ETSI-Alignment Architecture - Honolulu

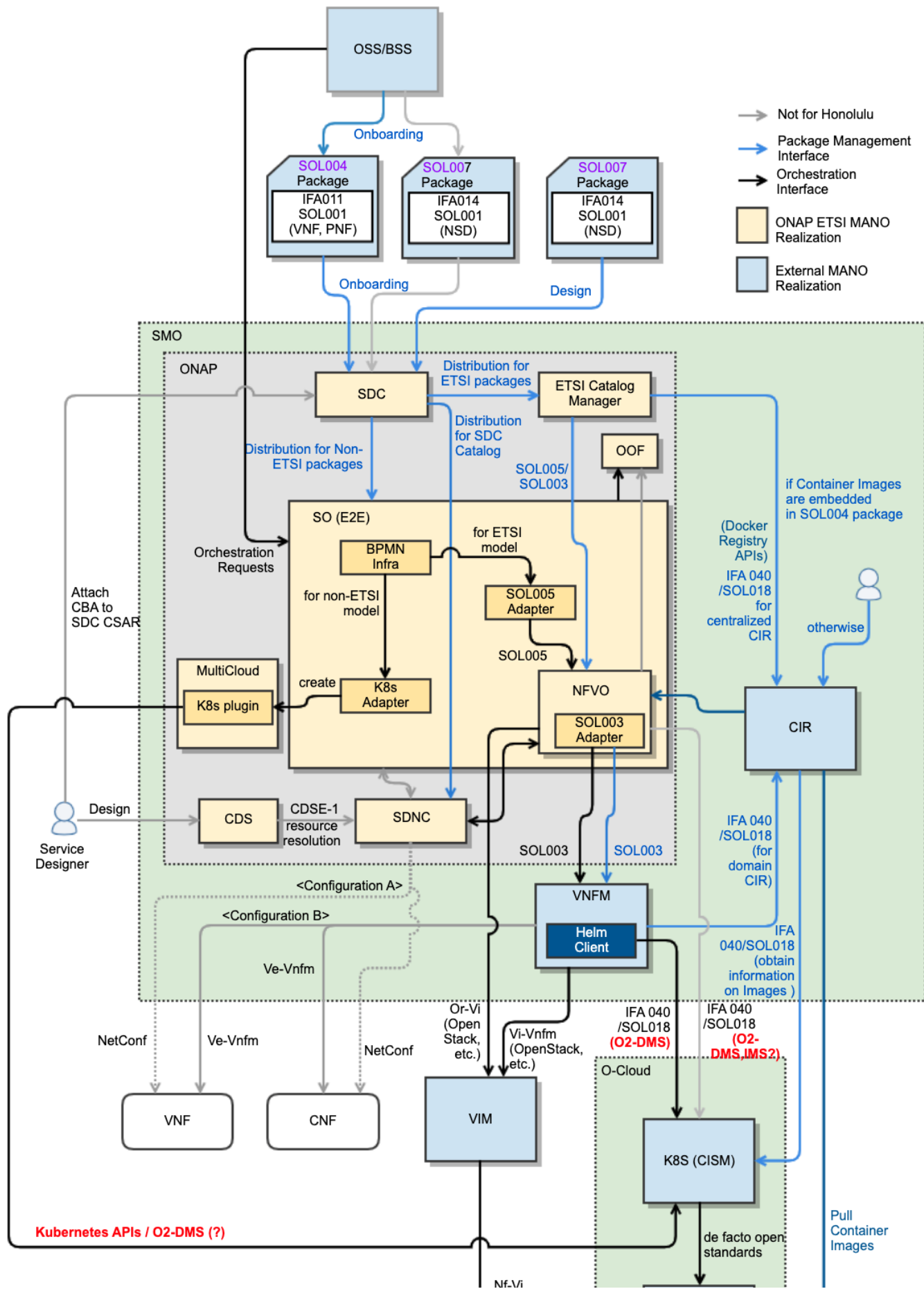## 1- Brief Project Overview (brief as it should be known)

The use case (**REQ-400 - Getting issue details...** `STATUS` **,** **ONAPARC-648 - Getting issue details...** `STATUS` **) intends to**

demonstrate the ETSI-Aligned modeling, package distribution, and orchestration by conforming to ETSI NFV. This use case shall support:
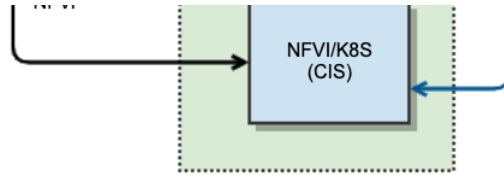
1. **VNF modeling with minimum CNF enhancements from ETSI NFV 4.1.1, by conforming to the ETSI SOL004 Package and SOL001 Modeling standards**
2. **automated NSD and VNF/CNF package E2E distribution, by conforming to SOL005 and SOL003 Package Management APIs**
3. **NS and VNF Orchestration through a modular architecture and standards-based (ETSI NFV) interfaces, by conforming to SOL005 and SOL003 LCM standards**

1.1 Overall Architecture

The following diagram depicts overall ETSI-Alignment Architecture for Honolulu.

- SDC supports SOL004 Package onboarding
- SDC supports SOL007 Design
    - SOL007 Onboarding is not yet supported in Honolulu
- ETSI Catalog Manager receives SDC Package notification
- ETSI Catalog Manager gets the original vendor ETSI NS and VNF packages and stores them into its database
- ETSI Catalog Manager sends the package onboarding notification to its subscribers (SO NFVO, SOL003 Adapter, in the future SOL005 Adapter)
- If the SOL004 VNF package embeds Container Images, ETSI Catalog Manager pushes the images to CIR via Docker Registry APIs
    - Currently, ETSI Catalog Manager defined this Container Image handling as a stretch goal
    - If not (due to SDC 2MB Image file size limitation), ETSI Catalog Manager does not handle Container Images. In that case, the operator will upload Container Image(s) to the CIR component directly
    - In this case, SOL004 Package metadata will reference Container Image file(s)
- CIR will be realized by Nexus with Docker Registry (GitHub is under consideration)
    - CIR component can be delivered part of ONAP or the operator can leverage their CIR component(s)
        - Multiple CIR instances can be used
        - Registration CIR into AAI for ETSI Catalog Manager to find a proper instance of CIR
    - Access APIs to CIR will be Docker Registry APIs, by conforming to ETSI IFA040/SOL018
    - K8S CISM and CIS need to have their own way to find a proper CIR instance to pull Container Images.
- K8S CISM will obtain information on Container Image(s) thru Docker Registry APIs
- K8S CIS will pull Container Image(s) thru Docker Registry APIs
- The Interface between VNFM and K8S CISM will follow K8S APIs
    - Interface between VNFM and CISM: use k8s APIs, potential to leverage an alignment with O2-DMS
    - Interface between MultiCloud K8s Plugin and CISM: use k8s APIs, potential to leverage alignment with O2-DMS
    - VNFM Enhancement (PoC) to invoke CISM on O-Cloud through O2-DMS
    - Note: support O2-DMS is a stretch goal; most likely use K8S APIs
- Architecture wiki page: ETSI-Aligned CNF Support - Honolulu

Legend:
- Not for Honolulu
- Package Management Interface
- Orchestration Interface
- ONAP ETSI MANO Realization
- External MANO Realization

OSS/BSS

Onboarding

SOL004 Package — IFA011 SOL001 (VNF, PNF)
SOL007 Package — IFA014 SOL001 (NSD)
SOL007 Package — IFA014 SOL001 (NSD)

Onboarding

Design

SMO

ONAP

SDC

Distribution for ETSI packages

ETSI Catalog Manager

Distribution for Non-ETSI packages

Distribution for SDC Catalog

OOF

SOL005/ SOL003

if Container Images are embedded in SOL004 package

SO (E2E)

Orchestration Requests

BPMN Infra

for ETSI model

for non-ETSI model

SOL005 Adapter

(Docker Registry APIs)

IFA 040 /SOL018 for centralized CIR

otherwise

Attach CBA to SDC CSAR

MultiCloud

K8s plugin

create

K8s Adapter

SOL005

NFVO

SOL003 Adapter

CIR

Design

Service Designer

CDS

CDSE-1 resource resolution

SDNC

SOL003

SOL003

IFA 040 /SOL018 (for domain CIR)

<Configuration A>

<Configuration B>

VNFM

Helm Client

IFA 040/SOL018 (obtain information on Images)

Ve-Vnfm

NetConf

Ve-Vnfm

NetConf

Or-Vi (Open Stack, etc.)

Vi-Vnfm (OpenStack, etc.)

IFA 040 /SOL018 (O2-DMS)

IFA 040 /SOL018 (O2-DMS,IMS2)

VNF

CNF

VIM

O-Cloud

K8S (CISM)

Kubernetes APIs / O2-DMS (?)

Pull Container Images

de facto open standards

Nf-Vi

1.2 ONAP SO CNF Orchestration Support

There are two tracks in ONAP for CNF support: ONAP ETSI-Alignment and Direct Path for CNF (Based on models, SO selects a proper path):

- **ETSI-Alignment path**
    - Following ETSI NFV IFA specifications for CNF
    - Leveraging and extending existing ONAP ETSI-Alignment Architecture
- **Direct Path**
    - Some think there is no need for the full VNFD for LCM, just Helm Charts and Images; delegate LCM to CISM; it may work for simple CNFs
    - Note: there are several cases where a more complete VNFD will be needed, such as 5G Core NFs.
- **Collaboration Areas:**
    - Unified Modeling, Packaging & Onboarding (SDC) – optional VNFD?
    - Package Distribution (SDC – ETSI Catalog Manager – CIR)
    - Common Components (e.g., CIR, A&AI, CDS)
- **Note: CIR will be realized by Nexus with Docker Registry (possibly migrate it to Github), and CIR access APIs will be based on Docker Registry APIs.**

2- New component capabilities for Honolulu, i.e. the functional enhancements, if applicable

- **SDC**
    - Enhancements related to ETSI 3.3.1 SOL004 compliant VNF / CNF package onboarding and distribution
        - ETSI Package Management (SDC Enhancements) - Honolulu
        - ETSI Package Management (SDC Enhancements) - Honolulu#ETSIPackageManagementArchitecture
    - Supports ETSI 3.3.1 VNF with minimum CNF enhancements from 4.1.1
        - VNF/CNF Data Model Based on ETSI v3.3.1 SOL001 plus CNF Enhancements
        - CSAR Models: ETSI-Alignment VNFD and Example(s)
    - Enhancements related to ETSI 3.3.1 SOL007 compliant NSD package Design and Distribution
        - ETSI Package Management (SDC Enhancements) - Honolulu#NSOnboardingDesign: Design SOL007
        - ETSI Package Management (SDC Enhancements) - Honolulu#DesignNS: Design NS
    - Enhancements related to ETSI 3.3.1 SOL004 VNF and SOL007 NS package security
        - ETSI Package Management (SDC Enhancements) - Honolulu#SDCSOL004/SOL007VNFPackageSecurity
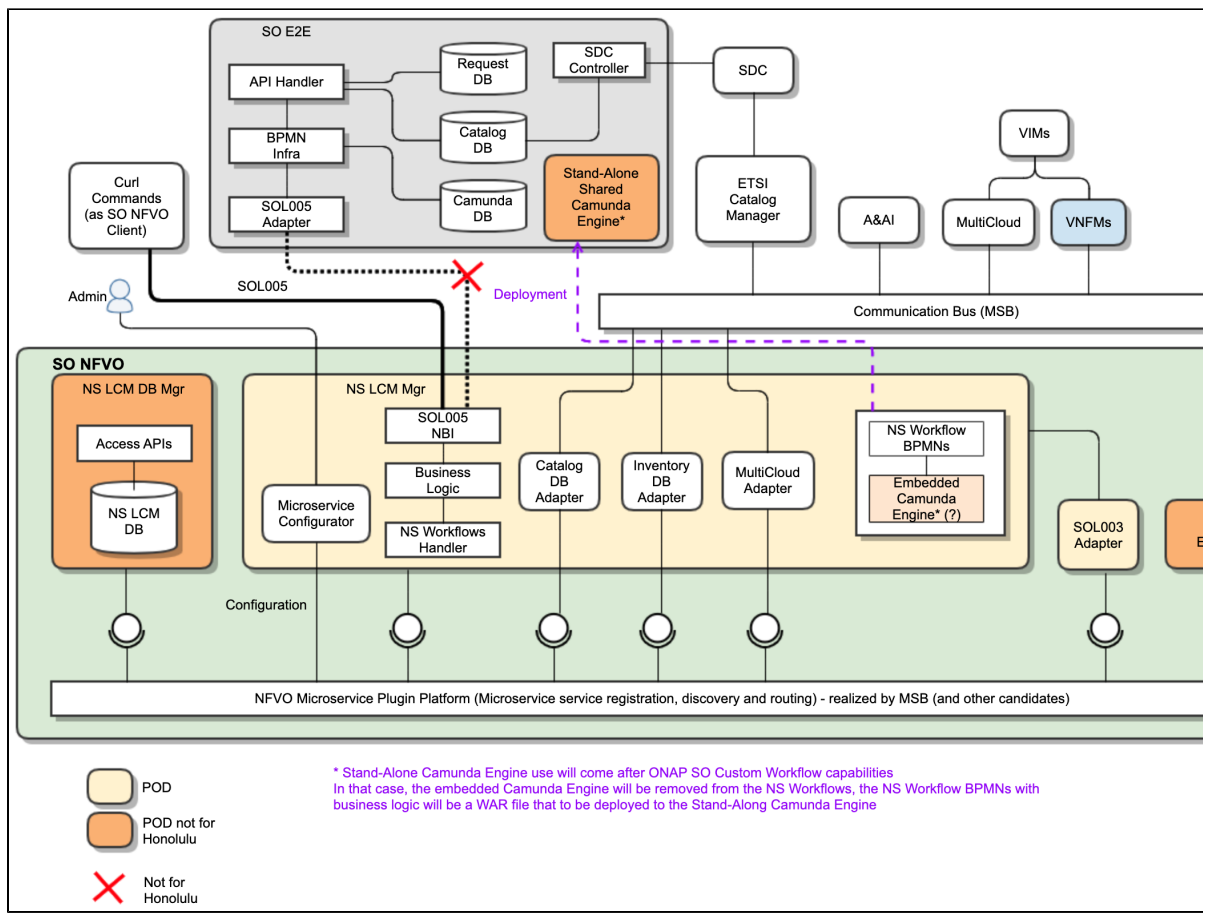   - Architecture Diagram for ETSI package handling and distribution

ONAP ETSI-ALIGNMENT PACKAGE DISTRIBUTION

The ETSI Catalog Manager supports the following features in Honolulu

- Support Subscription and Notification for NSD packages, https://wiki.onap.org/display/DW/Modeling+R8+Honolulu+Architecture+Review
  - SO NFVO will subscribe to ETSI Catalog Manager and provide a NSD package notification by conforming to SOL005
- Support Container Images Management to CIR
  - ETSI Catalog Manager pushes Container Images to the selected CIR if the SOL004 VNF package embeds Container Images
  - Leverage Docker Registry APIs
  - CIR will be realized by Nexus with Docker Registry

CIR will be realized by Nexus with Docker Registry

- Register CIR to A&AI
- ETSI Catalog Manager selects the CIR instance
- CIR provides Container Image uploading in the case where the SOL004 VNF package does not embed Container images

For SDC Modeling Support, see the subsequence page

- docker tag <imageId or imageName> <nexus-hostname>:<repository-port>/<image>:<tag>
- docker tag af340544ed62 nexus.example.com:18444/hello-world:mytag
- docker push <nexus-hostname>:<repository-port>/<image>:<tag>
- Docker pull <nexus-hostname>:<repository-port>/<image>:<tag>
  - docker push nexus.example.com:18444/hello-world:labeltest
  - docker pull nexus.example.com:18443/hello-world:labeltest

---

- **SO / SO NFVO**
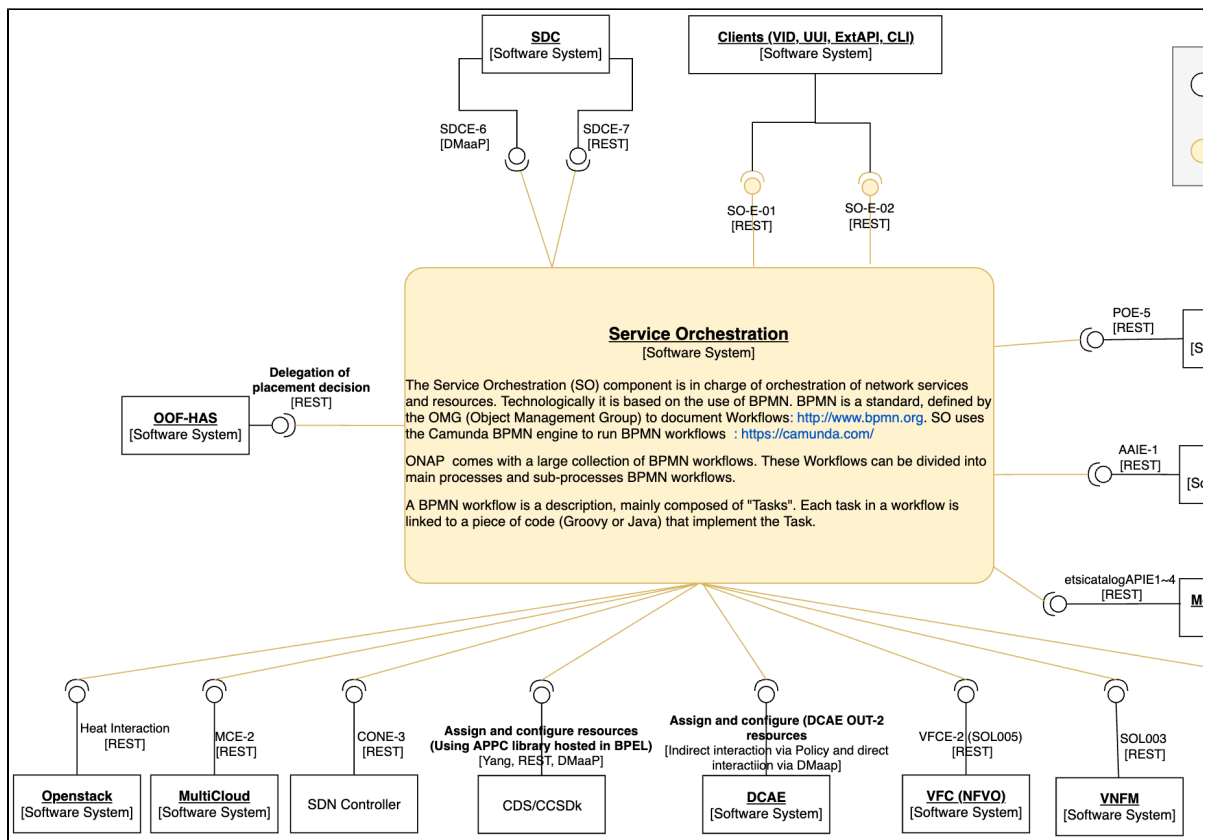 - SO NFVO continues to support the same SOL005 API scope as Guilin: Create NS, Instantiate NS, Terminate NS, Delete NS and Get NS Operation Status
    - In addition, it will support NSD subscription and notification, interacting with ETSI Catalog Manager
    - However, SO NFVO will upgraded ETSI v3.3.1 SOL005 LCM APIs from ETSI v2.7.1 SOL005 LCM APIs
      - ETSI v3.3.1 SOL005 API swagger
        - Use the latest SOL005 API swagger from https://forge.etsi.org/jenkins/view/All+jobs/job/NFV+-+Network+Functions+Virtualisation/job/sol005-master/lastSuccessfulBuild/artifact/build/
        - SOL005-NSLifecycleManagement-API.json
        - SOL005-NSLifecycleManagementNotification-API.json
        - SOL005-NSPackageManagement-API.json
        - SOL005-NSPackageMaangementNotification-API.json
      - ETSI v3.3.1 SOL003 API swagger
        - Use the latest SOL003 API swagger from https://forge.etsi.org/jenkins/view/All%20jobs/job/NFV%20-%20Network%20Functions%20Virtualisation/job/sol002-sol003-master/lastSuccessfulBuild/artifact/build/
        - SOL003-VNFLifecycleManagement-API.json
        - SOL003-VNFLifecycleManagementNotification-API.json
        - SOL003-VNFPackageManagement-API.json
        - SOL003-VNFPackageManagementNotification-API.json
    - SO NFVO and SOL003 Adapter CSIT will be supported to assist SO E2E Integration testing
    - Epics and User Stories: ONAP SO Hierarchical Orchestration (SO NFVO) - Honolulu#EpicandUserStory
  - SO NFVO Honolulu Architecture Highlights
    - SO NFVO will be invoked by Curl Commands (as SO NFVO Client) since the SOL005 Adapter enhancements are out of scope
    - SO NFVO NS Workflow BPMNs and business logic will be packaged as a war file and deployed to ONAP SO Standalone /Clustered Camunda Engine
      - Dynamic Orchestration support is a trial goal for Honolulu. In that case, NS Workflow BPMNs and business logic (with Embedded Camunda engine) will be part of SO NFVO, as is
    - SO NFVO and SOL003 Adapter communications will be treated as an internal communication via HTTP
    - SO NFVO interacts with ETSI Catalog Manager thru MSB for NSD subscription and notification
  - Architecture Diagram: ONAP SO Hierarchical Orchestration (SO NFVO) - Honolulu#HonoluluSONFVOTesting

SO E2E

API Handler
Request DB
SDC Controller
SDC

BPMN Infra
Catalog DB
Stand-Alone Shared Camunda Engine*

SOL005 Adapter
Camunda DB

Curl Commands (as SO NFVO Client)

ETSI Catalog Manager

VIMs

A&AI
MultiCloud
VNFMs

Admin

SOL005

Deployment

Communication Bus (MSB)

SO NFVO

NS LCM DB Mgr

Access APIs

NS LCM DB

NS LCM Mgr

Microservice Configurator

SOL005 NBI

Business Logic

NS Workflows Handler

Catalog DB Adapter

Inventory DB Adapter

MultiCloud Adapter

NS Workflow BPMNs

Embedded Camunda Engine* (?)

SOL003 Adapter

Configuration

NFVO Microservice Plugin Platform (Microservice service registration, discovery and routing) - realized by MSB (and other candidates)

POD

POD not for Honolulu

Not for Honolulu

* Stand-Alone Camunda Engine use will come after ONAP SO Custom Workflow capabilities
In that case, the embedded Camunda Engine will be removed from the NS Workflows, the NS Workflow BPMNs with business logic will be a WAR file that to be deployed to the Stand-Along Camunda Engine

- SO Interface Update
    - The following diagram depicts SO interfaces. The interactions with ETSI Catalog Manager (a.k.a. Modeling etsicalog) are added
   - See the link for details, ARC Service Orchestrator Component Description - Honolulu-R8

SDC
[Software System]

Clients (VID, UUI, ExtAPI, CLI)
[Software System]

SDCE-6
[DMaaP]

SDCE-7
[REST]

SO-E-01
[REST]

SO-E-02
[REST]

POE-5
[REST]

Service Orchestration
[Software System]

The Service Orchestration (SO) component is in charge of orchestration of network services and resources. Technologically it is based on the use of BPMN. BPMN is a standard, defined by the OMG (Object Management Group) to document Workflows: http://www.bpmn.org. SO uses the Camunda BPMN engine to run BPMN workflows : https://camunda.com/

ONAP comes with a large collection of BPMN workflows. These Workflows can be divided into main processes and sub-processes BPMN workflows.

A BPMN workflow is a description, mainly composed of "Tasks". Each task in a workflow is linked to a piece of code (Groovy or Java) that implement the Task.

Delegation of placement decision
[REST]

OOF-HAS
[Software System]

AAIE-1
[REST]

etsicatalogAPIE1~4
[REST]

Heat Interaction
[REST]

MCE-2
[REST]

CONE-3
[REST]

**Assign and configure resources
(Using APPC library hosted in BPEL)**
[Yang, REST, DMaaP]

Assign and configure (DCAE OUT-2 resources
[Indirect interaction via Policy and direct interactiion via DMaap]

VFCE-2 (SOL005)
[REST]

SOL003
[REST]

Openstack
[Software System]

MultiCloud
[Software System]

SDN Controller

CDS/CCSDk

DCAE
[Software System]

VFC (NFVO)
[Software System]

VNFM
[Software System]

- **ETSI Catalog Manager**
    - ○ Enhancements related to direct SDC Package distribution support thru SDCE-6
    - ○ Enhancements related to NSD Subscription and Notification
    - ○ Architectural Relationships and Interfaces support: ARC Modeling Component Description – Honolulu-R8
    - ○ Architecture Diagram:
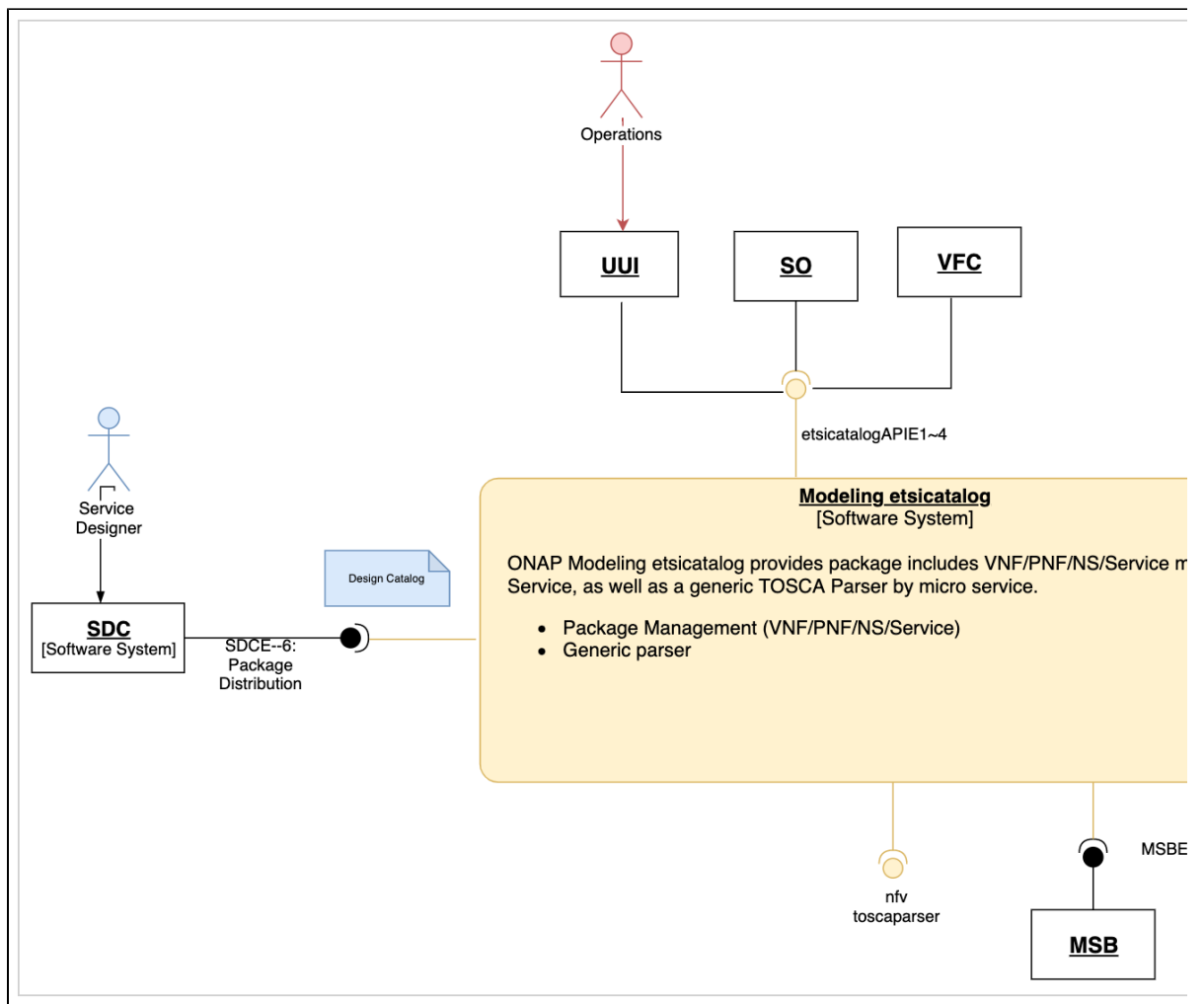
- Enhancements related to Container Image handling is a stretch goal
    - Most likely due to SDC 2MB file size limitation, Container Images will be uploaded to the CIR component by the Admin
    - i.e., in that case, no impact on ETSI Catalog Manager in Honolulu for Container Image handling
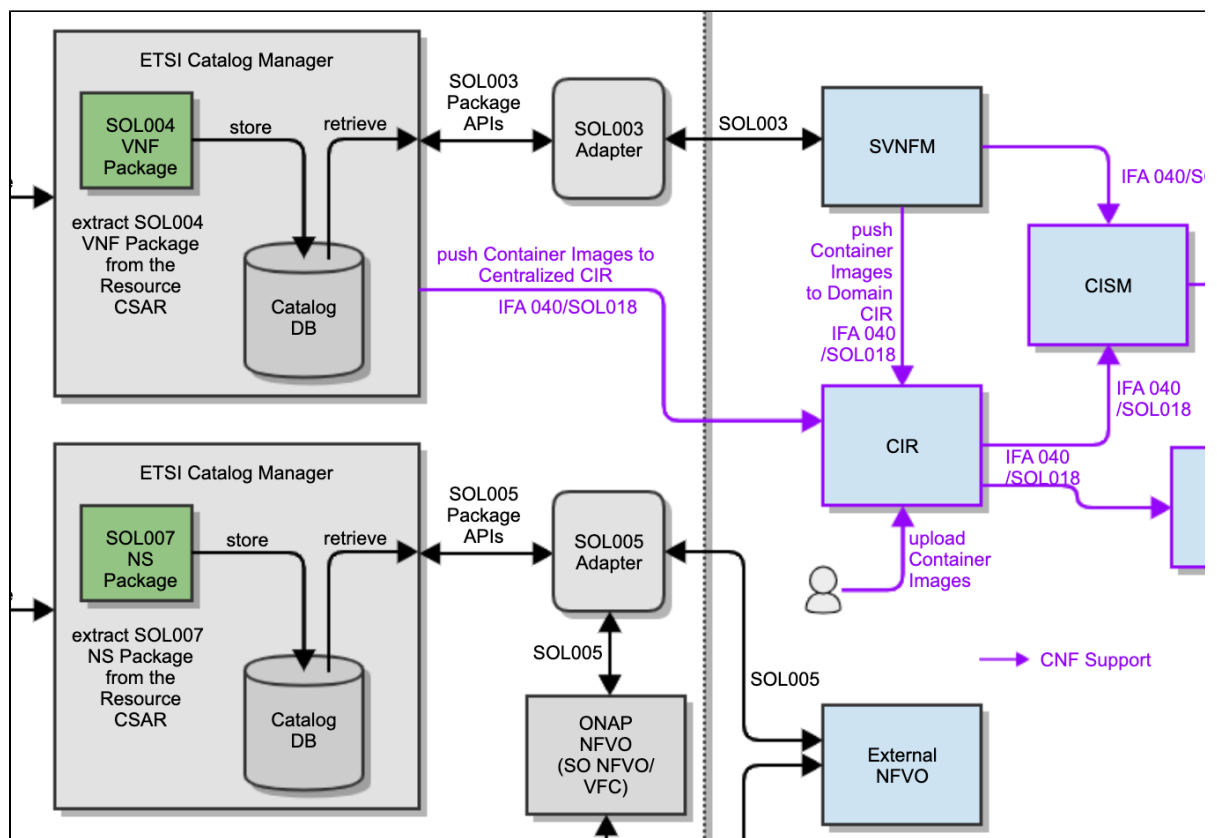
- **AAI**
  - No impacts since the SO NFVO does not support Scaling or Healing
  - AAI impacts will be analyzed for the Istanbul release

3- New or modified interfaces

- SDC ETSI Catalog Manager
- ETSI Catalog Manager SO NFVO

4- If they are modified, are they backwards compatible?

- SDC Onboarding supports one-level backward compatibility.
  - For Honolulu release, SDC supports ETSI v3.3.1 and v2.7.1/2.6.1
- SDC data model does not have the version support, the SDC data models will be super-set data structures to hold both versions
  - SDC data model version support would be an Istanbul release consideration

6- Interface naming (point to an example)

7- Consumed API from other projects

Project	API Dependency	Notes	APi Specs (Swagger)
SO / SO NFVO	ETSI Catalog Manager Package Management APIs for NS	ETSI Catalog Manager APIs	https://wiki.onap.org/download/attachments/79200593/etsicatalog_API_v1.yaml?version=1&modificationDate=1579574658000&api=v2

SOL003 Adapter	ETSI Catalog Manager Package Management APIs for VNF	ETSI Catalog Manager APIs	https://wiki.onap.org/download/attachments/79200593/etsicatalog_API_v1.yaml?version=1&modificationDate=1579574658000&api=v2

8- Published API

Project	API	Notes	APi Specs (Swagger)
SOL003 Adapter	ETSI SOL003 Package Management & LCM APIs	Support APIs for VNFM	https://forge.etsi.org/jenkins/view/All%20jobs/job/NFV%20-%20Network%20Functions%20Virtualisation/job/sol002-sol003-master/lastSuccessfulBuild/artifact/build/

9- Reference to the interfaces.

(Reference to the the swagger.json file(s) whenever possible)

10- What are the system limits?

.

11- Involved use cases, architectural capabilities or functional requirements.

12- Listing of new or impacted models used by the project (for information only).

- Identify any High Level Information Model Requirements.   See: ONAP R8 Modeling High Level Requirements
    - Models based on information exchanges from Use Cases
    - Models documenting existing implementations
    - Forward looking models that may be implemented in future releases
- Describe how exposed APIs are mapped to information models

(list all the relevant Jira tickets)

13-Test plan/Testing Strategy

1. Unit Testing
2. Dev-to-Dev Testing and
3. Integration

14- Any other details that are specific to this functional enhancement or UseCase.