ONAP Operations Manager for Lifecycle Management of the ONAP Platform

A Proposal to Linux Foundation ONAP Project

Submitted by AT&T
Updated June 6, 2017

Proposed ONAP Project

- Project Name: ONAP Operations Manager
- Description: Propose to define the ONAP Operations Manager to manage platform level lifecycle management or OAM (Operations, Administration and Management) in an organized and consistent fashion
- Key functions
  - TOSCA based orchestration for deployment automation of platform modules
  - Inventory and state management of all platform modules
  - Execute corrective control actions
  - Automate change management
  - Support multiple ONAP instance management
- Benefits
  - Provides framework for full OAM of ONAP platform(s)
  - Drives agility into ONAP deployment automation and change management
  - Monitor health and state of ONAP and enable control loop actions
  - Improves DevOps experience and platform support
- Use Cases
  - ONAP platform and/or component deployment and instantiation
  - Monitor state (health) of components/subcomponents and services
  - Control loop actions
  - Change management – software upgrades

ONAP Operations Manager for ONAP Lifecycle Management

The ONAP Operations Manager (OOM) is being proposed as a new platform component that deploys and manages the lifecycle of the ONAP platform and all of the software required to make it operational.

Problem Statement:

- Disparate management and control of all the ONAP components as individual entities
- Platform deployment of ONAP components is done via simple script not easily rolled back, does not allow for troubleshooting deployment failures, and does not support change management
- Inadequate Platform lifecycle management – such as inventory, state management, control loop actions – is hindering platform advancement
- Implementations are currently tightly coupled with each component

ONAP Operations Manager (OOM)

Scope

- **Unify deployment, management and control capabilities for all of ONAP**
  - all ONAP components, shared software modules and portal applications
  - micro-services used to manage resources and services
  - messaging, event and data management fabric
  - operate by a single 'logical' team via one runbook (as if by a single user)
- **Scalable solution to manage increased volumes and specialization**
  - deploy and manage additional ONAP instances as volume/demand grows
  - manage OOM scale via hierarchy of instances (as needed)

ONAP OOM Use Cases

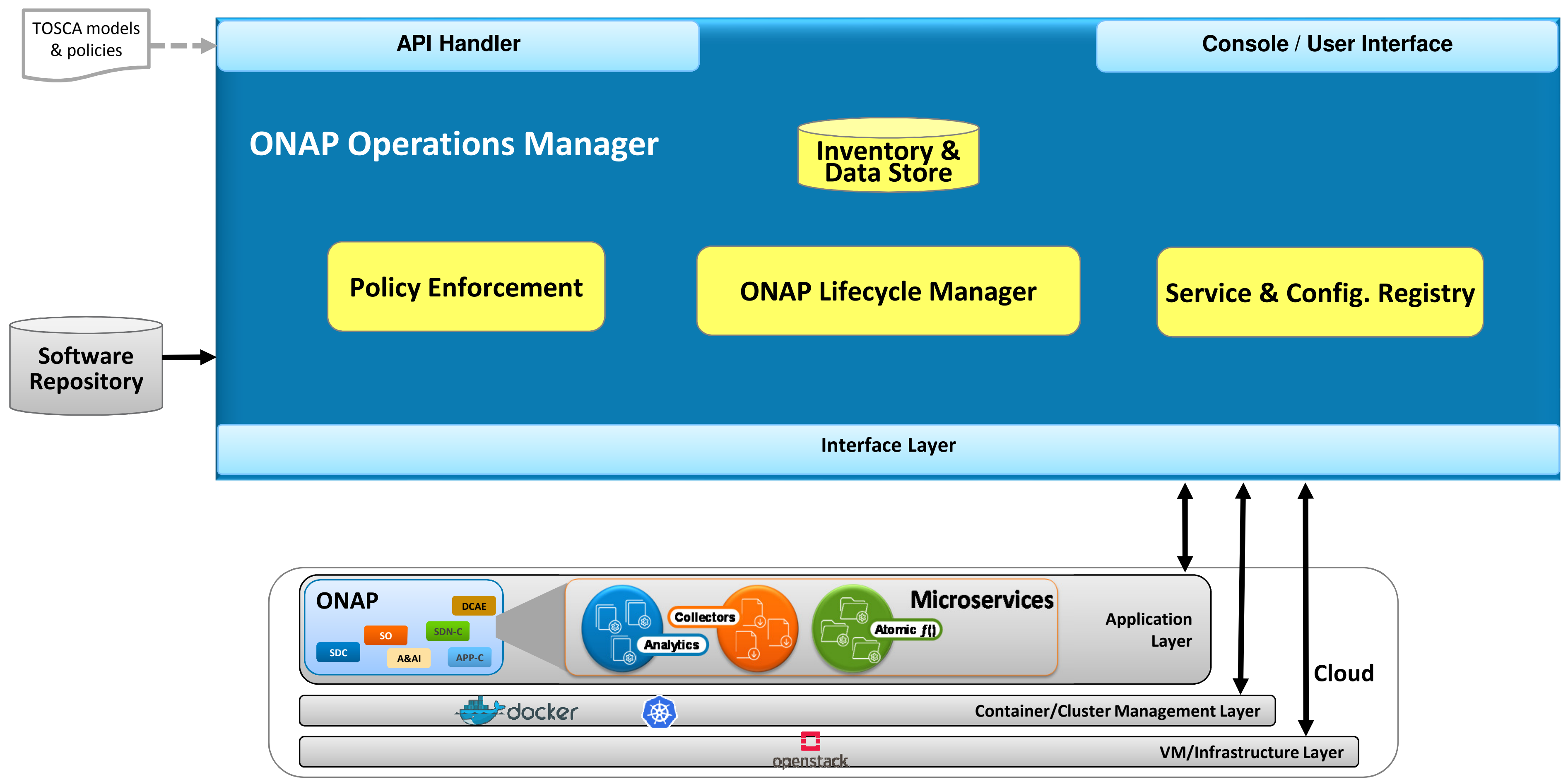
2017 Use Cases (Initial)

1	***Instantiate ONAP Components***	• Stand up an ONAP platform include all or a subset of components via single build/deploy method • Use of TOSCA based orchestration to enable deploy and un-deploy of a platform in test/prod environment • Integrate with health checks to verify deployment success
2	***Change Management***	• Leverage TOSCA orchestration of auto-deployment to support updates of individual software modules
3	***Monitor the State of ONAP***	• Use of health-checks to determine the health and state of each ONAP component and subcomponent • Discovery of ONAP components and subcomponents via self-registration process • Enable health-check to validate the deployment
4	***Control Loop Functions***	• Initial capabilities include start, stop, and restart, with additional control loop functions to be defined
5	***Micro-Services Onboarding***	• Initial micro-services support will be focused on DCAE related analytics & data collectors • Additional micro-services (include ONAP Portal UI apps) will be introduced later.

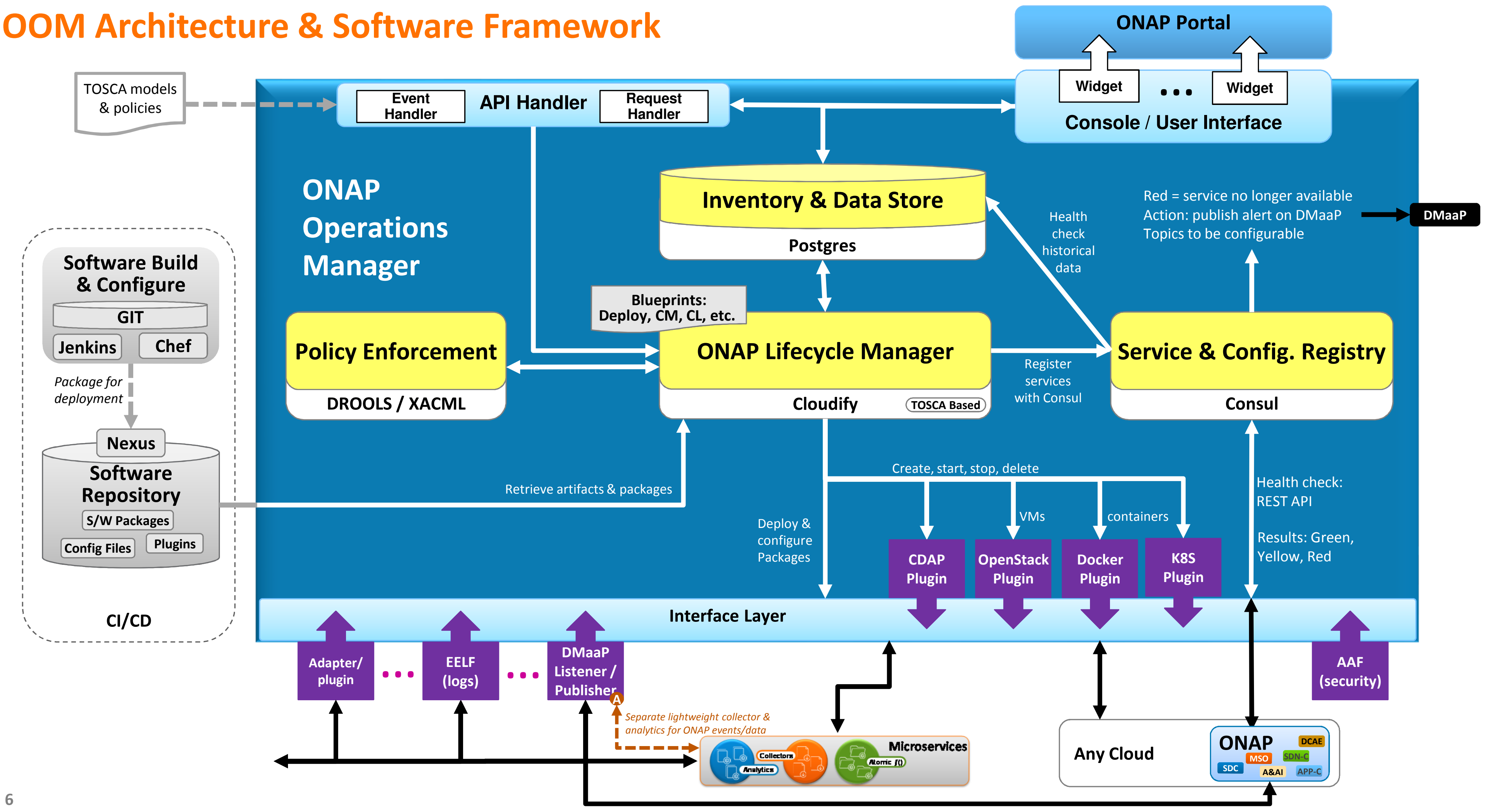
2018+ Use Cases

6	***Policy Framework Integration***
7	***Additional Control Loop Functions*** *– restart, scale, etc.*
8	***OSS Integration*** *– ticketing, performance, etc.*
9	***Additional Microservices*** *– portal, shared ONAP, optimization, etc.*
10	***Onboard BU/Services/Ops apps***
11	***Evolve Container Support*** *– additional Kubernetes features, etc.*
12	***Security integration (ASPR, certificates)***
13	***Network integration***
14	***Management of multiple ONAP instances***
15	***etc.***

ONAP Operations Manager Functional View

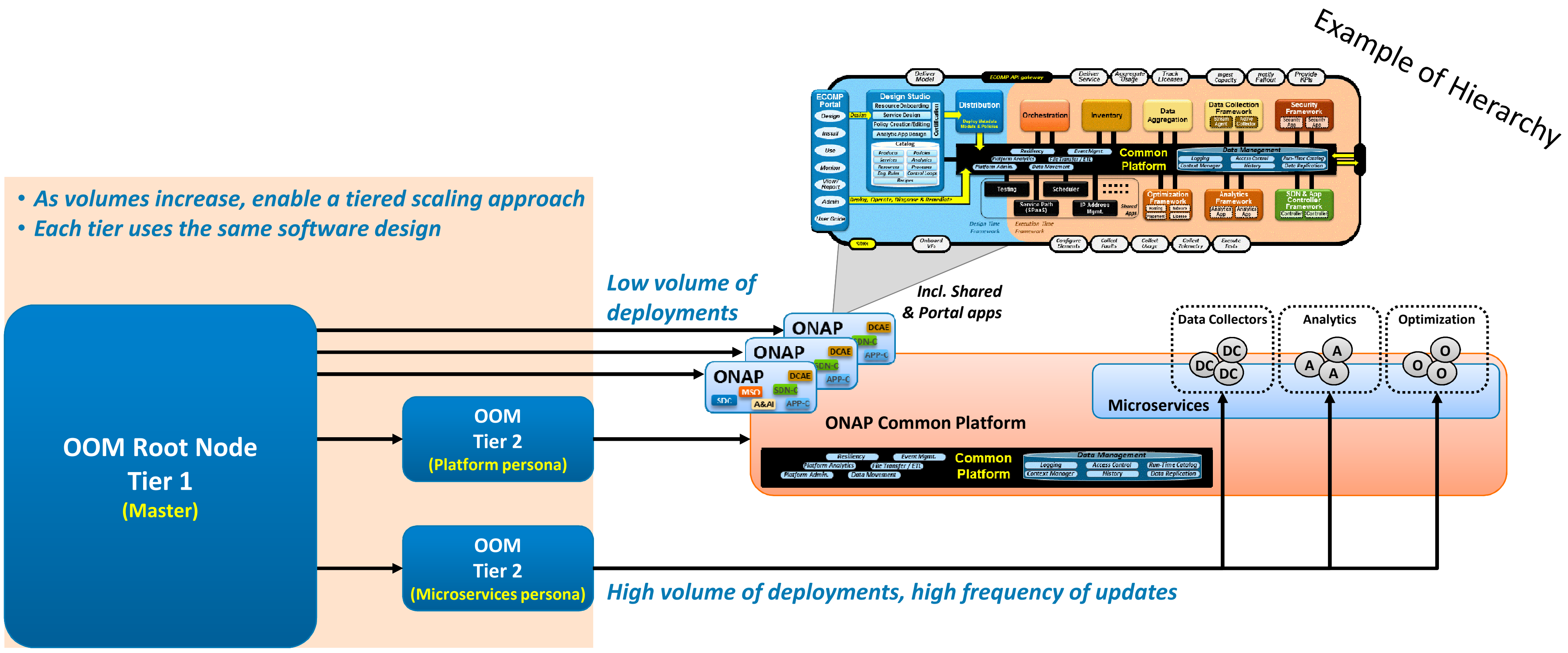


OOM Architecture & Software Framework



ONAP Operations Manager (OOM) – Hierarchy to Accommodate Scaling & Specialization

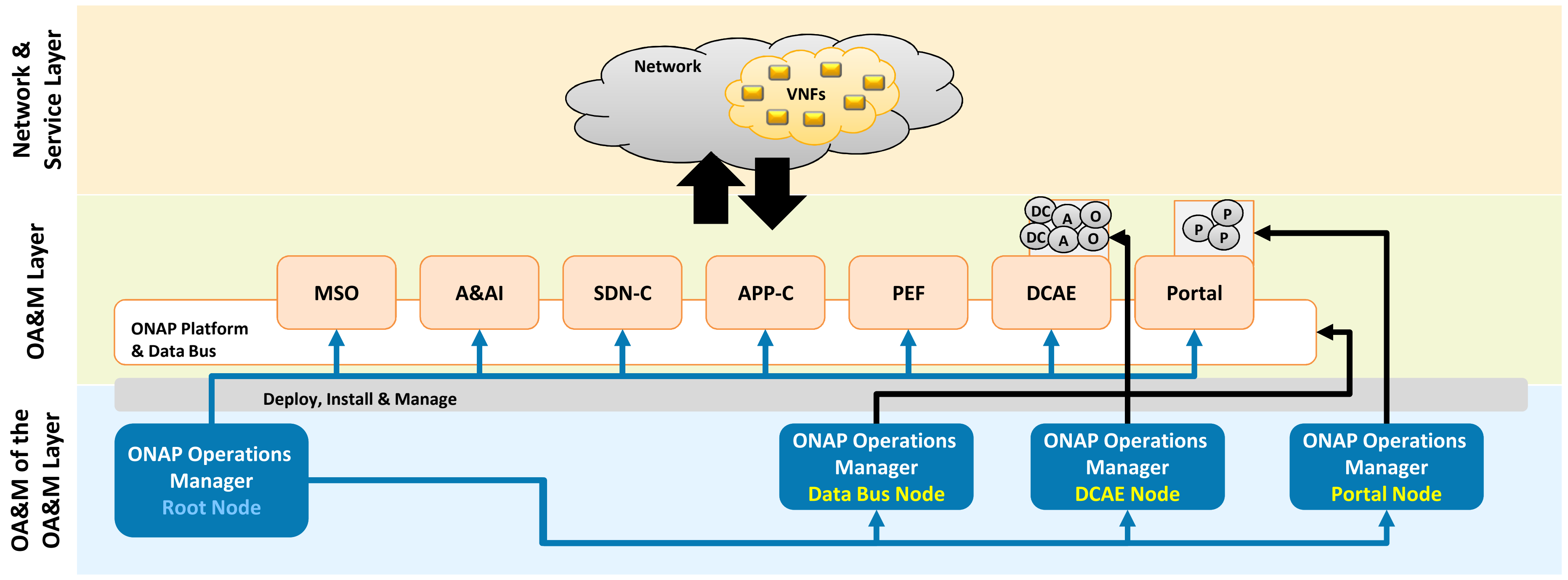
- *As volumes increase, enable a tiered scaling approach*
- *Each tier uses the same software design*

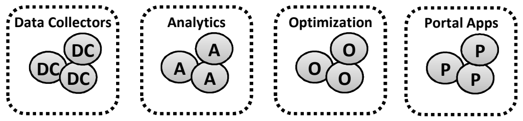


Notes:
- The role and scope of ONAP Operations Manager Nodes are not static.
- For initial contribution, the Tier 1 Root Node and the Tier 2 DCAE Node will be provided.

ONAP Operations Manager (OOM) vs. ONAP OA&M Functions vs. Services

OA&M = Operations Administration & Management

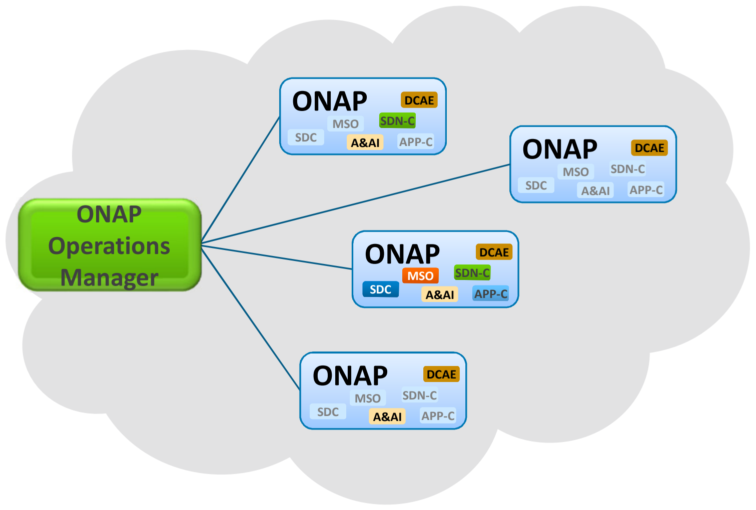




Notes:
- The role and scope of ONAP Operations Manager Nodes are not static.
- For initial contribution, the Root Node and the DCAE Node will be provided.

ONAP Operations Manager (OOM) Architecture

- Controls multiple ONAP platforms
  - Single ONAP Operations Manager (OOM) may manage dev/test, IST, Prod Support, Production
  - Single OOM may manage multiple regional ONAP instances
- Orchestrated deployment and lifecycle support
- Subset and entire ONAP platform can be instantiated



Back-Up Slides

ONAP Operations Manager Benefits

- Unify the deployment and lifecycle management of all ONAP platform components
  - Eliminate component specific deployments
  - Manage the dynamic onboarding of micro-services
  - "X" # of components → single, configurable deployment
- Automate platform deployment, scaling and management
  - Consolidate the release build process across components
  - Automate health check and state management of inventory of ONAP components, software modules, and services
  - Automate scale of ONAP components
  - Automate fault, performance and outage recovery
- Simplify remaining manual operations
  - Single dashboard
  - Monitor consolidated state with drill downs by component/module/microservice
  - Reboot, reset or redeploy any/all components from a single place
  - Enable consistent platform operations and management via a single playbook
  - Provide the "database of record" for the state of ONAP components
- Provide ONAP Platform Support/DevOps teams a consolidated view and user experience
  - Current ONAP platform instances and their components
  - Component and subcomponent current health and history
  - Automated and manual lifecycle management tasks – restart, scale, resiliency, etc.
- Enable end to end agile delivery via CI/CD integration with configurable orchestration
- Facilitate agile testing through dynamic creation of test instances (deploy/un-deploy/create subset)

ONAP Operations Manager – Architecture Principles and Design Goals

Principles

1. Simplicity… more reliable, more available, less moving parts to break or manage
2. Develop as an iterative, open-sourced solution… plan for collaboration
3. Deploy and manage the collection of ONAP components as a single unit (a platform)
4. Automate the deployment and management as much as possible
5. Simplify whatever is left after automation via extensive but terse support for manual functions
6. Overall solution should have the same open source as other ONAP components
7. Design recursion out of the solution… "the buck stops here"

Design Goals

1. Use open source and provide to open source where we need to build
2. Deliver a modular design
3. Design for change, scale, flexibility and obsolescence of both software technology and operation
4. Enable technologies and software packages to be swapped out with minimal impacts
5. Enable a management model that provides agile onboarding of new and varying objects controlled/managed
6. Unify deployment, management and control capabilities of the platform
7. Operate by a single 'logical' team via one runbook (as if by a single user)
8. Use, to the extent possible, the same technology stack (i.e., dbms) as the ONAP platform
9. Accommodate use of diverse fundamental operations that enable variability where it makes sense