



ONAP container

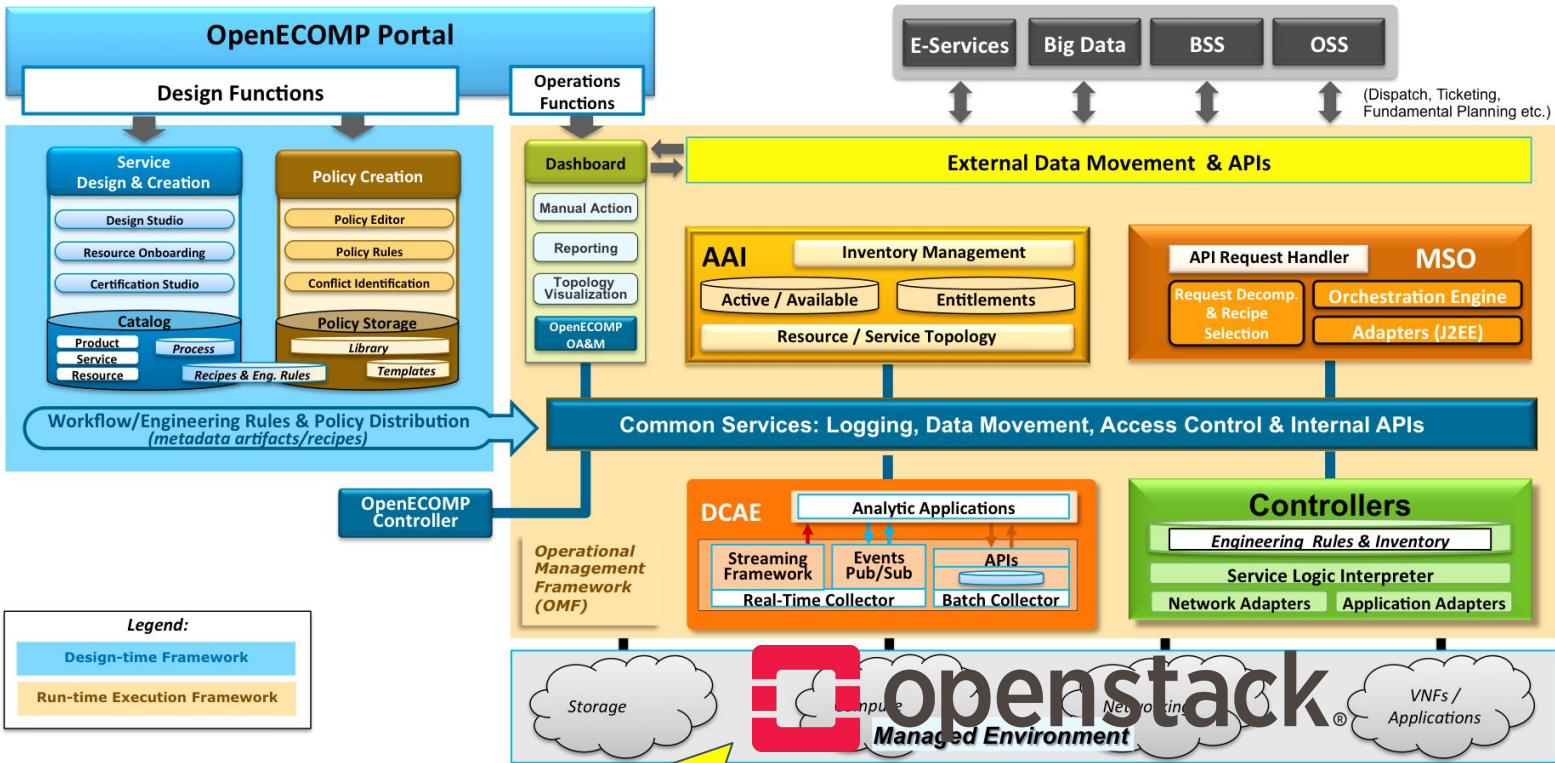
For architecture subcommittee review

November 14, 2017

Isaku Yamahata

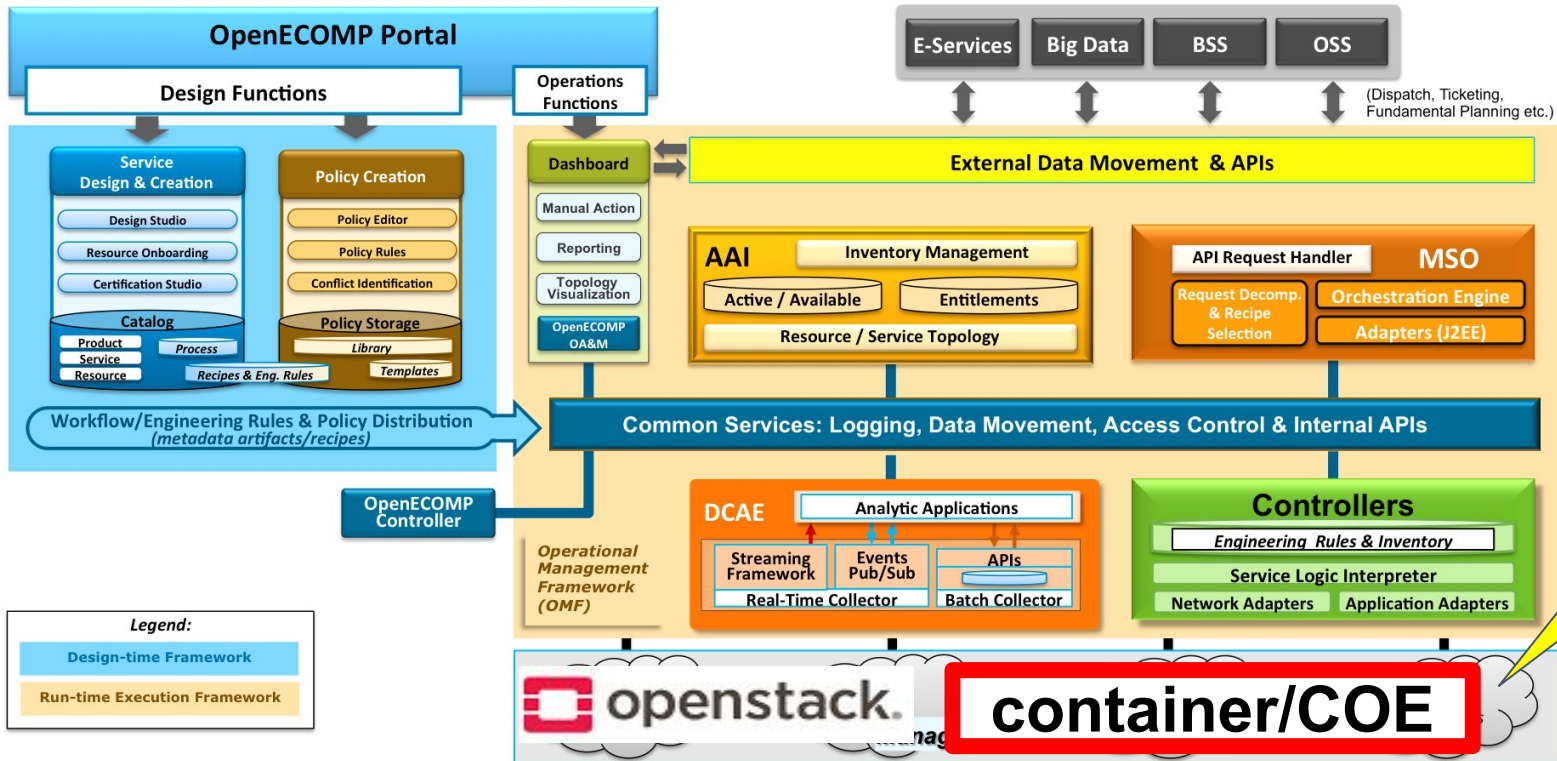
Goal and scope

Current ONAP architecture



Currently ONAP is heavily tied into OpenStack(or VM)

Goal and Scope



Proposal:
teach
ONAP
container/
COE

influence/
feedback/
contribute

Related projects
(TOSCA, container,
COE etc...)

Goal and scope contd

Goal

- Have ONAP take advantage of container/COE technology for cloud native era
- Utilizing of industry momentum/direction
- Influence/feedback the related technologies(e.g. TOSCA, container/COE)

Scope

- Teach ONAP container/COE instead of openstack so that VNFs can be deployed/run over container/COE in cloud native way
- Related project: multicloud and related projects which use multicloud.

Benefit for container/COE technology

- container/COE has many cloud native features.
 - Easy deployment/version upgrade
 - Autoscaling/autohealing with lower response time
 - E.g. K8s: wisdom of google's past 15+ experience
- Industry trends
 - Take advantage of industry investment
 - Align with industry trends
- For more cloud native VNF

risk

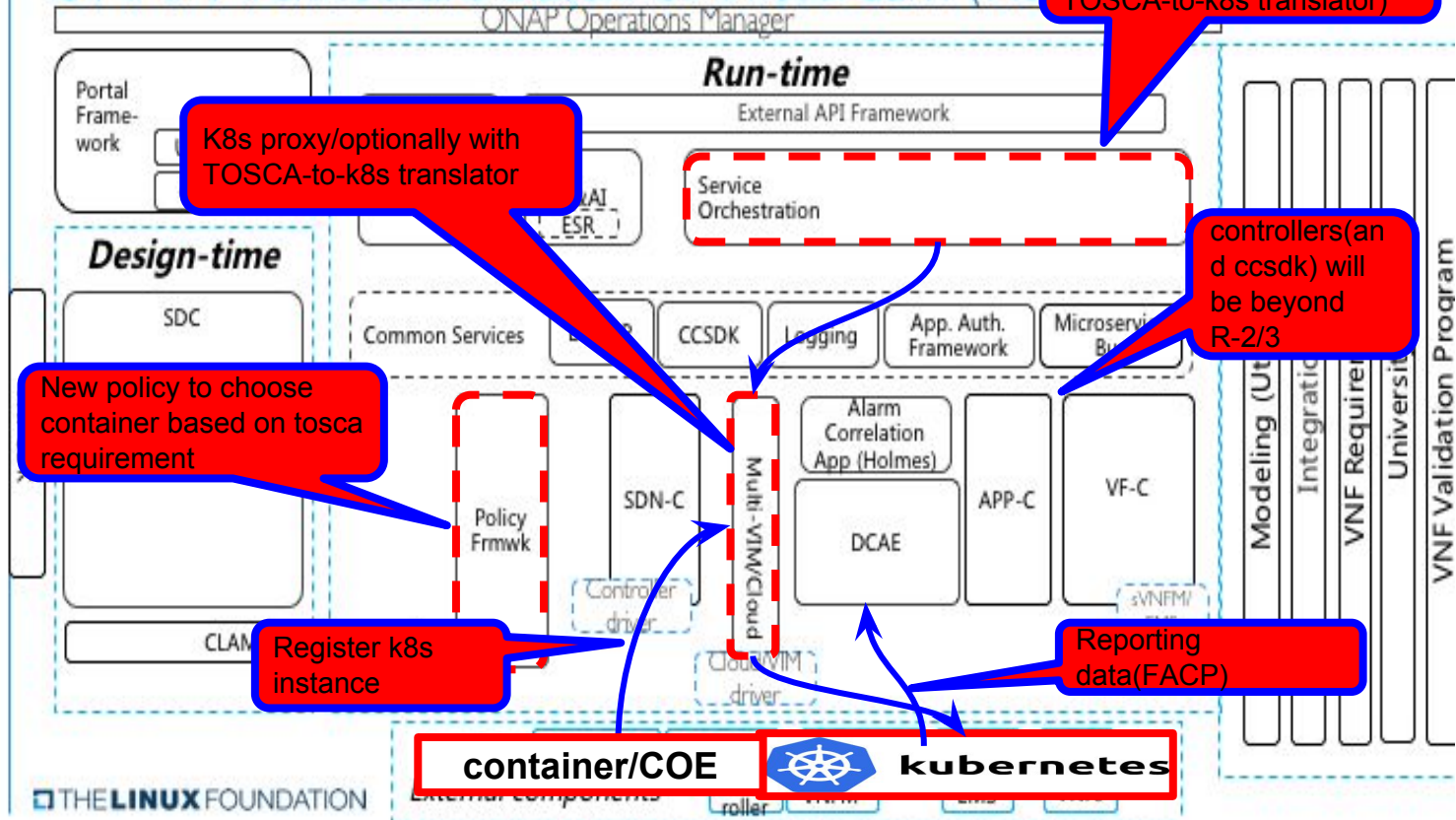
- container/coe technology may not be as mature as VM(openstack) technology

Non goal/out-of-scope

- Not installer/deployment. ONAP running over container
 - OOM project ONAP on kubernetes
 - <https://wiki.onap.org/pages/viewpage.action?pageId=3247305>
 - <https://wiki.onap.org/display/DW/ONAP+Operations+Manager+Project>
 - Self hosting/management might be possible. But it would be further phase.
- container/COE deployment
- On-demand Installing container/coe on public cloud/VMs/baremetal as cloud deployment
 - This is also out of scope for now.
 - For ease of use/deployment, this will be further phase.

Proposed architecture and Choices for project/architecture

ONAP Architecture Baseline Amsterdam (Release 2.0)



<https://wiki.onap.org/display/DW/Architecture>

Design principle

- ONAP should be able to take all the advantage of container/coe technology
- The design should not tie in to any single container technology
- design/project should align with other ONAP architecture/projects/future direction
- Should leverage the existing components
- Balance long term direction and short term achievement
 - E.g. TOSCA -> container API isn't available. It would take long time. Multiple steps will be needed.

architecture choices

item	recommendation
Project home(new project or subproject of an existing project?)	New projet
How to use container	Use container orchestration engine.e.g. k8s
API design: especially workload/lifecycle, model driven or not?	Model driven. Allow SO to use COE API directly. Multicloud as proxy: Open in multicloud discussion
Where to convert TOSCA to container API (if model driven). SO/controllers or multicloud	SO/controllers (eventually as CCSDK)
Source code repo	Subrepo of multicloud as starter

Next steps

- Agree on its design/architecture and project home
 - Usecases, story, integration(containerised VNFs)
 - 5G, edge cloud
- Define the next development scope and propose new project
 - Feature, functionality and task
 - Release-3?
 - activity can start from R-2 and in R-3 the effort can be official.
 - Pick one usecase and containerize it: vCPE?
- participants start development

Detailed Discussion

Recommendation
in following slides

Second Recommendation as
short term work around

Project home: new project or existing project?

Project home	pros	cons
Multicloud subproject	Least overhead for project management	architecture/design/implementation would be tracted to the existing design
New project for any container	New design for container is possible. The scope would be much more than multicloud project.	Overhead of project management
New project for each container technology	ditto	There is commonality among container technology. Too much overhead of project management

How to use container

How to use container/coe	comment
Run container under VM based orchestration. E.g. openstack nova-docker, zun, magnum)	Little benefit to use container
Use container (e.g. docker) and manage other resource by ONAP.	Resource(host/network/etc) orchestration layer is missing. Needs to be implemented in ONAP
Use container orchestration engine. E.g. k8s/docker swarm/...	Full advantage of container/coe

There are also technologies for COE to use VM. e.g. virtlet, kubevirt. It would make sense as VNF migration from VM to container as transition path.

Api choice	Align with multicloud direction?	New API? API consumer needs to be enhanced	container/coe feature can be easily utilized?	Other comment
Re-use the existing multicloud API: Coerce into the existing VIM API	Align with the existing code. No with future direction.	No	No	
Define new API for container	No with future direction	Yes	Yes	
Expose container/COE API directly	No with future direction	Yes	Yes	heavily depends on container/coe technology
Model driven API: without enhancement: coerce into model driven VM based API	yes	No. no additional changes to API consumer	No	
Model driven API: With enhancement. Probably in long term, contributing to TOSCA	Yes with future direction.	Yes	Yes	Needs to implement/invent conversion logic from TOSCA to container/COE Reasonable abstraction among container/COE technologies

Where to translate TOSCA to K8S: Open

Now under discussion: Fuel for discussion

Eventually follow the community decision

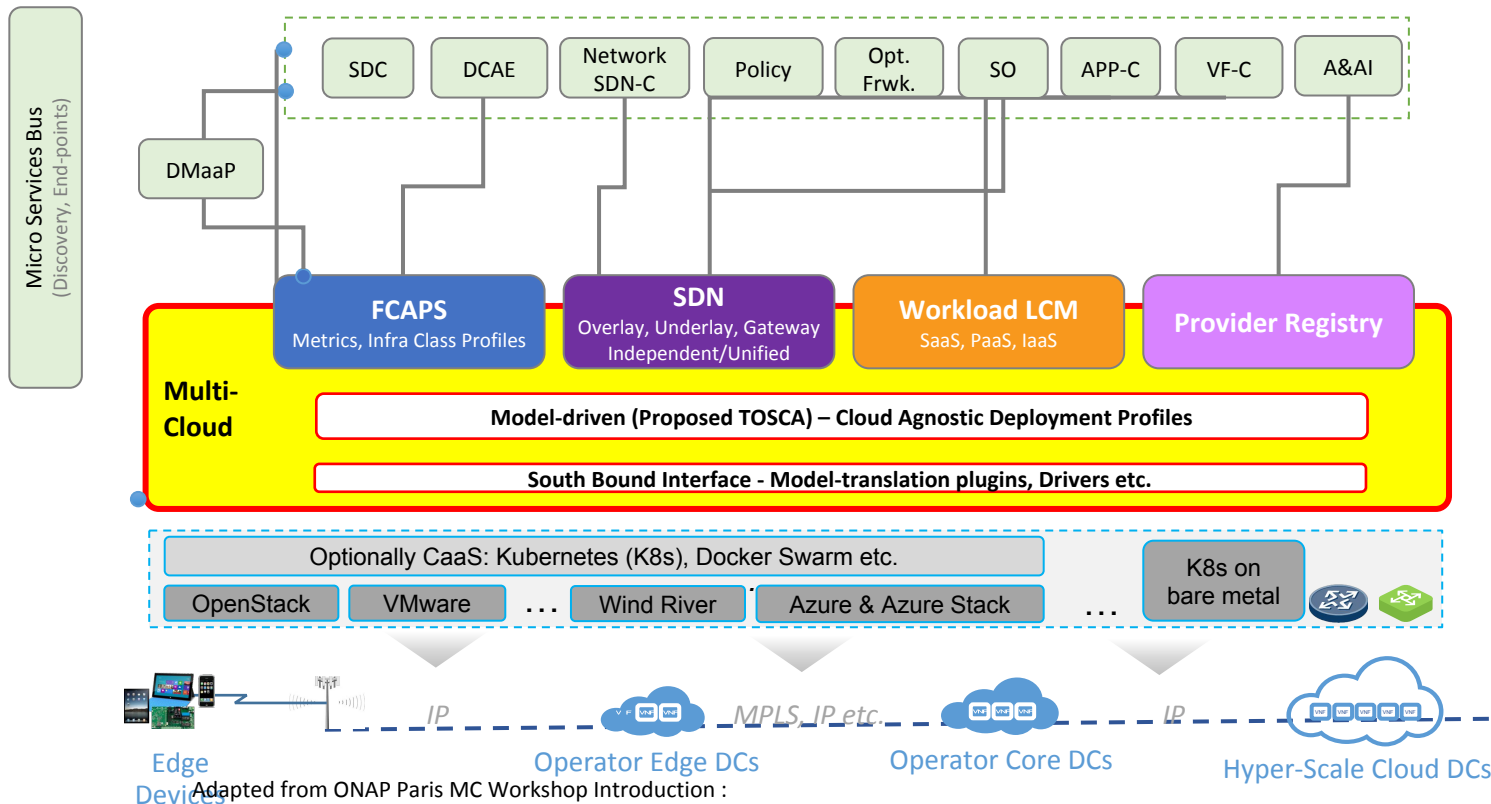
Where to convert	comment
caller(adaptor in SO/CCSDK)	Multiple place to host code/process it
Callee (in multcloud or new project)	Single place to host code/process it

Source code repository

Source code repository	comment
Subdirectory of one of the multicloud repositories	No appropriate repo
New subrepo for any container under multicloud	Doesn't align with the current multicloud practice
New subrepo under Multicloud per container tech	Each container technology support can be evolved independently. Least overhead for repo management
New repo for any container technology under new project	
New repo per container technology under new project	Each container technology support can be evolved independently.

backup

Multi-Cloud Reference Architecture – R2 & Beyond



Adapted from ONAP Paris MC Workshop Introduction : <https://wiki.onap.org/download/attachments/11928197/ONAP-mc-intro.pdf?version=1&modificationDate=1506518564000&api=v2>

Task list

- Registry
- Tosca-to-x translator (x=kubernetes)
- Integration
 - SO and CCSDK
 - For integration, with 1st phase, k8s api would be used directly
 - Later k8s api dependency would be removed with full fledged TOSCA model driven API
- Containerized VNF
 - Based on sample VNFs
- DCAE plugin: can be deferred