

CNFO – Simplified AAI CNF Model

Konrad Bańka (Samsung)
Łukasz Rajewski (Orange)
Seshu Kumar (Huawei)

CNFO - Summary

Executive Summary - Provide CNF orchestration support through integration of K8s adapter in ONAP SO

- Support for provisioning CNFs using an external K8s Manager
- Support the Helm based orchestration
- leverage the existing functionality of Multi cloud in SO
- Bring in the advantages of the K8s orchestrator
- Set stage for the Cloud Native scenarios

Owners: Lukasz Rajewski (Orange), Seshu Kumar M (Huawei)

Business Impact - Enables operators and service providers to orchestrate CNFs based services along with the VNFs and PNFs

Business Markets - All operators and service providers that are intended to use the CNFs along with PNFs / VNFs

Funding/Financial Impacts - Reduction in the footprint of the ONAP for CNF support.

Organization Mgmt, Sales Strategies - *There is no additional organizational management or sales strategies for this requirement outside of a service providers "normal" ONAP deployment and its attendant organizational resources from a service provider.*

REQ-341
Guilin

REQ-458
Honolulu

REQ-627
Istanbul+

AAI CNF Model – Overview

- Currently no CNF Resources information is visible in ONAP AAI
- Some interfaces are already implemented (Multicloud-k8s Status/Query API) that allow retrieval of detailed resources information
- Initial implementation of CNF Model in AAI should be simple and allow user to know about resources available and where to get their exact status from
- Long-Term solution should design appropriate CNF Resources in AAI, providing only the most important data and relationships about them.

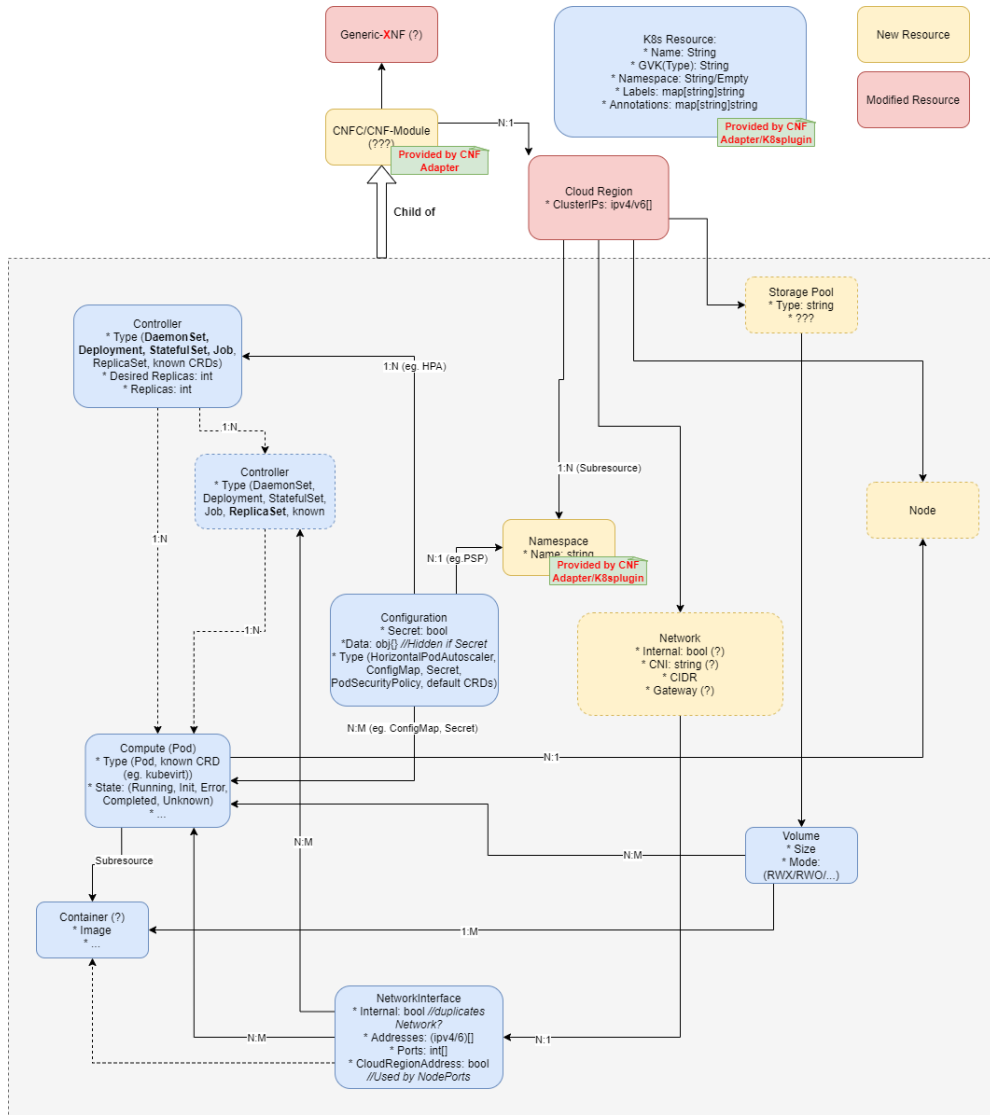
Simple AAI CNF Model – Istanbul

- Create additional AAI Object storing **Any** CNF Resource in K8s
- This new object (eg. k8sobject) should be related to generic-vnf/vf-module and cloud-region (subresource), using similar relationship matrix as vservers
- Data stored within AAI would be very generic, containing more or less only:
 - Name [string; Primary Key]
 - Group, Version, Kind [strings; Primary Key]
 - Namespace [string; Primary Key]
 - Labels [map[string]string or []string] *(depending on AAI capabilities)*
 - Annotations [map[string]string or []string] *(depending on AAI capabilities)*
 - **DetailsReflink** [string or object]
 - This field allows AAI Object consumer to specify query toward SO CNF Adapter to get full object data

Simple AAI CNF Model – Jakarta

- Provide **the most meaningful data** directly into AAI (according to predefined schema)
- Define **high level AAI objects** to describe CNF Resources with meaningful data for Closed-Loops, OSS/BSS and Operators
- Source of data for K8s objects may differ depending on actual resource:
 - Some objects can change their state over time, they're mostly related to given CNF/*VF-module* (eg. Pods)
 - Some objects would be static by nature, they're mostly associated with Cloud-Region (eg. Namespaces, Worker Nodes, CNIs)

Simple AAI CNF Model – Jakarta - Initial model



This model was drafted for reference and requires further tuning and agreement. We can discuss it further on next Modelling Subcommittee sessions or create dedicated ones, just like it happened during Guilin.

Main assumptions of this model:

- Update existing resources schema (Cloud-Region, Generic-VNF) and/or create CNF counterparts (VF-Module)
- Store Resources definitions about Controllers (eg. Deployments), Compute (Pods, Containers), Storage (PV/PVC), Network (Pod Interfaces, Services) and Configuration resources (eg. ConfigMaps, Secrets).
- Store unclassified resources like CRDs under “Uncategorized”-like object