)

# Suggested action plan B (Phase 1)

- **Phase 1: Centralize the representation of cloud regions**
  - The centralization representation of cloud regions facilitate the on-boarding process of new VIM/Cloud instances.
  - Nevertheless with **constraints** with regarding to the Identification of a cloud region
    - value of “cloud-owner”: underscore character “\_” cannot be used
    - “cloud-region-id” should be unique across all cloud regions.
  - Impacted project: **SO**
    - Remove hard coded cloud sites
    - Integrate with MultiCloud for heat based VNF orchestration.
      - With this integration SO does not need to store/access cloud region representation any more
  - Impacted project: **Integration**
    - Refactor robot scripts: depreciate the usage of local configuration file: vm\_properties.py; allow scripts users to specify the “cloud-owner” and “cloud-region-id”
    - Refactor robot scripts: populate the default cloud region into AAI with correct information: align to ESR VIM registration.

# Suggested action plan B (Phase 2)

- **Phase 2: Consistent ID of a cloud region**
  - **Agree on the usage of consistent ID to specify a cloud region**
    - Composed keys {cloud-owner} , {cloud-region-id} is preferred choice
  - **MultiCloud**
    - Upgrade the API version, use composed keys {cloud-owner} + {cloud-region-id} instead of {vim-id} to specify a cloud region
  - **VID/SO/SDNC/OOF**
    - Use composed keys {cloud-owner} + {cloud-region-id} instead of just {cloud-region-id} to specify a cloud region.
  - **VFC,UUI**
    - Use composed keys {cloud-owner} + {cloud-region-id} instead of {vim-id} to specify a cloud region

# Suggested action plan B (Phase 3)

- **Phase 3: Correlate and align dcaeLocation to AAI's cloud region**
  - **In case that dcaeLocation should be correlated/aligned to AAI's cloud region**
  - dcaeLocation is equivalent to AAI's cloud region + tenant.
  - DMaaP may have their own cache/storage of the dcaeLocation
  - Make sure the dcaeLocation is derived from AAI's cloud region
    - Cloud Region is on-boarded into AAI once, no need to create dcaeLocation
    - CLAMP user retrieves the Cloud Region information from AAI and create dcaeLocation accordingly
  - **CLAMP**
    - Retrieve cloud region from AAI, then distribute the cloud region's information to DCAE/DMaaP to create dcaeLocation
  - **DCAE/DMaaP**
    - Some synchronization mechanism should be applied to ensure the updated cloud region information will be synced to dcaeLocation





# ONAP

OPEN NETWORK AUTOMATION PLATFORM

谢谢