[ExtAPI] Beijing M1 Release Planning Template

ONAP External APIs expose the capabilities of ONAP. They allow ONAP to be viewed as a "black box" by providing an abstracted view of the ONAP capabilities.

They support that an external consumer of ONAP capabilities can be authenticated and authorized. They can also be used for connecting to systems where ONAP uses the capabilities of other systems.

Note 1: External API does not include all the B2B capabilities of exposure (e.g. partner management)

Note 2: The case where trusted providers of a service (e.g. operator owned transport, or cloud infrastructure) do not need to pass through External API. For example, External APIs between ONAP and BSS/OSS allow Service Providers to utilize the capabilities of ONAP while using their existing BSS/OSS environment minimizing customization

It is envisioned that from a Service Provider to Partner Provider interaction context (i.e. MEF Interlude), the ONAP External API will support the following types of interacts:

•Service Provider controls aspects of the Service within the Partner domain (on behalf of the Customer) by requesting changes to dynamic parameters as permitted by service policies.

•Service Provider queries state of the Service.

•Service Provider requests change to administrative state or permitted attributes of a Service.

•Service Provider request creation of connectivity between two Service Interfaces as permitted by established business arrangement.

•Service Provider request instantiation of functional service components as permitted by established business arrangement.

•Service Provider queries the Partner for detailed information related to Services provided by the Partner to the Service Provider.

•Service Provider receives Service specific event notifications (e.g., Service Problem Alerts) from the Partner.

•Service Provider receives Service specific performance information from the Partner.

•Service Provider request Service related test initiation and receive test results from the Partner.

The content of this template is expected to be fill out for M1 Release Planning Milestone.

Overview

Project Name	ExtAPI
Target Release Name	Beijing, Casablanca, Dublin
Project Lifecycle State	**Incubation**; Refer to ONAP Charter, section 3.3 Project Lifecycle for further information
Participating Company	AT&T, CenturyLink, China Mobile, China Telecom, Orange, PCCW Global, Turk Telekom, Verizon, Amdocs, Ciena, Huawei, Intel, Netcracker, ZTE, MEF

Scope

What is this release trying to address?

- Deliver points of interoperability between ONAP and External Systems
- Focus on ONAP External APIs to BSS/OSS (i.e., MEF Legato)
  - **Service Catalog**
    - Add notification for serviceCatalog API
      - Description:
        - Allow BSS catalog function to receive service catalog notification as serviceSpec status change or characteristic change (new value in an enum list for example). Could be interesting to track these serviceSpec update to update accordingly productSpec
      - Relevance:
      - Complexity: Easy
      - Prerequisites: It requires to have a notification from SDC because NBI will not pool AAI
      - Resources:
    - Improve ServiceCatalog API for service characteristics
      - Description:
        - Expose from NBI json (or other format) file describing the serviceSpec characteristic (same type of file we can retrieve on MEF Git Hub to describe an UNISpec for example)
        - Convert YAML in CSAR to ONAP wide consistent JSON schema for Service Characteristic Input parameters and provide across the ServiceCatalog API
      - Relevance:
      - Complexity: Easy
      - Prerequisites: Need SDC improvements
      - Resources:
  - **Service Ordering**
    - Add Service Request Status for serviceOrder API
      - Description:
        - Provide summary and detailed status for both Infrastructure requested VNFs and Customer requested VNFs.
        - Requestors that create transaction requests on the ONAP Platform, have a need to effectively manage the requests they create. The Requestor is the responsible owner for the transaction and thus, need the ability to have visibility into the status of their requests
        - BSS/OSS system will call the API providing one of the following options:
          1.Specific Service Request ID
          2.Many Service Requests based on Selection Criteria such as: Service Request Status and/or Time Frame where appropriate, (e.g., past day completions)
          ONAP will respond with the current status of the workflow for the selected request(s). The status can include not only a status summary of the requests but also the steps taken, start/stop time of the steps taken, notifications generated, and the remaining steps that need to be taken. Different workflows will have different numbers of steps, so this API will need to provide a name or identifier for any and all steps
      - Relevance:
      - Complexity: Easy
      - Prerequisites: Could implement in two phases, 1st phase implement status details that are currently available in the Service Orchestration (SO) API, while the 2nd phase would enhance SO and/or the SO API with any additional status information
      - Resources:
    - Add notification for serviceOrder API
      - Description:
        - Allow BSS (or any other) system to receive order/OrderItem update. BSS (or any other system) will not have to pool. We can allow several distinct notification (Nice to have: let subscriber specify notification contains). Minimum is to provide ServiceOrderStateChangeNotifications etc to HUB subscriber. After if we're able to get a notification from SO it will be perfect but initial requirement is only at external API northbound
        - Notifications related to ServiceOrder: - ServiceOrderCreationNotification - ServiceOrderAttributeValueChangeNotification - ServiceOrderStateChangeNotification - ServiceOrderInformationRequiredNotification - ServiceOrderRemoveNotification
      - Relevance:
      - Complexity: Easy
      - Prerequisites: Nothing for basic deliver…SO notifications to have high performance (without SO notification, NBI will pool SO as of today)
      - Resources:
    - Update ServiceOrder to to accommodate Service Chaining.
      - Description:
        - Enhance the Service Order API (TMF 641) to allow BSS/OSS the ability to flag services as part of a "group". Enhance the Service Inventory API (TMF 638) to allow BSS/OSS to retrieve Service Inventory by "group".
      - Relevance:
      - Complexity: Easy
      - Prerequisites: Service Orchestrator (SO) and its external API, A&AI and its API to SO and its external API
      - Resources:
    - Update ServiceOrder to manage Service modification request UC
      - Description:
        - This will allow BSS system to trigger service modification request. By modification we mean: characteristic value change, status change (other ?). Minimum could be to handle modification that

can be managed in SO with a Delete Service and then Add service (this is a change up to nbi but remove/add down to nbi). This is not service order modification butt service modification on existing service instance in the inventory (new service order with action change)
- Possibly related to CC VPN use case, explore other use cases
  - Relevance:
  - Complexity: Average to High depending on SO capability to handle service modification
  - Prerequisites: could require SO upgrade – Check if some use case can be handle by NBI only (triggering add /remove in SO)
  - Resources:
- Update ServiceOrder to provide user and system workload information for both Infrastructure requested VNFs and Customer requested VNFs
  - Description:
    - Requestors that create transaction requests on the ONAP Platform, have a need to effectively manage the requests they create. The Requestor is the responsible owner for the transaction and thus, need visibility into the volumes of requests created by a requestor, in order to better balance the workload among their labor resources.
    - •BSS/OSS system will call the API providing one of the following options:
      1.Specific Service Request ID
      2.Specific User ID of the Requestor
      3.All Users
      ONAP will respond with what activity is happening, who is doing it, and how old is it. Profile information can include Service Request Type, Creation/Completion Dates, and User ID of the Requestor. This information may require a dip into VID
  - Relevance:
  - Complexity: Average
  - Prerequisites: Could implement in two phases, 1st phase implement request profile information that is currently available in the Service Orchestration (SO) API, while the 2nd phase add additional status information from VID
  - Resources:
- **Service Inventory**
  - Add notification for serviceInventory API
    - Description: Allow BSS (or any other) system to receive service state update.
    - Relevance:
    - Complexity: Easy
    - Prerequisites: It requires to have a notification from AAI because NBI will not pool AAI
    - Resources:
  - Update Service Inventory to accommodate Service Chaining.
    - Description:
      - Enhance the Service Order API (TMF 641) to allow BSS/OSS the ability to flag services as part of a "group". Enhance the Service Inventory API (TMF 638) to allow BSS/OSS to retrieve Service Inventory by "group".
    - Relevance:
    - Complexity: Easy
    - Prerequisites: Service Orchestrator (SO) and its external API, A&AI and its API to SO and its external API
    - Resources:
  - Improve ServiceInventory API
    - Description:
      - As of now we retrieve very few information from AAI – Perhaps digging more in the instantiated VNF or VF could allow us to have more information as service state or serviceCharacteristic for example.
    - Relevance:
    - Complexity
    - Prerequisites: Need AAI expertise; Need enhancement to AAI UI to see more topology details across API
    - Resources:
- **Performance Management** (specification focus) (stretch goal: implementation)
  - Provide performance data for both Infrastructure VNFs and Customer VNFs
    - Description:
      - Capacity planning engineers must be able to determine the performance and quantities of VNFs running on the network for planning purposes. Performance Management data is required to develop Site Level Tenant and VNF & VM Forecasts that are the basis for Capacity Planning of the Network Infrastructure. May also be used in support of customer self service to provide customers with performance information about their specific services and VNFs
      - BSS/OSS system will call the API providing starting and ending dates and times. This API will be called either on-demand or on a regular timed basis. ONAP will respond with performance management data from DMaaP and Performance Data Store, may be trended for forecasting process, and include specific metrics & KPIs for all existing VNFs/VMs in service. May make use of TM Forum Performance Management API (TMF 628).
    - Relevance:
    - Complexity: Easy
    - Prerequisites: Could implement in two phases, 1st phase implement performance information that is currently available in the DMaaP API, while the 2nd phase enhance the DMaaP API with additional performance information. Requires additional micro-service development for collecting information from DMaaP or data store to store and aggregate the information
    - Resources:
- Service Topology (stretch goal) (specification focus)
- License Usage (stretch goal) (specification focus)
- **Integration**
  - Integrate External API/NBI within ONAP MSB
    - Description: May need to consider how External API agent functionality can be decoupled from MSB
    - Relevance:

- - - Complexity
      - Prerequisites:
      - Resources:
    - Build End-to-End Use Case
      - Description: Showcase External API from a complete Service Lifecycle perspective. Apply ONAP Use Cases.
      - Relevance:
      - Complexity
      - Prerequisites:
      - Resources:
  - Initial focus specification of ONAP External APIs supporting Inter-Provider (i.e., MEF Interlude)
    - Service Control (specification focus)
    - Service State (operational state) (specification focus)
    - Service Inventory / Details (specification focus)
  - Explore Role-based view of single APIs descriptors for both Legato and Interlude
  - Alignment with MEF Legato, MEF Interlude and TM Forum APIs
  - Definition of Use Cases, Interactions, and Information Model engaging service providers and BSS/OSS vendors
  - UML Models (Eclipse Papyrus) (with Modeling sub-committee) and API definition (JSON Swagger) for

    - License Usage
    - Service Modeling and Service Topology
    - Service Inventory
    - Service State Management
    - Service Quality Management
  - Define API Styles to be applied to External APIs (along with Micro-service Bus (MSB) and Modeling Project)
  - API development (in conjunction with specific ONAP component projects)

    - Well defined specifications for the NB APIs (e.g., JSON Swagger / OpenAPI).
    - ONAP implementation of these APIs
  - Architecture for External APIs
    - Identification and involvement of stakeholder ONAP projects
    - Describe key External API foundation functionalites
    - Work with Architecture and MSB projects
  - Document the role and requirements of External APIs in Model Driven ONAP
    - Work with Modeling project and sub-committee to explore a Model Driven approach: a cohesive way to have a shared view of information across ONAP external interfaces that can be used for or be input into a model driven process whereby the cost of delivering platform functionality is drastically reduced and the time to delivery is dramatically decreased.
  - Explore use of Model Driven Tool Chain to automatically generate APIs based on models with Modeling Project

Use Cases

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

The TSC identified the following Use cases for Release A:

TSC Use Case	VNFs identified in TSC Use case
(obselete)Use Case: VoLTE (vIMS + vEPC)	N/A
Use Case: Residential Broadband vCPE (Approved)	vBNG, vG_MUX, vG, vAAA, vDHCP, vDNS
Use Case: vFW/vDNS (Approved)	vFW, vPacketGenerator, vDataSink, vDNS, vLoadBalancer, all VPP based.
Use Case: VoLTE(approved)	vSBC, vPCSCF, vSPGW, vPCRF, VI/SCSCF, vTAS, VHSS, vMME

The External API developed by this project are applicable to the Services identified in the TSC E2E use cases.

Minimum Viable Product

- Documentation of User Stories; Use Cases and Interactions (e.g., UML); Information Models (e.g., UML); Data Models (e.g., JSON); Interface Profiles and Functional Definition;
- ONAP Component Mapping and Functional Analysis;
- Code contribution for External API Agent functionality.

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.

- Initial focus on ONAP External APIs to BSS/OSS (i.e., MEF Legato)
  - Service Catalog
  - Service Ordering (including Service Instantiation)
  - Service Inventory
  - Service Topology (stretch goal) (specification focus)

- License Usage (stretch goal) (specification focus)
- Initial focus specification of ONAP External APIs supporting Inter-Provider (i.e., MEF Interlude)
  - Service Control
  - Service State (operational state)
  - Service Inventory / Details

Epics

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
EXTAPI-593	Finalized Code Submission	⚡	Jun 23, 2021	Jun 23, 2021	Aug 26, 2021	Unassigned	None	═	OPEN	Unresolved
EXTAPI-587	Release Candidate 2 Integration and Test	⚡	Apr 09, 2021	Apr 23, 2021	Apr 22, 2021	Unassigned	None	═	CLOSED	Done
EXTAPI-582	Release Candidate 1 Integration and Test	⚡	Mar 17, 2021	Mar 23, 2021	Mar 25, 2021	Unassigned	None	═	CLOSED	Done
EXTAPI-573	Release Candidate 0 Integration and Test	⚡	Feb 23, 2021	Mar 15, 2021	Mar 11, 2021	Unassigned	None	═	CLOSED	Done
EXTAPI-552	Feature Freeze	⚡	Feb 01, 2021	Mar 08, 2021	Feb 25, 2021	Unassigned	None	═	CLOSED	Done
EXTAPI-542	Specification Freeze	⚡	Dec 23, 2020	Feb 15, 2021	Jan 21, 2021	Unassigned	None	═	CLOSED	Done
EXTAPI-537	Release Planning	⚡	Dec 12, 2020	Jan 25, 2021	Jan 11, 2021	Unassigned	None	═	CLOSED	Done
EXTAPI-533	TSC must for REQ for Release 8	⚡	Nov 18, 2020	Feb 19, 2021		Unassigned	None	═	CLOSED	Done
EXTAPI-529	Release Candidate 2 Integration and Test	⚡	Nov 06, 2020	Nov 24, 2020	Nov 19, 2020	Unassigned	None	═	CLOSED	Done
EXTAPI-525	Release Candidate 1 Integration and Test	⚡	Oct 22, 2020	Nov 02, 2020	Nov 05, 2020	Unassigned	None	═	CLOSED	Done
EXTAPI-515	Release Candidate 0 Integration and Test	⚡	Sep 18, 2020	Oct 15, 2020	Oct 08, 2020	Unassigned	None	═	CLOSED	Done
EXTAPI-491	Code Freeze	⚡	Aug 14, 2020	Sep 14, 2020	Sep 10, 2020	Unassigned	None	═	CLOSED	Done
EXTAPI-468	Functionality and API Freeze	⚡	Jul 21, 2020	Aug 31, 2020	Aug 06, 2020	Unassigned	None	═	CLOSED	Done
EXTAPI-465	Epic to cover the must have that effect External API Framework projec	⚡	Jul 13, 2020	Nov 18, 2020		Unassigned	None	═	CLOSED	Done
EXTAPI-454	Release Planning	⚡	Jun 20, 2020	Jul 28, 2020	Jul 09, 2020	Unassigned	None	═	CLOSED	Done
EXTAPI-439	Release Candidate 2 Integration and Test	⚡	Apr 21, 2020	May 21, 2020	May 14, 2020	Unassigned	None	═	CLOSED	Done
EXTAPI-436	Release Candidate 1 Integration and Test	⚡	Apr 21, 2020	May 04, 2020	Apr 30, 2020	Unassigned	None	═	CLOSED	Done
EXTAPI-403	Release Candidate 0 Integration and Test	⚡	Mar 01, 2020	Apr 07, 2020	Mar 26, 2020	Unassigned	None	═	CLOSED	Done

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
EXTAPI-387	Code Freeze	⚡	Jan 31, 2020	Mar 19, 2020	Mar 05, 2020	Unassigned	None	=	CLOSED	Done
EXTAPI-356	Functionality and API Freeze	⚡	Nov 22, 2019	Jan 23, 2020	Jan 21, 2020	Unassigned	None	=	CLOSED	Done

Showing 20 out of 53 issues

Stories

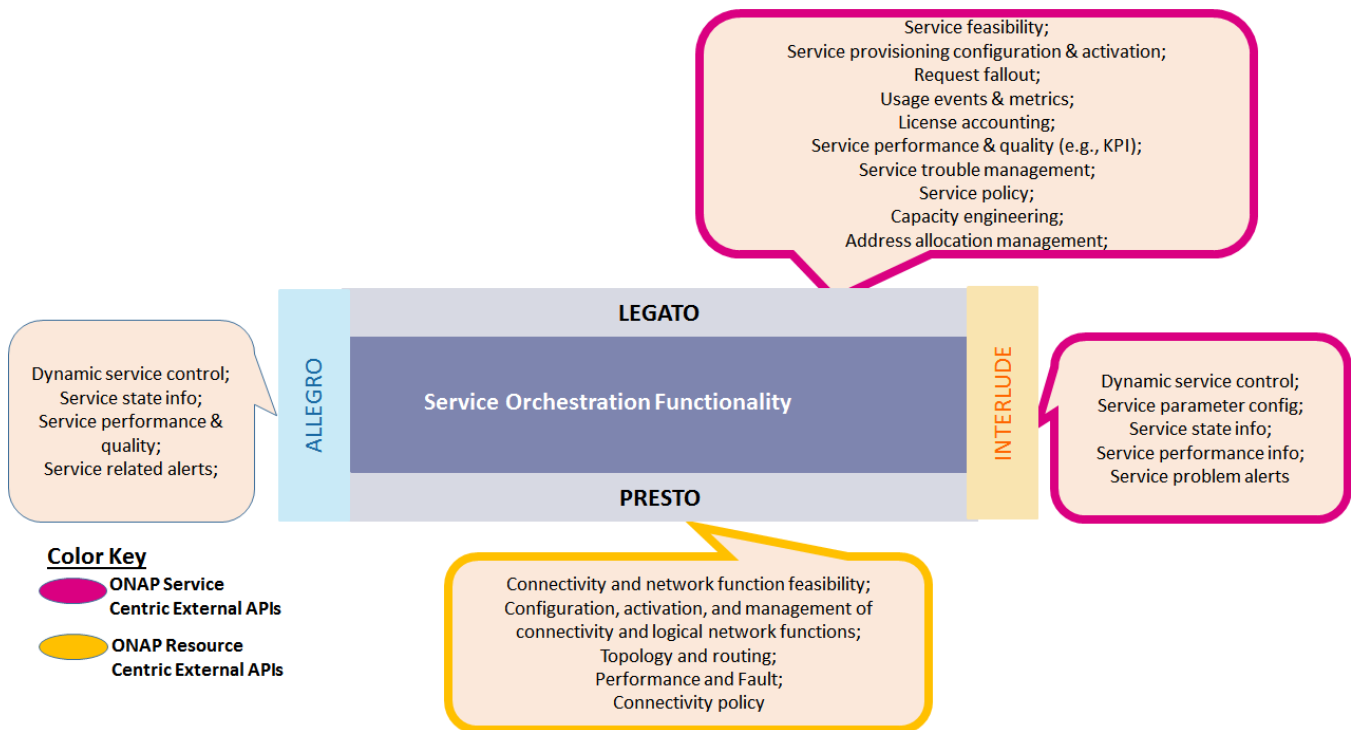| Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
EXTAPI-16	Analyze TM Forum APIs for beyond release 1	🔖	Aug 28, 2017	Aug 08, 2018		Unassigned	None	=	CLOSED	Done
EXTAPI-12	Track SDC and SO API Definition for Service Instantiation	🔖	Jul 03, 2017	Aug 08, 2018		Unassigned	None	^	CLOSED	Done
EXTAPI-9	Define UML Model Service Instantiation	🔖	Jul 03, 2017	Aug 08, 2018		Unassigned	None	^	CLOSED	Done
EXTAPI-8	Analyze Service Abstraction Information Models	🔖	Jul 03, 2017	Aug 08, 2018		Unassigned	None	=	CLOSED	Done
EXTAPI-7	Describe high level Use Cases for BSS/OSS Interactions with ONAP	🔖	Jul 03, 2017	Aug 08, 2018		Unassigned	None	^	CLOSED	Done
EXTAPI-6	Define Target Capabilities for External Interfaces	🔖	Jul 03, 2017	Aug 08, 2018		Unassigned	None	=	CLOSED	Done
EXTAPI-5	Define Functional Reference Architecture	🔖	Jul 03, 2017	Aug 08, 2018		Unassigned	None	^	CLOSED	Done
EXTAPI-4	Analyze existing SO Service Instantiation APIs	🔖	Jul 03, 2017	Aug 08, 2018		Unassigned	None	^	CLOSED	Done

8 issues

Longer term roadmap

Provide a clear and unambiguous ONAP service abstraction so that the BSS/OSS can exchange service requirements and service capabilities in a common and consistent fashion.

Provide a way to rapidly integrate new Services and Service Components into ONAP so that they can quickly introduce capabilities for their customers and within their infrastructure.

Enable management the entire lifecycle of Services within ONAP in a common way so that they can ensure orchestration, manageability and control of each Service in an easily integrateable and low cost way.

Model Driven approach: a cohesive way to have a shared view of information across ONAP external interfaces that can be used for or be input into a model driven process whereby the cost of delivering platform functionality is drastically reduced and the time to delivery is dramatically decreased.
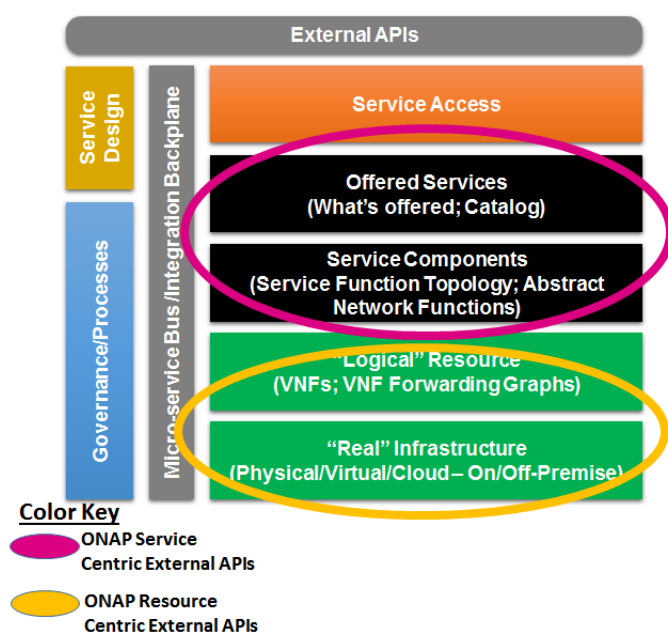
Service feasibility;
Service provisioning configuration & activation;
Request fallout;
Usage events & metrics;
License accounting;
Service performance & quality (e.g., KPI);
Service trouble management;
Service policy;
Capacity engineering;
Address allocation management;

LEGATO

ALLEGRO

Service Orchestration Functionality

INTERLUDE

PRESTO

Dynamic service control;
Service state info;
Service performance & quality;
Service related alerts;

Dynamic service control;
Service parameter config;
Service state info;
Service performance info;
Service problem alerts

Color Key

⬤ **ONAP Service Centric External APIs**

⬤ **ONAP Resource Centric External APIs**

Connectivity and network function feasibility;
Configuration, activation, and management of connectivity and logical network functions;
Topology and routing;
Performance and Fault;
Connectivity policy

It is envisioned that from a Service Provider to BSS/OSS interaction context (i.e. MEF Legato), the ONAP External API will support the following types of interacts:

- BSS/OSS retrieves Service Models
- BSS/OSS requests service feasibility determination.
- BSS/OSS requests reservations of capabilities related to a potential Service.
- BSS/OSS requests activation of Service.
- BSS/OSS receives Service activation tracking status updates.
- BSS/OSS retrieves Service Inventory
- BSS/OSS receives usage events due to a Customer initiating dynamic activity on their Service (e.g., increase in bandwidth).
- BSS/OSS receives a summary of Service quality and usage information.
- BSS/OSS receives Service state and fault event information
- BSS/OSS receives Service Activation Testing results.
- BSS/OSS receive capability information about the Service layer.
- BSS/OSS manages Licenses
- BSS/OSS receives License Usage information

It is envisioned that from a Service Provider to Partner Provider interaction context (i.e. MEF Interlude), the ONAP External API will support the following types of interacts:

- Service Provider controls aspects of the Service within the Partner domain (on behalf of the Customer) by requesting changes to dynamic parameters as permitted by service policies.
- Service Provider queries state of the Service.
- Service Provider requests change to administrative state or permitted attributes of a Service.
- Service Provider request creation of connectivity between two Service Interfaces as permitted by established business arrangement.
- Service Provider request instantiation of functional service components as permitted by established business arrangement.
- Service Provider queries the Partner for detailed information related to Services provided by the Partner to the Service Provider.
- Service Provider receives Service specific event notifications (e.g., Service Problem Alerts) from the Partner.
- Service Provider receives Service specific performance information from the Partner.
- Service Provider request Service related test initiation and receive test results from the Partner.

Governance/Processes
Policy/Rule/Constraints, SLA/OLA Definition, Asset/Capacity Management, Configuration/Change/Incident Management, Catalog Management

Service Design
Network function on-boarding
Service definition, including Service blueprints

Service Access
User/Tenant Management, Identity Management, RBAC Entitlement Control

Offered Services
Policy/Rule/Constraints, SLA/OLA Definition Asset/Capacity Management, Configuration/Change/Incident Management

Services Components
Decomposition/Instantiation/Assurance/Remediation
Service Level Optimization, QoS Optimization, Policy Enforcement,

Logical Resource
Resource Matchmaking/Optimization
Resource Reservations/Allocations, Installation/Configuration

Real Infrastructure
Provider Registration, Infrastructure Discovery/Monitoring
Capacity Optimization

Release Deliverables

Deliverable Name	Deliverable Description
Documentation	Documentation of User Stories; Use Cases and Interactions (e.g., UML); Information Models (e.g., UML); Data Models (e.g., JSON); Interface Profiles and Functional Definition;ONAP Component Mapping and Functional Analysis
External API	JSON Swagger / OpenAPI for the External Interface
External API Agent Software	Code contribution for External API Agent functionality

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 are consolidate in a single centralized place. Edit the Release Components name for your project in the centralized page.

1. External API Agent:
    a. Core Agent Functionality
    b. Service Catalog API
    c. Service Ordering API
    d. Service Inventory API

ONAP Dependencies

List the other ONAP projects your depends on.

Dependent on APIs from SDC, SO, and AAI

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.

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.
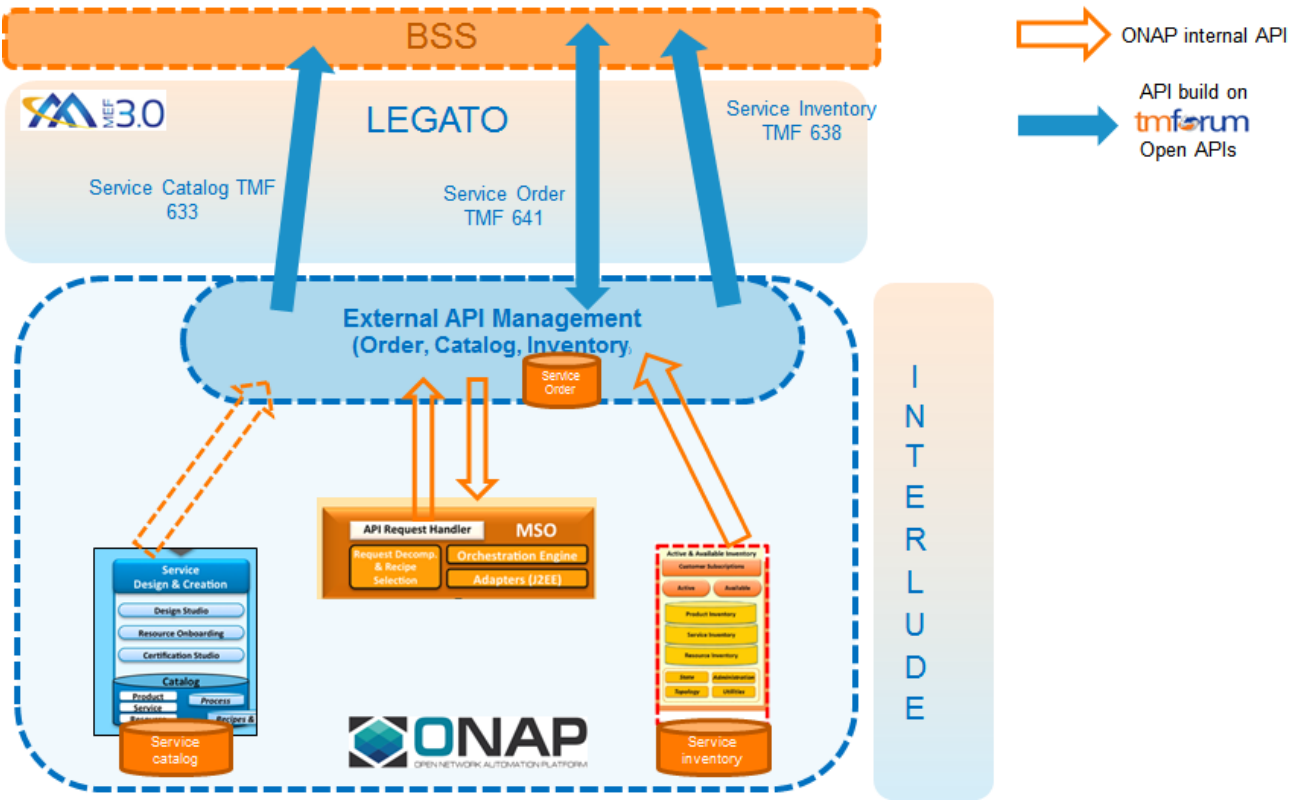
Platform Maturity

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	0		• 0 -- none • 1 – baseline performance criteria identified and measured • 2 & 3 – performance improvement plans created & implemented
Stability	0	0		• 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	0	0		• 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		• 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	0		• 0 – no ability to scale • 1 – single site horizontal scaling • 2 – geographic scaling • 3 – scaling across multiple ONAP instances

Managea bility	0	0		• 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	0		• 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: Catalog API	Exposes Service Catalog	TBD	TBD	https://wiki.onap.org/display/DW/SDC+API
SO: Service Instantiation API	Requests for Service Instantiation	TBD	TBD	
AAI: Service Inventory API	Query for Service Inventory	TBD	TBD	

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)
ExtAPI: Service Catalog	External Service Catalog API	TBD	TBD	TBD
ExtAPI: Service Ordering	External Service Ordering API	TBD	TBD	TBD
ExtAPI: Service Inventory	External Service Inventory API	TBD	TBD	TBD

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
Springboot	Java Platform Framework	TBD
MariaDB	MySQL open fork	

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.

Potential Test Cases for External API include:

- BSS/OSS retrieves Service Models
- BSS/OSS orders a new Service.
- BSS/OSS subscribes to Service order tracking status notifications.
- BSS/OSS retrieves Service Inventory

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
TBD	TBD

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
EXTAPI-602	multiple instantiations of same service fails	🔴	Mar 21, 2022	Apr 06, 2022		Unassigned	None	=	IN PROGRESS	Unresolved
EXTAPI-599	API Redoc Documentation	🔴	Oct 04, 2021	Jun 02, 2022		Unassigned	None	=	CLOSED	Done
EXTAPI-535	Error in NBI serviceSpecification API	🔴	Nov 30, 2020	Feb 22, 2021		Unassigned	None	=	CLOSED	Done
EXTAPI-512	remove offending yaml from csar used in Karate testing	🔴	Sep 11, 2020	Sep 14, 2020		Unassigned	None	^	CLOSED	Done
EXTAPI-510	1 NBI pod has no limit	🔴	Sep 10, 2020	Oct 15, 2020		Unassigned	None	^	CLOSED	Duplicate
EXTAPI-509	NBI has root pods	🔴	Sep 09, 2020	Sep 10, 2020		Unassigned	None	^	CLOSED	Done
EXTAPI-428	Fix logger in SoClient	🔴	Apr 09, 2020	Apr 22, 2020		Unassigned	None	=	CLOSED	Done
EXTAPI-424	Public HTTP port open	🔴	Apr 03, 2020	Apr 07, 2020		Unassigned	None	^	CLOSED	Done
EXTAPI-423	Check for CST template is case sensitive	🔴	Mar 31, 2020	Jun 30, 2020		Unassigned	None	=	CLOSED	Done
EXTAPI-366	Fix NBI builds	🔴	Nov 26, 2019	Dec 01, 2019		Unassigned	None	=	CLOSED	Done
EXTAPI-354	NBI ServiceSpecification GET fail	🔴	Nov 11, 2019	Nov 13, 2019		Unassigned	None	^	CLOSED	Not a Bug
EXTAPI-307	nbi pod not started after a fresh master installation	🔴	Sep 02, 2019	Sep 04, 2019		Unassigned	None	^	CLOSED	Cannot Reproduce
EXTAPI-305	No Need for "ReadWriteMany" access on storage when deploying on Kubernetes	🔴	Aug 29, 2019	Sep 04, 2019		Unassigned	None	^	CLOSED	Done
EXTAPI-287	NBI to SDC connectivity health checks fail	🔴	Aug 08, 2019	Aug 12, 2019		Unassigned	None	=	CLOSED	Done
EXTAPI-251	Add external var for tmp directory	🔴	May 24, 2019	Feb 15, 2021		Unassigned	None	=	CLOSED	Done
EXTAPI-249	Change to oom dockers causing permissions failing when tosca parsing	🔴	May 22, 2019	Aug 12, 2023		Unassigned	None	=	CLOSED	Done
EXTAPI-248	ExtAPI should not be polling SDC-DISTR-NOTIF-TOPIC-AUTO without authenticating	🔴	May 22, 2019	Aug 12, 2023		Unassigned	None	=	CLOSED	Done
EXTAPI-247	Fix API version in helm chart	🔴	May 16, 2019	May 17, 2019		Unassigned	None	^	CLOSED	Done

EXTAP I-246	Fix broken sonar code coverage	🔴	May 16, 2019	May 17, 2019	Unassigned	None	═	CLOSED	Done
EXTAP I-241	Adapt NBI test	🔴	May 14, 2019	May 16, 2019	Unassigned	None	═	CLOSED	Done

Showing 20 out of 70 issues

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 Plan
TBD	TBD	TBD

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
TBD	TBD	TBD

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.