Resource Provisioning - DDP BA

TMF518_RP

Version 1.2

September, 2011

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

List of Use Cases

List of Figures

List of Tables

Executive Summary

This document entails the Business Agreement (BA) aspect of the MTNM / MTOSI Resource Provisioning (RP) Document Delivery Package (DDP). As it name indicates, it covers requirements and use cases concerning the provisioning of network resources.

The following management capabilities are covered:

- Connection Control
- Equipment Provisioning
- Flow Domain Control
- Gui Cut-Through Control
- Software and Data Control
- Termination Point Control
- Transmission Descriptor Control
- Assignment of Transmission Descriptors
- Topological Link Control

This document generalizes and extends the resource provisioning requirements and use cases from TMF 513 v3.0/v3.1. TMF 513 focuses exclusively on the NML-EML interface. However, this document considers the more general scenario of OS-OS communications with NML-EML as a special case

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

The following sections are included in this document:

- Section 1 is this introduction.
- Section 2 defines the business problem and project scope

- Section 3 has the requirements and associated descriptive text.
- Section 4 contains the use cases.
- Section 5 has traceability matrices between the use cases and the requirements.
- Section 6 provides a list of open issues to be considered in later versions of this document.
- Section 7 lists references and states IPR claims, if any.
- Section 8 provides administrative details such as document history and acknowledgements.

1.3 Terminology Used In This Document

Many of the object types used in this document are defined in the associated BAs for the NRB DM-DDP (TMF518_NRB) and NRF DM-DDP (TMF518_NRF).
For other terms refer to the SD0-1 supporting document.

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**). 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.

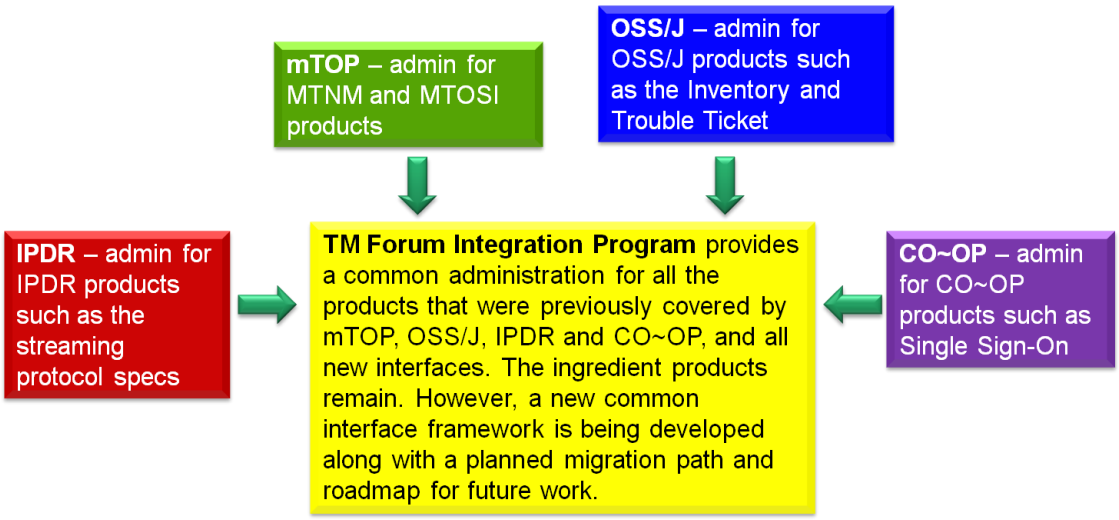

Figure 2-1. Inputs to the TM Forum Integration Program

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.

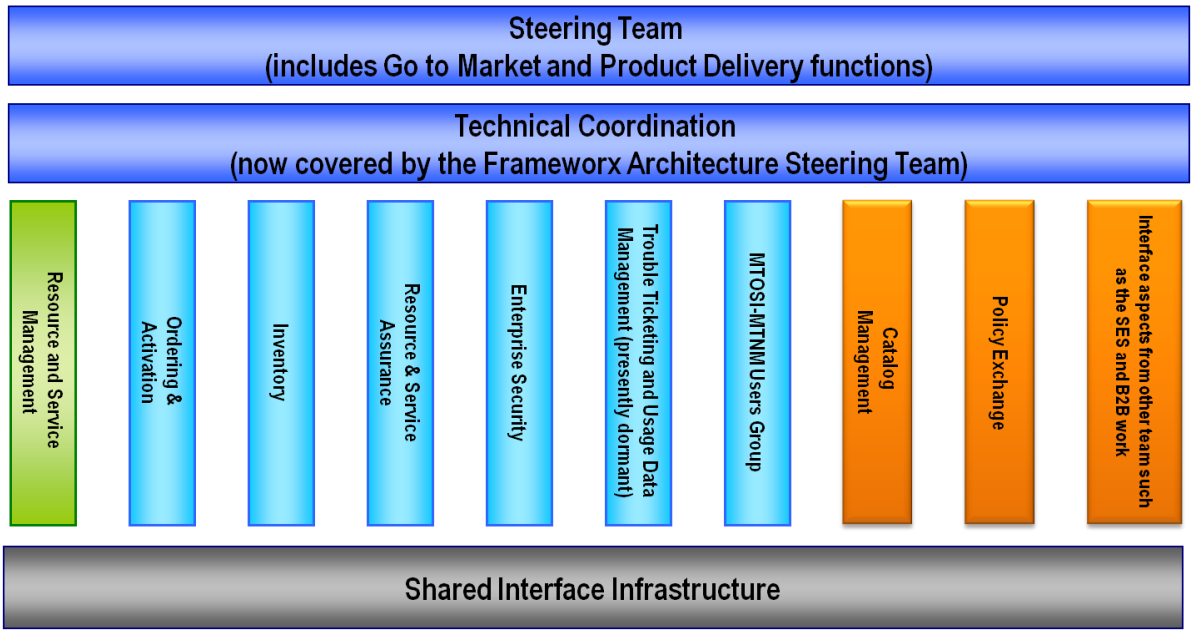

Figure 2-2. TM Forum Integration Program

2.2 Benefits

MTOSI and MTNM provide a set of Interface specifications that allow for resource and service management (with only MTOSI covering service management, but with MTOSI and MTNM both covering resource management, using very much the same information model).

These specifications are intended to lower design, implementation, Verification Validation & Testing (VVT), and maintenance costs for management interfaces. These Interfaces are intended for use by service providers, suppliers of equipment and OSS suppliers. The intention is to also encourage system integrator usage of management systems that make use of the Interfaces.

In particular, the followed approach tends to minimize the cost of integration, provide access to all necessary information and control, and support all vendor/operator differentiation. The intent of the interface is to provide compatibility among different version, for a detailed description see SD2-6_VersioningAndExtensibility.

2.2.1 Service Provider Benefits

The service provider benefits are as follows:

- One stop shopping concerning feature requests for much of the TM Forum contract specification work is part of the defined Change Control Group (CCG) process that TM Forum makes available in order to control the interface.
- The technical deliverables are also of high value to the service provider. The Interface specifications allow for an open, multi-supplier environment, shorten delivery times and lower integration costs.

- The MTOSI and MTNM products provide an integrated, multi-technology interface with support for most key layer 1 and layer 2 transport technologies. This is in contrast to earlier approaches where each technology-specific forum provided a single-technology management interface. The service provider was faced with having to use many different, uncoordinated management interfaces.
- These products are not bound to any one middleware, transport or computing language. So, the service provider will be able to evolve to new technologies as they arise.

2.2.2 Supplier Benefits

The supplier benefits are as follows:

- Fewer Adapters leads to Lower Costs – in as much as MTOSI and MTNM gain market penetration (and there has already been significant market acceptance of these interfaces), the supplier is faced with the need to build fewer adapters between their products and the products of their partners. A supplier can also directly see cost savings in the use of the Interfaces among its own products (as the need for an open interface arises).
- Lower Middleware Transitions Costs – the Interfaces are defined to be middleware and transport independent. So, the supplier can migrate from one middleware or transport technology to another without changing the supporting business logic in the code.
- Increase Usage by System Integrators (SIs) – a supplier's support of their own "open" interfaces goes only so far to encourage SIs. Clearly, an SI would like to make use of supplier products (both equipment and OSS suppliers) that make use of well supported standard interfaces rather than supplier specific interfaces. The latter case forces the SI into a situation characterized by many pair-wise negotiations between various suppliers.
- Lower Training Cost – in as much as a supplier re-uses the Interfaces for multiple products and for multiple customers, the various training costs are lower because the designers, system engineers, developers and testers are using the same Interfaces over and over again.

3 Business Processes

3.1 Business Requirements

No requirements have been identified for this category.

3.2 Category I: Static and Structural Requirements

Refer to the TMF518_NRF BA document.

3.3 Category II: Normal Sequences, Dynamic Requirements

3.3.1 Connection Control

R_TMF518_RP_II_0002	The Interface shall allow the requesting OS to create a planned Subnetwork Connection (SNC) given the requesting OS specified data listed in R_TMF518_RP_II_0005. The requesting OS specified SNC data is a result of the (successful) completion of this request, the target OS shall create an object representing the SNC, but shall not attempt to establish (on NEs) any of the cross-connections associated with the SNC. The successfully created SNC will be in pending state.
Source	TMF 513 v3.0, Requirement II.082

R_TMF518_RP_II_0003	The Interface shall allow the requesting OS to create point to multi-point configurations. Each leg of a point to multi-point configuration shall be represented by its own Subnetwork Connection (SNC) which allows for individual management of each leg. (e.g. add or remove). All SNCs of a point to multi-point configuration share the same aEnd Termination Point (TP).
Source	TMF 513 v3.0, Requirement II.083

R_TMF518_RP_II_0004	The Interface shall allow the requesting OS to create a

	Subnetwork Connection (SNC) without specifying the ending Termination Point (TP) instances, but only the ending Managed Element (ME) instances or containing TP instances. The end TP instances will be chosen by the target OS, at SNC creation time. These ME or TP instances are therefore identified by the name value “Target OS assigned”.
Source	TMF 513 v3.0, Requirement II.153

R_TMF518_RP_II_0005	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS create a Subnetwork Connection (SNC): 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the SNCs within the target OS. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Directionality This parameter shall represent the directionality of the SNC (bidirectional or unidirectional). The target OS shall set the directionality to unidirectional if either an ingress or egress Traffic Descriptor is zero for an ATM Subnetwork Connection. 5. Static protection level This parameter shall indicate the degree of internal resilience/protection of the SNC e.g., to indicate whether the Subnetwork Connection should be Protected, Preemtible, or Unprotected. The target OS will be required to create a SNC with the specified protection level. Refer to the supporting document SD1-36_SNCTypes for details on the static protection level parameter. 6. Protection effort This parameter shall indicate whether the resilience requested must be achieved of not. Refer to the supporting document SD1-36_SNCTypes for details on the protection effort parameter. 7. SNC Type Refer to R_TMF518_FMW_I_0029 in the TMF518_FMW BA. 8. Layer Rate Refer to R_TMF518_NRB_I_0003 in the TMF518_NRB BA. 9. Routing constraint data Refer to R_TMF518_NRF_I_0030 in the TMF518_NRF BA.

	10. Complete route This parameter shall indicate whether the routing constraint specifies a complete route. 11. Network routed This parameter shall indicate whether the route shall be computed by the network. 12. Reroute allowed This parameter shall indicate if the SNC may be rerouted. 13. Network reroute This parameter shall indicate if the reroute (if allowed) shall be computed by the network, by the target OS, or by either. 14. Revertive This parameter shall indicate whether the SNC shall always attempt to return to its intended Route. 15. Priority This parameter shall represent the priority of the SNC (i.e. highest (0) to lowest). 16. Exclusive intended route This parameter shall indicate if the intended route that is provided as part of this creation data is an exclusive route. 17. aEnd TP(s) This parameter represents a list of the names of the aEnd Termination Point (TP) (s) that shall be the aEnd points of the SNC. The names may be either specific or generic. The TPs may of the following type: - Connection Termination Point (CTP) - Group Termination Point (GTP) - Floating Termination Point (FTP) 18. zEnd TP(s) This parameter represents a list of the names of the zEnd Termination Point (TP) (s) that shall be the zEnd points of the SNC. The names may be either specific or generic. The TPs may of the following type: - Connection Termination Point (CTP) - Group Termination Point (GTP) - Floating Termination Point (FTP) 19. Bundled SNC This parameter shall indicate if the SNC to be created is a bundled SNC 20. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001 21. GTP deletion This parameter shall only be used when creating bundled SNCs. It shall indicate that the target OS has to delete all the interior GTPs supporting the bundled SNC when the SNC is deleted. 22. Alarm reporting

	This parameter shall indicate whether alarm reporting for the SNC is to be enabled or disabled. 23. Alarm severity assignment profile This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that is to be assigned to the SNC. 24. aEnd point role This attribute shall represent the role of the aEnd Termination Point (TP)s of the SNC. Refer to R_TMF518_NRF_I_0026 in the TMF518_NRF BA. 25. zEnd point role This attribute shall represent the role of the zEnd Termination Point (TP)s of the SNC. Refer to R_TMF518_NRF_I_0026 in the TMF518_NRF BA. 26. Network Access Domain This attribute represents the Network Access Domain (NAD) to which this Subnetwork has been assigned. 27. Grade of impact This parameter shall indicate the degree to which the creation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes for details on the grade of impact parameter. 28. OS freedom level This parameter shall indicate the level of freedom given to the target OS in determining the effect on the Subnetwork when performing SNC operations. 29. Name Refer to R_TMF518_FMW_I_0001 30. AliasNameList Refer to R_TMF518_FMW_I_0001
Source	TMF 513 v3.0, Requirement II.084

R_TMF518_RP_II_0103	The Interface shall allow the requesting OS to specify the following parameters when it requests to check if a valid Subnetwork Connection (SNC) can be created (the other parameters used for the actual creation – as shown in R_TMF518_RP_II_0005 – are not necessary for the check): 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the SNCs within the target OS.

	3. Directionality This parameter shall represent the directionality of the SNC (bidirectional or unidirectional). The target OS shall set the directionality to unidirectional if either an ingress or egress Traffic Descriptor is zero for an ATM Subnetwork Connection. 4. Static protection level This parameter shall indicate the degree of internal resilience/protection of the SNC e.g., to indicate whether the Subnetwork Connection should be Protected, Preemtible, or Unprotected. The target OS will be required to create a SNC with the specified protection level. Refer to the supporting document SD1-36_SNCTypes for details on the static protection level parameter. 5. Protection effort This parameter shall indicate whether the resilience requested must be achieved of not. Refer to the supporting document SD1-36_SNCTypes for details on the protection effort parameter. 6. SNC Type Refer to R_TMF518_FMW_I_0029 in the TMF518_FMW BA. 7. Layer Rate Refer to R_TMF518_NRB_I_0003 in the TMF518_NRB BA. 8. Routing constraint data Refer to R_TMF518_NRF_I_0030 in the TMF518_NRF BA. 9. Complete route This parameter shall indicate whether the routing constraint data attribute specifies a complete route. 10. Network routed This parameter shall indicates whether the route shall be computed by the network. 11. Exclusive intended route This parameter shall indicate if the intended route that is provided as part of this creation data is an exclusive route. 12. aEnd TP(s) This parameter represents a list of the names of the aEnd Termination Point (TP) (s) that shall be the aEnd points of the SNC. The names may be either specific or generic. The TPs may of the following type: - Connection Termination Point (CTP) - Group Termination Point (GTP) - Floating Termination Point (FTP) 13. zEnd TP(s) This parameter represents a list of the names of the zEnd Termination Point (TP) (s) that shall be the zEnd points of the SNC. The names may be either specific or generic. The TPs may of the following type:

	- Connection Termination Point (CTP) - Group Termination Point (GTP) - Floating Termination Point (FTP) 14. Bundled SNC This parameter shall indicate if the SNC to be created is a bundled SNC 15. aEnd point role This attribute shall represent the role of the aEnd Termination Point (TP)s of the SNC. Refer to R_TMF518_NRF_I_0026 in the TMF518_NRF BA. 16. zEnd point role This attribute shall represent the role of the zEnd Termination Point (TP)s of the SNC. Refer to R_TMF518_NRF_I_0026 in the TMF518_NRF BA. 17. mustConsiderResources This parameter provides a list of TPs and parameters that would be applied to the potential SNC. 18. tpDataListToModify This parameter indicates whether or not the Subnetwork's equipment capabilities and the current resource states must be considered for potential activation of the Subnetwork Connection. 19. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001
Source	TMF 518_RP Version 1.0

R_TMF518_RP_II_0006	The Interface shall allow the requesting OS to check if a valid Subnetwork Connection (SNC) can be created given the SNC creation data specified by the requesting OS as specified in R_TMF518_RP_II_0103. The requesting OS may also request the target OS to consider the SNC resources to determine whether activation of the SNC, if applied, will succeed. The validity check will take into consideration the subnetwork's equipment capabilities and the current resource states. This is a best effort guarantee as the resources may not be available when The requesting OS tries to actually activate the SNC.
Source	TMF 513 v3.0, Requirement II.085

R_TMF518_RP_II_0007	The Interface shall allow the requesting OS to request that a target OS activate a specified Subnetwork Connection (SNC) specified by the requesting OS given the SNC data specified by

	the requesting OS as listed in R_TMF518_RP_II_0008 As a result of the (successful) completion of this request, the target OS will have issued the required commands such that all Cross Connect (CC)s associated with (comprising) the SNC are in place (i.e., are provisioned on NE(s)). The state of the successfully activated SNC will transition to active. If the SNC specified by the requesting OS has more than one route, this operation unlocks all the routes, delegating the target OS and/or the network (e.g. restoration process) the actual activation of the appropriate route.
Source	TMF 513 v3.0, Requirement II.086

R_TMF518_RP_II_0008	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS activates a Subnetwork Connection (SNC): 1) SNC name This parameter shall indicate the name of the previously created SNC that is to be activated. 2) Grade of impact This parameter shall indicate the degree to which the activation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes for details on the grade of impact parameter. 3) OS freedom level This parameter shall indicate the level of freedom given to the target OS in determining the effect on the Subnetwork when performing SNC operations. 4) Termination Point (TP)s to modify This parameter shall identify a list of TPs that are to be modified as part of the activation request. Each item in the list shall contain the following: • TP name (generic TP names are not allowed) • Transmission parameters • Ingress transmission descriptor name • Egress transmission descriptor name • Mapping mode
Source	TMF 513 v3.0, Requirement II.087

R_TMF518_RP_II_0009	The Interface shall allow the requesting OS to request that a Subnetwork Connection (SNC) be both created and activated (as a result of a single request from the requesting OS) given the SNC data specified by the requesting OS as listed in R_TMF518_RP_II_0005. As a result of (successful) completion of this request, the target OS shall create an object representing the SNC and shall have issued required commands such that all cross-connections associated with (comprising) the SNC are in place (i.e., are provisioned on NE(s)). The state of the successfully created and activated SNC will transition to active. If the SNC specified by the requesting OS has more than one route, this operation unlocks all the routes, delegating the target OS and/or the network (e.g. restoration process) the actual activation of the appropriate route.
Source	TMF 513 v3.0, Requirement II.088

R_TMF518_RP_II_0010	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS create and activate a Subnetwork Connection (SNC): 1. The parameters required to create an SNC (Refer to R_TMF518_RP_II_0005). 2. Termination Point (TP)s to modify This parameter shall identify a list of TPs that are to be modified as part of the activation request. Each item in the list shall contain the following: • TP name (generic TP names are not allowed) • Transmission parameters • Ingress transmission descriptor name • Egress transmission descriptor name • Mapping mode 3. BLSRDirection (BidirectionalLineSwitchedRingDirection) This parameter is used in conjunction with the timeslot when the target OS cannot use the routing constraints for a BLSR case. 4. Timeslot This attribute is used in conjunction with blsrDirection.. 5. PotentialFutureSetupIndicator (PotentialFutureSetupIndicator) This attribute refers to the aEnd of the SNC and is used to convey the likely future (or current) configuration of the SNC.

	See attached supporting document SD1-16_LayeredParameters.
Source	TMF 513 v3.0, Requirement II.089

R_TMF518_RP_II_0011	The Interface shall allow the requesting OS to add a protection route to a Subnetwork Connection (SNC) specified by the requesting OS in a target OS given the SNC data specified by the requesting OS as listed in R_TMF518_RP_II_0012. As a result of (successful) completion of this request, the target OS shall add the new route to the SNC, but shall not attempt to establish (on NEs) any cross connections as side effect of this operation, because the route is created in locked state.
Source	TMF 513 v3.0, Requirement II.241

R_TMF518_RP_II_0012	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS adds a route to a Subnetwork Connection (SNC) 1) SNC name This parameter shall represent the name of the SNC to which the Route is to be assigned that is to be activated. 2) Grade of impact This parameter shall indicate the degree to which the activation of the added route may impact the Subnetwork. (Refer to the supporting document SD1-36_SNCTypes for details on the grade of impact parameter. 3) OS freedom level This parameter shall indicate the level of freedom given to the target OS in determining the effect on the Subnetwork when performing SNC operations. 4) Intended This parameter shall indicate if this route is the new intended route for the SNC, as opposed to a back-up route. 5) Exclusive This parameter shall indicate if this route is an exclusive route for this SNC. An exclusive route is a route that no other SNCs can share any of its CCs or CTPs. 6) Routing constraint data

	Refer to R_TMF518_NRF_I_0030. 7) Complete route This parameter shall indicate whether the routing constraint data attribute specifies a complete route.
Source	TMF 513 v3.0, Requirement II.242

R_TMF518_RP_II_0013	The Interface shall allow the requesting OS to remove a protection route from a Subnetwork Connection (SNC) specified by the requesting OS given the SNC data specified by the requesting OS as listed in R_TMF518_RP_II_0014. As a result of (successful) completion of this request, the target OS shall delete the protection route of specified SNC. The specified route must not be in the unlocked state, and must not be the intended route. Of course it is possible to delete a locked backup route which is “in use” by other SNC route, because this operation has no side effect on routes of any other SNCs, even if sharing Cross-Connect (CC) s/ Connection Termination Point (CTP) s.
Source	TMF 513 v3.0, Requirement II.243

R_TMF518_RP_II_0014	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS remove a route from a Subnetwork Connection (SNC). 1. SNC name This parameter shall represent the name of the SNC from which the Route is to be removed. 2. Route identifier This parameter shall represent the identifier of the route that is to be removed. 3. OS freedom level This parameter shall indicate the level of freedom given to the target OS in determining the effect on the Subnetwork when performing SNC operations. 4. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001
Source	TMF 513 v3.0, Requirement II.244

R_TMF518_RP_II_0015	The Interface shall allow the requesting OS to modify (a route of) a Subnetwork Connection (SNC) specified by the requesting

	OS in the target OS given the SNC data specified by the requesting OS as listed in R_TMF518_RP_II_0016. As a result of (successful) completion of this request, the target OS shall update the route description, but shall not attempt to establish (in the network) any of the Cross-Connect (CC) s associated with the modified route. The state of the successfully modified route will be locked. If the target OS can preserve the name of the requesting OS specified SNC then the route is modified. If the target OS cannot preserve the name of the requesting OS specified SNC then the target OS shall create a new pending SNC from an existing pending or active SNC.
Source	TMF 513 v3.0, Requirement II.245

R_TMF518_RP_II_0016	The Interface shall allow the requesting OS to specify in addition to the creation parameters specified in R_TMF518_RP_II_0005, the following parameters when it requests that a target OS modifies a Subnetwork Connection (SNC): 1. SNC name This parameter shall represent the name of the SNC to be modified. 2. Route identifier This parameter shall represent the identifier of the Route that is to be modified. If not specified, the intended Route shall be used 3. Modification type This parameter shall indicate the type of modification to be performed (i.e. rerouting or add/remove protection). 4. Retain SNC This parameter shall indicate if when modifying an SNC whether the original SNC shall be deleted or put into the pending state. 5. Modify server layers This parameter shall indicate whether the target OS is allowed to modify the server layers to fulfill the protection constraints identified by this request. 6. Added or new route This parameter shall represent (depending on the modification type), the route of a new protection leg or the whole SNC. When it describes a segment to be added, either the SNCP cross-connects or the switch TPs that will be changed in the segment may be specified by the requesting OS. The target OS then chooses the missing segments. Alternatively, the requesting OS may specify the full route.

	7. Removed route This parameter shall represent the protection leg that is to be removed from the SNC. Either the last cross-connects (that contain the SNCP) are specified by the requesting OS or the full route may be specified. This parameter can be used in conjunction with Added Or New Route only to reroute a segment. 8. Termination Point (TP)s to modify This parameter shall identify a list of TPs that are to be modified as part of the modification request. Each item in the list shall contain the following: • TP name (generic TP names are not allowed) • Transmission parameters • Ingress transmission descriptor name • Egress transmission descriptor name • Mapping mode 9. TolerableImpactEffort This parameter qualifies the tolerable conditions under which the SNC modification may be performed.
Source	TMF 513 v3.0, Requirement II.246

R_TMF518_RP_II_0017	The Interface shall allow the requesting OS to modify a Subnetwork Connection (SNC) specified by the requesting OS in the target OS given the SNC data specified by the requesting OS as listed in R_TMF518_RP_II_0016. As a result of (successful) completion of this request, the target OS shall modify and activate the SNC. The end result of this operation is equivalent to a requesting OS using the create-modify operation followed by the swap operation.
Source	TMF 513 v3.0, Requirement II.257

R_TMF518_RP_II_0018	The Interface shall allow the requesting OS to activate a pending Subnetwork Connection (SNC) and deactivate an active SNC in a single atomic operation given the SNC data specified by the requesting OS as listed in R_TMF518_RP_II_0019.
Source	TMF 513 v3.0, Requirement II.258

R_TMF518_RP_II_0019	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS swap

	two Subnetwork Connection (SNC)s. 1) SNC name to be deactivated This parameter shall indicate the name of the active SNC that is to be deactivated. 2) SNC name to be activated This parameter shall indicate the name of the previously created SNC that is to be activated. 3) Grade of impact This parameter shall indicate the degree to which the activation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes for details on the grade of impact parameter. 4) OS freedom level This parameter shall indicate the level of freedom given to the target OS in determining the effect on the Subnetwork when performing this operation. 5) Termination Point (TP)s to modify This parameter shall identify a list of Termination Point (TP)s that are to be modified as part of the swap request. Each item in the list shall contain the following: • TP name (generic TP names are not allowed) • Transmission parameters • Ingress traffic descriptor name • Egress traffic descriptor name • Mapping mode
Source	TMF 513 v3.0, Requirement II.259

R_TMF518_RP_II_0020	The Interface shall allow the requesting OS to switch a route of a Subnetwork Connection (SNC) specified by the requesting OS given the SNC data specified by the requesting OS as listed in R_TMF518_RP_II_0021. As a result of (successful) completion of this request, the target OS shall activate in the network the input route, and deactivate the currently active route, plus all the partial routes, if any. The operation is refused if performed on a pending SNC, or on a locked route. The operation does not affect the administrative state of any route. The restoration process is still allowed to re-route again, e.g. in case of failures.
Source	TMF 513 v3.0, Requirement II.247

R_TMF518_RP_II_0021	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS switched the route of a Subnetwork Connection (SNC): 1. SNC name This parameter shall represent the name of the SNC for which the Route is to be switched. 2. Route identifier This parameter shall represent the identifier of the route that is to be switched. 3. Grade of impact This parameter shall indicate the degree to which the route switch of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes for details on the grade of impact parameter. 4. OS freedom level This parameter shall indicate the level of freedom given to the target OS in determining the effect on the Subnetwork when performing SNC operations. 5. Termination Point (TP)s to modify This parameter shall identify a list of TPs that are to be modified as part of the switch route request. Each item in the list shall contain the following: • TP name (generic TP names are not allowed) • Transmission parameters • Ingress transmission descriptor name • Egress transmission descriptor name • Mapping mode 6. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001
Source	TMF 513 v3.0, Requirement II.248

R_TMF518_RP_II_0022	The Interface shall allow the requesting OS to set the administrative state of one or more of the routes of a Subnetwork Connection (SNC) specified by the requesting OS given the SNC data specified by the requesting OS as listed in R_TMF518_RP_II_0023.

	As a result of (successful) completion of this request, the target OS shall update the administrative state of addressed routes. The unlocked routes of an SNC are the set of resources the restoration process is allowed to work with.
Source	TMF 513 v3.0, Requirement II.249

R_TMF518_RP_II_0023	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS set the administrative state of one or more of the routes of a Subnetwork Connection (SNC): 1. SNC name This parameter shall represent the name of the SNC for which the Route is to be switched. 2. Route identifiers This parameter shall represent a list of the identifiers of the routes that are to have their administrative state set. 3. Administrative state values This attribute shall represent a list of the values for the administrative states of the routes specified. 4. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001
Source	TMF 513 v3.0, Requirement II.250

R_TMF518_RP_II_0024	The Interface shall allow the requesting OS to set a route as the intended one of a Subnetwork Connection (SNC) specified by the requesting OS given the SNC data specified by the requesting OS as listed in R_TMF518_RP_II_0025. As a result of (successful) completion of this request, the addressed route is the intended one, and the formerly intended route is a backup one.
Source	TMF 513 v3.0, Requirement II.251

R_TMF518_RP_II_0025	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS set a route as the intended one for a Subnetwork Connection (SNC): 1. SNC name This parameter shall represent the name of the SNC for which the Route is to be switched.

	2. Route identifier This parameter shall represent the identifier of the route that is to be set to the intended route of the SNC. 3. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001
Source	TMF 513 v3.0, Requirement II.252

R_TMF518_RP_II_0026	The Interface shall allow the requesting OS to request that a target OS deactivate a Subnetwork Connection (SNC) specified by the requesting OS. Refer to R_TMF518_RP_II_0027 for the data provide as part of the deactivation request. As a result of (successful) completion of this request, the target OS will have issued required commands such that all cross-connections associated with (comprising) the Subnetwork Connection have been removed (i.e., are no longer provisioned on NE(s)), but the target OS shall preserve the object representing the Subnetwork Connection. The successfully deactivation of the specified SNC will transition to pending. If the SNC specified by the requesting OS has more than one route, this operation locks all the routes, which means that target OS and/or the network (e.g. restoration process) have no more control over these routes. All the currently active Cross-Connect (CC)s for this SNC shall be removed, of any (active or partial) route.
Source	TMF 513 v3.0, Requirement II.090

R_TMF518_RP_II_0027	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS de-activate a Subnetwork Connection (SNC): 1) SNC name This attribute shall indicate the name of the SNC that is to be deactivated. 2) Grade of impact This attribute shall indicate the degree to which the activation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes for details on the grade of impact parameter. 3) OS freedom level This attribute shall indicate the level of freedom given to the target OS in determining the effect on the

	Subnetwork when performing SNC operations. 4) Termination Point (TP)s to modify This attribute shall identify a list of TPs that are to be modified as part of the activation request. Each item in the list shall contain the following: • TP name (generic TP names are not allowed) • Transmission parameters • Ingress transmission descriptor name • Egress transmission descriptor name • Mapping mode
Source	TMF 513 v3.0, Requirement II.091

R_TMF518_RP_II_0028	The Interface shall allow the requesting OS to request that a target OS delete a Subnetwork Connection (SNC) specified by the requesting OS. Refer to R_TMF518_RP_II_0029 for the data provide as part of the delete request. As a result of the (successful) completion of this request, the target OS shall delete the object representing the SNC. The target OS shall refuse/fail this request if any of the cross-connections associated with (comprising) the SNC were in place (i.e., are provisioned on NE(s)) at the time of the request (i.e., the SNC must be successfully deactivated before the target OS will allow the SNC to be deleted). If the SNC has more than one route, then the operation deletes the SNC, its intended and all back-up route(s).
Source	TMF 513 v3.0, Requirement II.092

R_TMF518_RP_II_0029	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS delete a Subnetwork Connection (SNC): 1) SNC name This attribute shall indicate the name of the previously created SNC that is to be activated. 2) OS freedom level This attribute shall indicate the level of freedom given to the target OS in determining the effect on the Subnetwork when performing SNC operations.
Source	TMF 513 v3.0, Requirement II.093

R_TMF518_RP_II_0030	The Interface shall allow the requesting OS to request that a target OS both deactivate and delete (as a result of a single request) a Subnetwork Connection (SNC) specified by the requesting OS. Refer to R_TMF518_RP_II_0031 for the data provided as part of the deactivate and delete request. As a result of (successful) completion of this request, the target OS will have issued required commands such that all cross-connections associated with (comprising) the Subnetwork Connection have been removed (i.e., are no longer provisioned on NE(s)), and the target OS shall delete the object representing the Subnetwork Connection. If the SNC specified by the requesting OS has more than one route, this operation locks all the routes, which means that target OS and/or the network (e.g. restoration process) have no more control over these routes. All the currently active Cross-Connect (CC) s for this SNC shall be removed, of any (active or partial) route. Then the operation deletes the SNC, its intended and all back-up route(s).
Source	TMF 513 v3.0, Requirement II.094

R_TMF518_RP_II_0031	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS de-activate and delete a Subnetwork Connection (SNC): 1) SNC name This attribute shall indicate the name of the SNC to be activated and deleted. 2) Grade of impact This attribute shall indicate the degree to which the activation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes for details on the grade of impact parameter. 3) OS freedom level This attribute shall indicate the level of freedom given to the target OS in determining the effect on the Subnetwork when performing SNC operations. 4) Termination Point (TP) s to modify This attribute shall identify a list of TPs that are to be modified as part of the activation request. Each item in the list shall contain the following: • TP name (generic TP names are not allowed)

	• Transmission parameters • Ingress transmission descriptor name • Egress transmission descriptor name • Mapping mode
Source	TMF 513 v3.0, Requirement II.095

R_TMF518_RP_II_0032	The Interface shall allow the requesting OS to de-activate and delete the leg of a point to multipoint configuration. Each leg of a point to multipoint configuration shall be a separate Subnetwork Connection (SNC).
Source	TMF 513 v3.0, Requirement II.096

3.3.2 Equipment Provisioning

R_TMF518_RP_II_0034	The Interface shall allow the requesting OS to provision an Equipment in an Equipment Holder given the requesting OS specified data listed in R_TMF518_RP_II_0035.
Source	TMF 513 v3.0, Requirement II.136

R_TMF518_RP_II_0035	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS provision an Equipment: 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS to check whether the user label is unique amongst the equipments within the target OS. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001 5. Name Refer to R_TMF518_FMW_I_0001 6. AliasNameList Refer to R_TMF518_FMW_I_0001

	7. Network Access Domain 8. Expected equipment type This parameter shall represent the type of the expected equipment. 9. Alarm reporting This parameter shall indicate whether alarm reporting for this equipment is to be enabled or disabled. 10. Alarm severity assignment profile This parameter shall represent the name of the Alarm Severity Assignment Profile (ASAP) that is to be assigned to the Equipment. 11. Manufacturer This parameter shall represent the name of the equipment vendor. 12. Protection role This parameter shall represent the protection role (e.g. primary or secondary) that the equipment plays in case it takes part in an equipment protection scheme 13. Protection scheme state This parameter shall indicate the state of the protection scheme (i.e. whether it is to be active or locked).
Source	TMF 513 v3.0, Requirement II.263

R_TMF518_RP_II_0036	The Interface shall allow the requesting OS to unprovision an Equipment from an Equipment Holder.
Source	TMF 513 v3.0, Requirement II.262

3.3.3 Flow Domain Control

3.3.3.1 Matrix Flow Domain Management

3.3.3.1.1 Creation of Matrix Flow Domain (MFD)s

R_TMF518_RP_II_0037	The Interface shall allow the requesting OS to create a Matrix Flow Domain (MFD) within an Managed Element (ME); given the requesting OS specified data listed in R_TMF518_RP_II_0038
Source	TMF 513 v3.1, Requirement II. 305

3.3.3.1.1.1 Matrix Flow Domain (MFD) Creation Data

R_TMF518_RP_II_0038	The Interface shall allow the requesting OS to specify the following parameters when it requests the creation of a Matrix

	Flow Domain (MFD): 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the MFDs within the target OS domain. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Network Access Domain This attribute represents the Network Access Domain (NAD) to which the new MFD shall be assigned to. 5. List of unassigned CPTPs This parameter contains the list of unassigned CPTPs which shall be assigned to the MFD to be created. 6. Connectionless Layered Parameters This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB) that are supported by the MFD. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters for details of the currently defined connectionless parameters. 7. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001
Source	TMF 513 v3.1, Requirement II. 306

3.3.3.1.2 Modification of Matrix Flow Domain (MFD)s

R_TMF518_RP_II_0039	The Interface shall allow the requesting OS to modify a Matrix Flow Domain (MFD); given the requesting OS specified data listed in R_TMF518_RP_II_0040. The modification of the MFD shall be done on a best effort basis. All attributes that could not be modified shall be returned in the reply.
Source	TMF 513 v3.1, Requirement II. 307

3.3.3.1.2.1 Matrix Flow Domain (MFD) Modification Data

R_TMF518_RP_II_0040	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS modify

	an Matrix Flow Domain (MFD): 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the MFDs within the target OS. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Network Access Domain This attribute represents the new Network Access Domain (NAD) to which the MFD shall be assigned to. 5. Connectionless Layered Parameters This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB) that are to be changed in the MFD. Refer to chapter "Connectionless Technology Parameters" of the supporting document SD1-16_LayeredParameters for details of the currently defined connectionless parameters. 6. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001
Source	TMF 513 v3.1, Requirement II. 308

3.3.3.1.3 Deletion of Matrix Flow Domain (MFD)s

R_TMF518_RP_II_0041	The Interface shall allow the requesting OS to delete a Matrix Flow Domain (MFD) from a Managed Element (ME) given the requesting OS specified MFD name. The request shall be denied if the MFD to be deleted is still associated to a Matrix Flow Domain (MFD) The request shall be denied if the MFD to be deleted is fixed. The CPTPs shall be automatically de-assigned from the MFD (i.e., change the TP role to unassigned) before the MFD is deleted.
Source	TMF 513 v3.1, Requirement II. 309

3.3.3.1.4 CPTP Management

3.3.3.1.4.1 CPTP Assignment to Matrix Flow Domain (MFD)

R_TMF518_RP_II_0042	The Interface shall allow the requesting OS to assign CPTPs to a Matrix Flow Domain (MFD).

	The requesting OS shall provide a list of unassigned CPTP names to be associated to the MFD. The provided CPTPs must be potential CPTPs for this MFD (e.g. have to be on the same equipment or same rack with backplane connectivity). The request shall be denied if the MFD is “fixed”.
Source	TMF 513 v3.1, Requirement II. 310

3.3.3.1.4.2 CPTP Un-assignment from Matrix Flow Domain (MFD)

R_TMF518_RP_II_0043	The Interface shall allow the requesting OS to un-assign CPTPs from a Matrix Flow Domain (MFD). The requesting OS shall provide a list of assigned CPTPs to be un-assigned from the MFD. The request shall be denied if the MFD is “fixed”. A request to un-assign an FD Edge CPTP from a MFD shall be denied if a Flow Domain Fragment uses this CPTP. The CPTPs which are also associated as Flow Domain Edge CPTP (FD Edge CPTP)s to a Flow Domain (FD) shall automatically be de-associated from the FD; refer to Figure “State diagram for Port TP Role State” of the supporting document SD1-44_ConnectionlessTechnologyManagement.
Source	TMF 513 v3.1, Requirement II. 311

3.3.3.1.5 Transmission Descriptor (TMD) Management

3.3.3.1.5.1 Association of a Transmission Descriptor (TD) to a Matrix Flow Domain (MFD)

R_TMF518_RP_II_0044	The Interface shall allow the requesting OS to associate a Transmission Descriptor (TMD) to a given Matrix Flow Domain (MFD) identified by a requesting OS specified MFD name. The association will overwrite specific transmission parameters of the MFD with the corresponding parameter values contained in the TMD. The Behaviour is defined in R_TMF518_RP_II_0098.
Source	TMF 513 v3.1, Requirement II. 312

3.3.3.1.5.2 Disassociation of a Transmission Descriptor (TD) from a Matrix Flow Domain (MFD)

R_TMF518_RP_II_0045	The Interface shall allow the requesting OS to de-associate a Transmission Descriptor (TMD) from a given Matrix Flow Domain (MFD) identified by a requesting OS specified MFD name. The de-association shall not influence any transmission parameter configured in the MFD. The Behaviour is defined in R_TMF518_RP_II_0098.
Source	TMF 513 v3.1, Requirement II. 313

3.3.3.2 Flow Domain (FD) Management

3.3.3.2.1 Creation of Flow Domain (FD)s

R_TMF518_RP_II_0046	The Interface shall allow the requesting OS to create a Flow Domain (FD) within the target OS; given the requesting OS specified data listed in R_TMF518_RP_II_0047 The association of the CPTPs to the FD shall be done on a best effort basis.
Source	TMF 513 v3.1, Requirement II. 321

3.3.3.2.1.1 Flow Domain (FD) Creation Data

R_TMF518_RP_II_0047	The Interface shall allow the requesting OS to specify the following parameters when it requests the creation of a Flow Domain (FD): 1. Name This parameter defines the identifier of the new FD which will be used over the interface. The target OS has to make sure that the name of the FD is unique within the target OS domain. If no name is provided by the requesting OS, the target OS has to define a unique name. 2. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 3. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the FDs within the target OS domain. 4. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 5. Network Access Domain This parameter shall indicate the Network Access Domain (NAD) to which this FD will be associated to. 6. List of Matrix Flow Domains This parameter identifies the list of MFDs to be associated to the new FD. The MFDs to be associated must exist and must not be associated to another FD, i.e., they have to be un-associated. 7. Connectionless layered parameters This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB, Fiber Channel) that are supported by the FD. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-

	16_LayeredParameters for details of the currently defined connectionless parameters. 8. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001.
Source	TMF 513 v3.1, Requirement II. 322

3.3.3.2.2 Modification of Flow Domain (FD)s

R_TMF518_RP_II_0048	The Interface shall allow the requesting OS to modify a Flow Domain (FD); given the requesting OS specified data listed in R_TMF518_RP_II_0049.
Source	TMF 513 v3.1, Requirement II. 323

3.3.3.2.2.1 Flow Domain (FD) Modification Data

R_TMF518_RP_II_0049	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS modify an Flow Domain (FD): 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the MFDs within the target OS. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Network Access Domain This parameter shall indicate the Network Access Domain (NAD) to which this FD will be associated to. 5. Connectionless layered parameters This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB, Fiber Channel) that are to be changed in the FD. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters for details of the currently defined connectionless parameters. 6. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001.
Source	TMF 513 v3.1, Requirement II. 324

3.3.3.2.3 Deletion of Flow Domain (FD)s

R_TMF518_RP_II_0050	The Interface shall allow the requesting OS to request the deletion of a Flow Domain (FD) from the target OS given the requesting OS specified FD name. The request shall de-associate all Matrix Flow Domain (MFD)s and Flow Domain Edge CPTP (FD EdgeCPTP)s from the FD and then delete the FD. The Port TP role state indication of the CPTPs which corresponds to the de-associated FD Edge CPTPs shall be set to “assigned”. The request shall be denied if Flow Domain Fragment (FDFr)s are provisioned on any of the FD Edge CPTPs associated to the FD
Source	TMF 513 v3.1, Requirement II. 325

3.3.3.2.4 CPTP Management

3.3.3.2.4.1 Association of a CPTP to a Flow Domain (FD)

R_TMF518_RP_II_0051	The Interface shall allow the requesting OS to associate CPTPs to a Flow Domain (FD).The requesting OS shall provide a list of assigned CPTPs to be associated to the FD. These CPTPs have to be already assigned to one of the Matrix Flow Domain (MFD)s that are associated to the FD. The successful association of the CPTPs shall update the Port TP role state of the CPTPs (i.e., newly associated CPTPs become “FD Edge”). The association of the CPTPs to the FD shall be done on a best effort basis.
Source	TMF 513 v3.1, Requirement II. 326

3.3.3.2.4.2 Disassociation of a CPTP from a Flow Domain (FD)

R_TMF518_RP_II_0052	The Interface shall allow the requesting OS to dissociate Flow Domain Edge CPTP (FD Edge CPTP)s from a Flow Domain (FD) The requesting OS shall provide a list of FD Edge CPTPs to be dissociated from the FD. The request shall be denied if a Flow Domain Fragment (FDFr) uses this FD Edge CPTP. The successful dissociation of the CPTPs shall update the Port TP role state of the CPTPs (i.e., newly dissociated CPTPs lose

	their “FD Edge” property, i.e., become “assigned”).
Source	TMF 513 v3.1, Requirement II. 327

3.3.3.2.5 Matrix Flow Domain (MFD) Management

3.3.3.2.5.1 Association of a Matrix Flow Domain (MFD) to a Flow Domain (FD)

R_TMF518_RP_II_0053	The Interface shall allow the requesting OS to associate Matrix Flow Domain (MFD)s to a Flow Domain (FD). The requesting OS shall provide a list of existing MFDs to be associated to the FD. The MFDs to be associated must not be associated to another FD
Source	TMF 513 v3.1, Requirement II. 328

3.3.3.2.5.2 Disassociation of a Matrix Flow Domain (MFD) from a Flow Domain (FD)

R_TMF518_RP_II_0054	The Interface shall allow the requesting OS to dissociate Matrix Flow Domain (MFD)s from a Flow Domain (FD).. The requesting OS shall provide a list of associated MFDs to be dissociated from the FD. The request shall be denied if a Flow Domain Fragment (FDFr) uses Flow Point (FP)s that are served by an Flow Domain Edge CPTP (FD Edge CPTP) which is assigned to this (edge) MFD. A target OS working in the “connectivity-aware” mode shall also deny the request if the MFD to be dissociated is not at the edge of an FD but is used by an FDFr. The request dissociates all FD Edge CPTPs which are associated to the MFD to be dissociated from the FD and sets the Port TP role state indication of the CPTPs to “assigned
Source	TMF 513 v3.1, Requirement II. 329

3.3.3.3 Traffic Conditioning (TC) Profile Management

3.3.3.3.1 Creation of Traffic Conditioning (TC) Profiles

R_TMF518_RP_II_0055	The Interface shall allow the requesting OS to create a Traffic Conditioning (TC) Profile in the target OS given the requesting OS specified data listed in R_TMF518_RP_II_0056.
Source	TMF 513 v3.1, Requirement II. 333

3.3.3.3.1.1 Traffic Conditioning (TC) Profile Creation Data

R_TMF518_RP_II_0056	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS creates Traffic Conditioning (TC) Profile: 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the Traffic Conditioning Profiles within the target OS domain. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Traffic conditioning parameters This parameter shall represent a list of traffic conditioning parameters which can be set and/or retrieved at a specified connectionless layer on a Termination Point (TP) having this TC Profile associated. Refer to chapter "Traffic Conditioning Parameters" of the supporting document SD1-16_LayeredParameters for details of the currently defined connectionless parameters. 5. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001 6. Name Refer to R_TMF518_FMW_I_0001 7. AliasNameList Refer to R_TMF518_FMW_I_0001 8. Network Access Domain
Source	TMF 513 v3.1, Requirement II. 334

3.3.3.3.2 Modification of Traffic Conditioning (TC) Profiles

R_TMF518_RP_II_0057	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS modifies Traffic Conditioning (TC) Profile: 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the Traffic Conditioning Profiles within the target OS domain.

	3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Traffic conditioning parameters This parameter shall represent a list of traffic conditioning parameters which can be set and/or retrieved at a specified connectionless layer on a Termination Point (TP) having this TC Profile associated. Refer to chapter "Traffic Conditioning Parameters" of the supporting document SD1-16_LayeredParameters for details of the currently defined connectionless parameters. 5. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001 6. AliasNameList Refer to R_TMF518_FMW_I_0001 7. Network Access Domain Note: When traffic conditioning parameters are modified, this will automatically modify the traffic conditioning in all associated TPs.
Source	TMF 513 v3.1, Requirement II. 335

3.3.3.3.3 Deletion of Traffic Conditioning (TC) Profiles

R_TMF518_RP_II_0058	The Interface shall allow the requesting OS to delete a Traffic Conditioning (TC) Profile in the target OS given A requesting OS specified TC Profile name. The target OS shall reject the deletion request if any Flow Domain Edge CPTP (FD Edge CPTP) or Flow Point (FP) is still associated with this TC Profile. The target OS shall reject the deletion request if the requesting OS wants to delete the default TC Profile.
Source	TMF 513 v3.1, Requirement II. 336

3.3.3.3.4 Traffic Conditioning (TC) Profile Configuration

R_TMF518_RP_II_0059	The Interface shall allow the requesting OS to configure the mappings of specific Traffic Classes and Traffic Conditioning (TC) Profiles to specific groups of traffic units (e.g., Ethernet frames identified by VLAN-Id and Priority) at the following points: a) a CPTP b) a FDFr point (Flow Point (FP)

	The requesting OS shall provide a complete (except default) set of new mappings which overwrite all (except default) existing mappings in the Traffic Mapping Table. The provisioning of an empty mapping set removes all (except default) existing mappings; i.e., the traffic units are conditioned according to the default mapping. Note: The traffic units may still be conditioned specifically by another TP on this port. As a result of (successful) completion of this request, the target OS shall condition the traffic units flowing through the resource as defined by the Traffic Class and the TC Profile, or as defined by the default Traffic Class and default TC Profile respectively.
Source	TMF 513 v3.1, Requirement II. 337

3.3.3.4 Flow Domain Fragment (FDFr) Management

3.3.3.4.1 Creation of Flow Domain Fragment (FDFr)s

R_TMF518_RP_II_0060	The Interface shall allow the requesting OS to create a Flow Domain Fragment (FDFr) given the requesting OS specified data listed in R_TMF518_RP_II_0061. This includes FDFrs for flows of untagged frames. The target OS shall create all necessary Flow Point (FP)s and Matrix Flow Domain Fragment (MFDFr)s (not visible as separate named objects at the interface) comprising the FDFr. The request is successful if at least two (edge) FPs can be created. In case not all requested (edge) FPs can be connected when the FDFr is created, the target OS returns the list of the not connectable FPs in the success reply. If internal CPTPs are provided by the requesting OS and if it is not possible to include all provided internal CPTPs to the route of the FDFr, the operation will be rejected. If the target OS works in the “connectivity-aware” mode, the requesting OS can request one of two creation results when not all FPs have potential connectivity to one another: 1) reject the creation request 2) add all Flow Points regardless of potential connectivity. The target OS shall check if the new requested FDFr already

	exists, either entirely or partially or differently. The following cases have to be respected: 1) An existing FDFr matches the FDFr being requested ("matches" means that the set of flow points resulting from the activation of the FDFr is the same. Creation of two successive FDFr's on the same set of CPTPs, but with different VLAN IDs do NOT match). a) If the name specified by the requesting OS is the same as the name of the existing FDFr, or no name is specified: The operation will succeed (if all goes well). In this case (same name, or no name) the FDFr will be returned with the original name. b) If the name specified by the requesting OS is not the same as the name of the existing FDFr, the operation is rejected with exception Object In Use. 2) The existing FDFr is a subset or superset of the requested FDFr. a) If the requesting OS did not give a name, or if the requesting OS name is different from the name of the existing FDFr the operation is rejected with exception Object In Use. b) If the requesting OS name is the same as the name of the existing FDFr, the operation is rejected (because this operation does not allow to change the endpoints). Note: The above does not depend on the "ACTIVE" or "PARTIAL" state of the previous FDFr, but on the set of endpoints and where applicable the set of MFDFrs, of each FDFr.
Source	TMF 513 v3.1, Requirement II. 345

3.3.3.4.1.1 Flow Domain Fragment (FDFr) Creation Data

R_TMF518_RP_II_0061	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS creates a Flow Domain Fragment (FDFr): 1. Name This parameter defines the identifier of the new FDFr which will be used over the interface. The target OS has to make

	sure that the name of the FDFr is unique within the containing Flow Domain. If no name is provided by the requesting OS, the target OS has to define a unique name. Note: The name of the FDFr is not changeable after the FDFr is created. 2. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 3. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the FDFrs within the target OS. 4. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA . 5. Network Access Domain This parameter indicates the Network Access Domain to which this FDFr has to be assigned. 6. Connectionless layered parameters This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB, Fiber Channel) that are to be changed in the FD. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters for details of the currently defined connectionless parameters. 7. Directionality This parameter shall indicate the directionality of the new FDFr (bidirectional or unidirectional). Note: In the case of Ethernet, Directionality is always bidirectional. 8. aEnd TPs This parameter shall represent a list of of CPTP names that delimit the FDFr and characterize its edges (entrance and/or exit points). As a result of creating the FDFr, FPs are created as clients of the FD Edge CPTPs. In case of a unidirectional FDFr this attribute contains the list of source FD Edge CPTPs. In case of a bidirectional FDFr this attribute may be combined with the zEnd TPs attribute to obtain all the FD Edge CPTPs that are associated to the FDFr. Note: For a bidirectional point to point FDFr it is suggested, but

	not mandatory, to put one TP in the aEnd and one in the zEnd, as with SNCs and TLs. For a multipoint FDFr, or a point-to-point FDFr that may be expanded to multipoint, it is suggested to put all the TPs in the aEnd. 9. zEnd TPs In case of a unidirectional FDFr this attribute contains the list of sink FD Edge CPTPs that delimit the FDFr and characterize its edges (exit points). As a result of creating the FDFr, FPs are created as clients of the FD Edge CPTPs. In case of a bidirectional FDFr this attribute may be combined with the aEnd TPs attribute to obtain all the FD Edge CPTPs that are associated to the FDFr. 10. Internal TPs An optional (possibly empty) list of internal CPTP names that must be included in the route of the FDFr. As a result of creating the FDFr, FPs are created as clients of the internal CPTPs. 11. MFDFrs An optional (possibly empty) list of MFDFrs that make up the route of the FDFr. This attribute may be omitted if the FDFr is routed by the network. As a result of creating the FDFr, MFDFrs are created in the various MFDs. 12. Full route This parameter shall identify if the internal TPs and MFDFrs describe the full route of the FDFr (as opposed to only a partial constraint). When no routing constraints are specified the value of false must be used. 13. Termination Point(s) to configure This parameter shall identify a list of CPTPs and FPs that are to be configured as part of the creation request. Only CPTPs and FPs related to the new FDFr can be configured. Each item in the list shall contain the following: • termination point name • transmission parameters (incl. Traffic Mapping Table, Alarm Severity Assignment Profile names and TCA Parameter Profile names) • ingress/egress transmission descriptor names 14. FDFr type This parameter shall identify the type of the new FDFr (point-to-point, point-to-multipoint (E-tree), multipoint). 15. Connectivity requirement This parameter shall identify (for a “connectivity-aware” target OS) the requested operation mode in case not all FPs have potential connectivity to one another: • reject the creation request • add all Flow Points regardless of potential

	connectivity. If the target OS is not connectivity-aware, this parameter is ignored. 16. Administrative state This parameter shall indicate whether the FDFr shall be locked (i.e., traffic units cannot flow through the FDFr) or unlocked (i.e., traffic units are allowed to flow through the FDFr). 17. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001
Source	TMF 513 v3.1, Requirement II. 346

3.3.3.4.2 Modification of Flow Domain Fragment (FDFr)s

R_TMF518_RP_II_0062	The Interface shall allow the requesting OS to modify a Flow Domain Fragment (FDFr); given the requesting OS specified data listed in R_TMF518_RP_II_0063.
Source	TMF 513 v3.1, Requirement II. 347

3.3.3.4.2.1 Flow Domain Fragment (FDFr) Modification Data

R_TMF518_RP_II_0063	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS modify a Flow Domain Fragment (FDFr): 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA . 2. User label uniqueness This parameter shall indicate to the target OS that the value of the new user label attribute must be unique amongst all FDFrs within the target OS domain. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA . 4. Network Access Domain This parameter indicates the new Network Access Domain to which this FDFr has to be assigned. 5. Connectionless layered parameters This parameter shall represent the connectionless technology parameters associated with the layer (e.g., Ethernet, DVB) that the FDFr is connecting. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters for details of the currently defined connectionless parameters.

	6. TPs to remove This parameter shall represent a list of CPTP names that must be removed from the Flow Domain Fragment (FDFr). As a result of modifying the FDFr, the client Flow Points are deleted. 7. aEnd TPs This parameter shall represent a list of additional CPTP names that delimit the Flow Domain Fragment (FDFr) and characterize its edges (entrance and/or exit points). As a result of modifying the FDFr, Flow Points are created as clients of the FD Edge CPTPs. 8. zEnd TPs This parameter shall represent a list of additional CPTP names that delimit the Flow Domain Fragment (FDFr) and characterize its edges (exit points). As a result of modifying the FDFr, Flow Points are created as clients of the FD Edge CPTPs. 9. Internal TPs This parameter shall represent a list of additional internal CPTP names that must be added to the route of the Flow Domain Fragment (FDFr). As a result of modifying the FDFr, Flow Points are created as clients of the internal CPTPs. 10. Termination Point(s) to modify This parameter shall identify a list of Flow Points (FPs) that are to be modified as part of the modification request. Only FPs related to the FDFr can be modified. Each item in the list shall contain the following: • termination point name • transmission parameters (incl. Traffic Mapping Table, Alarm Severity Assignment Profile names and TCA Parameter Profile names) • ingress/egress transmission descriptor names 11. Administrative state This parameter shall indicate whether the FDFr shall be locked (i.e., traffic units cannot flow through the FDFr) or unlocked (i.e., traffic units are allowed to flow through the FDFr). 12. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001.
Source	TMF 513 v3.1, Requirement II. 348

3.3.3.4.3 Deletion of Flow Domain Fragment (FDFr)s

R_TMF518_RP_II_0064	The Interface shall allow the requesting OS to delete a Flow Domain Fragment (FDFr)given the requesting OS specified

	FDFr name.
Source	TMF 513 v3.1, Requirement II. 349

3.3.3.4.4 Flow Point (FP) Addition

R_TMF518_RP_II_0065	The Interface shall allow the requesting OS to add Flow Point (FP)s, identified by their server Flow Domain Edge CPTP (FD Edge CPTP)s, or by their FD Internal CPTPs, to a Flow Domain Fragment (FDFr). The requesting OS provides a list of FD Edge CPTPs to be added to the FDFr as additional end points. As a result of adding the CPTPs, FPs are created as clients of the CPTPs. The requesting OS provides a list of FD Internal CPTPs to be added to the FDFr as additional internal points. It shall be possible for the requesting OS to configure each Flow Point to be added (created) with the following Parameters: • transmission parameters (incl. Traffic Mapping Table, Alarm Severity Assignment Profile names and TCA Parameter Profile names) • ingress/egress transmission descriptor names If the target OS works in the “connectivity-aware” mode, the requesting OS can request one of two results when not all FPs have potential connectivity to all FPs already in the FDFr: • reject the request add all Flow Points regardless of potential connectivity
Source	TMF 513 v3.1, Requirement II. 350

3.3.3.4.5 Flow Point Removal

R_TMF518_RP_II_0066	The Interface shall allow the requesting OS to remove Flow Point (FP)s from a Flow Domain Fragment (FDFr). The requesting OS provides a list of existing CPTPs to be removed from the FDFr. The Flow Points who are clients of the removed CPTPs and are associated with the FDFr are deleted. The removal of CPTPs shall be rejected if less than two end Flow Points would remain after successful completion of the request.
Source	TMF 513 v3.1, Requirement II. 351

3.3.4 GUI Cut-Through Control

For the requirements in this section, it was agreed to consider only the NMS to EMS interface and not to generalize them at the OS to OS generic level. This generalization will be considered for further study.

R_TMF518_RP_II_0067	The NML-EML Interface shall allow the NMS to access the EMS user interface.
Source	TMF 513 v3.1, Requirement II.147

R_TMF518_RP_II_0068	The GCT feature must be supported by a generic cross-platform interface. The NML-EML Interface shall be supplemented by a client-server window system that would facilitate the actual launch of the GCT. The window system protocol (e.g. X-protocol) providing the solution for the actual launch of the GCT is outside the MTNM interface solution set. In order for the MTNM GCT interface to be truly platform independent it should not be based on the implementation details relevant to any specific window system.
Source	TMF 513 v3.1, Requirement II.148

R_TMF518_RP_II_0069	EMS GCT functionality on NMS is the same as the one available to an EMS client user when invoked within the EMS. The security issues, however, are not currently addressed in this interface (at this release bilateral agreement is required, in the future security issues may be defined as a part of the interface). When there is a bilateral agreement regarding the definition of users the following applies: The EMS allows control of the various EMS features depending on capability level of the user. Within a given window context, user access should be limited only to those operations, which are allowed given the (EMS) user info. User identification may have the following behavior (based on bilateral agreement): As long as at least one GUI cut-through session is active, then the user remains logged in. For a less secure, seamless cut-through the EMS user login and password is maintained at the NMS (outside the scope of the Interface). When the cut-through session is invoked, then the login and password is passed across the Interface When there is a bilateral agreement regarding the definition of user capabilities the following applies: The EMS shall allow the restriction of user access by allowing an NMS to explicitly specify the user capability.

	(When there is no specified user id the capability value alone determines the functionality of the GCT.) Typical user capabilities that are envisioned are read-only and read-write. When both user identification and user capability are both agreed upon and both are specified then the user capability is to be applied as a further restriction to the capability implicit in the user information.
Source	TMF 513 v3.1, Requirement II.149

R_TMF518_RP_II_0070	The GUI Cut-through shall allow the context to be specified. The user can make a request for a specific object in different window contexts and the user's entry point into the EMS should be in the same context or window type as requested by the NMS. The only window context that must be supported is the Top-Level context. If the EMS does not provide data for a window type, the NMS will use the data for the top level window. The suggested optional window contexts are Fault, Configuration (software management/connection management), Accounting, Performance, Security and Systems Management. When the desired window context is unavailable for the given object scope, then a window context that contains and/or allows navigation into the requested context is provided. The interface will allow the NMS to retrieve all the GCT contexts that are supported by the EMS and all supported GCT window types.
Source	TMF 513 v3.1, Requirement II.150

R_TMF518_RP_II_0107	The GCT feature may apply to different objects managed within each EMS, i.e. the GCT request must have a scope (either EMS or ME). The scope field in the GCT data record reflects the supported scope of the GCT operation. When the GCT is not implemented for the requested object scope, the EMS will launch the GCT of the closest superior object available (this information is available through the hierarchy of the object name). For example, if a certain GCT is unavailable for the requested ME then a GCT (of the same context) should be launched for the EMS instead. The NMS should request the narrowest scope desired. Alternatively the NMS shall only request a scope that is known to be supported according to the profile/GCT capability information received.
Source	TMF 513 v3.1, Requirement II.151

R_TMF518_RP_II_0072	If possible, the EMS should be able to actively manage the GCT application windows. 1. Integrate the EMS window within NMS window hierarchy (e.g. associate it to the NMS main window on creation of

	the GCT window) if made possible by the window system protocol. 2. The EMS should be able to close all of the GCT windows upon request by the NMS, or notify the NMS that closing of the GCT is disabled. Optional based on EMS.
Source	TMF 513 v3.1, Requirement II.152

3.3.5 Software and Data Control

R_TMF518_RP_II_0073	The Interface shall allow the requesting OS to request the target OS to backup the data of the Managed Element (ME) specified by the requesting OS.
Source	TMF 513 v3.0, Requirement II.229

R_TMF518_RP_II_0074	The Interface shall allow the requesting OS to retrieve the current status of the database backup for a Managed Element (ME) specified by the requesting OS.
Source	TMF 513 v3.0, Requirement II.230

R_TMF518_RP_II_0075	The Interface shall allow the requesting OS to request the target OS abort the database backup for a Managed Element (ME) specified by the requesting OS.
Source	TMF 513 v3.0, Requirement II.231

R_TMF518_RP_II_0076	The Interface shall allow the requesting OS to retrieve the names of all the database backups for a list of Managed Element (ME) names specified by the requesting OS that are available on the target OS.
Source	TMF 513 v3.0, Requirement II.232

3.3.6 Termination Point Control

R_TMF518_RP_II_0077	The Interface shall allow the requesting OS to create a Group Termination Point (GTP) in the target OS, given the requesting OS specified data listed in R_TMF518_RP_II_0078
Source	TMF 513 v3.0, Requirement II.164

R_TMF518_RP_II_0078	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS creates

	a Group Termination Point (GTP): 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS to check whether the user label is unique amongst the GTPs within the target OS. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Contained CTPs This parameter shall represent a list of the names of the Connection Termination Point (CTP) that are to be contained by the GTP. 5. Starting CTP name In cases where the CTPs are contiguous and of the same layer rate, this parameter shall indicate the first CTP in the group. This parameter is used in lieu of the Contained CTPs parameter. 6. Number of CTPs This parameter is used in conjunction with the starting CTP name parameter. It shall indicate the number of contiguous CTPs that follow the first CTP in the group. It equals 1 minus the total number of CTPs in the GTP. 7. GTPEffort This parameter is used to indicate if the GTP bandwidth specification can be met with a different list of CTPs 8. isReportingAlarms This parameter indicates whether alarm reporting for the GTP to be created is administratively activated or de-activated. True = alarm reporting is activated; False = alarm reporting is de-activated. 9. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001 10. Name Refer to R_TMF518_FMW_I_0001 11. AliasNameList Refer to R_TMF518_FMW_I_0001 12. Network Access Domain
Source	TMF 513 v3.0, Requirement II.165

R_TMF518_RP_II_0079	The Interface shall allow the requesting OS to delete a Group Termination Point (GTP) given the GTP name specified by the

	requesting OS from the target OS.
Source	TMF 513 v3.0, Requirement II.166

R_TMF518_RP_II_0080	The Interface shall allow the requesting OS to add or remove Connection Termination Pont (CTP)s to/from a Group Termination Point (GTP) given the CTP names and the GTP name specified by the requesting OS.
Source	TMF 513 v3.0, Requirement II.167

R_TMF518_RP_II_0081	The Interface shall allow the requesting OS to create a Termination Point Pool (TP Pool) in the target OS, given the requesting OS specified data listed in R_TMF518_RP_II_0082.
Source	TMF 513 v3.0, Requirement II.264

R_TMF518_RP_II_0082	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS create a Termination Point Pool (TP Pool): 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS to check whether the user label is unique amongst the TP Pools within the target OS. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Containing Subnetwork This parameter shall represent the name of Subnetwork containing the TP Pool. 5. Contained members This parameter shall represent a list of the names of the Termination Point (TP)s or Group Termination Point (GTP) s, all taken from MEs that belong to the above-specified Subnetwork, that are to be contained by the TP Pool. 6. Layered transmission parameters This parameter shall represent the common layers and transmission parameters the above-specified Contained TPs, or TPs contained in Contained GTPs, are required to have (e.g., ATM VP layer with prescribed traffic characteristics). Note: This does not affect the values of the layered transmission parameters of the contained TPs. 7. Description of use

	This attribute shall describe the specific use of the TP pool, in particular how its members are collected and administered. 8. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001 9. Name Refer to R_TMF518_FMW_I_0001 10. AliasNameList Refer to R_TMF518_FMW_I_0001 11. Network Access Domain
Source	TMF 513 v3.0, Requirement II.265

R_TMF518_RP_II_0083	The Interface shall allow the requesting OS to delete a Termination Point Pool (TP Pool) given the TP Pool name specified by the requesting OS from the target OS.
Source	TMF 513 v3.0, Requirement II.266

R_TMF518_RP_II_0084	The Interface shall allow the requesting OS to add or remove Termination Point (TP) s or Group Termination Point (GTP) s to or from a TP Pool given TP or GTP names and the TP Pool name specified by the requesting OS.
Source	TMF 513 v3.0, Requirement II.267

R_TMF518_RP_II_0085	The Interface shall allow the requesting OS to request that a Connection Termination Point (CTP) specified by the requesting OS be terminated and mapped. This request, if successful, will configure the CTP specified by the requesting OS such that it will then be capable of supporting lower rate connections. If the CTP is successfully configured such that it is capable of supporting lower rate connections, then the CTP's mapping mode should indicate as such. Note that before carrying out the requesting OS' request, the target OS should confirm that the CTP specified by the requesting OS is capable of being terminated and mapped and that it is not involved in an active cross-connection at the CTP's rate. Some examples: • Termination and mapping of an STS1 CTP such that it will support VT1.5 CTPs. • Termination and mapping of a T3 CTP such that it will support T1 CTPs.
Source	TMF 513 v3.0, Requirement II.068

R_TMF518_RP_II_0086	The Interface shall allow the requesting OS to request that a Connection Termination Point (CTP) specified by the requesting OS no longer be terminated and mapped. This request, if successful, will configure the CTP specified by the requesting OS such that it will then be capable of supporting cross-connections at the CTP's rate. If the CTP is successfully configured such that it is capable of supporting cross-connections at the CTP's rate, then the CTP's mapping mode should indicate as such. Note that before carrying out the requesting OS' request, the target OS shall confirm that the CTP specified by the requesting OS is not supporting an active Cross Connection (CC) at a client layer rate (e.g., for A requesting OS specified STS1 CTP, target OS should confirm that no contained VT1.5 CTPs are involved in an active cross-connection).
Source	TMF 513 v3.0, Requirement II.069

R_TMF518_RP_II_0088	The Interface shall allow the requesting OS to provision Termination Point (TP) transmission parameters. Refer to supporting document SD1-16_LayeredParameters for the currently defined set of supported TP parameters.
Source	TMF 513 v3.0, Requirement II.072

R_TMF518_RP_II_0089	Only applicable to ATM technology. The Interface shall allow the requesting OS to configure terminated and mapped VP (or VC) Connection Termination Point (CTP) s at the end of a VP (or VC) trail. A terminated (and available for mapping) VP or VC CTP can be turned to non terminated (nor available for mapping), (and can therefore be deleted on the NE if required), only if not used as a server by other lower level CTPs (e.g., a terminated VP CTP can be deleted only if does not carry any VC CTPs).
Source	TMF 513 v3.0, Requirement II.073

R_TMF518_RP_II_0090	The Interface shall allow the requesting OS to create a Floating Termination Point (FTP) given the requesting OS specified data

	listed in R_TMF518_RP_II_0091.R_TMF518_RP_II_0081 R_TMF518_RP_II_0002 R_TMF518_RP_II_0003 R_TMF518_RP_II_0004 R_TMF518_RP_II_0005 R_TMF518_RP_II_0006 R_TMF518_RP_II_0007 R_TMF518_RP_II_0008 R_TMF518_RP_II_0009 R_TMF518_RP_II_0010 R_TMF518_RP_II_0011 R_TMF518_RP_II_0012 R_TMF518_RP_II_0013 R_TMF518_RP_II_0014 R_TMF518_RP_II_0015 R_TMF518_RP_II_0016 R_TMF518_RP_II_0017 R_TMF518_RP_II_0018 R_TMF518_RP_II_0019 R_TMF518_RP_II_0020 R_TMF518_RP_II_0021 R_TMF518_RP_II_0022 R_TMF518_RP_II_0023 R_TMF518_RP_II_0024 R_TMF518_RP_II_0025 R_TMF518_RP_II_0026

	R_TMF518_RP_II_0027 R_TMF518_RP_II_0028 R_TMF518_RP_II_0029 R_TMF518_RP_II_0030 R_TMF518_RP_II_0031 R_TMF518_RP_II_0032 R_TMF518_RP_II_0033 R_TMF518_RP_II_0034 R_TMF518_RP_II_0035 R_TMF518_RP_II_0036 R_TMF518_RP_II_0037 R_TMF518_RP_II_0038 R_TMF518_RP_II_0039 R_TMF518_RP_II_0040 R_TMF518_RP_II_0041 R_TMF518_RP_II_0042 R_TMF518_RP_II_0043 R_TMF518_RP_II_0044 R_TMF518_RP_II_0045 R_TMF518_RP_II_0046 R_TMF518_RP_II_0047 R_TMF518_RP_II_0048 R_TMF518_RP_II_0049 R_TMF518_RP_II_0050 R_TMF518_RP_II_0051 R_TMF518_RP_II_0052

	R_TMF518_RP_II_0053 R_TMF518_RP_II_0054 R_TMF518_RP_II_0055 R_TMF518_RP_II_0056 R_TMF518_RP_II_0057 R_TMF518_RP_II_0058 R_TMF518_RP_II_0059 R_TMF518_RP_II_0060 R_TMF518_RP_II_0061 R_TMF518_RP_II_0062 R_TMF518_RP_II_0063 R_TMF518_RP_II_0064 R_TMF518_RP_II_0065 R_TMF518_RP_II_0066 R_TMF518_RP_II_0067 R_TMF518_RP_II_0068 R_TMF518_RP_II_0069 R_TMF518_RP_II_0070 R_TMF518_RP_II_0072 R_TMF518_RP_II_0073 R_TMF518_RP_II_0074 R_TMF518_RP_II_0075 R_TMF518_RP_II_0076 R_TMF518_RP_II_0077 R_TMF518_RP_II_0078 R_TMF518_RP_II_0079

	R_TMF518_RP_II_0080 R_TMF518_RP_II_0082 R_TMF518_RP_II_0083 R_TMF518_RP_II_0084 R_TMF518_RP_II_0085 R_TMF518_RP_II_0086 R_TMF518_RP_II_0088 R_TMF518_RP_II_0089 R_TMF518_RP_II_0090 R_TMF518_RP_II_0091 R_TMF518_RP_II_0092 R_TMF518_RP_II_0093 R_TMF518_RP_II_0094 R_TMF518_RP_II_0095 R_TMF518_RP_II_0097 R_TMF518_RP_II_0098 R_TMF518_RP_II_0099 R_TMF518_RP_II_0100 R_TMF518_RP_II_0101 R_TMF518_RP_II_0102 R_TMF518_RP_II_0103 The target OS will attempt to fulfill the request including the creation of the appropriate fragment CTPs if applicable (e.g., VCAT or LAG).
Source	TMF 513 v3.0, Requirement II.293

R_TMF518_RP_II_0091	The Interface shall allow the requesting OS to specify the following parameters when it requests the creation of a Floating Termination Point (FTP) object. 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS that the value of the user label attribute must be unique amongst the FTPs within the target OS domain. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Network Access Domain This attribute represents the Network Access Domain (NAD) to which the new FTP shall be assigned to. 5. Equipment name Equipment hosting the new FTP in the ME. 6. Ingress TMD This attribute shall represent the name of the ingress Transmission Descriptor (TMD) associated with this FTP. 7. Egress TMD This attribute shall represent the name of the egress Transmission Descriptor (TMD) associated with this FTP. 8. tpMappingMode (TerminationMode) Within the façade definition, the CTP/FTP can act as an aggregate of associated G.805 TCPs, G.805 Termination Functions and G.805 CPs at one or more LayerRates. The CTP is contained within a PTP or FTP. The tpMappingMode attribute indicates and controls the connection of the named CP at a specified LayerRate to the dedicated G.805 TCP and associated G.805 Termination Function at the same LayerRate within the CTP/FTP. 9. Layered transmission parameters This parameter represents the client layer rate (e.g., LR_Ethernet), the layer rate (e.g., LR_Encapsulation, LR_Fragment, LR_LAG_Fragment) and the server layer rate (e.g., VC12, any supported concatenated layer rate, including LR_Fragment) of the new CPTP. A list of technology-specific transmission parameters is associated to every layer rate. Refer to the supporting documents SD1-18_layers and SD1-16_LayeredParameters for details of the currently defined layer rates and transmission parameters. Note: Virtual Concatenation (VCAT) is indicated by the presence

	of the LR_Fragment layer rate and its associated layered transmission parameters (e.g., number and rate of the server CTPs), and that Link Aggregation (LAG) is indicated by the presence of the LR_LAG_Fragment layer rate and its associated layered transmission parameters. 10. Name Refer to R_TMF518_FMW_I_0001. 11. AliasNameList Refer to R_TMF518_FMW_I_0001 12. Direction (Directionality) This attribute identifies the required direction of the Floating Termination Point.
Source	TMF 513 v3.0, Requirement II. 299

R_TMF518_RP_II_0092	The Interface shall allow the requesting OS to delete a Floating Termination Point (FTP) given the requesting OS specified FTP name from the target OS. The deletion request shall fail if: • the FTP or any of its contained server CTPs (VCAT and LAG case) is an Subnetwork Connection (SNC) endpoint, or • if the FTP cannot be explicitly deleted (e.g., was automatically created by the target OS). • if the FTP is a CPTP whose role is "*assigned*" or "*fdEdge*".
Source	TMF 513 v3.0, Requirement II. 294

3.3.7 Transmission Descriptor Control

R_TMF518_RP_II_0093	The Interface shall allow the requesting OS to create a Transmission Descriptor (TMD) in the target OS given the requesting OS specified data listed in R_TMF518_RP_II_0094
Source	TMF 513 v3.0, Requirement II.190

R_TMF518_RP_II_0094	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS creates a Transmission Descriptor (TMD):

	1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This attribute shall indicate to the target OS that the value of the user label attribute must be unique amongst the TMDs within the target OS. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Layered transmission parameters This attribute shall represent a list of transmission parameters which can be set and/or retrieved at a specified layer on a TP having this TMD assigned as egress or ingress TMD. Specific parameters include, for example, frame format, line code, alarm reporting control (enable/disable), TP service state (In Service, Out Of Service etc.). 5. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001 This attribute shall represent any additional information parameters which can be set on the Termination Point (TP) having this TMD assigned as an egress or ingress TMD. 6. ContainingTmdRef This attribute contains the name of another Transmission Descriptor (TMD) which is considered to contain this TMD. The semantics of the containment is that the TMD shall inherit the layered Transmission Parameters and additional Object information from the containing TMD. 7. External representation This attribute shall represent a reference to the external representation of the TMD (e.g., an XML file name). The content of this information is opaque at the Interface and not utilized. 8. Name Refer to R_TMF518_FMW_I_0001. 9. AliasNameList Refer to R_TMF518_FMW_I_0001. 10. Network Access Domain
Source	TMF 513 v3.0, Requirement II.191

R_TMF518_RP_II_0106	The Interface shall allow the requesting OS to modify a Transmission Descriptor (TMD) in the target OS given the NMS specified data listed in the updated R_TMF518_RP_II_0094 for the creation of a TMD. When transmission parameters are modified, this will

	automatically modify the corresponding parameters in all associated TPs/MFDs on a best effort basis. TPs/ MFDs whose parameters could not be modified shall be returned by the target OS. Note: Only the modified parameters will be updated in the TPs/MFDs; i.e., the TMD parameter/value list may be inconsistent with the corresponding parameter/ value list of the associated TPs/MFDs.
Source	TMF 513 v3.1, Requirement II.353

R_TMF518_RP_II_0095	The Interface shall allow the requesting OS to delete a Transmission Descriptor (TMD) given the requesting OS specified TMD name. The target OS shall refuse/fail this request if any Termination Point (TP)s are associated with this TMD.
Source	TMF 513 v3.0, Requirement II.192

3.3.8 Assignment of Transmission Descriptor (TMD)s

R_TMF518_RP_II_0097	The Interface shall allow the requesting OS to associate an ingress and/or egress Transmission Descriptor (TMD) with a given Termination Point (TP) identified by its TP name specified by the requesting OS.
Source	TMF 513 v3.1, Requirement II.194

R_TMF518_RP_II_0098	The assignment of a Transmission Descriptor (TMD) to a Termination Point (TP) or Matrix Flow Domain (MFD) by using the TMD's name amounts to an overwriting of the layered transmission parameters of the TP or MFD by the layered transmission parameters of the TMD and to an overwriting of the additional information parameters of the TP or MFD by the additional Object information parameters of the TMD. Note that these parameters may also be set according to R_TMF518_RP_II_0088 and without using a TMD. Current parameters of the TP or MFD that are not present as parameters of the TMD are left unchanged by the TMD assignment. The unassignment of a TMD from a TP or MFD (by using the empty TMD name) has no effect on the parameters of the TP or MFD, i.e. the layered transmission parameters and additional info parameters of the TP or MFD remain unchanged.
Source	TMF 513 v3.0, Requirement II.277

R_TMF518_RP_II_0099	The Interface shall allow the requesting OS to validate the Transmission Descriptor (TMD)s assigned to a Termination Point (TP) or Matrix Flow Domain (MFD) given a TP or MFD name specified by the requesting OS. The following is provided for clarification: The assignment of a Transmission Descriptor (TMD) to a Termination Point (TP) or Matrix Flow Domain (MFD) is called consistent, if whenever a TMD transmission parameter is also present as a TP transmission parameter or a TMD additional Object information parameter is also present as a TP or MFD additional information parameter and the common parameters of the TMD and TP or MFD have the same values.
Source	TMF 513 v3.1, Requirement II.278

3.3.9 Topological Link Control

R_TMF518_RP_II_0100	The Interface shall allow the requesting OS to create a Topological Link (TL) in the target OS, OS given the requesting OS specified data listed in R_TMF518_RP_II_0101.
Source	TMF 513 v3.0, Requirement II.168

R_TMF518_RP_II_0101	The Interface shall allow the requesting OS to specify the following parameters when it requests that a target OS create a Topological Link (TL): 1. User label Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 2. User label uniqueness This parameter shall indicate to the target OS to check whether the user label is unique amongst the TLs within the target OS. 3. Owner Refer to R_TMF518_FMW_I_0001 in the TMF518_FMW BA. 4. Directionality This attribute shall represent the directionality of the TL (bidirectional or unidirectional). 5. aEnd Termination Point (TP) This parameter shall represent the name of the aEnd TP of the TL. 6. zEnd Termination Point (TP)

	This parameter shall represent the name of the zEnd TP of the TL. 7. Layer Rate This parameter shall represent the layer rate of the TL. Refer to R_TMF518_NRB_I_0003 in the TMF518_NRB document. 8. Network Access Domain This attribute represents the Network Access Domain (NAD) to which this TL has been assigned. 9. Alarm reporting This attribute shall indicate whether alarm reporting for the TL is enabled or disabled. 10. Alarm severity assignment profile This attribute indicates the required assignment of an Alarm Severity Assignement Profile (ASAP) to the TL. 11. VendorExtensions / AdditionalInfo Refer to R_TMF518_FMW_I_0001 12. Name Refer to R_TMF518_FMW_I_0001 13. AliasNameList Refer to R_TMF518_FMW_I_0001
Source	TMF 513 v3.0, Requirement II.169

R_TMF518_RP_II_0102	The Interface shall allow the requesting OS to delete a Topological Link (TL) given the TL name specified by the requesting OS.
Source	TMF 513 v3.0, Requirement II.170

3.4 Category III: Abnormal or Exception Conditions, Dynamic Requirements

See section 4.1.1 in the TMF518_FMW for the complete list of exceptions that may be raised by a target OS in response to a requesting OS.

3.5 Category IV: Expectations and Non-Functional Requirements

No requirements have been identified for this category.

3.6 Category V: System Administration Requirements

No requirements have been identified for this category.

4 Use Cases

4.1 Connection Control

4.1.1 The requesting OS creates a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0001
Use Case Name	The requesting OS creates a Subnetwork Connection (SNC)
Summary	This use case describes how a target OS can create either a point-to-point SNC or the leg of a broadcast connection. This use case requires that the target OS supports the Pending state for SNCs. The target OS is required to create an SNC of the specified SNC type, layer rate, directionality and grade of impact between the specified A end and Z end termination points. If the target OS cannot meet any of these parameters, an appropriate exception is raised. If non-network routed protocols are used for route determination and the requesting OS did not request network routing: The target OS creates an SNC of the specified static protection level that best matches the specified protection effort. If the protection effort is *same*, the target OS creates an SNC with the specified static protection level. If the protection effort is *sameOrBetter* or *sameOrWorse*, the target OS first attempts to provide the protection level requested. If it is not possible, the target OS attempts to provide better or worse static protection according to the protection effort parameter. However, if the requesting OS requests static protection level *partiallyProtected* with protection Effort *sameOrBetter*, the target OS may attempt to provide *fullyProtected* first. If the requesting OS specifies routing constraints in the request and the target OS supports this feature, the target OS is required to either include or exclude specified resources during route selection based on the provided criteria even if there are pending, partial, or active SNCs using the required parts of the route.The requesting OS may specify the full route in the request. If the requesting OS does not specify routing constraints or does not provide a full route, then the target OS itself will select the full or partial route respectively. If the target OS does not support the routing constraints feature and routing constraints are specified in the request, or supports the feature but cannot satisfy the requesting OS routing criteria, an exception is raised. If the target OS cannot use the routing constraints for a BLSR

	case, *BLSRDirection* and *Timeslot* parameters may be used for route selection if specified by the requesting OS. If network routed protocols are used for route determination and the requesting OS requests network routing: The target OS requests the network determine the route of the SNC of the specified static protection level. If the requesting OS specifies routing constraints in the request and the network supports this feature, the target OS passes these constraints to the network. If the network does not support the routing constraints feature or supports the feature but cannot satisfy the requesting OS routing criteria, an exception is raised. If the target OS supports the capability to manage more than one route for the same SNC, then the route of the newly created SNC shall be the *intended* route. If an *exclusive* SNC has been specified, then the target OS must find a route that does not conflict or share CCs or CTPs with any other existing SNC route, in any administrative state. Once an *exclusive* (intended) route has been created by the target OS, any further creation operations in which conflicts are detected with the *exclusive* route shall raise an exception. For DWDM: A routing constraint for DWDM (frequency) can be specified similar to the use of timeslot for SONET/SDH. If an existing SNC respects all the conditions specified in the requesting OS request, the target OS is allowed to return the existing SNC. It is also allowed to attempt to create a different SNC. For rerouting behavior, please see UC_TMF518_RP_0004 and UC_TMF518_RP_0005. Where applicable, the exception contains a list of the failed cross-connections and the reason(s) for failure. Notes: From a target OS perspective, there is no limit on the number of SNCs in the pending state that use the same route. However, the target OS may limit the number of pending SNCs (depending on specific target OS implementation).
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends the SNC creation request to the target OS.
Description	1. The requesting OS sends the request to create an SNC to the target OS.

	2. The target OS validates the request. If the request is not valid (invalid parameters, or the target OS does not support Pending SNC state), an exception is raised. 3. If an SNC with the same properties as specified in the requesting OS request already exists, the target OS may return that SNC. 4. If the target OS does not support routing constraints and they are specified in the request, or any of the mandatory parameters cannot be satisfied, or the requesting OS requested user label uniqueness and the specified user label already exists in the target OS, an exception is raised. 5. If non-network routed protocols are used for route determination and the requesting OS did not request network routing the target OS determines the route of the requested SNC using the specified static protection level and protection effort and optional routing constraints. If the routing constraints are not specified or a complete route is not defined by the requesting OS, then the target OS itself selects a full or partial route respectively. 6. If a server SNC is required and the target OS has freedom to create the server SNC for this request, the server SNC is created. If the target OS does not have this level of freedom, an exception is raised. Note that a server SNC is defined to be a SNC on a server layer needed to establish the traffic for the (client) SNC to be created. 7. If network routed protocols are used for route determination and the requesting OS requested network routing the target OS creates the SNC (this does not involve the entering of the cross-connection(s) at the ME(s)). The target OS initializes the SNC with the specified parameters of the request and the route. The SNC state is Pending. 8. The target OS replies with a success indication. 9. The target OS forwards a SNC object creation notification to the notification service.
Ends When	In case of success: 1. The requesting OS receives an indication of success of the action. 2. The target OS returns the created SNC name and its distinguishing information. In case of failure: The requesting OS receives an indication of failure of the action or exception.
Post-Conditions	In case of success: 1. The SNC has been created and not activated. 2. The target OS has forwarded a SNC object creation notification

	to all the OSs having registered to be notified 3. If generic end point(s) were specified, then the defined end point(s) are replied. In case of failure: The SNC has not been created.
Exceptions	1. Internal error: The requested operation could not be performed. 2. Not implemented: The target OS does not support this service. 3. Invalid input: At least 1 of the CTP references is invalid. 4. Invalid input: At least 1 of the CTP parameters is invalid. 5. Invalid input: In case a bundled SNC is requested, the GTP endpoints of the SNC do not match, i.e., the GTPs do not have the same number of CTPs and in a particular order with respect to their layer rates or are not of the same layer rate 6. Protection effort not met: The target OS cannot create a SNC with the requesting OS-specified static protection level and protection effort. 7. Unsupported routing constraints: The SNC can not be created because of cross-connection or CTP conflicts with other SNCs. 8. Unsupported routing constraints: Timeslot conflicts with other SNCs. 9. Unable to comply: The number of total pending SNCs has exceeded the maximum limit. The limit is dependent on target OS implementation. 10. Unsupported routing constraints: The target OS does not support routing constraints specified. 11. User label in use: The user label uniqueness constraint is not met. 12. Unsupported routing constraints: SNC cannot be created because of conflict with another active or partial SNC. 13. Object in use: The intended route is in conflict with an “exclusive” route of another SNC. 14. Comm Loss (Communication loss). 15. Entity not found
Traceability	R_TMF518_RP_II_0002, R_TMF518_RP_II_0003, R_TMF518_RP_II_0005 This use case is a generalization of Use Case 5.6.1 from TMF 513 v3.0

4.1.2 The requesting OS activates a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0002
Use Case Name	The requesting OS activates a Subnetwork Connection (SNC)
Summary	The requesting OS requests to activate a subnetwork connection (SNC). An SNC can be activated while in any state. If transmission parameters are specified for A end or Z end CTPs, the target OS will apply them either before or after the creation of the cross-connections, as appropriate. The alarm reporting on the CTPs and the containing TPs may be turned on by the target OS, unless otherwise specified via the alarm reporting transmission parameter. An already activated SNC can be activated again. Where applicable, the exception or error reason contains a list of the failed CTPs and the reason(s) for failure. If the addressed SNC has more routes, this operation unlocks all the routes, delegating the target OS and/or the network (e.g. restoration process) the actual activation of more appropriate route
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends the SNC activation request to the target OS.
Description	1. The requesting OS sends the request to activate the specified SNC to the target OS. 2. The target OS validates the SNC reference (e.g. name). If the request is not valid, an exception is raised. 3. If the SNC is already activated, then the target OS replies with a success indication. The target OS may not send the commands to the NE a second time for the cross-connection establishment. However the commands may be sent for the transmission parameters. 4. If the SNC of the referenced name is pending and some contained cross-connections are on another (active) SNC, the target OS behavior depends on whether the SNC is a part of point-to-multipoint configuration or not. If the SNC is a part of point-to-multipoint configuration, the use case is carried out, otherwise an exception (resources occupied) is raised if cross-connection sharing is not supported. If no cross-connections to be established are on another active SNC, the use case is carried out. 5. If the specified aEnd and/or zEnd CTPs are not terminated and mapped at the appropriate connection rate, then the target OS behavior depends on the requesting OS specified target OS

	freedom. If the target OS does not have this level of freedom to terminate and map/un-terminate and un-map TPs, an exception is raised. If the requesting OS specified this level of freedom in the request, the target OS either terminates and maps containing or un-terminates and un-map the contained termination points of the aEnd and/or zEnd CTPs. Refer to UC_TMF518_RP_0037_ and UC_TMF518_RP_0038. 6. If the TP parameters were specified for at least 1 of the A-end and/or Z-end CTPs then the target OS may provision the parameters on these CTPs before activation of the cross-connections. Refer to UC_TMF518_RP_0007, UC_TMF518_RP_0008 and UC_TMF518_RP_0009. The target OS may put all the containing TPs in service. 7. If the requesting OS specified this level of freedom in the request and a server SNC is involved and is not active, the server SNC is activated. 8. The target OS initiates the activation of the SNC (which involves entering the cross-connection(s) at the ME(s)). 9. If TP parameters have not been applied before activation of cross-connections, they are applied now. See step #6. 10. If network routed protocols are used for route determination the target OS requests that the network determine the route of the requested SNC using the specified static protection level and protection effort and optional routing constraints. 11. If all of the cross-connections comprising the SNC have been established, then the target OS sets the SNC state of the SNC to active. The target OS updates the connection state of the affected CTPs to either sink, source, or bidirectionally connected. If there are more routes, then it is up to target OS the choice of better route to activate. Once a route has been successfully activated (its actual state is “active”) then the SNC is enabled to transit to active state. 12. If network routed protocols are used for route determination and a complete route cannot be determined: • If the requesting OS requests that rerouting can be performed. The SNC state shall be set to partial and the network attempts to determine a new route for the SNC. Refer to UC_TMF518_RP_0004 and UC_TMF518_RP_0005 • If the requesting OS requests that rerouting cannot be performed. The SNC state shall be set to partial. 13. If not all cross-connections comprising the SNC have been established(*), then the target OS sets the SNC state of the SNC to partial. The target OS updates the connection state of those CTPs that were successfully established to either sink, source, or bidirectionally connected. The target OS replies with a failure

	indication and error reason. (*) in case of multi-route SNC, this means that the target OS (or the network) was not able to activate any (just unlocked) route of the SNC 14. If there has been a SNC state change, then the target OS forwards a SNC state change notification to the notification service. 15. If there has been a CTP connection state change, then the target OS forwards a CTP state change notification to the notification service. 16. If there are error conditions, (e.g. failure to provision TP transmission parameters) existing after establishing all the CCs, then the target OS will reply with a failure indication and error reason. In this case the SNC state will be active. Otherwise if there are no error conditions the target OS replies with a success indication.
Ends When	In case of success: The requesting OS receives an indication of success of the action. In case of failure: The requesting OS receives an indication of failure of the action.
Post-Conditions	In case of success: 1. The SNC has been activated. 2. If the target OS has provisioned TP parameters, refer to the post-conditions as specified in UC_TMF518_RP_0039. 3. If the target OS has performed terminate and map or un-terminate and un-map, refer to the post-conditions (in case of success) as specified in to UC_TMF518_RP_0037 and UC_TMF518_RP_0038. 4. If there has been a change in the SNC state of the SNC, then the target OS will forward a state change notification for a SNC state to the requesting OS. 5. If there has been a CTP connection state change, then state change notification(s) are emitted to the requesting OS. In case of failure: 1. The SNC has not been completely activated, (i.e. zero or more cross-connections comprising of the SNC have been activated). 2. If the target OS has successfully provisioned TP parameters and failed to activate the SNC, refer to the post-conditions as specified in UC_TMF518_RP_0039. 3. If the target OS has successfully performed terminate and map or un-terminate and un-map and failed to activate the SNC, refer

	to the post-conditions as specified in to UC_TMF518_RP_0037 and UC_TMF518_RP_0038. 4. If there has been a change in the SNC state of the SNC as a result of a failure, then the target OS will forward a state change notification for a SNC state to the requesting OS. 5. If there has been a CTP connection state change as a result of the failure, then state change notification(s) are sent to the requesting OS.
Exceptions	1. Invalid input: Any input parameter is syntactical incorrect. 2. At least 1 of the CTP references failed to be provisioned. See exception list from UC_TMF518_RP_0039. 3. Not in valid state: At least 1 of the containing termination points of the CTP references failed to be terminated and mapped or contained CTP(s) failed to be un-terminated and unmapped. See exception list from UC_TMF518_RP_0037 and UC_TMF518_RP_0038. 4. Comm Loss (Communication loss). 5. Object in use: CC or CTP conflicts between the active route (with equal or higher priority) of this and other SNCs or when CC creation would involve a TP that has an existing fixed CC that does not match that required for the SNC. 6. Timeslot in use: Timeslot conflicts with other SNCs. 7. Unable to comply: The SNC is in pending state and is in conflict with another active or partial SNC. 8. Entity not found: The SNC reference (e.g. name) or one or more TP references reference objects that do not exist. 9. Not implemented: The target OS does not support this service. 10. Internal error: The requested operation could not be performed.
Traceability	R_TMF518_RP_II_0007, R_TMF518_RP_II_0008 This use case is a generalization of Use Case 5.6.2 from TMF 513 v3.0

4.1.3 The requesting OS creates and activates a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0003
Use Case Name	The requesting OS creates and activates a Subnetwork Connection (SNC)
Summary	This operation provides a way to create and activate a point-to-point subnetwork connection or a leg of a broadcast configuration in one

	request. The target OS is required to create an SNC of the specified SNC type, layer rate, directionality and grade of impact between the specified A end and Z end termination points. If the target OS cannot meet any of these parameters, an appropriate exception is raised. If non-network routed protocols are used for route determination and the requesting OS did not request network routing: The target OS creates an SNC of the specified static protection level that best matches the specified protection effort. If the protection effort is *same*, the target OS creates an SNC with the specified static protection level. If the protection effort is *sameOrBetter* or *sameOrWorse*, the target OS first attempts to provide the protection level requested. If it is not possible, the target OS attempts to provide better or worse static protection according to the protection effort parameter. However, if the requesting OS requests static protection level *partiallyProtected* with protection Effort *sameOrBetter*, the target OS may attempt to provide *fullyProtected* first. If the requesting OS specifies routing constraints in the request and the target OS supports this feature, the target OS is required to either include or exclude specified resources during route selection based on the provided criteria even if there are pending, partial, or active SNCs using the required parts of the route.The requesting OS may specify the full route in the request. If the requesting OS does not specify routing constraints or does not provide full route, then the target OS itself will select the full or partial route respectively. If the target OS does not support the routing constraints feature and routing constraints are specified in the request, or supports the feature but cannot satisfy the requesting OS routing criteria, an exception is raised. If the target OS cannot use the routing constraints for a BLSR case, *BLSRDirection* and *Timeslot* parameters may be used for route selection if specified by the requesting OS. If network routed protocols are used for route determination and the requesting OS requests network routing: The target OS requests the network determine the route of the SNC of the specified static protection level. If the requesting OS specifies routing constraints in the request and the network supports this feature, the target OS passes these constraints to the network. If the network does not support the routing constraints feature or supports the feature but cannot satisfy the requesting OS routing criteria, an exception is raised. If the target OS supports the capability to manage more routes for the same SNC, then the route of the newly activated SNC is the INTENDED route. The activation implies that the route is unlocked. If EXCLUSIVE SNC has been specified, then the target OS must find

	a route that does not conflict or share CCs or CTPs with any other existing SNC route, in any administrative state. Once an EXCLUSIVE (intended) route has been created by target OS, any further creation operation which conflicts with the exclusive route shall be refused. For DWDM: A routing constraint for DWDM (frequency) can be specified similar to the use of timeslot for SONET/SDH. If transmission parameters are specified for A end or Z end CTPs, the target OS will apply them either before or after the creation of the cross-connections, as appropriate. The alarm reporting on the CTPs and the containing TPs may be turned on by the target OS, unless otherwise specified via the alarm reporting transmission parameter. If the pending state is supported, it is possible for the SNC to be created but activation to be rejected if conflicting active or partial SNCs, the resulting SNC will be in pending state. If the pending state is not supported, then this is not possible and the SNC will not be created if activation is rejected. If an existing SNC respects all the conditions specified in the requesting OS request, the target OS is allowed to return the existing SNC. It is also allowed to attempt to create and activate a different SNC. For rerouting behavior, refer to UC_TMF518_RP_0004 and UC_TMF518_RP_0005. Where applicable, the exception contains a list of the failed cross-connections and the reason(s) for failure. **Note:** 1. From a target OS perspective, there is no limit on the number of SNCs in the pending state that use the same route. However, the target OS may limit the number of pending SNCs (depending on specific target OS implementation). 2. The target OS may not support pending SNCs at all. 3. For ATM specific behavior, refer to UC_TMF518_RP_0047.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends the SNC create and activate request to the target OS
Description	Case A: Pending SNC(s) not supported: 1. The requesting OS sends the request to create and activate a SNC to the target OS. 2. The target OS validates the request. If the request is not valid, an exception is raised.

	3. If an SNC with the same properties as specified in the requesting OS request already exists, the target OS may reuse that SNC. 4. If the target OS (or the network for network routed) does not support routing constraints and they are specified in the request, or any of the mandatory parameters cannot be satisfied, or the requesting OS requested user label uniqueness and the specified user label already exists in the target OS, an exception is raised. 5. If some of cross-connections to be established for a referenced SNC are on another, active or partial SNC, the target OS behavior depends on whether the requested SNC is to be a part of broadcast configuration or not. If the requested SNC is to be a part of the broadcast configuration, the use case is carried out, otherwise an exception (resources occupied) is thrown if cross-connection sharing is not supported. If none of the cross-connections to be established are on another, active or partial SNC, the use case is carried out. 6. If non-network routed protocols are used for route determination and the requesting OS did not request network routing, the target OS determines the route of the requested SNC using the specified static protection level, protection effort and optional routing constraints. If routing constraints are not specified or a complete route is not defined by the requesting OS, then the target OS itself selects a full or partial route respectively. 7. If network routed protocols are used for route determination and the requesting OS requests network routing, the network determines the route of the requested SNC using the specified static protection level and optional routing constraints. If routing constraints are not specified or a complete route is not defined by the requesting OS, then the network itself selects a the route. 8. If a server SNC is required and the target OS has freedom to create the server SNC for this request, the server SNC is created if it does not already exist and then activated. If the requesting OS did not specify this level of freedom in the request, an exception is raised. 9. If the TP parameters were specified for at least 1 of the A-end and/or Z-end CTPs in the original create request, then the target OS may provision the parameters on these CTPs before activation of cross-connections. Refer to UC_TMF518_RP_0039. 10. If the specified aEnd and/or zEnd CTPs are not terminated and mapped at the appropriate connection rate, then the target OS behavior depends on the requesting OS specified target OS freedom. If the target OS does not have this level of freedom to terminate and map/un-terminate and un-map TPs, an exception is raised. If the requesting OS specified this level of freedom in the request, the target OS either terminates and maps

	containing or un-terminates and un-maps the contained termination points of the aEnd and/or sEnd CTPs. Refer to R_TMF518_RP_II_0018.and R_TMF518_RP_II_0019. 11. The target OS initiates the activation of the SNC (which involves entering the cross-connection(s) at the ME(s)). 12. If TP parameters have not been applied before activation of cross-connections, they are applied now. See step #9. 13. If all of the cross-connections comprising the SNC have been established, then the target OS sets the SNC state of the SNC to active. The target OS updates the SNC state of the affected CTPs to either sink, source, or bidirectionally connected. The target OS replies with a success indication. The target OS forwards a SNC object creation notification to the notification service.. The target OS forwards a CTP connection state change notification for all affected CTPs to the notification service. 14. If there are error conditions, (e.g. failure to provision TP transmission) existing after establishing all the cross-connections, then the target OS will reply with a failure indication and error reason. In this case the SNC state of the SNC will be active. 15. If one or more (possibly all) of the cross-connections comprising the SNC have been established (*) then the target OS sets the SNC state of the SNC to partial. The target OS updates the connection state of those CTPs that were successfully established to either sink, source, or bidirectionally connected. The target OS replies with a failure indication and error reason. (*) in case of multi-route SNC, this means that the target OS (or the network) was not able to activate any (just unlocked) route of the SNC. 16. The target OS forwards a SNC object creation notification to the notification service. 17. The target OS forwards a CTP connection state change notification for all CTPs that were successfully established to the notification service. 18. If none of the cross-connections comprising the SNC have been established, then the target OS replies with a failure indication and error reason. The SNC is not created. Case B: Pending SNC(s) supported: 1. The requesting OS sends the request to create and activate an SNC to the target OS. 2. The target OS creates the SNC. Refer to UC_TMF518_RP_0001. 3. The target OS activates the SNC. Refer to UC_TMF518_RP_0002.

Ends When	Case A: Pending SNC(s) not supported: In case of success: 1. The SNC has been created and activated. 2. If the target OS has provisioned TP parameters, refer to the post-conditions as specified in UC_TMF518_RP_0039. 3. If the target OS has performed terminate and map or un-terminate and un-map, refer to the post-conditions (in case of success) as specified in R_TMF518_RP_II_0018.and R_TMF518_RP_II_0019. 4. The target OS has forwarded a SNC object creation notification to the notification service. 5. The target OS has forwarded CTP connection state change notification(s) to the notification service. In case of failure: 1. The SNC has not been created (i.e. invalid request or no cross-connections comprising the SNC were activated). 2. The SNC has been created but has not been completely activated, (i.e. one or more cross-connections comprising of the SNC have been activated). 3. If the target OS has successfully provisioned TP parameters and failed to activate the SNC, refer to the post-conditions as specified in UC_TMF518_RP_0039. 4. If the target OS has successfully performed terminate and map or un-terminate and un-map and failed to activate the SNC, refer to the post-conditions as specified in to R_TMF518_RP_II_0018.and R_TMF518_RP_II_0019. 5. If there has been a CTP connection state change as a result of the failure, then state change notification(s) are emitted to the notification service. Case B: Pending SNC(s) supported: Refer to UC_TMF518_RP_0001 and UC_TMF518_RP_0002.
Post-Conditions	Case A: Pending SNC(s) not supported: 1. At least 1 of the CTP references is invalid. 2. At least 1 of the CTP parameters is invalid. 3. The target OS cannot meet specified static protection level and protection effort for the referenced SNC. 4. The target OS does not support the routing constraints specified. 5. Timeslot conflicts with other SNCs. 6. The number of total pending SNCs has exceeded the maximum limit. The limit is dependent on target OS implementation.

	7. At least 1 of the CTP references failed to be provisioned. See exception list from UC_TMF518_RP_0039. 8. At least 1 of the containing termination points of the CTP references failed to be channelized or contained CTP(s) failed to be de-channelized. See exception list from R_TMF518_RP_II_0018.and R_TMF518_RP_II_0019. 9. Communications failure between the target OS and the ME(s) and this prevents creation and activation of the SNC. 10. The SNC is in conflict with another active or partial SNC and can not be created. 11. Cross-connection or CTP conflicts with other SNCs. 12. UserLabel uniqueness constraint is not met. 13. Non-specific target OS internal failure. Case B: Pending SNC(s) supported: 1. See exception list from UC_TMF518_RP_0001. 2. See exception list from UC_TMF518_RP_0002.
Exceptions	1. See exception list from UC_TMF518_RP_0001. 2. See exception list from UC_TMF518_RP_0002.
Traceability	R_TMF518_RP_II_0009, R_TMF518_RP_II_0010 This use case is a generalization of Use Case 5.6.3 from TMF 513 v3.0

4.1.4 The requesting OS adds a route to a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0004
Use Case Name	The requesting OS adds a route to a Subnetwork Connection (SNC)
Summary	The requesting OS requests to add a protection route to a given Subnetwork Connection in A target OS. As a result of (successful) completion of this request, the target OS shall add the new route to the Subnetwork Connection, but shall not attempt to establish (on NEs) any cross connections as side effect of this operation, because the route is created in locked state.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. An SNC exists
Begins When	The requesting OS sends an add route request to the target OS.
Description	1. It is possible to specify if the new added route becomes the

	intended one, and / or if it is exclusive. 2. It is possible to describe zero, more or all routing constraints, i.e. the whole route description.
Ends When	The target OS has completed the route creation.
Post-Conditions	The newly added route is available at the interface, in locked state.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Internal error: The requested operation could not be performed. 3. Invalid input: Any input parameter is syntactical incorrect. 4. Entity not found: Fields of input parameters reference objects that do not exist. 5. Protection effort not met: The requesting OS requests a route with a static protection level (inherited from SNC) that cannot be met by the target OS. 6. Unable to comply: The target OS is unable to find an appropriate route. 7. Object in use: The route is in conflict with an “exclusive” (even locked) route of another SNC. 8. Comm loss (Communication loss). 9. Unsupported routing constraints: The target OS does not support the routing constraints specified.
Traceability	R_TMF518_RP_II_0011, R_TMF518_RP_II_0012 This use case is a generalization of Use Case 5.6.4 from TMF 513 v3.0

4.1.5 The requesting OS removes a route from a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0005
Use Case Name	The requesting OS removes a route from a Subnetwork Connection (SNC)
Summary	The requesting OS requests to remove a protection route from a subnetwork connection. As a result of (successful) completion of this request, the target OS shall delete the protection route of addressed Subnetwork Connection. Of course it is possible to delete a locked backup route which is “in use” by other SNC route, because this operation has no side effect on routes of any other SNCs, even if sharing CCs/TPs.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed

	the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. An SNC exists 3. The addressed route must not be in the unlocked state 4. The addressed route must not be the intended route
Begins When	The requesting OS sends a remove route request to the target OS.
Description	The route is removed, so at least the SNC remains with only the intended route.
Ends When	The target OS has completed the route removal.
Post-Conditions	The route is no longer exists.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Internal error: The requested operation could not be performed. The target OS does not support the requested feature. 3. Invalid input: Any input parameter is syntactical incorrect. 4. Entity not found: Fields of input parameters reference objects that do not exist. 5. Not in valid state: The route is in the UNLOCKED state, or the route is the intended one. 6. Comm loss (Communication loss). 7. Entity not found.
Traceability	R_TMF518_RP_II_0013, R_TMF518_RP_II_0014 This use case is a generalization of Use Case 5.6.5 from TMF 513 v3.0

4.1.6 The requesting OS creates-modifies the route of a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0006
Use Case Name	The requesting OS creates-modifies the route of a Subnetwork Connection (SNC)
Summary	The requesting OS requests to modify a route of a Subnetwork Connection. As a result of (successful) completion of this request, the addressed SNC route is modified. If the SNC was in PENDING or PARTIAL state, then the state is unchanged. If the SNC was in ACTIVE state, then the output state is PARTIAL. In case the SNC has several routes, then the administrative state of the addressed route will always transit to LOCKED state.
Actor(s)	The requesting OS

Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a modify route request to the target OS.
Description	Two classes of modification are available: • add protection leg, remove protection leg • reroute It is possible to describe zero, more or all routing constraints, i.e. the whole route or leg description. At least, it must be possible to • add/remove a protection leg to/from a simple SNC • change from simple to add drop type and vice versa
Ends When	The target OS has completed the route modification.
Post-Conditions	The modified route is available at the interface, in locked state. If the current Route is the just modified one, then is not retrievable in the network.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Internal error: The requested operation could not be performed. 3. Invalid input: Any input parameter is syntactical incorrect. 4. Object in use: The SNC can not be created because of CC or CTP conflicts with other SNCs. 5. Entity not found: Fields of input parameters reference objects that do not exist. 6. Protection effort not met: The requesting OS requests a new SNC with a static protection level and protection effort that cannot be met by the target OS. 7. Unable to comply: The SNC cannot be created because it cannot comply with any of the input parameter constraints for a reason different than the ones above. 8. Unsupported routing constraints: The target OS does not support the routing constraints specified. 9. User label in use: The user label supplied by the requesting OS is already in use. 10. Comm loss (Communication loss).
Traceability	R_TMF518_RP_II_0015, R_TMF518_RP_II_0016 This use case is a generalization of Use Case 5.6.6 from TMF 513 v3.0

4.1.7 The requesting OS deactivates a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0007
Use Case Name	The requesting OS deactivates a Subnetwork Connection (SNC)
Summary	The requesting OS requests that an SNC be deactivated from the target OS' managed subnetwork (i.e., de-provisioned from the target OS' managed subnetwork). However, as a result of this action, the target OS continues to maintain the SNC object within the target OS. The deactivate operation requires that the target OS support the Pending state for SNCs. Deactivating an SNC implies deletion in the ME of all the non-shared cross-connects that belong to this SNC. The PTPs are left in the same service state and are not put out-of-service. Some examples of why a target OS would use this use case include: • To free-up resources in the underlying managed subnetwork yet maintain a record of the SNC such that it could be quickly reactivated. For example, this is useful if 2 or more SNC share resources at different times of the day. • To maintain a record of the SNC in the target OS such that the target OS maintains knowledge of the network resources (e.g., aEnd CTP(s), zEnd CTP(s), route, etc.) which would be allocated for the SNC. If the addressed SNC has more routes, this operation locks all the routes, delegating the target OS and/or the network (e.g. restoration process) the actual deactivation of all CCs which are not shared with (routes of) other SNCs.
Actor(s)	requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. The requesting OS must have knowledge of the identification of the SNC (e.g., the identifying name of the SNC).
Begins When	The requesting OS sends a request to the target OS to deactivate an SNC.
Description	1. The target OS validates the specified SNC. 2. The target OS initiates the deactivation of the SNC. SNC deactivation involves target OS communication with Managed Elements. The target OS attempts to remove, from the applicable Managed Elements, all the non-shared cross-connections which comprise this SNC. • An already deactivated SNC can be deactivated again with success (the target OS is not required to send the commands to the ME a second time, however). • While in SNCS_PARTIAL state, it is possible to deactivate an SNC again, this corresponds to a retry.

	3. If the target OS succeeds in deactivating the SNC (i.e., if all the non-shared cross-connections comprising the SNC on applicable Managed Elements have been removed), the state of the SNC changes to PENDING (or remains in PENDING if already in the PENDING state). The supporting CTPs are left in the same state and are not put out of service. The deactivation is successful even if some CCs representing fixed connectivity cannot be deleted. An SNC cannot be deactivated if it is composed solely of fixed cross-connects. • The target OS provides a success indication to the requesting OS. 4. If the target OS fails to deactivate the SNC, a failure indication is sent to the requesting OS, and: • If at least one, but not all of the non-shared cross-connections comprising the SNC on applicable Managed Elements have been removed, the SNC's state changes to PARTIAL (or remains in PARTIAL if already in the PARTIAL state); • If none of the non-shared cross-connections comprising the SNC on applicable Managed Elements have been removed, the SNC's state remains ACTIVE. 5. If there has been an SNC state change, the target OS generates a state change notification when the SNC's state has been changed and sends it to the notification service. 6. For any cross-connection that has been successfully removed as a result of the deactivate SNC action, the target OS generates a state change notification for the associated connection termination points (CTPs), that have transitioned to Not Connected. Note that the CTP may still be Connected in the opposite direction as part of another SNC. 7. The requesting OS can request that the target OS set TP transmission parameters as a side-effect of the deactivate SNC action. • The target OS should send AVC notifications for the successfully modified TPs. • If a given entry in the list of transmission parameters specified by the requesting OS can not be successfully applied to the TP, for any reason, then an error reason is returned to the requesting OS. • Existing TP transmission parameters for which no changes were requested will be left unchanged. • The alarm reporting on the CTPs and the containing TPs may be turned off by the target OS as part of this request, unless otherwise specified by the requesting

	OS 8. If a Server SNC was involved in this SNC, the Server SNC does not support any other Client SNCs, and all cross-connections comprising the referenced SNC have been removed, then the target OS behavior depends on whether the target OS has freedom to deactivate the Server SNC. If the target OS does not have this level of freedom, only the Client SNC is deactivated.
Ends When	In case of success: The requesting OS receives an indication of success of the request. In case of failure: 1. The requesting OS receives an indication of failure of the request, or 2. The request times out.
Post-Conditions	In case of success: 1. The network resources (CTPs) which had been used in the SNC are freed in the managed subnetwork (really, are no longer in active use by the SNC). 2. The target OS maintains the SNC object. 3. The SNC state is the PENDING state. 4. If a change of SNC state has occurred, the target OS has sent a state change notification to the Notification Service. 5. For any change of TP connection state, the target OS has sent a state change notification to the notification service. 6. For the successfully modified TPs, the target OS should send AVC notifications to the notification service In case of failure: 1. Some or all of the network resources (CTPs) associated with the SNC are still in active use by the SNC, as indicated in the TP Connection State. 2. The target OS maintains the SNC object. 3. The SNC state is either the ACTIVE (all network resources associated with the SNC are still in use by the SNC) or PARTIAL state (some network resources associated with the SNC are still in use by the SNC). 4. If a change of SNC state has occurred, the target OS sends a state change notification to the notification service. 5. For any change of TP connection state, the target OS has sent a state change notification to the notification service.
Exceptions	1. Not implemented: The target OS does not support this service.

	2. Internal error: The requested operation could not be performed. 3. Unable to comply: The SNC is fixed and can not be deactivated 4. The sncName provided by the requesting OS does not refer to an SNC object, or when any field in tpsToModify is invalid 5. The sncName or tpsToModify (provided by the requesting OS) references an object that does not exist 6. The SNC is fixed and can not be deactivated 7. Comm loss (Communication loss); the communication has been interrupted between the target OS and one or more of the underlying (applicable) Managed Elements. This exception is only used in the case where no change has been made to the SNC object or to any CTP. **Note:** Whenever an exception is raised, it can be assumed that no network changes have been made to the SNC.
Traceability	R_TMF518_RP_II_0026, R_TMF518_RP_II_0027 This use case is a generalization of Use Case 5.6.7 from TMF 513 v3.0

4.1.8 The requesting OS deletes a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0008
Use Case Name	The requesting OS deletes a Subnetwork Connection (SNC)
Summary	The requesting OS requests that an SNC be deleted in the target OS (i.e., SNC object maintained by the target OS be deleted as a result of this action). This use case also includes the deletion a leg from a broadcast system. The delete operation requires that the target OS supports the Pending state for SNCs. Some examples of why a target OS would use this use case include: • To delete a record of the SNC from the target OS to free-up target OS resources. • To delete a record of the SNC from the target OS such that the target OS no longer has any network resources (e.g., aEnd CTP(s), zEnd CTP(s), route, etc.) allocated for the SNC. If the SNC has more routes, then the operation deletes the SNC, its intended and all back-up route(s).
Actor(s)	The requesting OS

Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. The requesting OS must have knowledge of the identification of the SNC (e.g., the identifying name of the SNC). 3. The SNC must be in Pending state.
Begins When	The requesting OS sends a request to the target OS to delete an SNC.
Description	1. The target OS validates the SNC provided by the requesting OS as part of the request. 2. The target OS will only delete an SNC in the PENDING state. 3. The target OS initiates the deletion of the SNC in the target OS. SNC deletion is not expected to involve target OS communication with Managed Elements. • SNC deletion involves the target OS deleting the SNC object that the target OS maintains. If successful, the target OS provides a success indication to the requesting OS. • If the SNC is part of a broadcast system, the target OS only deletes the specified leg of the broadcast system and the other legs are left unchanged. 4. If the target OS fails to delete the SNC, a failure indication is sent to the requesting OS. The SNC state remains in PENDING. 5. The target OS generates an object deletion notification when the SNC object is deleted and sends it to the notification service. 6. If a Server SNC was involved in this SNC, and the Server SNC does not support any other Client SNCs, then the target OS behavior depends on whether the target OS has freedom to delete the Server SNC. If the target OS does not have this level of freedom, only the Client SNC is deleted.
Ends When	In case of success: The requesting OS receives an indication of success of the request. In case of failure: 1. The requesting OS receives an indication of failure of the request, or 2. The request times out, or 3. The requesting OS receives an indication that the request was rejected (e.g., if the SNC that was requested to be deleted was in other than the PENDING state).
Post-Conditions	In case of success:

	1. The SNC object within the target OS has been deleted. In case of failure: 1. The SNC object still exists within the target OS.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Internal error: The requested operation could not be performed. 3. Invalid input: Any input parameter is syntactically incorrect. 4. Entity not found: The sncName or tpsToModify (provided by the requesting OS) references an object that does not exist. 5. Unable to comply: The SNC is fixed and can not be deleted 6. Comm loss (Communication loss) **Note:** Whenever an exception is raised, it can be assumed that no network or target OS database changes have been made to the SNC.
Traceability	R_TMF518_RP_II_0028, R_TMF518_RP_II_0029 This use case is a generalization of Use Case 5.6.8 from TMF 513 v3.0

4.1.9 The requesting OS deactivates and deletes a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0009
Use Case Name	The requesting OS deactivates and deletes a Subnetwork Connection (SNC)
Summary	The requesting OS requests that an SNC be deactivated and deleted. As a result of successfully completing this use case: • The SNC will be de-provisioned from the target OS' managed subnetwork, and • The SNC object will be removed from the target OS. If PENDING SNC(s) are supported by the target OS, this use case first uses the requesting OS Deactivates a Subnetwork Connection use case. If that use case is successfully completed, then the requesting OS Deletes a Subnetwork Connection use case is employed. This use case assumes that the deletion of the SNC **will never fail if pending SNC(s) are not supported by the target OS**. It is up to the target OS to enforce this assumption. Some examples of why a target OS would use this use case include: • To accomplish via a single request: the freeing-up of resources in the underlying managed subnetwork • The deletion of the record of the SNC from the target OS to

	free-up target OS resources • The removal of any target OS knowledge of the network resources allocated for the SNC. If the addressed SNC has more routes, this operation locks all the routes, which means that target OS and/or the network (e.g. restoration process) have no more control over these routes. All the currently active CCs for this SNC shall be removed, of any (active or partial) route. Then the operation deletes the SNC, its intended and all back-up route(s).
Actor(s)	The requesting OS
Pre-Conditions	The pre-conditions of UC_TMF518_RP_0007.
Begins When	The requesting OS sends a request to the target OS to deactivate and delete an SNC.
Description	Case A: Pending SNC(s) not supported 1. The target OS validates the specified SNC. If the request is invalid, an exception is raised. 2. The target OS initiates the deactivation of the SNC. SNC deactivation involves target OS communication with managed elements. The target OS attempts to remove, from the applicable managed elements, all the non-shared cross-connections which comprise this SNC. 3. If the target OS succeeds in deactivating the SNC (i.e., if all the cross-connections comprising the SNC on applicable managed elements have been removed), the target OS initiates the deletion of the SNC in the target OS. SNC deletion is not expected to involve target OS communication with managed elements. SNC deletion involves the target OS deleting the SNC object that the target OS maintains. The target OS provides a success indication to the requesting OS. The target OS generates an object deletion notification when the SNC object is deleted and sends it to the notification service. The deactivation is successful even if some CCs representing fixed connectivity cannot be deleted. An SNC cannot be deactivated if it is composed solely of fixed cross-connects. 4. If the target OS fails to deactivate the SNC, a failure indication is sent to the requesting OS, and: 5. If at least one, but not all cross-connections comprising the SNC on applicable managed elements have been removed, the SNC's state changes to PARTIAL (or remains in PARTIAL if already in the PARTIAL state); 6. If none of the cross-connections comprising the SNC on applicable managed elements have been removed, the SNC's state remains ACTIVE. 7. If there has been an SNC state change, the target OS generates a state change notification when the SNC's state has been

	changed and sends it to the notification service. 8. For any cross-connection that has been successfully removed as a result of the deactivate SNC action, the target OS generates a state change notification for the associated connection termination points (CTPs), indicating that the TP connection state has transitioned to Not Connected. 9. If an Server SNC was involved in this SNC, the Server SNC does not support any other Client SNCs and all cross-connections comprising the SNC have been removed, the target OS behavior depends on whether the target OS has freedom to deactivate and delete the Server SNC. If the target OS does not have this level of freedom, only the Client SNC is deactivated and deleted. Case B: Pending SNC(s) supported This use case first uses the "The requesting OS Deactivates a Subnetwork Connection" use case. If the "The requesting OS Deactivates a Subnetwork Connection" use case is successfully completed, this use case then uses the "The requesting OS Deletes a Subnetwork Connection" use case.
Ends When	In case of success: The requesting OS receives an indication of success of the request. (Note that only a single success indication should be provided, not an indication for the successful deactivation of the SNC followed by another indication for the successful deletion of the SNC). In case of failure: 1. The requesting OS receives an indication of failure of the SNC deactivation action, or 2. The requesting OS receives an indication of failure of the SNC deletion action, or 3. The request times out.
Post-Conditions	In case of success: This use case uses the Post Conditions of the Delete SNC use cases. In case of failure: 1. The SNC state is either ACTIVE (all network resources associated with the SNC are still in use by the SNC) or PARTIAL (some network resources associated with the SNC are still in use by the SNC). • Whenever an exception is raised, it can be assumed that no network changes have been made to the SNCState. • The target OS maintains the SNC object.

	2. The network resources (CTPs) which had been used in the SNC are freed in the managed subnetwork and the SNC is in the PENDING state. However, (for whatever reason) the target OS maintains the SNC object. 3. The request times out. 4. If a change of SNC state has occurred, the target OS sends a state change notification to the notification service. For any change of TP connection state, the target OS has sent a state change notification to the notification service.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Internal error: The requested operation could not be performed. 3. Invalid input: Any input parameter is syntactical incorrect (e.g the sncName provided by the requesting OS does not refer to an SNC object, or when any field in tpsToModify is invalid). 4. Entity not found: The sncName or tpsToModify (provided by the requesting OS) reference an object that does not exist. 5. Unable to comply: The SNC is fixed and can not be deactivated. 6. Comm loss (Communication loss). **Note:** Whenever an exception is raised, it can be assumed that no network changes have been made to the SNC
Traceability	R_TMF518_RP_II_0030, R_TMF518_RP_II_0031 This use case is a generalization of Use Case 5.6.9 from TMF 513 v3.0

4.1.10 The target OS reroutes a Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0010
Use Case Name	target OS reroutes a Subnetwork Connection (SNC)
Summary	The target OS has the ability to reroute an SNC. There are a variety of reasons why this may happen. The requesting OS should know if it is possible for the target OS to reroute the SNC. The requesting OS will be notified when reroutes occur. The following are examples of when a reroute would occur in A target OS domain: a. Failure in the SNC or reversion (this is the main use) b. A lower cost route is available.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the

	TMF518_FMW BA document. 2. The requesting OS has registered with the notification service. 3. The SNC is in an ACTIVE state before the route change is initiated. The SNC can also be in a PARTIAL state if the route change failed or in PENDING state if a route is available at that time and the route change is re-initiated.
Begins When	The requesting OS creates an SNC
Description	1. The requesting OS can specify at SNC create time if it wishes the target OS to perform automatic rerouting of SNCs. 2. The requesting OS can specify one of the following reroute behaviors (don't care", "yes" or "no") • If the requesting OS specifies don't care", the target OS can choose the behavior. • If the requesting OS specifies "yes" and the target OS does not support this feature, the target OS will raise an unable to comply exception with the error reason indicating re-route not supported. • If the requesting OS specifies "no" and the target OS only supports the specified connection using reroute, the target OS will raise an unable to comply exception with the error reason indicating only re-route supported. • The actual value for the SNC ("yes" or "no") instance is contained in the reply. 3. A condition occurs that results in the reroute of the SNC: • A route change notification is sent indicating that a re-route has started; the notification contains the original route that is no longer available. • The SNC is re-routed successfully a route change notification is sent indicating that the SNC has been re-routed successfully; the notification contains the new route. • The SNC cannot be re-routed a route change notification is sent indicating that the SNC re-routing has failed; the notification contains the original route. 4. If the target OS attempts to reroute an SNC it will send appropriate 'notifications' to the notification service. The target OS may send SNC state changes. The SNC state may not be impacted if the target OS can create a new connection in the network before deleting old.
Ends When	The target OS has completed it re-routing and the SNC has either been successfully re-routed or has failed.
Post-Conditions	The SNC will have particular reroute behavior as defined in the target OS reply.

Exceptions	7. Not implemented: The target OS does not support this service. 8. Internal error: The requested operation could not be performed. 9. Unable to comply. 10. Comm loss (communication loss). 11. The target OS does not support the requested feature.
Traceability	R_TMF518_RP_II_0005 This use case is a generalization of Use Case 5.6.10 from TMF 513 v3.0

4.2 Equipment Provisioning

4.2.1 The requesting OS unprovisions equipment

Use Case Id	UC_TMF518_RP_0012
Use Case Name	The requesting OS unprovisions equipment
Summary	An operator permanently unprovisions an equipment from the managed element. The successful result of this operation is the potential deletion of the equipment object and all of its related objects, such as Termination Points.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to unprovision an equipment or the requesting OS detects equipment has been unprovisioned.
Description	1. The requesting OS sends the request to the target OS 2. The target OS validates the equipment name. If the name cannot be found an exception is thrown. 3. The target OS may check if a service is assigned to a port of the equipment (e.g. a supported PTP is in service) in which case an exception will be thrown. 4. The target OS deletes all of the equipment related objects, such as Termination Points. 5. The target OS should attempt to set the equipment state to out of service by maintenance prior to unprovisioning the equipment. 6. The target OS unprovisions the equipment.

	7. Appropriate notifications shall be sent.
Ends When	The target OS sends a response.
Post-Conditions	The Equipment is unprovisioned and all its related are objects deleted.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Internal error: The requested operation could not be performed. 3. Invalid input: The equipment name does not reference an equipment object. 4. Object in use: The equipment resources are in use. 5. Entity not found: The equipment name references an object which does not exist. 6. Unable to comply: The equipment can not be unprovisioned at the managedElement. 7. Comm loss (Communication loss).
Traceability	R_TMF518_RP_II_0036 This use case is a generalization of Use Case 5.9.1 from TMF 513 v3.0

4.2.2 The requesting OS provisions equipment

Use Case Id	UC_TMF518_RP_0013
Use Case Name	The requesting OS provisions equipment
Summary	An operator permanently provisions an equipment in an equipment holder in a ME. The result of this operation may be the creation of the equipment object and all of its related objects such as Termination Points.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to provision an equipment or the requesting OS detects equipment has been provisioned.
Description	1. The requesting OS sends request to target OS to provision an equipment. 2. The target OS validates the equipment CreateData. 3. If the equipmentHolder does not exist, an exception is thrown. 4. If the equipmentHolder already has an expected equipment, an

	exception is thrown. 5. The target OS provisions the equipment object and all of its related objects. 6. Appropriate notifications shall be sent.
Ends When	The target OS sends a response.
Post-Conditions	The new Equipment object is created and depending on the target OS, any object supported by the Equipment.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Internal error: The requested operation could not be performed. 3. Object in use: The equipment holder already has an expected equipment. 4. Invalid input: The equipment holder does not reference an equipmentHolder object. 5. Entity not found: The equipmentHolder references object that does not exist. 6. Unable to comply: The equipment can not be created at the managedElement. 7. Comm loss (Communication loss). 8. User label in use.
Traceability	R_TMF518_RP_II_0034 This use case is a generalization of Use Case 5.9.2 from TMF 513 v3.0

4.2.3 Craft inserts a plug-in card

Use Case Id	UC_TMF518_RP_0014
Use Case Name	Craft inserts a plug-in card
Summary	The Craft inserts a plug-in card.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. The requesting OS is attached to a notification service.
Begins When	The Target OS detects insertion of plug-in card.
Description	1. If new equipment and auto-provisioning is supported, the following notifications are sent to the notification service:

	• Equipment object creation notification • Equipment Holder State Change • If it is a equipment supporting TPs, PTP Object Creation Notifications may be sent to the notification service (Refer to the provision Equipment use case) 2. If auto-provisioning is not supported, the following notifications may be sent to the notification service: • Equipment object creation notification if the equipment has not been provisioned. • Equipment Holder State Change 3. If it is insertion of a provisioned plug-in, the following notifications may be sent to the notification service. • Equipment Holder State Change • Equipment State changes and AVCs.
Ends When	The target OS sends applicable notifications to the notification service.
Post-Conditions	The requesting OS is aware of the potential or modification of the PTPs, the state changes of the contained CTPs and the creation of any topological links.
Exceptions	No Interface exceptions as no requesting OS-initiated operations
Traceability	This use case is a generalization of Use Case 5.10.4 from TMF 513 v3.0

4.3 Flow Domain Control

4.3.1 Flow Domain Management Use Cases

4.3.1.1 The requesting OS creates a Flow Domain

Use Case Id	UC_TMF518_RP_0015
Use Case Name	The requesting OS creates a Flow Domain
Summary	This operation allows A requesting OS to create a Flow Domain (FD). The requesting OS can also associate Matrix Flow Domains (MFDs) and CPTP to this new FD. The association of the CPTPs is done on a best effort basis.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.

Begins When	The requesting OS sends a request to the target OS to create a Flow Domain.
Description	1) The requesting OS sends the request to the target OS to create an FD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a FD object with the name specified already exists, then an Object In Use exception is raised. c) If uniqueness of the user label is required, the target OS checks the user label for uniqueness; i.e., if a FD object with the same user label exists already, then a User Label In Use exception is raised. d) If one of the specified resources (i.e., MFDs and/or CPTPs) does not exist an Entity Not Found exception is raised. e) If any of the MFDs to be associated is already associated to another FD, an Object In Use exception is raised. f) If any of the MFDs to be associated could not be associated to the FD, no MFD is associated and an Unable To Comply exception is raised. g) If a CPTP is not already assigned to one of the provided MFDs, an Unable To Comply exception is raised. 3) If the request is valid the target OS creates the FD as requested. 4) The target OS associates the requested CPTPs to the new FD (i.e., the “Port TP role state” attribute of the CPTPs is set to “fdEdge”). All names of the CPTPs that could not be associated shall be returned in the reply. 5) The target OS replies with a success indication 6) The target OS shall send an FD object creation notification to the notification service.
Ends When	In case of success: 1) The requesting OS receives an indication of the success of the operation. 2) The target OS returns the distinguishing information of the created FD. 3) In case not all requested CPTPs could be associated, the target OS returns the list of these CPTPs. In case of failure: The requesting OS receives an exception as an indication of

	the failure of the request.
Post-Conditions	In case of success: 1) The FD has been created. 2) The MFDs have been associated to the new FD. 3) The CPTPs have been associated to the FD (i.e., the Port TP role state attribute of the CPTPs has been set to "fdEdge") or have been returned in the reply. In case of failure: Nothing has changed in the System, i.e., the FD has not been created and the appropriate exception is reported to the requesting OS.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0046, R_TMF518_RP_II_0047 This use case is a generalization of Use Case 9.3.1 from TMF 513 v3.1

4.3.1.2 The requesting OS deletes a Flow Domain

Use Case Id	UC_TMF518_RP_0016
Use Case Name	The requesting OS deletes a Flow Domain
Summary	This operation allows A requesting OS to delete an existing Flow Domain (FD). The target OS is required to verify that no Flow Domain Fragments (FDFrs) exist within the FD. The operation dissociates the FD Edge CPTPs, dissociates the Matrix Flow Domains (MFDs) and deletes the FD.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to delete a Flow Domain.
Description	1) The requesting OS sends the request to the target OS to delete an FD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the FD object is not known to the target OS, an Entity

	Not Found exception is raised. c) If any FDFr is contained in the FD, an Object In Use exception is raised. d) If any of the MFD can not be dissociated from the specified FD, no MFD is dissociated and an Unable To Comply exception is raised. e) If any FD Edge CPTPs can not be dissociated from the FD (i.e., the Port TP role state attribute of the CPTPs was set to “assigned”) no FD Edge CPTP are dissociated and an Unable To Comply exception is raised. 3) If the request is valid the target OS deletes the Flow Domain. 4) The target OS replies with a success indication. 5) The target OS shall sends an FD object deletion notification to the notification service.
Ends When	In case of success: The requesting OS receives an indication of success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: 1) The FD Edge CPTPs associated to the FD have been de-associated (i.e., the Port TP role state attribute of the CPTPs has been set to "assigned"). 2) The MFDs associated to the FD have been de-associated. 3) The FD has been deleted. In case of failure: Nothing has changed in the System, i.e., the FD has not been deleted and the appropriate exception is reported to the requesting OS.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0050 This use case is a generalization of Use Case 9.3.2 from TMF 513 v3.1

4.3.1.3 The requesting OS modifies a Flow Domain

Use Case Id	UC_TMF518_RP_0017

Use Case Name	The requesting OS modifies a Flow Domain
Summary	This operation allows A requesting OS to modify a Flow Domain (FD) that already exists. The requesting OS can modify the user label, the owner, the network access domain, the connectionless layered parameters or the additional information of an existing FD.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to modify the attributes of a Flow Domain.
Description	1) requesting OS sends the request to the target OS to modify a FD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FD is not known to the target OS, an Entity Not Found exception is raised. c) If uniqueness of the user label is required, the target OS checks the user label for uniqueness; i.e., if a FD object with the same user label exists already, then a User Label In Use exception is raised. d) If the target OS can not satisfy any attribute that needs to be modified, an an Unable To Comply exception is raised. 3) If the request is valid: a) If the FD already has all the required information (i.e., no changes are required), the target OS shall not send any notifications to the notifcation service. b) Otherwise the target OS modifies the attributes of the FD as requested. 4) The target OS replies with a success indication 5) If the target OS made a change then appropriate notifications shall be sent to the notification service.
Ends When	In case of success: 1) The requesting OS receives an indication of the success of the operation. 2) The target OS returns the distinguishing information of the modified FD.

	In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The FD has been modified as requested. In case of failure: Nothing has changed in the System, i.e., the FD has not been modified.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0048, R_TMF518_RP_II_0049 This use case is a generalization of Use Case 9.3.3 from TMF 513 v3.1

4.3.1.4 The requesting OS associates Matrix Flow Domain(s) to a Flow Domain

Use Case Id	UC_TMF518_RP_0018
Use Case Name	The requesting OS associates Matrix Flow Domain(s) to a Flow Domain
Summary	This operation allows A requesting OS to associate one or more Matrix Flow Domain(s) (MFD(s)) to an existing Flow Domain (FD). The target OS is required to validate the data provided by the requesting OS, and associates the requested MFD(s) to the specified FD. If the target OS cannot associate the MFD as specified, an appropriate exception is raised. Note that best effort is not supported. The target OS verifies the server layer connectivity between the associated MFD(s), the “FD Connectivity State” attribute of the FD is modified accordingly; fully connected, not fully connected, unknown.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to associate one or more MFDs to a FD.
Description	1) The requesting OS sends the request to the target OS to associate additional MFD(s) to an existing FD. 2) The target OS validates the request: a) If the syntax is in errror, an Invalid Input exception is raised. b) If the specified FD is not known to the target OS, an

	Entity Not Found exception is raised. c) If one or more of the specified MFDs is not known to the target OS, an Entity Not Found exception is raised. d) If one or more of the specified MFDs are already associated to another FD, an Object In Use exception is raised. e) If any of the MFDs could not be associated, no MFD is associated and an Unable To Comply exception is raised. 3) If the request is valid the target OS associates the specified MFD(s) to the FD. 4) The target OS verifies the server layer connectivity between the associated MFD(s), the "FD Connectivity State" attribute is modified accordingly; fully connected, not fully connected, unknown. 5) The target OS replies with a success indication. 6) The target OS sends an attribute value change notification to the notification service; i.e., via a notification of the FD's "Matrix Flow Domains" attribute. Note that the notification includes the complete list of MFD names that are associated to the FD. 7) If the "FD Connectivity State" attribute is modified, the target OS sends a state change notification to the notification service.
Ends When	In case of success: The target OS returns the distinguishing information of the modified FD. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The FD has been modified as requested. In case of failure: Nothing has changed in the System, i.e., the FD has not been modified.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0053 This use case is a generalization of Use Case 9.3.4 from TMF 513 v3.1

4.3.1.5 The requesting OS dissociates Matrix Flow Domain(s) to a Flow Domain

Use Case Id	UC_TMF518_RP_0019
Use Case Name	The requesting OS dissociates Matrix Flow Domain(s) to a Flow Domain
Summary	This operation allows A requesting OS to dissociate one or more Matrix Flow Domain(s) (MFD(s)) from an existing Flow Domain (FD). The target OS is required to validate the data provided by the requesting OS, and dissociates the requested MFD(s) from the specified FD. If the target OS cannot dissociate the MFD(s) as specified, an appropriate exception is raised. Note that best effort is not supported. The operation also dissociates the FD Edge CPTPs which are associated to the MFDs to be dissociated. Following the dissociation, the target OS verifies the server layer connectivity between the associated MFD(s), the “FD Connectivity State” attribute of the FD is modified accordingly; “fully connected”, “not fully connected”, or “unknown”.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to dissociate one or more MFD(s) to a FD.
Description	1) The requesting OS sends the request to the target OS to dissociate MFD(s) from an existing FD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FD is not known to the target OS, an Entity Not Found exception is raised. c) If one or more of the specified MFDs was not previously associated to the FD, an Unable To Comply exception is raised. d) If one or more of the specified MFDs has traffic (i.e., FDFr uses this MFD), an Object In Use exception is raised. e) If any FD Edge CPTPs can not be dissociated from the MFDs that are being dissociated (i.e., the Port TP role state attribute of the CPTPs was set to “assigned”) no FD Edge CPTP are dissociated and an Unable To Comply exception is raised. f) If any of the MFDs could not be dissociated, no MFD is dissociated and an Unable To Comply exception is

	raised. 3) If the request is valid the target OS dissociates the specified MFD(s) from the FD. 4) The target OS replies with a success indication. 5) The target OS sends an attribute value change notification to the notification service; i.e., via a notification of the FD's "Matrix Flow Domains" attribute. Note that the notification includes the complete list of MFD names that are associated to the FD. 6) The target OS verifies the server layer connectivity between the associated MFD(s), the “FD Connectivity State” attribute is modified accordingly; fully connected, not fully connected, unknown. 7) If the "FD Connectivity State" attribute is modified, the target OS sends a state change notification to the notification service.
Ends When	In case of success: The requesting OS receives an indication of success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: 1) The requested MFD(s) have been dissociated from the FD. 2) The corresponding FD Edge CPTPs have been dissociated from the FD. In case of failure: Nothing has changed in the System, i.e., no change in the MFD association to the FD.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0054 This use case is a generalization of Use Case 9.3.5 from TMF 513 v3.1

4.3.1.6 The requesting OS associates CPTP(s) to a Flow Domain

Use Case Id	UC_TMF518_RP_0020
Use Case Name	The requesting OS associates CPTP(s) to a Flow Domain
Summary	This operation allows A requesting OS to associate one or more

	CPTPs to an existing Flow Domain (FD). The target OS is required to validate the data provided by the requesting OS, and associate the requested CPTP(s) to the specified FD. The operation is best effort, i.e., the list of CPTPs that could not be associated will be returned in the reply.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document
Begins When	The requesting OS sends a request to the target OS to associate CPTP(s) to a FD.
Description	1) The requesting OS sends the request to the target OS to associate additional CPTP(s) to an existing FD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FD is not known to the target OS, an Entity Not Found exception is raised. c) If one or more of the specified CPTPs is not known to the target OS, an Entity Not Found exception is raised. d) If one or more of the specified CPTPs is not in the “assigned” CPTP “Port TP role state”, an Unable To Comply exception is raised. e) If one or more of the specified CPTPs is assigned to an MFD that is not associated to the FD, an Unable To Comply exception is raised. 3) If the request is valid the target OS associates the specified CPTP(s) to the FD (i.e., the “Port TP role state” attribute of the CPTPs is set to “FD Edge”). All CPTPs that could not be associated have to be returned in the reply. 4) The target OS replies with a success indication 5) The target OS sends an attribute value change notification to the notification service; i.e., a notification of the FD's “FD Edge CPTPs” attribute. Note that the notification includes the complete list of FD Edge CPTP names that are associated to the FD.
Ends When	In case of success: The requesting OS receives an indication of success of the request and a list of all CPTPs that could not be associated. In case of failure: The requesting OS receives an exception as an indication of

	the failure of the request.
Post-Conditions	In case of success: The requested CPTPs (except the ones returned in the list of not associated CPTPs) have been associated as FD Edge CPTPs to the FD. In case of failure: Nothing has changed in the System, i.e., no change in the CPTP association to the FD.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0051 This use case is a generalization of Use Case 9.3.6 from TMF 513 v3.1

4.3.1.7 The requesting OS dissociates FD Edge CPTPs from a Flow Domain

Use Case Id	UC_TMF518_RP_0021
Use Case Name	The requesting OS dissociates FD Edge CPTP(s) from a Flow Domain
Summary	This operation allows A requesting OS to dissociate one or more FD Edge CPTP(s) from an existing Flow Domain (FD). The target OS is required to validate the data provided by the requesting OS, and dissociates the requested FD Edge CPTP(s) from the specified FD. The operation will be rejected if one or more of the specified FD Edge CPTPs carries traffic. If the target OS cannot dissociate the FD Edge CPTP(s) as specified, an appropriate exception is raised. Note that best effort is not supported.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to dissociate FD Edge CPTP(s) from an FD.
Description	1) The requesting OS sends the request to the target OS to disassociate FD Edge CPTPs from an existing FD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FD is not known to the target OS, an Entity Not Found exception is raised.

	c) If one or more of the specified FD Edge CPTPs is not known to the target OS, an Entity Not Found exception is raised. d) If one or more of the specified FD Edge CPTPs is not in the “FD Edge” CPTP “Port TP role state”, an Unable To Comply exception is raised. e) If one or more of the specified FD Edge CPTPs is assigned to an MFD that is not associated to the FD, an Unable To Comply exception is raised. f) If one or more of the specified FD Edge CPTPs carries traffic (i.e., an FDFr is provisioned on this CPTP), an Object In Use exception is raised. g) If any of the CPTPs could not be dissociated, no CPTP is dissociated and an Unable To Comply exception is raised. 3) If the request is valid the target OS dissociates the specified FD Edge CPTP (s) from the FD (i.e., the “Port TP role state” attribute of the CPTPs is set to “assigned”). 4) The target OS replies with a success indication. 5) The target OS send an attribute value change notification to the notification service; i.e., via a notification of the FD's "FD Edge CPTPs" attribute. Note that the notification includes the complete list of FD Edge CPTP names that are associated to the FD.
Ends When	In case of success: The requesting OS receives an indication of success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The requested FD Edge CPTPs have been disassociated from the FD. In case of failure: Nothing has changed in the System, i.e., no change in the FD Edge CPTP association to the FD.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0052 This use case is a generalization of Use Case 9.3.7 from TMF 513 v3.1

4.3.2 Matrix Flow Domain Management Use Cases

4.3.2.1 The requesting OS creates a Matrix Flow Domain (MFD)

Use Case Id	UC_TMF518_RP_0022
Use Case Name	The requesting OS creates a Matrix Flow Domain (MFD)
Summary	This operation allows A requesting OS to create a Matrix Flow Domain (MFD). The target OS creates the requested MFD consistent with its implementation and the requesting OS specifications. If the target OS cannot create the MFD as specified, an appropriate exception is raised. The requesting OS specifies the MFD through a list of FD Edge CPTPs and/or internal TPs that the MFD must contain, and a set of attributes that the MFD must satisfy. After the target OS validates the input data, and if it can satisfy the input constraints, it proceeds with implementing the MFD. If the target OS fails to validate the input, it raises an appropriate exception. The target OS implements the Matrix Flow Domain as specified by the requesting OS. It also assigns all the specified TPs to the MFD.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to create a MFD.
Description	1) The requesting OS sends the request to the target OS to create an MFD with the provided parameters. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If uniqueness of the user label is required, the target OS checks the user label for uniqueness; i.e., if an MFD object with the same user label exists already then a User Label In Use exception is raised. c) If one of the specified TPs is not known to the target OS, an Entity Not Found exception is raised. d) If at least one of the MFD parameters could not be set, an Unable To Comply exception is raised. e) If any of the specified TPs is already in use by another MFD, an Object In Use exception is raised. 3) If the request is valid: a) the target OS creates the MFD. b) The target OS assigns the requested CPTPs to the

	MFD. 4) The target OS replies with a success indication 5) The target OS sends object creation notifications to the notification service.
Ends When	In case of success: The requesting OS receives an indication of the success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: 1) The MFD has been created. 2) The requested TPs have been associated with the MFD. 3) The provided parameters have been set for the MFD. 4) The requested FD Edge CPTPs have been associated with the FD. In case of failure: Nothing has changed in the System, i.e., the MFD has not been created.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0037, R_TMF518_RP_II_0038 This use case is a generalization of Use Case 9.4.1 from TMF 513 v3.1

4.3.2.2 The requesting OS deletes a Matrix Flow Domain (MFD)

Use Case Id	UC_TMF518_RP_0023
Use Case Name	The requesting OS deletes a Matrix Flow Domain (MFD)
Summary	This operation allows the requesting OS to delete a Matrix Flow Domain (MFD) that already exists. The target OS is required to validate the MFD, verify that the MFD is not associated with a Flow Domain.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to delete an

	MFD.
Description	1) The requesting OS sends the request to the target OS to delete an MFD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified MFD object is not known to the target OS, an Entity Not Found exception is raised. c) The MFD to be deleted must not be associated with a Flow Domain. If it is still associated, an Object In Use exception is raised. d) The MFD to be deleted must not be “fixed”. If it is fixed, an Unable To Comply exception is raised. 3) If the request is valid: a) The target OS deletes the MFD. b) The target OS releases all assigned TPs from the MFD. 4) The target OS replies with a success indication. 5) The target OS sends an object deletion notification to the notification service.
Ends When	In case of success: The requesting OS receives an indication of the success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The assigned TPs have become unassigned and the MFD has been deleted. In case of failure: Nothing has changed in the System, i.e., the MFD has not been deleted and no TPs have been modified.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0041 This use case is a generalization of Use Case 9.4.2 from TMF 513 v3.1

4.3.2.3 The requesting OS modifies a Matrix Flow Domain (MFD)

Use Case Id	UC_TMF518_RP_0024
Use Case Name	The requesting OS modifies a Matrix Flow Domain (MFD)
Summary	This operation allows the requesting OS to modify a Matrix Flow Domain (MFD) that already exists. The requesting OS can modify the user label, the owner, the attributes (e.g., Spanning Tree Protocol (STP) parameters) or the additional information of an existing Matrix Flow Domain.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to modify an MFD.
Description	1) The requesting OS sends a request to the target OS to modify an MFD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified MFD is not known to the target OS, an Entity Not Found exception is raised. c) If uniqueness of the user label is required, the target OS checks the user label for uniqueness; i.e., if a MFD object with the same user label exists already, then a User Label In Use exception is raised. d) If the target OS can not satisfy any attribute that needs to be modified, an an Unable To Comply exception is raised. e) If the MFD already has the required information, the target OS replies with a success indication but no notification is generated. 3) If the request is valid: a) If the MFD already has all the required information (i.e., no changes are required), the target OS shall not send any notifications to the notifcation service. b) Otherwise the target OS modifies the attributes of the MFD as requested. 4) The target OS replies with a success indication. 5) If the target OS made a change then appropriate notifications shall be sent to the notification service.
Ends When	In case of success:

	1) The requesting OS receives an indication of the success of the operation. 2) The target OS returns the distinguishing information of the modified MFD together with the list of attributes which could not be modified. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The MFD has been modified as requested. In case of partial success: The MFD has been modified but not all requested modifications have been completed. In case of failure: Nothing has changed in the System, i.e., the MFD has not been modified.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0039, R_TMF518_RP_II_0040 This use case is a generalization of Use Case 9.4.3 from TMF 513 v3.1

4.3.2.4 The requesting OS assigns CPTP(s) to a Matrix Flow Domain

Use Case Id	UC_TMF518_RP_0025
Use Case Name	The requesting OS assigns CPTP(s) to a Matrix Flow Domain
Summary	This operation allows A requesting OS to assign one or more CPTPs to an existing Matrix Flow Domain (MFD). The target OS is required to validate the data provided by the requesting OS, and assign the requested CPTP(s) to the specified MFD. Note that best effort is not supported.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document
Begins When	The requesting OS sends a request to the target OS to assign a list of CPTP(s) to an MFD
Description	1) The requesting OS sends a request to the target OS to assign

	a list of CPTP(s) to an existing MFD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified MFD is not known to the target OS, an Entity Not Found exception is raised. c) If the specified MFD is fixed, an Unable To Comply exception is raised. d) If one or more of the specified CPTPs is not known to the target OS, an Entity Not Found exception is raised. e) If one or more of the specified CPTPs is not a potential CPTP for this MFD (i.e., is not in the "unassigned" CPTP "Port TP role state" or is not on the same equipment or same rack with backplane connectivity), an Unable To Comply exception is raised. 3) If the request is valid the target OS assigns the specified CPTP(s) to the MFD (i.e., the “Port TP role state” attribute of the CPTPs is set to “assigned”). 4) The target OS replies with a success indication. 5) The target OS send an attribute value change notification to the notification service; i.e., via a notification of the MFD's “Assigned CPTPs” attribute. Note that the notification includes the complete list of CPTP names that are assigned to the MFD.
Ends When	In case of success: The requesting OS receives an indication of success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The requested CPTPs have been assigned to the MFD. In case of failure: Nothing has changed in the System, i.e., no change in the CPTP assignment to the MFD.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0042 This use case is a generalization of Use Case 9.4.4 from TMF 513 v3.1

4.3.2.5 The requesting OS un-assigns CPTP(s) from a Matrix Flow Domain

Use Case Id	UC_TMF518_RP_0026
Use Case Name	The requesting OS un-assigns CPTP(s) from a Matrix Flow Domain
Summary	This operation allows A requesting OS to un-assign one or more CPTP(s) to an existing Matrix Flow Domain (MFD). The target OS is required to validate the data provided by the requesting OS, and to un-assign the requested CPTP(s) from the specified MFD. Note that best effort is not supported.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to un-assign a list of CPTP(s) from an MFD
Description	1) The requesting OS sends a request to the target OS to un-assign a list of CPTP(s) from an existing MFD. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified MFD is not known to the target OS an Entity Not Found exception is raised. c) If the specified MFD is fixed, an Unable To Comply exception is raised. d) If one or more of the specified CPTPs is not known to the target OS, an Entity Not Found exception is raised. e) If one or more of the specified CPTPs is in the “unassigned” “Port TP role state”, a Not In Valid State exception is raised. f) If one or more of the specified CPTPs is used by a Flow Domain Fragment, an Object In Use exception is raised. g) If one or more of the specified CPTPs is not assigned to the specified MFD, an Unable To Comply exception is raised. 3) If the request is valid the target OS un-assigns the specified CPTP(s) from the MFD (i.e., the "Port TP role state" attribute of the CPTPs is set to “unassigned”). If any of the CPTPs could not be un-assigned, no CPTP is un-assigned and an Unable To Comply exception is raised. 4) The target OS replies with a success indication. 5) The target OS sends the appropriate notifications to the

	notification service.
Ends When	In case of success: The requesting OS receives an indication of success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The requested CPTPs have been assigned to the MFD. In case of failure: Nothing has changed in the System, i.e., no change in the CPTP assignment to the MFD.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0043 This use case is a generalization of Use Case 9.4.5 from TMF 513 v3.1

4.3.3 Traffic Conditioning Profile Management Use Cases

4.3.3.1 The requesting OS creates a Traffic Conditioning Profile

Use Case Id	UC_TMF518_RP_0027
Use Case Name	The requesting OS creates a Traffic Conditioning Profile
Summary	This operation allows A requesting OS to create aTraffic Conditioning (TC) Profile. The target OS validates the data provided by requesting OS, and creates the requested TC Profile in accordance with the requesting OS parameter list. If the target OS cannot create the TC Profile as specified, an appropriate exception is raised.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to target OS to create a TC Profile.
Description	1) The requesting OS sends a request to target OS to create a TC Profile. The requesting OS provides the TC Profile name and a list of bandwidth parameters. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is

	raised. b) If uniqueness of the user label is required, the target OS checks the user label for uniqueness; i.e., if a TC Profile object with the same user label exists already, then a User Label In Use exception is raised. c) If the maximum number of TC Profiles in the target OS has already reached and the TC Profile cannot be created, a Capacity Exceeded exception is raised. 3) If the request is valid the target OS creates the TC Profile. 4) The target OS replies with success indication. 5) The target OS sends a TC Profile object creation notification to the notification service.
Ends When	In case of success: The requesting OS receives an indication of the success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The TC Profile has been created by the target OS. In case of failure: Nothing has changed in the System, i.e., the TC Profile has not been created.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0055, R_TMF518_RP_II_0056 This use case is a generalization of Use Case 9.5.1 from TMF 513 v3.1

4.3.3.2 The requesting OS deletes a Traffic Conditioning Profile

Use Case Id	UC_TMF518_RP_0028
Use Case Name	The requesting OS deletes a Traffic Conditioning Profile
Summary	This operation allows A requesting OS to delete a Traffic Conditioning (TC) Profile that already exists. The target OS is required to validate the request, verify that the TC Profile is not associated to a Flow Point (FP) or FD Edge CPTP (FD Edge CPTP), and deletes it.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use

	Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to target OS to delete a TC Profile.
Description	1) The requesting OS sends a request to target OS to delete a TC Profile. The requesting OS provides the TC Profile name. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specifed TC Profile object is not known to the target OS, an Entity Not Found exception is raised. c) If any FP or CPTP within the target OS has an association to the TC Profile, an Object In Use exception is raised. 3) If the request is valid the target OS deletes the TC Profile. 4) The target OS replies with success indication. 5) The target OS sends a TC Profile object deletion notification to the notification service.
Ends When	In case of success: The requesting OS receives an indication of the success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The TC Profile has been deleted by the target OS. In case of failure: Nothing has changed in the System, i.e., the TC Profile has not been deleted.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0058 This use case is a generalization of Use Case 9.5.2 from TMF 513 v3.1

4.3.3.3 The requesting OS modifies a Traffic Conditioning Profile

Use Case Id	UC_TMF518_RP_0029
Use Case Name	The requesting OS modifies a Traffic Conditioning Profile

Summary	This operation allows A requesting OS to modify aTraffic Conditioning (TC) Profile. target OS validates the data provided by requesting OS, and modifies the requested TC Profile in accordance with the requesting OS parameter list. If the target OS cannot modify the TC Profile as specified, an appropriate exception is raised.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to target OS to modify a TC Profile
Description	1) The requesting OS sends a request to target OS to modify a TC Profile. The requesting OS provides the TC Profile name and a list of bandwidth parameters. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specirfied TC Profile object does not exist, an Entity Not Found exception is raised. 3) If the request is valid the target OS modifies the TC Profile with the new parameter list. Note: This will automatically modify the traffic conditioning in all associated TPs. 4) The target OS replies with success indication. 5) The target OS sends a TC Profile attribute value change notification to the notification service.
Ends When	In case of success: The requesting OS receives an indication of the success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The TC Profile has been modified by the target OS. In case of failure: Nothing has changed in the System, i.e., the TC Profile has not been modified.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0056, R_TMF518_RP_II_0057

	This use case is a generalization of Use Case 9.5.3 from TMF 513 v3.1

4.3.3.4 The requesting OS configures Traffic Mapping Table

Use Case Id	UC_TMF518_RP_0030
Use Case Name	The requesting OS configures Traffic Mapping Table
Summary	This operation allows A requesting OS to configure the Traffic Mapping Table in an CPTP or FP. The requesting OS provides a complete (except default) new set of mappings which will overwrite all (except default) existing mappings.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to target OS to configure the Traffic Mapping Table of an FD Edge CPTP or FP.
Description	1) The requesting OS sends a request to target OS to configure the Traffic Mapping Table of an FD Edge CPTP or FP. 2) The requesting OS provides a complete (except default) new set of mappings. 3) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If at least one referenced TC Profile objects is not known to the target OS, an Entity Not Found exception is raised. 4) If the request is valid: a) If the provided set of mappings contains no mapping at all (i.e., is empty), all mappings, except the default one, are removed; i.e., the traffic units flowing through the TP are only conditioned by the default configuration. Note: The traffic units may still be conditioned specifically by another TP on this port. If the mappings cannot be removed, an Unable To Comply exception is raised. b) If the provided set of mappings contains mappings, all existing mappings (except default) are overwritten by the new set of mappings; i.e., the traffic units are conditioned by the new configuration. If the mappings cannot be overwritten, an Unable To Comply exception is raised.

	5) The target OS replies with a success indication. 6) The target OS sends an attribute value change notification to the notification service.
Ends When	In case of success: The target OS has set the new mappings in the Traffic Mapping Table. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The TP conditions the traffic units according to the modified Traffic Mapping Table. In case of failure: Nothing has changed in the System, i.e., the Traffic Mapping Tables have not been changed.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0059 This use case is a generalization of Use Case 9.5.4 from TMF 513 v3.1

4.3.4 Flow Domain Fragment Management Use Cases

4.3.4.1 The requesting OS creates and activates a Flow Domain Fragment

Use Case Id	UC_TMF518_RP_0031
Use Case Name	The requesting OS creates and activates a Flow Domain Fragment
Summary	This operation provides a way to create and activate a point-to-point or multipoint-to-multipoint Flow Domain Fragment (FDFr) in one request. The target OS is required to create an FDFr at the requested layer rate. If the target OS cannot meet any of the requested parameters, appropriate exceptions are raised. If the target OS works in the “connectivity-aware” mode, the requesting OS can request one of two creation results when not all Flow Points have potential connectivity to one another: • option 1) reject the creation request, or • option 2) add all FPs regardless of potential connectivity. If the FDFr is created, an accompanying FP is created for every edge

	CPTP specified. In case neither the target OS nor the network provides auto-routing, the requesting OS also has to provide the internal CPTPs that have to be used by the target OS when creating the FDFr. An accompanying FP is created for every internal CPTP specified. If transmission parameters are specified for the involved CPTPs and FPs, the target OS will apply them either before or after the creation of the Matrix Flow Domain Fragments (MFDFrs), as appropriate. The alarm reporting on the CPTPs and the containing FPs may be turned on by the target OS, unless otherwise specified via the alarm reporting transmission parameter. This operation is best effort except for option 1) when the target OS works in the “connectivity-aware” mode.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to create and activate a Flow Domain Fragment to the target OS.
Description	1) The requesting OS sends a request to create and activate a Flow Domain Fragment to the target OS. 2) The target OS validates the request: a) TIf the syntax is in error, an Invalid Input exception is raised. b) If one of the referenced CPTP objects is not known to the target OS, an Entity Not Found exception is raised. c) If the specified CPTPs are not associated with the referenced FD, a TP Invalid Endpoint exception is raised. d) If an FDFr with the same properties as specified in the requesting OS request already exists, the target OS may reuse that FDFr. e) If any of the specified edge CPTPs do not have the “FD Edge” role, a Not In Valid State exception is raised. f) If any of the specified internal CPTPs do not have the “FD Internal” role, a Not In Valid State exception is raised. g) If the requesting OS has provided an FDFr name and the target OS does not support requesting OS supplied names, an Unable To Comply exception is raised. h) If uniqueness of the user label is required, the target OS checks the user label for uniqueness; i.e., if a FDFr object with the same user label exists already, then a User Label In Use exception is raised.

	i) If any of the mandatory input parameters cannot be satisfied, an Unable To Comply exception is raised. j) If the target OS is "connectivity-aware" and option 1 (see summary) was requested, and if not all of the FPs (to be created) have connectivity with one another, an Unable To Comply exception is raised. k) In traffic mapping tables specified for created FPs, if any map column headers are not recognized by the target OS, an Unable To Comply exception is raised. l) In traffic mapping tables specified for created FPs, if there are mismatches between the TrafficMapping_Table_Count and any column length, an Unable To Comply exception is raised. m) In traffic mapping tables specified for created FPs, if table contents would result in frames being mapped to more than one FDFr, an Unable To Comply exception is raised. n) If the FDFr being created would have less than two edge FPs, an Unable To Comply exception is raised. o) If the FDFr cannot use all of the specified internal CPTPs and/or all of the MFDFrs in the specified route, an Unable To Comply exception is raised. 3) If the requesting OS provides transmission parameters for the involved CPTPs and FPs, it is up to the target OS to provision the parameters on these TPs before or after the activation of the Matrix Flow Domain Fragments, as appropriate. A CPTP or FP which cannot be provisioned, or for whom mandatory[1] transmission parameters cannot be set, must not be added to the new FDFr and has to be returned in the "failed TP list" list (for every FP which is not added to the FDFr, the corresponding server CPTP has to be added to the list). FPs for which only best-effort transmission parameters could not be set have to be added to the new FDFr and must be returned in the "parameter problems TP list" list. CPTPs for which only best-effort transmissions parameters could not be set have to be returned in the "parameter problems TP list". If an internal CPTP cannot be provisioned, it is returned either in the "failed TP list" list, when the provisioning is mandatory, or to the "parameter problems TP list" list, when the provisioning is best-effort. The alarm reporting on the CPTPs and the contained FPs may be turned on by the target OS as part of this request, unless otherwise specified via the transmission parameter "AlarmReporting". 4) If the specified value of the FDFr's IVID is not null, all FPs

[1]Parameters are classified into mandatory and best-effort by bilateral agreement or in the Implementation Statement document.

	specified with IVID different from the FDFr's IVID must not be associated to the new FDFr and their server CPTPs have to be returned in the “failed TP list” list. 5) The target OS initiates the activation of the FDFr (which involves establishing the Matrix Flow Domain Fragment(s) at the ME(s)). 6) If all of the Matrix Flow Domain Fragments comprising the FDFr have been established, then the target OS sets the FDFr state of the FDFr to active. The target OS updates the SNC state of the affected CTPs to either sink, source, or bidirectionally connected. 7) If there are error conditions, (e.g. failure to provision TP transmission) existing after establishing all the Matrix Flow Domain Fragments, then the target OS will reply with a failure indication and error reason. In this case the FDFr state of the FDFr will be active. 8) If one or more (possibly all) of the Matrix Flow Domain Fragment comprising the FDFr have not been established then the target OS sets the state of the FDFr to partial. The target OS updates the connection state of those CTPs that were successfully established to either sink, source, or bidirectionally connected. The target OS replies with a failure indication and error reason. 9) The target OS replies with a success indication. 10) The target OS sends FP and FDFr object creation notifications to the notification service.
Ends When	In case of success: 1) The target OS returns the distinguishing information of the created FDFr in the success indication of the action. 2) The target OS sends object creation notifications for the created FDFr and FPs. The OC notification of “CTP” like objects like FPs is new to the interface. This because: • congestion prevention at the interface • FPs (like CTPs) have a naming convention, hence it is not strictly necessary even to read them. 3) The target OS sends the necessary AVC and SC notifications for the transmission parameters changed in the CPTPs. 4) In case of best-effort, and in case not all requested CPTPs could be associated, the target OS returns the CPTPs that could not be associated in the “failed TP list”. 5) In case not all provided best-effort2 transmissions parameters

	could be set on some CPTPs or FPs as requested, the target OS returns these TPs in the list of “parameter problems TP list”. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: 1) The FDFr and its FPs have been created. 2) The FDFr has been activated, or it has been partially activated, i.e., some but not all the necessary provisioning operations have been performed. 3) All requested CPTPs which are not contained in the "failed TP list" list have been associated to the new FDFr. 4) The transmission parameters have been set, as requested, to all CPTPs and FPs which are not contained in the “parameter problems TP list”. In case of failure: Nothing has changed in the System, i.e., the FDFr has not been created.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0060, R_TMF518_RP_II_0061 This use case is a generalization of Use Case 9.6.1 from TMF 513 v3.1

4.3.4.2 The requesting OS deactivates and deletes a Flow Domain Fragment

Use Case Id	UC_TMF518_RP_0032
Use Case Name	The requesting OS deactivates and deletes a Flow Domain Fragment
Summary	The requesting OS requests that a Flow Domain Fragment be deactivated and deleted. As a result of successfully completing this use case: • The FDFr will be de-provisioned from the target OS' managed Flow Domain, and the FDFr will be removed from the target OS. • The removal of any network resources allocated for the FDFr.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.

Begins When	The requesting OS sends a request to the target OS to deactivate and delete an FDFr
Description	1) The requesting OS sends the request to deactivate and delete a Flow Domain Fragment (FDFr) to the target OS. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FDFr object is not known to the target OS, an Entity Not Found exception is raised. 3) If the request is valid: a) The target OS initiates the deactivation of the FDFr. FDFr deactivation involves target OS communication with managed elements. The target OS attempts to remove, from the applicable managed elements, all the Matrix FDFrs which comprise this FDFr. b) If any Matrix FDFr cannot be removed, its FPs must be returned in the “failed TP list” list. c) If the target OS succeeds in deleting the FDFr (i.e., if all the Matrix FDFrs comprising the FDFr on applicable managed elements have been removed), the target OS initiates the deletion of the FDFr in the target OS. FDFr deletion is not expected to involve target OS communication with managed elements. FDFr deletion involves the target OS deleting the FDFr object, and all associated FPs. d) If some MFDFrs were removed and some were not, the FDFr is not deleted from the target OS, and it is marked “Failed”. 4) The target OS provides a success indication to the requesting OS. 5) The target OS sends FP and FDFr object deletion notifications to the notification service.
Ends When	In case of success: 1) The requesting OS receives an indication of success of the request. 2) The FPs of any MFDFrs which could not be removed are returned in the “failed TP list” list. 3) Object Deletion notifications have been sent for all FPs which were removed. 4) If the FDFr has been removed, an Object Deletion notification has been sent. In case of failure: The requesting OS receives an exception as an indication of

	the failure of the request.
Post-Conditions	In case of success: 1) For all MFDFrs which are removed, the assocated FPs have been deleted. 2) If all MFDFrs were removed, the FDFr has been deleted. 3) If the FDFr was not deleted, it is marked as “Failed”. In case of failure: Nothing has changed in the System, i.e., the FDFr has not been created.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0064 This use case is a generalization of Use Case 9.6.2 from TMF 513 v3.1

4.3.4.3 The requesting OS modifies a Flow Domain Fragment

Use Case Id	UC_TMF518_RP_0033
Use Case Name	The requesting OS modifies a Flow Domain Fragment
Summary	This operation allows A requesting OS to add one or more Flow Points (FPs) to an existing Flow Domain Fragment (FDFr), and to remove one or more Flow Points (FPs) from an existing Flow Domain Fragment. Flow Points are added by designating the FD Edge CPTPs and FD Internal CPTPs to be added in two separate lists (one for edge and one for internal). Flow Points are removed by designating their server CPTPs (edge and internal) in a third list. This operation also allows A requesting OS to modify the attributes and parameters of the same Flow Domain Fragment, and of other involved CPTPs and FPs, in one request together with FP addition/removal. The requesting OS can modify the • user label • owner • network access domain • administrative state • connectionless layered parameters of the FDFr • layered parameters of the associated FPs and CPTPs • additional information

	of the FDFr. For each added CPTP and associated FP, the requesting OS can provide the following parameters: • Transmission parameters (incl. Traffic Mapping Table, Alarm Severity Assignment Profile names and TCA Parameter Profile names) • ingress/egress Transmission descriptor names • TCA parameter profile names. If the target OS works in the "connectivity-aware" mode, the requesting OS can request one of two modification results when not all new FPs have potential connectivity to all FPs already in the FDFr: • option 1) reject the modification request, or • option 2) add all FPs regardless of potential connectivity. The operation is best effort (except when the target OS is "connectivity-aware" and option 1 was requested).
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to modify a FDFr.
Description	1) The requesting OS sends a request to modify a Flow Domain Fragment (FDFr) to the target OS. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FDFr object is not known to the target OS, an Entity Not Found exception is raised. c) If uniqueness of the user label is required, the target OS checks the user label for uniqueness; i.e., if an FDFr object with the same user label exists already, then a User Label In Use exception is raised. d) If any of the TP objects specified for removal or modification is not known to the target OS, an Entity Not Found exception is raised. e) If any of the TP objects specified for removal or modification is not associated with the specified FDFr, an Entity Not Found exception is raised. f) If any of the CPTP objects specified for addition is not known to the target OS, an Entity Not Found exception is raised. g) If any of the CPTP objects specified for addition is not associated with the specified FDFr FD, an Entity Not

	Found exception is raised. Note that it is valid to include a CPTP which is already associated with the FDFr. No FP object creation notification is issued in this case, but there may be attribute value change notifications. h) If any of the CPTPs specified for addition as edge do not have the “FD Edge” role, a Not In Valid State exception is raised. i) If any of the CPTPs specified for addition as internal do not have the “FD Internal” role, a Not In Valid State exception is raised. j) If less than two end FPs would remain after successful completion of the operation, an Unable To Comply exception is raised. k) If any of the mandatory input parameters cannot be satisfied, an Unable To Comply exception is raised. l) In traffic mapping tables provided for specified FPs, if there is any unrecognized information or if there is internal inconsistency (e.g., inconsistent column lengths), an Unable To Comply exception is raised. m) In traffic mapping tables provided for specified FPs, if table contents would result in frames being mapped to more than one FDFr, Unable To Comply exception is raised. n) If fulfilling the command would create contradiction between the IVID value of the FDFr and the IVID value of any FP, an Unable To Comply exception is raised. o) If any of the mandatory transmission parameters of existing TPs cannot be set, an Unable To Comply exception is raised. 3) If the request is valid: a) If the existing value of the FDFr's IVID is not null (and is not being changed), any new FPs specified with IVID different from the FDFr's IVID must not be associated to the FDFr and their server CPTPs have to be returned in the “failed TP list” list. b) Any new CPTP or FP which cannot be provisioned, or for whom mandatory transmission parameters cannot be set, must not be added to the FDFr and must be returned in the “failed TP list” list (for every FP which is not added to the FDFr, the corresponding server CPTP has to be added to the list). New CPTPs and FPs, for which only best-effort transmissions parameters could not be set, have to be added to the new FDFr and must be returned in the “parameter problems TP list” list. Existing CPTPs and FPs, for which mandatory transmissions parameters could not be modified, must be returned in the “failed TP list” list. Existing CPTPs

	and FPs, for which only best-effort transmissions parameters could not be modified, must be returned in the "parameter problems TP list" list. If the target OS works in the "connectivity-aware" mode, it checks if the FPs to be added have potential connectivity to the already existing FPs of this FDFr. If not all of the new FPs have connectivity, and if option 1 (see summary) was requested, an Unable To Comply exception is raised c) The target OS modifies FDFr attributes and parameters as requested. Flow Points are created as clients of CPTPs which are successfully added to the FDFr, and their parameters are set. Flow Points, which are clients of CPTPs successfully removed from the FDFr, are deleted. The parameters of other involved TPs are modified as requested. 4) The target OS provides with a success indication. 5) The target OS sends object create, object delete, attribute value and state change notifications to the notification service.
Ends When	In case of success: 1) The requesting OS receives an indication of success of the action. 2) The target OS returns TPs which could not be added, or whose "mandatory" parameters could not be set/modified, in the "failed TP list". 3) The target OS returns TPs, whose "best effort" parameters could not be set/modified, in the "parameter problems TP list". 4) The target OS returns the modified FDFr with its distinguishing information, and sends In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: 1) The Flow Domain Fragment has been modified as requested. 2) The transmission parameters have been set/modified, as requested, to all CPTPs and FPs which are not contained in the "parameter problems TP list" or "failed TP list". 3) The TPs which could not be added to the FDFr are returned in the "failed TP list". 4) The target OS has forwarded a Flow Domain Fragment object modification notification and notifications for all objects which were added, removed, or modified. In case of failure:

	Nothing has changed in the System, i.e., the FDFr has not been created.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0062, R_TMF518_RP_II_0063, R_TMF518_RP_II_0065, R_TMF518_RP_II_0066 This use case is a generalization of Use Case 9.6.3 from TMF 513 v3.1

4.4 Gui CutThrough Control

For the three use cases in this section, it was agreed to consider only the NMS to EMS interface and not to generalize them at the OS to OS generic level. This generalization will be considered for further study.

4.4.1 The requesting OS retrieves GUI Cut-Through window data

Use Case Id	UC_TMF518_RP_0034
Use Case Name	The NMS retrieves GUI Cut-Through window data
Summary	The NMS is required to launch GCT. It retrieves the relevant GCT commands required to launch the GCT for each supported window type, or scope and context. Additionally, it receives indication if the EMS supports server-launch. The retrieval of the data is done through the NMS-EMS interface.UC_TMF518_RP_0035 describes the client-based launch and UC_TMF518_RP_0036 describes the server-based launch.
Actor(s)	The NMS
Pre-Conditions	The NMS and EMS have executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The NMS requires the relevant GCT window data
Description	The NMS gets the list of GCT parameters for all windows supported by EMS through the NMS-EMS interface. EMS indicates if it also supports server-based launch.
Ends When	In case of success: The NMS has all required GCT information. In case of failure: The NMS does not have the required GCT window data; The NMS cannot attempt to launch a GCT outside the interface.
Post-Conditions	The NMS has retrieved the GCT information from the EMS.
Exceptions	1. Internal error: The requested operation could not be performed.

	2. Not implemented: The target OS does not support this service.
Traceability	R_TMF518_RP_II_0070 This use case is a generalization of Use Case 5.12.1 from TMF 513 v3.0

4.4.2 Server based GCT launch

Use Case Id	UC_TMF518_RP_0036
Use Case Name	Server based GCT launch (e.g. using an X-protocol)
Summary	The GUI can be launched via a pure X-protocol, initiated by the EMS / server side once it is requested to do so through the MTNM interface. This USE CASE addresses the optional server-based GCT launch.
Actor(s)	The NMS
Pre-Conditions	none
Begins When	User request to display a GUI component that is managed by the EMS.
Description	1. The NMS request a launch of the GCT via the MTNM interface. 2. The EMS runs the GCT application on the X-server and attempts to redirect the display to the NMS display address. This step is outside of the MTNM interface and is shown as a dashed line in the figure below. 3. Once the GCT is displayed on the NMS and the system call returns, the server launch operation returns and indicates if it supports closing of GCT window.
Ends When	The display of a given GUI application becomes available at the NMS X-terminal.
Post-Conditions	none
Exceptions	none
Traceability	R_TMF518_RP_II_0072 This use case is a generalization of Use Case 5.12.3 from TMF 513 v3.0

4.5 Software and Data Control

No Use Case identified.

4.6 Termination Point Control

4.6.1 The requesting OS provisions the mapping mode of a CTP

Use Case Id	UC_TMF518_RP_0037
Use Case Name	The requesting OS provisions the mapping mode of a CTP
Summary	The requesting OS sets the mapping mode of a CTP (e.g., DS3, STS1, or VC4) to support client layer rate connections.
Actor(s)	requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. The requesting OS knows the CTP it wishes to terminate and map, e.g. has a handle for the CTP.
Begins When	The requesting OS sends a request to the target OS to terminate and map the CTP (server layer CTP).
Description	1. The target OS validates the request against the server layer CTP specified as described in steps 2 through 8. 2. The target OS validates if specified CTP exists. If not exception 1) is thrown. 3. The target OS validates if the specified CTP is capable of being mapped. 4. The target OS validates if the specified CTP is involved in an existing server layer cross-connection. If yes, the CTP cannot be terminated and mapped. 5. The target OS validates ME containing the specified CTP is not accessible, if not then exception 4) is thrown. 6. The target OS ignores any pending SNCs involving this CTP at either the server layer rate or a client layer rate. 7. Note that it may be possible to set a CTP that is involved in a partial subnetwork connection to terminated and mapped (even if the partial SNC is at the server layer rate) if this CTP is not involved in an existing active cross-connection. 8. The target OS validates if the CTP was already terminated and mapped, then the operation shall be considered a success. 9. The target OS terminates and maps the CTP specified. An AVC will be sent to the Notification Service. 10. If in the specific implementation of NE/target OS the client layer CTPs are actually created as a result of this operation, then it is assumed that the alarm reporting state of the newly created client layer CTPs will be disabled by default. Object creation notifications will not be generated for the newly created client

	layer CTPs. The target OS sends a response to the requesting OS
Ends When	In case of success: The requesting OS receives a response. In case of failure: The requesting OS receives a negative response or exception.
Post-Conditions	In case of success: 1. The server layer trail termination functions are available. 2. If in the specific implementation of NE/target OS the client layer CTPs are actually created then it is assumed that the alarm reporting state of the client layer CTPs will be disabled by default on application of the terminate and map operation. In case of failure: None.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Entity not found: The specified CTP does not exist. 3. Internal error: The requested operation could not be performed. 4. Unable to comply: The CTP is involved in an existing, active cross connection at the CTP's native rate (CTP's layer rate); or the CTP is not in a valid state. 5. Comm loss (Communication loss).
Traceability	R_TMF518_RP_II_0085 This use case is a generalization of Use Case 5.5.1 from TMF 513 v3.0

4.6.2 The requesting OS un-maps a server layer CTP

Use Case Id	UC_TMF518_RP_0038
Use Case Name	requesting OS un-maps a server layer CTP
Summary	The requesting OS sets the mapping mode of a server layer CTP (e.g., DS3, STS1, or VC4) to no longer be mapped to client layer capacity (e.g., VT Group/TUG/VT1.5/VC12 etc.) and to no longer terminate the corresponding server layer G.805 trail.
Actor(s)	requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. The requesting OS has a handle for a CTP that it wishes to

	modify.
Begins When	The requesting OS sends a request to the target OS to un-terminate and un-map a specified server layer CTP
Description	1. The target OS validates the request against the server layer CTP specified 2. The target OS validates if the specified server layer CTP exists 3. The target OS validates if the specified server layer CTP is capable of being un-mapped 4. The target OS validates if specified server layer CTP is involved in (is not supporting) an existing Active client layer cross-connection (i.e., the client CTPs (if any exist) are in a not connected connection state). If yes, the server layer CTP cannot be un-mapped. 5. Note that it is possible to adjust a Server layer CTP that is involved in a pending subnetwork connection, whether there are pending SNCs at a client layer rate or at the server layer rate. In other words, this operation ignores pending subnetwork connections involving the Server layer CTP or any client CTPs. 6. Note that it is possible to adjust a CTP that is involved in a partial subnetwork connection (even if the partial SNC is at a client layer rate) if none of the client CTPs are involved in an existing active client layer cross-connection. 7. If the Server layer CTP was already un-terminated and unadapted, then the operation shall be considered a success. 8. The target OS sets the mapping mode on the Server layer CTP to un-channelized/un-terminated. AVC will be sent to the Notification Service. 9. The target OS applies any necessary operations on the NE to deactivate traffic and management functions on the G.805 TTP, e.g., disables path trace processing, disables multiplexing/channelizing/mapping functions (makes the server layer CTP available for cross-connection at the server layer), removes VT visibility, etc. The alarm reporting state of the server layer CTP (and aggregated G.805 TTP) remains unchanged from its setting prior to use of this operation 10. The target OS sends a response to the requesting OS.
Ends When	In case of success: The requesting OS receives a response. In case of failure: The requesting OS receives a negative response.
Post-Conditions	In case of success: 1. The appropriate structure is set on the NE so that the Server layer CTP (e.g., an STS1 CTP) is then available to be part of an

	SNC at the CTP rate (e.g., STS1). In case of failure: None. The requesting OS has received a negative response.
Exceptions	1. Entity not found: The specified CTP does not exist 2. Internal error: The requested operation could not be performed. 3. Unable to comply: The CTP is supporting (containing) one or more lower order (client) CTPs that are involved in existing, active cross connections at the lower order CTP rates 4. Comm loss (Communication loss).
Traceability	R_TMF518_RP_II_0086 This use case is a generalization of Use Case 5.5.2 from TMF 513 v3.0

4.6.3 The requesting OS Provisions the TP Transmission Parameters

Use Case Id	UC_TMF518_RP_0039
Use Case Name	The requesting OS Provisions the TP Transmission Parameters
Summary	1. The requesting OS requests that a termination point's (TP) transmission parameters be provisioned. For example, the types of TP transmission parameters that can be provisioned include for SONET/SDH/DWDM: • For DS1: frame format (e.g. SF, ESF, and Unframed) and line coding (e.g. B8ZS and AMI) • For DS3: frame format (e.g. M23, C-bit parity, and Unframed) • For STS-1: expected incoming and outgoing path trace • For DWDM TP: TunnedFequency, etc. 2. The requesting OS must be aware that the provisioning of a TP transmission parameter when the TP is actively involved in a SNC may cause service disruption. The provisioning of a connected TP does not involve in the tearing down of the associated cross-connect. **Note:** The specific TP transmission parameters and values varies across different vendors and technologies.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA

	document.
Begins When	The requesting OS sends the request to set a transmission parameter(s) on the TP.
Description	1. The requesting OS sends the request to set transmission parameter(s) on the specified TP. 2. The target OS validates the TP reference (e.g. name). 3. The target OS validates the request (e.g. transmission parameter name(s) exists and value(s) are valid) and supported. 4. The target OS sets the specified TP's transmission parameter(s) to the specified value(s). 5. If there has been a transmission parameter value change(s), then the target OS forwards attribute value change notification(s) to all subscribing requesting OSs. 6. The target OS replies with a success indication.
Ends When	In case of success: The requesting OS receives an indication of success of the action. In case of failure: The requesting OS receives an indication of failure of the action.
Post-Conditions	In case of success: 1. The TP's transmission parameter(s) have been set to the specified value. In case of partial success: If the request to set the transmission parameter(s) succeeded and the TP's transmission parameter(s) have changed in value, then the target OS has forwarded attribute value change notification(s) to all subscribing requesting OSs. In case of failure: Appropriate exception is reported to the requesting OS.
Exceptions	6. Not implemented: The target OS does not support this service. 7. Internal error: The requested operation could not be performed. 8. Unable to comply. 9. Invalid input: The TP reference is invalid. 10. Comm loss (Communication loss).
Traceability	R_TMF518_RP_II_0088 This use case is a generalization of Use Case 5.5.7 from TMF 513 v3.0

4.6.4 The requesting OS creates a Group Termination Point

Use Case Id	UC_TMF518_RP_0049
Use Case Name	The requesting OS creates a Group Termination Point
Summary	The use case describes how the requesting OS requests that a target OS create a new GTP. The requesting OS requests the creation of the GTP by either: 1. listing the CTPs that comprise the GTP, or 2. in the case of contiguous CTPs of the same layer rate, the NMS may list the first CTP and the number of following (contiguous) CTPs. The resulting GTP is returned as a result to the requesting OS.
Actor(s)	
Pre-Conditions	3. The requesting OS and the target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 4. The requesting OS has defined a Topological Link
Begins When	The requesting OS sends the request to the Target OS to create a Group Termination Point (GTP).
Description	1. The requesting OS sends a create GTP request to Target OS with the GTP distinguishing information. The requesting OS can provide either a listing of the specific CTPs that comprise the GTP, or the requesting OS may list the first CTP and the number of following CTPs (this second approach is only valid for contiguous CTPs of the same layer rate). The requesting OS also needs to determine the gtpEffort level. If gtpEffort is set to EFFORT_SAME, then the Target OS must create the GTP with the exact same list of CTPs as provided with the GTP creation request. If EFFORT_WHATEVER is specified, then the Target OS may comply with the total bandwidth requirement by using a different set of CTPs. It should be noted that this mode (i.e., EFFORT_WHATEVER) allows for the GTP components to be instantiated at a later time by the ME (e.g., upon detection of user's signal). Therefore the operation may successfully return a new GTP with an empty listOfTPs attribute (to be updated at a later time once the component CTPs are created in the ME). 2. The Target OS validates the request. The target OS creates the GTP, assigns a unique name to the GTP and stores it persistently. If the requesting OS has specified EFFORT_WHATEVER, the target OS may create the GTP with a different list of CTPs than specified by the requesting OS, or create the GTP with an empty list of CTPs. In the latter case, the Target OS supplies the CTPs at some later point in time (typically upon detection of a user signal connecting to the GTP). 3. The target OS replies with a success indication. 4. A Create GTP Notification is sent by Target OS to the notification service.

	Note: It is up to the internal implementation of Target OS which data are stored in Target OS persistently and which ones will be queried from ME as required.
Ends When	In case of success: The requesting OS receives an indication of success of the action. In case of failure: The requesting OS receives an indication of failure of the action.
Post-Conditions	In case of success: The GTP is available. In case of failure: None. The requesting OS has received an indication of failure of the action or exception
Exceptions	1. Not implemented: The target OS does not support this service. 2. Internal error: The requested operation could not be performed. 3. Unable to comply. 4. Invalid input: The TP reference is invalid. 5. Comm loss (Communication loss). 6. The user label supplied by the requesting OS is already in use. 7. The Input data is valid but the Target OS cannot create the GTP because there are insufficient resources in the network to create the GTP.
Traceability	R_TMF518_RP_II_0077 This use case is a generalization of Use Case 7.5.18 from TMF 513 v3.1

4.6.5 The requesting OS modifies a Group Termination Point

Use Case Id	UC_TMF518_RP_0050
Use Case Name	The requesting OS modifies a Group Termination Point (GTP)
Summary	The use case describes how the requesting OS requests that an target OS modify an existing GTP that has a non-empty list of CTPs. The modify GTP operation is used to add or delete CTPs from a GTP. For a given request, the requesting OS can only add or delete CTPs, not both. It is not possible to add a CTP that is already involved in a cross connection or SNC, or that is part of another GTP. Attempts to modify a GTP that is involved in a cross connection (or SNC) should be rejected by the target OS. The operation is best-effort, i.e., the target OS will add or delete as many of the identified CTPs as possible. If the service is called with the name of a non-existent GTP or CTP, it will fail. If the GTP was initially created with gtpEffort equal to EFFORT_SAME, the target OS should reject the modification request.

Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and the target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. The GTP to be modified already exists, and the target OS has assigned a set of CTPs to the GTP.
Begins When	The requesting OS sends the modify GTP request to the target OS.
Description	1. The requesting OS sends a modify GTP request to target OS with the GTP distinguishing information. 2. If the request involves CTP additions, the target OS will attempt to add as many of the CTPs as possible. If the request involves CTP deletions, the target OS will attempt to delete as many of the requested CTPs as possible. 3. The target OS replies with a success indication and an updated listing of the CTPs that comprise the modified GTP. 4. The target OS sends an AVC notification to the notification service, indicating that the list of CTPs for the GTP has changed.
Ends When	The target OS sends a reply to the requesting OS.
Post-Conditions	The GTP has been modified in the target OS.
Exceptions	1. Service not implemented by the target OS. 2. The requesting OS has supplied invalid input data. 3. The user label supplied by the requesting OS is already in use. 4. The target OS has not yet assigned a set of CTPs to the GTP. 5. The GTP was initially created with gtpEffort equal to EFFORT_SAME. 6. The Input data is valid but the target OS cannot modify the GTP because there are insufficient resources in the network to make the requested modification.
Traceability	R_TMF518_RP_II_0080 This use case is a generalization of Use Case 7.5.19 from TMF 513 v3.1

4.6.6 The requesting OS deletes a Group Termination Point

Use Case Id	UC_TMF518_RP_0051
Use Case Name	The requesting OS deletes a Group Termination Point (GTP)
Summary	The requesting OS deletes a GTP. This operation is idempotent. If the service is called with the name of a non-existent GTP, it will succeed.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and the target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA

	document.
Begins When	The requesting OS sends the delete GTP request to the target OS.
Description	1. The requesting OS sends a delete GTP request to the target OS. 2. The target OS validates the GTP identifier and checks that this GTP is not being used in a XC or SNC. 3. If no XCs or SNCs are using this GTP, the target OS deletes the GTP. 4. The target OS replies with a success indication. 5. A GTP object deletion event is sent by target OS to the notification service.
Ends When	The target OS sends a reply to the requesting OS.
Post-Conditions	The GTP has been deleted in the target OS.
Exceptions	1. Service not implemented 2. Communication Failure with NE. 3. The GTP is being used by an XC or an SNC. 4. The requesting OS has supplied invalid input data (The TMD Name is invalid).
Traceability	R_TMF518_RP_II_0079 This use case is a generalization of Use Case 7.5.20 from TMF 513 v3.1

4.6.7 The requesting OS creates a Termination Point Pool

Use Case Id	UC_TMF518_RP_0052
Use Case Name	The requesting OS creates a Termination Point Pool (TP Pool)
Summary	The use case describes how the requesting OS requests that an target OS create a TP Pool. The requesting OS passes the TP Pool create information to the target OS. The resulting TP Pool is returned as a result to the requesting OS. Remark: • It is up to the requesting OS whether it maintains TP Pools across multiple administrative subnetworks and hence target OSs.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and the target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends the request to the target OS to create a TP Pool.
Description	1. The requesting OS sends a create TPPool request to target OS with the TPPool creation information. The requesting OS may request that the target OS enforce a unique user label (user label

	is selected by the requesting OS and the target OS makes sure the user label is already in use). 2. The target OS validates the request. The target OS creates the TP Pool, assigns a unique name to the TPPool and stores it persistently. 3. The target OS replies with a success indication. 4. A TPPool object creation event is sent by target OS to the notification service.
Ends When	In case of success: The requesting OS receives an indication of success of the action. In case of failure: The requesting OS receives an indication of failure of the action.
Post-Conditions	In case of success: The TPPool is available. In case of failure: None. requesting OS has received an indication of failure of the action or exception
Exceptions	1. Service not implemented by the target OS 2. The requesting OS has supplied invalid input data. 3. The user label supplied by the requesting OS is already in use. 4. The Input data is valid but the target OS cannot create the TPPool because the target OS has already exceeded its maximum number of TP Pools or the requested TPPool is too large.
Traceability	R_TMF518_RP_II_0081 This use case is a generalization of Use Case 7.5.21 from TMF 513 v3.1

4.6.8 The requesting OS modifies a Termination Point Pool

Use Case Id	UC_TMF518_RP_0053
Use Case Name	requesting OS modifies a Termination Point Pool (TP Pool)
Summary	The requesting OS changes the content of a TPPool.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and the target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. 2) The TP pool to be modified has to exist. 3. 3) TPs or GTPs to be removed have to be idle. 4. 4) TPs or GTPs to be added have to exist and have to belong to the TPPool's Subnetwork and must not be contained in a GTP or another TP pool.

Begins When	The requesting OS sends the modify TPPool request to the target OS.
Description	1. The requesting OS sends the modify TPPool request to the target OS, i.e. requests to add or delete TPs or GTPs to or from the TPPool. 2. The target OS validates the passed TP identifiers. 3. The target OS validates the TPPool identifier. 4. The target OS changes the TPPool attributes if possible. This will fail under the following conditions: • One or more passed TPs are “in use”. • One or more passed TPs are contained in a GTP. 5. The target OS replies with a success indication. 6. The target OS will generate an AVC for the TPPool with the changed attributes.
Ends When	The target OS sends a reply to the requesting OS.
Post-Conditions	The TPPool has been modified in the target OS.
Exceptions	1. Service not implemented by the target OS. 2. Non-specific target OS internal failure. 3. The requesting OS has supplied invalid input data (i.e., the TPPool identifier of one or more TP identifiers are invalid). 4. One or more TPs or GTPs do not exist or do not belong to the TPPool's Subnetwork, and so the addressed TPPool can not be modified. 5. An input parameter references an object that does not exist. 6. One or more TPs or GTPs to be removed are not idle, or one or more TPs or GTPs to be added are contained in a GTP or another TPPool.7)The operation would result in resources
Traceability	R_TMF518_RP_II_0084 This use case is a generalization of Use Case 7.5.22 from TMF 513 v3.1

4.6.9 The requesting OS deletes a Termination Point Pool

Use Case Id	UC_TMF518_RP_0054
Use Case Name	The requesting OS deletes a Termination Point Pool (TP Pool)
Summary	The requesting OS deletes a TPPool. Remark: • The requesting OS is responsible to maintain the consistency of TPPool across multiple administrative subnetworks and hence target OSs. Routine integrity checking may be required. requesting OS must also check that the TPPool is not being used in any target OS under its network domain, before the

	TPPool can be deleted from the persistent storage. • This operation is idempotent. If the service is called with the name of a nonexistent TPPool, it will succeed.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and the target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. The TP pool to be deleted must not be in use, i.e. all of its members must be idle.
Begins When	The requesting OS sends the delete TPPool request to the target OS.
Description	1. The requesting OS sends a delete TPPool request to the target OS. 2. The target OS validates the TPPool identifier and checks that this TPPool is not being used on the target OS, i.e. has all its member TPs or GTPs idle. 3. If no TPs or GTPs are using this TPPool, the target OS deletes the TPPool.4)The replies with a success indication. 4. A TPPool object deletion event is sent by target OS to the notification service.
Ends When	The target OS sends a reply to the requesting OS.
Post-Conditions	The TPPool has been deleted in the target OS. (Note: The TPPool can be deleted in the requesting OS if and only if no target OS uses this TD.)
Exceptions	1. Service not implemented 2. Non-specific target OS internal failure. 3. The TPPool is being used (i.e., not all contained TPs or GTPs are idle). 4. The requesting OS has supplied invalid input data (i.e., the TPPool name is invalid).
Traceability	R_TMF518_RP_II_0083 This use case is a generalization of Use Case 7.5.23 from TMF 513 v3.1

4.6.10 The requesting OS creates a Floating Termination Point

Use Case Id	UC_TMF518_RP_0040
Use Case Name	The requesting OS creates a Floating Termination Point
Summary	This operation allows A requesting OS to create a Floating Termination Point (FTP).

	The target OS creates the requested FTP consistently with its implementation and the requesting OS specifications. If the target OS cannot create the FTP as specified, an appropriate exception is raised. The requesting OS specifies the FTP through choosing a location within the network element (e.g., shelf and card), a list of layer rates and related transmission parameters. After the target OS validates the input data, and if it can satisfy the input constraints (e.g., specific bandwidth as a virtually concatenated layer rate), it proceeds with implementing the FTP (and its contained CTPs, if applicable) and returns the name of the new FTP and all its attributes to the requesting OS. If the target OS fails to validate the input, it raises an appropriate exception. If transmission parameters are specified for the contained CTPs which are created (where applicable), the target OS will apply them after the creation of the FTP. The alarm reporting on the FTPs and the contained CTPs may be turned on by the target OS, unless otherwise specified via the alarm reporting transmission parameter. An object creation notification is sent to notify the requesting OS(s) about the existence of the new FTP (note that no notification is sent related to the contained CTPs in case virtual concatenation applies).
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends a request to the target OS to create an FTP.
Description	1) The requesting OS sends the request to the target OS to create an FTP with the provided parameters. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If explicit creation of the requested FTP not supported, an Unable To Comply exception is raised. c) If the specifed equipment object is not known to the target OS, an Entity Not Found exception is raised. d) If the target OS supports name specification by the requesting OS, and if a name was specified, and if there is already an FTP with the specified name, an Object In Use exception is raised. e) If uniqueness of the user label is required, the target OS checks the user label for uniqueness; i.e., if an FTP object with the same user label exists already then a User Label In Use exception is raised. f) If not all required resources are available (e.g., enough overall back-plane bandwidth, enough usable timeslots,

	etc.), a Capacity Exceeded exception is raised. g) If the input parameters are incompatible, an Unable To Comply exception is raised. 3) If the request is valid: a) The target OS creates the FTP and its contained CTPs if applicable (e.g., in case the FTP layer rate implies inverse multiplexing). CTP instances are either chosen by the target OS, or created according to requesting OS specification. b) If contained CTPs were created, and if transmission parameters were specified for them, the target OS applies the parameters. 4) The target OS replies with a success indication. a) The target OS returns the name of the new FTP. If the requesting OS specified a name, and if the target OS supports naming by requesting OS, the requesting OS specified name is returned, else A target OS created name is returned. 5) The target OS sends the appropriate object creation notification to the notification service. No notification is sent for the contained CTPs. (The CTPs can be retrieved by explicit request.)
Ends When	In case of success: The target OS returns the created FTP name and all its attributes to the requesting OS. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: 1) The FTP has been created. 2) The contained fragment CTPs have been created (if applicable). In case of failure: No change occurred to the system, i.e., the FTP has not been created.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0090, R_TMF518_RP_II_0091 This use case is a generalization of Use Case 9.2.1 from TMF 513 v3.0

4.6.11 The requesting OS deletes a FloatingTermination Point

Use Case Id	UC_TMF518_RP_0041
Use Case Name	The requesting OS deletes a Floating Termination Point
Summary	This operation allows the requesting OS to delete a Floating Termination Point (FTP) that already exists. The target OS is required to verify that the FTP or the contained fragment CTPs are not cross-connected. The deletion request will fail • if the FTP is a Call endpoint, or • if the FTP or any of its contained server CTPs is an SNC endpoint, or • if the CPTP cannot be explicitly deleted (e.g., was automatically created by the target OS). • If the FTP is a CPTP which is assigned or has the fdEdge role.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends the request to the target OS to delete an FTP.
Description	1) The requesting OS sends the request to the target OS to delete an FTP. 2) The target OS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FTP object is not known to the target OS, an Entity Not Found exception is raised. c) If the specified FTP object or any of its contained CTP objects are cross-connected, an Unable To Comply exception is raised. d) If the specified FTP object terminates a Call, an Unable To Comply exception is raised. e) If the specified FTP is a CPTP, and it is not in the unassigned role, an Object In Use exception is raised. f) If the specified FTP cannot be deleted (e.g., was not automatically created by the target OS), an Unable To Comply exception is raised. g) If there is any incompatibility among the input parameters (e.g., layered transmission parameters consistent with the list of layer rates), an Unable To Comply exception is raised.

	3) If the request is valid the target OS deletes the FTP and any contained CTPs. 4) The target OS replies with a success indication. 5) The target OS sends an FTP object deletion notification to the notification service.
Ends When	In case of success: The requesting OS receives an indication of the success of the request. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The FTP and its contained CTPs have been deleted. In case of failure: No change occurred to the system, (i.e., the FTP or any of its contained CTPs have not been deleted or modified).
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements.
Traceability	R_TMF518_RP_II_0092 This use case is a generalization of Use Case 9.2.2 from TMF 513 v3.0

4.7 Transmission Descriptor Control

Use Case Id	UC_TMF518_RP_0042
Use Case Name	The requesting OS creates a Transmission Descriptor (TMD)
Summary	The use case describes how the requesting OS requests that the target OS create a Transmission Descriptor (TMD). The requesting OS passes the TMD information to the target OS. The resulting TMD is returned as a result to the requesting OS.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends the create TMD request to the target OS.
Description	1. The requesting OS sends a create TMD request to the target OS with the TMD creation information. The requesting OS may

	request that the target OS enforce a unique user label (the user label is selected by the requesting OS and the target OS makes sure the user label is not already in use). 2. The target OS validates the request. The target OS creates the TMD, assigns a unique name to the TMD and stores it persistently. 3. The target OS replies with a success indication. 4. A TMD object creation event is sent by the target OS to the notification service. **Note:** It is up to the internal implementation of the target OS which data are stored in the target OS persistently and which ones will be queried from ME as required.
Ends When	The target OS sends a reply to the requesting OS.
Post-Conditions	The TMD has been created in the target OS
Exceptions	1. Not implemented: The target OS does not support this service. 2. Invalid input: The requesting OS has supplied invalid input data. 3. User label in use: The user label supplied by the requesting OS is already in use. 4. Capacity exceeded: The Input data is valid but the target OS cannot create the TMD because the target OS has already exceeded its maximum number of TMDs. 5. Internal error: The requested operation could not be performed. 6. Comm loss (Communication loss).
Traceability	R_TMF518_RP_II_0093 This use case is a generalization of Use Case 5.5.12 from TMF 513 v3.0

Use Case Id	UC_TMF518_RP_0043
Use Case Name	The requesting OS modifies a Transmission Descriptor (TMD) on a TP
Summary	The requesting OS changes the Transmission Descriptor (TMD) (ingress and/or egress) for a TP.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. TransmissionDescriptor to be associated has to exist.
Begins When	The requesting OS sends the modify TP request to the target OS.
Description	1. The requesting OS sends the modify TP request to the target OS. 2. The target OS validates the TP identifier.

	3. The target OS validates the TMD Name(s). 4. The target OS communicates with the Network Elements (NE). 5. The target OS changes the TMD(s) if possible. This will fail under the following conditions: • The TP is terminating, but was not explicitly created. • The input data is valid but the target OS cannot change the TMD because resources are not available on the NE. • The input data is valid but the target OS cannot change the TMD because the new TMD does not provide enough resources for the existing SNCs running over the TP. 6. The target OS replies with a success indication. 7. The target OS will generate an AVC for the TP with the new TMD name.
Ends When	The target OS sends a reply to the requesting OS.
Post-Conditions	The TMD(s) on the TP has been updated.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Internal error: The requested operation could not be performed. 3. Invalid input: The requesting OS has supplied invalid input data (e.g. the TP or TMD identifier is invalid). 4. Entity not found: The TP or TMD does not exist. 5. Unable to comply: The transmission parameter values could not be configured in the TP. 6. Comm loss (Communication loss).
Traceability	R_TMF518_RP_II_0097 This use case is a generalization of Use Case 5.5.13 from TMF 513 v3.0

Use Case Id	UC_TMF518_RP_0056
Use Case Name	NMS sets the Transmission Descriptor (TMD) Profile Pointer
Summary	The purpose of this use case is to configure the Layered Transmission Parameters of a Termination Point (TP) or a Matrix Flow Domain (MFD) by modifying the assignment of the Transmission Descriptor (TMD) to this TP/MFD.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The NMS sends the request to the EMS.
Description	1. The NMS sends the request to the EMS to set the TMD pointer. 2. The NMS may provide:

	• an ingress TMD name (in case of a TP) or • an egress TMD name (in case of a TP)or • a TMD name (in case of an MFD) If a TMD has to be removed, the NMS has to provide an “empty TMD name” instead of the TMD name. 3. The EMS checks the syntax of the request. If the request is not valid, an Invalid Input exception is raised. 4. If a TMD has to be removed: The EMS checks the existence of the TMD to be removed. If the TMD is not associated, an Entity Not Found exception is raised. Only the pointer attribute in the TP/MFD which contains the name of this TMD has to be updated and an AVC notification has to be send. Note: The parameter values in the TP/MFD remain unchanged. 5. If a TMD has to be added: The EMS checks the existence of the TMD to be added. If the TMD does not exist,an Entity Not Found exception is raised. If the TMD to be added is already associated to the TP/MFD, the EMS shall overwrite all corresponding parameters in the TP/MFD with the values contained in the TMD. Parameters which are not already in the TP/MFD will be added to the TP/MFD. This request ensures consistency between the parameter values of the TMD and the parameter values of the TP/MFD. (Note: Individual values may have been changed in the TP/MFD before via the use cases "NMS provisions the TP Transmission Parameters" and "NMS modifies a Matrix Flow Domain".) If the parameter values can not be set on the TP/MFD, an Unable To Comply exception is raised. If the TMD is not already assigned to the TP/MFD the EMS shall set all parameter values as defined in the TMD profile. If the parameter values can not be set on the TP/MFD, an Unable To Comply exception is raised. The pointer attribute in the TP/MFD which contains the name of this TMD has to be updated and an AVC notification has to be send.
Ends When	In case of success: The parameter values of the TP/MFD are consistent with the parameter values of the (newly) associated TMD profile. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The parameter values of the TP/MFD are consistent with the parameter values of the (newly) associated TMD profile. In case of failure: Nothing has changed in the System.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements

Traceability	R_TMF518_RP_II_0044 R_TMF518_RP_II_0045 R_TMF518_RP_II_0097 R_TMF518_RP_II_0098 This use case is a generalization of Use Case 7.5.14 from TMF 513 v3.1

Use Case Id	UC_TMF518_RP_0055
Use Case Name	The requesting OS modifies a Transmission Descriptor (TMD)
Summary	The purpose of this use case is to modify the attributes and transmission parameters in an already created Transmission Descriptor (TMD). This operation overwrites specific parameter values of the TMD with the new provided parameter values. Existing Parameters can also be removed from the TMD. The corresponding parameter values of all associated TPs/MFDs are changed accordingly.
Actor(s)	The requesting OS
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. The TMD to be modified must exist. 3. A communication between the target OS and the relevant MEs has to be active.
Begins When	The requesting OS sends the request to the target OS.
Description	1. The requesting OS sends the request to the target OS to modify the TMD parameters. 2. The requesting OS provides the list of attribute values and transmission parameter values which have to be modified in the TMD. 3. The target OS checks the syntax of the request. If the request is not valid, an Invalid Input exception is raised. 4. The target OS checks the existence of the TMD to be modified. If the TMD does not exist, an Entity Not Found exception is raised. 5. The target OS overwrites the attributes and parameters of the TMD with the provided ones. This includes the deletion of parameters. If it is not possible to change any attribute/parameter value, an Unable To Comply exception is raised. 6. The target OS checks the connection to all MEs that contains associated TPs or MFDs. If connection to all MEs is lost, an NE Comm Loss exception is raised. Otherwise the target OS has to return the names of the MEs which are not connected. 7. The target OS changes all parameter values in all TPs and MFDs associated to this TMD according to the modified values; i.e., only the modified parameters will be updated.

	Note: Parameters which have been deleted in the TMD will not be changed in the associated TPs and MFDs. 8. The target OS has to return the names of all TPs and MFDs that could not be changed to the new parameter values due to some error reasons.
Ends When	In case of success: The target OS returns the distinguishing information of the modified TMD. In case not all associated TPs/MFDs could be updated to the modified attribute and parameter values, the target OS returns the list of these TPs/MFDs. In case of failure: The requesting OS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The TMD contains the new provided attribute and parameter values. Every TP/MFD (except the TPs/MFDs returned as failed TPs/MFDs) associated to this TMD has been set to the new parameter values and will use the new values. In case of failure: Nothing has changed in the System.
Exceptions	Refer to Category III: Abnormal or Exception Conditions, Dynamic Requirements
Traceability	R_TMF518_RP_II_0106 This use case is a generalization of Use Case 7.5.15 from TMF 513 v3.1

Use Case Id	UC_TMF518_RP_0044
Use Case Name	The requesting OS deletes a Transmission Descriptor (TMD)
Summary	The requesting OS deletes a Transmission Descriptor (TMD). Remark: • The requesting OS is responsible to maintain the consistency of TMD across multiple administrative subnetworks. Routine integrity checking may be required. requesting OS must also check that the TMD is not being used in any the target OS under its network domain, before the TMD can be deleted from the persistent storage. • This operation is idempotent. If the service is called with the name of a non-existent TMD, it will succeed.
Actor(s)	The requesting OS
Pre-Conditions	9. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends the delete TMD request to the target OS.
Description	1. The requesting OS sends a delete TMD request to the target OS.

	2. The target OS validates the TMD identifier and checks that this TMD is not being used on the target OS. 3. If no TPs are using this TMD, the target OS deletes the TMD. 4. The target OS replies with a success indication. 5. A TMD object deletion event is sent by the target OS to the notification service.
Ends When	The target OS sends a reply to the requesting OS.
Post-Conditions	The TMD has been deleted in the target OS.
Exceptions	1. Not implemented: The target OS does not support this service. 2. Comm loss (communication loss) 3. Object in use: The TMD is being used. 4. Entity not found: The TMD does not exist in the target OS. 5. Invalid input: The requesting OS has supplied invalid input data (The TMD Name is invalid). 6. Comm loss (Communication loss).
Traceability	R_TMF518_RP_II_0095 This use case is a generalization of Use Case 5.5.14 from TMF 513 v3.0

4.8 Topological Link Control

4.8.1 The requesting OS creates a Topological Link (TL)

Use Case Id	UC_TMF518_RP_0045
Use Case Name	The requesting OS creates a Topological Link (TL)
Summary	The requesting OS creates a Topological Link (TL).
Actor(s)	The requesting OS.
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends the request to the target OS to create a Topological Link.
Description	1. The target OS validates the request syntax. If not exception 1) is thrown. 2. The target OS validates the TL name is not already in use. As owner of TPs and TLs, the target OS may perform this validation against data that was manually entered at the target OS or auto-discovered. This target OS authority to use auto-discovered information also applies to the following data validations. If not

	exception 2) is thrown. 3. The target OS validates that the A and Z end TPs referenced in the request exist in its domain of control. If not exception 3) is thrown. 4. The target OS validates that neither TP is already assigned to a TL. If not exception 2) is thrown. 5. The target OS validates that the rate specified in the request, as well as any other additional creation information, is consistent with its own data. If not exception 2) is thrown. 6. As owner of TPs and TLs, the target OS validates the request against its own criteria. If not exception 4) is thrown. 7. The target OS creates the TL. 8. The target OS sends a response to the requesting OS.
Ends When	In case of success: The requesting OS receives an indication of success of the action. In case of failure: The requesting OS receives an indication of failure of the action.
Post-Conditions	In case of success: The TL is available. In case of failure: None. The requesting OS has received an indication of failure of the action or exception
Exceptions	1. Invalid input: request syntax is not valid. 2. Object in use: the request attempts to create a TL that already exists, or the request references TPs that are already associated with TLs. 3. Entity not found: the request references an entity that does not exist. 4. Unable to comply: the target OS rejects the request based on its own criteria. 5. Internal error: The requested operation could not be performed. 6. User label in use: The user label supplied by the requesting OS is already in use. 7. Comm loss (communication loss). 8. Not implemented.
Traceability	R_TMF518_RP_II_0100 This use case is a generalization of Use Case 5.5.10 from TMF 513 v3.0

4.8.2 The requesting OS deletes a Topological Link (TL)

Use Case Id	UC_TMF518_RP_0046
Use Case Name	The requesting OS deletes a Topological Link (TL)
Summary	The requesting OS deletes a Topological Link (TL).
Actor(s)	The requesting OS.
Pre-Conditions	1. The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document. 2. The requesting OS has identified an existing Topological Link for deletion.
Begins When	The requesting OS sends the request to the target OS to delete a Topological Link.
Description	1. The target OS validates the request syntax. If not exception 1) is thrown. 2. The target OS validates that the TL exists in its span of control. If not, exception 2 is thrown. 3. As owner of TLs, the target OS validates the request against its own criteria. If not exception 3) is thrown. 4. The target OS deletes the TL. 5. The target OS sends a response to the requesting OS.
Ends When	In case of success: The requesting OS receives an indication of success of the action. In case of failure: The requesting OS receives an indication of failure of the action.
Post-Conditions	In case of success: The TL is deleted. In case of failure: None.
Exceptions	1. Invalid input: request syntax is not valid. 2. Entity not found: the request references a TL that does not exist. 3. Unable to Comply: the target OS rejects the request based on its own criteria. 4. Internal error: The requested operation could not be performed.

	5. Comm loss (communication loss). 6. Not implemented.
Traceability	R_TMF518_RP_II_0102 This use case is a generalization of Use Case 5.5.11 from TMF 513 v3.0

4.9 ATM Connection Management

4.9.1 The requesting OS creates and activates an ATM Subnetwork Connection (SNC)

Use Case Id	UC_TMF518_RP_0047
Use Case Name	The requesting OS creates and activates an ATM Subnetwork Connection (SNC)
Summary	The requesting OS sets up a Subnetwork Connection on A target OS Subnetwork. The requesting OS supplies the CTP names or NSAP addresses. This use case extends the UC_TMF518_RP_0003. The only differences are with respect to that use case are specified here.
Actor(s)	The requesting OS
Pre-Conditions	The requesting OS and target OS have successfully executed the Use Case 0001 OS (Re) Starts as defined in the TMF518_FMW BA document.
Begins When	The requesting OS sends the Create and Activate SNC request to the target OS.
Description	1) The target OS configures the end point CTPs. The target OS should try to determine if the operation will succeed before making any configuration changes. This would be done as part of validation. It is preferable not to change the configuration of one CTP if the target OS can determine that the next CTP cannot be changed. For each CTP in the SNC the target OS will do the following: • If the CTP is already used by a PARTIAL or ACTIVE SNC, the request is rejected. • For a connection at the VP layer, i) If the VP CTP was previously explicitly configured to contain VC CTPs the request will be rejected. (i.e. if it is TERMINATED_AND_AVAILABLE_FOR_MAPPING with an assigned Traffic Descriptor (assigned bandwidth)) ii) If there are VC CTPs using the VP CTP, then the

	request will be rejected. iii) If the VP CTP is already NEITHER_TERMINATED_NOR_AVAILABLE_FOR_MAPPING it will be left in this mode. • For a connection at the VC layer, i) If the VP CTP was previously either explicitly or implicitly configured to contain VC CTPs the mapping mode of the VP CTP will be left unchanged (TERMINATED_AND_AVAILABLE_FOR_MAPPING). ii) If the VP CTP is configured as NEITHER_TERMINATED_NOR_AVAILABLE_FOR_MAPPING, the VP CTP will be implicitly configured to contain VC CTPs. Note that there will be no bandwidth assigned to the terminating VP CTP (The Traffic Descriptors will be empty). In the above cases, if the mapping mode is changed, an AVC will be generated. The TP Traffic Descriptors specified on the end points are used. 2) For the intermediate points along the route of the SNC • If the SNC is created using network routing protocols, there will be no notifications regarding the connection state or mapping mode of the intermediate points along the route • For all other cases, the target OS may have to configure the mapping mode. Also appropriate notifications will be sent for the connection state and the mapping mode changes. 3) The target OS activates the required SNC in the network. A network routed SNC will be an atomic operation from the cross-connect perspective. However the SNC can still use the PARTIAL state. If an SPVX source switch cannot reach the destination, it is considered to be PARTIAL because one or more resources have been allocated. The behavior is that the network routing protocol will try to establish the connection when the required links are in place.
Ends When	A target OS routed SNC may use the PARTIAL SNC state In a network routed SNC, the configuration of the endpoints (if they are in the target OS domain) will be included as part of the activate operation. If the target OS fails to create and activate the SNC it will be in the PARTIAL state as some resources are allocated.
Post-Conditions	Refer to UC_TMF518_RP_0003.
Exceptions	Refer to UC_TMF518_RP_0003
Traceability	R_TMF518_RP_II_0009, R_TMF518_RP_II_0010 This use case is a generalization of Use Case 5.14.1 from TMF 513 v3.0

5 Traceability Matrices

Table 5-1. Use Cases – Requirements Traceability Matrix

Use Case Id	Use Case Name	Requirements
UC_TMF518_RP_0001	The requesting OS creates a Subnetwork Connection (SNC)	R_TMF518_RP_II_0002, R_TMF518_RP_II_0003, R_TMF518_RP_II_0005 This use case is a generalization of Use Case 5.6.1 from TMF 513 v3.0
UC_TMF518_RP_0002	The requesting OS activates a Subnetwork Connection (SNC)	R_TMF518_RP_II_0007, R_TMF518_RP_II_0008 This use case is a generalization of Use Case 5.6.2 from TMF 513 v3.0
UC_TMF518_RP_0003	The requesting OS creates and activates a Subnetwork Connection (SNC)	R_TMF518_RP_II_0009, R_TMF518_RP_II_0010 This use case is a generalization of Use Case 5.6.3 from TMF 513 v3.0
UC_TMF518_RP_0004	The requesting OS adds a route to a Subnetwork Connection (SNC)	R_TMF518_RP_II_0011, R_TMF518_RP_II_0012 This use case is a generalization of Use Case 5.6.4 from TMF 513 v3.0
UC_TMF518_RP_0005	The requesting OS removes a route from a Subnetwork Connection (SNC)	R_TMF518_RP_II_0013, R_TMF518_RP_II_0014 This use case is a generalization of Use Case 5.6.5 from TMF 513 v3.0
UC_TMF518_RP_0006	The requesting OS creates-modifies the route of a Subnetwork Connection (SNC)	R_TMF518_RP_II_0015, R_TMF518_RP_II_0016 This use case is a generalization of Use Case 5.6.6 from TMF 513 v3.0
UC_TMF518_RP_0007	The requesting OS deactivates a Subnetwork Connection (SNC)	R_TMF518_RP_II_0026, R_TMF518_RP_II_0027 This use case is a generalization of Use Case 5.6.7 from TMF 513 v3.0

UC_TMF518_RP_0008	The requesting OS deletes a Subnetwork Connection (SNC)	R_TMF518_RP_II_0028, R_TMF518_RP_II_0029 This use case is a generalization of Use Case 5.6.8 from TMF 513 v3.0
UC_TMF518_RP_0009	The requesting OS deactivates and deletes a Subnetwork Connection (SNC)	R_TMF518_RP_II_0030, R_TMF518_RP_II_0031 This use case is a generalization of Use Case 5.6.9 from TMF 513 v3.0
UC_TMF518_RP_0010	target OS reroutes a Subnetwork Connection (SNC)	R_TMF518_RP_II_0005 This use case is a generalization of Use Case 5.6.10 from TMF 513 v3.0
UC_TMF518_RP_0012	The requesting OS unprovisions equipment	R_TMF518_RP_II_0036 This use case is a generalization of Use Case 5.9.1 from TMF 513 v3.0
UC_TMF518_RP_0013	The requesting OS provisions equipment	R_TMF518_RP_II_0034 This use case is a generalization of Use Case 5.9.2 from TMF 513 v3.0
UC_TMF518_RP_0014	Craft inserts a plug-in card	This use case is a generalization of Use Case 5.10.4 from TMF 513 v3.0
UC_TMF518_RP_0015	The requesting OS creates a Flow Domain	R_TMF518_RP_II_0046, R_TMF518_RP_II_0047 This use case is a generalization of Use Case 9.3.1 from TMF 513 v3.1
UC_TMF518_RP_0016	The requesting OS deletes a Flow Domain	R_TMF518_RP_II_0050 This use case is a generalization of Use Case 9.3.2 from TMF 513 v3.1
UC_TMF518_RP_0017	The requesting OS modifies a Flow Domain	R_TMF518_RP_II_0048, R_TMF518_RP_II_0049 This use case is a generalization of Use Case 9.3.3 from TMF 513 v3.1
UC_TMF518_RP_0018	The requesting OS associates Matrix Flow Domain(s) to a Flow Domain	R_TMF518_RP_II_0053 This use case is a generalization of Use Case 9.3.4 from TMF 513 v3.1

UC_TMF518_RP_0019	The requesting OS dissociates Matrix Flow Domain(s) to a Flow Domain	R_TMF518_RP_II_0054 This use case is a generalization of Use Case 9.3.5 from TMF 513 v3.1
UC_TMF518_RP_0020	The requesting OS associates CPTP(s) to a Flow Domain	R_TMF518_RP_II_0051 This use case is a generalization of Use Case 9.3.6 from TMF 513 v3.1
UC_TMF518_RP_0021	The requesting OS dissociates FD Edge CPTP(s) from a Flow Domain	R_TMF518_RP_II_0052 This use case is a generalization of Use Case 9.3.7 from TMF 513 v3.1
UC_TMF518_RP_0022	The requesting OS creates a Matrix Flow Domain (MFD)	R_TMF518_RP_II_0037, R_TMF518_RP_II_0038 This use case is a generalization of Use Case 9.4.1 from TMF 513 v3.1
UC_TMF518_RP_0023	The requesting OS deletes a Matrix Flow Domain (MFD)	R_TMF518_RP_II_0041 This use case is a generalization of Use Case 9.4.2 from TMF 513 v3.1
UC_TMF518_RP_0024	The requesting OS modifies a Matrix Flow Domain (MFD)	R_TMF518_RP_II_0039, R_TMF518_RP_II_0040 This use case is a generalization of Use Case 9.4.3 from TMF 513 v3.1
UC_TMF518_RP_0025	The requesting OS assigns CPTP(s) to a Matrix Flow Domain	R_TMF518_RP_II_0042 This use case is a generalization of Use Case 9.4.4 from TMF 513 v3.1
UC_TMF518_RP_0026	The requesting OS un-assigns CPTP(s) from a Matrix Flow Domain	R_TMF518_RP_II_0043 This use case is a generalization of Use Case 9.4.5 from TMF 513 v3.1
UC_TMF518_RP_0027	The requesting OS creates a Traffic Conditioning Profile	R_TMF518_RP_II_0055, R_TMF518_RP_II_0056 This use case is a generalization of Use Case 9.5.1 from TMF 513 v3.1
UC_TMF518_RP_0028	The requesting OS deletes a Traffic Conditioning Profile	R_TMF518_RP_II_0058 This use case is a generalization of Use Case 9.5.2 from TMF 513 v3.1

UC_TMF518_RP_0029	The requesting OS modifies a Traffic Conditioning Profile	R_TMF518_RP_II_0056, R_TMF518_RP_II_0057 This use case is a generalization of Use Case 9.5.3 from TMF 513 v3.1
UC_TMF518_RP_0030	The requesting OS configures Traffic Mapping Table	R_TMF518_RP_II_0059 This use case is a generalization of Use Case 9.5.4 from TMF 513 v3.1
UC_TMF518_RP_0031	The requesting OS creates and activates a Flow Domain Fragment	R_TMF518_RP_II_0060, R_TMF518_RP_II_0061 This use case is a generalization of Use Case 9.6.1 from TMF 513 v3.1
UC_TMF518_RP_0032	The requesting OS deactivates and deletes a Flow Domain Fragment	R_TMF518_RP_II_0064 This use case is a generalization of Use Case 9.6.2 from TMF 513 v3.1
UC_TMF518_RP_0033	The requesting OS modifies a Flow Domain Fragment	R_TMF518_RP_II_0062, R_TMF518_RP_II_0063, R_TMF518_RP_II_0065, R_TMF518_RP_II_0066 This use case is a generalization of Use Case 9.6.3 from TMF 513 v3.1
UC_TMF518_RP_0034	The NMS retrieves GUI Cut-Through window data	R_TMF518_RP_II_0070 This use case is a generalization of Use Case 5.12.1 from TMF 513 v3.0
UC_TMF518_RP_0036	Server based GCT launch (e.g. using an X-protocol)	R_TMF518_RP_II_0072 This use case is a generalization of Use Case 5.12.3 from TMF 513 v3.0
UC_TMF518_RP_0037	The requesting OS provisions the mapping mode of a CTP	R_TMF518_RP_II_0085 This use case is a generalization of Use Case 5.5.1 from TMF 513 v3.0
UC_TMF518_RP_0038	requesting OS un-maps a server layer CTP	R_TMF518_RP_II_0086 This use case is a generalization of Use Case 5.5.2 from TMF 513 v3.0

UC_TMF518_RP_0039	The requesting OS Provisions the TP Transmission Parameters	R_TMF518_RP_II_0088 This use case is a generalization of Use Case 5.5.7 from TMF 513 v3.0
UC_TMF518_RP_0040	The requesting OS creates a Floating Termination Point	R_TMF518_RP_II_0090, R_TMF518_RP_II_0091 This use case is a generalization of Use Case 9.2.1 from TMF 513 v3.0
UC_TMF518_RP_0041	The requesting OS deletes a Floating Termination Point	R_TMF518_RP_II_0092 This use case is a generalization of Use Case 9.2.2 from TMF 513 v3.0
UC_TMF518_RP_0042	The requesting OS creates a Transmission Descriptor (TMD)	R_TMF518_RP_II_0093 This use case is a generalization of Use Case 5.5.12 from TMF 513 v3.0
UC_TMF518_RP_0043	The requesting OS modifies a Transmission Descriptor (TMD) on a TP	R_TMF518_RP_II_0097 This use case is a generalization of Use Case 5.5.13 from TMF 513 v3.0
UC_TMF518_RP_0044	The requesting OS deletes a Transmission Descriptor (TMD)	R_TMF518_RP_II_0095 This use case is a generalization of Use Case 5.5.14 from TMF 513 v3.0
UC_TMF518_RP_0045	The requesting OS creates a Topological Link (TL)	R_TMF518_RP_II_0100 This use case is a generalization of Use Case 5.5.10 from TMF 513 v3.0
UC_TMF518_RP_0046	The requesting OS deletes a Topological Link (TL)	R_TMF518_RP_II_0102 This use case is a generalization of Use Case 5.5.11 from TMF 513 v3.0
UC_TMF518_RP_0047	The requesting OS creates and activates an ATM Subnetwork Connection (SNC)	R_TMF518_RP_II_0009, R_TMF518_RP_II_0010 This use case is a generalization of Use Case 5.14.1 from TMF 513 v3.0
UC_TMF518_RP_0049	The requesting OS creates a Group Termination Point	R_TMF518_RP_II_0077 This use case is a generalization of Use Case 7.5.18 from TMF 513 v3.1

UC_TMF518_RP_0050	The requesting OS modifies a Group Termination Point (GTP)	R_TMF518_RP_II_0080 This use case is a generalization of Use Case 7.5.19 from TMF 513 v3.1
UC_TMF518_RP_0051	The requesting OS deletes a Group Termination Point (GTP)	R_TMF518_RP_II_0079 This use case is a generalization of Use Case 7.5.20 from TMF 513 v3.1
UC_TMF518_RP_0052	The requesting OS creates a Termination Point Pool (TP Pool)	R_TMF518_RP_II_0081 This use case is a generalization of Use Case 7.5.21 from TMF 513 v3.1
UC_TMF518_RP_0053	requesting OS modifies a Termination Point Pool (TP Pool)	R_TMF518_RP_II_0084 This use case is a generalization of Use Case 7.5.22 from TMF 513 v3.1
UC_TMF518_RP_0054	The requesting OS deletes a Termination Point Pool (TP Pool)	R_TMF518_RP_II_0083 This use case is a generalization of Use Case 7.5.23 from TMF 513 v3.1
UC_TMF518_RP_0055	The requesting OS modifies a Transmission Descriptor (TMD)	R_TMF518_RP_II_0106 This use case is a generalization of Use Case 7.5.15 from TMF 513 v3.1
UC_TMF518_RP_0056	NMS sets the Transmission Descriptor (TMD) Profile Pointer	R_TMF518_RP_II_0044 R_TMF518_RP_II_0045 R_TMF518_RP_II_0097 R_TMF518_RP_II_0098 This use case is a generalization of Use Case 7.5.14 from TMF 513 v3.1

Table 5-2. Requirements – Use Cases Traceability Matrix

Requirement Id	Use Case Name	Use Case Id
R_TMF518_RP_I_0081		
R_TMF518_RP_II_0002	The requesting OS creates a Subnetwork Connection (SNC)	UC_TMF518_RP_0001

R_TMF518_RP_II_0003	The requesting OS creates a Subnetwork Connection (SNC)	UC_TMF518_RP_0001
R_TMF518_RP_II_0004		
R_TMF518_RP_II_0005	target OS reroutes a Subnetwork Connection (SNC) The requesting OS creates a Subnetwork Connection (SNC)	UC_TMF518_RP_0010 UC_TMF518_RP_0001
R_TMF518_RP_II_0006		
R_TMF518_RP_II_0007	The requesting OS activates a Subnetwork Connection (SNC)	UC_TMF518_RP_0002
R_TMF518_RP_II_0008	The requesting OS activates a Subnetwork Connection (SNC)	UC_TMF518_RP_0002
R_TMF518_RP_II_0009	The requesting OS creates and activates an ATM Subnetwork Connection (SNC) The requesting OS creates and activates a Subnetwork Connection (SNC)	UC_TMF518_RP_0047 UC_TMF518_RP_0003
R_TMF518_RP_II_0010	The requesting OS creates and activates an ATM Subnetwork Connection (SNC) The requesting OS creates and activates a Subnetwork Connection (SNC)	UC_TMF518_RP_0047 UC_TMF518_RP_0003
R_TMF518_RP_II_0011	The requesting OS adds a route to a Subnetwork Connection (SNC)	UC_TMF518_RP_0004
R_TMF518_RP_II_0012	The requesting OS adds a route to a Subnetwork Connection (SNC)	UC_TMF518_RP_0004
R_TMF518_RP_II_0013	The requesting OS removes a route from a Subnetwork Connection (SNC)	UC_TMF518_RP_0005
R_TMF518_RP_II_0014	The requesting OS removes a route from a Subnetwork Connection (SNC)	UC_TMF518_RP_0005
R_TMF518_RP_II_0015	The requesting OS creates-modifies the route of a Subnetwork Connection (SNC)	UC_TMF518_RP_0006
R_TMF518_RP_II_0016	The requesting OS creates-modifies the route of a Subnetwork Connection (SNC)	UC_TMF518_RP_0006
R_TMF518_RP_II_0017		

R_TMF518_RP_II_0018		
R_TMF518_RP_II_0019		
R_TMF518_RP_II_0020		
R_TMF518_RP_II_0021		
R_TMF518_RP_II_0022		
R_TMF518_RP_II_0023		
R_TMF518_RP_II_0024		
R_TMF518_RP_II_0025		
R_TMF518_RP_II_0026	The requesting OS deactivates a Subnetwork Connection (SNC)	UC_TMF518_RP_0007
R_TMF518_RP_II_0027	The requesting OS deactivates a Subnetwork Connection (SNC)	UC_TMF518_RP_0007
R_TMF518_RP_II_0028	The requesting OS deletes a Subnetwork Connection (SNC)	UC_TMF518_RP_0008
R_TMF518_RP_II_0029	The requesting OS deletes a Subnetwork Connection (SNC)	UC_TMF518_RP_0008
R_TMF518_RP_II_0030	The requesting OS deactivates and deletes a Subnetwork Connection (SNC)	UC_TMF518_RP_0009
R_TMF518_RP_II_0031	The requesting OS deactivates and deletes a Subnetwork Connection (SNC)	UC_TMF518_RP_0009
R_TMF518_RP_II_0032		
R_TMF518_RP_II_0034	The requesting OS provisions equipment	UC_TMF518_RP_0013
R_TMF518_RP_II_0035		
R_TMF518_RP_II_0036	The requesting OS unprovisions equipment	UC_TMF518_RP_0012
R_TMF518_RP_II_0037	The requesting OS creates a Matrix Flow Domain (MFD)	UC_TMF518_RP_0022
R_TMF518_RP_II_0038	The requesting OS creates a Matrix Flow Domain (MFD)	UC_TMF518_RP_0022
R_TMF518_RP_II_0039	The requesting OS modifies a Matrix Flow Domain (MFD)	UC_TMF518_RP_0024
R_TMF518_RP_II_0040	The requesting OS modifies a Matrix Flow Domain (MFD)	UC_TMF518_RP_0024
R_TMF518_RP_II_0041	The requesting OS deletes a Matrix Flow Domain (MFD)	UC_TMF518_RP_0023

R_TMF518_RP_II_0042	The requesting OS assigns CPTP(s) to a Matrix Flow Domain	UC_TMF518_RP_0025
R_TMF518_RP_II_0043	The requesting OS un-assigns CPTP(s) from a Matrix Flow Domain	UC_TMF518_RP_0026
R_TMF518_RP_II_0044	NMS sets the Transmission Descriptor (TMD) Profile Pointer	UC_TMF518_RP_0056
R_TMF518_RP_II_0045	NMS sets the Transmission Descriptor (TMD) Profile Pointer	UC_TMF518_RP_0056
R_TMF518_RP_II_0046	The requesting OS creates a Flow Domain	UC_TMF518_RP_0015
R_TMF518_RP_II_0047	The requesting OS creates a Flow Domain	UC_TMF518_RP_0015
R_TMF518_RP_II_0048	The requesting OS modifies a Flow Domain	UC_TMF518_RP_0017
R_TMF518_RP_II_0049	The requesting OS modifies a Flow Domain	UC_TMF518_RP_0017
R_TMF518_RP_II_0050	The requesting OS deletes a Flow Domain	UC_TMF518_RP_0016
R_TMF518_RP_II_0051	The requesting OS associates CPTP(s) to a Flow Domain	UC_TMF518_RP_0020
R_TMF518_RP_II_0052	The requesting OS dissociates FD Edge CPTP(s) from a Flow Domain	UC_TMF518_RP_0021
R_TMF518_RP_II_0053	The requesting OS associates Matrix Flow Domain(s) to a Flow Domain	UC_TMF518_RP_0018
R_TMF518_RP_II_0054	The requesting OS dissociates Matrix Flow Domain(s) to a Flow Domain	UC_TMF518_RP_0019
R_TMF518_RP_II_0055	The requesting OS creates a Traffic Conditioning Profile	UC_TMF518_RP_0027
R_TMF518_RP_II_0056	The requesting OS modifies a Traffic Conditioning Profile The requesting OS creates a Traffic Conditioning Profile	UC_TMF518_RP_0029 UC_TMF518_RP_0027
R_TMF518_RP_II_0057	The requesting OS modifies a Traffic Conditioning Profile	UC_TMF518_RP_0029
R_TMF518_RP_II_0058	The requesting OS deletes a Traffic Conditioning Profile	UC_TMF518_RP_0028
R_TMF518_RP_II_0059	The requesting OS configures Traffic Mapping Table	UC_TMF518_RP_0030

R_TMF518_RP_II_0060	The requesting OS creates and activates a Flow Domain Fragment	UC_TMF518_RP_0031
R_TMF518_RP_II_0061	The requesting OS creates and activates a Flow Domain Fragment	UC_TMF518_RP_0031
R_TMF518_RP_II_0062	The requesting OS modifies a Flow Domain Fragment	UC_TMF518_RP_0033
R_TMF518_RP_II_0063	The requesting OS modifies a Flow Domain Fragment	UC_TMF518_RP_0033
R_TMF518_RP_II_0064	The requesting OS deactivates and deletes a Flow Domain Fragment	UC_TMF518_RP_0032
R_TMF518_RP_II_0065	The requesting OS modifies a Flow Domain Fragment	UC_TMF518_RP_0033
R_TMF518_RP_II_0066	The requesting OS modifies a Flow Domain Fragment	UC_TMF518_RP_0033
R_TMF518_RP_II_0067		
R_TMF518_RP_II_0068		
R_TMF518_RP_II_0069		
R_TMF518_RP_II_0070	The NMS retrieves GUI Cut-Through window data	UC_TMF518_RP_0034
R_TMF518_RP_II_0072	Server based GCT launch (e.g. using an X-protocol)	UC_TMF518_RP_0036
R_TMF518_RP_II_0073		
R_TMF518_RP_II_0074		
R_TMF518_RP_II_0075		
R_TMF518_RP_II_0076		
R_TMF518_RP_II_0077	The requesting OS creates a Group Termination Point	UC_TMF518_RP_0049
R_TMF518_RP_II_0078		
R_TMF518_RP_II_0079	The requesting OS deletes a Group Termination Point (GTP)	UC_TMF518_RP_0051
R_TMF518_RP_II_0080	The requesting OS modifies a Group Termination Point (GTP)	UC_TMF518_RP_0050
R_TMF518_RP_II_0081	The requesting OS creates a Termination Point Pool (TP Pool)	UC_TMF518_RP_0052
R_TMF518_RP_II_0082		
R_TMF518_RP_II_0083	The requesting OS deletes a Termination Point Pool (TP Pool)	UC_TMF518_RP_0054

R_TMF518_RP_II_0084	requesting OS modifies a Termination Point Pool (TP Pool)	UC_TMF518_RP_0053
R_TMF518_RP_II_0085	The requesting OS provisions the mapping mode of a CTP	UC_TMF518_RP_0037
R_TMF518_RP_II_0086	requesting OS un-maps a server layer CTP	UC_TMF518_RP_0038
R_TMF518_RP_II_0088	The requesting OS Provisions the TP Transmission Parameters	UC_TMF518_RP_0039
R_TMF518_RP_II_0089		
R_TMF518_RP_II_0090	The requesting OS creates a Floating Termination Point	UC_TMF518_RP_0040
R_TMF518_RP_II_0091	The requesting OS creates a Floating Termination Point	UC_TMF518_RP_0040
R_TMF518_RP_II_0092	The requesting OS deletes a Floating Termination Point	UC_TMF518_RP_0041
R_TMF518_RP_II_0093	The requesting OS creates a Transmission Descriptor (TMD)	UC_TMF518_RP_0042
R_TMF518_RP_II_0094		
R_TMF518_RP_II_0095	The requesting OS deletes a Transmission Descriptor (TMD)	UC_TMF518_RP_0044
R_TMF518_RP_II_0097	NMS sets the Transmission Descriptor (TMD) Profile Pointer The requesting OS modifies a Transmission Descriptor (TMD) on a TP	UC_TMF518_RP_0056 UC_TMF518_RP_0043
R_TMF518_RP_II_0098	NMS sets the Transmission Descriptor (TMD) Profile Pointer	UC_TMF518_RP_0056
R_TMF518_RP_II_0099		
R_TMF518_RP_II_0100	The requesting OS creates a Topological Link (TL)	UC_TMF518_RP_0045
R_TMF518_RP_II_0101		
R_TMF518_RP_II_0102	The requesting OS deletes a Topological Link (TL)	UC_TMF518_RP_0046
R_TMF518_RP_II_0103		
R_TMF518_RP_II_0106	The requesting OS modifies a Transmission Descriptor (TMD)	UC_TMF518_RP_0055

6 Future Directions

7 References

7.1 References

[1] TMF518_FMW, Framework DDP BA

[2] TMF513, Multi-Technology Network Management (MTNM) Business Agreement, Version 3.1, March 2007

[3] TMF517, Multi-Technology Operations System Interface (MTOSI) Business Agreement, Version 1.2, December 2006

[4] SD1-16, Layered Parameters

[5] SD1-36, SNC and Protection

[6] SD0-1, Dictionary

7.2 Source or use

The various sources for the requirements in this document are listed in the “Source” field of each requirement.

7.3 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 resource provisioning. 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 TMF612_RP 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	September 2007	This is the first version of the document and as such, there are no changes to report.
1.1	May 2008	Updated based on review and consolidation comments for the preparation of the MTOSI 2.0 release.

1.2	September 2011	Updated sections 1.1 and 2. Replaced mTOP by MTNM / MTOSI everywhere in the document

8.4 Company Contact Details

Document Contact	
Name:	Michel Besson
Company:	Cramer > Amdocs OSS division
Email:	MichelBesson@Amdocs.com
Phone:	+44 (7717) 692178

8.5 Acknowledgments

This document was prepared by the members of the TM Forum MTNM / MTOSI RM team.

- Keith Dorking, Ciena Corporation, document editor
- Steve Fratini, Telcordia Technologies, MTNM / MTOSI Program Director
- Bernd Zeuner, Deutsche Telekom AG
- Jérôme Magnet, Ciena