## webwxgetmsgimg (7).jpegO-RAN-WG6.AppLCM-Deployment-R003-v02.00

*Technical Report*

**O-RAN Working Group 6 Application Life Cycle Management (LCM)**

**for Deployment Technical Recommendation**

Copyright © 2023 by the O-RAN ALLIANCE e.V.

The copying or incorporation into any other work of part or all of the material available in this specification in any form without the prior written permission of O-RAN ALLIANCE e.V. is prohibited, save that you may print or download extracts of the material of this specification for your personal use, or copy the material of this specification for the purpose of sending to individual third parties for their information provided that you acknowledge O-RAN ALLIANCE as the source of the material and that you inform the third party that these conditions apply to them and that they must comply with them.

O-RAN ALLIANCE e.V., Buschkauler Weg 27, 53347 Alfter, Germany

1

Contents

[List of figures 3](#_bookmark0)

[List of tables 3](#_bookmark1)

[Foreword 4](#_bookmark2)

[Modal verbs terminology 4](#_bookmark3)

[Executive summary <optional> 4](#_bookmark4)

[Introduction <optional> 4](#_bookmark5)

1. [Scope 5](#_bookmark6)
2. [References 5](#_bookmark8)
   1. [Informative references 5](#_bookmark9)
3. [Definition of terms, symbols and abbreviations 6](#_bookmark10)
   1. [Terms 6](#_bookmark11)
   2. [Symbols 6](#_bookmark12)
   3. [Abbreviations 7](#_bookmark13)
4. [Background 7](#_bookmark15)
   1. [Introduction 7](#_bookmark16)
   2. [App Package 8](#_bookmark18)
      1. [App Package Concepts 8](#_bookmark19)
      2. [App Package Security 9](#_bookmark20)
      3. [App Package DeploymentDescriptor 10](#_bookmark22)
      4. [CloudNativeDescriptorFile and ExecutableImage 11](#_bookmark24)
5. [Use Cases 11](#_bookmark25)
   1. [Deployment Flow For ASD And Cloud Native Artifacts Use Case 11](#_bookmark26)
      1. [Introduction 11](#_bookmark27)
      2. [UML Sequence Diagram 12](#_bookmark28)
      3. [Sequence Table 14](#_bookmark30)
      4. Recommended Requirements Error! Bookmark not defined.
6. [Information Requirements for K8s-Aware Solution 22](#_bookmark36)
   1. [DeploymentDescriptor 22](#_bookmark37)
      1. [Metadata 29](#_bookmark41)
      2. [Security 29](#_bookmark42)
      3. [Resource Requirements 29](#_bookmark43)
      4. [DeploymentItems 29](#_bookmark44)
      5. [Other 29](#_bookmark45)
   2. [DeploymentItem 29](#_bookmark46)
      1. [Deployment Sequencing 29](#_bookmark47)
      2. [Lifecycle Requirements 29](#_bookmark48)
7. [Information Requirements for ETSI-based Solution 30](#_bookmark49)
   1. [DeploymentDescriptor 30](#_bookmark50)
   2. [DeploymentItems 30](#_bookmark51)
8. [Formatting Recommendations 30](#_bookmark52)
   1. [CSAR 30](#_bookmark53)
   2. [SOL004 30](#_bookmark54)
   3. <here is a sample Annex sub-heading> Error! Bookmark not defined.

[History 33](#_bookmark56)

# List of figures

[Figure 4.1-1 Common Application Life Cycle Management for ASD 8](#_bookmark17)

[Figure 4.2.2-1 Package Security Approaches 10](#_bookmark21)

[Figure 4.2.3-1 Deployment Descriptor Model 10](#_bookmark23)

[Figure 5.1.2-1 Deployment Flow for ASD and Cloud Native Artifacts Use Case 14](#_bookmark29)

[Figure 5.2.2-1 Flow of NF Deployment instantiation with VNFD 19](#_bookmark33)

[Figure 6.1.1-1 DeploymentDescriptor with ASD representation 23](#_bookmark38)

# List of tables

[Table 1.1-1 Revision per Topic 5](#_bookmark7)

[Table 3.3-1 Abbreviations & Acronyms 7](#_bookmark14)

[Table 5.1.3-1 Deployment Flow for ASD and Cloud Native Artifacts Use Case 14](#_bookmark31)

[Table 5.1.4-1 Recommended Requirements for Deployment flow for ASD 17](#_bookmark32)

[Table 5.2.3-1 Flow of NF Deployment instantiation with VNFD 19](#_bookmark34)

[Table 5.2.4-1: Potential requirements related to VNFD for application deployment 22](#_bookmark35)

[Table 6.1.2-1 ASD Information Model 24](#_bookmark39)

[Note 1: Applies only for ExtCpd representing secondary network interfaces in a POD. Note 2: Several ExtCpd may](#_bookmark40) [refer to same additional network interface requirements 25](#_bookmark40)

[Table – Revision History 32](#_bookmark55)

[Table - History of the Document 33](#_bookmark57)

# Foreword

This Technical Report (TR) has been produced by O-RAN Alliance.

# Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the O-RAN Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in O-RAN deliverables except when used in direct citation.

# Executive summary

This is a Technical Recommendation from the WG6 Cloudification & Orchestration working group that documents the deployment flow use case, application package, package artifacts, and life cycle management.

The content of this TR itself is not normative. However, it will recommend possible requirements that are recommended to be incorporated into the Orchestration Use Case and Requirements document. Furthermore, the various sections of this document such as use case flows, application package, and package artifacts related to provisioning will be ported into their appropriate normative documents.

Because of the scope and nature of this document, there are a number of major sections in this document that will be tracked separately as they become mature.

# Introduction

Orchestration is the process of onboarding packages that have applications so that application artifacts can be made available during LCM. This is a Technical Recommendation from the WG6 Cloudification & Orchestration working group that documents the deployment flow use case, application package and life cycle management. The Deployment flow in this document shows detailed flows of what happens to accomplish deployment of applications.

The content of this TR itself it not normative.But, it will recommend possible requirements that are recommended to be incorporated into the Orchestration Use Case and Requirements document. The Deployment flow Use case is recommended to be integrated into the Orchestration Use Case document.

Because of the scope and nature of this document, there are a number of major sections in this document that will be tracked separately as they become mature. The document contains content for K8s aware and ETSI NFV aware solutions.

The recommendation section of this document provides recommendations of where the content should be incorporated into what normative specifications.

# Scope

The contents of the present document are subject to continuing work within O-RAN and may change following formal O-RAN approval. Should the O-RAN Alliance modify the contents of the present document, it will be re-released by O- RAN with an identifying change of version date and an increase in version number as follows:

version xx.yy.zz where:

xx: the first digit-group is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have xx=01). Always 2 digits with leading zero if needed.

yy: the second digit-group is incremented when editorial only changes have been incorporated in the document.

Always 2 digits with leading zero if needed.

zz: the third digit-group included only in working versions of the document indicating incremental changes during the editing process. External versions never include the third digit-group. Always 2 digits with leading zero if needed.

The present document specifies a Technical Recommendation from the WG6 Cloudification & Orchestration working group that documents the deployment flow use case, application package and life cycle management.

This document covers the orchestration & cloudification aspects for the Application Package model. The model is developed in the OAM Specification [1].

# Revision History per Topic

**Table 1.1-1 Revision per Topic**

|  |  |
| --- | --- |
| **TOPIC** | **REVISION STATUS** |
| Application Package Concepts & Security | Clause 4.2 / Reviewed / March 2023 Release Train |
| Deployment Flow for ASD and Cloud Native Artifacts Use Case | Clause 5.1 / Reviewed / March 2023 Release Train |
| Deployment Flow for VNFD Use Case | Clause 5.2 / Reviewed / March 2023 Release Train |
| Potential & Recommended Requirements | Clause 5.2 / Reviewed / March 2023 Release Train |
| Deployment Descriptor with ASD (Information Requirements for K8s-aware solution) | Clause 6.1 / Reviewed / March 2023 Release Train |
| Deployment Descriptor with VM (Information Requirements for ETSI VNF VM-aware solution) | Clause 6.x / |

# References

## Informative references

References are either specific (identified by date of publication and/or edition number or version number) or

non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, O-RAN cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document, but they assist the user with regard to a particular subject area.

1. O-RAN.WG10.OAM-Architecture: “O-RAN Operations and Maintenance Architecture”.
2. Kubernetes label key-value-nomenclature [https://kubernetes.io/docs/concepts/overview/working-with-](https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/) [objects/labels/](https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/)
3. Kubernetes node features [https://kubernetes-sigs.github.io/node-feature-discovery/v0.9/get-](https://kubernetes-sigs.github.io/node-feature-discovery/v0.9/get-started/features.html#table-of-contents) [started/features.html#table-of-contents](https://kubernetes-sigs.github.io/node-feature-discovery/v0.9/get-started/features.html#table-of-contents)
4. O-RAN WG10 App Package Specification
5. O-RAN Cloudification & Orchestration Use Case and Requirements
6. ETSI GS NFV-SOL 004 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; VNF Package and PNFD Archive specification".
7. ETSI GS NFV-IFA 011 (V4.3.1): "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; VNF Descriptor and Packaging Specification".
8. O-RAN.WG6.O2DMS-INTERFACE-ETSI-NFV-PROFILE (V03.00): "O-RAN Working Group 6: O2dms

Interface Specification: Profile based on ETSI NFV Protocol and Data Models".

1. O-RAN WG6 O2DMS INTERFACE K8S PROFILE (v02.00): “O-RAN Working Group 6 O2dms Interface

Specification: Kubernetes Native API Profile for Containerized NFs”.

1. O-RAN WG6 O2IMS INTERFACE (v03.00):” O-RAN Working Group 6 O2ims Interface Specification”.
2. ETSI GS NFV-SEC 021 (V2.6.1): “Network Functions Virtualisation (NFV) Release 2; Security; VNF Package Security Specification”
3. O-RAN.WG11.Security-Requirements-Specification.O (R003-v05.00.01): “O-RAN Working Group 11 Security Requirements Specifications”
4. O-RAN.WG6.ASD Specification “O-RAN Working Group 6 ASD Specification”

# Definition of terms, symbols, and abbreviations

## Terms

For the purposes of the present document, the following terms apply:

**DeploymentDescriptor:** a template validated by the Solution Provider describing the deployment options for the application that includes one or more **DeploymentItems**.

**DeploymentItem:** a template that describes the requirements for the deployment of *a component of* the application on the O-Cloud.

## Symbols

Void

## Abbreviations & Acronyms

For the purposes of the present document, the abbreviations apply:

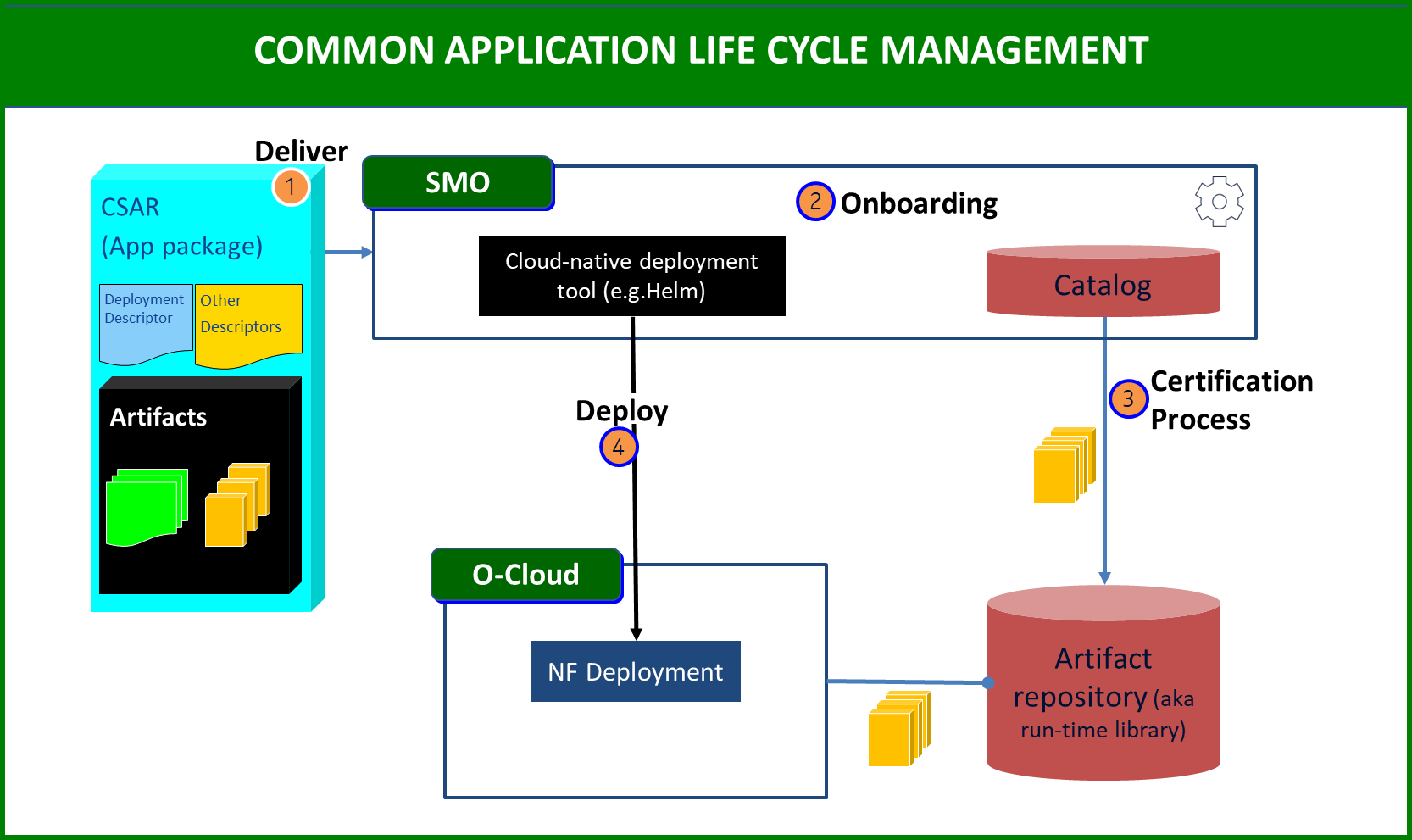
**Table 3.3-1 Abbreviations & Acronyms**

|  |  |
| --- | --- |
| **Abbreviation & Acronyms** | **Definition** |
| AI / ML | Artificial Intelligence & Machine Learning |
| ASD | Application Service Descriptor |
| CSAR | Cloud Service Archive |
| DMS | Deployment Management Services |
| ETSI | European Telecommunications Standards Institute |
| IMS | Infrastructure Management Services |
| K8s | Kubernetes |
| LCM | Life Cycle Management |
| OA&M | Operations, Administration & Management |
| O-RAN | Open Radio Access Network |
| SMO | Service Management & Orchestration |
| VNFD | VNF Descriptor |

# Background

## Introduction

This section describes the Common Application Life Cycle Management of applications that are developed by a Solution Provider and delivered to a Service Provider or Network Operator for deployment in an O-Cloud. Once the Solution Provider delivers the NF deployment, it is validated in a test environment prior to giving to operations to deploy. The description of the delivery of the NF deployments to the Service Provider as part of LCM is detailed in the OAM architecture document. The framework for Lifecycle Management is defined in the OAM Architecture document [1]. The O-RAN Cloudification & Orchestration aspects described here complements the concepts in the OAM Architecture specification. The following diagram in **Error! Reference source not found.** is an example that applies to the ASD solution for didactic purposes.



**Figure 4.1-1 Common Application Life Cycle Management for ASD**

The above diagram depicts some of the key steps in basic application package orchestration:

* Step 1, **DELIVER**: a solution provider (vendor) delivers an application package (app package) which may include a variety of artifacts. These artifacts may include application descriptors, model descriptors, informational artifacts, cloud native artifacts, metadata, and dictionaries.
* Step 2, **ONBOARDING**: Onboarding involves the ingestion, verification, and unpacking of the package and storage into a catalog. The SMO performs these onboarding activities and may usea cloud-native onboarding tools for that purpose. During this process, an AI/ML model can be trained.
  + Security Package Validation – During onboarding, the security validation of the package or package artifact(s) is performed. See the OAM Architecture document [1] clause 4.1.3, and clause 4.1.4.
* Step 3, **CERTIFICATION PROCESS**: After onboarding, the package undergoes a certification process from the catalog in the SMO. Afterwards, it is published to an artifact repository (aka the run-time library). Certification of the package may include the following: security validation of artifacts, metadata verification, artifact verification against indicated metadata locations. See the OAM Architecture document [1].
  + Note: The package cannot be certified until training is finished. If the AI/ML training cannot complete, the package is not certified.
  + Note: The package cannot be certified until security validation of the package is completed.
  + Note: For some operators, the package may go through part of the certification process before onboarding (Step 1 and Step 2).
* Step 4, **DEPLOY**: In deployment, The deployment tool is given a reference to the image repository entry from which the image can be pulled. Inventory is updated.

## App Package

### App Package Concepts

The package concepts are motivated from the ETSI GS NFV-SOL 004 [2] standards which describe the directory structure and required files for a package.

The Application Package contains:

Manifest file – The manifest file describes the location and types of files in the package.

Metadata – The metadata contains reference information related to the package itself. It is information about the information in the package, hence it is called metadata.

Other artifacts – The package can also contain other artifacts as well such as informational artifacts. Note: The OAM Architecture specification [1] does not cover details about the application package.

### App Package Security

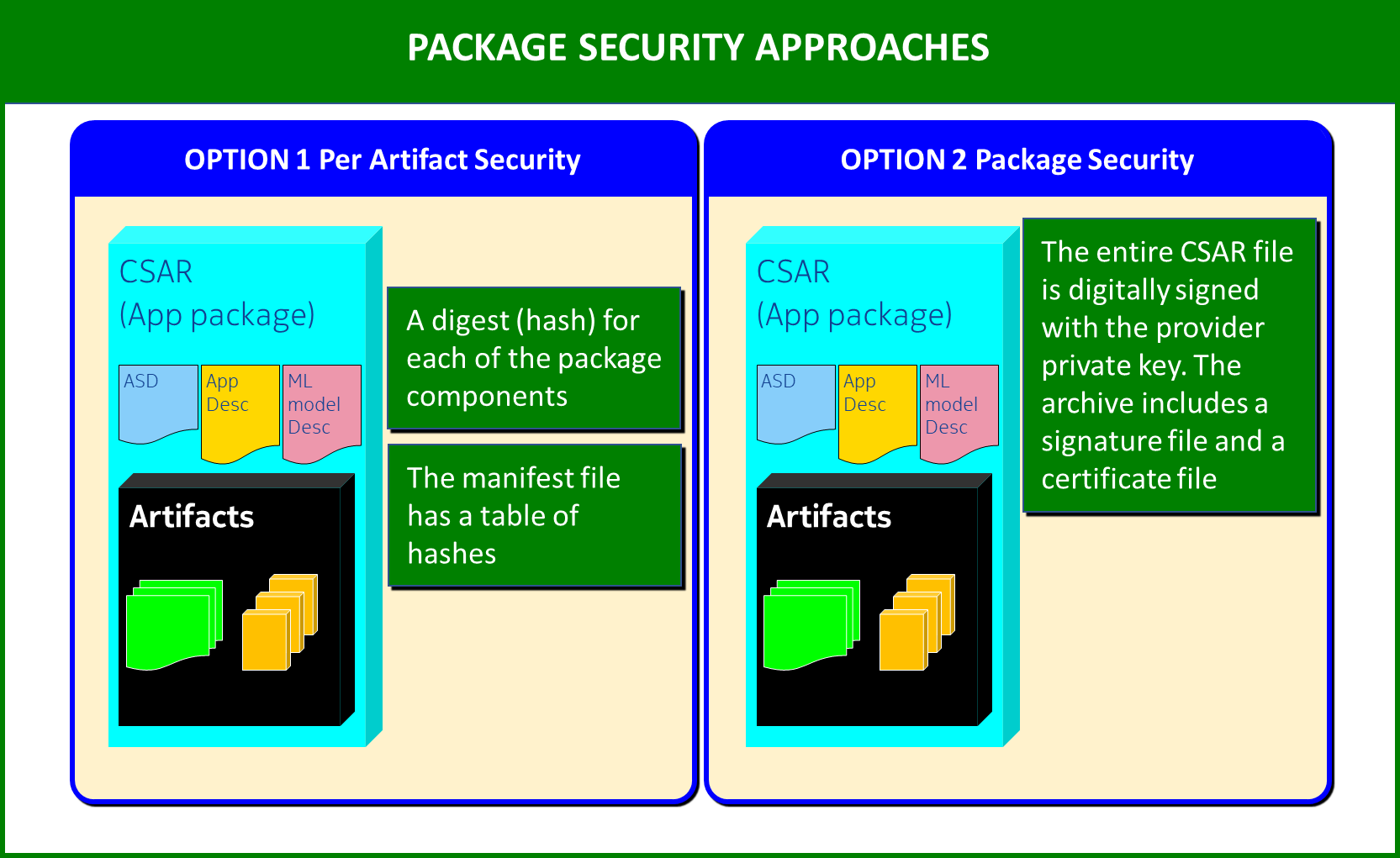
An important aspect of managing application packages of CNFs, VNFs, rApps, and xApps as it relates to O-RAN Cloudification & Orchestration are the security aspects.

The WG11 Security Requirements specification [12] covers security requirements and controls related to package security. These requirements follow ETSI GS NFV-SEC 021 [11], ETSI GS NFV-IFA011 [3] (stage 2) and ETSI GS NFV-SOL 004 [2] Clause 5.1 (stage 3) which describes two options for package security:

**Option 1 Per Artifact**: The VNF package shall contain a Digest (a.k.a. hash) for each of the components of the VNF package. The table of hashes is included in the *manifest file*, which is signed with the VNF provider private key. In addition, the VNF provider shall include a signing certificate that includes the VNF provider public key, following a pre-defined naming convention and located either at the root of the archive or in a predefined location (e.g., directory). The certificate may also be included in the signature container if the signature format allows that. For example, the CMS format allows to include the certificate in the same container as the signature.

All package artifacts shall be individually signed by the VNF provider as per ETSI GS NFV-SOL 004 [2] Clause 5.4 (stage 3) and the WG11 Security Requirements specification [12].

**Option 2 Per CSAR Package**: The complete CSAR file shall be digitally signed with the VNF provider private key. The VNF provider delivers one zip file consisting of the CSAR file, a signature file and a certificate file that includes the VNF provider public key. Only one signature of the CSAR file shall be present. The certificate may also be included in the signature container if the signature format allows that. The manifest shall be signed in both option 1 and option 2. Only one signature of the manifest shall be present.

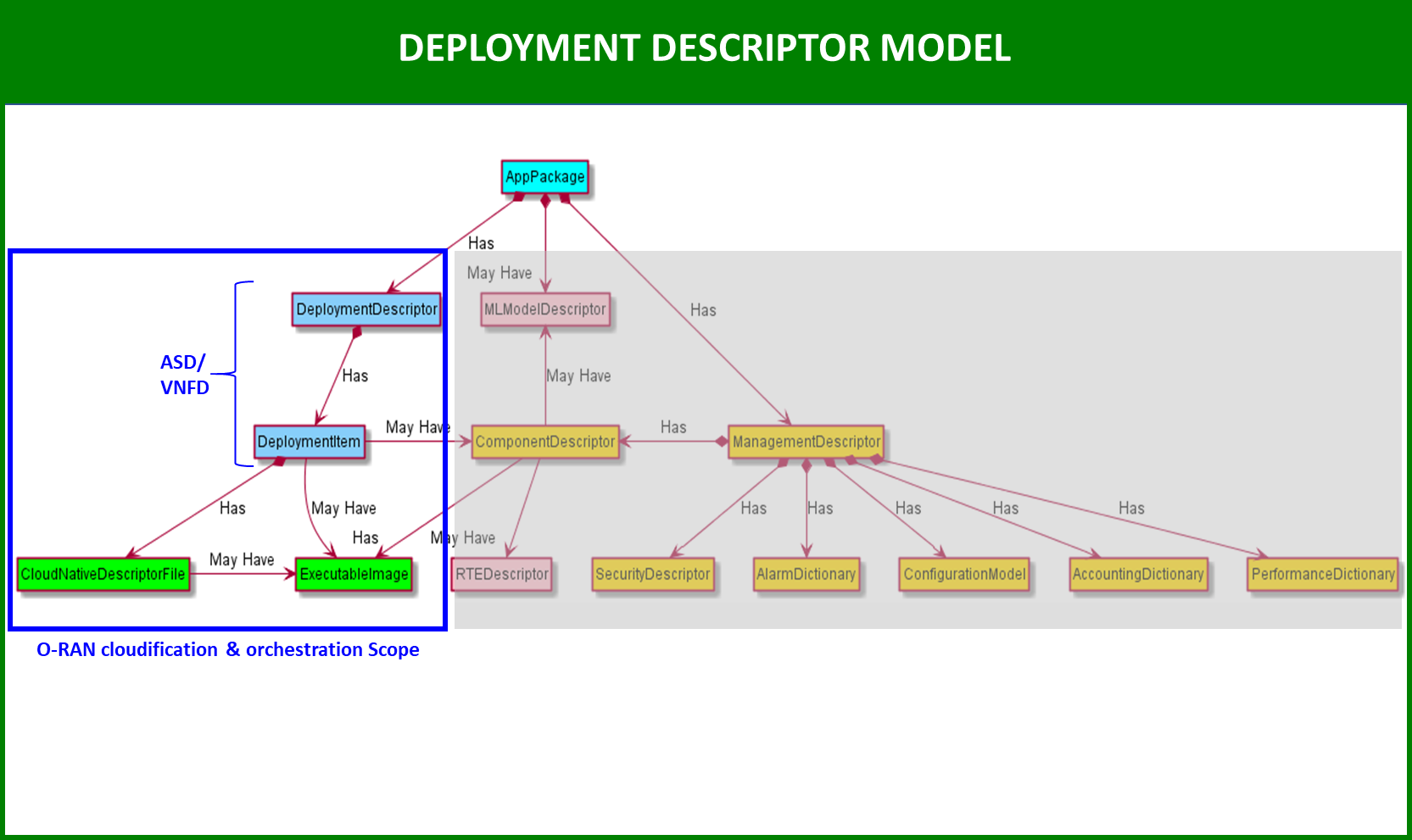


**Figure 4.2.2-1 Package Security Approaches**

### App Package DeploymentDescriptor

This section describes the Application Package (App Package) as it relates to O-RAN Cloudification & Orchestration. The OAM Architecture document (see reference [1]) details the Application Package model which details the high-level composition of the Application Package (AppPackage). This is shown in figure 2. The part of that model that is pertinent to O-RAN cloudification & orchestration that is within scope of the [Work Item Technical Report] is demarcated with a blue box in figure 2.

This section also describes the Application Service Descriptor (ASD) and the VNF Descriptor (VNFD) in the application package. These two things serve the same function in their respective profiles, and these terms are used interchangeably through the rest of the document. The ASD is a simplified deployment descriptor compared to the VNFD for CNFs, xApps and rApps to leverage enhancements in orchestrators.



**Figure 4.2.3-1 Deployment Descriptor Model**

In figure 2, the part of the model highlighted with the blue box, shows that the DeploymentDescriptor is contained within an AppPackage. The DeploymentDescriptor has a composition relationship to Deployment Item(s). The Deployment Item(s) has a composition (containment) relationship to one cloud native descriptor files. The DeploymentItem has an association relationship with the ExecutableImage(s) which has a cardinality of zero to many.

In figure 2, the part of the model highlighted with the blue box, ties back to the application package model management descriptor which is described in more detail in this TR. Note: future versions of this document will further describe this part of the model (in the blue box).

This TR suggests that the DeploymentDescriptor and DeploymentItem(s) be contained within one artifact in the package as an Application Service Descriptor (ASD) or VNF Descriptor (VNFD). The DeploymentItem is part of the ASD. However, it is possible that the DeploymentDescriptor and DeploymentItem(s) could be in separate artifact.

There will be metadata pertaining to the cloud native descriptor file. The metadata will include the checksum/hash, and type of file (helm, K8s manifest, SOL doc). From a workflow perspective, it uses the metadata to determine the appropriate action to take. The ASD allows the SMO to determine the resource requirements based on variables of the cloud native descriptor file. The deployment requirements can be derived for the executable image from the deployment item and the cloud native descriptor file.

The OAM architecture document and Information Model specification in reference [1] in the Lifecycle Management section describes the full model and the components show in more detail.

* + 1. CloudNativeDescriptorFile and ExecutableImage

# Use Cases

## Deployment Flow for ASD And Cloud Native Artifacts Use Case

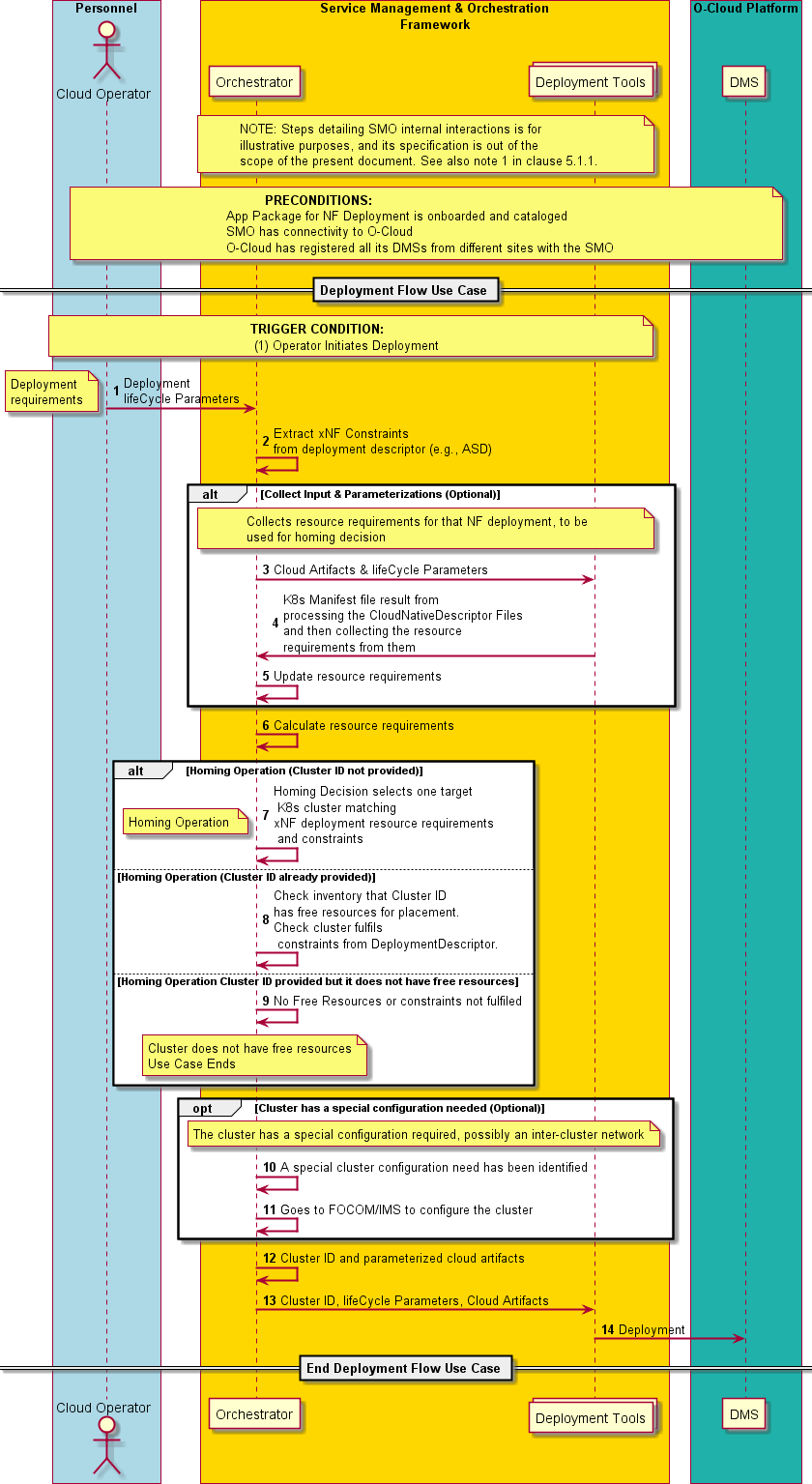
### Introduction

This describes a use case and flow to deploy an application (such as a NF deployment) into the O-Cloud. The deployment uses a deployment tool that is appropriate for the artifactType in the DeploymentItem.

This use case is for an application deployment on a K8s Cluster with an ASD.

NOTE 2: The use case flow illustrates interactions among entities within the SMO and between the SMO and the O- Cloud.

### UML Sequence Diagram



**Figure 5.1.2-1 Deployment Flow for ASD and Cloud Native Artifacts Use Case**

### Sequence Table

**Table 5.1.3-1 Deployment Flow for ASD and Cloud Native Artifacts Use Case**

|  |  |  |
| --- | --- | --- |
| USE CASE STAGE | EVOLUTION / SPECIFICATION | <<Uses>> Related Use |
| Use Case Title | Deployment Flow using ASD and Cloud Native Artifacts Use Case |  |
| Goal | The SMO uses the data available in the respective Deployment Descriptor to extract the specific xNF constraints (e.g., additional connectivity requirements, specific infrastructure capabilities needed, the specific input values provided in the lifecycle Parameters, etc.).  The SMO makes the homing decision (choses a Kubernetes cluster) that matches the xNFs deployment requirements and parameterizes the cloud artifacts according to the received input (such as K8s resource descriptions, ASD, and other relevant data available for respective deployment). |  |
| Actors and Roles | Cloud Operator – An operator of the Cloud.  SMO: Orchestrator – One of the components in the SMO that coordinates orchestration.  SMO: Deployment Tool – One of the components in the SMO that manages the cloud native deployment.  O-Cloud DMS – The O-Cloud Deployment Management Services. |  |
| Assumptions | Deployment tool – It is assumed that the SMO has the proper deployment tool that corresponds to the artifactType in the DeploymentItem. For example, helm.  Deployment descriptors – This consists of a DeploymentDescriptor and one to many CloudNativeDescriptorFiles. |  |
| Preconditions | Onboarded & Cataloged – The App Package for NF Deployment is onboarded and cataloged.  Connectivity – SMO has connectivity to O-Cloud.  O-Cloud Operational – O-Cloud has registered its DMS with the SMO. |  |
| Begins when | This use case begins when the Cloud Operator initiates the creation of a NF Deployment. |  |
| Step 1 (M) | The Cloud operator initiates a deployment operation which consists of a set of a DeploymentItem, which supports complex applications that require multiple artifacts (like Helm charts) to be installed in a particular order.  Each deploymentOrder is from a list of DeploymentItem constructs. Each DeploymentItem has lifeCycleParameters that show how the input parameters for that deployment are mapped into their respective CloudNativeDescriptorFiles. |  |
| Step 2 (M) | EXTRACTS xNF CONSTRAINTS |  |

|  |  |  |
| --- | --- | --- |
|  | The orchestrator extracts xNF constraints from the NF deployment descriptor, in this case the ASD. |  |
| Step 3 (ALT, OPT) | COLLECT INPUT & PARAMETERIZATIONS(Optional)  The purpose of the collect input & parameterizations step is to use input received from the Cloud operator for this deployment and map it into the CloudNativeDescriptorFiles according to mapping provided by the lifeCycleParameters and determine a K8s Resource Descriptor.  The deployment tool receives and processes the cloud artifacts and lifeCycleParameters native to the cloud artifact. The artifactType defines the type of artifacts within the DeploymentItem and what will be native to those artifacts.  The lifeCycleParameters are the list of parameters that can be overridden at deployment time (e.g., the list of parameters in the values.yaml). |  |
| Step 4 (ALT, OPT) | OUTPUT K8s MANIFEST FILES  The deployment tool returns the K8s manifest files and collects the resources requirements. |  |
| Step 5 (ALT, OPT) | PROCESS K8s MANIFEST FILE  K8s Manifest is processed to update the xNF constraints. |  |
| Step 6 (M) | CALCULATE RESOURCE REQUIREMENTS  The orchestrator calculates the resource requirements. |  |
| Step 7 (ALT) | HOMING DECISION (CLUSTER ID IS NOT PROVIDED)  The orchestrator performs the homing step.  The homing decision uses the aggregated resource requirements, constraints fetched from the DeploymentDescriptor (e.g., ASD) and data available locally for the SMO to process the decision. Note: data may be anything that the SMO knows out of the scope of the ASD and K8s resource descriptor.  The orchestrator makes a homing decision that matches the NF deployment requirements. It identifies a K8s cluster type or instance to deploy upon. The Homing operation selects the DMS that will be used. If an operator has multiple clouds (each one with multiple DMSes), the orchestrator will identify which cloud and DMS that are used. |  |
| Step 8 (ALT) | HOMING DECISION (CLUSTER ID IS AVAILABLE)  The orchestrator checks that the inventory has the referred cluster with free resources for placement based on the aggregated resource requirements. Also, it checks that the cluster fulfils the constraints from the DeploymentDescriptor. |  |
| Step 9 (ALT) | POSSIBLE ALTERNATIVE FLOW: HOMING DECISION (CLUSTER ID PROVIDED BUT NO FREE RESOURCES OR CONSTRAINTS NOT FULFILED)  The cluster designated by the Cluster ID doesn’t have the free resources needed for the application deployment.  A failure notification is sent.  The orchestrator continues to seek a suitable cluster, going back to step 5. |  |

|  |  |  |
| --- | --- | --- |
|  | Note: The repetition, defaults for retries of the homing decision are implementation specific. |  |
| Step 10 (ALT) | ALT FLOW: ADDITIONAL CLUSTER PROVISIONING IS NEEDED  This alt flow is used if additional cluster provisioning is required.  For example, it might be the case that the deployment requires an inter-cluster network. The deployment may span more than one cluster. In this case, the orchestrator may have to go back to FOCOM/IMS because the cluster networks would need to be provisioned in the IMS. |  |
| Step 11 (ALT) | ALT FLOW: CONFIGURE THE CLUSTER  The orchestrator will instruct FOCOM/IMS to configure the cluster. |  |
| Step 12 (M) | DEPLOYMENT REQUEST  The deployment is performed with the cluster ID, and the required cloud artifacts (e.g., K8s manifest). |  |
| **Step 13 (M)** | DEPLOYMENT REQUEST  The orchestrator (SMO) requests for the deployment tools (SMO) for the deployment operation with the Cluster ID, lifeCycle Parameters and Cloud Artifacts. |  |
| Step 14 (M) | DEPLOYMENT  The deployment tool requests for the deployment operation towards O2 DMS over the O2dms Interface.  The Inventory is updated within the SMO for the deployment. |  |
| Ends When | The application has been deployed to the appropriate cluster. This use case ends when the containerized workload is up and running in a suitable Kubernetes cluster. |  |
| Exceptions | Exceptions – A failed deployment occurs; thus, the application could not be deployed. A cluster that matches the requirements was not available for this deployment. |  |
| Post-conditions | Success – In a successful post-condition, the containerized workload is up and running in the chosen Kubernetes cluster with a given cluster ID. O1 registration occurs with the PNF/xNF registering with O1.  Failure – The post-condition upon a failure the deployment did not occur. It is expected that the O-Cloud & O-Cloud infrastructure resources are in the same state.  For further details see the instantiate use case in K8s [12] and NFV profiles [11]. and instantiate Use Case in Orchestration Use Case. |  |
| Traceability | (See Requirements Below) [REQ-ORC-xx+1], [REQ-ORC-xx+2], [REQ-ORC- xx+3], [REQ-ORC-xx+4], [REQ-ORC-xx+5], [REQ-ORC-xx+6] |  |

### Recommended Requirements for Deployment Flow for ASD and Cloud Native Artifacts Use Cases

ORCHESTRATION REQUIREMENTS RELATED TO O2 (Recommended section for the Orchestration & Cloudification Use Case document)

**POTENTIAL NEW REQUIREMENTS**:

**Table 5.1.4-1 Recommended Requirements for Deployment flow for ASD**

|  |  |  |
| --- | --- | --- |
| [REQ-ORC-xx+1] | The deployment descriptors provided for the NF deployment shall contain necessary information to enable the SMO, together with other input data received from the cloud operator to perform a homing decision. | Deployment Flow for ASD And Cloud Native Artifacts Use Case |
| [REQ-ORC-xx+2] | The DeploymentDescriptor (e.g., ASD) shall contain deployment parameters to enable the SMO to perform application deployment.  Note: for further information on the DeploymentDescriptor and Information Model (IM) see reference [13]. | Deployment Flow for ASD And Cloud Native Artifacts Use Case |
| [REQ-ORC-xx+3] | The DeploymentDescriptor (e.g., ASD) shall contain “lifecycleParameters” to support the SMO to map cloud operator customizable deployment parameters for the application deployment.  Note: For further information about the “lifecycleParameters” see reference [13]. | Deployment Flow for ASD And Cloud Native Artifacts Use Case |
| [REQ-ORC-xx+4] | The DeploymentDescriptor shall support providing default values in the chart enabling the SMO (Orchestrator & Deployment Tools) to complement them with instance specific parameter values in order to construct a values file. | Deployment Flow for ASD And Cloud Native Artifacts Use Case |
| [REQ-ORC-xx+5] | The DeploymentDescriptor (e.g., ASD) shall contain necessary deployment information enabling the SMO (Orchestrator & Deployment Tools) to render corresponding concrete K8s manifest descriptions.  Note: This aspect is under investigation and will be reflected in a future version of this document. | Deployment Flow for ASD And Cloud Native Artifacts Use Case |
| [REQ-ORC-xx+6] | The DeploymentDescriptor (e.g., ASD) shall have workload descriptors that can be parsed by the SMO.  Note: This aspect is under investigation and will be reflected in a future version of this document | Deployment Flow for ASD And Cloud Native Artifacts Use Case |
| [REQ-ORC-xx+7] | The DeploymentDescriptor (e.g., ASD) shall support describing resource requirements to enble the SMO (Orchestrator & Deployment Tools) to extract cloud resource requirements including but not limited to quantitative requirements. | Deployment Flow for ASD And Cloud Native Artifacts Use Case |

## Deployment flow for VNFD use case

### Introduction

Clause 5.2 describes a use case and flow to deploy an application, such as an NF Deployment into the O-Cloud, using the ETSI NFV defined Virtualised Network Function (VNF) Descriptor (VNFD) specified in ETSI GS NFV-IFA 011 [3].

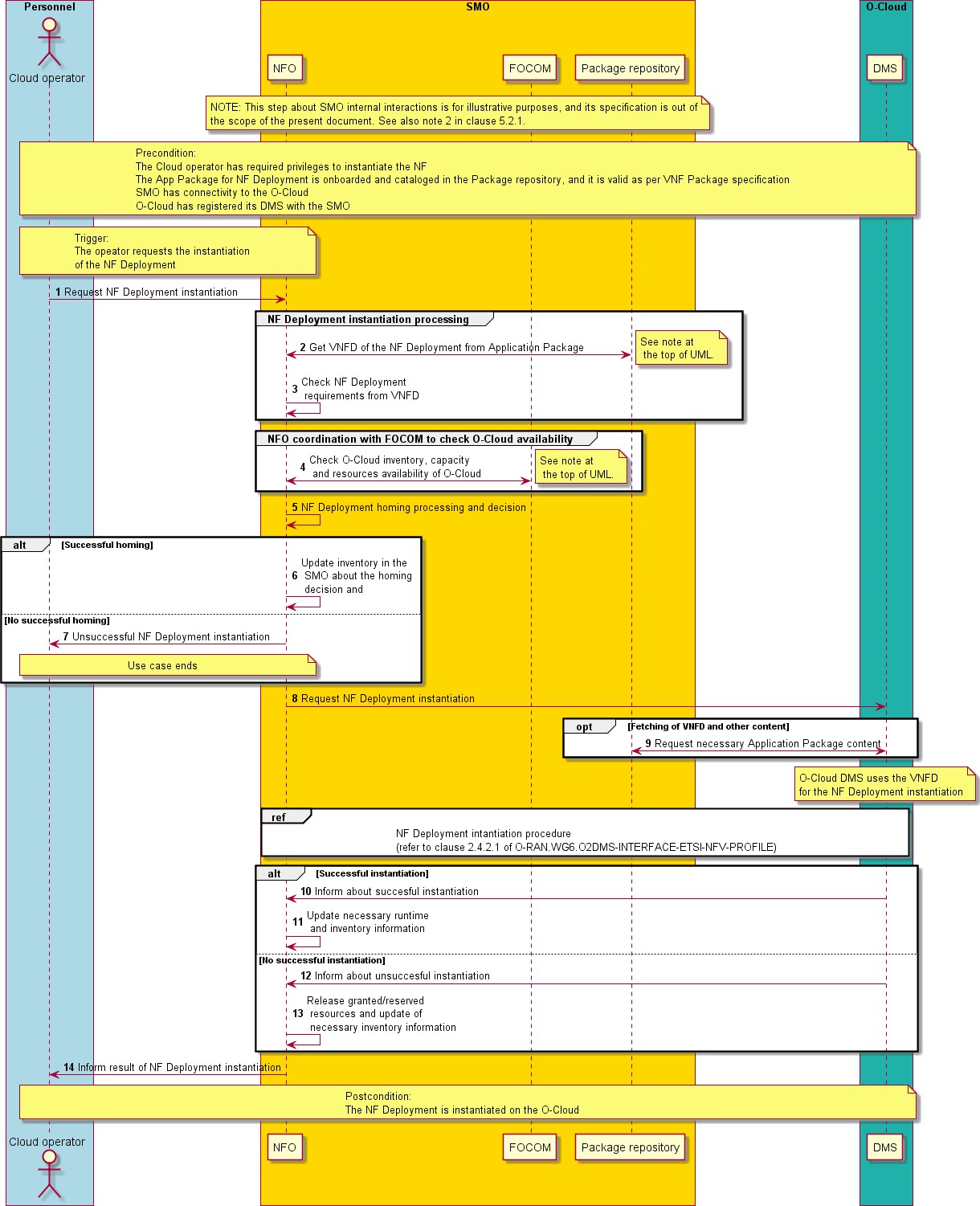
NOTE 1: It is assumed in the use case flow that the SMO interacts with an O-Cloud DMS based on the O2-D demarcation #1 documented in the O-RAN.WG6.O2DMS-INTERFACE-ETSI-NFV-PROFILE [4]. The use case flow could be revisited, or new ones be derived, in future versions of the present document based on further updates performed to the referenced document.

NOTE 2: The use case flow illustrates interactions among entities within the SMO and between the SMO and the O-Cloud. Clause 2.3.2 of O-RAN.WG6.O2DMS-INTERFACE-ETSI-NFV-PROFILE [4] provides the

mapping and correspondence of management functions and functional blocks between the O-RAN and ETSI NFV architectural frameworks.

### UML sequence diagram

Figure 5.2.2-1 illustrates the use case flow for the instantiation of an NF Deployment based on VNFD.



**Figure 5.2.2-1 Flow of NF Deployment instantiation with VNFD**

### Sequence table

**Table 5.2.3-1 Flow of NF Deployment instantiation with VNFD**

|  |  |  |
| --- | --- | --- |
| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |

|  |  |  |
| --- | --- | --- |
| Goal | Application deployment (NF Deployment) into the O-Cloud using VNFD. |  |
| Actors and Roles | Cloud operator: an operator of the O-RAN NF and O-Cloud. SMO logical functions: NFO, FOCOM  Package repository within SMO (see note 1)  O-Cloud DMS: the O-Cloud Deployment Management Services.  NOTE 1: It is out of scope of the present document to specify whether the SMO supports package repository functions and/or services. |  |
| Assumptions | O-Cloud DMS offers NF Deployment management functionality according to the O2-D demarcation #1 documented in O- RAN.WG6.O2DMS-INTERFACE-ETSI-NFV-PROFILE [4]. |  |
| Pre-conditions | 1. The Cloud operator has required privileges to instantiate the NF 2. The Application Package for NF Deployment, which includes the VNFD and other artifacts, is onboarded and catalogued, and it is valid as per VNF Package and VNFD specification (reference is provided to ETSI GS NFV-IFA 011 [3] as "VNF Package and VNFD specification"). 3. SMO has connectivity to O-Cloud 4. O-Cloud has registered its DMS with the SMO. |  |
| Begins when | Cloud operator requests the instantiation of an NF Deployment. |  |
| Step 1 (M) | Cloud operator requests to instantiate an NF Deployment based on a VNFD, which is identified in the request. The VNFD contains one or more DeploymentItems describing the requirements for the NF  Deployment. |  |
| Step 2 (M) | NFO queries the Package repository to get the VNFD from the Application Package corresponding to the NF Deployment requested to be instantiated. |  |
| Step 3 (M) | NFO checks the NF Deployment metadata, resource requirements (types and amount), deployment lifecycle parameters and deployment items information from the VNFD. |  |
| Step 4 (M) | NFO coordinates with FOCOM to check O-Cloud inventory, capacity and resources availability of the O-Cloud. The request could include a list of identified O-Cloud Sites, if a pre-selection or requirements about specific NF Deployment location are available to the NFO (e.g., provided by the Cloud operator), so that O-Cloud DMS instance that  matches the application requirements is selected |  |
| Step 5 (M) | NFO performs an NF Deployment homing, which determines the O- Cloud (see note 2) and the respective O-Cloud DMS responsible for the NF Deployment instantiation.  NOTE 2: It is assumed that the NF Deployment instantiation is confided to a single O-Cloud. |  |
| Step 6 (M) | In case of successful homing, the NFO updates inventory information in the SMO based on the homing decision. The NFVO also prepares  necessary input data for the NF Deployment instantiation request. |  |
| Step 7 (M) | In case of unsuccessful homing, the NFO informs the Cloud operator of unsuccessful NF Deployment instantiation. |  |
| Step 8 (M) | NFO requests the O-Cloud DMS to instantiate the NF Deployment. The instantiation request(s) can provide necessary input lifecycle and configuration data for the instantiation. Information about granted resources, target O-Cloud and placement constraints for the NF  Deployment can also be provided. |  |
| Step 9 (M) | In case the O-Cloud DMS does not have the VNFD and/or other information needed for the instantiation of the NF Deployment (e.g., certain artifacts), the DMS fetches from the Package repository the  necessary content of the Application Package. |  |
|  | O-Cloud DMS proceeds with NF Deployment instantiation as described in clause 2.4.2.1 of O-RAN.WG6.O2DMS-INTERFACE-ETSI-NFV-  PROFILE. The O-Cloud DMS uses the information provided at runtime on the request, information from the VNFD and contents from the Application Package to perform the allocation of necessary O-Cloud resources for the NF Deployment. | Clause 2.4.2.1 of O-  RAN.WG6.O2DMS- INTERFACE-ETSI- NFV-PROFILE |
| Step 10 (M) | In case of successful instantiation of the NF Deployment, the O-Cloud DMS updates the runtime information of the NF Deployment and  informs to the NFO of the successful instantiation. |  |
| Step 11 (M) | NFO updates the inventory information according to the successful NF Deployment instantiation. |  |

|  |  |  |
| --- | --- | --- |
| Step 12 (M) | In case of unsuccessful instantiation of the NF Deployment, the O- Cloud DMS informs to the NFO about the failures/errors. The O-Cloud  DMS proceed with the rollback of the instantiation. |  |
| Step 13 (M) | NFO proceeds with necessary rollback handling, freeing any of the granted/reserved resources for the NF Deployment and updating the  inventory information considering the unsuccessful instantiation. |  |
| Step 14 (M) | NFO informs the Cloud operator about the result (either successful or not successful) of the instantiation of the NF Deployment. |  |
| Ends when | The Cloud operator is informed about the NF Deployment instantiation result. |  |
| Exceptions | * NF Deployment homing fails: O-Cloud does not have enough or required resources for the NF Deployment. * NF Deployment instantiation fails: O-Cloud DMS notifies SMO and SMO informs Cloud operator, and granted/reserved resources are freed, and inventory data is updated accordingly. |  |
| Post-conditions | The NF Deployment is instantiated on the O-Cloud. |  |
| Traceability | REQ-SMO-VNFD.0a, REQ-SMO-VNFD.0b, REQ-SMO-VNFD.0c, REQ- SMO-VNFD.0d, REQ-SMO-VNFD.0e, REQ-SMO-VNFD.0f |  |

### 5.3.2 Potential New Requirements for Deployment flow for VNFD Use Case

Table 5.2.4-1 lists potential requirements related to VNFD support application deployment.

**Table 5.2.4-1: Potential requirements related to VNFD for application deployment**

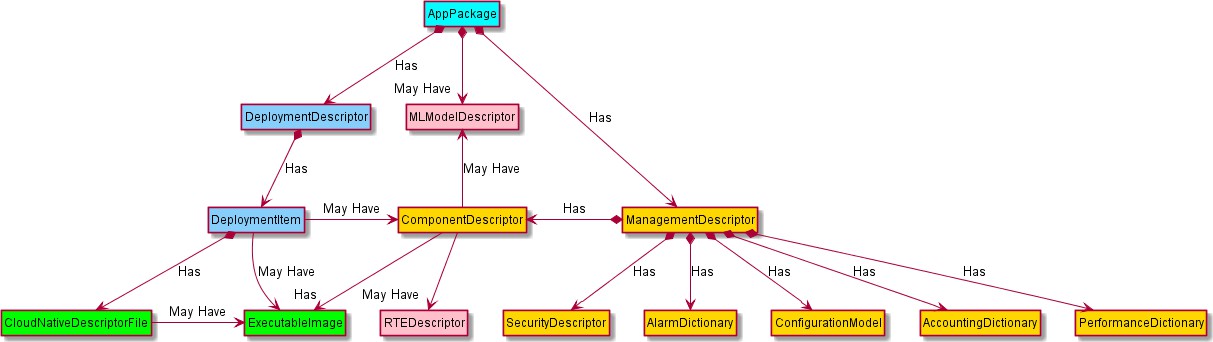
|  |  |  |
| --- | --- | --- |
| **Identifier** | **Requirement** | **Additional notes** |
| REQ-SMO- VNFD.0a | The deployment descriptors provided for the NF Deployment shall contain necessary information to enable the SMO, together with other  input data received from the Cloud operator, perform a homing decision for the application deployment. | Deployment flow for VNFD use case |
| REQ-SMO- VNFD.0b | The DeploymentDescriptor (e.g., VNFD) shall contain NF Deployment resource requirements (type and amount) and NF Deployment units’ affinity/anti-affinity constraints to enable the SMO (NFO) to parse,  process and use the deployment information for the application deployment. | Deployment flow for VNFD use case |
| REQ-SMO- VNFD.0c | The DeploymentDescriptor (e.g., VNFD) shall support defining lifecycle parameters for the corresponding NF Deployment to enble the SMO  (NFO) to parse, process and determine the lifecycle operation to perform. | Deployment flow for VNFD use case |
| REQ-SMO- VNFD.0d | The DeploymentDescriptor (e.g., VNFD) shall support defining necessary input lifecycle parameters to enable the SMO (NFO) to  assign corresponding values from input data provided to the SMO when requesting the O-Cloud DMS the NF Deployment lifecycle operation. | Deployment flow for VNFD use case |
| REQ-SMO- VNFD.0e | The DeploymentDescriptor (e.g., VNFD) shall support defining necessary input lifecycle parameters to enable the SMO (NFO) generate, based on orchestration runtime data, corresponding values when requesting the O-Cloud DMS the NF Deployment lifecycle  operation. | Deployment flow for VNFD use case |
| REQ-SMO-VNFD.0f | The DeploymentDescriptor (e.g., VNFD) shall conform to a VNF Package and VNFD specification to enable the SMO (NFO) to check if an onboarded application package is valid. | Deployment flow for VNFD use case.  Reference is provided to ETSI GS NFV-IFA 011  [3] as "VNF  Package and VNFD specification". |

# Information Requirements for K8s-Aware Solution

## DeploymentDescriptor with Application Service Descriptor

Application Service Descriptor (ASD) describes application metadata for the orchestrator to be used as input for deployment of an application workload that is described by the ASD.

### Conventions



The attributes of the ASD information elements are described in the tables provided in section 6.1. Each table has 5 columns, with the following significance:

* + - * The "Attribute" column provides the attribute name.
      * The "Qualifier" column indicates whether the support of the attribute is mandatory, optional or conditional from the SMO perspective.
      * The "Cardinality" column contains the minimum and maximum cardinality of this information element (e.g., 1, 0 … N, 1 … N). A cardinality range starting with 0 indicates that the attribute need not always be included and 1 indicates that the attribute needs to be always included.
      * The "Type" column provides information on the type of the attribute values. It can be the name of an Information Element. If a cell in the "Type" is marked as "Not specified", which indicates that the specification of the type is left to the data model.
      * The "Description" provides a brief explanatory description, additional constraints, and examples.

The ASD is a complementary descriptor to the cloud-native deployment artifacts hence it contains design-time information provided by the application vendor that is not described/available in the cloud native artifacts. Such complementary information in the ASD can include specific infrastructure capabilities requirements for the deployment of the application software workload, constraints on the Kernel type needed, additional networking requirements needed beyond primary connectivity described in the cloud native artifacts, any special CNI (Container Network Interfaces), etc.

ASD can support an application workload deployment on OS container infrastructures, and it is applicable to O-RAN NFs, xApps, and rApps.

The following figure illustrates the representation of O-RAN OAM Application Packaging Model [WG10. Common Package] and shows how the Application Service Descriptor is positioned within the Application Package Model.

has

has

May have

DockerImage (ExecutableImage)

Helm Chart (CloudNative DescriptorFile)

**ASD DeploymentItem (DeploymentItem)**

**ASD (DeploymentDescriptor)**

Deployment related artifacts

**Figure 6.1.1-1 DeploymentDescriptor with ASD representation**

### Application Service Descriptor

For the O-RAN NFs the Application Service Descriptor (ASD) is a main top level descriptor template which consists of the attributes described in Table 6.2-1 below.

**Table 6.1.2-1 ASD Information Model**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Qualifier** | **Cardinality** | **Type** | **Description** |
| asdId | M | 1 | Identifier | Identifier of this ASD information element. This attribute shall be globally unique. The format will be defined in the data model specification phase. |
| asdVersion | M | 1 | String | Identifies the version of the ASD |
| asdSchemaVersion | M | 1 | Version | Specifies the version of the ASD’s schema.  Note 1: |
| asdProvider | M | 1 | String | Provider of the application and of the ASD. |
| asdApplicationName | M | 1 | String | Name to identify the application. Invariant for the application lifetime. |
| asdApplicationVersion | M | 1 | Version | Specifies the version of the application.  Note 2. |
| asdApplicationInfoName | M | 0..1 | String | Human readable name for the application. Can change during the ASD lifetime. |
| asdInfoDescription | M | 0..1 | String | Human readable description of the application. Can change during the ASD lifetime. |
| asdExtCpd | M | 0..N | ExtCpd | Describes the externally exposed connection points of the application. |
| enhancedClusterCapabilities | M | 0..1 | EnhancedClus terCapabilities | A list of expected capabilities of the target deployment cluster to aid placement of the application on a suitable cluster. |
| deploymentItems | M | 1..N | DeploymentIt em | Cloud native deployment artifacts associated with the application. |
| asdInvariantId | M | 1 | Identifier | Identifier of this descriptor in a version independent manner. This attribute is invariant across versions of ASD. It is in UUID format as specified in RFC 4122 |
| Note 1: Version change when a change is made to any one of the ASD attributes.  Note 2: Version value change when software or ASD values changes | | | | |

### Deployment Item (DeploymentItems) Information Element

The CloudNativeDescriptorFile defined in [WG10. Common Package] is realized by the deploymentItems IE which has the DeploymentItems structure described in Table 6.3-1 below.

DeploymentItems IE describes the type of cloud native artifact, its order of deployment in case multiple cloud native artifacts are employed with the same ASD, and the list of parameters that can be parameterized in this cloud native deployment artifact at deployment time.

**Table 6.1.3-1 DeploymentItems IE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .**Attribute** | **Qualifier** | **Cardinality** | **Type** | **Description** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| deploymentItemId | M | 1 | Identifier | The identifier of this deployment item |
| artifactType | M | 1 | String | Specifies the artifact type. One of following values can be chosen: "helm\_chart", "helmfile", "crd", "terraform". |
| artifactId | M | 1 | String | Unique reference to a deployment item (cloud native deployment artifact). It can be a URI or file path. |
| deploymentOrder | M | 0..1 | Integer | Specifies the deployment order of the deployment item. A lower value specifies that the deployment item belongs to an earlier deployment stage, i.e., needs to be installed prior to a deployment item with higher deploymentOrder values. If not specified, the deployment of the deployment itemt can be done in arbitrary order and decided by the  orchestrator. |
| lifecycleParameters | M | 0..N | String | The list of parameters that can be overridden at deployment time (e.g., the list of parameters in the values.yaml which can be overridden at deployment time) |
| Note 1: An initial list of artifactType. Other type of artifact is for further study | | | | |

### External connection point (asdExtCpd) Information Element

ExtCpd describes application exposing endpoints to enable orchestrators to string together or optimally place linked applications.

**Table 6.1.4-1 ExtCpd Information Element (IE)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Qualifier** | **Cardinality** | **Content** | **Description** |
| id | M | 1 | String | The identifier of this ExtCpdData |
| description | M | 1 | String | Describes the service exposed. |
| virtualLinkRequirement | M | 1 | String | Refers in an abstract way to the network that the ExtCpd shall be exposed for (ex: OAM, EndUser, backhaul, LI, etc). The intent is to enable a network operator to identify which extCpd was designed to be connected to which VPN. . |
| networkInterfaceRealizati onRequirements | M | 0..1 | NetworkInte rfaceRealiza tionRequire ments | Details container implementation specific requirements on the NetworkAttachmentDefinition. See Note 1 & 2. |
| inputParamMappings | M | 0..1 | ExtCpd.Para mMappings | Information on what parameters is required to be provided to the deployment tools for the asdExtCpd instance. |
| resourceMapping | M | 0..1 | String | Resource name for the cloud native resource manifest, as specified in the cloud native descriptor declaring the network interface (e.g., in the helm.chart, for the: service, ingress or pod resource). Enables, together with knowledge on namespace, the orchestrator to lookup the runtime data related to the extCpd. |

**Note 1: Applies only for ExtCpd representing secondary network interfaces in a POD. Note 2: Several ExtCpd may refer to same additional network interface requirements.**

### NetworkInterfaceRealizationRequirements Information Element

NetworkInterfaceRealizationRequirements IE describes details related to secondary networks that attach the OS containers to the logical or physical networks.

**Table 6.1.5-1 NetworkInterfaceRealizationRequirements IE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Qualifier** | **Cardinality** | **Content** | **Description** |
| trunkMode | M | 0..1 | ”false” or “true” | If not present or set to”false”, means that this interface shall connect to single network. If set to ”true” then the network interface shall be a trunk interface (connects to multiple VLANS). |
| ipam | M | 0..1 | "infraProvided", "orchestrated", "userManaged" | The default value ("infraProvided") means that the CNI specifies how IPAM is done and assigns the IP address to the pod interface. |
| interfaceType | M | 0..1 | "kernel.netdev", "direct.userdriver", "direct.kerneldriver", "direct.bond", "userspace | This attribute is applicable for passthrough and memif interfaces. Value default value is ”kernel.netdev”. |
| interfaceOptions | M | 0..N | "virtio", "memif" | Alternative vNIC configurations the network interface is verified to work with. |
| interfaceRedundancy | M | 0..1 | "infraProvided", "left", "right",  "activeActiveBond", "activePassiveBond", "activePassivel3", | ”infraProvided” The application sees one vNIC, but the infrastructure provides redundant access to the network via both switch planes. For interface of type kernel.netdev the redundancy is provided by the vSwitch or Linux bonding. For interfaces of type direct, it requires that the physical NIC is connected to both switch planes (a.k.a. smartNIC). Thus, it imposes hardware requirements on the infrastructure. For interface of type userspace redundancy handled by a DPDK-OVS or VPP vSwitch ”left” and ”right” indicates  a vNIC connected non-redundantly to  the network via one specific (left or right) switchplane.  All other attributes indicate a mated vNIC pair in the Pod, one connecting to  the network via left switchplane and the other connecting to the network via the right switchplane, and with application using them together as a redundant network interface using a particular redundancy method that need to be accomodated in the node infrastructure.  "activeActiveBond": Requires a static multi- chassis LAG in active-active mode from the infrastructure. Typically corresponds to bonding mode-2 “balance-xor” "activePassiveBond": The interfaces are bonded in active-passive mode in the application with move of bond MAC address. No specific requirements on DC fabric. Typically corresponds to bonding mode-1 “active-backup”  "activePassivel3": Application moves its IP address by attaching it to one of the two MACs by use of GARP messages.  Editor's note: Change the link to reference section https://nfvwiki.etsi.org/index.php?title=Cni.. |
| nicOptions | M | 0..N | "i710",  "e810",  "mlx-cx5",  "mlx-cx6" | nics a direct user space driver the application is verified to work with. The content aligned with ETSI HPC registry Editor's note: Change the link to reference section https://nfvwiki.etsi.org/index.php?title=Cni. |

### ExtCpdParamMappings Information Element

ExtCpdParamMappings IE describes required information on what parameters to be provided to the deployment tools for the ExtCpd instance.

**Table 6.1.6-1 ExtCpdParamMappings IE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Qualifier** | **Cardinality** | **Content** | **Description** |
| loadbalancerIP | M | 0..1 | String | When present, this attribute specifies the name of the deployment artifact input parameter through which the orchestrator can configure the loadbalancerIP parameter of the K8s service or ingress controller that the ExtCpd represents.  *Note 2* |
| externalIPs | M | 0..N | String | When present, this attribute specifies the name of the deployment artifact input parameter through which the orchestrator can configure the extermalIPs parameter of the K8s service or ingress controller, or the pod network interface annotation, that the ExtCpd represents. The param name and provided IP address(es) value will be passed to the deployment tool when deploying the *DeploymentArtifacts.*  *Note 2* |
| nadNames | M | 0..N | String | These attributes specify, for an ExtCpd representing a secondary network interface, the name(s) of the deployment artifact input parameter(s) through which the orchestrator can provide the names of the network attachment definitions (NADs) the orchestrator has created as base for the network interface the ExtCpd represents.  *It is expected that the NADs themselves have been created prior to the deployment of the deployment artifacts.*  *Note 1,2,3* |
| nadNamespace | M | 0..1 | String | Specifies, for an ExtCpd representing a secondary network interface, the name of the deployment artifact input parameter through which the orchestrator can provide the namespace where the NADs are located.  Attribute may be omitted if the namespace is same as the application namespace.  *Note 2* |

Note 1: When the ExtCpd represent a networkRedundant/mated pair of sriov interfaces, there are references to 2 or 3 related NADs needed to be passed, while for other interface types only one NAD reference is needed to be passed.

Note 2: The format of the Content strings is specific for each different orchestration templating technology used (Helm, Teraform, etc.). Currently only a format for use with Helm charts is

suggested: "<helmchartname>:[<subchartname>.] 0 ... N[<parentparamname>.] 0 … N<paramname>”. Whether the optional parts of the format are present depends on how the parameter is declared in the helm chart. An example is: "chartName:subChart1.subChart2.subChart3.Parent1.Parent2.Parent3.parameter".

Note 3: A direct attached (passthrough) network interface, such as an sriov interface, attaches to a network via only one of the two switch planes in the infrastructure.

When using a direct attached network interface one therefore commonly in a pod uses a mated pair of sriov network attachments, where each interface attaches same network but via different switchplane.

The application uses the mated pair of network interfaces as a single logical “swith-path-redundant” network

interface – and this is represented by a single ExtCpd.

Also, there is a case where a third “bond” attachment interface is used in the pod, bonding the two direct interfaces so that the application does not need to handle the redundancy issues – application just uses the bond interface.

In this case all three attachments are together making up a logical “switch-path-redundant” network interface represented by a single ExtCpd. When three NADs are used in the ExtCpd the NAD implementing the bond attachment interface is provided through the parameter indicated in the third place in the nadNames attribute.

### EnhancedClusterCapabilities Information Element

EnhancedClusterCapabilities IE describes information which is used to aid placement of the application service on a suitable cluster.

**Table 6.1.7-1 EnhancedClusterCapabilities IE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Qualifier** | **Cardinality** | **Content** | **Description** |
| minKernelVersion | M | 1 | String | Describes the minimal required Kernel version, e.g., 4.15.0. Coded as displayed by linux command uname –r |
| requiredKernelModules | M | 0..N | String | Required kernel modules are coded as listed by linux lsmod command, e.g., ip6\_tables, cryptd, nf\_nat etc. |
| conflictingKernelModules | M | 0..N | String | Kernel modules, which must not be present in the target environment. The kernel modules are coded as listed by linux lsmod command, e.g., ip6\_tables, cryptd, nf\_nat etc. Example: Linux kernel SCTP module, which may conflict with use of proprietary user space SCTP stack provided by the application. |
| requiredCustomResources | M | 0..N | Structure (inlined) | List the required custom resource types in the target environment, identifying each by the "kind" and "apiVersion" field in the K8S resource manifests and in the application. The list shall include those custom resource types which are not delivered with the application.  Example: requiredCustomResources:  -{kind: "Redis", apiVersion: "[kubedb.com/v1alpha1](http://kubedb.com/v1alpha1)"} |
| >kind | M | 0..1 | String | Kind of the custom resource |
| >apiVersion | M | 0..1 | String | apiVersion of the custom resource |
| clusterLabels | M | 0..N | String | This attribute indicates the required O-Cloud Node Cluster capabilities.  These can indicate special infrastructure capabilities (e.g., NW acceleration, GPU compute, etc.). The intent of these O-Cloud Node Cluster capabilities is to serve as a set of values that can support the SMO in application placement decisions. See note 1.  clusterLabels follow the Kubernetes label key-value- nomenclature [5] . This attribute should use the standardized values for the O-RAN O-Cloud Node Cluster capabilities [reference tbd], the existing values for node  features [6], and any new capabilities identified in O-RAN specifications and allows also vendor-specific extensions. |
| requiredPlugin | M | 0..N | Structure (inlined) | A list of the names and versions of the required K8s plugin (e.g., multus v3.8) |
| >requiredPluginName | M | 0..1 | String | The names of the required K8s plugin (e.g., multus) |
| >requiredPluginVersion | M | 0..1 | String | The version of the required plugin (e.g., 3.8) |
| Note 1: The SMO uses the set of standardized capabilities to identify O-Cloud Node cluster capabilities required in the DeploymentDescriptors and to match them to the capabilities available in the O-Cloud. | | | | |

### Metadata

NOTE: Content is expected in future versions of the present document.

### Security

NOTE: Content is expected in future versions of the present document.

### Resource Requirements

NOTE: Content is expected in future versions of the present document.

### DeploymentItems

NOTE: Content is expected in future versions of the present document.

### Other

NOTE: Content is expected in future versions of the present document.

## DeploymentItem

### Deployment Sequencing

NOTE: Content is expected in future versions of the present document.

### Lifecycle Requirements

NOTE: Content is expected in future versions of the present document.

# Information Requirements for ETSI-based Solution

## DeploymentDescriptor

NOTE: Content is expected in future versions of the present document.

## DeploymentItems

NOTE: Content is expected in future versions of the present document.

# Formatting Recommendations

## CSAR

NOTE: Content is expected in future versions of the present document.

## SOL004

NOTE: Content is expected in future versions of the present document.

# Recommendations

## Introduction

This TR is structured in a modular fashion organized by major topics and the following sections provide relevant recommendations by topic.

## Recommendation for Deploy flow use case & requirements

The deployment flow use case for ASD is recommended to be incorporated into the Cloudification & Orchestration Use Case and Requirements document.

**Some part of the flow is relevant to WG1 & WG10 OAM Architecture document. Some part of the flow is relevant to WG6 for interaction between NFO & DMS.**

The deployment flow based on ASD does not show interactions between NFO & DMS beyond those that are just stating “request to deploy”. So, it is not clear what needs to be documented in the Orchestration Use Case Specification. This still has yet to be resolved.

**SUMMARY**: The recommendation from this TR is that Deployment Flow Use cases should go into the Cloudification & Orchestration Use Case and Requirements specification.

## Recommendation for DeploymentDescriptor based on ASD

This is described in Clause 6.1 and Clause 6.2

The team producing this TR has considered four options for where that content might be placed:

1. **O2 DMS Interface Spec (new) WG6 Stage 2** –

The O2 DMS Interface Specification would be a sister document in WG6 to the existing O2 IMS Interface Specification. It would be a new document which would also be the home for other O2 DMS topics, such as O2 DMS fault and O2 DMS performance management. So, it could be created for DeploymentDescription as well as the other O2 DMS topic areas.

1. **WG10 App Pkg Spec** [out of place wrong WG]

Another possibility is to put into the WG10 App Package Specification; however, it is not proper working group.

1. **O2 DMS LCM - O2 DeploymentDesc for ASD** (new WG6 document) A new separate dedicated for ASD modelling could be an option.
2. **O2 DMS K8s Profile** [this is a Stage 2 material being put into Stage 3 document]

Incorporating the material into the Stage 3 O2 DMS K8s Profile document. The problem with that is that it is stage 2 material going into a stage 3 document.

**SUMMARY**: This TR recommends that a new specification is created for the ASD content as described in Option #3. The content in Clause 6 from the present document can be used as the seed content and baseline for the new ASD specification.

## Recommendation for DeploymentDescriptor based on VNFD

Recommendation for Deployment Flow for VNFD Use Case

**Some part of the flow is relevant to WG1 & WG10 OAM Architecture document Some part of the flow is relevant to WG6 for interaction between NFO & DMS.**

**SUMMARY**: The recommendation from this TR is that Deployment Flow for Deployment Descriptor based on VNF use cases should go into the Cloudification & Orchestration Use Case and Requirements specification.

# Revision History

**Table – Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revis ion** | **Description** |
| July , 2023 | 02.00 | Incorporated CR to update Security and Recommendations section:  - NOK-2023.06.30-CR0060-WG6 AppLCMDeployment-v01.00 |
| Mar 08,  2023 | 00.10 | Addition of Deployment Flow for VNFD Use Case   * Clause 5.2.1 and Clause 5.2.2 addition of Deployment Flow for VNFD Use Case |
| June 02,  2022 | 00.01 | Initial Draft Discussion versions |

# History

History of the Document

**Table - History of the Document**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision** | **Description** |
| 2023.03 | 01.00 | Published as Final version 01.00 |