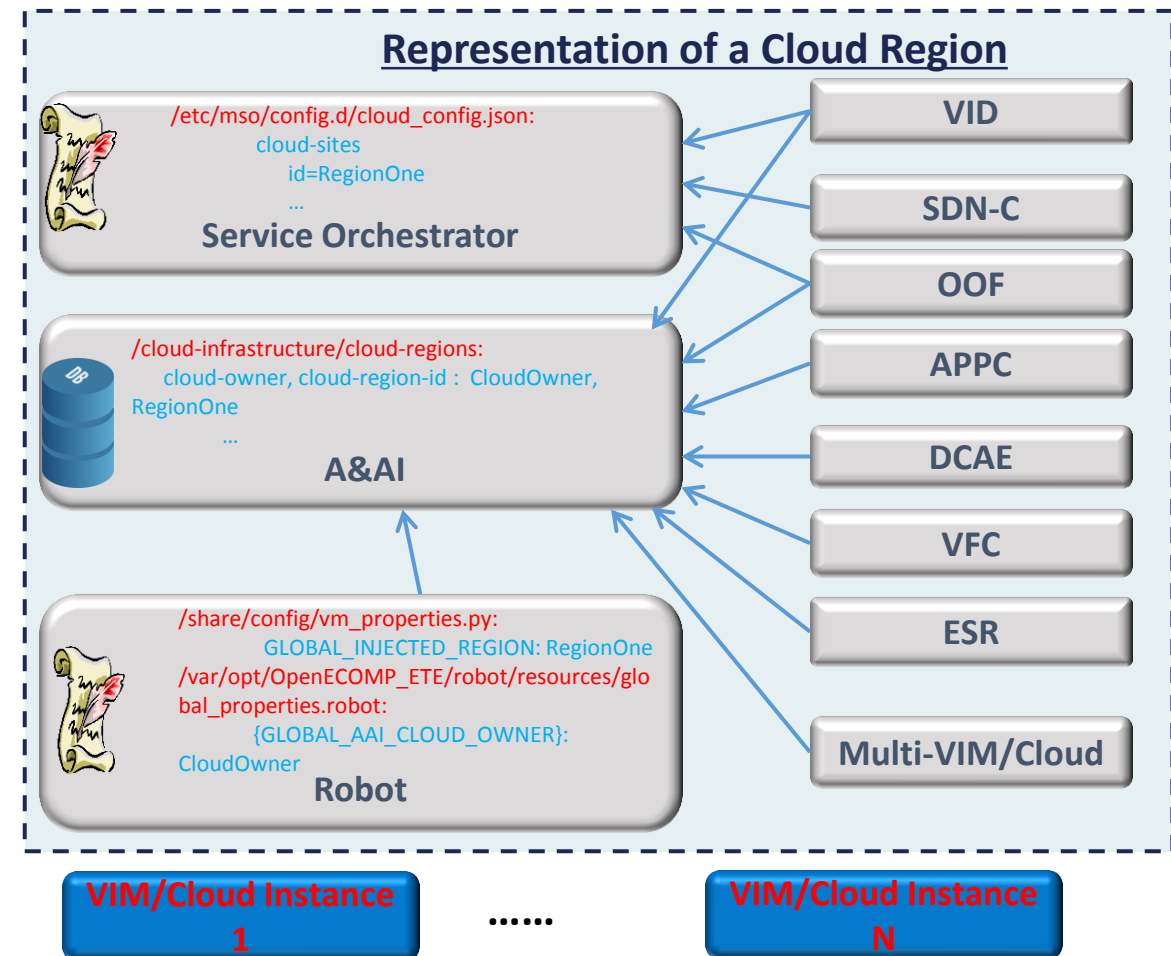


ONAP Casablanca Requirement: Centralized representation and consistent identification of a cloud region

Bin Yang (Wind River)
ONS LA, March 26, 2018

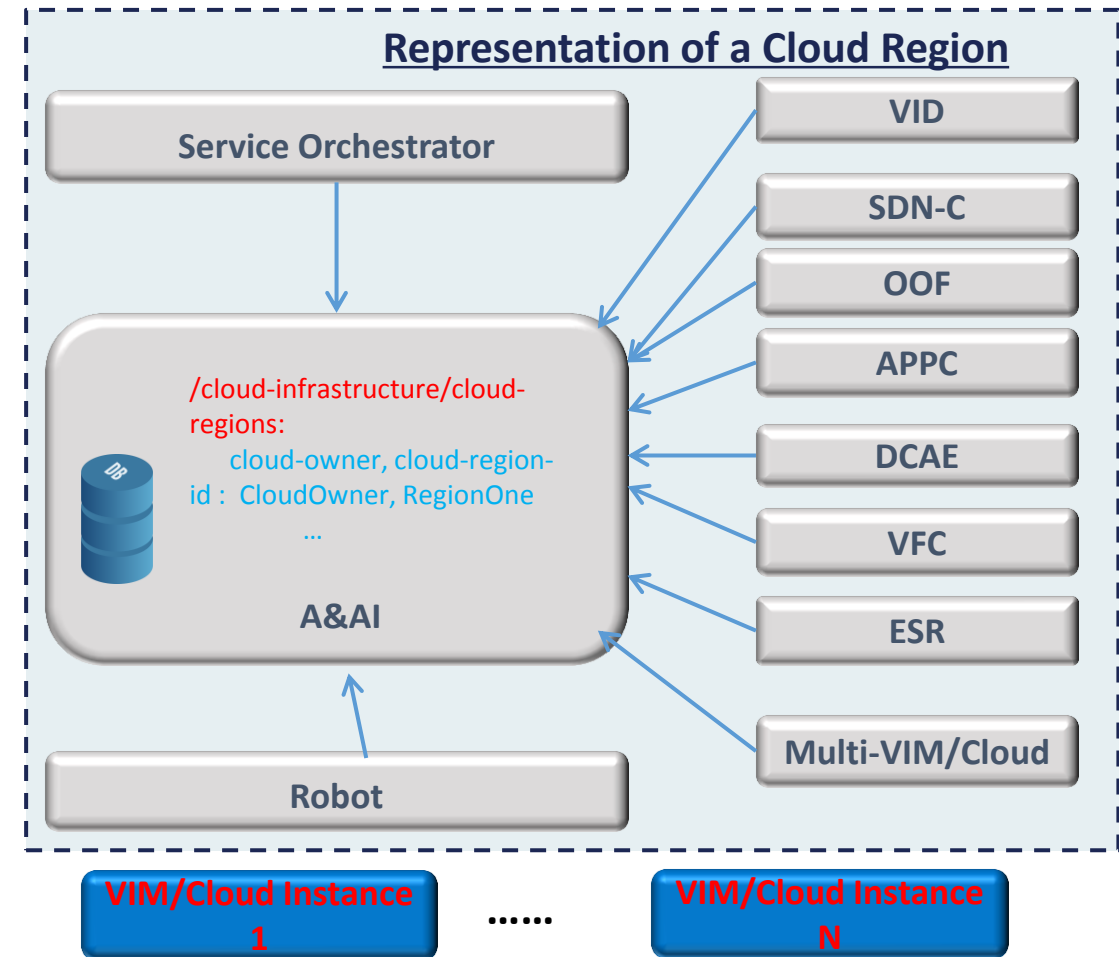
Status quo: multiple representation a Cloud Region

- The facts in ONAP A (and B) releases:
 - 3 representations for a single cloud region
 - Cloud region in AAI
 - Cloud-sites in SO
 - Vm_propertise in Robot VM
 - 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
- 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}
- Impacted projects
 - Multiple ONAP projects are involved
 - VID, SO, SDNC, OOF, VFC, MultiCloud, Integration, ESR, UII
 - Some consumers of MultiCloud need to be scrutinized
 - APPC, DCAEgen2, etc.
 - They are using the “vim-id” but in an transparent way: fetch it from AAI and pass it to MultiCloud



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 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)
- **MultiCloud and its consumers (OOF, VFC, UI, ESR)**
 - 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
- **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)



ONAP

OPEN NETWORK AUTOMATION PLATFORM

谢谢