Service Activation - DDP BA - Part 3: Service Component Activation

TMF518_SA_3

Version 1.2

September, 20011

Notice

No recipient of this document and code shall in any way interpret this material as representing a position or agreement of TM Forum or its members. This material is draft working material of TM Forum and is provided solely for comments and evaluation. It is not “Forum Approved” and is solely circulated for the purposes of assisting TM Forum in the preparation of final material in furtherance of the aims and mission of TM Forum.

Although it is copyrighted material of TM Forum:

- Members of TM Forum are only granted the limited copyright waiver to distribute this material within their companies and may not make paper or electronic copies for distribution outside of their companies.
- Non-members of the TM Forum are not permitted to make copies (paper or electronic) of this draft material other than for their internal use for the sole purpose of making comments thereon directly to TM Forum.
- If this material forms part of a supply of information in support of an Industry Group Liaison relationship, the document may only be used as part of the work identified in the Liaison and may not be used or further distributed for any other purposes

Any use of this material by the recipient, other than as set forth specifically herein, is at its own risk, and under no circumstances will TM Forum be liable for direct or indirect damages or any costs or losses resulting from the use of this material by the recipient.

This material is governed, and all recipients shall be bound, by all of the terms and conditions of the Intellectual Property Rights Policy of the TM Forum (http://www.tmforum.org/Bylaws/1094/home.html) and may involve a claim of patent rights by one or more TM Forum members or by non-members of TM Forum.

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

Table of Contents

List of Requirements

Formatted: English (U.K.)

Field Code Changed

Formatted: English (U.K.)

Field Code Changed

Formatted: English (U.K.)

Formatted: English (U.K.)

Formatted: French (France)

Field Code Changed

Formatted: French (France)

Field Code Changed

Formatted: English (U.K.)

Formatted: English (U.K.)

List of Figures

List of Tables

Executive Summary

This document is Part 3 of the MTOSI Service Activation Document Delivery Package (DDP). It covers requirements and use cases for a Service Component Activation Interface (SCAI). This interface is entirely within the eTOM SM&O layer. As such, it receives activation requests from another SM&O application that is normally responsible for the orchestration of end-to-end service activation. The interfaces exposed by the SCAI hide the complexity of the underlying resource activation process through the use of service templates and references to service access points.

1 Introduction

1.1 DDP Structure

In order to allow for more efficient release delivery, the previous monolithic BA, IA and SS documents have been partitioned into smaller self-contained (though not independent) units called Document Delivery Packages (DDPs).

This is similar to the 3GPP concept of Integration Reference Point (IRP). The basic idea is that the Interface, which is specified by the entire document set (of a release), is partitioned into DDPs where each DDP specifies “a certain aspect” of the Interface, which needs to be very clearly scoped.

There are three kinds of DDPs:

- the FrameWork DDP (FMW) – this DDP contains the generic artifacts that are applicable to all the other DDPs.
- Data Model DDP (DM-DDP) – a DDP that concerns a data model (entities, data structures, attributes, state, but no operations)
- Operation Model DDP (OM-DDP) – a DDP that concerns a computational model (operations, notifications, transactions) for a given functional area (such as resource inventory management)

The unified deliverables structure for any given MTOSI / MTNM product release is as follows:

- Product Release Notes:
  - a scope specification for the type and extent of the delivered product,
  - the partitioning of the release into DDPs (i.e., definitions of various aspects of the release),
  - and an overview of the release’s (delta) deliverables;
- For each DDP:
  - Business Agreements (BAs): a business view specification
  - Information Agreements (IAs): a system view specification
  - Interface Implementation Specifications (ISSs): implementation and deployment view specification per supported enabling technology (mapping of the IA to either CORBA (IDL, services usage) or XML (WSDL, XSD, bindings…)
  - Supporting Documentation: normative and informative supporting documents.
- Reference Implementation (optional) of core IIS fragments for selected interfaces and enabling technologies.

1.2 Document Structure

This document covers the TMF Service Component Activation Interface (SCAI) Business Agreement. Service components are activated as part of an end-to-end activation process managed at the SM&O layer of the eTOM. The general business requirements for service component activation are defined, followed by the functional requirements and use cases.

This document has the following sections:

- Section 1~~1~~ is the introduction.
- Section 2~~2~~ defines the business problem and scope.
- Section 3~~3~~ contains the requirements. Lists the functional requirements to be fulfilled by each interface for each business scenario as described in the problem statement.
- Section 4~~4~~ has the use cases.
- Section 5~~5~~ provides a traceability matrix between use cases and requirements, and vice versa.
- Section 6~~6~~ provides a summary and list of open issues.
- Section 7~~7~~ lists the references used in this document and notes any IPR claims.
- Section 8~~8~~ notes the contact for this document and has administrative information such as the document version history and list of acknowledgements.

1.3 Terminology Used In This Document

The terminology used in this document is covered in the Service Basic DDP BA and SD0-1..

In addition, the following terms are noted

- The "***requesting OS***" in this document is the OS that is using the interface. It is assumed that requesting OS has Customer Relationship Management (CRM) capabilities (at least enough intelligence to formulate the Service Activation Interface (SAI) requests).

2 Business Problem Description, Project Scope

2.1 Project Scope

The TM Forum Integration Program is responsible for all of the interface and business services work within the TM Forum. In some cases, interface work is delegated to other teams but the final verification for technical uniformity and integrity is the responsibility of the TM Forum Integration Program.

Initially, the TM Forum Integration Program was formed to coordinate the various existing TM Forum interfaces activities (as shown in Figure 2-1~~Figure 2-1~~). In particular, the responsibility for maintaining MTOSI and MTNM is now covered by the MTOSI-MTNM Users Group which is a team within the TM Forum Integration Program. The long term plan (which is already well under progress) is to migration the various input work to a single harmonized suite of interfaces.

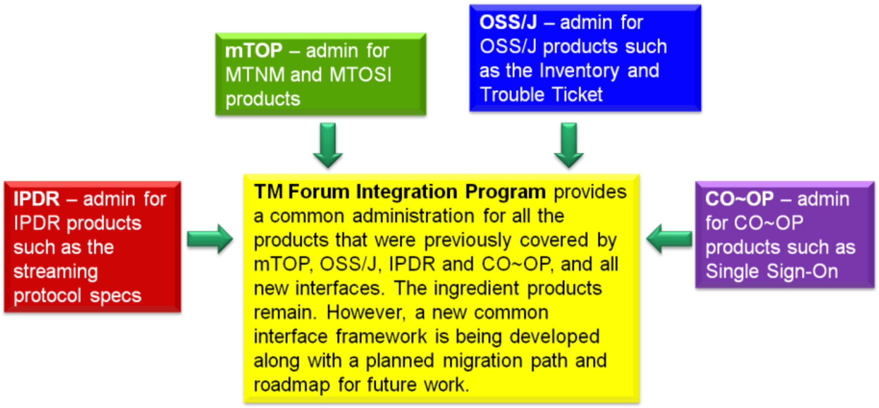

mTOP – admin for MTNM and MTOSI products
OSS/J – admin for OSS/J products such as the Inventory and Trouble Ticket
IPDR – admin for IPDR products such as the streaming protocol specs
TM Forum Integration Program provides a common administration for all the products that were previously covered by mTOP, OSS/J, IPDR and CO~OP, and all new interfaces. The ingredient products remain. However, a new common interface framework is being developed along with a planned migration path and roadmap for future work.
CO~OP – admin for CO~OP products such as Single Sign-On

Figure 2-1. Inputs to the TM Forum Integration Program

Figure 2-2~~Figure 2-2~~ provides a summary of the team within the TM Forum Integration Program as well as a few teams outside of the program but which also do some interface work. In terms of MTOSI and MTNM, the main input for updates come from the Resource and Service Management Team.

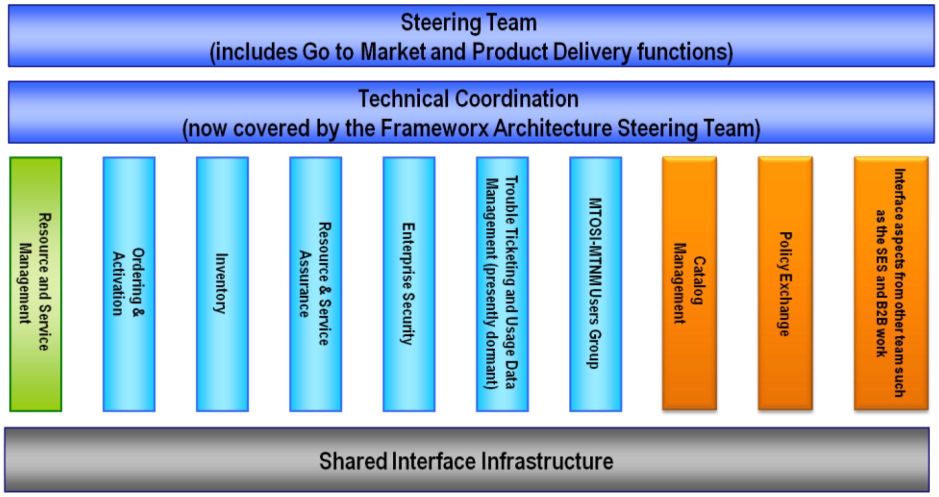

Steering Team
(includes Go to Market and Product Delivery functions)
Technical Coordination
(now covered by the Frameworx Architecture Steering Team)
Resource and Service Management
Ordering & Activation
Inventory
Resource & Service Assurance
Enterprise Security
Trouble Ticketing and Usage Data Management (presently dormant)
MTOSI-MTNM Users Group
Catalog Management
Policy Exchange
Interface aspects from other team such as the SES and B2B work
Shared Interface Infrastructure

Figure 2-2. TM Forum Integration Program

The Service Component Activation Interface, or SCAI, addresses the need to activate multiple components in an end-to-end service activation scenario in a consistent and simplified manner. Today's service activation solutions are often costly and complex due to the diversity of northbound OSS interfaces from underlying NMS/EMS software and the exposure of resource level data.

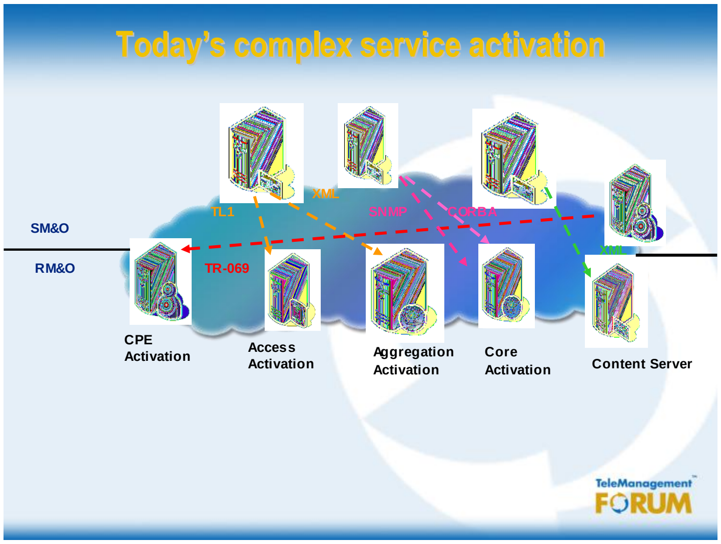


Figure 2-3~~2~~. Existing Service Activation Paradigm

It is not uncommon today for OSS Activation applications to have to understand the resource level details for every single component in the end-to-end activation process. The goal of this project is to provide a unified activation interface whereby the OSS Activation application needs only to have knowledge of the service definition, service templates and service access points as defined in the service design phase, in order to activate the end-to-end service instance and their individual service component instances. See TMF518_SA_1 for details of the various phases.

The scope is limited to the activation of the resources required for service delivery. No restrictions are placed on the types of services, nor the underlying technologies that are delivering the services.

2.2 Supported Business scenarios

The Service Component Activation Interface (SCAI) allows a traditional element or network management OS, or any other generic resource management OS, to provide an abstracted "service aware" interface to an SM&O layer OS. In doing so, it hides the details and complexity of the underlying resources, thus simplifying the activation of multiple components involved in an end-to-end service.

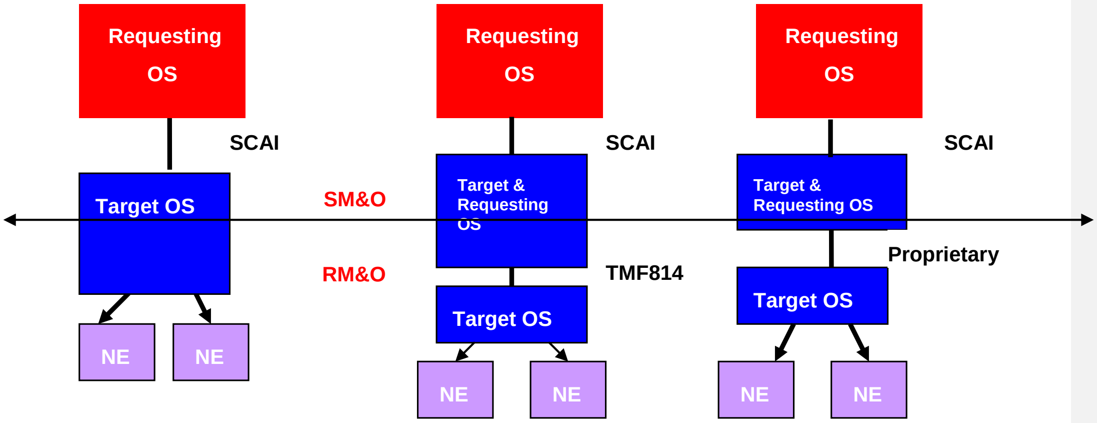
The interface supports static provisioned services whereby each component involved in the activation process has one or more "service access points" that serve as an entry point into the component to be activated. The notion of a "SAP" at the service component level (Resource Facing Service, RFS) differs from the SAP for an end-to end-service (Customer Facing Service, CFS) in that it may, or may not, be the actual point of entry to the service for a given user.

The Service Component Activation Interface provides an "abstracted", or "service aware" view of the activation of a "resource facing service". The SCAI only understands and manipulates service objects such as the RFS, and by reference, the Service Definition, the ServiceTemplate and the ServiceAccessPoints. How the service provider determines which services are to be "customer facing" and which are to be "resource facing" is not within the scope of this document. This interface merely provides the means, once a "resource facing service" is identified, to activate the service and associate it to a subscriber (optional) and a given service access point.

The target OS that processes the incoming service component activation request will either directly or indirectly, follow-up with the activation of the associated resources. This can be done in one of three different ways, and possibly more:

1. Direct activation of resources as a result of the request, in which case the target OS has both service provisioning and resource provisioning capabilities
2. Activate resources via a "standard" resource provisioning request to another OS, in which case the target OS provides service provisioning capabilities, and perhaps NMS capabilities
3. Via a proprietary interface to another EMS/NMS that will activate the resources.

These different scenarios are illustrated below:



2.3 Benefits

The Service Management SCAI is intended to provide the following benefits over and above those already cited in the Service Activation Overview document TMF518_SA_1:

- Simplification of the service configuration and activation overall process through
  - Reduction in number and type of disparate northbound OSS interfaces from the RM&O layer to the SM&O layer
  - Abstraction of the resource layer through the implementation of service templates that reference resources rather than the passing of actual resource values at service configuration and activation time
  - Consistent naming of resources visible over the SCAI through the use of a standardized model
- Reduction of OSS integration costs through
  - Reduction in number of systems to integrate with existing service provider OSS
  - Reduction of technical expertise and staff dedicated to integration of disparate systems
  - Repeatable integration process independent of technology and services requiring integration.

2.4 Mapping of Processes to eTOM Business Framework

The eTOM is a business process framework, or model, that provides the enterprise processes required for a service provider. There are various levels of granularity, or decomposition, of processes that have been defined. The MTOSI service management initiative has considered a Level 3 decomposition of the eTOM SM&O layer in order to fully evaluate the process flow for service configuration and activation.

The eTOM Level 3 decomposition takes the process called “Service Configuration & Activation” found at Level 2 and breaks it down into the following sub-processes:

- *Design Solution* – Develop an end-end specific service design which complies with a particular customer's requirement
- *Allocate Service Specific Parameters to Services* – Issue service identifiers for new services.
- *Track & Manage Service Provisioning* – Launch all the operational tasks needed to fix each solution requirement.
- *Implement, Configure & Activate Service* – Ensure service provisioning activities are assigned, managed and tracked efficiently.
- *Test Service End-to-End* – Test specific services to ensure all components are operating within normal parameters, and that the service is working to agreed performance levels
- *Issue Service Orders* – Issue correct and complete service orders
- *Report Service Provisioning* – Monitor the status of service orders, provide notifications of any changes and provide management reports.
- *Close Service Order* – Close a service order when the service provisioning activities have been completed
- *Recover Service* – Recover specific services that are no longer required by customers.

These processes are illustrated below in the context of their relationship with the CRM and the RM&O layers. The identified interfaces include: A) Service Activation interface, B) Resource Activation interface, C) Service Inventory interface, and D) Resource Inventory Interface. It is to be noted that the Service Component Activation Interface (SCAI) is not defined between any two layers of the eTOM, but is found with the SC&A processes (E)

Note: Not all Layer 3 processes are shown, in particular those that would not be part of the MTOSI-SA interfaces.

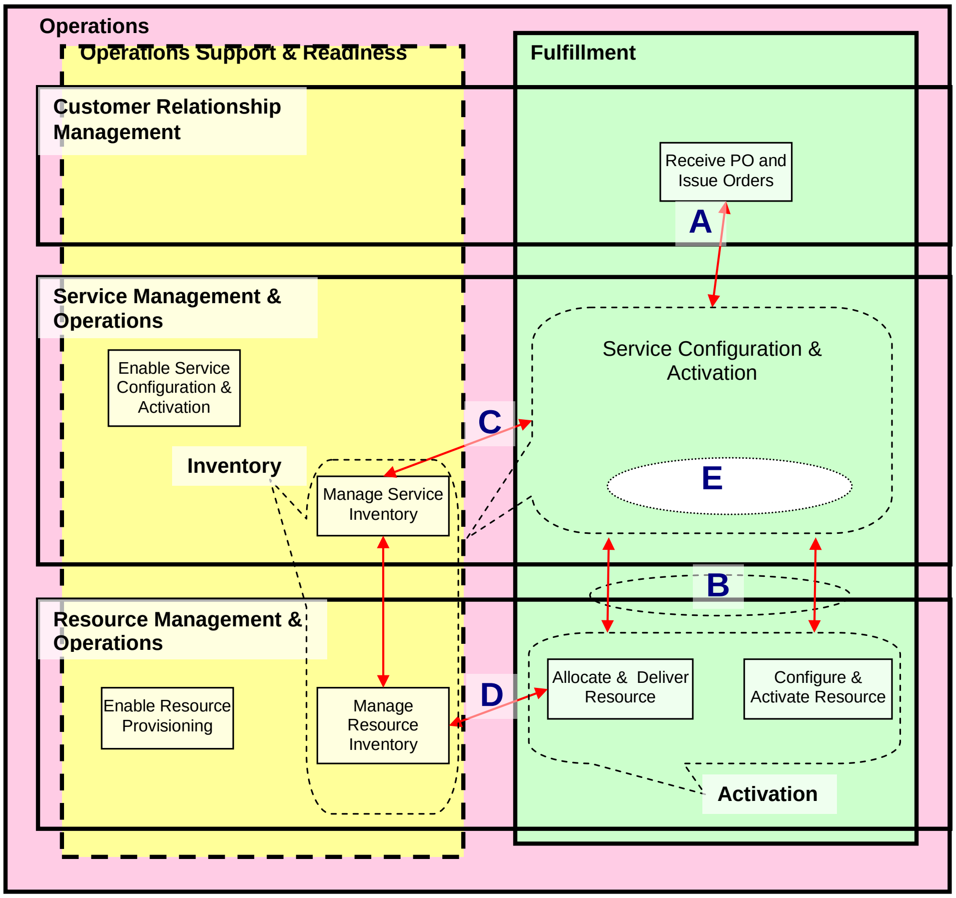

Operations
Operations Support & Readiness
Fulfillment
Customer Relationship Management
Receive PO and Issue Orders
A
Service Management & Operations
Enable Service Configuration & Activation
Service Configuration & Activation
C
Inventory
Manage Service Inventory
E
B
Resource Management & Operations
Enable Resource Provisioning
Manage Resource Inventory
D
Allocate & Deliver Resource
Configure & Activate Resource
Activation

Figure 2-~~4~~3. eTOM Laye3 Decomposition (Inventory and Activation)

provides a high-level mapping between the eTOM processes identified earlier in this section and the support of these processes in terms of the operations defined in this document.

Table 2-1. eTOM to SCAI Mapping

eTOM L3 Process	Support by Operations and Notifications in this document
Design Solution	N/A
Allocate Service Specific Parameters to Services	N/A.
Track & Manage Service Provisioning	This process is supported by the provision request and associated responses in

	R_TMF518_SA_3_II_0024
Implement, Configure & Activate Service	This process is supported by the reserve, provision, and activate, and modify requests in R_TMF518_SA_3_II_0023 R_TMF518_SA_3_II_0024 R_TMF518_SA_3_II_0025 R_TMF518_SA_3_II_0028
Test Service End-to-End	This component test is supported by the feasibility check R_TMF518_SA_3_II_0022
Issue Service Orders	N/A
Report Service Provisioning	This is specifically supported by service component activation events
Close Service Order	N/A
Recover Service	This process is supported by the terminate R_TMF518_SA_3_II_0027

3 Business Processes

3.1 Business Requirements

R_TMF518_SA_3_BR_0001	The Service Component Activation Interface (SCAI) shall support the activation of resource facing service components within the same service provider administrative domain.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_BR_0002	The Interface shall be technology and service agnostic and thus allow for implementation by any type of service using any kind of support technology. Example of such services include (amongst others): • Network Transport Services: DSL, MPLS VPN, E-Line, E-LAN, Frame Relay, and ATM • Signaled Services: Voice over IP (VoIP), Voice (Mobile and Fixed), Video on Demand (VoD) • Application Services : Email, SMS, Voice Mail, etc. Triple Play Services: VoIP, IPTV, and Internet Access.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_BR_0003	The SCAI shall support the activation of a resource facing service component that is: • Part of an end-to-end customer facing service managed by a service activation OS. • The sole RFS associated to a CFS • An RFS that is associated to multiple CFS
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_BR_0004	The SCAI shall provide template based activation whereupon service templates residing on the component activation OS, or target OS, are referenced by the service activation OS, or requesting OS.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_BR_0005	The SCAI shall support endpoint based activation whereby a *service access point* (SAP) that serves as the point of entry to the service component is passed. This access point is represented as a physical or logical resource.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_BR_0006	Subscriber and/or user related information may be passed to the service component at service activation time, but is not required.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_BR_0007	The SCAI shall support the processing of service operations based upon service related information from a higher SM&O layer OS. It acts upon resource facing services as a result of operations received.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_BR_0008	The SCAI shall optionally support a rollback mechanism such that the underlying resources supporting a given service component instance shall not be left in an undetermined state should an activation request be unsuccessful.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_BR_0009	The SCAI shall support synchronous operations. The synchronous mode is blocking, and the SCAI request must complete before the response is returned from the service component activation OS.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_BR_0010	The SCAI shall also support asynchronous operations. The asynchronous mode allows for the service component activation OS to initiate the activation request and immediately return a response indicating whether the initiation was successful. An event will be generated when the actual service component activation is completed.
Source	TMF518_SA_3 version 1.0

3.2 Category I: Static and Structural Requirements

Related static requirements can be found in TMF518_SB, Service Basic - DDP BA.

3.3 Category II: Normal Sequences, Dynamic Requirements

In the following requirements, the follow conventions are used:

- The OS sending a request over the Interface is called the “requesting OS” and the OS receiving the request is called the “target OS”.
- Unless stated otherwise, the term “Interface” refers to the Service Component Activation Interface (SCAI).

3.3.1 Service Component Activation

It should be noted that the Interface under study in this document, i.e., the Service Component Activation Interface (SCAI), is between a requesting OS that understands the concepts of the eTOM SC&A process and a target OS that also understands the concepts of the eTOM SM&O layer, but has either direct, or indirect knowledge of the RM&O layer. There are a number of options for implementing this interface. These include, but are not limited to the following:

- The target OS is a service component activation OS that then requests resource activation to an EMS/NMS
- The target OS is an NMS that exposes the service layer SCAI, but then requests resource activation to an EMS either via a TMF814 interface, or proprietary interface
- The target OS is an EMS that exposes the service layer SCAI, then directly activates the associated resources

R_TMF518_SA_3_II_0011	The Service Component Activation Interface shall support the following types of operations: 1. ***Feasibility***– the service component OS is requested to determine whether the necessary resources are available and sufficient for the instantiation of a given resource facing service. It should be noted that this is just an initial feasibility check and that no RFS is created. 2. **Reservation-** the service component OS is requested to reserve a resource, or set of resources, required for a *resource facing service*. This situation allows for the requesting OS to determine whether the underlying resources are available and reserve them in order to support a given RFS. As a result of the reservation request, an RFS is instantiated

	3. **Provision** – the service component OS is requested to allocate and configure all underlying resources that comprise a resource facing service. The RFS is instantiated as a result of this request, and is managed by the service component OS. When this request is complete, the RFS shall be in a Provisioned_Inactive state, and may be activated at a later time. 4. ***Activation*** – the service component OS is requested to activate a given resource facing service such that the component is fully available and active as part of the customer facing service. When this request is complete, the RFS shall be in a Provisioned_Active state. 5. ***Modification –*** the service component OS is requested to modify a given resource facing service. For example, the requesting OS may wish to add service access points and/or users, to change existing service characteristic values, or other modifications. 6. ***Deactivation*** – the resource facing service is deactivated and thus is no longer available for service. It remains allocated to the CFS that is managed by the higher level service activation OS. 7. ***Termination*** – the resource facing service is deleted and thus unallocated from the CFS. All associated resources are freed and made available for service to other users.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0012	The service states that may associated to an RFS or a SAP are: 1. Reserved – the resources required to instantiate the RFS have been reserved, but not provisioned and activated 2. Provisioned_Inactive – the resources required to instantiate the RFS have been allocated and configured, but not activated. The underlying resource facing service is not available for use by a CFS. 3. Provisioned_Active – the RFS instance is now available for use by a CFS.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0013	The Interface shall allow the requesting OS to request the **change of service characteristic values** in an RFS This is a specific type of modify request. For example, one could change the service characteristic value associated to the service characteristic “bandwidth” from 1M to 2M. After the change service characteristic values request is complete, the RFS is in the same state as before the request.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0014	The Interface shall allow the requesting OS to request the **addition of new service access points** in an RFS This is a specific type of modify request. For example, one could add a new site to a VPN, with the SAP representing the logical port on a router. After the addition of the new SAP is complete, the RFS is in the same state as before the request.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0015	The Interface shall allow the requesting OS to request the **deletion of service access points** from an RFS This is a specific type of modify request. For example, one could remove one or more sites from a VPN. After the removal of the SAPs is complete, the RFS is in the same state as before the request.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0016	The Interface shall allow the requesting OS to request the **replacement of service access point(s)** from an RFS After the removal of the SAPs is complete, the RFS is in the same state as before the request.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0017	The Interface shall allow the requesting OS to request the **activation of service access point(s)** in an RFS This is a specific type of modify request. For example, an RFS may be in a Provisioned_Active state, but a particular SAP has previously been deactivated. This operation allows for the SAP to be activated.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0018	The Interface shall allow the requesting OS to request the **deactivation of service access point(s)** in an RFS This is a specific type of modify request. It allows one to deactivate a given SAP or set of SAP(s) without deactivating the entire RFS.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0019	The Interface shall allow the requesting OS to request the **addition of a users** to an RFS This is a specific type of modify request. It allows one to add users to an existing resource facing service. Depending upon the component RFS, there may or may not be associated SAPs for the users. After the addition of users is complete, the RFS is in the same state as before the request.
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0020	The Interface shall allow the requesting OS to request the **deletion of users** from an RFS This is a specific type of modify request. It allows one to delete users from an existing resource facing service. After the addition of users is complete, the RFS is in the same state as before the request
Source	TMF518_SA_3 version 1.0

R_TMF518_SA_3_II_0021	The Service Component Activation interface shall send the following notifications, as defined in the MTOSI Framework, for clients that have autonomous notification detection capability and have subscribed to any of the following types of events: ○ *Object Creation Notification* When an RFS object is created ○ *Object Deletion Notification* When an RFS object is deleted ○ *Attribute Value Change Notification* When RFS attributes change values ○ *Service State Notification* When RFS serviceState or operationalState changes
Source	TMF518_SA_3 version 1.0

3.3.1.1 Detailed Behavior

This section provides detailed operations signatures for the various SCAI operations. It should be noted that these definitions are not intended to imply a specific Message Exchange Pattern (MEP).

R_TMF518_SA_3_II_0022	The **FeasibilityCheck** request shall convey the information and support the behavior described in **Table 3-1**~~**Table 3-1**~~.
Source	TMF518_SA_3, Version 1.0

Table 3-1. Operation Signature for FeasibilityCheck

Operation Signature	Request: This operation passes the necessary information to check if an “rfs” could be created given the associated serviceTemplate, subscriber, user, and SAP information. ***feasibilityCheck*** (rfsCheckData) Response: 1. ***Response*** (passed or failed) – this provides an indication of whether or not the requested RFS could be instantiated or not. In either case, it is just a check and no actual RFS is created. Exceptions: 1. In the event that the target OS in unable to execute the request, a standard exception is returned.
Behavior	This operation verifies the feasibility of provisioning and activating an RFS in support of a CFS. No RFS is actually created
Pre-conditions	The RFS does not exist
Post-conditions	In case of success: • The target OS sends “passed” back to the requesting OS In case of failure: • The target OS sends “failed” back to the requesting OS
Arguments	
Request	rfsCheckData – This information is the same as if one wanted to provision an RFS. The only difference is that the service component activation OS will only check to see if the RFS could actually be instantiated, but will not actually create it. See: resource facing service definition in TMF518_SB
Response	**Passed** – this is an indication that the necessary resources are available for allocation and configuration in order to support the desired resource facing service **Failed** – this is an indication that the feasibility check was not successful and that

	the necessary resources are not available, or not sufficient, for allocation and configuration of the RFS
Exceptions	The following exceptions are allowed: • Subset of basic exception (see R_TMF518_SA_3_III_0029) - invalidInput - unableToComply - communicationFailure - internalError

R_TMF518_SA_3_II_0023	The **reserve** request shall convey the information and support the behavior described in **Table 3-2**~~**Table 3-2**~~.
Source	TMF518_SA_2, Version 1.0

Table 3-2. Operation Signature for Reserve

Operation Signature	Request: ***reserve*** (rfsCreateData, expiringTime) Response: Choice of one of the following 1. ***rfsName*** – this provides the name of the RFS that has been created 2. ***rfsCreation***– provides the data relative to the creation of an RFS. Exceptions: 1. In the event that the target OS is unable to execute the request, an exception is returned.
Behavior	This operation allocates and reserves the necessary resources to support a given RFS for a given subscriber and associated user(s) until the "expiring time" is met. If the requesting OS does not provision and activate the RFS before the "expiration time" is met, the target OS will delete the RFS and free all the associated resources. If the target OS does not support the specification of an "expiring time", it may set an internal timer to ensure that un-provisioned RFS are cleaned up on a regular basis. Note: This differs from the SAI where it amends the reserve request to confirm the

	request to go ahead and provision the resources. In this case, the when the SAI receives the amend, or has direct confirmation for the provisioning request, it must in turn invoke the SCAI “provision” in order to bring the RFS out of a “reserve” state.
Pre-conditions	The RFS does not exist
Post-conditions	In case of success: • The RFS(s) is instantiated and placed in the “reserved” state In case of failure: • An exception is thrown In case of timer expiration: • RFS is deleted and any reserved and allocated resources are freed
Arguments	
Request	For the definitions of “rfsCreateData” information see Table **3-1**~~Table 3-1~~. expiringTime (optional) – the date and time by which an RFS that remains in a “reserved” state (i.e. it has not been provisioned and activated) shall be removed from the system.
Response	The target OS may return the name of the RFS that has been created or the object creation structure containing the entire RFS that was created
Exceptions	The following exceptions are allowed: • Subset of basic exception (see R_TMF518_SA_3_III_0029) - invalidInput - unableToComply - communicationFailure - internalError - objectInUse
Notifications	• Object Creation Notification (see R_TMF518_SA_3_III_0030)

Formatted: Font: Not Bold

R_TMF518_SA_3_II_0024	The **provision** request shall convey the information and support the behavior described in **Table 3-3**~~**Table 3-3**~~
Source	TMF518_SA_3, Version 1.0

Table 3-3. Operation Signature for Provision

Operation Signature	Request: ***provision*** (rfsCreateData or rfsName) Response: Choice of one of the following 1. **rfsName** – this provides the name of the RFS that has been provisioned 2. ***rfsCreation***– provides the data relative to the creation of an RFS. 3. ***rfsStateChange***– provides the data relative to a state change of an RFS. 4. ***rfsDeletion***– returned if the RFS previously existed (i.e. it was “reserved”), but could not be provisioned. Thus, as a result the RFS was deleted. Exceptions: 1. In the event that the target OS in unable to execute the request, an exception is returned.
Behavior	This operation provisions an RFS in support of a CFS When this request is complete, the RFS shall be in the Provisioned_Inactive state
Pre-conditions	The RFS may exist and be in a “reserved” state, in which case the resources that have been reserved will be provisioned accordingly. The RFS may not exist, in which case the resources will be allocated and provisioned.
Post-conditions	In case of success: • The RFS is in a Provisioned_Inactive state In case of failure: • An exception is thrown
Arguments	
Request	For the rfs info, see TMF518_SB
Response	See rfs info and notification structures
Exceptions	The following exceptions are allowed: • Basic exception (see R_TMF518_SA_3_III_0029)

Notifications	- Object Creation Notification (see R_TMF518_SA_3_III_0030). - State Change Notification (see R_TMF518_SA_3_III_0031) - Object Deletion Notification (see R_TMF518_SA_3_III_0032)

R_TMF518_SA_3_II_0025	The **activate** request shall convey the information and support the behavior described in **Table 3-4**~~**Table 3-4**~~
Source	TMF518_SA_3, Version 1.0

Table 3-4. Operation Signature for Activate

Operation Signature	Request: ***activate*** (rfsCreateData or rfsName) Response: Choice of one of the following 1. ***rfsName*** – this provides the name of the RFS that has been activated 2. ***rfsCreation***– provides the data relative to the creation of an RFS. 3. ***rfsStateChange***– provides the data relative to a state change of an RFS. Exceptions: 1. In the event that the target OS in unable to execute a synchronous request, an exception is returned.
Behavior	This operation activates an RFS in support of a CFS. If the RFS has not been previously provisioned, it will be done as part of this operation. If it has been provisioned, then the service component OS will activate the RFS. When this operation is complete, the RFS will be in a Provisioned_Active state
Pre-conditions	The RFS either does not exist, or is in a “Reserved” or “Provisioned_Inactive” state.
Post-conditions	In case of success: - The RFS shall be in a Provisioned_Active state. In case of failure: - An exception is thrown
Arguments	
Request	Either an rfsName of an rfs that has been previously “reserved” or “provisioned” or an rfs that is to be provisioned and activated.
Response	The name of the rfs that has been activated, or if in asynchronous mode an object creation or state change notification

Exceptions	The following exceptions are allowed: - Basic exception (see R_TMF518_SA_3_III_0029)
Notifications	- Object Creation Notification (see R_TMF518_SA_3_III_0030) - State Change Notification (see R_TMF518_SA_3_III_0031)

R_TMF518_SA_3_II_0026	The **deactivate** request shall convey the information and support the behavior described in **Table 3-5**~~**Table 3-5**~~
Source	TMF518_SA_3, Version 1.0

Table 3-5. Operation Signature for Deactivate

Operation Signature	***deactivate*** (rfsName) Response: Choice of one of the following 1. ***rfsName*** – this provides the name of the RFS that has been deactivated 2. ***rfsStateChange***– provides the data relative to a state change of an RFS.
Behavior	This operation deactivates the RFS which results in the RFS not be available for use When this request is complete, the RFS shall be in the Provisioned_Inactive state
Pre-conditions	The RFSs supporting a given CFS should be in the Provisioned_Active state.
Post-conditions	In case of success: - The RFS is in the Provisioned_Inactive state. In case of failure: - An exception is thrown
Arguments	
Request	rfsName – Name of the RFS to be deactivated
Response	rfsName – Name of the RFS that has been successfully deactived
Exceptions	The following exceptions are allowed: - Basic exception (see R_TMF518_SA_3_III_0029) - State Change Notification (see R_TMF518_SA_3_III_0031)

R_TMF518_SA_3_II_0027	The **terminate** request shall convey the information and support the behavior described in **Table 3-6**~~Table 3-6~~..
Source	TMF518_SA_3, Version 1.0

Table 3-6. Operation Signature for Terminate

Operation Signature	***Terminate*** (rfsName) Response: 1. ***Response*** (rfsName) – this provides the name of the RFS that has been terminated. Note: the RFS itself no longer exists Exceptions: 1. In the event that the target OS in unable to execute a request, an exception is returned.
Behavior	This operation deletes an RFS. It is no longer available for use
Pre-conditions	The RFSs supporting a given CFS may be in one of the following states: reserved, Provisioned_Inactive, or Provisioned_Active. Note: It is not recommended to “terminate” an RFS that is in a Provisioned_Active state, but it is not forbidden.
Post-conditions	In case of success: • The RFS is deleted In case of failure: • An exception is thrown
Arguments	
Request	rfsName – Name of the RFS to be terminated
Response	rfsName – Name of the RFS that has been successfully terminated
Exceptions	The following exceptions are allowed: • Basic exception (see R_TMF518_SA_3_III_0029) • Object Deletion Notification (see R_TMF518_SA_3_III_0032)

R_TMF518_SA_3_II_0028	The **modify** request shall convey the information and support the behavior described in **Table 3-7**~~**Table 3-7**~~
Source	TMF518_SA_3, Version 1.0

Table 3-7. Operation Signature for Modify

Operation Signature	Request: ***modify*** (rfs modify data,action, targetState) – this operation allows for modification of information relative to an RFS, including the description, userLabel, associated service characteristics, subscribers, users, and SAPs. Response: Choice of one of the following 1. ***rfsName*** – this provides the name of the RFS that has been deactivated ***serviceState*** -the service state of the RFS 2. ***rfsModification*** – this provides the RFS attributes that have changed Exceptions: 1. In the event that the target OS in unable to execute a request, an exception is returned.
Behavior	This operation modifies the attributes associated to an RFS. If successful, the target OS then transitions the RFS to the desired state as expressed in "targetState" For all subscriber, user, and SAP information, it is assumed that the relationship amongst them is maintained elsewhere.
Pre-conditions	The RFS exists and is in a Provisioned_Inactive state, though this is not mandatory.
Post-conditions	In case of success • The RFS will be modified per the request. In case of failure: • An exception is thrown
Arguments	
Request	targetState – this is the state to which the target OS wants the RFS transitioned when the request is complete action – This indicates whether the requested modification is a "modify", "add", "delete", "replace", "activate", or "deactivate"
Response	The attributes of the RFS that have been modified
Exceptions	The following exceptions and notifications are allowed:

	• Basic exception (see R_TMF518_SA_3_III_0029) • Attribute Value Change Notification (see R_TMF518_SA_3_III_0033)

3.4 Category III: Abnormal or Exception Conditions, Dynamic Requirements

3.4.1 Exceptions concerning Service Component Activation Requests

R_TMF518_SA_3_III_0029	The following types of standard MTOSI exceptions shall be supported by the service component activation interface: • entityNotFound– An action requested resulted in the target OS not being able to locate the associated entity • notImplemented – Raised when the target OS does not support this operation. • invalidInput – Raised if any input parameter is syntactically incorrect. • unableToComply – Raised when the target OS cannot fulfill the request. This can be used in either the initial or final response. • communicationFailure – Raised when the target OS is not able to communicate with one or more entities and as a consequence could not complete the rollback. • internalError – Raised when a target OS internal processing error occurs • objectInUse – Raised when an attempt to modify an object that is already in use for another request
Source	TMF518_SA_3, Version 1.0

R_TMF518_SA_3_III_0030	The ***Object Creation Notification*** is an indication that the target OS created an RFS The following parameters are supported: • commonEventInfo • objectType – RFS • objectName – Name of RFS • osTime – Time that the service component activation OS created the RFS • object – optional inclusion of the RFS object

Source	TMF518_SA_3, Version 1.0

R_TMF518_SA_3_III_0031	The ***State Change Notification*** is an indication that an attribute, or set of attributes, associated to an RFS have changed states The following parameters are supported: • commonEventInfo • objectType – RFS • objectName – Name of RFS • osTime – Time that the service component activation OS created the RFS • attributeList– New values of state attributes that have changed
Source	TMF518_SA_3, Version 1.0

R_TMF518_SA_3_III_0032	The ***Object Deletion Notification*** is an indication that the target OS deleted an RFS The following parameters are supported: • commonEventInfo • objectType – RFS • objectName – Name of RFS • osTime – Time that the service component activation OS created the RFS • object – optional inclusion of the RFS object deleted
Source	TMF518_SA_3, Version 1.0

R_TMF518_SA_3_III_0033	The ***Attribute Value Change Notification*** is an indication that an attribute, or set of attributes, associated to an RFS have changed. The following parameters are supported: • commonEventInfo • attributeList This container encapsulates the object element populated with the attributes whose values have changed.
Source	TMF518_SA_3, Version 1.0

3.5 Category IV: Expectations and Non-Functional Requirements

None have been identified thus far

3.6 Category V: System Administration Rquirements

None have been identified thus far

4 Use Cases

4.1 Service Component Activation

4.1.1 Feasibility Check

Use Case Id	UC_TMF518_SA_3_0001
Use Case Name	Feasibility Check
Summary	A requesting OS would like to determine whether it would be possible, or "feasible" to provision an RFS for a given subscriber/user.
Actor(s)	Requesting OS
Pre-Conditions	The requesting OS will have retrieved the associated service design and service templates from the Service Inventory.
Begins When	The use case begins when the requesting OS sends a feasibility check request.
Description	1. The requesting OS sends a feasibility check request to the target OS. The service template and the associated list of subscribers, users, and service access points provided in the request identify (either directly, or by reference) what underlying resources may require configuration and/or need to be associated to the component service instance. Such resources may include modems, ports, VPI/VCI, VLANS, profiles etc. 2. If the request is valid, the target OS proceeds to check if it is possible to fulfill the request by verifying whether the resources required are available and sufficient. 3. The target OS responds to the server OS
Ends When	In case of success: The target OS returns the value "passed" In case of failure: The target OS returns the value "failed" if the operation was not successful, or an exception if the operation could not be processed
Post-Conditions	No RFS is created as a result of this request, and the associated resources requested are not changed.
Exceptions	basic exception (see R_TMF518_SA_3_III_0029)
Traceability	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0022 R_TMF518_SA_3_III_0029

4.1.2 Reserve Service

Use Case Id	UC_TMF518_SA_3_0002
Use Case Name	Reserve Service
Summary	A requesting OS would to "reserve" a resource facing service for a given subscriber/user until a certain point in time
Actor(s)	Requesting OS
Pre-Conditions	The requesting OS will have retrieved the associated service design and service templates from the Service Inventory. The requesting OS may, or may not, have previously issued a "feasibility check" for the service
Begins When	The use case begins when the requesting OS sends a reserve service request.
Description	1. The requesting OS sends a reserve service request to the target OS. The service template, and any additional service characteristics specify (either directly, or indirectly through reference) that need to be reserved for a given subscriber/user and associated service access points. 2. The target OS validates the request 3. If the request is valid, the target OS proceeds to check if it is possible to fulfill the request by verifying whether the resources required are available and sufficient. 4. If the target OS is able to honor the reserve request, it then: ➢ Creates an RFS using the attributes as specified in the request to allocate and reserve the resources ➢ If no "expiringTime" is specified on the request, or the target OS doesn't support the configuration of such a timer, the target OS may do one of two things: a. set an internal timer according to a pre-defined rule b. set no timer value, in which case the RFS will continue to exist until it is provisioned and/or terminated ➢ If the target OS does support the "expiringTime", it sets the timer as requested. ➢ The RFS will continue to exist until it is provisioned (and service characteristics may be changed at the actual provisioning time), or the timer goes off. ➢ If the timer expires and the RFS has not been provisioned, the target OS will delete the RFS. ➢ If the target OS supports asynchronous notifications, it will send an *Object Creation Notification*

	The target OS responds to the server OS
Ends When	In case of success: The target OS returns the name of the RFS created. *Object Creation Notification* is sent In case of failure: The target OS returns an exception. In the event it can process the request, but is unable to reserve the resources necessary to fulfill the request, an "UNABLE_TO_COMPLY" exception is returned.
Post-Conditions	An RFS is created and is put into a state "Reserved". The associated resources are allocated and reserved for the given subscriber/user If a timer has been specified, and the requested time lapses without further operations on the RFS, it will then be deleted by the target OS
Exceptions	basic exception (see R_TMF518_SA_3_III_0029)
Notifications	Object Creation Notification (see R_TMF518_SA_3_III_0030)
Traceability	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0023 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0032

4.1.3 Provision

Use Case Id	UC_TMF518_SA_3_0003
Use Case Name	Provision Service
Summary	A requesting OS would like to provision a resource facing service that is a component in an end-to-end service. The target OS will ensure that all resources required to deliver the RFS are configured.
Actor(s)	Requesting OS
Pre-Conditions	The requesting OS will have retrieved the associated service design and service templates from the Service Inventory. The requesting OS may have previously issued a feasibility check and/or a reserve request for the RFS If the RFS has been previously reserved, this request will proceed with the configuration of the associated resources
Begins When	The use case begins when the requesting OS sends a provision service request.

Description	1. The requesting OS sends a provision service request to the target OS. 2. The target OS validates the request. 3. If the request is valid, the target OS proceeds to fulfill the provision request: ➢ If the RFS exists and is in the reserved state, the target OS will proceed with the provisioning of the resources if required (i.e. resources required may have changed since the initial reservation) ➢ If the RFS does not exist, the target OS will proceed with the provisioning of the resources required and instantiate the associated RFS ➢ The RFS is then marked "Provisioned_Inactive" if the request is successfully fulfilled. ➢ If the target OS supports asynchronous notifications, it will send an Object Creation Notification. if an RFS has been instantiated. If the RFS already existed and was "reserved", it sends a State Change Notification 4. If the request is valid, but the target OS is not successful in the provision request, it may roll-back resources to their previous state if such a mechanism is supported. 5. The target OS responds to the server OS
Ends When	In case of success: The server OS returns the name of the RFS that has been provisioned In case of failure: The server OS sends an exception indicating the reason for failure. In the event it can process the request, but is unable to provision the resources necessary to fulfill the request, an "UNABLE_TO_COMPLY" exception is returned. If the RFS had been in a reserved state, and the provisioning request is not successful, then the RFS is deleted and an object deletion notification is sent.
Post-Conditions	An RFS is in the "Provisioned_Inactive" state on the target OS In the case of failure, the resources may or may not be roll-backed, depending on the capabilities of the server OS. If the RFS had previously been in a "Reserved" state, and the provisioning was unsuccessful, the RFS is deleted
Exceptions	basic exception (see R_TMF518_SA_3_III_0029)
Notifications	Object Creation Notification (see R_TMF518_SA_3_III_0030) State Change Notification (see R_TMF518_SA_3_III_0031)

	Object Deletion Notification (see R_TMF518_SA_3_III_0032
Traceability	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0024 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0031 R_TMF518_SA_3_III_0032

4.1.4 Activate Service

Use Case Id	UC_TMF518_SA_3_0004
Use Case Name	Activate Service
Summary	A requesting OS would like to activate a resource facing service that is a component in an end-to-end service. The target OS will ensure that all resources required to deliver the RFS are configured and activated.
Actor(s)	Requesting OS
Pre-Conditions	The requesting OS will have retrieved the associated service design and service templates from the Service Inventory. A requesting OS may have previously done a feasibility check, reserve, and/or provision request for the RFS.
Begins When	The use case begins when the requesting OS sends an activate service request.
Description	1. The requesting OS sends a request to the target OS to activate a resource facing service. 2. The target OS validates the request. 3. If the request is valid, the target OS proceeds to fulfill the activate request: ➢ If the RFS exists and is in the “reserved” state, the target OS will proceed with the provisioning and activation. ➢ If the RFS exists and is in the “Provisioned_Inactive” state, it proceeds with the activation of the RFS ➢ If the RFS does not exist, the target OS will proceed with the provisioning of the resources required and instantiate the associated RFS ➢ The RFS is then marked “Provisioned_Active” if the request is successfully fulfilled. ➢ If the target OS supports asynchronous notifications, it will send an rfsCreationEvent if an RFS has been instantiated. If the RFS already existed and was in a state of “Provisioned_Inactive”, it sends an State Change Notification

	4. If the request is valid, the RFS did not previously exist, and the target OS is not successful in the activate request, it may roll-back resources to their previous state if such a mechanism is supported. 5. The target OS responds to the server OS
Ends When	In case of success: The server OS returns the name of the RFS that it has been activated In case of failure: The server OS sends an exception indicating the reason for failure. In the event it can process the request, but is unable to provision the resources necessary to fulfill the request, an “UNABLE_TO_COMPLY” exception is returned.
Post-Conditions	In the case of success, the RFS is activated on the service component OS. The resource facing service is considered to be ready and available for service In the case of failure, the RFS is left is its previous state (Reserved, or “Provisioned_Inactive”)
Exceptions	basic exception (see R_TMF518_SA_3_III_0029)
Notifications	Object Creation Notification(see R_TMF518_SA_3_III_0030) State Change Notification (see R_TMF518_SA_3_III_0031))
Traceability	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0025 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0031

4.1.5 Deactivate Service

Use Case Id	UC_TMF518_SA_3_0005
Use Case Name	Deactivate Service
Summary	A requesting OS would like to deactivate a resource facing service.
Actor(s)	Requesting OS
Pre-Conditions	The requested RFS to be deactivated exists
Begins When	The use case begins when the requesting OS sends a request to

	deactivate a service
Description	1. The requesting OS sends a request to deactivate a service to the target OS. 2. The target OS validates the request. 3. The target OS determines the "serviceState" of the RFS to be deactivated. If it is not in a "Provisioned_Active" state, it returns an error 4. The target OS proceeds to fulfill the deactivate request: ➢ The RFS is deactivated and the serviceState is changed to "Provisioned_Inactive" ➢ If the target OS supports asynchronous notifications, it will send an State Change Notification The target OS responds to the server OS
Ends When	In case of success: The server OS returns the name of the RFS that was deactivated In case of failure: The server OS sends an exception indicating the reason for failure. In the event it can process the request, but is unable to provision the resources necessary to fulfill the request, an "UNABLE_TO_COMPLY" exception is returned.
Post-Conditions	In the case of success, the RFS is in a "Provisioned_Inactive" state If the target OS is unable to deactivate the RFS, it remains in the same state as before the operation was requested (i.e. "Provisioned_Active")
Exceptions	basic exception (see R_TMF518_SA_3_III_0029)
Notifications	State Change Notification (see R_TMF518_SA_3_III_0031)
Traceability	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0026 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0031

4.1.6 Terminate Service

Use Case Id	UC_TMF518_SA_3_0006
Use Case Name	Terminate Service
Summary	A requesting OS would like to terminate a resource facing service.

Actor(s)	Requesting OS
Pre-Conditions	The requested RFS to be terminated exists and is in a “Reserved” , “Provisioned_Inactive” , or “Provisioned_Active” state. Note: it is recommended not to terminate an RFS in a “Provisioned_Active” state, but not forbidden.
Begins When	The use case begins when the requesting OS sends a request to terminate a service
Description	1. The requesting OS sends a request to terminate a service to the target OS. 2. The target OS validates the request 3. The target OS proceeds to fulfill the terminate request: ➢ Some resources may be un-configured and freed for use at a future time (i.e. a port), while some may stay configured and active for other users (i.e. a VLAN) ➢ The RFS is then deleted ➢ If the target OS supports asynchronous notifications, it will send an Object Deletion Notification The target OS responds to the server OS
Ends When	In case of success: The server OS returns the name of the RFS that was deleted In case of failure: The server OS sends an exception indicating the reason for failure. In the event it can process the request, but is unable to provision the resources necessary to fulfill the request, an “UNABLE_TO_COMPLY” exception is returned.
Post-Conditions	The RFS no longer exists on the target OS If the target OS is unable to terminate the RFS, it remains in the same state as before the operation was requested
Exceptions	basic exception (see R_TMF518_SA_3_III_0029)
Notifications	Object Deletion Notification R_TMF518_SA_3_III_0032
Traceability	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0027 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0032

4.1.7 Modify Service

Use Case Id	UC_TMF518_SA_3_0007
Use Case Name	Modify Service
Summary	A requesting OS would like to modify an RFS
Actor(s)	Requesting OS
Pre-Conditions	1. The requesting OS will have retrieved the associated service design and service templates from the Service Inventory 2. The requesting OS will have previously provisioned the RFS. The RFS should be in a Provisioned_Inactive state in order for a modification to occur without possibly a service interrupt. However, this is not mandatory.
Begins When	The use case begins when the requesting OS sends a modify request.
Description	1. The requesting OS sends a modify service request, providing the service characteristics to be changed 2. The target OS validates the request 3. If the request is invalid, an exception is returned 4. At this point, the target OS examines the "action" field in the request: *Case*: "Modify" ➢ This is the case where existing service attributes and/or characteristic values are to be modified. In this case, the RFS should be in a state "Provisioned_Inactive". The service characteristics may have been originally defined in the template. This operation does not affect the template definition, only the values associated to a particular service instance. The new values would be specified in the "described by" list. *Case*: "Add" ➢ This is the case where the request is to add a user(s) and/or SAP(s) to the existing RFS. The list of users and/or SAPS to be added are found in the user reference list and SAP reference list. Existing SAPs don't have to be listed. It is assumed by default that there is a one-to-one correspondence between user/sap, but this is not imposed. *Case*: "Delete" ➢ This is the case where the request is to delete a user(s) and/or SAP(s) to the existing RFS. Note: In order to delete SAPs, the serviceState of each SAP to be deleted must be "Provisioned_Inactive". The list of users and/or SAPs to be deleted are specified in the user reference and SAP reference lists.

	Case: “Replace” ➢ This is the case where the request is to replace a SAP(s) on the existing RFS. In this case, the sapList should be composed of SAP(s) as follows: sapRefList = {oldSAP1, newSAP1….oldSAPx, newSAPx} Note: In order to replace SAPs, the serviceState of each SAP to be replaced must be “Provisioned_Inactive”. Likewise, the user reference list can be updated *Case*: “Activate” ➢ This is the case where the request is to activate a SAP(s) on an existing RFS. This is different from the “activate” service request in that the service may be already active, but a SAP or set of SAPs may have been individually deactivated. Thus, this would allow for their activation. The SAP(s) to be activated are in the sapRefList Note: The RFS must be in a “Provisioned_Active” state in order to activate individual SAP(s) on the RFS. The RFS SAP state should in theory be “Provisioned_Inactive” in order to be activated, but this is not mandatory. *Case*: “Deactivate” ➢ This is the case where the request is to deactivate a SAP(s) on an existing RFS. This is different from the “deactivate” service request in that one can “deactivate” a SAP, or set of SAPs on an RFS, while the RFS remains in an activated state. Note: The RFS must be in a “Provisioned_Active” state or “Provisioned_Inactive” state in order to deactivate individual SAP(s) on the RFS. The RFS SAP state should in theory be “Provisioned_Active” in order to be deactivated, but this is not mandatory. 5. Depending on the case specified above, the target OS determines whether it can fulfill the request. If so, the RFS is modified as a result. 6. If the request is valid, and the target OS is not successful in the modify request, it may roll-back resources to their previous state if such a mechanism is supported. 7. If the target OS supports asynchronous notifications, it will send an Attribute Value Change Notification In the event that any existing attributes of the RFS have been changed. 8. The target OS then responds to the server OS
Ends When	In case of success: The target OS returns the name of the resource facing service that has been successfully modified with the attributes that have been successfully updated

	In case of failure: The server OS sends an exception indicating the reason for failure. In the event it can process the request, but is unable to provision the resources necessary to fulfill the request, an "UNABLE_TO_COMPLY" exception is returned.
Post-Conditions	In the case of success, the service attributes that were requested to be modified are changed in the RFS. The service is then transitioned to the desired target state. If this transition is unsuccessful, the RFS remains in its previous state. If the target OS is unable to modify the resource facing service, the RFS and all its attributes remain in the same state
Exceptions	basic exception (see R_TMF518_SA_3_III_0029)
Notifications	attribute value change event (see R_TMF518_SA_3_III_0033)
Traceability	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0028 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0033

5 Traceability Matrices

Table 5-1. Use Cases – Requirements Traceability Matrix

Use Case Id	Use Case Name	Requirements
UC_TMF518_SA_3_0001	Feasibility Check	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0022 R_TMF518_SA_3_III_0029
UC_TMF518_SA_3_0002	Reserve Service	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0023 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0032
UC_TMF518_SA_3_0003	Provision Service	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0024 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0031 R_TMF518_SA_3_III_0032
UC_TMF518_SA_3_0004	Activate Service	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0025 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0031
UC_TMF518_SA_3_0005	Deactivate Service	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0026 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0031

UC_TMF518_SA_3_0006	Terminate Service	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0027 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0032
UC_TMF518_SA_3_0007	Modify Service	R_TMF518_SA_3_II_0011 R_TMF518_SA_3_II_0012 R_TMF518_SA_3_II_0028 R_TMF518_SA_3_III_0029 R_TMF518_SA_3_III_0030 R_TMF518_SA_3_III_0033

Table 5-2. Requirements – Use Cases Traceability Matrix

Requirement Id	**Use Case Name**	**Use Case Id**
R_TMF518_SA_3_BR_0001	Activate Service	UC_TMF518_SA_3_0004
R_TMF518_SA_3_BR_0002	All use cases	
R_TMF518_SA_3_BR_0003	All user cases	
R_TMF518_SA_3_BR_0004	Activate Service	UC_TMF518_SA_3_0004
R_TMF518_SA_3_BR_0005	Activate Service	UC_TMF518_SA_3_0004
R_TMF518_SA_3_BR_0006	Activate Service	UC_TMF518_SA_3_0004
R_TMF518_SA_3_BR_0007	All use cases	
R_TMF518_SA_3_BR_0008	Provision Service Activate Service Modify Service	UC_TMF518_SA_3_0003 UC_TMF518_SA_3_0004 UC_TMF518_SA_3_0007
R_TMF518_SA_3_BR_0009	All use cases	
R_TMF518_SA_3_BR_0010	All use cases	

R_TMF518_SA_3_II_0011	Modify Service Terminate Service Deactivate Service Activate Service Provision Service Reserve Service Feasibility Check	UC_TMF518_SA_3_0007 UC_TMF518_SA_3_0006 UC_TMF518_SA_3_0005 UC_TMF518_SA_3_0004 UC_TMF518_SA_3_0003 UC_TMF518_SA_3_0002 UC_TMF518_SA_3_0001
R_TMF518_SA_3_II_0012	Activate Service Reserve Service Provision Service Activate Service Deactivate Service	UC_TMF518_SA_3_0004 UC_TMF518_SA_3_0002 UC_TMF518_SA_3_0003 UC_TMF518_SA_3_0004 UC_TMF518_SA_3_0005
R_TMF518_SA_3_II_0013	Modify Service	UC_TMF518_SA_3_0003
R_TMF518_SA_3_II_0014	Modify Service	UC_TMF518_SA_3_0003
R_TMF518_SA_3_II_0015	Modify Service	UC_TMF518_SA_3_0003
R_TMF518_SA_3_II_0016	Modify Service	UC_TMF518_SA_3_0003
R_TMF518_SA_3_II_0017	Modify Service	UC_TMF518_SA_3_0003
R_TMF518_SA_3_II_0018	Modify Service	UC_TMF518_SA_3_0003
R_TMF518_SA_3_II_0019	Modify Service	UC_TMF518_SA_3_0003
R_TMF518_SA_3_II_0020	Modify Service	UC_TMF518_SA_3_0003
R_TMF518_SA_3_II_0021	Modify Service Terminate Service Deactivate Service Activate Service Provision Service Reserve Service	UC_TMF518_SA_3_0007 UC_TMF518_SA_3_0006 UC_TMF518_SA_3_0005 UC_TMF518_SA_3_0004 UC_TMF518_SA_3_0003 UC_TMF518_SA_3_0002
R_TMF518_SA_3_II_0022	Feasibility Check	UC_TMF518_SA_3_0001
R_TMF518_SA_3_II_0023	Reserve Service	UC_TMF518_SA_3_0002
R_TMF518_SA_3_II_0024	Provision Service	UC_TMF518_SA_3_0003
R_TMF518_SA_3_II_0025	Activate Service	UC_TMF518_SA_3_0004
R_TMF518_SA_3_II_0026	Deactivate Service	UC_TMF518_SA_3_0005
R_TMF518_SA_3_II_0027	Terminate Service	UC_TMF518_SA_3_0006
R_TMF518_SA_3_II_0028	Modify Service	UC_TMF518_SA_3_0007
R_TMF518_SA_3_III_0029	All Use Cases	

R_TMF518_SA_3_III_0030	Reserve Service Provision Service	UC_TMF518_SA_3_0002 UC_TMF518_SA_3_0003
R_TMF518_SA_3_III_0031	Activate Service Deactivate Service Provision Service	UC_TMF518_SA_3_0004 UC_TMF518_SA_3_0005 UC_TMF518_SA_3_0003
R_TMF518_SA_3_III_0032	Terminate Service Provision Service	UC_TMF518_SA_3_0006 UC_TMF518_SA_3_0003
R_TMF518_SA_3_III_0033	Modify Service	UC_TMF518_SA_3_0007

6 Future Directions

The following areas have been identified for future study in relation to the SCAI:

- The existing interface has been positioned primarily as an “extension upwards” from the RM&O layer, and thus providing the ability to abstract resources. A request has been made to have the equivalent of a SCAI interface, but more as a peer-to-peer with another activation OS. In theory, this is possible, but further study is required
- The SCAI has been developed primarily with “static, provisioned” services in mind. That is to say, services are being activated in relationship to a direct request from a CRM to the SAI, and then the SCAI. There are relatively few components involved in the activation process, and the associated subscriber/user data is also minimal.
- The relationship between policy management and the use of “service templates” that are essentially a logical grouping of “service characteristics” that could be considered to be a basic policy.
- The relationship between TISPAN SuM and the configuration of service and user profiles to MTOSI-SM and the configuration of subscriber related information.
- A closer examination of the “modify service” functionality as related to the MTOSI MART feature

7 References

7.1 References

[1] TMF518_FMW, Framework DDP-BA

[2] TMF518_SB, Service Basic DDP-BA

[3] TMF518_SA_1, Service Activation DDP-BA, Overview

[4] TMF518_SA_2, Service Activation DDP-BA, Service Activation Interface

[5] SD0-1, Dictionary

7.2 IPR Releases and Patent Disclosure

There are no known IPR claims on the material in this document. As per the TM Forum bylaws, any TM Forum member company that has IPR claims on this or any TM Forum specification needs to make the claims known to the TM Forum membership immediately.

8 Administrative Appendix

This Appendix provides additional background material about the TM Forum and this document.

8.1 About this document

This document has been generated from the SD0-3_Template_BA.dot Word template.

8.2 Use and Extension of a TM Forum Business Agreement

This document defines the business problem and requirement model for <<problem area>>. The Business Agreement is used to gain consensus on the business requirements for exchanging information among processes and systems in order to solve a specific business problem. The Business Agreement should feed the development of Information Agreement(s), which is a technology-neutral model of one or more interfaces. While the Business Agreement contains sufficient information to be a “stand alone” document, it is better read together with the Information Agreement document <<(TMF <<number>> if available)>> when the Information Agreement is available. Reviewing the two documents together helps in gaining a full understanding of how the technology neutral information model solution is defined for this requirement model. An initial Business Agreement may only deal with a subset of the requirements. It is acceptable for subsequent issues of the document to add additional requirements not addressed by earlier releases of the BA. Business Agreements are the basis for requirement traceability for information models.

It is expected that this document will be used:

- As the foundation for a TM Forum Information Agreement(s)
- To facilitate requirement agreement between Service Providers and vendors
- As input to a service Provider’s Request for Information / Request for Proposal (RFI/RFP—RFX)
- As input for vendors developing COTS products
- As a source of requirements for other bodies working in this area

8.3 Document History

Version	Date Modified	Description of changes
1.0	October 2007	This is the first version of the document and as such, there are no changes to report.
1.1	May 2008	Consolidation based on team discussions and review comments
1.2	June 2011	Updated sections 1.1 and 2. Replaced mTOP by MTNM / MTOSI everywhere in the document

8.4 Company Contact Details

Company	Team Member Representative
Ciena	*Name:* Jessie Jewitt *Email:* JJewitt@ciena.com *Phone:* +1 650 714-2593

8.5 Acknowledgments

This document was prepared by the members of the TM Forum MTOSI Service Management team

- Jessie Jewitt Ciena, **Editor**
- Michel Besson Cramer/Amdocs, Project Manager
- Shlomo Cwang Cramer/Amdocs MTOSI Service Management Leader
- Steve Fratini Telcordia MTOSI SAI

Additional input was provided by the following people:

- Giuseppe Ricucci Telecom Italia