TM Forum How To Guide

ODA Production Implementation Guidelines

GB999

Team Approved Date: 03-Apr-2020

Release Status: Production	Approval Status: TM Forum Approved
Version 4.0.1	IPR Mode: RAND

Notice

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100

Parsippany, NJ 07054 USA

Tel No. +1 973 944 5100

Fax No. +1 973 998 7916

TM Forum Web Page: www.TM Forum.org

Table of Contents

List of Figures

Executive Summary

This Guideline provides a deeper implementation focused analysis of the ODA Production Function Grouping within the ODA Functional Architecture [1]. It defines the detailed logical functional architecture, and implementation best practice for realizing these functions as ODA Components on integration fabrics used to support hybrid networks, comprising both physical appliances and virtualized applications.

Automated Lifecycle management of ODA implementation components is integrated with the Onboarding Automation defined in IG1141 Procurement and Onboarding Suite [2] and IG1176 TOSCA Guide for Model-Driven Automation [3].

These guidelines have been validated with the results of six 5G based catalysts and two 'Network as a Service' catalysts exhibited at:

1. Digital Transformation World (DTW) Nice May 2018.
2. Digital Transformation Asia KL, November 2018.
3. Digital Transformation World (DTW) Nice May 2019 .

Based on an analysis of these catalysts, this Implementation Framework addresses:

1. Realization of Zero-Touch Management Automation (using established concepts from ZOOM Project and Open APIs).
2. Alignment with Automated Lifecycle Model for network and other resource onboarding described in IG1176 TOSCA Guide for Model-Driven Automation [3] and IG1141 Procurement and Onboarding Suite [2].
3. A Baseline of the preferred implementation framework for the Open Digital Architecture (ODA) Production Functional Grouping.
4. Documentation of the Use Cases used in the Catalysts.
5. A Connectivity Service technology neutral model fulfilling the ODA requirement to decouple ODA Production functions (NMS, OSS) from ODA Core Commerce and other BSS functions.
6. Connectivity Service Definitions - aligned with GB922 Information Framework models and ONF TR512 Core Information Model - for use with the NaaS API Component Suite TMF909 [4].
7. A Framework facilitating adoption of supplier implementations using techniques such as: intent based management, closed control loops, policy management and Artificial Intelligence/Machine Learning.

The Implementation Framework described in this report provides a systematic approach for:

1. Linking Communication Service Provider (CSP) defined Operational Domain boundaries, and the 'Network as a Service' APIs / capabilities that each expose as Customer Facing Services (CFS)[1].
2. Mapping Operational Domains to supplier defined Technical Domain service boundaries for multiple technologies. (Resource Facing Services RFS, Resource Functions and Resources).

3. Using the mature TM Forum GB922 Information Framework Models for Services, Resources and Products.

This snapshot addresses most of the current ODA and ODA Production requirements through this Implementation focused Framework model.

The principle additions in this version have been to address the issues identified in the previous release GB999 R19.5.1 'Appendix B: ODA Production related JIRA Change Requests' and make consequential adjustment to the models in this version, and the addition of a formal UML model for connectivity services based on GB922 Information Framework that is extensible to multiple transport technologies e.g. 5G, SD-WAN through the ONF TR 512 models. Examples used are focused on 5G work in the Catalysts.

The team now wishes to solicit wider member review and feedback as it moves to the next stage of developing these models into more detailed specifications for:

- Validating the Core Connectivity Service Models based on the TM Forum Information Framework.
- Transformation of the Connectivity Service models into JSON for TM Forum API that are aligned with published GB922 Logical and Compound Resource Computing and Software Resource [28] now incorporated into GB922 Resource Domain Business Entities [50], TMF 664 Resource Function Activation and Configuration API R19-5 [45] , the TR255 Suite of documents and Formal Change requests to SID models.
- Extension rules for supporting other network technologies within ODA Production.

Whilst the document is evolving, the ODA team thinks wider member input would be valuable, and specifically the team is soliciting additional JSON and UML modelling expertise to complete the next Release 20.0 version.

---

[1] Differences and need for Customer Facing Service and Resource Facing Services are described in sections 4.1 & 5.5.1. See also discussion of use of Resource Function concept described in Reference [28]

Readers Guide

Document structure

ODA Production Implementation Guidelines is re-structured in this release into four main Parts each aimed at a specific set of audience needs and distinct topics:

- **Part I: ODA and ODA Production Definition**
  Audiences: Product Managers, Architects and Developers.
- **Part II: ODA Production Implementation Framework**
  Audiences: Architects and Developers.
- **Part III: ODA Production Modelling Principles**
  Audiences: Architects and Developers.
- **Part IV: ODA Production Implementation Exemplars**
  Audiences: Product Managers, Architects and Developers.
- **Appendices:**
  Appendix A: Network Slice Management Requirements
  Appendix B: ODA Production related JIRA Change Requests
  Appendix C: ODA Production GB922 Information Framework Model (new)

Audience topics

Product Managers

- Part I: Gives the Product Manager an overview of the main concepts and principles in implementing ODA Production.
- Part III: Defines a Connectivity Service model that ODA Core Commerce can consume without knowing the details of the network technologies used to provide it, including those defined by other Organizations. An example of the connectivity requirements they are addressing is captured in TR276 Monetizing 5G [18] which has been assembled from member product manager inputs.
  Examples used are based on 5G Catalysts.
- Part IV: Provides examples taken from catalysts of how the ODA Production Implementation Framework is related to practical implementations.

Architects and Developers

- Part I: Provides the overall rational for the ODA Production concepts and principles.
- Part II: Provides the technical overview of the ODA Production Implementation Framework.
- Part III: Provides the detailed APIs and Connectivity Models needed to realize the ODA Production Implementation Framework.
- Part IV: Provides the 5G examples and results of the 5G and other catalysts.
- Appendices: provide technical details of the Network Slicing requirements and a draft of the Connectivity Service representation in GB922 Information Framework models.

It is recognized that:

- 5G implementation architecture will evolve as 3GPP 5G specifications have yet to be finalized at the publication date of this Guidebook. This will affect the specifics of mappings from the NaaS Connectivity Service defined here to 3GPP Network Slice model and the use of GSMA Generic Slice Template [19].
- Also, other Standards Defining Organization (SDO) will evolve their specifications and the ODA Production approach decouple these changes from operation of ODA Core Commerce in particular.

Legacy solutions may only ever realize a subset of the ODA Production Implementation Framework e.g. the NaaS APIs and Connectivity Service models developed in this document.

Part I: ODA and ODA Production Definition

1. Introduction

1.1. The Open Digital Architecture Framework

The Open Digital Architecture (ODA) described in:

- Open Digital Architecture White Paper [5].
- IG1166 ODA Architecture Vision [6].
- IG1167 ODA Functional Architecture [1]
- GB998 Open Digital Architecture (ODA) Concepts & Principles [7].

ODA sets out as one of its key principles the need to decouple different areas of the Functional Architecture and applications implementing them (GB998 Principle 3.04).

In realizing ODA, each functional area publishes the services (collection of functions) that it exposes in a Catalog (which also provides the information necessary to use the services exposed through Open APIs). This is intended to be part of a progressive move towards a fully model driven approach to realizing the lifecycle management of ODA.

1.2. ODA Production and Network as a Service (NaaS)

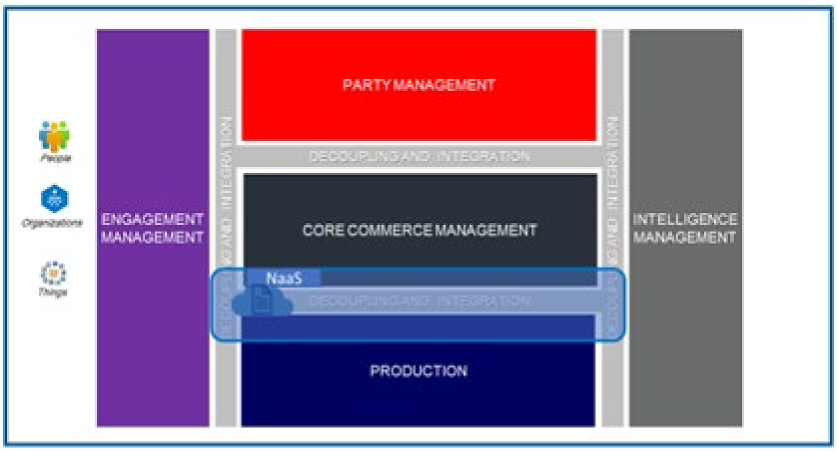


Figure 1.2.1: ODA Functional Architecture - High Level view (Level 0)

The benefits of decoupling the lifecycle of Production Functions from the lifecycle of Core Commerce Function are improved business agility in:

- Introducing new product offering by Core Commerce by reusing exposed Production services from ODA Production which change less frequently. (Production Services are also referred to as Network as a Service - NaaS, exposed via NaaS API Component Suite [4] plus service models)
- Introducing new and evolving technology solutions and resources to realize ODA Production without necessarily changing the exposed Production Services.
  i.e. decouple technology lifecycle changes from ODA Production Service lifecycle
- The ability to quickly create Products by assembling multiple reusable NaaS (exposed services) at either the OSS application level or from within a network operational/technical domain saves project integration time and costs.

For communication services the concept of Network as a Service (NaaS), shown above, has emerged as a critical Service Provider requirement for decoupling ODA Production from other ODA functional groups. NaaS is interpreted broadly covering both connectivity services - this document's primary focus - and end point services like applications.

Using NaaS API component suite to implement the services exposed by ODA Production enables:

- Decoupling the implementation detail of ODA Production (the 'How ') from the exposed Production Services by use of Intent Based Production Service models. These Production services are published in a catalog so that any changes are propagated automatically supporting a model driven approach.
- Removing the time-consuming, inflexible and complex IT system integration costs of introducing new network technology particularly where these are based on detailed resource hierarchy models.
  (see TMF909 NaaS Component Suite [4] and IG1170 Network as a Service (NaaS) Package [8]).
- Managing within ODA Production the lifecycle of all service and resource changes.
- Moving to model driven zero friction integration of the components used to realize ODA Production, and their lifecycle management see IG1176 [3] TR262 [17], IG1141 [2].

Additionally, A generalized (abstract) connectivity / connection model is needed for ODA Production that is substantially free of any specific technology implementation detail, thus allowing these services to be realized with multiple and evolving choices of network and technology domains. However, noting that connectivity Connection termination Points have characteristics dependent on the protocol level and technology used for the specific service e.g. Mobile, Carrier Ethernet Level 1 / Level 2 protocols.

1.3. ODA Production

ODA Production is a NaaS from an abstraction perspective. From the background of business and technology trends, the implementation examples of ODA Production are as follows:

1. Cloud Native NaaS
2. 5G NW Slice Service
3. Hybrid Connectivity Service

1.4. Business Challenge

The objective of this Implementation Guide is to achieve ODA decoupling by separating Customer Facing Services (CFS) exposed by ODA Production from their implementation details.

1.4.1. ODA Production Challenge

Communication Service Providers (CSP) need to integrate multiple technologies and manage them as services offered as products to their customers. Depending on the CSP organization, there is typically little reuse of network services and in the case of older technologies (e.g. transmission) management may still be done manually due to complexity and/or lower volume. Consequently, when a CSP creates a new product, it becomes a new OSS/IT project to integrate technology services together. Similarly, changing a product that includes a physical component to be a virtual component will trigger a new OSS/IT project and most likely the creation of a new product.

To achieve the needed agility in the 5G ecosystem, and to take advantage of newer cloud and virtualization technologies, CSPs need a simpler approach to integration decoupling via APIs and NaaS exposed services (the 'What') from 'how' ODA Production functions realize these services. ODA Production also performs the lifecycle management of the NaaS exposed to the OSS/IT systems in the Core Commerce Management functions.

.

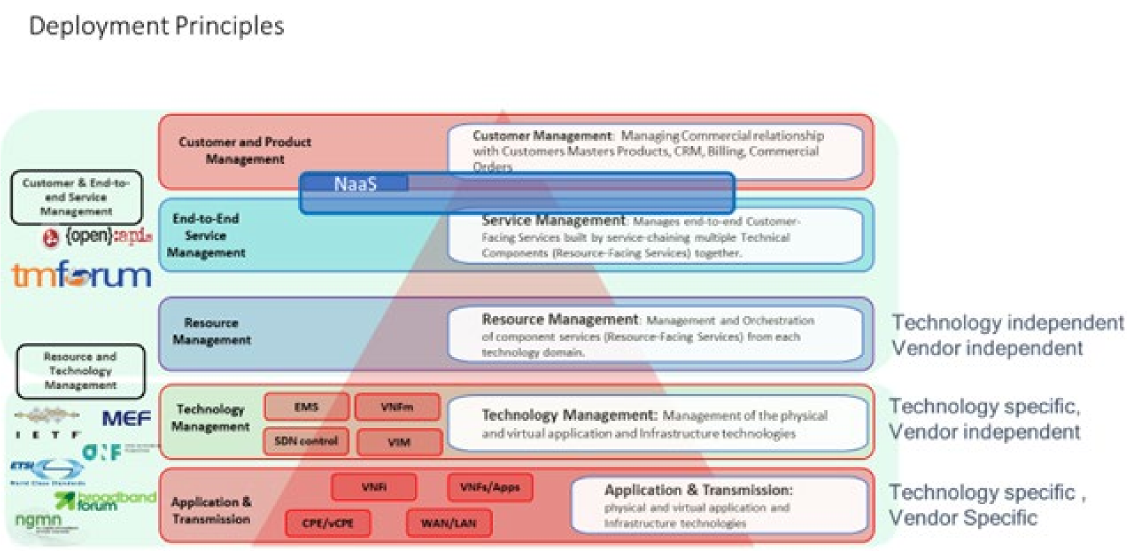


Figure 1.4.1. Application of Network as a Service

The figure above derived from TR262 [17] and Vodafone practices shows that the critical feature is End to End Service Management (which provides Customer Facing Services) to Customer and Product Management. In an ODA context this is the primarily focus for the Network as a Service (NaaS) solution exposed by ODA Production.

- Service Management composes both CFS and Resource Facing Services (RFS) to provide the services exposed through NaaS and ultimately the definition of an E2E Service delivered to customers.
- Resource Management composes components into:

- Resource Facing Services from multiple Technology Domains. An ODA Production feature of Resource Facing Services is that they are technology specific but vendor independent.
- Supporting Resource Functions (RF) which permit abstraction of Resources into functions and characteristics which is needed for virtualization e.g. NFV and cloud solutions. A particular advantage is that RF can be composed into Products or Services (CFS and RFS) These concepts are discussed more fully in Sections 4.1 and 5.5.1 and Appendix C and are derived from TR 255 [13] [14] [15] and implemented in TMF 664 Resource Function Activation and Configuration API.

- Technology Management manages individual technology domains using management interfaces that are technology specific and may be vendor specific using interface technologies selected and defined by multiple resource and technology management Standards Defining Organizations (SDO).

This model provides a set of requirements for abstracting / generalizing the management of Applications and transmission technology into the services offered within product. That is going from technology domains with a diverse set of management interfaces and technologies, to services offered using NaaS with consistent TM Forum Open APIs and Connectivity Service models.

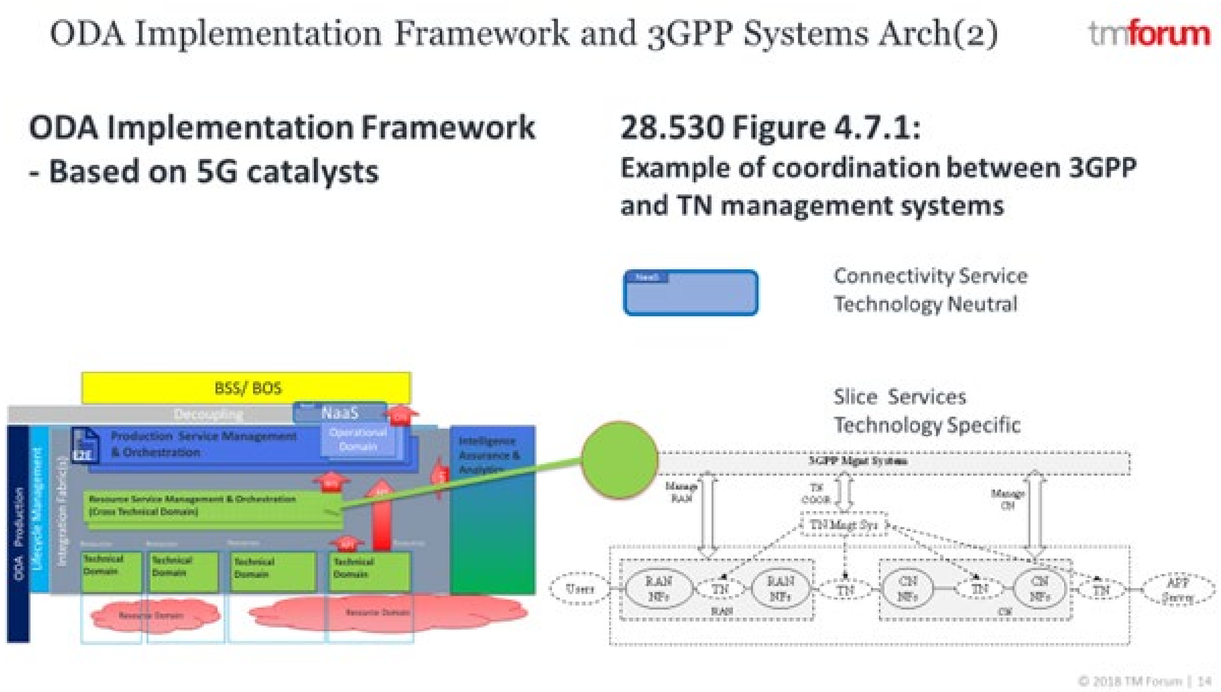
ODA Production covers the bottom four layers of Figure 1.4.1.

The design of the Service APIs (TMF641 TMF640) makes them useable to expose RFS services from the Resource Service Management & Orchestration function block provided the connectivity service model is designed to support both technology neutral (CFS) and also technology specific (RFS) extensions.

2. Terminology

Term	Definition / Source
Connectivity Service	A communication capability that is offered and allows connections to be established between two or more Service Access Points with associated transfer, storage and processing functions. Source: GB999 *Note: a Communication Service (more generalized term) may include either or both a connectivity service and an endpoint service.*
Connectivity Service Domain	A Domain which exposes a Connectivity Service as a Customer Facing Service (CFS) or as a Resource Facing Service (RFS) and publishes the capability in a Catalog accessible by potential clients / tenant. It may be the complete capability required for a customer or a part of an end to end service. *(This accommodates the fact that network domains have different roles - access /core – and may be divided geographically across multiple CSPs*.) Source: GB999
Flow/Connection	An instance of a Connectivity Service established for a specific client as a tenant of the Connectivity Service. Source: GB999 and TR255 Suite.
Management Domain	A ManagementDomain class represents a special grouping of ManagedEntities that has two important properties. First, it is used to partition managed objects into a meaningful logical grouping. One important use of such a grouping is to provide a means to define which EMS (as well as which NMS) manages, monitors, etc. which set of devices. It also provides a means to show how management functions are distributed and scaled. Second, it defines a common administrative domain that is used to administer the managed objects that it contains. This implies that all the managed objects contained in this Management Domain are administered similarly - either by the same user, group of users or policy. Source SID Root Entities, TR275.
Operational Domain	A domain whose boundary is set by a CSP based on organization assets processes and information. A type of Management Domain. May be associated with one or more Communication Service Domains. Source GB999.

Part II: ODA Production Implementation Framework



3. ODA Production Overview

This section provides an overview of the main concepts in realizing ODA Production and is suitable for Product Managers, Architects and Developers requiring a high-level overview.

3.1. ODA and ODA Production Implementation Framework

The following diagrams describe the relationship between TM Forum Open Digital Architecture (ODA) work and ODA Production showing how it is related to the 5G catalysts and 3GPP 5G systems architectures.

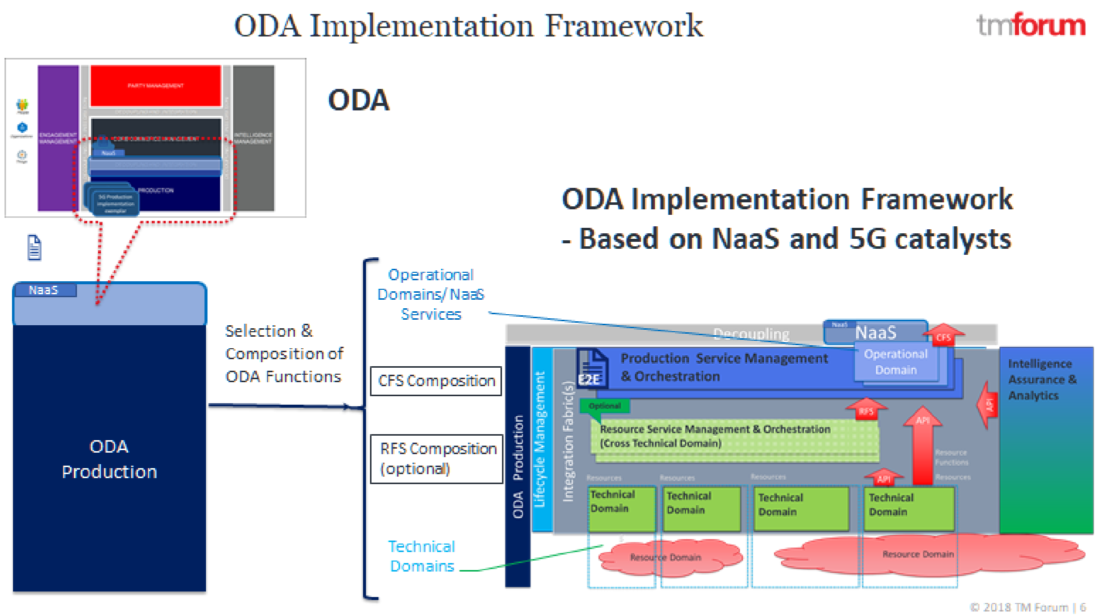


Figure 3.1.1: Relationship between ODA and ODA Production Architecture

On the upper left of the above diagram the main ODA Function Groupings are shown:

- Engagement Management (people, organizations).
- Party Management.
- Core Commerce Management (technology / production independent BSS functions. i.e. handling commercial processes and has no detailed knowledge of production technologies).
- Production which includes traditional Service and Resource Management Functions.
- Intelligence Management that coordinates intelligence knowledge across all the ODA Functional blocks using a set of patterns including closed control loop, policy and AI/ML. See IG1177 ODA Intelligence Management Implementation Guide [20].

On the lower left side of the above diagram:

- The ODA Production Function Group exposes its capabilities including Connectivity Services enabled by TM Forum Open APIs (REST) including ‘Network as a Service’ (NaaS) [4].
- These exposed services /capabilities correspond to Customer Facing Services (CFS) in the TM Forum Information Model (aka SID), a stable concept for over 20 years.
- ODA Production has a principle that Production should be decoupled from Core Commerce which specifically means that the services exposed by Production should be resource/network technology neutral.

Connectivity Service models defined in Section 5.5.1 for these NaaS services, are:

- Simple, demanding as little information as possible while being highly reusable from end users of Core Commerce products,
- Resource technology neutral to the maximum extent.

On the right side of Figure 3.1.1 is shown the current ODA Production Functional Framework described in the next sections.

3.2. ODA Production Functional Framework Overview

The purpose of this framework is to model a set of functions (based on the ODA Functional Architecture) which can be selected to compose ODA Production Solutions. It is not mandatory to include all the functions in a specific implementation but the minimum Viable Product (MVP) is to support the ODA NaaS Service (APIs Plus Connectivity Models)[1]. Having an agreed set of Framework functions drives up the consistency of ODA Production Implementations.

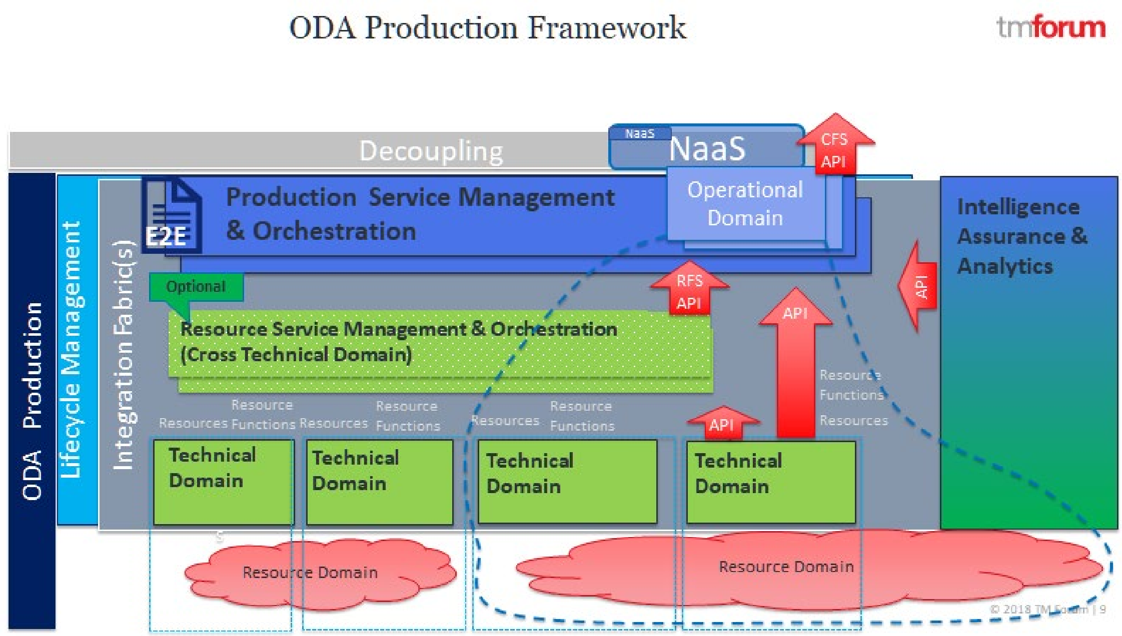


Figure 3.2.1: 5G use of ODA Production Architecture/ Implementation Framework

These components have been distilled out of the implementation patterns from six 5G catalyst PoC, catalysts looking at autonomic computing and closed control loop techniques, two AI catalysts and the NaaS in Action catalysts.

- The Framework is based on exposed capabilities/ API services shown as Red Arrows from each Functional Block. Services from the Function Blocks are exposed and consumed through the Integration Fabrics (Dark Grey Box). There may be multiple integration fabrics depending on the legacy situation e.g. NFV NFVI, Netconf/Yang, etc.
- ODA Production Functions are lifecycle managed by the lifecycle management capabilities (Blue box) using a lifecycle model that is integrated with the prior work on Onboarding Automation [2], and concepts of ‘software’ factories in Monetizing 5G [18].
- Production Service Management & Orchestration realize the exposed Production CFS Services through the Decoupling layer, principally through NaaS using Intent Based Service Models, and can reuse/ compose other CFS services (iterative composition pattern). The e2e Service View is provided by the composite CFS. Using composition of CFSs has advantages when incorporating legacy physical networks, especially in those situations where there are no standard Resource APIs.

- The exposed capabilities / CFS services realized by NaaS are associated with operator or CSP determined Operational Domains (setting operational and governance boundaries across a set of function instances spanning Multiple Function Blocks).
  In the diagram we show the Operational Domains bound to NaaS services and CFS Functions which captures what and how they exposed their services.
  However, strictly the realization of an Operational Domain will also incorporate functions from Resource Service Management Functions and Technical Domain Function Blocks within its operational scope and governance. To illustrate this we have shown using a dotted line an example scope for the governance of an Operational Domain
  The exact boundary and scope of the Operational Domains is an individual operator design and operational decision, and is best illustrated by using examples of the framework above as described in catalyst examples in Part IV: ODA Production Implementation Reference.
- At the bottom are the Technical Domains whose scope and functionality are defined by suppliers and/or groups such as 3GPP and may use many interfaces/API technologies.
- In the center the optional Resource Service Management and Orchestration expose Resource Facing Services (RFS), and optionally compositions of other RFS, that pragmatically map and integrate what is provided by suppliers (Technical Domains) into the Operational Domain requirements of CSPs.
  The requirement for these function block arises where ISV or CSP have existing Multi-technology Domain orchestration tools, or Vendors supply Multi-technology managers / orchestrators, as has been seen in several of the catalyst implementation inputs to this Framework. If this function block is not needed for a particular integration, then it can be omitted and the Technical Domains APIs (upright red Arrow RHS) can be directly integrated by the Production Service Management & Orchestration functions.
- RFS components typically do some of the following:
  - Expose technology specific services.
  - Map between the native protocols and models used to expose management of resources e.g. Netconf, YANG, SNMP, etc. and the preferred TM Forum Rest APIs.
  - Provide some integrated approach to provisioning and assuring RFSs that span multiple technical domains e.g. slices across RAN, Core and transport.
  - They may be operator, SP, ISV or supplier provided.
- All APIs may support the use of Information Framework concept of Resource Functions which abstract resources and leads to several implementation advantages [14]
- Intelligence and Assurance and Analytics plays a key role in managing Closed control Loops by providing analysis of monitoring results against the intent based service target Service Level Agreements (SLA). See IG1177 [20] for a complete architectural description.

3.2.1. ODA Production Lifecycle

The ODA Production Functions are transformed into an implementation that composes and packages these functions into components that expose NaaS services.

ODA Production Implementation Packaging can be done in multiple ways which reflects legacy, operational and technical decisions of individual CSPs. Hence, the ODA Production Implementation Framework need to cover many deployment options and may follow multiple implementation standards e.g. Kubernetes and ODA Components [21] [22], when deployed.

However, ODA Production must also support the composition of current physical network appliances and virtualized networks and applications. i.e. Hybrid Network operations which requires a common approach for physical and virtualized approaches.

Because there are a rich set of options with both physical and virtualized solutions for ODA Production the framework uses concepts from the ODA Vision Lifecycle model [6] that separate functional, implementation, and deployment concerns in a structured way.

Figure 3.2.2: ODA Production separation of Functional and orthogonal Lifecycle and Infrastructure aspects

This shows that lifecycle management and Integration Fabrics are orthogonal concerns that apply to the packaging of all ODA Production Functions, and ideally supporting any component model realizing ODA functions.

This is related to the ODA vision lifecycle [6] shown below:

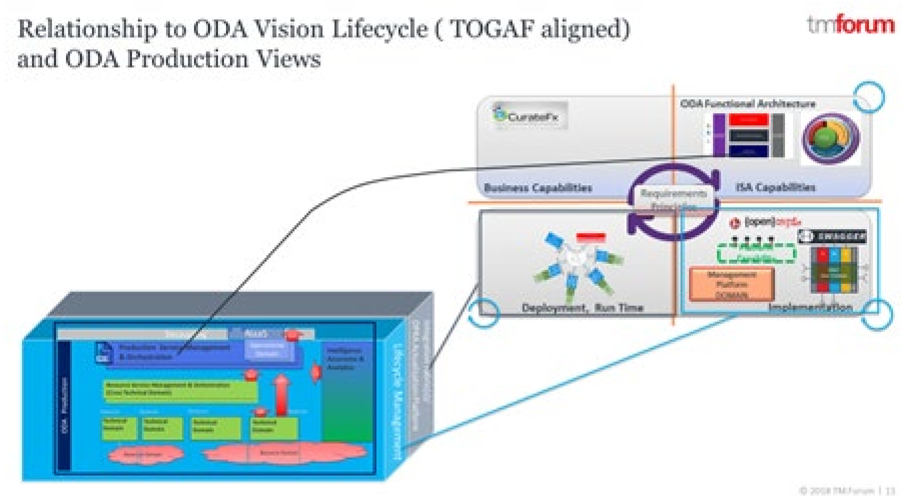


Figure 3.2.3: ODA Production Implementation and ODA vision lifecycle

In the ODA vision there are four viewpoints:

1. The Business Architecture which defines business capabilities that are reusable in multiple business and ecosystem deployment.
2. The Information Systems Architecture which defines IT reusable capabilities / functions that are implementation independent. This is where the ODA Functional architecture resides including ODA Production.
3. The Technical Architecture / Implementation viewpoint which groups ODA functions into reusable components that expose ODA Production function and are well enabled to allow lifecycle management
4. Deployment / runtime which is where the components are run e.g. Kubernetes, cloud, physical appliances.

This guideline covers the final three aspects with a mapping that is shown in Figure 3.2.3

3.2.2. ODA Lifecycle Management

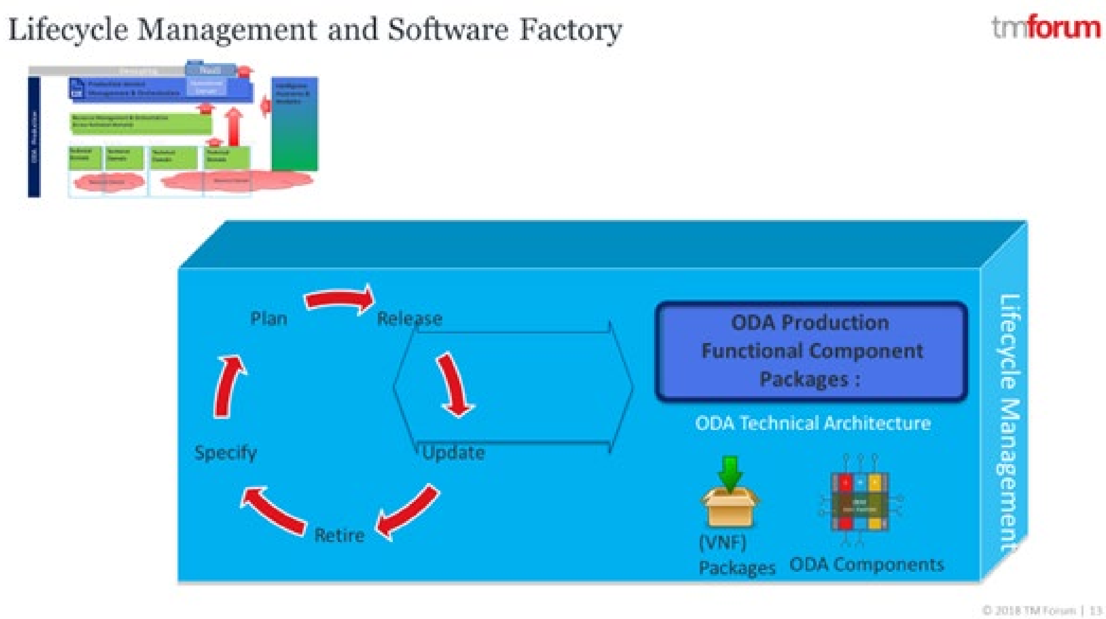


Figure 3.2.4: ODA Production Lifecycle Management

Lifecycle management covers:

- The ODA Function definitions themselves.
- The components that package groups of ODA Functions: examples being ODA Components [21] and VNF / NFV Packages [23].
- Supports Continuous Development /Continuous Integration (CD/CI) lifecycle model stages: Specify, Plan, Release, Update, Retire.
- Automated onboarding of components which may be virtualized or physical using concepts in IG1141 Procurement and Onboarding Suite [2] and IG1176 TOSCA Guide for Model-Driven Automation [3].

- Integration of Components into NaaS Connectivity services published in CFS catalog exposed by ODA Production with associated configuration metadata defining automated model driven automation of the implementation e.g. fulfillment, assurance and billing.

3.2.3. ODA Production Integration Fabrics

The run time environments for ODA Production are diverse including the physical appliances, traditional servers and cloud PaaS/IaaS virtualization environment such as virtual machine, and containers. This reflects the need to support hybrid physical and virtualized network deployments such as are encountered with NFV. Examples are shown below:

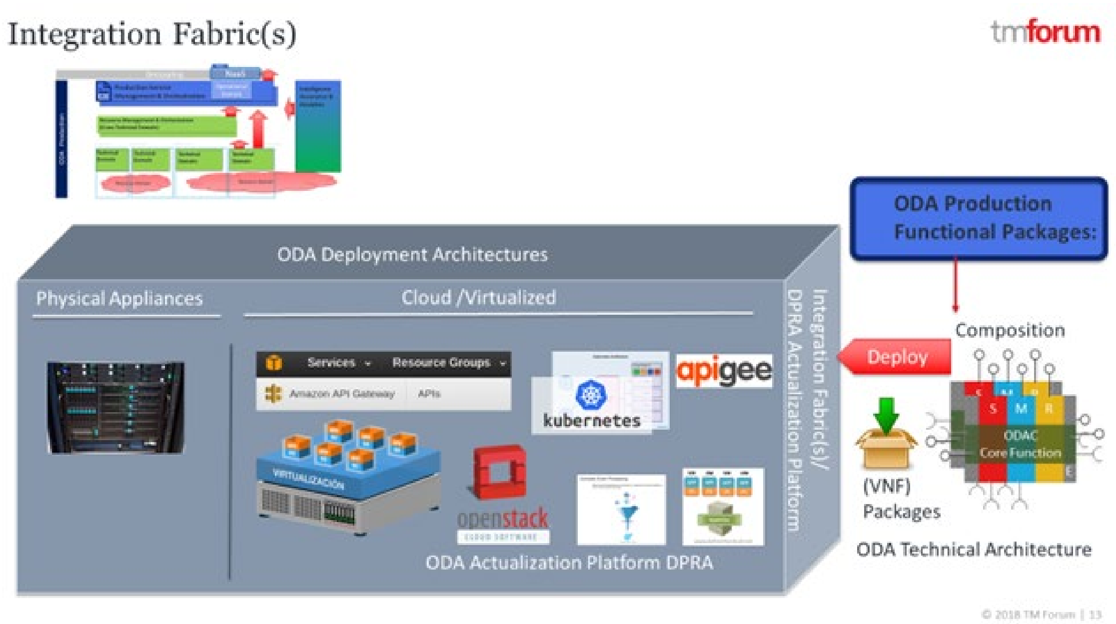


Figure 3.2.5: ODA Production Deployment Options

The deployment viewpoint covers:

- Cloud related solutions
- Repositories of software images, e.g. docker registries.
- Virtual Machines, containers and associated tools.
- Configuration metadata for:
  - the automated deployment of software images (referred to as lifecycle management in ETSI NFV, and technical onboarding in IG1141 [2])
  - Operational management of running components applications: e.g. K8s Pods/Clusters, virtual machines such as data streaming, performance metrics, run time policies, usage metrics and Security monitoring.
- Physical Appliances.
- In any typical deployment there may be multiple integration fabrics as NEP and ISV may make different choices over time, as may CSPs. This is another reason for decoupling the

ODA Production Integration Fabrics from the Functional view as they can have different life-cycles.

Additional detailed deployment examples have been developed and recorded in IG1167 ODA Functional Architecture [1] Section 4.5.9 Deployment and Scenario examples.

Formal modeling of Logical Resources defined in the Information Framework, and their relationship to ETSI NFV virtualization concepts is detailed in GB922 Resource Domain Business Entities Section 3.2.23 Mapping ETSI NFV Model to SID' [50].

3.3. 5G enabled Services

In a 5G context, ODA Production defines generalized Connectivity Services for 5G Enabled services for NaaS and assumes the use of intent based management so that the detailed implementation of these services based on 3GPP is not directly exposed in the Service definition. This is a realization of the concepts advanced in:

- Network as a Service see IG1170 Network as a Service (NaaS) Package [8].
- TMF909 NaaS API Component Suite [4].
- TR262 Management Platform Blueprint and Application to Hybrid Infrastructure [17].
- TR263F Relationships Between the Concepts of Domain and Platform [24].
- G1194 Focus on Services Not Slices R19.5. Derived from the 5G Rider on the Storm Catalysts, describes how Generic Slice Template Attributes (defined by GSMA) can be organized into Resource Functions that support intent based connectivity services and that can be mapped by profiles into detailed Resource Function characteristics (also based on GSMA definitions) at the Resource level based on 3GPP standards or RFS.
- Developments of TMF 664 Resource Function Activation and Configuration [36] now include the 3GPP 28.541 Network Resource Model (NRM). and is well suited for orchestration in both ODA Production Service Management and Resource Management function blocks when managing Resource Functions.

3.3.1. ODA Production support for 5G

On the diagram below, the common case of 5G implementation is shown:

- The Operational Domains at the Service Management & Orchestration layer may include at least Mobile data connect services and often other applicative services such as IoT services or Video services.
- The E2E Resource layer includes end-to-end slice orchestration functions.
- The Technical Domains include Core, Access and Transport together with some 4G & 3G components in a hybrid environment. Additional technical domains that correspond to the operational domains may be there. In other deployment examples they can be provided by partners that are digital operators.

The Figure 3.3.1 captures the relationship between ODA Production Implementation Framework and 5G Management Systems.

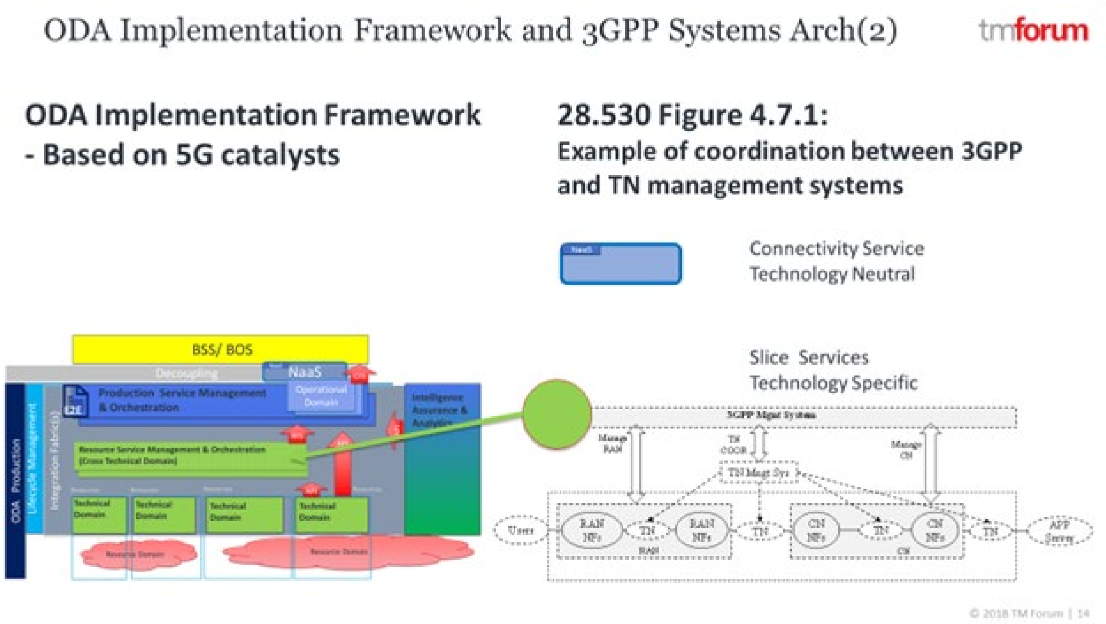


Figure 3.3.1: ODA Production Implementation Framework relationship with 5G Management Systems

The 3GPP slice management functions correspond to RFS framework components in ODA Production shown by the green circle and arrow.

These exposed Network Slice and Slice instance capabilities are technology specific which are then generalized by the CFS into technology neutral connectivity/ communications services that are exposed into the core commerce grouping (Communication Service Customer CSC in 3G terminology).

Note: 3GPP Management Systems shown in on the right-hand side Figure 3.3.1 are intended by 3GPP in 28.530 to compose Transport (TN) AN and Core Network to provide an overall managed view which corresponds to the RFS pattern.

4. ODA Production Architecture Principles

Aimed primarily at Architects and Developers this guideline provides detailed implementation principles for ODA Production of which some apply to other ODA Functional Groups.

The previous section has a holistic overview of the main aspects of the ODA Production Implementation Framework. This framework can be realized in several different ways and must accommodate many CSP choices:

- CSP choices for Operational Domain boundaries.
- Precise CSP choices of Connectivity Services exposed NaaS.
- Choice of which Operational Domain is responsible for each NaaS Connectivity Service. e.g. Access, Core, VPN, etc.
- Legacy networks and the CSPs vendor choices for Technical Domains.

This diversity requires a framework, and principles which set out the best practice, that can be 'localized' to accommodate individual CSP choices rather than a prescriptive rigid and static architecture. However, these principles need to ensure interoperability between diverse implementations at both protocol and semantic / model levels.

Flexibility is supported by allowing different choices of components realizing collections of ODA functions, and well-defined Connectivity Services exposed by NaaS implemented as TM Forum Open APIs.

This flexibility leverages the polymorphic design of TM Forum Open APIs that allows a deployed instance of a NaaS API Suite to be dynamically configured to support different service model payloads. This relies on each ODA Production components publishing their Service Capabilities in a Catalog that can be reached by prospective clients of these Service Capabilities. A significant business benefit resulting from this approach is that each operational domain can start exposing their services in the catalog when they are ready avoiding the need for a big bang transformation.

4.1. General Modelling Principles for CFS /RFS Services, and Resource Functions

Critical to this Framework is to note that both CustomerFacingService and ResourceFacingServices are both modeled as services. So, what are the general guidelines from the Information Framework separating them and deciding whether they are needed in a specific implementation?

- GB922 Service Overview distinguishes Customer Facing and Resource Facing services in the following text (GB922 Service overview sec 1.1.3):
    - "This enables us to model a wide variety of services in a common class hierarchy while differentiating between Services that are obtained as a Product by a Customer versus those that aren't. As we will see, a CustomerFacingService is one that is obtained as a Product by a Customer. Therefore, the Customer may have specific control over this Service via its associated Product. In contrast, the Customer never knows explicitly which ResourceFacingServices are being used to support a CustomerFacingService. More importantly, the Customer shouldn't have

to know which ResourceFacingServices are being used, since the Customer hasn't explicitly obtained them."

- CFS are associated with resource technology neutral services i.e. they describe general capabilities and have attributes that are general across many technologies e.g. throughput, latency, SLA /loss rate, availability.
- CFS and RFS typically have different life-cycles, CFS are related to customer and product changes and RFS to technology changes.
- RFS are associated with resource technology specific services i.e. they have attributes that predominately relate to a specific technology.

- RFS typically do the following:
  - Map between the native protocols used to expose management of resources e.g. Netconf, YANG, SNMP, etc.
  - Provide some integrated approach to provisioning and assuring RFS that span multiple technical domains e.g. slices across RAN and Core.
  - They may be Operator, SP, ISV or Supplier provided.
  - In ODA Production RFS are optional as they are typically used to provide a service abstraction of physical resources.
- Resource Functions
  - An alternative to modelling resources as a service i.e. RFS, Resource Functions are convenient for virtualization since they model Resources as Functions which may represent legacy network appliances or functionally equivalent network applications. They have a more flexible set of relationships and can be supported by products or services. They support the use of profiles to relate an abstract intent service view to the underlying technologies as described in IG1194 Focus on Services not Slices [35]

***See also additional ODA CFS RFS Modelling principles for Connectivity Services in Section 5.5.1 ODA additional CFS and RFS* Best Practice**.

4.2. Operational and Technical Domains

In realizing implementation of the ODA Production domains there are several architectural modelling activities that take place to realize ODA Production capabilities.

These include:

- Defining the exposed capabilities and services from the ODA Production including NaaS.
- Operators implementing ODA Production Capabilities need to decide the Operational Domains that will expose each capability realized as Open APIs. (This decision is about setting an operational boundary and its linkage to exposed capabilities).
  Note Operational Domain and Platform concepts are related see:
  TR263F Relationships Between the Concepts of Domain and Platform [24]
  and specifically Section 5 on information models for relationship between concepts of Domain Platform and Entity. For example, some but not all domains can overlap/intersect, whereas platforms cannot.
- The Customer Facing Services (CFS) exposed by the ODA exposed capabilities using NaaS.

- Composition of subordinate CFSs to support the CFSs exposed by Production capabilities (iterative composition pattern). These subordinate CFSs may be from other Operational Domains both within the same operator or acquired from third party operators as happens with wholesale interconnect.
- Mapping of CFS to internal Resource Facing Services (RFS) that abstract into services the resource capabilities defined by Suppliers' Technical Domains whose boundaries are defined by technology and supplier choices. This mapping links the boundary decisions of Operational Domains to the Technical Domain boundary decisions of suppliers.
- RFSs can be atomic or composite to include other RFS (iterative composition pattern). This is a decision taken by the Operations / Integrator composing or creating RFSs based on deployment needs.
- In a Service Oriented Architecture any exposed services can be consumed by any other service. Specifically, in Figure 3.3.1 the resources service can be consumed directly by CFSs without going through an RFS Service.
- Others tba.

4.3. Integrating third party products / services

Several implementation examples require services from other operators e.g. Wholesale interconnect such as that in MEF Carrier Ethernet, cloud Services, etc.

Composition of these partner operator services with those of the operator may take place in the product / product offering or at ODA Production level. Where these services are composed and orchestrated is an operator decision that needs to consider the following:

- Where the partner operator service is unrelated to those provided internally it can be composed in ODA Core Commerce.
- Where there is strong technical coupling such as with Carrier Ethernet and 5G RAN Core Network (through Control User Plane Separation-CUPS) it is more likely that composition will be as a CFS composed within ODA Production.

4.4. Support of Automation & Closed Control Loops

Various kinds of automation and closed-control loop scenarios are possible within the ODA Production area. These are several flavors of closed-loops, depending on the level of the management that is required and on the policy management application selected:

- CFS/RFS based closed-loops, managed at the Service Orchestration layer. These, for example, may include the preservation of End-to-end slice SLAs/OLAs, as demonstrated in the 2018 catalysts.
- End-to-End Resource closed-loops managed at the End-to-End Resource layer. A good example for that will be a need to "replace" a slice caused by a technical malfunction or resource congestion.
- Technical Domain closed-loops, typically at a very low latency. Many of the domain resource problems fall under this category.

These closed-loops will use the policy management functions that exist in each one of the participating OSS applications when the policy actions can be cascaded. However, there is always the policy that controls the entire loop. That may be a policy within the highest-level Orchestration management system, whether it is at a Service level or at a Resource level. Alternatively, an external centralized, policy manager can be used to control many of these cases, as suggested by ONAP.

The diagram below shows different cases of closed-loops within the ODA Production Functional Group derived from multiple 5G catalysts.

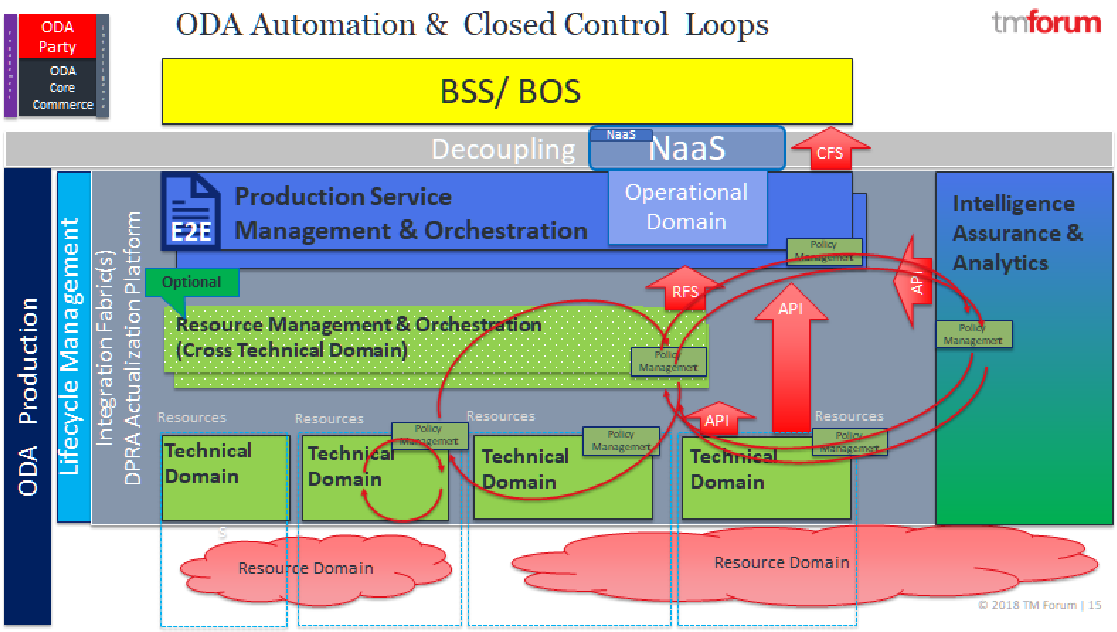


Figure 4.4.1: Automation & Closed-control loops

4.5. 5G Management Architecture Overview

This chapter includes an analysis of related work by other standardization organizations (SDOs) as well as open source work.

4.5.1. 3GPP Slice Management Architecture

Slice management and Communication Service concepts are being progressively defined in several 3GPP documents which at the time of this report are still under development. This material is based on understanding at the time of this report December 2019 and is subject to change.

The relationship between Network Slices' and Communication Service is described in the section after the core 3GPP documents are introduced below.

Slices and Sub slices

These are introduced as concepts in:

- TS 23.501 V16.1.0 (2019-06) System Architecture for the 5G System; Stage 2: (Release 16) [10]:
  Defines in sec 5.15 the core network characteristics of a network slice and the precise

attributes in the S-NSSAI and the NSSAI identifiers needed to support networking – which are 5G and mobile specific. It introduces the definitions:

- **Network Slice:** A logical network that provides specific network capabilities and network characteristics.
- **Network Slice Instance (NSI):** A set of Network Function instances and the required resources (e.g. compute, storage and networking resources) which form a deployed Network Slice.
- **Network Slice Subnet Instance (NSSI):** a set of network functions and the resources for these network functions which are arranged and configured to form a logical network.
    - 3GPP TR 28.801 V1.2.0 Study on management and orchestration of network slicing for next generation network (Release 15) [11].
      This lists a comprehensive set of use cases for Network slices including a management exposure requirement to support customer defined Network slices instances (section 5.1.9).
- 3GPP TS 28.530 V1.0.0 Management of 5G networks and network slicing; Concepts, use cases and requirements (Release 15) [27]:
  Examines use cases slice models examples between Network Operators (NOP) and Communications SP (CSP). This study assumes that 5G slices instance are exposed directly among NOPs and CSPs. However, this assumption does not align with the ODA Core Commerce to ODA Production requirement that the NaaS APIs are technology neutral.

Network Slice Models

Identification of a Network Slice is done via the Single Network Slice Selection Assistance Information (S-NSSAI). The NSSAI (Network Slice Selection Assistance Information) is a collection of S-NSSAIs. Currently 3GPP allows up to eight (8) S-NSSAIs in the NSSAI sent in signaling messages between the User Equipment (UE) and the Network. This means a single User Equipment (UE) may be served by at most eight Network Slices at a time. The S-NSSAI signaled by the UE to the network, assists the network to select a particular Network Slice instance. An S-NSSAI consists of:

- ·A Slice/Service Type (SST), which refers to the expected Network Slice behavior in terms of features and services.
- ·A Slice Differentiator (SD), which is an optional information that complements the Slice/Service type(s) to differentiate amongst multiple Network Slices of the same Slice/Service type.

The S-NSSAI may be associated with a PLMN (e.g., PLMN ID) and have network-specific values or have standard values. An S-NSSAI is used by the UE in access network in the PLMN that the S-NSSAI is associated with.

An example of communications services using multiple NSI(s) are illustrated below:

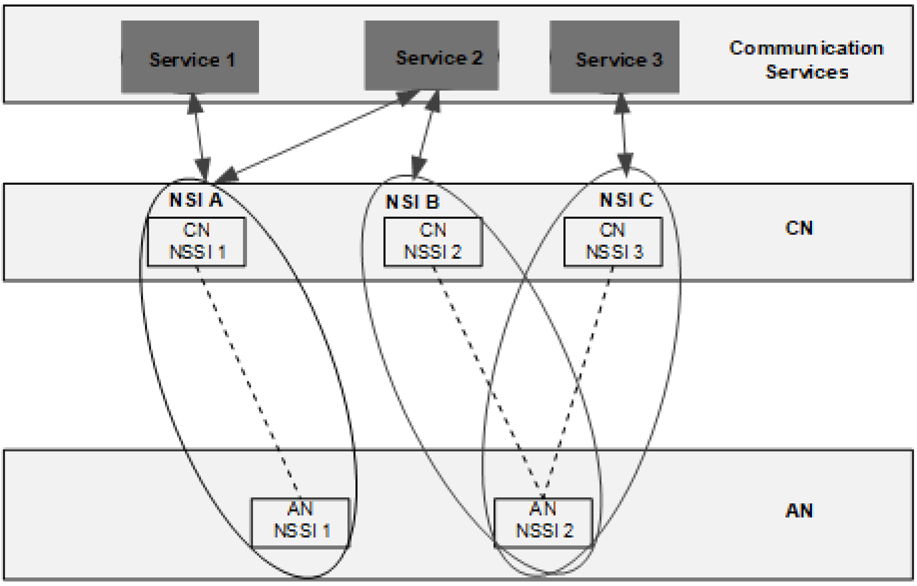


Figure 4.5.1: Examples of NSI and NSSI relationships in CN and AN

As described in 11] the following phases describe the network slice lifecycle:

- Preparation phase
- Instantiation, Configuration and Activation phase
- Run-time phase
- Decommissioning phase

This is illustrated below:

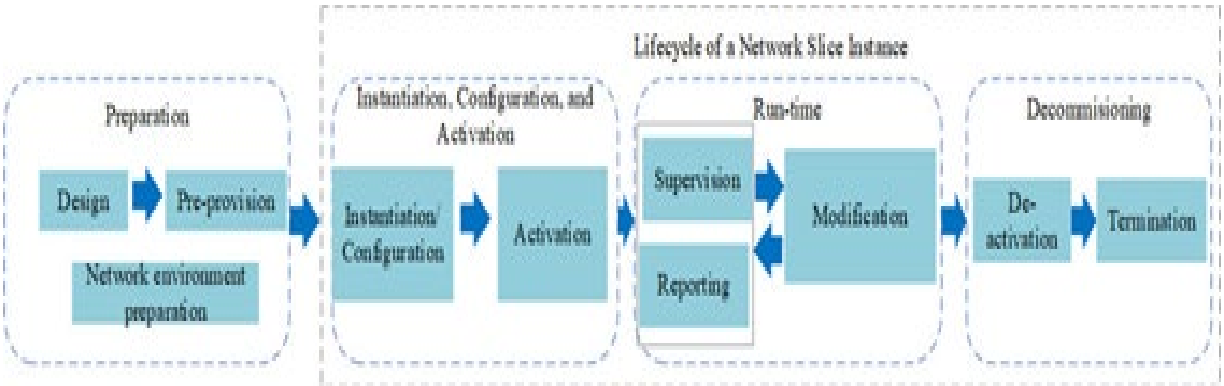


Figure 4.5.2: Network Slice Lifecycle

4.5.2. 3GPP Communication Services and Slices Management

Communication Services:

- 3GPP 28.530 Management and orchestration; Concepts, use cases and requirements [27] introduced the concept of a Communication Service (CS) that sits above and is distinct from Network Slices.
- 3GPP 28.805 Study on management aspects of communication services [12] outlines a model in Sec 4.2 and 4.3 which introduces the notion of Customer Facing Comms Service CFCS) and Resource Facing Comms Services (RFCS) within Customer Service Management

Functions/ Service Management Layer which sit above Network Slices. Their purpose is to allow customer communication service instances to be tenants on Network Slices and for details of Network Slices to be hidden through use of Profiles Refer to the following diagrams in 28.805:

- Figure 4.2.2.1: Customer aspects as well as, service and resource aspects regarding CSMF.
- Figure 4.3.1: Management model for management of communication services.
- An outline Management Model in Figure 4.4.2.1: Example of relationships shown between different entities.

- The ODA Connectivity Service model described in Part III provides a more detailed model that fulfils the detailed requirements in 28.530, 28.805 such that these services can be integrated with ODA Implementations that use the TM Forum Information Model GB922 (aka SID).

Relationship Communications Service and Network Slices

From 3GPP TS 28.530 V15.0.0 (2018-09) Management and orchestration; Concepts, use cases and requirements (Release 15) there is a proposal on a general architecture to support the use cases proposed which are mostly between the CSP and the NOP (definitions below diagram).

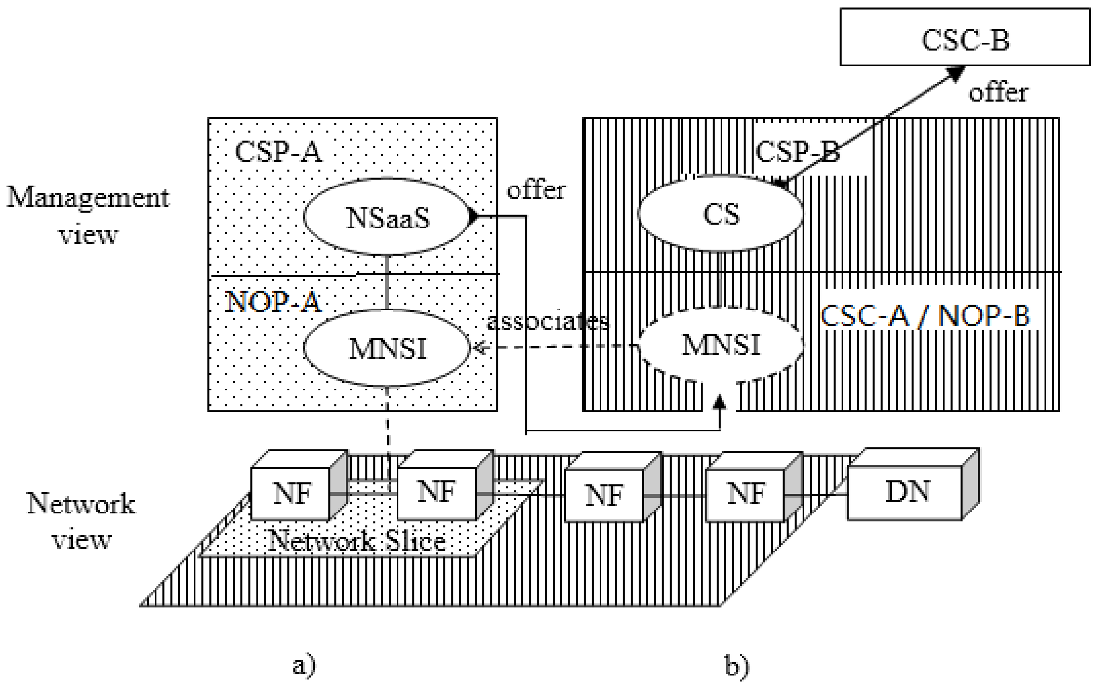


Figure 4.5.3: 28.530 Figure 4.1.6.1: Examples of Network Slice as a Service being utilized to deliver communication services to end customers

KEY

CS Communication Service

CSC Communication Service Customer

CSP Communication Service Provider

NOP Network Operator

- Related to 28.530 Fig 4.1.6.1: In the 3GPP document 23.501 5G System Architecture there is a definition of Network Slices and the S-NSSAI which is the identifier for Network Slices. The constraints on S-NSSAI are in section 5.15.2.1 … 'An S-NSSAI can have standard values or non-standard values. S-NSSAIs with PLMN-specific values are associated to the PLMN ID of the PLMN that assigns it. An S-NSSAI shall not be used by the UE in access stratum procedures in any PLMN other than the one to which the S-NSSAI is associated.
- Progressively the notion of Communication Services being the abstraction of network slices presented to customer is merging strongly in 3GPP 28.805 Study on management aspects of communication services [12].

Summary

This review confirms the need for separation of Customer Facing and Resource Facing Services to model 3GPP Communication Services especially from recent results in 28.805 and also positions Network Slices as RFS or resource level concepts.

4.5.3. Network Slice Management Requirements

The 5G Slice management requirements derived from these catalysts are documented in Appendix A: Network Slice Management Requirements.

4.6. Connectivity Service Overview

To realize the ODA decoupling principle between ODA Production and ODA Core Commerce and other ODA functional grouping for communication services, a technology neutral, generalized Connectivity Service model [1] needs to be exposed from ODA Production using the TMF 909 NaaS Component API Suite [4].

This model is designed primarily to align with TM Forum SID models in GB922 Resource Domain Business Entities, and the abstracted connectivity concepts in ONF TR -512 [30]. The reason for these choices is to provide a clean integration approach with the GB922 Information Model that is the basis of all ODA Functional Groupings.

This ODA Connectivity Service model can be extended to support technology specific Connectivity Service that supports the ODA Production Implementation Framework RFS services.

The NaaS API Component Suite TMF909 can be used for either CFS or RFS services since both are based on the SID Service Model.

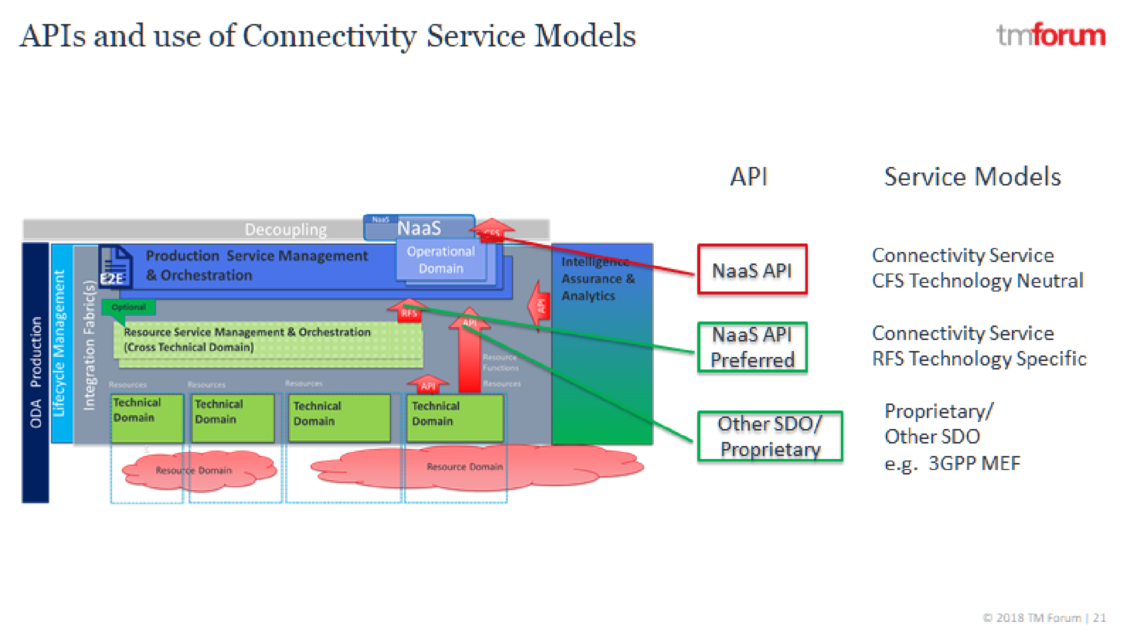


Figure 4.6.1: Connectivity Service Model for ODA Production

The key modelling aspects shown above are:

1. Use of the Connectivity Service model (resource technology neutral) for CFS (maybe composed CFSs) exposed services from ODA Production Service Management & Orchestration using TMF 909 NaaS API Component Suite.
2. Connectivity Service model with Technology extensions for RFS (maybe composed RFSs) for exposed services form ODA Resource Service Management & Orchestration for which use of TMF NaaS API Component suite is preferred.
3. Technology Domain exposed services which may use proprietary or other SDO interface specifications and may use proprietary or other SDO models e.g. 3GPP 5G specifications, MEF Carrier Ethernet Proposals, Open Daylight SDN, etc.

In addition to the service models for the API shown above it will be necessary to create:

1. Best practices for Mapping /extension between these service types (e.g. use of Open API extension mechanism for going between technology neutral and technology specific models).
2. Example mapping to proprietary models from other SDO such as for 5G.

---

[1] We use term Connectivity Service rather than Network Service as this has a particular meaning in ETSI NFV virtualization studies and 3GPP. In this context Connectivity Services are about the exposed operational service and cover: connectivity over transport networks and services which have embedded processing (Edge) and Storage (vCDN) as described in TR 255. [13], [14], [15]. The 3GPP Communications Services concept is similar to SID Service used by both CFS and RFS concepts.

---

[1] This is different from a typical Architecture where all architectural components are mandatory.

Part III: ODA Production Modelling Principles

5. Connectivity Service and Slice Management Data Modelling

5.1. Modelling requirements

The primary modelling goal is to define the resource technology neutral Connectivity Service Models that can be exposed by ODA Production Operational Domains as CFSs and how these are mapped and transformed onto technology /resource level models.

Service Models enable intent based Service Models realized using closed control loop and autonomic self-healing implementations within the Operational/ Connectivity Service Domains. This facilitates ODA Production decoupling and improved business agility.

5.1.1. Business Requirements Goals

Req 1. Resource Technology Decoupling:
ODA Production Connectivity Services shall be defined to be resource technology independent. For example, as 5G models and specifications evolve, the mapping and transformations can be changed without impacting the exposed Connectivity Service models from ODA Production and hence OSS/BSS implementations. It also permits hybrid 5G/4G operation of Connectivity Services.

Req 2. Multiple Operational Domain Deployments:
Support multiple ways for CSPs to group / compose Technical Domains, e.g. 5G, core, into Operational Domains.

Req 3. Operational Domain composition:
Support Composition of Operations Domains including composition of Operational Domains from other CSPs.

Req 4. Abstraction:
Define the necessary transformations from the Operational Domains of the CSP to the Technical Domains from the suppliers (to achieve ODA Production decoupling from technology specific models).

Req 5. Simple Intent based Connectivity Service models:
To support decoupling of Connectivity Services exposed by ODA Production Operational Domains (CFS) and internal RFS shall be modelled as services having functions (a SID concept modelled formally as ResoureceFunctions with configuration Features and characteristics) that are mapped /transformed to the detailed technology specific resource models and their configuration .

Use of ResourceFunctions avoids complex resource hierarchies being directly exposed from the ODA Production Domain CFS or in internal RFS Service which is a major source of integration complexity and coupling with traditional OSS/BSS.
An example of these principles based on the GSMA Generic Slice Template (GST) attributes has been developed in the Digital Transformation World 5G Riders on the Storm catalyst and documented in IG1194 Focus on Services not Slices. [35]

Req 6. Orchestration:
Support multiple levels of orchestration each decoupled within ODA Production Operational and Technical Domains - supporting composition of both CFS and RFS services. Where each level exposes Intent-based services that are realized internally with autonomic network solutions using closed control loops, policy management, AI and other technologies.

5.1.2. Connectivity related Requirements

Req 7. Concatenation:
Connectivity Models shall support concatenation of connectivity services provided across multiple Operational Domains and multiple CSPs.

Req 8. Composition:
Connectivity Models shall support composition of connectivity services (CFS and RFS) and abstraction so that simple e2e connectivity services /CFS can be constructed free from details of component CFS and RFS and specific technologies.

Req 9. Connectivity Service Abstraction:
Define a generalized Connectivity Service Model that abstracts of multiple technology specific models from other groups covering 5G Network Slices, Carrier Ethernet, SD-WAN and other technologies. Model must be extensible to support technology specific Connectivity Service models.

5.1.3. Information Model Requirements

Req 10. Mappings:
Information Models shall provide the mapping and links between the Connectivity Services Data Model and the Information Framework (aka SID) so that Connectivity Services can integrated seamlessly with the models in other ODA groupings, such as products in ODA Core Commerce, and RFS derived from other SDO models.

Req 11. Key Model Elements:
The main elements that shall be modelled are the specifications for the exposed Connectivity Services (and constraints such as Point of Presence / Service Access Points), the specification of individual customer or tenant connections/ flows (which may be multi point and involve storage and processing e.g. vCDN), and interconnections between Operational and Technical Domains.

5.1.4. Data Models Requirements

Req 12. Operational Domain:
Define Data Models that can support multiple operational domain deployment models both within a single CSP and across multiple CSPs.

Req 13. Technology neutral and specific:
Data Models shall cover both CFS (technology neutral) and rules for defining RFS (technology specific) services and APIs.

Req 14. Open APIs Data models for Connectivity Services:
Shall augment the existing TM Forum Service Open APIs defined in TMF909 NaaS Component Suite which can be used to expose Connectivity Services (CFS) from ODA Production Operational Domains.

Req 15. API Schemas:
Connectivity Service API Data Models shall be in the form of schemas together with their mapping to the SID Information Models

5.1.5. Model Mapping requirements

Req 16. Validate though use of exemplar deployment/ implementation models:
– using catalyst and 5G results - that these Connectivity Service models can be mapped to 5G RFS and 3GPP Resource models using the Implementation Framework architecture models described in this Guideline.

5.2. 5G Communication Services and Network Slices

Recent 3GPP Studies on communication services modeling and relationship to Network Slices [12] support this approach but are currently less detailed and less generalized than these TM Forum models.

The architectural models described in this Guideline show that 3GPP 5G Network slices models are Resource Facing Service (RFS) and Resource models which need to be transformed, abstracted and orchestrated into intent based Connectivity Service models that are exposed as NaaS Services (Customer Facing Services in SID terms) from ODA Production Operational Domains[1].

5.3. Connectivity Service Model Overview

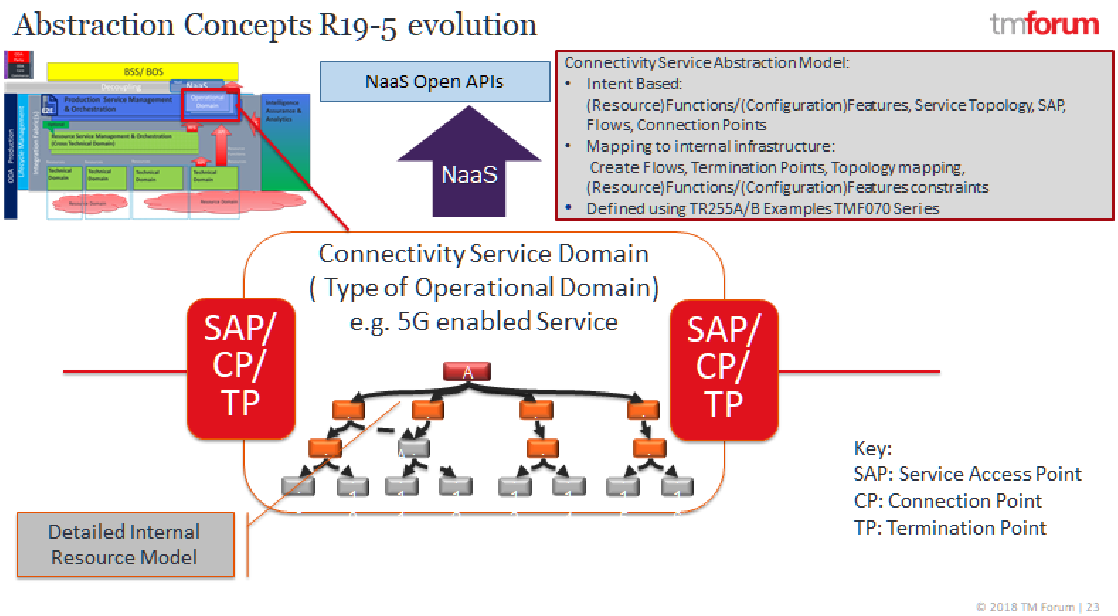


Figure 5.3.1: IG1165 [29] Abstraction Concepts

This diagram shows the key concepts for a Connectivity Service abstraction of technologies such as 5G Network Slices:

- The Connectivity Service Domain – a type of Operational Domain that exposes and publishes (in a catalog) the Connectivity Services and the type of connection /flows that can be established across it.
- The Connectivity Service abstraction model - for example a 5G enabled Connectivity Service based on a RAN 5G slice Network Segment Domain exposed by a Connectivity Service Domain using NaaS APIs. This supports:

- Intent based operations by modelling
    - ResourceFunctions to define the capabilities of the Connectivity Service Instances and Flows/ Connections.
    - Service Access Points (SAP) which are the 'static' logical view of where the Connectivity Service Domain can deliver the Connectivity Service Flows (e.g. LAN, Aggregation, Point to Point) at i.e. London, Paris, etc., and the logical location of internal nodes. Note Internal nodes are supported in TR255A [13] but are not needed in the examples considered in this guideline.
    - (Service) Topology (aka Connectivity Potential TR255A): the external view of the Service Topology and constraints, and any internal service nodes, examples being MEC and vCDN; and constraints on how flows can be established by tenants of the exposed Connectivity Services. See TR 255A [13]
      Note Studies in R19-5 suggest that Topology and Connectivity Potential can be modelled with Connectivity and Adjacency Graphs which is a SID concept.
    - Connection Points (CP) which identify the endpoints characteristics of Connectivity Service Flows.
    - Termination points (TP) associated with SAP where the User and Control plane Flows are physically or logically delivered and are protocol specific.
- Mapping to internal Resource Infrastructure.
    - The internal detailed resource model – addressed by work in 3GPP and elsewhere is hidden by the Connectivity Service by use of Resource Functions (with Configuration Features and Characteristics) and constraints associated with Flows.
    - Supports creation of Flows among Connection Points.
    - Mapping from service topology to internal resource topology.
    - Enforces ResourceFunction/ConfigurationFeature constraints.

- NaaS Open APIs providing the management of instances of Connectivity Service Flows across the Connectivity Service Domain.

The patterns and principles for modeling Connectivity Services and flows across them are documented in:

- TR255A Connectivity Patterns for Virtualization Management [13] that provides a series of examples and defines a notation and patterns for combining Resource Functions with graphs can be used to describe a wide range of examples. Uses Resource Functions (with supporting Configuration Feature aka Feature & Feature Group and Characteristic) for intent-based management abstractions that decouple service models from internal detailed resource realization.
- TR255B Specification Requirements for Resource Functions [14] sets out the Specification Requirements on Resource Functions that support both intent and detailed (traditional) resource management and hybrid deployments using both approaches. It also covers API patterns for abstracted intent and detailed based models and uses a Firewall example. These are the basis for TMF664 Resource Function Activation and Configuration API.

5.4. CFS Core Connectivity Service Model

The model is based on elaborating prior ZOOM Project related work reported in:

- TR255A Connectivity Patterns for Virtualization Management [13]
- TR255B Specification Requirements for Resource Functions [14]
- GB922 Resource Domain Business Entities formerly GB922 Logical and Compound Resource [28]
- IG1165 NaaS Network Model and Termination Point Composition Best Practice [29]

Note throughout this section the names of entities are used; however, in the formal model described in Appendix C is necessary to distinguish specification and entity classes following the SID Specification Entity Pattern. The simplified approach used in this document keeps the discussion short and simple by not introducing too many similar terms. Readers should be aware that there are synonyms used for the same concepts across TR 255A/B and Open APIs including TMF 664 Resource Function Activation and Configuration [36] and may be harmonized in a later release.

A simple mapping is:

TR255	GB922 Resource and Common	Open APIs
Function and Resource Function {1}	Resource Function {2}	Resource Function {3}
Feature / Feature Group {1}	Configuration Feature (with composite pattern) {4}	Feature with 'is bundle' attribute for composition {1}

Appendix D contains a more comprehensive setting of mappings and identifies known issues that are being addressed by mainly the Information Framework team.

Notes:

{1} **TR255** Resource Function Activation and Configuration Suite R17.5.0 [40]

{2} **GB922 Resource** v19.5.1 Section 3.2.11 Resource Function: Support for Virtualization and Resource Abstraction pages 38ff [41]

{2} TMF664 Resource Function Activation and Configuration API REST Specification R17.0.1 [42]

{4} **GB922 Common** v19.5.1 Section 2.16 Configuration and Profiling ABE Pages 394ff [44]

5.4.1. Connectivity Service Domains Static Connectivity Service elements

Purpose

The primary purpose of the Connectivity Service Model is to define how connectivity services should be modelled in a Service Catalog. The static part of this model captures in a machine-readable manner the material usually captured in a Product/Service Brochure, where, for automated service lifecycle management, part of this information needs to be formally modeled as it set constraints on what can be dynamically ordered. For example, if a Service Order specifies a Service Access Point (SAP) not listed in the Connectivity Service Static model it will fail. Clearly as services evolve the number and locations of SAPs may change which is why it is advantageous to have this information explicitly held in a Catalog.

Modeling Background

IG1165 [29] focused on a review of industry network models which traditionally have been focused on the static physical aspects of Network Service/ Connectivity Service Domains and concentrate on resources rather than services. This reflects the traditional situation where networks are physical and relatively static as they are planned and built over many months.

For more modern virtualized networks, the Connectivity Services exposed by these domains can change more frequently which requires them to be formally specified and published, rather than implicitly described.

Connections or flows for individual customers or tenants can then be created over these Connectivity Service Domains. This impacts the way that the Connectivity Service Domains are composed and concatenated which is described later in Section 5.5.2.

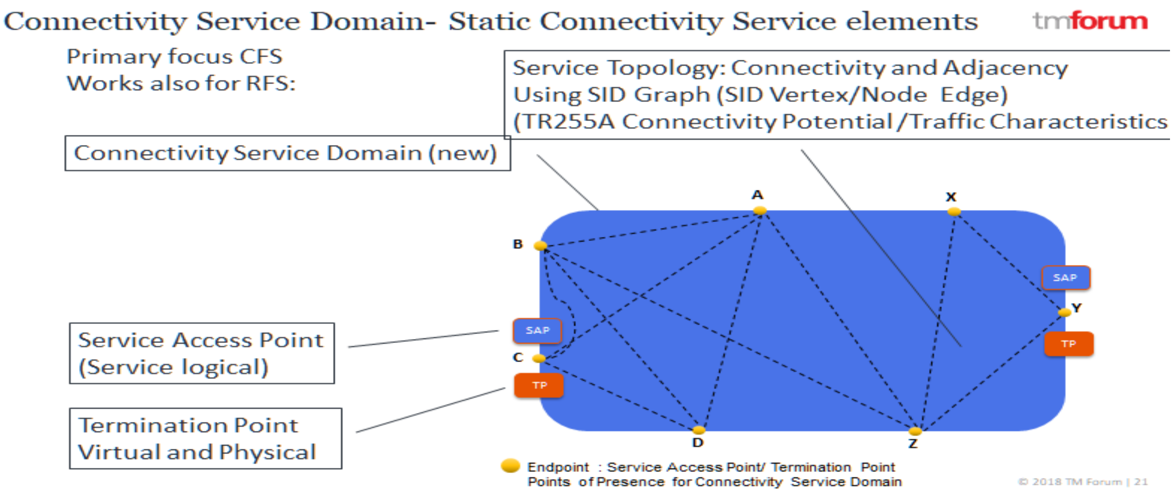


Figure 5.4.1: 'Static' Connectivity Service model

These Connectivity Service Domain elements describe the service that is published in a catalog and are largely concerned with:

- Connectivity Service Domain (CSD) that defines the scope of the services i.e. regions and locations and is associated with an Operational Domain defined by the CSP controlling that service. It may support multiple Connectivity Services
- Service Access Points (SAP) where flows can be exposed and internal resources that need to be modeled as part of the CSD. e.g. MEC /vCDN Nodes.
- Termination Point (TP) where technically Connectivity Services are physically /logically delivered.
- Service topology that set constraints on the Flows/Connections that can be established (including performance, capacity and SLA limitations).
  TR255 has concepts of Connectivity Potential and Traffic Characteristic which are modelled in the SID as types Graphs.
  There is an ongoing Information Framework discussion as to how best to model graphs in a simple extensible manner especially when they need to be represented in Open API JSON schemas. The emerging preference is to use Adjacency Graph and Connectivity Graph to construct topology in a simple but extensible manner

This information is published as part of the service catalog information.

Example:

Typically, in catalysts such as Skynet this is part of the catalyst scenario: e.g. the service is available at London, Paris, Tokyo, Ivory Coast and is regarded as static and invariant. However, in the dynamic world of virtualization and use of catalogs this information can change albeit relative slowly and must be captured formally in the Service Catalog.

General Example from TR255A:

Connectivity Service Domain:

- Feature Group: Connectivity Topology
  - Connectivity graph: List of SAP /Nodes and links/ Directed Edges
  - Connectivity Potential: traffic matrix for Directed Edges.
  - Ingress traffic: traffic characteristics for each SAP.

Note SID Topology Model has been changed extensively in R19.5. This means terms used in TR255 R17.5.1 need to be mapped to a different SID model to that when they were created. **Affects Connectivity Graph, Vertex, Directed Edge, Connectivity Potential, & Traffic Matrix**

Connectivity Service 'static' elements:

Entity	Source	Definition
Connectivity Service Domain	New entity	May be contained within a Management Domain SID and may be re factored from TIP Management Domain
Connectivity Service	New	ServiceSpecification for Flows / Connections. Subclass of SID Service Specification GB922 Service
Service Access Point	TR255A page 8 TR255 pg76	a kind of RF that handles access into and out of another RF (such as an

Entity	Source	Definition
		application RF or virtualized appliance RF) Note: Requires association with Location page 9 and Ingress Traffic Characteristic page 9.
Service Access Point	SID	Represents a set of parameters associated (directly or indirectly) with a unique (logical and/or physical) resource where the Service can be accessed
Connectivity Potential/ Resource Graph	TR 255A pg10	Matrix of traffic characteristics between nodes/SAP pg. 10 Model as Adjacency and connectivity relationship in a type of SID Graph called Resource Graph (new proposal for R19.5)
Resource Function	GB922 Resource Domain Business Entities formerly GB922 Logical and Compound Resource Computing and Software adopted by TR 255A pg. 9, pg. 21 & 28 TR255 pg. 75	Specifies a function as a behavior to transform inputs of any nature into outputs of any nature independently of the way it is provided. (GB922 Resource Domain Business Entities formerly GB922 Logical and Compound Resource Computing and Software)

Table 1: Connectivity Service Static Elements

Note SID Topology Model has been changed extensively in R19.5.

5.4.2. Connectivity Service Domain Flow/Connection Elements

The model allows for flows supporting tenants to be established across the Connectivity Service Domain Service Model.

Specification of Flow/Connections are typically held in a Service Catalog.

Instances of Flow/Connections are typically held in a Service Inventory.

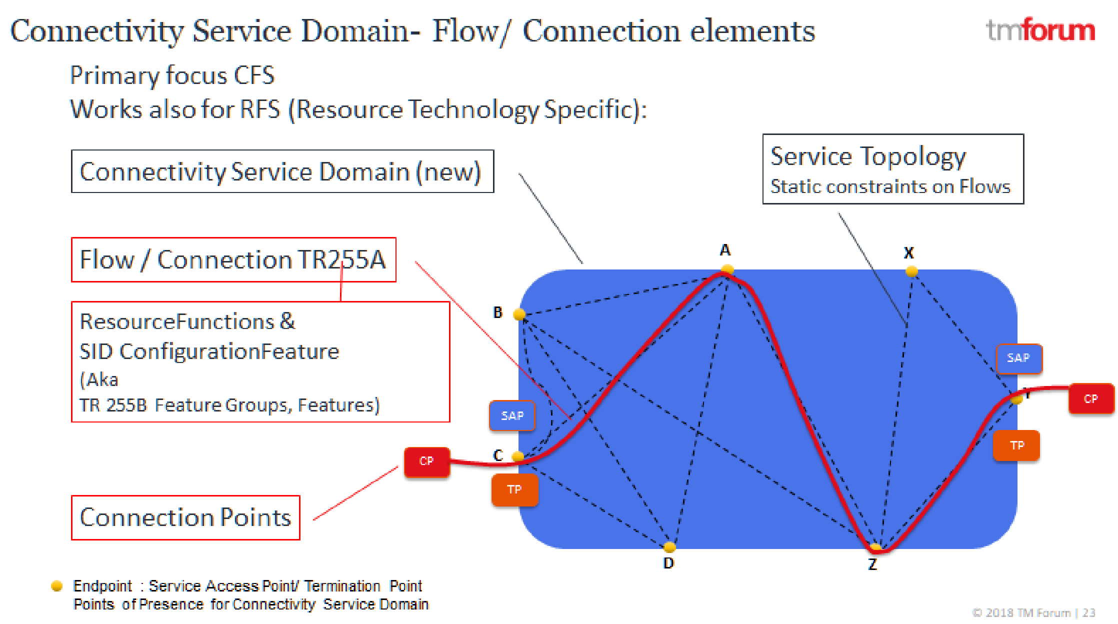


Figure 5.4.2: Core Connectivity/ Flow Service Model

The notion is to provide an abstracted technology neutral view of a Connectivity Service and instances of Flows/connections that can or have been established across them, aka described as tenant services when described in a Cloud content.

The model above is a simplified example of a point-to-point Connection /Flow as this document's focus is 5G Communication Services enabled by Network Slices.

TR255A connectivity models and patterns provide strong guidance for more complex cases including those requiring internal processing and storage such as is needed for Mobile Edge Computing (MEC) and Content Delivery networks vCDN. These cases are not addressed in this snapshot but will be needed for future catalysts and are readily added using the connectivity patterns in TR255A.

The Connectivity Service model consists of several related entities that provide a static view of the Connectivity Service Flows that can be established across the static entities in support of different tenants or users of those Connectivity Services.

Examples:

In 5G Riders on the Storm multiple slices are established and connection and flows are moved between them to address changing network conditions during a flood emergency in response to changing operation needs and priorities for emergency services.

General Example from TR 255:

Flow feature group:

Flow Topology

- Flow Type: p2p (MEF: eLine), multi-point (MEF eLAN), aggregation (MEF e-Tree)
  - Flow direction: unidirectional or bidirectional
- Flow characteristics at ingress SAPs:
  - List of SAPs
  - For each SAP

- Static bandwidth
- Mean bit rate
- Burst bit rate
- Peak Bit rate
- Latency
- Loss rate

Connectivity Service Flow elements:

The TR 255A concepts of Features Feature Groups are replaced by the concept of Configuration Feature and in conjunction with use of Resource Functions makes composition of CFS, RFS and Resources more straightforward and is needed for simplification and automation of the concatenation and composition models described later.

For this release Features, Feature Groups and Resource Functions can be regarded as specializations of Configuration Feature.

Entity	Source	Definition
Flow/ connection	TR 255A Modeled as an RF pg. 14	An instance of connectivity across a connectivity potential where nodes and hops are visible list of hops with associated Traffic Characteristics
Resource Function	GB922 RD	A ResourceFunction is a function to transform inputs of any nature into outputs of any nature independently of the way it is provided. Each time a ResourceFunctionSpecification is provided whatever the way it is done (SoftwareSpecification, SoftBlackBoxSpec or PhysicalBlackBox), a ResourceFunction is created with its ConnectionPoints. The ResourceFunction is provided either by an InstalledSoftware, a SoftBlackBox, or a PhysicalBlackBox. These relationships are exclusive.
ConfigurationFeature	GB922 RD	SID Configuration Feature A ConfigurationFeature is a type of Entity described by a ConfigurationFeatureSpec. It is used to expose one or several ResourceFunctions to a consumer in such a way that it is easier to configure these ResourceFunctions from a consumer view point. A ConfigurationFeature may have characteristics configured according to the

Entity	Source	Definition
		possible configurations specified by the ConfigurationFeatureSpec.
Connection Point	GB922 RD	A ConnectionPoint implements a ConnectionPointSpec A ConnectionPointSpec is used to describe the information/data input to a ResourceFunctionSpec or output from a ResourceFunctionSpec. This description is captured independently of the way it is realized
TR255 Terms		
Function (Feature)	TR 255B	TR255 A Feature is how an RF exposes its capabilities to consumers. Unlike an RF, a feature cannot be independently deployed. GB922 Resource Domain Business Entities pg. 48 Modeled in SID as Configuration Feature(Spec)
Function (Feature Group)	GB922 RD and TR 255B	Feature Group is an aggregation of Features. A feature group may have characteristics in addition to the characteristics of the constituent features. TR 255B pg. 35 Modeled in SID as Configuration Feature(Spec)

Key: RD = Resource Domain Business Entities

Table 2: Connectivity Service Tenant Elements

Termination Point related entities are defined later in this document Section 7.6 Connectivity Service Best Practice as they are important for concatenation and composition modelling specifically: 5.5.3, 5.5.4, & 5.5.5

Examples:

Skynet BOS and 5G Rider catalyst all developed simple descriptions of Flows:

- For Skynet, it was a UK Japan Remote health Diagnostics check service,
- For BOS, it was a drone service,
- For 5G Riders on the Storm it was a set of eMBB based services.

Notably the exposed CFS was combination of a connectivity service and an application which reinforces the need to keep the Connectivity service definition simple and capable of being composed with other non-network based CFSs. Using widely different Connectivity Service models for each of these services will increase the complexity of mapping and undermine the economies of scale for CSP services by inhibiting reuse.

General Example from TR 255:

- FeatureGroup: Flow Topology
- Flow Type: p2p (MEF: eLine) (MEF eLAN), aggregation (MEF e-Tree)

5.5. Connectivity Service Best Practice

5.5.1. ODA additional CFS and RFS Best Practice

See also general CFS and RFS principles in Section 4.1 General Modelling Principles for CFS/ RFS Services and Resource Functions

For modeling connectivity service and mapping to underlying technology RFS models the following additional ODA best practice characteristics are adopted:

- Use SID TR 255A ZOOM terminology for core Connectivity Model for CFS and linking to ServiceSpecification and supporting Resource Function and ConfiguationFeature (aka Feature / Feature Group in TR255). This creates a self-similar / pattern for CFS, RFS and Resource Function models making abstraction simpler and easier to automate.
- Customer Facing Service Model Elements:
    - For ODA CFSs must be technology neutral.
    - Exposed by Operational Domains e.g. Connectivity Service Domains.
    - Different Instances of the same CFS service definition can be exposed by multiple Operational Domains (Re-use). E.g. multiple CSPs may use the same Carrier Ethernet Service Spec
    - exposed Connectivity Service are invariant to different ODA Production arrangements of Operational Domains and Technical Domains and any changes/substitutions (ODA Decoupling Principle)
    - CFSs can be composed.
- Resource Facing Service Model Elements
    - Service offered are dependent on technologies.
    - Exposed Services are technology specific and vendor neutral.
      i.e. changing technology causes change of RFS Specification and any subsequent mapping to CFS.
    - Different instances of same RFS Service definition can be exposed by multiple Technical Domains (Re-use). E.g. multiple vendors may provide vCPE to the same RFS Specification
    - RFSs can be composed and aggregated from multiple Technical Domains.
- Resource Functions (a subclass of Logical resource are used to abstract resources in support of intent based models and virtualization and can be used to support Products and Service Specifications. IG1194 Focus on Services not Slices [35] shows how characteristics or what GSMA describe as Generic Slice templates attribute can be grouped as candidate ResourceFunctions which can be used to support both service and resource models.

5.5.2. Concatenation and Composition Best Practice

Connectivity Service Domains (CSD) are defined by individual CSPs. However, for many practical cases e2e customer services /products are composed by combining services from multiple CSD and multiple CSPs.

This section explores best practices for concatenation and composition of services from multiple Connectivity Services.

Two key requirements were identified in IG1165 [29] for addressing both Composition and Concatenation of Connectivity Service Domains provided by multiple Operational Domains either within a single CSP or across multiple CSPs.

- Composite network Connectivity Service Domains may be composed (either CFS or RFS based from multiple Network Technology Domains and /or Network Segment Domains (RFS based) performing specific roles e.g. Backhaul, Core (see TR263F Relationships between the Concepts of Domain and Platform for definitions [24]).
  This implies the need for self-similar models for Resource, Resource Facing Service and Customer Facing Services to make composition straightforward and automated.
  Composition may be black box hiding or abstracting the internal connectivity detail or may be glass box where the internal details are exposed.
- Concatenation of network Connectivity Service Domains through Cross-connect Links and Termination Points, at Service Access Points (for clarity not shown below but discussed later).

Note: Composition using solely CFS services between Domains can be a simpler approach and is described in Section 9.2. Minimum ODA Production implementation example.

These simple goals are captured in the following examples:

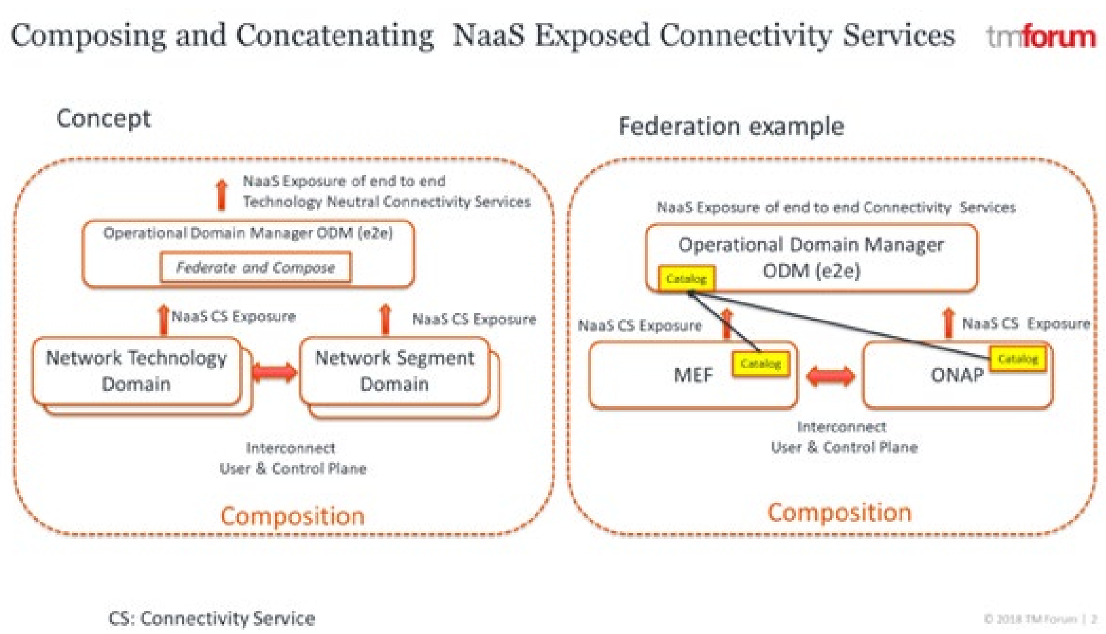


Figure 5.5.1: Composing & Concatenating NaaS examples

On the left is shown an end-to-end composite Operational Domain Manager federating and composing several NaaS Connectivity Services (RFS) exposed by several component Network

Segment /Technology Domains. It in turn exposes an end-to-end Technology Neutral NaaS Connectivity Service (CFS).

On the right side, integration is facilitated by the specification and publication in catalogs of the exposed NaaS Connectivity Services. To make composition an automated process - such as use of TOSCA configuration files as is done in ONAP - there are requirements on both what is in held in the catalog and conventions for defining the User and Control plane interconnects from a management perspective.

The following examples of the concatenation and composition of Operational and Technical/Technology Domain management are considered based on the 5G Exemplars used in IG1165 [29].

Example 1 network slice instance

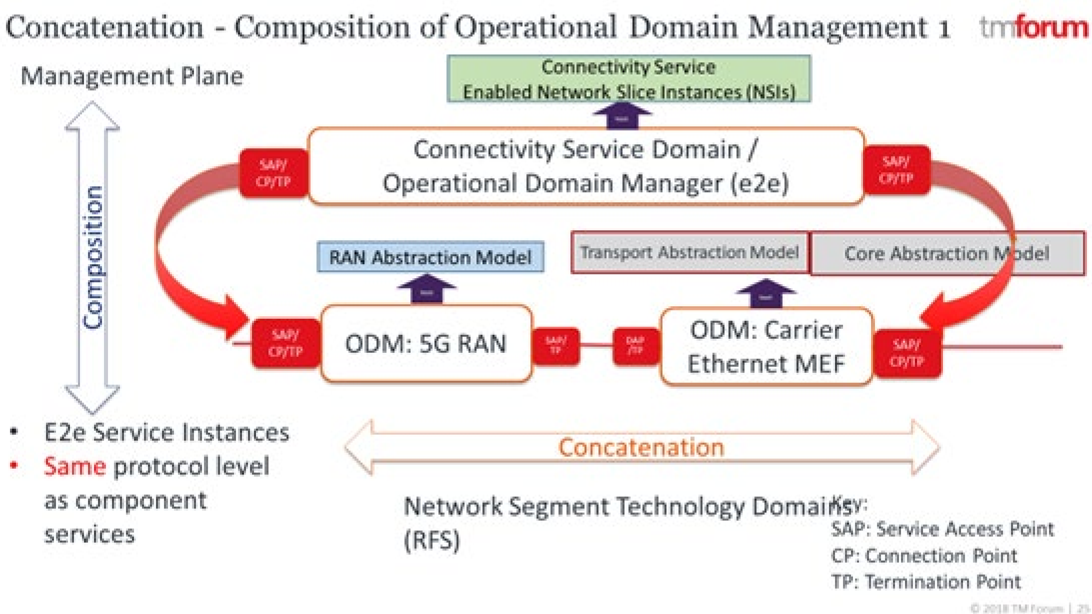


Figure 5.5.2: Concatenation - composition of component Operational Domain Management Example 1

In this example it is assumed that concatenation of the Network Segment/Technology Domains are between Termination Points (TP) operating at the same Protocol Level (in the OSI Management sense). Most commonly these are Layer 3 and Layer 2 protocols although application L7 protocols also need to be supported (see IG1165 Appendix 1 [29]).

It is assumed in this example that the e2e Connectivity Service Instances (CFS or RFS) are simply an abstraction of the underlying component network technical/ segment Domains services (RFS) i.e. a simplified abstracted connectivity model; and that the exposed end service Termination Points are identical to the component Termination Points i.e. operating at the same Protocol Level.

The requirements are:

1. Ability to concatenate Component Service Domains by interconnecting at the same protocol layer(s).
2. Ability to expose the customer facing Service Access Point, Connection points and Termination Points of the component services as e2e Connectivity Service Termination Points as shown by the red arrows. This may be by direct exposure, or mapping/translation. E.g. as might happen with IP NAT
3. Ability to hold internal mapping of the (interior) links between component service Termination Points that are not exposed to customer. Noting that for physical network this may cover layer 1, 2 and 3, but for virtualization-based solutions this may only require layer 3 termination points.
4. Ability to model abstracted services that are vendor and technology neutral and roll them up into an e2e abstracted service model.

Example 2 network slice instances

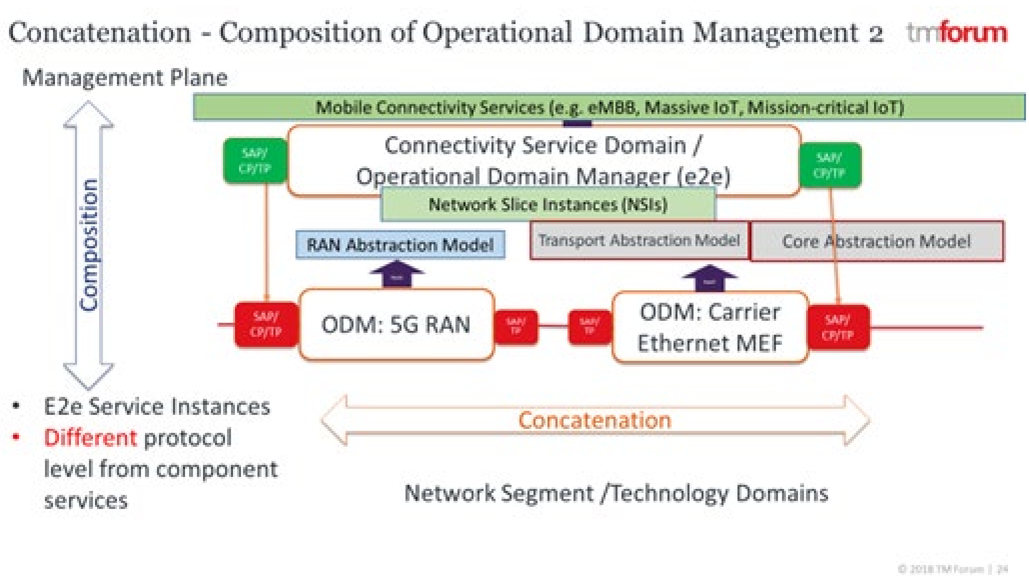


Figure 5.5.3: Concatenation - composition of Operational Domain Management Example 2

This case differs only in that the e2e ODM uses the network slice model internally and exposes a connectivity services that is tailored to specific applications e.g. eMBB, Massive IoT, Mission-critical IoT. I.e. a different abstract service model (CFS) to network Slicing Model, RAN, Transport and Core Abstraction Model (RFS / Resource).

The consequences are that the e2e service is operating at an Application level and will have a different protocol layer potentially with different customer facing Termination Points (e.g. different identifiers). That means the e2e ODM must hold a mapping of the application level customer facing Termination Points to the component Service Termination Points. This situation will also arise for over the top services such as Content Delivery networks.

5.5.3. Concatenation of Connectivity Service Domains (CSD) Service Topology

In many practical use cases the scope of a Connectivity Service delivered to the end customer cannot be realized by a single Operational Domain or by a single CSP. This means that multiple

Operational Domains / Connectivity Service Domains must concatenate Service Topologies in order to support e2e flows that deliver the overall service. Examples of these needs are seen in the MEF Sonata interface and some of the 5G multiple operator roaming models.

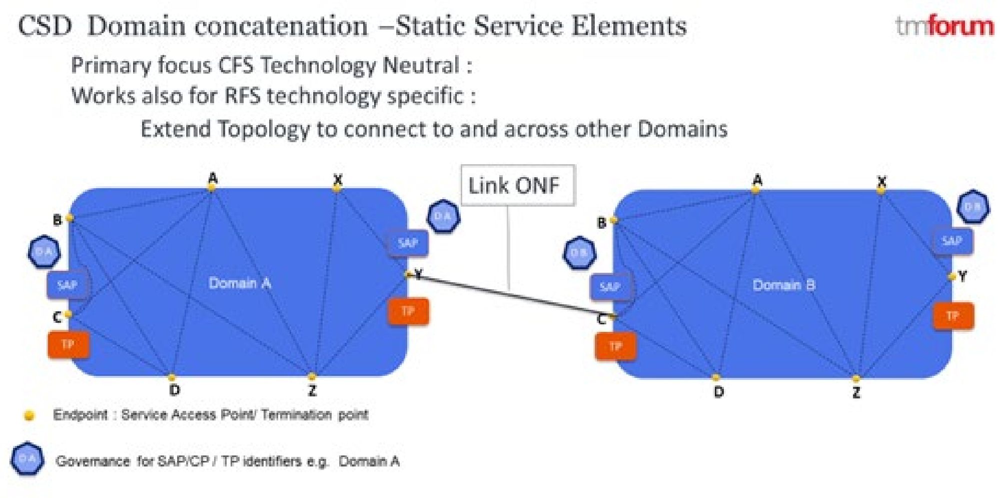


Figure 5.5.4: CSD Domain Concatenation – Topology aspects

This example shows two Communication Service Domains (topologies) are concatenated interconnected with a Link between an SAP on one Connectivity Service Domain /Operational Domain with an SAP on another domain. This requires the Link and the SAP to manage and associate with additional information about Termination Points associated with specific Layer Protocols available at either end of the link. However, the internal technology of each Domain is not exposed.

Note that the SAP and Termination Points (TP) on Domain A are governed by Domain A. i.e. rules for allocating identifiers, and similarly SAP and TP on Domain B are governed by Domain B.

This model can be used for either CFS or RFS services supported by common Resource Functions.

These additional entities effectively bind the SAP for the CFS Connectivity Service to a specific (logical) termination, port and layer protocol which is essential to be able to interconnect and interoperate the Link to convey Flows from one Operational Domain to another.

In TR255A links are Functions a.k.a. Resource Functions so that they can be composed of lower level protocol domains in the same way as ONF propose. This self-similar recursive pattern is necessary to support the many levels of abstraction needed and used in the ONF work.

ONF TR 512 [30] has a stable, conceptually well-thought-out model for representing interconnection of Forwarding Construct (FC) Domains using FC Ports that are associated with Logical Termination Point and Layer Protocols and this proposal re- uses those concepts and specifications.

Link and Termination Point related entities

Note these definitions also apply at the exposed e2e service Termination Points discussed earlier in sections 5.5.1 and 5.5.2 and are located here as the relationship with Links (Section 5.5.3) can also be covered in a single table.

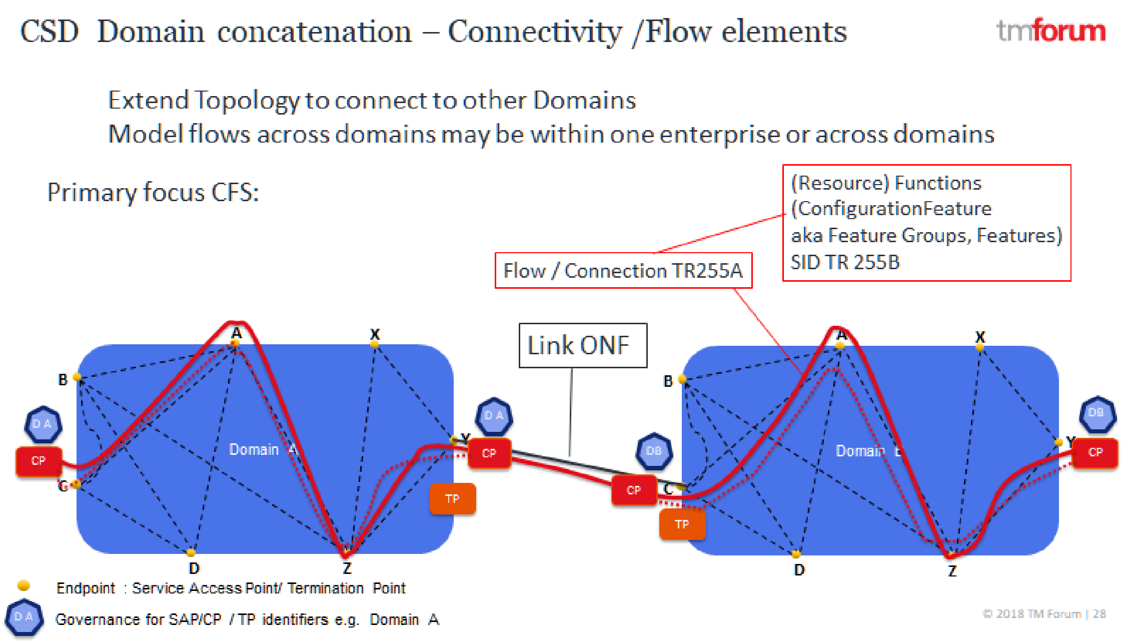
Entity	Source	Definition
Logical Termination Point (LTP)	ONF TR512 Equiv. to SoftConnection Point GB922 Resource Domain Business Entities and Endpoint in TR 255C and TOSCA. Synonyms: Termination Point	Encapsulates the termination and adaptation functions of one or more transport layers represented by instances of LayerProtocol
FcPort	ONF TR512	Association of the FC to LTPs is made via FcPorts. The FcPort class models the access to the FC function
ForwardingConstruct (FC)	ONF TR512	Class models enabled constrained potential for forwarding between two or more LTPs at a particular specific LayerProtocol. Like the LTP, the FC supports any transport protocol including all circuit and packet forms.
Layer Protocol (LP)	ONF TR512	Projection of an LTP into each transport layer is represented by a LayerProtocol (LP) instance.
ForwardingDomain (FD) Synonym of preferred term Connectivity Service Domain	ONF TR512	Class models the topological component that represents the opportunity to enable forwarding (of specific transport characteristic information at one or more protocol layers) between points represented by the LTP in the model. ONF concept equivalent to Connectivity Service Topology.
Termination Point Encapsulation	GB922 TR275	The TerminationPointEncapsulation (TPE) represents one or more functions that terminate/originate a signal that adapt a signal for use, and that enable a signal to propagate. Hence, a TPE can represent the end point of a signal flow. The TPE can

Entity	Source	Definition
		also represent the intermediate point of a signal flow. A TPE is capable of encapsulating multiple transport functions (G.805 termination functions, adapters, points etc.) at many different layers where the encapsulated transport functions are all related to the same signal flow. The encapsulated layers may be exposed via usage of instances of LayerTermination (LT). TPE is the agreed touchpoint between the Information Framework and the ONF TR 512 Common Information Model as defined in TR275 Core Networking Resources Business Entities [37].

Table 3: ONF TR512 Concepts and Definitions

5.5.4. Concatenation of Connectivity/Flows

This example shows a flow that an end customer / tenant wishes to consume, realized by concatenating across two Connectivity Service Domains.



SAP not shown for simplicity

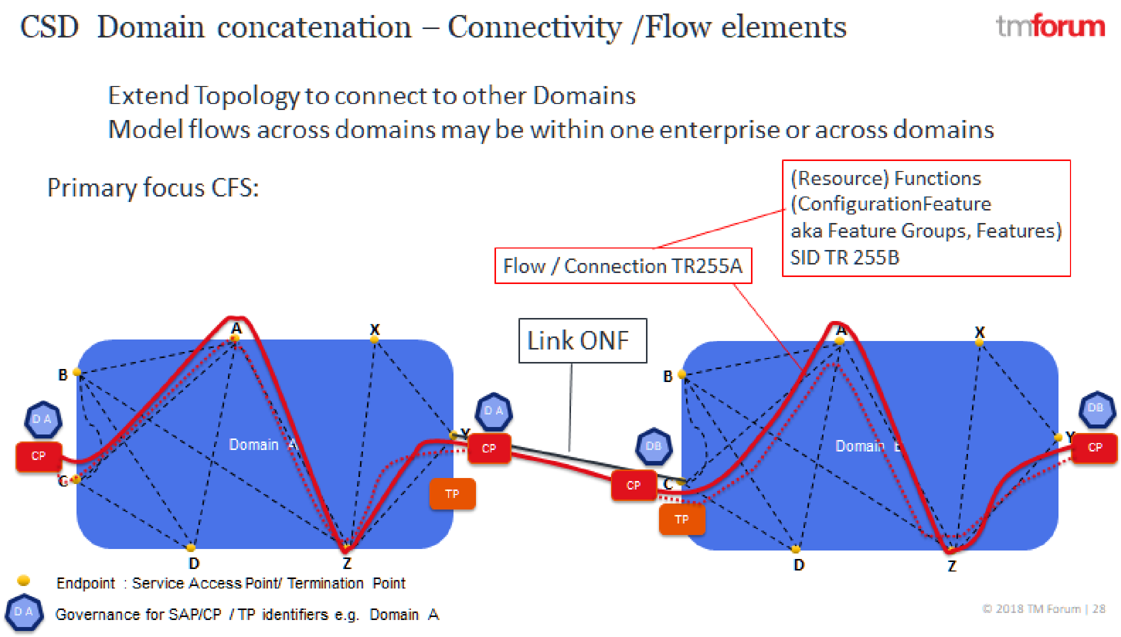
Figure 5.5.5: CSD Domain Concatenation – Flow aspects

This example shows a Connection /Flow (solid red line) established across two Domains.

Logically each Domain sets up an internal connection Flow shown as red dotted connection. By associating the Connection Points (CP) with TP and SAP on either end of the link between Domains an end to end external Connection/Flow can be formed.

However, the CPs on Domain A are managed by Domain A and the CPs on Domain B are managed by Domain B. A further model element Domain C must be introduced as shown in the next section.

5.5.5. Composition of Connectivity Service Domains



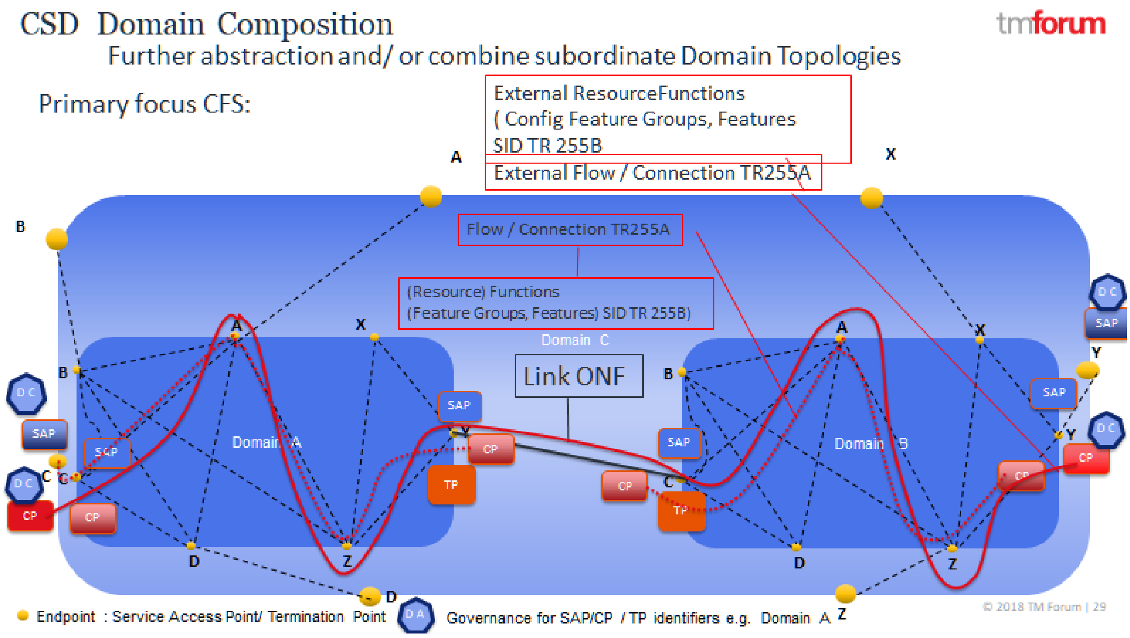
Some Termination Points not shown for simplicity.

Figure 5.5.6: CSD Domain Concatenation – e2e Domain

A composed Domain C is needed to:

- Manage establishment of Connectivity / Flows and assignment of CPs across Domain A and Domain B to support the overall connectivity flow across Domain C.
- Abstract the individual composing Connectivity Service Domain (and Connectivity Service Flow) Functions (Features and Feature Groups) into a simpler set for the e2e Composed Connection /flow.
- Permitting additional behaviors and constraints on the Flows to be defined and enforced by Domain C.
- Hide the view of the enclosed domains A& B from the clients of Domain C as this may change and be reconfigured dynamically.
- Manage the relationship between SAPs and TPs that link component Domain A with Domain B.
- Manage the relationships between the SAPs, TPs and CPs exposed by Domain C to those on Domain A and B by providing ownership of relationships between the SAPs of different CSDs and the associated LTP, Ports and Layer protocol specifications.

Specifically, one option is for Domain C to 'promote' SAP, TP and CP from Domain A and B as shown in the next section.



Internal Connection Flows and some TP not shown for simplicity

Figure 5.5.7: CSD Domain Composition

In this example Domain C 'promotes' SAP, TP and CP from the enclosed domains, and maintains the required relationships. The' external' CPs and SAPs may be simply exposed or may be renamed by Domain C. Termination points can also be exposed in a similar manner, but renaming is more problematic as these have control and data plane implications for the Connectivity Service.

When done in this way the external view of Domain C is self-similar with the external view of Domains A and B which makes composition easier to describe in a machine-readable way and automate provisioning from configuration files (model and data driven).

Realizing Abstracted Connectivity Services using composition and mapping to Resource Facing Services

While exposed technology neutral Connectivity Services (CFS) assist in decoupling ODA Production Function Group capabilities from other ODA function grouping e.g. ODA core Commerce, these capabilities and their functions/features do need to be implemented by resources, whether traditional network appliances or virtualized equivalents.

In the ODA Production Implementation Framework Production Service Management and Orchestration can consume exposed technology specific RFS services exposed by Resource Service Management and Orchestration or resource-based APIs. These RFS services can be either Intent based or detailed/declarative based services as is encountered with legacy implementations.

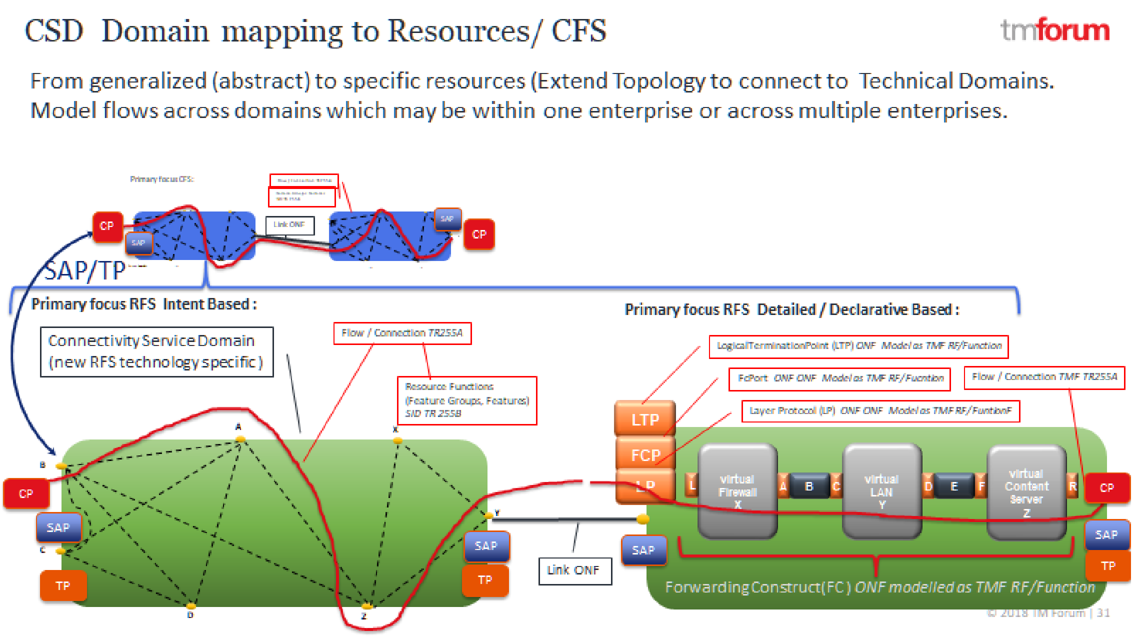


Figure 5.5.8: Intent and declarative patterns for RFSs realizing Connectivity Service CFSs

This diagram shows how one Connectivity Service Domain (CFS - Blue) might be realized by concatenating flows across two technology specific Connectivity Service domains (RFS - Green) whose connectivity management is technology specific and whose scope is determined by supplier.

At least two possibilities exist for implementation of these Technical Domains:

1. A declarative or detailed approach shown on the lower right which is the traditional element management thinking where details and topology of resource functions and resources are exposed - examples include Carrier Ethernet, SD WAN.
2. An intent based approach which will become more prevalent as Connectivity Service solutions from vendors become more autonomic by adopting Closed control Loops, use of AI analytics. Examples being numerous catalyst examples described in IG1128 Dynamic Control Architecture for Managing a Virtualized Eco-System [31].

An intent based Connectivity Service Domain is self-similar to the pattern adopted for Connectivity Service Domain by exposing its capabilities in the form of Functions (Features and Feature Groups) but in this case the Feature are technology specific.

As Connectivity and networks become more virtualized the intent pattern will become dominant in support of model driven automation and it is possible that supplier may create solutions which are technology neutral across multiple generations of technology e.g. 4G and 5G. Thus, vendor RFS solutions may evolve to CFS services which can be composed into a composite CFS exposed from ODA Production by a Connectivity Service /Operational Domain. This will simplify the ODA Production Implementation Framework and deliver opportunities for Vendor to deliver value added innovation that can be rapidly adopted by CSP though NaaS and CFS composition which is intrinsically simpler -due to abstraction - than RFS composition. Noting that the governance of the scope of Operational Domains is by the service provider and Technical Domain by the vendors.

5.6. Data Model proposal for Connectivity Services

Work in progress.

This section addresses how the model in TMF633 Service Catalog [38] is aligned linked to the Connectivity Service Model above. The focus is ServiceSpecification and ServiceCandidate since these define the exposed CFS information in the Service Catalog.

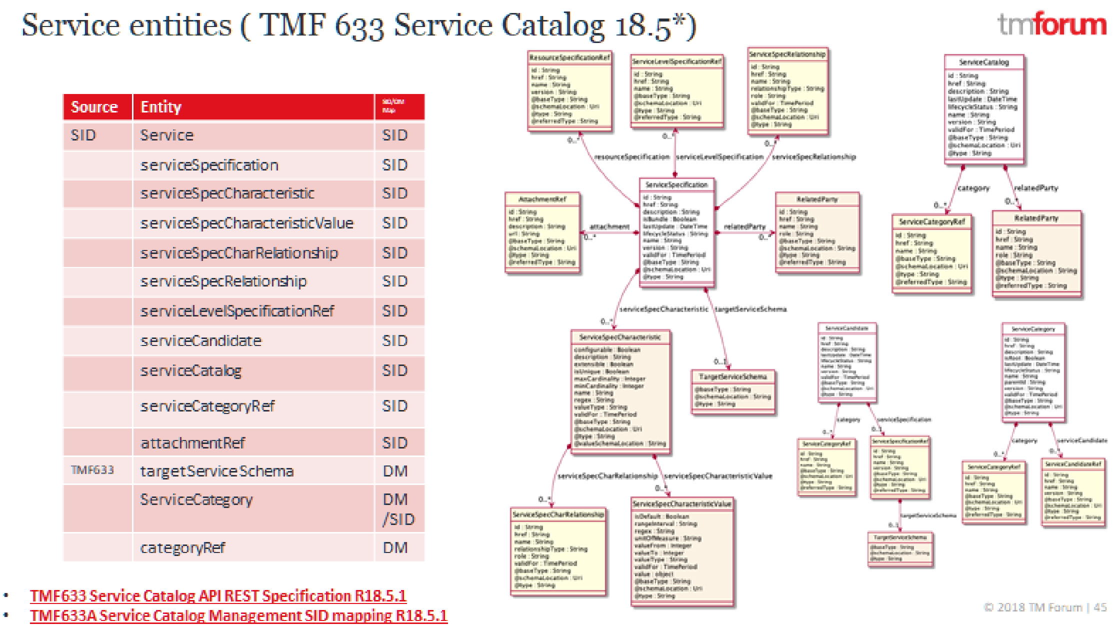


Figure 5.6.1: Summary of service entities in TMF633 Service catalog

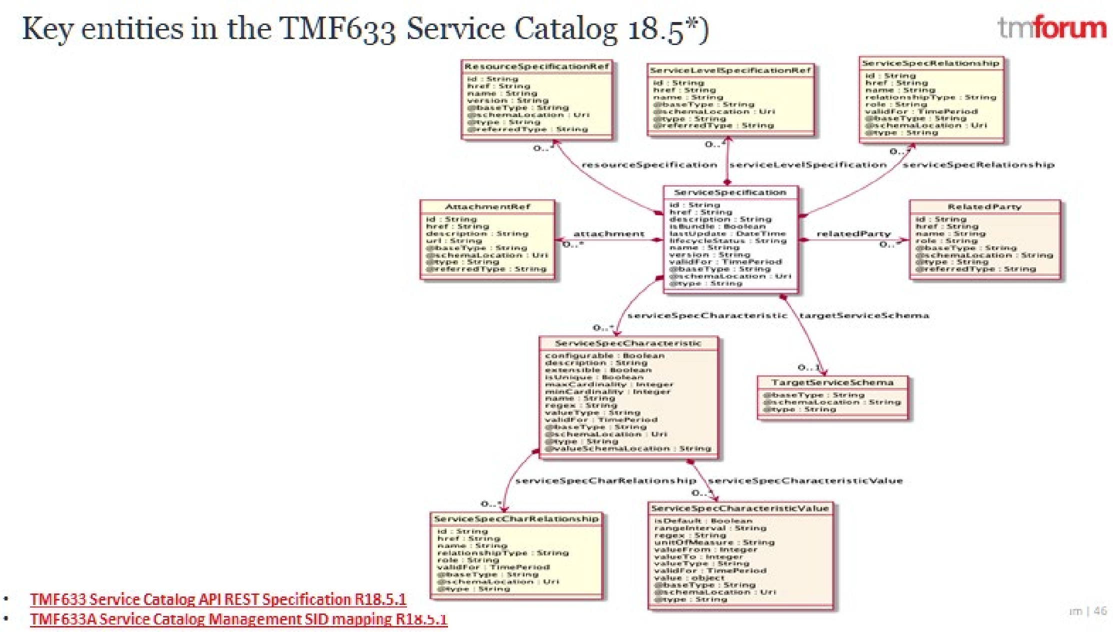


Figure 5.6.2: Key entities in the TMF633 Service Catalog R 18.5.1 UML Data Model

A future challenge is to map Connectivity Service model entities to general Service model captured in a Service Catalog described above. The SID mappings in TMF633A [39] are quite complete. These do not include an explicit formal mapping to Resource Functions and this requirement for Connectivity Services will need to be investigated in a later release. There are technical solutions which can address these requirements using API polymorphic behavior mechanisms.

5.7. Future Work

In the next version of this document the following candidate items have been identified:

- Improving the current version by progressing the SID and API CR that have been identified by this document. This will result in formal SID information models updates which will lead to some editorial changes in this document.
- Definitions of a set of Data Models Schema (in the API team) aligned with SID Connectivity Service models initially for use by the NaaS API Component Suite
- Updates based on following progress on standardization results in 3GPP and GSMA.

Part III: ODA Production Implementation Exemplars

6. Slice Management APIs

An analysis of the API requirements was done in this release through the review of catalyst project use-cases and solutions. The APIs were reviewed in the context of achieving End-to-End slice management. Analysis of the APIs requirements was structured by ODA management function groups:

- Core Commerce APIs (also known used by the BOS project) – The Business layer APIs are outside the scope of this document.
- The ODA Production APIs used between the ODA Core Commerce layer to the ODA Production layer.
- ODA Production Orchestration APIs (i.e. within a domain or across domains in the case of network slice).
- ODA Assurance APIs.

The sections below include details of the more common APIs at the ODA Production layer. It includes descriptions of the APIs and analysis of their data entities. It is worth noting that currently for Service level APIs the SID Service data entities may allow both Customer Facing Services (CFSs) and Resource Facing Sources (RFSs). However, the APIs include aspects of life cycle management that dictate the level of these services. Therefore, the analysis here, includes such aspects.

6.1. From CCM to Production

The overall set of NaaS APIs is not covered here. Yet, as part of the Slice management use cases, several APIs were used to initiate dynamic orchestration of services and slices:

API	Description	Data Model Comments
TMF641 Service Order	The Service Ordering API provides a standardized mechanism for placing a service order with all the necessary order parameters. It allows users to create, update & retrieve Service Orders and manages related notifications.	Main entities: • Order – services are embedded in the order structure • Declaring support of both CFS& RFS using a “Service Type” attribute to distinguish.

API	Description	Data Model Comments
		ABEs: • Service order ABE
TMF645 Service Qualification	The Service Qualification API is used to perform a technical eligibility. It allows to retrieve a list of services that are technically available in the context of the interaction (place, party, service characteristics).	Main entities: • Service Qualification Item that references Services • Not restricted to CFS but focuses on it. A “Category” attribute to distinguish. ABEs: • Customer order ABE

Notes

- These APIs are currently at the Order entity level, where the Customer Facing Service (CFS) is part of the order.
- TMF 641 Service Order has reference implementation. This was produced by Skynet catalyst project (DTW2019) based on IG1176 TOSCA data models.

6.2. ODA Production Orchestration APIs

TM Forum developed a set of APIs related to Service and Resource Orchestration:

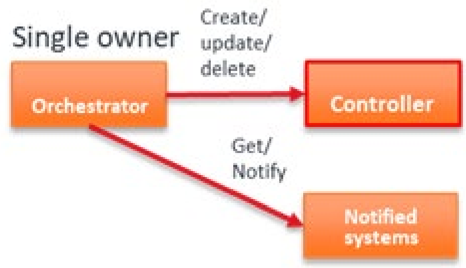
API	Description	Data Model Comments
TMF633 Service Catalogue	This API allows the management of the entire lifecycle of the catalog elements, the consultation of catalog elements during service processes.	Main entities: • CFS oriented (partners, distributors), includes a “Launched” state • Includes a “Type” attribute to distinguish between CFS & RFS ABEs:

API	Description	Data Model Comments
		• Service Specification ABE
TMF634 Resource Catalog	The Resource Catalog Management API REST specification allows the management of the entire lifecycle of the Resource Catalog elements, the consultation of resource catalog elements during several processes such as ordering process, campaign management, and sales management.	Main entities: • Resource (among partners, distributors) • Same state machine as the Service catalogue ABEs: • Resource Catalog ABE • Resource Specification ABE
TMF638 Service Inventory	The Service Inventory API can be used to query the service instances for a customer via Self Service Portal or the Call Centre operator can query the service instances on behalf of the customer while a customer may have a complaint or a query.	Main entities: • Service. A “Category” attribute to distinguish between CFS & RFS. • Operational states ABEs: • Service ABE
TMF639 Resource Inventory	This Resource Inventory API provides a consistent/standardized mechanism to query and manipulate the Resource inventory.	Main entities: • Resource • Specific operations for logical resources and physical resources • Resource-type operations ABEs: • Resource ABE • It is suggested to use the Topology ABE to address the relationship of layers. This is work in progress.

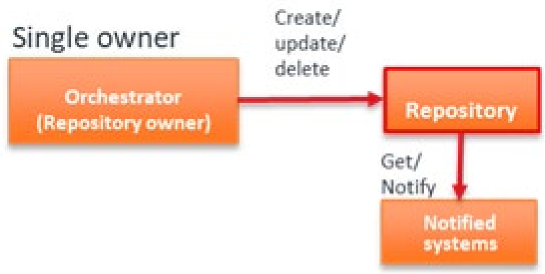
API	Description	Data Model Comments
		Topology ABE will add the relationship of layers.
TMF664 Resource Function Activation & Configuration	This API is to be used for provisioning of components in support of cloud services. The components may be virtualized or physical.	Main entities: • CFS (suggested) • RFS (suggested) • Resource • Operational states ABEs: • Service ABE (suggested) • Resource ABE Note: it is suggested to extend the definition of this API beyond resources
TMF640 Service Activation & Configuration	Service activation and configuration may be the process to support the accomplishment of sales or customer request, normally it requires human handled work. It can be activated or configured by the Activation and Configuration API.	Main entities: • Service (CFS & RFS) ABEs: • Service ABE
Recommended: Slice & sub-slice management **Currently proprietary**	This API allows managing Slice managers (Domain managers) that comply with 3GPP definitions.	Main Entities: • RFS

The Orchestration APIs can be categorized to their functionality:

- Activation & Configuration – This is used between a management/orchestration system to an activating system to perform activation and configuration changes of resources or services.



- Inventory - This is used between a management/orchestration system to an inventory system to perform inventory updates.



- Catalogue - This is used between a management/orchestration system to a catalog system to perform catalog updates that are at the specification level (not at the instance level).

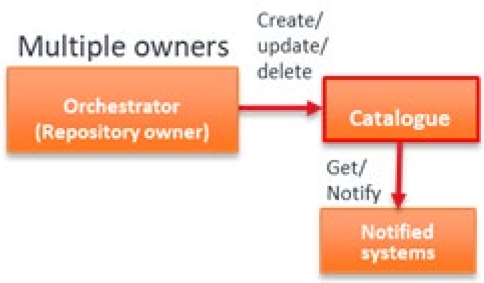


Figure 6.2.1: Orchestration API Pattern Categorization

An example of this pattern is the Telstra NaaS Operational Domains that expose their service definitions through the catalog API (in Telstra NaaS, each domain POST their new service definition to the NaaS master service catalog which then advertise to its subscribers (Domains and BSS/OSS) the new updates).

6.3. ODA Production Assurance APIs

TM Forum has a set of APIs for Assurance:

API	Description	Data Model Comments
TMF653 Service Test	This API provides test procedure for the certain service. The Service Test API provides a standardized mechanism for placing a service test with all of the necessary test parameters.	Main entities: • Service (CFS/RFS) ABEs: • Service Test ABE
TMF656 Service Problem	This SPM API is used for the service providers to manage the service problems in their service area.	Main entities: • A service problem (attached to a service) ABEs: • Service Problem ABE
TMF628 Performance Management	The Performance Management API provides standardized mechanism for performance management such as creation, partial or full update and retrieval of the resources involved in performance management (Measurement Production Job, Measurement Collection Job, and Ad hoc Collection). It allows also notification of events related to performance.	Main entities: • Performance (Common Entities) ABEs: • Performance ABE Note: The Measurement ABE may be used in the future versions
TMF642 Alarm Management	The Alarm Management API provides the standardized client interface to Alarm Management systems for creating, tracking and managing alarms among partners. The interface supports alarm management on both resources and services. The alarmed objects are not restricted to any particular technology or a vendor, so the API can be used in a wide variety of fault management cases.	Main entities: • Alarm – the alarmed object is referenced by value. Therefore, it supports resources, services and any alarmed object ABEs: • Resource Trouble ABE (yet, not necessarily attached to resources)

API	Description	Data Model Comments
TMF649 Performance Thresholds	The Performance Thresholding API provides a standardized client interface to Service and Resource Performance Management Systems for manipulating (create/update/delete) threshold/violation/exception rules. It enables alarms/notifications on exceptions and scheduling of threshold/violation/exception evaluation.	Main entities: • Performance (Common Entities) ABEs: • Performance ABE
TMF623 SLA Management	The SLA API provides a standardized interface for SLA life cycle Management (SLA Negotiation, SLA configuration SLA Activation/enforcement, SLA Operations, SLA violation / consequence handling, SLA reporting) between a Customer and a Service Provider which provides offers (product with attached SLA in its catalog) the customer can discover, browse, trigger and order.	Main entities: • Service Level Agreement (references Service indirectly) ABEs: • Service Level Agreement ABE
TMF657 Service Quality	This API provides Quality Management for the certain service. It enables easy integration of Service Quality Management applications and client applications within the Digital Ecosystem, where Service Quality Management application may reside in one enterprise and client applications may be in multiple other Enterprises.	Main entities: • Service Performance (Service is referenced indirectly) ABEs: • Service Performance ABE
TMF621 Trouble Ticket	The Trouble Ticket API provides a standardized client interface to Trouble Ticket Management Systems for creating, tracking and managing trouble tickets among partners as a result of an issue or problem identified by a customer or another system.	Main entities: • Trouble Ticket (common entities) ABEs: • Trouble Ticket ABE

6.4. Operational APIs for closed-loops

Looking at the various practical use cases of closed-loops, using policies, the typical use cases include the following run-time steps, as depicted in the figure below:

- A monitoring system provides assurance data to the Policy Management system.

- The Policy management system decides what is the required action, yet since it is not the executing system, it passes the decision to the executing system, using a closed-loop event. Sometimes, there may be several levels of decisions, so the decision may be propagated to another Policy Manager that will get it and as a result make a localized decision.
- The Executing system translates the decision to action, orchestration of changes in many cases.

APIs for the following parts are required:

- Passing Assurance events – This can be done using TMF 642, the alarm management API. However, a more general API, an event API may be useful here.
- Applying policies, passing decisions, triggering closed-loops. Such API may pass the following:
    - The policy details: ID, name (closed-loop event name).
    - Details on the triggering event.
    - Details on the decision, such as ID of an action with possible parameters.
    - Orchestration APIs are used. These can be TMF 664 or TMF 640 (as discussed in section ODA Production Orchestration APIs section 6.2).

Additionally, a design time API for defining the policy is required

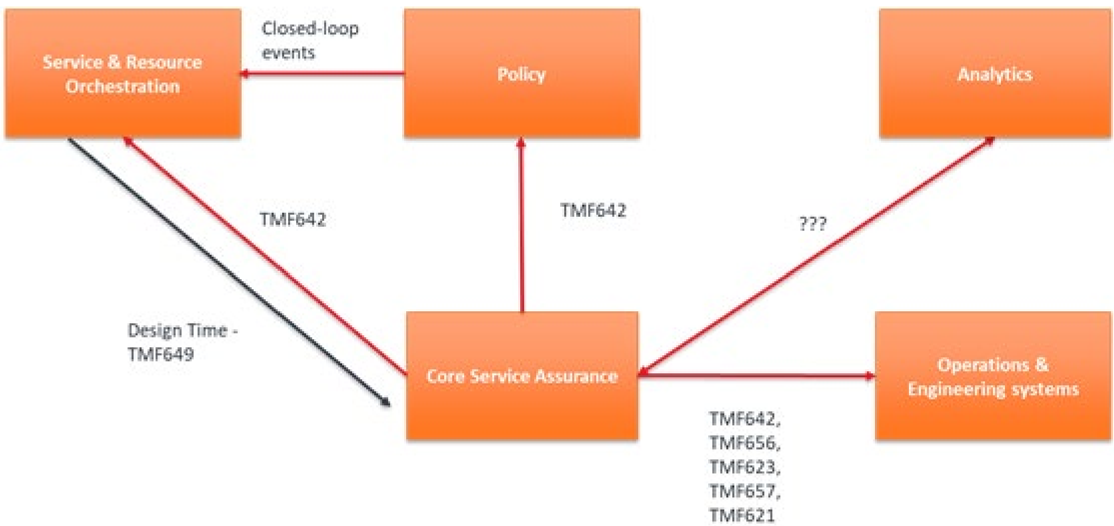


Figure 6.4.1: Open APIs for Closed-Loops

External Analytics may be involved in this process. The core assurance system may need to configure external Analytics tools.

There are several recommendations for related TM Forum work items:

- Using a single closed-loop triggering API (probably the generic Event API).
- Consider a new API for Policy Management.
- Further discuss the recommendation on API for applying Analytics ODA.

API	Description	Data Model Comments
Topology query API	This API provides multi-layers Topology information to ODA Intelligent Management.	Main entities: • Resource • Specific operations for logical resources and physical resources • Resource-type operations ABEs: • Resource ABE • Topology ABE to add the relationship of layers

7. Positioning 3GPP 5G work and ODA

As the ODA Production Architecture Framework is developed it will be desirable to establish a liaison work item with 3GPP on mappings between the detailed ODA Connectivity Service models and the Network Resource Models especially when this is extended to support inter-operator roaming – expected in Release 16.

The goals of this joint activity are to ensure that:

- the mappings from this CFS ODA Connectivity Service model to the RFS and resource level models in 3GPP are practical and validated through Proof of concept Catalysts projects.
- Maximum flexibility is retained for service provider choices for Operational Domain boundaries.
- Decoupling of ODA Core Commerce (BSS) from specific technologies such as 5G.
- Alignment with ODA production and the SID of the recent outline model proposals in3GPP TR 28.805 Study on management aspects of communication services v1.0.0 (Release 16) [12]. This describes the notion of 'Customer Facing Communication Services (CFCS) and Resource Facing Communication (RFCS) Services associated with Communication Service Management Functions and their relationship to Network Management Functions and associated Network Slice as a Service.

Part IV: ODA Production Implementation Exemplars

8. ODA Production Implementation Guidelines

Several examples of implementing ODA Production are included in IG1167 ODA Functional Architecture [1] Section 4.4.6. Deployment Options and Scenario Examples; these and recent catalyst results have been used to distil out ODA Production Implementation Guidelines and Framework.

This ODA Production Implementation Guideline is primarily focused on 'How' to implement ODA Production behind Connectivity Services defined in this guide that are realized through NaaS APIs. It describes:

- **ODA Production Framework** which describes how vendor resources are systematically integrated and managed to expose NaaS Customer Facing Services (CFS - technology neutral) in some cases using Internal Resource Facing Services (RFS - technology specific) See Section 3.1.
- **ODA Production Connectivity Services** - a generalized connectivity model i.e. realization of ODA Production using NaaS based on earlier work in TR255A Connectivity Patterns for Virtualization Management [9] See Section ⍰.
- **5G Network Slice implementation exemplar** that abstracts the 3GPP 5G implementation (for network slice) detail into the ODA NaaS Functions and connectivity model. See Section 1.
- **Validation of ODA Production** functions, architecture and implementation using:
    - Mapping of this generalized connectivity service model to the systems implementations used in six 5G catalyst project at DTW Nice 2018 & 2019
    - Mapping to the 5G systems architecture (3GPP 23.501) Release 15. [10], 3GPP TR 28.801 'Study on management and orchestration of network slicing for next generation network' [11] considering draft proposals for Communication Services in 3GPP TR 28.805 [12].

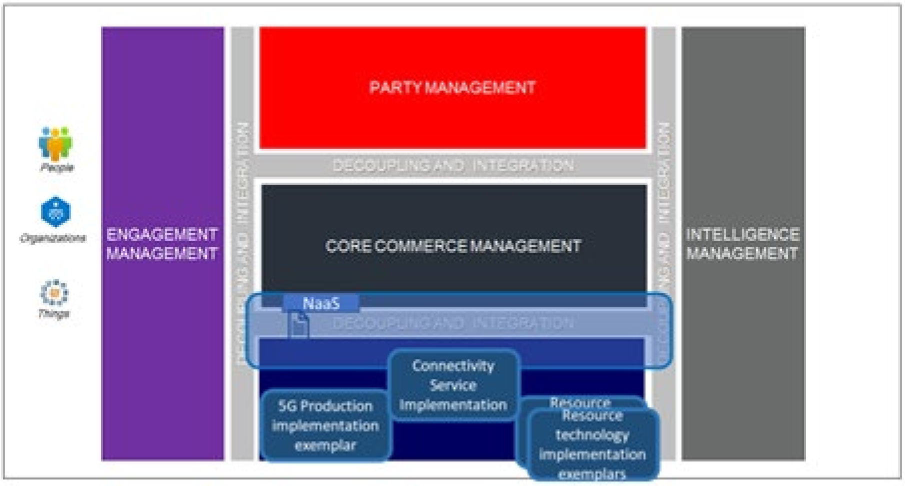


Figure 6.4.1: Exposed Production Connectivity Service and implementation exemplars

Where the exposed connectivity services for decoupling are covered by the NaaS concept, supported by a Connectivity Service model, and the validation of 5G Production implementations are covered by a series of exemplar implementation functions and catalyst architectures in Part II. Other connectivity service examples are in VPN TR 255A & B [13], [14], Firewalls TR255C [15], and vCDN TR 245 [16].

9. TM Forum catalysts – Slice management Architecture & Use Cases

9.1. ODA Production examples

9.1.1. ODA functional Architecture examples

As mentioned earlier there are several examples of ODA Production IG1167 ODA Functional Architecture [1] Section 4.4.6. Deployment Options and Scenario Examples.

9.2. Minimum ODA Production implementation example

9.2.1. Architecture example

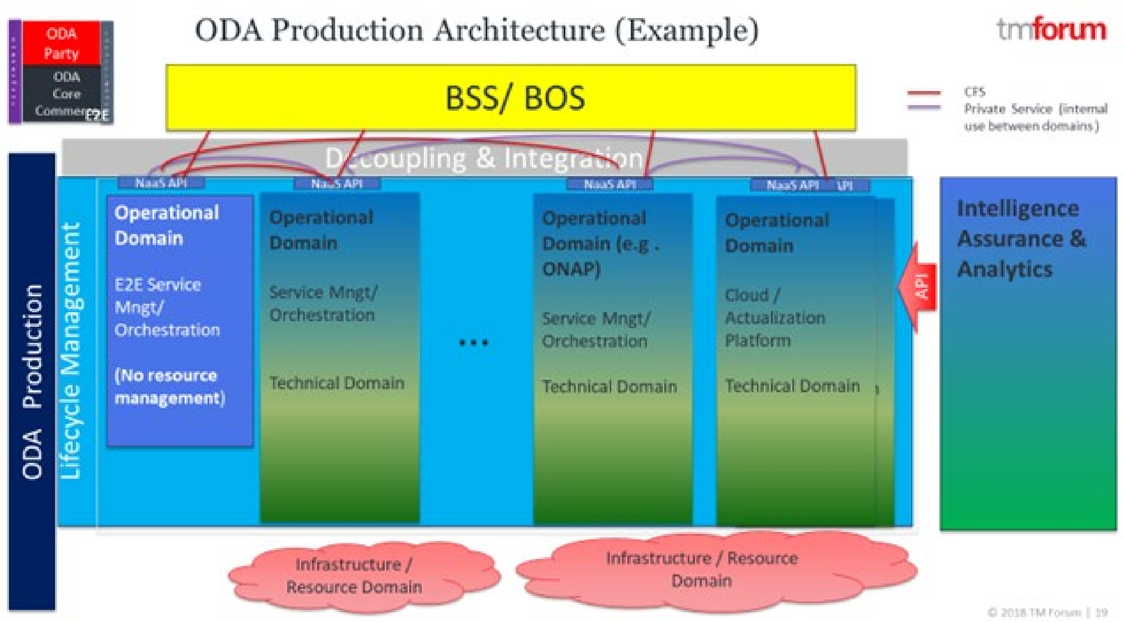


Figure 9.2.1: Minimum ODA Production implementation example

This example of an implementation of ODA Production leverages the API economy concept by exposing network capabilities as a Network as a Service. These can be referred to as CFS for services reused as part of customer Products and private services for services used between operational domains. Operational Domains can manage technical domain(s) i.e. resource level both physical and virtual. Operational Domain(s) can also expose composite services (CFS) from a composition/orchestration and management of NaaS (CFS and/or private) from other domains and may not directly manage any resources (also referred as E2E service management or cross-domain management).

The benefits of this implementation include:

- Usage of NaaS API component suite for all network technologies to communicate with the BSS/BOS (or CCM domain) eliminating custom integration time and costs for all NaaS.

- Reuse of technology domain exposed services (NaaS) in multiple products (e.g. Firewall as a Service used in Managed Firewall product as well as in Internet Security product and any other products using a firewall) hence reducing the duplication and project integration time and costs.
- Simplification of BSS/OSS as there is no need for these systems to know anything about "How" the service is created (i.e. no resource level knowledge) reducing and/or eliminating impact of any resource level changes (e.g. change of supplier or release update).
- Responsibility of every operational domain is to support the complete lifecycle of their exposed NaaS including performing Closed-Loop assurance (e.g. scaling and healing) of the services and/or resources used. Any problem that can't be resolved within the domain is notified to the consumer of that service which can then be acted upon at the service level (e.g. E2E Service Management Operational Domain activates a Mobile backup service to replace a fixed access service being down).
- Migration or refactorization of Operational Domain can be complex e.g. representing multiple technologies, or can expose only a few NaaS, the decision lies within the CSP and may change as they migrate and start exposing their network capabilities as NaaS and form the contract with the BSS/BOS through the NaaS APIs.
- Increase automation can be delivered the more NaaS exposes its capabilities. For example, during the design NaaS can expose the attributes to be used for the service qualification, the service configuration, the SLA(s) that can be offered, the type of tests that can used, etc., and CCM applications can extract that information in an automated fashion.

9.3. 5G Intelligent Service Planning and Optimization Catalyst

9.3.1. Catalyst Architecture

The 5G Intelligent Service Planning & Optimization catalyst included the systems as follows:

- Digital Enablement Platform (an Ordering system, Wipro)
- Service & Resource Orchestrator (Ericsson)
- Adaptive Inventory (Ericsson)
- Planning Analytics (Aria)
- Technical Domain Orchestrators (Ericsson)
    - A Cloud Orchestrator
    - A WAN Orchestrator
- Service Assurance& Analytics (MYCOM-OSI)

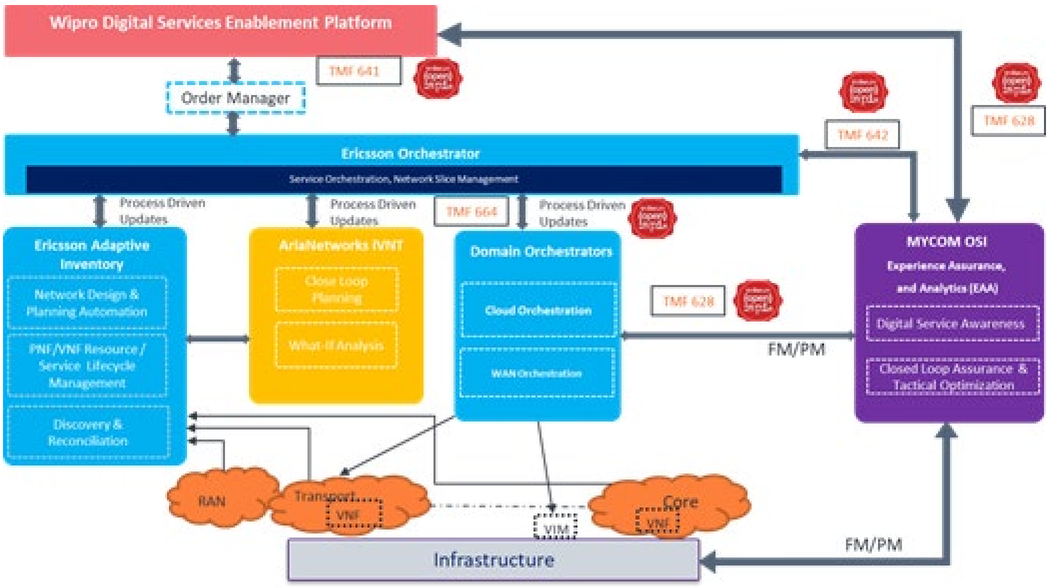


Figure 9.3.1: 5G Intelligent Service Planning and Optimization Catalyst

9.3.2. Catalyst Use Cases and APIs

The 5G Intelligent Service Planning & Optimization catalyst demonstrated several use cases:

- Network Slice Design
- Network Slice Activation
- Reactive & Proactive closed-loop
- Customized Customer Experience (Customer level)

For implementing these use cases the following APIs were used:

- TMF641 – Order management
- TMF628 – Performance Management
- TMF642 – Alarm Management
- TMF664 – Resource Function Activation & Configuration. This was used between Ericsson Service & Resource Orchestrator to the Domain Orchestrators
- Ericsson specific interfaces among the Ericsson components
- Analytics specific API between the Orchestrator to the Planning Analytics

9.4. 5G Intelligent Service Operations Catalyst

9.4.1. Catalyst Architecture

The 5G Intelligent Service Operations catalyst included the systems as follows:

- Order Management (NetCracker)
- Service Orchestrator (NetCracker)
- Policy Management (Tech Mahindra)

- Service Assurance& Analytics (TEOCO)
- 5G Lab (BT)

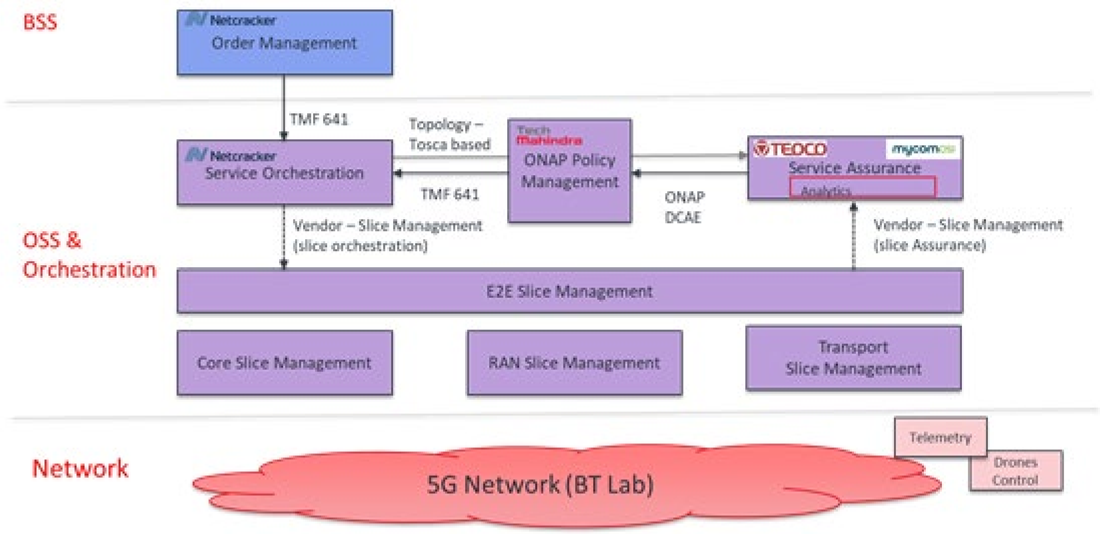


Figure 9.4.1: 5G Intelligent Service Operations Catalyst

9.4.2. Catalyst Use Cases and APIs

Two use cases of operational closed-loop were demonstrated, showing slice SLA preservation for different kinds of services (Drone based broadcasting, Riders Telemetry).

For implementing these use cases the following APIs were used:

- TMF641 – Order management (in two different contexts)
- ONAP Policy Management API (REST)
- Network Vendor slice management APIs (Provisioning, Assurance)
- TOSCA based Service/Resource API was suggested

9.5. Blade Runner Catalyst

9.5.1. Catalyst Architecture

The Blade Runner catalyst included the systems as follows:

- Customer & Partner Management – CRM, Order Management, Billing (BearingPoint Infonova BSS platform)
- End to End Customer Facing Service Management (Rift)
- Domain Specific Resources Facing Service Management (ONAP based – Infosys, Riverbed)
- Service Assurance (CNEX recently acquired by Ericsson)
- Technology Specific Resource Management (ONAP based – Exfo)

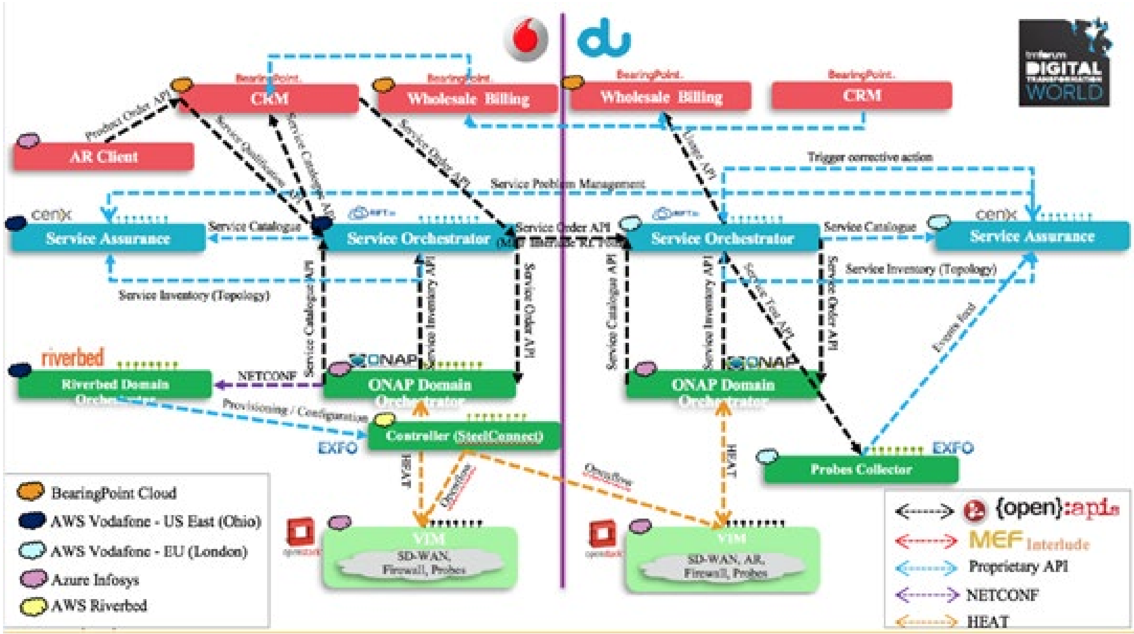


Figure 9.5.1: Blade Runner Catalyst Architecture

9.5.2. Catalyst Use Cases and APIs

The Blade Runner catalyst demonstrated several use cases:

- Service Onboarding
- Service Provisioning & Activation
- Operational closed-loop

For implementing these use cases the following APIs were used:

- Service Order API (x2)
- Service Catalogue API (x2)
- Service Inventory API
- Product Order API
- Service Test API
- Service Qualification API
- Service Problem API
- Some of the APIs were TM Forum APIs, some ONAP APIs and others proprietary.

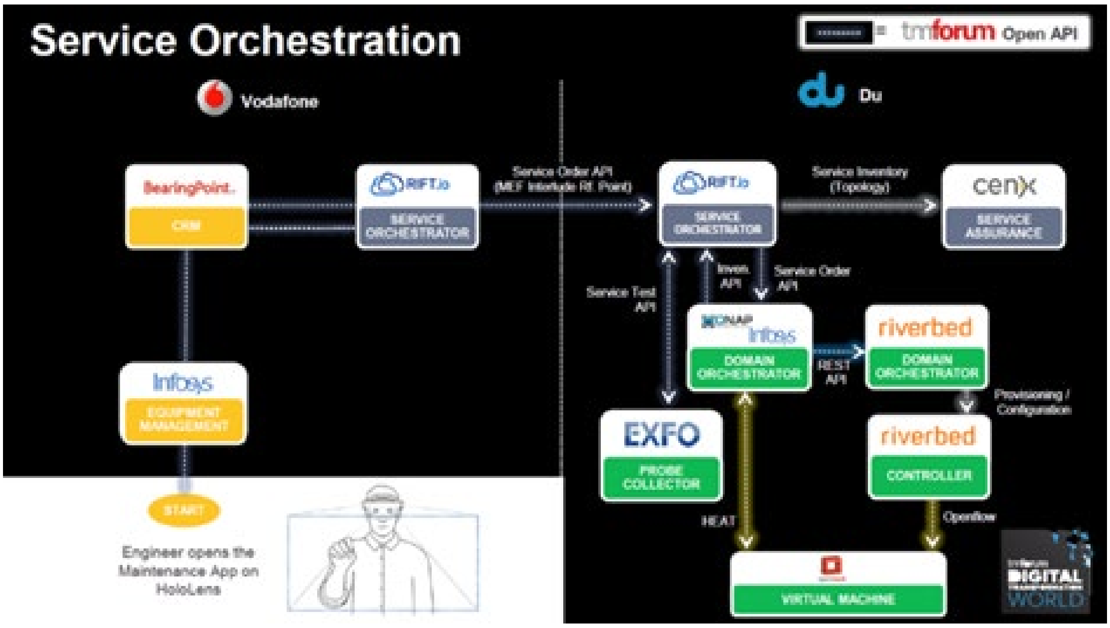


Figure 9.5.2: Blade Runner Catalyst Service Orchestration

9.6. Skynet Catalyst - 5G models

9.6.1. Catalyst use cases

Skynet (DTW 2019) looked at advancing the Blade Runner work by developing a multi-country multi-provider multi-organization health epidemic management use case.

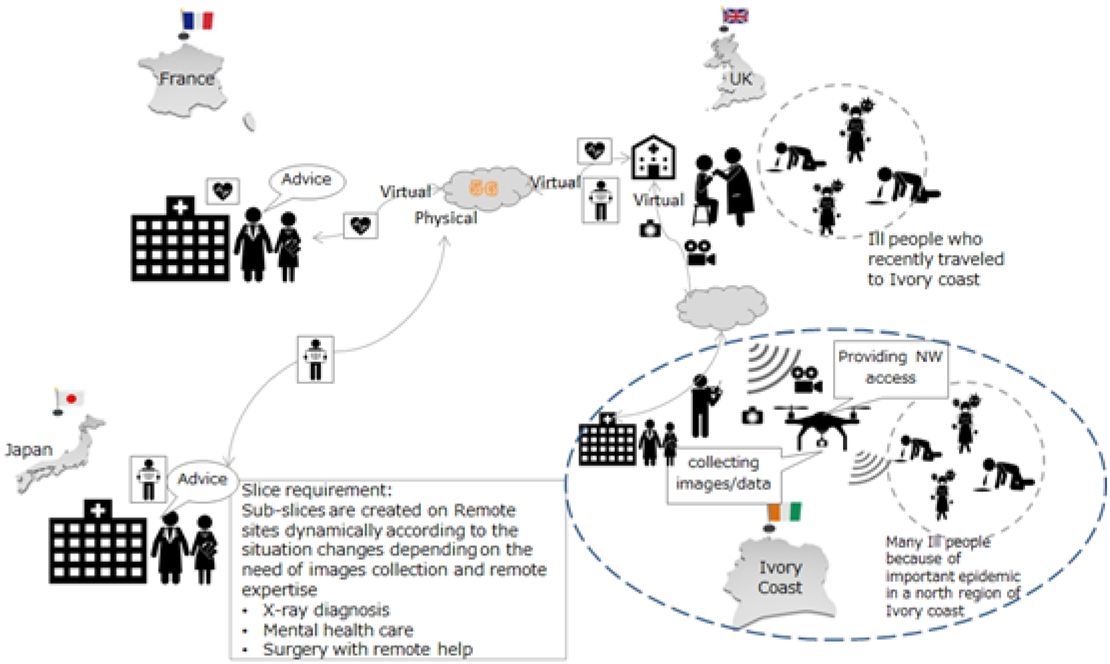


Figure 9.6.1: Skynet Remote Health Care Business Scenario

Scenario Context:

Several persons have reported sick in UK hospital immediately after a voyage to Ivory Coast. World Health Organization (WHO) is involved to ensure the disease is contained. UK Hospital needs to collect some samples from Ivory Coast and then consult doctors in France and Japan to quickly identify root cause of the problem and control the spread of disease.

9.6.2. Catalyst 5G architecture and models

In Skynet Catalyst, TOSCA has been used to describe a hybrid network connectivity that is composed of virtual network and non-virtual network. Especially, NTT has explored about the needed node types and relation types for the non-virtual part of the hybrid connectivity. The hybrid network TOSCA generation and orchestration based on TOSCA are mainly implemented in UK-JP connectivity service ( ).

TOSCA is used to model the connectivity in multiple layers. In the Service Orchestrator (SO) layer, TOSCA is used to model the abstract/'intent' the connectivity requirements, including the source/destination point of the connectivity and the quality of the connectivity. In the Domain Orchestrator (DO) layer, TOSCA is used to specify the physical/virtual infrastructures, the logical resources and their relationships needed to realize the connectivity service.

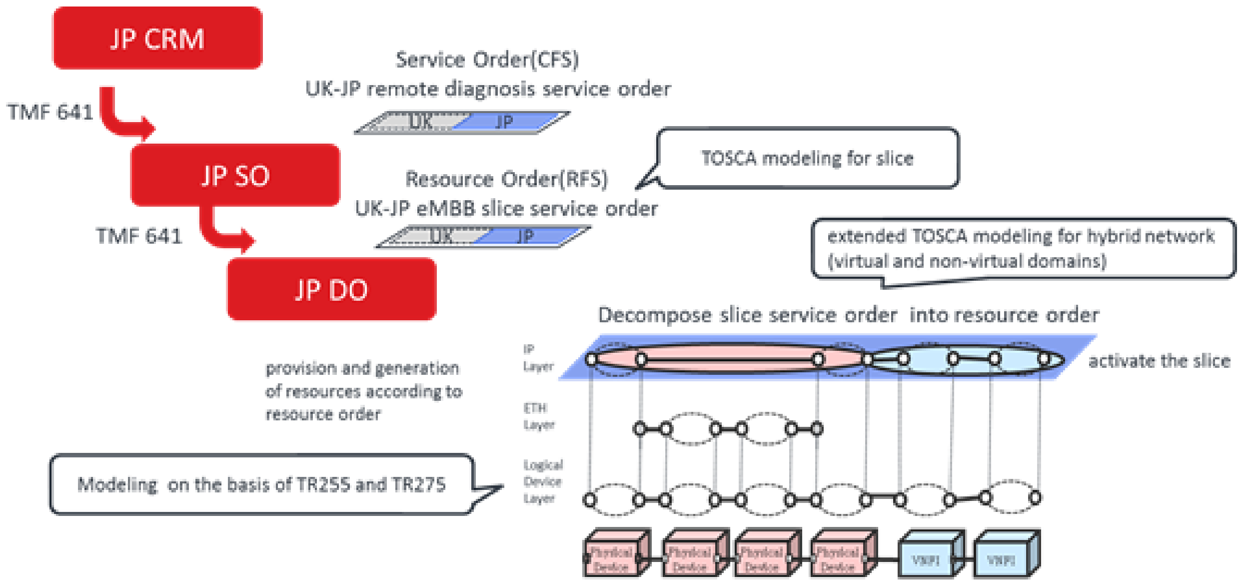


Figure 9.6.2Hybrid Connectivity modelling and activation

What is interesting is that the service order uses a generalized service 'UK -JP remote diagnostics order' at CFS level whereas the slices resource order is at an RFS level 'UK-JP eMBB Slice Service Order' which is consistent with the ODA Production Implementation architecture proposal.

10. References

[1] IG1167 ODA Functional Architecture R19.0.1

[2] IG1141 Procurement and Onboarding Suite R17.5.1

[3] IG1176 TOSCA Guide for Model-Driven Automation R19.0.0

[4] TMF909 NaaS Component Suite to be published. Add reference when formally published.

[5 Open Digital Architecture White Paper

[6] IG1166 ODA Architecture Vision R18.0.0

[7] GB998 Open Digital Architecture (ODA) Concepts & Principles R18.0.1

[8] IG1170 Network as a Service (NaaS) Package R18.0.0 NaaS package

[9] TR255A Connectivity Patterns for Virtualization Management R17.5.0

[10] 3GPP TS 23.501: "System Architecture for the 5G System" V16.1.0 (2019-06)

[11] 3GPP TR 28.801: "Study on management and orchestration of network slicing for next generation network" V15.1.0 (2018-01)

[12] 3GPP TR 28.805 Study on management aspects of communication services v1.0.0 (2019-05) (Release 16)

[13] TR255A Connectivity Patterns for Virtualization Management R17.5.0

[14] TR255B Specification Requirements for Resource Functions R17.5.0

[15] TR255C TOSCA Representation of Virtual Firewall R17.5.0

[16] TR245 Network Service Analysis: Virtual Content Delivery Network (vCDN) R15.0.0

[17] TR262 Management Platform Blueprint and Application to Hybrid Infrastructure R17.5.1

[18] TR276 Monetizing 5G R18.0.1

[19] GSMA Generic Slice Template NG.116 CR1001 Generic Network Slice Template

[20] IG1177 ODA Intelligence Management Implementation Guide R18.5.1

[21] IG1171 ODA Component Definition R18.5.1

[22] IG1174 Model-Driven Design of Management Interfaces for ODA Component R18.5.1

[23] ETSI GS NFV-MAN 001

[24] TR263F Relationships Between the Concepts of Domain and Platform R18.0.1

[25] 3GPP 28.541 Network Resource Model

[26] 3GPP TS 28.533 V15.0.0 (2018-09) Management and orchestration; Architecture framework (Release 15)

[27] 3GPP TS 28.530 V15.1.0 (2018-09) Management and orchestration; Concepts, use cases and requirements (Release 15)

[28] GB922 Logical and Compound Resource Computing and Software R18.0.1

[29] IG1165 NaaS Network Model and Termination Point Composition Best Practice R18.0.0

[30] ONF TR-512 Core information Model

[31] IG1128 Dynamic Control Architecture for Managing a Virtualized Eco-System R16.0.1

[32] TM Forum -TR229A Chapter 4. Network Slicing User Stories R17.0.1

[33] ONF TR 526 'Applying SDN Architecture to 5G Slicing

[34] IG1157 Digital Platform Reference Architecture Concepts and Principles R17.0.1

[35] IG1194 Focus on Services not Slices R19.5

[36] TMF664 Resource Function Activation and Configuration R19.5.0

[37] TR275 Core Networking Resources Business Entities R18.5.1

[38] TMF633 Service Catalog Management API REST Specification R18.5.1

[39] TMF633A Service Catalog Management API Mapping R18.5.1

[40] TR255 Resource Function Activation and Configuration Suite R17.5.0

[41] GB922 Resource v19.5.1 Section 3.2.11 Resource Function: Support for Virtualization and Resource Abstraction pages 38ff was GB922 Logical and Compound Resource Computing and Software R19.0.1 (LCRCS)

[42] GB922 Resource v19.5.1 Section 3.2.15 Use of ConfigurationFeature to support ResourceFunction

[43] GB999 ODA Production Implementation Guidelines v3.0.1

[44] GB922 Common v19.5.1 Section 2.16 Configuration and Profiling ABE Pages 394ff Was GB922 Configuration R14.5.1

[45] R19.5 was TMF664 Resource Function Activation and Configuration API REST Specification R17.0.1

[46] TMF071 ODA Terminology R19.0.1

[48] GB922 Models GB922 Information Framework Models Suite R19.0

[49] GB922 Resource v19.5.1 Sec 3.2.16. ResourceFunctionSpec and related entities 50ff

[50] GB922 Resource Domain Business Entities R19.5

11. Appendix A: Network Slice Management Requirements

Following is a set of requirements on Network Slicing management, covering various management aspects have been established from service providers.

Requirement Category	Requirement Name	Requirement	Management Function type
General (cross-functions)			
	Slice types	Various slice types should be supported: uRLLC, eMBB, etc. Various slice types (SST) and ability for SP defined slice sub-types should also be supported.	CFS & RFS both
	Cross-Domain	It should be possible to manage slices E2E across domains: Core, Edge, RAN, Cloud	
	Multi-carrier	Multi-carrier slice management should be supported (for example: charging, domain specific).	
Service Order, Orchestration & Inventory			
	Service Design	Slices should be supported as part of the Service Design, Service Catalog & order Template	CFS
	Slice Selection	It should be possible to select Slices and Subnetworks as	RFS, CFS-RFS interfaces

Requirement Category	Requirement Name	Requirement	Management Function type
		part of the Service design	
	Slice LCM	Full slices LCM (Life Cycle management) should be supported: instantiation, configuration, enablement de-allocation or removal.	RFS
	Slice Path Inventory	Slice Path Inventory information should be managed: network path selection, VNF, PNF resources, Connectivity choices, domains	RFS, Infrastructure
	Slice Virtual Resources	It should be possible to assign and orchestrate the virtual resources of a slice: VNFs, Microservices, Virtual networks, RAN resources	RFS/Resource order and management
	MEC Support	Application hosting and onboarding at MEC should be supported for the slice when applicable.	RFS: Resource Management/Orchestration
	Load Balancing	It should be possible to load balance or re-allocate network and compute resources as	RFS: Resource Management

Requirement Category	Requirement Name	Requirement	Management Function type
		required for SLA maintenance.	
Service Assurance			
	Slice SLA	The Slice SLA should be monitored	CFS: Assurance
	Performance Management	Performance measurements for the slice various components should be available from the different domains: RAN, MEC, Core and Virtual resources. This includes: gNB, CPE WAN interfaces, VNF performance data, VPN and network element performance.	RFS: Performance measurement of the network and compute resources
	Fast Collection & Processing	Service Assurance should provide processed measurements fast enough to answer the needs of critical applications	CFS & RFS
	SA at the Edge	Intelligent feedback processing at the edge or far edge is required	RFS, CFS
Interfaces			
	Slice support	All functional interfaces should support slices: customer apps, BSS, RAN, MEC, Orch stack, Cloud	RFS (domain and resource API)

Requirement Category	Requirement Name	Requirement	Management Function type
	Multi-carrier APIs	Carrier to Carrier API on slices at different domains should be supported	CFS, partner management
Billing/Charging		OSS to BSS API for dynamic slice charging is required	Product offer management, CFS

12. Appendix B: ODA Production related JIRA Change Requests

In the development of this document and a companion document IG1176 TOSCA Guide for Model-Driven Automation R19.0.0 several enhancements and examples have been identified that are needed to the SID in R19-5. In particular to move the model automation of GB999 and IG 1176 fully aligned with SID models

These are available to project participants at:

1. FP-836 Create classes representing specialization of the general “Function” concept
2. FP-835 Create “Function” as a general concept
3. FP-834 LogicalPhysicalResource redundant with ResourceRelationship
4. FP-833 Changes to the description of HostingPlatformRequirementSpec
5. FP-832 Clarify definition of “SoftwareResourceSpecifict
6. FP-838Revisit the concept of Graph
7. FP-837Deprecate “Feature” and “FeatureGroup”
8. ODA-162 Create SID aligned ODA /5G production examples for GB222 Service Overview and Logical and Compound Resource
9. FP-856 GB922 Service Overview) Enhance SID Service Overview with ODA /5G Connectivity Service example
10. FP-857 GB922 Logical and compound Resource Software and Compute Enhance with ODA /5G Connectivity Service examples

All of these have been addressed and resolved. The graph model changes haven been incorporated in to TMF 664 and will need some further work to support a proposed Topology Discovery API for Big Data and AI analytics of transports networks, including 5G Network Slices.

13. Appendix C: ODA Production GB922 Information Framework Model (new)-Informative

13.1. Introduction and Background

The focus of the Modeling update to GB999 between the R19.0 and R 19.5 was to review the Connectivity Service Model in Section 5 and establish and confirm the precise relationship of those concepts with the formal model elements in the Information Framework - mainly in GB922 Resource Domain Business Entities [50] formerly GB922 Logical and Compound Resource Computing and Software [18]; and concurrently address JIRA change proposals identified in R19 and recorded in Section 12 Appendix B: ODA Production related JIRA Change Requests.

This has resulted in agreements of which the most significant being (JIRA Items in Braces):

- Retain concept of ResourceFunctions and do not introduce a generalized Function (FX-835, FX836)
- Use the existing concept of ConfigurationFeature to model Features and Feature Groups in the Information Framework (FP-837.).
  TR255 uses equivalent terms: Feature & Feature Group.
  TMF664 Resource Function Activation and Configuration API maps ConfigurationFeature to Feature object in which 'is Bundle' attribute is used to support composition.
  Also described in TR 255B R17.5.1 pg. 36
- A study on how Topology should be captured in the Information Framework model.
  The Topology discussion is based on JIRA request in ODA 310 about how to create a topology model to support a Topology API that discovers Topology over multiple transport layers and concatenated networks.
  The exact best modelling practice for Topology is currently under development in the Information Frameworks team for R19-5 for incorporation in GB999 R20.0.

These Information Model changes, and some subsidiary issues had to be resolved prior to presenting a formal Connectivity Service Model in this document.

Working through the JIRA changes has been a substantial activity as each change proposal required considerable research and discussion. This has resulted in an updated GB922 Resource Domain Business Entities R19.5. [50] (formerly GB9222 Logical and Compound Resource Computing and Software [18]) with additional examples and clarification of the models and their use.

13.2. Connectivity Service Model

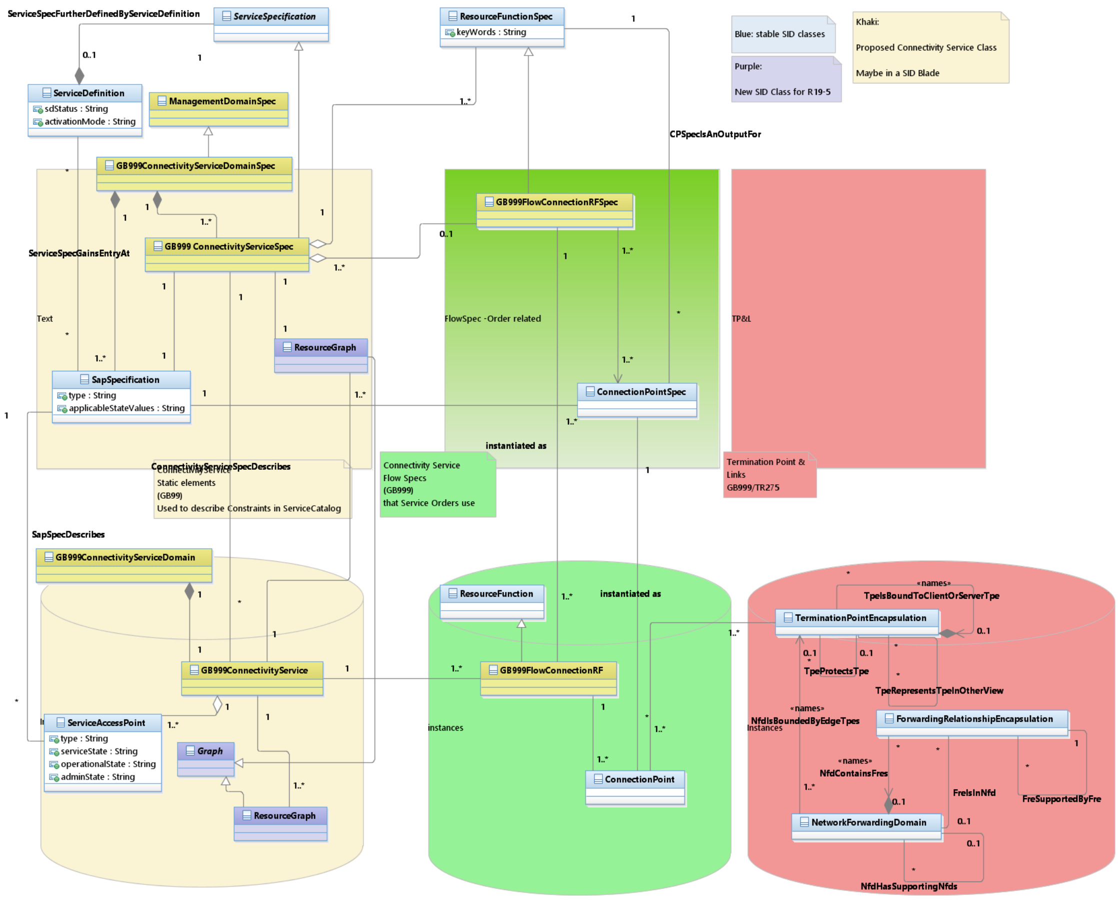


Figure 13.2.1: Overview of Connectivity Service Model

This Model satisfies the core requirements listed in Section 5.1.

A goal in evolving this R19-5 model from R19 has been to simplify it and to make it easier to adopt and use; While also having the necessary extensibility to support:

- Connectivity Service mapping to specific connectivity technology.
- Connectivity that goes beyond Point to Point and Point to Multipoint connectivity, and includes internal processing and storage nodes e.g. Node Computing and vCDN Services. as described in TR 255 [13]

The key elements of the model structure illustrated in Fig 13.2.1 above are:

- Common SID Entities and Specifications used to underpin the model are shown at the top.
- The left-hand groupings in Yellow define the Connectivity Service Domain and Connectivity Service Static Model elements.
- The center groupings in Green define the Flow/Connection model elements of the Connectivity Services (ResourceFunction based).

- The right-hand groupings in Red define Termination Points & Links model elements. These Termination Point Elements at the edge of Connectivity Service Domain(CSD) are needed to:
    - Support concatenation of CSDs.
    - Link to industry transport models, specifically those defined in TR275 as the touch points agreed with the Open Network Foundation (TR512 and TR513) that provide the model bridge to other networking organization models such as those from MEF, BBF and others.

The colored grouping are separated into those that model 'Specifications' (Box) and those that model 'entities' (Drum.)

The absence of entities in the Specification part for Termination Points & Links is because these are part of the Common Information Model agreement with ONF (defined in TR275 and ONF TR512), and solely instances from those models need to be linked into this model. This also helps simplify the model.

This separation of 'Specifications' and 'Entities' is needed for the follow purposes:

- **Service Catalogs:**
    - Any Connectivity Service Domain is required to publish in a catalog the Connectivity Service capabilities that they have. This is formed of two parts
        - **The static elements** of the service defining the Service Access Points (SAP) and relationships between them. Typically, this would be captured in a product /service brochure but for lifecycle and operational automation this data has to be in a machine-readable form.
            - Examples include: An MPLS VPN available in Chicago, Dallas London and Paris and that each group of SAPs has a service topology model such as:
                - Any to Any (AN).
                - Multipoint where one SAP acts as an aggregation point for the other SAPs.
                - Many other possibilities identified in TR255.
            - The model elements in the Yellow Box is the Specification or metadata for the instances of Connectivity Services defined in the Yellow Drum that are held in a catalog i.e .the specific Connectivity Services like a MPLS VPN.
        - The **Dynamic Flow/ Connection elements** (Green Box) specify types of Flows (modelled as ReourceFucntions) that can be instantiated. E.g. Flow Connection definitions such as a multipoint connection for vCPE.
        - Instances of Flows are held in a Service Inventory.
- **Ordering and maintaining Connections/Flows for tenants:**
    - Ordering processes use the Connectivity Service definitions (Yellow Drum) typically stored in a Service Catalog, to establish constraints on ordering i.e. which SAPs can support the Connectivity Service and any topology constraints

  - Once Connectivity Service constraints for a CSD are known Ordering Processes can request an instance of a Connectivity Service specifying the required Flow/Connection definition (specified in the Green Box) that is typically also stored in the Service Catalog.
- **Concatenation and composition of networks and flows:**
  - Requirements in Section 5.1.2 are to allow Connectivity Services Domains to be concatenated to allow e2e Connectivity across multiple CSD and for their composition into a higher level e2e Connectivity Service Domain.
  - To concatenate and compose a self similar model is needed at all level of abstraction for
    - connection points at the edge of CSD that expose the Flows
    - Termination Points that allow physical and logical interconnection of CSD.
  - ONF models are clearly the favoured mechanism which is why the model support Termination Point Encapsulation as the Information Framework representation and touch point to ONF concepts of
    - ONF (Logical)Termination Point.
    - ONF Network Forwarding Domain equivalent of CSD.
    - ForwardingRelationshipEncapsulation to define the protocol level related attributes of the Termination Point.
- **Extensibility**
  - This core model can be extended to support specific Connectivity Services and Flows. Currently this can be achieved by injecting a specific Connective Service definition into schemas that extend Open APIs using the API polymorphic pattern.
  - There is also a case for adding these specific Connectivity Service statically to the Information Model to permit automation in going from UML models to REST API specifications. This approach is for further study.

Formal modeling of Logical Resources defined in GB922 Resource Domain Business Entities section 3.2.23 [50] (formerly GB922 Logical and Compound Resource Computing and Software [28]) provides the relationship between the Connectivity Service model above and ETSI NFV virtualization concepts.

This snapshot is subject to further evolution based on the changes developed for Information Framework some of which are not finalized.

14. Appendix D Modelling Terminology across GB999, TR255 Suite, Information Framework and Open APIs

In developing this document GB999 and TR255[36] and TMF 644 Open API [45] several mappings and terminology issues were identified across ODA ZOOM, Information Framework and OPEN APIs. These have been mostly resolved by studies in the Information Framework team and the production of two revised guidelines:

GB922 Resource v19.5.1 [41] [42] [50]

GB922 Common v19.5.1 [44]

Currently the terms selected to describe the same concept differ across: ODA (ODA Production Framework and TR255), the Information Framework(SID) and the Open APIs notably TMF 664 Resource Function Activation and Configuration.

Differences in choices of terms are driven by the context and the abstraction relevant to those views for example Information models need terms that are generalized and based on entity Specification patterns whereas APIs are implementations and solely need entity definitions in their Data /Resource Models.

Changing any of these terms is a complex and time-consuming task and for the immediate timeframe it is more practical to provide a mapping table.

The requirements driving the need to capture these mappings include:

- ResourceFunction:
- Models need to support both Specs and entities - some documents are a little imprecise on this point
- APIs
- Configuration
    - Configuration and ConfigurationSpec need to be a first class concept in these models.
- Resource Function introduced to support intent based management which require a separation of the Function from the resources that implement it - needed for virtualised and hybrid solutions.
- (Traditional network resources were modelled with RFS Spec ResourceSpecs and associated CharacteristicSpecs
- Feature and Feature group introduced in TR255 to support Resource Function (these are all entities / instances) and are configuration parameters intended to support intent based management and affect behaviour of Resource Functions)
- SID GB922 uses ConfigurationFeatureSpec rather than Feature and Feature Group
- ConfigurationFeature Spec is a spec whereas Feature and FeatureGroups in TR255 are entities
- TMF 664 uses FeatureSpec/ Feature but does not use group - it uses a tag "is Bundled".

ODA/ Frameworx/ TR255	SID GB922	TMF 664 Resource Function Activation and Configuration API
Instance Concepts		
Resource Function {1} ***Defn***: A Resource Function (RF) specifies a function as a behavior to transform inputs of any nature into outputs of any nature independently from the way it is provided. It is typically created by a function designer who may not have specific knowledge on realization architecture (for example using English text explanations with diagrams, as in RFC standards, or preferably a machine interpretable language). NetworkFunction, OfficeFunction as well as GameFunction are examples of specialization (of ResourceFunction). *Source: TR255 {1}* *TMF071 {6}* • Examples of RFs from ETSI NFV are VNF (an atomic RF) and Network Service (a composite RF). • Examples of RFs from IETF / SDN are Service Function and Service Function Chain	**Resource Function** {2} ***Defn***: A ResourceFunction is described by a ResourceFunctionSpec specifying the possible configurations. A ResourceFunction is a function to transform inputs of any nature into outputs of any nature independently from the way it is provided. Each time a ResourceFunctionSpecification is provided whatever the way it is done (SoftwareSpecification, SoftBlackBoxSpec or PhysicalBlackBox), a ResourceFunction is created with its ConnectionPoints. The ResourceFunction is provided either by an InstalledSoftware, a SoftBlackBox, or a PhysicalBlackBox. These relationships are exclusive.	**Resource Function** {3} TMF664 Resource model. ***Defn:*** A ResourceFunction is a behavior to transform inputs of any nature into outputs of any nature independently from the way it is provided
Feature/ Function {1} ***Defn***: A Feature is how an Resource Function (RF) exposes its capabilities to	**ConfigurationFeature** {5} ***Defn:*** A ConfigurationFeature is a type of Entity described by a ConfigurationFeatureSpec.	**Feature** {3} In TMF 664 Feature Sub-resource modelled

ODA/ Frameworx/ TR255	SID GB922	TMF 664 Resource Function Activation and Configuration API
consumers. Source: TR 255B {1} and GB922 Resource {2}	It is used to expose one or several ResourceFunctions to a consumer in such a way that it is easier to configure these RFunctions from a consumer view point. A ConfigurationFeature may have characteristics configured according to the possible configurations specified by the ConfigurationFeatureSpec. Unlike a Resource Function (RF), a feature cannot be independently deployed. {5}	as Configuration Feature
Feature Group{1} **Defn:** Feature Group is an aggregation of Features. A feature group may have characteristics in addition to the characteristics of the constituent features. pg 35. Source TR 255B	**ConfigurationFeature {5)** **Defn:** as above Note there are some gaps being addressed in the SID as ConfigurationFeature needs to be rooted to support the required composition required and the unconditional associate with Installed software needs to be modified to conditional	Uses flag "isBundle" to represent composition in instances. {3) Follows API defined patterns in TMF 630
Template /Profile Used informally but based on MTOSI Model in SID {4} pg 405	**Configuration {4}** ***Defn***: A Configuration (also referred to as a Profile) defines how a Resource, Service, or Product operates or functions in terms of Characteristic Specifications and related Product/Service/Resource Specifications as well as Characteristics and related Product/Service/Resource entities. A Configuration may contain one or more parts (which is realized by using the Atomic/Composite pattern, but it is represented as a single entity	Not used at present. In TMF664. IG1194 Focus on Services not Slices {7} shows an example of how templates / Profile can be added.

ODA/ Frameworx/ TR255	SID GB922	TMF 664 Resource Function Activation and Configuration API
	- ConfigurationRelationship), and each part may contain zero or more fields.	
Specification Concepts		
ResourceFunctionSpec {1} ***Defn***: A **Resource Function Spec** is a specification for a type of Resource Function.	**Resource FunctionSpec {8} {9}** ***Defn***: A ResourceFunctionSpec specifies a function as a behaviour to transform inputs of any nature into outputs of any nature independently from the way it is provided. It is typically created by a function designer who may not have specific knowledge on realisation architecture (for example using English text explanations with diagrams, like RFC standards, or preferably a machine interpretable language). NetworkFunctionSpec, OfficeFunctionSpec as well as GameFunctionSpec are examples of specialisation. The realization/deployment of a ResourceFunctionSpec may be achieved by different ways; either SoftwareSpecification, PhysicalBlackBoxSpec or SoftBlackBoxSpecification. Examples: manually or automated by software or hardware implementing algorithms etc… without any impact on the definition of the functions itself. A ConnectionPointSpec is used to describe the information/data input to a ResourceFunctionSpec or output from a ResourceFunctionSpec. This	Resource FunctionSpec not used as APIs are instance based on entities and data models

ODA/ Frameworx/ TR255	SID GB922	TMF 664 Resource Function Activation and Configuration API
	description is captured independently from the way it is realised.	
Feature/ Function Spec {1} ***Used informally***	**ConfigurationFeatureSpec {8} {9}** ***Defn:*** A ConfigurationFeatureSpec is a type of EntitySpecification. It is used to expose one or several ResourceFunctionSpecs to a consumer in such a way that it is easier to configure these RFSpecs from a consumer view point, at the deployment stage (intend based deployment as opposed to detailed based deployment). •Some ConfigurationFeatureSpec may be mandatory while others can be selected or not by the consumer (for a given instantiation). • ConfigurationFeatureSpec may have characteristics which are settable by the consumer (at instantiation time and/or during the lifetime of the ResourceFunction).	FeatureSpec is used in TMF 634 Resource Catalog but TMF664 uses Feature as its focus is on instances in the Data Model.
Template /Profile Spec Used informally	**ConfigurationSpec (8}** **Defn:** The definition of how a Resource, Service, or Product operates or functions in terms of CharacteristicSpecification(s) and related ResourceSpec(s), ProductSpec(s), ServiceSpec(s).	Not used at present

Notes for table:

{1} **TR255** Resource Function Activation and Configuration Suite R17.5.0 [40]

{2} **GB922 Resource** v19.5.1 Section 3.2.11 Resource Function: Support for Virtualization and Resource Abstraction pages 38ff [41]

{3} TMF664 Resource Function Activation and Configuration API REST Specification R17.0.1 [45] Needs updating

{4} **GB922 Common** v19.5.1 Section 2.16 Configuration and Profiling ABE Pages 394ff [44]

{5} **GB922 Resource** v19.5.1 Section 3.2.15 Use of ConfigurationFeature to support ResourceFunction pg 48 [42]

{6} TMF071 **TMF071** ODA Terminology R19.0.1 [46]

{7} **IG1194** Focus on Services not Slices v1.0.1 [35]

{8} GB922 Models GB922 Information Framework Models Suite R19.0 [??]

{9} **GB922 Resource** v19.5.1 3.2.16. ResourceFunctionSpec and related entities 50ff [49]

15. Administrative Appendix

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

15.1. Document History

15.1.1. Version History

Version Number	Date Modified	Modified by:	Description of changes
0.1.0	05-Sep-2018	Yuval Stein	Creation of Draft V01: Shared with NTT, Steve, Dave
1.0.0	23-Dec-2018	Alan Pope	Final edits for initial release
1.0.1	31-May-2019	Dave Milham	Partial updates excluding chapter for R19
2.0.0	19-Jun-2019	Alan Pope	Final edits for R19.0 release
2.0.1	02-Oct-2019	Adrienne Walcott	Updated to reflect TM Forum Approved Status
2.0.2	21-Nov-2019	Dave Milham	Restructure to meet NTT proposal in ODA-310 Updates include snapshot UML Model based on SID ODA discussions in Appendix C
3.0.0	02-Dec-2019	Alan Pope	Final edits for v3.0 release
4.0.0	03-Apr-2020	Dave Milham	Editorial updates to address technical deficit decribed in ODA-406 Common Resource concepts Mapping table across TR255/GB999/GB922/TMF664. Also updates to address restructuring changes ito GB922 Guides.
4.0.1	25-May-2020	Adrienne Walcott	Updated to reflect TM Forum Approved Status

15.1.2. Release History

Production Release	Date Modified	Modified by:	Description of changes
18.5.0	23-Dec-2018	Alan Pope	Initial Release
19.0.0	19-Jun-2019	Alan Pope	Updated for Release 19.0

Production Release	Date Modified	Modified by:	Description of changes
19.0.1	02-Oct-2019	Adrienne Walcott	Updated to reflect TM Forum Approved Status
v3.0.0	02-Dec-2019	Alan Pope	Final edits for v3.0 release
Pre-production	03-Apr-2020	Alan Pope	Final edits for v4.0 release
Production	25-May-2020	Adrienne Walcott	Updated to reflect TM Forum Approved Status

15.2. Acknowledgements

This document was prepared by the members of the TM Forum ODA project:

- Yuval Stein, TEOCO Ltd
- Samita Chakrabarti, Verizon
- Johanne Mayer, Mayer Consulting
- Neetu Singhal, Nokia
- Massimo Banzi, Telecom Italia
- Shingo Horiuchi, NTT
- Yuki Nakata, NTT
- Chao Wu NTT
- James O'Sullivan, Huawei
- Kevin McDonnell, Huawei
- Vance Shipley, Sigscale
- Dave Milham, TM Forum
- Dirk Rejahl, TM Forum
- Alan Pope, TM Forum