1. Application LCM Use Case

Introduction

The Application LCM use case is being defined to demonstrate how to package an application for standardized onboarding and the initial states expected to be required to move through the SMO states from Onboarding to Ready for Deployment.

Further details:

- Cherry Application LCM Step 1 - Create Application Package
- Cherry Application LCM Step 2 - Package Validation
- Cherry Application LCM Step 3 - Catalog Package
- Cherry Application LCM Step 4 - Create Config

Background and Goal of the Use Case

As the O-RAN Alliance defines the standardized way to onboard applications for the Non-RT RIC (rAPP) and Near-RT RIC (xAPP), with or without Machine Learning, the OAM Architecture is defining the process in which the SMO governs the applications and progresses them to a point ready for deployment. The SMO Network Function Orchestrator (NFO) will be simulated to draw out requirements for an SMO implementation. The tools and microservices developed as part of this Use Case should be able to be integrated to any SMO environment. Stereotypes are used to indicate which project and capability is being developed or used to provide the described role of the actor in the use case. The following actors are used to demonstrate the flow:

- <<RICAPP>> Package Manager: This function provides the ability to assemble a valid SMO application package from a build directory. The function can also be used by the SMO to validate a package that it receives during onboarding.
- <<INT OTF>> Application Developer: The OTF environment will be used to initiate the normally manual actions required by the application developer when creating an application package.
- <<INT OTF>> SMO Operator: The OTF environment will be used to initiate the normally manual actions requires by the SMO Operator when managing an application through its life cycle.
- <<INT OTF>> Network Function Orchestrator. The OTF environment will be used to sequence together the steps (even if initially manual) to move an application from its onboarded state to a ready-to-deploy state. Later some of these steps can be removed as more automation is developed.
- <<SMO>> MVC State Manager: This new Model/View/Controller Software Pattern provides state management according to a State Model provided as configuration and storage for associated metadata according to a Schema Definition. The function is deployed as three different instances for three different roles:
    - Model Catalog: This function manages the SMO states from Onboarding through Available for Deployment. It represents the design and inventory of onboarded applications as a service model and contains the catalogued components of the application package
    - Instance Config: This functions manages the SMO states from Available for Deployment through Deployed. It represents the plan to deploy to a location and allows pre-definition of the deployment configuration values and application configuration values in the SMO.
    - Virtual Inventory: The virtual inventory created by the deployment needs to be tracked. ONAP inventory does not yet align with the new aspects of the O-Cloud and therefore is not intended to be used for this release. Instead a simple version with enough intelligence to track the deployments will be implemented.

The key goal is that the O-RAN capabilities exist:

- Build an Application which can be packaged for deployment as an rAPP or xAPP.
- Onboard a package, validate the package and unpack it into the SMO Model Catalog
- Be able to define a configuration for an expected instance of an application deployment such that the application becomes "deployment ready"

The implementation of this use case will enable and demonstrate what application developers need to do in order to onboard their application to the SMO. It will also provide orchestration requirements for an SMO implementation in order to manage an onboarded application through its Life Cycle.

In addition, the use case documentation provides a mapping of the use case requirements to the EPICs defined for O-RAN Software Community (O-RAN SC) Cherry Release. The list below enumerates the EPICs to be added to the Cherry Release. Note that these Epics are of lower priority than RAN functionality required to support a UE to connect to the RAN:

- <<RICAPP>> Provide a tool to Build an App Package
- <<RICAPP>> Provide a tool to validate an APP Package
- <<SMO/OAM>> Create a Metadata driven Model/View/Controller for managing a package with configurable metadata
- <<SMO/OAM>> Onboard an App Package
- <<SMO/OAM>> Validate an App Package and move to Onboarding
- <<SMO/OAM>> Unpackage an App Package in the Onboarding State and set next State (ML Required, Available)
- <<SMO/OAM>> Create an potential configuration for an App Package in Available State and set next State (Deployable).

The following End-to-End flows are proposed to illustrate the activities in the major steps of the demonstration. Each step is executed in order as the preceding step is a prerequisite for the current step. The follows are intended to be illustrative and not necessarily exactly as implemented. Some of the activities is intended to drive needed knowledge and requirements to feed back into O-RAN to strengthen the specification being developed concurrently.

- **Step 1: Package Creation -** OTF will be used to simulate the Application Developer by creating an application package from a completed build from the repository.
- **Step 2: Package Validation -** OTF will be used to simulate the SMO Onboarding Process. This normally would be handled by the SMO Design Time Environment. This step represents the first step which is to validate the package.
- **Step 3: Package Cataloging -** OTF will be used to simulate the SMO Onboarding Process. This step takes the process from validated to available which in ONAP terms would be distributed to the Run Time Environment.

- **Step 4: Application Pre-Configuration -** OTF will be used to simulate SMO Operator which will define the configuration of an expected deployment of the application. This will serve for deployments as an rAPP or xAPP.

New Functionalities

SMO

Model Catalog

The Model Catalog keeps track of onboarded packages from the moment they are validated through the "Available" state. Some of the states represent Service Design the final "Available" state is when it is made available to the Run-time environment. This functionality amalgamates