TM Forum Specification

Network as a Service (NaaS) API Component Suite Profile

TMF 909A

Team Approved Date: 22/Nov/2019

Release Status: Pre-production	Approval Status: Team Approved
Version 3.0	IPR Mode: RAND

Notice

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100
Parsippany, NJ 07054 USA
Tel No. +1 973 944 5100
Fax No. +1 973 998 7916
TM Forum Web Page: www.TM Forum.org

Table of Contents

List of Figures

List of Tables

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 provider 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.

1. 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 reduces 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** Advanced Platform Deployment Blueprints R17.5.1 section 8

- 8.2. Product Pre-Ordering
- 8.3. Product Ordering and Activation and Configuration (A&C)
- 8.4. SLA Violation Reporting and Dynamic Product Instance Inventory
- 8.5. SLA Violation Correction

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).

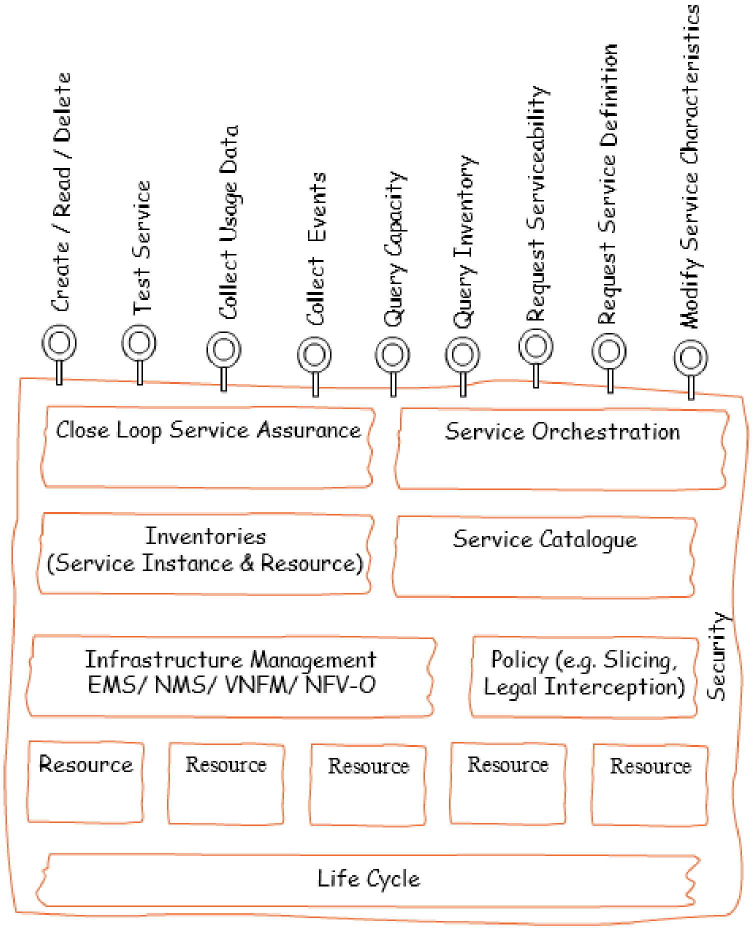
Create / Read / Delete
Test Service
Collect Usage Data
Collect Events
Query Capacity
Query Inventory
Request Serviceability
Request Service Definition
Modify Service Characteristics
Close Loop Service Assurance
Service Orchestration
Inventories (Service Instance & Resource)
Service Catalogue
Infrastructure Management EMS/ NMS/ VNFM/ NFV-O
Policy (e.g. Slicing, Legal Interception)
Security
Resource
Resource
Resource
Resource
Resource
Life Cycle

Figure 1. Operational Domain Management Functions and Capabilities

The NaaS API component suite supports the lifecycle functions required to manage the network capabilities exposed as Network as a Service and managed by operational domains.

2. 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. Similarly, 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 drives the usage of TMF 641 and/or 640 and is at the discretion of each CSP.

2.1. Requirements

The NaaS API component suite includes APIs classified in 5 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)
- Concept to Market
    - The output of service design process are the service specifications.   The existence of the service specifications in the TMF Service Catalog (TMF633) is a pre-condition for this component suite standard

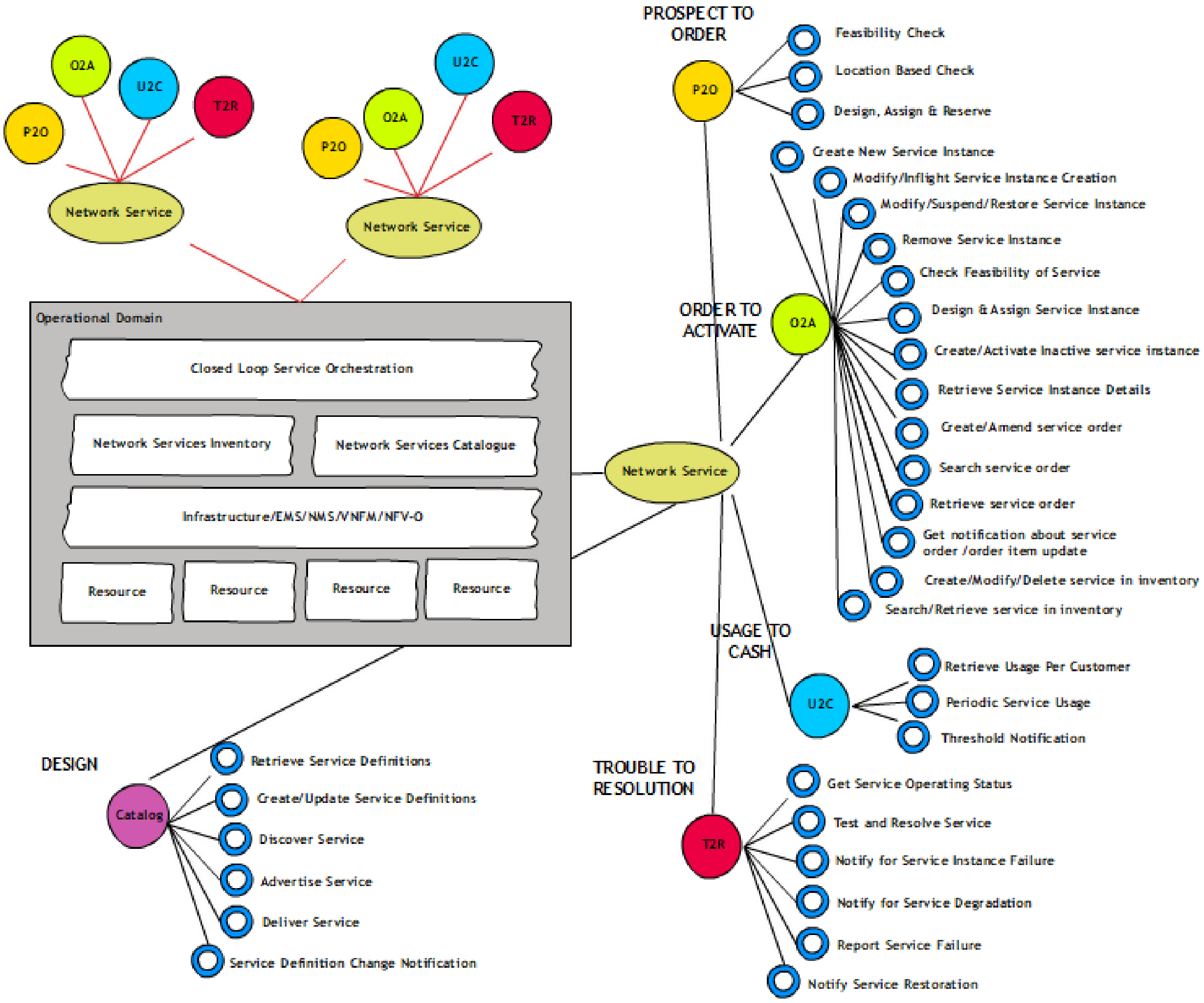
O2A
U2C
T2R
P2O
Network Service
U2C
O2A
T2R
P2O
Network Service
Operational Domain
Closed Loop Service Orchestration
Network Services Inventory
Network Services Catalogue
Infrastructure/EMS/NMS/VNFM/NFV-O
Resource
Resource
Resource
Resource
PROSPECT TO ORDER
P2O
Feasibility Check
Location Based Check
Design, Assign & Reserve
Create New Service Instance
Modify/Inflight Service Instance Creation
Modify/Suspend/Restore Service Instance
Remove Service Instance
Check Feasibility of Service
ORDER TO ACTIVATE
O2A
Design & Assign Service Instance
Create/Activate Inactive service instance
Retrieve Service Instance Details
Create/Amend service order
Search service order
Retrieve service order
Get notification about service order /order item update
Create/Modify/Delete service in inventory
Search/Retrieve service in inventory
Network Service
USAGE TO CASH
U2C
Retrieve Usage Per Customer
Periodic Service Usage
Threshold Notification
DESIGN
Catalog
Retrieve Service Definitions
Create/Update Service Definitions
Discover Service
Advertise Service
Deliver Service
Service Definition Change Notification
TROUBLE TO RESOLUTION
T2R
Get Service Operating Status
Test and Resolve Service
Notify for Service Instance Failure
Notify for Service Degradation
Report Service Failure
Notify Service Restoration

Figure 2. NaaS API component suite classified in the 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.

2.1.1.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.

2.1.2.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.

2.1.3.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.

2.1.4.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.

2.1.5.Service Catalog (Design Time)

Service Catalog provide a centralized repository of all the service definitions offered by an internal (or a partner) domain (s).

The Service Catalog Management allows the management of the entire lifecycle of the Service Catalog elements and the consultation of service catalog elements during several processes such as service activation, service assurance, etc.

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

2.2. Use Cases

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

2.2.1.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
- Create new service order
- Amend existing service order
- Search service order(s)
- Retrieve service order
- Search service (in the inventory)
- Retrieve service (in the inventory)
- Create service (in the inventory – reserved for admin)
- Modify service (in the inventory – reserved for admin)
- Delete service (in the inventory – reserved for admin)

2.2.2.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.

2.2.3.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 detects service problems.

2.2.4.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

2.2.5.Service Catalog (Design Time)

The key Design Time business functions are:

- Retrieve Service Definitions (design and run time)
- Create Service Definitions *
- Update Service Definitions *
- Discover Services
- Advertise Services
- Deliver Services
- Service Definition Change Notification.

* Restricted to the operational domain providing the service.

3. Functions

3.1. 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.

3.2. 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

3.3. 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

3.4. 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

3.5. 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

3.6. 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

3.7. 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

3.8. 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.

3.9. 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.

3.10. 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.

3.11. 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.

3.12. 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

3.13. Create new service order

Pre-condition: for service order item describing new service, the service specification has been defined in the service catalogue. For service order item describing service update, this service exists in the inventory. All Referred to class references such as relatedParty, appointment, and place must exist if id provided.

Post condition: Service order instance has been created. Service order fulfillment could start.

Depending on the service order, the success or failure of the request will be immediately returned. In other cases, the immediate response will simply be that the service order request is processing.

Error: If incorrect id is provided (see pre-conditions) the creation of the service order could fail. It is important to note that the service order is a distinct resource from the service itself. A service order creation could be successful, but the service creation described in one service order item could fail.

3.14. Amend existing service order

Pre-condition: service order exists, and it could be updated.

Post condition: Service order instance has been updated.

Error: Service order did not exist or could not be updated.

3.15. Search service order

Pre-condition: Service orders are stored in the database**.** A list of service order requested criteria(s) is provided.

Post condition: A list of service orders matching request criteria is retrieved.

Error: Not applicable – an empty list because no service order match requested criteria(s) is not considered as an error.

3.16. Retrieve service order

Pre-condition: Service order exists

PostCondition: Attributes of the service order are returned

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

3.17. Search service (in the inventory)

Pre-condition: Service does exist

PostCondition: Attributes of the service are returned

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

3.18. Retrieve service (from the inventory)

Pre-condition: Service does exist

PostCondition: Attributes of the service are returned

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

3.19. Create service (from the inventory – reserved for admin)

Pre-condition: Service does not exist in the inventory. This function can only be supported from an administrative account with the right level of privileges.

PostCondition: Service instance has been created in the inventory.

Error: (?? If not the right privilege or ?)

3.20. Modify service (from the inventory – reserved for admin)

Pre-condition: Service exists in the inventory. The requester has admin privileges to perform the action.

PostCondition: Service attributes have been updated.

Error: Request for non-existing service or attributes will return an error. Request not performed with the right level of privileges.

3.21. Delete service (in the inventory – reserved for admin)

Pre-condition: Service exists in the inventory. The requester has admin privileges to perform the action.

PostCondition: Service has been deleted from the inventory.

Error: (?? If not the right privilege or ?)

3.22. 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

3.23. 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.

3.24. 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.

3.25. 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, a 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.

3.26. 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.

3.27. 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.

3.28. 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.

3.29. 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.

3.30. 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.

3.31. 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.

3.32. Retrieval of Usage per customer/on request[1]

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.

3.33. Threshold Notification for Service

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

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.

3.34. Retrieve Service Definition

Pre-condition: Service specification and candidate exists in catalogue.

Post condition:
The details on the service specification are returned.

Error: If specified service specification (or candidate) is not found, an error will be returned.

3.35. Create Service Definition

Pre-condition: Service specification must not exist in catalogue.

Post condition:
The service specification and service candidate 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.

3.36. Update Service Definition

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

[1] This function is generally used for specific queries such as customer interest or functions such as reviewing usage to trigger special offers, etc.   Per scheduled push/pull of data used for routine billing cycles is not included in this standard.

3.37. Discover Services

Pre-condition: Service candidate exists in catalogue.

Post condition:
The service candidate and its related specification are returned.

Error: If specified service candidate (or specification) is not found, an error will be returned

3.38. Advertise Services

Pre-condition: Service candidate must not exist in catalogue.
Post condition:
The service candidate is specification creation notifications are sent to all the subscribers of catalog.

Error: NA, implementation specific.

3.39. Deliver Services

Pre-condition: Service candidate exists in catalogue.

Post condition:
The service specification is created under service candidate to advertise that a service definition exists for consumption.

Error: NA, implementation specific.

3.40. Service Definition Change Notification.

Pre-condition: Service catalog elements exists in catalogue.

Post condition:
Changes to service catalog elements results in notification which gets delivered to subscribed consumers.
Error: NA, implementation specific.

4. Component Capabilities, Flows and Sequence Diagrams

4.1. 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 reserve resources • Create an inactive service • Activate an inactive service • Retrieve service details • Create new service order • Amend existing service order • Search service order(s) • Retrieve service order • Search service (in the inventory) • Retrieve service (in the inventory) • Create service (in the inventory – reserved for admin) • Modify service (in the inventory – reserved for admin) • Delete service (in the inventory – reserved for admin)
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

Requirement	Function
	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 Definitions • Create Service Definitions • Update Service Definitions • Discover Services • Advertise Services • Deliver Services • Service Definition Change Notification.

Table 1. NaaS Requirements, Functions and associated APIs

4.2. Sequence Diagrams

The following diagrams are not exhaustive and are examples only.

4.2.1.O2A CreateService-Atomic-Async

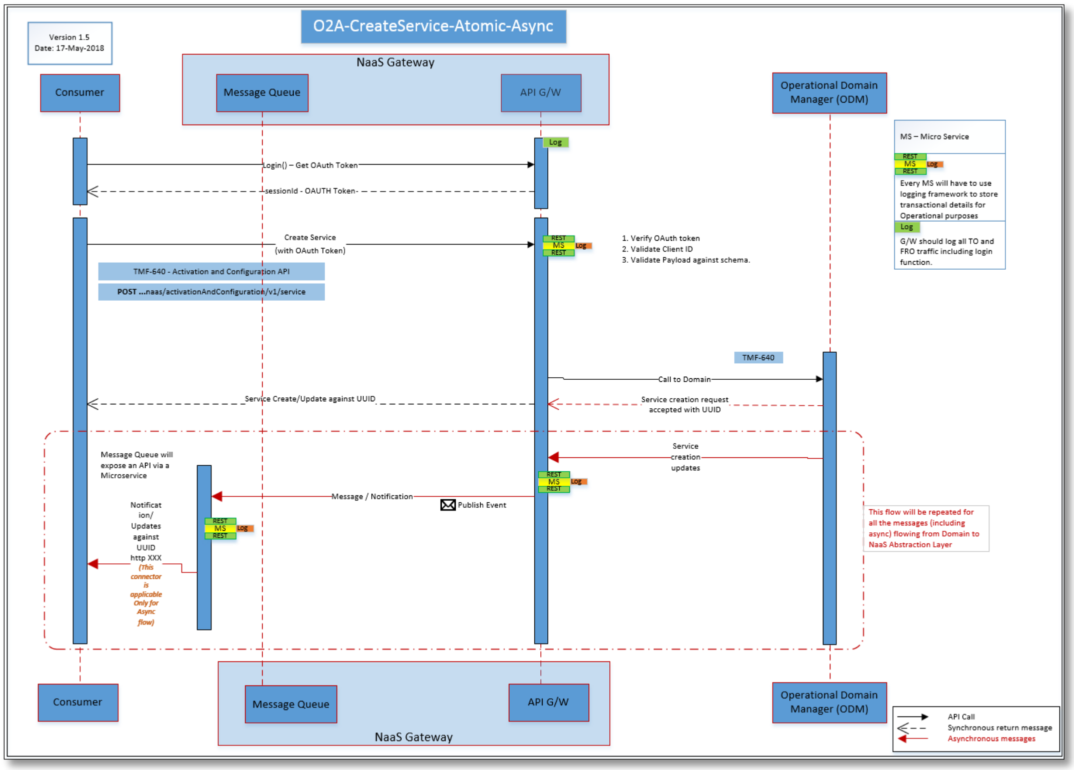
Version 1.5
Date: 17-May-2018
O2A-CreateService-Atomic-Async
NaaS Gateway
Consumer
Message Queue
API G/W
Operational Domain Manager (ODM)
Log
Login() – Get OAuth Token
sessionId - OAUTH Token
Create Service (with OAuth Token)
TMF-640 - Activation and Configuration API
POST ...naas/activationAndConfiguration/v1/service
REST
MS
REST
Log
1. Verify OAuth token
2. Validate Client ID
3. Validate Payload against schema.
MS – Micro Service
Every MS will have to use logging framework to store transactional details for Operational purposes
G/W should log all TO and FRO traffic including login function.
TMF-640
Call to Domain
Service Create/Update against UUID
Service creation request accepted with UUID
Service creation updates
Message Queue will expose an API via a Microservice
Message / Notification
Publish Event
Notification/ Updates against UUID http XXX
(This connector is applicable Only for Async flow)
This flow will be repeated for all the messages (including async) flowing from Domain to NaaS Abstraction Layer
Consumer
Message Queue
API G/W
NaaS Gateway
Operational Domain Manager (ODM)
API Call
Synchronous return message
Asynchronous messages

Figure 3. O2A Create Service Atomic Async Sequence diagram

4.2.2.O2A Modify Atomic Service

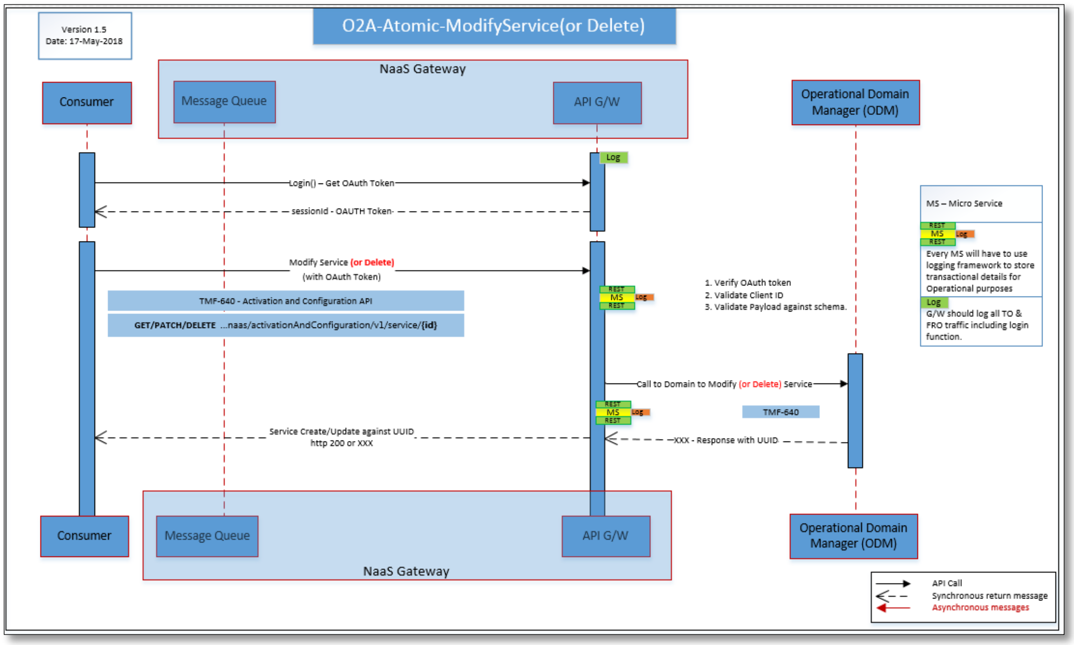
Version 1.5
Date: 17-May-2018
O2A-Atomic-ModifyService(or Delete)
NaaS Gateway
Consumer
Message Queue
API G/W
Operational Domain Manager (ODM)
Log
Login() – Get OAuth Token
sessionId - OAUTH Token
Modify Service (or Delete) (with OAuth Token)
TMF-640 - Activation and Configuration API
GET/PATCH/DELETE ...naas/activationAndConfiguration/v1/service/{id}
REST
MS
REST
Log
1. Verify OAuth token
2. Validate Client ID
3. Validate Payload against schema.
MS – Micro Service
Every MS will have to use logging framework to store transactional details for Operational purposes
G/W should log all TO & FRO traffic including login function.
Call to Domain to Modify (or Delete) Service
TMF-640
Service Create/Update against UUID http 200 or XXX
XXX - Response with UUID
Consumer
Message Queue
API G/W
NaaS Gateway
Operational Domain Manager (ODM)
API Call
Synchronous return message
Asynchronous messages

Figure 4. O2A Modify Service Atomic Sequence diagram

4.2.3.O2A – Manage Service Order

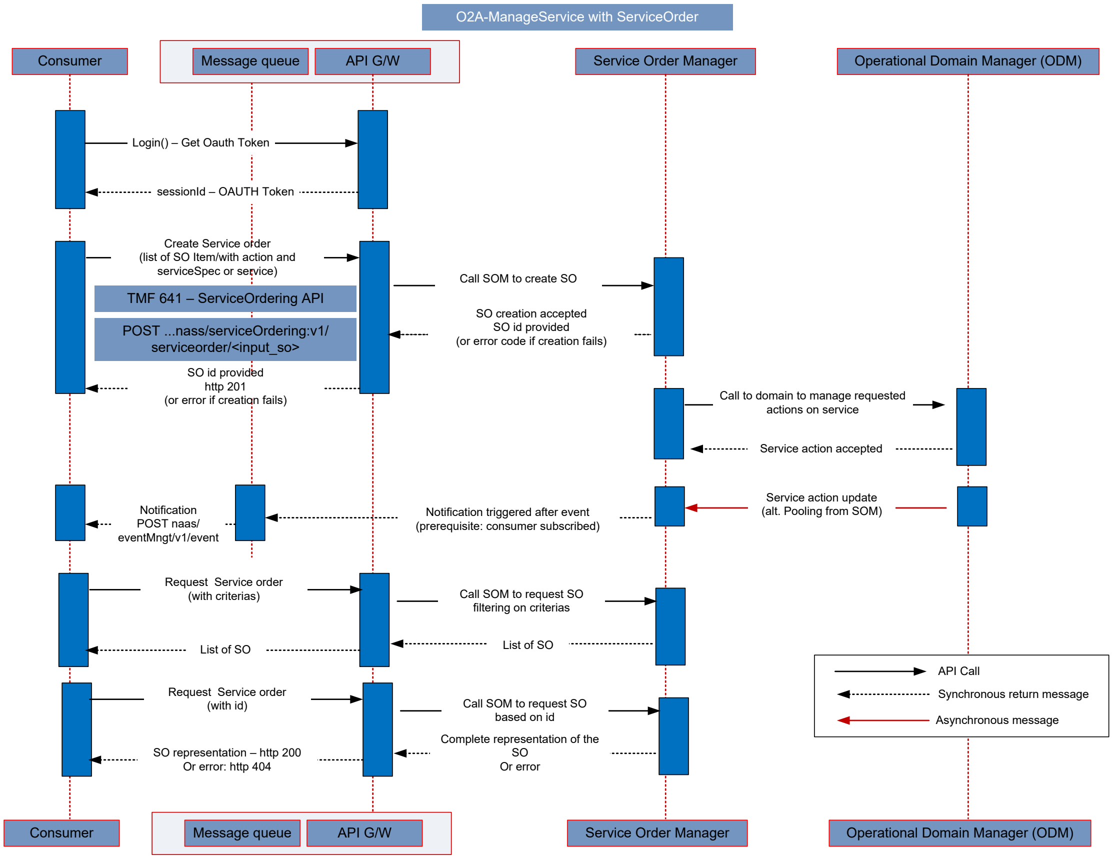
O2A-ManageService with ServiceOrder
Consumer
Message queue
API G/W
Service Order Manager
Operational Domain Manager (ODM)
Login() – Get Oauth Token
sessionId – OAUTH Token
Create Service order (list of SO Item/with action and serviceSpec or service)
TMF 641 – ServiceOrdering API
POST ...nass/serviceOrdering:v1/ serviceorder/<input_so>
SO id provided http 201 (or error if creation fails)
Call SOM to create SO
SO creation accepted SO id provided (or error code if creation fails)
Call to domain to manage requested actions on service
Service action accepted
Service action update (alt. Pooling from SOM)
Notification triggered after event (prerequisite: consumer subscribed)
Notification POST naas/ eventMngt/v1/event
Request Service order (with criterias)
Call SOM to request SO filtering on criterias
List of SO
List of SO
Request Service order (with id)
Call SOM to request SO based on id
Complete representation of the SO Or error
SO representation – http 200 Or error: http 404
API Call
Synchronous return message
Asynchronous message
Consumer
Message queue
API G/W
Service Order Manager
Operational Domain Manager (ODM)

Figure 5. Manage Service order diagram

4.2.4.Trouble to Resolve – Reactive Report Service Problem - Sync

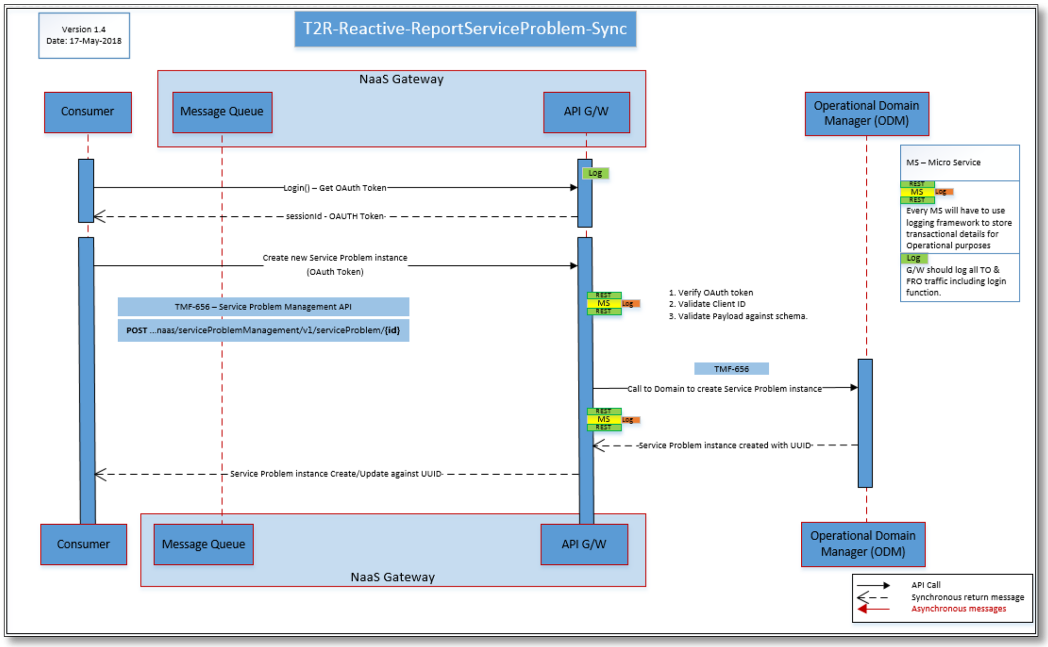
Version 1.4
Date: 17-May-2018
T2R-Reactive-ReportServiceProblem-Sync
NaaS Gateway
Consumer
Message Queue
API G/W
Operational Domain Manager (ODM)
MS – Micro Service
REST
MS
Log
REST
Every MS will have to use logging framework to store transactional details for Operational purposes
Log
G/W should log all TO & FRO traffic including login function.
Log
Login() – Get OAuth Token
sessionId - OAUTH Token
Create new Service Problem instance (OAuth Token)
TMF-656 – Service Problem Management API
POST ...naas/serviceProblemManagement/v1/serviceProblem/{id}
REST
MS
Log
REST
1. Verify OAuth token
2. Validate Client ID
3. Validate Payload against schema.
TMF-656
Call to Domain to create Service Problem instance
REST
MS
Log
REST
Service Problem instance created with UUID
Service Problem instance Create/Update against UUID
Consumer
Message Queue
API G/W
NaaS Gateway
Operational Domain Manager (ODM)
API Call
Synchronous return message
Asynchronous messages

Figure 6. T2R - Reactive Report Service Problem - Sync

4.2.5.Trouble to Resolve – Reactive Update Service Problem – Sync

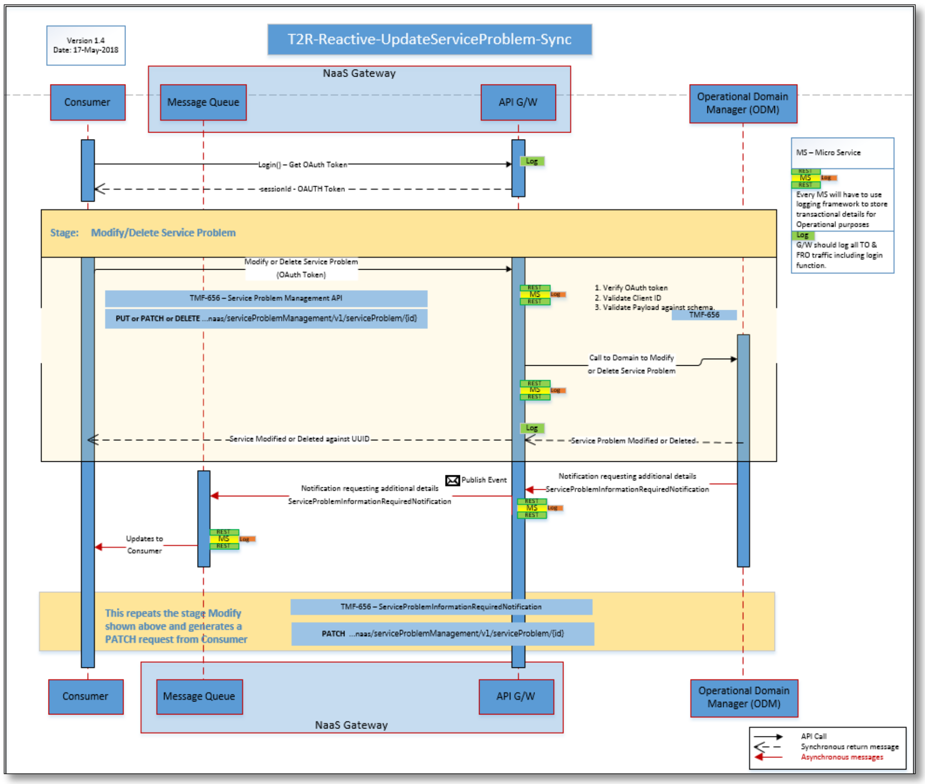
Version 1.4
Date: 17-May-2018
T2R-Reactive-UpdateServiceProblem-Sync
NaaS Gateway
Consumer
Message Queue
API G/W
Operational Domain Manager (ODM)
MS – Micro Service
REST
MS
Log
REST
Every MS will have to use logging framework to store transactional details for Operational purposes
Log
G/W should log all TO & FRO traffic including login function.
Login() – Get OAuth Token
sessionId - OAUTH Token
Stage: Modify/Delete Service Problem
Modify or Delete Service Problem (OAuth Token)
TMF-656 – Service Problem Management API
PUT or PATCH or DELETE ...naas/serviceProblemManagement/v1/serviceProblem/{id}
1. Verify OAuth token
2. Validate Client ID
3. Validate Payload against schema.
TMF-656
Call to Domain to Modify or Delete Service Problem
Service Modified or Deleted against UUID
Service Problem Modified or Deleted
Publish Event
Notification requesting additional details
ServiceProblemInformationRequiredNotification
Notification requesting additional details
ServiceProblemInformationRequiredNotification
Updates to Consumer
This repeats the stage Modify shown above and generates a PATCH request from Consumer
TMF-656 – ServiceProblemInformationRequiredNotification
PATCH ...naas/serviceProblemManagement/v1/serviceProblem/{id}
Consumer
Message Queue
API G/W
Operational Domain Manager (ODM)
NaaS Gateway
API Call
Synchronous return message
Asynchronous messages

Figure 7. T2R - Reactive Update Service Problem - Sync

4.2.6.Trouble to Resolve – Proactive Obtain Service Problem Details – Async

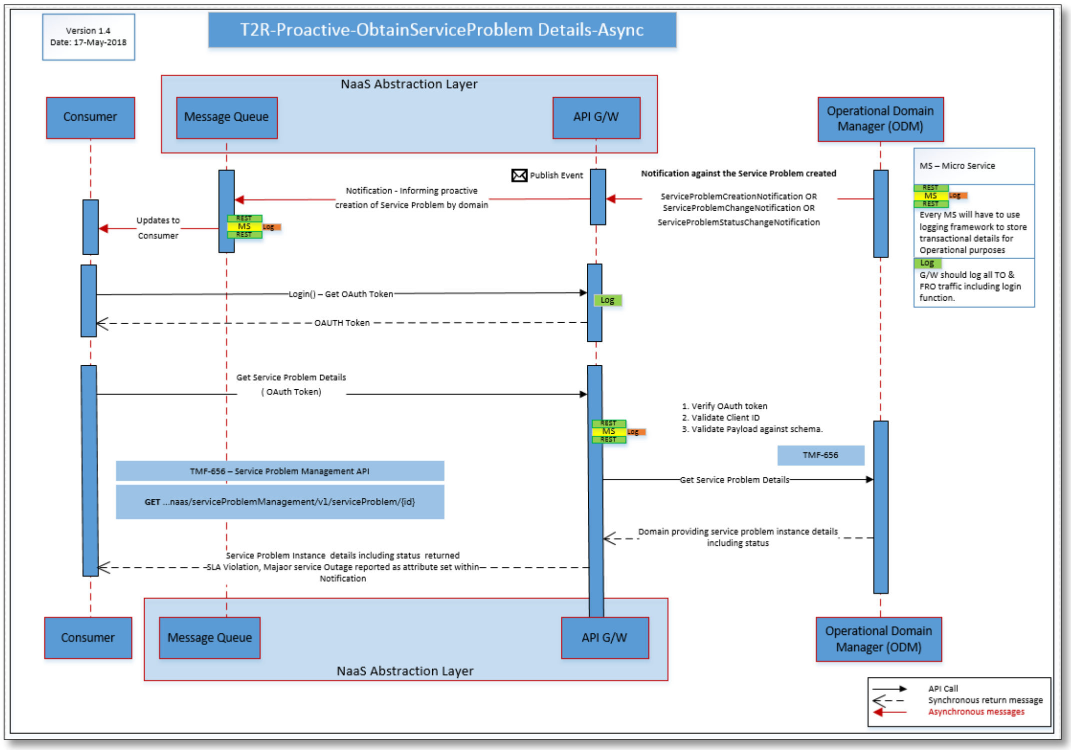
Version 1.4
Date: 17-May-2018
T2R-Proactive-ObtainServiceProblem Details-Async
NaaS Abstraction Layer
Consumer
Message Queue
API G/W
Operational Domain Manager (ODM)
MS – Micro Service
REST
MS
Log
REST
Every MS will have to use logging framework to store transactional details for Operational purposes
Log
G/W should log all TO & FRO traffic including login function.
Publish Event
Notification against the Service Problem created
ServiceProblemCreationNotification OR
ServiceProblemChangeNotification OR
ServiceProblemStatusChangeNotification
Notification - Informing proactive creation of Service Problem by domain
Updates to Consumer
REST
MS
Log
REST
Login() – Get OAuth Token
Log
OAUTH Token
Get Service Problem Details
(OAuth Token)
1. Verify OAuth token
2. Validate Client ID
3. Validate Payload against schema.
REST
MS
Log
REST
TMF-656
TMF-656 – Service Problem Management API
GET ...naas/serviceProblemManagement/v1/serviceProblem/{id}
Get Service Problem Details
Domain providing service problem instance details including status
Service Problem Instance details including status returned
SLA Violation, Majaor service Outage reported as attribute set within Notification
Consumer
Message Queue
API G/W
Operational Domain Manager (ODM)
NaaS Abstraction Layer
API Call
Synchronous return message
Asynchronous messages

Figure 8. T2R – Proactive Obtain Service Problem Details – Async

4.2.7.U2C - Retrieval of Usage Record

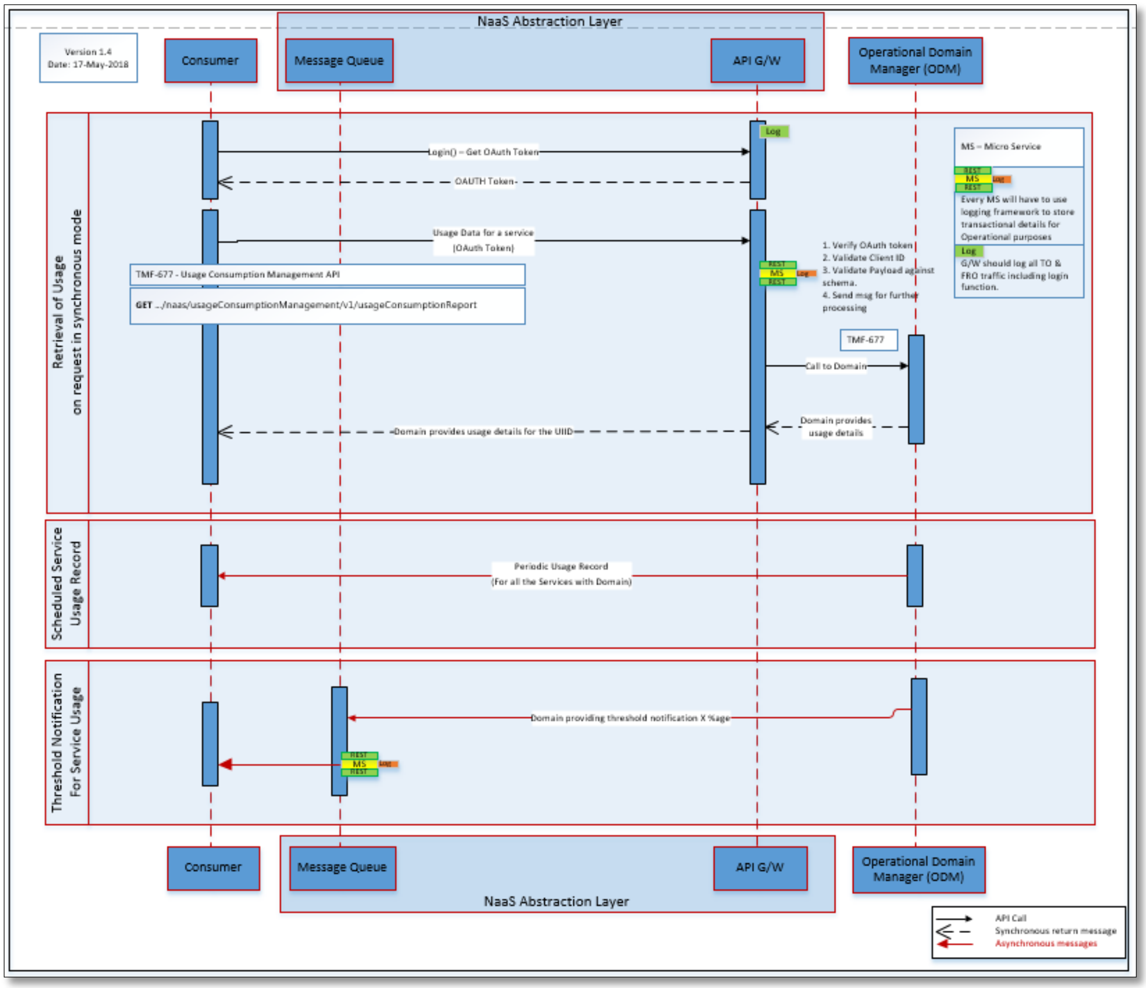
NaaS Abstraction Layer
Version 1.4
Date: 17-May-2018
Consumer
Message Queue
API G/W
Operational Domain Manager (ODM)
Retrieval of Usage
on request in synchronous mode
Log
Login() – Get OAuth Token
OAUTH Token
Usage Data for a service
(OAuth Token)
TMF-677 - Usage Consumption Management API
GET …/naas/usageConsumptionManagement/v1/usageConsumptionReport
REST
MS
Log
REST
1. Verify OAuth token
2. Validate Client ID
3. Validate Payload against schema.
4. Send msg for further processing
MS – Micro Service
REST
MS
Log
REST
Every MS will have to use logging framework to store transactional details for Operational purposes
Log
G/W should log all TO & FRO traffic including login function.
TMF-677
Call to Domain
Domain provides usage details
Domain provides usage details for the UIID
Scheduled Service
Usage Record
Periodic Usage Record
(For all the Services with Domain)
Threshold Notification
For Service Usage
Domain providing threshold notification X %age
REST
MS
Log
REST
Consumer
Message Queue
API G/W
Operational Domain Manager (ODM)
NaaS Abstraction Layer
API Call
Synchronous return message
Asynchronous messages

Figure 9. U2C – Retrieval of usage – Sync

4.2.8. Master Service Catalog

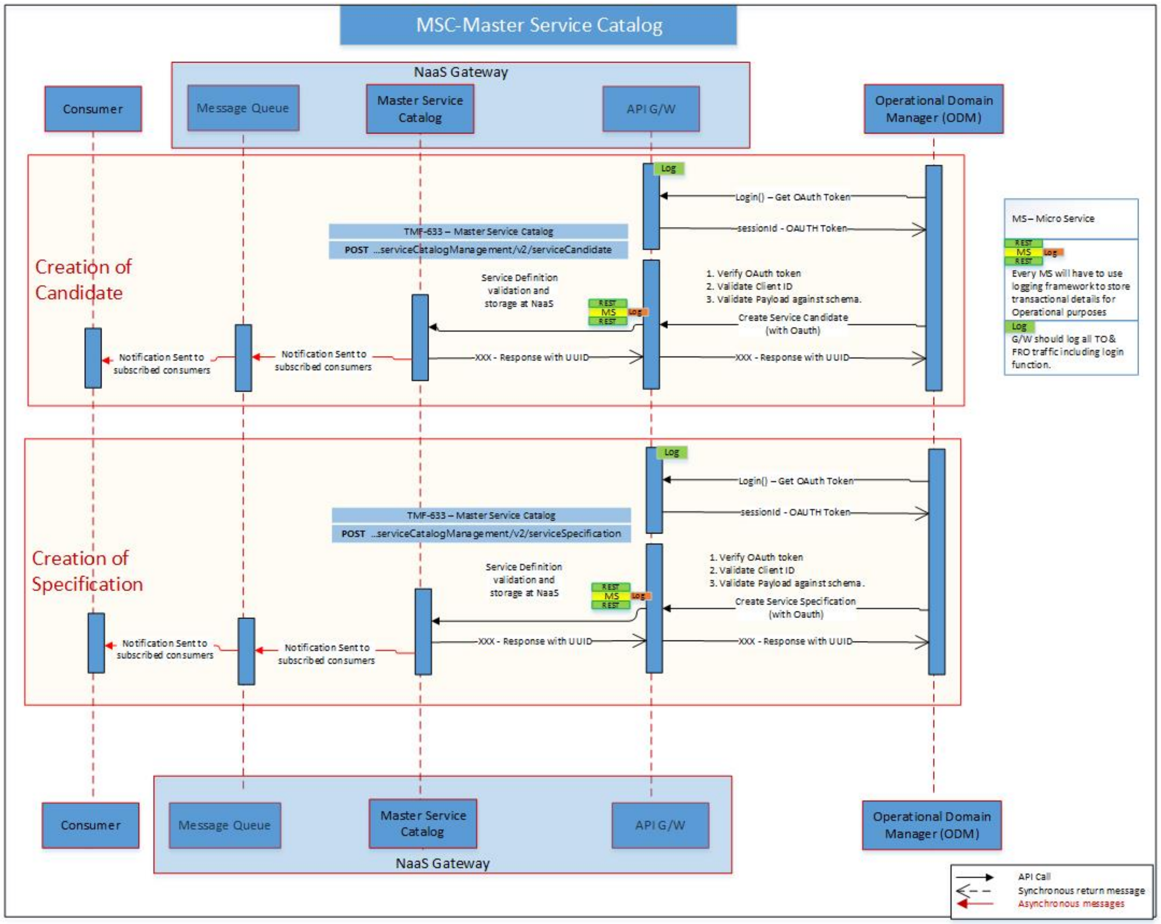
MSC-Master Service Catalog
NaaS Gateway
Consumer
Message Queue
Master Service Catalog
API G/W
Operational Domain Manager (ODM)
Creation of Candidate
Log
Login() – Get OAuth Token
sessionId - OAUTH Token
TMF-633 – Master Service Catalog
POST ...serviceCatalogManagement/v2/serviceCandidate
Service Definition validation and storage at NaaS
REST
MS
Log
REST
1. Verify OAuth token
2. Validate Client ID
3. Validate Payload against schema.
Create Service Candidate (with Oauth)
XXX - Response with UUID
XXX - Response with UUID
Notification Sent to subscribed consumers
Notification Sent to subscribed consumers
MS – Micro Service
REST
MS
Log
REST
Every MS will have to use logging framework to store transactional details for Operational purposes
Log
G/W should log all TO & FRO traffic including login function.
Creation of Specification
Log
Login() – Get OAuth Token
sessionId - OAUTH Token
TMF-633 – Master Service Catalog
POST ...serviceCatalogManagement/v2/serviceSpecification
Service Definition validation and storage at NaaS
REST
MS
Log
REST
1. Verify OAuth token
2. Validate Client ID
3. Validate Payload against schema.
Create Service Specification (with Oauth)
XXX - Response with UUID
XXX - Response with UUID
Notification Sent to subscribed consumers
Notification Sent to subscribed consumers
Consumer
Message Queue
Master Service Catalog
API G/W
NaaS Gateway
Operational Domain Manager (ODM)
API Call
Synchronous return message
Asynchronous messages

Figure 10. Design – Creation of Service Candidate and Service Specification in Master Service Catalog

5. Functions and API Mappings

Function Name	Already Defined	Candidate for Common API	API Operation and Notification Mapping	Comment and Constraints
	Y	NA	**Activation And Configuration (TMF 640)** https://projects.tmforum.org/wiki/display/PUB/TMF640+Service+Activation+and+Configuration+API+REST+Specification+R18.5.1	…
Create new service			**POST** https://...../naas/ServiceActivationAndConfiguration/v4/service	
Monitor long running service creation process			**GET** https://...../naas/ServiceActivationAndConfiguration/v4/monitor/{id}	
Modify inflight service creation			**PATCH** https://...../naas/ServiceActivationAndConfiguration/v4/service/{id}	
Cancel inflight service creation			**DELETE** https://...../naas/ServiceActivationAndConfiguration/v4/service/{id}	
Modify existing service			**PATCH** https://...../naas/ServiceActivationAndConfiguration/v4/service/{id}	
Suspend service / restore service			**PATCH** https://...../naas/ServiceActivationAndConfiguration/v4/service/{id}	
Remove service			**DELETE** https://...../naas/ServiceActivationAndConfiguration/v4/service/{id}	
Check feasibility of a service			**POST** https://...../naas/ServiceActivationAndConfiguration/v4/service	
Design and Assign a service and reserve resources			**POST** https://...../naas/ServiceActivationAndConfiguration/v4/service	
Create an inactive service			**POST** https://...../naas/ServiceActivationAndConfiguration/v4/service	
Activate an inactive service			**PATCH** https://...../naas/ServiceActivationAndConfiguration/v4/service/{id}	
Retrieve service details			**GET** https://...../naas/ServiceActivationAndConfiguration/v4/service/{id}	

Function Name	Already Defined	Candidate for Common API	API Operation and Notification Mapping	Comment and Constraints
Create new service order **Amend existing service order** **Search service order(s)** **Retrieve service order**	**Y**	**NA**	**Service ordering (TMF 641)** https://projects.tmforum.org/wiki/display/PUB/TMF641+Service+Ordering+API+REST+Specification+R18.5.1 **POST https://…/naas/serviceOrdering/4.0.0/serviceOrder/<SO_input>** **PATCH https://…/naas/serviceOrdering/4.0.0/serviceOrder/<serviceOrderId>** **GET https://…/naas/serviceOrdering/4.0.0/serviceOrder/<serviceOrderId>** **GET https://…/naas/serviceOrdering/4.0.0/serviceOrder/?<criteria=value&filtering>**	
Search service (in the inventory) **Retrieve service (in the inventory)** **Create service (in the inventory – reserved for admin)** **Modify service (in the inventory – reserved for admin)** **Delete service (in the inventory – reserved for admin)**	Y	NA	**Service Inventory (TMF638)** https://projects.tmforum.org/wiki/display/PUB/TMF638+Service+Inventory+API+REST+Specification+R18.5.1 **GET https:://…/naas/serviceInventory/4.0.0/service/?<criteria=value&filtering>** **GET https:://…/naas/serviceInventory/4.0.0/service/<serviceId>** **POST https:://…/naas/serviceInventory/4.0.0/service/<service_input>** **PATCH https:://…/naas/serviceInventory/4.0.0/service/<service_updateInput>** **DELETE https:://…/naas/serviceInventory/4.0.0/service/<serviceId>**	
Location based check **Feasibility check** **Design, assign, and**	Y	NA	**Service Qualification (TMF 645)** https://projects.tmforum.org/wiki/display/PUB/TMF645+Service+Qualification+API+REST+Specification+R18.5.1 **POST** https://…../naas/serviceQualificationManagement/v4/serviceQualification **GET** https://…../naas/serviceQualificationManagement/v4/serv	

Function Name	Already Defined	Candidate for Common API	API Operation and Notification Mapping	Comment and Constraints
reserve			iceQualification/{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**	Y	NA	**Service Problem Management (TMF 656)** https://projects.tmforum.org/wiki/display/PUB/TMF656+Service+Problem+API+REST+Specification+R18.5.1 **POST** https://...../naas/serviceProblemManagement/v4/serviceProblem **POST** https://...../naas/serviceProblemManagement/v4/serviceProblem **GET** https://...../naas/serviceProblemManagement/v4/serviceProblem/{id} **PATCH** https://...../naas/serviceProblemManagement/v4/serviceProblem/{id} **DELETE** https://...../naas/serviceProblemManagement/v4/serviceProblem/{id} **Service Test Management (TMF 653)** https://projects.tmforum.org/wiki/display/PUB/TMF653+Service+Test+API+REST+Specification+R18.5.1 **POST** https://...../naas/serviceTestManagement/v4/serviceTest **GET** https://...../naas/serviceTestManagement/v4/serviceTest/{id} **PATCH** https://...../naas/serviceTestManagement/v4/serviceTest/{id} **ServiceProblemCreateNotification** **ServiceProblemCreateNotification**	

Function Name	Already Defined	Candidate for Common API	API Operation and Notification Mapping	Comment and Constraints
Identification of service affecting performance issue **Identification of a notifiable service degradation or any major outage** **Resolution or restoration of service**			**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/PUB/TMF677+Usage+Consumption+API+REST+Specification+R18.5.1 **GET** https://...../naas/usageConsumption/v4/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** **Advertise Service Definition**	Y	NA	**Service Catalog Management (TMF 633)** https://projects.tmforum.org/wiki/display/PUB/TMF633+Service+Catalog+API+REST+Specification+R18.5.1 **GET** https://...../naas/serviceCatalogManagement/v4/serviceSpecification/{id} **POST** https://...../naas/serviceCatalogManagement/v4/serviceSpecification **PATCH** https://...../naas/serviceCatalogManagement/v4/serviceSpecification/{id} **POST** https://...../naas/serviceCatalogManagement/v4/serviceCandidate **PATCH**	

Function Name	Already Defined	Candidate for Common API	API Operation and Notification Mapping	Comment and Constraints
Deliver Service Definition			https://…../naas/serviceCatalogManagement/v4/serviceCandidate/{id} **ServiceCandidateCreationNotification** **ServiceCandidateRemoveNotification** **ServiceSpecificationCreationNotification** **ServiceSpecificationRemoveNotification**	

5.1. Notification Tables

API Name	Notifications
Activation and Configuration API	serviceCreationNotification serviceValueChangeNotification serviceStateChangeNotification serviceDeletionNotification monitorCreationNotification monitorValueChangeNotification monitorStateChangeNotification monitorDeletionNotification
Service Ordering API	ServiceOrderCreateNotification ServiceOrderAttributeValueChangeNotification ServiceOrderStateChangeNotification ServiceOrderDeleteNotification
Service Inventory API	ServiceCreateNotification ServiceAttributeValueChangeNotification ServiceStateChangeNotification ServiceBatchNotification ServiceDeleteNotification
Service Catalog API	ServiceCandidateCreationNotification ServiceCandidateRemoveNotification ServiceSpecificationCreationNotification ServiceSpecificationRemoveNotification
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

6. 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)

7. 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.

7.2. Appendix A: References

Reference	Description	Source	Brief Use Summary

7.3. Document History

7.3.1.Version History

Version Number	Date Modified	Modified by:	Description of changes
0.1	28-Mar-2018	Johanne Mayer	first issue of document
1.0	02-May-2018	Johanne Mayer	Updated followed feedback
1.1	06-May-2018	Johanne Mayer, Pierre Gauthier, Ludovic Robert	Addition of Service Order API, update to the API table
1.2	12-Jun-2018	Johanne Mayer, Corey Clinger	Updated document to include API team review feedback
1.3	14-Jun-2018	Johanne Mayer	Removed reference to ONAP extAPI as per comment request.
1.4	03-Oct-2018	Johanne Mayer	Removed references to TMF 641 and Service ordering operational functions
1.5	10-Oct-2018	Corey Clinger Johanne Mayer	Added Monitor query and events. Formatting corrections Remove service order functions from Figure 2.
1.6	03-Jan-2019	Adrienne Walcott Derek Flexer	Convert to recent TM Forum branding
1.7	13-Mar-2019	Abdul Majid Hussain, Varun Nair	Major updates as below • Updated URI formats • Added flow diagram for MSC • Addedd/updated use case for MSC • Added section 7.5 to track the reference API and its versions
1.8	29-Aug-2019	Johanne Mayer	Addition of TMF641 service ordering and TMF638 service inventory APIs Updated all API information to align with 18.5 versions

Version Number	Date Modified	Modified by:	Description of changes
			Removal of section 7.5 since all APIs are aligned with 18.5
3.0	22-Nov-2019	Alan Pope	First Version of the NaaS API component suite published

7.3.2 Release History

Production Release Status	Date Modified	Modified by:	Description of changes
Pre-production	22-Nov-2019	Alan Pope	First version of the NaaS API component suite published.

7.4. 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
- Varun Nair, 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)