Modeling Honolulu Release Planning

- 1 Overview
- 2 Scope
  - 2.1 What is this release trying to address?
  - 2.2 Requirements
  - 2.3 Minimum Viable Product
  - 2.4 Functionalities
    - 2.4.1 Epics
    - 2.4.2 Stories
  - 2.5 Longer term roadmap
- 3 Release Deliverables
- 4 Sub-Components
- 5 Architecture
  - 5.1 High level architecture diagram
  - 5.2 Platform Maturity
  - 5.3 API Incoming Dependencies
  - 5.4 API Outgoing Dependencies
  - 5.5 Third Party Products Dependencies
- 6 Testing and Integration Plans
- 7 Gaps
- 8 Known Defects and Issues
- 9 Risks
- 10 Resources
- 11 Release Milestone
- 12 Team Internal Milestone
- 13 Documentation, Training
- 14 Other Information
  - 14.1 Vendor Neutral
  - 14.2 Free and Open Source Software

Overview

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Enter the name of the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Release Name</td>
<td>Honolulu</td>
</tr>
<tr>
<td>Project Lifecycle State</td>
<td>Mature</td>
</tr>
<tr>
<td>Participating Company</td>
<td>China Mobile, ZTE, Huawei</td>
</tr>
</tbody>
</table>

Scope

What is this release trying to address?

etsicatalog project:
  - Enhance function of Subscription and Notification for NSD packages
  - Support upload Container Images to CIR
  - Satisfy TSC must have items

Modelspec for modeling subcommittee:
  - Create UML models for new model proposals or enhancements in Honolulu release
  - Document approved models as model specs

Requirements

Describe the use case this release is targeted for (better if reference to customer requirements).

Modeling/etsicatalog will contribute to the following functional requirements:

Functional requirements:
ETSI-Alignment Support for Honolulu

REQ-400 - Getting issue details…

modelspec will address the following modeling requirements:

- ONAP R8 Modeling High Level Requirements

Minimum Viable Product

Describe the MVP for this release.

1) ETSI Interface alignment (SOL005 and SOL003) API implement
2) Code contribution to support the ETSI-Alignment Support for Honolulu
3) Modelspec from modeling subcommittee

Functionalities

List the functionalities that this release is committing to deliver by providing a link to JIRA Epics and Stories. In the JIRA Priority field, specify the priority (either High, Medium, Low). The priority will be used in case de-scoping is required. Don’t assign High priority to all functionalities.

Epics

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>T</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
<th>Assignee</th>
<th>Reporter</th>
<th>P</th>
<th>Status</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODELI NG-512</td>
<td>Release Candidate 2 Integration and Test</td>
<td>🌟</td>
<td>Apr 09, 2021</td>
<td>May 07, 2021</td>
<td>Apr 22, 2021</td>
<td>Yuanhong Deng</td>
<td>David McBride</td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>MODELI NG-507</td>
<td>Release Candidate 1 Integration and Test</td>
<td>🌟</td>
<td>Mar 17, 2021</td>
<td>Apr 15, 2021</td>
<td>Mar 25, 2021</td>
<td>Yuanhong Deng</td>
<td>David McBride</td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>MODELI NG-497</td>
<td>Release Candidate 0 Integration and Test</td>
<td>🌟</td>
<td>Feb 23, 2021</td>
<td>Apr 15, 2021</td>
<td>Mar 11, 2021</td>
<td>Yuanhong Deng</td>
<td>David McBride</td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>MODELI NG-480</td>
<td>Feature Freeze</td>
<td>🌟</td>
<td>Feb 01, 2021</td>
<td>Mar 16, 2021</td>
<td>Feb 25, 2021</td>
<td>Yuanhong Deng</td>
<td>David McBride</td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
</tr>
<tr>
<td>MODELI NG-461</td>
<td>This epic covers the work to satisfy TSC Must-Have Items for Honolulu Release</td>
<td>🌟</td>
<td>Nov 25, 2020</td>
<td>May 07, 2021</td>
<td>Yuanhong Deng</td>
<td>Yuanhong Deng</td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
<td></td>
</tr>
<tr>
<td>MODELI NG-458</td>
<td>Support Subscription and Notification for NSD packages</td>
<td>🌟</td>
<td>Nov 23, 2020</td>
<td>May 07, 2021</td>
<td>Yuanhong Deng</td>
<td>Byung-Woo Jun</td>
<td></td>
<td>CLOSED</td>
<td>Done</td>
<td></td>
</tr>
</tbody>
</table>

8 issues

Stories

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>T</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
<th>Assignee</th>
<th>Reporter</th>
<th>P</th>
<th>Status</th>
<th>Resolution</th>
</tr>
</thead>
</table>
Longer term roadmap

Indicate at a high level the longer term roadmap. This is to put things into the big perspective.

Release Deliverables

Indicate the outcome (Executable, Source Code, Library, API description, Tool, Documentation, Release Note, etc) of this release.

<table>
<thead>
<tr>
<th>Deliverable Name</th>
<th>Deliverable Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Code</td>
<td>Source code for etsicatalog</td>
</tr>
<tr>
<td>Maven Artifacts</td>
<td>Maven Artifacts for etsicatalog</td>
</tr>
<tr>
<td>Docker Containers</td>
<td>Docker container associated with etsicatalog</td>
</tr>
<tr>
<td>Documentation</td>
<td>Modeling detailed documentation</td>
</tr>
</tbody>
</table>

Sub-Components

List all sub-components part of this release. Activities related to sub-components must be in sync with the overall release.

Sub-components are repositories and are consolidated in a single centralized place. Edit the Release Components name for your project in the centralized page.

Architecture

High level architecture diagram

At that stage within the Release, the team is expected to provide more Architecture details describing how the functional modules are interacting.

Indicate where your project fit within the ONAP Architecture diagram.

Block and sequence diagrams showing relation within the project as well as relation with external components are expected.

Anyone reading this section should have a good understanding of all the interacting modules.
Platform Maturity

Please fill out the centralized wiki page: Honolulu Release Platform Maturity

**API Incoming Dependencies**

List the API this project is expecting from other projects.
Prior to Release Planning review, Team Leads must agreed on the date by which the API will be fully defined. The API Delivery date must not be later than the release API Freeze date.

Prior to the delivery date, it is a good practice to organize an API review with the API consumers.

<table>
<thead>
<tr>
<th>API Name</th>
<th>API Description</th>
<th>API Definition Date</th>
<th>API Delivery Date</th>
<th>API Definition link (i.e. swagger)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDCE-6</td>
<td>SDC Interface for ONAP components subscribe to service notification from DMaaP</td>
<td></td>
<td></td>
<td>SDC API</td>
</tr>
<tr>
<td>SDCE-7</td>
<td>SDC Interface for ONAP components retrieve service models from the Design Catalog</td>
<td></td>
<td></td>
<td>SDC API</td>
</tr>
<tr>
<td>DMaaP-3</td>
<td>DMaaP Message Router Consuming Interface to provide a message receiving service to the DMaaP user</td>
<td></td>
<td></td>
<td>DMaaP API</td>
</tr>
<tr>
<td>MSBE-1</td>
<td>Micro Service Bus Interface for registration and discovery with the ONAP Micro Service Bus</td>
<td></td>
<td></td>
<td>Microservice Bus API Documentation</td>
</tr>
</tbody>
</table>

**API Outgoing Dependencies**

This project is delivering to other projects.

<table>
<thead>
<tr>
<th>API Name</th>
<th>API Description</th>
<th>API Definition Date</th>
<th>API Delivery date</th>
<th>API Definition link (i.e. swagger)</th>
</tr>
</thead>
<tbody>
<tr>
<td>etsicatalogAPIE-1</td>
<td>Catalog API</td>
<td></td>
<td></td>
<td>etsicatalog API Document</td>
</tr>
<tr>
<td>etsicatalogAPIE-2</td>
<td>NSD Package Management API</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>etsicatalogAPIE-3</td>
<td>VNF Package Management API</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>etsicatalogAPIE-4</td>
<td>Parser API</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Third Party Products Dependencies**
Third Party Products mean products that are mandatory to provide services for your components. Development of new functionality in third party product may or not be expected.

List the Third Party Products (OpenStack, ODL, RabbitMQ, ElasticSearch, Crystal Reports, ...).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Django</td>
<td><a href="https://www.djangoproject.com/">https://www.djangoproject.com/</a></td>
<td>2.1.10</td>
</tr>
<tr>
<td>djangorestframework</td>
<td><a href="https://www.django-rest-framework.org/">https://www.django-rest-framework.org/</a></td>
<td>3.10.3</td>
</tr>
</tbody>
</table>

In case there are specific dependencies (Centos 7 vs Ubuntu 16, Etc.) list them as well.

1. **Testing and Integration Plans**

   Provide a description of the testing activities (unit test, functional test, automation,...) that will be performed by the team within the scope of this release.

   Describe the plan to integrate and test the release deliverables within the overall ONAP system.

   Confirm that resources have been allocated to perform such activities.

   For Unit Test, Modeling/esticatalog is required to maintain 55% codecoverage at the minimum.


   Functional test plan:

   Modeling/esticatalog Honolulu Functional Test Cases

2. **Gaps**

   This section is used to document a limitation on a functionality or platform support. We are currently aware of this limitation and it will be delivered in a future Release.

   List identified release gaps (if any), and its impact.

<table>
<thead>
<tr>
<th>Gaps identified</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>To fill out</td>
<td>To fill out</td>
</tr>
</tbody>
</table>

3. **Known Defects and Issues**

   N/A

4. **Risks**

   List the risks identified for this release along with the plan to prevent the risk to occur (mitigation) and the plan of action in the case the risk would materialized (contingency).

   Please update any risk on the centralized wiki page - Honolulu Risks

5. **Resources**

   Please see the INFO.yaml files associated with each repo as the authoritative sources of information. https://gerrit.onap.org/r/admin/repos/q/filter:modeling

6. **Release Milestone**

   The milestones are defined at the Release Planning: Honolulu and all the supporting project agreed to comply with these dates.

7. **Team Internal Milestone**

   This section is optional and may be used to document internal milestones within a project team or multiple project teams. For instance, in the case the team has made agreement with other team to deliver some artifacts on a certain date that are not in the release milestone, it is recommended to provide these agreements and dates in this section.

   It is not expected to have a detailed project plan.

<table>
<thead>
<tr>
<th>Date</th>
<th>Project</th>
<th>Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>To fill out</td>
<td>To fill out</td>
<td>To fill out</td>
</tr>
</tbody>
</table>
• **Documentation, Training**

Please update the following centralized wiki: [Honolulu Documentation](#)

That includes:

- Team contributions to the specific document related to the project (Config guide, installation guide...).
- Team contributions to the overall Release Documentation and training asset
- High level list of documentation, training and tutorials necessary to understand the release capabilities, configuration and operation.
- Documentation includes items such as:
  - Installation instructions
  - Configuration instructions
  - Developer guide
  - End User guide
  - Admin guide
  - ...

---

**Note**

The Documentation project will provide the Documentation Tool Chain to edit, configure, store and publish all Documentation asset.

---

**Other Information**

- **Vendor Neutral**

If this project is coming from an existing proprietary codebase, ensure that all proprietary trademarks, logos, product names, etc. have been removed. All ONAP deliverables must comply with this rule and be agnostic of any proprietary symbols.

- **Free and Open Source Software**

FOSS activities are critical to the delivery of the whole ONAP initiative. The information may not be fully available at Release Planning, however to avoid late refactoring, it is critical to accomplish this task as early as possible.
List all third party Free and Open Source Software used within the release and provide License type (BSD, MIT, Apache, GNU GPL,... ).
In the case non Apache License are found inform immediately the TSC and the Release Manager and document your reasoning on why you believe we can use a non Apache version 2 license.

Each project must edit its project table available at [Project FOSS](#).

---

**Charter Compliance**

The project team comply with the [ONAP Charter](#).