Network as a Service (NaaS) API Component Suite Profile

TMF 909A

Version 1.5

October 2018

Notice

Copyright © TeleManagement Forum 2018. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](http://www.tmforum.org/IPRPolicy/11525/home.html), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,

East Tower – 10th Floor,

Morristown, NJ  07960 USA

Tel No.  +1 973 944 5100

Fax No.  +1 973 944 5110

TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org/)

Table of Contents

[Notice 2](#_Toc526935351)

[Table of Contents 3](#_Toc526935352)

[List of Figures 5](#_Toc526935353)

[List of Tables 6](#_Toc526935354)

[Executive Summary 7](#_Toc526935355)

[1. Introduction 8](#_Toc526935356)

[2. Requirements and Use Cases 10](#_Toc526935357)

[2.1. Requirements 10](#_Toc526935359)

[2.1.1. Order to Activate 11](#_Toc526935360)

[2.1.2. Prospect to Order 11](#_Toc526935361)

[2.1.3. Trouble to Resolution 12](#_Toc526935362)

[2.1.4. Usage to Cash 12](#_Toc526935363)

[2.1.5. Service Catalog (Design Time) 12](#_Toc526935364)

[2.2. Use Cases 12](#_Toc526935365)

[2.2.1. Order to Activate Use Cases 12](#_Toc526935366)

[2.2.2. Prospect to Order Use Cases 13](#_Toc526935367)

[2.2.3. Trouble to Resolution Use Cases 13](#_Toc526935368)

[2.2.4. Usage to Cash Use Cases 13](#_Toc526935369)

[2.2.5. Service Catalog (Design Time) 13](#_Toc526935370)

[3. Functions 14](#_Toc526935371)

[3.1. Create a new service 14](#_Toc526935372)

[3.2. Monitor long running service creation process 14](#_Toc526935373)

[3.3. Modify inflight service creation 14](#_Toc526935374)

[3.4. Cancel inflight service creation 14](#_Toc526935375)

[3.5. Modify existing service 14](#_Toc526935376)

[3.6. Suspend service / restore service 15](#_Toc526935377)

[3.7. Remove service 15](#_Toc526935378)

[3.8. Check feasibility of a service 15](#_Toc526935379)

[3.9. Design, Assign and Reserve a service 15](#_Toc526935380)

[3.10. Create an inactive service 16](#_Toc526935381)

[3.11. Activate an inactive service 16](#_Toc526935382)

[3.12. Retrieve service details 16](#_Toc526935383)

[3.13. Location based check (service qualification request) 17](#_Toc526935384)

[3.14. Reactive Trouble to Resolve - Validate and Test Service 17](#_Toc526935385)

[3.15. Fetch Service Test Result/Status 17](#_Toc526935386)

[3.16. Modify Service Test 17](#_Toc526935387)

[3.17. Notifiable event occurs 17](#_Toc526935388)

[3.18. Create Service Problem (Creation of ticket proactively by domain) 18](#_Toc526935389)

[3.19. Obtain Existing Service Problem Details 18](#_Toc526935390)

[3.20. Modify/Update Service Problem Details 18](#_Toc526935391)

[3.21. Delete Service Problem 18](#_Toc526935392)

[3.22. Proactive Trouble to Report 19](#_Toc526935393)

[3.23. Retrieval of Usage per customer/on request](#_Toc526935394)[1](#_Toc526935394) [19](#_Toc526935394)

[3.24. Threshold Notification for Service 19](#_Toc526935395)

[3.25. Retrieve Service Specification 19](#_Toc526935396)

[3.26. Create Service Specification 20](#_Toc526935397)

[3.27. Update Service Specification 20](#_Toc526935398)

[4. Component Capabilities, Flows and Sequence Diagrams 21](#_Toc526935399)

[4.1. Requirements to Functions 21](#_Toc526935400)

[4.2. Sequence Diagrams 22](#_Toc526935401)

[4.2.1. O2A CreateService-Atomic-Async 23](#_Toc526935402)

[4.2.2. O2A Modify Atomic Service 23](#_Toc526935403)

[4.2.3. Trouble to Resolve – Reactive Report Service Problem - Sync 24](#_Toc526935404)

[4.2.4. Trouble to Resolve – Reactive Update Service Problem – Sync 25](#_Toc526935405)

[4.2.5. Trouble to Resolve – Proactive Obtain Service Problem Details – Async 27](#_Toc526935406)

[4.2.6. U2C - Retrieval of Usage Record 28](#_Toc526935407)

[5. Functions and API Mappings 29](#_Toc526935408)

[5.1. Notification Tables 31](#_Toc526935409)

[6. Component API Specification 33](#_Toc526935410)

[7. Administrative Appendix 34](#_Toc526935411)

[7.2. Appendix A: References 34](#_Toc526935412)

[7.3. Document History 34](#_Toc526935413)

[7.3.1. Version History 34](#_Toc526935414)

[7.4. Acknowledgments 34](#_Toc526935415)

List of Figures

[Figure 1 Operational Domain Management Functions and Capabilities 9](#_Toc526938662)

[Figure 2 NaaS API component suite classified in 4 main lifecycle areas 11](#_Toc526938663)

[Figure 3 O2A Create Service Atomic Async Sequence diagram 23](D:\\Users\\d877911\\Documents\\TMF APIs\\TMF 909 - API_Suite_component_for_NaaS v1_5.docx" \l "_Toc526938664)

[Figure 4 O2A Modify Service Atomic Sequence diagram 24](D:\\Users\\d877911\\Documents\\TMF APIs\\TMF 909 - API_Suite_component_for_NaaS v1_5.docx" \l "_Toc526938665)

[Figure 5 T2R - Reactive Report Service Problem - Sync 25](D:\\Users\\d877911\\Documents\\TMF APIs\\TMF 909 - API_Suite_component_for_NaaS v1_5.docx" \l "_Toc526938666)

[Figure 6 T2R - Reactive Update Service Problem - Sync 26](#_Toc526938667)

[Figure 7 T2R – Proactive Obtain Service Problem Details – Async 27](#_Toc526938668)

[Figure 8 U2C – Retrieval of usage - Sync 28](#_Toc526938669)

List of Tables

[Table 1 NaaS Requirements, Functions and associated APIs 22](#_Toc516587337)

[Table 2 NaaS API notifications 32](#_Toc516587338)

Executive Summary

The scope of this document is to define the API Component Suite in support of a set of Operational Domains exposing and managing “Network” Services. These services are referred to as Network as a Service (NaaS) and include all services offered from a service providers including connectivity, end points, media services, etc. An Operational Domain supports a set of functions and capabilities responsible for the complete lifecycle of services and resources within its domain, including exposure of services supported by SLAs, interfacing via standard TMF APIs with the BSS systems and other operational domains. The API set supports the control plane for the services and not the delivery of the service.

The first aim and objectives of the NaaS component suite is to set the API control plane contract between OSS/BSS (IT systems) and operational domains as well as between operational domains to operational domains.

Whilst the API standards in this document define the framework and common header information for exposed services, at this time the service details are to be defined by the individual operational domains.

Parallel work will focus on a Standard data model from a set of CSP top service priority. The result can be referenced and feedback added in this document in future releases.

# Introduction

This component suite covers the operations required to be exposed in order to provide the functionality required by Operational Domains interworking with OSS/BSS applications and/or other domains from one service provider or from 3rd parties. One of the key requirements is the re-use of an API functionality rather than using a large set of specific APIs. This also simplifies the number of APIs needed and reduce the initial and maintenance costs as suppliers typically charge on a per API basis.

The NaaS API component suite implements a subset of the proposed scenarios in HIP Management platform **[TMF070B](https://www.tmforum.org/resources/specification/tmf070b-advanced-platform-deployment-blueprints-r17-5-0/)** [Advanced Platform Deployment Blueprints R17.5.1](https://www.tmforum.org/resources/specification/tmf070b-advanced-platform-deployment-blueprints-r17-5-0/) section 8

* [8.2. Product Pre-Ordering](https://projects.tmforum.org/wiki/display/PUB/TMF070B+Advanced+Platform+Deployment+Blueprints+R17.5.1?_ga=2.107166212.723378726.1524824266-753906893.1507067474" \l "TMF070BAdvancedPlatformDeploymentBlueprintsR17.5.1-ProductPre-Ordering)
* [8.3. Product Ordering and Activation and Configuration (A&C)](https://projects.tmforum.org/wiki/display/PUB/TMF070B+Advanced+Platform+Deployment+Blueprints+R17.5.1?_ga=2.107166212.723378726.1524824266-753906893.1507067474" \l "TMF070BAdvancedPlatformDeploymentBlueprintsR17.5.1-ProductOrderingandActivationandConfiguration(A&C))
* [8.4. SLA Violation Reporting and Dynamic Product Instance Inventory](https://projects.tmforum.org/wiki/display/PUB/TMF070B+Advanced+Platform+Deployment+Blueprints+R17.5.1?_ga=2.107166212.723378726.1524824266-753906893.1507067474" \l "TMF070BAdvancedPlatformDeploymentBlueprintsR17.5.1-SLAViolationReportingandDynamicProductInstanceInventory)
* [8.5. SLA Violation Correction](https://projects.tmforum.org/wiki/display/PUB/TMF070B+Advanced+Platform+Deployment+Blueprints+R17.5.1?_ga=2.107166212.723378726.1524824266-753906893.1507067474" \l "TMF070BAdvancedPlatformDeploymentBlueprintsR17.5.1-SLAViolationCorrection)

The APIs:

* Will not be defining customer products or offers
* Will be implemented using the Hypermedia APIs with dynamic payloads that make the services self-descriptive wherever possible
* Will be operating on customer facing services or private services, the latter services only exposed towards other operational domains
* Will perform lifecycle actions to the expose network services (generally CFS)
* Communicate what the BSS needs back from the Network Services
* Will not be dealing with CRM, or Billing and will not have to care about a network function being virtualized or physical (That’s the domain responsibility) unless it is mandatory (such as a service end point).

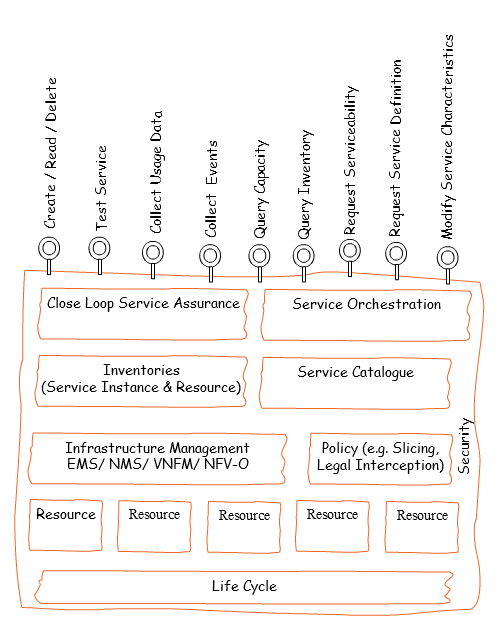


Figure 1 Operational Domain Management Functions and Capabilities

# Requirements and Use Cases

This section describes the functions that are needed to deliver Network as a Service.

While “service” can be the service specification or service instance, within this document the use of unqualified service will always mean service instance. Service specification will also be fully qualified.

A network service typically has a long life span and can undergo numerous changes throughout its life. A service may be created, fail, be restored, be made inactive (maybe for non-payment), be made active again, fail, and be restored, etc. throughout its life. As such this standard does not try to provide all the possible ordering of actions against a service. The material is all the atomic actions as the ordering of the various actions are generally independent or the preconditions are documented. Similar, notifications are only explicitly mentioned in sections 3 and 4 when they originate autonomously from the operational domain. Notifications that are caused by API calls are not covered in section 3 and 4 as they are generally redundant (e.g., a POST call to create a service will cause a serviceCreationNotification and/or a monitor\*ChangeNotification) as they simply document what the API call caused. Similar the link to a returned Monitor for long running transactions is generally not mentioned as this pattern is true for all long running transactions.

There is some inherent flexibility in the TMF Open API design to support different business models. One area where this flexibility is clear is whether the network exposes single services and service order processing is fully a BSS responsibility. Or the network may accept the full or partial service order and service order processing is a network responsibility. The decision would drive the usage of TMF 641 and/or 640. For this version of the NaaS API component suite, the focus will be on TMF 640 activation and configuration. A separate profile may be created in the future in support of the TMF 641 Service Ordering.

## Requirements

The NaaS API component suite includes APIs classified in 4 main lifecycle areas:

* Prospect to Order (P2O)
* Order to Activate (O2A)
* Trouble to Resolution (T2R)
* Usage to Payment (also referred to as U2C - Usage to Cash)
* The output of service design process (part of concept to market) are the service specifications. The existence of the service specifications in the TMF Service Catalog (TMF633) is a pre-condition for this standard

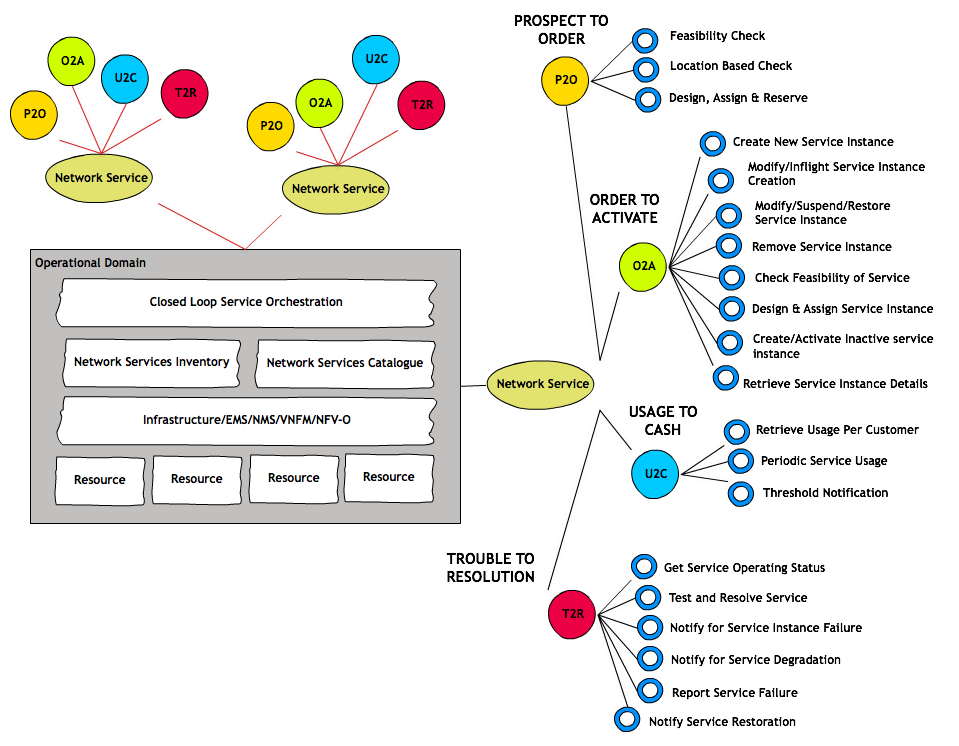


Figure 2 NaaS API component suite classified in 4 main lifecycle areas

**Note as there is overlap between Prospect to Order and Order to Activate within the TMF standards, the Prospect to Order follows Order to Activate**. This is simply to improve readability of this document.

### Order to Activate

The order to activate functional area includes all activities to support the business/customer layer in delivering ordered services. This includes full lifecycle support for all network services.

Order to activate also includes any changes (inflight and post activation) along with life cycle management of the service.

### Prospect to Order

The prospect to order process includes all activities supporting the business/customer layer in evaluating what products can be offered to a customer. This ranges from simple geographically based checks to see if a service is offered to fully designing and assigning all aspects of a service and reserving the resources to provide the service.

There are three main scenarios:

* Location based check. The check is fully automatic.
* Feasibility check. The complexity of this check ranges from fully automatic to extensive manual design. The results of a feasibility check are saved.
* Design, assign, and reserve. The service is fully designed and assigned. The underlying network resources are reserved. The results are saved.

### Trouble to Resolution

The Trouble to Resolution functional area provides the business/customer layer with the operating status of service instances and the ability to initiate testing of service instances. This includes full lifecycle support for all network services to be supported for service assurance.

Operational Domain Management functions will not expose resource level functions (such as Performance Management and Fault Management) to the BSS domain or consumer systems as these functions will remain internal to the operational domain itself and will work primarily on information abstracted from the network layer. Operational Domains will not be responsible for processing Trouble ticket nor SLA but they can send status to these applications when thresholds are crossed.

### Usage to Cash

The usage to payment includes all activities supporting the business/customer layer in evaluating Usage Consumption for subscribed communication products. It also allows customers or users to be informed on usage and remaining credits available for the subscribed service.

There are three scenarios that need to be supported:

* + - Retrieval of Usage per customer/on request: - This can be a synchronous call to get the information and we can also anticipate that few systems might require an asynchronous behaviour to provide the usage consumption information. When such need arises, notification can be used to pass the usage information.
    - Scheduled Service Usage Record for a service.
    - Threshold Notification for Service Usage to indicate that a customer has already consumed X% of the allocated quota.

### Service Catalog (Design Time)

During runtime, the service catalog is only used to retrieve service specifications. During the design time, we need to retrieve, create and update the service specification. The concept to market activities of service design are not in the scope of this standard.

## Use Cases

This section provides examples of how the most common business functions are implemented using the API standard.

### Order to Activate Use Cases

The key order to activate business functions are:

* Create new service
* Monitor long running service creation process
* Modify inflight service creation
* Cancel inflight service creation
* Modify existing service
* Suspend service / restore service
* Remove service
* Check feasibility of a service
* Design and Assign a service and reserve resources
* Create an inactive service
* Activate an inactive service
* Retrieve service details

### Prospect to Order Use Cases

The key prospect order to activate business functions are:

* Location based check
* Feasibility check
* Design, assign, and reserve

The latter two are used during the prospect to order and order to activate business processes. For example, a feasibility check can be supporting the business processes such as configure, price and quote and used as part of evaluating alternative options for an order. From a network perspective, the process is the same. As such the Feasibility Check and the Design, Assign and Reserve process are identical to the discussion in the O2A section.

### Trouble to Resolution Use Cases

Trouble to Resolution area can be sub-classified broadly into two main categories:

* Reactive T2R
* Proactive T2R

Reactive T2R process is when a consumer system reports a problem. Proactive T2R processes is notifying consumer system(s) when operational domain detect service problems.

### Usage to Cash Use Cases

The key usage to cash business functions are:

* Retrieval of Usage per customer/on request for a Service
* Periodic Service Usage Record for a Service
* Threshold Notification for a Service

### Service Catalog (Design Time)

The key Design Time business functions are:

* Retrieve Service Specification (design and run time)
* Create Service Specification\*
* Update Service Specification\*

\* Restricted to the operational domain providing the service.

# Functions

## Create a new service

**Pre-condition:** service specification has been defined in the service catalogue and a service instance does not exist

**Post condition:**

Service instance has been created and is captured in the service instance inventory of the domain.

Depending on the service and sometimes the details of a particular service creation request, the success or failure of the request will be immediately returned. In other cases, the immediate response will simply be that the request is processing. The latter will always occur if the requested service has a manual design step.

**Error:** If the resources are not available an error will be returned.

## Monitor long running service creation process

**Pre-condition:** Client wants to monitor the service creation. A GET may be issued to the monitor for the service to get the current status.

Alternatively, the client listens for the serviceCreationNotification, the monitorValueChangeNotification, and/or monitorStateChangeNotification events.

**PostCondition:**  Client is passively notified when the service creation/modification process has completed if it used the event notification pattern

**Error:** None identified

## Modify inflight service creation

**Pre-condition:** service instance creation is in process

**Post condition:**

Service creation process has been modified and ultimately captured in the service instance inventory of the domain.

**Error:** domain can determine that the request is not feasible and return an error

## Cancel inflight service creation

**Pre-condition:** service instance is in process

**Post condition:**

Service creation process has been canceled and the service instance inventory is unchanged

**Error:** domain can determine that the request is not feasible and return an error

## Modify existing service

**Pre-condition:** service specification has been defined in the service catalogue and a service instance exists

**Post condition:**

Service instance has been modified and updates captured in the service instance inventory of the domain.

Depending on the service and sometimes the details of a particular service creation request, the success or failure of the request will be immediately returned. In other cases, the immediate response will simply be that the request is processing. The latter will always occur if the requested service has a manual design step.

**Error:** If the requested change is not feasible e.g. resources are not available an error will be returned and service instance will be unchanged

## Suspend service / restore service

**Pre-condition:** service specification has been defined in the service catalogue and a service instance exists in the active/suspended state

**Post condition:**

Service instance has been modified to new state and captured in the service instance inventory of the domain.

**Error:** If the requested change is not feasible e.g. resources are not available an error will be returned and service instance will be unchanged

## Remove service

**Pre-condition:** service specification has been defined in the service catalogue and a service instance exists

**Post condition:**

Service instance has been removed and longer exist in the service instance inventory of the domain.

**Error:** If the service removal fails e.g. there is a dependent service an error will be returned and service instance will be unchanged

## Check feasibility of a service

**Pre-condition:** service specification has been defined in the service catalogue and a service instance

**Post condition:**

Required design manual and/or automatic has been done and informational record is created in the domain. Informational record captures the design and tentative resources needed to provide the service.

Depending on the service and sometimes the details of a particular service creation request, the success or failure of the request will be immediately returned. In other cases, the immediate response will simply be that the request is processing. The latter will always occur if the requested service has a manual design step.

**Error:** If the service qualification could not be done e.g. the service spec does not exist

Note: The “Check feasibility of a service” capability is used by both Prospect to Order and Order to Activate business functions. It is the same capability for both business functions.

## Design, Assign and Reserve a service

**Pre-condition:** service specification has been defined in the service catalogue and a service instance may or may not exist

**Post condition:**

Required design manual and/or automatic has been done and service instance is created in the service inventory of the domain. The state of the service instance will be reserved.

Depending on the service and sometimes the details of a particular service creation request, the success or failure of the request will be immediately returned. In other cases, the immediate response will simply be that the request is processing. The latter will always occur if the requested service has a manual design step.

**Error:** If the requested service cannot be reserved e.g. resources are not available an error will be returned. If a service instance existed already it will be restored to the initial state.

Note: The “Design, Assign and Reserve a service” capability is used by both Prospect to Order and Order to Activate business functions. It is the same capability for both business functions.

## Create an inactive service

**Pre-condition:** service specification has been defined in the service catalogue and a service instance may or may not exist

**Post condition:**

Required design manual and/or automatic has been done and service instance is created in the service inventory of the domain. The state of the service instance will be inactive.

Depending on the service and sometimes the details of a particular service creation request, the success or failure of the request will be immediately returned. In other cases, the immediate response will simply be that the request is processing. The latter will always occur if the requested service has a manual design step.

**Error:** If the requested service cannot be reserved e.g. resources are not available an error will be returned. If a service instance existed already it will be restored to the initial state.

## Activate an inactive service

**Pre-condition:** service specification has been defined in the service catalogue and a service instance exists in an inactive state

**Post condition:**

The state of the service instance has been updated to be active.

**Error:** If the state change is not possible an error will be returned. The service instance will remain in an inactive state.

## Retrieve service details

**Pre-condition:** Service does exist

**PostCondition:**  Attributes of the service are returned

**Error:** Request for non-existing service or attributes will return an error

## Location based check (service qualification request)

**Pre-condition:** A service specification exists

**PostCondition:**  Availability of requested service is confirmed.

**Error:** if the service qualification could not be done e.g. the service spec does not exist

## Reactive Trouble to Resolve - Validate and Test Service

**Pre-condition:** service specification has been defined in the service catalogue including a service test function and a service instance exists.

**Post condition:**

Test is completed and results are returned. If self-healing was requested, the success or failure of the self-healing is reported.

Depending on the service and sometimes the details of a particular service test request, the success or failure of the request will be immediately returned. In other cases, the immediate response will simply be that the request is processing. If the test is long running, the requester is notified when the test has completed.

**Error:** If the service instance does not exist or the test could not be performed an error will be returned. The service instance status will be returned even if the test could not be performed.

## Fetch Service Test Result/Status

**Pre-condition:** A service test has been requested and has indicated the response will be asynchronous. The test is either running or completed

**Post condition:**

If the test is completed, the full test results will be returned. If the test is still running, an indication of test in progress will be returned.

**Error:** If the test id is not valid, an error will be returned.

## Modify Service Test

**Pre-condition:** A service test has been requested and has indicated the response will be asynchronous. The test is either running or completed.

**Post condition:**

If the test is still running and the test can be modified, a success message will be returned. If not an message will be returned indicating the test cannot be modified. If the test is completed, the full test results will be returned.

**Error:** If the test id is not valid, an error will be returned.

## Notifiable event occurs

**Pre-condition:** A condition occurs the triggers a notification; such as:

* Identification of a notifiable SLA violation
* Identification of service affecting performance issue
* Identification of a notifiable service degradation or any major outage
* Resolution or restoration of service

**Post condition:**

A notification is generated providing details on the problem / status change and sent to the registered call back.

**Error:** If the callback id an error will be logged.

## Create Service Problem (Creation of ticket proactively by domain)

**Pre-condition:** Service instance is determined to have a service problem. The root cause of the service problem could be resource failures, degradation of service and/or SLA violations.

**Post condition:**

An event was triggered documenting service problem and related information on the problem. The event message was sent to each callback URL that registered for information on service problems for this service specification type. The trouble ticket system could register for this notification to create a trouble ticket.

**Error:** If callback URL are not available, the process will retry until a timeout is reached. The inability to deliver a message will be logged.

## Obtain Existing Service Problem Details

**Pre-condition:** Service problem exists and a service problem ID has been created.

**Post condition:**

The current information on the service problem will be returned.

**Error:** If service problem id is not found, an error will be returned.

## Modify/Update Service Problem Details

**Pre-condition:** Service problem exists and a service problem ID has been created.

**Post condition:**

The service problem information will be updated.

**Error:** If service problem id is not found, an error will be returned.

## Delete Service Problem

**Pre-condition:** Service problem exists and a service problem ID has been created and the issue has been resolved.

**Post condition:**

The service problem information will be deleted.

**Error:** If service problem id is not found, an error will be returned.

## Proactive Trouble to Report

**Pre-condition:** Service instance is determined to have a service problem. The root cause of the service problem could be resource failures, degradation of service and/or SLA violations.

**Post condition:**

An event was triggered documenting service problem and related information on the problem. The event message was sent to each callback URL that registered for information on service problems for this service specification type.

**Error:** If callback URL are not available, the process will retry until a timeout is reached. The inability to deliver a message will be logged.

## Retrieval of Usage per customer/on request1

**Pre-condition:** Service instance exists and is generating usage

**Post condition:**

The usage information for either the time period requested or the current bucket value is returned.

**Error:** If service instance does not exist or the requested usage type does not exist an error will be returned.

## Threshold Notification for Service

**Pre-condition:** Service instance exists and is generating usage. The service configuration includes a notification a predetermined usage values.

**Post condition:**

An event is generated and sent to the requesting consumer with the information.

**Error:** If callback URL are not available, the process will retry until a timeout is reached. The inability to deliver a message will be logged.

## Retrieve Service Specification

**Pre-condition:** Service specification exists in catalogue.

**Post condition:**

The details on the service specification are returned.

**Error:** If specified service specification is not found, an error will be returned.

## Create Service Specification

**Pre-condition:** Service specification must not exists in catalogue.

**Post condition:**

The service specification will be loaded into the service catalogue and a success message will be returned.

**Error:** If specified service specification already exists, an error will be returned.

## Update Service Specification

**Pre-condition:** Service specification exists in catalogue.

**Post condition:**

The service specification will be updated in the service catalogue and a success message will be returned.

**Error:** If specified service specification is not found, an error will be returned

# Component Capabilities, Flows and Sequence Diagrams

## Requirements to Functions

|  |  |
| --- | --- |
| Requirement | Function |
| Order to Activate | * Create new service * Monitor long running service creation process * Modify inflight service creation * Cancel inflight service creation * Modify existing service * Suspend service / restore service * Remove service * Check feasibility of a service * Design and Assign a service and reserveresources * Create an inactive service * Activate an inactive service * Retrieve service details |
| Prospect to Order | * Location based check * Feasibility check * Design, assign, and reserve |
| Trouble to Resolution | * Obtain existing Service Problem details * Create Service Problem * Modify / Update Service problem * Delete service problem that is no longer valid and was raised by consumer system earlier * Operational domain requesting additional information about ticket * Validate and Test service for reported problems by creating service test * Fetch Service test result on completion * Modify service test parameters * Creation of ticket proactively by domain * Identification of a notifiable SLA violation * Identification of service affecting performance issue * Identification of a notifiable service degradation or any major outage * Resolution or restoration of service |
| Usage to Cash | * Retrieval of Usage per customer/on request for a Service * Periodic Service Usage Record for a Service * Threshold Notification for a Service |
| Service Catalog | * Retrieve Service Specification * Create Service Specification * Update Service Specification |

Table 1 NaaS Requirements, Functions and associated APIs

## Sequence Diagrams

The following diagrams are not exhaustive and are examples only.

### O2A CreateService-Atomic-Async

Figure 3 O2A Create Service Atomic Async Sequence diagram

### O2A Modify Atomic Service

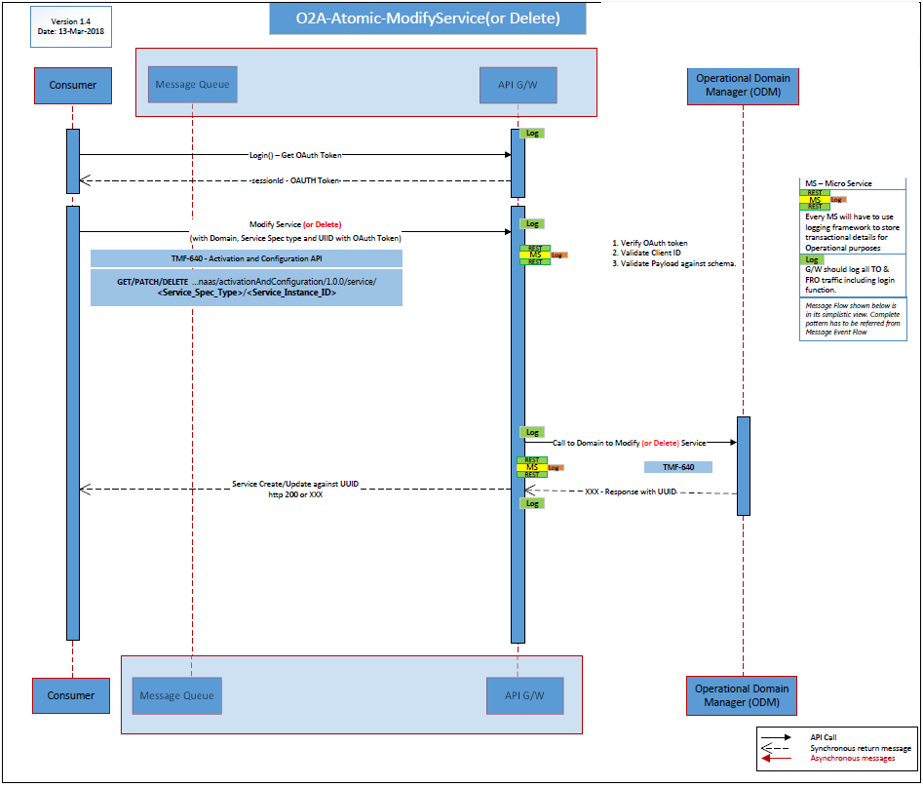


Figure 4 O2A Modify Service Atomic Sequence diagram

### Trouble to Resolve – Reactive Report Service Problem - Sync

### Trouble to Resolve – Reactive Update Service Problem – Sync

Figure 6 T2R Reactive Report Service Problem - Sync

Figure 5 T2R - Reactive Report Service Problem - Sync

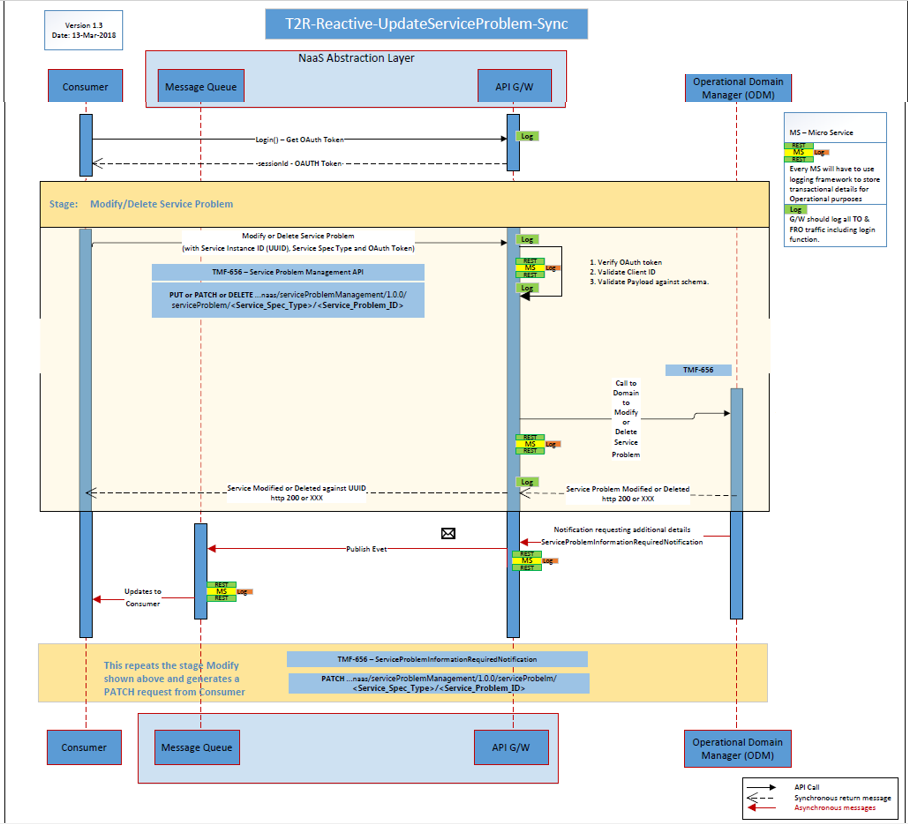


Figure 6 T2R - Reactive Update Service Problem - Sync

### Trouble to Resolve – Proactive Obtain Service Problem Details – Async

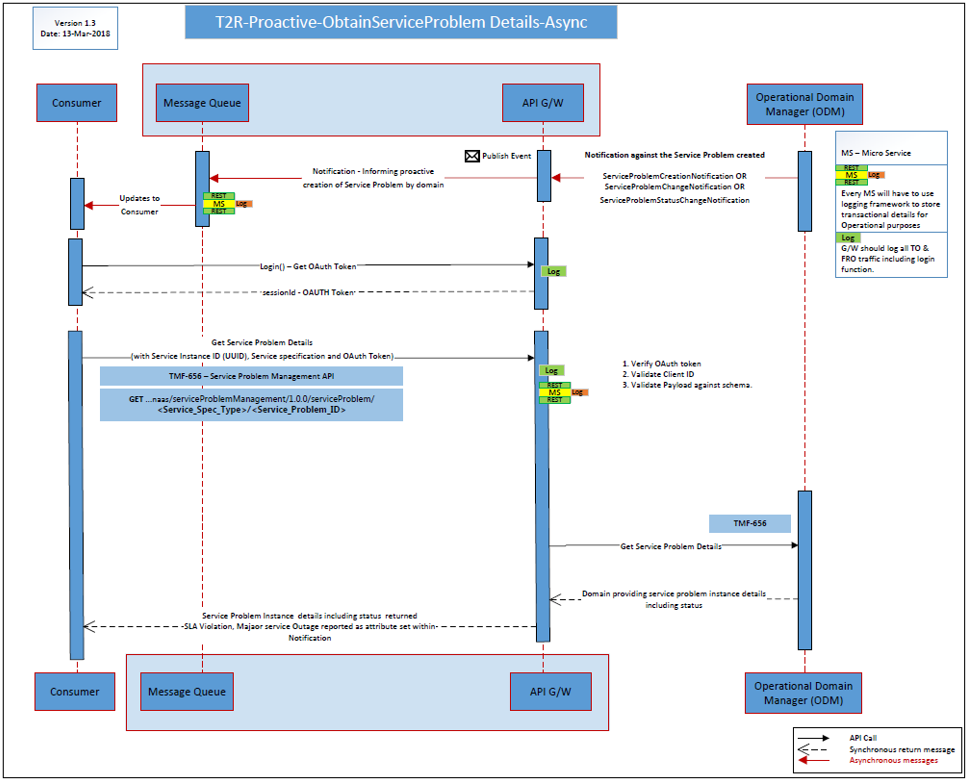


Figure 7 T2R – Proactive Obtain Service Problem Details – Async

### U2C - Retrieval of Usage Record

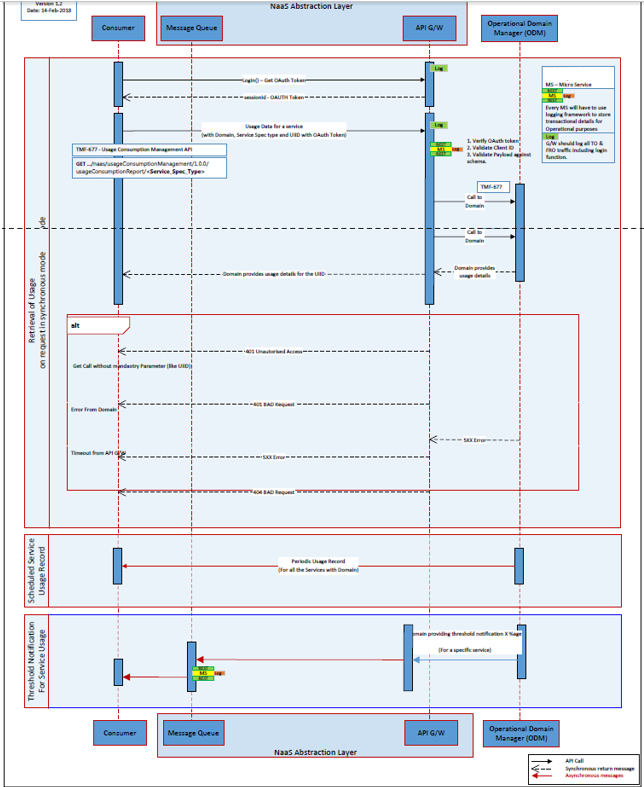


Figure 8 U2C – Retrieval of usage - Sync

# Functions and API Mappings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | **Already Defined** | **Candidate for Common API** | **API Operation and Notification Mapping** | **Comment and Constraints** |
| **Create new service**  **Monitor long running service creation process**  **Modify inflight service creation**  **Cancel inflight service creation**  **Modify existing service**  **Suspend service / restore service**  **Remove service**  **Check feasibility of a service**  **Design and Assign a service and reserve resources**  **Create an inactive service**  **Activate an inactive service**  **Retrieve service details** | Y | NA | **Activation And Configuration (TMF 640)** <https://projects.tmforum.org/wiki/download/attachments/84580852/TMF640_Activation_Configuration_API_REST_Specification_R15.5.1.docx?api=v2>  **POST <https://...../naas/activationAndConfiguration/1.0.0/service>**  **GET https://...../naas/activationAndConfiguration/1.0.0/service /{ID}/Monitor and Monitor State Change Notification**  **PATCH** <https://...../naas/activationAndConfiguration/1.0.0/service/<Service_Instance_ID>**>**  **DELETE https://...../naas/activationAndConfiguration/1.0.0/service /{ID}**  **PATCH https://...../naas/activationAndConfiguration/1.0.0/service /{ID}**  **PATCH <https://...../naas/activationAndConfiguration/1.0.0/service/<Service_Instance_ID>>**  **DELETE** <https://...../naas/activationAndConfiguration/1.0.0/service/<Service_Instance_ID>**>**  **POST <https://...../naas/activationAndConfiguration/1.0.0/service>**  **POST <https://...../naas/activationAndConfiguration/1.0.0/service>**  **POST <https://...../naas/activationAndConfiguration/1.0.0/service>**  **PATCH <https://...../naas/activationAndConfiguration/1.0.0/service/<Service_Instance_ID>>**  **GET <https://...../naas/activationAndConfiguration/1.0.0/service/<Service_Instance_ID>>**  **GET https://...../naas/serviceInventory/1.0.0/service/<Service\_Instance\_ID>** | … |
| **Location based check**  **Feasibility check**  **Design, assign, and reserve** | Y | NA | **Service Qualification (TMF 645) <https://projects.tmforum.org/wiki/display/API/TMF645+Service+Qualification+API+REST+Specification+R18.0.0>**  **POST https://...../naas/ServiceQualification/1.0.0/serviceQualification**  **GET https://...../naas/ServiceQualification/1.0.0/serviceQualification/Service\_Instance\_ID>**  **See Order to Activate (Activation And Configuration) for Feasibility, design and Assign functions.** |  |
| **Create Service Problem**  **Creation of ticket proactively by domain**  **Obtain existing Service Problem details**  **Modify / Update Service problem**  **Delete service problem that’s no longer valid and was raised by consumer system earlier**  **Validate and Test service for reported problems by creating service test**  **Fetch Service test result on completion**  **Modify service test parameters**  **Identification of a notifiable SLA violation**  **Identification of service affecting performance issue**  **Identification of a notifiable service degradation or any major outage**  **Resolution or restoration of service** | Y | NA | **Service Problem Management (TMF 656)**  **<https://projects.tmforum.org/wiki/display/API/TMF656+Service+Problem+Management+API+REST+Specification+R16.5.1>**  **POST <https://...../naas/serviceProblemManagement/1.0.0/serviceProblem>**  **POST <https://...../naas/serviceProblemManagement/1.0.0/serviceProblem>**  **GET <https://...../naas/serviceProblemManagement/1.0.0/serviceProblem/<Service_Problem_ID>>**  **PATCH <https://...../naas/serviceProblemManagement/1.0.0/serviceProblem/<Service_Problem_ID>>**  **DELETE <https://...../naas/serviceProblemManagement/1.0.0/serviceProblem/<Service_Problem_ID> >**  **Service Test Management (TMF 653)**  **<https://projects.tmforum.org/wiki/display/API/TMF653+Service+Test+Management+API+REST+Specification+R16.5.1>**  **POST <https://...../naas/serviceTestManagement/1.0.0/serviceTest/>**  **GET [https://...../naas/serviceTestManagement/1.0.0/serviceTest/<Service\_Test\_ID](https://...../naas/serviceTestManagement/1.0.0/serviceTest/%3cService_Test_ID)>**  **PATCH <https://...../naas/serviceTestManagement/1.0.0/serviceTest/<Service_Test_ID>>**  **ServiceProblemCreateNotification**  **ServiceProblemCreateNotification**  **ServiceProblemCreateNotification**  **ServiceProblemStateChangeNotification** |  |
| **Retrieval of Usage per customer/on request for a Service**  **Periodic Service Usage Record for a Service**  **Threshold Notification for a Service** | Y | NA | **Usage Consumption Management (TMF 677)**  **<https://projects.tmforum.org/wiki/display/API/TMF677+Usage+Consumption+API+REST+Specification+R17.5.0>**  **GET https://...../naas/usageConsumptionManagement /1.0.0/usageConsumptionReport//QueryParameter}**  **UsageReportNotification**  **UsageReportNotification** | • Threshold Notification for Service Usage to indicate that a customer has already consumed X% of the allocated quota. No TMF API support for this requirement. Custom Notification will be used to map this use case until defined by TMF. |
| **Retrieve Service Specification**  **Create Service Specification**  **Update Service Specification** | Y | NA | **Service Catalog Management (TMF 633)**  **<https://projects.tmforum.org/wiki/display/API/TMF633+Service+Catalog+Management+API+REST+Specification+R17.5.0>**  **GET <https://...../naas/catalogManagement/1.0.0/serviceSpecification/<Service_Spec_ID>>**  **POST <https://...../naas/catalogManagement/1.0.0/serviceSpecification>**  **PATCH <https://...../naas/catalogManagement/1.0.0/serviceSpecification/<Service_Spec_ID>>** |  |

## Notification Tables

| **API Name** | **Notifications** |
| --- | --- |
| Activation and Configuration API | serviceCreationNotification  serviceValueChangeNotification  serviceStateChangeNotification  serviceDeletionNotification  monitorCreationNotification  monitorValueChangeNotification  monitorStateChangeNotification  monitorDeletionNotification |
| Service Catalog API | Service Specification Creation  Service Specification Update  Service Specification Remove |
| Service Qualification API | ServiceQualificationStateChangeNotification |
| Usage Consumption Management API | UsageConsumptionReportRequestStateChange |
| Trouble to Resolution API | **serviceProblemManagement**  ServiceProblemCreationNotification  ServiceProblemStatusChangeNotification  ServiceProblemChangeNotification  serviceProblemInformationRequiredNotification  **serviceTest**  ServiceTestCreationNotification  ServiceTestAttributeValueChangeNotification  ServiceTestRemoveNotification  **serviceTestSpecification**  ServiceTestSpecificationCreationNotification  ServiceTestSpecificationAttributeValueChangeNotification  ServiceTestSpecificationRemoveNotification |

Table 2 NaaS API notifications

# Component API Specification

At this time, there is no need to define any new API for this version of the NaaS API Component Suite (First release)

# Administrative Appendix

This Appendix provides additional background material about the TM Forum and this document. In general, sections may be included or omitted as desired, however a Document History must always be included.

## Appendix A: References

|  |  |  |  |
| --- | --- | --- | --- |
| Reference | Description | Source | Brief Use Summary |
|  |  |  |  |

## Document History

### Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date Modified** | **Modified by:** | **Description of changes** |
| 0.1 | 28/Mar/2018 | Johanne Mayer | first issue of document |
| 1.0 | 2/May/2018 | Johanne Mayer | Updated followed feedback |
| 1.1 | 6/May/2018 | Johanne Mayer, Pierre Gauthier,  Ludovic Robert | Addition of Service Order API, update to the API table |
| 1.2 | 12/June/2018 | Johanne Mayer, Corey Clinger | Updated document to include API team review feedback |
| 1.3 | 14/June/2018 | Johanne Mayer | Removed reference to ONAP extAPI as per comment request. |
| 1.4 | 3/October/2018 | Johanne Mayer | Removed references to TMF 641 and Service ordering operational functions |
| 1.5 | 10/October/2018 | Corey Clinger  Johanne Mayer | Added Monitor query and events. Formatting corrections  Remove service order functions from Figure 2. |

## Acknowledgments

This document was prepared by the members of the TM Forum Open API team:

* Johanne Mayer, Telstra
* Corey Clinger, Telstra
* Abdul Majid Hussain, Telstra
* Pierre Gauthier, TM Forum
* Ludovic Robert, Orange
* Dave Milham, TM Forum
* Steve Harrop, Vodafone

Additional input was provided by the following people:

API Governance Group, TM Forum (review, comments, explanations)