

Vision and Evolution of Networks through Ubiquitous Softwarisation White Paper

Version 1.0

22 Sept 2026

Security Classification: Non-Confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Copyright © 2026 GSM Association

Disclaimer

The GSMA makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Compliance Notice

The information contain herein is in full compliance with the GSMA Antitrust Compliance Policy.

This Permanent Reference Document is classified by GSMA as an Industry Specification, as such it has been developed and is maintained by GSMA in accordance with the provisions set out GSMA AA.35 - Procedures for Industry Specifications.

Table of Contents

1 Executive summary 5
2 Introduction 6
2.1 Overview 6
2.2 Scope 7
2.3 Definitions 8
2.4 Abbreviations 9
2.5 References 11
3 Conventions 16
4 Standardisation and open source in network softwarisation 16
4.1 Introduction 16
4.2 Standardisation activities 16
4.2.1 ETSI ISG NFV 16
4.2.2 3GPP 17
4.2.3 ITU-T 18
4.2.4 O-RAN ALLIANCE 18
4.2.5 GSMA 19
4.3 Open-source activities 20
4.3.1 Cloud Native Computing Foundation 20
4.3.2 ETSI SDG OSM 22
4.3.3 Anuket 22
4.3.4 Sylva 23
4.3.5 CNTi test suite 23
4.3.6 Nephio 24
4.3.7 OpenStack and Tacker 25
4.4 Analysis of standardisation and open-source activities 25
4.4.1 Standards and open source overview 25
4.4.2 Interplay between standards and open-source activities 28
4.4.3 Potential way forward – harmonization between standards and opensource 29
5 Telecom industry softwarisation trends and maturity 30
5.1 Introduction 30
5.2 Key trends on softwarisation of telecom networks 31
5.2.1 Introduction 31
5.2.2 Containerization 31
5.2.3 Microservices 32
5.2.4 Declarative and intent-based management 32
5.2.5 Infrastructure as Code 33
5.2.6 GitOps 33
5.2.7 CRDs, custom operators and controller-based orchestration 33
5.2.8 Serverless and FaaS 34
5.2.9 AI native 34
5.2.10 eBPF 35

5.2.11 Observability 36
5.3 New prospects, business roles and deployment scenarios of telecom networks 37
5.3.1 Deployment maturity levels in softwarised telecom networks 37
5.3.2 Business roles and stakeholders in softwarised telecom networks 39
5.3.3 Deployment scenarios: on-premises, public Clouds and hybrid Cloud 41
6 MNO's pain-points, requirements and gap analysis 43
6.1 Overview of MNO challenges 43
6.2 Technical and operational pain points 43
6.2.1 Hardware dependency in data plane processing 43
6.2.2 Consistency issues between NF-to-NF communication and Kubernetes networking models 44
6.2.3 Constraints on service mesh and automation frameworks 44
6.2.4 Inconsistent virtualisation models across NFs 45
6.2.5 High-coupling design of NFs' architecture 45
6.2.6 Product lifecycle and support 45
6.2.7 Limited NF's design visibility and integration with management stacks 45
6.2.8 Upgrade procedures 46
6.2.9 Vulnerability and transparency in open-source supply chains 46
6.2.10 Operational complexity of fragmented Autonomous Network (AN) tools 47
6.2.11 Siloed domain-specific management and orchestration 47
6.3 Requirements for softwarised networks 47
6.3.1 Overview 47
6.3.2 Standardised NF-to-NF communication model 47
6.3.3 Clear responsibility boundaries between NFs and the platform 47
6.3.4 Common framework for CNF lifecycle operations 48
6.3.5 Guidelines for performance assurance 48
6.3.6 Proper stateless design to enable better agility of NFs 48
6.3.7 Stable and long-term support models and interfaces 49
6.3.8 Simplified exposure towards MNO's other OSS systems 49
6.3.9 Capabilities for energy efficiency and sustainable operations 49
6.3.10 Open-source based supply chain transparency and security assurance via SBOM 49
6.3.11 In-service upgrade support 49
6.3.12 GitOps and declarative management 49
6.4 Gap analysis vs. existing standardisation and open-source initiatives 50
6.4.1 Overview 50
6.4.2 Alignment among standards 50
6.4.3 Limitations of the open-source based solution ecosystem 51
6.4.4 Lack of an integrated framework 51
7 Vision and recommendations for future telecom network softwarisation standardization and open source 51
7.1 Vision for future telecom network softwarisation 51
7.2 Recommendations 53

7.2.1 Introduction 53
7.2.2 Recommendations related to standardisation and open-source activities 54
7.2.3 Recommendations related to key trends in network softwarisation 55
7.2.4 Recommendations related to mobile network deployment and business roles 55
7.2.5 Recommendations related to technical and operational pain-points, and gaps regarding development of softwarised mobile networks 56
8 Conclusion 57
Annex A Document Management 59
A.1 Document History 59
A.2 Other Information 59

1 Executive summary

Network softwarisation has become an intrinsic feature in how mobile networks are designed, deployed and operated. Supported by paradigms and underpinning technologies from Network Functions Virtualisation (NFV), Software-Defined Networking (SDN), and Cloud, mobile network operators (MNO) have pursued towards further network disaggregation and openness. The openness is clearly reflected with the diverse options that MNOs can nowadays consider regarding their networks' deployment, both from an actual deployment and implementation perspectives. On the one hand, on-prem and Public Cloud deployment options are being explored, not only from a proof-of-concept (PoC) point of view, but also into actual commercialization. MNOs have more choices regarding the kind of resources to leverage and tailor to whichever telecom network workloads are the target for being softwarised. On the other hand, newer and multiple actors and roles become part of the ecosystem, expanding beyond MNO and network equipment providers; Cloud providers, IT and software suppliers, likewise system integrators play also now an important role in the overall network design, deployment and operation.

MNO's use cases are expanding from the initial 4G/EPC basic NFV-based VNF deployments, prominently based on virtual machine (VM) technologies, towards more sophisticated network-service level orchestration with fully enriched cloud-native solutions, either NFV or pure open-source based. In fact, some 4G and the large majority of 5G network deployments are backed up by solutions based on standards or open-source de-facto solutions related to network softwarisation. Standards, such as ETSI NFV, ITU-T, 3GPP and O-RAN ALLIANCE, and open-source activities, like CNCF and LFN projects and OpenStack, are numerous and diverse, spanning solutions across layers (infrastructure, platform and services) and domains (from RAN, through edge, to core network). Such rich ecosystem is a reality of the interest of MNOs and other industry stakeholders, yet the potential for fragmentation and unclear end-to-end system integration is always at risk. A stronger collaboration and harmonization between standards and open-source have the potential to help MNOs efficiently rely on open-source for commercial deployment while providing necessary standard domain-specific abstractions to ensure long-term support.

From a technology and conceptual standpoint, NFV and cloud-native are regarded as the umbrella of a larger landscape of well-established current and devised new technology trends. Well-established technologies like containerization, micro-services, and approaches like declarative management, custom operators, and controller-based orchestration are being further complemented with technology trends like GitOps, serveless, Function as a Service (FaaS), extended Berkeley Packet Filter (eBPF), together with the omnipresence of AI-native features.

While the advances in network softwarisation and the use of related technologies are a reality, the "network softwarisation journey" is still ongoing. It is a fact that MNOs are still facing challenges in how the technologies and concepts are being applied. This uncovers multiple technical and operational pain points, like hardware dependency in data plane processing, inconsistent networking models between NF-to-NF communication and underlying Cloud platforms, inconsistent virtualisation models across NFs, product lifecycle and support, upgrades, etc. to name a few. Thus, the present white paper provides recommendations and guidelines to the industry, and standards and open-source

organizations to address the identified gaps, challenges and pain-points. Recommendations are grouped around four main aspects: standardisation and open-source activities, key technology and operational trends, mobile network deployment and business roles, and those addressing specific technical and operational pain-points. Furthermore, future activities are envisioned in a context of technologies rapidly evolving, including also AI, where network softwarisation is expected to enable new customer added-value services, make the softwarised networks more Cloud and AI-native, with expanded and true automation at all levels towards simplifying the deployment and operation of the networks.

With this white paper, the community is called for action to address the identified gaps and requirements with the purpose to promote that the industry, as a whole, can work towards uplifting the merits of network softwarisation and creating a stronger ecosystem around it.

2 Introduction

2.1 Overview

Since mid-2010's Network Functions Virtualisation (NFV) [1] and Software-Defined Networking (SDN) [2] initiatives have led a major digital transformation to telecommunication networks, and in particular to mobile networks. The transformation has been even more heightened with the adoption of cloudification technologies and cloud-native concepts coming from a larger pervasive ecosystem around Cloud computing. Mobile network operators (MNO) are fully embarked on the network softwarisation journey with plenty of mobile networks deployed worldwide on virtualised and cloudified infrastructure platforms.

In particular, 5G networks, and the 5G core more precisely, are regarded to be one of the most relevant examples of applying network softwarisation and cloud-native principles, not only on their deployment form, but also conceptually in their reference architecture, which have shown a trend of moving towards further network disaggregation. Network softwarisation is also expanding across multiple network domains, as depicted on figure 1. Common IT/Cloud infrastructure and virtualization and cloudification platforms are an enabler for deploying diverse kinds of network functions and other applications, all as software entities.

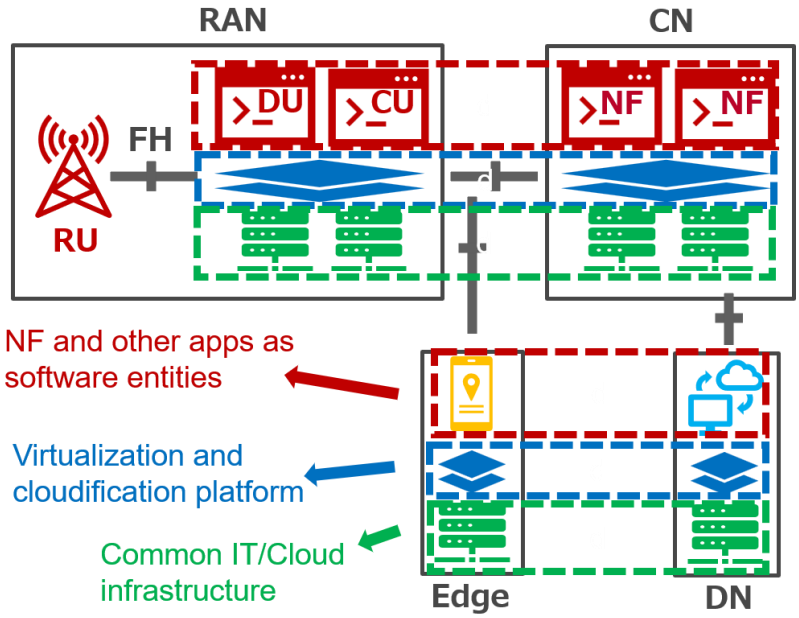

Figure 1: Softwarisation of telecom networks across network domains

As the telecom industry is moving forward in the adoption of network softwarisation and cloud-native concepts and technologies, reality has also shown that MNOs are still facing challenges and pain-points when it comes to developing and deploying their networks. A survey run within GSMA gauged MNOs on their current virtualised/cloudified network deployments and future prospects, confirming the gap between the forecasted and planned network deployments with respect to those actually fulfilled. The survey also provided insights on how far the MNOs have adopted NFV and other network softwarisation solutions, envisioning a clear trend toward more cloud-nativeness. But the survey analysis also indicated several issues that MNOs perceive, like standards and solutions fragmentation, lack of interoperability, higher complexity in network development and unfulfilled expectations on total cost of ownership (TCO) reduction.

As the results showed, telecom industry needs to progress further in a united way, promoting collaboration among all industry stakeholders and driving further interoperability regarding network softwarisation. In this context, the present white paper aims at identifying the trends and maturity of the telecom industry related to network softwarisation and analysing relevant standardisation and open source activities. As such learnings are clearly introduced and compared against MNO's pain-points and requirements, the white paper devises recommendations for future standardisation and open source work, and the telecom industry as a whole, to drive further interoperability of solutions and ease the path of MNOs in adopting at its fullness virtualisation and cloud-native technologies to softwarise mobile networks.

2.2 Scope

The present white paper analyses the standardization and open source activities related to network softwarisation. It further documents the maturity and deployment options of softwarised mobile networks, and the current and future key technology and conceptual trends driving the future of the network softwarisation. Finally, the present white paper

provides recommendations and guidelines for the industry, and in particular, to future standardisation and open-source related activities in the field of telecom network softwarisation. Such recommendations are based on the analysis of pain-points, challenges and high-level requirements devised by MNOs and other industry stakeholders compared to current standards and open source activities, and current and future trends, and network deployments maturity.

2.3 Definitions

Term	Description
Containerization	Process of packaging application code, including libraries and dependencies required to run the code, into a single lightweight executable, called container image, which is deployed and runs as operating system (OS) containers on the host computer by a container runtime. NOTE: For further information regarding "container image", refer to [10].
Container runtime	Software-based runtime which provides an abstraction of the underlying operating system and translates the commands of the program running inside the container into respective actions for the operating system. NOTE: For further information regarding "container runtime", refer to [10].
Declarative management	Management paradigm that enables consumers to declare the desired state of one or multiple managed objects without defining the actions to reach that desired state. NOTE: For further information regarding "declarative management", refer to [9].
eBPF	General-purpose engine (virtual machine) with origins in the Linux kernel that can run programs in a privileged context such as an operating system kernel.
GitOps	Set of practices for managing software applications and infrastructure by continuously evaluating and reconciling their desired states as defined in a version control system, such as Git, against their actual state. NOTE: Reconciling states includes changing the resources to match the defined state via deployment or updates [10].
Infrastructure as Code	Practice of storing the definition of infrastructure as one or more files. NOTE: For further information regarding "Infrastructure as Code", refer to [10].
Intent-based management	Management paradigm that enables consumers to declare expectations including requirements, goals and constraints given to a system, without specifying how to achieve them.
Kubernetes controller	Software program implementing control loop that watches the state of the Kubernetes cluster, then makes or requests changes where needed to move the current cluster state closer to the desired state. NOTE: For further information regarding "Kubernetes controller", refer to [11].
Kubernetes operator	Software extension to Kubernetes that makes use of custom resources to manage applications and their components running on a Kubernetes cluster.

Term	Description
	NOTE: These operators follow principles, notably the control loop-based reconciliation performed by Kubernetes controllers [11].
Microservices architecture	Architectural approach that breaks applications into individual independent (micro)services, with each service focused on a specific functionality. NOTE: For further information regarding "Microservices architecture", refer to [10].
Monolithic design	Design of applications that contain all the functionality in a single deployable program. NOTE: Once they grow in complexity, monolithic applications can become hard to maintain [10].
OS container	Lightweight virtualisation container utilizing a shared operating system kernel of its host. NOTE 1: The host providing the shared OS kernel can be a physical compute node or another virtualisation container [9]. NOTE 2: An OS container runs as a process with resource and capability constraints managed by the OS. NOTE 3: The files available to the container process are packaged as a container image [10].
Serverless	Cloud computing execution model where the cloud provider dynamically manages the allocation and provisioning of servers, and applications run in stateless compute containers (or other types of workloads) that are event-triggered, ephemeral, and fully managed by the cloud provider.

2.4 Abbreviations

Term	Description
AAL	Acceleration Abstraction Layer
AI	Artificial Intelligence
API	Application Programming Interface
ASD	Application Service Descriptor
ATCA	Advanced Telecommunications Computing Architecture
BSS	Business Support Systems
CaaS	Containers as a Service
CCM	CIS Cluster Management
CIS	Container Infrastructure Services
CISM	CIS Management
CNCF	Cloud Native Computing Foundation
CNF	Cloud-native Network Function
CNI	Container Network Interface
COTS	Commercial-off-the-shelf
CRD	Custom Resource Definition
CSAR	Cloud Service ARchive
CSP	Communications Service Provider

Term	Description
DMS	Deployment Management Services
DPDK	Data Plane Development Kit
DPU	Data Processing Unit
eBPF	extended Berkeley Packet Filter
FaaS	Function as a Service
FOCOM	Federated O-Cloud Orchestration and Management
IaC	Infrastructure as Code
IaaS	Infrastructure as a Service
IMS	Infrastructure Management Services
ISG	Industry Specification Group
IT	Information Technology
KRM	Kubernetes Resource Model
LCM	Lifecycle Management
LFE	Linux Foundation Europe
LFN	Linux Foundation Networking
LLM	Large Language Model
MANO	Management and Orchestration
MNO	Mobile Network Operator
Near-RT RIC	Near-Real-Time RAN Intelligent Controller
NF	Network Function
NFO	Network Functions Orchestration
NFV	Network Functions Virtualisation
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NS	Network Service
NUMA	Non-uniform memory access
OAM	Operations, Administration and Management
O-DU	O-RAN Distributed Unit
OS	Operating System
OSM	Open Source MANO
OSS	Operating Support Systems
PaaS	Platform as a Service
PRD	Permanent Reference Document
SDG	Software Development Group
SMO	Service Management and Orchestration
SR-IOV	Single root input/output virtualization

Term	Description
SRE	Site Reliability Engineering
TCO	Total cost of ownership
TOSCA	Topology and Orchestration Specification for Cloud Applications
VIM	Virtualised Infrastructure Manager
VNF	Virtualised Network Function
VNFM	VNF Manager

2.5 References

Ref	Doc Number	Title
[1]	N/A	M. Chiosi et al., *Network Functions Virtualisation – An Introduction, Benefits, Enablers, Challenges & Call for Action*, 2012, white paper. [Online]. Available at: https://docbox.etsi.org/ISG/NFV/Open/Publications_pdf/White%20Papers/NFV_White_Paper1_2012.pdf
[2]	N/A	Open Networking Foundation, *Software-Defined Networking: The New Norm for Networks*, ONF White Paper, April 2012. [Online]. Available at: https://opennetworking.org/wp-content/uploads/2011/09/wp-sdn-newnorm.pdf
[3]	ETSI GS NFV 002	ETSI GS NFV 002 V1.2.1 (2014-12): "Network Functions Virtualisation (NFV); Architectural Framework". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_nfv002v010201p.pdf
[4]	ETSI GS NFV-IFA 036	ETSI GS NFV-IFA 036 V5.3.1 (2025-08): "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements for service interfaces and object model for container cluster management and orchestration specification". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/036/05.03.01_60/gs_nfv-ifa036v050301p.pdf
[5]	ETSI GS NFV-IFA 040	ETSI GS NFV-IFA 040 V5.2.1 (2024-12): "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/040/05.02.01_60/gs_nfv-ifa040v050201p.pdf
[6]	ETSI GS NFV 006	ETSI GS NFV 006 V5.2.1 (2024-12): "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Architectural Framework Specification". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/006/05.02.01_60/gs_nfv006v050201p.pdf
[7]	ETSI GS NFV 008-1	ETSI GS NFV 008-1 V6.1.1 (2026-01): "Network Functions Virtualisation (NFV) Release 6; Architecture; Architectural Framework for Telco Cloud". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/00801/06.01.01_60/gs_nfv00801v060101p.pdf

Ref	Doc Number	Title
[8]	ETSI GS NFV 008-2	ETSI GS NFV 008-2 V6.1.1 (2026-04): "Network Functions Virtualisation (NFV) Release 6; Architecture; Part 2: Functional and Non-Functional Requirements for Telco Cloud". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/00802/06.01.01_60/gs_nfv00802v060101p.pdf
[9]	ETSI GR NFV 003	ETSI GR NFV 003 V1.9.1 (2024-12): "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV". [Online]. Available at: https://www.etsi.org/deliver/etsi_gr/NFV/001_099/003/01.09.01_60/gr_NFV003v010901p.pdf
[10]	N/A	Cloud Native Glossary. [Online]. Available at: https://glossary.cncf.io/
[11]	N/A	Kubernetes Documentation. [Online]. Available at: https://kubernetes.io/docs/home/
[12]	TM Forum IG1523A	IG1523A V1.1.0: "AN L4 Wireless Network Fault Management Solution Package"
[13]	TM Forum IG1523B	IG1523B V1.1.0: "AN L4 Core Network Fault Management Solution Package"
[14]	TMF921	TM Forum Specification TMF921 V5.0.0: "Intent Management".
[15]	N/A	Kubernetes Operator pattern. [Online]. Available at: https://kubernetes.io/docs/concepts/extend-kubernetes/operator/
[16]	N/A	Anuket, a Linux Foundation Project. [Online]. Available at: https://anuket.io/
[17]	GSMA - NG.126	Cloud Infrastructure Reference Model, Version 5.0, 10 June 2025. [Online]. Available at: https://www.gsma.com/newsroom/wp-content/uploads//NG.126-v5.0-3.pdf
[18]	GSMA - NG.133	Cloud Infrastructure Reference Architecture managed by OpenStack, Version 2.0, 09 November 2022. [Online]. Available at: https://www.gsma.com/newsroom/wp-content/uploads/NG.133-v2.0-11.pdf
[19]	GSMA - NG.139	Cloud Infrastructure Reference Architecture managed by Kubernetes, Version 2.0, 10 June 2025. [Online]. Available at: https://www.gsma.com/newsroom/wp-content/uploads//NG.139-v2.0.docx
[20]	N/A	Sylva Documentation. [Online]. Available at: https://sylva-projects.gitlab.io/docs/1.4/getting-started/solution-overview
[21]	N/A	Sylva GitLab repositories. [Online]. Available at: https://gitlab.com/sylva-projects/sylva
[22]	N/A	Cloud Native Telecom Initiative (CNTi). [Online]. Available at: https://lfnetworking.org/cloud-native-telecom-initiative/
[23]	N/A	Nephio documentation. [Online]. Available at: https://docs.nephio.org/docs/
[24]	N/A	Nephio R5 Release Notes. [Online]. Available at: https://docs.nephio.org/docs/release-notes/r5/
[25]	N/A	Nephio project GitHub. [Online]. Available at: https://github.com/nephio-project

Ref	Doc Number	Title
[26]	N/A	eBPF Documentary. [Online]. Available at: https://ebpf.io/
[27]	10.1109/ACCESS.2023.3281480	D. Soldani, et al., “eBPF: A New Approach to Cloud-Native Observability, Networking and Security for Current (5G) and Future Mobile Networks (6G and Beyond).” [Online]. Available at: https://ieeexplore.ieee.org/document/10138542
[28]	N/A	A. Green, “eBPF for the Infrastructure Platform – How Modern Applications Leverage Kernel-Level Programmability.” [Online]. Available at: https://www.linuxfoundation.org/hubfs/eBPF/The_State_of_eBPF25_111925.pdf
[29]	N/A	D. Brown, “Rakuten Mobile Adopts eBPF to Strengthen Anomaly Detection and Security in Cloud-Native Telecom Networks.” [Online]. Available at: https://ebpf.foundation/rakuten-mobile-adopts-ebpf-to-strengthen-anomaly-detection-and-security-in-cloud-native-telecom-networks/
[30]	N/A	O-RAN ALLIANCE, "About O-RAN ALLIANCE". [Online]. Available at: https://www.o-ran.org/about
[31]	O-RAN.WG1.TS.OAD	O-RAN.WG1.TS.OAD-R004-v15.00 (October 2025): "O-RAN Work Group 1 (Use Cases and Overall Architecture); O-RAN Architecture Description". [Online]. Available at: https://specifications.o-ran.org/download?id=932
[32]	O-RAN.WG6.TS.O2IMS-INTERFACE	O-RAN.WG6.TS.O2IMS-INTERFACE-R004-v10.00 (October 2025): "O-RAN Working Group 6; O2ims Interface Specification". [Online]. Available at: https://specifications.o-ran.org/download?id=966
[33]	O-RAN.WG6.O2DMS-INTERFACE-ETSI-NFV-PROFILE	O-RAN.WG6.O2DMS-INTERFACE-ETSI-NFV-PROFILE-R005-v10.00 (February 2026): "O-RAN Working Group 6; O2dms Interface Specification: Profile based on ETSI NFV Protocol and Data Models". [Online]. Available at: https://specifications.o-ran.org/download?id=1044
[34]	O-RAN.WG6.O2DMS-INTERFACE-K8S-PROFILE	O-RAN.WG6.O2DMS-INTERFACE-K8S-PROFILE-R004-v06.00 (June 2025): "O-RAN Working Group 6; O2dms Interface Specification: Kubernetes Native API Profile for Containerized NFs". [Online]. Available at: https://specifications.o-ran.org/download?id=894
[35]	O-RAN.WG6.AAL-GAnP	O-RAN.WG6.AAL-GAnP-R004-v13.00 (October 2025): "O-RAN Working Group 6 (Cloudification and Orchestration Workgroup); Acceleration Abstraction Layer General Aspects and Principles". [Online]. Available at: https://specifications.o-ran.org/download?id=969
[36]	N/A	OpenInfra Foundation, "OpenStack". [Online]. Available at: https://www.openstack.org/

Ref	Doc Number	Title
[37]	N/A	OpenInfra Foundation, "Tacker Documentation". [Online]. Available at: https://docs.openstack.org/tacker/latest/
[38]	N/A	OpenInfra Foundation, "Tacker Architecture". [Online]. Available at: https://docs.openstack.org/tacker/latest/user/architecture.html
[39]	ETSI GS NFV-SOL 001	ETSI GS NFV-SOL 001 V5.3.1 (2025-08): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; NFV descriptors based on TOSCA specification". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/001/05.03.01_60/gs_nfv-sol001v050301p.pdf
[40]	ETSI GS NFV-SOL 006	ETSI GS NFV-SOL 006 V4.3.1 (2022-09): "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; NFV descriptors based on YANG Specification". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/006/04.03.01_60/gs_nfv-sol006v040301p.pdf
[41]	ETSI GS NFV-SOL 002	ETSI GS NFV-SOL 002 V5.3.1 (2025-08): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/002/05.03.01_60/gs_nfv-sol002v050301p.pdf
[42]	ETSI GS NFV-SOL 003	ETSI GS NFV-SOL 003 V5.3.1 (2025-08): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/003/05.03.01_60/gs_nfv-sol003v050301p.pdf
[43]	ETSI GS NFV-SOL 005	ETSI GS NFV-SOL 005 V5.3.1 (2025-08): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/005/05.03.01_60/gs_nfv-sol005v050301p.pdf
[44]	ETSI GS NFV-SOL 014	ETSI GS NFV-SOL 014 V5.3.1 (2025-08): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; YAML data model for descriptor-based virtualized resource management". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/014/05.03.01_60/gs_nfv-sol014v050301p.pdf
[45]	ETSI GS NFV-SOL 018	ETSI GS NFV-SOL 018 V5.3.1 (2025-08): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; Profiling specification of protocol and data model solutions for OS Container management and orchestration". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/018/05.03.01_60/gs_nfv-sol018v050301p.pdf
[46]	ETSI GS NFV-SOL 020	ETSI GS NFV-SOL 021 V5.2.1 (2025-03): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; Specification of protocols and data models for Container Infrastructure Service Cluster Management". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/020/05.02.01_60/gs_nfv-sol020v050201p.pdf

Ref	Doc Number	Title
[47]	ETSI GS NFV-IFA 049	ETSI GS NFV-IFA 049 V5.4.1 (2026-01): "Network Functions Virtualisation (NFV) Release 5; Architectural Framework; VNF generic OAM functions and other PaaS Services specification". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/049/05.04.01_60/gs_nfv-ifa049v050401p.pdf
[48]	ETSI GS NFV-SOL 024	ETSI GS NFV-SOL 024 V5.3.1 (2025-09): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; Specification of protocols and data models solutions for VNF Generic OAM functions and PaaS Services". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/024/05.03.01_60/gs_nfv-sol024v050301p.pdf
[49]	3GPP TR 28.834	3GPP TR 28.834 V18.0.1 (2023-06): "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on management of cloud-native Virtualized Network Functions (VNF) (Release 18)".
[50]	3GPP TR 28.869	3GPP TR 28.869 V20.0.0 (2025-12): "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on cloud aspects of management and orchestration (Release 20)".
[51]	SP-251653	3GPP New SID: "Study on 6G Management and Orchestration". [Online]. Available at: https://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_110_Baltimore_2025-12/Docs/SP-251653.zip
[52]	ITU-T Y.3532	ITU-T Recommendation Y.3532 (05/2023): "Series Y: Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities; Cloud computing; Cloud computing – Functional requirements of Platform as a Service for cloud native applications". [Online]. Available at: https://www.itu.int/rec/T-REC-Y.3532-202305-I
[53]	ITU-T Y.3535	ITU-T Recommendation Y.3535 (02/2022): "Series Y: Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities; Cloud computing; Cloud computing – Functional requirements for a container". [Online]. Available at: https://www.itu.int/rec/T-REC-Y.3535-202202-I
[54]	ETSI GS NFV-SOL 004	ETSI GS NFV-SOL 004 V5.1.1 (2024-07): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; VNF Package and PNFD Archive specification". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/004/05.01.01_60/gs_nfv-sol004v050101p.pdf
[55]	ETSI GS NFV-SOL 007	ETSI GS NFV-SOL 007 V5.3.1 (2025-08): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; Network Service Descriptor File Structure Specification". [Online]. Available at: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/007/05.03.01_60/gs_nfv-sol007v050301p.pdf
[56]	ETSI GS NFV-SOL 026	ETSI GS NFV-SOL 026 V0.0.15 (Draft) (2026-04): "Network Functions Virtualisation (NFV); Protocols and Data Models; Specification of protocol and data model solutions for PIM services ". [Online]. Available at:

Ref	Doc Number	Title
		https://docbox.etsi.org/ISG/NFV/Open/Drafts/SOL026/NFV-SOL026v0015.docx
[57]	O-RAN.WG6.ASD	O-RAN.WG6.ASD-R004-V02.00 (June 2025): "O-RAN Work Group 6; Application Service Descriptor specification". [Online]. Available at: https://specifications.o-ran.org/download?id=889

3 Conventions

If the document includes binding material this section shall contain the following statement: The key words “must”, “must not”, “required”, “shall”, “shall not”, “should”, “should not”, “recommended”, “may”, and “optional” in this document are to be interpreted as described in RFC2119 **Error! Reference source not found.**.

4 Standardisation and open source in network softwarisation

4.1 Introduction

Standardization and open-source activities regarding network softwarisation of relevance to telecom networking have been set in place since early 2010’s decade. ETSI Industry Specification Group (ISG) on Network Functions Virtualisation (NFV) kicked off towards the end of 2012, and since then a plethora of related activities have been taking place both in standards development organisations and open-source communities. Not only new dedicated standards organisations and open-source projects have been created, but also other long-lasting organisations, like 3GPP, considered and developed activities in relation to their focused domains, like mobile network and management and orchestration. For instance, 3GPP developments of 5G were heavily influenced by NFV and other softwarisation perspectives.

Clause 3 introduces and researches on the status of standardisation and open-source activities related to network softwarisation, including virtualisation, cloudification, cloud-native and NFV, providing a brief summary of their scopes and relevant developments. The documented activities are further analysed to identify complementary and also, potentially, activities with similar scope and outputs, whenever there is such a case. Overall, the present clause describes the main “playground” for MNOs and other industry stakeholders to drive further interoperability in network softwarisation.

4.2 Standardisation activities

4.2.1 ETSI ISG NFV

ETSI ISG NFV has contributed to softwarisation transformation of the telecommunications industry for more than a decade. Established in 2012, its primary mission has been to standardize the shift from proprietary hardware and tightly coupled network functions to softwarised network functions running on cloud infrastructure.

A foundational concept developed by the group is the NFV architectural framework, which introduced the Management and Orchestration (MANO) components to manage the lifecycle of Virtualised Network Functions (VNFs) (refer to ETSI GS NFV 006 "NFV-MANO

Architectural Framework" [6]). At this stage, generic MANO components including the NFV Orchestrator (NFVO), VNF Manager (VNFM), Virtualised Infrastructure Manager (VIM), container infrastructure services management (CISM), CIS cluster management (CCM), and physical infrastructure management (PIM) and their interfaces to MANO-internal or external components are specified, including stage 3 protocol and data models in ETSI GS NFV-SOL 002 [41] for the "reference point between VNFM and EM/VNF", ETSI GS NFV-SOL 003 [42] for the "reference point between NFVO and VNFM", ETSI GS NFV-SOL 005 [43] for the "reference point between NFVO and OSS/BSS", ETSI GS NFV-SOL 014 [44] for "virtualised resource descriptors", ETSI GS NFV-SOL 018 [45] for "CISM and image registry APIs" (based on Kubernetes®, Helm® and OCI™), ETSI GS NFV-SOL 020 [46] for "CCM APIs" (based on Cluster API), and ETSI GS NFV-SOL 026 [56] for "PIM APIs and models" (based on DMTF's Redfish®), among others. Regarding descriptors, VNF and Network Service (NS) descriptors applicable in the lifecycle management of VNF/NS instances have also been specified up to stage 3, like ETSI GS NFV-SOL 001 [39] for "NFV descriptors based on TOSCA specification", and ETSI GS NFV-SOL 006 [40] on "NFV descriptors based on YANG specification". An additional aspect considered in ETSI NFV is the specification of generic Platform as a Service (PaaS) capabilities to support the OAM of VNF and other managed objects, specified in stage 2 and stage 3 deliverables as ETSI GS NFV-IFA 049 [47] and ETSI GS NFV-SOL 024 [48], respectively. The latter takes as a basic principle the profiling of existing cloud-native ecosystem solutions to capabilities of the platform regarding telemetry, network connectivity and configuration, among others. To test the implementation of these concepts, the ISG established a Proof of Concept (PoC) framework, encouraging collaborations across the ecosystem to validate NFV implementations. As technology evolved, the group's focus expanded to embrace cloud-native principles, resulting in changes that include Cloud-native Network Functions (CNFs) and the promotion of container infrastructure management, declarative APIs and Kubernetes-based orchestration for automation and agility features. Specifications of Container Infrastructure Service Management and CIS Cluster Management interfaces are some of the main deliverables during this stage.

In its ongoing evolution, ETSI ISG NFV is now advancing towards a "Telco Cloud" architecture, accommodated by ongoing Release 6 group specification work of ETSI GS NFV 008-1 "Telco Cloud architectural framework" [7] and ETSI GS NFV 008-2 "Functional requirements for the Telco Cloud" [8]. This platform-oriented framework is specified in a way to simplify and make more modular the Telco Cloud. Key principles guiding this evolution include simplification, infrastructure-agnostic, cloud-native integration, software portability across different infrastructures (e.g., hybrid cloud, edge cloud) and enhanced automation with Artificial Intelligence (AI). This new architecture includes features intended to support not only traditional network functions but also new workloads like AI and video processing on a unified cloud infrastructure, ensuring backward compatibility with existing NFV deployments. Through these efforts, ETSI ISG NFV has provided the critical standardization needed to foster an open ecosystem and promote the industry's transition to softwarised, agile networks.

4.2.2 3GPP

3GPP SA5 initiated a study of management of cloud-native VNFs in Rel-18, the result of which was captured in the technical report 3GPP TR 28.834 [49]. In Rel-19 the study was

generalized to cloud aspects of management and orchestration and concluded in 3GPP TR 28.869 [50], promoting the incorporation of management requirements and solutions for network functions that comply with cloud-native design principles into the 3GPP standards.

At the end of 2025, 3GPP SA5 kicked off the study of 6G OAM (FS_6G_OAM in SP-251653 [51]) in Release 20. The study includes management and orchestration related requirements and solutions for cloud-native network functions, driving the full adoption of cloud-native technologies in 6G networks.

4.2.3 ITU-T

Since 2020, ITU-T has launched standardization activities on cloud-native platforms and services within SG13 WP2. It has completed the release of multiple standards, including those for cloud-native PaaS functional requirements and management architecture (Y.3532) [52], container functional requirements and management architecture (Y.3535) [53]. Currently, ITU-T is developing standards centered on cloud-native service development principles, establishing a full-stack cloud-native standard system that covers infrastructure, platform capabilities, and service architecture.

Going forward, ITU-T plans to focus on foundational cloud-native capabilities in the cloud computing domain, and to continue promotion of collaboration on telco cloud architecture among standardization organizations such as GSMA, ETSI and 3GPP.

4.2.4 O-RAN ALLIANCE

O-RAN ALLIANCE is a standards organisation which was founded in February 2018 by five MNOs, and it now established as a world-wide community of MNOs, vendors, and research & academic institutions. Its mission is to re-shape the RAN industry towards more intelligent, open, virtualised and fully interoperable mobile networks [30].

O-RAN ALLIANCE is structured into several working groups and task groups. Among these, WG6 on “Cloudification and Orchestration” is tasked with the standardization in the area of cloudified infrastructure and NF workloads software management and orchestration. From the O-Cloud’s perspective, a component part of the O-RAN architectural framework specified in O-RAN.WG1.TS.OAD [31], two interfaces are specified:

- O2ims: produced by the Infrastructure Management Services (IMS) in the O-Cloud is used for the management of the O-Cloud infrastructure resources, and
- O2dms: produced by the Deployment Management Services (DMS) in the O-Cloud is used for the management of the NF Deployments, which are referred as the software deployment on O-Cloud resources that realizes, all or part of a Cloudified NF.

The stage 2 and 3 of the O2ims are specified in O-RAN.WG6.TS.O2IMS-INTERFACE [32], while for the O2dms, two profiles are specified, one based on the ETSI NFV protocol and data models for VNF lifecycle, fault and performance management (refer to O-RAN.WG6.O2DMS-INTERFACE-ETSI-NFV-PROFILE [33]) and one based on the Kubernetes native application programming interface (API) resources (refer to O-RAN.WG6.O2DMS-INTERFACE-K8S-PROFILE [34]). Descriptors for containerized NF Deployments are specified as application service descriptor (ASD) in O-RAN.WG6.ASD [57].

On the Services Management and Orchestration (SMO) side, the Network Functions Orchestration (NFO) serves as the consumer of the services and APIs produced by the DMS performing the orchestration of NF Deployments for Cloudified NFs, while the Federated O-Cloud Orchestration and Management (FOCOM) is the consumer of the IMS services to perform the federation and orchestration of O-Clouds and its resources.

While the focus of the O-RAN ALLIANCE regarding network softwarisation lies on the specification of standard solutions concerning the network softwarisation of the RAN and its NFs, the use of the O-Cloud is not limited to them. For instance, xApps can be deployed and operated as part of the Near-Real-Time RAN Intelligent Controller (Near-RT RIC), which is also considered as an O-RAN NF that can be deployed on the O-Cloud.

Finally, one of the focuses of O-RAN WG6 is the specification of the acceleration framework for RAN NFs, necessary for accelerating tasks and operations in high-performance and demanding functions, like the O-RAN Distributed Unit (O-DU). The framework, referred as Acceleration Abstraction Layer (AAL) comprises the framework itself, the specification of AAL-Common APIs, further split into application and management, and of dedicated AAL profiles for acceleration. An overview of the AAL specification scope and principles is available in O-RAN.WG6.AAL-GAnP [35].

4.2.5 GSMA

GSMA is the standardization organization fostering global interoperability of mobile networks and MNOs. In the domain of network softwarisation, GSMA has focused its activity in promoting the open-source and open standards for Telco Cloud infrastructure. Such activity has been performed by the Networks Group (NG), and aims at (as stated in their mission statement):

- developing and maintaining specifications for: hardware and software profiles, standardized reference points, technology agnostic requirements, interoperability, and
- promoting innovation: showcase the way, ahead of development curve, by building generic and flexible and yet concrete models and implementation architectures, deal with gaps and promote competitive technology evolution

In terms of specification outcomes, three PRDs have been produced by the NG:

- GSMA PRD NG.126: "Cloud Infrastructure Reference Model" [17]: the document identifies the abstractions and associated concepts to represent cloud infrastructure, and specifies cloud infrastructure software and hardware profiles for the infrastructure hosting virtualised and containerized network functions.
- GSMA PRD NG.133: "Cloud Infrastructure Reference Architecture managed by OpenStack" [18]: the document is a follow-up specification to NG.126 [17] focusing on OpenStack as a VIM for the cloud infrastructure used to host NFV workloads, primarily VNFs based on VM.
- GSMA PRD NG.139: "Cloud Infrastructure Reference Architecture managed by Kubernetes" [19]: the document follows a similar scope and principle as the NG.133 but dedicated to the profiling and specification of Kubernetes-based platform as cloud infrastructure.

The "Cloud Infrastructure Reference Model" is based on the infrastructure layer of the ETSI NFV architectural framework initially specified in the ETSI GS NFV 002 [3]. More precisely, as indicated in clause 6.2 of GSMA PRD NG.126 [17], the scope of the Reference Model encompasses the NFV Infrastructure (NFVI), with its hardware (compute, storage and network) resources, the virtualisation layer and the virtual compute, storage and network resources, as well as the VIMs in the NFV-MANO stack. Furthermore, the GSMA PRD NG.133 [18] explicitly references the VIM functional block as the focus of creating the software and hardware profiles when using OpenStack services as a VIM solution. Finally, the GSMA PRD NG.139 [19], even though not explicitly referenced in the document, covers the scope of the CISM and CIS clusters as specified in ETSI GS NFV-IFA 040 [5] and ETSI GS NFV-IFA 036 [4], respectively. In this case, the GSMA PRD NG.139 [19] specifies the software and hardware profiles to consider when using Kubernetes control plane, as CISM for orchestration of container workloads, and Kubernetes clusters, as CIS clusters.

It is worth remarking that the Anuket project (see clause 3.3.3) further develops the actual integrated implementations of the reference architectures and models specified in the referred GSMA PRDs; hence establishing a direct transfer between standards specification work and open-source related outcomes.

4.3 Open-source activities

4.3.1 Cloud Native Computing Foundation

The Cloud Native Computing Foundation® (CNCF®) hosts multiple opensource projects related to cloud native technologies that enable building, packaging, deploying, and performing lifecycle management and operations of cloud native software applications across private, public or hybrid cloud platforms.

Techniques like containers, container orchestration, service meshes, microservices, immutable infrastructure, declarative management, etc. can play an important role in modernisation, softwarisation and cloudification of telecom networks. Virtualisation and OS container technologies have already made it feasible to deploy network functions (RAN, Core, etc.) and other telco applications as VMs or OS containers on top of Commercial-off-the-shelf (COTS)-based infrastructure, enabling decoupling between software and hardware. Open-source technologies developed and hosted by CNCF can provide additional cloud-native operations and management solutions for the virtualised/containerised network functions and applications deployed in the network operator’s cloud infrastructure.

Table 1 provides a summary of "CNCF graduated" projects that are relevant to management and operations of cloud-native mobile networks.

Category	Project(s)	Description
Orchestration and Management	Kubernetes® (K8s®)	Orchestration system for automating deployment, scaling, and management of containerized applications. One of the sub-projects of Kubernetes is "Cluster API", which provides a declarative way to provision, upgrade, and operate multiple Kubernetes clusters. **Relevance:** Management and provisioning of Kubernetes clusters across different network domains (RAN, Edge, Core, IT etc.) for

Category	Project(s)	Description
		deployment of containerized network functions and other applications. Deployment, management and operations of containerized network functions and other applications deployed in the network operator's cloud-based network.
App Definition and Development	Helm®	As the Kubernetes package manager, Helm streamlines the installation and management of containerized applications on Kubernetes-based infrastructure. Helm uses Charts that are packages of pre-configured Kubernetes resources required to deploy an application. **Relevance:** Packaging, installation and release management of container-based network function and other applications deployed in the network operator's cloud-based network.
Observability	Prometheus®	Monitoring system that collects metrics from configured targets (systems and services) at given intervals, evaluates rule expressions, displays the results, and can trigger alerts when specified conditions are observed. **Relevance:** Collection and monitoring of performance and fault management related metrics associated with infrastructure, platform components, workloads and applications deployed in the network operator's cloud-based network.
	fluentd®	Collects events and logs from various data sources and writes them to target destinations, e.g., files, databases, cloud storage, etc. to provide a unified logging layer. **Relevance:** Collection and aggregation of events and logs related to infrastructure, platform components, workloads and applications deployed in the network operator's cloud-based network.
Continuous Integration and Delivery	Argo®, Flux	Continuous delivery and continuous deployment solutions that enable GitOps based management paradigm in Kubernetes-based deployments. **Relevance:** Enable automation via GitOps-based deployment, management, and operations related to infrastructure, platform components, workloads and applications deployed in the network operator's cloud-based network.
Service Mesh	Isti®	Service mesh layer for distributed applications providing a uniform and efficient way to secure, connect, and monitor services. **Relevance:** Domain-specific service mesh solutions like traffic enforcement, networking configurations, certificate management for container-based network function and other

Category	Project(s)	Description
		applications deployed in the network operator's cloud-based network.

Table 1: Overview and description of relevant CNCF projects

4.3.2 ETSI SDG OSM

ETSI SDG Open Source MANO (OSM) was launched in 2016 as a community-driven initiative under ETSI to deliver a practical implementation of ETSI NFV's NFV-MANO framework. Since its inception, OSM has evolved into an open-source reference orchestration platform designed to simplify the management of infrastructure resources and applications across heterogeneous cloud environments.

From the beginning, OSM has provided imperative orchestration for NFV, enabling automated deployment and configuration of Network Functions and Network Services. This capability ensures end-to-end service delivery, proper connectivity between components, and lifecycle exposure through standardized, cloud-native interfaces aligned with best practices from IT and public cloud domains.

To achieve this, OSM leverages and extends ETSI ISG NFV specifications, including information model (ETSI GS NFV-SOL 006 [40]), package formats (ETSI GS NFV-SOL 004 [54] and ETSI GS NFV-SOL 007 [55]), and API specifications (ETSI GS NFV-SOL 003 [42] and ETSI GS NFV-SOL 005 [43]). These standards enable deployment, management, and Day-1/Day-2 configuration of Network Functions and Network Services.

Since 2023, OSM has expanded its scope to also provide declarative, cloud-native orchestration for infrastructure, platforms, and services. This includes Kubernetes cluster management and orchestration of both network and IT applications. The declarative framework builds on proven open-source projects and CNCF technologies, such as:

- Cluster API for cluster lifecycle management.
- CrossPlane® for cluster and cloud resource management.
- Flux as GitOps framework for intent-based cluster and application management.
- Argo Workflows for defining cloud-native workflows.

4.3.3 Anuket

Anuket [16] provides a common model, standardized reference infrastructure specifications, and conformance and performance frameworks for virtualised and cloud-native network functions. These elements enable faster, more reliable onboarding into production environments and help accelerate digital transformation in communications. The initiative encompasses the collection and normalization of network operator requirements, open-source software development, and an industry verification program for ecosystem implementations, all integrated into a single effort.

By addressing common requirements for both infrastructure and network functions, Anuket reduces the number of individual clouds needed and shortens the onboarding process for new functions. Ensuring infrastructure conformance enhances interoperability, decreases development efforts, and minimizes the need for extensive customization.

Conformant infrastructure minimizes the level of customized support needed for each application. By creating conformant versions of products, suppliers can assure network operators that network functions will deploy and run reliably.

As a key partner in the Anuket project, GSMA has published the Cloud Infrastructure Reference Model, GSMA NG.126 [17], along with the Cloud Infrastructure Reference Architectures based on OpenStack, GSMA NG.133 [18] and Kubernetes, GSMA NG.139 [19]. The strategic partnership between Anuket and GSMA brings together the global MNO community with innovative technical solutions, supporting robust 5G and future network deployments.

4.3.4 Sylva

Sylva [20] provides a telco-grade cloud stack that enables scalable and reliable cloud infrastructure. It is an open-source platform serving as the technical backbone for Telco Cloud, allowing network operators to build flexible, scalable, and interoperable network infrastructures. Developed under the Linux Foundation Europe (LFE), its scope extends globally.

LFE's mission is to accelerate open-source collaboration across Europe, fostering innovation and supporting European companies in participating in worldwide initiatives, e.g., by means of contributions and validation centers.

Sylva is an open-source platform for deploying and managing Kubernetes clusters across various infrastructures, including OpenStack, VMware, Bare Metal, and more, with support for acceleration. It offers automation, lifecycle management, and a curated set of cloud-native tools tailored for enterprise environments. Sylva project provides three major releases per year.

Leveraging Infrastructure as Code (IaC) and GitOps principles, Sylva ensures full automation, high performance, and security in managing CNFs. Its concepts are based on the Anuket project (see clause 3.3.3).

Designed to address telecom challenges such as operational effort reduction, network agility, and security compliance, Sylva supports use cases like 5G Core, Open RAN, and edge computing through a standardized, open, and programmable platform. Sylva supports ETSI GS NFV-SOL 018 [45], which profiles the Kubernetes API, and ETSI GS NFV-SOL 020 [46], which profiles the Kubernetes Cluster API. It also supports O-RAN ALLIANCE defined O2-DMS (dedicated to the LCM of Deployments) and O2-IMS (dedicated to the LCM of the cloud infrastructure itself). It is used internally by major telecom network operators to promote a horizontal, open, and interoperable network management approach, maximizing open interfaces and improving energy efficiency. The project code and all repositories are available on Sylva Gitlab [21], which also contains the Sylva Roadmap and white papers from network operators.

4.3.5 CNTi test suite

The Cloud Native Telecom Initiative (CNTi) [22], launched by the Linux Foundation Networking (LFN) in 2024, aims to accelerate the adoption of cloud-native technologies

within telecom networks. CNTi focuses on identifying, developing, and promoting vendor-neutral best practices for cloud-native telecom networking applications.

The initiative focuses on three key areas:

- Developing cloud-native and Kubernetes-native networking best practices for telecom, including CNFs.
- Maintaining an open-source, supplier-neutral test suite to validate and verify adherence to these best practices.
- Creating a Cloud Native Telecom certification program to build Communications Service Provider (CSP) confidence, promote adoption, and reduce interoperability challenges.

Sylva and CNTi collaborate to advance the adoption of cloud-native best practices in support of telecom and edge computing requirements. A central element of this partnership is the CNTi Test Suite - a supplier-neutral, open-source testing framework that verifies CNFs against core cloud-native principles.

The Sylva Validation Center integrates the CNTi Test Suite into its toolchain to ensure workloads running on Sylva-compliant stacks meet cloud-native principles. This joint effort plays a vital role in reducing fragmentation and fostering interoperability.

4.3.6 Nephio

Nephio [23] is an LFN project designed to deliver carrier-grade, simple, open, Kubernetes-based cloud-native intent automation and common automation templates. Its primary goal is to simplify the deployment and management of multi-supplier cloud infrastructure and network functions (modeled as Kubernetes resources) across large-scale edge deployments.

Nephio streamlines automation tooling by enabling the deployment of cloud infrastructure and network functions without the need for out-of-band configuration management. It allows Kubernetes to handle the configuration of infrastructure and CNFs directly, replacing vendor-specific tools of NF-related management systems. Nephio utilizes the Kubernetes Resource Model (KRM) to manage CNF configurations and adapt Kubernetes clusters to meet CNF requirements.

Building on the foundation of previous releases, Nephio R5 [24] introduces further enhancements in functionality, usability, security, and scalability, preparing for broader adoption and new use cases. Key features of Nephio R5 include:

- O-RAN O-Cloud cluster provisioning
- bare-metal cluster provisioning
- support for ArgoCD GitOps workload reconciliation
- platform improvements
- security enhancements

Additional updates include several dependency upgrades and security fixes.

Nephio complements many existing open-source initiatives, aiming to develop open automation configurations that conform to O-RAN and 3GPP specifications. It can interface with ONAP for end-to-end service orchestration via open APIs for domain automation. EMCO, another LFN project, can also integrate with Nephio.

The project code and all repositories are available on GitHub [25], which also includes Nephio's Go API objects, Custom Resource Definitions (CRDs), libraries for manipulating them within KRM functions, as well as user-facing documentation and tutorials.

4.3.7 OpenStack and Tacker

OpenStack [36] has been since the early specification and development stages of NFV a key component to manage the compute node resource pools and virtualised networking providing necessary virtual resource for VNFs, and has served as the baseline open-source solution for VM-based network softwarisation.

Tacker is an OpenStack service for NFV orchestration with a general purpose VNFM to deploy and operate VNFs and Network Services on an NFV Platform [37]. It is based on the ETSI NFV's NFV-MANO Architectural Framework, see ETSI GS NFV 006 [6], and provides the OpenStack's NFV Orchestration API. Tacker, as it has become a mature project, is being used for large-scale commercial and production level as a Generic VNFM to manage all types of VNFs on multiple infrastructures, both for VMs and OS containers. Although integration in a telecom network operator environment is required, such as handling carrier-grade features like database redundancy, monitoring and user management, Tacker is deemed as simple and light solution covering a key are of management in network softwarisation, the VNFs. There are several deployment options, container based, VM-based and standalone depending on the network operator's environment.

One of the key aspects of Tacker is that it implements standardized APIs, in particular, diverse APIs specified in the ETSI GS NFV-SOL deliverables like VNF LCM, performance, fault and package management APIs from ETSI GS NFV-SOL 002/003 [41][42] and ETSI GS NFV-SOL 005 [43], in addition to the support of the VNF Package and VNF Descriptors from ETSI GS NFV-SOL 004 [54] and ETSI GS NFV-SOL 001 [39], respectively. More information about the current support of APIs and specifications is available in the Tacker architecture documentation [38].

4.4 Analysis of standardisation and open-source activities

4.4.1 Standards and open source overview

As introduced in previous clauses, there exist several and diverse standardisation and open-source activities related to network softwarisation. Some of these activities handle different kinds of deliverables, like standards versus open-source code, while some other activities produce same kind of deliverables and in certain cases with similar technical scope.

Figure 2 provides a high-level overview of the scope covered by the activities, both standardisation and open source, reported in previous sub-clauses. A corresponding placement of the "logo" of a respective organisation indicates that in its scope of work the organisation produces related deliverables such as specifications and/or code/integrated code. It is noted that a standards organization or open-source project might cover several areas, in which cases the corresponding logo is replicated wherever applicable.

NOTE 1: OpenStack® logos and project names are a registered trademark of OpenInfra Foundation.

NOTE 2: Kubernetes, Prometheus, Helm, fluentd, Istio, Argo logos and project names are a registered trademark of Linux Foundation in the U.S. and other countries.

NOTE 3: ETSI™ and 3GPP™ logos and names are a registered trademark of the European Telecommunications Standards Institute.

NOTE 4: O-RAN® ALLIANCE logo and name are a registered trademark of O-RAN ALLIANCE e.V.

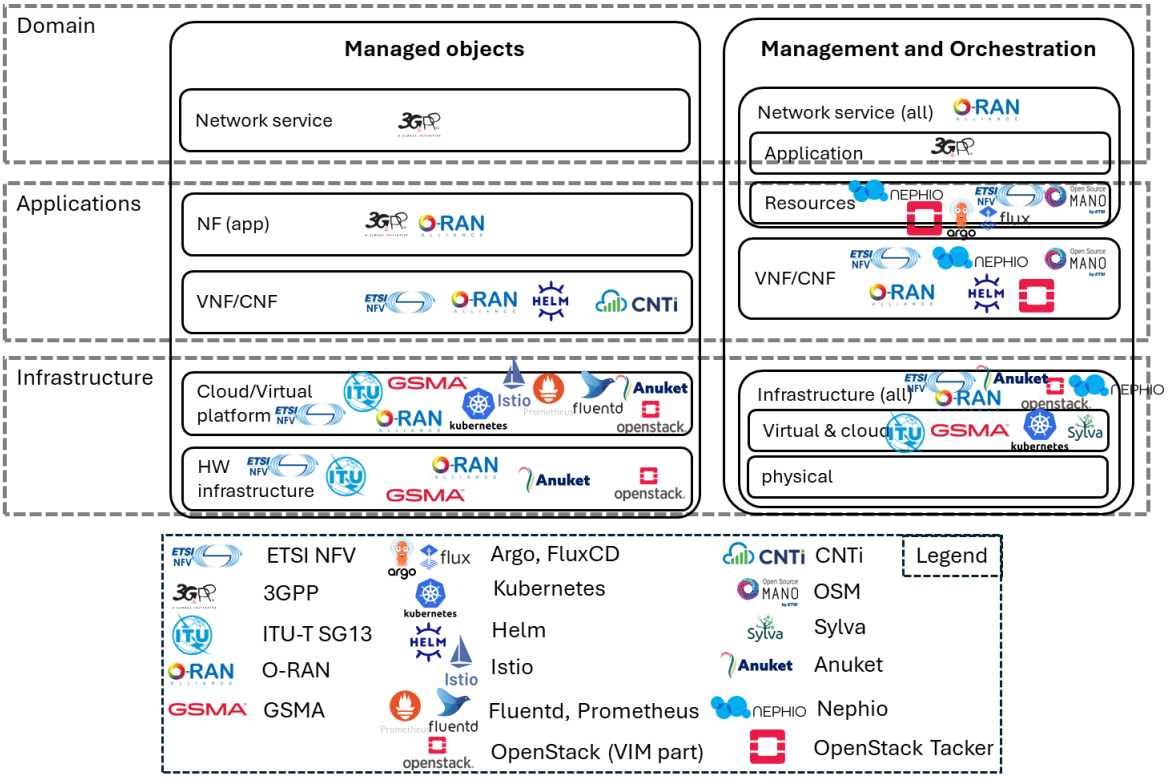

Figure 2: Overview of standards and open-source activities in network softwarisation

The activities are grouped into three major areas, with the following meanings.

- Infrastructure: it comprises the structure and composition of the hardware infrastructure and the virtualised/cloudified platform, including clusters for virtualised and containerized workloads. It also includes the necessary underlay and overlays networking connecting the infrastructure as well as the deployed virtual and containerized workloads. On the management and orchestration stack it comprises the management functions used to manage the infrastructure objects.
- Applications: it comprises the areas of NF softwarisation and the application logics. Regarding the first aspect, it further comprises the deliverables defining the requirements, composition and structure of software components and artifacts realizing the NF (commonly referred as "descriptors"). On the management and orchestration stack, it comprises the associated management functions for VNF/CNF, including lifecycle management, provisioning and telemetry.
- Domain: it comprises the areas of interconnecting and assembling NFs into composed objects to deliver connectivity and network services, for instance, the

composition and structure of a mobile network that an MNO can service and operate. On the management and orchestration stack, it comprises the aspects of network and application management, and the orchestration and management (monitoring and fulfilment) of resources realizing the composed network services.

The three areas are related to how mobile networks are built, deployed and operated in a network softwarisation environment. Common Cloud/virtual infrastructure provides the necessary resources where the NF application's software can be deployed and executed. When considering the full stack of management layers, these support the lifecycle management, monitoring and operation of the infrastructure resources, software components, and the mobile network as built from the composition of the necessary NFs. Hence, as depicted in Figure 2, it can be clearly observed that diverse standards and open-source organizations provide related assets in various areas, all necessary to deploy and operate the mobile network end-to-end.

Regarding the scope and deliverables of the various documented standardization and open-source activities, the following highlights can be summarized based on Figure 2. Firstly, several organizations and open-source projects scope their work in cloud/virtual platform infrastructure. A number of these open-source projects focus on different functionalities such as OpenStack for virtualised infrastructure, Kubernetes for containerized infrastructure, Fluentd and Prometheus for telemetry, etc. However, from a standards point of view, different organizations tackle same functional areas both in the domain of containerized and virtualised infrastructure, e.g., GSMA, ETSI NFV and O-RAN ALLIANCE. GSMA and O-RAN ALLIANCE had expanded scope in the area of hardware infrastructure; in the case of GSMA, to complement the specification of the profiles to also cover hardware aspects, and in the case of O-RAN ALLIANCE to cover the specifics of hardware acceleration (initially covered in Release 2 of ETSI NFV for VM-based deployments) as a key technical requirement to support RAN workloads.

A similar situation can be reported regarding the management and orchestration of infrastructure: both ETSI NFV and O-RAN ALLIANCE cover management and orchestration of virtualised and containerized workloads as well as the management of the physical and platform infrastructure. In the case of ETSI NFV, the management and orchestration of infrastructure leverage existing standards or open-source solutions wherever de-facto solutions are feasible, like Kubernetes, Cluster API, OpenStack and Redfish (see clause 3.2.1). In the case of O-RAN ALLIANCE, only Kubernetes is referred as open-source de-facto solution for the orchestration of containerized workloads, while alternative O-RAN specific solutions are currently pursued for Cloud infrastructure management.

Secondly, regarding the area of applications, from a standards point of view, both ETSI NFV and O-RAN ALLIANCE perform similar activity: in terms of management and orchestration of VNF/CNF, both specify their own solutions. However, it is important to remark that in the case of O-RAN ALLIANCE no new specific solutions are delivered: for the case of management of NF Deployments, there is a specific profile based on ETSI NFV, hence leveraging solutions to maximize the reuse of existing standards, and another profile based on use of Kubernetes API resources. In terms of descriptors, use of VNF descriptors is leveraged in the case of ETSI NFV solutions, while O-RAN ALLIANCE has specified an alternative descriptor, the ASD, for the case of containerized NFs.

Within the VNF/CNF management and orchestration, in the context of open-source projects, various solutions are provided by OpenStack Tacker, ETSI SDG OSM and Nephio, some of them acknowledging the implementation of standard specified APIs and data models, like Tacker and OSM concerning ETSI NFV standards and Nephio concerning the case of O-RAN ALLIANCE standards. In addition, it is regarded that Helm provides similar functionality for aspects regarding to the LCM of containerized NFs, focusing on the orchestration of multiple containerized workloads on top of Kubernetes infrastructure. As such, Helm Charts modelling and its tooling cover aspects of VNF/CNFs as managed object, as well as their LCM. It is important to highlight that Helm is also profiled as part of ETSI NFV specifications for supporting the orchestration and deployment of containerized workloads (refer to ETSI GS NFV-SOL 018 [45]).

Finally, in the area of domain management and network services it is worth highlighting the scope delineation between ETSI NFV and 3GPP standards, the former focusing on the management and orchestration for resource fulfilment of NFs, while the latter focusing more on the actual application, network and service management. O-RAN ALLIANCE covers both aspects since O-RAN Alliance also specifies RAN specific NFs (e.g., O-RU and the Near-RT RIC), but still leveraging 3GPP standards concerning the management of RAN NFs. In terms of open source, same projects referred for the case of VNF/CNF management and orchestration also cover aspects of resource orchestration for the deployed softwarised NFs.

4.4.2 Interplay between standards and open-source activities

The CNCF technologies and projects like Kubernetes, Helm, Prometheus, Argo, Flux, etc., can be seen as the de-facto baseline for building domain-specific solutions, including container-based network functions deployed in a mobile network. This is due to their open-source nature, rich ecosystem and widespread adoption, which is particularly evident in other relevant open-source activities. For example, OSM's framework uses CNCF technologies like Kubernetes, Cluster API, Crossplane, Flux and Argo to provide declarative, cloud-native orchestration for infrastructure, platforms, and services. Similarly, Sylva relies heavily on Kubernetes and Cluster API for provisioning and management of container clusters across heterogenous infrastructure and container-based network functions deployed on those clusters. Moreover, Nephio makes use of KRM and extends it where needed by leveraging Kubernetes' extensibility to define CRDs for its management framework.

This pattern – of CNCF technologies as de-facto standard or baseline – can also be seen in relevant standards organizations, and the reference architectures and frameworks developed by those organizations. For example, ETSI ISG NFV has incorporated open-source technologies (e.g., those developed by the CNCF such as Kubernetes, Helm) into its management and orchestration framework, NFV-MANO. The APIs exposed by de-facto open-source solutions (Kubernetes, Cluster API etc.) have been used to develop stage 3 (protocol and data model) profiles of the respective NFV-MANO standard interfaces.

Other solutions developed by the CNCF related to observability, service mesh, GitOps, etc. have also been used in the standard reference architectures, like the one developed by ETSI ISG NFV. Tools and solutions like Istio, Prometheus, fluentbit have been adopted as generic Operations, Administration and Management (OAM) tools and common platform services that support management and operations of container-based network functions in the NFV-MANO framework.

Finally, it is important to remark that open-source projects do also leverage standard solutions, e.g., by implementing specified APIs and data models. Examples are the OpenStack Tacker, which has implemented the VNF LCM, performance, fault and package management APIs from ETSI GS NFV-SOL deliverables (see clause 3.3.7), and also O-RAN Software Community (SC), which implements interfaces and data models specified in O-RAN ALLIANCE deliverables, like O2dms and O2ims, or the Nephio project, that also implements O2IMS provisioning services and O2DMS NF Deployment lifecycle management service (K8s profile).

4.4.3 Potential way forward – harmonization between standards and opensource

On the one hand, standard frameworks, such as NFV-MANO, lay out a reference architecture comprising of various functional components, services offered by those components and a standard way of communication between the internal components of the framework as well as between the framework itself and the external entities. This is achieved via standard interfaces, and protocols that enable information exchange over those interfaces, e.g., via RESTful APIs. Industry solutions implementing a standard functionality thus adhere to the functional requirements and interface specifications developed by the standard organization. This ensures multi-vendor interoperability between different products and solutions that implement standard APIs, e.g., based on REST, for inter-communication.

On the other hand, open-source solutions, such as those developed by the CNCF, focus on providing some concrete functionality developed by the relevant community as per their business needs and use cases. These software solutions often expose APIs – such as REST-based, gRPC-based – that the internal or external entities can use to interact with the software solution.

Standard frameworks (reference architectures, requirements, interface specifications) and open-source frameworks (software solutions, APIs) need not be mutually exclusive or direct alternatives to each other, rather they can co-exist in a harmonized way. Relevant open-source software solutions can be seen as implementation of some standard functions and services whereas their exposed APIs can be used as the stage 3 (protocol and data model) equivalent of standard interfaces. Furthermore, API schemas of open-source solutions can be considered while designing standard descriptors wherever appropriate.

Therefore, a comprehensive standard framework may comprise of a harmonized reference architecture, which consists of standard functions, services and interfaces as well as various relevant open-source technologies and solutions that may be seen as implementations of (some of) those standard functions and services. One such example is the NFV-MANO standard framework, which adopts some of the de-facto open-source solutions (such as Helm, Kubernetes, Cluster API) and their APIs as standard profiles (stage 3 specifications) for relevant standard interfaces and services in its reference architecture.

This approach of combining standard frameworks and open-source solutions not only fosters dynamic collaboration between standards and open-source organizations but also avoids duplication of work. It can also demarcate the common underlying layer made up of general-purpose open-source components from higher domain-specific standard abstractions provided by the dedicated management and operations framework. Standards can help

MNOs efficiently rely on open-source for the commercial deployment, e.g., considering timeline, stability, telco-grade, and developer involvement, while providing necessary standard domain-specific abstractions to ensure long-term support regardless of the adopted valuable open-source solutions.

5 Telecom industry softwarisation trends and maturity

5.1 Introduction

Technologies, an in particular those streamed up from other fields like Cloud computing, are in continuous evolution. But not only technologies evolve, the way these are being used to realise actual network deployments are also changing the landscape and telecom ecosystem.

From a technology and conceptual standpoint, NFV and cloud-native are regarded as the umbrella of a larger ecosystem of well-established current and devised new technology trends. In some cases, these are already being used by MNOs and other stakeholders for deploying and operating their networks, yet the scope of their usage can still be further expanded. In other cases, new technologies might not be used yet extensively, but are expected these can become key components in future networks, like the foreseen 6G. Clause 4.2 further describes key trends on softwarisation of telecom networks explaining their relevance and rationale for its use. Figure 3 provides and overview of the documented trends.

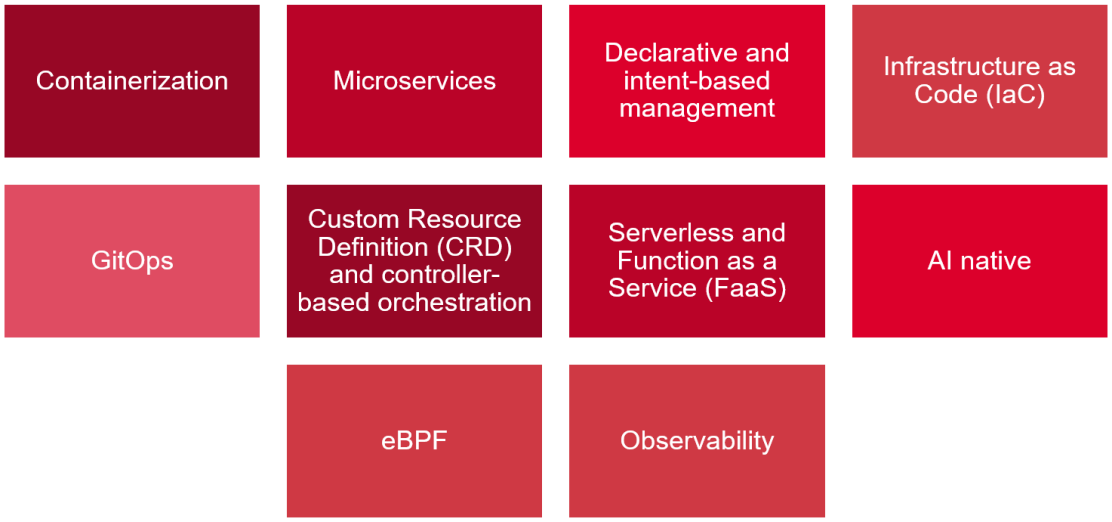

Figure 3: Overview of key trends in network softwarisation

From an actual network deployment and business perspective, the adoption of network softwarisation is not only changing the way MNOs build, deploy and operate their networks, but is also opening up to new business forms, telecom services and the overall industry stakeholder's ecosystem beyond MNOs and telecom equipment providers. For this, clause 4.3 further describes the new opportunities, business roles and typical/exemplary deployment scenarios of mobile networks when network softwarisation is employed.

5.2 Key trends on softwarisation of telecom networks

5.2.1 Introduction

The present clause describes the key characteristics of softwarised telecom network deployments as well as highlights the technology trends that the industry has been experiencing in the ongoing transformation of telecom networks towards softwarised, cloud-native and autonomous networks. These trends are at different maturity levels. Some of them have already been materialized, partially or fully, in commercial deployments while others are still being conceptualized and standardized in relevant organizations for commercial, standards-compliant deployments. The documented trends are summarized in the following subclauses in no particular order.

5.2.2 Containerization

Telecom network deployments have been seeing a shift from VM-based network function deployments to container-based deployments in the recent years. Containers offer various benefits such as lightweight footprint, faster provisioning, seamless scalability, rich open source and standards ecosystem around their orchestration and management. Due to these reasons, network function vendors and network operators are now leaning towards packaging and deploying Core and RAN network functions as OS containers instead of VMs.

Existing de-facto solutions like Kubernetes are used to orchestrate containerized workloads realizing the container-based network functions on the underlying infrastructure. Additional vendor-proprietary or open-source tools (some of them are mentioned in clause 3.3.1) can be used for other OAM aspects related to containerized deployments, e.g., packaging and versioning, monitoring and observability, security, etc.

In terms of implementation, container-based deployments can be divided into two types depending on the form of cluster node, either VM or bare-metal based: containers in VM and containers on Bare-Metal, each with its own advantages. Containers in VMs have high security and technological maturity, while containers on Bare-Metal have more advantages in performance, resource utilization, and architectural simplicity.

With the development of telecom services, containers on Bare-Metal have become a significant direction for infrastructure evolution due to their higher performance and other characteristics. However, its architecture, which relies on the host kernel, poses potential security risks. To address this issue, currently the industry is introducing secure container technology, which runs each container in a lightweight VM or microkernel. This solution brings higher security isolation boundaries through independent kernels.

In addition, the deep integration of container technology and hardware acceleration technology will become an important direction for future evolution. This is, for instance, reflected in the use of new kinds of computational hardware, such as Data Processing Unit (DPU), to offload container networks and storage stacks, intended to be orchestrated and managed through a unified framework, to achieve hybrid deployment of different types of containers such as native containers and secure containers.

5.2.3 Microservices

With the advent of Kubernetes and maturity of container runtimes, the stateless microservices design pattern has been gaining popularity in the IT industry as an alternative to traditional monolithic design. This trend has also permeated to the telecom industry where complex systems (like a 5G Mobile Core) are being modularized into individual services.

The use of containers as packaging and runtime technology for applications coupled with Kubernetes primitives (Deployment, Services, etc.) for their orchestration and deployment provide developers an effective way to design and build complex software applications as modular services.

To fully exploit the advantages offered by modern platforms, statelessness is one of the primary requirements a microservice needs to meet. It allows for greater scalability and resilience by removing the stickiness on a single instance of an application, instead allowing for containers to be replaced smoothly and without disruption.

Telecom network functions are categorized as large and complex applications, and the microservice transformation needs to ensure the high reliability and stability of telecom services. Therefore, it is necessary to implement effective governance and resilience enhancement for core processes such as "registration discovery & load balancing", "rate limiting, degradation & circuit breaker", and "tracing & alerting" in microservice operation.

As a dedicated infrastructure for microservice communication, service mesh has become a key component of the microservice governance system. For telecom scenarios, service mesh needs to undergo carrier-grade enhancements including performance optimization, advanced traffic governance, and secure communication.

5.2.4 Declarative and intent-based management

Management of telecom networks have traditionally followed an imperative paradigm where management systems focus on the fine-grained tasks (or steps) to execute in order to achieve a desired state. This paradigm is now being challenged and gradually replaced by a declarative management paradigm where management systems are given the desired state of the overall system ("system" here refers to the telecom network, either as a whole or in part) in a declarative manner. The management system itself is responsible for achieving this desired state declared by the user/operator. This shifts the focus on the end goal rather than the steps that are needed to get there.

Following the declarative approach, consumers of the system’s API(s) do not need to be aware of the detailed interim steps and granular operational tasks, which are taken care of by the system itself in an autonomous fashion. This decreases the OAM complexity and number of interactions between different systems, which is particularly beneficial in large-scale and complex systems that involve multiple components and countless interactions among them.

Declarative management paradigm also gives way to intent-based network management and orchestration, which is among the leading trends towards simplified management of modern telecom networks. Network operators or OSS/BSS can declare their intent about the desired state of the system either using declarative API objects (similar to how Kubernetes

does it) or in natural language or semi-structure data schema that can be translated by an AI-based system into necessary schemas and API calls towards the target OAM framework.

Industry alliances like TM Forum are leading initiatives to incorporate intent-based automation, such as TM921 [14], in telecom networks in order to realize autonomous networks, such as in TM Forum IG1523A [12] and IG1523B [13].

5.2.5 Infrastructure as Code

Infrastructure as Code (IaC) offers a way to provision and manage infrastructure resources in the form of code that can also be stored in Git repositories for version control. Infrastructure configurations are provided as code files and IaC tools take care of provisioning or updating the infrastructure resources as per desired configurations. These tools can also make use of the declarative approach described above, further simplifying infrastructure management tasks and offloading them from consumers to the tools.

In cloud-based telecom networks, deployment configurations can change frequently based on changes in network traffic and other factors. Infrastructure resources like servers, VMs, Kubernetes clusters may need to be provisioned or updated dynamically and on-demand. IaC tools can be very useful in tackling these dynamic infrastructure provisioning and management needs effectively.

5.2.6 GitOps

GitOps is another popular operational paradigm being already utilized by various telecom network operators in deploying and managing their networks. GitOps enables a version-control oriented approach to managing both infrastructure resources as well as workloads. Infrastructure configurations (as IaC), deployment manifests and artifacts (application packages, container images, configuration files etc.) are stored in a Git repository serving as the "single source of truth". GitOps controllers running in a deployment environment (e.g., a Kubernetes cluster) constantly monitor the declarative configurations stored in the Git repository and try to maintain the desired state in the actual environment. In case any changes are observed in the desired state, the controllers take necessary actions to match the actual state with the desired state accordingly.

Key characteristics of GitOps like declarative approach, version control, single source of truth, etc. enable automated operations, transparency and auditability, seamless rollback and avoid configuration drift by design.

5.2.7 CRDs, custom operators and controller-based orchestration

The Kubernetes' Operator Pattern [15] and its capability to be extended via CRDs enable leveraging the same Kubernetes control plane for controlling and managing various new kinds of custom resources, e.g., machines, servers, certificates, managed databases, complex applications etc. Each custom resource is controlled/reconciled by its own custom controller running on the Kubernetes cluster and is represented and configured via the Kubernetes API like any other Kubernetes resource.

Complex, stateful, and domain-specific software applications, such as cloud-native telecom network functions, can be modelled as Kubernetes custom resources (potentially comprising of further custom resources) and be managed via respective operators and controllers

residing in the extended Kubernetes's Control Plane. Introduction and adoption of such domain-specific custom resources for telecom use cases have recently gathered momentum in standards (e.g., ETSI ISG NFV) and open-source ecosystems (e.g., LFN Nephio).

5.2.8 Serverless and FaaS

In the future, in addition to providing legacy connectivity services, telecom network operators may expand their business to more diverse services. For example, utilize idle infrastructure resources in their mobile communication networks to provide computing services by offloading some computing tasks on terminals to the computing nodes provided by the network operator. Such computing services requires network operators to adopt a more flexible approach to dynamically deploy computing services and underlying computing resources on demand. In this context, serverless is regarded as a viable solution for dynamic deployment of computing services. The advantages of serverless include:

1) serverless can enable rapid elastic scaling, accurately matching resource demands to avoid over-provisioning or under-provisioning
2) the lightweight nature and fast startup of serverless functions make them well-suited for the distributed deployment architecture of telecom edge nodes (e.g., for Mobile Edge Computing), without occupying excessive edge resources
3) serverless allows billing based on the actual infrastructure resources used during application runtime, providing greater cost advantages

Function as a service (FaaS) platform providing serverless architecture deployment, orchestration, and management, allows serverless application providers to run and manage applications in the form of functions with an event-driven computing framework. FaaS platform can enable application providers to focus on the application development without worrying about the management of infrastructure resources.

As more telecom network applications adopt microservice architecture design, FaaS platform in the future may become a mainstream tool to simplify the deployment and operation on a telco cloud of service instances handling stateless tasks with short-triggered lifecycles. With FaaS, service instances of network applications can be quickly scaled in/out based on the real-time changes of traffic load, which can improve the resource utilization efficiency of telco cloud infrastructure.

5.2.9 AI native

AI native represents a future paradigm shift where artificial intelligence becomes an intrinsic, foundational element of telco software system architecture rather than an add-on feature. While NFV/SDN technologies introduced with the decoupling between software and hardware, and between control and data plane, network softwarisation establishes a broader concept of programmability, automation and virtualisation. AI native builds upon these foundations by embedding AI directly into the software-defined fabric, transforming networks from being programmable to becoming cognitive, enabling self-optimization, predictive operations and intent-driven management. Key characteristics of AI-nativeness include:

1) native integration: AI capabilities are deeply embedded into the software system design from inception, enabling seamless intelligence across all layers.

2) autonomous decision-making: Systems possess self-learning and self-optimization capabilities, making real-time decisions based on continuous data analysis without human intervention.
3) predictive intelligence: Proactive problem-solving through predictive analytics, anticipating issues like network congestion or security threats before they occur.
4) continuous evolution: Systems evolve through machine learning, improving performance over time without requiring manual updates.
5) multi-agent collaboration: Multiple AI agents work collaboratively across different domains to solve complex problems.

AI native influences both the architecture and interfaces of softwarised networks. Through data-driven control loops and cognitive layers, it refines network architectural models such as control planes and orchestration, and may evolve APIs and communication protocols to support intelligent, adaptive interactions.

The emergence of Agentic AI further extends this evolution, introducing autonomous, goal-oriented agents that can dynamically coordinate network functions. While Agentic AI primarily reshapes architectural constructs, such as decentralized decision-making and self-organizing systems, its impact on interfaces may focus more on intent-based abstraction rather than low-level protocol changes.

From a development perspective, AI-powered tooling and platforms are transforming how network software is designed, tested and operated, enabling higher-level abstraction, simulation and continuous integration of intelligence into network functions.

Thus, AI native should be viewed not as external disruptions, but as integrated, advanced features within the softwarisation continuum, deepening agility, enhancing autonomy, and preparing networks for increasing complex and intent-driven demands.

To practically realize AI-native architectures in future networks, the underlying cloud-native platform must natively support the orchestration of heterogeneous computing resources (e.g., GPUs, NPUs). Standardized frameworks for AI workload lifecycle management and high-performance data pipelines are essential to prevent creating isolated AI silos within the telco cloud.

5.2.10 eBPF

eBPF, originally an acronym for "extended Berkeley Packet Filter" but now a standalone term, is a powerful virtual machine embedded within the Linux® kernel [26]. eBPF allows programs to run in a privileged context, transforming the kernel from a passive component into an active, programmable platform. eBPF programs are highly portable across kernel versions, can be updated atomically without disrupting workloads or requiring reboots, and are rigorously verified at load time to prevent system instabilities. These event-driven programs execute directly in the kernel, with user-space applications responsible for loading and interacting with them via eBPF Maps [27].

NOTE: Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Inherent strengths of eBPF include:

1) programmability: offering dynamic, real-time customization of kernel behaviour without source code modification, accelerating innovation while maintaining kernel safety and stability through strict verification. This empowers rapid iteration for DevOps and Site Reliability Engineering (SRE) teams
2) visibility and control: enhancing through deep observability, granular traffic shaping (QoS), advanced filtering for security, and real-time threat response directly at the kernel level
3) low overhead: ensuring minimal resource impact, making it ideal for high-density, cost-sensitive cloud and edge environments. It integrates seamlessly across various operating systems, reducing architectural complexity by eliminating the need for sidecars
4) substantial business impact, leading to reduced operational complexity, accelerated time-to-production for new features, and a future-proof platform adopted by leading enterprises, hyper scalers, and MNOs

eBPF is to the OS kernel what JavaScript is to the web browser: it transforms passive infrastructure into an active, programmable platform for innovation. This broad capability fuels eBPF major use cases across cloud-native environments. In observability, it enables kernel-level monitoring of infrastructure, networks, and applications, with deep Kubernetes integrations seen in many available products. For security, eBPF provides runtime threat detection, file integrity monitoring, and container fingerprinting. While networking is surprisingly underrepresented in product count, eBPF underpins critical Kubernetes Container Network Interface (CNI) plugins, offering significant performance gains. Beyond these, eBPF is also being used for innovative use cases like cloud cost attribution and monitoring large language model (LLM) API calls [28].

In essence, eBPF provides the foundational technology to build truly resilient (antifragile), secure, and performant telecom networks capable of adapting to the complex and evolving threat landscape of the future [29].

5.2.11 Observability

In a cloud native environment every deployed component typically produces outputs which can be monitored to gain insights into a service's performance and internal state. The typical types of observability data are metrics, logs and traces. These are typically collected in a central observability platform, ideally in near real time, allowing for immediate adjustments when necessary and supporting efficient issue analysis. To further enhance this process all observability data is enhanced with meta data providing additional context. Moreover, automated anomaly detection and alerting rules can be used to increase the reaction speed to arising issues.

This observability approach is tailored to the current day highly distributed cloud-native environments enabling a quick detection and resolution of problems across the entire system.

In practice, the following components are needed:

1) Data needs to be scraped or exported from the data sources across the environment and moved into a centralized.

2) The collected data needs to be exposed through a monitoring interface allowing for further analysis via queries or charts.
3) Built on this foundation alerting, detection and automation tools can be implemented.

5.3 New prospects, business roles and deployment scenarios of telecom networks

5.3.1 Deployment maturity levels in softwarised telecom networks

MNOs have been using standards and de-facto Cloud computing solutions to softwarise their mobile networks since mid-2010's. After almost a decade of widespread deployment of virtualised and cloudified telecom networks, industry stakeholders have many more insights about the maturity of such network deployments. As the starting context of MNOs were not the same, such as different country regulations, product lifecycles, target deployments, etc., the levels of penetration of network softwarisation into each MNO have not been the same.

When it comes to softwarising telecom networks, two reference model approaches can be regarded as the baseline: a) based on standards, and b) based on open source. It is also possible a combination of the two approaches, as per MNOs suitability and availability of solutions. As a matter of, standard based solutions have leveraged open source for diverse aspects, predominantly for infrastructure, container and platform as a service functions.

Regarding the standards approach, the key main outcome has been the NFV reference model produced by the ETSI NFV (refer to clause 3.2.1). As for the open source approach, solutions within and around the Kubernetes ecosystem have become prominent (refer specifically to clause 3.3.1 and other sub-clauses in clause 3.3).

Figure 4 illustrates a deployment maturity quadrant. Such illustration aims to characterise the maturity and target network deployments by MNOs. The X-axis represents the maturity level: on the left, simpler use cases, like solutions for deploying only an NF, while on the right, more complex ones; the Y-axis represents the approach: on the top, more NFV and standards-centric, while on the bottom, plain container and open-source-centric.

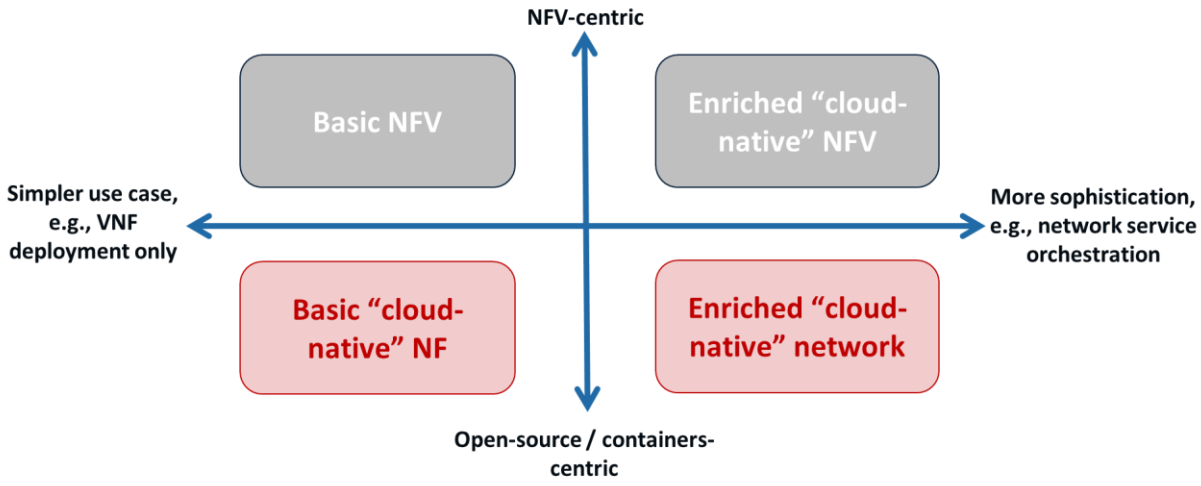

Figure 4: Deployment maturity level quadrant for softwarised telecom networks

The four resulting positions on the quadrant are described as follows:

- **Basic NFV**: leveraging early delivered ETSI NFV specifications and/or its general architectural framework, focusing on simple use cases of VNF deployment, and using prominently VM-based deployments.
- **Enriched "cloud-native" NFV:** leveraging ETSI NFV specifications and/or its general architectural framework enhanced for container-based deployments and with additional functionality to cover more complex use cases like end-to-end network orchestration, automation, etc.
- **Basic "cloud-native" NF:** leveraging more open-source solutions without referencing ETSI NFV specifications (i.e., no reference to the ETSI NFV framework from a functional point of view) to cover basic use case to enable simple NF deployment, and targeting prominently container-based deployments.
- **Enriched "cloud-native" network:** leveraging more open-source solutions without referencing ETSI NFV specifications (i.e., no reference to the ETSI NFV framework from a functional point of view) and addressing additional functionality to cover more complex use cases like end-to-end network orchestration, automation, etc.

Simpler network deployments, as depicted on the left-side of the X-axis on Figure 4, are regarded to be the two main entry points for MNOs. Figure 5 illustrates possible entry points and evolution paths for MNOs, taking as a baseline the maturity quadrant from Figure 4. As depicted on Figure 5, "Basic NFV" was the early entry point for many MNOs to start with the softwarisation of 4G/EPC. The availability of a reference architecture for NFV and the maturity of VM-based technologies for virtualisng the core network functions were the main drivers for this entry point.

With respect to the evolution paths, on the one hand, MNOs that started with the virtualisation of 4G/EPC may have further evolved or changed in three possible directions, in particular when realizing the deployment of 5G/5GC network and towards the support of container-based NF deployments:

- Functionality enriching the "Basic NFV" with cloud-native support and more sophisticated automation and network service level orchestration;
- Migrating towards an "Enriched cloud-native network" leveraging more open source and focused on container-centric deployments; or
- Changing the technologies from standard-based and VM-support to "Basic cloud-native NF" using de-facto open-source solutions, without implementing sophisticated automation and network service-level orchestration support.

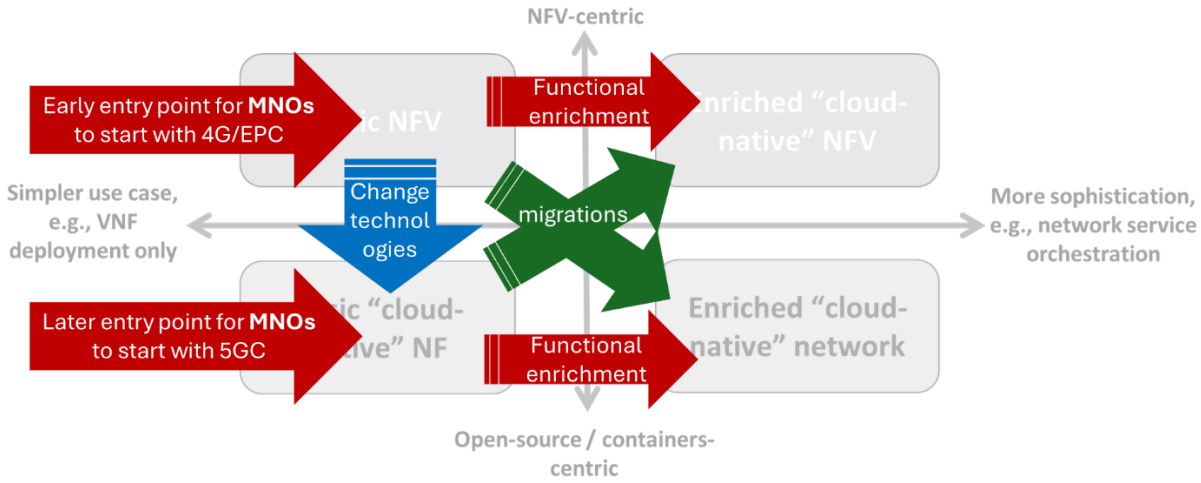

Figure 5: Entry points and potential evolution paths towards more network deployment sophistication

On the other hand, MNOs that started with the virtualisation of 5G/5GC using basic cloud-native NF and container-based support leveraging mostly open-source solutions may further evolve by:

- Functionality enriching the "Basic cloud-native NF" towards more sophisticated features of automation and network service-level orchestration using the same open-source-centric approach; or
- Migratin towards "Enriched cloud-native NFV" on a standard-centric approach leveraging and targeting longer-term support provided by such standards.

Regarding the possible migration paths, MNOs expect a more enriched cloud-native network support in future deployments, like 6G. Such a trend confirms the objective of MNOs to cover more sophisticated use cases of mobile network softwarisation, with greater levels of automation and orchestration and toward leveraging lightweight containerization and open-source based solutions to underpin the baseline infrastructure and platform capabilities for softwarised mobile networks. Standardization needs are also identified (see clause 5) to address complexity, interoperability and automation issues that cannot be simply resolved with de-facto open-source based solutions.

5.3.2 Business roles and stakeholders in softwarised telecom networks

The softwarisation of telecom networks has substantially evolved the role of MNOs and other telecom industry stakeholders. This has mainly been driven by the network softwarisation principles, like the decoupling between physical infrastructure resources and NF's software, the use of virtualisation and cloudification for the deployment and operation of the software, the use of COTS-based hardware, and the disaggregation of the mobile network in more granular and specialized NFs and network services.

Compared to legacy physical NF deployments, in softwarised telecom networks the number of stakeholders has expanded beyond MNOs, telecom equipment suppliers, OSS/BSS solution suppliers, and system integrators. This growing number of roles in softwarised telecom networking is represented in Figure 6.

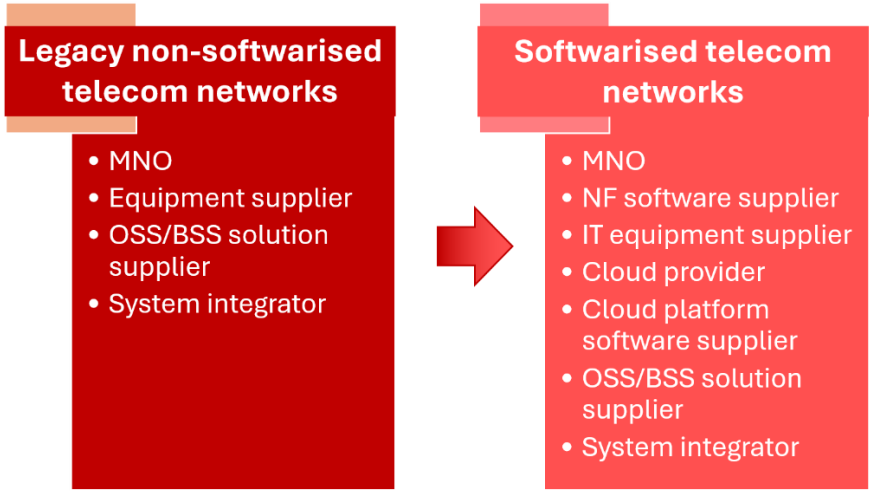

Figure 6: New stakeholders and roles in softwarised telecom networks

Thus, in softwarised telecom networks, the following roles can be identified, including:

- MNOs, who operate the softwarised mobile network be means of operating support systems & business support systems (OSS/BSS)
- NF software suppliers, who provide the NF software to be deployed and operated on virtualised and cloudified infrastructure. The NF software, together with the underlying Cloud resources, build the softwarised mobile network
- information Technology (IT) suppliers, who provide generic and general-purpose IT hardware such as compute servers and storage which conform the Cloud resources infrastructure for virtulisation and cloudification. IT suppliers may also further integrate various compute, networking and storage subsystems, such as chipsets, network interface cards, storage drives, etc. These are not further detailed, neither in Figure 6 nor in Figure 7
- Cloud providers, also referred as Hyperscalers, who provide resources on Public Clouds as infrastructure as a service (IaaS) and containers as a service (CaaS),
- Cloud platform software suppliers, who provide only the software enabling the cloudified and virtualised platform
- OSS/BSS solution suppliers, who provide tools and software systems used by the MNOs to operate the mobile network, and
- system integrators (SI), who integrate the various components and systems to deliver an end-to-end solution. SIs can be specialized to integrate infrastructure, Cloud platform software, NFs and OSS/BSS systems, or any combination thereof

Note that a telecom stakeholder, like for instance an MNO, may play several roles, e.g., considering in-house development of specific OSS/BSS software, end-to-end system integration, etc. As another example, Cloud providers may also provide Cloud platform software to run on-premises (private Clouds), thus also performing the role as a Cloud platform software supplier.

Figure 7 illustrates the main roles and assets concerning softwarised telecom networks. The arrows indicate which role relates to which asset to build and operate the softwarised telecom network. The figure also depicts the possibility of deploying and operating the

network (or part of it) either or both on Public and on-prem Clouds. In the former case, the Cloud provider provides both the HW infrastructure and Cloud/virtual software platform. In the Figure 7, HW infrastructure and Cloud/virtual platform are represented spanning both the on-prem and Public Clouds, but different subsystems and instances of these are in practice realized (the figure only provides a simple representation). The role of the SI surrounds the whole set of assets, as an SI could integrate all systems end-to-end or only a subset of these, depending on the choice of deployment by the MNO and the reuse of existing and deployment of new assets.

The assets in Figure 7 are meant to include different aspects and functionalities of control, data, management and security functions, without further details provided.

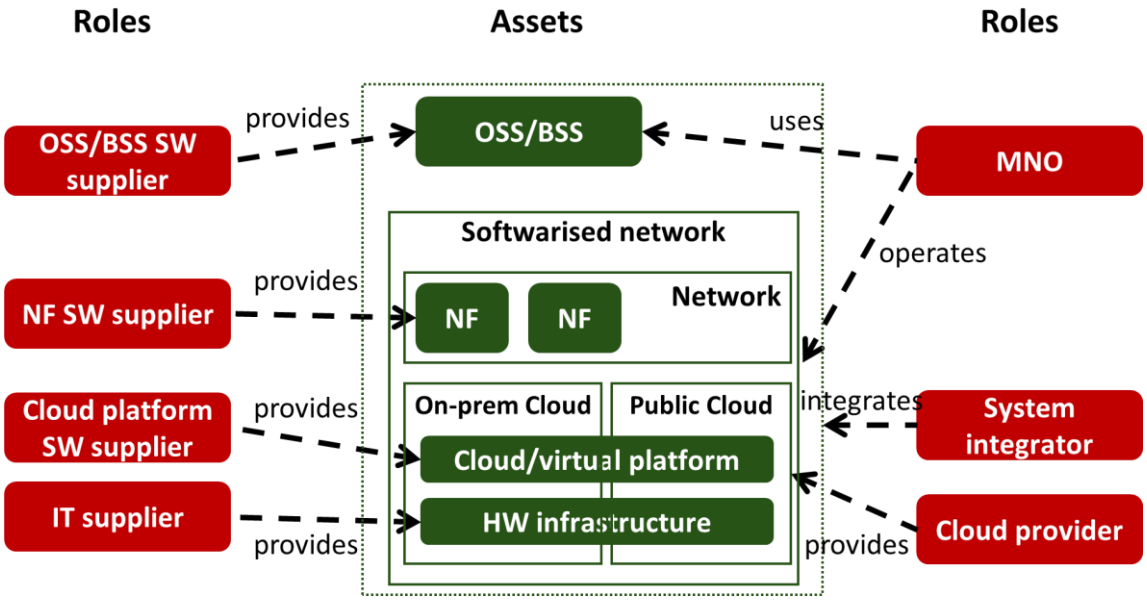

Figure 7: Major roles and assets in softwarised telecom networks

The expansion of roles and assets exemplifies the openness that disaggregated software telecom networks brings to MNOs and the possibility of new entry players into the telecom network industry, like Cloud platform SW and IT suppliers. Such an aspect offers more choices to MNOs to choose among different and diverse solutions that can suit better their network development, business and consumer service end goals.

Another aspect to highlight is the additional network deployment options available to MNOs, including the use of private/on-premise infrastructure or Public Cloud, or even hybrid cases whereby part of network assets is deployed on-premise while others on Public Cloud. This aspect is further elaborated in the following clause 4.3.3.

5.3.3 Deployment scenarios: on-premises, public Clouds and hybrid Cloud

As introduced in clause 4.3.2, network softwarisation offers new options to MNOs to realize the deployment of mobile networks. While legacy mobile network generations until 4G made use of on-premises dedicated infrastructure (private Cloud), either based on advanced telecommunications computing architecture (ATCA) or COTS-based if already virtualised, MNO's strategy for 5G and future generations is envisioning the use of also public Cloud

infrastructure to deploy part of the mobile network and other systems, like OSS/BSS components. Various factors for leveraging public Cloud infrastructure include reducing total cost of ownership (TCO), expanding network capacity for peak demand, and increasing network resiliency for disaster recovery, among others. MNO's specific use cases and business goals play an important role regarding the needs and considerations of such various factors. Nonetheless, the expansion of the infrastructure beyond on-premises (private Cloud) is an important factor to consider.

The expansion of infrastructure is becoming overtime more feasible due to the commoditization of the hardware, virtualisation/cloudification platform software and platform services usage across different Cloud domains. Use of common COTS-based hardware and open-source based IaaS/CaaS software plays an important role.

Because not all MNO service priorities and context (e.g., location, regulation, sovereignty requirements in different countries) might be the same, it is MNO's choice which deployment models to follow and which and where telecom network workloads are expected to be run. Nonetheless, from a service performance perspective, the suitability of telecom network workloads for deployment on-premises and on public Cloud needs to be also considered. Table 2 summarizes and exemplifies various kinds of telecom network workloads and their suitability to be run on-premises and public Cloud.

NOTE: Table 2 is only indicative, it is up to MNO's decision to determine the placement of relevant telecom network and other application workloads.

On-premises	Public Cloud
• Telecom network workloads with strict latency requirements, e.g., RAN functions. • Telecom network workloads with high ingress/egress bandwidth requirements, e.g., 5GC user plane functions. • Telecom network workloads with telecom-specific resource usage, e.g., RAN functions. • Telecom network workloads with predictable and generally high resource usage patterns. • Telecom network workloads with specific security and regulatory related requirements.	• Telecom network management workloads, e.g., OSS/BSS components such as for observability, visualization, optimization tools, network planning, etc. • Supporting application workloads with compute intensive processing, e.g., data analytics functions, AI training, etc. • Supporting application and network management workloads with transactional patterns, e.g., billing and customer relationship management (CRM) functions.

Table 2: Examples of telecom network workloads on-premises and public Cloud

As the deployment options showcase a no "one size fits all" solution as well as the interest of MNOs on diverse options, it is important to highlight that solutions provided are not unique to a single MNO's deployment, thus, even in some cases MNOs may choose to deploy and migrate back-and-forth workloads between on-premises and public Cloud depending on specific MNO needs and circumstances, e.g., for disaster recovery, additional network capacity, etc. This approach is referred as hybrid Cloud, whereby the MNO leverages virtualised and cloudified infrastructure from both domains.

This case highlights the importance of enabling mobile network deployments, and other components, to be ported to diverse environments. To achieve this, use of standard APIs and models and de-facto industry-wide cloud-native open-source solutions play a key role, as well as a framework and components that help shielding the differences of lower-layer platform implementations towards higher-layer management and orchestration systems.

6 MNO's pain-points, requirements and gap analysis

6.1 Overview of MNO challenges

The softwarisation of telecom networks introduces challenges that do not arise in generic cloud-native environments. These challenges primarily come from three main sources:

- Technical constraints arising from telecom-specific performance requirements.
- Operational complexity caused by heterogeneous virtualisation models across NFs.
- Lack of alignment across standardisation and open-source initiatives.

As MNOs progress toward software-driven networks, they face several recurring challenges across both technical and operational domains. Such technical and operational challenges represent major bottlenecks hindering the cloud-native transformation of telecom networks. Figure 7 represents the headlines of the documented challenges grouped into high-level categories regarding operational, networking, NF softwarisation and security.

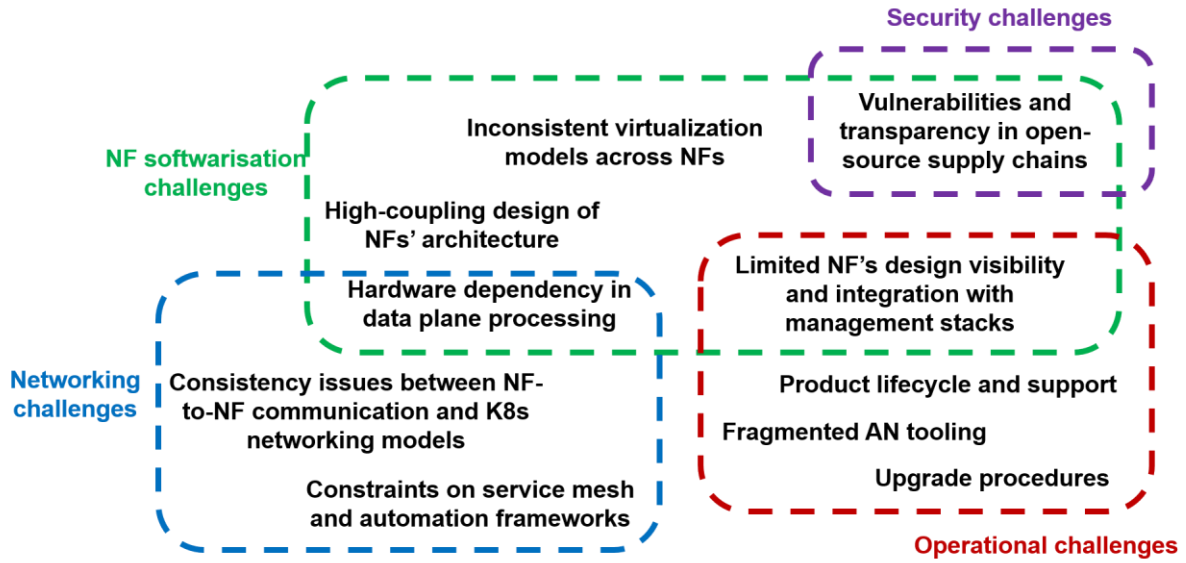

Figure 8: Technical challenges grouped into high-level categories

These structural challenges are outlined at a high level and further detailed in clause 5.2.

6.2 Technical and operational pain points

6.2.1 Hardware dependency in data plane processing

High-performance data plane workloads depend on hardware-optimised technologies such as Data Plane Development Kit (DPDK), single root input/output virtualization (SR-IOV), and non-uniform memory access (NUMA). However, Kubernetes is designed around the principle that workloads should be portable across any node with minimal hardware

dependency. This inherent mismatch makes it difficult to apply hardware-specific optimisations consistently at the container or workload level. As a result, CNF onboarding efforts and performance validation become resource-intensive, slowing down deployment cycles.

6.2.2 Consistency issues between NF-to-NF communication and Kubernetes networking models

Kubernetes typically assigns a single network interface to each Pod, commonly referred as primary interface. In contrast, NF-to-NF communication (e.g., 5G core network functions like UPF, AMF, SMF, PCF) may require specialised transport handling that cannot be accommodated via the primary interface. To support such requirements, NF implementations commonly rely on secondary network attachments enabled by CNI meta-plugins such as Multus. This kind of CNI meta-plugins allow Pods to attach multiple network interfaces, but these operate outside the scope of Kubernetes' native load balancing model.

Operationally, this leads to the following challenges from an MNO's perspective:

- NF vendors implement secondary interface usage in different ways, while the resulting operational complexity is handled by the MNO.
- Lifecycle management of secondary interfaces (connectivity validation, exclusion from Kubernetes LoadBalancer scope, fault isolation, and troubleshooting) is largely performed by the MNO.
- Telecom-specific requirements, such as multiple interfaces and protocol constraints, introduce additional complexity compared to general IT workloads.

As a result, while secondary interfaces address functional connectivity requirements, the associated operational responsibility and complexity are largely absorbed by the MNO, which goes beyond its responsibility to provide and maintain the network service level connectivity.

6.2.3 Constraints on service mesh and automation frameworks

Secondary interface configurations often bypass or conflict with the traffic paths assumed by service meshes such as Istio. This limits the use of cloud-native automation and observability capabilities —such as traffic steering, telemetry, and policy-based control— and presents obstacles to the microservice-oriented decomposition of NFs. As a consequence, the use of secondary interfaces limits the practical applicability of cloud-native operational models in telecom environments.

Furthermore, networking for NFs is still not suitable for cloud-native environments because static network configurations are still required, while networking in cloud-native environments assume more dynamic and non-stitched connectivity.

Finally, while adoption of networking principles in cloud-native environments with the use of service meshes and Kubernetes would facilitate the integration into such environments, planning of capacity and QoS for the flat networking principles in cloud-native environments raises difficulties for the MNO to plan the necessary underlying networking resources and monitor the resulting networking patterns.

6.2.4 Inconsistent virtualisation models across NFs

The coexistence of VNFs, CNFs, and pseudo-microservice architectures prevents MNOs from establishing a unified operational model for lifecycle management, CI/CD, observability, and automated healing. Hybrid operations (VNF + CNF) are expected to persist for the foreseeable future, making standardisation and automation particularly challenging. In some cases, workloads labelled as CNFs still run as monolithic single-Pod implementations, limiting the benefits of cloud-native design principles.

6.2.5 High-coupling design of NFs' architecture

The stateless design of telecom network functions refers to an architecture model for cloud-native deployment, where components processing service logic (such as data packets or signalling) in network functions are designed to store no session, connection, user context, or other states. All necessary information is obtained through external components or retrieved in real time from shared storage, enabling rapid fault recovery, elastic scaling, and flexible deployment and upgrade.

In a 5G era, some network functions still adopt the stateful design in terms of product development, where the service logic processing is closely coupled with stateful data in the same component and deployed in the same VMs. With this kind of design architecture, network function instance failures can lead to state information loss and service interruptions, resulting in slow recovery. In case of OAM operations like software upgrade, it requires the migration or rebuild of state, which brings more complexity and higher possibility to cause service traffic interruption.

Even when stateless design is adopted, e.g., via a Shared Data Layer (SDL), MNOs face operational challenges in managing the strict dependency between stateless NF instances and the centralized data layer. Standardized fallback mechanisms and latency-aware state synchronization are required to prevent single points of failure in external storage.

6.2.6 Product lifecycle and support

Adoption of software design principles and practices in virtualised and cloud-native environments enable MNOs to innovate at a faster pace and make the network deployment and maintenance more agile. However, MNOs are also required to keep providing their services for long periods of time (+10 years in most of the cases) in a multi-generational network environment. The issue arises by the fact that product lifecycle and support for different components involved in the softwarised network like infrastructure, platform as well as NFs is too short.

This issue is further expanded by limited choice of platform solutions for NFs. In principle, horizontal decoupling available in network softwarisation should allow MNOs to evaluate and have at disposal more options for infrastructure, platform and NFs. However, it is typically the case that NFs can only support certain platform solution, narrowing down the options for MNOs to integrate with existing or new planned platform assets.

6.2.7 Limited NF's design visibility and integration with management stacks

While legacy NF deployments based on ATCA-based platforms did not necessarily disclose details of implementation of the NF, in a network softwarisation scenario, NFs make use of

generic platform and virtualised/cloudified infrastructure. Even though design implementation of NFs need not be disclosed and are still left as an implementation matter, the lack of and limited visibility into the NF's design (e.g., structure of the NF in terms of deployable software components, dependencies and relationship between components, mapping information between components and infrastructure resources, etc.) makes MNOs integration and deployment of NF's complex and time-consuming.

A more concrete issue concerns the difficulty to map across the different layers of network and compute in the infrastructure physical layer (e.g., compute nodes and L2/L3 network devices), virtualised/cloudified infrastructure (e.g., clusters and virtual network segments), workloads deployments (e.g., Pods and their interfaces) and the application layer. Such mapping is needed so that troubleshooting procedures can be handled to keep network services reliable and highly available.

Furthermore, while NF's disaggregation becomes a reality, OAM for NFs becomes a challenge due to the lack of unified OAM information and management procedures. Considering that platform and infrastructure resources are generic to potentially many NFs, specific OAM procedures assumed from legacy implementations become a challenge for MNOs and make integration to other MNO's OSS assets complex and sometimes unfeasible, thus forcing unwillingly MNOs to adopt specialized and vendor-specific EM/NM systems.

6.2.8 Upgrade procedures

Complex and lengthy upgrade procedures are required to realize the infrastructure, platform and NF upgrades. This is aggravated by the coupling in actual NF implementations among the different components. In many cases these are induced also by limited in-service software upgrades (ISSU) support. The great number of NF instances that MNOs in practice need to handle for nation-wide mobile networks and the short/limited maintenance windows make the upgrade process even more complex. Furthermore, as the network gets expanded to cover nation-wide coverage, the number of resources and software components building the infrastructure and platform increases considerably, potentially expanding across different resources generations and software versions. This also brings the complexity of having to validate the upgrades of infrastructure and platform for the NF instances workloads in a continuous flow.

Even though, network softwarisation brings more flexibility and agility in handling the network upgrades compared to legacy physical network implementations, overall, upgrades become a pain point for many MNOs, which, depending on the size of the network can result in lengthy procedures and year-level development projects.

6.2.9 Vulnerability and transparency in open-source supply chains

As network softwarisation heavily relies on open-source software and third-party containers, the attack surface expands leaving MNOs vulnerable to zero-day vulnerabilities and supply chain attacks, increasing the operational burden of security compliance.

6.2.10 Operational complexity of fragmented Autonomous Network (AN) tools

Integrating diverse tools for Autonomous Networks (AN) introduces extreme operational complexity. Without standardized observability and clear intent mapping between domain-specific closed-loop controls, simultaneous autonomous actions can lead to resource conflicts and unpredictable network behaviors.

6.2.11 Siloed domain-specific management and orchestration

In many MNO's environments, lifecycle management, orchestration and assurance are handled through multiple domain-specific solutions based on a mix of standard-based, open-source and proprietary implementations. This often leads to fragmented operational models, inconsistent Day 0/1/2 procedures, duplicated integration efforts, and limited portability across multi-vendor deployments and network domains. Therefore, a lack of a common cross-domain orchestration framework spanning RAN, Core, and other domains, including cloud/platform represent an impediment for MNOs to fully leverage network softwarisation end-to-end.

6.3 Requirements for softwarised networks

6.3.1 Overview

Based on the pain points described in clause 5.2, MNOs require a set of common capabilities and specifications to enable the scalable and consistent softwarisation of their networks. These requirements form the foundation needed by MNOs to advance network softwarisation effectively.

Clause 5.3 summarises key requirements grouped in different areas.

6.3.2 Standardised NF-to-NF communication model

MNOs need consistent guidelines on how NF-to-NF communication should be implemented in cloud-native environments. Today, the use of primary versus secondary interfaces varies widely across vendors and NFs, leading to inconsistent implementation approaches. 3GPP has already standardized NF-to-NF communication, including the definition of service-based interfaces. However, the specifications do not describe how these standardized interfaces should be mapped and implemented in cloud-native environments. As a result, the interpretation of how NF-to-NF communication is realized on cloud-native platforms largely depends on individual NF vendors and platform implementations. This leads to inconsistent assumptions between NF developers and platform operators. Therefore, there is a need to clarify a common interaction model between 3GPP-defined NF communication and cloud-native networking constructs.

6.3.3 Clear responsibility boundaries between NFs and the platform

In many deployments, the demarcation between platform responsibilities (infrastructure layer) and NF responsibilities (application layer) is not clearly defined. While existing standards define functional responsibilities at a high level, ambiguities arise when these responsibilities are applied to CNFs running on Kubernetes-based platforms. For example, node-level customizations in a DPDK-enabled CNF can cause unexpected Kubernetes pod restarts during fault testing.

This illustrates that unless node-level customization responsibilities (including what can be configured via the NF packaging & descriptors (e.g., Cloud Service Archive (CSAR)) or by the MNO) are clearly defined, the operational risk ultimately falls on the MNO. Therefore, the responsibility boundaries described in existing standards should be further clarified with respect to cloud-native deployments.

6.3.4 Common framework for CNF lifecycle operations

Across Day-0 (design and build), Day-1 (deployment), and Day-2 (operations), operators require a unified lifecycle management framework enabling automation, monitoring, and recovery for CNFs. Existing orchestration frameworks are still largely optimised for NFV/VNF models and provide only limited support for CNF-specific operational needs. To address this, standardised APIs and data models for policy-based scaling, health monitoring, and automated healing are needed.

A common framework will facilitate the unification and generalization of LCM and OAM procedures of NFs running on virtualised/cloudified infrastructure and platforms, and facilitate the integration of NFs, regardless their type and technology implementation bases (e.g., VMs, OS containers, etc.), to the rest of MNO’s assets (network and OSS). In fact, as NFs can be handled as "software entities" running on a unified infrastructure and platform, generic IT/Cloud industry practices for provisioning and telemetry/monitoring can be extensively applied in current and future network softwarisation.

6.3.5 Guidelines for performance assurance

When CNFs rely on DPDK, SR-IOV, or other hardware-accelerated methods, performance validation and configuration guidelines should be standardised. Today, validation methods differ significantly across vendor implementations, making it difficult for MNOs to ensure consistent performance. A shared performance assurance framework between the infrastructure layer and the NF layer is required.

6.3.6 Proper stateless design to enable better agility of NFs

In the future, stateless design can become an important enabler for telecom network functions to achieve carrier-grade high availability with elastic scaling and rapid fault recovery. As the evolution of telco cloud infrastructure, the use of cloud-native technologies like container and microservice can better facilitate telecom network functions adopting towards stateless design and implementation.

However, the potential challenges brought by stateless design also need to be taken seriously. They can become a performance bottleneck due to frequent data communication with external storages or components, and state inconsistency due to data operations by multiple instances simultaneously. Therefore, to get the best trade-off between agility and other aspects, which components of a network function and in what granularity to implement stateless design shall be handled properly.

For instance, achieving a stateless design for the UPF is challenging because its processing logic is often tightly integrated with hardware accelerators. Easing such hardware dependencies, e.g., via abstraction layers, is an important step to ensure that state can be effectively separated from processing logic in high-performance NFs.

6.3.7 Stable and long-term support models and interfaces

To minimize the impacts of frequent network and infrastructure upgrades, stable and long-term supported standardized models and interfaces are required. In this regard, long-term support of network services by MNOs and the rapid advancement of underpinning cloud-native technologies require specific considerations to ensure the portability and reusability of the standardized models across different potential underlying solution implementations.

6.3.8 Simplified exposure towards MNO's other OSS systems

While the network disaggregation and softwarization provides greater flexibility to the MNO, it also discloses details into the network deployment not considered in legacy deployments, thus bringing in more complexity. While this is a necessary step for "openness and agility" of the network, network softwarization management and orchestration systems should handle the complexities of the network softwarization and "shield" other MNO's OSS systems from handling such complexity. This approach can further facilitate the integration of network softwarisation with existing or evolved MNO OSSs.

Furthermore, a common cross-domain orchestration and lifecycle management framework aligned across standards and open-source communities would enable MNOs to cope with the fragmented operational models spanning RAN, CN and other network domains. This will ensure to reduce operational silos, improve portability and provide a more practical integration model for multi-vendor cloud-native deployments.

6.3.9 Capabilities for energy efficiency and sustainable operations

Reducing energy consumption is a top priority for MNOs. Softwarised networks require standardized capabilities to dynamically monitor energy metrics at the infrastructure and pod levels, enabling intelligent workload consolidation (e.g., spinning down underutilized nodes) without compromising the strict performance requirements of telecom services.

6.3.10 Open-source based supply chain transparency and security assurance via SBOM

To mitigate software supply chain risks, standardized frameworks must enforce the generation and continuous validation of Software Bill of Materials (SBOM) for all deployed NFs and platform components. Transparent, machine-readable SBOMs that seamlessly integrate into the operator's DevSecOps pipelines are required for automated vulnerability scanning.

6.3.11 In-service upgrade support

Rolling out updates quickly and frequently is a key requirement of softwarised networks to fully exploit the advantages of flexible deployment and development cycles. Most crucial is the rolling update allowing to deploy a new version without downtime. However, alternative deployment types, such as for example canary deployments, are of interest as well since they allow the support of new scenarios like partial roll outs.

6.3.12 GitOps and declarative management

To make full use of the operational advantages offered by softwarised networks, GitOps and declarative management should be adopted as core principles. This approach enables faster

reliable rollouts due to the standardized workflow which includes reviews, approvals, automated testing and deployment strategies. Changes can be applied automatically across systems reducing the operational effort while improving traceability.

Expressing the desired system state in version-controlled repositories also ensures changes remain auditable, traceable, repeatable and easier to roll back. Automated reconciliation additionally prevents configuration drift and minimizes manual intervention. This needs to be supported by both the CSP and the software function vendor.

6.4 Gap analysis vs. existing standardisation and open-source initiatives

6.4.1 Overview

Many of the operator requirements described above are partially addressed by existing standardisation or open-source activities. However, no framework currently provides comprehensive support for CNF-oriented operations. This section analyses the gaps relative to major efforts such as ETSI, CNCF/Kubernetes, O-RAN ALLIANCE, and 3GPP, among others.

6.4.2 Alignment among standards

The ETSI NFV-MANO framework was originally designed around VM-based VNFs, reflecting the architectural assumptions of early NFV deployments. As the industry has shifted toward cloud-native architectures, ETSI NFV has introduced a series of extensions —including container-infrastructure and cluster-management capabilities— to better support CNFs and Kubernetes-based environments. These enhancements broaden the applicability of MANO and provide mechanisms for Day-0 to Day-2 lifecycle operations for CNFs.

Nevertheless, the operational models and tooling for CNF management continue to evolve, and industry practices have not yet reached the same level of maturity or consistency seen with VM-based VNFs. As a result, further refinement and closer alignment across standards bodies and implementations remain necessary to achieve consistent and fully integrated cloud-native lifecycle management within the MANO framework.

In parallel, other standardization bodies are also addressing cloud deployments from different perspectives. While the O-RAN ALLIANCE promotes cloud-ready architectures for the RAN domain, specifications for CNF operations and integration with Kubernetes remain limited. Similarly, 3GPP defines NF-to-NF interfaces and OAM models, but does not define how these functions should interact with cloud-native infrastructure. This results in unclear responsibility boundaries between NF developers and platform operators.

From an MNO perspective, these gaps across multiple standards highlight the need for clearer alignment and well-defined responsibility boundaries among standardization bodies. Without such alignment, MNOs are continuously required to bridge the gaps and fragmentation among standards through vendor-specific implementations and operational workarounds, increasing complexity and operational risk in cloud-native network deployments.

6.4.3 Limitations of the open-source based solution ecosystem

While it is fully recognised the value and merits of the open source in network softwarisation, how to address telecom related requirements in open-source based solutions is still and open issue. For instance, the CNCF ecosystem provides a strong foundation for general-purpose container orchestration, but does not explicitly target telecom workloads requiring deterministic low latency or high throughput. Extensions exist for SR-IOV and DPDK-enabled networking, but there is no standardised configuration model or performance assurance framework. This increases the need for vendor-specific CNI plugins and limits portability.

6.4.4 Lack of an integrated framework

Standardisation bodies and open-source communities operate independently, and no unified reference framework exists for CNF adoption. Standards tend to provide high-level abstract models, while in some cases open-source projects focus on implementation without ensuring alignment with standards. MNOs therefore require a reference integration model that bridges these domains and provides practical implementation guidance.

Existing standards and open-source initiatives provide essential building blocks for network softwarisation but lack a unified framework covering operational automation and NF–platform integration. A cross-industry reference model that harmonises standards and open source is necessary to support reliable CNF deployment at scale.

7 Vision and recommendations for future telecom network softwarisation standardization and open source

7.1 Vision for future telecom network softwarisation

While network softwarisation has evolved substantially since the early days of NFV and SDN, the journey is still underway. Technologies underpinning network softwarisation, including also AI, are evolving rapidly, and as a consequence, there is room for the MNOs to consider, together with network and service requirements, how to further improve the way softwarised telecom networks are designed, developed, built and operated.

In this white paper it is envisioned that telecom network softwarisation can and should further evolve upon following principles:

- **New customer added-value services enabler:** network softwarisation platforms and technologies need to elevate their capabilities to support new kinds of customer services, and not only from a pure network implementation standpoint. Telecom networks need to play a critical role in both economical and societal perspectives (services beyond media and leisure consumption, such as health, care, etc.).
- **Cloud and AI-nativeness of softwarised networks:** it is envisioned that telecom networks will integrate compute, Cloud and AI in a heavily distributed environment. Telecom networks should be capable to connect users, devices, agents and offer connectivity, compute and data services, and fully support the evolution of network applications and of other types of applications. Network connectivity needs to consider more compatibility with cloud-native environments and foster the adoption of network protocols that simplify network connectivity management.

- **Rapid technology evolution:** network softwarisation frameworks need to enable embracing rapid advancements in virtualisation, containerization, and softwarisation technologies in general, aligning to cloud-native principles and enabling true software portability across hardware infrastructures and platforms.
- **Automation at all levels:** network softwarisation frameworks need to embrace "automation", with smart and intelligent orchestration and management, as a mandatory asset. This is critical to address the network disaggregation complexity and simplify network development, construction and operation.
- **Elevating non-functional qualities:** network softwarisation needs to fully support and enable the telecom networks to become more robust and resilient, and high-performance to cope with capacity and quality requirements of future 6G and beyond.
- **Simple network operations:** network softwarisation frameworks need to embrace simplicity via declarative and intent-based management, with highly efficient monitoring capabilities (e.g., stream-based), and simple and replicable code and declared-based expressions.
- **Multi-domain end-to-end management and orchestration:** ensuring network softwarised frameworks unify operations across multi-vendor and multi-domain network environments on the grounds of a unified cloudified and virtualised infrastructure.
- **Sustainable, economical and long-term stable evolution**, ensuring backward compatibility (where needed) and long-term support, upgrade paths and robust version management capabilities.
- **Flexibility and simplicity:** network softwarisation frameworks will become more flexible, modular and loosely coupled, enabling to adapt their setting and usage to the needs by MNOs in terms of target network deployments and integration with other MNO assets.

Figure 9 summarizes the key principles driving the evolution of telecom network softwarisation as described in the preceding listed items.

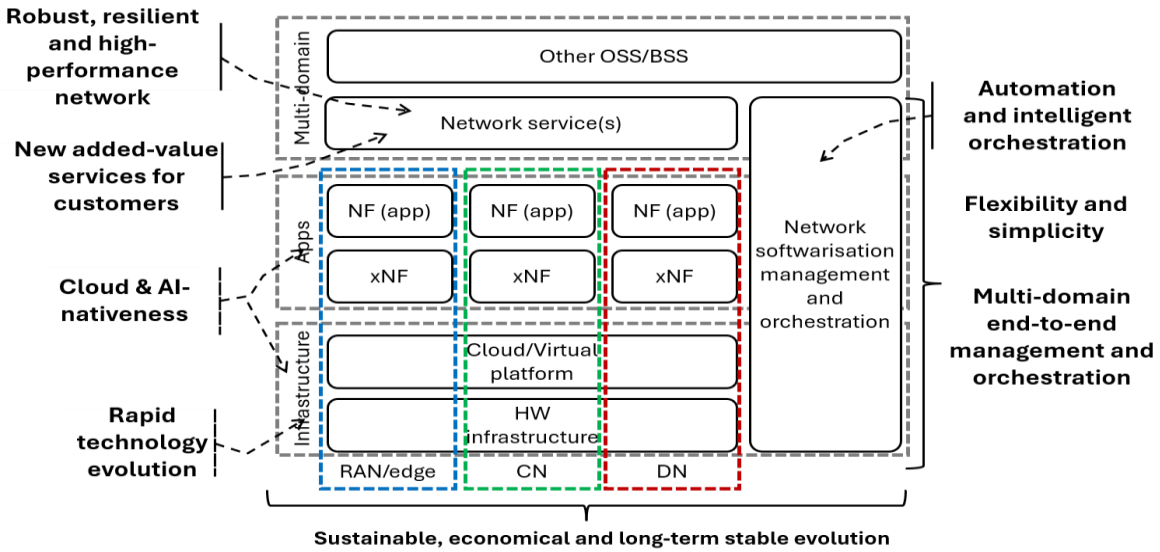

Figure 9: Key principles for network softwarisation evolution

7.2 Recommendations

7.2.1 Introduction

As documented throughout the white paper, there are various aspects to consider when developing, deploying and operating softwarised mobile networks. These include aspects concerning standards and open-source activities, wherein both play an important role in setting up the frameworks upon which MNOs and other telecom industry stakeholders can further develop.

The purpose of the present clause 6 is to summarize the key findings and main recommendations in aspects concerning standardisation and open-source activities, key trends in network softwarisation, network deployment and business roles, and specifically identified technical and operational pain points and challenges. The aim is that these recommendations can guide and direct the industry to take further action to ensure network softwarisation can be fully embraced and its merits uplifted accordingly.

Figure 10 represents high-level recommendations with highlighted requirements described in clause 5.3. Recommendations are further developed in the following clause 6.2.2.

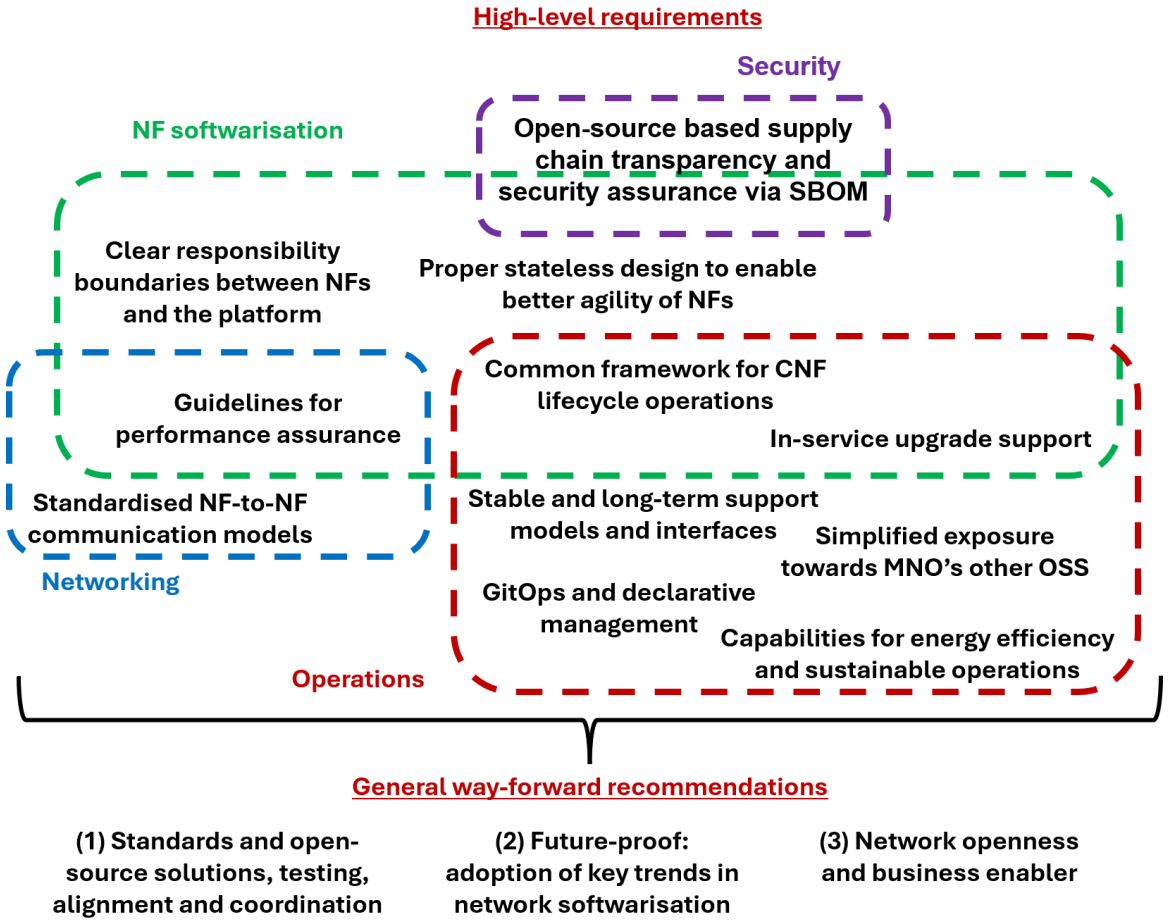

Figure 10: Technical challenges requirements and derived high-level recommendations

7.2.2 Recommendations related to standardisation and open-source activities

As documented in clause 3 of the present white paper, there are various past and ongoing standards and open-source activities related to network softwarisation. As part of these activities, there are several examples of collaboration between open source and standards work in which deliverables and outcomes are leveraged in both ways, from open source to standard, and vice-versa from standard to open source. It is recommended that such collaboration between standards and open-source activities continues and further tightens to leverage and reuse each other organizations' solutions and avoid industry fragmentation and "re-inventing the wheel" syndrome. In fact, a good balance between standards and open source is necessary: standards provide the means for long-term support addressing carrier grade requirements required by MNOs, while open source can be used for faster innovation and leverage of Cloud/IT technologies. In this regard, two points should be further emphasized:

- the need to establish a clear and careful demarcation between underlying general-purpose open-source technologies and higher domain-specific standard abstractions, and
- the importance of open-source communities, and its involved stakeholders, and projects to closely cooperate with standards to avoid fragmentation and address any gaps between telecom-driven requirements and open-source implementations, so as to ensure the open-source technologies can be efficiently used by telecom industry without the need to handle specific telecom extensions.

A careful handling of telecom requirements would promote that standards organizations may adopt open-source solutions as standard profiles, wherever applicable, and develop stage 3 interface and models specifications based on the APIs and schema of such relevant open-source solutions.

From the white paper it is also observed that there are multiple standards activities, some with same scope regarding infrastructure, platform and management and orchestration. It is recommended that standards organizations acknowledge existence and reuse of available specifications, profiling, when necessary, rather than creating new alternative standards. Importantly is how to also close the gaps among standardized frameworks, and with open source, to ensure consistent integration of end-to-end solutions for the MNOs. As documented in the white paper, when it comes to building and operating software networks, many different components from infrastructure, platform, software, etc. need to be considered. In this respect, it is expected that standardization work should be delivered determining the interplay and integration of the various frameworks, standard functions, services and interfaces, and potential de-facto open-source solutions, acknowledging the harmonization between standards and open source.

Finally, a unified approach to testing and validation is necessary to verify the interplay between different domain-specific standard abstractions and underlying open-source solutions. Test suites and validation frameworks developed by standard organizations and open-source communities should be used in combination to validate end-to-end interoperability between different functions and services that lie in various layers. This

approach can help identify crucial gaps on both sides, which can be worked together to realize the overall harmonized framework. Overall, standards can help MNOs efficiently rely on open source for their commercial deployment by considering timelines, stability, telco-grade and developer involvement. This activity should further be reinforced with industry wide testing and plugfest events whereby solutions and products from vendors, standards and open-source projects can be tested and interoperability among different solutions promoted. In this regard, collaboration between standards and open-source communities should be further coordinated establishing a rich (e.g., in number of projects and solutions to test) and continuous (i.e., on a regular basis to ensure advances in standards and new technical features can be handled promptly) plugfest and testing activity flow.

7.2.3 Recommendations related to key trends in network softwarisation

As documented in clause 4.2, several technologies and conceptual trends are already or are expected to become very relevant for future development and operation of softwarised mobile networks. While technologies in network softwarisation evolve very rapidly, telecom industry should adopt concepts and approaches from Cloud/IT and AI domains and make them workable for MNOs and the telecom environment. New virtualization technologies, microservices design, observability frameworks, controller-based orchestration are already prominent in today's network softwarisation solutions. Nonetheless, with the advent of 6G and the possibility of broadening up the kind of services and usages of mobile networks, based on a stronger synergy and compatibility between network, compute and AI, the relevance of new technologies and advances underpinning the mobile networks development and operation should not be underestimated. In this regard, standards should provide the tooling for MNOs to dive in between the operator-grade requirements and the latest technologies, and deliver specifications that easy adoption of "cutting-edge" technology developments, including open-source based, for instance in the form of generic abstractions or as profiling recommendations on how to use the technologies considering MNO requirements.

7.2.4 Recommendations related to mobile network deployment and business roles

While the adoption of NFV and relevant technologies to softwarising mobile networks has not been equal by all MNOs, a trend towards more network softwarisation sophistication is the end-goal for many MNOs. The trend towards cloud-native and AI-native networks is and will dominate the telecom industry in the foreseeable future. Because of this and taking into consideration the different "starting point and state of current softwarised network deployments", it is crucial that telecom industry, on the one hand, provides mature, standard and de-facto solutions towards cloud-native, and on the other hand, enables standards and solutions to consider how MNOs can perform a smooth migration between NFV-centric and open-source/container-centric based deployments.

The deployment of mobile networks in a softwarised form will continue toward 6G. To ensure a broad range of possible integrated solutions, multi-vendor interoperability in highly disaggregated telecom networks is very important. Therefore, it is expected that standards and open-source communities will develop standards and technology ensuring the openness of the whole mobile network system, vertically along the implementation, platform and management stacks, and horizontally, across the whole network. While openness and

disaggregation are important for MNOs, the attack surface expands, hence, it is critically important that vulnerabilities and supply chain attacks in open source are carefully and rapidly addressed and delivered artifacts follow machine-readable SBOM to enable automated vulnerability scanning and avoid increased operational burden of security compliance.

Finally, it is also critically important that standards and de-facto open-source-based solutions enable portability of NFs and other applications/workloads across different infrastructures and platforms and at different locations in a Cloud continuum. This will enable MNOs to continuously deploy softwarised networks using various environments and strategies, e.g., on-premises, public and hybrid Cloud, to broaden the scale, capacity and resiliency of mobile networks.

7.2.5 Recommendations related to technical and operational pain-points, and gaps regarding development of softwarised mobile networks

Clause 5 of the white paper documents several pain-points, challenges and high-level requirements regarding technical and operational aspects in softwarised mobile networks. The documented challenges comprise various characteristics including, NF's design and its softwarisation, connectivity and networking capabilities, security and operational aspects (like product lifecycle, upgrades, among others).

Regarding NF's design and their softwarisation, a couple of challenges refer to the tight-coupling design of NF's architecture and their hardware dependency in data place processing. As a basic principle of network softwarisation, it is critical that standards and de-facto open-source solutions define open models, interfaces that further decouple the hardware and software elements, and set up clear responsibility boundaries between NF's software and the platform & infrastructure on which they run. A unified, common and general-purpose (even if heterogeneous from a hardware perspective) infrastructure and orchestration framework will enable ease of NF's portability. Related to this, mobile networks should become more stateless from a logical network perspective, wherever feasible, while from a network softwarisation perspective, technologies and solutions should support and enable how to efficiently decouple state from NF application's processing. Furthermore, pain-points related to limited NF's design visibility and integration with management stacks recommend that standardized and de-facto models clearly show and describe the boundaries between the different managed objects composing the NFs. In this regard, it is expected that the complexity of disaggregated NF's as softwarised components needs to be managed by the standardized and de-facto management and orchestration frameworks responsible for the software network deployments. This will enable shielding the complexity of such deployments toward higher-level OSSs of the MNOs. Here, the objective is to ensure that such demarcation can facilitate the adoption of network softwarisation and the integration with existing MNO's assets.

In terms of network connectivity, as documented in clauses 5.2.2 and 5.2.3, there are issues between NF-to-NF communication and networking models that are de-facto and commonly used in cloud-native environments. In this regard, there is a strong need to clarify the communication models defined at NF-level, like those concerning the 3GPP mobile network NFs, and their mapping/support by cloud-native networking constructs. A possible way

forward is that telecom networking protocols are specified in a way to make them more suitable to cloud-native connectivity patterns, where feasible. In short, the definition of NF-to-NF communication models and NF-platform responsibility boundaries should be prioritised in future standardisation efforts.

Finally, with regard to operational aspects, while variety of softwarisation models across NFs are present and likely expected to persist for many years, it is important that a generic and common standardized framework is defined able to handle all kinds of xNF lifecycle, provisioning and assurance operations, aiming at breaking deployment and management silos of VNF vs. CNF vs. xNFs. Standards and de-facto solutions should introduce more cloud-native operational needs, while keeping the framework generic to handle different deployment technologies (containers, VMs, and newer ones). The use of GitOps and declarative management can help fully leveraging the advantages offered by softwarised networksk, hence network softwarisation frameworks should consider introducing more declarative management and compatibility with modernized cloud-native operations. Furthermore, while network softwarisation brings to MNOs benefits like more network agility and facilitates faster time-to-market, which are seen very much through the lens of day-0 and day-1 operations, day-2 and product lifecycle support is critically important. To address this, standards (models and interfaces) should provide the means enabling longer-term support of solutions at all layers (NF, platform and infrastructure); this means that higher-level networking and network operations systems should be shield from constantly evolving technologies. Two important aspects need to be reflected: on the one hand, NFs should be able to perform in-service software upgrade, while standardized models and de-facto solutions ensure that upgrades can be coordinated efficiently between network (NF) and platform & infrastructure. On the other hand, considering the importance of open-source based implementations, open-source communities are very much encouraged to develop long-term support versions and provide clear migration paths of the implemented software solutions to help MNOs achieve sustainable, stable and reliable network operations and services for longer-term, als in consideration of making softwarised networks more energy efficient.

8 Conclusion

Network softwarisation, including both virtualisation and cloudification approaches, has positively impacted in how MNOs can deploy and operate mobile networks. The horizontal network and vertical stack disaggregation accommodated by network softwarisation frameworks and technologies have steered toward more openness. The merits of softwarisation applied to mobile networks are extensive, like enhanced lifecycle and service agility, resiliency and scalability.

Numerous standards organizations and open-source communities contribute to providing specifications for interfaces, models and architecture, as well as actual open-source based implementations. Clause 3 of the white paper provides an overview and state of the art of major standards and open-source activities related to network softwarisation. All these activities expose and nurture a rich ecosystem that MNOs can leverage, yet, not in all cases efficiently. While a rich technology ecosystem is key for the success of network softwarisation, the potential for fragmentation and lack of clear and easy end-to-end systems integration are risks that MNOs care about. A coordinated action among standards and open

source is becoming more important as MNOs go farther in their adoption of network softwarisation.

The use and maturity of network softwarisation is reported in clause 4, while aspects of how network softwarisation is opening up the options for MNOs to build and operate their networks are also described. Use of public Cloud resources is not anymore strange to MNOs as an alternative or additional capacity to on-prem. Furthermore, as technologies and software paradigms are in continuous evolution and progress, several are the trends that are expected to play a key role in how mobile networks based on sofwarisation will evolve. While containerization and microservices have been a key driver in recent mobile network developments (e.g., 5G), many more technologies are expected to underpin future network generations like 6G.

Finally, as reported in clauses 5 and 6, several challenges and pain-points are still present in nowadays network softwarisation, and in particular for telecom mobile networks. Clause 6 documents, on the one hand, a high-level vision for the future and needs of network softwarisation, and on the other hand, several recommendations covering aspects of standardisation and open-source activities, key trends in network softwarisation, network deployment and business roles, and specifically identified technical and operational pain points and challenges.

Telecom industry and its stakeholders, but particularly standardization and open-source communities are called for action to address the identified gaps and requirements. A consistent and aligned approach across the whole industry will uplift the merits of network softwarisation and create a stronger and longer-sustainable marketplace for all industry stakeholders and across all layers, from infrastructure, through platform, software, applications and network services.

Annex A Document Management

A.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	22/09/2026	CR1001 NG.157 White Paper First version	ISAG	Joan Triay (NTT DOCOMO)

A.2 Other Information

Type	Description
Document Owner	NG
Editor / Company	Joan Triay, NTT Docomo

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com

Your comments or suggestions & questions are always welcome.