OPEN NETWORK AUTOMATION PLATFORM

Application Service Descriptor (ASD) for K8s NFs

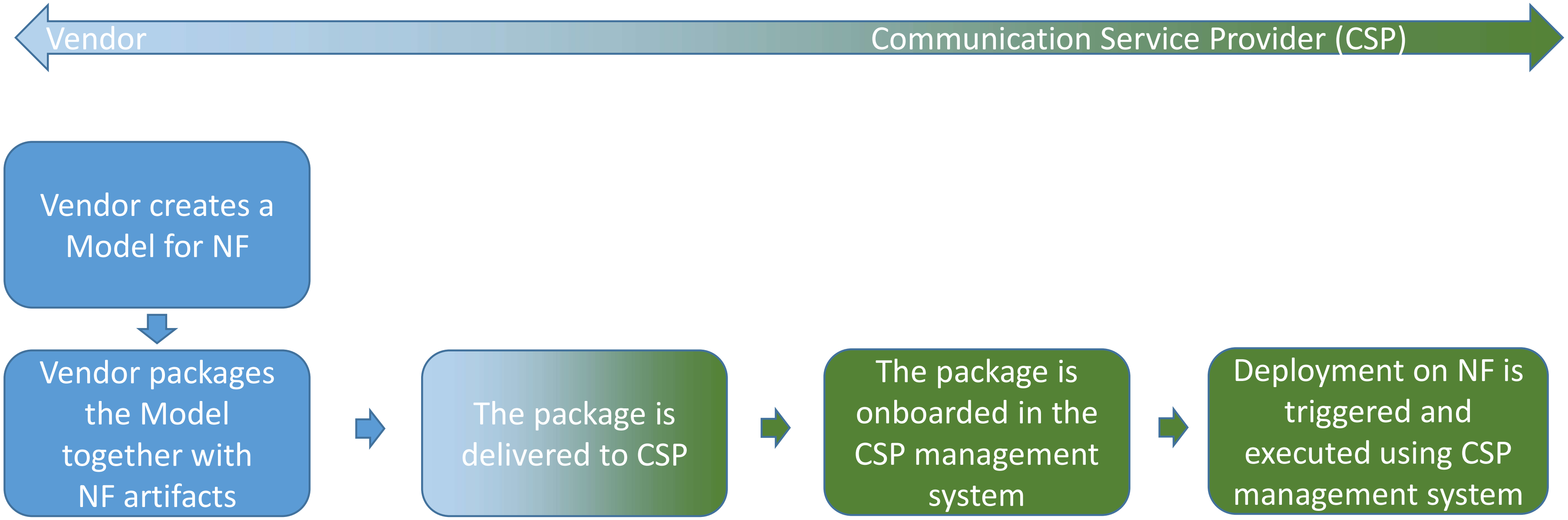
Presentation at Autonomous Networks - SDO Information Exchange meeting
Marian Darula, Byung Woo Jun, Zu Qiang, Ericsson

November 7 , 2022

Agenda

- Application Service Descriptor (ASD), Concept and motivation
- ASD PoC in ONAP
- Next steps

Typical NF modelling and packaging in LCM process



Vendor

Communication Service Provider (CSP)

Vendor creates a Model for NF

Vendor packages the Model together with NF artifacts

The package is delivered to CSP

The package is onboarded in the CSP management system

Deployment on NF is triggered and executed using CSP management system

Application Service Descriptor (ASD) Concept

- Utilizing cloud native modeling
  - E.g. based on helm but designed to support other methods, too
- Targeting deployment in Kubernetes environment
  - The Information model governed by Kubernetes APIs
- Providing just minimum additional information enabling orchestrator to take proper decisions
  - Relying of cloud native models and tooling
- Single Kubernetes cluster scope
  - Network service (NS) modelling linked with ASD(s) to cater for complex network solutions

Status of Art - ASD vs alternative NF modelling and packaging options

Plain K8S Helm

- De-facto standard solution
- Not good enough for orchestrator LCM as it is missing some information, e.g., cluster info, external network information, etc
- Good enough for CI/CD pipeline onboarding for GitOps use case

ETSI NFV VNF

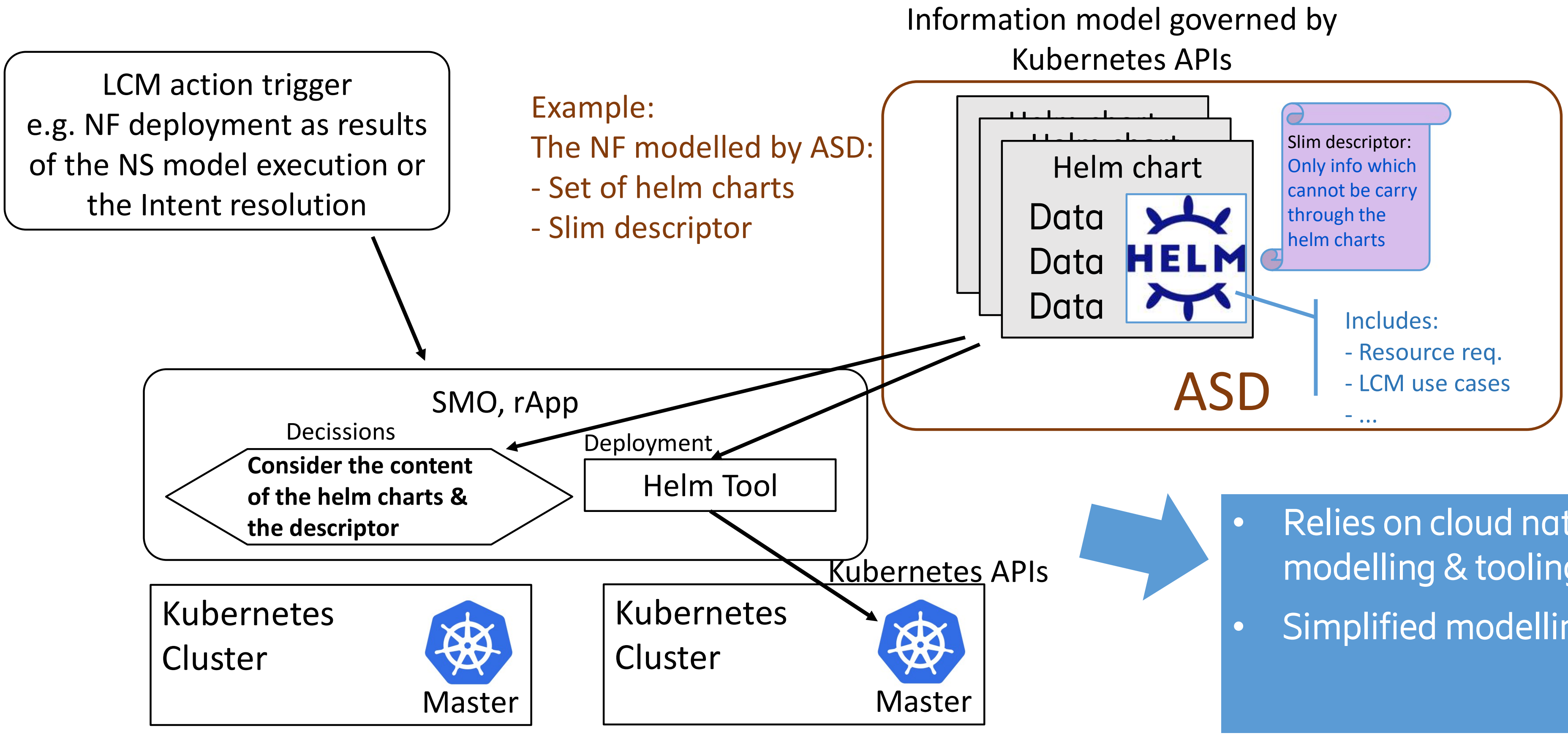
- Evolved from VM based modelling
- A static list of VNFCs.
- Each VNFC is linked to a set of HelmChart.
- VNFD and functional artifacts provides information for OSS components with largely duplicated information which is avaible in helm charts.
- Standardization is completed by ETSI

ASD

ASD

- A static list of deployment items.
- Each deployment item is a URL of a HelmChart or any other format.
- A slim descriptor with artifacts provides enough information for OSS orchestration;
- ASD concept is approved in ONAP
- In process of anchoring in ORAN for both NFs and rApps.

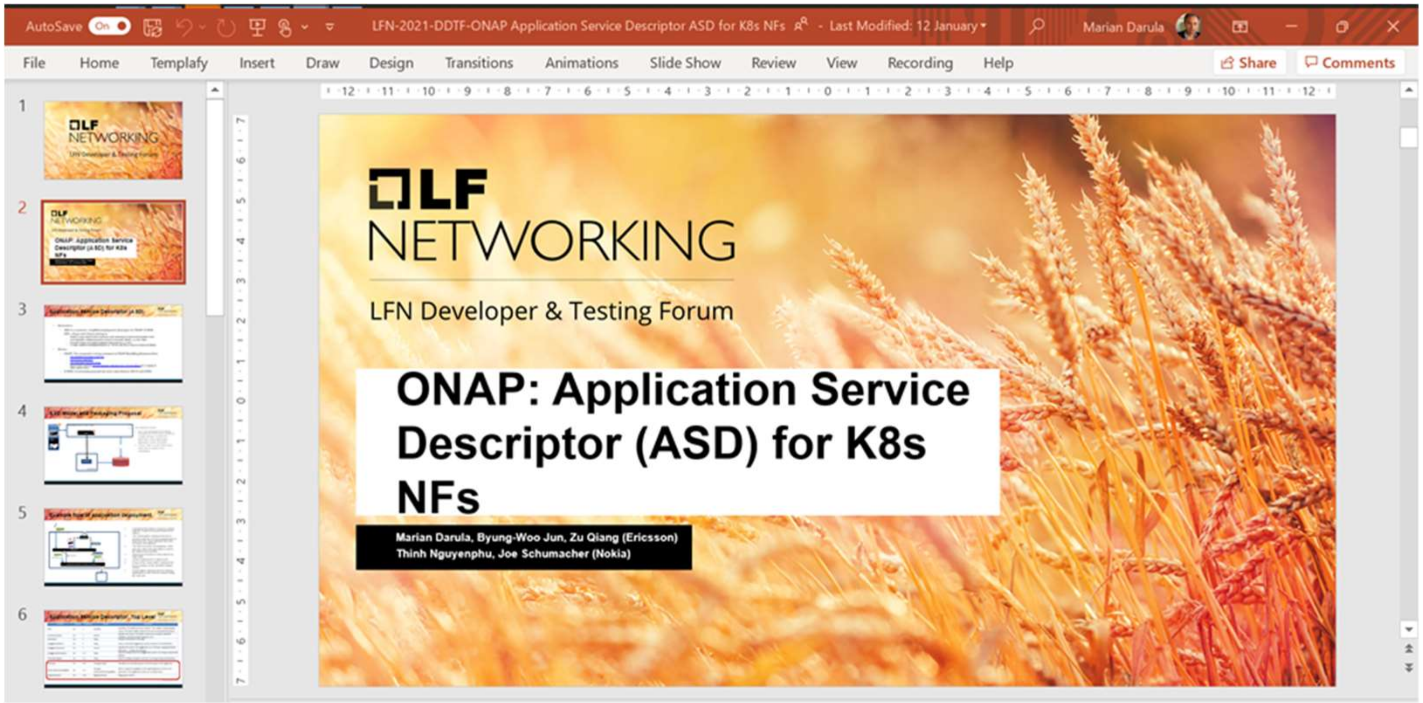
ASD based modeling and deployment of NF LCM flow, Helm example



- Relies on cloud native modelling & tooling
- Simplified modelling

Application Service Descriptor ASD, bit deeper dive

- Presentation material and recording from LFN Developer & Testing Forum 2022-01-12 LINK:



LFN-2021-DDTF-ONAP Application Service Descriptor ASD for K8s NFs - Last Modified: 12 January

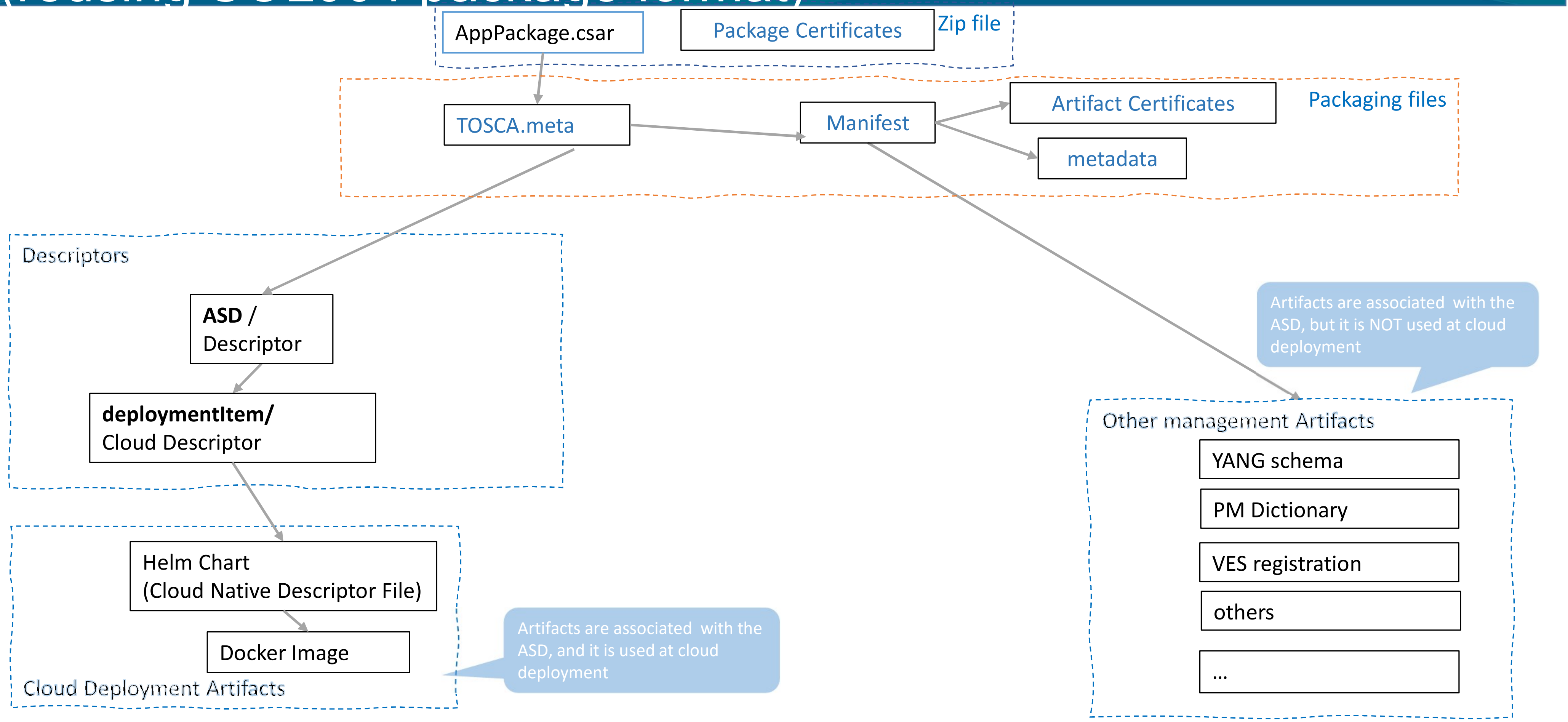
LF NETWORKING

LFN Developer & Testing Forum

ONAP: Application Service Descriptor (ASD) for K8s NFs

Marian Darula, Byung-Woo Jun, Zu Qiang (Ericsson)
Thinh Nguyenphu, Joe Schumacher (Nokia)

ONAP: Proposal NF onboarding package (reusing SOL004 package format)



Artifacts are associated with the ASD, but it is NOT used at cloud deployment

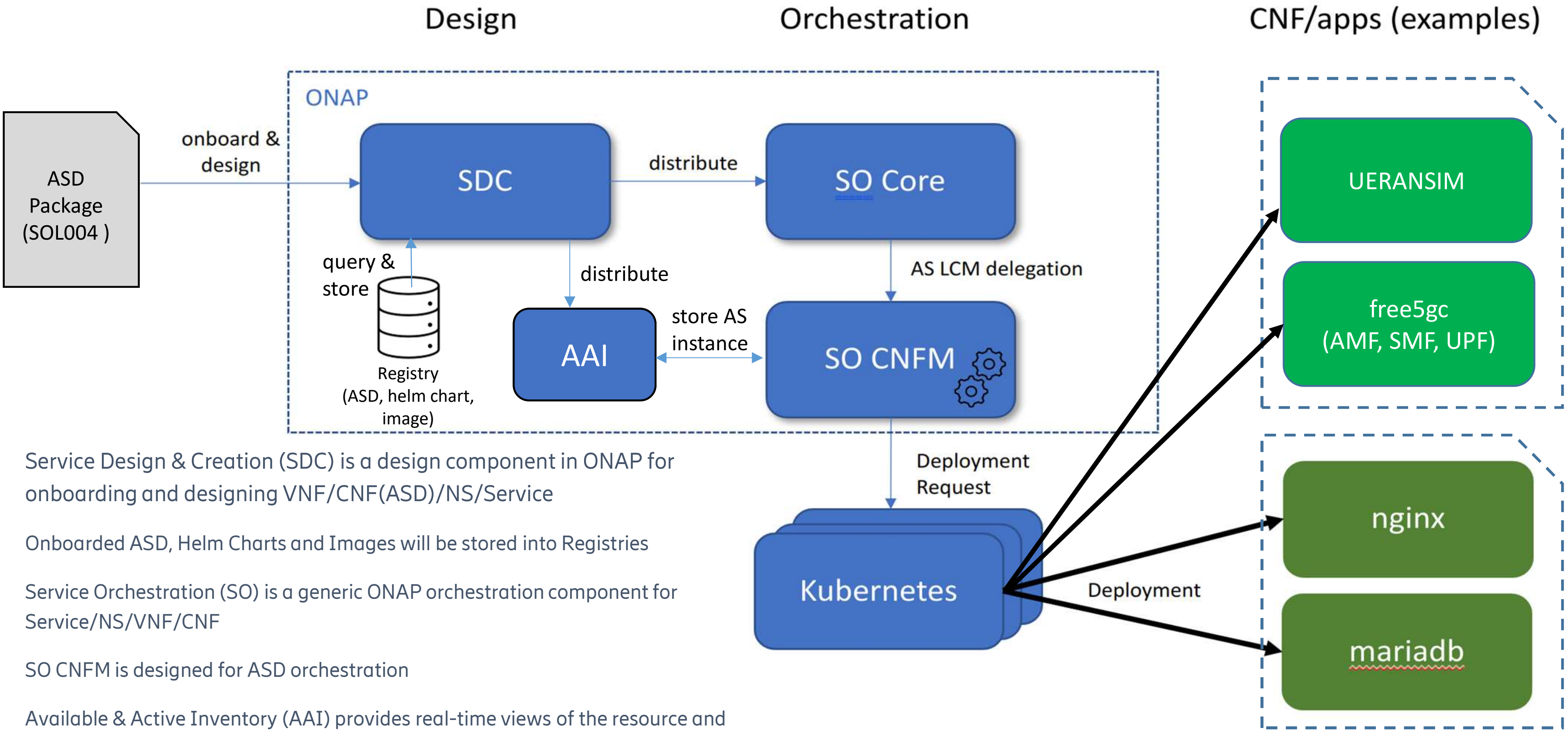
Artifacts are associated with the ASD, and it is used at cloud deployment

ASD Proof of Concept (PoC) in ONAP*

- The PoC was executed in ONAP framework to demonstrate viability of ASD concept for NF modelling and packaging.
- Sample applications like {MariaDB, nginx}, {Free5GC and UERANSIM} have been modelled and packaged following ASD concept.
- The packages have been onboarded in ONAP management system.
- Using ONAP components, the sample application modelled and packaged using ASD concept, have been successfully instantiated in the underlaying Kubernetes cluster.

*ONAP-Open Network Automation Platform, https://www.onap.org/

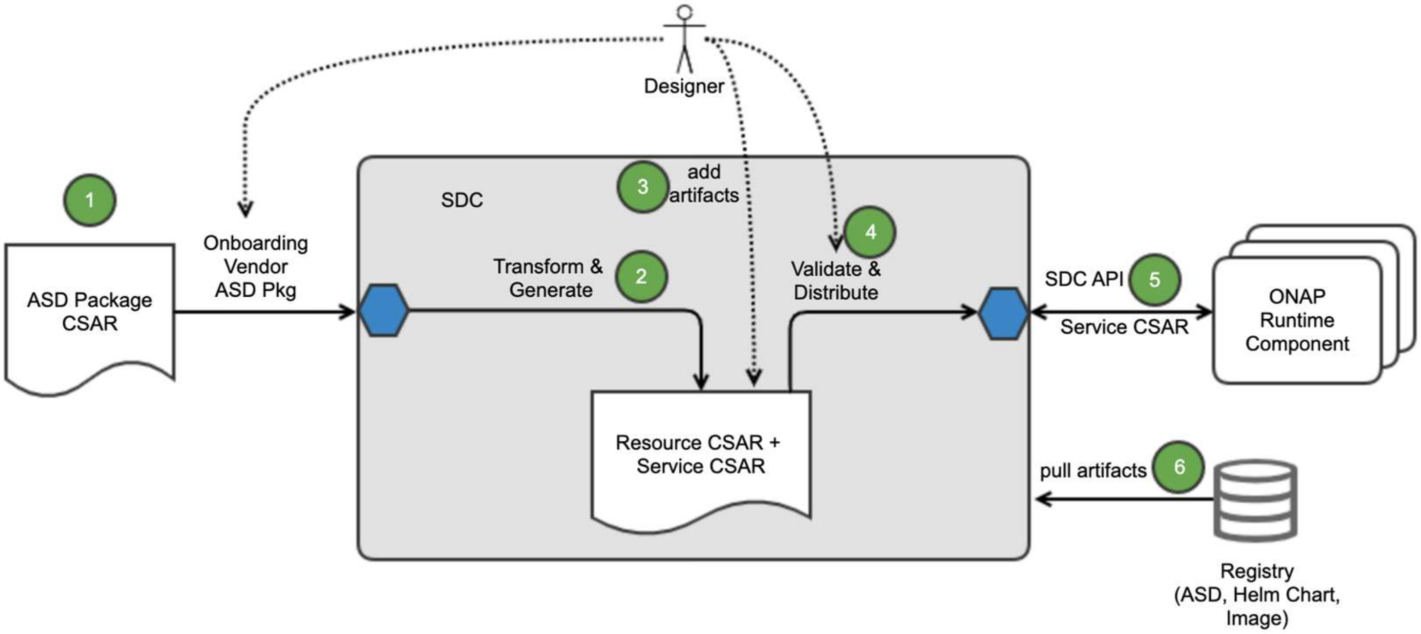
ASD Design & Orchestration Overview



- Service Design & Creation (SDC) is a design component in ONAP for onboarding and designing VNF/CNF(ASD)/NS/Service
- Onboarded ASD, Helm Charts and Images will be stored into Registries
- Service Orchestration (SO) is a generic ONAP orchestration component for Service/NS/VNF/CNF
- SO CNFM is designed for ASD orchestration
- Available & Active Inventory (AAI) provides real-time views of the resource and service instances and relationships

ASD Onboarding to ONAP SDC

1. Onboard vendor ASD packages and package validation
2. Transform onboarding ASD models into ONAP models and generate Resource CSAR and Service CSAR for distribution
3. Add additional artifacts such as vendor licensing models
4. Validate and decide for distributing the Service CSAR to ONAP runtime
5. Distribute the Service CSAR to ONAP runtime
6. Pull & store ASD, Helm Charts and Images from SDC to Registries (future implementation)



ASD Orchestration By ONAP SO

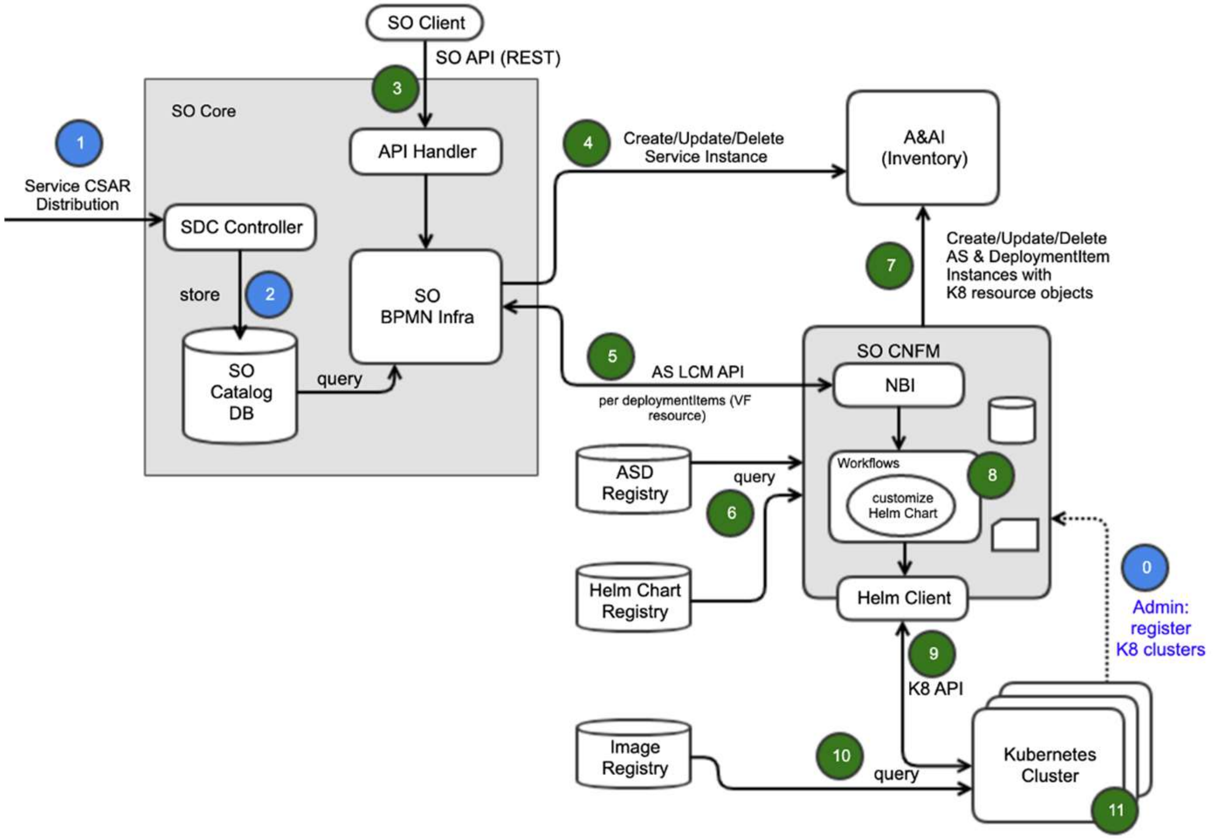
0. Register multiple Kubernetes Clusters

SO Core

1. Onboard ASD Service CSAR into SO runtime
2. Store ASD Metadata into SO Catalog DB
3. Support SO API Endpoint for ASD-CNF orchestration
4. Create/Update/Delete a Service Instance in AAI

SO CNFM

5. Use AS LCM APIs as NBI
6. Query ASD and Helm Charts from the registries
7. Store ASD & DeploymentItems instances with K8 resources (kind list) in AAI
8. Customize instance-level Helm Chart values, based on User Input and ASD
9. Deploy ASD-based CNFs to Kubernetes clusters through Helm Commands and Kubernetes APIs
10. Query Images from the registry
11. Applications deployed in the K8s and operational



Next steps

- Standardize ASD in O-RAN for both NFs as well as rApps
- Mandate support for ASD in O-RAN SMO and ONAP orchestration
- Fine-tune the ASD concept based on results from ONAP PoC for ASD modelling and deployment of sample applications

References

- Application Service Descriptor (ASD) onboarding Information Model, ver. 1.0.
https://wiki.onap.org/pages/viewpage.action?pageId=128715159

- Application Service Descriptor (ASD) Resource Data Model
https://wiki.onap.org/display/DW/Application+Service+Descriptor+%28ASD%29+Resource+Data+Model

- ASD Onboarding and Orchestration PoC
https://wiki.onap.org/display/DW/ASD+Onboarding+and+Orchestration+PoC