Service Orchestrator Beijing Release

Draft copy for proposal

Overview

Project Name	Enter the name of the project
Target Release Name	Beijing
Project Lifecycle State	Incubation
Participating Company	Amdocs, AT&T, China Mobile, Huawei, Intel, Nokia, Orange, ZTE

Scope

What is this release trying to address?

1. Platform maturity - enhance stability, performance, security etc. to strive for carrier-grade maturity.

2. New requirements:

    a. Make ONAP SO carrier grade

    b. **TBD**

Use Cases

SO will continue to support:

Use Case: vFW/vDNS (Approved)

Use Case: Residential Broadband vCPE (Approved)

Use Case: VoLTE(approved)

Minimum Viable Product

- Enhance ONAP SO run-time orchestration framework to support orchestration driven from declarative and imperative models.
- Perform lifecycle operations based on a HEAT and TOSCA model, including:
    1. Deployment
    2. Undeployment

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

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
SO-390	Support homing solution in SO workflows through OOF (HPA)		Jan 25, 2018	Aug 12, 2023		Unassigned	Seshu Kumar Mudiganti		CLOSED	Done
SO-375	To make SO carrier Grade		Jan 08, 2018	Aug 12, 2023		Seshu Kumar Mudiganti	Seshu Kumar Mudiganti		CLOSED	Done
SO-368	To improve the SO Amsterdam release		Jan 03, 2018	Aug 12, 2023		Seshu Kumar Mudiganti	Seshu Kumar Mudiganti		CLOSED	Done
SO-24	In order to implement use cases for alternative 1, they must be modeled.		Jun 29, 2017	Dec 28, 2017		Unassigned	None		CLOSED	Done
SO-23	In order for use cases to be modeled, ONAP services need to be represented as TOSCA types		Jun 29, 2017	Jul 25, 2018		Unassigned	None		CLOSED	Done
SO-22	Alternative 1 requires that the TOSCA orchestrator be part of the SO.		Jun 29, 2017	Aug 12, 2023		Unassigned	None		CLOSED	Done

6 issues

Stories

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
SO-553	Add "controllerType" field to requestParameters in APIH		Mar 30, 2018	Dec 07, 2018		Unassigned	None		CLOSED	Done
SO-537	remove SO security vulnerability and licensing		Mar 27, 2018	Aug 12, 2023		Byung-Woo Jun	Byung-Woo Jun		CLOSED	Done
SO-526	Change management in-place software upgrade workflow		Mar 22, 2018	Aug 12, 2023		Unassigned	None		CLOSED	Done
SO-500	SO String concatenation optimization		Mar 19, 2018	Apr 10, 2018		Byung-Woo Jun	Byung-Woo Jun		CLOSED	Done
SO-499	Increase the size of your varchar column for InfraActiveRequests AIC_CLOUD_REGION		Mar 19, 2018	Dec 07, 2018		Unassigned	None		CLOSED	Done
SO-489	Improve code coverage in Beijing release		Mar 15, 2018	Apr 10, 2018		Seshu Kumar Mudiganti	None		CLOSED	Done

SO-467	Improve Code coverage		Mar 09, 2018	Apr 18, 2018	Unassigned	None		CLOSED	Done
SO-458	Resolve the critical vulnerabilities in the third party libraries of SO-libs		Mar 06, 2018	Aug 12, 2023	Byung-Woo Jun	Seshu Kumar Mudiganti		CLOSED	Done
SO-457	Resolve the critical vulnerabilities in the third party libraries of SO		Mar 06, 2018	Aug 12, 2023	Byung-Woo Jun	Seshu Kumar Mudiganti		CLOSED	Done
SO-441	Create Vdu Adapter for ARIA		Feb 22, 2018	Apr 19, 2018	Unassigned	None		CLOSED	Done
SO-430	Write Junit for MSO-Adapter package		Feb 19, 2018	Mar 07, 2018	Unassigned	None		CLOSED	Done
SO-428	Add VDU code		Feb 16, 2018	Apr 25, 2018	Unassigned	None		CLOSED	Done
SO-418	Model-driven service orchestration and execution		Feb 13, 2018	Mar 29, 2018	Unassigned	Seshu Kumar Mudiganti		CLOSED	Done
SO-370	Improve the E2E service custom workflow to generic flows		Jan 03, 2018	Mar 29, 2018	Unassigned	Seshu Kumar Mudiganti		CLOSED	Done
SO-353	Fix sonar issues		Dec 01, 2017	Apr 10, 2018	Unassigned	None		CLOSED	Done
SO-298	Add JavaDoc comments to SDNOValidator		Nov 01, 2017	Oct 25, 2018	Unassigned	None		CLOSED	Done

16 issues

Longer term roadmap

SO aims at providing model driven orchestration to be provided to ONAP users.

Release Deliverables

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

Deliverable Name	Deliverable Description	Deliverable Location
SO Docker Images	Executable	Docker image available on nexus3
MariaDB Docker Image	Executable	Docker image available on nexus3
SO Libs	Jar files	Available on nexus
Deployment Scripts	Chef recipes used to configure the Docker containers.	found on SO Git repositories

Sub-Components

List all sub-components part of this release. Activities related to sub-component 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

Refering to CII Badging Security Program and Platform Maturity Requirements, fill out the table below by indicating the actual level , the targeted level for the current release and the evidences on how you plan to achieve the targeted level.

Area	Actual Level	Targeted Level for current Release	How, Evidences	Comments
Performance	0	1	Need to work with the benchmarking project and define the critera	• 0 -- none • 1 – baseline performance criteria identified and measured • 2 & 3 – performance improvement plans created & implemented
Stability	0	1	1.The load should be defined with the consensus of the benchmarking project 2.Need to soak test the SO for 72 hours with the defined load	• 0 – none • 1 – 72 hours component level soak w/random transactions • 2 – 72 hours platform level soak w/random transactions • 3 – 6 months track record of reduced defect rate
Resiliency	1	1 (Stretch goal 2)	Have to incorporate the rollback flows for all the active workflows. SO needs to check on component level resiliency.	• 0 – none • 1 – manual failure and recovery (< 30 minutes) • 2 – automated detection and recovery (single site) • 3 – automated detection and recovery (geo redundancy)
Security	0	1	1.The current code coverage needs improvement from 29% to 50% 2.Improve the documentation (readthedocs) of the SO describing what the software does (what problem does it solve?). 3. use at least one automated test suite that is publicly released as FLOSS	• 0 – none • 1 – CII Passing badge + 50% Test Coverage • 2 – CII Silver badge; internal communication encrypted; role-based access control and authorization for all calls • 3 – CII Gold
Scalability	0	1	SO will have 1. new APIs for manual and Auto scaling for platform enhancement 2. at component level SO is identifying the ways to support scaling.	• 0 – no ability to scale • 1 – single site horizontal scaling • 2 – geographic scaling • 3 – scaling across multiple ONAP instances
Manageability	0	1	1.SO should use the common logging for all its logs 2.SO dockers (both MSO and MariaDB) should be instantiated within one hour	• 1 – single logging system across components; instantiation in < 1 hour • 2 – ability to upgrade a single component; tracing across components; externalized configuration management
Usability	0	1	1.Improve SO user guide 2. Improve the API documentation	• 1 – user guide; deployment documentation; API documentation • 2 – UI consistency; usability testing; tutorial documentation

API Incoming Dependencies

List the API this release is expecting from other releases. 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.

API Name	API Description	API Definition Date	API Delivery date	API Definition link (i.e.swagger)

SDC	APIs for distribute service models	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
A&AI	APIs for inventory data	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
SDN-C	APIs for network controller	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
APP-C	APIs for application controller	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
VF-C	APIs for Network Service	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Multi-VIM	APIs for Multi-VIM	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link

API Outgoing Dependencies

API this release is delivering to other releases.

API Name	API Description	API Definition Date	API Delivery date	API Definition link (i.e.swagger)
Create service instance	Create a service instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Delete service instance	Delete a service instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Create vnf instance	Create vnf instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Delete vnf instance	Delete vnf instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Create vf module instance	Create vf module instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Delete vf module instance	Delete vf module instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Create volume group instance	Create volume group instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Delete volume group instance	Delete volume group instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Create network instance	Create network instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Delete network instance	Delete network instance	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Get orchestration requests	Get orchestration requests	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Get a specific orchestration request	Get a specific orchestration request	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Create E2E service	Create E2E service	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Delete E2E service	Delete E2E service	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link
Query progress status	Query progress status	Defined in Amsterdam	Delivered in Amsterdam	Interfaces Link

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, ...).

Name	Description	Version
Camunda	An open source platform for workflow and business process management	
ARIA	Tosca parser and TOSCA workflow engine	
JBOSS	An open source application server program	
MariaDB	One of the most popular open source database server	

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

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.

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.

Gaps identified	Impact
To fill out	To fill out

Known Defects and Issues

Provide a link toward the list of all known project bugs.

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
No issues found										

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).

Risk identified	Mitigation Plan	Contingency
1. Resources for functional requirement as was committed during discussion. 2.code is planned to be migrated from 1806 economy does not break anything major.(Low risk)	1. Need to decide on the on boarding by M2. 2. Att is ready to provide the required help to resolve issues.	1. M2 will rule out the functional requirement which don't keep up these needs. 2. Need to recheck on the s3p goals.

Resources

Fill out the Resources Committed to the Release centralized page.

Release Milestone

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

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 erecommended to provide these agreements and dates in this section.

It is not expected to have a detailed project plan.

Date	Project	Deliverable
To fill out	To fill out	To fill out

Documentation, Training

- Highlight the team contributions to the specific document related to he project (Config guide, installation guide...).
- Highlight the 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.

Release Key Facts

Fill out and provide a link toward the centralized Release Artifacts.