TELECOM INFRA PROJECT

Open Core Network Project Group

Orchestration

Technical Requirements v1.0

Authors:

- **Selcuk Duman**
  - Vodafone – Enterprise Architect – Cloud Automation
- **Riccardo Gasparetto Stori**
  - Vodafone – Cloud Automation Architect
- **Emmanuel Sarris**
  - Vodafone – Orchestration Architect
- **Borislav Glozman**
  - Amdocs – Software Architect

Editors:

- **Chris Morton**
  - Isn't That Write – Principal

Change Tracking

Date	Revision	Author(s)	Comment
	V0.1	Vodafone	Draft version
	V0.2	Vodafone	Software Architecture added
	V0.3	Vodafone	Use cases categorised
	V0.4	Vodafone	North/Southbound APIs updated
	V0.5	Vodafone/Amdocs	Updated with comments from Amdocs
	V0.6	Amdocs	Added Requirements section
	V0.7	Vodafone/Amdocs	Moved SOL005 to NBI
	V1.0	TIP/Isn't That Write	Cleaned up formatting. Proofreading and line editing. Minor editorial revising. Checked conformance with (US) Plain Writing Act of 2010. Graphics manipulation and captioning.

C2 General
Copyright © 2020 Telecom Infra Project, Inc.

Table of Contents

C2 General

Copyright © 2020 Telecom Infra Project, Inc.

1. Introduction

The dynamic nature of the next generation core network services benefits from an orchestration framework that enables automated management and operations of the underlying services. The orchestration platform enables deployment, management, and assurance of core network services, and provides overall capacity management.

Core networks support widely deployed services, such as mobile broadband, and specialized instances to support specific use cases (e.g., dedicated enterprise networks, consumer IoT).

This document describes the technical requirements of an orchestration framework to manage the services and resources related to TIP open core network (OCN) components and provides use case examples of the expected functionality.

1.1. Project Goal

The goal of this project is to develop an orchestration framework integrated with the OCN and able to manage the lifecycle of its components and the complete system—including microservices-based network functions, core applications, and connectivity. The framework is planned to permit operator input and provide a closed loop logic, managing fault and performance measurements by receiving analytics information from the open core components and an external service.

1.2. Document Scope

This document underlines the expectations from the OCN functions to enable their management via standard-based protocols and descriptors.

Its purpose is to describe the platform architecture of the orchestration system to manage OCN components, including the underlying infrastructure and services.

1.3. Document Structure

This document is structured as follows:

- Chapter 1: Introduction
- Chapter 2: Use Cases
- Chapter 3: Requirements
- Chapter 4: Platform Architecture & Specifications
- Chapter 5: Glossary

C2 General
Copyright © 2020 Telecom Infra Project, Inc.

2. Use Cases

2.1. NSSMF

5G core activation

- Configure network functions (NFs) to adapt to resource management actions such as migration of an NF between sites
- Dynamically configure instantiated NFs in edge sites
- Support zero touch provisioning (ZTP) use cases for CNF/PNF deployments in central and edge sites

5G core network slicing – capabilities to support eMBB, uRLLC, mIOT, V2X use cases

- CNF provisioning/scaling/termination
- CNF Day 0/1/2 configuration

2.2. NFVO

5G core NF instantiation/activation (shared and dedicated NFs) in the same site.

- CNF provisioning (via VIM/K8S APIs)
- CNF Day 0 configuration (via cloud-init/ConfigMaps)
- Data center network configuration (via a SDN controller or a configuration manager)

5G core instantiation/activation on multiple sites.

- Provision and configure 5G core NFs in multiple sites based on network service capacity and resiliency requirements.
- Dynamically migrate NFs from one site to another based on NS templates.

5G core network function instantiation/activation – edge sites

- Dynamically provision, configure, and scale NFs in edge sites
- Support collocated NFs and other software applications (consumer and enterprise)

5G core NF modification in central and edge sites

- Dynamically scale in/out 5G core NFs based on user input or feedback from the assurance, FM/PM systems

Software delivery & upgrade

- Support continuous delivery and testing of CNF software updates
- Hitless upgrade of NFs

C2 General
Copyright © 2020 Telecom Infra Project, Inc.

2.3. Assurance & Analytics Platform

- Collect KPIs from OCN functions
- FM/PM data from the NF level can be used as an input for the KPI data (NSSI level)
- SLA fulfilment monitoring

3. Requirements

Requirement	Description	Comment
REQ-ORC-010	Open core orchestrator shall support design and modeling of the open core in TOSCA	TBD: version of OASIS TOSCA and reference
REQ-ORC-015	Open core orchestrator shall support design and modeling of the open core in YANG	YANG models for slice/subslice TBD
REQ-ORC-020	Open core orchestrator shall support runtime implementation of open core TOSCA models	Runtime catalog service specification (see point above)
REQ-ORC-025	Open core orchestrator shall support GSMA Generic Network Slice Templates v2 for modeling design and management of network slice	Design time
REQ-ORC-030	Open core orchestrator shall provide inventory of OCN services (NSs) and NFs, including their runtime information	Resource inventory (TMF639)
REQ-ORC-040	Open core orchestrator shall store the runtime configuration of all NFs	X versions of configuration shall be stored
REQ-ORC-045	Open core orchestrator shall support open core provisioning feasibility	3GPP TS 28.530/531
REQ-ORC-050	Open core orchestrator shall support open core creation	
REQ-ORC-045	Open core orchestrator shall support open core allocation/deallocation of NSSI	
REQ-ORC-060	Open core orchestrator shall support open core activation	
REQ-ORC-065	Open core orchestrator shall support open core modification	
REQ-ORC-070	Open core orchestrator shall support open core deactivation	
REQ-ORC-080	Open core orchestrator shall support open core termination	

C2 General
Copyright © 2020 Telecom Infra Project, Inc.

REQ-ORC-090	Open core orchestrator shall have NFVO functionality required to perform life cycle management (LCM) operations on network services that comprise the open core	
REQ-ORC-110	Open core orchestrator shall have NFMF functionality required to manage NFs that comprise the open core	
REQ-ORC-120	Open core orchestrator shall support orchestration of composite and nested NSs that model open core	
REQ-ORC-130	Open core orchestrator shall support orchestration of shared and dedicated nested NSs that model open core	
REQ-ORC-140	Open core orchestrator shall support orchestration of cloud native network functions (CNFs) on k8s	
REQ-ORC-150	Open core orchestrator shall support configuration of CNFs on k8s	e.g., Day 0/1 configuration using k8s configMaps, CRDs, cloud-init Check if CaaS is present
REQ-ORC-160	Open core orchestrator shall support network configuration	via a datacenter SDN controller or configuration manager
REQ-ORC-170	Open core orchestrator shall support Open Core components deployment on multiple sites (edges)	Edges can be in one or multiple k8s cluster(s)
REQ-ORC-173	Open core orchestrator shall provide highly efficient monitoring capabilities of the servers, pods, and applications of the open core	
REQ-ORC-175	Open core orchestrator shall collect, aggregate, store, and present PM/FM data of open core components	

C2 General

Copyright © 2020 Telecom Infra Project, Inc.

REQ-ORC-180	Open core orchestrator shall support selection of site, cluster, node to deploy the NSs and NFs that open core consists of according to capacity, resiliency, cost, anti/affinity and other requirements	e.g., policies, AI/ML, Data locality for low latency app Data + service anti/affinity policies per pod Data + service anti/affinity policies across pods of different services Placement restrictions based on labels Placement restrictions based on NUMA constraints Placement restrictions based on multi-tenancy isolation policies Placement restrictions based on IO profile types
REQ-ORC-190	Open core orchestrator shall support dynamic migration of NFs from one site to another	
REQ-ORC-200	Open core orchestrator shall support scaling of NFs	K8s features may be used
REQ-ORC-210	Open core orchestrator shall support healing of NFs	K8s features may be used
REQ-ORC-220	Open core orchestrator shall support modification of NSs and NFs based on user request or as a result of closed loop operations based on FM/PM collection	
REQ-ORC-230	Open core orchestrator shall support dynamic configuration of NFs based on user request or as a result of closed loop operations based on FM/PM collection	e.g., Day 2 configuration
REQ-ORC-240	Open core orchestrator shall support software upgrade of NFs	
REQ-ORC-250	Open core orchestrator shall support network slicing of open core, including slice design, modeling, configuration, provisioning and storing in inventory	Different types of slices shall be supported: eMBB, uRLLC, mIOT, V2X, and future defined slices (TS 23.501)
REQ-ORC-260	Open core orchestrator shall support standard APIs for northbound communication (CSMF/NSMF/NSSMF/NS)	TMF-641, TMF-633, TMF-638, TMF639, TMF642, SOL005, 3GPP APIs (28.533, 28.531, 28.541, 28.545) RAN and transport are out of scope

C2 General

Copyright © 2020 Telecom Infra Project, Inc.

REQ-ORC-270	Open core orchestrator shall support southbound APIs towards all VIM types that manage the NFVIs that open core is deployed on	e.g., Helm charts, K8s, APIs, CRDs, operators
REQ-ORC-280	Open core orchestrator shall support southbound APIs toward open core NFs	e.g., NETCONF for configuration
REQ-ORC-290	Open core orchestrator shall support standard southbound APIs toward SDN controllers for transport establishment between open core NFs	e.g., NETCONF/RESTCONF
REQ-ORC-300	Open core orchestrator shall be deployed as a cloud-native system	
REQ-ORC-310	Open core orchestrator shall be designed and implemented according to microservices paradigm	
REQ-ORC-320	Open core orchestrator shall expose and consume secure APIs	e.g., TLS 1.3
REQ-ORC-330	Open core orchestrator shall store all open core related information in a secure manner	e.g., encrypted at rest
REQ-ORC-340	Open core orchestrator shall not expose any sensitive information	e.g., passwords
REQ-ORC-345	Open core orchestrator shall operate with minimum privileges	non-root user
REQ-ORC-346	Open core orchestrator shall support role-based access control (RBAC)	
REQ-ORC-347	Open core orchestrator shall support integration with standard user management platforms	e.g., LDAP and AD
REQ-ORC-350	Open core orchestrator shall provide the ability to be deployed in a highly available setup for control and data planes	e.g., Active/Standby or Active/Active
REQ-ORC-360	Open core orchestrator shall be able to be deployed in a georedundant setup	
REQ-ORC-370	Open core orchestrator shall support continuous integration/continuous delivery (CI/CD) of OCN services and NSs	K8s features may be used
REQ-ORC-380	Open core orchestrator shall be automatically installed with full functionality	
REQ-ORC-390	Open core orchestrator shall allow scaling	
REQ-ORC-400	Open core orchestrator shall provide a precheck script for validation of its full operability	
REQ-ORC-410	Open core orchestrator shall be able to run on any environment	e.g., bare metal, VM, Web-scale

C2 General

Copyright © 2020 Telecom Infra Project, Inc.

REQ-ORC-420	Open core orchestrator deployment shall support availability zones and multi-AZ configuration	
REQ-ORC-430	Open core orchestrator shall support automatic upgrade (for example, rolling upgrades, CNI plugins addition/ modification, blue-green/canary releases) of its components	
REQ-ORC-440	Open core orchestrator shall support data management, including auto discovery, snapshot/rollback, clone, backup/restore, migrate	
REQ-ORC-450	Open core orchestrator shall provide highly efficient monitoring capabilities of its servers, pods, microservices	
REQ-ORC-460	Open core orchestrator shall collect and store the CPU, memory, storage, and network metrics of its microservices with configurable retention	
REQ-ORC-470	Open core orchestrator shall emit or stream faults, alerts, and events	
REQ-ORC-480	Open core orchestrator shall write, collect, and store logs of its microservices, permitting aggregation, access, export, streaming, and analysis	
REQ-ORC-490	Open core orchestrator shall perform rapid failover on server fault while it honors service/app level anti/affinity policies	
REQ-ORC-500	Open core orchestrator shall support multi-tenancy	
REQ-ORC-510	Open core orchestrator shall be developed in an extendable manner so it can be used for orchestration of other domains	

C2 General

Copyright © 2020 Telecom Infra Project, Inc.

4. Platform Architecture & Specifications

4.1. Assumptions

Open core software components to be orchestrated by the orchestrator system shall support the APIs, communication, and packaging standards as described in section 4.3.

Underlying infrastructure components are able to communicate with the orchestrator as detailed in section 4.3.5 *Southbound APIs*.

Open core software is developed as a cloud-native system. All components are designed as microservices able to run independently and (where possible) in a stateless manner, so that each function can be initiated, scaled, healed, and terminated with minimal or no impact on the rest of the components.

The functions within the OCN framework are represented with declarative templates (i.e., TOSCA), and leverage the cloud platform capabilities to automate provisioning (e.g., cloud-init, K8S config-maps, CRDs, and operators).

The services provide capability to the orchestrator to modify service configuration through standard APIs.

This document will be updated with additional requirements such as security, geo-redundancy and high availability that will be addressed in the later phases.

4.2. Orchestration Definition

NFs and management systems are built compliant to standards developed by institutions (including ETSI and 3GPP). These functions are usually pre-packaged and include VNF descriptors (e.g., TOSCA), defining their topology, automatic life cycle management workflows, policies, and KPIs. To avoid silos and run these functions within the same infrastructure, the need to coordinate/chain multiple NFs and manage capacity along with automatic life cycle operations is evident.

Orchestrators are developed to fulfil such requirements of coordination, infrastructure reuse, and service consistency. At the same time, they permit easier introduction of future services such as distributed core functions (MEC) and 5G slices. These will require a large amount of highly dynamic resources to be deployed in many locations and running simultaneously with other software applications. Together with its API layer, an orchestration framework will also enable customers to request services online and see them automatically fulfilled.

C2 General
Copyright © 2020 Telecom Infra Project, Inc.

4.3. Software Architecture

4.3.1. Cloud-Native

The orchestration framework components shall be built as a cloud-native system, enabling each component to run as an microservice thar can be instantiated, scaled, healed, and terminated without affecting the rest of the system.

The components of the system should be able to run on the same type of platform as the remainder of the OCN functions to provide unity and scalability.

Resource management and automation layer services shall interact with respective infrastructure functions through virtualization and SDN controllers.

4.3.2. Standards Support

The orchestration system shall support relevant standards from 3GPP, ETSI, IETF—including data models and APIs required for managing cloud-native systems. The standards version the orchestration system is compliant with shall be specified for each release. These standards will be categorized according to characteristics and suitability of use to develop VNF/CNF and NS descriptors, northbound APIs, southbound APIs, and cloud-native APIs.

4.3.3. VNF/CNF and NS descriptors

The descriptors for open core NFs are expected to be deployed in a topology and orchestration specification for cloud applications (TOSCA)-compliant package. TOSCA specifications can be used to describe both component services and end-to-end (E2E) services, including their dependencies, requirements, capabilities, and the relationship between them.

The ETSI SOL001 standard describes specifications for using TOSCA-based service templates for the virtualized network function descriptor (VNFD), network service descriptor (NSD), and physical network function descriptor (PNFD).

These service descriptors are packaged together in a TOSCA YAML cloud service archive (CSAR) as specified in the ETSI SOL004 standard. This will provide the capability to manage all the descriptors along with the software images and other artifacts (e.g., custom workflows) required to deploy a network function in a single file.

4.3.4. Northbound APIs

The orchestrator framework shall be able to interact with other northbound systems such as an E2E service orchestrator or an assurance system via industry standard protocols. The protocols that will be used to support the network service APIs are listed next:

C2 General
Copyright © 2020 Telecom Infra Project, Inc.

TM Forum 633 Service Catalogue Management API REST Specification will offer the orchestration platform a standard set of capabilities for querying external service catalogues to determine the available and relevant service types for the domain.

TM Forum 638 Service Inventory Management API REST Specification will offer the orchestration platform a standard means of querying and updating the service inventory that resides in an external system and updates it for changes in the service state.

TM Forum 639 Resource Inventory Management API REST Specification will offer the orchestration platform and external systems a standard means of querying and updating the orchestrator's resource inventory and update external systems regarding changes to the resource based on properties defined in the service catalogue.

TM Forum 641 Service Ordering Management API REST Specification will offer the orchestration platform and external systems a standard means of ordering services from the orchestrator, thus enabling the NaaS concept. Notifications of changes to the order item(s) state and service instance will also be offered. Error handling notifications will also be exposed northbound to the external systems such as an E2E service orchestrator.

TM Forum 645 Service Qualification API REST Specification will offer the orchestration platform and external systems a standard means of validating whether specific service characteristics can be offered for a given infrastructure and/or at a given geographic location.

ETSI SOL005: ETSI SOL005 is used for communications between an NFVO and external systems such as assurance systems or northbound applications such as a network service orchestrator or a network slice subnet management function (NSSMF).

4.3.5. Southbound APIs

The orchestration framework for the OCN platform shall be able to support and orchestrate the infrastructure providers for the microservices framework.

In the proposed architecture—with the assumption of NSs being developed as cloud-native functions using Kubernetes as the infrastructure provider—the orchestration framework should be able to support Kubernetes APIs in addition to specific CaaS APIs (if a CaaS system is provided to support multiple Kubernetes clusters).

Resource management options for creation of virtualized resources (e.g., PODs, virtual networks):

- **Kubernetes API** – External Kubernetes resource manager (in the event Kubernetes is used as the infrastructure provider)
- **No API** – Internal capabilities for resource management (in the event capability is integrated as part of the orchestrator)

C2 General
Copyright © 2020 Telecom Infra Project, Inc.

During the LCM operations of a microservice within the OCN services, the horizontal configuration of the related VNFs and PNFs—including the underlying virtual and physical network equipment (e.g., routers and switches)—should be managed by the orchestrator through a configuration management capability using standards-based protocols such as 1) NETCONF/YANG, 2) an SDN controller API, and 3) open mechanisms provided by the infrastructure such as ConfigMaps, CRDs, operators, and cloud-init.

5G Core Service Orchestration options for 5G core service activation:

- **Netconf**/YANG or RESTconf/YANG
- **Kubernetes** with ConfigMaps (for stateless network functions)
- **Ansible** or similar configuration management tool

4.3.6. 5G Readiness

Traditionally, service providers use a siloed set of management tools specific to systems to be managed. Such systems are called element management systems (EMS) and are designed to manage vendor products and systems using standard and proprietary protocols.

With the advent of 5G, 3GPP created some standards to enable standards-based management of the 5G core network functions. These NFs and associated standards based on a service-based architecture are detailed in TS 23.501.

Another concept defined by 3GPP that uses service-based architecture is the management and orchestration of network slices as defined in TR 28.801. The following are among the related management functionality defined in TS 28.533.

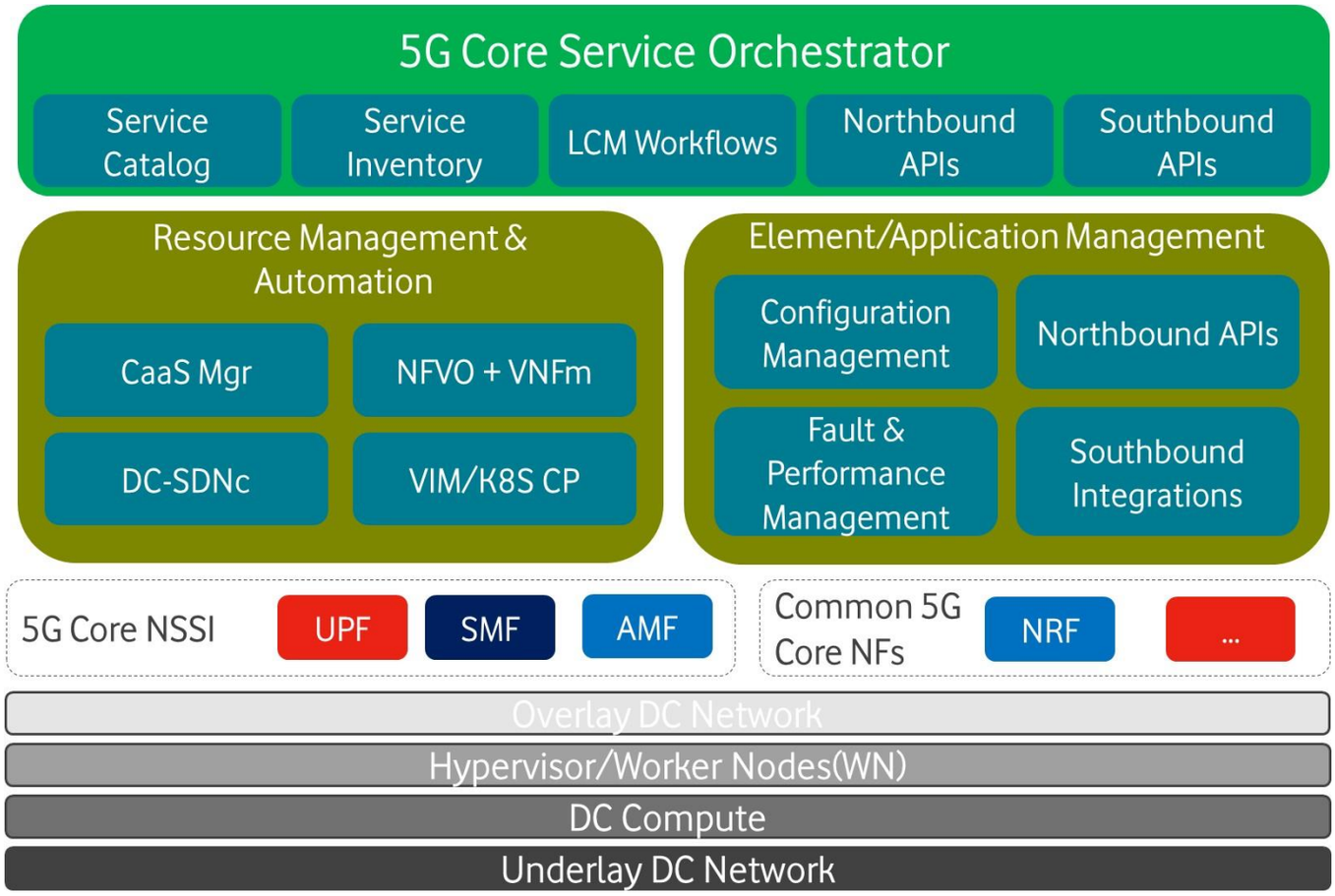
Communication Service Management Function (CSMF) – Responsible for translating the communication service-related requirement to network slice related requirements.

Network Slice Management Function (NSMF) – Responsible for management and orchestration of NSI. NSMF derives network slice subnet-related requirements from network slice-related requirements.

Network Slice Subnet Management Function (NSSMF) – Responsible for NSSI management and orchestration.

- OCN orchestrator should be able to manage slicing requirements of a service as communicated through NSSMF or—in the case that the orchestrator has NSSMF functionality—through NSMF.
- The orchestrator should manage the lifecycle of the 5G core network services to be provisioned.
- NFs to be managed should be stateless. Service characterization and state information will be stored within the application management components.

C2 General
Copyright © 2020 Telecom Infra Project, Inc.



5. References

References to the standards in the document will be listed here.

6. Glossary

C2 General

Copyright © 2020 Telecom Infra Project, Inc.