ONAP DCAE Transformation proposal

12/09/2020

Vijay Venkatesh Kumar
(vv770d@att.com)

Agenda

- Introduction
- DCAE Architecture (Guilin)
- Decentralized DCAE Architecture (Target proposal)
- Migration Plan
- Honolulu release Objective
- ONAP Cross-project Impacts

Transformation Objective

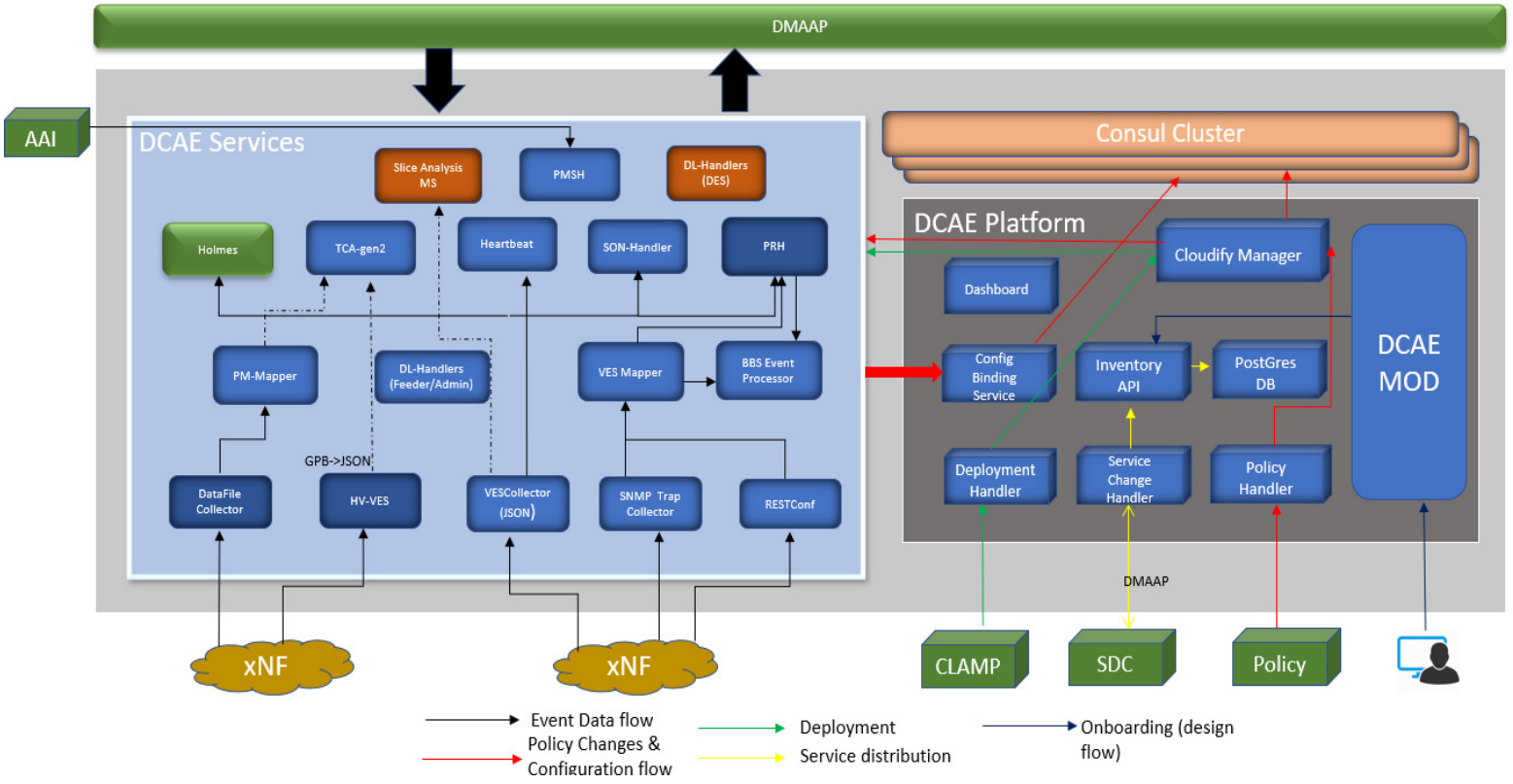
Simplify DCAE architecture by offloading platform functions into generic k8s/cloud native function to be controlled based on Microservice deployment via helm

Present Mode of Operation

- DCAE platform provides standardized functions for DCAE MS's
  - Configuration management and retrieval through CBS
  - DMaap Topic provisioning
  - Consolidated view of DCAE deployments (across clusters)
  - Platform API for LCM of DCAE MS
  - Policy Interaction abstracted and managed by Policy Handler
  - Postgres DB initialization

- DCAE MOD generates Cloudify Blueprints (used for dynamic/on-demand deployment)

ONAP DCAE Architecture (Guilin)



Cloudify Manager
Primary orchestrator within in DCAE through which all DCAE MS are deployed. Cloudify, through its arsenal of plugins, is capable of relationship-based orchestration in many levels and cross different technologies.

Deployment Handler
Provides API for deploying DCAE MS into DCAE. Used by CLAMP and Dashboard

InventoryAPI
Provides API for storing and retrieving Service blueprints into Postgres

Service Change Handler
Retrieves DCAE specific blueprint composition distributed by SDC/DCAE-DS

Policy Handler
Retrieves configuration for DCAE components from Policy Engine. Listens on updates from Policy, identifies target mS and pushes update into mS (or Consul)

Dashboard
An UI for operation team to manage, deploy and track services component in DCAE

Config Binding Service (CBS)
Standard API layer for DCAE components to retrieve configuration (from Consul or other sources)

Consul
Provides KV store for DCAE MS configuration. Service registration is used for selected components

DCAE MOD (NiFI)
Design platform for onboarding and service composition creation

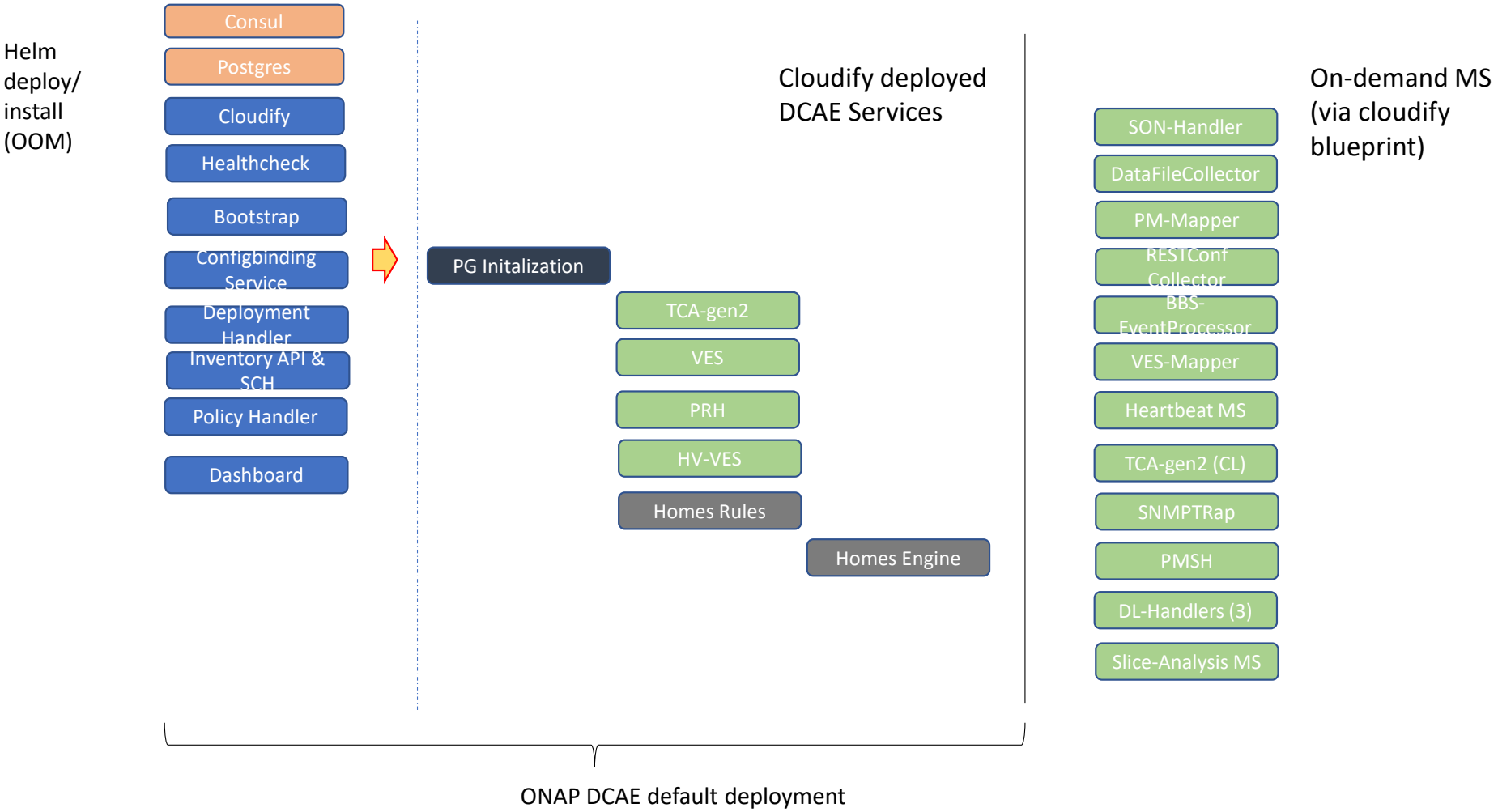
Benefits through Cloudify architecture not fully utilized in ONAP

- Infrastructure management support
- Support Hierarchical and distributed deployment (with central and regional/Edge site)
- Model driven approach enabling design flow integration
- Support containerized and VM based workload deployment on heterogenous cloud environment
- Supports service composition design (flow based) and instantiation of multiple related mS
- Dynamic DMAAP MR/DR provisioning capabilities part of orchestration.

Challenges

- Added complexity on DCAE Platform maintenance due to cross component dependency
- Tight coupling with Cloudify orchestration introduced onboarding challenge (maintaining k8s plugin <-> blueprint-gen compatibility)
- Community adoptions not diverse
- Security patches, ONAP compliance, S3P goals dependent Cloudify support
- Kubernetes and Cloudify state consistency

ONAP DCAE Deployment (Guilin Release)



ONAP Community Request

- Helm adoption for alignment with rest of ONAP deployments
- Remove second-level orchestration for DCAE Service components to remove underlying complexity

Transformation Goals

Decentralization of platform function

Remove centralized external Config Store

Remove Cloudify centric

Adopt industry standard and tools

Easier onboarding and deployment of DCAE mS

CI/CD based workflow automation

Pros

- Simplified & Open architecture
- Connectivity on distributed deployment
- Address latency and throughput concerns
- Cloud native solution enabling broader community support

Cons

All DCAE Service component impacted with migration

Transformation Impact for DCAE Services

Application Configuration Management	Deployment	Policy Interaction	DMaap Topic/feed provisioning	MOD (Design Platform)
• Switch Microservice employing Configbinding-Service API's (for configuration fetch from Consul) to retrieve config through K8S Configmap • Alternate option via K8S operator to replicate current platform functions	• Build Helm chart for existing services (with configuration defined as resource definition under helm). Generic template being explored • Common helm registry (ONAP CI integration) • OOM/Service (API support) for dynamic helm chart deployment post ONAP installation	• Build SDK or sidecar container to periodically fetch and retrieve active policy configuration into application container (MS impact to retrieve policy configuration outside of CBS api)	• K8s operator for components require secure dynamic AAF based topic • Facilitate Native/Kafka integration for required DCAE MS (Note: Impact of Istio/Service mesh to be assessed)	• Onboarding, Deployment & Catalog Management for helm based component • Generation of helm chart based on meta-data/spec onboarded • Integration with native Helm registry (which will be used for ONAP CI/CD or OOM/Service)

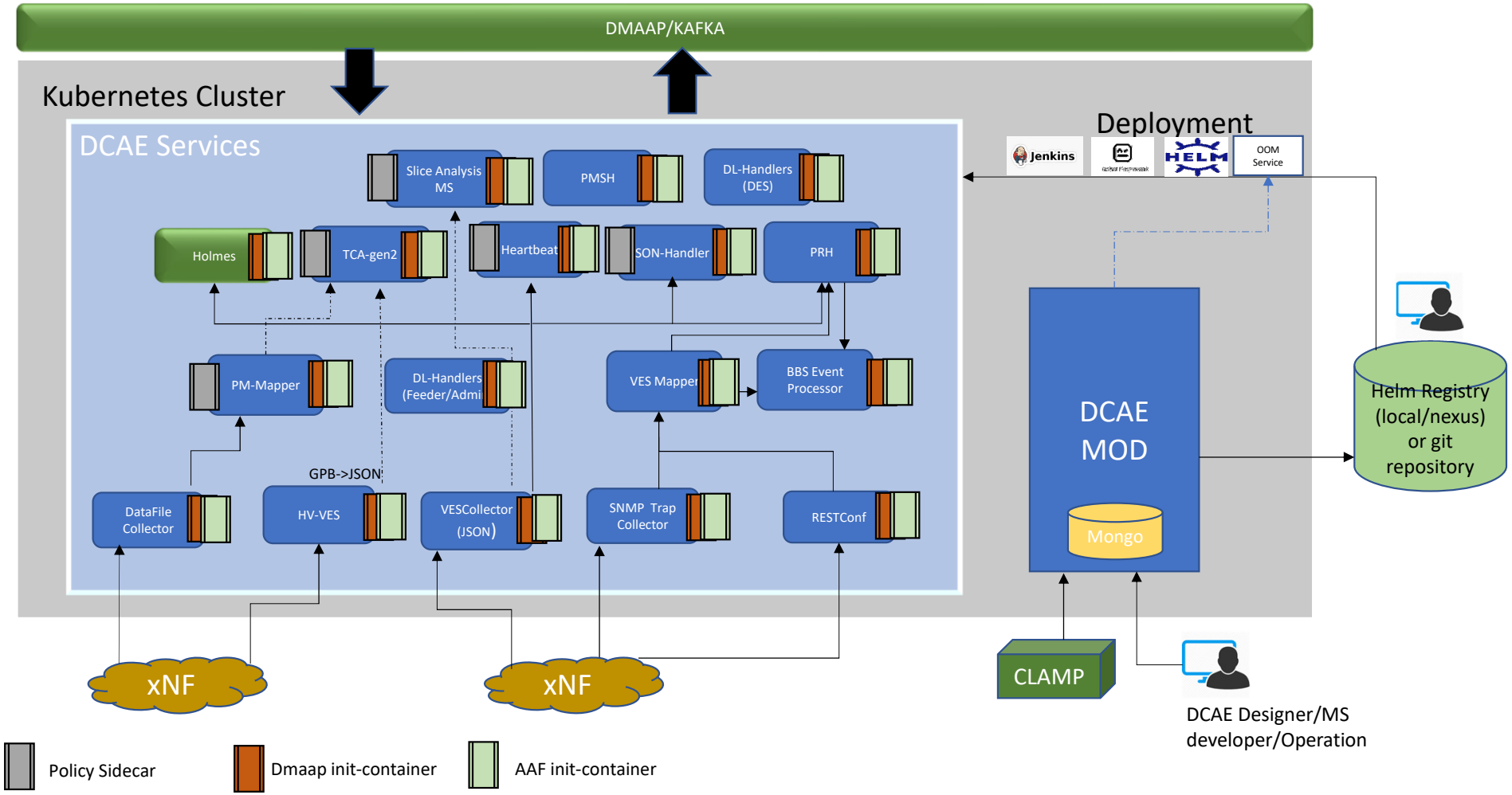
ONAP Phasing Plan (Proposal)

Phase 1 (H release candidate)
- Build standalone features for Helm integration
- Introduce framework & CI pipeline for supporting target architecture
- Design for transformation feature/work as POC

Phase 2 (I release candidate)
- Pilot few DCAE service components to leverage new feature
- Harden DCAE-MODv2 for helm support and CI for dynamic deployment

Phase 3 (J release)
- Migrate all existing/active DCAE services to helm
- Deprecate Cloudify/associated platform components

ONAP DCAE Architecture (FMO)



DCAE Honolulu release Plan

- Cloudify remains primary orchestration for dynamic deployments (MOD/CLAMP flows)

- Migrate bootstrap service components to Helm (continue Consul/CBS)
  - Build generic common helm template (to be used by VESCollector, TCAgen2, HV-VES, PRH, Holmes*)

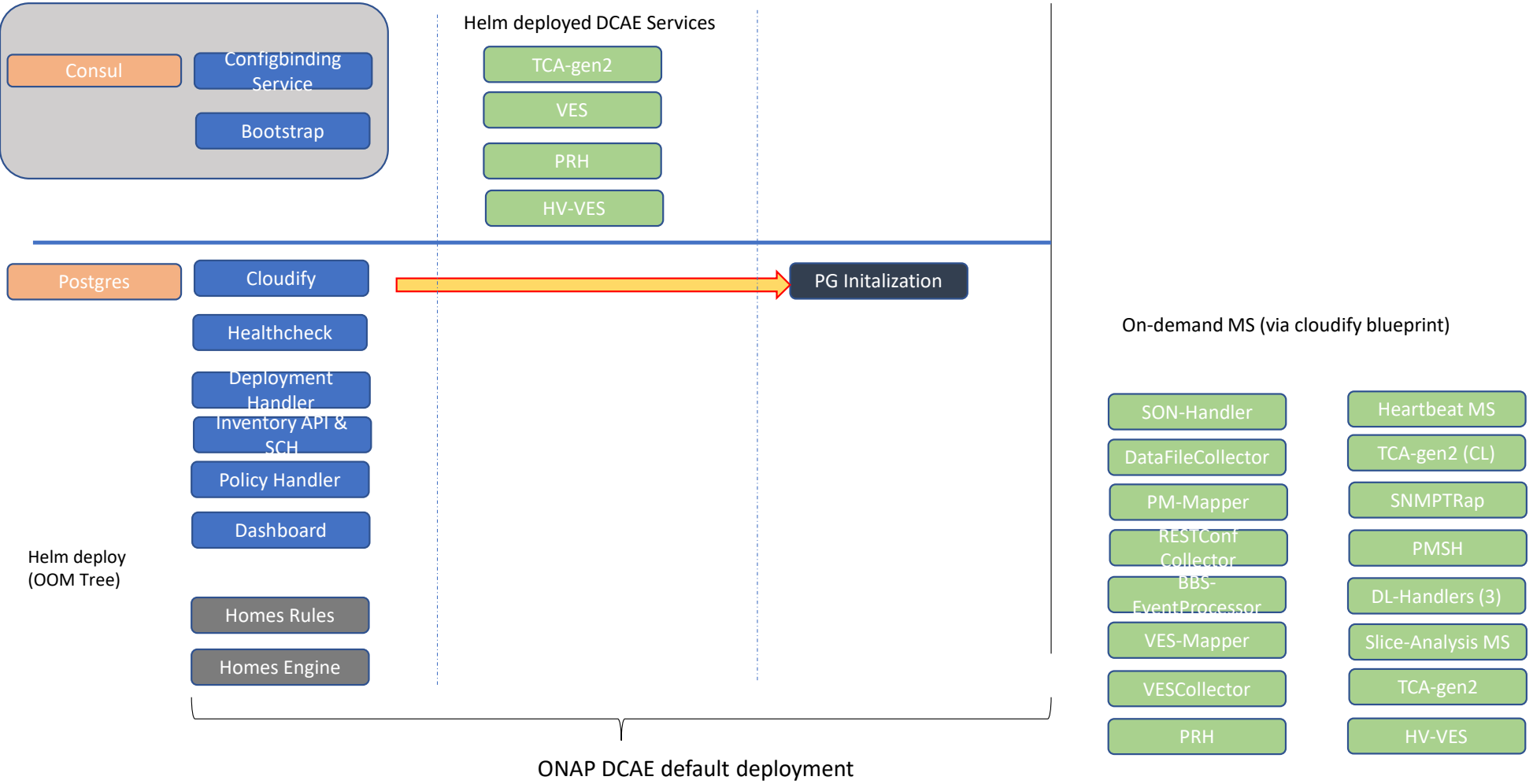
- Create Helm Repo within DCAE and trigger separate gating (OOM/integration and management TBD)
  - ONAP/Jenkins integration for helm chart build and push into ONAP/nexus
  - Dependency on oom/common to be available under ONAP/nexus

- Design configuration management for service component outside of Consul/CBS

- Building current DCAE platform function as sidecar/init container functionality *(Stretch-goal)*
  - Dynamic Topic/feed (DMAAP) provisioning through helm & K8s operator
  - Policy Handling

- DCAE MODv2 Enhancement for Helm support *(Stretch-goal)*

ONAP DCAE Honolulu Deployment



ONAP Cross project impact

- OOM
  - LF nexus integration for hosting Helm chart & Versioning
  - OOM service for supporting on-demand deployment from external registry (after main ONAP)
  - Support of K8s operator for DMaap topic/feed provisioning

- CLAMP/Policy
  - Design integration through MOD for Helm deployed component
  - Deployment integration through MOD (for Helm flow)
  - Helm override support in CLAMP GUI

- DMAAP
  - DMaap Operator Support (OOM/DCAE/DMAAP – TBD)
  - Support for Native/Kafka integration
  - Topic/feed provisioning in ONAP independent of AAF
    - Unauthenticated topic supported; feeds require access control for subscribers.

Call for Support

- ✓ More company engagement/resources to support transformation efforts.
- ✓ Features can be developed independently and integrated with MS during deployment

Seeking community contributions/support on below

- ❖ ONAP CI/CD integration (for helm chart)
- ❖ OOM/Helm service for supporting dynamic deployment
- ❖ Building sidecar/init container functionality
  - Dynamic Topic/feed (DMAAP) provisioning through helm & K8s operator
- ❖ Design for Standardized Configuration management for application
- ❖ DCAE MOD Enhancement for Helm based components (plugin or adapter for helm generation)

Next Steps

DCAE Weekly Meeting - https://lists.onap.org/g/onap-meetings/viewevent?repeatid=31910&eventid=925967&calstart=2020-12-02

REQ-479 - DCAE Transformation to support Helm

EPIC JIRA - DCAEGEN2-2488: DCAE Transformation to support Helm

Wiki - https://wiki.onap.org/pages/viewpage.action?pageId=92997528

Questions ?

ONAP Future release scope

Phase 2 (I release candidate)

- **New services introduced into ONAP/DCAE will be deployed on cloudify independent path**
- Deprecate DMaap Plugin
  - Migration DFC and PM-Mapper to use new Helm/K8s operator based Dmaap topic/feed provisioning
- Migrate bootstrap service components to leverage sidecar/init-container feature
  - TCA-gen2 to leverage Policy side-car
- Migrate bootstrap service components to support Configmap driven config management
  - VESCollector, TCAgen2, Holmes
- Harden DCAE-MODv2 helm support and CI integration for dynamic deployment*

Phase 3 (post I release)

- Migrate all remaining services to Helm w/sidecar and init container
- Deprecate DCAE Cloudify based platform components

Backup

Policy reconfiguration flow

1. Microservice deployment – with Policy ID
2. Env and config info stored in ConfigMap
3. Sidecar is attached to each POD
4. Sidecar will store input Policy info in persistent volume
5. Sidecar will listen for policy updates
6. Following flows will be supported:
   I. New configuration
   II. Retrieve configuration from sidecar, filter and store in PV
      I. Notify POD
      II. Provide the updated info
      III. Microservice decides when to apply it
   III. Reconfigure (add/update/delete) via either:
      I. Deployment
      II. Policy